CISQ

Consortium for Information & Software Quality ™

State of the Industry Report on Software Quality Analysis

Introducing the State of the Industry Report

The Consortium for Information & Software Quality™ (CISQ™) launched its "State of the Industry" survey, the first comprehensive study of software quality analysis that covered tool vendors, system integrators, managers, and engineers at end-user organizations. The survey was open from July 2019 to January 2020.

The impetus for this study was the alarming increase in software quality-related incidents and CISQ member concerns that organizations are not getting the basics right. We wanted to see how the move to Agile development and DevOps is changing not only software quality practices, but developer attitudes and behavior when it comes to code quality. It is also important to see how software quality standards are being utilized by system integrator and end-user organizations; which standards are being used, which sectors are driving adoption, and how organizations are deriving value from software quality standards.

Methodology

The findings in the *State of the Industry Report on Software Quality Analysis* come from 82 responses to an online survey, 155 telephone conversations with enterprise IT leaders, their teams, and IT vendor managers, discussions at CISQ-hosted workshops, and LinkedIn discussion forums. This report includes survey results, observations, and recommendations.

The report is split into three sections:

ENGINEERING

SYSTEM INTEGRATORS

VENDOR MANAGEMENT

The report can be downloaded from the CISQ website at: **www.it-cisq.org/state-of-the-industry.htm**.

What Do Developers Think of Software Quality Analysis (SQA)?

INTRODUCTION

The move to Agile development and DevOps and the increasing velocity that teams are experiencing is unprecedented. Teams are delivering software at a rate they never have before. At the same time, we are dealing with much higher levels of risk and vulnerability.

The issues of poor quality software and technical debt have been with us since the dawn of IT. It would appear we have reached a crisis point where we need to be much more serious about how we address these issues and we need to start thinking about how we make developers engineers.

This means taking individuals and teams that might be very good at cutting code and looking at the other aspects of engineering including quality assurance, security, robustness, and the long-term viability of the solutions they are developing. With the move towards SaaS and cloud-everything, the enterprise and engineering teams may feel this is not a problem that affects them, but it does. SaaS solutions are built by engineers. If those SaaS solutions have weaknesses, this affects thousands, if not millions, of individuals.

With CISQ's *State of the Industry Report on Software Quality Analysis*, we wanted to test the hypothesis that reliability, security, performance efficiency, and maintainability, because they are often called "non-functional requirements," are treated secondary to customer-facing features by product owners, managers, and teams.

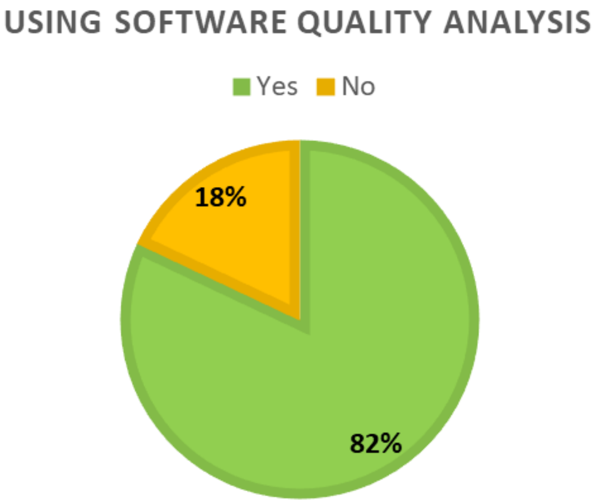
Let's jump into the survey results.

SO, ARE DEVELOPERS USING SQA?

Question: Do you currently use software quality analysis?

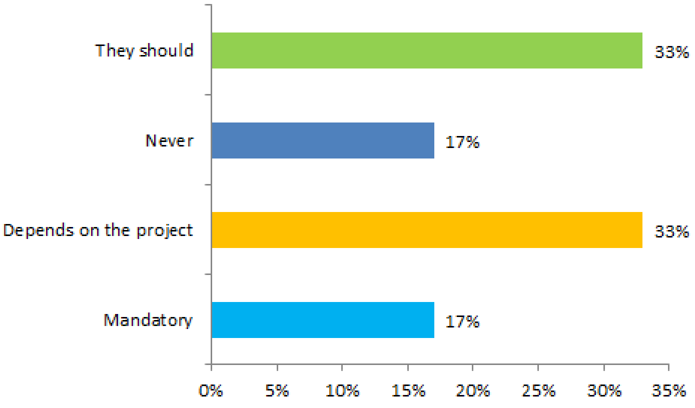
Results: The majority of developers, 82%, report using software quality analysis. Digging deeper, 33% report they always use static code analysis compared to 17% who always use dynamic code analysis. 32% of developers report using static and dynamic code analysis frequently, i.e., on a daily or weekly basis.

Observations: Anecdotally, there has been a belief that developers have been reluctant to use software quality analysis in any form, static or dynamic. However, the results we have collected would appear to show this is far from the case with over 82% of developers claiming to use SQA of some form. We believe the increased focus on continuous integration (CI) and continuous delivery (CD) within the DevOps community and the availability of open source tools for SQA is driving this.

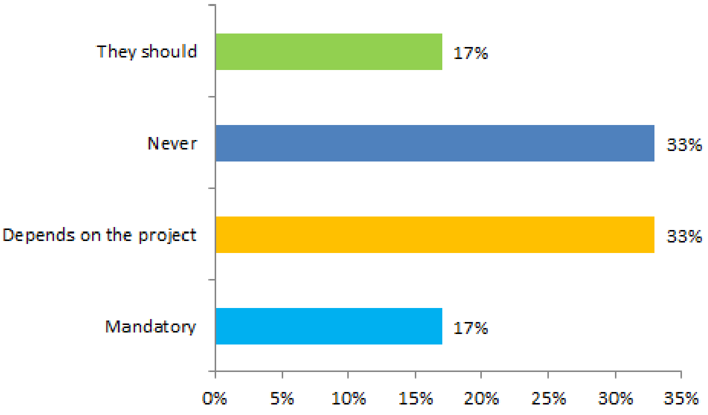


WHO IS TELLING DEVELOPERS TO USE SQA?

Question: Does your internal regulatory function or QA function mandate the use of software quality analysis with your development teams?



Regulatory Function



QA Function

Results: 17% of developers report the use of SQA is mandatory per the internal regulatory function, a third say it is project-dependent, and another third say the regulatory function should mandate SQA. This data suggests there is greater opportunity and openness between engineering and governance, risk and compliance (GRC) to reduce software risk, which has not been fully exploited. The quality assurance function mirrors regulatory policy, with 17% of developers reporting it is mandatory and 33% project-dependent. However, a third of say the quality assurance function never mandates the use of code analysis.

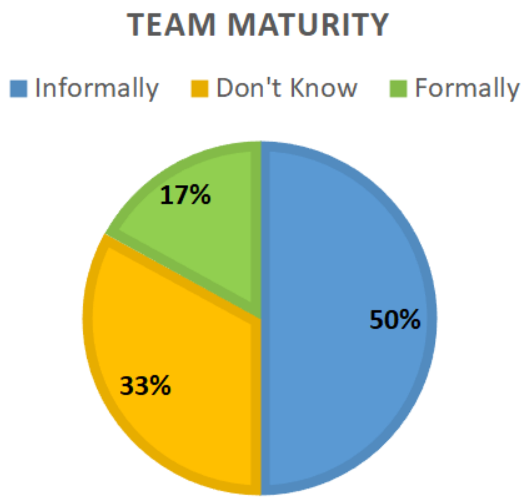
Observations: It is clear that both the internal regulatory function and, somewhat surprisingly, the quality assurance function are not mandating the use of code quality tools. It would appear the vast majority of developers are electing to use SQA on their own volition. There is still a project-by-project approach which appears very dominant and one could question whether this is advisable given the level of IT and security risk we face today.

ARE TEAMS EARNING AUTONOMY WITH GOOD CODING PRACTICE?

Question: Is the level of autonomy your team is given linked to the level of maturity the team has with software quality analysis and managing technical debt?

Results: 50% of developers report the level of autonomy the team is granted is informally linked to their use of software quality analysis. 17% report that autonomy is formally linked to SQA. 33% of developers are unaware of any formal or informal relationship between code analysis and team autonomy.

Observations: Agile and DevOps is based on an autonomous organizational structure with self-directed teams. However, very few organizations appear to have made autonomous units earn their autonomy by being aligned with correct behaviors and best practices. Only 17% of developers report their teams are formally assessed regarding code structural quality.



We find this somewhat surprising given how prevalent the issues of technical debt are in most organizations. Autonomy should be linked to best practices and behaviors regarding software quality.

> *"In the age of DevOps and release on demand, 'built-in-quality' isn't simply a saying, it is essential for survival."*
>
> *- Dean Leffingwell, Founder, Scaled Agile Framework (SAFe)*

WHAT STANDARDS ARE DEVELOPERS USING FOR SQA?

Question: The following standards have been identified as being related to software quality analysis and code vulnerability. How frequently do you use these standards?

Results: The most "frequently" used standards include: OWASP Top 10, ISO 25000, US CERT
Standards with "occasional" use include: MISRA, MITRE CWE, SANS/CWE Top 25, OMG/CISQ

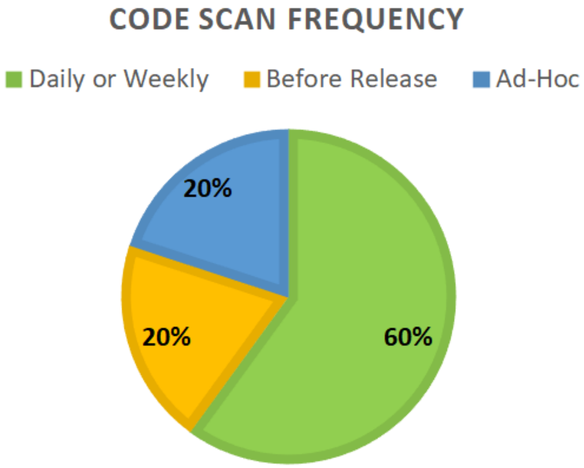
Observations: Not surprisingly, given its prevalence within the industry, the OWASP Top 10 is one of the most commonly cited standards. We should be mindful there is a difference between being aware of and referencing a standard and developing code compliant to that standard. MISRA is an example of something we expect to see more of in the future – industry-specific software quality standards. We predict cyber-physical devices and IoT will increase the use of domain-specific software quality standards.

HOW OFTEN ARE DEVELOPERS DOING IT?

Question: Which is the more common scenario regarding your frequency of software quality analysis - is it run daily or weekly, before release, or ad-hoc? It can be the whole or part of the code base.

Results: 60% of developers report code is scanned on a daily or weekly basis, 20% report code is scanned before release at a quality gate, and 20% report the use of code analysis is reactive and undertaken on an ad-hoc basis when there is an issue.

Observations: Over half of developers report using SQA on a daily or weekly basis. This is to be expected given our belief that DevOps and CI/CD is driving greater adoption of SQA. We should take care not to over-emphasize the frequency of SQA as it is the results of the scanned code and subsequent refactoring that is most important. There is still a high proportion of teams where SQA is not undertaken on a continuous basis. SQA should be integrated into the DevOps toolchain.

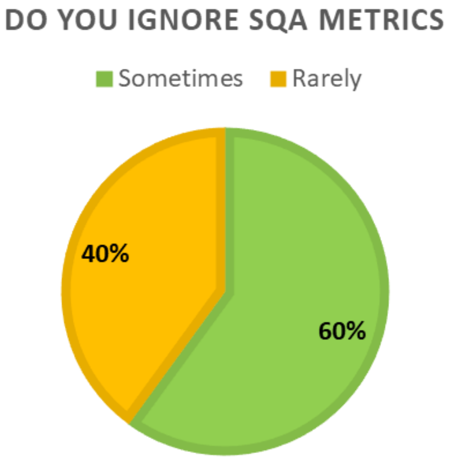


ARE DEVELOPERS IGNORING SQA METRICS?

Question: How often do you ignore software quality analysis metrics?

Results: 60% of developers report they "sometimes" ignore SQA results and 40% report they "rarely" do.

Observations: Although SQA tools have been in existence for some time, they are not perfect, and it should be no surprise that they occasionally provide false positives, i.e., the indication of a known vulnerability or code weakness that in fact does not exist. We are not surprised given the above statement that developers sometimes choose to ignore the results. We have reason to hope, however, that developers are spending time refactoring code as 40% only rarely ignore the scanner.



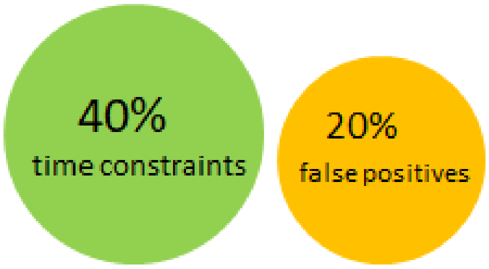
Development teams that frequently get false positives should invest in configuring the tooling to reduce the number of false positives. In the case of the 60% that "sometimes" ignore the SQA reports, we believe it is because they are using un-configured tools out of the box. It is important that DevOps teams are supported by mature SQA tooling that adheres to mature software quality standards to minimize the issue of false positives and false negatives.

WHY DO DEVELOPERS IGNORE SQA METRICS?

Question: What is the most common reason you ignore software quality analysis metrics?

Results: The most common reason developers ignore SQA metrics is because of time constraints (40% of responses), 20% report false positives, and the remainder marked "not relevant" or "other."

Observations: One fifth of developers ignoring SQA metrics because of false positives is a sign of increasing maturity of the tools, as we expected this figure to be higher. It should be of concern to the business and application managers that 40% of developers are ignoring results because of time constraints. This runs counter to the Agile values of delivering value to the customer and has its roots in the poor understanding of the impact of non-functional requirements by product owners and product managers.



It is not surprising that in our conversations with developers they often feel SQA is a wasted effort if they are receiving mixed messages from the business and its proxies. Software quality needs a champion in the enterprise and it should be a business-led issue.

WHO IS BENEFITING FROM SQA METRICS?

Question: Which stakeholders get the most value from the use of software quality analysis? Rank on a scale of 1 to 3, where 1 = not at all, 2 = somewhat, and 3 = considerably.

Results: The top three stakeholders getting the most value from SQA are in this order: 1) QA/Testers, 2) Developers, and 3) Operations.

Observations: An interesting and somewhat surprising result was how few developers feel SQA is of direct benefit to the end customer or business sponsor. Given our earlier observation that 40% of developers ignore SQA because of time pressure, one can hardly blame them if they are picking up on a message from the business that it is not important to the customer.

Stakeholder	**Ranking**
QA/Testers	2.80
Developers	2.60
Operations	2.40
Audit Function	2.20
Management	2.20
Architects	2.00
Business Sponsor	1.60

With the majority believing that QA/Testers are the major beneficiary of SQA, this indicates there is still an us-and-them attitude with developers not fully owning the quality of their code. "I cut code, you test." This attitude runs counter to the best practices of Agile and DevOps. Developers should take greater ownership of the quality of their code.

WHAT ARE THE BENEFITS OF SQA METRICS?

Question: Which code quality areas do you feel gain the most from the use of SQA? Rank on a scale of 1 to 5, where 1 = not at all, 3 = moderately, and 5 = considerably.

Results: The top three code quality areas gaining the most from code analysis are in this order: 1) Reliability, 2) Maintainability, and 3) Performance Efficiency.

Observations: It is clear that developers understand the relationship between SQA and the classic non-functional areas. This gives us concern, going back to the question on page 6, that SQA results are ignored by the teams.

Quality characteristic	**Ranking**
Reliability	4.80
Maintainability	4.60
Performance Efficiency	4.40
Security	4.20
Portability	3.8

It also infers an interesting observation. If developers feel SQA does not add strong value to the end customer as we saw in the question above, then do they not feel that reliability and performance are of value to the customer? These somewhat contradictory results bring us back to our old friend, non-functional requirements, and the conscious or subconscious treatment of NFRs as second-class citizens.

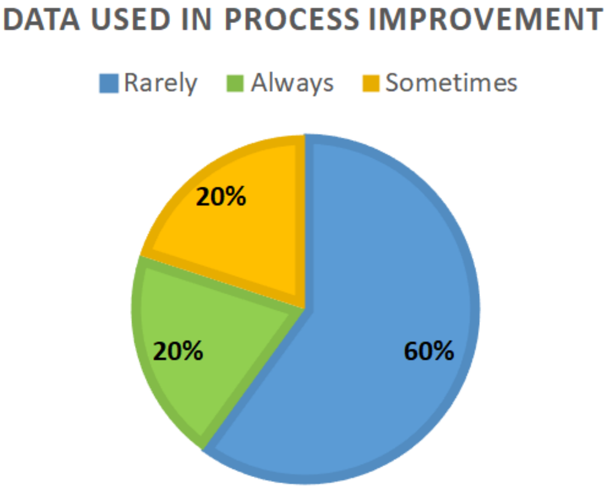
Not surprisingly, reliability and security are ranked highly in terms of benefits of SQA.

ARE TEAMS ACTING ON SQA FOR IMPROVEMENT?

Question: Does your team use software analysis data for process improvement, for example, at retrospectives?

Results: 60% of developers say their teams "rarely" use software analysis data in retrospectives or for process improvement, 20% "sometimes" do, and 20% "always" do.

Observations: It is somewhat surprising that only one fifth of developers work in teams where SQA results are always used as part of process improvement and retrospectives. SQA is an indicator of individual and team maturity regarding software development and directly associated with supporting practices and roles. We would expect this figure to be higher. Our recommendation is for SQA to be used not only in retrospectives, but as part of an individual's cross-skilling process to help developers improve their technical skills and code architecture.

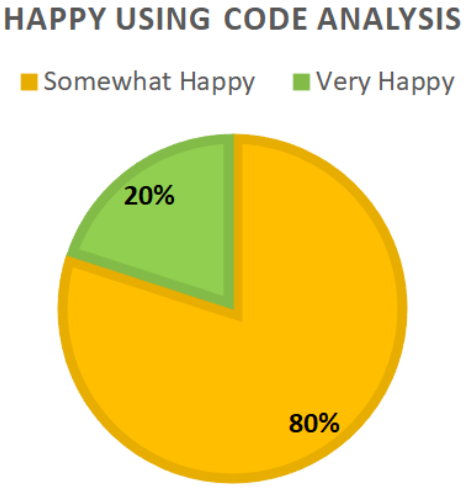


DO TEAMS LIKE USING SQA TOOLS?

Question: In general, are you happy to use software quality analysis tools?

Results: 80% of developers report they are "somewhat happy" using code analysis tools and 20% report "very happy." No respondents report being "unhappy."

Observations: Although none of the survey or interview respondents said outright they do not like SQA tools, it is our impression there are individuals that are not happy with the use of tools but choose not to speak out. As it is, the majority (80%) say they are only "somewhat happy." This could be due to the lack of calibration and integration of many of the tools, which results in manual processes and false positives. Also, generally speaking, developers may not like having a machine (or anyone, for that matter) pointing out things they have done incorrectly.



We still have some ways to go before SQA is seen as important as continuous integration (CI) and test-driven development (TDD), but the conversation has to start with the business and a movement away from our ill-named non-functional requirements. If reliability, performance efficiency, security, and maintainability are tagged as non-functional, they will continue to come in second place to customer features.

Recommendations for Engineering

CONCLUSION

Time pressure on developers, product owners, and product managers adds to the negative behavior that we see around software quality analysis and non-functional requirements. We should be mindful of the fact that developers ignore SQA results and are not particularly happy about using the tools. It is easy to blame the developers. We think it is a reflection of the environment they are in. Our recommendation is to ban the phrase non-functional requirements (NFRs). Until we do this, NFRs will be of secondary concern. It is clear that the developers are receiving mixed messages. Although a high percentage of developers use SQA, the percentage that use it proactively for process improvement and act on the findings is lower than we would like.

MANAGEMENT TEAM

Application managers and scrum managers should be mindful of their behavior and attitude towards non-functional requirements. They should proactively work with product owners or project managers to ensure NFRs are given adequate attention. We have found that having this conversation with the product management team when it is only focused on issues of technical debt is not constructive. This discussion has to be held in the context of business outcomes and risk.

The management team should ensure the correct use of SQA. For example, teams should be tuning the SQA tooling to reduce false positives. They should also ensure SQA tools are integrated into the toolchain. We should recognize this might mean a restructuring of the QA and testing functions to allow more quality testing related to NFRs to take place within the teams.

It is best practice that the level of autonomy granted to each team be linked clearly and consistently to relevant quantitative and qualitative team key performance indicators (KPIs) or objective and key results (OKRs), specifically, the team's ability to deliver high quality code with low levels of technical debt and with high levels of maintainability, reliability, and performance. Obviously, code security is another key indicator of team maturity. Those teams that do not show consistent best practices in these areas will be constrained regarding the frequency of release, their ability to sign-off on changes without approval, and frequency of code reviews.

Recognizing we are in a world where organizations are allowing development teams greater autonomy, we should be mindful of the need for enterprise-wide standards. During this study, we found those teams that have a named individual responsible for the standards the team will use have lower incidents of software defects and less technical debt and more consistent tooling, i.e., less false positives. Therefore, our recommendation is for every team to have a quality champion who promotes and supports the use of related standards, being mindful this individual should not take on the responsibility of quality, they are just the champion within the team. We have found communities of practice (COPs) to be a useful tool for developer-led adoption of standards.

From a development culture perspective, and in line with Agile and DevOps, we need to change the behavior and actions of the teams from one focused on quality assurance, to one focused on quality engineering. This goes beyond test-driven development or behavior-driven development practices towards a fundamental shift in lean ethos for engineering quality in each stage.

Recommendations for Engineering

DEVELOPMENT TEAM

It heartening to see the level of SQA use higher than we expected, however, it's clear that we have some ways to go before SQA is used effectively. Our first recommendation is for teams to stop using the phrase non-functional requirements (NFRs) and to educate their business analysts and product owners on the importance of NFRs to the end customer.

We recommend that SQA tools are treated in the same way as CI/CD tooling and integrated into the toolchain with a high level of automation. For this to be successful and not troublesome to the developers, the SQA tools will have to be tuned to the coding environment and relevant coding standards to reduce the number of false positives.

Teams should adopt a data-driven approach to prioritize refactoring tasks. SQA tools that are tuned to the team's code base and coding styles can aid in prioritization and help build the case with the product owner or product manager for refactoring time. To set prioritization objectives and to reduce internal debate, we recommend the consistent use of code quality standards with SQA tools. Furthermore, teams should adopt standards that can be automated and do not require manual intervention.

Given the low levels of SQA metrics being used for process improvement, we recommend all retrospectives include the review of SQA dashboards, if even a quick review to stave off complacency. If the teams are working within organizations where autonomy has to be earned by tangible metrics, this becomes even more important.

Finally, teams need to build a convincing business case for the use of SQA, especially if they work in environments where NFRs are treated secondary to customer-facing features. We have found the best approach is to point out the causal relationship between poor quality and customer experience, and second, the positive effect of using a standards-based approach to SQA to reduce technical refactoring, which today is typically 10-15% of time spent per sprint.

> *"Non-functional requirements are under-emphasized in project management and a major source of project overruns and failures. Non-functional requirements are success-critical and more emphasis must be placed on preparing for maintainability, which is critical to total cost of ownership."*
>
> *- Dr. Barry Boehm, Chief Scientist, SERC, TRW Professor of Software Engineering and Director, Center for Software Engineering, University of Southern California*

What are System Integrators Saying About SQA?

INTRODUCTION

Code architecture and its structural quality is most effectively coded into a system as it is developed, not reworked at a later date, months or even years, after development. This is true of all systems, but especially so when a system is developed under contact by a third party.

Late discovery of poor structural quality, often because of an incident, can result in costly and time-consuming repudiations, soured relationships, and even litigation. Indeed, the aforementioned situation is one of the reasons Systems Integrators (SIs) have sponsored industry bodies such as CISQ in the hope that SQA standards can improve quality and the relationship between customers and SIs.

However, the desire for SQA standards and their development is moot if standards are not being actively used within the customer-SI relationship. Therefore, this section of the report examines to what extent SQA and related standards have been adopted and enlisted by the SI and enterprise.

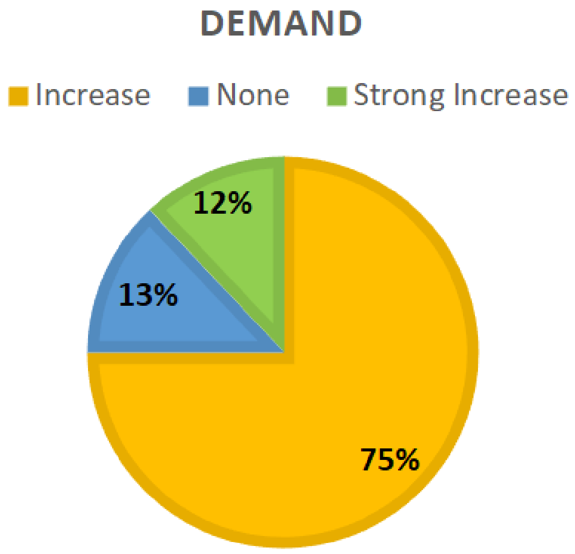
ARE CUSTOMERS ASKING FOR GREATER SOFTWARE STRUCTURAL QUALITY?

Question: Are you seeing an increased demand for software quality analysis by customers wishing to control and reduce technical debt?

Results: 75% of SI respondents report they see an increase in customers wishing to control and reduce technical debt through SQA. 12% of respondents report a strong increase in demand.

Observations: We are not surprised that 75% of SIs are seeing an increase in demand for SQA and we believe this is due to greater awareness within IT and the business of the benefits of SQA. It is also driven by auditors and regulators who are under pressure to reduce risk and incidents.

This is also an indication of how poorly SQA has been controlled or addressed in the past. If companies had been paying attention to SQA, we would not see such a large increase.



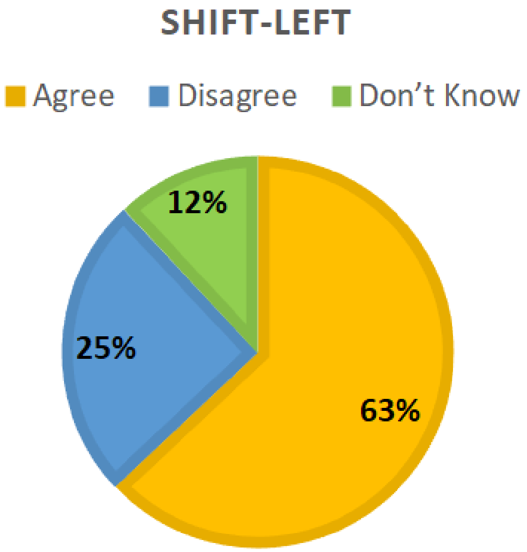
DO CUSTOMERS WANT INFORMATION EARLIER ON SQA?

Question: As part of the "shift-left" trend, are you seeing more customers use software quality analysis earlier in the development process?

Results: 63% of respondents agree their customers are using SQA earlier in the software development process and 25% do not agree.

Observations: Due to greater awareness of the risk of structural quality and technical debt, we are seeing the use of SQA tools and practices earlier in the software development lifecycle. This is a side effect of Agile development and DevOps where SQA is used in line with TDD, i.e., earlier in the software development process and on an incremental basis.

It is worth noting it can be be more difficult to "shift-left" in an outsourced mode of work compared to in-house development.



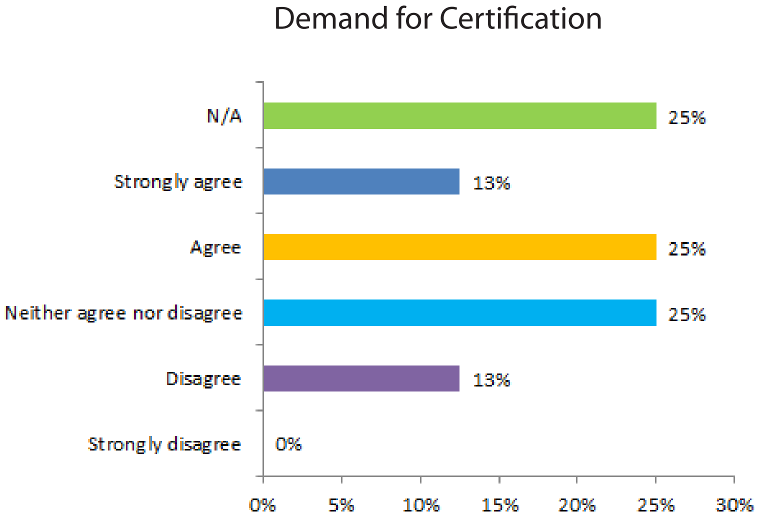
DO CUSTOMERS WANT SOFTWARE STRUCTURAL QUALITY CERTIFICATION?

Question: Are you seeing increased demand for software quality certification services?

Results: Only 12% of rspondents "strongly agree" there is increased demand for software quality certification and 25% "agree" that they see some increased demand.

Observations: In general, we are seeing an increase in customers requesting SQA in their contracts with suppliers, however, the desire for formal certification normally lags the adoption of industry standards by as much as three years.

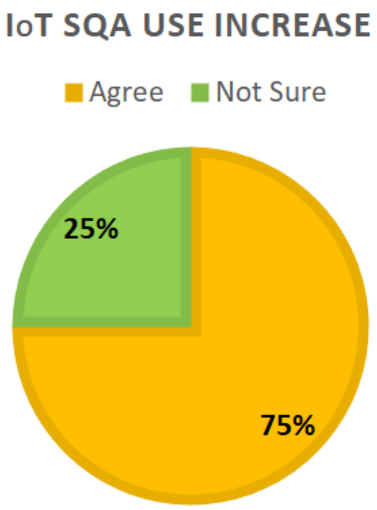
When there is greater acceptance of relevant standards, we will see more requests for certification. The audit community's awareness of standards such as CISQ should drive demand for software quality certification over the next two years.



WHAT EFFECT ARE CYBER-PHYSICAL SYSTEMS AND IOT HAVING ON SQA?

Question: Are you seeing increased demand for software quality analysis in support of Cyber-Physical Systems and IoT?

Results: 75% of respondents agree they see increased demand for software quality analysis within the cyber-physical and IoT domain.



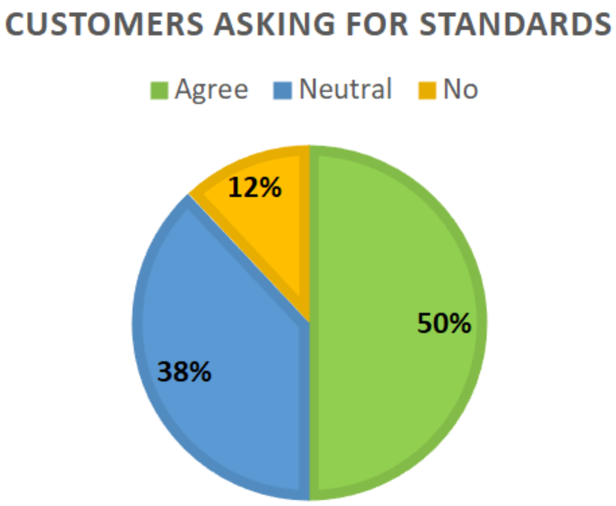
Observations: The figure of 75% is not surprising given the increase in cyber-physical devices and IoT. This stat should be reassuring considering the impact of cyber-physical systems on the physical world. However, we should not be complacent. 25% are "not sure" regarding this question and given the criticality of cyber-physical systems, even 5% is too much.

We believe performance, security, and maintainability are given a higher priority in the cyber-physical world compared to enterprise systems where they are commonly tagged as non-functional requirements.

ARE CUSTOMERS ASKING SIS TO USE SQA STANDARDS?

Question: Are you seeing increased demand for software quality standards support from your customers and prospects?

Results: 50% of respondents report they see increased demand for SQA standards from customers and prospects, 38% of respondents are neutral, and 12% say no.



Observations: It is encouraging to see that half of SI respondents see more demand for the use of SQA standards from their customers. Generally, we have seen a lack of awareness of SQA standards. Customers are indeed requesting a greater focus on SQA (75% increase, see page 10) but they are not always doing so formally tied to a standard.

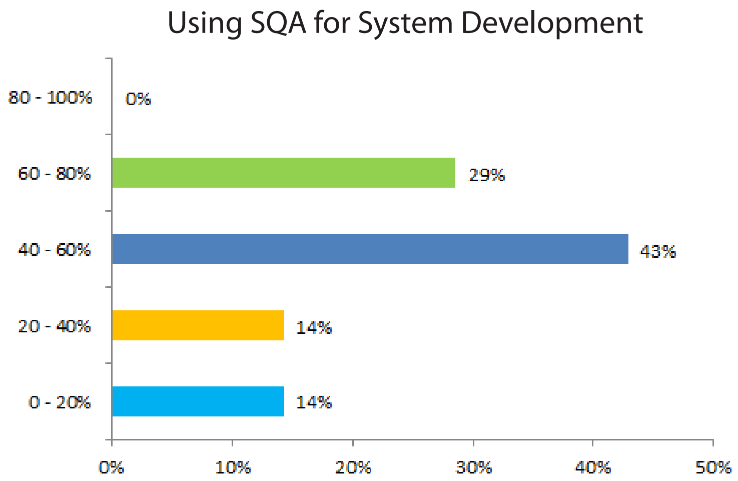
ARE SIS USING SQA WITH THEIR CUSTOMERS?

Question: What percentage of the systems your organization was contracted to develop or maintain in the last 2 years used software quality analysis?

Results: 43% of respondents report they used SQA in 40-60% of the systems they developed over the last 2 years, and 29% report 60-80% of systems.

Observations: The result is effectively showing the majority of systems developed or maintained by SIs use SQA. No respondent replied 100%, which should be of concern to the industry. More worryingly, 14% of respondents say less than 20% of their projects in the last 2 years used SQA in any meaningful way.

This may reflect the experience of the engineering teams where SQA may not be mandatory but project-dependent in these cases as we discussed on page 3.



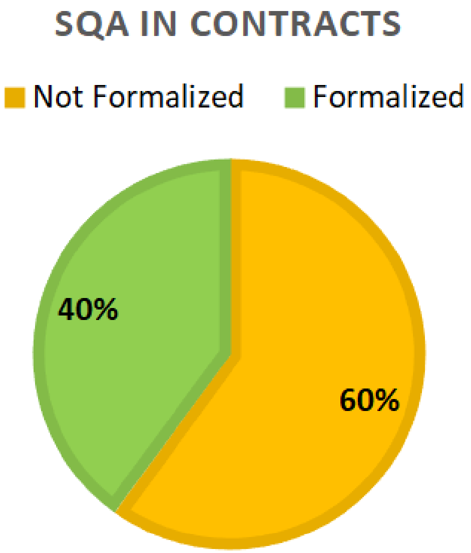
ARE SI CUSTOMERS FORMALIZING SQA IN CONTRACTS?

Question: What percentage of your customers have formalized software quality analysis measures into their contracts?

Results: Up to 40% of contracts “formally” cite SQA measurement and related KPIs.

Observations: The majority of enterprise customers do not formally contract for software quality and the reduction of technical debt through software quality analysis measures in agreements with suppliers.

However, digging deeper, 57% of SIs report they use SQA even when the customer does not request or mandate it in the contract. Still, 29% report “no” or “occasionally,” so it’s clear this is somewhat arbitrary.



WHICH STANDARDS ARE SI CUSTOMERS ASKING FOR?

Question: The following standards have been identified as being related to software quality analysis and code vulnerability. How frequently do you see these standards referenced by customers and prospects, if applicable? Rank of a scale of 1-3, where 1 = not at all, 2 = occasionally, 3 = frequently.

Results: SIs report the most frequently requested standards are: MITRE CWE, OMG/CISQ, SANS/CWE Top 25, and OWASP Top 10.

Observations: It is important to remember the context of this question, as it is only relevant in the 40% of cases where a customer has asked the SI to follow a standard contractually. It is also important to remember the standards are not mutually exclusive. For example, the OMG/CISQ standards are comprised of MITRE CWEs, as is the SANS/CVWE Top 25.

Standard	Ranking
MITRE CWE	2.29
OMG/CISQ	2.14
SANS/CWE Top 25	2.14
OWASP Top 10	1.86
ISO 25000	1.71
US CERT	1.57
MISRA	1.43

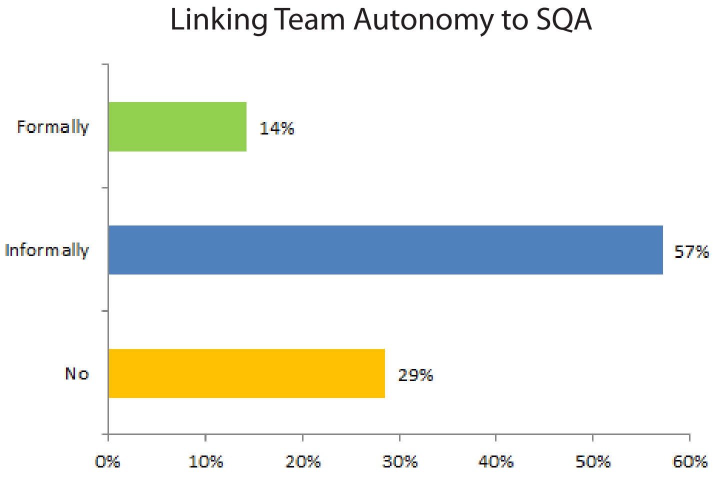
TEAM AUTONOMY AND SQA?

Question: Have you linked the level of autonomy a team is given to the level of maturity the team has with software quality analysis and managing technical debt?

Results: 14% of SIs have "formally" linked SQA to team autonomy and 57% have done so "informally."

Observations: This finding correlates with the engineering section of this report in terms of the level of autonomy granted to the teams and the use of SQA and the management of technical debt.

29% of SI respondents appear to have not considered the concept. We encourage SIs to leverage this practice with their teams to improve maturity.



Recommendations for SIs

CONCLUSION

Part of the impetus for this study was CISQ member concerns that SIs are not taking advantage of the opportunity that a standards-based approach to SQA can afford. As more SIs come under pressure from the use of SaaS-based models and packages, they need to use every weapon in their arsenal to differentiate themselves and attract customers.

SIs that have a proven track record in not only Agile and DevOps-based delivery, but also in producing secure and high quality software, have a unique advantage among their peers. SIs are not leveraging quality standards to their maximum in contracting in such a way that they safeguard themselves. A standards-based approach removes the subjective nature of many of the friction points SIs encounter with customers regarding quality. It is also an opportunity to reduce the burden of GRC and auditing for both the end customer and the SI as the auditors can audit to an agreed upon standard.

It is clear from the report there are a number of opportunities for improvement to increase the overall maturity of both customers and SIs when it comes to software quality and security. Chief amongst these is an educational exercise with potential customers and current customers on the benefits of contracting for quality against an agreed upon standard. SIs should not only be working with the vendor management community of their customers, but also the relevant business units with a strong focus on risk mitigation.

Although the figures are encouraging, it is clear we still have some ways to go internally with the SIs. There needs to be a stronger focus on SQA tool integration and greater alignment of team autonomy and evidence-based maturity of said teams. SIs should ensure team autonomy is linked to their use of SQA and the team's ability to manage technical debt.

The SIs should adopt standards that have a high level of maturity and depth. We recommend the adoption of SQA champions who can work with the delivery units, be it squads or agile release trains, to help them identify and implement the best SQA standards for the products they are developing.

SIs should spearhead the cultural change that we need within IT and the digital business when it comes to making NFRs first-class artifacts.

What are Vendor Managers Doing to Reduce the Risk of Poor Code Quality?

INTRODUCTION

The role of vendor management is critical given the vast majority of enterprise software is delivered as SaaS or developed by system integrators. Vendor management is ideally placed to ensure that software structural quality is considered right at the start of the contracting process and not as an afterthought.

Contracting against established software quality standards such as CISQ allows the customer to specify the level of structural quality in a quantifiable manner within the formal contract document. It removes ambiguity between the customer and SI as to what is expected regarding structural quality and how delivered code will be assessed by the customer.

However, it has been our observation that very few vendor management teams formally contract against software quality standards. In part, this is due to the lack of visibility of the standards, but we believe the major issue is governance. Whose job is it to specify which standards to contract against? Many vendor managers are hesitant to specify IT-related standards as they feel they are overstepping their remit with the IT team.

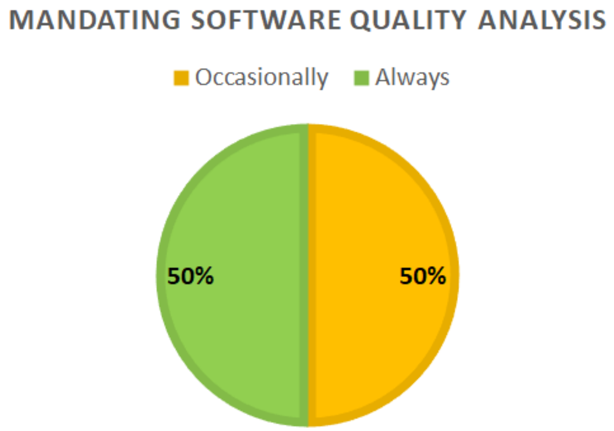
The emphasis on digital transformation has given the vendor management community an opportunity to make a real difference regarding software quality, risk, and customer experience. However, it takes courage to step into a space that should have been filled by IT but in many cases has not been.

ARE VENDOR MANAGERS ASKING FOR SQA WITH SUPPLIERS?

Question: Do you mandate the use of software quality analysis with your suppliers?

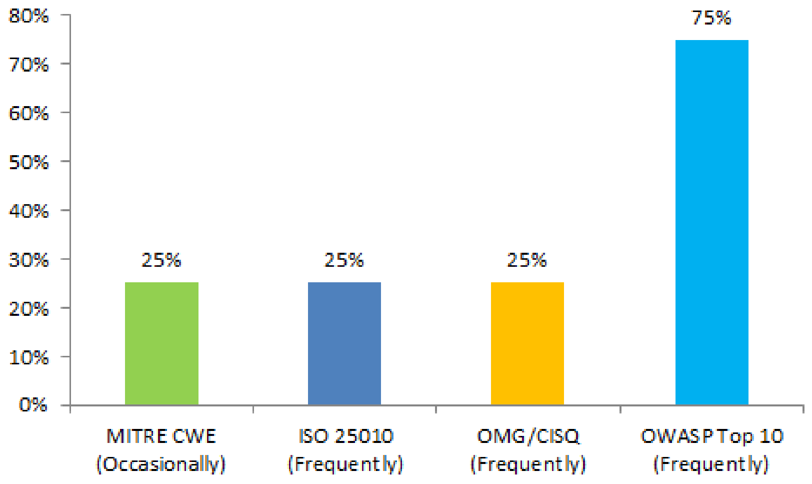
Results: 50% of vendor managers report they "always" mandate the use of SQA with suppliers and the remaining 50% saying they "occasionally" mandate SQA with suppliers.

Observations: Given the greater levels of automation within software development, we believe it is a reasonable expectation for a supplier to embed SQA into their toolchain. We therefore recommend vendors managers mandate the use of SQA standards with suppliers regardless of project type. We also recommend that vendor managers go further and have the use of SQA applied to maintenance contracts.



WHICH STANDARDS ARE VENDOR MANAGERS USING?

Question: The following standards have been identified as being related to software quality. How frequently do you use these standards with your suppliers?



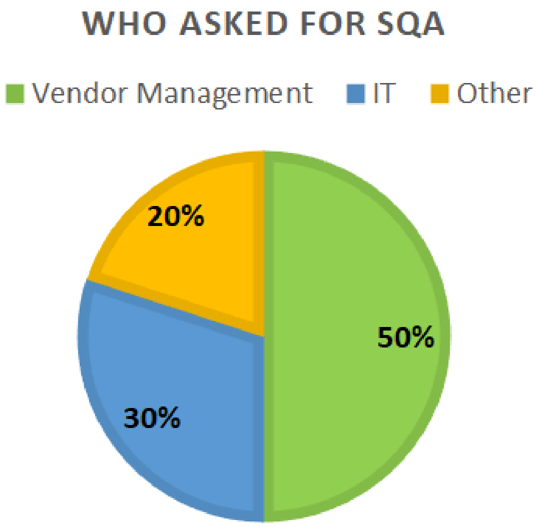
Results: OWASP Top 10 is cited as used "frequently" by 75% of vendor managers. The more specific and detailed SQA standards appear more rarely with only 25% of vendor managers report "frequently" or "occasionally" using these.

Observations: This result leads us to believe vendor managers are relying on their suppliers to decide what standards to use and if SQA is to be used at all. They are only scratching the surface and not using standards that have greater maturity in terms of SQA, such as OMG/CISQ, MITRE CWE, SANS Top 25, and ISO 25000.

WHO IS ASKING FOR SQA?

Question: Who initiated the use of SQA with your suppliers? (If applicable)

Results: 50% of vendor managers using SQA with suppliers report the vendor management team initiated its use. 30% report the IT function made an explicit request for its use.



Observations: This finding is positive, suggesting the vendor management function is becoming more proactive to reduce risk. Based on conversations we have had with vendor managers, the 50% figure may be a bit higher than the reality as respondents can be reluctant to report they have not initiated something that they suddenly realize they should have.

The most common reason cited by vendor managers for not initiating the use of SQA (after the obvious, they were not aware they could), is the concern of overstepping their remit and causing friction with IT.

"IT vendor managers need to be bold, reinvent themselves, and realize their function can have a big impact on the enterprise to reduce maintenance costs and risk. The agile enterprise and rapid growth of SaaS and niche service providers has increased the complexity of the vendor ecosystem and increases the probability of changes to the ecosystem impacting business functions. Third party risk management, data sovereignty and cyber security need to be managed holistically in today's environment. Technology enabled governance solutions are being adopted across organizations to manage the multiple relationships and complexity. "

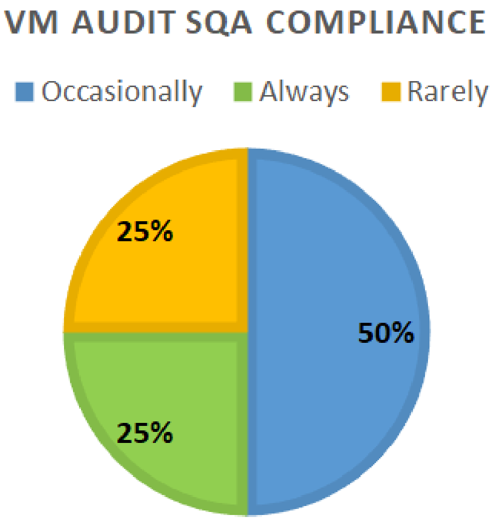
- Steve Hall, Partner and President, ISG and Advisory Board Member, CISQ

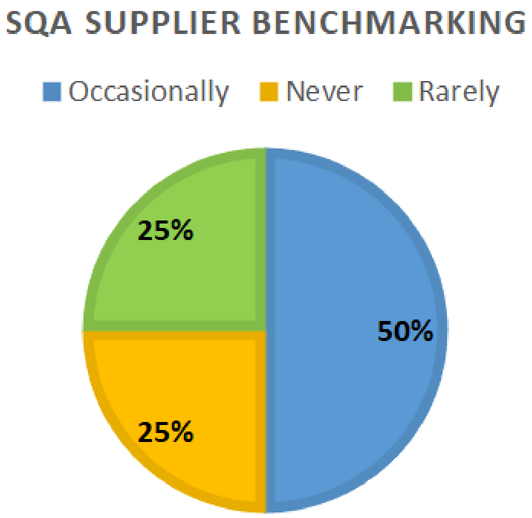
ARE VENDOR MANAGERS PUSHING SQA MATURITY WITH SUPPLIERS?

Question: Do you audit your suppliers for software quality compliance? Do you use software quality analysis to benchmark your suppliers?

Results: 25% of vendor managers report they "always" audit suppliers for SQA compliance, 50% report "occasionally," and 25% "rarely." When it comes to benchmarking, 50% of vendor managers "occasionally" benchmark their suppliers' SQA capability and the remainder report "rarely" or "never."

Observations: Regarding the maturity of SQA and IT vendor management, we are still at a comparatively low level. Few vendor management functions have linked SQA formally to contracts, auditing, or benchmarking of suppliers. We encourage greater collaboration between vendor managers and suppliers using agreed upon standards.





Recommendations for Vendor Management

CONCLUSION

Given the greater levels of automation within software development, it is a reasonable expectation for suppliers to embed software quality analysis into their toolchain. We therefore recommend that vendor management teams mandate the use of SQA with their suppliers regardless of project type. We also recommend that vendor managers go further and have the use of SQA applied to software maintenance contracts.

We recommend the vendor management community take steps to educate itself on SQA standards beyond the use of the "Top 10" lists. Currently, vendor managers are too reliant on SIs when it comes to SQA and they need to place themselves in a position of strength to safeguard their organization and business. Care must be taken not to introduce overly-bureaucratic vendor management processes. We need to improve the maturity of quality and SQA, and we believe the best way to do this is to mandate the use of mature standards.

In the digital economy, vendor management needs to take a more proactive approach with suppliers and internal customers. Vendor managers should communicate internally with IT and the business units that in addition to the basic vendor management competencies, the team can provide greater support to the enterprise for quality, security, and the reduction of total cost of ownership. Recognizing the concern that many vendor managers have with overstepping their bounds with IT, we recommend that vendor management have an honest discussion with IT and make it clear that vendor management needs to play a more proactive role given the importance of digital to the organization.

All development and maintenance contracts should be linked to positive improvements in structural quality and technical debt based on mature SQA standards. Vendor managers should enter into contracts with defined acceptable limits that are fair for both parties and with SQA metrics that can be measured automatically with low overhead to the supplier. We recommend the use of SQA in benchmarking suppliers so the vendor management function is better placed to balance value to the enterprise, cost to the organization, and quality and security to the end users. Again, the use of a mature standard places all vendors on a level playing field as they are aware of what they are being benchmarked against and clear on what they need to do to achieve high maturity.

With the increased use of SaaS-based subscription models and packaged development, it is easy for us to become complacent. However, vendor management has a role to play to ensure all systems entering the organization, whether built, bought, or rented, have a base level of security and quality. Vendor management should work with the business units to help them understand the risks of uncontrolled shadow IT. Vendor management should also work with the IT function to make sure the team is aware that the vendor management community can work with suppliers using a standards-based approach to SQA and security.

CISQ

Consortium for Information & Software Quality ™

About CISQ

The Consortium for Information & Software Quality™ (CISQ™) is an industry leadership group that develops international standards for automating the measurement of software size and structural quality from the source code. The standards written by CISQ enable organizations developing or acquiring software-intensive systems to measure the operational risk software poses to the business, as well as estimate the cost of ownership.

CISQ was co-founded by the Object Management Group® (OMG®) and Software Engineering Institute (SEI) at Carnegie Mellon University.