EMA™

IT & DATA MANAGEMENT RESEARCH,
INDUSTRY ANALYSIS & CONSULTING

The 2021 State of Software Quality: A View From the Enterprise

May 2021 EMA eBook
By Steve Hendrick

Table of Contents

What is software quality, and why is it important to enterprises?

It's easy to answer the question, "Is software quality important to the enterprise?" since nearly all enterprises are already in the midst of a digital transformation. Answering the question, "What is software quality?" is more complicated. This is because software quality is often a goal, and there are many ways to achieve this goal.

Software quality includes three attributes that exist and can change across the lifecycle of the software. These three attributes include:

- **The ability to meet functional needs.** This includes addressing application capabilities, security considerations, any compliance needs.

- **The ability to meet or exceed customer needs.** This includes addressing application usability concerns to ensure that the application meets or exceeds customer expectations.

- **The ability to meet non-functional needs.** Included here is addressing application availability, scalability, reliability, maintainability, extensibility, and performance.

Achieving software quality takes people, policies, processes, and products. In other words, it takes enterprise commitment at many levels to drive and improve software quality.

Together with Micro Focus, EMA set out earlier this year to understand how enterprises were approaching the subject of software quality. EMA gathered data from senior IT staff by surveying 316 enterprises of various sizes around the world. Details about this survey can be found in Appendix A. All of the research presented in this eBook is from the 2021 State of Software Quality (SoSQ) survey.

EMA's objective was to understand enterprise-wide perspectives on software quality, and also determine if these perspectives differed based on leadership in software quality or DevOps. To determine software quality and DevOps leadership, EMA asked each respondent to self-assess the level of software quality in the products they produce and maturity of their DevOps practice. This enabled EMA to identify both software quality and DevOps leaders, mainstreamers, laggards, and outliers. Details on how EMA mapped respondents into these four categories can be found in Appendix B.

The **goal of software quality** is to effectively address functional, customer, and non-functional needs.

Software quality's relationship to enterprise revenue

The majority of enterprises believe that software quality is important. A true test of this is evaluating the impact that software quality has on enterprise revenue. Overall, 70% of enterprises said that application development has a significant impact on revenue generation. Of the remaining 30%, 22% said application development has some impact on revenue generation and 8% said it had no impact.[1]

EMA looked more deeply into the impact of software quality on revenue by providing a more granular question that evaluated software's impact on revenue. This revealed a more nuanced perspective into how software quality impacts enterprise revenue. Because 2020 was a unique year, EMA phrased the question to look at the revenue impact across 2019 and 2020.

Despite the adverse impact on business and revenue caused by the COVID-19 pandemic, 44% of enterprises reported that the quality of their software products had a high impact on revenue. Notably, 27% of enterprises said that the quality of their software products is having a very high or extremely high impact on revenue. This contrasts with the 2% who reported that the quality of their products had no impact or a negative impact on revenue. Collectively, this means 98% of enterprises report that software quality is having a positive impact on driving revenue.

Over the last two years, how has the quality of your software products impacted enterprise revenue?
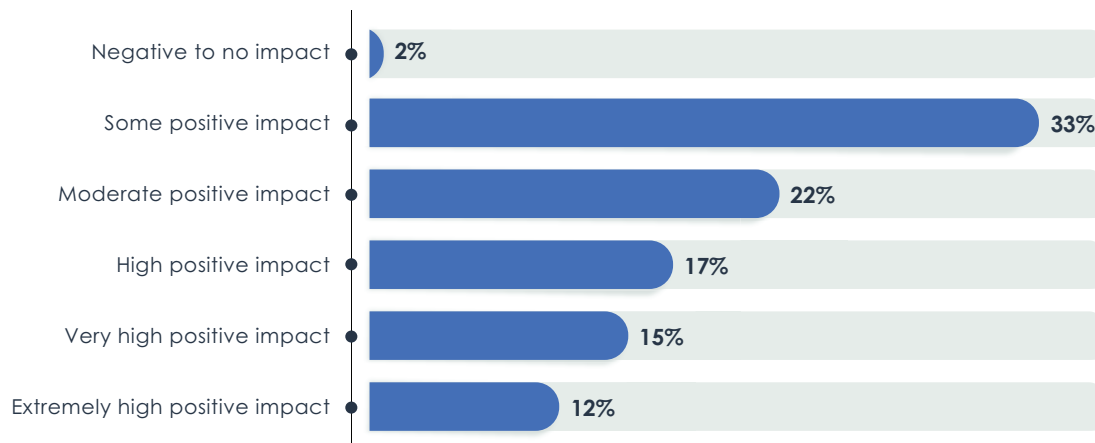
Category	%
Negative to no impact	2%
Some positive impact	33%
Moderate positive impact	22%
High positive impact	17%
Very high positive impact	15%
Extremely high positive impact	12%

98% of enterprises report that software quality is having a positive impact on driving revenue.

[1] 2021 Micro Focus SoSQ Survey, q15, N=316

Micro Focus 2021 SoSQ Survey
N=311, Valid cases

The importance of people in driving software quality

People matter, but other factors individually matter almost as much

The journey to software quality involves people, policies, processes, and products. People remain the focal point of software development because people write code, define policy that governs activities, determine processes, and use products. In order to test just how important people are to driving software quality, EMA developed a zero-sum question in which each respondent was given 100 points to be allocated across five categories: people, policies, processes, products, and audit/compliance. The expectation was that people would be most important, and that proved to be true based on its 25% overall mean share.

Overall mean shares across the other four categories were distributed almost evenly, with shares ranging between 18% and 20%. Surprisingly, this overall distribution changed very little when segmented by software leadership. The distributions for software quality leaders, laggards, and mainstreamers were almost identical to the one pictured. EMA would posit that better policies, processes, and products are making it easier and faster to build better software. A more formal and structured approach to software development provides opportunities to minimize defects through good design and comprehensive testing.

The net is that software development is undergoing its own digital transformation. Forty years ago, developing software was more art than science. Today, software development is more science than art and technologies like AI/ML will continue to push software development in the direction of science.

How does your enterprise balance the contribution to software quality that is made by people, policy, processes, and products (development and DevOps tools)?

- 25%
- 19%
- 18%
- 20%
- 18%

- ● The importance of people
- ● The importance of policy for guiding the SDLC
- ● The importance of processes for managing the pipelines
- ● The importance of products (tools used by Dev and Ops)
- ● The importance of passing audits and compliance

Micro Focus 2021 SoSQ Survey
N=316, Valid cases
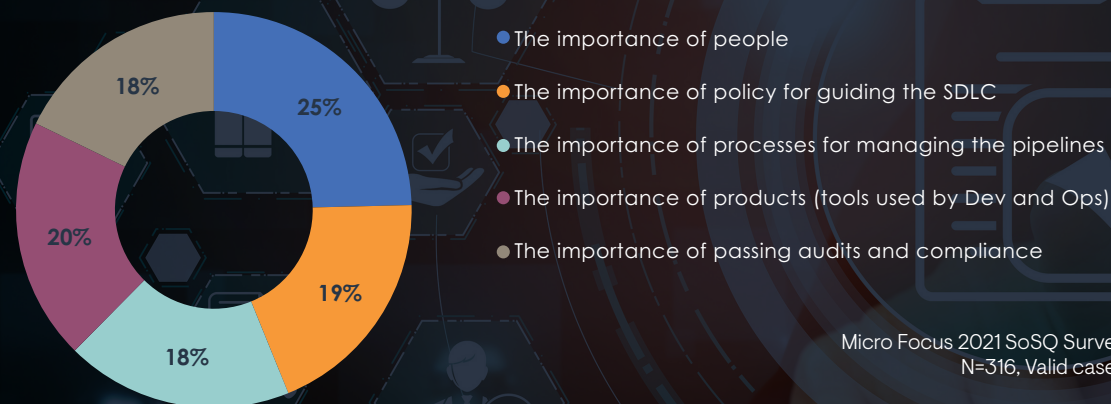
People matter, but collectively, policies, processes, and products **matter more**.

Who contributes to software quality?

When EMA asked which groups and teams contributed to driving software quality in the enterprise, we found that it takes a village to develop software. Application development (60%) and ITSM teams (60%) were the most critical to driving software quality, followed closely by software quality assurance teams (57%). Development and QA team involvement in software quality are integral parts of software development. The importance of ITSM in addressing software quality is due to the focus on incident management, problem management, and change/release management, which demonstrates an operational dimension of software quality.

Customers and users (48%) and C-level management (47%) are also closely aligned with addressing software quality. Customer advisory boards (CABs) and the increasing popularity of Canary deployments for non-mission-critical apps is becoming more common. Customers and users are both knowingly and unknowingly contributing to how software quality is improved.

C-level management is often where initiatives to improve software quality begin. C-level management usually consists of key stakeholders in any software quality initiatives, especially when it comes to defining policy or implementing a software quality center of excellence (CoE).

The increasing popularity of platform teams that deal with configuration and deployment requirements will likely increase the software quality emphasis that currently exists with operations teams (39%) and release management teams (36%).

Who contributes to driving software quality at your enterprise?

Team	Percentage
Application development teams	60%
IT service management teams	60%
Software quality assurance teams	57%
Customers and users	48%
C-level management	47%
Operations teams	39%
Release management teams	36%
Change advisory board	22%
Customer success teams	17%

Micro Focus 2021 SoSQ Survey
N=316, Valid cases

Successful enterprises
address software quality across the SDLC.

Who is primarily responsible for software quality?

While it takes a village to drive software quality, is there one group within IT primarily responsible for software quality? This research found that when asked what one group in the enterprise is primarily responsible for driving enterprise software quality, the response was mixed.

Beliefs about who drives software quality depend on IT role. Application developers see software quality as their responsibility more than anyone else. Likewise, IT operations see software quality (in the areas of IT operations and ITSM) as primarily their responsibility. Finally, C-level management sees software quality as their primary responsibility—a view that is not often shared by developers or IT operations.

The reality is that everyone is correct, and no one is correct. Everyone is correct because from their role and perspective, they have unique ways to address software quality that are important to the enterprise. No one is correct because we already know that it takes a village to address software quality needs.

What one group in your enterprise is primarily responsible for driving enterprise software quality? – segmented by – Which of the following best or most closely describes your primary job role or title within your organization?

	Application Development	IT Operations	C-Level
IT service management teams	19%	41%	25%
Software quality assurance teams	27%	16%	15%
C-level management	10%	6%	25%
Application development teams	22%	9%	13%
Customers and users	12%	3%	5%
Operations teams	3%	13%	5%
Release management teams	4%	6%	5%

● Application Development   ● IT Operations   ● C-Level
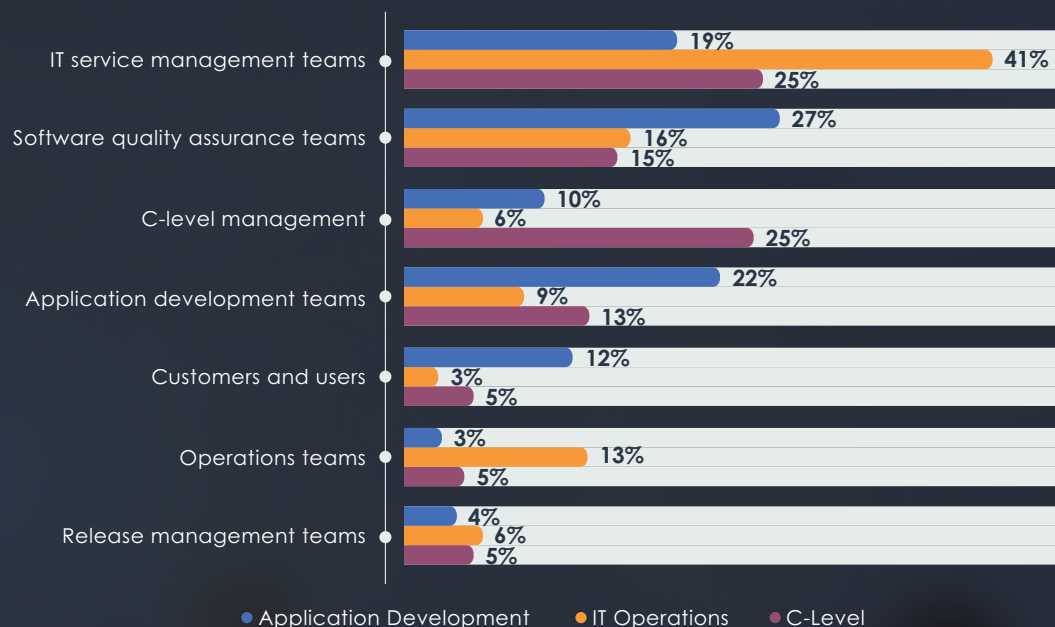
It takes **leadership** and **key contributions** across the SDLC to address software quality in a comprehensive way.

The importance of policy and process to improving software quality

Most software development methodologies in use are agile or agile-like

There is no shortage of software development methodologies. Agile methods (extreme programming, Crystal, scrum, Lean, and Kanban) and agile-like methods (Incremental, Spiral, V-model, Rapid application development, Prototype, and others) are the dominant approaches used today.

When asked what methodologies are being used to develop software code, 67% identified an agile or agile-like methodology. Homegrown approaches accounted for 23% of enterprises and 14% for waterfall.

A homegrown approach is favored by software quality leaders more so than mainstreamers or laggards. EMA believes that enterprises using a homegrown approach have probably adopted a hybrid methodology that includes best practices from various approaches they have used in the past.

The 14% share of enterprises using waterfall may seem high, but with methodologies like SAFe, which can look like a more iterative approach to waterfall, it is likely that today's approach to waterfall looks far more lean and agile than it did 20 years ago.

What share of software code is currently being developed across the following software methodologies?

- Homegrown — 23%
- Agile (Scrum, SAFe, XP...) — 22%
- Incremental — 15%
- Spiral — 13%
- V-Model — 13%
- Waterfall — 14%
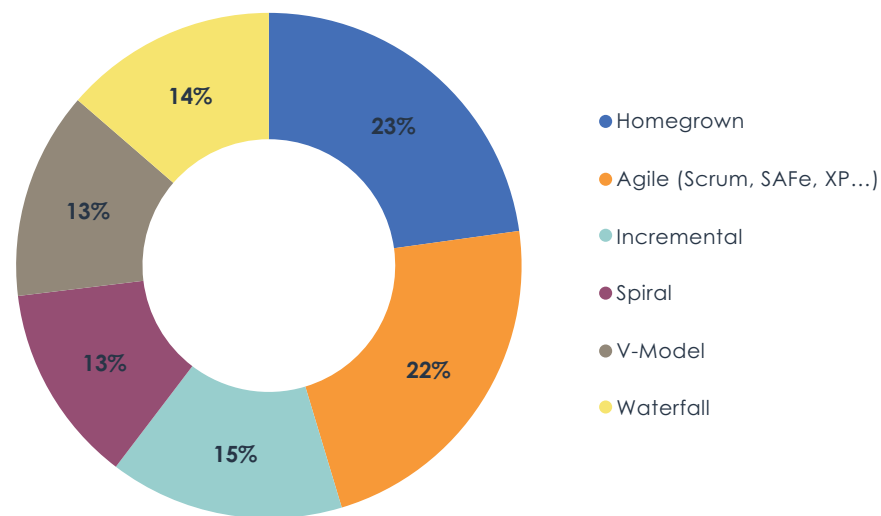
The **majority** of enterprises today have embraced agile or agile-like methodologies.

Micro Focus 2021 SoSQ Survey
N=316, Valid cases

Software quality leadership aligns with use of agile methods

Although 67% of enterprises use agile or agile-like methods, how pervasive is this use and does it vary by software quality leadership? The answer is that reliance on agile methods is considerably different depending on the quality of software being produced.

Thirty-eight percent of enterprises that are software quality leaders say that nearly all of their teams are using agile methods, compared to just 16% of mainstreamers and 24%

of laggards. The distribution of responses for each of the three software leadership categories is even more revealing.

The largest subsegment (or peak value) of software quality leaders is the 38% in which nearly all teams are using agile methods. The largest subsegment for software quality mainstreamers or laggards is where *most* teams are using agile methods.

This data clearly shows that software quality leaders are more pervasive in their use of agile methods. This strong correlation between agile adoption and software leadership is just one example of many correlations between software leadership and DevOps maturity, cloud adoption, use of containers, and scope/penetration of DevOps tools.

How would you describe the planned adoption of agile methodologies in your enterprise? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

	SW Quality Laggards	SW Quality Mainstreamers	SW Quality Leaders
We have no plans to implement agile methods	2%	0%	2%
We have agile plans, but have not begun to implement them	6%	4%	3%
We are just beginning to adopt agile methods	11%	6%	7%
A few teams are using agile methods	6%	5%	4%
Some teams are using agile methods	17%	23%	16%
Most teams are using agile methods	33%	46%	29%
Nearly all teams are using agile methods	24%	16%	38%
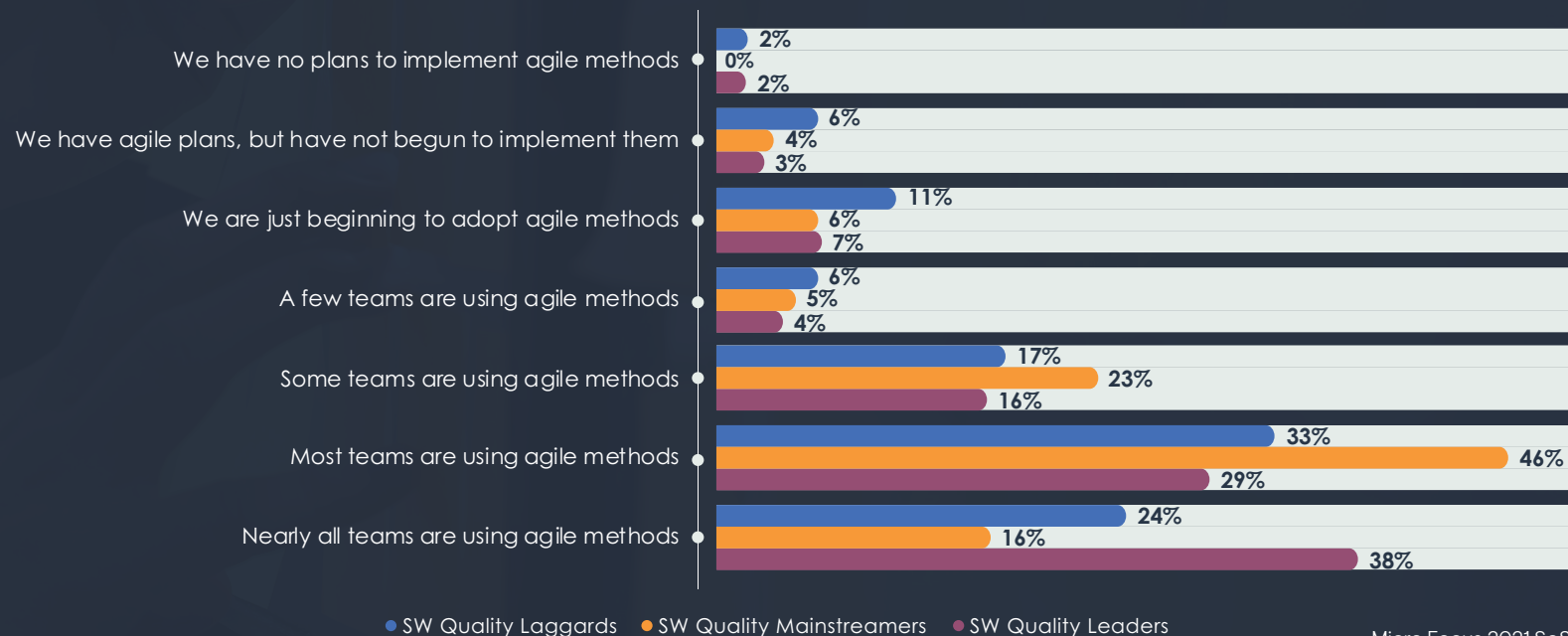
Software quality leaders excel at requirements and testing

Requirements and testing are two areas of the SDLC that have a profound impact on the quality of software developed. Requirements are established early in the SDLC and testing occurs throughout CI and CD activities. Since it is many times more expensive to resolve defects found in production than found during development, careful attention to requirements and testing will have a significant positive impact on the speed and ease in developing high-quality software.

Measuring quality in requirements, testing, and all of the other lifecycle stages

listed here means coming to terms with how to measure quality, what to measure, and how well the quality objectives were met. Collecting measurements facilitates more precision in management.

Software quality leaders are almost always more highly invested in measuring activities across the SDLC compared to software followers. This is especially true in requirements and testing. Requirements and testing play such an active role in driving quality in the early stages of software development that being especially attentive to

measuring and managing requirements and testing should be considered a best practice.

Performance monitoring is another area in which metrics are invaluable in application management. While the differences between software leaders and followers is most acute in requirements and testing, performance monitoring is another area where there is a material difference. While performance monitoring is more focused on production applications, it fulfills a critical role in meeting SLAs and supporting incident-related RCA.

In what phases of the SDLC are you measuring quality? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Phase	SW Quality Laggards	SW Quality Mainstreamers	SW Quality Leaders
Requirements Management	36%	44%	60%
Version Management	31%	46%	50%
Build Management	40%	44%	46%
Deployment Management	38%	50%	51%
Test Management	41%	50%	59%
Release Management	40%	39%	40%
Operations Management	47%	58%	50%
Change Management	23%	33%	37%
Performance Monitoring	23%	34%	40%

● SW Quality Laggards ● SW Quality Mainstreamers ● SW Quality Leaders
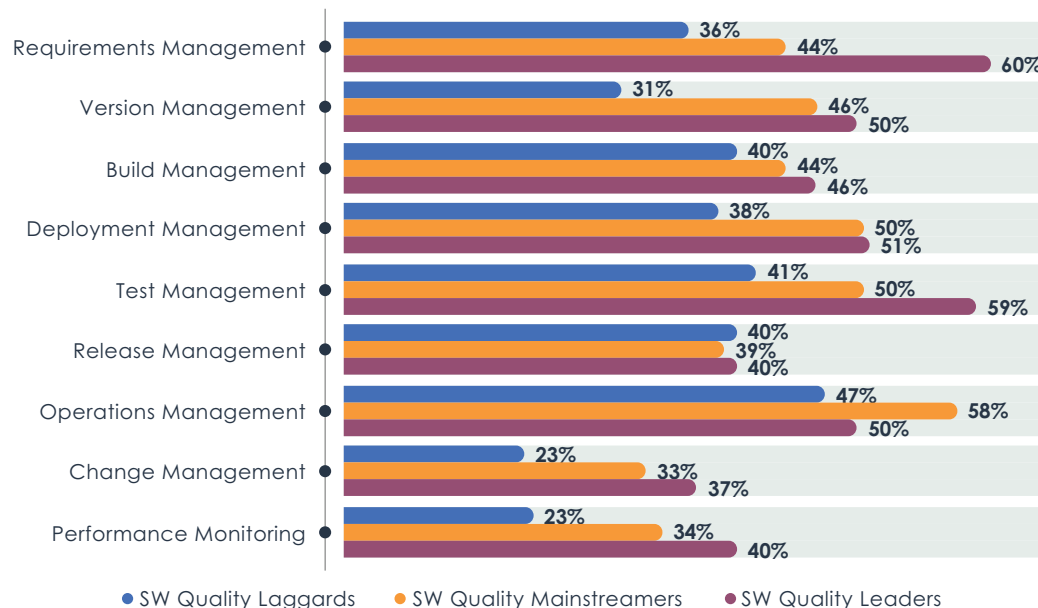
Leveraging metrics that evaluate requirements and testing effectiveness is a **best practice** in CI/CD.

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
Laggards (5-7), Mainstreamers (8), Leaders (9-10)

Software quality leaders understand the importance of metrics

Measurement is the foundation for management and improvement. This is clearly a message that software quality leaders understand. This survey question was designed to present a mini maturity model focused on the collection of metrics across the SDLC.

EMA would naturally expect software quality leaders to adopt a comprehensive approach to measuring across the SDLC and their value streams, but we did not expect to see a difference of this magnitude.

Forty-six percent of software quality leaders collected metrics across all activities in a formal and structured way. This contrasts with 11% of mainstreamers and 7% of laggards. The wording of the responses to the question being asked is very important. Most DevOps tools provide a degree of observability out of the box, so most enterprises have a heterogeneous collection of some metrics across the lifecycle. However, investing in a framework to integrate and facilitate management across value streams in a structured way is a far more complex activity. This data reveals the challenges and benefits from adopting value stream management tools and principles.

How effectively is your enterprise leveraging software quality metrics to manage development? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

We don't collect enough metrics to have them help manage development
- 2%
- 0%
- 3%

Some metrics are being collected, but management doesn't really rely on them
- 12%
- 5%
- 3%

Some metrics are collected and help monitor and manage CI activities
- 21%
- 8%
- 3%

Metrics are being collected across key SDLC activities, including CI, and weuse these metrics in an informal way to manage development
- 20%
- 23%
- 7%

Metrics are being collected across key SDLC activities, including CI, and we use these metrics in a formal way to manage development
- 23%
- 24%
- 13%

Metrics are collected across most SDLC activities and are used in a formal way to manage development
- 15%
- 28%
- 23%

Metrics are collected across all activities that matter and are used in a formal and structured way to manage all of our software development
- 7%
- 11%
- 46%

● SW Quality Laggards ● SW Quality Mainstreamers ● SW Quality Leaders

Software quality leaders take **value stream management** seriously.

The benefits of leveraging software quality metrics

Using metrics to drive software development has many advantages. The list of benefits and the accompanying chart are ordered based on overall enterprise response. Increased productivity (56%) and better resource management (55%) are the two most frequently cited ways that metrics drive benefits, and this perspective is essentially shared by all enterprises regardless of their level of software leadership.

Improvements to software quality (53%) is the third-ranking overall benefit. However, software quality laggards struggle to recognize this benefit relative to mainstreamers and leaders. The challenge that software laggards have is a much lower incidence of measuring SDLC activities including requirements, testing, and performance monitoring. Because software quality laggards capture fewer metrics across the lifecycle, they are less well-positioned to use these metrics for improving software quality.

The differences between software leaders and mainstreamers compared to software laggards becomes even more glaring when it comes to metrics used to manage the software development pipeline. Sixty percent of software quality leaders are well-positioned to manage their pipeline due to the comprehensive and coherent way they collect, integrate, and leverage metrics. Software quality leaders (60%) are well ahead of mainstreamers (45%) and both of these groups are well ahead of software quality laggards (35%).

What is most beneficial about collecting and analyzing software quality metrics? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Our metrics have led to increased productivity
- 54%
- 55%
- 58%

Our metrics enable us to manage our resources better
- 52%
- 58%
- 56%

Our metrics have improved product quality
- 45%
- 64%
- 59%

Our metrics enable us to manage our pipeline better
- 35%
- 45%
- 60%

Our metrics enable us to quickly understand what's working and what's not
- 21%
- 29%
- 38%

- ● SW Quality Laggards
- ● SW Quality Mainstreamers
- ● SW Quality Leaders

Resource productivity, resource manageability, and software quality are the **top three benefits** from instrumenting the SDLC.

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
Laggards (5-7), Mainstreamers (8), Leaders (9-10)

DevOps maturity and a software quality CoE are key to a software quality strategy
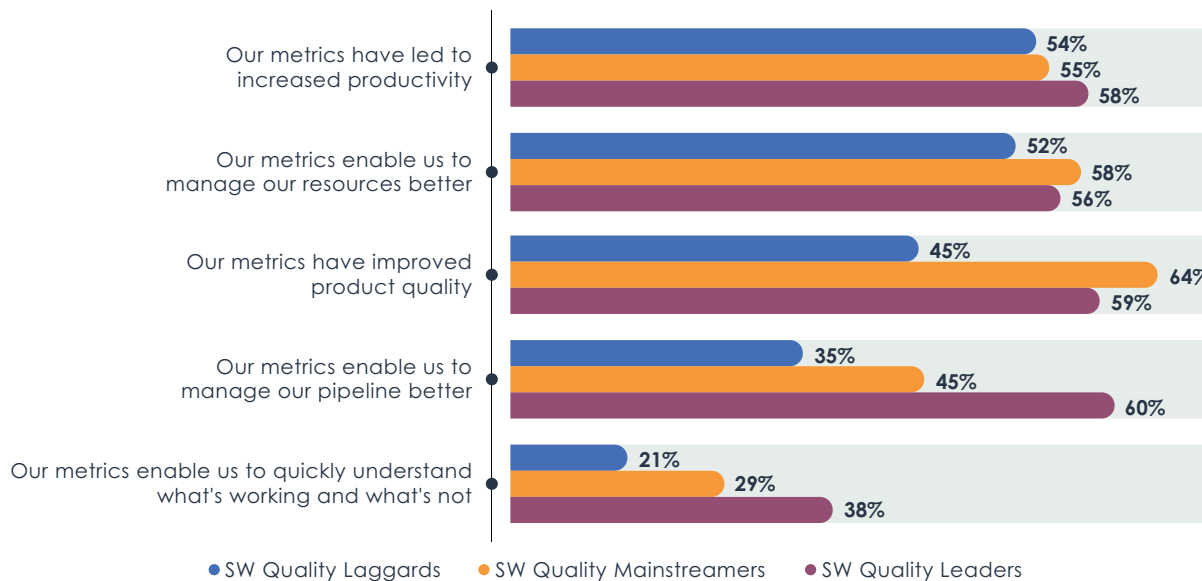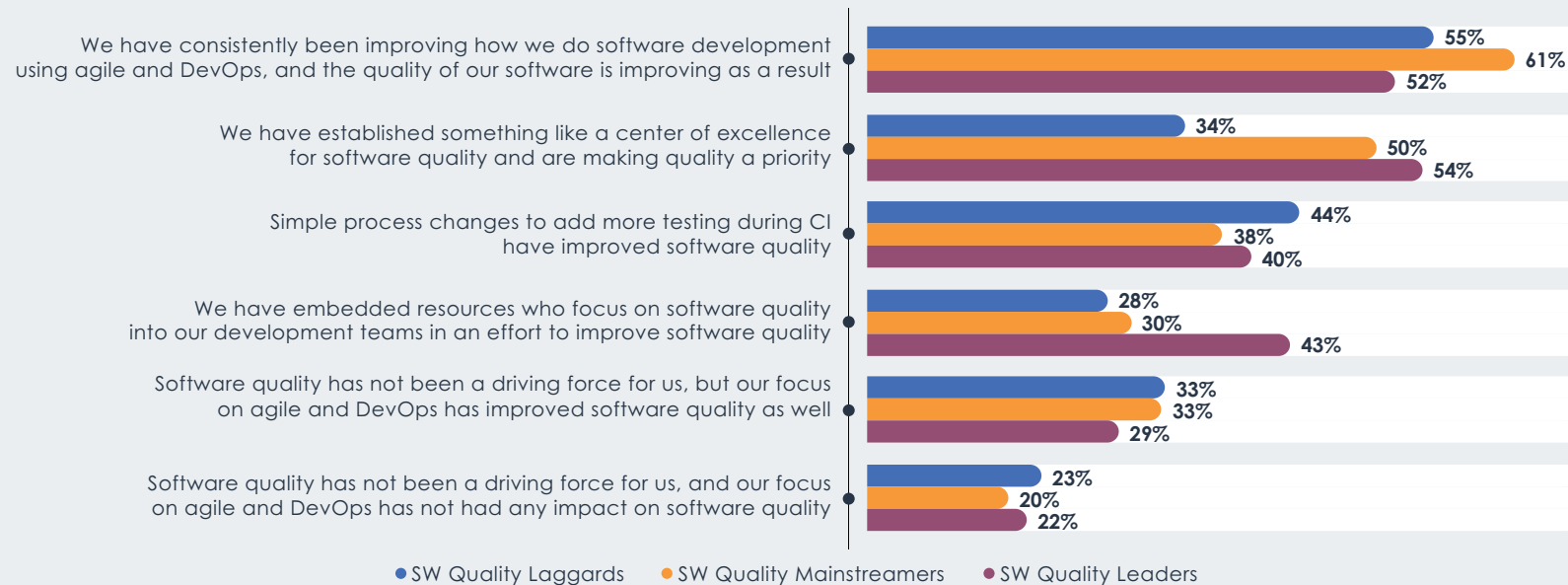
Improvements to software quality are functions of people, policies, processes, and products. Fifty-five percent of enterprises, including those at all levels of software leadership, state that the increasing use of agile methods and improvements to their DevOps toolchain have likewise improved the software quality of their products. This is a message that has been unfolding throughout this report, but methodology and process are just two elements that need attention on a software quality journey.

Establishing a software quality center of excellence (CoE) was identified by 44% enterprises overall as a way to make software quality a priority. However, this was a perspective shared far more by software quality leaders and mainstreamers than it was by software quality laggards.

Another best practice of software quality leaders is the attention to software quality by their development teams. EMA believes that software quality is a consideration shared by many members in the team, including business analysts, developers, quality assurance, SREs, and the team manager. Forty-three percent of software quality leaders report that their teams have embedded resources that have responsibility in part to drive software quality. This contrasts with 30% of mainstreamers and 28% of laggards who have also adopted this approach.

What best describes your software quality journey over the last two years? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

We have consistently been improving how we do software development using agile and DevOps, and the quality of our software is improving as a result
- 55%
- 61%
- 52%

We have established something like a center of excellence for software quality and are making quality a priority
- 34%
- 50%
- 54%

Simple process changes to add more testing during CI have improved software quality
- 44%
- 38%
- 40%

We have embedded resources who focus on software quality into our development teams in an effort to improve software quality
- 28%
- 30%
- 43%

Software quality has not been a driving force for us, but our focus on agile and DevOps has improved software quality as well
- 33%
- 33%
- 29%

Software quality has not been a driving force for us, and our focus on agile and DevOps has not had any impact on software quality
- 23%
- 20%
- 22%

● SW Quality Laggards   ● SW Quality Mainstreamers   ● SW Quality Leaders

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
Laggards (5-7), Mainstreamers (8), Leaders (9-10)

The importance of product decisions to improving software quality

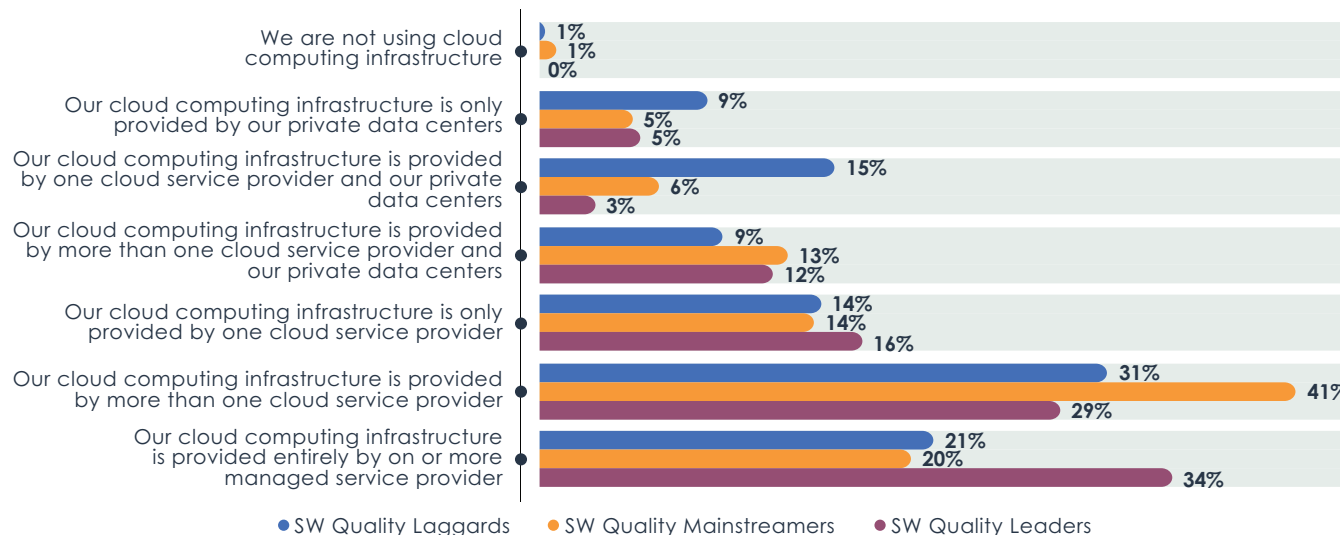## How cloud computing strategies impact software quality

How enterprises approach cloud computing has a significant impact on the level of software quality they deliver. Just 1% overall of the enterprises EMA encountered were not using cloud computing infrastructure, 27% were using a hybrid cloud model, 47% appear to be single/multi-cloud-centric, and 25% were consuming cloud services through a SaaS or managed service provider (MSP).

The exciting find when viewing this data through the lens of software quality leadership is that software quality leaders are significantly more committed to a SaaS and MSP delivery model. The MSP model includes third parties that were engaged to manage aspects of the SDLC, especially those focused on production workloads. SaaS products provide a similar experience because their operation and management is also addressed by a third party. The SaaS/MSP delivery model is embraced by 34% of software quality leaders, compared to just 20% of mainstreamers and 21% of laggards.

The SaaS/MSP delivery model is significantly different from the multi-cloud model used by the highest percentage of software quality mainstreamers and laggards. This difference stems from the efficiency of outsourcing complex aspects of operations, software delivery, and even software development. This leaves the enterprise with more bandwidth to pursue developing new revenue channels and addressing existing customer/partner needs.

How would you describe your enterprise's approach to cloud computing? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

	SW Quality Laggards	SW Quality Mainstreamers	SW Quality Leaders
We are not using cloud computing infrastructure	1%	1%	0%
Our cloud computing infrastructure is only provided by our private data centers	9%	5%	5%
Our cloud computing infrastructure is provided by one cloud service provider and our private data centers	15%	6%	3%
Our cloud computing infrastructure is provided by more than one cloud service provider and our private data centers	9%	13%	12%
Our cloud computing infrastructure is only provided by one cloud service provider	14%	14%	16%
Our cloud computing infrastructure is provided by more than one cloud service provider	31%	41%	29%
Our cloud computing infrastructure is provided entirely by on or more managed service provider	21%	20%	34%

● SW Quality Laggards   ● SW Quality Mainstreamers   ● SW Quality Leaders

Software quality leaders **embrace** SaaS/MSP cloud solutions.

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4), Laggards (5-7), Mainstreamers (8), Leaders (9-10)

Approaches to DevOps tool selection differ at the extremes

There are multiple enterprise strategies for acquiring DevOps tools. An enterprise can lead to a strategic vendor or partner for tools (28% overall), seek out a heterogeneous (36%) or homogeneous (49%) toolchain based on best of breed, or acquire DevOps tools on a case-by-case basis, specifically matched to enterprise needs (55%).

The distribution of responses shows consistency across DevOps leadership categories when selecting heterogeneous best-of-breed or homogeneous best-of-suite alternatives.
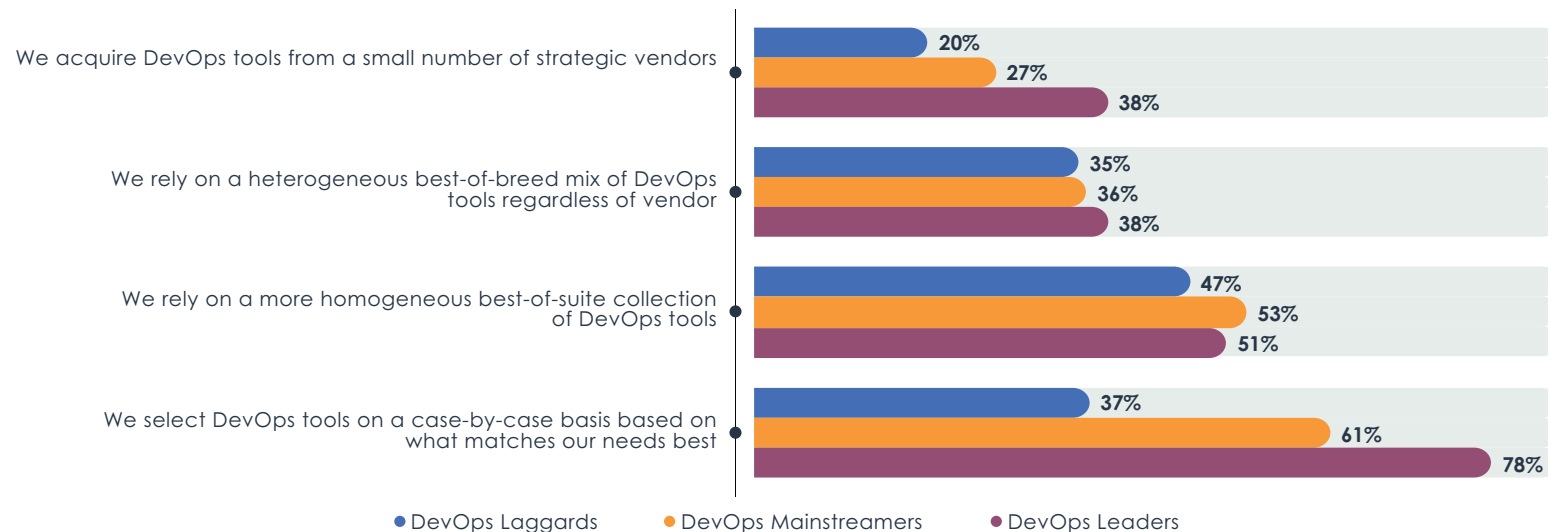
However, it is surprising to see significant preference for homogeneous over heterogeneous solutions. The reason for this is an adverse reaction to the longstanding fragmentation of the DevOps market and the efforts of leading vendors to develop tools in adjacent DevOps market segments.

At the extremes, there is a significant difference in support for acquiring tools from strategic vendors or on a case-by-case basis. DevOps leaders have a strong preference for acquiring tools from strategic vendors,

who were on a case-by-case basis relative to DevOps mainstreamers. They had a nearly 2x preference relative to DevOps laggards. DevOps leadership and software quality leadership are highly correlated, but segmenting this question by DevOps leadership is more relevant.

The importance of developers to DevOps leaders was also shown in a related question on developer empowerment. This question clearly showed that IT teams of DevOps leaders drove IT tool selection.[2]

What is your organization's approach to DevOps tool selection? – segmented by – How would you describe your enterprise's approach to DevOps?

We acquire DevOps tools from a small number of strategic vendors
- 20%
- 27%
- 38%

We rely on a heterogeneous best-of-breed mix of DevOps tools regardless of vendor
- 35%
- 36%
- 38%

We rely on a more homogeneous best-of-suite collection of DevOps tools
- 47%
- 53%
- 51%

We select DevOps tools on a case-by-case basis based on what matches our needs best
- 37%
- 61%
- 78%

● DevOps Laggards   ● DevOps Mainstreamers   ● DevOps Leaders

---

[2] Micro Focus 2021 SoSQ survey, q23, N=316

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers=no plans or plans, laggards=automated some or beginning, mainstreams=automated many, Leaders=automated nearly all
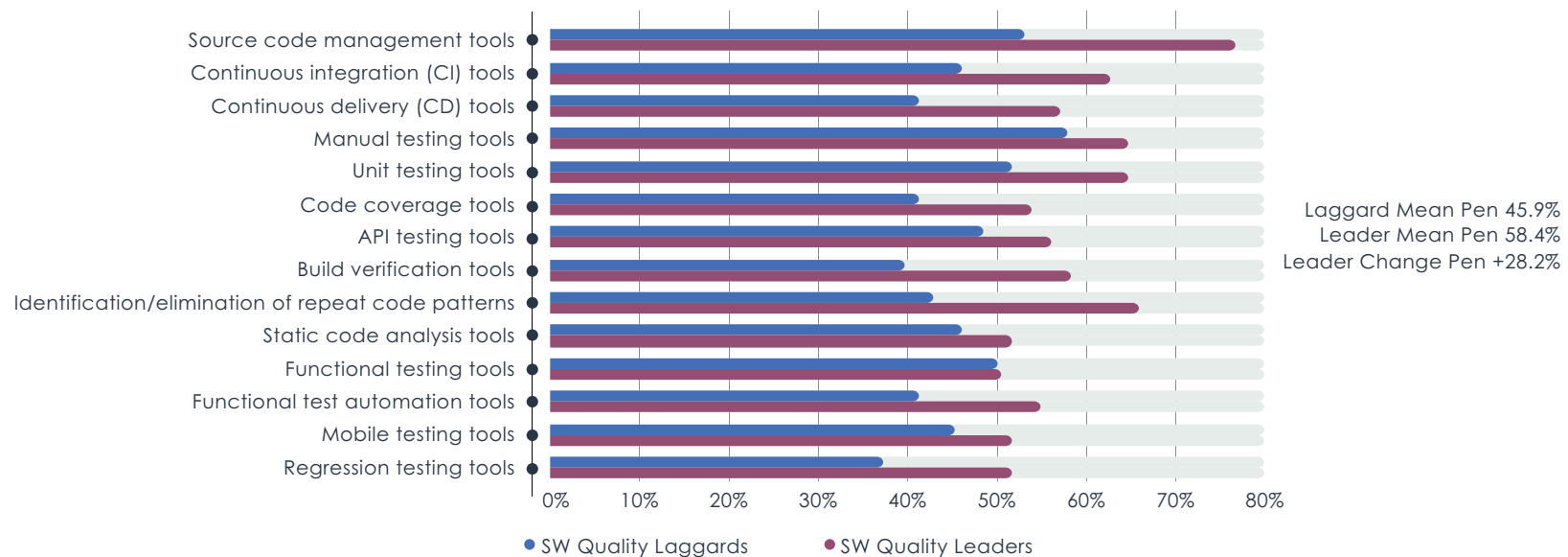
Development toolchain attributes of software quality leaders

DevOps staff use a variety of tools to support application development, deployment, and operations. EMA tracked the current and projected penetration of over 40 tool categories that span DevOps, looking at tool categories specifically oriented around CI (development of a working application). Across these development tool categories, EMA tracked the penetration rates of tool categories for software quality leaders and laggards.

The mean penetration rate for software quality laggards is 46% compared to 58% for software quality leaders. This means that the average penetration for software quality leaders is 28% higher than laggards. This higher penetration of tool categories by leaders means more tools are in use by software quality leaders relative to laggards. Greater tool use provides more opportunities for increased developer team productivity and toolchain automation.

Taking into account that DevOps is 11 years old and virtually every enterprise's DevOps journey began with continuous integration, EMA would not expect the difference in penetration across software leadership categories to be too extreme. However, this is exactly the case, especially when compared to deployment and operation toolchain attributes on the following pages.

What DevOps tools are regularly used or will be used in application development? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Laggard Mean Pen 45.9%
Leader Mean Pen 58.4%
Leader Change Pen +28.2%

● SW Quality Laggards   ● SW Quality Leaders

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
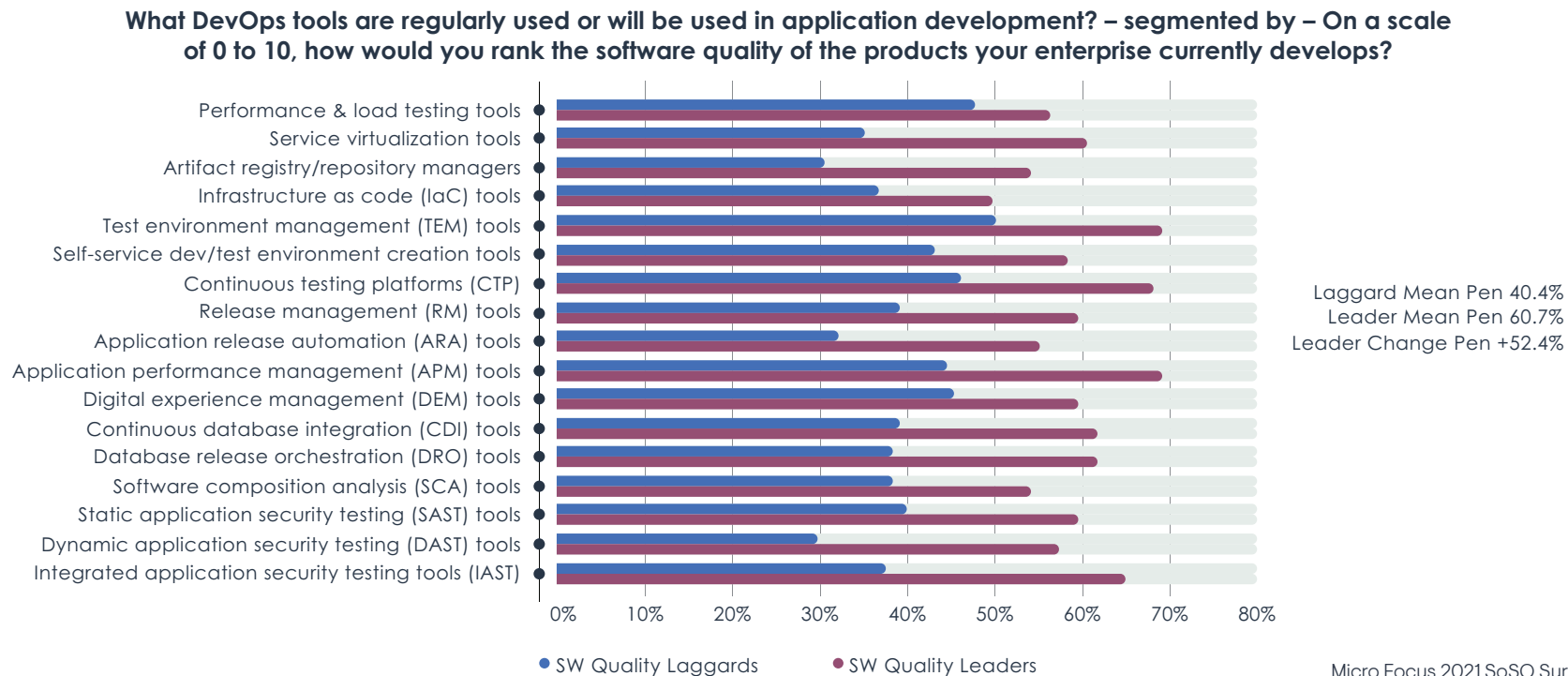Laggards (5-7), Mainstreamers (8), Leaders (9-10)

Deployment toolchain attributes of software quality leaders

EMA is characterizing application deployment as the configuration and transition of an application from test across staging and production environments. This involves an array of specialized testing and configuration to ensure the reliability and scalability of the production application. Consequently, application deployment is often the most challenging defect-prone collection of activities in the pipeline.

The complexities of application deployment are compounded by the early emphasis on CI and the more recent introduction of technologies like containerization, Kubernetes, and infrastructure as code (IaC) tools. Software quality leaders, given their deeper experience in DevOps and the cloud, are better positioned to leverage deployment tools. This is reflected in penetration rates. Software quality leader mean penetration rates across these toolchain categories is 61% compared to laggards, who are at 40%.

This creates an even more significant gulf between leaders and laggards, with the average penetration of leaders at 52% higher than laggards.

Because the hardware and software stack changes considerably during the transition from QA to staging and then to production, testing becomes critical. Performance, load, database, and security testing are necessary to ensure the configuration of the application is production-ready.

What DevOps tools are regularly used or will be used in application development? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Laggard Mean Pen 40.4%
Leader Mean Pen 60.7%
Leader Change Pen +52.4%

● SW Quality Laggards   ● SW Quality Leaders

Operational toolchain attributes of software quality leaders

Operational tools, including the tool categories encompassed here, continue to transform at a rapid rate. The reason for this transformation is a data-driven aspect of these tools that encourages an overlay of advanced analytics, AI/ML, and policy engines that are significantly increasing functionality on a quarterly basis.

The rapid evolution of many operational tools, combined with tool categories including observability, value stream management, and AIOps that were born in the last five years, delayed the adoption by software quality laggards. The result is penetration rates that largely mimic application deployment, with a mean penetration rate for software quality laggards at 41% and 60% for software quality leaders. This creates a large gap between software quality leaders and laggards, with the mean penetration rate of software quality leaders being 46% higher than laggards.

Value stream management (VSM) is an interesting case in point. VSM tools had low double-digit overall penetration rates back in 2019. Today, the 51% mean penetration of VSM tools for software quality leaders is 70% higher than the 30% mean penetration for software quality laggards. The VSM market is intensely data-driven and stands to benefit from the overlay of advanced analytics, AI/ML, and inference engines that will drive high growth for years to come.

What DevOps tools are regularly used or will be used in application development? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Laggard Mean Pen 41.4%
Leader Mean Pen 59.7%
Leader Change Pen +46.2%
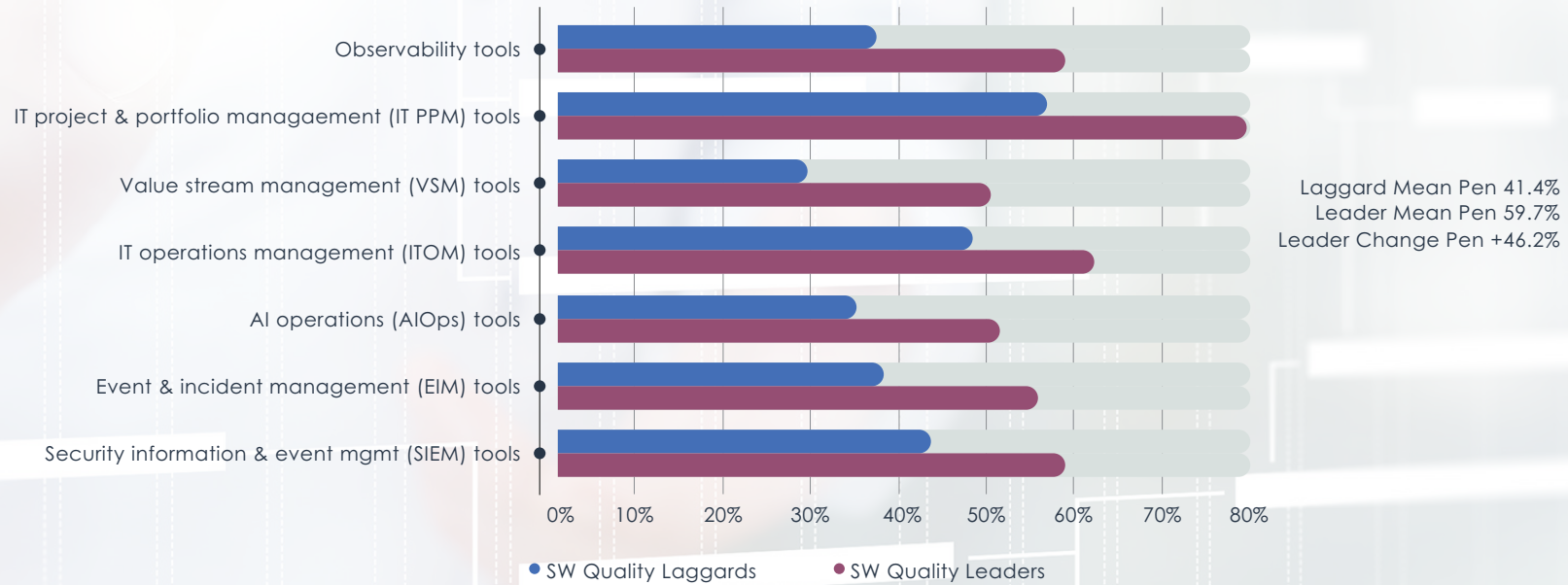
● SW Quality Laggards   ● SW Quality Leaders

Characteristics of software quality leaders

Why enterprises are improving their software quality

Evaluating the top three scores of software quality leaders proves insightful. Software quality leaders are highly focused on how to rethink software quality to retain and expand their customer base (61%), realizing that software can drive a competitive advantage (57%), and senior management has a better vision for the role of software in their enterprise (56%). These scores characterize enterprises fixated on digital transformation. All three of these responses have a strategic orientation, unlike the more
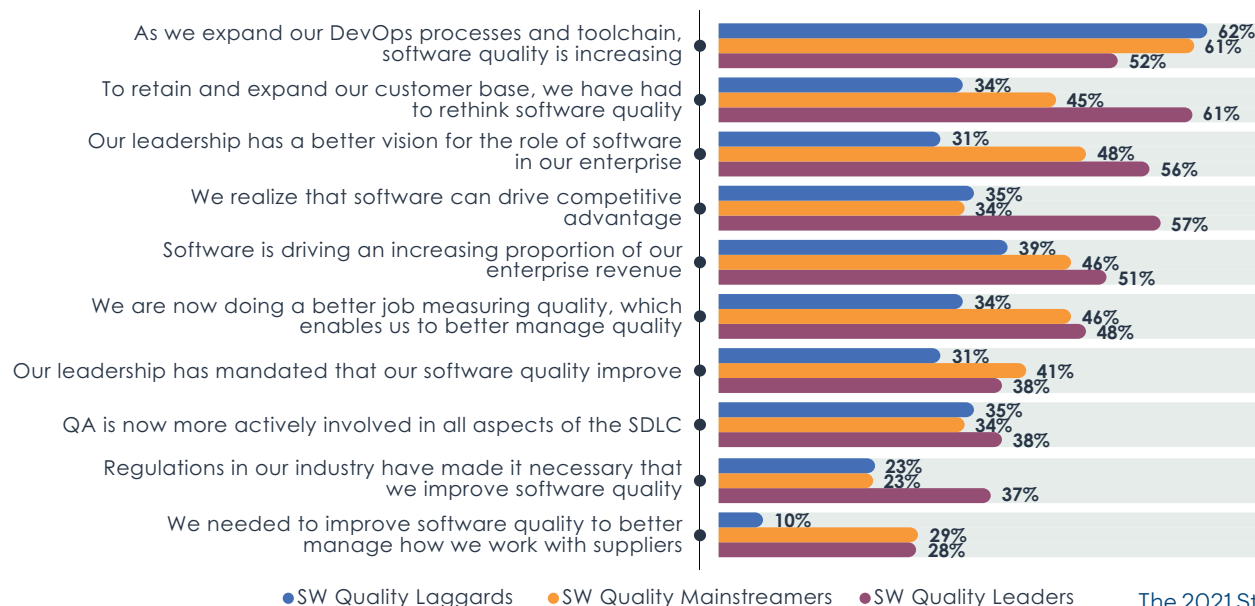
tactical responses elsewhere in this question. This digital transformation emphasis, combined with a strong cloud SaaS/MSP orientation provides these enterprises with the agility and flexibility to pivot rapidly toward new opportunities.

There is a strong consensus that improvements to DevOps are also helping to improve software quality. Overall, 58% of enterprises support this finding, with software quality mainstreamers and laggards

both identifying it as their top driver to improve software quality.

Comparing the emphasis of software quality leaders to followers shows a significant difference in their strategic outlook despite sharing a variety of tactical objectives. Becoming a software quality leader requires understanding the many ways that software quality leaders differ from followers and systematically looking for ways to improve software quality across the SDLC.

Why has your approach to software quality improved since the beginning of 2019? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Statement	SW Quality Laggards	SW Quality Mainstreamers	SW Quality Leaders
As we expand our DevOps processes and toolchain, software quality is increasing	62%	61%	52%
To retain and expand our customer base, we have had to rethink software quality	34%	45%	61%
Our leadership has a better vision for the role of software in our enterprise	31%	48%	56%
We realize that software can drive competitive advantage	35%	34%	57%
Software is driving an increasing proportion of our enterprise revenue	39%	46%	51%
We are now doing a better job measuring quality, which enables us to better manage quality	34%	46%	48%
Our leadership has mandated that our software quality improve	31%	41%	38%
QA is now more actively involved in all aspects of the SDLC	35%	34%	38%
Regulations in our industry have made it necessary that we improve software quality	23%	23%	37%
We needed to improve software quality to better manage how we work with suppliers	10%	29%	28%

● SW Quality Laggards   ● SW Quality Mainstreamers   ● SW Quality Leaders

Software quality leaders and followers are on the same journey but reflect significantly different levels of maturity.

How enterprises are improving their software quality

The top three overall responses to how enterprises are improving their software quality included investing more in understanding customer needs (53%), increased automation across the SDLC (50%), and more regular collaboration with customers (48%). While significant differences do exist between software quality leaders and laggards, the percentage responding across all categories is 40% or higher, which is compe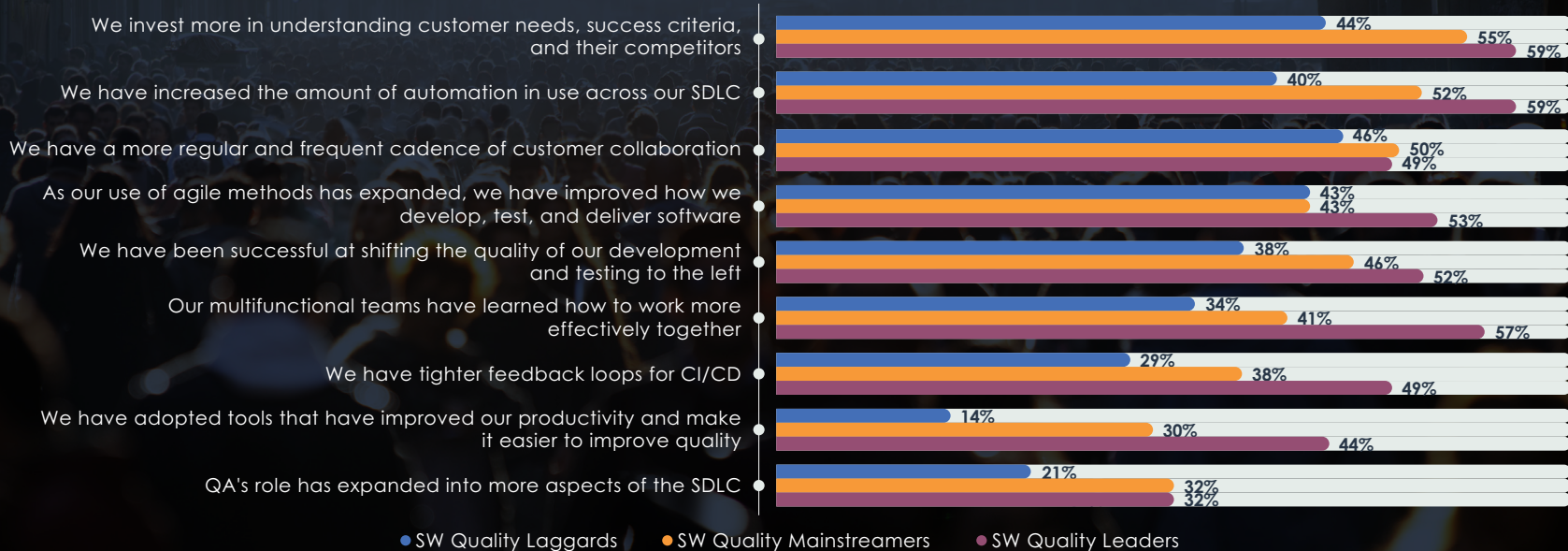lling. Higher levels of communication and collaboration between enterprises and their customers should be a win/win for both parties.

Software quality leaders excel at every aspect of DevOps and development presented in this question. The difference between software quality leaders and laggards was statistically significant across almost every response. Responses in which this difference was more than 20 percentage points should include the greater effectiveness of multifunctional teams, tighter feedback loops for CI/CD, and the adoption of tools that have improved productivity.

While software quality laggards are behind in every response, they are performing well (40% or more) in areas jointly identified as important, which include the top four responses.

How has your approach to software quality improved since the beginning of 2019? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Response	SW Quality Laggards	SW Quality Mainstreamers	SW Quality Leaders
We invest more in understanding customer needs, success criteria, and their competitors	44%	55%	59%
We have increased the amount of automation in use across our SDLC	40%	52%	59%
We have a more regular and frequent cadence of customer collaboration	46%	50%	49%
As our use of agile methods has expanded, we have improved how we develop, test, and deliver software	43%	43%	53%
We have been successful at shifting the quality of our development and testing to the left	38%	46%	52%
Our multifunctional teams have learned how to work more effectively together	34%	41%	57%
We have tighter feedback loops for CI/CD	29%	38%	49%
We have adopted tools that have improved our productivity and make it easier to improve quality	14%	30%	44%
QA's role has expanded into more aspects of the SDLC	21%	32%	32%

● SW Quality Laggards   ● SW Quality Mainstreamers   ● SW Quality Leaders

Software quality initiatives in place

Improving software quality is not just a byproduct of making changes elsewhere in the enterprise, like expanding a DevOps practice, working more closely with customers, or collecting more metrics. Most enterprises elect to take a more proactive stance on software quality and implement initiatives with the expectation of seeing improvements.

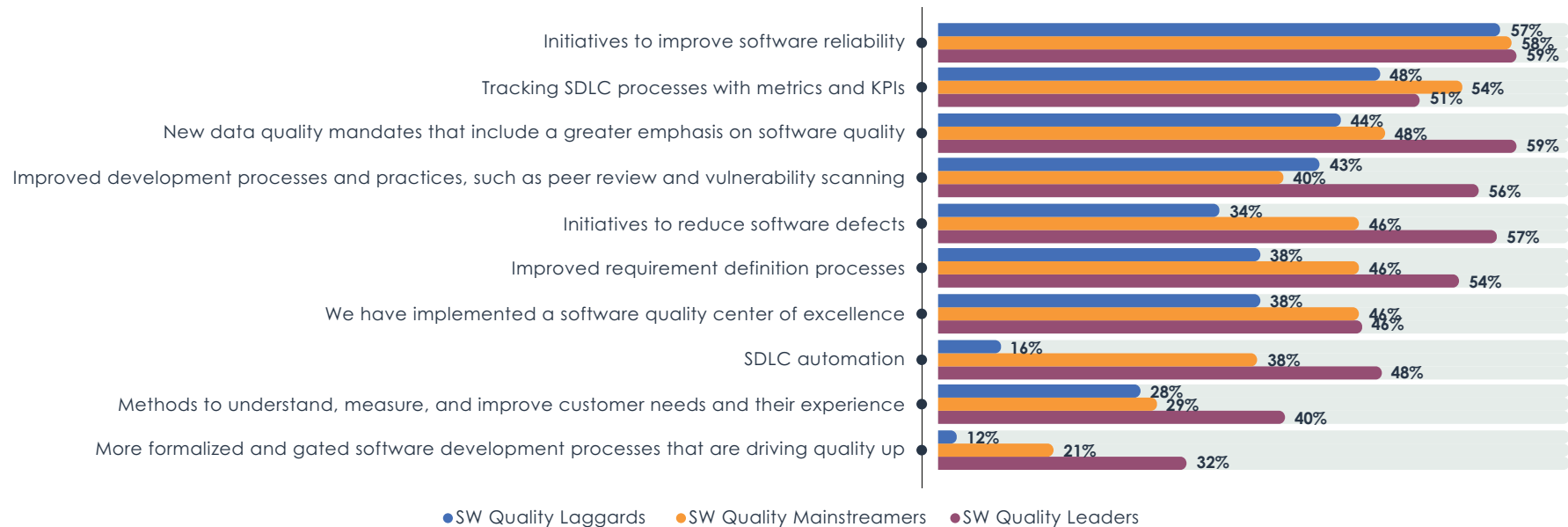The leading initiatives based on the overall percentage of enterprise involvement include initiatives to improve software reliability (58%), tracking SDLC processes with metrics (51%), new data quality mandates (49%), improved development practices like peer review (46%), initiatives to reduce software defects (45%), improved requirements (45%), and the implementation of a software quality CoE (42%). This list includes 70% of the possible responses to this question and all of them showed enterprise involvement above 40%.

Software quality leaders also show 40% or more involvement across 90% of these responses, compared to 70% involvement for mainstreamers and 40% involvement for laggards.

Recurring software quality themes identified in this question include an emphasis on software reliability, the importance of metrics, a focus on peer programming, improved requirements, and a software quality CoE. The fact that enterprises consistently keep identifying these activities indicates they are important drivers of software quality.

What types of software quality initiatives are in place? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Initiatives to improve software reliability — Laggards 57%, Mainstreamers 58%, Leaders 59%

Tracking SDLC processes with metrics and KPIs — Laggards 48%, Mainstreamers 54%, Leaders 51%

New data quality mandates that include a greater emphasis on software quality — Laggards 44%, Mainstreamers 48%, Leaders 59%

Improved development processes and practices, such as peer review and vulnerability scanning — Laggards 43%, Mainstreamers 40%, Leaders 56%

Initiatives to reduce software defects — Laggards 34%, Mainstreamers 46%, Leaders 57%

Improved requirement definition processes — Laggards 38%, Mainstreamers 46%, Leaders 54%

We have implemented a software quality center of excellence — Laggards 38%, Mainstreamers 46%, Leaders 46%

SDLC automation — Laggards 16%, Mainstreamers 38%, Leaders 48%

Methods to understand, measure, and improve customer needs and their experience — Laggards 28%, Mainstreamers 29%, Leaders 40%

More formalized and gated software development processes that are driving quality up — Laggards 12%, Mainstreamers 21%, Leaders 32%

● SW Quality Laggards ● SW Quality Mainstreamers ● SW Quality Leaders

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
Laggards (5-7), Mainstreamers (8), Leaders (9-10)

"Aha!" moments on the software quality journey

Over the past two years, enterprises have been reacting to digital transformation needs—first in a pre-COVID-19 way during 2019 and then fully impacted by COVID-19 during 2020. This question was designed to capture enterprise experience both before and during the pandemic to avoid any bias.

Overall, enterprises align most with the top three responses: improving software quality requires a top-down commitment (46%), senior management is most interested in managing timelines and cost rather than software quality (41%), and shifting left
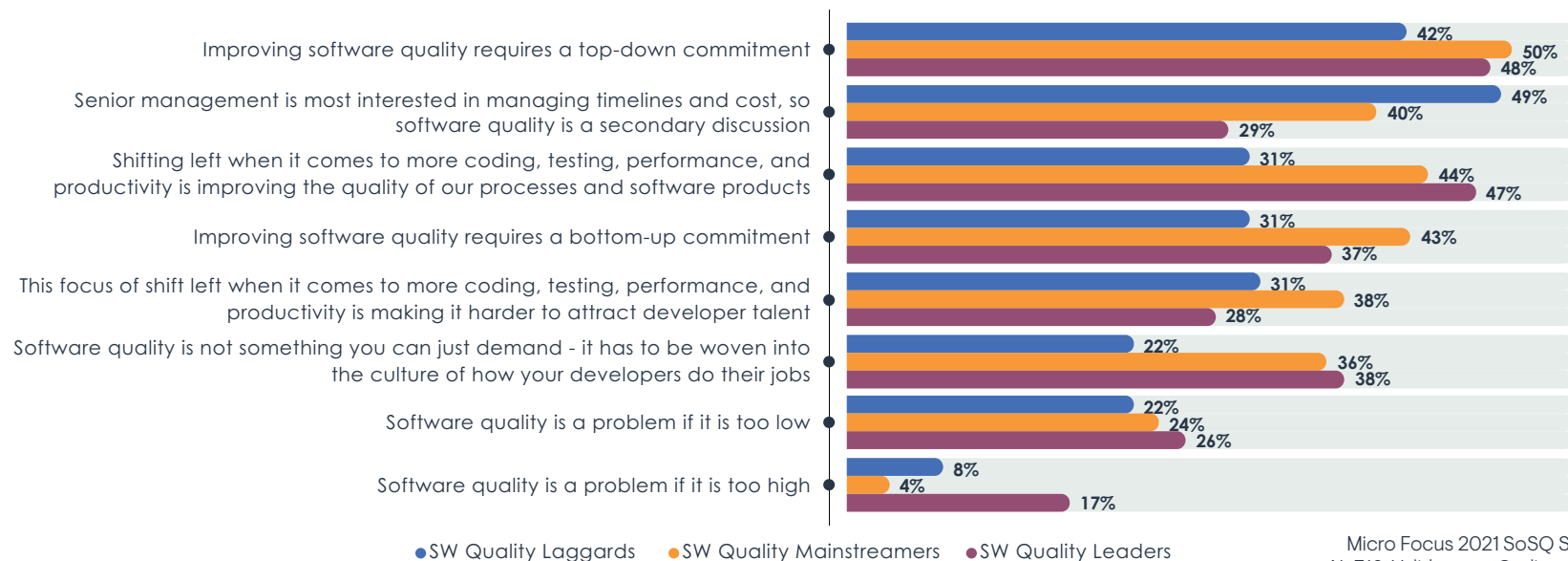
improves the quality of processes and products (40%).

A top-down commitment by C-level management is necessary to emphasize the importance of an initiative like software quality. This is a strong reason why software CoEs are popular. What's striking is how the importance of top-down commitment dovetails with the priority that senior management attaches to timelines and costs, with software quality being a secondary concern. However, the emphasis on timelines and cost exists to communicate that the enterprise must address multiple

competing priorities. This is designed not to dismiss the importance of software quality, but instead to provide a context for making informed decisions about quality, cost, and time.

Shifting left is mentioned again, but with an explicit focus on testing, performance, and productivity. Testing, both functional and performance, are important to address early in the lifecycle—but not arbitrarily early. There is an optimal point in the SDLC for each of the many different types of software tests.

**What are the most important realizations that you have had about software quality over the last two years? – segmented by –
On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develop?**

	Laggards	Mainstreamers	Leaders
Improving software quality requires a top-down commitment	42%	50%	48%
Senior management is most interested in managing timelines and cost, so software quality is a secondary discussion	49%	40%	29%
Shifting left when it comes to more coding, testing, performance, and productivity is improving the quality of our processes and software products	31%	44%	47%
Improving software quality requires a bottom-up commitment	31%	43%	37%
This focus of shift left when it comes to more coding, testing, performance, and productivity is making it harder to attract developer talent	31%	38%	28%
Software quality is not something you can just demand - it has to be woven into the culture of how your developers do their jobs	22%	36%	38%
Software quality is a problem if it is too low	22%	24%	26%
Software quality is a problem if it is too high	8%	4%	17%

● SW Quality Laggards   ● SW Quality Mainstreamers   ● SW Quality Leaders

Worst software quality practices

Best practices are always exciting to identify because of the wisdom they can impart. However, making mistakes is also an important part of the learning process. The challenge is learning how to make incremental mistakes to contain the blast radius, fail fast, learn, and move forward.
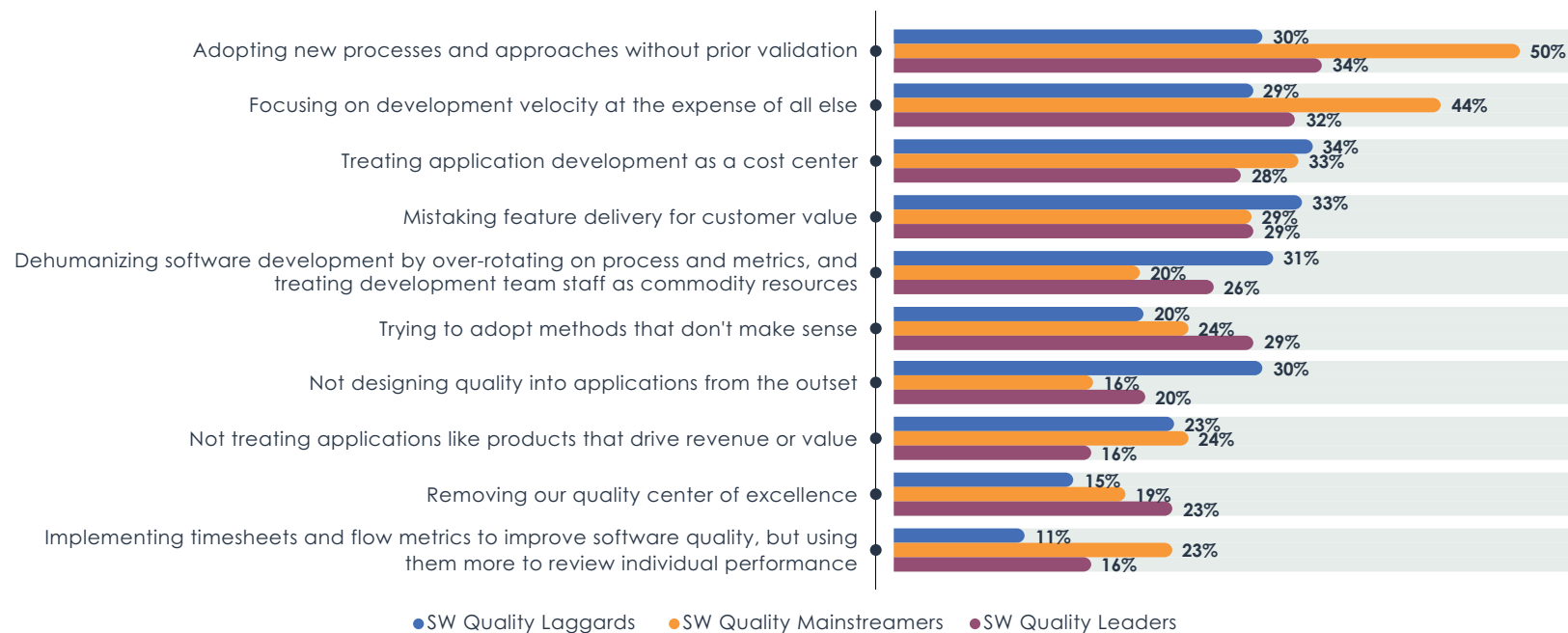
Enterprise involvement on an overall basis across all of these worst practices never exceeded 37%. However, software quality mainstreamers were a significant exception, which likely is a consequence of DevOps growing pains. The top four overall practices to be avoided include adopting new processes without prior validation (37%), focusing on development velocity at the expense of all else (34%), treating application development as a cost center (34%), and mistaking feature delivery for customer value (32%).

Each of these top four worst practices has merit. While the expansion of a DevOps practice has been cited multiple times as the path to improving software quality, there is a dark side to making process changes. Focusing on customer timelines is a software development priority, which must be balanced by also considering quality and cost.[3] Finally, mistaking feature delivery for customer value is a common dilemma best resolved through higher levels of customer communication and collaboration.

What worst practices have you experienced in your software quality journey over the last two years that should be avoided? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Practice	SW Quality Laggards	SW Quality Mainstreamers	SW Quality Leaders
Adopting new processes and approaches without prior validation	30%	50%	34%
Focusing on development velocity at the expense of all else	29%	44%	32%
Treating application development as a cost center	34%	33%	28%
Mistaking feature delivery for customer value	33%	29%	29%
Dehumanizing software development by over-rotating on process and metrics, and treating development team staff as commodity resources	31%	20%	26%
Trying to adopt methods that don't make sense	20%	24%	29%
Not designing quality into applications from the outset	30%	16%	20%
Not treating applications like products that drive revenue or value	23%	24%	16%
Removing our quality center of excellence	15%	19%	23%
Implementing timesheets and flow metrics to improve software quality, but using them more to review individual performance	11%	23%	16%

● SW Quality Laggards   ● SW Quality Mainstreamers   ● SW Quality Leaders

[3] Anyone still treating application development as a cost center needs to read Mik Kersten's book, *Project to Product*.

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
Laggards (5-7), Mainstreamers (8), Leaders (9-10)

Best software quality practices

The leading best practices based on overall vendor experience include the following: improved development practices (such as peer review and peer programming) (47%), adapting a more comprehensive approach to software testing (45%), implementation and use of flow metrics to improve software development and quality (42%), and the use of containers to improve efficiency and velocity (41%).
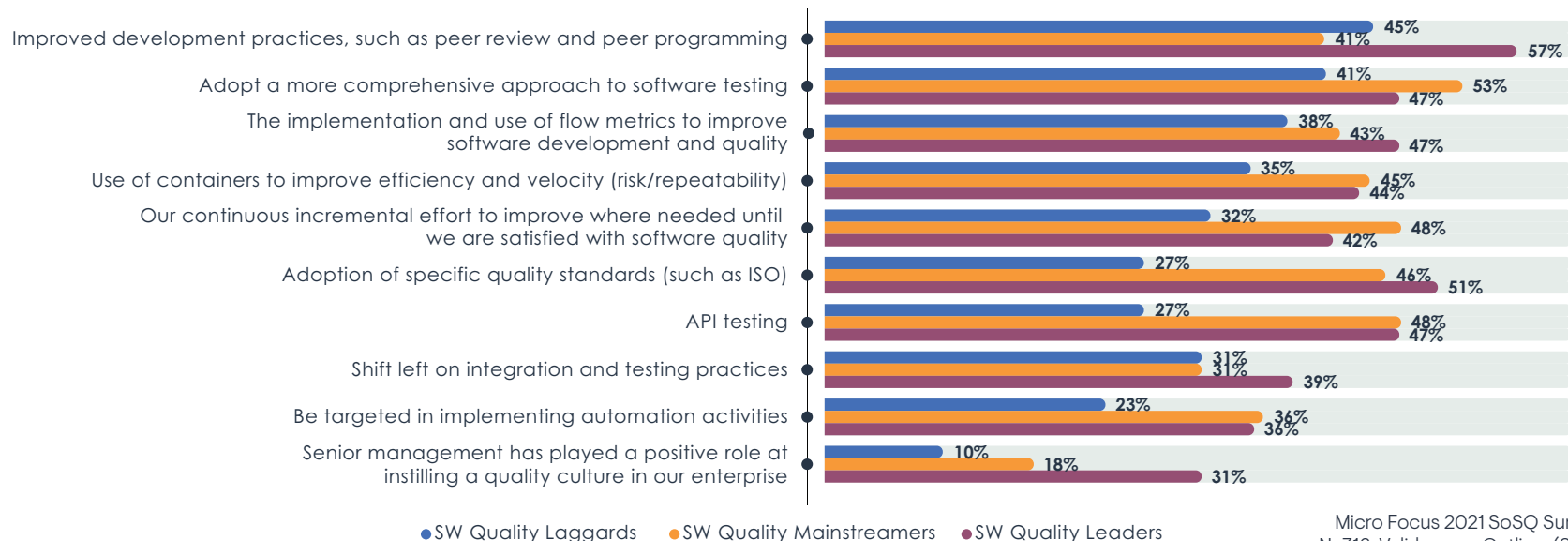
Peer review and peer programming have been common themes throughout this eBook. As the overall leading best practice and the leading best practice for software

quality leaders, peer review and programming are techniques that every agile enterprise should leverage. Adopting a more comprehensive approach to software testing is also a practical and highly effective best practice. Testing across the lifecycle is imperative, and a continuous testing platform is useful to support test automation, test management, and gated CI/CD. Implementation and the use of metrics to improve development and quality are also very strong themes of this eBook. Value stream management provides an opportunity to instrument and manage product

lifecycles. The use of containers to improve efficiency and velocity is making an appearance for the first time in this eBook. However, adoption of containers was investigated in the survey and its maturity is highly correlated with software quality leadership.

A special mention should also be given to the adoption of specific quality standards and API testing. Both of these best practices are important to software quality leaders despite the significant difference of their importance to software quality laggards.

What best practices have you experienced in your software quality journey over the last two years that should be followed? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Best practice	Laggards	Mainstreamers	Leaders
Improved development practices, such as peer review and peer programming	45%	41%	57%
Adopt a more comprehensive approach to software testing	41%	53%	47%
The implementation and use of flow metrics to improve software development and quality	38%	43%	47%
Use of containers to improve efficiency and velocity (risk/repeatability)	35%	45%	44%
Our continuous incremental effort to improve where needed until we are satisfied with software quality	32%	48%	42%
Adoption of specific quality standards (such as ISO)	27%	46%	51%
API testing	27%	48%	47%
Shift left on integration and testing practices	31%	31%	39%
Be targeted in implementing automation activities	23%	36%	36%
Senior management has played a positive role at instilling a quality culture in our enterprise	10%	18%	31%

● SW Quality Laggards ● SW Quality Mainstreamers ● SW Quality Leaders

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
Laggards (5-7), Mainstreamers (8), Leaders (9-10)

The discipline of software quality leaders

High software quality occurs when enterprises make software quality a priority, address software quality across the SDLC, and continually look for ways to improve software quality. Over time, software quality becomes a part of their DevOps culture.

One of the best ways to focus an organization on software quality is to implement a software quality CoE. However, CoE stakeholders and especially C-level executives must be consistent and coherent in their support of software quality, particularly because of competing priorities including customer demands, timelines, and development costs. The CoE provides a framework that enables resources across roles to find and engage in activities that will drive improvements to software quality. This eBook has identified many actions from requirements through the pipeline to production that will help enterprises address software quality.

The most significant finding in this research was the transformation that occurs, enabling an enterprise to become a software quality leader. Three characteristics of software quality leaders stand out:

1. Software quality leaders see software as an important way to drive revenue. Communication and collaboration with customers is paramount so requirements can be well understood, driving the development of software solutions that have the right capabilities and are highly reliable. This connection between business opportunities, software, and revenue is far stronger with software quality leaders compared to followers.

2. Software quality leaders have extensive IT experience and have deep expertise understanding all things IT, including DevOps, cloud, microservices, and containers. Software quality leaders are not always large enterprises with billion-dollar IT budgets. Software quality leaders come in all sizes, from SMBs to large multinational enterprises.

3. Software quality leaders are willing to not let IT get in the way. Software quality leaders are largescale consumers of managed services and SaaS services. Their heavy reliance on leveraging MSPs allows them to remain more flexible and agile when addressing business opportunities. This is very different from software quality followers, who are involved with cloud services but are not strong advocates of MSP outsourcing.

The differing characteristics of software laggards, mainstreamers, and leaders provides insight into how enterprises evolve from one stage of software leadership to the next.

Appendix A

Defining software quality leadership categories

EMA asked enterprises to self-assess the software quality of the products they internally develop. We expected a distribution much like the one seen in this chart, which enabled us to divide the responses into four categories.

Software quality leaders were those enterprises that self-assessed with scores of nine or 10 and accounted for 30% of the sample. Software quality mainstreamers included enterprises with the software quality score of 8 and represented 25% of the sample. Software quality laggards scored between 5 and 7 and included 40% of the sample. Finally, software quality outliers scored between 0-4 and represented only 5% of the sample. Software quality outliers accounted for only 16 responses in the sample, which made this segment too small to be displayed as a category in this analysis.

Across the entire sample, the mean software quality score was 7.5.

Average Software Quality Score
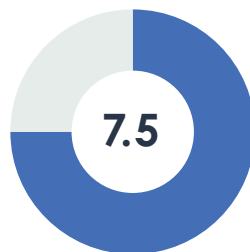
7.5

On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

Software Quality (0=poor, 10=high)

0	0%
1	0%
2	1%
3	1%
4	2%
5	7%
6	13%
7	20%
8	25%
9	17%
10	13%
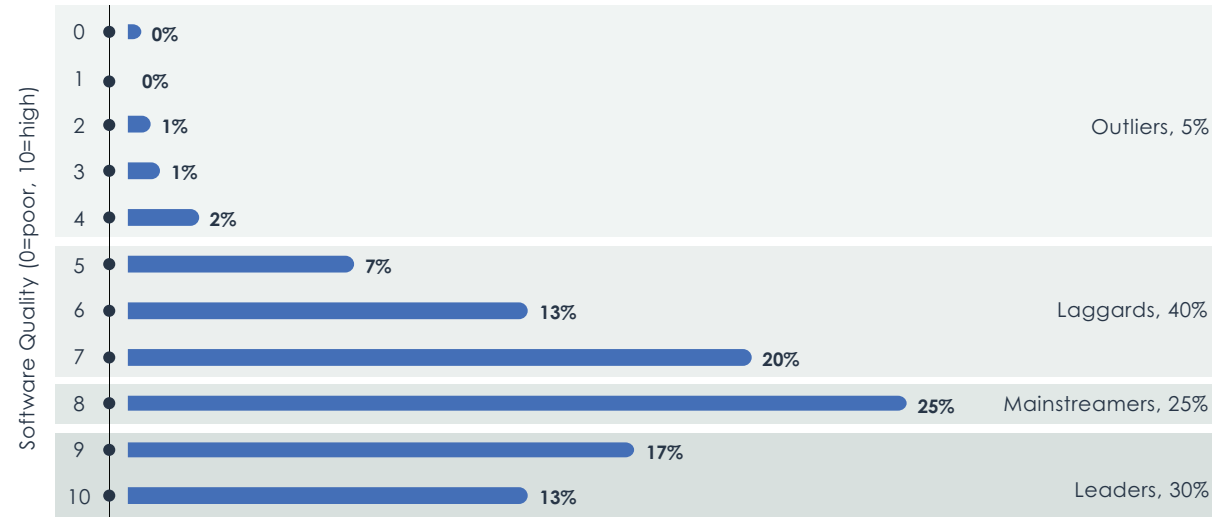
Outliers, 5%

Laggards, 40%
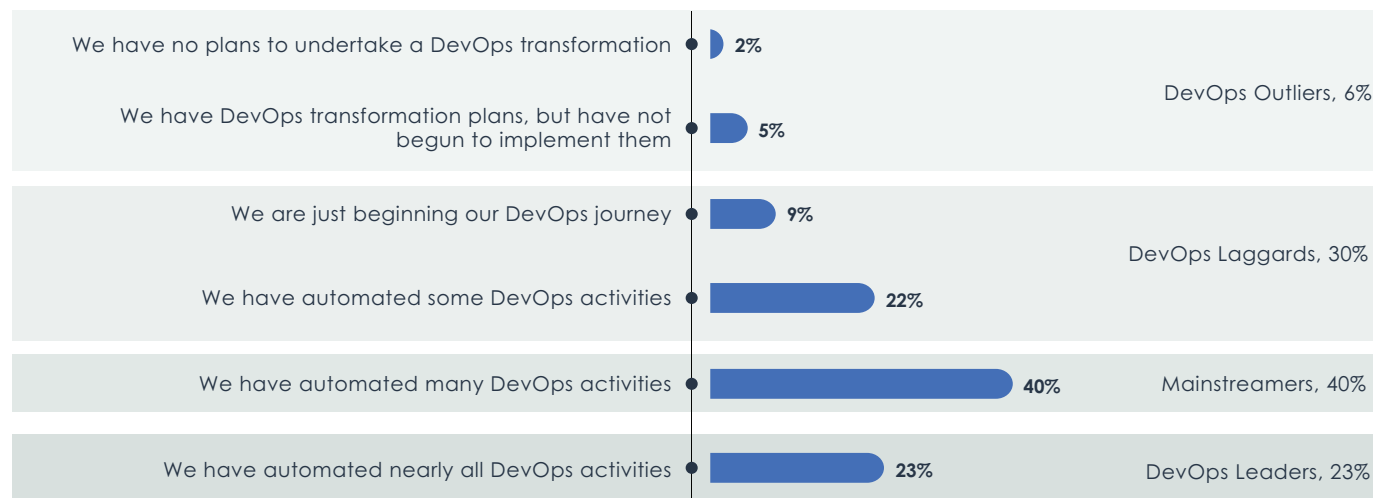
Mainstreamers, 25%

Leaders, 30%

Defining DevOps leadership categories

This survey also asked enterprises to self-assess their DevOps maturity. The distribution of enterprises by their approach to DevOps is similar to the software quality distribution. This is not surprising, and there is a strong correlation between the two variables.

DevOps leaders are enterprises that have automated nearly all of their DevOps activities, and this include 23% of the sample. DevOps mainstreamers are enterprises that have automated many DevOps activities and included 40% of the sample. DevOps laggards are enterprises that are either just beginning their DevOps journey or have automated some DevOps activities

and account for 30% of the sample. Finally, DevOps outliers are enterprises that have either no plans to leverage DevOps or have DevOps plans, but have yet to implement them, and represent just 6% of the sample. Data describing the performance of DevOps outliers is not displayed in any of the charts in this eBook due to small segment size concerns.

How would you describe your enterprise's approach to DevOps?

We have no plans to undertake a DevOps transformation	2%	
We have DevOps transformation plans, but have not begun to implement them	5%	DevOps Outliers, 6%
We are just beginning our DevOps journey	9%	
We have automated some DevOps activities	22%	DevOps Laggards, 30%
We have automated many DevOps activities	40%	Mainstreamers, 40%
We have automated nearly all DevOps activities	23%	DevOps Leaders, 23%

Appendix B

Demographics

Role

App Dev	**31%**
IT Operations	**22%**
CTO, CIO, CDO	**47%**

Geography

N America	**41%**
W Europe	**36%**
Asia Pacific	**23%**

Company Size

2K - 5K Emp	**31%**
5K - 10K Emp	**32%**
10K - 20K+ Emp	**37%**

IT Budget

$1M - $50M	**27%**
$50M - $500M	**41%**
$500M - $5B	**32%**

Vertical Industries

HC/LS	**19%**
Fin Svcs	**15%**
Retail/WS	**15%**
Manufacturing	**12%**
Telco	**11%**
Media	**5%**
Other	**23%**

Which of the following best or most closely describes your primary job role or title within your organization?
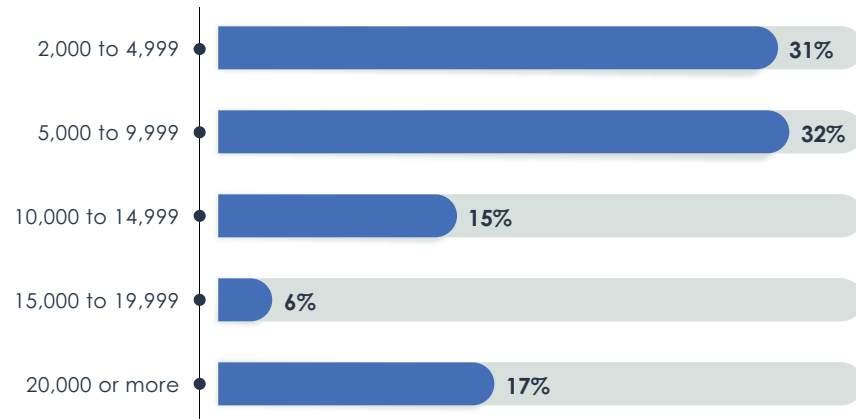
Role	
Application Development - Director or Vice President	31%
IT Operations - Director or Vice President	20%
QA - Director or Vice President	2%
Chief Data Officer	6%
CTO	19%
CIO	23%
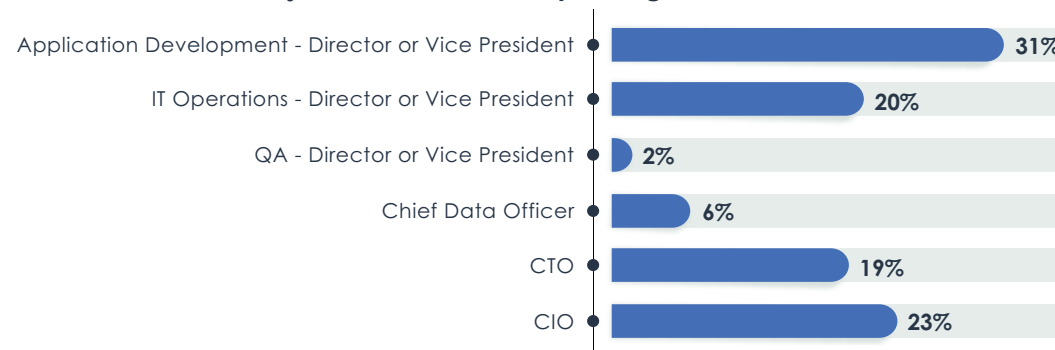
Micro Focus 2021 SoSQ Survey, March 2021
N=316, Valid cases

What country do you live in?

Country	Percentage
United States	41%
Canada	
UK	10%
Germany	10%
France	16%
Australia	10%
India	9%
Singapore	4%
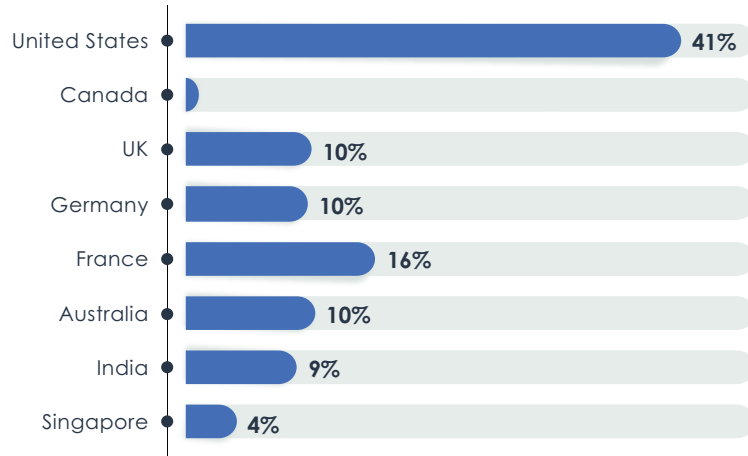
Please estimate how many total employees your organization has worldwide

Employees	Percentage
2,000 to 4,999	31%
5,000 to 9,999	32%
10,000 to 14,999	15%
15,000 to 19,999	6%
20,000 or more	17%

Micro Focus 2021 SoSQ Survey
N=316, Valid cases

What is your organization's annual IT budget for 2020? – segmented by – On a scale of 0 to 10, how would you rank the software quality of the products your enterprise currently develops?

IT Budget	SW Quality Laggards	SW Quality Mainstreamers	SW Quality Leaders
Less than $1 million	1%	3%	
$1 million to less than $5 million	3%	4%	3%
$5 million to less than $10 million	6%	5%	8%
$10 million to less than $25 million	9%	8%	5%
$25 million to less than $50 million	11%	10%	9%
$50 million to less than $100 million	9%	10%	14%
$100 million to less than $250 million	16%	14%	4%
$250 million to less than $500 million	17%	16%	18%
$500 million to less than $1 billion	16%	20%	20%
$1 billion or more	10%	11%	18%

● SW Quality Laggards   ● SW Quality Mainstreamers   ● SW Quality Leaders

Micro Focus 2021 SoSQ Survey
N=316, Valid cases, Outliers (0-4),
Laggards (5-7), Mainstreamers (8), Leaders (9-10)

The 2021 State of Software Quality: A View From the Enterprise   . 27
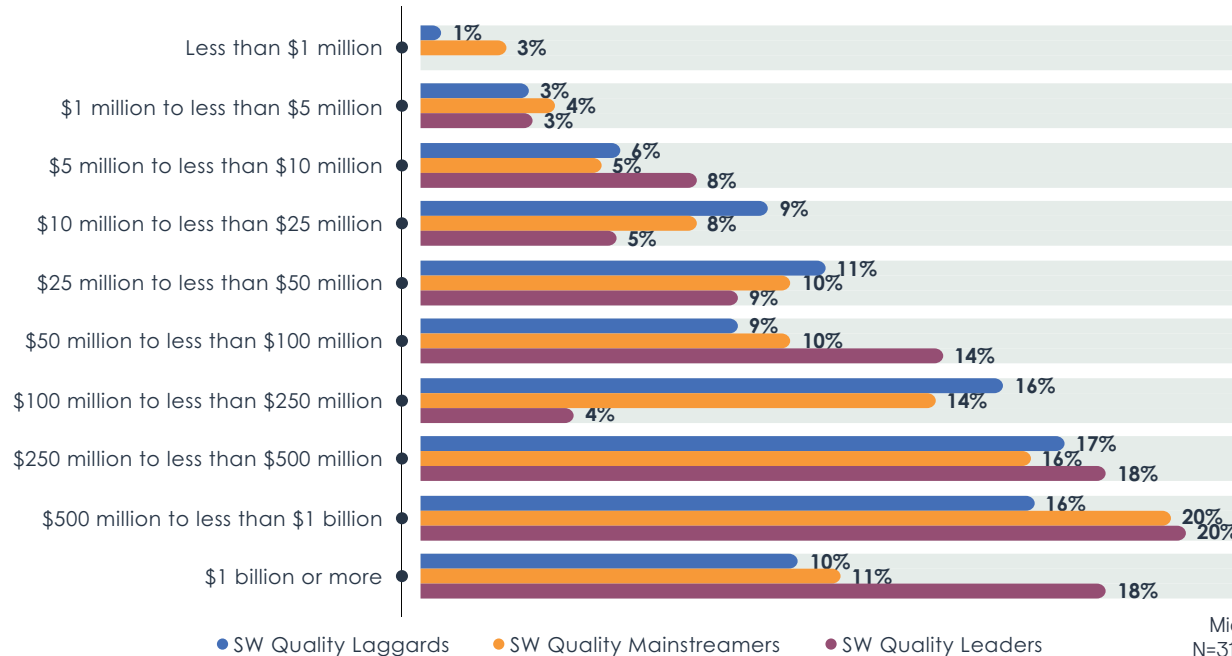
What is your organization's primary industry?

Industry	Percentage
Financial Services (Banking/Insurance/Securities)	15%
Retail	14%
Manufacturing	12%
Business Services (accounting, consulting, legal, etc.)	11%
Telecommunications/ISP/Web Hosting	11%
Healthcare	7%
Life Sciences (biotech, pharmaceuticals, etc.)	7%
Media (broadcast communications, entertainment, publishing, web site, social networking, etc.)	5%
Government (State/Province/County/Municipal/other local government)	4%
Utilities	4%
Transportation & Logistics	4%
Other	3%
Consumer Packaged Goods	2%
Wholesale	1%
Government (Federal/National)	1%
Oil & Gas	0%