voke Research

MARKET SNAPSHOT™ REPORT:

Service Virtualization

By Theresa Lanowitz, Lisa Dronzek | January 21, 2015

voke®
Moving markets beyond the status quo!

MARKET SNAPSHOT™ REPORT:

Service Virtualization

By Theresa Lanowitz, Lisa Dronzek | January 21, 2015

SUMMARY

From a benefits perspective, lifecycle virtualization technologies deliver a quick and measurable economic impact, just as server virtualization provides for the datacenter. Lifecycle virtualization includes the technology of service virtualization.

Service virtualization enables development and test teams to statefully simulate and model their dependencies of unavailable or limited services and data that cannot be easily virtualized by conventional server or hardware virtualization means. Service virtualization removes the constrains and wait times frequently experienced by development and test teams needing to access components, architectures, databases, mainframes, mobile platforms, and so on.

Service virtualization technology is critical to the success of a vibrant application economy. Organizations using service virtualization experience fewer defects, reduced software cycles, and increased customer satisfaction.

This Market Snapshot report provides real-world data to help organizations justify the investment in service virtualization.

TABLE OF CONTENTS

Executive Overview 2

Market Overview 3

Market Evolution 3

- Vendor Overview 4

Survey Methodology 5

- Survey Objectives 5
- Participant Demographics 6

Market Snapshot 9

- The Need for Service Virtualization 9
- Use of Service Virtualization 13
- Benefits and Challenges 19
- Service Virtualization ROI 24

Net / Net 31

Appendix: Related Research 33

EXECUTIVE OVERVIEW

From August 2014 through October 2014, voke conducted an independent survey of 505 participants from both technology and non-technology companies of varying sizes. We explored their use of service virtualization and the results they experienced. This Market Snapshot report identifies the data we gathered and provides our analysis of the participants' responses about how service virtualization is used, why the technology is adopted, as well as its general perceptions, challenges, and benefits. This report is voke's second Market Snapshot[1] on service virtualization; the first was published in 2012.

The use of virtualization in the pre-production portion of the software lifecycle is defined by voke as lifecycle virtualization.[2] Lifecycle virtualization, in pre-production, allows overutilized or excessively needed resources to be virtualized and shared across the entire software supply chain.

From a benefits perspective, lifecycle virtualization technologies deliver a quick and measurable economic impact, just as server virtualization provides for the datacenter. Lifecycle virtualization includes the technology of service virtualization.

Service virtualization enables development and test teams to statefully[3] simulate and model their dependencies of unavailable or limited services and data that cannot be easily virtualized by conventional server or hardware virtualization means. Service virtualization removes constrains and wait times frequently experienced by development and test teams needing to access components, architectures, databases, mainframes, mobile platforms, and so on.

Service virtualization technology is critical to the success of a vibrant application economy. Organizations using service virtualization experience reduced defects, reduced software cycles, and increased customer satisfaction.

This Market Snapshot report provides real-world data to help organizations justify the investment in service virtualization.

[1] *voke Market Snapshot™ Report: Service Virtualization* – December 11, 2012
[2] *voke Category Snapshot™ Report: Lifecycle Virtualization* – November 7, 2011
[3] A stateful component means that the virtualized asset is capable of storing the interaction and triggering the interaction with other components or virtualized assets. See section Mocks and Stubs vs. Service Virtualization, page 23.

MARKET OVERVIEW

Organizations of all types and sizes will benefit from the use of service virtualization. voke predicts that lifecycle virtualization will be the hub of the modern software lifecycle. With virtualization at the hub of the software lifecycle, much needed collaboration increases among development, quality assurance (QA), and operations. All teams must work to support the line of business and share the common goal of shipping software that will deliver valuable business outcomes.

Service virtualization adoption is on the upswing, but has yet to attain critical mass. To achieve the levels of server and desktop virtualization, vendors must invest in marketing and promote increased market awareness of service virtualization. Service virtualization adoption will be driven by the pervasiveness of complex software supply chains working together seamlessly to deliver business benefit.

Success with service virtualization comes when teams are united and collaborative, leaders are supportive of the technology, and the benefits delivered across the software lifecycle are visible. Service virtualization ultimately helps the entire software project by solving classic computing problems that teams have struggled with for years.

Service virtualization is a proven technology that provides a tremendously positive impact for organizations that adopt, use, and implement it.

MARKET EVOLUTION

voke's research report[4] on lifecycle virtualization, published in 2011, defined lifecycle virtualization and identified service virtualization as part of the broader category. Now, in 2015, we are seeing the three most commonly used and adopted types of lifecycle virtualization as:

- Service virtualization
- Virtual and cloud-based labs[5]
- Network virtualization[6]

---

[4] *voke Category Snapshot™ Report: Lifecycle Virtualization* – November 7, 2011

[5] Virtual and cloud-based labs are defined as the creation of a virtual instance of a physical environment, either on-premise or in a cloud, to enable anyone to access a virtual equivalent of any environment anywhere, on-demand without the need to wait for a physical machine setup or additional hardware. See *voke Market Snapshot™ Report: Virtual and Cloud-based Labs* – August 21, 2014.

[6] Network virtualization is defined as enabling pre-production teams to model and simulate a network as close to production as possible for the purpose of assessing the impact of the network on the user experience and optimizing network parameters. The goals of network virtualization are to ensure that the infrastructure is not a bottleneck and make sure that anything added to the infrastructure does not negatively impact the network or other connected elements.

We are beginning to see more market awareness of service virtualization than other technologies in the lifecycle virtualization category. Depending on the nature of their architectures and platforms, organizations may need all forms of lifecycle virtualization, including:

- Service virtualization to statefully simulate services, components, data, or applications regardless of the level of completeness or accessibility
- Virtual or cloud-based labs to enable on-demand access to development and test environments
- Network virtualization to model and simulate networks to assess the impact of the network on the user experience

The service virtualization market initially solidified after acquisitions by CA and IBM along with innovation from HP and Parasoft. A new vendor, Tricentis, emerged in 2014 after the release of our 2012 Market Snapshot report. The addition of a single new vendor is the only significant change in technology vendors within a two-year period. Currently, Microsoft is the only major application lifecycle management (ALM) vendor without a service virtualization offering.

We should expect the market to continue to evolve and see service virtualization technology integrating with other such lifecycle solutions as automated release management, development testing, performance testing, virtual and cloud-based labs, and defect virtualization.

⊙ VENDOR OVERVIEW

Today the service virtualization vendor landscape consists of vendors with service virtualization products along with other complementary lifecycle automation solutions. The acquisition of ITKO by CA in 2011 legitimized the market and made it a necessity for any vendor serious about the software lifecycle to offer a service virtualization solution. Before selecting any service virtualization product, it is strongly advised to conduct a proof of concept (POC) with more than one vendor.

Vendor Name	Product Name	Background
CA	CA Service Virtualization (formerly ITKO LISA)	• CA acquired ITKO in August 2011. • ITKO introduced service virtualization in its ITKO LISA product in 2007.
HP	HP Service Virtualization	• HP introduced its HP Service Virtualization product in July 2011.
IBM	IBM Rational Test Virtualization Server (formerly GH VIE)	• IBM acquired Green Hat in January 2012. • Green Hat introduced its Virtual Integration Environment (VIE) product in 2011.
Parasoft	Parasoft Virtualize	• Parasoft introduced its Parasoft Virtualize product in 2011, which was previously part of Parasoft SOAtest.
Tricentis	Tosca Virtualize	• Tricentis introduced its Tosca Virtualize product in 2014.

SURVEY METHODOLOGY

The voke survey was conducted from August 2014 to October 2014 using online interviews and in-person telephone interviews with 505 participants from diverse organizations, market segments, geographies, and roles. All survey participation is anonymous and responses are confidential. Participant responses are given in the aggregate without company or individual names identified.

We asked a variety of question types in the survey, including multiple choice, check all that apply, and open-ended. Please note that data in tables not adding up to 100% resulted from a check all that apply question type.

voke thanks all of our survey participants for their time and insight. Please visit www.vokeinc.com if you would like to share your input in current or future surveys.

⊙ SURVEY OBJECTIVES

The primary goal of this survey is to gain real-world insights about service virtualization including its need, use, adoption, benefits, challenges, and return on investment (ROI). Additionally, we sought to identify new insight and market changes since our first survey on this topic in 2012.

This research focuses on the following primary areas:

- Demographics
- Needs
- Uses
- Challenges and benefits
- ROI

Overall, the research focuses on service virtualization and its benefit to delivering more predictable, effective, and efficient business outcomes through software.

⊙ PARTICIPANT DEMOGRAPHICS

The demographic questions in the survey characterized the participants in terms of company size, geographic location, and roles.

The following tables and charts, beginning with vertical segments, summarize the demographic data on survey participants.

Technology-based organizations represent 51% of the survey participants, and 49% of the participants were in a variety of segments from non-technology industries. This almost even split of participants from technology-based and non-technology-based organizations is in contrast to the 2012 voke Market Snapshot. In our 2012 research, technology-based participants were 60% while non-technology-based participants were 40%.

This increase in non-technology-based participants is indicative of service virtualization becoming more widely adopted in enterprise IT organizations.

Vertical Segment	Participant Percentages
TECH: Software	26.6%
TECH: Service Provider/Professional Services	11.9%
TECH: Telecommunications	5.0%
TECH: Wholesale/Retail/Distributor	3.8%
TECH: E-commerce/Internet	2.5%
TECH: Networking	0.9%
TECH: Hardware	0.3%
NONTECH: Finance/Banking/Accounting	21.0%
NONTECH: Retail/Wholesale/Distributor	7.8%

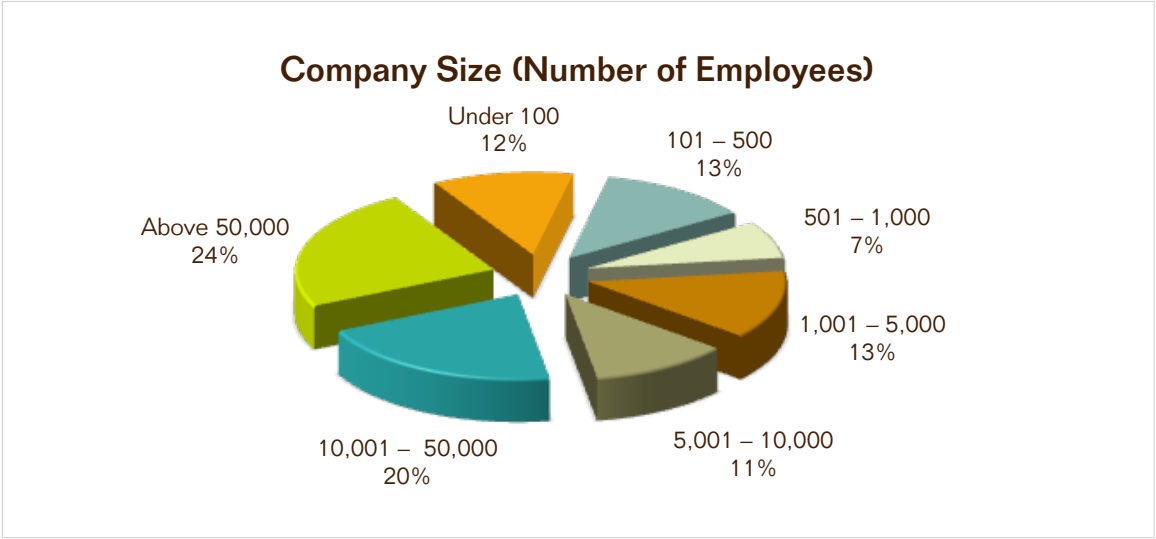
Vertical Segment (continued)	Participant Percentages
NONTECH: Healthcare/Medical/Pharmaceutical/Bio-tech	3.8%
NONTECH: Insurance/Real Estate/Legal	3.8%
NONTECH: Travel/Hospitality/Entertainment/Recreation	2.5%
NONTECH: Transportation/Utilities	2.2%
NONTECH: Education	1.6%
NONTECH: Government/Military	1.6%
NONTECH: Media	1.6%
NONTECH: Manufacturing	1.3%
NONTECH: Advertising/Marketing	0.9%
NONTECH: Aerospace/Defense Contractor	0.6%
NONTECH: Business Services/Consultant	0.3%

Survey participants were geographically dispersed, with 58% working in the United States and 42% working in other geographies.

Geography	Company Headquarters	Primary Work Location
United States	63%	58%
Canada	3%	3%
South / Central America	3%	4%
Europe (Including UK)	24%	24%
India	3%	5%
Rest of Asia (excluding India)	1%	1%
Australia / New Zealand	2%	3%
Middle East	1%	1%
South Africa	0%	1%

Survey participants came from a diverse range of company sizes, with 25% from smaller organizations under 500 employees, 31% ranging from over 500 up to 10,000 employees, and 44% from larger organizations with more than 10,000 employees.

This diversity in company size is indicative of the value that service virtualization delivers to organizations of all sizes.



Additionally, participants identified the following company classifications. Please note, this table represents data from a check all that apply question type, therefore, company classification is not mutually exclusive.

Company Classification	Participant Percentages
Small business	14%
Startup	4%
Privately held	25%
Publicly traded	41%
Fortune 100	13%
Fortune 500	24%
Fortune 1000	7%
Forbes Global 2000	5%
Government	5%
Not for profit	3%
Education	1%

Survey participants also represented a broad distribution of employee job levels ranging from C-level executives (e.g., CIO, CEO, CTO) to practitioners across all roles of the software lifecycle.

The distribution of employee job levels is in stark contrast to our 2012 research where 53% of survey participants were individual contributors.

In 2014, 36% of survey participants identified their job levels as director and above. This is indicative of the awareness of service virtualization by those job levels higher in the organization. Service virtualization adoption and implementation is more successful if it has visibility and acceptance by leadership of the organization.

Job Level	Participant Percentages
Director and above	36%
Manager	27%
Lead	18%
Individual Contributor	19%

We found survey participants across key functional areas with 40% of participants working in QA organizations.

By comparison, our 2012 research identified 56% of its participants working in a QA role. This wider distribution across various functional roles in 2014 shows how service virtualization is expanding its reach and usage throughout the lifecycle.

Functional Role	Participant Percentages
QA	40%
Development	19%
Line of Business	17%
Operations	14%
Professional Services / Consulting	10%

MARKET SNAPSHOT

With these demographically diverse participants, the research examined the need, use, benefits, and ROI of service virtualization.

⊙ THE NEED FOR SERVICE VIRTUALIZATION

Survey participants provided their input about the impact of various challenges faced by their organizations.

Work Delayed While Waiting for Dependencies

We asked survey participants if their developers or QA engineers experienced delays in waiting for a service, component, application, API, user interface, or dataset during their current or last release cycles.

We found that 81% of participants identified development delays of waiting for a dependency in order to develop software, reproduce a defect, or fix a defect. Additionally, 84% of participants identified QA delays of waiting for a dependency in order to begin testing, start a new test cycle, test a required platform, or to verify a defect. These numbers suggest that delays in waiting for dependent services, components, applications, APIs, user interfaces, or datasets are equally troublesome and disruptive to both development and QA teams. Think about your own organization and the cost, quality, and schedule impacts that occur while waiting for dependencies across your teams, organization, software supply chain, and third parties.

Impact of Waiting	Participant Percentages
Development work delayed	81%
QA work delayed	84%

Anecdotally, participants reported the following reasons for delays **before** the use of service virtualization:

- A protracted timeframe to access unstable middleware environments
- The inability to develop a shared service across disparate organizations and third-parties
- The inability to control data for testing purposes

Participants noted that the types of delays identified above occurred before implementing a service virtualization solution. These delays were subsequently resolved with the use of a service virtualization solution. Anecdotally, one participant identified 90% virtualization of all dependencies resulting in increased productivity.

Think about your own organization and the opportunity to positively impact productivity through the virtualization of dependencies.

Access to Required Dependencies

For the purposes of this research, a dependency is defined as a required element used by development and testing for pre-production purposes. Examples of dependencies include, but are not limited to:

- Services
- Components
- Applications
- APIs
- User interfaces
- Datasets

We asked survey participants to characterize the total number of dependencies required and the total number of dependencies that are available with unrestricted access.

Dependencies	Needed	Available
None	0%	9%
1 to 5	15%	33%
6 to 10	15%	25%
11 to 15	8%	11%
16 to 20	7%	5%
21 to 30	14%	3%
31 to 40	13%	5%
41 to 50	11%	3%
51 to 60	3%	1%
61 to 70	1%	0%
71 to 80	4%	0.5%
81 to 90	0%	0%
91 to 100	2%	1%
101 to 175	3%	2%
200 to 400	3%	1%
500 to 2000	1%	0.5%

On average, participants require access to 52 dependent elements for development or testing. On average, survey participants reported having unrestricted access to only 23 of the 52 dependent elements needed for development and testing. A majority of participants, 67%, report unrestricted access to only 10 or fewer dependent elements.

Our 2012 research revealed that participants needed access to 33 elements for development or testing and had unrestricted access to only 18. This number is in stark contrast with the 2014 numbers of 52 elements required and 23 available. The reasons for an increase in the number of required elements for development or testing include:

- Greater application complexity
- An increase in the number of mobile dependencies
- An expansion in the size and scope of the software supply chain
- A larger need for integration of all types of applications including legacy
- More composite applications
- A higher demand for testing earlier in the software lifecycle

voke data shows an increase in the number of dependencies required between 2012 and 2014. Yet, the most significant increase is the discrepancy between what is available and what is needed. While the number of dependent systems has increased, the percentage gap between unavailable systems and what is required has widened. With an increase in the number of dependencies comes an even greater need to conduct testing earlier and focus on end-to-end integration. voke predicts the number of dependencies to continue on an upward trend.

On the large scale, one participant reported the maximum number of dependencies needed is 2,000 and the maximum number of dependent elements reported available is 1,000.

Think about your own organization and if you have unrestricted access to everything required for development or testing. Most organizations experience a deficit of unrestricted access to elements needed for development or testing. The result of this deficit is that quality, cost, and schedule are severely impacted. Many organizations are unaware of the hidden cost of rework[7] that the lack of unrestricted access to required elements creates for development or testing. Every organization should know its cost of rework and the consequences.

Third-Party Access Challenges

A total of 79% of participants reported third-party restrictions, time limits, or fees for the services or applications needed for development and testing.

Participants reported accessibility costs, schedule impacts, and restricted access as the primary constraints encountered when needing to access third-party dependencies. The following lists are indicative of the primary factors identified for constrained access to third-party dependencies.

Accessibility Costs

- Additional fees beyond a basic number of records to test for performance and integration
- Costs for third-party reports that are required for end-to-end integration testing
- Fees for each transaction conducted on a test system each time the test is run
- Fees for mainframe time to test composite applications
- Infrastructure costs to access such dependencies as cloud storage, databases, and web hosting

[7] *voke Strategic Brief™ Report: Reducing the Cost of Rework* – October 21, 2014

Schedule Impacts

- Third-party interface changes impacting overall development schedule
- Time limits imposed on access to third-party dependencies
- Software supply chain not synchronized when accessibility to dependencies is granted
- Integration bottlenecks across the supply chain due to limited third-party dependency access
- Coordination of testing activities across the supply chain
- Insufficient or incomplete test data
- Manual provisioning and scheduling
- Batch processing of dependent components

Restricted Access

- Limited or no access to third-party test environments
- Lack of licenses for third-party components
- Reduced testing due to lack of test systems for third-party software
- Inability to conduct load testing on third-party APIs
- Limited or no preproduction test environments due to cost restrictions
- Test access limited to overnight batch runs
- Performance testing windows must be scheduled with third-party providers
- Inadequate time for performance testing with third-party assets
- Fewer transactions tested without an integrated end-to-end test across the software supply chain
- Limited third-party mainframe access

Think about your own organization and the macro scope of dependent elements and third-party dependencies needed across all projects. Are you experiencing schedule delays, cost overruns, and incomplete testing due to the lack of unrestricted access to necessary third-party dependencies? If the answer is yes, investigate service virtualization.

⊙ USE OF SERVICE VIRTUALIZATION

Of the total survey participants, 88% told us about their experiences with service virtualization solutions. We asked those participants about the adoption status of service virtualization in their organizations.

In our 2012 research, 44% of survey participants identified "currently using solutions." In 2014, the number of survey participants identifying "currently using solutions" increased to 57%. This change speaks to an increase in the awareness, adoption, and use of service virtualization.

Status of Adoption	Participant Percentages
Evaluating solution(s)	17%
Completed evaluation and preparing to implement	16%
Currently using solution(s)	57%
Currently using solution(s), but evaluating other options	7%
Currently using solution(s), but in the process of implementing another solution(s)	3%

We also see a broad mix in terms of its length of use, with the majority, 64%, using service virtualization technology for two years or less.

In 2012 our research showed that 55% of participants were using service virtualization for two years or less. Again, this increase to 64% in 2014 suggests a greater awareness, adoption, and use of service virtualization.

Length of Use	Participant Percentages
Less than 1 year	32%
Between 1 to 2 years	32%
Between 2 to 3 years	22%
Between 3 to 4 years	6%
Between 4 to 5 years	4%
More than 5 years	4%

Furthermore, we asked participants to describe how widespread their use of service virtualization is across their organizations, partners, and supply chains. The majority of participant use, 69%, is project-based or departmental. We see that 38% offer enterprise-wide usage or availability through a center of excellence, and 16% extend usage to third-party offshore teams or their software supply chains.

The data represented in this table comes from a check all that apply question. The data shows usage spanning from the smallest scale of a pilot project to the largest scale accessible to an entire software supply chain.

Nature of Use	Participant Percentages
Pilot project	22%
Project-based	38%
Departmental	31%
Available via center of excellence (CoE)	19%
Enterprise-wide	19%
Accessible to third-party offshore teams	10%
Accessible to entire software supply chain including third-party partners and suppliers	6%

Anecdotally, participants indicated that developer buy-in was a challenge to achieving enterprise-wide deployment. Participants cited the primary reasons for developers refusing to adopt service virtualization as unwillingness to change, reluctance to include a new tool, and the belief that using stubs and mocks in lieu of service virtualization is acceptable (see analysis of mocks and stubs vs. service virtualization on page 23).

While there is an increase in service virtualization awareness, adoption and enterprise-wide use is still lagging.

Organizational Insight

We asked participants which group in their organizations discovered or recommended service virtualization technology.

We see that 39% of participants identify the QA/test team as the group discovering or recommending service virtualization. Ideally, all QA/test teams would use service virtualization because the technology allows testing to be done on incomplete or unavailable elements.

Recommended By	Participant Percentages
QA / test	39%
Development / engineering	26%
Center of excellence	9%
IT operations / production deployment	6%
C-level executive	4%
Corporate	4%
IT infrastructure / lab support	4%
Line of business	3%
Sales	2%
Third party professional services / consultant	2%
Support	1%

Similarly, we asked participants which group in their organizations funded the purchase of their service virtualization solutions. We see a broad mix of groups funding service virtualization technology.

Group Funded	Participant Percentages
QA / test	21%
Development / engineering	18.5%
Corporate	16.5%
C-level executive	12.5%
Center of excellence	7.5%
IT operations / production deployment	6.5%
Line of business	6.5%
IT infrastructure / lab support	3%
Project-based funding	3%
Centrally funded	2%
Sales	1.5%
Support	1%
Jointly funded	0.5%

Funding should not be an issue if your organization believes in removing barriers between development, QA, and operations to deploy software to production with less effort and fewer defects. The application economy requires software that works as expected. In order to provide software that delivers valuable business outcomes, organizations must focus on removing budgets allocated for single organizations, such as development or testing, and embrace technology that will enable the common goal of shipping great software.

We also asked which groups and roles used service virtualization technology.

We see that the initial beneficiary of service virtualization is the QA/testing team. However, the use of service virtualization by the QA/testing team for earlier and more comprehensive testing impacts the entire software lifecycle and software supply chain.

Used By	Participant Percentages
QA – software QA or testing (functional, performance, security)	72%
Development – software developers	58%
QA – architects	39%
Development – architects	37%
Release engineering or management	32%
QA – managers	32%
Center of excellence (CoE)	27%
Consultants / professional services	23%
Development – managers	19%
IT – infrastructure	16%
IT – operations / production deployment	14%
IT – lab managers / lab engineers	13%
Training	10%
IT – patch management	6%
IT – system administrator	6%
IT – security	5%
Anyone on-demand	5%
IT – management	4%
Project management	4%
Support	4%
Sales	1%
Technical publications	1%

Types of Assets Virtualized

We asked survey participants to identify the types of assets virtualized through service virtualization. They most frequently cited SOA, web services, and APIs as the types of assets virtualized.

The assets cited as the most frequently virtualized are consistent with the immediate pain points experienced by software project teams. Anecdotally, we see that once the immediate pain point of removing restrictions to necessary dependencies in the software project is resolved, the team quickly finds other elements to virtualize. Service virtualization use expands quickly beyond virtualization of the initially identified assets.

The smaller percentages associated with virtualizing mobile assets suggest that mobile testing is a lower priority and not automated.

Types of Assets Virtualized	Participant Percentages
SOA / web services	71%
APIs	64%
Middleware	38%
Applications – legacy	36%
Mainframe	35%
Data – test data	34%
Data – databases	33%
External software – third-party software or services	33%
Applications – ERP / packaged	29%
Applications – new	29%
External software – partner software or services	27%
Applications – SaaS	17%
Architectures	15%
Lab environments	15%
Data – mobile	12%
Mobile – architectures	11%
External software – entire software supply chain	10%
Mobile – carrier networks	10%
User interfaces	10%
Mobile – devices	9%
Mobile – development platforms	8%
Mobile – user interfaces	7%
Network infrastructure	7%
Data – big data	6%
Operating systems	6%
Embedded systems	5%
Networked elements or appliances	4%
Mobile – operating systems	3%

Rationale for Adoption

We asked participants to indicate the rationale for adopting service virtualization solutions in their organizations.

In our 2012 research, participants cited "restricted access to dependent services, components, or systems" as the primary reason for adoption. Comparing the 2012 response to the 2014 primary reason for adoption of "improving time-to-market" suggests an increase in the understanding of business needs or unrealistic schedule demands.

Rationale for Adoption	Participant Percentages
Improve time-to-market	68%
Test earlier in the lifecycle	54%
Enable continuous integration	52%
Performance testing	48%
Parallel development	47%
Scheduling constraints	40%
Restricted access to dependent services, components, or applications	38%
Test data management	35%
Reduce production defects	34%
Reduce capital expenditures (CAPEX)	32%
Simulation of new software	31%
Reduce operational expenditures (OPEX)	31%
Third-party access fees	28%
Mobile development and testing	22%
Network constraints	18%
Simulation of hardware	14%
Replace an internally developed service virtualization solution	12%

Additionally, participants cited the need for test automation and how service virtualization is used as an enabler for automated regression testing. Automation is a key enabler for meeting the business need of time-to-market. Without automation the balance of project cost, quality, and schedule are adversely impacted.

⊙ BENEFITS AND CHALLENGES

We asked participants to describe the benefits and challenges of service virtualization.

User Benefits

Participants reported the following benefits in the context of schedule, cost, and quality gained from using service virtualization. Overall, participants noted that schedule was more important than cost.

Schedule Benefits Realized	Participant Percentages
Simulate services, APIs, components, or applications prior to availability of production ready assets	58%
Reduce scheduling conflicts of constrained services, applications, or components	53%
Reduce bottlenecks associated with internal and external partners and resources	48%
Improve time-to-market	46%
Reduce business risk of schedule delays	41%
Enable more efficient parallel development	40%
Reduce mainframe dependencies	37%

Another notable schedule benefit was meeting deadlines to external customers.

Cost Benefits Realized	Participant Percentages
Provide unrestricted, on-demand access to services, components, and applications required for development, testing, and integrations	65%
Reduce operational expenditures (OPEX)	44%
Reduce costs associated with third-party access fees	35%
Reduce capital expenditures (CAPEX)	34%
Eliminate the expense of building internal tools	29%

Additionally, mainframe accessibility to support legacy systems was another important cost benefit.

Quality Benefits Realized	Participant Percentages
Enable more testing earlier in the lifecycle	66%
Unrestricted access to realistic test environments	58%
Enable performance testing earlier in the lifecycle	52%
Manage the demand for complex testing scenarios	50%
Improve development and test productivity	49%
Reduce the risk of application failures in production	38%
Enable continuous integration	38%
Improve test data management	36%
Better collaboration across internal teams	30%
Better collaboration across external teams or third parties	26%
Increase team morale	16%

Participants highlighted the ability to start coding in parallel to a dependent third-party service that is concurrently under development. This enables both earlier development and testing of dependent services throughout the software supply chain.

Overall, participants cited improved QA team morale from using service virtualization. Because of the immediate benefits realized through the use of service virtualization, teams are eager to promote the use of service virtualization internally and throughout the software supply chain. Many QA teams act as service virtualization ambassadors and influence change throughout the organization.

Benefits and Challenges of Using Vendor Products

Our research also revealed common participant comments about each vendor in the service virtualization market.

The following table represents the primary vendors, in alphabetical order, along with the respective service virtualization products, as well as the most frequently mentioned survey participant comments about these products.

Vendor	Product Name	Most Frequent Comments
CA Technologies	CA Service Virtualization	• Complete and mature solution • Flexible and extensible • Integration with a variety of lifecycle tools • Easy to use and implement • Quick ROI • Costly
HP	HP Service Virtualization	• Integration with existing HP testing tools • Integration with HP Network Virtualization • Easy to use and implement • Needs more promotion from HP
IBM	IBM Rational Test Virtualization Server	• Mainframe support • Not able to complete proof of concept • Complex and difficult to use • Requires professional services to implement

Vendor (cont.)	Product Name	Most Frequent Comments
Parasoft	Parasoft Virtualize	• Easy to use • Technically sound • Extensive protocol support • Affordable/low cost of ownership • Limited global presence
Tricentis	Tosca Virtualize	• Integration with existing test management and automation solutions • Unknown in service virtualization market

Expansion Challenges

In addition to the benefits, we also asked participants about the challenges they faced with the adoption and expansion of service virtualization in their organizations.

voke research shows that 69% of service virtualization solutions are used at the project or departmental level. In contrast, 19% of service virtualization solutions are used at the enterprise level. Given that service virtualization is a critical technology in removing cost, quality, and schedule constraints and offering a measurable and proven ROI, the technology should be experiencing broader enterprise-wide adoption and usage.

The following reasons were cited as the most common challenges to enterprise-wide adoption of service virtualization.

- Key stakeholders not understanding the service virtualization value proposition
- An unwillingness to expand beyond stubbing and mocking
- Lack of funding especially when budgetary silos exist
- Limited resources for training
- Lack of an executive champion to transform the lifecycle and increase collaboration and productivity through new technology
- Lack of education about what service virtualization is, why it is important, and how it can help

Think about your own organization, do you already have an executive champion willing to proactively invest in new technology to prevent unnecessary delays and quality risks, or do you need to find a change agent to help build business buy-in?

Mocks and Stubs vs. Service Virtualization

Our research revealed that one of the most common barriers to service virtualization adoption is the confusion between stubbing/mocking and service virtualization. It is common to see developers resist service virtualization use because they believe that this type of work is already achieved through stubbing and mocking.

Stubbing and mocking became popular with developers working in parallel development environments. In a parallel development situation, multiple teams are working on the same codebase with the same components. A typical example is a new product team and a maintenance team. Both teams need access to components created by the other in order to finish their work. This need to access components by both teams is known as a deadlock.

Developers created the concept of stubbing or mocking as a way to make the dependent resource available for parallel development teams. Stubbing and mocking exemplifies the age-old adage of "today's solutions are tomorrow's problems". With stubbing and mocking quality is an issue because the stubs and mocks may not be an accurate reflection of the complete or final behavior of the dependent resource. The impact to quality has a cascading effect—from introducing unnecessary and undetectable defects to functionality failures to an increase in rework costs.

From a testing perspective, stubs and mocks make the test suite ignore unavailable components. In doing so, vital components are left out of testing and may not be tested in the aggregate until a final end-to-end test prior to going live. And in the worst case, these components may not be tested at all prior to production.

Both stubs and mocks and service virtualization require that assets must be replaced with real components and fully tested prior to release. The advantage with service virtualization is the ability to virtually test behavior incrementally prior to full availability of all components.

Service virtualization provides a virtualized asset that:

- Represents realistic behavior
- Allows for sharing across multiple teams in the software supply chain
- Allows modifications to create different conditions and behaviors
- Represents composite behavior and maintains "statefulness"

Service virtualization, unlike stubbing and mocking, allows for a "stateful" component. A stateful component means that the virtualized asset is capable of storing the interaction and triggering the interaction with other components or virtualized assets. Unrestricted access to stateful components is essential for today's complex software.

Service virtualization eliminates the problems introduced by stubbing and mocking, such as unintentional defect introduction, inability to conduct integration testing throughout the lifecycle, and reducing the cost of rework to remediate defects.

Service virtualization is an architected technology; stubbing and mocking are workarounds. Developers must not be given the opportunity to decline the use of service virtualization because they have built stubs or mocks. Service virtualization is a technology that impacts the entire software supply chain for the betterment of the software.

Expanding the Use of Service Virtualization

Many organizations are intellectually aware of the benefits of service virtualization but struggle with the first steps to implement the technology. Suggested first steps to leverage service virtualization are:

- Requiring virtualized assets when source code is checked in by development
- Virtualizing third-party elements
- Virtualizing any part of the software supply chain that is fee-based access
- Virtualizing assets that are a part of an organization's core technology
- Virtualizing reusable assets
- Requiring the use of service virtualization throughout the software supply chain by all participants

Each of these steps removes a roadblock and promotes enterprise-wide adoption. Successful implementation of service virtualization is a team effort. While the testing team gains the immediate benefit of being able to test earlier with unavailable or incomplete services, the true value of service virtualization is realized with better business outcomes delivered through software.

⊙ SERVICE VIRTUALIZATION ROI

We asked participants to share their metrics and ROI data resulting from the use of service virtualization in their organizations.

ROI — Time Saved

We asked survey participants how long they wait to access such required systems as APIs, applications, components, datasets, user interfaces, services, and so on before and after using service virtualization. On average, participants wait 32 days to access required systems before using service virtualization. After using service virtualization, wait times are essentially eliminated with nearly on-demand access to all required systems.

The table highlights the percentages of participants and their wait times before and after using service virtualization.

Wait Time	Before Service Virtualization	After Service Virtualization
On-demand (no wait)	0%	27%
Seconds to Minutes	0%	14%
30 minutes to 2 hours	0%	10%
4 to 7 hours	2%	17%
1 day	1%	11%
2 days	1%	10%
3 days	9%	8%
4 days	3%	1%
1 week	8%	1%
2 weeks	15%	1%
3 weeks	27%	0%
1 month	14%	0%
2 months	10%	0%
3 months	5%	0%
4 to 6 months	3%	0%
Never (no access ever)	2%	0%

The above table is summarized below in terms of average and median wait times before and after using service virtualization.

Required Systems Wait Time	Average	Median
Before service virtualization	32 days	21 days
After service virtualization	1 day	1 hour

If your organization is waiting a month for dependencies or not testing prior to a release, your loss of productivity and business risks far outweigh the cost of this technology.

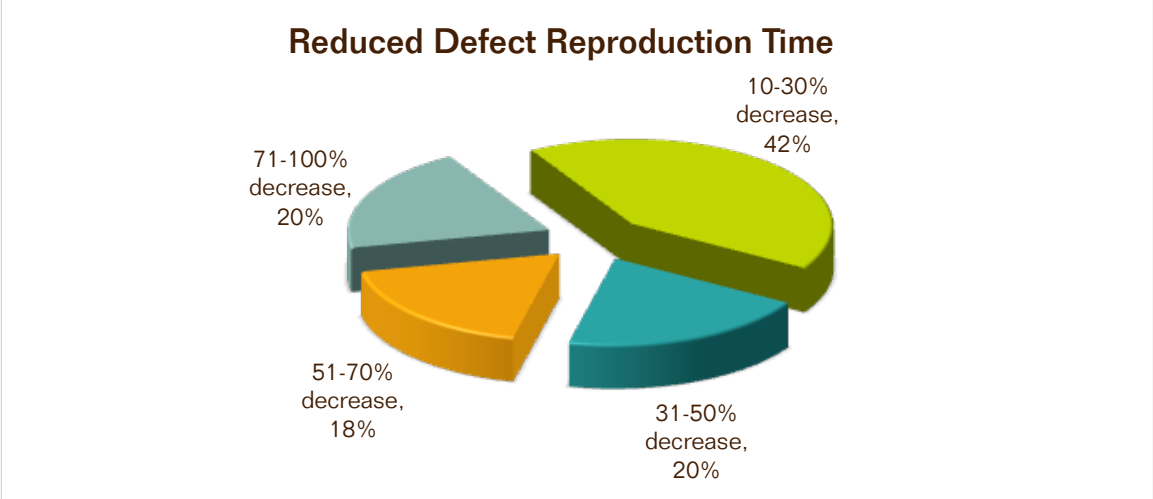
Think about your own organization and the amount of time spent waiting for required dependencies. This information alone may be sufficient to justify an investment in service virtualization.

ROI — Defect Metrics

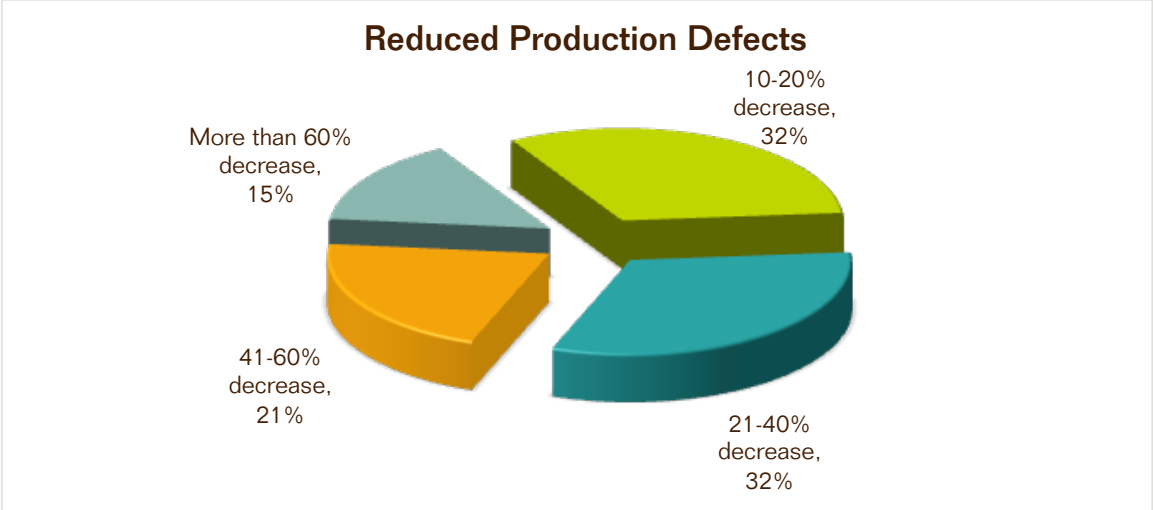
We asked participants about improvements related to defects they experienced since using service virtualization. Overall, participants indicated that the ability to provide the test

and the environment that contains the defect reduced the defect reproduction time. This is a classic use case of lifecycle virtualization and having everything when needed for as long as necessary to reproduce a defect.

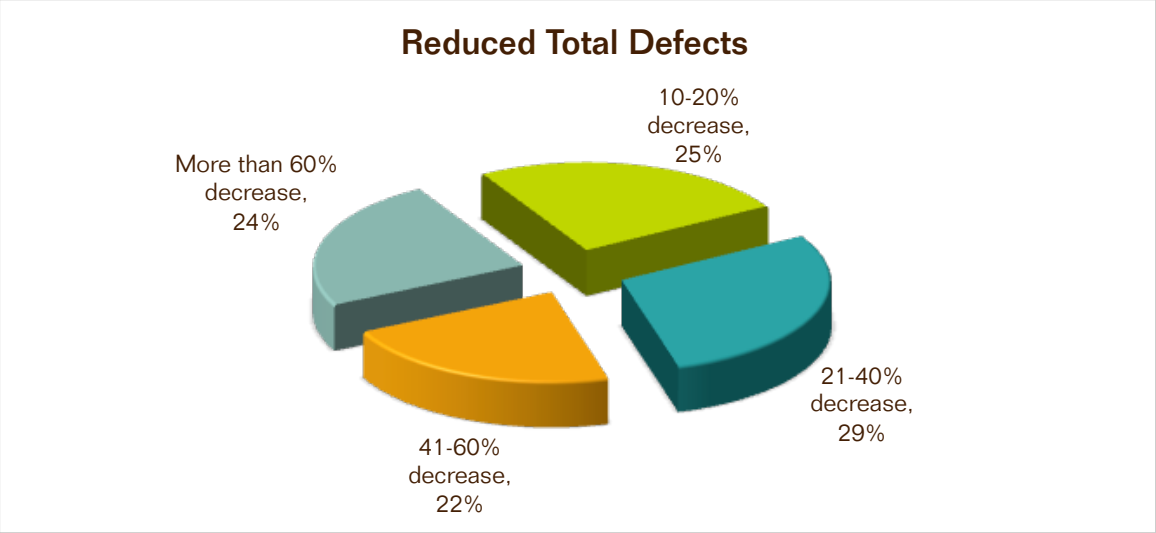
We see that 38% of participants reduced their defect reproduction time by 50% or more.



We found that 36% of participants reported a reduction in production defects by more than 41%.



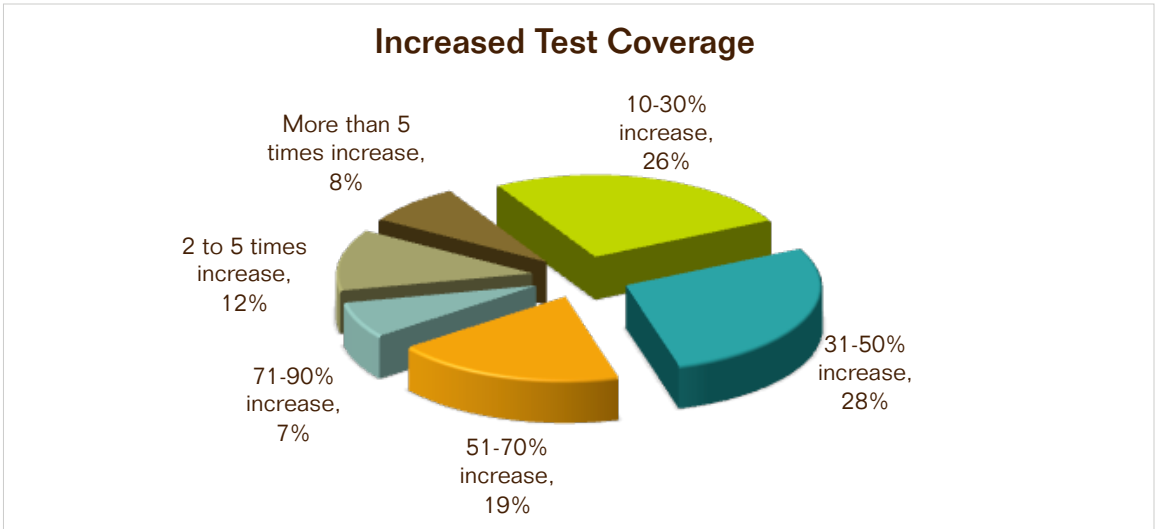
Almost half, 46%, of our survey participants also experienced more than 41% reduction in total defects.



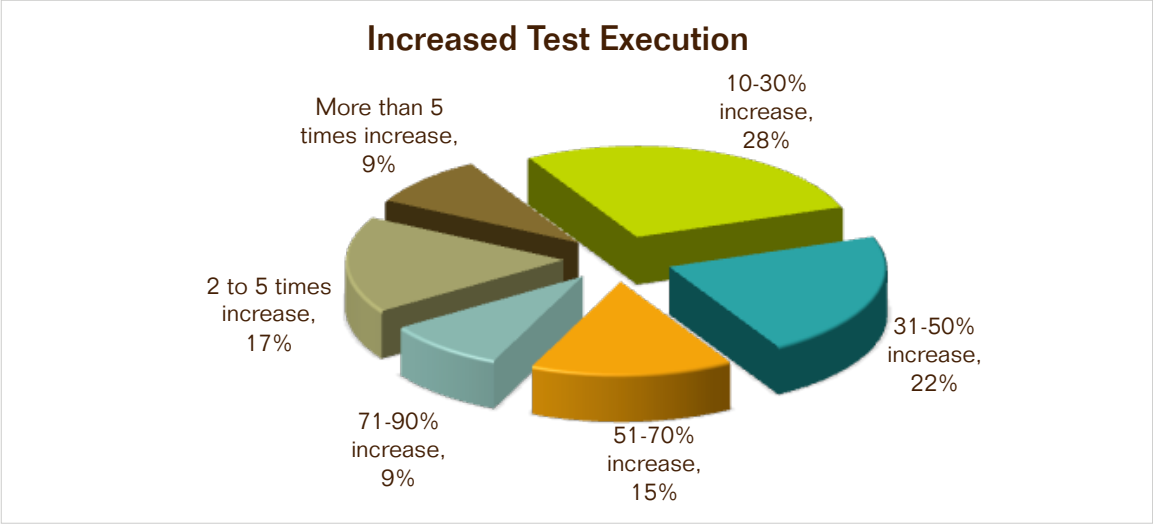
ROI — Test Productivity

We asked participants to describe the improvements related to testing that they experienced since using service virtualization. Overall, participants identified service virtualization as a way to enable more test automation, test earlier, reduce testing cycles, and test what is required. Think about your own organization and whether you have the availability of dependencies and the capability to test what is required.

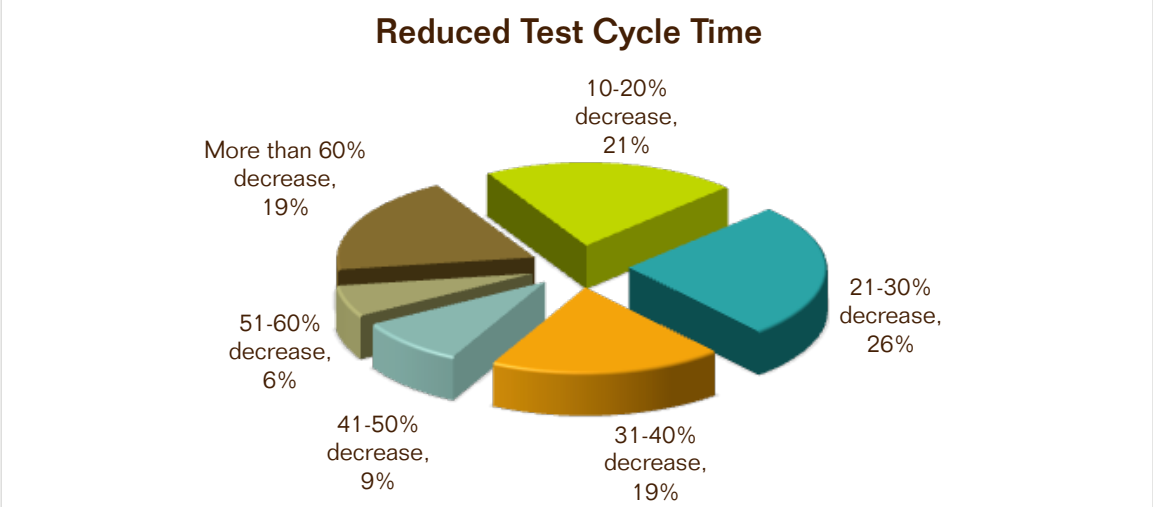
We found that 20% of participants more than doubled their test coverage since using service virtualization.



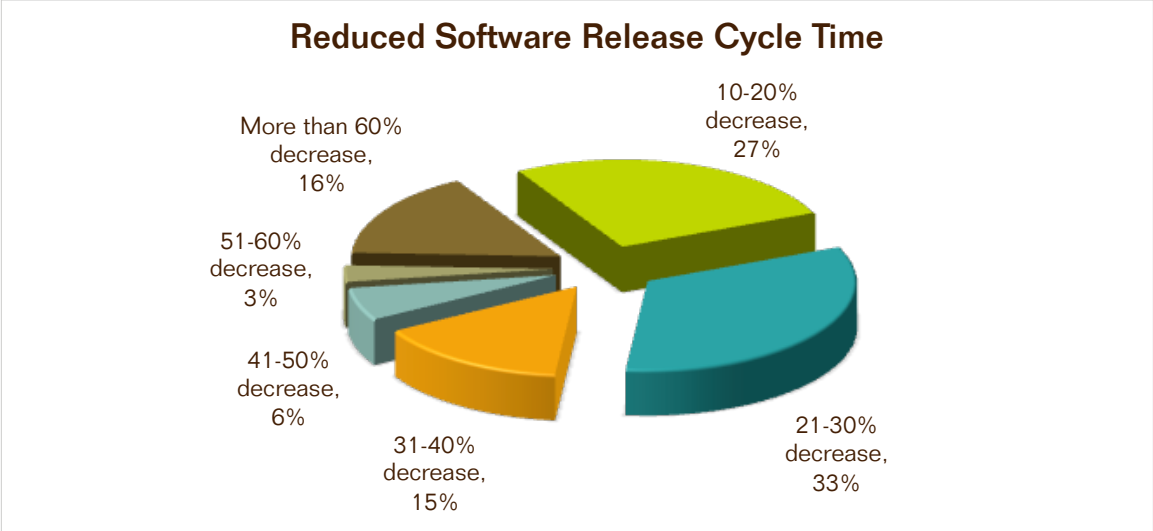
We see that 26% of participants doubled (or better) their test execution rates.



We see that 34% of participants experienced a 50% or greater reduction in test cycle times.



A total of 40% of participants saw their software release cycles decrease by 40% or greater.



Participants commented that service virtualization provides clearer insight to requirements failures. Because of a better understanding of the impact of and cost of requirements failures, more time is spent on requirements. The conclusion is that spending more time on requirements equates to less time spent developing and, ultimately, fewer requirements related to defects. Greater customer satisfaction is the end result of this requirements-centric approach to the software lifecycle.

ROI Summary

Service virtualization is a critical technology for the vitality of the application economy. Adoption and implementation of service virtualization yields proven ROI metrics that are quick and easy to realize. The following table summarizes the ROI benefits identified by survey participants.

ROI Summary	Metric
Reduced defect reproduction time	38% achieved a greater than 50% reduction in defect reproduction time
Reduced production defects	36% achieved a greater than 41% reduction in production defects
Reduced total defects	46% achieved greater than 41% reduction in total defects

ROI Summary (continued)	Metric
Increased test coverage	20% achieved more than two times the test coverage
Increased test execution	26% achieved an increase of two times or greater of test execution rates
Reduced test cycle time	34% achieved a decrease of 50% or greater in test cycle time
Reduced software release cycle time	40% achieved a decrease of 40% or greater in software release cycle time

ASSESS YOUR ORGANIZATION

Service virtualization is a cornerstone of testing automation throughout the lifecycle. The inability or unwillingness to automate testing is a barrier to releasing software faster and with greater quality. This automation barrier impacts the ability of the business to deliver valuable outcomes through software.

Service virtualization is no longer a nascent technology. Service virtualization is an essential technology with a strong and proven ROI, a critical technology used to deliver software that drives optimal business outcomes, and a technology that removes constraints throughout the lifecycle and across teams.

Assess your organization to determine where you are in the evaluation, adoption, implementation, and use of service virtualization. Determine how to best manage the use of service virtualization in your organization by discussing the following assessment questions.

- Is your organization committed to delivering valuable business outcomes through software?
- Is collaboration among teams a reality or just a buzzword?
- Have you released software to production to meet schedules at the expense of quality?
- Is your cost of rework unmanageable?
- Is automation a reality in your organization?
- Is service virtualization being adopted and used by multiple teams in your organization?
- What is the impact to your development and testing productivity because of limited or no access to required systems or components?
- Does your software supply chain suffer because of restricted access to dependencies?
- Is your quality level transparent throughout the software lifecycle?

Service virtualization is a must have technology for the modern software lifecycle. Without the use of service virtualization organizations will be unable to drive valuable business outcomes through software. And, they will suffer from dubious quality and experience negative financial impacts through excessive rework costs.

Evaluate service virtualization to make virtualization the hub of your application lifecycle.

NET / NET

Software complexity, heterogeneity, and interdependence are increasing at a rapid rate. Service virtualization removes the constraints associated with components connected to an application under test that may be:

- Incomplete or undergoing changes
- Controlled by a partner or third-party member of the software supply chain
- Unavailable at the time required by either the development or testing team
- Inaccessible for the frequency or full duration of time required to test especially for performance and load testing
- Difficult to provision or configure in a test environment
- Needed concurrently by multiple teams with different test data

The ultimate goal for every individual involved in a software project is to deliver software that meets or exceeds the needs of the business. Meeting the goal of valuable software requires collaboration among teams and a willingness to explore and embrace new technologies that will add value to the entire software project.

Service virtualization assists in delivering on the overarching software goal while reducing risk and cost. Service virtualization will:

- Deliver the necessary service dependency for development or testing
- Enhance testing through more code coverage and test execution
- Remove the constraint of waiting for services, components, or the necessary elements to test
- Improve time-to-market

Service virtualization is a technology that removes barriers across teams, the software supply chain, and the organization. A thriving application economy is built with software that delivers valuable business outcomes through a collaborative and cooperative effort on the part of the team.

Teams that are truly collaborative and focused on the common goal of building and delivering great software to provide valuable business outcomes are firm believers in service virtualization. These types of organizations will thrive in the application economy.

Service virtualization is a transformational technology. Organizations of all sizes and market segments must understand the value of service virtualization and move towards adoption and implementation across the software supply chain.

APPENDIX: RELATED RESEARCH

You may find it helpful to consult the following voke reports for more information about related topics.

voke Category Snapshot™ Report: Lifecycle Virtualization – November 7, 2011

voke Market Snapshot™ Report: Service Virtualization – December 11, 2012

voke Market Snapshot™ Report: Virtual and Cloud-based Labs – August 21, 2014

voke Strategic Brief™ Report: Reducing the Cost of Rework – October 21, 2014

Contact Information

Corporate Headquarters

voke, inc.
2248 Meridian Boulevard
Suite H
Minden, NV 89423
USA

Phone: +1-866-895-9045
Web: www.vokeinc.com
Blog: www.voke.blogspot.com

⊙ ABOUT VOKE

voke, founded in 2006, is a modern independent technology analyst firm focused on the edge of innovation. voke's primary coverage area is the application lifecycle and its global transformation, including virtualization, cloud computing, embedded systems, mobile and device software.

voke provides data and analysis for the economy of innovation. Companies of all sizes, financial firms, and venture capital organizations turn to voke to harness strategic advice, independent research, and speaking services to move markets beyond the status quo. Please visit www.vokeinc.com to become a research client.

vOke®
Moving markets beyond the status quo!