

## White Paper



### Driving Better Business Process Scalability with Modern Software Quality

---

Gaurish Vijay Hattangadi, Rajeev Gupta



To remain competitive, today's enterprise must be able to rapidly support new business processes with an increasingly heterogeneous, changing environment of IT systems. How does a business manager validate this new paradigm, when the old QA approaches do not mitigate the risk of failure? What methodology, systems and tools can we use for coping with the challenges of providing quality in today's agile business environments? How do you know that your CXO's business vision is matched by the quality of systems you invest in?

## Executive Summary

Ensuring quality in modern IT is a demanding job. Pushed by the CXO's vision, management implements modern applications - and suddenly one day the realization dawns: classical QA no longer completes the picture. The data warehouses, cloud computing applications and SOA systems of today can't be tested with the same old approaches of yesteryear. Classical testing was based on a model of validating specific application silos, which is no longer the case. As systems are aligned to serving a business process, IT managers are increasingly forced into aligning their testing and validation across multiple services and components in the context of a business process, so how do they do this?

In a recent [SOA survey](#)<sup>1</sup> the results indicated that the top SOA benefits organizations aim for are improved data integration (32%), to enable legacy application integration (32%) and integrated disparate department applications (23%). These are very practical approaches that play to SOA's strengths, but they also increase the stakes for automated testing. In addition, a majority of enterprises will increase or maintain

*Note: For the purpose of this paper we are not specifically discussing quality and testing related to requirements management, design and architecture or coding of service components which are amply covered elsewhere. While ITIL, SOA governance and ALM initiatives form a major part of any IT implementation, this is considered outside the scope of this paper.*

<sup>1</sup> Survey sees SOA strength, Jack Vaughan, ITKnowledgeExchange, TechTarget, March 2009

## The Modern Application World: SOA and BPM

For an IT strategy to enable differentiation in the market, it must enable a company to arbitrage a business opportunity before its competitors do. In today's world, the early bird gets the worm, and everyone else just goes hungry. In effect, IT either helps build a strategic advantage, or it hinders a business process that does. Think of the first to market advantage that Amazon enjoyed in eCommerce, or the first gas filling station accepting credit cards at the pump, or the automated check-in kiosks at airports. When these business processes are executed successfully, leveraging new IT functionality, they give their business owners a significant first mover advantage.

SOA promises a methodology for strategic IT, and therein lies its value. "SOA" in very informal terms, is the premise of developing systems that serve a business function and can be composed together from disparate services, to serve a business need. SOA promises to deliver a business process oriented, agile IT. We are told that these new integration technologies will extend application lifetimes and integrate disparate heterogeneous systems. Vendor lock-in would be a thing of the past, application silos would be broken, and the great IT dream of agility would come true. While SOA does provide the technology enablers, the devil lies in the details - the implementation, process and testing details, and many more unique and fascinating complexities that must be validated when we attempt to implement such new paradigms.

The world is littered with failed SOA and BPM projects - projects where smart people had good intentions, purchased leading technology, but failed. Why would a technological marvel that promises so much be so hard to attain? Smaller applications with defined consumers have enviable simplicity, as it is easier to just apply the same old development and testing tools and get better results. But larger, enterprise wide SOA apps are far harder to manage and test - which very often ends up making SOA into D.O.A. - and this illustration gives you a brief idea of why:

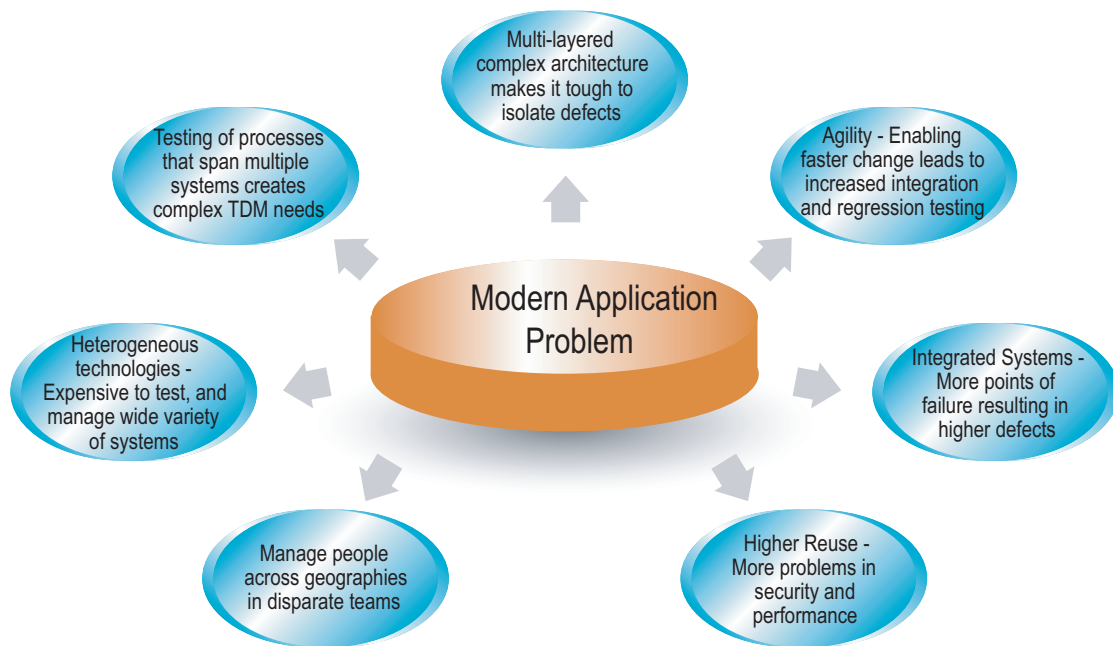


Figure 1: The Implementation roadblocks that modern integrated applications face

### With so many endpoints, where can QA begin?

We have all witnessed managers who inspired by the SOA dream have rushed into service-enablement implementations, purchased the latest technologies and implemented leading integration tools only to wonder: 'How do I test this new monster?'. That is because classical testing tools, testing and methodologies don't fit into the modern application world. For example here are a few questions that we have been asked recently:

How does a old-school manual tester verify service interfaces? Not all of them are transparently exposed through a UI, nor are all services in an application easily covered by plain UI testing.

When an integrated process has 5 disparate systems, setting up test beds for them can be tough. Imagine setting up the test data for 2 mainframes, 2 web applications and a cloud computing SaaS system for a single test run.

My testers are a sincere bunch, they are ready to test any process, and all is well until they are constrained by lack of access to the resources they need. The mainframe that the application integrates is too critical to be made available for testing. The service interface underneath this app is still under development. Sigh....

Furthermore, a business workflow is neither instantaneous nor continuous. For instance, a business process takes a different path when a user does not approve a quote request for 5 days. Well, how do I test this alternate path?

There are also hidden costs - a CRM SaaS system that is used by my order fulfillment process has a per-use fee and is too expensive to test! Given some or all of the above issues, how do I test the integrated order fulfillment process?

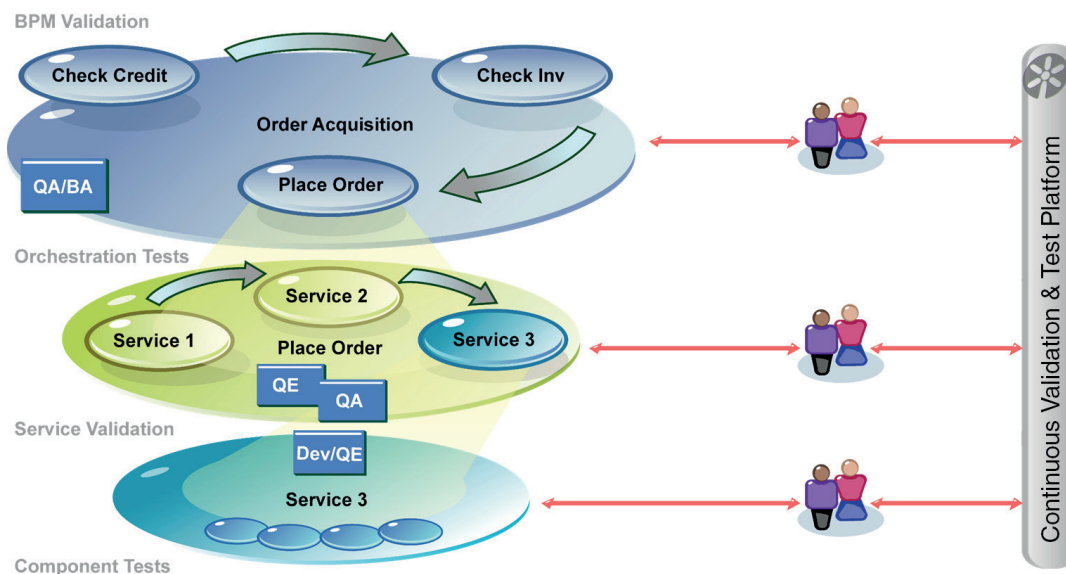
We have many more such questions that managers ask. When you bring in new IT paradigms, QA had better come up with modern processes to cover these assets. So now that we know what the problem is, let's answer the call for help and start mitigating risks!

## Orienting QA with the Business Process

As managers move from being in charge of application silos to being stakeholders in the business process they must align their testing and QA methodology. Managers must be able to gauge the effectiveness of an IT implementation of a business process and be willing to tackle new questions that range from process throughput to IT implementation quality of a business function.

Therefore the first objective of this new QA paradigm is identification of the business process to be tested. How is this different from the classical view of QA? By an order of magnitude, since the scope increases - an example, a CRM portal that must fetch data from CDI processed data, services exposed by ESB (e.g the distribution module from an ERP for showing order status for customers), public cloud computing application (e.g. mail integration, service center software showing problem tickets), etc. Classically, managers could have tested the portal in isolation and shrugged off responsibility for the quality of the dependent applications - but today we ask new questions. Questions including end to end performance, security, reliability and many more. We view the systems not in isolation but as a combined whole that must roll up to serve a business need.

Let's take a look at a typical business process for an Order Management application and simplify it to three levels. You have



Source: iTKO

Figure 2: Automated quality processes and tests need to be "rolled up" from a fine-grained component level, into sub-processes and overall business processes to allow fast collaboration among teams on defect resolution

the BPM team that is defining the overall process, the Integration team that is assembling workflows that support the process, and the individual delivery teams that are programming or offering Component-level services that are assembled within the workflows (that support the BPM-defined processes).

The component-level teams provide tests that bubble up to the Integration team as they assemble the services into a Workflow that conducts one aspect of the BPM process, for instance a service that pulls in the Product Data into the workflow of “Place Order.” Then the Integration team conducts their own tests of “Place Order,” which become part of the validation process for broader testing at the BPM level of “Order Acquisition.”

“Top Down” view: The guys at the top of the stack who are using BPM tools cannot possibly be expected to understand the issues and interactions that are happening at the Integration and Component levels. But they do need to know when an issue arises in their business process as it changes, and be able to report it to the teams who are responsible for delivering the underlying logic. They need to take the effort to validate their business processes, and trust that the underlying delivery teams have done so at their own level.

“Bottom Up” view: The key here is to sign up each team for delivering a set of tests as they contribute technology components or Services into the Workflows that are assembled by Integration teams. Integration teams can take those component tests as “test harnesses” that describe what each component should do, and then tie them together as they are doing integration to provide proof that the expected workflow is valid when it is assembled from underlying components.

Given that a business process must be served by multiple applications all integrated through a process and service layer we next examine these components. We build an inventory of all the systems that participate in fulfilling the process. This includes the applications that satisfy transactional needs, the data warehouses and portals that take care of reporting needs, the enterprise service inventories that serve business functions and the workflows that compose services together to serve the business process.

Each of these assets have their own unique testing needs. A baseline requirement for SOA is the belief that the individual systems that expose services are properly tested. The baseline requirement of validating siloed systems can still be met by classical testing methodologies. It is the service layers and complex integration scenarios that now warrant our attention

## Approaching Modern Application QA Solutions

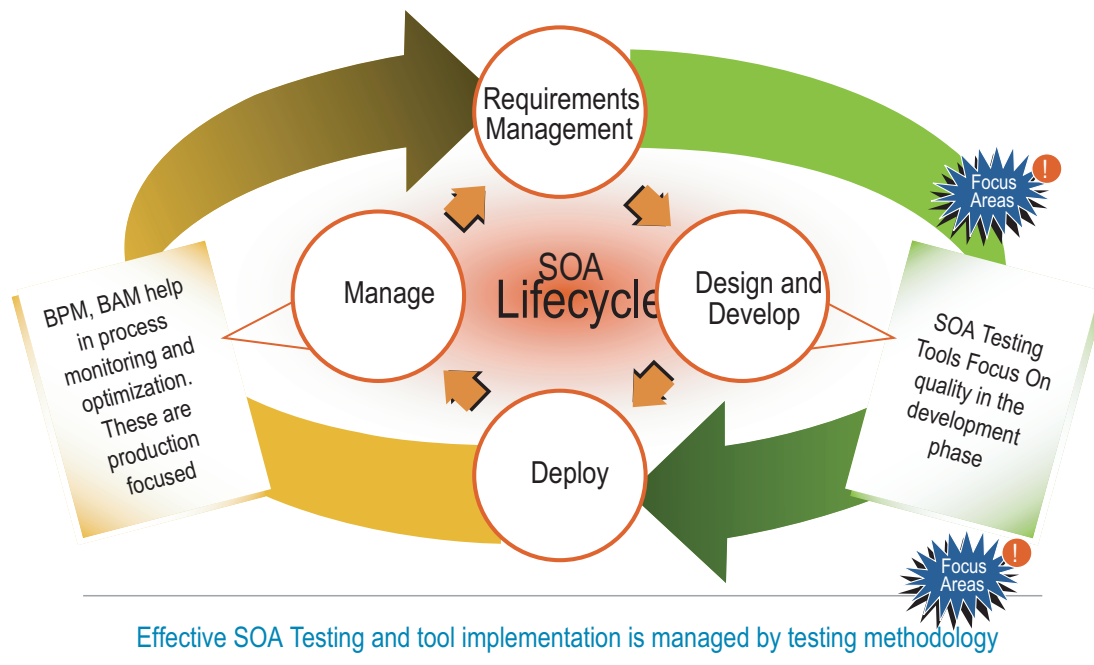


Figure 3: Modern SOA Systems need sophisticated Quality support in all stages of their lifecycle. Specialized SOA Testing tools provide support in development. BPM, BAM and repository tooling support deployment and production. All of this is executed in the context of a SOA Testing methodology.

In the interest of space we narrow our focus to the service layer and orchestrations and that make up the end-to-end application workflow, to take a deeper look at the methodology that underlies the testing of these SOA systems.

One way of looking at SOA from a quality perspective is to break the application lifecycle into development and production management. Testing tools provided by various vendors focus deeply on ensuring that services and orchestrations are correctly implemented. BPM tools ensure that SOA components such as services and orchestrations correctly serve a business process in production.

The usage of these tools is defined by a SOA testing methodology that governs SOA testing and specifies the what (what do you test), how (e.g. tools usage) and when (what part of an interface is tested at what part of the testing lifecycle). Of course, this is a very simplified view (BPM tools do contribute to development and SOA tools will sometimes tackle production monitoring aspects) - the point to note is that you need to cater to three elements for fulfilling the SOA testing needs of your organization.

## Modern Application Testing Methodology: What do I test, and when do I test?

Aligning testing to the business process leads to the identification of assets to be tested. The first priority is to ensure that these components have been unit tested and they are working as expected in isolation. Modern apps, which integrate disparate applications cannot be built successfully if the individual applications are not verified to be correctly implemented as well.

In a complex system, the more layers of abstraction you have, the less likely all of these layers will work together when you

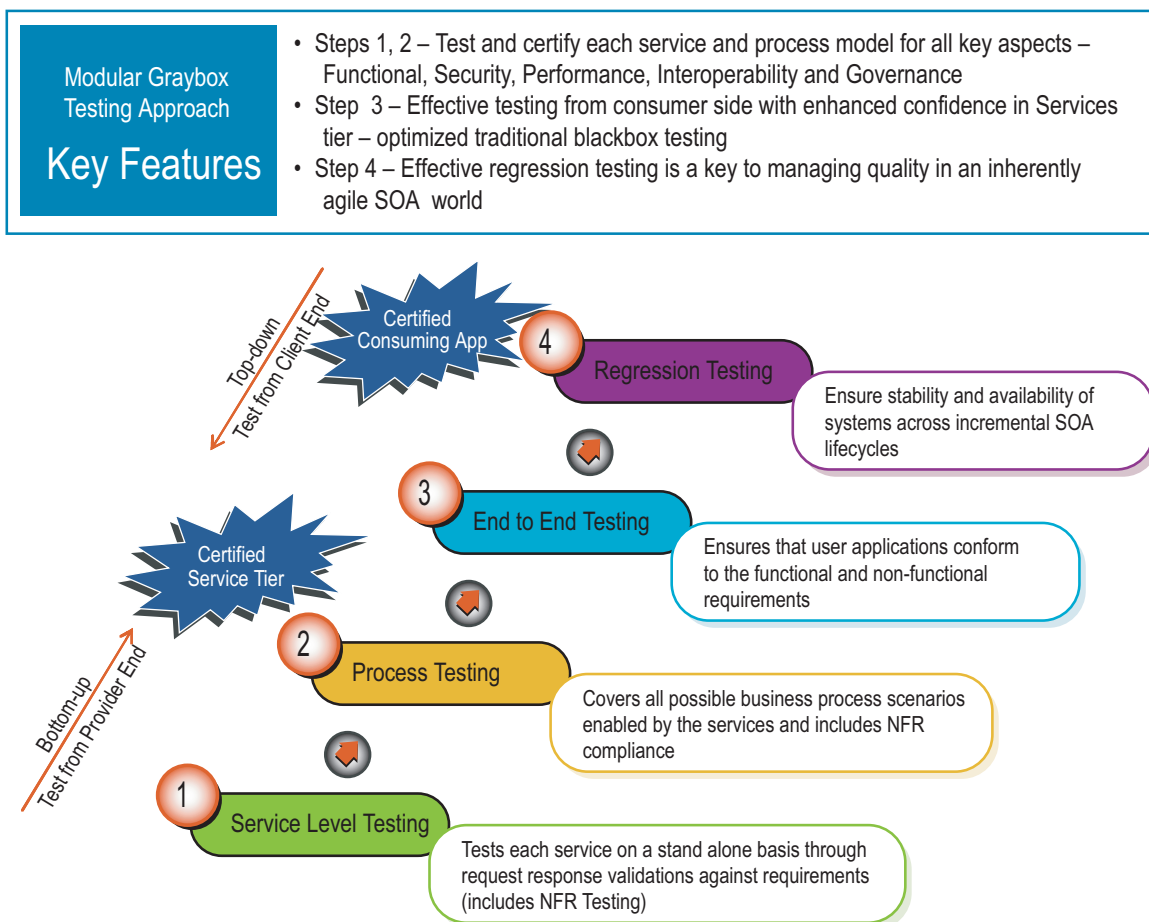


Figure 4: A simplified view of the proposed testing methodology for SOA systems. Testers begin from the provider side with services and ultimately scale up to test business process implementations (process models / orchestrations). Testers also test from the consumer side and these end to end tests evolve into regression suites.

hook them up, and the harder it can be to validate behaviors at other layers that may affect the one you are testing.

We begin our modern testing process from the testing of services (which serve as interfaces to application components). The diagram above provides a simple overview of a SOA testing process:

## A Point Of Interest: SOA Testing Evolution

Moving on from the methodology changes, perhaps the other interesting change in the modern development and testing world has been the evolution of new tools. Dedicated SOA testing tools that are now available, allow for testing most of the SOA system facets during development and through deployment. Newer concepts that aid the SOA testing paradigm include:

- **Service virtualization:** Allows for services and databases that serve a business process to be modeled in a virtual environment. These virtual models then behave very similar to the systems and service that they mimic, allowing for replication of expensive or hard to access systems (e.g. emulating, mainframes, SaaS and cloud computing systems that have per use costs), simulating consistent test data, and removing development bottlenecks of incomplete or unavailable systems. Service virtualization solutions (e.g. iTKO's LISA Virtualize and other approaches to stubbing and mocking) improve availability of scarce systems and resources. This is complemented by hardware virtualization (e.g. VMWare, Citrix, Xen) that share hardware resources amongst software systems.
- **Defect Tracing:** Sophisticated SOA testing tools are capable of tracking a defect across all layers and components that comprise an orchestration (The workflow implementation). Where classical testers could only test the UI and identify that something somewhere was wrong, these tools can walk through the complex web of service invocations in a process and identify problems.
- **Security and Regulatory Compliance:** SOA brings with it a complex set of standards and practices for addressing security requirements and regulatory compliance. Identities need to be managed, services need to be secured, malicious messages need to be filtered and many more aspects managed in a modern environment. Testing tools and BPM products together can verify that services are securely developed and compliant with standards, and finally correctly implemented in production. Regulatory compliance is addressed by some Policy management tools through centrally managed repositories. Service developers do not need to think of these security and compliance aspects when developing - that job is relegated to a security specialist who can selectively develop and apply policies throughout the enterprise.
- **"Stateful" Message Verification:** Modern SOA testing and BPM validation tools can examine messages as part of "stateful" conversations. Following messages in transit across systems is about more than verifying structure and contents against standards. Stateful testing is critical to testing middleware systems, for instance a web service that routes to an ESB, that processes orders and routes a message across multiple systems based on some business rules. We must look at the path the message takes, how each service modifies application state and how the message is transformed to support a complete workflow.
- **End To End Performance Testing:** It is insufficient for performance testing to be relegated to a single application these days. Newer testing tools can gauge the performance of not only the services and application that underlie a process but also the performance of the process itself. Process mining is a highly developed discipline today and provides a key input for process improvement. An example of the sophisticated performance testing needs of SOA applications today, is a request we have seen frequently where a portal page is composed of data sourced from ESB services and master records and also updated by external applications such as news tickers. How does a performance testing team measure the performance of such portals? How does one test the latency imposed by external partner driven widgets that update frequently?
- This is a small sampling of the evolution of testing tools to meet the requirements in the rich browser UIs of the modern application world. Testing tools today are extremely sophisticated to meet these types of demands

## Summing up

In summary, classical testing tools and processes do not adequately cover the new requirements imposed by SOA systems and modern applications. A combination of specialized SOA testing methodology, continuous validation of business outcomes atop BPM systems, and virtualization to eliminate lifecycle constraints can combine to provide a comprehensive quality methodology for SOA systems that can mitigate the risk of delivering IT business processes today.

## About the Authors

Gaurish Vijay Hattangadi (Gaurish\_Hattangadi@infosys.com)

Gaurish is a Solution Consultant with the Independent Validation and Testing Services Practice at Infosys.

Rajeev Gupta (Rajeev.Gupta@itko.com)

Rajeev is a Director of Solution Architecture and leads the Value Delivery practice at iTKO.



### About iTKO

iTKO helps customers optimize their software development and delivery lifecycle for greater quality and agility in an environment of constant change. iTKO's award winning LISATM product suite can dramatically lower quality assurance costs, shorten release cycles, reduce risks, and eliminate critical development and delivery constraints by virtualizing dependent IT resources to provide on-demand accessibility and capacity for teams across the software lifecycle.

For more information, visit <http://www.itko.com> or check out our blog at <http://blog.itko.com>.



About Independent Validation Solutions (IVS) Infosys' Independent Validation and Testing Solutions practice provides third-party validation solutions and services that help global 2000 companies win in a flat world.

The 8000 people strong practice offers end-to-end validation services which range from consulting to custom validation to specialized testing, ensuring minimal business disruptions. A strong complementary IP-based solutions program ensures significant improvements in testing efficiencies and cycles. Established best practices further help clients achieve operational superiority and improve time-to-market, while retaining a customer-centric focus.

With Infosys' Independent Validation and Testing Solutions, clients are assured of a transparent business partner, world-class processes, speed of execution and the power to stretch their IT budget by leveraging the Global Delivery Model that Infosys pioneered.

For more information, visit [www.infosys.com/testing-services](http://www.infosys.com/testing-services) or view our blog at [www.infosysblogs.com/testing-services](http://www.infosysblogs.com/testing-services)



For more information, contact [askus@infosys.com](mailto:askus@infosys.com)

#### About Infosys

Many of the world's most successful organizations rely on Infosys to deliver measurable business value. Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 30 countries build tomorrow's enterprise.

For more information about Infosys (NASDAQ:INFY), visit [www.infosys.com](http://www.infosys.com).