Next Generation Testing
**Introduction**

As business application development becomes more dynamic, distributed and component-based, new approaches are needed in testing to help business applications deliver on the demanding needs of the business. IBM has seen that testing—frequently mistakenly viewed as adding little business value—often consumes a minimum of 25 percent to as much as 50 percent of the time and cost associated with the typical application life cycle.

Testing clearly has taken on an increasingly important role in application development and is one of the biggest drivers of overall development cost and quality. A new approach to testing is needed to address these challenges—one that is linked across the life cycle, which gathers information and uses this information to model and predict outcomes in order to guide intelligent action. Such an approach would help to answer the following questions:

- How do we determine the quality of our testing effort?
- Are we getting our money’s worth out of testing?
- Are we doing too much or too little testing?
- How can we shift more of our investment to earlier project phases, including development?
- Will an increased testing investment drive any further improvement in quality?
- Our budget was cut—so what testing should we eliminate? What impact will it have on application production quality?

**Next Generation Testing**

Next Generation Testing represents how testing must evolve to address these challenges across the development life cycle. It is driven by four factors: cost reduction, quality improvement, risk management, and schedule acceleration. By focusing only on a single factor or fewer than all four factors, one’s view of the cost and value of testing can become distorted. A total view is required—one that looks at all four factors along with the impacts and relationships each has on one another.

At the core of the Next Generation Testing is data. Data is mined, analytics are applied, and insights are delivered. These practices lead to intelligent action that can be taken to enable change. Change is the imperative—acting on how we organize; what tools, methods and technologies we use; and what models become most influential in achieving our goals. It is also a main reason why we need more meaningful data in order to help predict and plan.

As change forces companies to adapt, organizations are developing new ways of working to manage change and balance the key factors of cost reduction, quality improvement, risk management, and schedule acceleration.

**Figure 1: Factors driving Next Generation Testing**

**Next Generation Testing can help organizations drive dramatic reductions in cost and schedule while improving overall quality and lowering business risk.**
As an example, Figure 2 illustrates work done for a client that applied this approach using IBM’s methodologies for defect correction. Their total cost due to defects, including the effort to test and the associated business impact from defects, was in excess of US$20 million. By having IBM apply innovative techniques to reduce the number of defects, the client identified the potential for 35 percent cost reduction and greater than 40 percent schedule acceleration.

In the client example in Figure 2, you see that although the cost of testing only went down by US$600,000, the overall cost—including the cost of fixing defects—was reduced by over US$7 million. This represents a new way to understand the real costs and impacts of testing, and a more insightful way to present this information to executives.

IBM believes it is critical to structure a Next Generation Testing Architecture across five layers: Data Intelligence, Simulation/Forecasting, Execution, Management, and Infrastructure.
Most testing tools available today have focused on the execution and management levers for functions such as test case management and execution and defect management. IBM believes the Next Generation Testing will drive significant improvements in quality and cost reduction by incorporating new tools across the entire architecture. IBM is developing capabilities that will allow clients to extend their capabilities to understand, plan, simulate, forecast, execute, and manage testing.

Progression toward a Test Utility

IBM has seen an evolution in client objectives when pursuing testing services engagement models. Initially, most clients purchased test services in a staff augmentation mode as depicted on the left hand side of Figure 3.

In the staff augmentation model, the client directs the testing efforts and the benefits are based purely on labor arbitrage. Expected savings for this model range from 10 percent to 20 percent of overall costs. Typically, there is no transformation included in this model.

In the last few years, IBM has seen a transition toward the Test Center of Excellence (TCOE) model, which is represented in the middle of Figure 3. The TCOE model is typically a managed services model where committed service levels and performance indicators drive the provider’s focus to the service delivered instead of just the resources delivering the service. This model is very effective in delivering high quality test services and is a model that supports the transformation of the client’s testing models. Savings with this model often range from 15 percent to 30 percent.

In the future, we see a move to a next generation model where the Test Utility will focus on providing services in an outcome-based pricing model—where a definable, repeatable and predictable “unit price” for test work can be put in place for services that are delivered through an on demand provisioning approach. This model is really an extension of the TCOE model, offering clients more flexibility. This will allow clients to closely correlate their testing spend to their actual test usage. As a client moves from staff augmentation to TCOE to a Test Utility, the potential cost improvements that can be expected increase as shown at the bottom of Figure 3.

Organizations are growing more and more complex, with numerous departments, vendors and locales. From a testing perspective, this complexity needs to be managed and coordinated into one smoothly operating, reliable test management system. A TCOE model provides a basis for effective standardization and communication across the testing organizations, the business units and ultimately for the end users and clients.

IBM’s TCOE model is driven by a common governance structure, methodology, tools, and processes. The TCOE model utilizes a factory method for execution, driving standardization across testing. It will perform critical change management from a test perspective and act as the focal point for change. It will promote key best practices across test management and execution such as forecasting, estimation, and upfront code analysis. It will also be the conduit to explore and pilot new
Organizations are growing more and more complex, with numerous departments, vendors and locales. From a testing perspective, this complexity needs to be managed and coordinated into one smoothly operating, reliable test management system.
Innovative methods. The TCOE can provide a flexible execution model that can be ramped up and down depending on the business demands. Figure 4 illustrates a successful TCOE model that IBM has implemented for one of its clients.

This banking client faced several testing technology issues. They were experiencing unreasonably high operational cost and the quality of the output was often compromised due to variations in their processes. Their primary objectives were to reduce the cost of quality, improve the application quality and time to market, increase test transparency, and free up internal resources.

IBM’s testing solution provides managed testing services via a test factory in a two-tier model in Bangalore, India; test automation via Rational tool suite; and continuous innovation injection through working closely with IBM Research. It also provides an accelerated and phased approach to transforming current methods, processes, and tools into an industrialized approach.

The solution targets to achieve the following results:

- On demand resource model (40 percent core team, 60 percent flexible resources)
- Ninety percent of worked performed off-shore
- Productivity improvements through 50 percent to 70 percent test automation and use of Rational tool suite
- Unit-based pricing (“test case-based”)
- Factory performance measured by 20 service level agreements

**Figure 4:** Global financial institution implements Test Center of Excellence
With an implementation of a TCOE, a client should realize the following benefits:

- Client management to focus on outcomes instead of test execution
- An on-shore model that allows direct responses to urgent project demands
- Access to a scalable and flexible resource pool to meet peaks and valleys
- Centralized harvesting of knowledge and retention of intellectual property
- A single test owner to provide a uniform process, efficiency improvements, consolidated test communications, and reduced administration overhead
- Recruitment effort transfer to partner and increased staff retention
- Higher test system utilization

IBM believes that a number of essential elements should be a basis for the TCOE model. These include:

- Professional Certification Metrics Management / Service Level Agreement (SLA)
- Factory Automation
- Test team independent of Development team
- Business Process testing (applicable to User Acceptance Testing (UAT)
- Testing as a profession
- Knowledge Management
- Virtualization
- Delivery Excellence
- Requirements Traceability
- Vendor Management
- Resource Capacity Management
- Test Automation
- Consolidation of Test Cases
- Periodic Self-Assessments
- Global Delivery
- Regression Testing

- Lab Management
- Risk-Based Testing
- Requirements-Based Testing
- Data Security
- Test Coverage

However, in the TCOE model from IBM, additional IBM unique best practices are integrated through IBM Total Test Quality to further enhance the benefits that are achievable by the client. These include utilizing research technology from IBM Research and patented methods to further enhance quality. Some of these practices are described in the next sections.

**Test Strategy**

Historically, test strategy work has always been more of an art than a science. It is very dependent on having a strong test architect who understands all aspects of the program/project that need to be tested. It is important to utilize intelligence and automation to generate key aspects of the strategy.

The test strategy needs to answer key business questions such as:

- How can I determine the quality of my testing effort?
- My budget was just cut—what testing should I eliminate? What impact will it have on application production quality?
- Am I getting my money’s worth out of testing?
- Am I doing too much or too little testing?

The creation of a test strategy can be both standardized and automated. The key theme is the power of using information to execute testing in a more intelligent fashion. The test strategy needs to be developed on an architecture that utilizes data intelligence and forecasting to provide accurate prediction of defects across the life cycle. This intelligence needs to be integrated with an execution architecture that moves defect identification earlier in the life cycle to provide the most effective use of resources.
IBM has developed a tool called the Test Planning and Optimization Workbench (TPOW) that will help a client determine the optimal amount of testing and answer key questions such as “If we extend testing by one month, what will the improvement be in overall quality as measured by size of the maintenance team required, predicted severity level one and two problems, and application availability?” Using empirical data collected over the years in conjunction with defect reduction experience and patented processes, IBM can predict forward and backward defects by life cycle phase.

This technology is critical in helping clients determine the right level of testing. Cost savings can be significant through reduction in schedules and elimination of problems reaching production, while also driving significant improvements in quality.

**Risk-Based Testing and Test Optimization**

Test organizations frequently face challenges to reduce costs and shorten schedules, while detecting defects before applications are implemented in production. Risk-based testing embraces a set of practices that help identify the business, technical and management risks associated with project requirements, and the software components of the system.

Through structured annotations and rules-driven risk classification and assessment, IBM’s test planning tools capture the risks associated with requirements, design and development artifacts. Captured risks will be used as inputs to the automated and manual creation and validation of test strategies, test plans, and test cases.

These risks help determine the level and types of quality assurance needed for each requirement and technical component. Testing policies and goals can be defined by risk category and used in the creation of the test strategies, plans, and cases. The testing policies and goals help define the reviews, inspections, and tests for artifacts across the life cycle. This traceability between the requirements, code, and test artifacts also enables a key feature of risk-based testing. When testing costs and schedules need to be reduced to meet external constraints, risk-based test planning tools can give greater priority to the tests addressing higher risk requirements and components, and meet the constraints by skipping tests that address lower risk items.

Test optimization provides additional capabilities to reduce the cost and schedule of testing efforts. Test optimization focuses on:

- Identifying tests that are overlapping or redundant,
- Using combinatorial analysis to minimize the number of test scenarios and configurations needed to identify defects, and
- Streamlining the execution of test suites using techniques such as intelligent selection to minimize test datasets from production data.

Test optimization from IBM covers a broad range of disciplines, including test automation, test coverage data collection and analysis, mathematical optimization and analytics, and data profiling and masking.

Key to the success of risk-based testing and test optimization is the use of tools that span the life cycle—from project initiation through production. The linkage of requirements, design and development tools with test management and test execution tools make it possible to create and refine test strategies and plans that can be updated with the latest status and can be re-planned on an as-needed basis to respond to changes in cost and schedule.

IBM has developed a factory-based model for application development and maintenance called Application Assembly Optimization, where the standardization of the inputs and outputs provides flexibility for the testing model to be used independently or in conjunction with the development factory.
Within this approach, the standardization enables the utilization of multiple geographical locations in a seamless manner, as they all use uniform processes, methods, and tools. The focus on a common model and tools also provides an increased ability to build the base of experience and facilitate benchmarking projects against each other.

**Quality of Test Process Inputs**
A TCOE should utilize a standardized model for inputs and outputs. This factory model has shown significant benefits for many organizations:

- Faster response to business needs and visibility to delivery execution
- Improved time to market
- Quality and productivity improvements in the range of 10 percent to 15 percent based on initial findings
- Dynamic mobility of work that drives lower development cost and risk
- Resource availability to match the client needs

**Defect Management**
Effective defect management is driven by the following two primary, but distinctly different focuses:

**Management** – the process by which defects are uncovered and resolved as efficiently as possible, while meeting the delivery schedule and analysis needs.

**Analysis** – the process by which defects are studied to yield useful information about weaknesses across the project life cycle, so future projects can both remove defects earlier and prevent as many as possible from occurring.
Unfortunately, defect management is often overlooked as an area in testing that needs a concrete management system and process in place at the start of a testing phase. Frequently, projects only focus on developing the test plan, the test cases and scripts, and the plan execution curve to complete the test cases and scripts. This approach tends to assume that by managing execution, defects will automatically be handled optimally. Many projects do not staff Defect Managers, assuming that the Test Lead will absorb any responsibilities in this area during execution. However, in practice, especially in complex system integration projects, this approach is often a disaster waiting to happen. None of this test execution preparation, for example, will help a project understand the consequences of a growing defect backlog or, at what point, that backlog will bring an entire test effort to a complete standstill.

IBM leverages a unique, full life cycle approach to defect analysis called the Defect Reduction Method™. With this approach to analysis, we can identify strengths and weaknesses across key risk and quality areas: Test Effectiveness, Test Design Effectiveness (which evaluates whether the test is adequately thorough/comprehensive), System and Code Stability, Functional Completeness, and overall Code Quality, including an analysis of where defects are being injected across the life cycle.

This model draws upon many years of project data so that we can tailor expected results to the industry, test phase, and type of testing (product versus system integration) in order to compare actual results against a best practice expectation. This enables projects to develop specific improvement plans that yield reliable, significant reductions in defect rates. Figure 6 illustrates the effect the Defect Reduction Method can have on the defect rate for projects that track, standardize, and implement defect prevention techniques.

**Reducing Test Cases**

During system test or system integration test, many clients develop regression test cases that they run as part of the test process. Over time, the number of regression test cases increases due to new requirements and the amount of time required to execute these tests grows as well. In most instances, it is not necessary to run the full suite of regression test cases. Most releases only change a subset of the applications within the portfolio (10 percent to 50 percent based on size and complexity of the changes).

By identifying only those applications that have been modified, along with their related components, organizations can reduce the number of test cases executed, resulting in schedule acceleration. In some cases, the numbers of test cases in system test and in systems integration testing have been reduced by as much as 50 percent and 30 percent, respectively.

**Automation**

IBM's perspective on test automation covers all aspects of the testing program and makes use of tools specifically designed to support the use cases for modern application development.
One of the most important features is that the requirements, test, and defect management tools are integrated to facilitate automated requirements traceability.

IBM has developed a suite of automation starter solutions that bring IBM Rational tools together with professional testing services to set up and apply the tools for Test Management, Functional Testing, Performance Testing, and Life Cycle Testing. These technologies are a combination of software products from Rational such as Quality Manager as well as research tools and infrastructure capabilities such as the IBM Smart Business Development and Test Cloud.

In terms of automating functional testing, the strategy typically employed is to identify candidates based on an assessment of the areas of highest risk to the business and the most stable areas of the software. To maximize the value of automation, it is important that the maintenance of scripts be minimized, which is best done by focusing effort on the areas of the application that are the most stable. Developing an automation framework is important in order to maximize the reusability of scripts. It is important to remember that not everything can be automated and that the automation strategy maintain focus on what is necessary and prioritize the automation effort.

Given that it is best to automate features that are business critical and stable, there tends to be a significant correlation between regression test candidates and automated test candidates. The benefit of automating regression testing is that the test team can focus its effort on new and changed functionality while not sacrificing test coverage in regression. Under the right conditions, this can enable the test team to effectively test more functionalities in less time.

Test Environment Management
The structure and design of business applications are changing from monolithic to more dynamic, distributed, and component-based. IT needs to be more responsive to business changes, while business is demanding software that is easier to use, more flexible, and more connected to real time data. There are many challenges to make applications deliver on the demanding needs of the business. Test and delivery of these applications is rising in importance to become a very critical function for many enterprises. The problem is that most organizations are stuck testing applications the old way—with manual user interface (UI) testing, manual validation, and physical test environments—and they simply do not work or cannot keep up with complex, distributed, multi-tier applications.

There are many environment issues inherent with today’s testing practices, including the following:

- 30 percent to 50 percent of all servers within a typical IT environment are dedicated to test.
- Most test servers run at less than 10 percent utilization.
- IT staff report a top challenge is finding available resources to perform tests in order to move new applications into production.
- Approximately 30 percent of defects are caused by wrongly configured test environments.
- Testing backlog is often very long and the single largest factor in the delay of new application deployments.
- Test environments are seen as expensive and providing little real business value.

Cloud Computing in the Test Environment
Cloud computing has emerged as a style of IT delivery in which applications, data, and IT resources are rapidly provisioned and provided as standardized offerings to users over the Web in a flexible pricing model.
Within the test environment, cloud computing is rapidly becoming an element of Next Generation Testing that can help create faster, less expensive methods of testing by enabling more parallel development across interdependent teams without dependencies on back-end systems and data stores.

Moving application testing to a cloud-computing environment provides the following benefits:

- **Reduced capital expenses** – By provisioning test servers on demand and making sure unused servers are re-provisioned, you can maximize capital usage.
- **Reduced operating expenses** – By automating the provisioning of test resources, you attack a key variable cost that has an impact on your bottom line—IT labor costs. This capability allows you to redirect key resources from manual configuration activities to more mission-critical and value-added tasks.
- **Reduced test cycle time** – The development and test cycle is a critical path to innovation and the market. Decreasing cycle time directly drives time to market and competitiveness.

**Application Virtualization**

Application virtualization can help lower quality assurance costs, shorten release cycles, reduce risks, and eliminate critical development and testing constraints by virtualizing IT resources to provide on demand accessibility and capacity for teams across the entire software life cycle. It helps clients optimize their software test and delivery life cycle for greater quality and agility in an environment of constant change by simulating the behavior of a physical asset in a software emulator and hosting that emulator in a virtual environment. Application virtualization models the behavior of the system under test and stores the model and behavior in a virtual environment for later testing. Application virtualization can customize multiple virtual test environments that can provide productivity and time-to-market gains by enabling more parallel development across interdependent teams. Application virtualization reduces or eliminates infrastructure costs for test labs, test harnesses, responders and mockups, including hardware, software licenses, and configuration costs.

The following are the additional benefits of application virtualization:

- Provides the ability to rapidly create multiple virtual test environments customized for development and testing teams
- Provides testing teams with 24/7 availability for testing, as needed, without dependencies on back-end systems and data stores
- Delivers realistic simulated development and test environments at a fraction of the cost of physical server environments
- Provides testing teams the ability to test more thoroughly throughout the life cycle without worrying about data or service availability
- Shortens release cycles, reduces risks, and eliminates critical development and testing constraints by virtualizing IT resources

**Test Tool Management**

IBM Testing Services’ approach to test tool management recognizes that a client may use a variety of tools across its software development life cycle. IBM’s approach to tool management is aligned with the roles, processes, and work products used in our methodologies.

Tools to support testing include not only test management and test execution tools, but also automation for test environment creation, test data creation and management, and requirements and modeling. IBM uses a combination of internal and commercially available tools and integrates, as needed, with the tools used by our clients.
IBM’s approach to test tools management also recognizes that there are often unique challenges and opportunities for the creation of new tools that directly address the problems faced by a client. IBM Testing Services draws upon the capabilities of IBM Research to create innovative solutions to difficult challenges in testing that are not addressed by commercially available tools.

**Performance Engineering and Management**
Any successful, well-designed system is going to have comprehensive nonfunctional requirements that represent those system requirements that go beyond the functional requirements and specify the operational aspects of a system such as recoverability, availability, capacity, and performance.

Performance engineering is a technical discipline employed on development projects to provide that a system meets a pre-specified set of performance objectives by applying specialized performance estimation and design skills to the system architecture. In addition, performance engineering manages risks associated with many of the nonfunctional requirements.

Because performance is often a defining factor to the success of a system, relying on performance test alone to address performance concerns is often “too little, too late”.

Performance engineering proactively addresses performance at the project's outset to provide that the system meets its requirements and expectations.

IBM implements performance engineering on development projects using the Performance Engineering and Management Method (PEMM) that is comprised of eight overlapping phases:

- **Requirements and Early Design** – completing PE activities in the earliest part of the project to meet performance and capacity requirements are well defined and assessed at the outset
- **“Volumetrics”** – analyzing the quantitative factors, both business and technical, that will affect system performance
- **Estimation and Modeling** – estimating and validating the performance and capacity of a solution
- **Technology Research** – determining previously unknown performance characteristics of selected solution components
- **Design, Development, and Tracking** – conducting performance engineering activities during the development life cycle, including supporting design and code reviews, and establishing performance time or resource utilization budgets
- **Performance Testing and Validation** – testing and assessing the performance of the live solution, usually by an independent performance test team
- **Live Monitoring and Capacity Management** – managing performance and capacity of the deployed solution in production, using monitoring tools and processes and SLAs
- **Risk and Performance Management** – reducing risks related to performance and identifying, assessing and addressing capacity

**Summary**
As change forces companies to adapt, organizations seek to balance cost reduction, quality improvement, risk management, and schedule acceleration.

IBM’s Total Test Quality approach defines how IBM is approaching the new requirements posed by Next Generation Testing. This transformative approach combines IBM Testing Services with comprehensive automation from IBM Software and cutting-edge innovation from IBM Research to help lower business risk and reduce life cycle development schedule and cost, while improving overall application quality by helping organizations simulate, predict, plan and achieve their unique balance of requirements.
Through innovative test architecture that combines a comprehensive suite of automation technology and cutting-edge innovation from IBM Research, IBM Testing Services can help provide the linkage between execution, management, infrastructure, data intelligence, and simulation/forecasting. IBM believes this is a breakthrough approach to application testing and a key component of world-class application development and management.

For more information
To learn more about IBM Testing Services, contact an IBM representative or visit: ibm.com/services/ams

About the Authors
Steven Kagan is the global leader for Application Management Services Offerings and Competencies. His responsibilities include global offering management for Application Management Services, including Application Outsourcing, Business Application Modernization and IBM Testing Services.

Steve's 25 years of experience with IBM include leading services practices and launching new business ventures within IBM. He has helped build these businesses by leveraging customer relationships, developing intellectual capital, and integrating cross-IBM capabilities into focused customer solutions. His mission has been the creation and deployment of strategies that substantially increase IBM's service market share by providing measurable results for clients. He continues to consult with analysts and clients worldwide and coordinate the work efforts of thousands of professionals globally.

Martin Rolek is the Global Offering Executive for IBM Testing Services. He is responsible for the development of new test offerings, industry-leading innovations, and advanced test optimizers.

Marty has over 30 years experience in business systems and IT management, IT and business process outsourcing and strategy consulting. He has led large global engagements in both sales and delivery roles, including application development, package implementation, and legacy application transformation projects, as well as large IT and business process outsourcing engagements. Marty has guided the development of IBM methodologies for legacy application transformation, full life cycle testing and application development effectiveness, training of IBM services and consulting professionals, and establishment of Center(s) of Excellence worldwide.

---

Steve Kagan
IBM Global Business Services
Vice President, Application Management Services Offerings and Competencies
smkagan@us.ibm.com

Martin Rolek
IBM Global Business Services
Director, Global Test Offerings
msrolek@us.ibm.com