Software Quality Transformation: Focus on Results, Not Process

Executive Summary

Based on an independent survey conducted by the Boston, Massachusetts-based IT project management research and consulting firm survey:

- Only about 37% of users feel their IT shops deliver on-time.
- Roughly 32% of IT projects were considered successful, meaning they were completed on time, on budget and with the required features and functions.

A study by Dynamic Markets Ltd. of 800 IT managers across eight countries shows that:

- 62 % of organizations experienced IT projects that failed to meet their schedules.
- 49% suffered budget overruns.
- 47% had higher-than-expected maintenance costs.
- 41% failed to deliver the expected business value and ROI.

If the above data is correct, software development is in a crisis - a crisis of quality that finds IT organizations all too often unable to deliver applications on time, on budget, and at the quality levels users need. Poor development and testing processes result in wasted money and time, business disruption, lost business opportunities, security breaches and risk of regulatory non-compliance.

To meet business needs, especially at a time when all parts of the business are being challenged to do more with less, companies need to radically transform how software development and testing is managed. This white paper explains how to achieve this transformation, by focusing on specific desired outcomes, and focusing efforts on improving only those processes that affect outcomes. This targeted approach saves money, increases effectiveness, and helps both the IT organization and the enterprise become more competitive.

The Crisis in Software Quality

A quick conversation with any CIO, or an in depth look at success rates of IT programs, provides sufficient evidence on the depth of the software quality problem.

Surveys show that only about 37% of users feel their IT shops deliver on-time, and that only 32% of IT projects were considered successful (completed on time, on budget and with the required features and functions).

These statistics clearly shows the magnitude of the problem at hand. They look even worse in light of the effort, time and cost that has been invested in making these applications work.

Even while businesses can quantify the costs of their application failures, they struggle to build business cases to justify proactive investments in application quality to prevent these failures .

Every time there is a failure it is attributed to process gaps. Therefore, IT organizations concentrate solely on fixing the process gap. Corporations have spent way too much money trying to address these process gaps and increasingly they are finding it difficult to clearly tie process improvements initiatives to direct business benefits. Hence, funding these process improvement initiatives is getting increasingly difficult.

Such expenditures do not solve the problem in software quality many corporations face, nor do they produce a compelling business case for funding them. The fundamental reason is that process improvements will bring about delivering consistent level of performance, but they fall short of meeting specific outcomes that business expects.

What organizations should instead do is first identify, very specifically and in quantitative terms, the business outcomes that are most important to them.

Then they should focus their time, effort and money on identifying the levers that can measure these outcomes, correlating the process that delivers that outcome. From there, they should fix those processes that will produce measurable outcomes.

Let's examine how organizations currently approach improving their test and development methodologies, how a new, more "results-oriented" approach would be better, and then how to move towards it.

Current Methods

Organizations hoping to build a "best-in-class" QA organization typically focus on the following categories:

- Organization, which includes establishing test policies and goals, the structure and governance of the QA function and performance and knowledge management.
- Process, which includes defining standards and guidelines, creating mechanisms for audit, process review and continuous process im provements, as well as techniques for predicting and preventing defects.
- Methodology, which includes estimation and planning, requirements management, test techniques and methods, metrics management, as well as defect, risk and change management.
- Operating model, which covers areas such as project structuring, demand and work load management, the ability of the QA organization to enforce the quality gates and how SLAs are managed and what should be the OLAs be tween service providers
- Tools and technology, which includes the test management tools, test automation tools, the integration of testing tools, license management, etc.
- Infrastructure, which includes managing the state and configuration of the test/QA environment, and the collection and maintenance of test data.

Organizations then typically judge their capabilities and baseline their test processes maturity against industry standard process/methodologies. These include the TPI (Test Process Improvement Institute), TMMi (Test Maturity Model), TMAP, etc. Using these models, an organization evaluates its performance in various areas. Most of these industry models will provide a means to deploy standard processes and can be used to periodically measure effectiveness of deployment of more mature test processes.

One thing to be aware of: They do not clearly identify outcomes an organization will be capable of delivering at an increased level of maturity. It could vary from company to company. Following these models essentially show the organization that it has the capability to deliver consistently at a predefined level of performance, which the organization defines for itself and can be improved upon over time.

Organizations spend large amounts of money, time and effort attempting to move up the maturity ladder for all of their development and test processes, assuming that the higher a maturity level any process achieves, the better its work will be. While that is generally true, it misses the critical point that not all processes have equal leverage in producing outcomes that are most important to the organization and that outcomes that are important at a moment in time can change based on business priorities.

The scope of testing and QA covers many different subtasks, ranging from test strategy definition, test design, test planning, test data preparation and maintenance, automation, through to test execution and reporting and defect management. These tasks span the development lifecycle cycle, and from smoke test to build verification to functional, non-functional and load testing, through user acceptance and production testing. While this is a very inside view of QA, these functions are impacted by various factors which are either internal to the organization or external. Factors such as compliance and regulation, technology advancements like virtualization and clouds, competitive threats also influence our approach and outcomes that relate to QA and testing some directly and some indirectly. Given all of the variables that influence an organization, the priorities of the organization can change from time to time and hence so do the expectations from the various entities that make up the organization, resulting in the expectations of the outcome as well.

Typically, organizations engage a consulting firm to review process maturity. The firm usually advises its client to implement a new process, or to improve an existing process, to achieve incremental benefits. The process improvement usually delivers value, but often there are challenges:

- Process improvement will definitely ensure hygiene and consistent delivery, however the measure of consistency could be different for different organization so indirectly put process assures consistency but does not guarantee outcome, the outcome is for us to determine.

- Secondly, process and maturity models often define an organization's maturity level. However, achieving greater maturity can wrongly suggest that organizations at the highest levels are the most optimized. However, this might be true primarily because the outcomes for each stage is determined by an individual company and could vary. In real life, where a customer's primary business is not IT, its IT organization needs not necessarily to focus on maturing all processes to the highest levels but to concentrate on meeting the quality of deliverables and business expectations of IT.

For an IT organization to meet its changing business expectations consistently over time, they have to have the right level of process or just enough process to drive consistent delivery and also have enough flexibility in their operating model to be able be more agile to changing expectations. Given this paradox organizations need a means to bring about not only consistent delivery of quality software, but also be adaptable to changing business priorities. A best-in-class QA organization will embody the right level of process that will help remain agile while changing its methods to suit changing business expectations.

What is best-in-class QA? The most significant difference between a traditional approach and one that we recommend is that it requires an IT organization to focus on creating a process to enhance delivery capabilities. Meeting consistent delivery standards, however, does not mean that the organization is automatically meeting business expectations. To meet business expectations, IT must define a specific set of outcomes its QA organization needs to achieve. To achieve these outcomes, IT must identify the set of processes that directly impact those outcomes and understand how they can be matured. So fundamentally the approach is two staged:

- Stage 1. Deploy standard processes to achieve consistent service standard.

- Stage 2. Identify outcomes that are most important to the business and map those processes in IT that most impact these outcomes and focus on maturing these specific processes to have the biggest influence on the outcome.

To facilitate this, we have developed a five-stage maturity model. The idea of this model is not to box a customer into a maturity level, but to identify the maturity level of each process and define what it should be at for delivering a desired outcome.

Best in Class Maturity Level

For example if an organization desires to be world-class in global delivery (measuring it as a leverage of onsite-offshore ratio of 10:90) it needs to closely examine processes such as governance, test data management, etc. and identify ways to improve them to the highest level of maturity rather than working on maturing all other processes to that level. What this means is focusing on the levers that have the highest impact/bearing on global delivery (in this case, relative to test data management and governance processes) and maturing these processes to the highest level of maturity to achieve a desired set of predefined outcomes.

The ensuing pages elucidate the approach we use to deploy/use our Best in Class (BIC) framework:

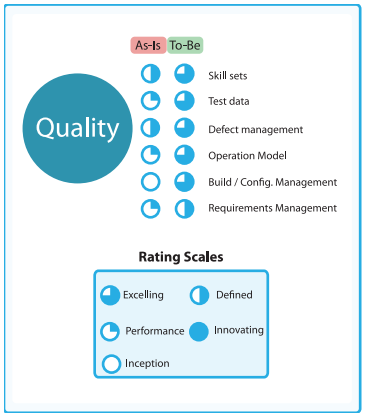
Say, for example, a hypothetical customer has identified the following as critical outcomes:

- Global deliver (Using a 10:90 onsite-offshore leverage)
- +/- 5% estimation accuracy
- +/- 5% schedule variance
- 27% automation savings
- Deliver at CMMi Level cost of quality (25%)

For the purpose of this paper we will restrict our discussion and analysis to just one outcome of the five mentioned. This will be the cost of quality (CoQ). We are making an assumption that this hypothetical organization is already following some standard process for software delivery and that it is delivering at certain capability. The outcome delivered by this capability on the parameters mentioned above may not be at desired levels of performance.

As a framework that can be easily customized for this hypothetical client, we can use BIC to analyze the parameters that contribute to increasing CoQ. The table below shows some of the key parameters and their current levels of maturity.

Calibrating CoQ



The chart above clearly articulates the process areas that have a direct bearing on the cost of quality, their current level of performance, and where they should be in terms of performance for this client to achieve its desired outcome. For example, a build and configuration management process needs to be at improved from its current state of inception. We need to validate what it means for a build management process to be at "inception" level and similarly, what it means to be at "excelling" level.

Based on our experience we have defined it to be as follows:

Broadly Assessing Process Maturity

Sub-Focus Area	Parameters	Inception	Performing	Defined	Excelling	Innovating
Release management	Release planning Release readiness tracking	Adhoc release notes	Release Planning & Tracking	Centralized, integrated Release planning & tracking (release readiness)	Strategic release planning & real-time monitoring	
Config Management	Tool Definition Access levels Version log Code Promotion	Basic CM process definition	Implement config tool	Implement multi-site config tools with branching & merging + enterprise rollout; Integrated with build/migration tools Ability to identify impacted components across systems	Structured defect prevention for CM related defects	
Change Management	Coordinate Implementation Coordinate Build & Test Change Authorization process Change definition & Log	Adhoc change requests; no consistent governance and impact analysis	Change Definition, CRs, Impact Analysis, Logs, Change Authorization process	Coordinate build & test Structured approval guidelines and change review board + tool-based, centralized change mgmt. function; Coordinate implementation Ensure validation of inter/intra system dependencies,		

Similarly, for defect management, this organization needs to move from its current level of performance to the one defined as excelling, which means its defect containment effectiveness will have to be at 85% to 98%. The organization will also need to have a structured root-cause analysis model for defect prevention.

Improving Defect Management

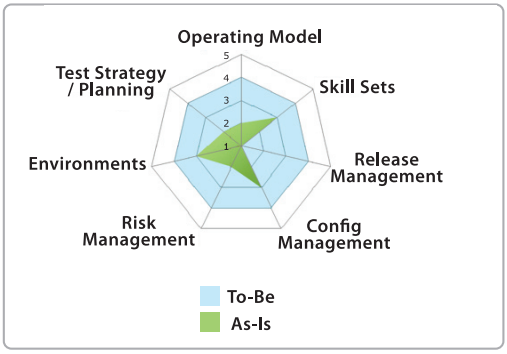
Sub-Focus Area	Parameters	Inception	Performing	Defined	Excelling	Innovating
Defect Management	Defect revention methods + Defect Lifecycle Management	Inadequate: Not all the defects are consistently tracked to closure	Defects logged, analyzed, fixed and tracked to closure	Organization defect database used for defect management activities	Structured root cause analysis for defect prevention	Defect prediction Models triggering innovation
	Defect containment effectiveness	50-60%	60-75%	>75-85%	>85% - 98%	>98%
	CoQ	70%	60%	45%	35%	25%

The next step in the process is to examine how to improve and what obstacles prevent improvement. In this specific case, the following were identified as the primary factors contributing to the high cost of quality/cost of testing and the recommendations to improve quality and lower costs:

- Enhance the current mix to have about 15% of resources with the right mix of product, technology, application and domain.
- Establish and institutionalize a Test Data CoE. Use industry standard tools for data automation and configuration management of data.
- Project overheads such as defect triage, causal analysis, RFQs, test data co-ordination, artifact maintenance, etc. can be reduced by auctioning adherence to SLA/OLA, 24X7 environment support, centralized test data management, etc.
- Negative impacts of parallel testing, SIT and UAT conducted in parallel results in duplication of effort. Where required evaluate alternative test methodologies to compress time (e.g., Risk Based Testing)
- The test approach needs to be flexible to enable altering methodology used for testing from pure waterfall to risk-based testing, depending on critical path and schedule requirements.
- Emphasize on adoption of quality gates. Define and implement queue entry criteria for build migration and quality of fixes.
- Institutionalize smoke and build verification tests to avoid the wrong builds and environmental issues surfacing after significant test effort has been undertaken.
    - In the last three months, 25 builds have been rejected due to poor quality of code. Most of the rejected builds continue to be tested by QA, since the SME waives off the entry criteria.
    - On an average, Cognizant is executing 0.5 cycles, more than what is planned for each project to achieve the desired level of quality. There are instances where 26 to 50 builds of the same application have been tested (Stores) and the defect yield was high.
- Enforce structured release management (ITIL) process for both BAU and large programs. Set up a shared services release management CoEs to manage the release calendar and review release readiness by platform.
  As is usually the case, every customer's business is different. Hence, each customer's operating model and expectations, as well as factors influencing performance levels, are different.

Similarly let us consider another outcome which is on-time delivery. The customer in this case was looking at best-in-class capability and had a goal of achieving +/- 5% schedule slippage. When we analyzed the factors/process areas contributing to this outcome, the following were ascertained:

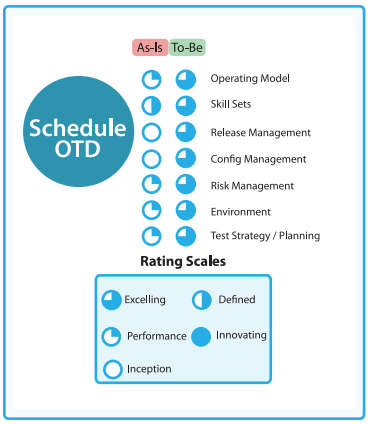
Gauging On-Time Delivery



There were eight sub focus/process areas that were directly contributing to the performance of this outcome. We compared the current level of performance of these sub-process areas with the following metrics:

A Parametric View

We also evaluated the other sub-focus areas and ascertained the current level maturity and identified at what level they needed to operate to deliver a +/- 5% outcome for on-time delivery.



Visualizing Transformation

Parameter	As-Is	To-Be	Best in Class
Demand Forecasting	80%	>80%	>80%**
Defect Removal Efficiency	40%	>80%	>80%*
Asset Reuse	20%	60% on BAL	>60%**
Risk Identification Efficiency		/ 5%	>/ 5%**
Risk Mitigation Efficiency		85%	>85%**

As we map the current level performance and the expected level of performance of these sub-focus areas to deliver a specific outcome, we also identify the means/metrics to help ascertain when the desired level of performance for a specific sub-focus area is reached. This can be viewed in a table that compares parameters across as-is and to-be needs.

We then fine grained our findings and developed a plan which laid out a series of initiatives that will be owned by the current QA organization. The aim is to help them improve sub-focus area performance to facilitate a gradual improvement in our ability to deliver on the OTM outcome.
Among the key challenges identified and performance improvement recommended were:

- Define BAU capacity comprising of IT initiated changes, small changes and regulatory needs. Implement released based testing for BAU by platform. Emphasize on Project/ Portfolio prioritization.
- Identify high level skills by platform and make sure these resources are utilized for SME. Create Core & Flex teams.
- Enhance current, demand forecasting at a project level to improve and cover engagement level requirements.
- Implement ITIL-based structured release management process.
- Quality of fixes for blockers: Over 14% of the blockers when fixed has to be retested more than once
- Establish/enforce SLA/OLA for availability of environment. Considerable time is lost due to application / third-party interfaces not being available. Also need was a24X7 support window for non-production environment.
- Entry criteria did not exist prior to commencement of testing, leading to a higher number of cycles. Enforce entry criteria for code quality before accepting code for test.
- Need dedicated performance test environment. At present, in some areas the environment is scaled down (1:10). In few cases, non-availability of environments is known toward the end (during execution), leading to slippages.

The aforementioned examples provide clear indication of the usefulness of an outcomes-based approach to improve software quality, why it's best-in-class, and why is it different from traditional methods.

Conclusion

In summary, organizations seeking to improve software quality need to focus on the following:

- Deploying a standard process is mandatory for delivering consistent outcomes - in our view that is hygiene. The difficulty with standards is that, beyond a certain point, it does not necessarily lead to guaranteed outcomes.
- A best-in-class approach is all about striving for outcomes first, and then mapping processes that have maximum leverage for delivering that outcome. It also means mapping the means to measure effectiveness of each of the processes deployed and compare then with the best in the industry or across industry.
- Don't expect every process to be at the highest level of maturity for delivering best-in-class outcomes as these outcomes could be very different for each customer and different for different business units within the same company.

- Our approach embraces the entire software lifecycle and elements that contribute to process performance compared with an isolated Software Testing Lifecycle approach. Lastly this approach requires that an organization has some level of consistency in their delivery before they c an embark on a BIC approach and also this is very custom developed for a customer as it factors the customer business/IT environment and various other factors for developing the deployment strategy.

- In our view, the best results for BIC implementations are obtained by organizations that have some level of process maturity to begin with. This enables them to deliver consistent performance and position the BIC to help deliver significant improvement to critical outcomes.

About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services. With over 50 global delivery centers and approximately 88,700 employees, we dedicate our global technology and innovation know-how, our industry expertise and worldwide resources to working together with clients to make their businesses stronger. A member of the NASDAQ-100 Index and S&P 500 Index, Cognizant is a Forbes Global 2000 company and a member of the Fortune 1000 and is ranked among the top information technology companies in BusinessWeek's Hot Growth and Top 50 Performers listings.

Start Today

For more information on how to drive your business results with Cognizant, contact us at inquiry@cognizant.com or visit our website at: www.cognizant.com

Cognizant
Passion for building stronger businesses

World Headquarters
500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters
Haymarket House
28-29 Haymarket
London SW1Y 4SP UK
Phone: +44 (0) 20 7321 4888
Fax: +44 (0) 20 7321 4890
Email: infouk@cognizant.com

India Operations Headquarters
#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com