

TESTING

CONTROVERSIES





Foreword

FOREWORD



With business and technology complexity growing, testing and QA have started to play a fundamental role in today's business. Further, with the scale of testing increasing rapidly, most firms need to do testing faster and cheaper. While there are different test methodologies and strategies for organizations to extract greater and higher levels of test efficiencies, what perplexes organizations is the decision on which methodology to use and when.

In my multiple years of consulting with numerous clients, I have realized that there is no single 'silver bullet' when it comes to test methodology or strategies for solving all the QA challenges that organizations face. Executives ask me whether one test methodology is better than the other, and I always tell them that the question isn't necessarily which kind of testing methodology is better, but to decide which methodology would prove to be most beneficial, given the business or test situation at hand. When you ask that question to yourself, the answer could be a single methodology or it could be a blend of multiple, used in varied proportions. So before deciding one way or the other, organizations should take stock of the variables at hand like type of program - Organizational Priorities, Compliance, ERP Rollout, etc., technology - Cloud, SOA, Mobility, etc., Time and Cost Constraints, Criticality of Business Application, etc., and then determine a test methodology or strategy to be adopted in order to maximize return on investment and accelerate business success.

In our endeavor to communicate this message, we have catalogued some of the most prominent controversies that have beset the world's testing communities for several years now. In our attempt to give you an in-depth overview of some of these controversies like, Automated vs. Manual Testing, Planned vs. Exploratory Testing and Risk vs. Regression Testing, we have come up with this series of papers. The papers are primarily aimed to help you better understand the testing methodologies in question, the roles each of them play and what determines their success under specific business situations.

We hope you find the reading informative and enjoyable.

Regards,

Aparna Sharma
Head of Client Services
Independent Validation and Testing Services, Infosys



Index

I N D E X

1. Manual Testing Versus Automated Testing
A Decision Point
-

6

2. Scripted Testing Versus Exploratory Testing
-

10

3. Regression Versus Risk-Based Testing
Managing Complexities
-

14

Manual Testing Versus Automated Testing

A Decision Point

- Vasudeva Naidu



Automation in a test environment can be expensive and doesn't always yield a high return. In fact, a high percentage of automation efforts fail. In this tip Vasudeva Naidu gives advice on key factors that will lead to success in your automation initiative. Naidu describes the history of automation in test, including reasons leading to failure. Done well, augmented by manual tests, automation can prove instrumental to a successful test effort.

Automation in a test organization can be a somewhat controversial decision. Many automation efforts fail or do not yield the expected return on investment. In this tip, we'll look at specific key factors to ensure success with your automation initiative.

Concerns

Automated test execution has become an accepted and well established software testing strategy in the last few years. Automation is not new anymore and IT organizations understand the strengths, weaknesses and role of automation in improving time-to market and test coverage while reducing manual labor cost of testing. Yet, surprisingly a lot of automation initiatives have failed and hence automation's popularity as an effective and efficient strategy is being questioned by many IT leaders. Equally surprising is the low level of automation of test execution that is noticeable in most IT organizations, across industry sectors like banking, insurance, healthcare, energy, manufacturing, etc., despite investing heavily in procurement of automation tools and infrastructure. Lack of success stories and insufficient data on return on investment is leading to concerns about the initial investments required to automate test execution. In fact, very few IT organizations capture basic automation testing metrics needed to assess the maturity of their automation processes and plan future improvements. No wonder most IT organizations still prefer manual testing strategies. So we are left wondering if automation will ever succeed. What variables ensure the success or failure of automation strategies? Will automation testing replace manual testing?

History of automation

To understand the answers to these questions let us step back in time and understand the evolution and need for test automation. In late 90's, many test automation tools, mostly relying on record and playback, started becoming popular. However most of the automation projects failed since record and playback was not a workable option. In early 2000, new versions of automation tools started emerging with increasing complexity of technology. Volume of testing increased, cost escalated and the pressure to attempt automation came naturally. This resulted in increased investments in

automation projects. Technological changes and legacy modernization increasing maintenance effort, the lack of effective governance to deal with these rapid changes led to more failures than success for the automation projects. With time, testing organizations started realizing the stringent selection and maintenance procedures needed to make automation successful. Even though they lacked a clear understanding of what parameters influenced the outcome of automation projects, this also moved the focus back towards manual testing.

These events provided IT organizations with important lessons and they realized 3 key points:

1. Automation is not just an one time initiative
2. Failure rates for automation are higher than success rates and hence careful planning is essential
3. Automation skills and tools are expensive. Effective governance is instrumental in making automation successful across the enterprise

IT Organizations were witnessing the revolution in technology advancement in the last decade and technology was becoming more complex, applications becoming more distributed and testing becoming an accepted discipline. Organization started realizing the following

1. Cost of testing would escalate and there was an immediate need to improve effectiveness and efficiency of testing processes
2. Automation was still an underutilized testing strategy
3. Automation as a concept had matured
4. Reasons for deciding between manual and automation testing were clear

All this led to the development of automation competencies in organizations which had direction, focus and funding to make test automation a success.

Ensuring successful automation

So how can organizations ensure the success of their automation projects? Some of the key factors which need to be taken into consideration to ensure successful automation are listed below:

Enterprise wide automation governance with clearly specified goals	<ul style="list-style-type: none">• Establish enterprise level automation governance with a clear objective of increasing automation• Automation competence has senior management sponsorship and a steering committee
Automation Process	<ul style="list-style-type: none">• Establish end-to-end automation processes including work intake, ROI analysis, automation testing plans and execution plan• Establish automation project management strategies
Automation Funding Model	<ul style="list-style-type: none">• Establish automation funding models governed by automation competency and adopt self funding models• Provide seed funding for incubation of the competency• Look for pay per use model for both tools and services from service providers and tool vendors
Test Labs, Environments and Infrastructure needs	<ul style="list-style-type: none">• Invest in necessary infrastructure for automation based on Return on Investment• Cross utilize infrastructure across enterprise
Tooling standards and advance automation frameworks	<ul style="list-style-type: none">• Establish uniform tooling standards and create policies and procedures to prevent multiple tools• Build enterprise wide automation frameworks to increase reuse and reduce effort
Test data for automated execution	<ul style="list-style-type: none">• Develop methods and procedures to create test data sheet for automation• Establish data refresh capabilities for faster automation cycle execution
Automation Metrics	<ul style="list-style-type: none">• Identify automation specific metrics which will help in measuring success of automation and provide action items for continuous improvements
Automation training and awareness	<ul style="list-style-type: none">• Provide a platform for people to develop automation specific skills• Publish success stories to increase awareness and acceptance of automation

Clearly there are several parameters which need to be taken into consideration to ensure the success of automation and reduce the percentage of manual execution. Organizations can lay a strong foundation to ensure the success of automation by focusing on the initiatives listed in the above table. This will help improve the effectiveness and usage of automation.

Role of manual testing

While automation continues to evolve rapidly, it is too early in the technological revolution to replace manual testing completely with automation. In fact most of the new features, complex validations and business intensive functionalities will continue to be tested manually. The goal of 100% automation is not just ambitious, it is also impractical. Typically 70% automation helps maximize return on investments. Hence manual testing remains and will continue to dominate in organizations with lower levels of automation maturity and in areas where ROI on automation is not significant.



Conclusion

While the focus on automation continues, manual testing will remain dominant for the next few years. While the urge to utilize automation grows, the urge to uncover the complex paradigm shift is necessary to increase automation and reduce manual testing still needs attention. It is clear that a careful and structured approach is necessary to increase the percentage of automation. Since multiple variables affect automation, a strong foundation followed by a focus on excellence in execution is necessary for succeeding in automation. Very soon, automation of test execution will become inevitable. Organization's preparedness will dictate its testing cost structure and future competitiveness.

The article originally appeared on SearchSoftwareQuality.com at <http://searchsoftwarequality.techtarget.com/tip/Manual-testing-vs-automated-testing-A-decision-point> in February 2011.

[SearchSoftwareQuality and the SearchSoftwareQuality logo are trademarks of TechTarget, Inc.]

Scripted Testing Versus Exploratory Testing

- Vasudeva Naidu



I recollect using exploratory test techniques to uncover hidden defects for a software product organization, in my role as a tester 12 years ago, despite lacking formal training or awareness. It was my urge to uncover hidden defects which drove me to adopt exploratory test methods. Today, even as software testing gurus continue to debate the merits and demerits of exploratory testing, its awareness and practice, with the exception of some software product development organizations, continues to shrink rapidly. When I started thinking about the possible reasons why majority of the testers today fail to adopt exploratory testing, I realized that the gap really lies in the design of the overall test strategy. Most of the test strategies today heavily focus on scripted test methods with limited, or no scope, for the usage of exploratory test methods. There is no doubt that exploratory testing does have certain advantages, but in order for us to leverage the same, we need to revisit our application testing strategies and ensure a fine balance between scripted and exploratory test methods and tools. This balance would ensure increase in the effectiveness of the testing function without compromising testing efficiencies.

What is the Current Role of Scripted Testing Versus Exploratory Testing?

In the last 15 years, as testing methodologies have undergone several changes, scripted testing has taken center stage. QA teams currently focus on ensuring the application under development complies with the stated requirement specifications, rather than trying to evaluate the behavior of the application under development through

- Extensive application and business knowledge
- Innovative and creative methods simulating possible user behavior
- Ensuring thorough evaluation while managing time

Some of the possible explanations/situations that led to the industry-wide adoption of scripted testing:

- a. Pace of technology advancement (from colossal Mainframes to iPad) has significantly driven change across IT systems. In turn, this has increased the volume of testing required drastically.
- b. Insufficient availability of SMEs to keep up with volume of testing
- c. Rising costs and the need to improve efficiency necessitated commoditization of the testing function. This resulted in reduced involvement of SMEs in the testing process.
- d. Improved IT Project management practices pushed the need for better control on budgets and the outcome of the Quality to be quantified. This has significantly pushed scripted testing in comparison to exploratory testing.
- e. Requirements management continues to be a pain point in many IT organizations resulting in a push to strengthen traceability of requirement to test cases

- f. As the size of applications increased and business process flows started to span multiple applications, it became necessary to adopt a more process oriented approach to test these complex systems. This is precisely what scripted testing offered.

Why we need Exploratory Testing and its significance in today's world

While IT organizations across industry verticals continue to improve the quality of their product, according to my experience the defect removal efficiency of the overall IT organizations has not improved beyond 85%. However, a majority of the testing groups within the IT Organizations have transformed themselves and become more efficient, reporting defect removal efficiencies of 95% to 98%. Most IT process gurus are analyzing this situation and trying to devise strategies to address this big quality gap that exists between what testing groups are claiming and what the overall IT application quality really is. It is very clear that quality is just not a testing problem and the entire IT organization needs quality focus to improve the defect removal efficiency of the entire IT organization from the current levels. One such strategy that can be adopted to ensure the same is exploratory testing. Of course it would need to be carefully married to the organization's master test strategy in order for it to yield the planned improvements in quality.

Implementing a combined strategy

Even though most testing professionals focus on scripted test execution, the natural tester's instinct pushes her/him to explore the system completely in the available

time. Structured methods and initiatives can improve the efficiency of exploratory testing when carefully combined with mainstream scripted execution. While attempting to devise a successful strategy one must keep in mind certain ground realities:

- IT organizations continue to struggle with SME availability
- To continue to provide testing resources aligned to an application/domain is not a best practice due to increasing cost of QA. This especially is not worth it when most CXOs are looking to either maintain

or reduce cost of testing in comparison to overall project cost

- Organizations' inability to measure and track Cost of Quality

However despite these challenges, testers are encouraged to think beyond their test scripts thanks to the focus of IT organizations on quality of applications. Table 1 outlines a few initiatives which I believe will help in successfully adopting exploratory testing to augment scripted testing:

Table 1: Suggested Initiatives to encourage exploratory testing

PARAMETER	SUGGESTED INITIATIVES TO ENCOURAGE EXPLORATORY TESTING
TEST METHOD/PROCESS	<p>Listed below are the methods which motivates and encourages a tester to adopt exploratory methods in addition to scripting methods</p> <ul style="list-style-type: none"> • Defect classification based on business impact if not detected • Quantifying business value to defects gives the sense of importance of a tester role in improving ROI • Encouraging early validation techniques including statistic testing of requirements, static testing of design • Validating Product design quality • Encouraging non functional requirements validation and performance profiling • Predictive profiling of defects • Closer collaboration of testing and production support team to provide feedback on defect trends driving creative thinking
TEST METRICS TO HELP IDENTIFY EXPLORATORY TEST NEEDS	<ul style="list-style-type: none"> • Testing effectiveness to identify the extent of exploratory testing needed • Annual Production support cost reduction % • Requirement stability index • Decrease in test case effectiveness (Ability to uncover defects)
PEOPLE	<ul style="list-style-type: none"> • Definitive role for SME in testing function who can really adopt exploratory methods • Business/Application steward role who can clearly identify the changes in the business transaction flow with every new change request or enhancement • Sufficient application training mandates



Conclusion

While I agree with industry gurus on the importance and role of exploratory testing, its role as the primary test strategy, as argued by many, is questionable due to its inability to support large volumes of testing and control the behavior of large test teams. At the end of the day economics dictate what is feasible and what isn't and industry indicators definitely show a move towards the commoditization of the testing function. While the concept of introducing quality into the product and pushing for early detection of defects is forcing QA teams to develop test strategies which adopt exploratory testing and integrate it with scripted testing seamlessly. It is also shifting the focus towards validation at the level of business transactions, grooming of SMEs and building business stewards (owners of a business process or end-to-end business transaction) who will have ability to adopt exploratory testing methods effectively. Training and awareness is also a key contributor for the success of exploratory testing methods in this ever-growing tester's community. QA managers must implement exploratory testing techniques as an additional strategy, combined with scripted testing, to improve the overall quality of applications.

*The article originally appeared on SearchSoftwareQuality.com at <http://searchsoftwarequality.techtarget.com/tip/Scripted-testing-vs-exploratory-testing> in March 2011
[SearchSoftwareQuality and the SearchSoftwareQuality logo are trademarks of TechTarget, Inc.]*

Regression Versus Risk-Based Testing

Managing Complexities

- Vasudeva Naidu



The role of regression testing is well known to IT industry. However, we always hear about the challenges regression testing poses. Based on my experience, I have noticed that more than 80% of organizations do not have a well-defined regression test strategy in-place even for the testing of their most critical business applications. If we all know its importance, why do organizations fail to establish a well defined regression testing strategy?

When I started analyzing this practice across 10 different organizations, which I had consulted for over a span of 3 years, I realized that the devil lied in the details. Recent increases in IT system complexities have introduced several new variables effecting the successful deployment of regression testing. Added to this, IT organizations are not adopting advance regression test strategies to keep pace with the changing IT dimensions. Risk Based Testing (RBT) is once such strategy which is barely used by organizations today. But in my view, this form of regression testing will eventually replace conventional regression testing methods in the near future.

Why do organizations struggle with regression testing?

Majority of testing organizations build regression test beds by accumulating test cases from several releases. Execution of these test beds is based on the availability of time, test environment and data. While the testers executing these regression test sets acquire skills by practice, they do not undergo any regression testing specific skill development or training to help them adopt advanced methods of testing. This results in the following gaps

- Lack of well-defined strategy for the creation, maintenance and execution of regression test suites
- Lack of prioritization of test cases based on business criticality
- Lack of requirements to test case traceability which needs to be updated after every release
- Lack of regression testing specific test environments and test data strategies
- Lack of impact assessment due to release specific changes to build critical regression test suite and the
- Lack of regression test automation strategy

Addressing these challenges is not easy unless undivided attention is provided to each and every point mentioned above. However, if these challenges exist today, then the question that comes to my mind is how are organizations managing their regression testing needs today? Knowingly or unknowingly, most of the organizations are meeting this demand by adopting “*Risk based testing, without proper risk assessment*”. This essential means regression testing is executed on the basis of time and resource availability. This approach significantly increases the probability of missing several regression defects, thus increasing the production

support cost considerably. The lack of an effective regression testing strategy can be attributed to 3 main points:

- Lack of business domain knowledge
- Increasing size of the application under test &
- Increasing size of application level regression test strategy (selection of regression test set and maintenance)

Role of risk based testing

Risk based testing approach addresses the *size and complexity parameters* of a regression testing in a methodical manner. Theoretically, we all know that there are infinite number of ways in which a product can fail. Practically speaking, this is the same with applications too. However, most business users and testers tend to test an application for every practical use. In doing so, they end up in challenges as described in previous section - increasing size and complexity of regression testing required. Risk based testing can help address this size and complexity issues by

- Forcing to prioritize test cases by clearly defining importance of a functionality or feature
- Encouraging impact assessment to identify likelihood of failure, thus addressing the testability of a requirement
- Improving testing effectiveness by ensuring testing of all critical functionalities, which are used most often
- Increasing test case effectiveness by eliminating non-productive test cases and
- Identifying test data needs during the planning phase itself

In spite of these positives, Risk Based Testing is still used or deployed in a very limited manner across organizations. The top two reasons for this are

- The lack of well defined methods to measure the success of RBT and
- The lack of stake holder involvement in the RBT planning exercise.

Making Risk Based Testing (RBT) a Success

Most test managers have heard and known the concept of RBT for a while now. However, its usage continues to be limited despite the fact that it can help us to measure the Quality Risk more quantitatively and also reduce the testing effort required significantly for regression testing. Below are a few suggestions that can help you in successfully implementing RBT in your respective organizations.

Table 1: Making RBT Successful

LEVER	SUGGESTED INITIATIVES/PLANNING STEPS
<p>AWARENESS/SENIOR MANAGEMENT BUY-IN</p>	<p>Identify a RBT champion with the primary responsibility of ensuring successful implementation of RBT and its adoption across the enterprise. His role would be to</p> <ul style="list-style-type: none"> • Build a business case for the need of RBT • Senior Management buy-in on the business case • Build communication material and identify stake holders • Conduct training to increase awareness of RBT and its usage across the enterprise • Identify pilot projects and supporting metrics • Identify process changes across life cycle stages • Identify roles and responsibilities of stake holders across life cycle in the RBT process
<p>PROCESS BUILD OUT AND PILOT</p>	<p>Some of the key process assets and changes which are necessary for success of RBT are</p> <ul style="list-style-type: none"> • Checklist to evaluate the need for RBT • Tailored test strategy for RBT • Risk catalogue for evaluation • Risk assessment and prioritization framework tailored to organization/application • Metrics to measure benefits and production issues due to RBT <p>While the processes are built, it is important to select the right project for pilot in which the RBT champion will be able to demonstrate quick wins. One of the techniques to conduct pilot is to apply RBT on a project, without actually executing RBT itself. It means that while the project team is planning and executing regression testing, the RBT champion should evaluate and plan RBT simultaneously for the same project. Post execution an evaluation of the risk assessment and prioritization framework's effectiveness. This feedback will provide you confidence on RBT strategy for the actual implementation. This will also improve acceptance and adoption by all stake holders.</p>
<p>MEASURE AND DEMONSTRATE</p>	<p>Showcasing the success of RBT quantitatively is key for its successful adoption. Tools, Metrics and Stake holder involvement is key to make this happen</p> <ul style="list-style-type: none"> • Involve business, system analyst and development in risk analysis and prioritization • Identify right tools to manage requirement traceability matrix, defects and test execution reports • Measure metrics including Requirements coverage, Cost of Quality, Test effectiveness, Defect density and failure rate to understand the effectiveness of RBT and to take corrective actions



Conclusion

The importance of end-to-end regression testing is taking center stage as business and technology complexity is increasing and application sizes are growing. The effectiveness of regression testing will either have “Law of diminishing returns” as the regression test case effectiveness will decrease or the business value of testing will be continuously questioned due to failures in identification of regression defects. Risk Based Testing is the answer to address both the above mentioned situations. The increasing distributed nature of IT topology will swallow the old school of thoughts around full blown regression testing and Risk Based Testing (RBT) will soon take center stage as the main stream regression testing strategy across organizations.

The article originally appeared on SearchSoftwareQuality.com at <http://searchsoftwarequality.techtarget.com/tip/Regression-vs-risk-based-testing-Managing-complexity> in July 2011

[SearchSoftwareQuality and the SearchSoftwareQuality logo are trademarks of TechTarget, Inc.]



Vasudeva Naidu

vasudeva_naidu@infosys.com

Vasudeva handles sales and client relationships for several Insurance and Healthcare clients. With over 13 years of experience, he has helped companies define and implement testing efficiency improvement programs and has successfully deployed TCOEs (Testing Center of Excellence). Vasudeva's key professional achievements include growing and managing large testing teams with team sizes exceeding 400 people. He has implemented several testing thought leadership initiatives over the years.

Learn more

www.infosys.com/testing-services

Infosys® |  SearchSoftwareQuality.com

Global presence

Americas

Brazil: Nova Lima **Canada:** Calgary, Toronto **Mexico:** Monterrey **United States:** Atlanta, Bellevue, Bentonville, Bridgewater, Charlotte, Fremont, Hartford, Houston, Lakeforest, Lisle, Minnesota, New York, Phoneix, Plano, Quincy, Reston, Southfield

Asia Pacific

Australia: Brisbane, Melbourne, Perth, Sydney **China:** Beijing, Dalian, Hangzhou, Shanghai **Hong Kong:** Central **India:** Bangalore, Bhubaneshwar, Chandigarh, Chennai, New Delhi, Gurgaon, Hyderabad, Jaipur, Mangalore, Mumbai, Mysore, Pune, Thiruvananthapuram **Japan:** Tokyo **Malaysia:** Kuala Lumpur **New Zealand:** Auckland, Christchurch, Wellington **Philippines:** Metro Manila **Singapore:** Singapore

Europe

Belgium: Brussels **Czech Republic:** Brno, Prague **Denmark:** Copenhagen **Finland:** Helsinki **France:** Paris, Toulouse **Germany:** Eschborn, Frankfurt, Stuttgart, Waldorf **Greece:** Maroussi **Ireland:** Dublin **Netherlands:** Amsterdam **Norway:** Oslo **Poland:** Lodz **Russia:** Moscow **Spain:** Madrid **Sweden:** Stockholm **Switzerland:** Basel, Geneva, Zurich **United Kingdom (UK):** London, Swindon

Middle East and Africa

Mauritius: Reduit **UAE:** Dubai, Sharjah

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.



Building  Tomorrow's Enterprise

For more information, contact askus@infosys.com

www.infosys.com