

# Tool Smiths in Off-Shored Work: Socio-Technical System of Quality Testing in India

Full Paper

Author 1

Author 2

Author 3

## ABSTRACT

The off-shored IT industry in India is perceived to be uniformly overhauled by automation replacing manual work practices. The consequence of viewing work automation as a face-off between human potential and smart technology denies the role of business acumen, market forces and social contexts that provide for and constitute work environments in any country. We adopt an anthropological approach and a socio-technical framework to understand the arrival of automation work practices in the quality testing segment of the IT industry in India. We highlight key socio-economic parameters influencing decisions to automate a testing environment and analyze the ensuing discourse emerging from the voices of Quality Testers imbuing value into the testing job profile

## KEYWORDS

IT Industry, India, Ethnography, Quality Testing

## 1 INTRODUCTION

India is referred as ‘the testing bowl’ in the off-shored IT industry universe - a stomping ground to arguably a million Quality Testers [henceforth QT] constituting a substantial percentage of the IT industry work force in the country. The past few years have seen the advent of automation replacing manual driven quality testing work practices, slowly and steadily changing the environment of job security, hiring patterns, work practices and job profiles. The paper is situated in the contextual universe of the off-shored IT industry in India to focus on the impacts of automation in the QT

job segment, and is the first attempt at synthesizing the initial understandings of a broader long-term project titled ‘The Socio-Economic Impact of Work Automation in The Off-Shored Job Sector in India’.

Our work is entirely based on ethnographic methods and our research outputs are a qualitative analysis of quality testing work in times of process driven work flows and automated testing replacing manual work practices. The paper draws from a four-month study of work culture and praxis amongst quality testers in the Indian city of Hyderabad, a bustling IT hub in South India. We refer to Quality Testers in India as ‘tool smiths’ in reference to the discourse arising from the IT work community in response to the low bar to enter the QT job segment, the nature of repetitive and recursive work practices, and the subsequent erosion of their work with the coming of automation. This paper will analyze and order the QT segment’s response to the idea of being called a ‘tool smith’ through a counter discourse vesting skill, creativity and productivity to QT as a job.

The past few decades is a witness to India’s emergence as a major player in the off-shored IT industry. Global IT spending is estimated to be \$3.5 trillion in 2017, with IT services forming the second largest spending segment of \$.95 trillion [12]. The global test automation market is expected to expand at a CAGR of 23.6% from 2016 to 2024, reaching US\$ 85.84 by the end of the forecast period [18]. Given India’s significance as a job haven for low cost IT outsource destinations the increasing adoption of automation practices over the past few years, aided by the availability of free, open-source automation tools, has become a cause of concern. High level business reports predict threat to jobs, high rates of unemployment, layoffs, particularly significant slowdowns in hiring for low-skilled, repetitive work that Indian workers specialize and, currently, automation processes are eroding into [11,43,46]. The consequence of viewing work automation as a face-off between human potential and smart technology denies the role of business acumen, market forces and social contexts that provide for and constitute work environments in any country. In this paper, we argue for and make visible the social processes surrounding business decisions to deploy automation in everyday work practices and the consequence of these decisions on Quality testers and their counter discourse to make their job profiles relevant to today’s technology-driven work environment. We employ an anthropological approach rooted in Science and Technology

\*Produces the permission block, and copyright information

<sup>†</sup>The full version of the author’s guide is available as *acmart.pdf* document It is a datatype.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICSE 2018

© 2018 Copyright held by the owner/author(s).

Studies (STS) frameworks to analyze the complex socio-technical system of a work place and the transformations brought about by the steady intensity of automation processes in the software industry. The introduction of a technological innovation into a work context is not simply a shift to more productive, efficient and profit creating business - it's a process that sets in motion formal transfiguration of everyday work practices, cultures alongside the restructuring of policy and legal frameworks, institutional setups and job profiles. However, these technology-driven shifts encounter existing and ground level processes that inevitably confront, shape and sometimes limit the acceptability of new technological processes. An STS framework of socio-technical systems [19], elaborated on in the literature review, offers ways to understand and analyze this two-pronged process in which contexts of work and the arrival of a new technology innovation together shape the level of productively, efficiency and business impacts any industry is able to attain. Our research uses the STS framing to stake a claim about the entry of automation in the Indian software industry as significantly dependent on the social, economic and political contexts that the IT industry is embedded in. The interplay of several factors- the type of industry [a service or a product company], the nature of ownership [multi-national or local], company/leadership vision about automation, tradeoffs between efficiency and costs, nature of business contracts with clients, the specific work cultures [work hierarchies, reward structure] prevalent on the shop floor are critical in determining the scope and specifications of technology innovation and their introductions into an industry. Automation, being profoundly disruptive of the work context it enters, requires a more nuanced understanding of its unfolding and adoption into a specific work milieu.

In this paper, we attempt to explore and discuss three broad research aims:

1. to design and frame a qualitative research method exploring software testing work practices in order to illumine a QT work culture
2. to highlight key socio-economic parameters influencing decisions to automate a testing environment
3. to analyze the nature of shifts in the domain of quality testing job profiles and the consequent discourse arising from the voices of quality testers.

We hope to elucidate our claim that software testing and automation is more complex than a purely technical endeavor, hence any efforts in the direction of greater value generation must consider non-technical factors that play a key role in shaping any industry.

We begin with a review of extant literature framing the more theoretical aspects of our paper offering a more statistical alongside a socio-technical framing of work contexts in the Indian IT industry; we move on to explain our qualitative methodologies in collecting and synthesizing primary data that will go to inform our learnings; in the findings section, we showcase results of our data analysis; in our discussion we attempt to synthesize first-person

voices of testers in the industry in their endeavor to render the QT job segment relevant, creative and productive.

## 2 LITERATURE REVIEW

Quality testing and assurance form the backbone of the Indian IT industry, which is the haven of cheap offshore labor that takes care of banal tasks of software testing and maintenance [17,18] The repetitive, recursive and predictable nature of this work makes it seem like an ideal target for automation, with India showing a 52% potential for technically automatable work, just behind the highest country at 56% [11]. The assumption in such an analysis is that automation tools deployed in other countries will work with the same efficiency in India given that work practices, processes and profiles in the Indian IT industry broadly remain the same. The flaw in this assumption emerges from a ground level survey referring to the lack of complete replacement and the persistence of manual work in work teams that are at the receiving end of the deployment of automation tools in their work a day. Indeed, 42% of teams surveyed still rely heavily or entirely on manual testing tools [39].

The technical reports on the coming of automation processes, are offered from the point of view of industry experts and high-level management that do not account for ground level imperatives the bring automation to the shop floor; moreover these reports often ignoring the reasons for the persistence of manual testing in the face of automation. While there is an increasing acknowledgement of the importance of 'people' in the IT industry [7], with numerous blogs and articles talking about the necessity of 'people skills' to progress in the industry, very little work has actually looked at people - both within and outside the industry - as individuals whose life histories, biases and skills have a significant bearing on introduction, design and output of technological tools [24]. There has been research into the human aspect of software engineering in general such as group dynamics [3, 26], organizational structure [16], management and organisational culture [31] but the bulk of this work has looked at isolated engineering practices. Work that has taken a more holistic view of software engineering seems to focus more on the development side, ignoring testing [35]. This sidelining of software testing in research, quite telling by the constant conflation of 'software engineering' with 'software development', can only be detrimental to the industry as a whole, given that testing on average accounts for 50 percent or over of total project costs [4, 43]. With efforts being made to reduce these costs via automation, the need for research into how current testing structures and processes are maintained and how they shape test automation is becoming more pressing. This is not to say that no work has been done on the subject. Feldman [9] talks about the importance of treating Quality Assurance as a process that shapes the project as a whole, and not a last minute add on. Several studies discuss the intricacies involved in automatically generating test data and the insufficiency of current metrics in gauging the quality of test data, which often needs contain errors in order to check software features [15, 27, 28]. Others draw a comparison between manual and automation testing, often to propose statistical models which can predict an optimal percentage of automation for a particular project [33, 44, 28, 32, 14, 13]. In doing so, manual and

automation testing are treated as two distinct ways to perform testing processes while they are often in actuality two different yet work processes that work synchronously [ gleaned from our own research]

We feel that an anthropological approach rooted in Science and Technology Studies (STS) frameworks can provide a way to analyze the complex, multi-layered and multi-sited system at hand. STS accounts recognize that history and culture matter, but not as a straightforward action-reaction model. A comprehensive framework that situates technologies within integrated material, moral and social landscapes is provided by the idea of socio-technical imaginaries. Socio-technical imaginaries are ‘collectively held, institutionally stabilized, and publicly performed visions of desirable futures, animated by shared understandings of forms of social life and social order attainable through, and supportive of, advances in science and technology’ [19]. This analysis implies that on one hand, the introduction of a technological innovation invariably pushes ethical, legal and moral boundaries of acceptability, forcing restructuring of policy and legal frameworks, institutional setups and job profiles. On the other, existing ideas of the scope, limit and acceptability of technological processes shape the contours of the human-computer systems that emerge - the way in which people perceive machines and their scope structures the interactions that occur between humans and computers, deciding the level of efficiency the system is able to attain. The application of such a framework requires following socio-technical systems through four phases; the origin of new technologies and the social arrangements they help sustain or rework; the embedding of these ideas into cultures, institutions and material channels; period of resistance or conflict with older ideas that make clear and visible the social apparatus that supports each of the two ideas; and extension, the complex of processes by which ideas gain traction and cross scales (of time and geopolitical boundaries) [20]. At each phase, we examine collectives and institutions to understand why it is that the technological development emerged or advanced at precisely that time. Our work uses this frame to demonstrate that the shape work automation takes in the Indian IT Industry depends heavily on the existing culture into which it enters, as well as international relations of power and knowledge exchange. The history of education and technical training, pre-existing work profiles, organizational structures and power hierarchies between employees all become critical in determining the spaces which automation is seen as fit for, its reach and scope. Broadening the horizons of this potentially disruptive technology in a direction that will boost creativity, work and productivity requires grasping the present imaginaries and their origins, and thus identifying the channels of work, communication and leadership vision in an industry.

The shift from manual to automation, or the use of both, is not a simple replacement but a restructuring of workflows. The ‘efficiency’ of automation cannot be grasped without an analysis of what this restructuring entail, along with context-specific human and technological limits to automation [11]. Our work attempts to more clearly delineate the exact, ground-level, day-to-day processes that are involved in automating QT services; the negotiations entailed in deciding when to automate and which tools

to use; the interplay between manual and automated testing; how both are often used to supplement each other; the reactions of stakeholders to automation, and how they skill themselves to deal with changing technology. Through this preliminary attempt, we hope to lay a foundation for more focused enquiry into the specific, context-driven considerations involved in introducing technological tools, and how these can be leveraged for better integration of such tools into the existing environment.

### 3 METHODOLOGY

The paper is derived from qualitative methods to develop a more social and contextual understanding of the QT job segment in India - both work practices and the more subjective voices of employees. We undertook three different methods for data collection that were carried out in parallel, with insights from one method being used to inform further probing using others: a deep investigation and analysis of blogs maintained by quality testers occupying diverse career paths in the Indian IT industry; depth interviews, both face to face and video calls, with quality testers from Hyderabad, Bangalore, Chennai and NCR; ethnography of a IT skill tutoring hub in the city of Hyderabad where the youth and mid-career professionals flock to upgrade their IT skills.

#### 3.1 Blog Reviews

The (often anonymous) space of the internet is a rich resource from which to access candid first person accounts of work environments. In our case, we conducted an extensive review of blogs written by testers and ex-testers, including comments, social media sites and chat forums dedicated to software testing. We engaged with blogs maintained by three quality testers beginning from 2004 and their regular commentators; more than two dozen blogs, that engaged with quality testing but not limited to it; a few global forums of quality testers alluded to by the Indian bloggers. The high traction these sites receive in the form of comments and length of discussion threads, despite some having been defunct for over 3 years, are a testament to their resonance with the thoughts of testers across the country. We came across day-to-day frustrations with banal work, puzzlement at archaic management and bureaucratic techniques that clearly impaired work, as well as informal survival guides and manuals for newbies. These were essential in pointing us towards specific research questions, and became the starting points from which to think of reconfiguring quality testing as a research site, providing a major critique of existing attitudes shaping work practices in the QT domain and informal ‘jugaads’, or workarounds to many everyday problems testers face on the shop floor.

#### 3.2 Interviews

We conducted a set of 18 open-ended, in-depth interviews with early-career, mid-career and senior testers in the IT industry in India, as well as industry leaders. Through these interviews we traced daily routines of testers, organisational structures and processes, their projected career paths, views on automation and the impact they had observed on both immediate tasks and long-term processes. We gleaned the considerations that go into the on-

ground implementation of test automation, the extent and reach of automation in India and the responses of those immediately affected to the changes automation brings.

### 3.3 Ethnography

Since June 2017, we have been conducting fieldwork in Ameerpet, a dense bustling commercial suburb of Hyderabad city and the largest IT skill tutoring hub in India, largely run by the private sector, offering a plethora of courses ranging from robotic process automation to manual testing courses. On one hand, interactions with students gave us a good baseline for the profiles of future employees entering the testing space in India, their perceptions of their work (and hence a projection of how they perceive efficiency in that work), and the kind of training they receive (which directly impacts how they go on to do their work). On the other, the instructors for these courses are IT employees themselves - either currently senior testers or co-owners of test outsourcing companies. These instructors provided valuable insights into interview, hiring and real-time testing practices in India. Interviews with them are another key resource that informs our understanding of current testing processes, the degree of automation in testing and QT work culture in India.

## 4 FINDINGS

This section will outline our findings in relation to ‘the everyday’ routines of testing, their implication for a manual tester’s work profile and the shift that work automation tools bring to both a conceptual and hands on level of the quality tester’s work a day. We first provide an overview of the day to day work of testers ; next, we lay out the stakeholders, who are essentially the managers and testers in the industry and the decision making process for introducing automation tools; we then consider this stakeholder view point on the potential limits and advances of manual and automation testing practices We begin by giving a current standard of manual testing, automation testing, and an average day’s workflow, deduced from interviews with testers. Our first sub-section highlights documentation as a significant but sometimes efficiency hindering practice in the tester’s work.

### 4.1 The Everyday of Testing

*‘First... the client requirement is analyzed by the functional analyst or business analyst. Then he prepares the BRS document (Business Requirement document) according to which the team prepares the test cases...an experienced person will go over the BRS and decide the scenario. Once it is finalized, then the whole team will start executing it.’ – Karthik, Male, mid-20s, Junior tester in a multi-national IT service industry*

The day to day work of a tester is highly dependent on the choice of a development methodology for a product. The product development phase begins with the task of documenting specific requirements and drawing out test plan, on the completion of which the testing of the product begins: however, this kind of testing clearly separates the documentation and the practical phases of testing. Current testing practices of individual modules begins

almost in parallel with development mostly executed by business analysts and senior testers with help of automation scripts to run and test repetitive tasks [Junior testers who earlier documented the test cycles often don’t participate in designing test cases]. Once tests are designed, they need to be logged into the test-case template as well as the requirement traceability matrix. Testers update the test-case template each time they execute a test case; a module is passed when enough for it is passed and results matching client specifications. This refers to the kind of software development methodology decided at the start of the project by the managers, clients and business analysts. The choice of methodology depends on a range of factors, including available resources, budget, and timeline of project. The methodology will impact not only when a tester begins to work, but also the amount of interaction they have with the development team, the building of the product and their input into designing test cases. The type of ‘development’ cycle also impacts the product’s testing life cycle: For example, the ‘waterfall’ style of development phase is a period of documentation for the tester in the post-development phase with little interaction between the developer and tester. In ‘Agile’, another kind of testing life cycle steadily replacing the waterfall model, there are regular team meetings and testing of individual modules begins almost in parallel with development. Ranjan, a mid-career tester, told us ‘In Agile ... [development team] will be cross-functional, which means that all members will know some part of each other’s skills. This helps in validation, and improves quality of the product’. Much of the variation we encountered in testing practices can be accounted for by the size of the team and interactions between testers, developers and business analysts or domain experts. In larger teams, particularly in large companies, documentation plays a key role in communication, roles are more rigidly defined and there is less scope for innovation as an entry-level tester.

[Figure 1](#) illustrates a typical testing lifecycle workflow, as described by our informants. Starting from the top left, the clients provide the requirements from which the managers and business analysts produce requirement documents.

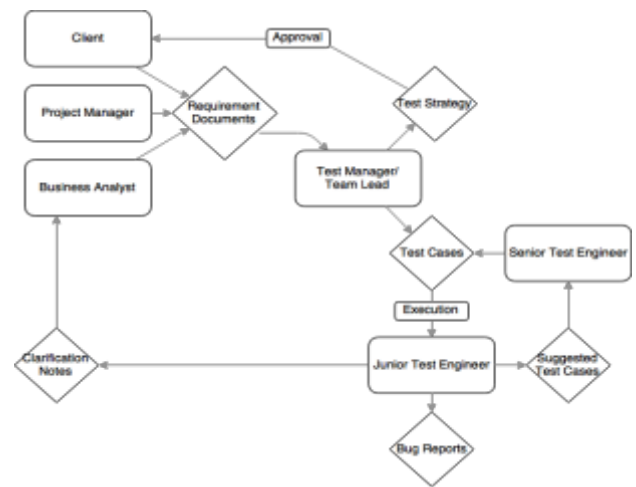


Figure 1: A typical testing lifecycle workflow

These documents are given to the Test Leads, who design a test strategy and present it to the clients. Once this strategy is approved, the senior testers in the team create test cases, passed on to junior testers who take care of execution. The level of involvement of senior testers depends on the complexity of a module.

As Karthik describes 'In complex projects, senior person is needed to make the test cases after understanding the requirement. He may not make the details of the test case ... but he'll make the overall frame and then the entire team will execute.' In case of any doubts, the analyst is consulted for clarification as they are most familiar with the requirements. Junior testers may also suggest new test cases, which are approved by a senior then incorporated into the test plan. Bug reports are submitted to the development team, which may respond with an updated module or leave the fix for later depending on the priority of the bug. This process continues till the module has been tested on all the decided test cases. The level of involvement of senior testers depends on the complexity of a module.

#### *4.1.1 Testing as Documentation*

The logging of individual test scenarios, cases, runs and the output forms the backbone of testing. The variation in the nature of testing practices is not a small one in India - While small firms taking up outsourced projects, formulation of test cases begins manually outlining each step in an excel sheet! In contrast, a typical multi-national IT service firm is an environment with little physical communication between large groups of developers, testers and managers working on projects spanning over months, documents become the primary mode of communicating tasks accomplished and to follow. In such a large set-up, test cases are necessary markers for tracking project trajectories. Testers acknowledge documents as maps to project lifecycles.

However, the ossification of documentation procedures cut into the timelines and efficiencies of projects. Rather than being a framework, documents turn into rigid procedural guides for running test cases. Testers need to go through a series of approvals to modify test cases rather than use self-made judgements to figure out the life cycles of a test case. Thus, the work of a junior tester becomes the banal task of following the step-by-step test case execution of documented procedure, dutifully logging the generated output. There is little contingency for exploration of the software or creative generation and running of test cases: As a senior tester quipped, "There are industry segments [such as medical equipment quality assurance] where test cases run by humans are absolutely required... but in most domains, even in entertainment industries [for example], they bore testers to death by making them do test case documentation." Not only does this excessive focus on documentation decrease attachment to work, consuming large amounts of time and effort, leaving testers mentally drained, it's positioning as a key metric by which testers are gauged has created a counterproductive work environment. While importance is given to the severity of a bug a tester manages to detect, their daily work output is measured by number of test cases executed and documented. Thus, the improvement of the product at hand takes a backseat and logging a greater number of cases becomes more important from a career advancement point of view.

The work perception hence created - of testing being equal to simple execution of a pre-given set of steps, and the logging of results - positions human work as robotic activity ideally conducive for an automation takeover. However, the envisioning of testing activity structures the form of automation poised to replace it. According to Pradeep, a senior tester and leader of a QT firm in Bangalore, "The reason test cases exist is to create a finite set of paths [to traverse and check the product] ... it requires skill to map out things you've already done. Unfortunately, this is the skill that's not taught - to create the map and not just follow it." Thus, automation is perceived as a work practice replacing or taking over the easy routine renditions of a test-case 'path'. A QT blogger Shrink suggests: "... We claim that this kind of automation, which only boost number of test cases run in a given time, is an inefficient use of technology which replaces humans with automation software but doesn't subsequently reallocate human effort and creativity to restructure work processes more creatively...". As Kiran, a young QT with a giant multi-national service firm, suggested: "... as testers we can map out the possibilities for new kinds of testing processes that automation opens up - we can smartly come up with more efficient classification and logging of bugs, the collection of data from test runs for analysis and better utilization of the time that we testers gain from automation in order to enhance the scope of our job profile..."

#### *4.2.2 The Slow [or Fast] March of Automation*

An overview of the QT blogosphere in India would suggest that IT firms are under the slow [or fast] march of automation, to increase levels of automation in order to achieve error-free and high-quality software products with a short run time to market. The introduction of automation testing in a project can have significant time and cost investments, which need to be approved by the client. As Shravan, a junior test engineer, in an Indian IT service giant company explained to us: "...testing is a service, we are a service company. Thus, the final decision of what to use rests solely with the client". Vishnu, automation director of a mid-sized QT firm, with over 15 years of experience added, "... though we do suggest options to the clients, once we've assessed the project... the decisions to follow a work flow or process is highly indicative of the type of client, their legacy systems that still control technology/platform choices, above all the attitudes towards, automating processes, cost trade offs..."

Three primary stakeholders emerge as decision makers in the process of automating a work stream

1. Clients are primary decision makers for a service industry. The client's perception of automated tools and the costs involved are critical- Balaram, a Test Director, at a mid-sized Indian company exclusively devoted to testing services, explained to us: "... some of our clients 'trust people more than tools' ... even when automation provides efficiency, manual testing methods persist. However, clients working in startups or with software that requires quick version updates are more likely to opt for automation..."

2. The leadership team in a company who evaluate automation costs on the basis of the following: on one hand, there is increased efficiency and shorter project turn-over freeing up resources for other projects. On the other, automation brings new costs and a new work culture requiring a period of intense business and cultural adjustments
3. Test Engineer: The engineers designing, executing test cases, working with tools, experience the least decision-making power to optimize automation practices. Testers grapple with contradictory impulses towards automation seems depending on their present skill level, opportunities and desire for up-skilling- Rahul, a tester with 3 years of experience in one of India's top 5 IT Services companies said: "...automation is very good for the testers, but they also need to upskill to learn..." Pradeep adds to this sentiment, "...They [testers] love it as long as they're a part of it. They want to be called an automation tester. They want to put it on their profile that they also know Selenium..."

We are seeing a recent trend in the visibility of voices like Monica Paul, senior blogger and a pro-automation voice: "...Clearly, automation testing is no longer an option anymore...not if you want to achieve optimal resource utilization while getting the product to market!"

While the jury is still out on 'script-less test automation', this process re-formulates coding programmes as a task in testing cycles: building test automation without actually coding large test scripts results in optimizing and increasing test coverage. Many other features of script less-automation up its ante - reviewing test suites on the go and utilizing the knowledge of manual testers seamlessly while lowering costs spent on building scripting skills. Accelerating and parallelizing development and testing processes renders resource intensive manual cycles redundant. A senior tester in a mid-sized Indian service company said: "... automation has condensed a three to six-week testing cycle to a few days! With applications becoming more complex and users now more digitally adept, chances of a slow or buggy application surviving in the market is nil!" The forecast on script-less automation and its impact on manual testing practices has increased the challenges of maintaining manual testing and ensuring manual skill sets do not become redundant or even counterproductive. Critical work processes maintaining QT cycles like, repetitive testing that run for multiple builds such as regression, intensive tests like load testing, unit testing are believed to be automation ready with better ease and perfection in executing results.

Contrary to this automation-ready picture, issues related to 'failed test automation experiments' persists, as the voices from the QT blogosphere we studied are quick to point out, and rest on the following: considerable time to create the test automation suite; difficulty of scripting and maintenance; the need for specialized development resources; and importantly setbacks from participation of a team of product owners and the subject matter

experts unable to work proficiently [40]. Test automation suites need essential and regular maintenance in order to be reusable and scale per demand requiring more investments in terms of time effort and costs. Many of the testers we spoke to alluded to the difference of opinion still persisting in the QT domain considering automated or manual modes of testing to unearth bugs hidden deep. Much of the dominant voices from the QT blogosphere we encountered and studied weren't unilaterally pro-automation, instead positioning and rooting for a more creative, blended and human-driven testing practices. Pradeep, in an interview, proposes a view on the blended QT practice: "...Test automation increases the challenges of manual testing. Development methodologies such as agile simply cannot be executed without automation... Again, to muddy the waters, automation testing being skill and effort-intensive an effective return on investment exercise is needed to assess and implement a mix of automated and manual tests... A 100 % automation drive might be counterproductive- especially in an off-shored labor such as India with high skill set availability at competitive price" This being the case diligent choices are made weighing factors like the complexity of automating test cases, frequency of executing test suites and contrasting them against time and effort involved in running manual tests.

## 4.2 Tools Don't Make Testers, Testers Make Tools

*"...a tester is never able to say "I am sure" - in relation to a bug free product! Can any QT come up with a bug-free product? How does one define or achieve a desired level of bug-freeness?"* – Balam, QT Manager of Mid-Sized Testing- focused Firm

This specific sub-section functions as a prelude to the discussion section foregrounding expressive voices of testers that not only 'show boat' manual testing practices but privilege the human centrism in testing as a work practice. In India, an active and multifaceted blogging community of testers vibrantly discuss a hybrid approach to testing. For these bloggers, a tool can always replace a tester who just executes the test cases but can never replace those planning/documenting and analyzing test plan/design/cases [23]. In India, though, a tester is wrongly conceptualized as the former, a person who is responsible for only reporting the bugs. Training a tester, particularly in the IT industry training centers across India, is often, about providing operational access and basic functional knowledge to the testing tools. A major takeaway from our reading of the QT blogosphere is that testers are viewed and trained as 'tool-smiths' and testing work as a mechanical skill devoid of finer quality [2]. Only a handful of people spend time in learning the frameworks, strategies and business intelligence of quality testing. Testers are ideally people who architect tools and harness these tools for creating test cases. Instead tool-learning is given primary importance over acquiring testing methodologies. Anirban, a senior consultant, said: "...it should be a tester's initiative to not only report issues or bugs but also come up with solutions for those to make the product more user-friendly... leaning testing must be a parallel process of testing skills, platforms, tools and learning the ropes in business..."

People like Anirban strongly support the idea that the role of an experienced tester for a product's release and maintenance can never be undermined. Testers can push the envelope of bug detection coupled with product enhancements - someone who not only can bust hidden yet easily identifiable bugs but employ exploratory techniques to unearth deeper product issues. This means an ace tester is alert and intuitive to spot a buried bug that could hamper the product's usage in its first round of release and user engagement. Much of the spotting and de-bugging process rests on the veracity and potency of the building, running and finishing test cases for a product. Pradeep, in our interview, suggests a far more demanding and diagnostic profile for a tester, "... A standing rule seems to suggest that a good tester should never say 'I am sure' since what you say is what you have observed, which is a conjecture, an inference and cannot be conflated with the 'truth' about the bug status in a product..."

A senior QT and Blogger Shrinik views the tester role with equal gravity. He opines in an idealistic vein that a tester is permanently testing, always an expert-in-progress. The QT universe being never-ending in terms of the range and variety of test cases, the more you test, the more you learn. To Shrinik, testers should ideally always be 'grown- up monkeys', referring to the idea that there is always to gain and learn from testing work: 'grown up monkey' is also a remark made in the context of a testing technique called Monkey Testing which is essentially providing random inputs in order to analyze the resulting behavior of an application or a product for bugs. Performing simple random experiments is part and parcel of monkey testing to get at that persistent bug going unnoticed, in recursive and automated cycles of testing. From our understanding in this study, it seems to be a general assumption that monkey testing is performed by testers who are junior and unskilled. It implies, according to Shrinik, new comers to QT can really benefit and begin to build work excellence from monkey testing methods.

Monkey testing attains a differing connotation in the actual work space. We learnt from our interviews that in the Indian work situation, it is par to assume that every fresh tester has to do monkey testing, thereby implying a lack of appropriate skills and experience when s/he enters the job market – a large pool in the entry level testing job profile lack any kind of specific training or formal education to conduct QT work. It is evident that voices from the blogospheres address contradictions in the potential and the actual experience of being a tester in the Indian IT industry context.

The blogosphere arguably carries a near-unanimous opinion that a tester is wrongly conceptualized as a 'bug reporter'. Instead s/he occupies a specialist location as a key stakeholder in the process of finding solutions to architect a user-friendly product. To the bloggers testers are ideally people who are able to architect tools and have the unique capability of harnessing the tool to create, run and pass test cases.

We will now move to the discussion section with a focus on the first-person voices of quality testers and a counter discourse establishing testing as creative and collaborative human-technical practice.

## 5 DISCUSSION

*"Tools don't test. Only people test. Tools perform actions that help people test"* - James Bach, on a context driven approach to the entire testing work culture [1].

In this section, we foreground a discussion from a qualitative analysis of our interviews with quality testers and a deep study of the blogosphere maintained by several quality testers working in the Indian IT industry. The first sub section will offer testing as a creative context-driven practice; this section will segue into the next with arguments offering a logical culmination of a tester profile to a business analyst; finally, we round off providing a summary of QT voices discussing the mettle and tenacity of the tester job profile

### 5.1 Thinking Beyond the Black Box of Testing

*"Testers aren't slaves of tools but rather should learn the art of testing to make themselves indispensable"* — Tester Tested Blog Post [38].

Indian bloggers build their critique of automation that inches forward day by day to take over manual testing work practices; they plead for a more balanced approach in incorporating a testing work etiquette. Black box testing refers to the form of testing where the tester is supposedly looking at the functionality of the software and not the internal structure i.e. modules/units/codes. Testers are only looking at the front end and UI of a product now under a testing cycle. Automation has steadily entered black box testing methods, eroding into the manual testing practices that was largely prevalent until a few years ago.

Going beyond the 'black box' is our way of referencing arguments the testing community makes to go beyond checking for simple functionalities of a product and for adopting the method of exploratory testing where one would be innovative and thoughtful in coming up with new test cases; test cases evolving out of one's own knowledge and experience leading to effective improvement in the identification of bugs. Several blog posts have pointed at skilled exploratory testers using creative ways to find and recognize bugs than may remain unidentifiable through running routine and repetitive test cases. An exploratory approach to product testing bring in multi-faceted kinds of information about product functionality, make for a good building of test runs and open up risk awareness to the type of product crash or non-usability. This kind of testing, that we call 'thinking beyond the black box of testing' not only follows the test plan, but allows for creative modifications to the plan at short notice, also affording testers to come up with appropriate solutions – sometimes pushing the boundary of testing by way of a futuristic vision to create bug-free products. As one of our interviewees, Sukumaran, a senior product developer and manager in an Indian IT software services giant turned entrepreneur said, "... in a way all testing is manual testing but is tool-supported. Tools are capable detecting only very specific bugs. Humans can, in principle, detect any kind of bug ... of course with the help of tools... Tools have also detected things that are not bugs... while missing out on an important and critical bug ... take the example of exploratory Testing – which is nothing but the tester's creative juices flowing.... You think tools take into account

the environment and customization of the product under quality testing...!”

Our primary suggestion in this subsection is the need for the development and testing communities to emphasize non-routine testing methods that are often kept aside in favor of more structured documentation. Automation testing makes quick work of all test cases that follow routine, predictable paths. However, this kind of testing cannot generate new test cases or understand why an error may have occurred - both these still require manual input. A pressing requirement in the IT industry is then to train testers ensuring their job profiles allow to gain a certain level of critical knowledge of QT cycles instead of becoming machine-operating tool smiths.

## 5.2 Expanding Role of Testing: From Tester to Business Analyst

*"Tomorrow or Today, if you are a CEO ensure that one of the senior you hire for your cabinet was a good tester and continues to be so even after becoming a senior" - Pradeep*

Quality testing in the past decade has evolved into a client focused and business strategizing task, particularly at senior levels of testing. Senior testers are often called upon to interface with business clients who are the primary stakeholders in the products that are being tested. This involves a good amount of immersion in the business contexts of clients and developing a sense of requirements not only to evolve testing parameters but customizing these parameters for the broader business context in which the product's life cycle begins. Testers over time develop a keen perception of business dynamics and their relationship to product testing. We unpack the above statement in the following portions of this section.

Quality testers in the course of everyday routines of testing a product go beyond their call of duty and begin to care about their client's company profile, market share, brand value, customers and not in the least the product's relationship to all of these. It seems testers slowly and surely begin to focus on goals their client business wants to achieve through their testing work: these might range from improving sales; focusing on conversion, funnel, lesser cost per customer acquisition, increase profitability, make customers happy by just allowing them to do what they intend to with the software. Parth, QT Manager in a Testing focused SME, said: "... Testing soon transforms into a work activity not just obsessed with the functionality and stability of the software under test. Testers begin to focus on human skills, develop skills to be clued into broad kinds of information, beginning from developing acumen to build test cases to business strategizing of the product under test." This already present work trend points to the kind of skills that all testers will soon need to develop in the face of automation - skills centered around context analysis and customer-facing interaction.

Yet, the initial training that is offered to testers in their work environments, or the way tester training courses are structured in informal learning markets, does not pay attention to these skills. As elaborated on earlier, testers are taught documentation and tool navigation. Key job skills such as requirement analysis and client

communication which are needed for career progress are not integrated into training modules. Moreover, testers, developers and analysts are segregated as soon as they are hired and training proceeds thereafter - this effectively limits the communication and cross-skilling occurring across these groups. It also has the unintended effect of making testing feel like less desired work. Pradeep recounted his experience of training a batch of 300+ new hires at a large IT firm in India, where he saw that 'the kids just didn't want to do testing, because they were told it had the lowest cut off. These were all bright, top-of-their-class kids. It felt degrading to them to be given that kind of work.'

Our findings point to a need in rethinking the training and support offered to early career testers if they are to succeed in the hybridized automation-manual testing scheme. There are certain skills, such as basic coding, which are now needed across the board. Similarly, the ability to read client requirements and analyze their implication in terms of product build is a quality required by testers, developers and analysts alike. Injecting value into the QT cycle must take cognizance of the expanding role of testers and provide for good utilization of their experience gained in the process of interfacing with clients, learning about a business culture while developing a sense for product functionalities

## 5.3 Injecting Value into Testing

*"Earlier testers used to take one hour to run the test cases. Now, automation can do it in 5 minutes! ... But all that extra 55 minutes, it doesn't go anywhere, it isn't used." — Pradeep, Senior Tester and Entrepreneur in an Interview*

*"Tools that are not tested well are time bombs! – a testing tool needs more testing than a product which it is aiding to be tested! Behind every life saved in an accident there is a tester and for every life killed in an accident there is a test case missing..." [42].*

The biggest value from automation is seen as time saved and time gained - automation tools can run more test cases, run test cases faster and at more precise time intervals. In all our interviews, the informants cited time-saving (and hence cost-saving for the client in terms of hours billed) as the biggest reason for automation: particularly for products with rapid release cycles, this is a significant value addition in the form of enhanced efficiency and productivity in the testing methods due to automation. By automating a manual process, free time is granted for testers to explore the product further. Instead we find that shop floors in the IT industry, some in our research sample, often don't have any set protocol for utilizing this saved time. Rahul, a mid-career tester, explained to us that 'if we need to submit a project one week later, but we're done will all test cases now then it's up to us. Some teams just submit, some wait and see if they can run more test cases. There's no set way'.

But there is also a positive flip side for testers in India. According to Vishnu, the automation director in a testing firm, "... we have elaborate work hierarchies and a transparent upward career path for our testers--- I started a newbie 11 years ago! ...many of us have learnt along the way to experiment with testing methods, particularly client focused ways to bring more value to our business – but we are primarily testers here - not a large service oriented



software giant ...”. Clearly there are some testing-oriented firms that have already taken into account the need to restructure in light of new testing practices, but this has not become standard across the industry.

Current testing practices, in the light of the benefits automation bring in, require a re-assessment in re-purposing gained time and efficiency. The possibility of testers critically assessing automation tools handed to them and proposing additions and changes in functionalities which can further aid QT still remains a hope in the context of the Indian IT industry

## 6 CONCLUSION

Quality testing has been acknowledged as a significant step in the product development process and a rapidly growing job segment in India. While the onset of automation had triggered a discourse of job-loss and the slow erosion of manual practices our paper demonstrates that the Indian IT industry is not a uniform terrain being transformed by automation. Instead, industry leaders have pointed to the more complex business decisions needed to usher in an automated work environment. However, voices from the blogosphere and personal interviews with the tester community hinge around the ‘idea’ of a hybrid environment for quality testing blending manual and automation practices. The creation of this ideal environment is supported by not only technological developments or the introduction of new tools, but a conducive business environment, the tailoring of tools to meet client deadlines and budgets, as well as a robust QT job profile committed to the optimal use of current tools and dynamically participating in the process of shop floor QT. Above all, the voices of testers challenge the idea of testing as purely a ‘tool smith’ or mechanical work in re-imagining the job of QT as critical in rendering a well-tested and bug-free product. It is imperative to use these voices, of people deeply familiar with the inner workings of the segment, as directions for change in the industry.

## ACKNOWLEDGMENTS

This work was partially supported by the xxxxx and by the xxxxx)[Anonymized for submission purposes]

## REFERENCES

- [1] <http://www.satisfice.com/articles/cdt-automation.pdf>
- [2] *Are you a Tester or a Toolsmith?*, 2006. <http://testertested.blogspot.in/2006/02/are-you-tester-or-toolsmith.html>
- [3] R. M. Belbin. 2003. *Management teams: why they succeed or fail*, 2nd Ed, Butterworth Heinemann, Oxford.
- [4] Thirumalesh Bhat and Nachiappan Nagappan. 2006. Evaluating the efficacy of test-driven development: industrial case studies. In *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering*. AM, New York, NY, 356-363.
- [5] Wiebe Bijker, Thomas Hughes, and Trevor Pinch, eds. 1987. *The Social Construction of Technological Systems: New Directions in the Sociology of History and Technology*. Cambridge, MA: MIT Press.
- [6] Judy Bullock (Ed.). 2004. *Management Teams: Why They Succeed or Fail (2nd. ed.)*. Leadership & Organization Development Journal, Vol. 25, Issue: 5, 477-479. DOI: <https://doi.org/10.1108/01437730410544791>
- [7] Bill Curtis, Herb Krasner and Neil Iscoe. 1988. A Field Study of the Software Design Process for Large System. *Comms of the ACM* 31(11). ACM, New York, NY, 1268-1287.
- [8] Reham Ejaz, Mubina Nazmeen and Maryam Zafar. 2010. A quality assurance model for analysis phase. In *Proceedings of the 2010 National Software Engineering Conference*. ACM, New York, NY, Article No. 1.
- [9] Stuart Feldman. 2005. Quality Assurance: Much More Than Testing. In *Magazine Queue - Quality Assurance*. ACM, New York, NY, 26-29.
- [1] Henning Femmer, Jan Kučera and Antonio Vetrò. 2014. On the impact of passive voice requirements on domain modelling. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, New York, NY, Article No. 21.
- [1] *Future that works: automation, employment, and productivity, 2017*. New York, NY: McKinsey Global Institute. Retrieved from [http://njit2.mrooms.net/pluginfile.php/688844/mod\\_resource/content/1/ExecutiveSummaryofMcKinseyReportonAutomation.pdf](http://njit2.mrooms.net/pluginfile.php/688844/mod_resource/content/1/ExecutiveSummaryofMcKinseyReportonAutomation.pdf)
- [1] *Gartner Says Global IT Spending to Reach \$3.5 Trillion in 2017*, 2016. <https://www.gartner.com/newsroom/id/3482917>
- [1] Milos Gligoric, Stas Negara, Owolabi Legunsen and Darko Marinov. 2014. An empirical evaluation and comparison of manual and automated test selection. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*. ACM, New York, NY, 361-372.
- [1] Benedikt Hauptmann, Maximilian Junker, Sebastian Eder, Christian Amann and Rudolf Vaas. 2014. An expert-based cost estimation model for system test execution. In *Proceedings of the 2014 International Conference on Software and System Process*. ACM, New York, NY, 159-163.
- [1] Johannes Held and Richard Lenz. 2012. Towards Measuring Test Data Quality. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*. ACM, New York, NY, 233-238.
- [1] Kim Herzig and Nachiappan Nagappan. 2014. The Impact of Test Ownership and Team Structures on the Reliability and Effectiveness of Quality Test Runs. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, New York, NY, Article No. 2 .
- [1] *India's cheap labour a 'myth', says Assocham President*, 2017. <http://www.livemint.com/Industry/4h5fHTN1tUJbgD1LYuctO/Indias-cheap-labour-a-myth-says-Assocham-President.html>
- [1] Infographics on IT Industry &ITeS in India, 2017. <https://www.ibef.org/industry/information-technology-india/infographic>
- [1] Sheila Jasanoff. “Future Imperfect: Science, Technology and the Imagination of Modernity” in *Dreamscapes of Modernity* (ed. Jasanoff S., S. Kim). Chicago: The University of Chicago Press. 2015: 1-33.
- [2] Sheila Jasanoff. “Imagined and Invented Worlds” in *Dreamscapes of Modernity* (ed. Jasanoff, Kim). Chicago: The University of Chicago Press. 2015: 321-42.
- [2] Bruno Latour . 1999. *Pandora's Hope: Essays on the Reality of Science Studies*. Massachusetts: Harvard University Press.

- [2] D. Jeya Mala and V. Mohan. 2010. Quality improvement and optimization of test cases: a hybrid genetic algorithm based approach. In *Newsletter of ACM SIGSOFT Software Engineering Notes*. Newsletter Vol. 35, Issue: 3, 1-14.
- [2] *Manual Testing - no more a shit job !*.  
[3] <http://testertested.blogspot.in/2006/05/manual-testing-no-more-shit-job.html>
- [2] J. A. McDermid and K. H. Bennett. 1999. Software Engineering Research: A Critical Appraisal. In *IEEE Proceedings On Software Engineering* 146(4). ACM, New York, NY, 179-186.
- [2] Cu D. Nguyen, Alessandro Marchetto and Paolo Tonella. 2013. Automated oracles: an empirical study on cost and effectiveness. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. ACM, New York, NY, 136-146.
- [2] Gary M. Olson, Judith S. Olson, Mark R. Carter and Marianne Storösten. 1992. Small group design meetings: an analysis of collaboration. *Human-Computer Interaction Journal*, Vol. 7, Issue: 4, 347-374. DOI: [http://doi.org/10.1207/s15327051hci0704\\_1](http://doi.org/10.1207/s15327051hci0704_1)
- [2] Fabio Palomba, Dario Di Nucci, Annibale Panichella, Rocco Oliveto and Andrea De Lucia. 2016. On the diffusion of test smells in automatically generated test code: an empirical study. In *Proceedings of the 9th International Workshop on Search-Based Software Testing*. ACM, New York, NY, 5-14.
- [2] Fabio Palomba, Annibale Panichella, Andy Zaidman, Rocco Oliveto and Andrea De Lucia. 2016. Automatic test case generation: what if test code quality matters?. In *Proceedings of the 25th International Symposium on Software Testing and Analysis*. ACM, New York, NY, 130-141.
- [2] *Pandavas and their testing skills ! (of course my own)*, 2006.  
[9] <http://testertested.blogspot.in/2006/01/pandavas-and-their-testing-skills-of.html>
- [3] Pablo Pedemonte, Jalal Mahmud and Tessa Lau. 2012. Towards automatic functional test execution. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, New York, NY, 227-236.
- [3] Václav Rajlich, Norman Wilde, Michelle Buckellew and Henry Page. 2001. Software Cultures and Evolution. *Computer Journal*, Vol. 34, Issue: 9, 24-28. DOI: <http://doi.org/10.1109/2.947084>
- [3] Rudolf Ramler and Klaus Wolfmaier. 2006. Economic Perspectives in Test Automation: Balancing Automated and Manual Testing with Opportunity Cost. In *Proceedings of the 2006 international workshop on Automation of software test*. ACM, New York, NY, 85-91.
- [3] Zahra Sahaf, VahidGarousi, Dietmar Pfahl, Rob Irving and Yasaman Amannejad. 2014. When to automate software testing? decision support based on system dynamics: an industrial case study. In *Proceedings of the 2014 International Conference on Software and System Process*. ACM, New York, NY, 149-158.
- [3] Yavuz Sancar, Frank Brüseke, Stefan Sauer, Gregor Engels and Hendrik Voigt. 2010. Towards economical software release recommendations. In *Proceedings of the 1st Workshop on Testing Object-Oriented Systems*. ACM, New York, NY, Article No. 9.
- [3] Helen Sharp and Hugh Robinson. 2005. Some Social Factors of Software Engineering: The Maverick, Community and Technical Practices. In *Proceedings of the 2005 workshop on Human and social factors of software engineering*. ACM, New York, NY, 1-6.
- [3] Software Sector Analysis Report, 2016. Equity Master. Retrieved from <https://www.equitymaster.com/research-it/sector-info/software/Software-Sector-Analysis-Report.asp>
- [3] Test Automation Market – Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2016 - 2024, 2016. New York, NY: Cision PR Newswire. Retrieved from <https://www.prnewswire.com/news-releases/test-automation-market---global-industry-analysis-size-share-growth-trends-and-forecast-2016---2024-300262224.html>
- [3] Tester Tested!, Pradeep Soundararajan. <http://testertested.blogspot.in/>  
[8]
- [3] Testing trends in 2017: a survey of software professionals, 2017. Dimensional Research: Sauce Labs. Retrieved from <http://info.saucelabs.com/rs/468-XBT-687/images/Sauce%20Labs%20-%20State%20of%20Testing%20Survey%20Results%20Jan,%202017.pdf>
- [4] The biggest benefits of scriptless test automation, 2017.  
[0] <http://www.qualitytesting.info/profiles/blogs/the-biggest-benefits-of-scriptless-test-automation>
- [4] Thinking Tester, Shrini Kulkarni. <http://shrini.blogspot.in/>  
[1]
- [4] Tools that slow you at work!, 2006.  
[2] <http://testertested.blogspot.in/2006/02/tools-that-slow-you-at-work.html>
- [4] Trends in software testing, 2016. Business Excellence: KPMG. Retrieved from <https://assets.kpmg.com/content/dam/kpmg/pdf/2016/07/Trends-in-software-testing-19july16.pdf>  
[3]
- [4] Auri M. R. Vincenzi, Tiago Bachiega, Daniel G. de Oliveira, Simone R. S. de Souza and José C. Maldonado. 2016. The complementary aspect of automatically and manually generated test case sets. In *Proceedings of the 7th International Workshop on Automating Test Case Design, Selection, and Evaluation*. ACM, New York, NY, 23-30.
- [4] We need Industrial Revolution 4.0, 2016.  
[5] <http://www.thehindubusinessline.com/opinion/automation-will-hit-indias-labour-advantage/article9262553.ece>
- [4] Why automation could be a threat to India's Growth, 2017.  
[6] <http://www.bbc.com/future/story/20170510-why-automation-could-be-a-threat-to-indias-growth>
- [4] Yahoo Messenger! Missed Feature?, 2006.  
[7] [http://testertested.blogspot.in/2006/02/yahoo-messenger-missed-feature\\_25.html](http://testertested.blogspot.in/2006/02/yahoo-messenger-missed-feature_25.html)