

Editorial

We see so much of strife, unrest, and violence around us in the world. Today, we have more comforts than we could have ever imagined or dreamed of. An average citizen possesses more than what people could afford a 100 years ago. But are we in any case happy? People are distressed.

We have seen that as society becomes more and more affluent, people become more and more distressed, desperate and hopeless. With increase in comforts outside, we become more uncomfortable inside.

Why is there so much crime and aggression?

Suppose right now you're pursuing your happiness very vigorously and I stand in your way. You will want to kill me, or somehow get me out of the way. It's a natural instinct. See, this is exactly what criminals are doing. They are only pursuing their happiness very vigorously, so vigorously that they are not concerned with what is happening to people around them.

Forces of love/compassion and hate/anger are always functioning in the world. It is a seesaw game. The question is: which end of the seesaw you want loaded?

To maintain sanity in the world and in our lives, we need to transform our anger, hate, jealousy into love and compassion—all forms of the same Energy.

Editorial Team

What's Inside

Editorial	1
Upcoming Conferences	2
Article	3
From ITB Desk	5
GTB	6
Interview	8

Resources

Black-Box Testing: Techniques for Functional Testing of Software and Systems

by Boris Beizer

Paperback: 320 pages

Publisher: Wiley (May 1995)

Language: English

ISBN-10: 0471120944

ISBN-13: 978-0471120940

Lessons Learned in Software Testing

by Cem Kaner, James Bach, and Bret Pettichord
(Paperback - Dec 15, 2001)

Paperback: 352 pages

Publisher: Wiley; 1st edition (December 15, 2001)

Language: English

ISBN-10: 0471081124

ISBN-13: 978-0471081128

Software Inspection (Paperback)

by Tom Gilb (Author), D. Graham (Author)

Paperback: 496 pages

Publisher: Addison-Wesley Professional (January 10, 1994)

Language: English

ISBN-10: 0201631814

ISBN-13: 978-0201631814

Guide to Advanced Software Testing

by Anne Mette Jonassen Hass

Hardcover: 427 pages

Publisher: Artech House Publishers (April 15, 2008)

Language: English

ISBN-10: 1596932856

ISBN-13: 978-1596932852

Upcoming Conferences

- **ANZTB Test 2010 Conference**

The ANZTB Test 2010 Conference will take place in Melbourne from Monday 1 - Wednesday 3 March 2010. The 2-day conference will feature keynote speeches and presentations from local specialists, as well as a Welcome Cocktail Party and associated Expo. The conference is expected to attract over 150 delegates and provides a wonderful opportunity to meet and extend contacts.

Above all else, we encourage you to enjoy the warm hospitality that Melbourne and its people have to offer, and that you too will share in the knowledge and ideas that the ANZTB Test 2010 Conference will generate.

- **STAREAST 2010**

April 26–30, 2010
Orlando, FL

<http://www.sqe.com/StarEast/Splash.aspx>

- **QUEST- Quality Engineered Software and Testing Conference Dallas 2010**

April 19-23, 2010
Dallas, Texas

<http://www.qaiquest.org/dallas/>

Random Collection: Blogs on Testing

Testlabs

<http://blog.testlabs.com/>

Testing at the Edge of Chaos

Matt Heusser's Blog

<http://blogs.stpcollaborative.com/matt/>

Notes and Rants

<http://blogs.msdn.com/alanpa/default.aspx>

Test Obsessed

<http://testobsessed.com/>

DevelopSense Blog

<http://www.developsense.com/blog.html>

Practical QA

<http://www.practicalqa.com/>

Abakas

<http://blog.abakas.com/>

Steve Rowe's Blog

<http://blogs.msdn.com/steverowe/default.aspx>

The Sounds of Software JakeBraking

<http://www.sqablogs.com/JakeBrake/>

PractiTest QA Blog

<http://qablog.practitest.com/>

Agile and Performance testing? “A Contradiction of terms?”

by Mieke Gevers,
AQIS, Belgium



Mieke has been in the IT industry for 20 years and since 2006, founder and managing director at AQIS (Agile Quality in Information Systems).

She has a Bachelors of Computer Science and started her career as a software developer, later moving on to various positions as an analyst, project manager and, ultimately, QA Manager for several different companies in the Benelux region and Germany. Until

November, 2006 she has been with Segue Software, Borland for more than eight years rising from Technical Sales Engineer, Technical Manager, EMEA/APAC, Alliance Architect EMEA/APAC to Solution Architect, EMEA at Borland.

Having developed special interest in the techniques and processes relating to test environment management and the impact of environmental factors on automated testing, she is a regular speaker throughout Europe, the United States, Asia and Australia. She is also EuroSTAR's country coordinator for Belgium, and member of the Eurostar 2007 Program Committee.

She also has been working with different automatic testing tools, specialising on Performance testing and monitoring.

Where does performance testing fit in an agile world? What are its challenges and advantages? Does it make sense to mix agile and performance testing?

Performance testing is an empirical and technical investigation, conducted to provide stakeholders with information about the quality of the product or service under test, with regard to speed, timings and resource measurements of all kinds (such as end-to-end response times, network delays, number of denial of service, infrastructure measurements, performance counters, amount of open connections, memory usage, ...)

So, let us first take a look at the definitions of what “Agile” means by looking at the “Manifesto for Agile Software Development”, where the definition and goals are:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools Working software over comprehensive documentation Customer collaboration over contract negotiation Responding to change over following a plan

Agile or iterative development environments have evolved to be a formal development methodology, where we are working towards releasing software of small functionalities, which all are making it into the big picture, namely the final release of the full functionalities.

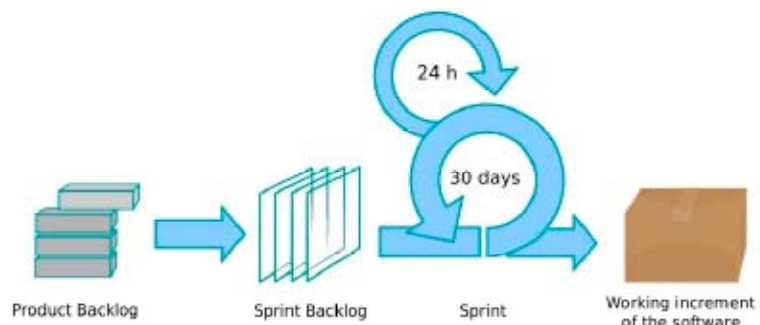
Agile also embraces the concept of using cross-functional teams; groups of testers, designers and developers, working together before and during the coding process. The cross-functional team should also include someone who is skilled in performance testing. We can't wait until development is finished, but need to test performance while development is ongoing, with a tight collaboration between developers and testers.

These days, Agile can address some of the issues that have caused problems in the waterfall model, such as:

- Developers needing to interrupt code writing to review tests and write patches.
- Dealing with large cycle time from developing and from test.
- Anticipating problems with long resolution times with the danger of inflated estimates or worse, underestimates.

Basically, we are working in groups and facing small time frames (called sprints) where functional testing tends to become more a white box than grey or black box testing approach.

Fulfillment of functional and performance requirements are essential to software quality. Both



are equal in their relevance to deliver a quality product, meeting the “good enough” level. Today, mission critical applications rely on adequate performance as much as on functional richness. Poorly performing software can potentially cause massive losses in revenue and put your business at risk. Ensuring software performance is a very challenging task, even more challenging than providing the correct functionality, because it depends on many variables, such as:

Bad code can cause an algorithmic performance degradation.

Article...

Article

4

Perf. Testing

- User behavior and load volume directly impact response times and resource usage.
- Improper configuration of application servers or frameworks may degrade performance significantly.
- Inadequate distribution of software components can introduce bottlenecks.

Many of these risk factors are specific to the underlying environment and can only be diagnosed in that specific setting, under specific circumstances. Performance engineers and developers find themselves increasingly confronted with performance problems that are difficult or impossible to diagnose. The increasing usage of third party code and highly distributed software environments adds additional complexity.

Performance, like the responsiveness of an application, is a part of the total end-user experience. Namely, the application should function and respond as requested and expected, but it should also perform and execute within the defined goals or SLAs (Service Level Agreements).

Who does not agree to the fact that finding errors in a later stage in the development phase comes with a cost increase? Fixing an issue in production is more costly than fixing it in an early stage of development.

When should performance testing start?

During a project, a customer showed us that even when using the traditional “V-Model”, they still gained advice from a performance tester from the very beginning of the project. Is this an extravagant luxury? Certainly not, and here is why. A performance tester has a different skill set and is looking at the application with another pair of glasses than the Product manager, developer, functional testers,...In general, when a performance tester is present at project inception (project start), it means there is already a specific, significant risk known for the tester to address.

Using these set of “other glasses” can lead to valuable advice on the application architecture layer, the infrastructure & HW choices, security, and other items that might have an impact on the application and its performance.

Now the combination Performance testing in an Agile development.

So what about doing performance testing in an agile project? Do we have to wait until we have developed the whole system or a reasonable sized set of deliverables before we start performance testing? When should performance testing start in an agile project? No doubt about it, my answer on this is “always immediately”. Even during development, programmers can easily put timers into their code

which is writing the output to simple log files, in addition to debugging tools. This approach will not only give a first insight into the application's performance but also gives extremely valuable information when you are comparing the results from several software versions with one another. In other words, it can be very helpful to do a performance non-regression & regression test to verify that there is no performance degradation noticeable in the new version or an increase of the application responsiveness.

Taking into account that Agile (Scrum) is working with short iterations, this gives an ideal base to include testing already at the very first iteration. It is common knowledge that this applies for functional testing (white and black box testing), but it is my opinion that Performance testing should not be any different.

At the first iteration, results like “response times”, “throughput”, “how many requests (or user actions) have run” and “performance counter information on the infrastructure like the CPU memory utilization” are giving an indication of the performance of a function, a method, a service,...the application. Some issues are only emerging under a certain “load”, for which you can use tooling from a simple java or JavaScript to a full blown performance testing tool, available either on the market or in the open source space.

After this first iteration or if you want to call it “test iteration”, with Agile, there is a daily feedback round between developers and testers where the developers can look at the performance testing results, log information and ask for more or other details, timers or other program or infrastructure counters that are interesting in order to pinpoint the issues. These daily meetings are very helpful in streamlining and adjusting the performance test plan, the test design and test execution. It is needless to say that for developers as well as testers, using .NET or Java, a profiling tool on its own or in combination with performance testing tools (profiling under load), can be very helpful to pinpoint the cause of the problem. Gaining visibility into for example; memory issues, synchronization problems, tread dumps, doffing capabilities (performance regressions),.....

In conclusion performance testing is an overall process and should be included at the earliest stage of the project, no matter which development methodology is being used.

New Certification

ITB introduces QAMP certification in India. ITB is the exclusive partner in India (National Authority) for QAMP.

QAMP, a new certificate for advanced vocational training of software engineers, supported by iSQI has successfully been introduced in co-operation with and according to the needs of the IT-Industry.

QAMP is targeted towards quality assurance managers, because they require competence in the entire development process. It tests the practical knowledge and the project experience of the certified employee. Theoretical and practical knowledge both have to be updated annually. In addition to a broad theoretical knowledge, there's a need to successfully pass three certification exams:

- Software Test (FL)
- Specified Module (AL)
- Requirement Engineering

Four Steps To QAMP

- Software Testing. (ISTQB Foundation Level)
- Specified Module .ISTQB Advance Level Test Manager/ Configuration Manager/Project Manager/Secure Software manager/Software Architecture
- Requirement Engineering (IREB)
- 2 years of practical experience

Upcoming IREB Examination is on December 20, 2009

For further information please visit our site www.istqb.in/ireb or mail us at ireb@indiantestingboard.com

New Affiliates

Harrier Information Systems Pvt. Ltd.

Website: www.harriersys.com

HARRIER INFORMATION SYSTEMS PVT. LTD.
THINKING LONG TERM ISO 9001:2000 Company

Coremind Technologies

www.coremindtechnologies.com

Core Mind Technologies

Upcoming ISTQB Examinations

- December 13, 2009: Chandigarh
- December 18, 2009: Trinulveli
- December 20, 2009: Pune, Hyderabad, Chennai, Trivandrum, Bangalore, Mumbai, Kolkata, Noida
- February 7, 2010: Pune, Hyderabad, Chennai, Trivandrum, Bangalore, Mumbai, Kolkata, Noida
- March 21, 2010: Pune, Hyderabad, Chennai, Trivandrum, Bangalore, Mumbai, Kolkata, Noida
- May 2, 2010: Pune, Hyderabad, Chennai, Trivandrum, Bangalore, Mumbai, Kolkata, Noida

For Online registrations, please visit http://www.istqb.in/enrollment_form.php

Last date for payment & registration is 10 days before the exam date

German Testing Board (GTB)

GTB Fact sheet	
Location	Germany / Erlangen, in the heart of the “Metropolregion Nürnberg” (http://www.em-n.eu/index.php?id=34&L=1)
Founded & operational since	GTB is a founder member of ISTQB; was created at the “Inaugural Meeting held at Solluntuna, Sweden” on 23rd November 2001
Board officials	Tilo Linz (President), Horst Pohlman (Vice President)
Certifications awarded	13.858 (until end of June 2009)
Language(s) for exams	GTB is responsible for German Language Exams in Germany. The exams are operated on behalf of GTB by 2 professional exam providers (DLGI: www.dlgi.de and iSQI: www.isqi.org).
Exam fee	GTB licenses the questionnaires to the exam providers and the fees are defined by the exam providers.
Software Testing market	<p>The great majority of German companies do not question the importance of software testing as an essential part of a successful and efficient business. For many, however, this appreciation of the worth of testing has so far not resulted in the use of managed and transparent test procedures. This is why analysts draw the conclusion that investment in systematic software testing continues to make a difference and consequently can create competitive advantages.</p> <p>PAC (Pierre Audoin Consultants) analyst Kerstin Dirtheuer is convinced: “As IT production processes become increasingly industrialized, there are more and more specialized quality assurance staff. As well as that, testing boosts the cost-effectiveness of software production, because the cost of retrospective correction of software flaws is significantly higher than expenditure on soundly based test work performed early on in the sequence.” PAC found considerable agreement on the question as to whether testing should be independent of development. At present this independent role is mostly taken on by in-house teams. On the other hand, many companies have experience with external test service providers; some have also gone down the route of off-shoring. There is also considerable market potential opening up for providers of specialized test tools.</p>
Population	About 82 Million
Capital	Berlin
Some details	With 82 million inhabitants, Germany accounts for the largest population among the member states of the European Union and is home to the third-largest number of international migrants worldwide. With the German reunification in 1990 the city of Berlin regained its position as the capital and largest city. On 3 October 1990 the five re-established states of the German Democratic Republic (East Germany) joined the Federal Republic of Germany (West Germany). The united Germany is a member of the United Nations, NATO, G8 and the OECD. It is a major economic power with the world’s fourth largest economy by nominal GDP and the fifth largest in purchasing power parity. It is the largest exporter and second largest importer of goods.

Message from the GTB President

Software testing has become an essential success factor not only for IT managers but for business as such and the competitiveness of companies. By offering and enhancing the worldwide standardized and established ISTQB® qualification scheme for software testers, the German Testing Board is helping to increase the value proposition of IT for business success.

Tilo Linz

President, German Testing Board (FSTB)

Flag	
The symbol for reunified Germany	 <p>The Brandenburg gate in Berlin</p>
Other info	<p>This link provides a wealth of interesting data: http://www.em-n.eu/index.php?id=34&L=1</p>
Contact	<p>German Testing Board e.V. Backoffice c/o Andrea Kränzlein Office-Management Koldestrasse 8 b D- 91052 Erlangen Tel: +49 / (0) 91 31 / 97 61 06 Fax: +49 / (0) 91 31 / 97 61 08 Mobil: +49 / (0) 171 / 63 90 749 E-Mail: backoffice@german-testing-board.info</p>



Hans Schaefer

President of the Norwegian Testing Board

Hans.schaefer@ieee.org

Hans Schaefer is an independent software testing consultant in Norway and the president of the Norwegian Testing Board. He has been consulting and teaching software testing methods since 1984. He organizes the Norwegian Special Interest Group in Software Testing for Western Norway. His work emphasis is on consulting; teaching and coaching test process improvement and test design techniques, as well as reviews.

ITB -- Hans, please tell us something about yourself, your background, your interests and things that you like to do in your free time.

Hans -- I am originally German, went to Norway in 1981 because my new country has more wild nature as well as more space and less bureaucracy. I studied software development and railway technology. Due to quality problems in the products we developed, I started working with systematic testing, and organizing the development work in order to better results. The latter I learned later is nowadays called "quality management". As the one-eyed amongst the blind, I was considered "expert" in testing, etc. and thus remained in this area. And this area IS interesting.

In my free time I walk in the mountains, either on foot or on ski (during winter). Sometimes, during weekends, I am a mountain guide. I am also volunteering at a museum railway, being fireman on our steam locomotive, and on other steam engines on the National network in Norway. This is a safety-critical job, and the seriousness of doing it gives an interesting contrast to the way most organizations work with software.

ITB -- Could you tell us something about your book *Software Testing Foundations*?

Hans -- Actually, there was an older book written in German by Tilo Linz and Andreas Spillner, and I was interested in translating it, because I needed a good introductory book in software testing. It ended up with the three of us translating it, fixing issues, and changing it to be consistent with the newest version of the ISTQB Foundation syllabus. Actually, the most interesting part of the work on the book was making sure it conforms to the syllabus and the terminology. We had many review cycles and many findings and discussions. The work really showed that when you use time to work through things, you find problems. Review is a very important technology!

I am using the book in my ISTQB Foundation courses and exam preparation workshops.

ITB -- What are the topics you address in your seminars? Please tell us something about a few of these topics?

Hans -- This is all dependent on the market. I have been teaching for a long time, both organizing and managing testing, test design techniques and review techniques. Nowadays the market is very interested in ISTQB compliant courses. This is why I do this part of the time.

However, I like teaching a few special topics:

- Risk based testing and risk management (on a broader scale than the ISTQB curricula). Risk management is life critical in the winter mountains, and one of the most effective methods (useful for both mountain use and software projects) was developed by mountain guide Werner Munter from Switzerland.
- Special methods for testing real time and embedded software (things that are mostly ignored in text books, but ever more important). These are things like varying response times, parallel processing, race conditions, deadlocks, priority inversion, etc.
- Learning from problems, i.e., how to analyze and classify problems occurring and use such classification to improve development and testing (root cause analysis and retrospectives).
- Reviews. There is a lot more than the minimum in the ISTQB curriculum. Leading reviews and leading meetings has very much in common. However, most organizations think that running reviews is easy. Thus I am more leading reviews as a neutral facilitator than teaching it.

ITB -- If you were asked to name five persons who have contributed the most to testing, who would they be?

Hans -- This is a very difficult task. There are definitely more than five persons. Personally I learned most from these people:

🔧 Tom Gilb about how to define requirements, especially “soft” requirements. His “Planguage” method is incredibly effective in helping to get testable requirements. He also gave a lot of input to improving reviews and inspections.

🔧 Boris Beizer, from him I got my first introduction to test design methods. His “Black Box Testing” book from 1995 essentially told a general approach to functional testing you could use whenever you have diagrams as a test basis.

🔧 John Musa about software reliability engineering. Even if his work requires developing usage profiles, statistical testing and simulation, the approach with reliability growth models seems to be a possible solution. I think this work is getting more and more important, the more complicated systems are.

🔧 Cem Kaner and Hung Nguyen. It started out with their book “Testing Computer Software”, but they have written a lot more. The book was the first one explicitly telling how to test under time and budget pressure. I had to learn this the hard way, since 1987: Any idea of trying to test “everything” will not work.

🔧 I have problems mentioning one person as number five, because there are plainly too many. Some of them are active in the ISTQB; some are in completely other fields. Maybe it is my wife Åshild Husum who is a medical doctor and who taught me that improvements, if they come at all, in most cases, come slowly. She also taught me that symptoms may be far removed from the cause, like in software testing.

ITB -- What, in your experience, are the areas of improvement for most testers and how can they improve?

Hans -- Do the testing job seriously. Find out what really matters. Find out how much value your job gives to your sponsors, in detecting trouble that otherwise might cost a lot more. And after that be open to improve whatever helps you to catch the worst problems you have in your organization.

As a tester you need to be very flexible! Sometimes you need to improve testing; sometimes you need to fight for improving requirements management, configuration management, risk management, reviews, meetings, project estimating and planning, or whatever else.

ITB -- Which are some of the testing books you would recommend to people?

Hans -- It totally depends on what you need to learn. My top choice about test design is “Black Box Testing” by Boris Beizer. My top choice about reviews is the Handbook on

Walkthroughs, Inspections and Technical Reviews by Weinberg and Freedman. For test management I recommend “Testing Computer Software” by Kaner, Falk, and Nguyen. “Software Test Automation” by Dorothy Graham and Mark Fewster is probably the best introduction into the technical problems in this area. My own book is good if you plan to appear for the ISTQB Foundation exam. But for any other topic there are many books.

ITB -- How can a tester do risk based testing when the organization is not following risk based testing? How can one introduce risk based testing to his/her organization?

Hans -- Any testing effort has the same problem: Test execution will be at the end of the project, and all delays accumulate there, creating the need to do test execution cheaper and faster. This means risk based testing is necessary anyway. If the term is not accepted, you may call it “surviving testing under time and budget pressure”. At any time, a tester should try to sort requirements (the test basis) into what definitely needs to be there, what should be there and what may be left out if there are problems.

A test plan should contain a section on “risks and contingencies”, and contingency strategies should be thought about before you embark on a testing project. Some project risks, like delays, sickness, changes, scope creep, nearly always materialize.

ITB -- What in your opinion is the state of software testing today and where do you see it going?

Hans -- It totally depends on the kind of product and the company. In the safety-critical domain, testing may be very advanced. Model based testing, combination testing, advanced test automation and very thorough regression testing as well as root-cause analysis are in daily use. I think more of this will be used in more parts of the industry.

On the other hand many organizations have nearly nothing in place. I think there is one principle that will need to be much more used: The principle that “two pairs of eyes find more bugs than one pair of eyes”. People and organizations are still too optimistic about what they do.

Another advance will probably be that fewer customers buy new software by thinking “buy it and forget it”, but will actively control and follow up supplier quality management and testing. Reliance on acceptance testing as a bug-finding activity will and should be less.

A third area will be the use of start criteria for test execution: It is wrong to find trivial problems during testing. Problems should be prevented. Buggy software should not even be accepted into a test lab.

Contact Us

ITB -- In an outsourcing environment how do you judge if a testing vendor has tested or not and has tested sufficiently?

Hans - I want to follow up the methods applied by the vendor, the test coverage, both black box and white box, and the problem rate during test execution. I definitely would like to speak to earlier clients of this testing vendor. Usually if clients have a good experience with some vendor, this vendor has less risk attached. The trouble with all this is that you have to use time to check out information about the vendor. Many organizations find it easier to "hope it will run well".

ITB – Testers are often asked to test even if the entry criteria are not met. This can cause a nightmare to testing. What in your opinion should these entry criteria be? What do you think are the risks in starting testing even if entry criteria are not met?

Hans - The risk is trouble. Many blocked test cases. A lot of wasted work, too many releases of the test object to the test lab, etc. If there is no motivation to use any kind of entry criteria, a test manager should try to collect at least some data about how much time

and effort is wasted due to problems that could have been prevented by checking entry criteria.

There are many possible entry criteria, but here is a list of what successful project I know have used most:

For starting test specification:

- All needed information is available (maybe not as documents).
- Documents have passed review with reasonable results.
- There is at least some reasonable budget and schedule to allow you to test.

For starting test execution:

- Test environment, tools, utilities, etc. exist and checked for being in working order.
- Review of specification (test basis) and product has been done and passed with reasonable results.
- Static analysis has been done and passed.
- The product to test is available.
- Clean compile, link, and load. (Build is tested).
- The test level before has been passed. The product is stable enough. Test coverage is high enough.
- A few test cases have run well ("smoke test").

Tell us... Contribute

Feed us back your comments, questions, or suggestions on what you'd like to see in iTEST. Write to feedback@indiantestingboard.com

If you're inspired to write, you may even send your contributions to us. The decision to include your contributions will rest solely with the Editorial Team.

Write to us at contribute@indiantestingboard.com



Head Office:

INDIAN TESTING BOARD
A-24, First Floor, Sector-59
NOIDA - 201301(U.P)

email: info@indiantestingboard.com

Phone: 0120-4355270

Branch Office :

INDIAN TESTING BOARD
1/2-1, 1st MAIN ROAD, 5th CROSS JUNCTION
PALACE LOOP ROAD,
VASANTH NAGAR
BANGALORE - 560052
Mobile : 09343494617

Accreditation Managers :

Vinay Baid

Cell: +91 9310373004

e-mail: vinay@indiantestingboard.com

Pravesh Pugalia

Cell: +91 9343494617

e-mail: pravesh@indiantestingboard.com