

Tea-time with Testers

MAY 2012 | YEAR 2 ISSUE IV

Jerry Weinberg

Why Congruence is Essential in Managing

Janet Gregory

Perils and Pitfalls of New "Agile" Tester

Mike Talks

Cost to Repair

Bernice Ruhland

Contributing to the Testing Community

Anurag Khode

Mobile Apps Testing

Joel Montvelisky

TV Series on Software Tester

T Ashok

Purity. Quality. Cleanliness Criteria.

Exclusive : Over a Cup of Tea with Minal Deshpande

**If you are reading
THIS**



**then you are one of those
17,900 smart testers
who regularly read**

TEA-TIME WITH TESTERS

Subscribe here Right Away to get our all Issues for FREE



TEA-TIME WITH TESTERS

First Indian Testing Magazine to reach 97 Countries in the world !

Created and Published by:

Tea-time with Testers.
Hiranandani, Powai,
Mumbai -400076
Maharashtra, India.

Editorial and Advertising Enquiries:

Email: editor@teatimewithtesters.com
Pratik: (+91) 9819013139
Lalit: (+91) 9960556841

This ezine is edited, designed and published by
Tea-time with Testers.

No part of this magazine may be reproduced,
transmitted, distributed or copied without prior written
permission of original authors of respective articles.

Opinions expressed in this ezine do not necessarily
reflect those of the editors of **Tea-time with Testers.**

Editorial

Special Issue with Special Stuff!

Dear Readers,

Things are changing on rapid pace and hence the expectations from testers too.

This week was special for me and also for those who could attend live webinar by James Bach.

James helped us understand what skills testers must possess and what he thinks is important to learn so that testers master those skills. There were many good points that he explained which are really important for testers if they desire to be great at testing.

My special thanks to James for showing us the right direction.

Well, do not get disappointed if you could not attend it. We have arranged another coaching session which is free of cost and I am sure that will help you to improve your testing skills. Say thanks to Ajay Balamurugadas, who extended his hand to do it. Details of this session are mentioned in this issue.

Articles by Janet Gregory, Mike Talks, Bernice Ruhland and contribution from our guiding angels Jerry Weinberg, Joel, T Ashok and Anurag have made this issue really special.

Another thing that has made this issue special is 'Interview of Minal Deshpande'. You'll surely get inspiration after reading her story.

That is all from my end this time. Let's meet in June again.

Sincerely Yours,

- **Lalitkumar Bhamare**



editor@teatimewithtesters.com



QuickLook



Editorial

What's making News?

Tea & Testing with Jerry Weinberg

Speaking Tester's Mind

Perils and Pitfalls of New "Agile" Tester – 18

Cost to Repair -23

In the School of Testing

Contributing to the Testing Community – 32

Mobile Apps Testing – 36

TV Series on Software Tester – 45

T' Talks

Purity. Quality. Cleanliness Criteria.-51

Cup of Tea with Minal Deshpande

Testing Puzzle – S.T.O.M. Contest

Our Testimonials

Family de Tea-time with Testers

Testing Puzzles

by Sebi



Crossword

by



qTrace tracks user's interactions with an application and uses that information to automatically create detailed defect documentation. qTrace supports the recording of a wide range of applications and technologies on the browser as well as desktop apps.



"Software development testers are frustrated by the amount of effort it takes to communicate complex defects clearly to the development team," says Vu Lam, CEO of QASymphony.

"qTrace is an elegant solution that makes it simple for testers to isolate, highlight, and report software issues.

It intelligently produces defect reports that contain detailed scripts of each test step, complete with screen shots and system information."

Vu Lam, CEO of QASymphony.

qTrace expedites the QA process with more accuracy and brings products to market by as much as 20% faster and cuts everyday defect reporting by 70%.

Useful editing features include annotation tools such as call-out boxes, pointers/arrows, blurring and cropping, making it easy to further enhance the clarity of the defect report.

In summary, qTrace:

- Reduces manual documentation efforts and increases accuracy of defect descriptions;
- Shortens defect resolution time by providing more information to the bug fix team;
- Complements and integrates with existing application lifecycle management (ALM) and defect tracking tools.

qTrace is available as a free edition or low-cost Pro edition

[DOWNLOAD NOW!](#)

Click below to see a video of qTrace:



About QASymphony

Founded in June 2011 with offices in Dublin, Calif. and Ho Chi Minh City, Vietnam, QASymphony is a software company built to revolutionize how software is tested, adopted, and supported. We are an early stage, self-funded, software-loving team, united by a common belief that software could be better and better tested. For more information, visit (qasymphony.com).

Website: www.qasymphony.com **Facebook:** www.facebook.com/qasymphony **Twitter:** www.twitter.com/qasymphony

Press Contact: [Victor Cruz](#) , Principal, MediaPR

For more updates on Software Testing, visit [Quality Testing - Latest Software Testing News!](#)

Are you interested in publishing the news about your own firm, tools, community and conferences in **Tea-time with Testers?**

Then write to us at:

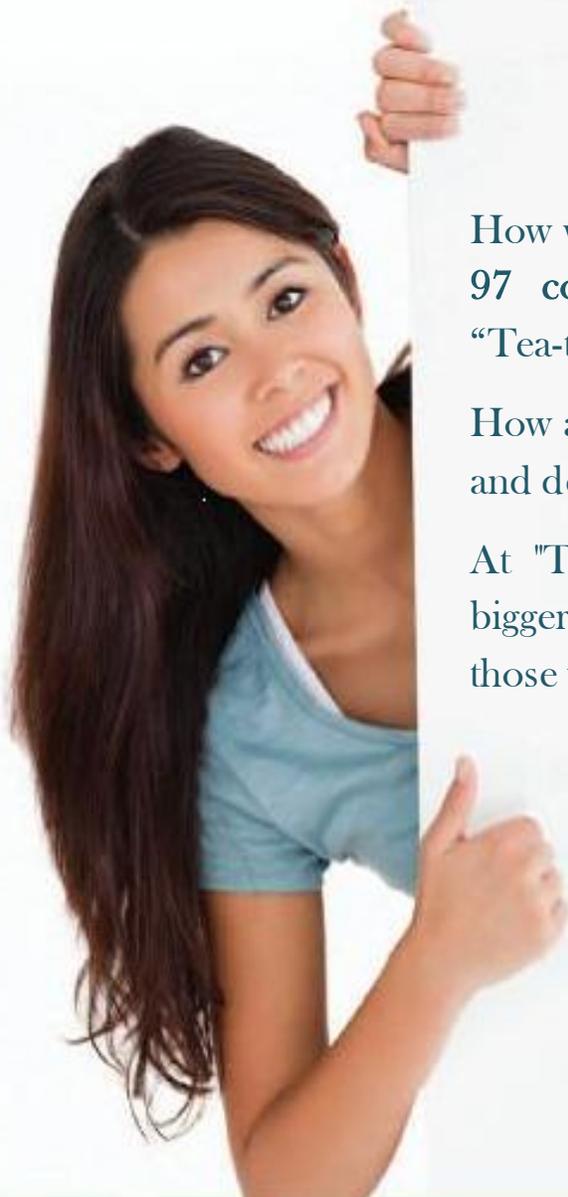
contact@teatimewithtesters.com

with “**News Enquiry**” in your subject line.

*ConditionsApply



Want to connect with right audience?



How would you like to reach over **17,900** test professionals across **97 countries** in the world that read and religiously follow "Tea-time with Testers"?

How about reaching industry thought leaders, intelligent managers and decision makers of organizations?

At "Tea-time with Testers", we're all about making the circle bigger, so get in touch with us to see how you can get in touch with those who matter to you!

ADVERTISE WITH US

To know about our unique offerings and detailed media kit

write to us at sales@teatimewithtesters.com

Announcing...

Smart Tester of the Month Award !!!

❖ Winner for Testing Crossword :

Name: Siji Kurian

Designation: Junior Software Test Analyst

Company: Hitech Outsourcing Services

Place: Kochi

Congratulations !



Discussion helps !

How about talking with us on Facebook?

Come ! Let's have a nice Tea-time there !

CLICK ON THE PAGE BELOW TO JOIN US

Tea-time with Testers



- Wall
- Info
- Photos
- Links
- Discussions
- WELCOME**

About

A Free Software Testing Magazine and forum to share, discuss,guide/learn an...

More

202 like this

9 talking about this

Likes

Teach Testing - an



Create a Page

Tea time with Testers ▶ WELCOME [Like](#)

Magazine

[f](#) [t](#) [in](#) [+](#) [+](#) [0](#) [+](#) [+](#) [0](#)



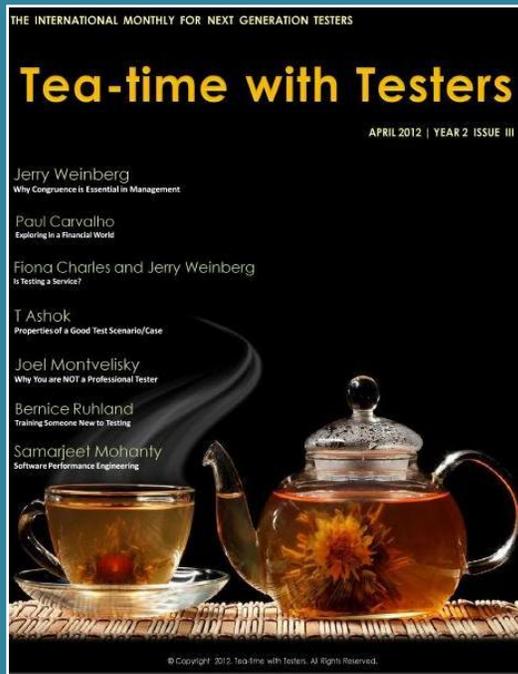
"Tea-time with Testers" is a FREE Software Testing Magazine dedicated to all disciplines in the field of Software Testing.

In this magazine you'll get to read variety of articles on software testing & also an opportunity where you can share your own views, win awards, contribute your own ideas, guide others and enjoy every bit of Software Testing.

We are sure , every sip of your tea will be worth enjoying while reading us.



 Find us on Facebook



April 2012

Our special thanks to all readers for your love and affection. Your letters have always remained our source of inspiration.

Do not hesitate to write us your feedback, opinion, suggestions.

Just email us on editor@teatimewithtesters.com

- Editor

Hello Team,

I am a regular reader of Tea-time with Testers. I always appreciate the initiative and the concept taken by Lalit and Pratik (and all those who are involved in making this a reality).

The unique thing about 'Tea-time with Testers' as compared to other ordinary magazines is that, not only IT professionals find it interesting but at times my non-IT friends also read it with equal interest. Latest news in software testing field, new ideas, expert's comments and guidance are few of those salient features.

This magazine will always stand apart from those stereo types, used to kind of technology magazines.

IT industry is ever growing field, where each day new innovations are taking place. Looking closely to any high revenue generating IT Company, the success part relies on its testing unit. Similarly, vice versa, if an IT company is not getting fruitful response from client, there is a lack of concentration on testing practices. This simply depicts the importance of "Software Testing and Quality Assurance". By each passing day, people are getting aware of this "Something" that was lacking in their success regime.

Being a Sr. Software Tester, I strongly feel the importance of testing and would not only recommend this to be a curriculum part of universities but would also love to support and give my best whenever required by "Tea-time with Testers" team.

My "Best Wishes" to "Tea-time with Testers" team. Keep it up.

- Maulik Joshi
Sr. QA Engineer
Cybage Software Pvt. Ltd.

Tea & Testing



Jerry Weinberg

Why Congruence is Essential for Managing (Part 3)

1.6 Helpful Hints and Suggestions

1. One of the surest indicators of incongruent managers is their belief they have power, and that power is one-sided. When a project fails, the employees, the customers, the vendors, or the fates are to blame. When it succeeds, it's brilliant management.
2. The concepts of congruence and requisite variety are of great generality and will appear throughout this book in a number of forms. The key underlying concept will always be *matching* two things, usually thoughts or feelings on the one hand and words or behaviors on the other.
3. As this book unfolds, it will elaborate many aspects of how to act congruently. As you read, remember that underlying all of these techniques is awareness and acceptance that you always have choices about how to respond. Whenever you find yourself saying, either to yourself or others, "I had no choice," you know you were acting incongruently and not using your full potential as a controller.

1.7 Summary

- i. This article was about how to become the kind of high-quality, effective software engineering manager needed to produce high-quality software. First and foremost among the requirements for such a manager is the ability to act in congruence with your beliefs.
- ii. Cybernetics says managers can be seen as controllers of feedback systems. To manage an engineering system by feedback control, a manager as controller needs to
 - plan what should happen
 - observe what significant things are really happening
 - compare the observed with the planned
 - take actions needed to bring the actual closer to the planned
- iii. Effective managers must know what to do, but they must also be able to act in accordance with that knowledge.
- iv. *Ashby's Law of Requisite Variety* says the action taken by the controller must be *congruent* with the situation. When people are not tapping their full variety of potential actions, they are coping *incongruently*.
- v. For control purposes, it doesn't matter *why* managers are unable to exhibit the requisite variety of action. Managers acting incongruently may not be capable of controlling the system they are trying to control. If so, the organization they manage will be unlikely to produce high-quality software, regardless of the reasons for their incongruence.
- vi. Technology is obviously important to the consistent delivery of high-quality software and software services, but in today's software organizations, management is the number one random process element. Not only that, but incongruent management stands in the way of improving all the other random process elements.
- vii. The personal effectiveness of people is what integrates all the other components of software engineering management. You'll never get a Pattern 3 organization with Pattern 2 managers. Instead, you start by getting the effective managers, then they lead the others.
- viii. The task of this volume is to address the major obstacles managers face in attempting to use full and appropriate variety in their actions.



1.8 Practice

1. (Contributed by Jim Batterson) Another common meaning of *congruence* is doing what you say you will do. It's really frustrating to work for a manager who you cannot count on to do what he/she says. Discuss how this definition of congruence is related to the one based on requisite variety.
2. Still another common meaning of *congruence* is appearing on the outside the way you feel on the inside. Discuss how this definition of congruence is related to the one based on requisite variety.
3. Discuss how all three of these definitions are related to the geometric concept of congruence, as when two triangles are congruent.
4. Give some examples where you have observed a manager who knew what to do, but was unable to do it. Give some examples where *you* were that manager.
5. (Norm Kerth) Review your past management training, textbooks, and articles. Which ones urge congruence versus incongruence? Which honor the human component and which ones ignore it, or try to suppress it?
6. (Norm Kerth) With congruence as a guiding theme, examine the biographies of leaders you admire. In particular, how did they respond when they were told they didn't have choices? How did they affect the choices available to others?



[Back To Index](#) 

Biography

Gerald Marvin (Jerry) Weinberg is an American computer scientist, author and teacher of the psychology and anthropology of computer software development.



For more than 50 years, he has worked on transforming software organizations. He is author or co-author of many articles and books, including *The Psychology of Computer Programming*. His books cover all phases of the software life-cycle. They include *Exploring Requirements*, *Rethinking Systems Analysis and Design*, *The Handbook of Walkthroughs, Design*.

In 1993 he was the Winner of the **J.-D. Warnier Prize for Excellence** in Information Sciences, the 2000 Winner of **The Stevens Award** for Contributions to Software Engineering, and the 2010 **Software Test Professionals first annual Luminary Award**.

To know more about Gerald and his work, please visit his Official Website [here](#).

Gerald can be reached at hardpretzel@earthlink.net or on twitter @JerryWeinberg

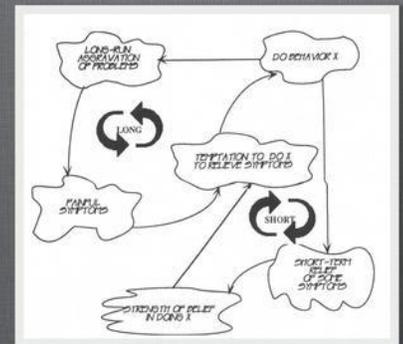
MANAGING YOURSELF AND OTHERS is yet another famous book written by Jerry.

Becoming an effective manager is the subject of this volume in Gerald M. Weinberg's highly acclaimed series, *Quality Software*. To be effective, managers must act congruently. Managers must not only understand the concepts of good software engineering, but also translate them into their own practices. Read this book to find out more.

Its sample can be read online [here](#).

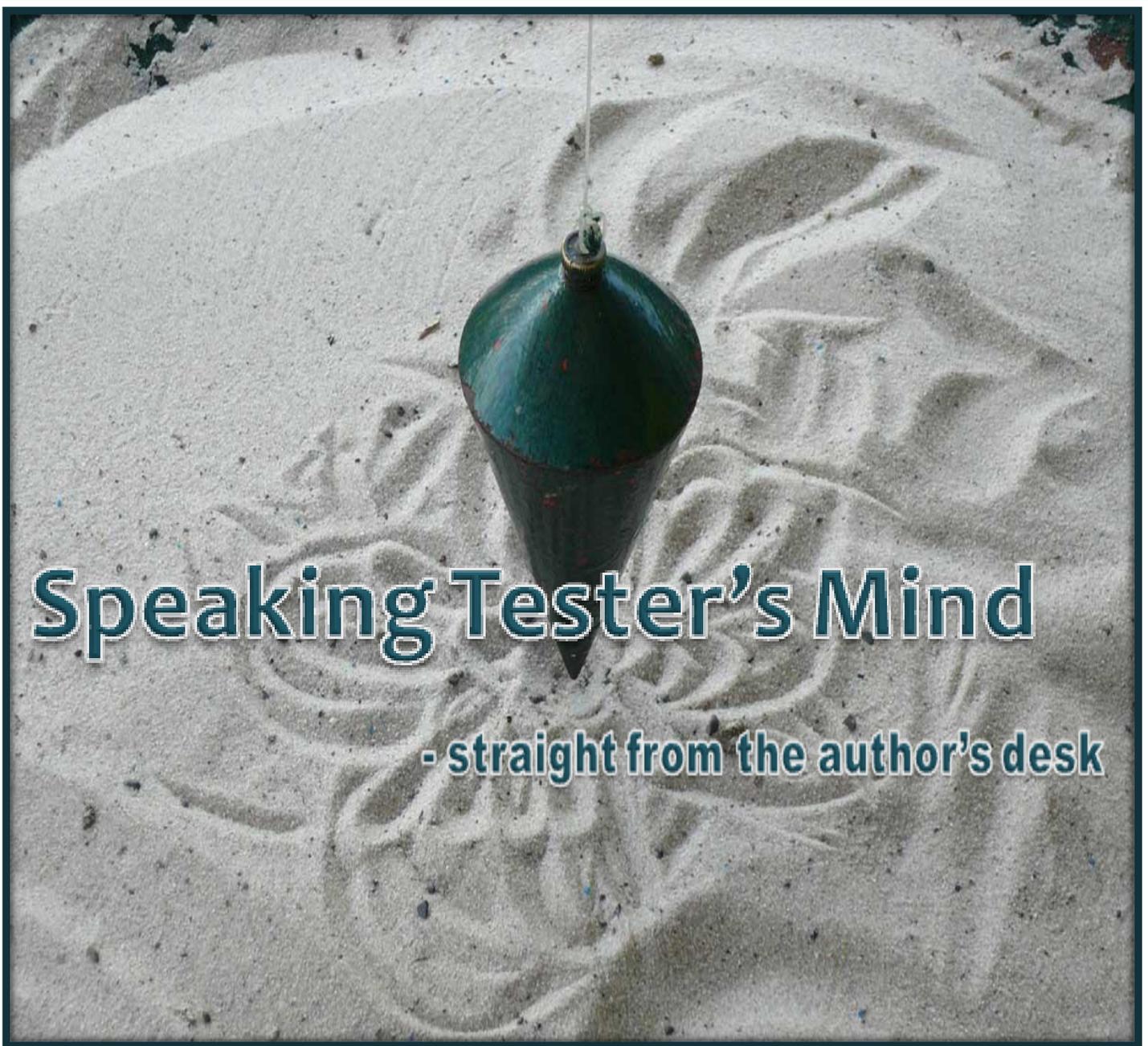
To know more about Jerry's writing on software please click [here](#).

MANAGING YOURSELF & OTHERS



GERALD M. WEINBERG

TTWT Rating: ★★★★★

A photograph of a green conical weight hanging from a thin string over a sand tray. The sand tray contains intricate, circular patterns drawn in the sand. The weight is positioned centrally over the patterns.

Speaking Tester's Mind

- straight from the author's desk

Perils & Pitfalls of the New “Agile” Tester

-part 1



by Janet Gregory

If your background is testing on traditional projects, you are used to receiving something called “requirements” that become the basis of your test cases. Then later, you’re given an operational system to test. On agile projects, you are expected to continually test changing code derived from “specifications / requirements / desirements” that are being discovered in almost real time.

Basic testing knowledge and skills are the same whether you are part of an agile or traditional project. However, there are significant differences in how a tester works.

In an agile environment, you are expected to:

- test without formal requirements documents
- contribute to fleshing out the requirements
- test in real time
- test code that is continually changing
- automate most tests
- be part of a close knit project team
- be able to step outside the tester role when needed to help other team members

If you are not prepared for these changes in your routine, you’ll find that it is hard to keep up with the team.

This article details eight pitfalls for the agile tester, along with their associated risks, and techniques to avoid each one. Let's start with some basic vocabulary.

Elisabeth Hendrickson uses this definition to describe agile teams, and I use it because of its simplicity. "Agile teams produce a continuous stream of value, at a sustainable pace, while adapting to the changing needs of the business." <http://testobsessed.com/blog/2010/12/14/the-agile-acid-test/>

Agile is the word used to describe methodologies that adhere to the values and principles outlined in the Agile Manifesto (www.agilemanifesto.org). Different methodologies use different words to describe the same concept, so I will use generic terminology and share my 'loose' definitions.

Iteration: a small chunk of time in which the team can deliver a potentially shippable product (1 – 4 weeks)

Release: a deliverable to the actual customer; it can be any length of time as long as it's frequent for your customers (let's say 3 months), and is usually multiple iterations

Potentially shippable product: a solid deliverable that may or may not be released to the customer

Feature: some business capability that has value to the customer; a feature has one to many stories

Story: smallest chunk of business value that could be delivered; usually in the format "As a ...(the person wanting the value), I want(what is to be built), so that(the reason)"

Done: is the term used to define when a story or feature has been completed and is potentially shippable.

Velocity: a loose measure of a team's productivity; used for planning

Continuous integration: the build environment which enables visibility into the build process. Every new successful code check-in runs all the unit tests and delivers a new build.

Whole team: the project team; may be extended to include other project teams if all are working on the same product.

Product Owner: the person representing the customer; prioritized features and stories for business value for the team.

Agile teams require the whole team to work together to successfully deliver solutions in a timely manner. It requires discipline from all team members to succeed.

The following pitfalls are in no particular order, but each by themselves can seriously hinder the success of the project.

1. Waiting for Tuesday's Build
2. Testers Are Not "Really" Part of the Team
3. Testing Only What the Programmers Build
4. Having No Input Into Requirements
5. Maintaining a "Quality Police" Mindset
6. Trying to Test Everything Manually
7. Thinking You Are Not Technical Enough
8. Forgetting the Big Picture

1. Waiting for Tuesday's Build

In traditional development projects, it is common to wait for a 'stable' build before starting to test. In an agile environment, it is imperative that testing begins as soon as, if not before, a story is completed. I named this "Waiting for Tuesday's Build" when a friend of mine was telling me about her experiences in her new agile team. She liked the collaboration in the team, but was finding it very difficult to keep up and found that she was testing at the end of the iteration and felt very rushed. When I asked why she wasn't testing every day, she said she had to wait for the new build to be ready on Tuesday. They were only building once a week.

Continual integration and daily builds allow a great opportunity to test code deployments on a regular basis rather than waiting until after all coding is complete or right before the release.

Four triggers to help you recognize that you are falling into this trap are:

- Stories are stacking up in the "To Test" pile
- Testers are complaining their test environments aren't available
- Stories are not tested within the iteration (carried over to the next)
- Deployments are not being tested regularly

When you wait to be able to test a story, the feedback arrives too late to be useful to the programmers. Defects or inconsistencies are not discovered until programmers are already on another story and do not want to be interrupted to fix them. Bugs pile up, and automation suffers because it can't be completed. As a tester, you get behind and may end up testing stories an iteration or two after they have been coded.

To avoid this pitfall:

- Work with the person(s) responsible for build to get regular builds and deployments to your test environment
- Learn the benefits of continuous integration; be able to articulate the reason you need it
- Implement a continuous integration that creates deployable software; make it a priority
- Plan for test environments that mimic production
- Be proactive - if you wait, you will be too slow
- Test immediately - as soon as the story is ready
- Test on the programmer's machine if necessary; understand the reason for this is to give feedback as early as possible
- Learn to pair test with the programmer
- Immediate feedback is key - provide it

2. Testers Are Not "Really" Part of the Development Team

Traditional test teams are independent. They often sit apart from the rest of the development team. In an agile team, the testers must be an integral part of the project delivery team.

It is easy to recognize this peril, but far more difficult to fix it. Watch to see if the testers are included in iteration and release planning sessions, or if discussions about stories are held between programmers and business users without a tester involved. Other symptoms which may not be so obvious include testers being told they can attend the meetings but are there to only watch, not contribute.

One of the biggest risks is that testers don't know what is going on. Changes are made to the story and testers aren't aware so don't know to test them. There is rework, and animosity may arise. Another risk is because of the fast pace and the close relationship you have with the development team, it can be easy to fall into the trap of testing what the programmer builds.

Whether you testing in a traditional or an agile environment, it is important to think about what you are testing based on the requirements. In agile, the trick is to be involved, collaborate to get a shared understanding with the customers using acceptance tests and conversation to build the right thing.

Some things you can do to prevent this issues are:

- Push the whole team attitude Check Lisa's experience - <http://lisacrispin.com/wordpress/2011/04/26/the-whole-team-approach-in-practice/>
- Logistics – sit with the programmers, not separately
- Invite yourself to the meetings and add value by asking relevant questions or offering suggestions
- Understand the story before coding starts
- Help customers develop stories and create acceptance tests first
- Ask for examples from the customer / product owner
- Use the "Power of Three" - make it mandatory; all three disciplines (domain expert / product owner, programmer, tester) are present at any discussion about a story or feature
- Be useful – constantly be testing and giving feedback to the team
- Use the "let's just try this" for an iteration or two to get things rolling

3. Having No Input Into Requirements

Agile projects do not require that requirements are gathered up-front by Business Analysts months in advance of beginning a project. Customers define features and stories, and prioritize them so the team knows the most valuable stories to be working on. Stories, along with their acceptance tests and the conversation around them are what we would term 'requirements'. Testers can provide a very critical role in requirements definition.

Symptoms of the team falling into this trap include testers not talking during iteration planning meetings, and business users define stories and acceptance tests by themselves without consulting any of the other team members. Another indication is that your testing is finding bugs that are actually missed requirements and programmers have to add extra code to finish the story after they think they are done.

There are several risks if testers don't have input into the requirements. Assumptions may not be uncovered early if no one asks the questions during the release and iteration planning sessions. Testers often have the greatest knowledge of impacts to the system so the risk is that impacts to the system are found late in the release rather than caught earlier. A less tangible risk is that not the tester's skill set is not being used to the team's advantage.

The risks are lessened if there is continual conversation between the business users, testers and the programmers. However, if there isn't, and the programmers 'guess' at what is right because they don't want to bother the users, there is a strong possibility that they will build it wrong, and will waste time and resources refactoring.

As a tester, there are some things that can be done to avoid this pitfall.

- Understand your role as tester
- Identify your test variations before coding is started and share them
- Have the programmers refuse to code without tests
- Be confident, be courageous
- Ask clarifying questions based on quality characteristics: complete, unambiguous, testable / verifiable, consistent, accurate
- Bring your testing tools with you to the planning meetings, and ask yourself 'how would I test this'

4. Not Keeping Up with the Programmers

In a traditional environment, testers don't get the application to test until after the programmers are finished coding the whole application or a large feature set. In an agile project, testers are expected to test each story as it is finished. This means the stories must be small (typically scheduled to be completed in 1-3 days in a 2 week iteration) and testable. The testers need to be ready to test them when the coding is completed.

As in "Waiting for Tuesday's Build", the symptoms are similar.

- All the stories aren't tested at the end of the iteration
- Stories stack up in the "To Test" pile

If testing activities don't get completed as soon as the programmers are coded, the stories get tested late in the game. When programmers don't get fast feedback such as suggestions and bugs, the testers can lose credibility. Velocity is affected because a story cannot be marked "done" until it is tested. This makes it harder to plan the next iteration.

To keep up with the programmers, try these techniques:

- Plan testing and programming tasks together
- Include testing tasks in velocity
- Make sure the stories are both small and testable
- Anyone can sign up for testing tasks
- Automate as you go – get programmers involved
- Plan so that testing is balanced with coding

[Back To Index](#) 

to be continued in next issue...

An agile testing coach and practitioner, Janet Gregory is the co-author of Agile Testing: A Practical Guide for Testers and Agile Teams and a contributor to 97 Things Every Programmer Should Know. Janet specializes in showing agile teams how testers can add value in areas beyond critiquing the product; for example, guiding development with business-facing tests. For the past ten years, Janet has been working with teams to transition to agile development, and teaches agile testing courses and tutorials worldwide. Janet contributes articles to publications such as Better Software, Software Test & Performance Magazine and Agile Journal, and enjoys sharing her experiences at conferences and user group meetings around the world. Janet was named one of the 13 Women of Influence in testing by Software Test & Performance magazine – January 2010.

For more about Janet's work, visit www.janetgregory.ca or visit her blog at janetgregory.blogspot.com/





Cost to Repair

By Mike Talks

Back in January the air conditioning on our car decided to play up ... in the middle of the New Zealand Summer. The fan would now only work in one position, and in that position the air conditioning wouldn't work. The air conditioning would work if the fan was off, but it would only slowly seep into the car.

I drove it around to my local garage for their opinion. Before I volunteered my car keys over to them to work, I asked what the probable problem was, and the magic question "what was likely to be the cost to fix?"

It's true, whenever we need work done and need to get someone to do it for us, we like to have an estimate, a ball park figure of how much it'll cost. For the car problem, the car is 15 years old, and if work becomes more than a certain value to fix I have to question is the work worth it, or should I really think more of living with it for now, but look to replace the car later this year?

I know if the decision is over the engine or brakes, I need to make the decision now. But air con? I guess I can put it off for a couple of weeks if I need to.

There's a whole spectrum of judgment going on there which of course testers will be aware of. The car has a defect - it's not major, endangering the occupants, there is even a work around



(it's called open the windows). Hence I have to deal with "how much is it going to cost", even "heck does this need fixing or can I live with it" and "when can I get it fixed".

Returning to my previous article in Feb'12 issue of Tea-time with Testers, i.e. Understanding the Defect Mindset, this was something I touched a little on, but never elaborated on.

Jerry Weinberg offered the following feedback:

A defect could have important consequences but require a total rewrite of the system to correct it-- which may be deemed impractical. Perhaps the reason people don't mention "cost to repair" is that the testers may not be in a position to evaluate that cost. On the other hand, "consequences to our organization" may also be impossible for testers to evaluate.

"Cost to repair" is perhaps something as testers we're uneasy about. For good reasons – we like to think of ourselves as the people who find the problems. Someone else (developers) have to fix it, and someone else (project managers) have to pay for it.

But trying to estimate the cost and rework a defect entails is important. Whether we like it or not, money makes our projects possible, and there's only so much a project manager has available. As I test manager I am finding myself working more and more with project managers and the financial aspects of the project. A project manager only has so much budget given by "The Business" to achieve the project goals.

These estimates have to be as accurate as humanly possible – I know a few developers who always tell me it'll just take 30 minutes to fix an issue. Really? Is that find the cause of the problem, change it, build it, unit test it then package for production? In 30 minutes? It can be the equivalent of a car mechanic telling you changing an oil filter is a 2 minutes job ... but forgetting to tell you on your make of car he first has to winch out the engine then put it back afterwards (not 2 minutes).

If a project is encountering a lot of issues, it's going to run over. As projects run on budget, overtime usually equals over-budget. Yes savvy project managers keep a little bit of contingency budget in case things go a little wrong, but it's usually just a fraction.

When a project is looking to go over-budget there are a couple of things they can do – both of which involves going back to "The Business" ...

The Baloo method is to try and de-scope elements of the project, just try and deliver the bare necessities, but within budget.





The Oliver method is to ask the Business for more money.

In truth most Waterfall project managers use a combination of Baloo and Oliver methods to get into production. This usually happens at some kind of Project Steering Group, which is probably the reason your project manager seems grumpy some afternoons.

You see, Project Steering Groups generally don't like to hear that a project is going to cost more to deliver less. This is human nature, think of us in the garage with a surprise bill – we're not happy. But also we know the truth of technology – some jobs turn out to be much bigger than initially expected as we encounter problems as we go along.

Fortunately few of us testers get to see the Project Steering Group, and a summons from one is about as welcome as an appointment with the Spanish Inquisition,



Most project managers describe the process as like being a pig slowly roasted on a spit. Questions are asked along, "why are we having problems", "what are the problems", "how much more do you think you need", and "how can you be sure that's enough".

In such an intense situation, the project manager's best tool is the defect reports you as testers have put together. It will highlight the cause of the major defects encountered. If some are expensive to fix, the "cost to repair" can allow the project manager with the steering group decide if the repair is really worth it.

If development shows up a high impact and costly issue, it might be worth just stopping the project there if it costs too much to fix.

Especially if you're on a legacy system (so many of us are), it might be worth retaining the system "as is" but speeding up it's replacement because the system just cannot cope with some new areas of functionality. These aren't decisions developers or testers can or should make, but they need to assist the project managers and steering group to make the best informed decision they can make.





Mike Talks is the chief tester for Payment Services at Kiwibank in Wellington.

An ex-programmer – but don't let you hold that against him. He's worked for 16 years testing projects ranging from complex aircraft avionics to simple websites.

Cursed by gypsies at an early age, he finds himself invariably breaking anything he touches. Nowhere more so than when he tries to use technology.

He is on Twitter as TestSheepNZ



Do **YOU** have **IT** in you what it takes to be **GOOD** Testing Coach?

We are looking for skilled **ONLINE TRAINERS** for Manual Testing, Database Testing and Automation Tools like Selenium, QTP, Loadrunner, Quality Center, JMeter and SoapUI.

TEA-TIME WITH TESTERS in association with **QUALITY LEARNING** is offering you this unique opportunity.

If you think that **YOU** are the **PLAYER** then send your profiles to trainers@qualitylearning.in .

Click [here](#) to know more

There was a time when people did not have compass to find right direction. The only guide they had was that guiding star up in the sky.

Do you think that you are also stuck somewhere with technical issues? Do you need help in decision making or want guidance?

Well, the wait is now over . Introducing...

“The Guiding Star”

*The panel of our experts is now here to help you.
Send us your questions around software testing and our Guiding Stars will help you out.*



E-mail your question on –

theguidingstar@teatimewithtesters.com

Please Note :

1. This is not a job portal.
2. Typical interview questions will not be answered.
3. Questions should be on Software Testing or related topics only.



In the school of Testing

for your better learning & sharing experience

**we are pleased to announce
FREE skype coaching**

“Enhance Your Testing Skills”

~ Course Description ~

**Building a Feature Map
Information gathering through Application Tours
Effective bug hunting, investigation and reporting
Preparing a Test Report**

When?

Date: June 04th to July 06th 2012

Time: 10.00pm IST to 11.15pm IST (4.30 pm GMT to 5.45 pm GMT)

For other time zones, <http://wwp.greenwichmeantime.com/gmt-converter/>

Days: Monday, Wednesday, Friday

How to register?

Send an email to ajay184f@gmail.com with subject "AjJun2012"

and mention your [Skype id](#) in the body of the email.



~ PLEASE NOTE OUR NEW CONTACT DETAILS ~

➔ For Editorial enquiries, write to us on editor@teatimewithtesters.com

➔ To contact us, mail us at contact@teatimewithtesters.com



“Career Development and Learning Strategies for Testers” is a series of articles providing different approaches to develop testers’ skills and knowledge from both a managerial and tester perspective.

By Bernice Niel Ruhland

Contributing to the Testing Community



“Career Development and Learning Strategies for Testers” is a series of articles providing different approaches to develop testers’ skills and knowledge from both a managerial and tester perspective. The focus of this article is the Testing Community. Becoming involved in the Testing Community is important as it provides a wealth of knowledge and learning opportunities. Our world has no boundaries when we connect with testers around the world. Understanding how they are approaching similar problems can introduce new tools such as laying out testing strategies using a Mind Map.

There is not a one-size-fit-all approach. It is important that each tester tries different approaches to build a network and to develop those relationships (whether it is through email, Skype, or in person.) If you try Twitter and find the mode of communication too limiting, then you might want to try G+ or participate in Weekend Testing. The actual path you take is not important – it is important that you connect, build relationships, and build a reputation. Not only can this reputation help you find a job but it allows you an avenue to develop your skills and help other testers.

Social Media

There are many social media options that include Twitter, G+, Facebook, and LinkedIn. Understand the strengths and weaknesses of the different options to determine a strategy. For example, LinkedIn may be used to maintain connections with former co-workers, college friends, and testers. Belonging to LinkedIn groups may be helpful for in-depth discussions. Twitter is great for quick communication and keeping a pulse on what is going on in the testing community. G+ allows deeper conversations within the Technology community with the ability to “blog” within a comment. Some people maintain a private Facebook page for only people they personally know. Identifying a strategy for social media usage can help define parameters and encourage your own usage.

Remember the 80/20 rule? Basically it translates to 20% of something is responsible for 80% of the results. How might this affect your social media strategy?

- You may need to connect with more testers in order to have a meaningful amount of information.
- If you review your Facebook friends list, your twitter following list, etc what percentage of those contacts are really participating in the community? What percentage are you in?
- If you are not part of the 20% group, is social media the right strategy for you? Does anyone know who you are through social media?

If you are using Twitter, the blog referenced below provides 20 testers to follow. In addition, the testers who commented on the blog will suggest additional testers to follow. It is helpful to follow a diverse group of people to gain different viewpoints on the same subject.

See this blog: <http://blog.utest.com/top-20-software-testing-tweeps/2010/05/>

Local Testing Groups

If there is a testing group in your local area, try to connect and attend meetings. Many of these groups will have a LinkedIn page or use other forms of social media to connect. Attending local meetings can open new opportunities such as making a presentation at a meeting. All connections are important whether you are able to meet the person or not. When there are opportunities to meet someone in person it is a great way to further develop that relationship.

Weekend Testing

Weekend Testing is a wonderful way to develop relationships and improve your exploratory testing skills. Working with other testers helps you understand how they found a critical bug that you missed; how to manage your testing time within a time constraint; and think through testing strategy plus much more! The three chapters' websites are listed below. Contact the appropriate chapter for more information.

<http://weekendtesting.com/>

<http://weekendtesting.com/chapters/america>

<http://weekendtesting.com/chapters/europe>

Blogs and Websites

There are a lot of blogs and testing websites that it can be information overload and a time drain. A strategy is helpful in managing this problem. Below are a few suggestions.

- As you build relationships with certain testers (ie., Twitter, Facebook, email, skype) check out their blogs. Determine which ones you found valuable and consider following their blogs or at least reviewing them periodically.
- If a tester posts a link to his blog, try to read it and comment if appropriate. Testers put a lot of time into writing these blogs, we should acknowledge their efforts. Do not comment only on the popular testers – treat everyone equally.
- Check out this list of 100 blogs to see if any of them interest you. <http://www.testingminded.com/2010/04/top-100-software-testing-blogs.html>
- Allocate some time every day for reading blogs and articles. If you do not have much time, start with 15-minutes. Consider making it part of your lunch break.

Contribute to the Testing Community

Just as it is important to learn from the Testing Community, it is also important to give back to the community. Do not just read what everyone else posts – get involved! Be sure to provide input on questions posted in social media forums if you have expertise or knowledge in the area. If you read an article / blog that you think others may be interested, post it. However be careful not to spam your connections with too much information. This can include reposting information. Try to maintain a level of quality to your postings. Otherwise testers may drop you from their friends and following list.

Tips for Managers

- Incorporate elements of connecting with the Testing Community as part of training programs.
- Because there are so many options, try to direct testers to websites and options to connect to, that will get them started. Over time each tester should identify what works well for him.
- If you facilitate Journal Clubs, identify opportunities to introduce new connecting opportunities. For more information on Journal Clubs, see my article published in Tea-Time with Testers, February 2012 issue.

Tips for Testers

- Do not wait for your manager to make suggestions. Reach out to the testing community to start building a network.
- Try different avenues such as Twitter, Weekend Testing, and LinkedIn to determine what works best for you.
- Develop your own strategies for staying connected. Over time that strategy may change – and that is okay.
- As a potential measurement, ask yourself outside of the people you work with -- who knows you are a tester and your testing abilities? How large have you expanded your network? What is the quality of that network?

[Back To Index](#) 

Bernice Niel Ruhland is a Software Testing Manager for a software development company with more than 20-years experience in testing strategies and execution; developing testing frameworks; performing data validation; and financial programming. To complete her Masters in Strategic Leadership, she conducted a research project on career development and onboarding strategies. She uses social media to connect with other testers to understand the testing approaches adopted by them to challenge her own testing skills and approaches.

The opinions of this article are her own and not reflective of the company she is employed with.

Bernice can be reached at:

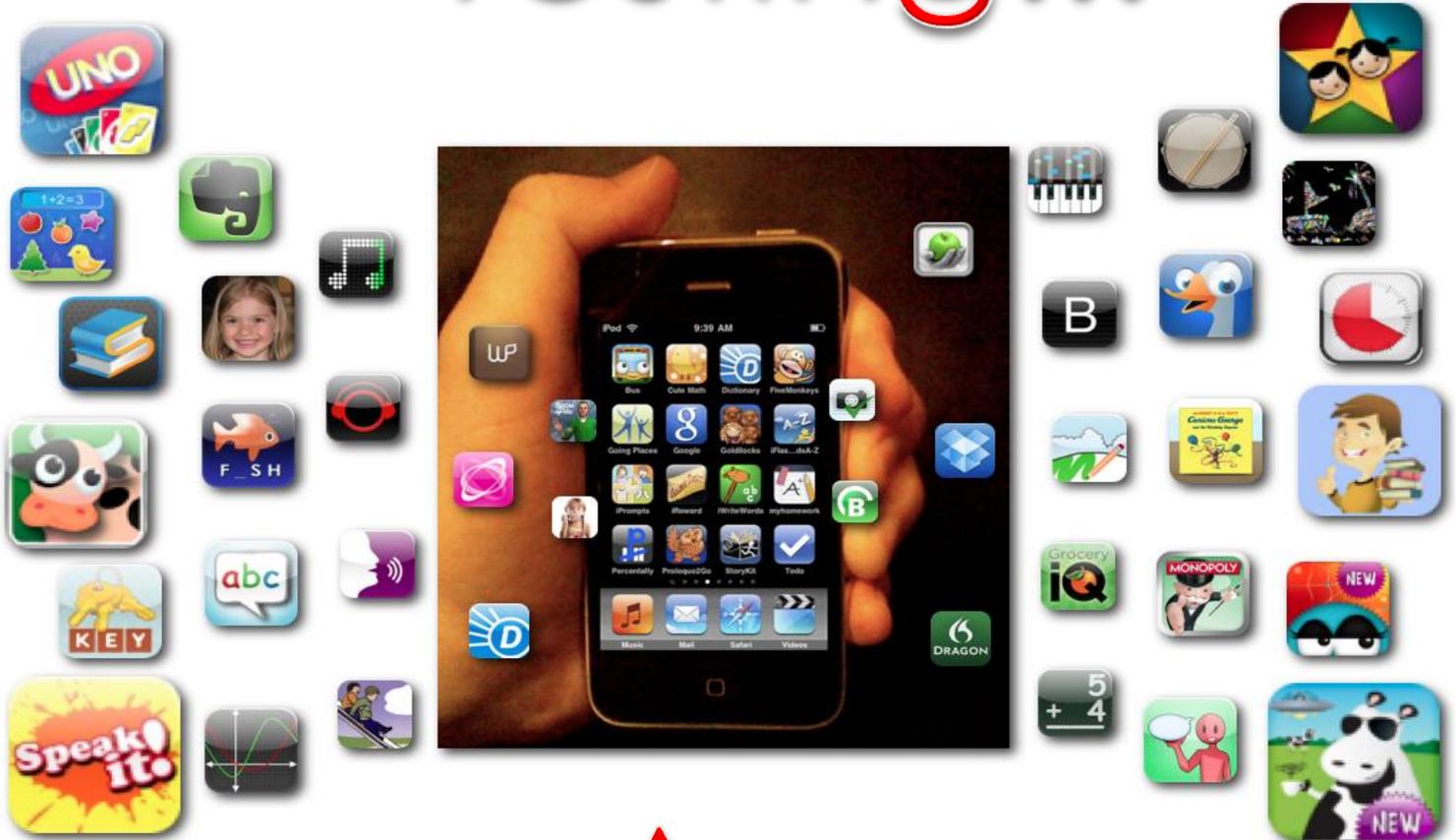
LinkedIn: <http://www.linkedin.com/in/bernicenielruhland>

Twitter: bruhland2000

G+ and Facebook: Bernice Niel Ruhland



Testing...



with Anurag

SeeTest: A Cross Platform Mobile Automation Tool

Mobile Apps Automation Testing is one of the hottest topic in the testing market now a days. Everybody is keen to know which mobile automation tool is best in order to fulfill their automation needs. Looking at the scope of the project and the development platform, everyone is coming up with their own mobile automation need. To satisfy such Automation Needs for your Mobile Applications, SeeTest (from Experitest) can surely help you.

What is SeeTest?

SeeTest is an Automation tool from Experitest which can help you automate functionalities of your mobile applications on various platforms like Android, iPhone, Blackberry, Symbian & WindowsPhone 7. It is one of the most comprehensive mobile test tool today in the market.

It covers almost all OS, all functionality. SeeTest has plugins into all testing environments such as QTP, Testcomplete, MSTest, JUnit, Perl and Python.

Features:-

- You can **Record and Play** test on real Smartphone devices. The C# code is generated automatically.
- SeeTest supports multiple platforms like *Android, iPhone, Blackberry, Symbian & WindowsPhone 7, HTML 5.*
- SeeTest supports all mobile OS versions and mobile device models.
- It supports all Gestures like Swipe, drag & drop, zoom in and zoom out, multitouch.
- Supports System alerts: Security pop ups.
- Virtual keyboards: It supports all keyboard configurations.
- SeeTest has plugins into all testing environments such as QTP, Testcomplete, MSTest, JUnit, Perl and Python. Plugin is also available for IBM's Rational.
- SeeTest has ability to run the very same test script on any mobile OS, physical device. With help of SeeTest you can run same script on multiple devices at a time.
- SeeTest object identification is based on 4 methods: image recognition, Native ID, Text recognition and Web DOM.
- SeeTest Supports Emulators and Devices also.
- Automatic Learning & Sensitivity setting of Elements – using advanced algorithm and self learning template matching.
- For devices No Jailbreak and No rooting is required.

Tool Type:-

Functional Testing Tool

Pricing:-

\$2499/ year (All Modules, All versions of SINGLE mobile OS iPhone / Android / Blackberry / Symbian WindowsMobile. You can checkout more about pricing [here](#)

Free Trial:-

You can download the 10 days free trial of SeeTest by filling short form [here](#)

Supporting docs & Videos for SeeTest:-

To know more about SeeTest you can watch a Video [here](#)

You can check for more details on [experitest's site](#)

I hope this info will help you explore this tool further. Just give a try with the Trial Version and let me know your experience. Till then stay tuned for my Next Article. Take Care ☺.



Anurag Khode is a Passionate Mobile Application test engineer working for Mobile Apps Quality since more than 4 years.

He is the writer of the famous blog **Mobile Application Testing** and founder of dedicated mobile software testing community **Mobile QA Zone**.

His work in Mobile Application testing has been well appreciated by **Software testing professionals** and **UTI** (Unified Testing Initiative, nonprofit organization working for Quality Standards for Mobile Application with members as Nokia, Oracle, Orange, AT & T, LG Samsung, and Motorola). Having started with this column he is also a Core Team Member of **Tea-time with Testers**.

Contact Anurag at anurag.khode@hotmail.com

There was a girl who graduated engineering in 1997. Later she joined one reputed IT firm through campus recruitment as a design engineer.

There came a moment when she had to make one conscious decision and she chose to be a *Software Tester*.

Today after 15 years, she is a Specialist Leader (Vice President India Level formerly) with Deloitte and leads the Testing Center of Excellence.

Tea-time with Testers decided to speak with her to discover that major credit of her consistent growth goes to the field of Software Testing, which she is immensely passionate about.

Won't you like to know how *software testing* helped her to climb the ladder?

Listen to **Minal Deshpande** in this exclusive interview with Tea-time with Testers.



Over a Cup of Tea with Minal Deshpande

Before we start, we would like to know how things started off. How did you land up in Software Testing?

Well, I graduated in engineering from a well known college in India. As a part of the campus recruitment, I joined a reputed IT firm as a Design engineer for commercial software. In about 6 – 8 months, the firm bought Rational tools with a validation perspective in mind. There were not many resources that were trained on this tool which would be used to validate the design that would go to the market with the product.

The firm was looking for volunteers who were willing to learn and take initiative to lead the group. Initially I was not much keen for this as I did not have prior experience with the tool. But later I decided to jump into this believing that it would provide me an opportunity to learn and explore new tools. So this was the first tool that I used from the testing perspective. I had to setup and learn everything by myself. Later it became a team of 2 and then 3 and I started leading them. And that's how my career started with testing.

Within a period of 3 years I had a very solid testing team. That's where my passion for software testing came up and I started liking whatever I was doing. It also gave me a lot of growth. The journey was a combination of learning and establishing something new and then growing it. All these things were outcome of testing and so that became my passion from that time onwards.

So, can we say that you became a tester more by chance than by choice?

It was more by choice because I jumped into it.

Initially I had planned to be in Design – Systems Analysis and testing at the same time but later my testing skills outgrew my design capability which was a very conscious decision at that time.



Minal Deshpande, Specialist Leader (Vice President India Level formerly) with Deloitte leads the Testing Center of Excellence. Minal is a dynamic women leader well known within Deloitte and in the Industry for her invaluable inputs and contributions in the field of technology. She is the brain behind India based Testing CoE and has successfully managed the Testing CoE for the last 7 years by developing niche and high end testing competencies. Minal has supported aggressive growth of testing practice across industry domains, technology sectors and functions.

Minal has extensive experience and success stories of supporting the large clients from proposal to delivery. Minal's strength lies in making client presentations, developing and extending high value testing services to the clients for better ROI. Minal played a key Leadership role for many of the Fortune 500 clients for Deloitte.

In addition to her technical background and experience, Minal has a strong support and is actively involved in advancement of career for women. She has contributed immensely to several national programs within Deloitte.

Minal has gained eminence not only within Deloitte but at the marketplace as well. She has been very proactive in representing and participating at Industry forums like "International Testing Conferences", WILL – Women in Leadership Forum for Corporates in India, "Green Group" etc. Minal has the honor of being the Advisory Board Member of ETI – QAI world wide's Testing Institute and Project Management Institute for BFSI conferences. Minal has to her credit many publications and her interview on "Future of Testing in India" is amongst the highest viewed video on Viddler and SQABlogs. Minal is also highlighted in "Women in Emerging Markets" as part of Diversity Journal (NY).

Minal sums up her achievements by adding, "I truly stand by "Pay it forward" and look ahead to take Testing in Industry and within Deloitte to next level of excellence. We have a long way to go and this is just the beginning"

You have been associated with Software testing since 1997 which is considerably long time. How do you compare maturity of Software testing compared to the quality processes in other industries in view of exponential growth of software industry?

Initially when I started, everything was very new, ad-hoc and based on personal heroics. It was based on what and how one wanted to do it. It was not very standardized. But over the time there has been a big difference. A lot many tools, techniques and process standardization have come up. People are more aware and focused on what they want to do, where their testing efforts will lead them and their organizations to. Earlier there used to be debate why we need a specialized tester when one can do development today and testing tomorrow.

Today things have changed that such work is outsourced to big organizations/ firms and at the same time the organizations have their own testing and quality control groups. This change is a big journey in itself. According to me it's not a big time-span, and testing has evolved to a great extent but there is a lot of scope for maturing even further. With the next leaders that are coming up, I am sure that they will be able to achieve it.

Now if you compare that with other manufacturing firms that have very strict quality control process, I don't think within software it is very similar. It is very robust and we do it well but it is not that strict as in manufacturing and I don't think it would even work in our organizations. Everything is a mandate there which cannot be done here. So we will achieve maturity in our own sense with the pace we are going at.

Do you think that the exponential growth of software industry has lead to compromising on the quality processes?

I do not think so. We all do our jobs well. Growing does not mean compromising but can be taken as less of time and efforts being put on research and building up stuff which would benefit us and help us grow. In recent years a lot of efforts have been taken by people like you guys and another group that I met in Bangalore. There are people who have dedicated their career to do research and coming up with new techniques. Such things were not happening earlier. So the additional intellectual stuff that was missing is now coming up. In comparison with past, now even IT firms have started to recognize the efforts being put into testing. Testing professionals are now getting more recognition than before.

Do you see any difference between testing philosophy that is typically followed by a testing unit within organizations and testing philosophy followed by consulting firms/independent consultants?

Yes, they do vary to some extent. Large Firms have their own tools, techniques and processes setup for themselves from testing perspective. Either they do it by themselves or they have other vendors do it for them. Consulting, pure play testing and independent consultants make clients move up the maturity chain in testing based on their experience in overall industry and it is not confined. However, the testing units have some more priorities like their own data security, compliance audits etc. It depends on services and priority of organization.



How do you differentiate the state of software testing today, four years ago and what is your predication about it for the next 5 – 10 years?

I think it is growing at a very fast rate. From all the discussions like whether testing needs to be a separate stream to where each firm having their own testing departments, this has been the growth in testing field. Then has come a phase where clients themselves are looking for independent testing and end to end testing offerings. This is a very big step that has evolved. According to me the next step evolving is Test Driven Development (TDD) that is testing leading it from the front. This is where we are headed. With all the Agile and other fast pace models coming up and tools being developed this will gain all the more importance.

The only thing we as a testing community needs to do is to up-skill ourselves to meet this. We should not fall behind in technical and /or functional knowledge because testing is a very niche skill according to me and not everyone can do it. It is a techno-functional domain which requires a mindset of its own to break rather than make and understanding bigger picture.

People think that anyone can do it but it's the biggest myth we can have. It requires skill, intelligence, mindset, technical and functional knowledge and we should not fall behind on it now. Now is the time that we lead. If we fall behind now, all the expected growth may not happen.

We were about to ask you how can a person grow as a tester and you already answered that.

I am equally passionate about testing as you guys are. So let me share how I grew in my testing career. There had been times when I had to move out of testing so as to accommodate with project requirements but yes, with software testing I have consistently grown in my career.

I always tell my group that it is not just about doing a thing. It is about doing things differently using tools and techniques. Let whatever the area be, you need to become techno-functional specialist and then wear that testing hat.

If we excel and do well, I think we will take over.



When you talk about passion, what would you advice someone who is mentoring others and trying to ignite a passion in them for testing?

(Smiles for a while) Things turn out very transactional at times. People show interest today but they disappear the next day.

Something that I have learned from my experience is that if you do anything passionately, be it testing or anything else, if you have a vision and you want to go towards it, first identify what you want to do, keep acting, don't give up . Challenges come and go but you have to stick to your mission. Unless one owns vision and passion, it's hard to rise in career. Don't shy away from putting efforts; don't shy away from putting in your 120%.

The second thing would be one's transactional view that I mentioned before. As long as you have a transactional view, you won't rise in your career. There has to be a large and strong vision towards which you are marching and I learned it in earlier days of my career. I have had very good mentors who groomed me and pushed towards it and this I got it very early in my career. Your efforts as a mentor are wasted if you stay with people who have transactional vision.

You have seen Indian Testing community since long. There are testers in India who have created their identity in global testing community and have earned respect of experts like James Bach. But do you think that number of such testers from India is still less? These testers know and practice the latest testing techniques but there are many testers in India who are still struggling with basics. What do you think should be done to mitigate this gap? How can we bring the whole community together, on same page?

There are folks who have done great work. We should put in efforts to make them recognized globally. We should work towards making a larger impact from these few and also try to have a larger group there. We need to find ways to promote these hardworking people, leaders, people who are coming next and put them in front.

Let me tell you my own experience, there was a first global conference on TMMi organized by TMMi. The chairperson and the whole board were present. It happened in Seoul, South Korea. They had called for papers and globally only six papers were selected. Ours was one of them. When we reached there we realized that most of them were related to TMMi. They had reached there because they had deep knowledge of it. We were the only people who were implementers. We didn't talk about the model, we talked about the implementation and what were the challenges that we faced. People who attended that were from industries and they were keen to know about its implementation, so our paper was very well received.

This is just an example and we need to do a lot more things which require a lot of efforts. To go over and above your duties, it sometimes becomes very difficult. We have always encouraged people who are willing and I think every firm should to do that. In all the various ways we can to promote our own community. If we don't promote ourselves it won't be useful.

That is all we have in our list for you in terms of testing. Apart from this we know that you are doing a lot of work in the field of women empowerment. Would you like to share some message with all our readers?

I always believe in giving back to the community. One of course is my testing community as a profession. It is about "paying it forward", the word that we use in Deloitte. And the second such community where I feel that I should give more is women. I want to empower them on their career paths.

I have many mentees who are women not only from testing. The same message of staying by your passion goes there also. If they have to do multiple things they have to manage their professional life. All of us get those kinds of situations. So for both my philosophy stands, you need to have a vision and passion for marching towards goals that we have.

How do you strike a balance between your personal and professional life?

First is to know yourself, identify vision and stay by it. At times some things will drop and some things will gear up. Be cognizant of that fact. You'll fall (I have fallen many times) but then we go with double success with that experience. You should not be bogged down by all those things. It might take a long time to grow but it becomes very strong and very mature.

The other day my client sent me this book, 'Why Success Always Starts with Failure'. Few of my folks worked for this client and he said this is a very good book, and wanted to share with me. It starts with your own lessons. It's about looking back, figuring out what you need to do and you then stay by it.

We heard that you do a lot of things around environmental awareness.

That was a long time ago. I used to do that before my marriage. Now I don't get much time to focus on it but I was associated with a group called "Green Group" for preventing environmental hazards, and promoting environment safety. I did work with them for around two years as a part of the community doing social work.

When you look back into past, how do you see your journey till now? Is it like I am in a better position or like miles to go before you sleep?

Miles to go before I sleep and this is what keeps us moving. I think I have come a long way but I do want to get into something else as well. I do have my vision for which I am doing it.

Last question, did you get a chance to read 'Tea-time with Testers'? We would love to have your feedback.

I liked the concept and initiative. Congratulations on the work you do and Thanks for your efforts.





are you one of those
#smart testers who
know d taste of #real
testing magazine...?



then you must be telling your friends about ..



Tea-time with Testers

Don't you? 😊



Tea-time with Testers !

first choice of every #smart tester !



testing intelligence

- its all about becoming an intelligent tester



an exclusive series by **Joel Montvelisky**

If there could have been a TV series on Software Testers...

Has anyone ever heard a kid saying: "When I grow up I want to be a Tester"?

I have 3 kids at home, and I've never heard any of them say something like that.



Up to now I've heard them say that when they grow up they want to be: a Doctor, a Policemen, a Firefighter, a Princess, a Veterinarian, a Dancer, a Soldier, a Bus driver, a Tractor driver, a Builder (or more specifically Bob the Builder!), they want to be Mickey Mouse, or Minnie Mouse, or Strawberry Shortcake, or Smurfette, or even Tom & Jerry...

But never have I heard them saying that when they grow up they want to be a Tester.

Testing is a grey profession

I guess after all testing is what they call a “grey” profession, or something that does not sound very exciting to kids; maybe it doesn’t sound very exciting to many grown-ups either?

On the other hand, there are many other “grey” professions that have enjoyed a surge in interest after getting some bright-color-lights in TV series or Hollywood movies. For example, archeology after Indiana Jones, crime-scene investigation after CSI (NY, LV, Miami, etc), pathology after Bones and NCIS, etc.

So maybe what we need to change the way people perceived testing is to come up with a TV series describing the work and adventures of an amazing team of clever software testers...



How would a TV series about Testers look like?

Let’s imagine a chapter out of our new TV drama:

T-G-I

(*Testing Geeks Incorporated*)

Scene 1 – Intro

The phone rings in the middle of morning, while the testing team is going over a regular task in the office (to make it look cool, let’s assume they are testing a multi-player virtual reality game of sorts).

One of the guys answers the phone and on the other side of the line there is someone asking for help with a very big and important problem.

The team scrambles to the Video Conference room and start a call with their customer, the *Department of Defense*.

In the conference a General explains the current crisis; half their users (including under-cover agents in enemy countries!) are not able to work with their encrypted communication software and are basically in the dark.

The General even shows how, when he tries to open this program on his laptop, he gets a general error message saying the system has performed an illegal operation and will shut itself down immediately.

The situation is critical and they need the **T-G-I** team to get in quickly and find the bug right away!

Scene 2 - The team goes into action

Our team of testers (two guys and two girls wearing jeans, sneaker and cool t-shirts) gather their equipment in their backpacks and scramble out of the office in a hurry as a military chopper lands on the lawn in front of their office, picks them up, and takes off without even slowing down its rotors.

They land in a military base and get out running towards the General who is waiting for them in a Jeep. He greets them and drives them into a secrete facility deep within a mountain cave.

Once inside the team divides into 2 groups. Team A will make a detailed analysis of the logs and the information gathered by the IT department, trying to look for clues on the bug. At the same time Team B will take a more “holistic” approach, playing with the bug itself and talking to some of the users who are experiencing the issue.

Scene 3 – Understanding the issue

We follow the “holistic” Team B as they interview a couple of communication technicians who were the first to report the bug. They use a heuristic-guided method to question them and gather information about the system and the problem at hand.

The technicians (a couple of Army Privates in charge of processing coded messages from the Middle East) explained that it all started 2 days ago when they got into their post to start their shift and one of the guys was not able to log into his communication program, he tried everything he could think of but nothing worked.

The strange thing is that a couple of hours later, when he got back from a smoke break he tried once again, and this time he was able to log in and work with it.

Then later in the day it started happening again, but this time it happened to both of them simultaneously. And when they reported the issue to their IT, it appeared that at least half the base was experiencing the same problem. Many operators and officials where not able to log into their communication platform!

Some users are not been able to connect at all, while others manage to connect sometimes.

They tried login in and out of their computers, modifying hardware, everything! Sometimes it helped for a number of minutes but then it stopped working once again. In fact, nothing makes sense to them.

Our testers ask to see one of these machines where things work intermediately. They play with it a little bit and install a service that will let them get more information. Then they ask one of the Privates to start working normally while they observe his operations from behind.

After a couple of minutes of work the error message pops up once again! But this time our testers are prepared and they move quickly to take charge of the machine.

They open the application they installed, that was gathering information on the system as the user worked, and start reviewing the activity log. After about 3 minutes one of our testers points at a specific line on the screen while the others nodes her head in agreement and smiles.

They run a couple of commands on the machine, then ask the user to try to run the program once again and “PRESTO!” The error message disappeared, while the IT guys who had been in the room all this time look in awe and disbelieve.

Scene 4 – Finding a solution

In the meantime Team A, the detailed analysis team, is gathering information about the latest upgrades installed on the system and the base. They check all maintenance logs and updates, as well as client software maintenance that have taken place in the last 72 hours.

The IT clerks report that nothing out of the ordinary was done in the last 3 days, or in the last 3 weeks for that matter. There were no major upgrades or any kind maintenance done.

As the testers start going over the routine upgrade list and cross-reference the actual components (libraries and executables) they get a call from the "holistic" team informing them about their breakthrough. The "analysis" testers ask for the list of processes they got from the internal monitoring agent and also to come and join them in the IT offices.

Based on the new information they modify their search and start analyzing all the data available once again.

The "holistic" team comes and joins in on the search. After a number of minutes one of the testers finds something that looks interesting and calls to the rest of the team to come and see.

She is not sure about the exact details but she remembers reading something about a component on the list some months or even years ago. With the help of another tester they start "googling" this information, searching for blogs, articles or release notes on the specific component, and within a couple of minutes one of them shouts "EUREKA! I found the problem and the solution!!!"

They download a file, give it to the IT technicians in the room and ask them to go to all the machines that are suffering the issue and replace the old file with the new one.

Scene 5 – Tying all the knocks together

The team gathers in the General's office and quickly goes over their findings.

They explain that the source of the problem is a file that was been feed by the translating system that included some uncommon (but legal) characters in an old middle eastern language, and was causing some of the computers to get stuck while trying to load it to memory.

The problem was even worst because this file, that had been fed into the system 3 days ago, was been periodically loaded into the local memory by the program in order to make the system work faster, just like the rest of the files that were received in the last 2 weeks.

This bug had already been fixed a number of years ago by the company developing the Operating System, it was a small bug in one of the language libraries. But since the computers used by his team and international operatives were disconnected from the Internet due to security measures, this specific updated had not been installed in any of the old computers as part of the regular software updates.

With this information in hand, our testing team simply provided the IT clerks with a new file that would immediately fix the problem.

BUG SOLVED!

Maybe not as good as Magnum PI, but what do you think?

Do you think we could persuade FOX to make this their next TV Drama?
I guess not but it was fun to write it anyway... 😊

I am also not sure this is what would make any of my kids turn and tell me that they want to be a Tester when they grow up, but it surely makes our work look as a lot more interesting (and less grey!).

BTW, many of the facts on this story came from actual bugs I found and was able to reproduce during my years working as a tester (not the part of the secret base in the mountains, in case you were wondering...).

Do you have other challenges or ideas for our **T-G-I** team? Feel free to share them!



Joel Montvelisky is a tester and test manager with over 14 years of experience in the field.

He's worked in companies ranging from small Internet Start-Ups and all the way to large multinational corporations, including Mercury Interactive (currently HP Software) where he managed the QA for TestDirector/Quality Center, QTP, WinRunner, and additional products in the Testing Area.

Today Joel is the Solution and Methodology Architect at PractiTest, a new Lightweight Enterprise Test Management Platform.

He also imparts short training and consulting sessions, and is one of the chief editors of ThinkTesting - a Hebrew Testing Magazine.

Joel publishes a blog under - <http://qablog.practitest.com> and regularly tweets as [joelmonte](#)

[Back To Index](#) 

Teach-Testing →

An Ambitious Campaign by Tea-time with Testers



Click here to Cast Your Vote



crossword

Find the hidden word!

by





Call for Articles !

Have you got something to say?

yes, we are listening you...!!!

"Tea-time with Testers" firmly believes that one of the best ways to improve upon software testing is to listen to the lessons learned by others and their experiences too.

So, if you have an interesting story that you'd like to share with the world, contact us at teatimewithtesters@gmail.com.

Submit your articles, stories, thoughts around software testing.

now its your chance to be heard...!

Click [HERE](#) to read our Article Submission FAQs !

T ' Talks



T. Ashok exclusively on software testing

Purity. Quality. Cleanliness Criteria.

What do we do when we want to ascertain the purity of a material? Firstly we clarify what the expected purity is by identifying the properties that it should satisfy. Then we check if the given material does indeed satisfy these properties.

Properties? Remember studying about physical and chemical properties in your school days? Physical property is one which does not alter the material while chemical property is one that can alter the material. Some examples of physical properties are density, viscosity, malleability, conductivity while flammability, enthalpy (of formation), toxicity are examples of chemical properties.

Interesting observation - To assess the 'Purity' of a material, we have to check the various '...ity'(s) i.e. Purity is really a degree of the how well the properties have been met. And properties are those that can be observed when evaluating the behavior of the material.

Let us shift gears to software now. One of the key objectives during the development of software is to assess the 'Quality'. Quality is about how well a system/software meets the expectations, similar to the notion of purity. It is no coincidence that they sound similar too. So by extending the previous logic, if 'Purity' is about satisfying properties i.e. collection of '...ity'(s), then (software) 'Quality' should also be about satisfying properties.

Another interesting observation- Quality is also a collection of '...ity'(s): Functionality, Security, Reliability, Usability etc. Cute eh? Let us extend the logic now. So what may this mean? The 'ity(s)' of software are really properties of software that we need to check for. So the act of evaluating the quality of software is really about checking how well properties have been met. Akin to the categorization of material world properties into physical and chemical, in the software world, we categorize these into functional and non-functional. The former is about the correctness of transformation (of input to output) while the latter is about attributes during the process of transformation.

So where am I leading you to? To a simple understanding that the act of testing is about clarifying/ setting-up the expectations by identifying properties and their intended value and then assessing them. Most often the act of identifying the properties and their intended value is very useful, as they result in interesting questions. And then choosing the appropriate tests and then applying suitable techniques to design test cases to assess the properties. ISO 9126 is an useful standard that lists these properties.

This leads us to a scientific approach to understanding expectations rather than only rely on past experience or resort to guess work or gut feel. An approach where understanding of expectations of entity under test is about identifying the properties that matter and then their intended values.

This is termed as a 'Cleanliness Criteria' in Hypothesis Based Testing (HBT) - A set of objective (i.e. testable) criteria that a system should satisfy.

So the next time you test software, identify the cleanliness criteria first i.e. list of properties that it should satisfy and the test purposefully.

It is great fun as it throws up more questions than answers.

Enjoy!



[Back To Index](#) 



T Ashok is the Founder & CEO of STAG Software Private Limited.

Passionate about excellence, his mission is to invent technologies to deliver "clean software".

He can be reached at ash@stagsoftware.com .





OUR PARTNERS

Quality Testing



Quality Testing is a leading social network and resource center for Software Testing Community in the world, since April 2008. QT provides a simple web platform which addresses all the necessities of today's Software Quality beginners, professionals, experts and a diversified portal powered by Forums, Blogs, Groups, Job Search, Videos, Events, News, and Photos.

Quality Testing also provides daily Polls and sample tests for certification exams, to make tester to think, practice and get appropriate aid.



Mobile QA Zone

Mobile QA Zone is a first professional Network exclusively for Mobile and Tablets apps testing.

Looking at the scope and future of mobile apps, Mobiles, Smartphones and even Tablets, Mobile QA Zone has been emerging as a Next generation software testing community for all QA Professionals. The community focuses on testing of mobile apps on Android, iPhone, RIM (Blackberry), BREW, Symbian and other mobile platforms.

On Mobile QA Zone you can share your knowledge via blog posts, Forums, Groups, Videos, Notes and so on.



Testing PUZZLES

by Sebi



Claim your **Smart Tester of The Month Award**. Send us an answer for the Puzzle and Crossword bellow b4 15th June 2012 & grab your Title.

Send -> teatimewithtesters@gmail.com with Subject: Testing Puzzle

**Exciting
PRIZE for 1st
three
WINNERS***

NOTE : S.T.O.M. contest comprises of Testing Puzzle + Crossword. To claim their prize, participants should to send answers both for puzzle and crossword.

Puzzle “Solve This Puzzle”



one hundred twelve
one hundred seventeen
One hundred and twenty two
One hundred and twenty two
One hundred and eight
One hundred and one
Thirty two
One hundred and five
One hundred and fifteen
Thirty two
One hundred and fifteen
One hundred and eleven
One hundred and eight
One hundred and eighteen
One hundred and one
One hundred
Thirty three
Thirty two
Ninty nine
One hundred and eleven

Biography



Blindu Eusebiu (a.k.a. Sebi) is a tester for more than 5 years. He is currently hosting European Weekend Testing.

He considers himself a context-driven follower and he is a fan of exploratory testing.

He tweets as @testalways.

You can find some interactive testing puzzles on his website www.testalways.com





TESTING CROSSWORD



1		2		3		4	
	■		■		■		■
5		6			■	7	
	■		■	8			■
	■	9			■	10	
11			■	12			■
	■		■	■	■		■
13							

Horizontal:

1. Any data developed for use in tests (8)
5. It is a technique used during the software development life cycle for software component and code error detection prior to application execution, in short form (2)
6. The first executable statement within a component, in short form (2)
7. _____ tests the ability of the software to prevent unauthorized access to the resources and data, in short form (2)
8. It is often the final step before rolling out the application (3)
9. It is a black-box GUI test automation tool. Its first 3 words (3)
10. Testing of individual software components, in short form (2)
11. Verifying a product is accessible to the people having disabilities, in short form (2)
12. You do it when bug needs an immediate fix (3)
13. A device that duplicates the functions of one system using a different system, so that the second system behaves like the first system (8)

Vertical:

2. It is a Mobile Test Automation tool for Android, iPhone, Blackberry, Symbian & WindowsPhone 7 (7)
3. It is an open-source web site stress test tool (6)
4. It is a provider of SaaS-based Test Management service for managing and executing manual and automated software tests and for reporting defects (7)

Answers for last month's Crossword:

J	I	T	S	A	X	E	N	A
A		E		G		M		
M	U	L	T	I		B	U	G
E		L		L		U		
S	A	U	C	E	P	N	L	P
B		R		R		I		A
A		I	E	R		T		I
C		U		O	L	V	E	R
H	T	M	L	R		I		T

Answer for last Puzzle:

http://www.testalways.com/boardwalk/boardwalk_solution.txt



We appreciate that you

"LIKE" US!



Join us on Facebook.

You are just a CLICK AWAY



Every Tester

who reads **Tea-time with Testers**,

Recommends it to friends and
colleagues .

What About You ?

Our Testimonials

'Tea-time with Testers' is such a creative name. I loved it and also the articles in your magazine.

Keep it up.

- Mariana Desouza

Came to know about TTWT via some QA forums. I wonder how didn't I notice it before.

I have downloaded your all issues and I'm surely going read them.

Great job. Thanks

- Neilson

I'm really happy to have such great site where I can see up to date articles by experts.

Thanks to Kumar brothers :)

Regards,
Syed

in ne>xt issue

articles by -



IT'S
ALWAYS
TEA-TIME

Jerry Weinberg

T Ashok

Joel Montvelisky

Janet Gregory

Anurag Khode

Bernice Ruhland

Samarjeet Mohanti

our family

Founder & Editor:

Lalitkumar Bhamare (Mumbai, India)

Pratikkumar Patel (Mumbai, India)



Lalitkumar



Pratikkumar

Contribution and Guidance:

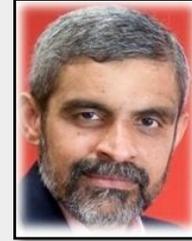
Jerry Weinberg (U.S.A.)

T Ashok (India)

Joel Montvelisky (Israel)



Jerry



T Ashok



Joel

Editorial | Magazine Design | Logo Design | Web Design:

Lalitkumar Bhamare

Cover Page Image- Roses and Teacups

Core Team:

Anurag Khode (Nagpur, India)

Dr.Meeta Prakash (Bangalore, India)



Anurag



Dr. Meeta Prakash

Testing Puzzle & Online Collaboration:

Eusebiu Blindu (Brno , Czech Republic)

Shweta Daiv (Mumbai, India)



Eusebiu



Shweta

Tech -Team:

Chris Philip (Mumbai, India)

Romil Gupta (Pune, India)

Kiran kumar (Mumbai, India)



Kiran Kumar



Chris



Romil

*// Karmanye vadhikaraste ma phaleshu kadachna |
Karmaphalehtur bhurma te sangostvakarmani //*

To get **FREE** copy ,
Subscribe to our group at

Google™

Join our community on



Follow us on



www.teatimewithtesters.com

