

# Tea-time with Testers

SEPTEMBER 2012 | YEAR 2 ISSUE VIII

Jerry Weinberg

The Fish-eye lens

Mike Talks

That Barton Fink Feeling...

Joe Demeyer

The Credible Tester

T Ashok

Behave Yourself !

Joel Montvelisky

5 Simple Tips to keep Testing Simple

Samarjeet Mohanty

Software Performance Engineering



# 11<sup>th</sup> International Conference on Software QA and Testing on Embedded Systems

**MORE  
TESTING**

**MORE  
QUALITY**

**MORE  
EMBEDDED**

**SOFTWARE**

This year we offer a special program: **3 Keynotes, 2 Tutorials, 8 Tracks**, from renowned companies such as: **Intel, Samsung, Phillips, Adobe, Oracle and Quality Perspectives.**

QA&TEST, the international Conference on Software QA and Testing on Embedded Systems, will be held on 17, 18 and 19 October in Bilbao, Spain, and gives you an excellent opportunity to increase your business network and establish contact with others professionals of the industry.

The main objective of QA&TEST is present the last technological developments in Software Testing and Quality Assurance, and showcase successful best practice to reduce costs and give companies a lead in global competition.

[www.qatest.org](http://www.qatest.org)

---

**Special offer 15% discount using the code TEATIME**

---

Organiser



Sponsor



October 17 · 18 · 19

**2012**

**Bilbao · Spain**



There are more than 40 leading  
organisations that host  
**Tea-time with Testers**  
on their knowledge portal.

**WHAT ABOUT YOURS?**

**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](mailto: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

## The bazaar of test cases

Since a couple of months, I have shifted to this new city. Though this city is new for me, I have many old friends living here already.

One fine evening, I met one of my old friends in a coffee shop. Both of us were quite happy that we met after a long time. After talking about this and that, we happened to talk about our professions and I learned that he also worked as a tester in one reputed firm.

We ordered two cups of hot Earl Grey and continued to talk. It's quite natural that when two testers meet over a tea, they'll discuss testing and we were no exception. My friend looked quite excited and he told me that very soon he was going to get some reward points from his manager. It naturally made me curious and when asked he said that he was getting that reward for earning more business for the firm.

"More business for the firm? How did you do that?" I asked. He said that he was quite a champ in writing more and more test cases and that he managed to write 400 test cases in addition to what was actually needed. By showing that inflated test case count, his firm showcased the need of additional tester. "And since delivery was critical, client had no other option than approving one more guy that gave us additional billing", said my friend.

I was speechless but assuming that I was impressed, he also told about his high performance rating since he managed to write *maximum* test cases and reported *maximum* defects. I did not fail to notice his emphasis on *maximum* though.

For me, it was hard to continue that discussion and I asked him to excuse me for I wanted to visit the bazaar.

He saw that I was little disturbed and gave me a free advice, "Don't worry Lalit! It's not that difficult. You just need to learn how to manage this bazaar of test cases."

I smiled and mumbled, "Yes, the *bazaar* of test cases".

*And, as long as people continue to count test cases without context, this bazaar is bound to flourish.*

Enjoy Reading !

Yours Sincerely,



**Lalitkumar Bhamare**

editor@teatimewithtesters.com



# QuickLook



Testing Puzzles  
by Sebi



Crossword  
by



## Editorial

### What's making News?

### Tea & Testing with Jerry Weinberg

### Speaking Tester's Mind

That Barton Fink Feeling - 18

The Credible Tester- 24

### In the School of Testing

Software Performance Engineering- 32

5 Simple Tips to Keep Testing Simple-37

### T ' Talks

Behave Yourself! 43

### Testing Puzzle – S.T.O.M. Contest

### Our Testimonials

### Family de Tea-time with Testers



# What's making News?

- find out the latest happenings in the technology world

## TuniTeam launches 'Beta Programm-2' and additional features in myTestingBank.

**TuniTeam** ([www.tuniteam.com](http://www.tuniteam.com)) has launched **myTesting Bank**, the test management tool for managing the QA / Validation activities and is offering free trials of this intuitive new test management tool with some new enhanced features this time.

### New Features:

Latest feature that is added into myTestingBank (aka myTB) is effective Bug tracking and Test Automation:

myTB now comes with strong bug tracking system which gives its user the facility to :

- Create automatic ticket during a test run if test is failed
- Follow reported issue from a test run to another test run
- Statistics about reported issues
- Define the road map for development team



## Beta Programme Phase 2:

After the first phase of Beta Programme, TuniTeam has now launched the phase two.

It includes the download of myTB binary and installation on local machine. The deployment of myTB on user's host will allow the free use of myTB at per user's wish. This will be behind user's firewall ensuring the secure use thereby.

## Special offers

- **On line use:** With this offer myTB will be accessible via a web interface that facilitates communication and collaboration between decentralized test teams.
- **Deploy:** You can download myTB and deploy on local host. Just order to get a License Key which is then used with the software "myTB" that you download.
- **White label:** With this offer you can customize the app and also get the right to rename the app with "Your company name" by changing the Logo. This offer will help small structures to use myTB as if it's their own testing tool.

**\*\*TuniTeam will provide full support for any kind of help needed.**

## About TuniTeam:

TuniTeam is a service company specializing in Mobile software development and validation activities. It is operational since 2009 and growing up mainly in European market.



Mobile Software

Web Design





# Want to connect with right audience?



How would you like to reach over **19,000** 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](mailto:sales@teatimewithtesters.com)

Announcing...

# Smart Tester of the Month Award !!!

## Prize winners

### *Testing Crossword:*

First Prize:

Name: Rajith Sudhakaran

Second Prize:

Name: Nirav Sampat

### *Testing Puzzle:*

Name : Venkatakanthi Datla

**Congratulations !**

**Note for Prize Winners: We will inform you about your prize details via e-mail.**

**Names of other winners who had sent us correct answers will be declared on our Facebook Page.**



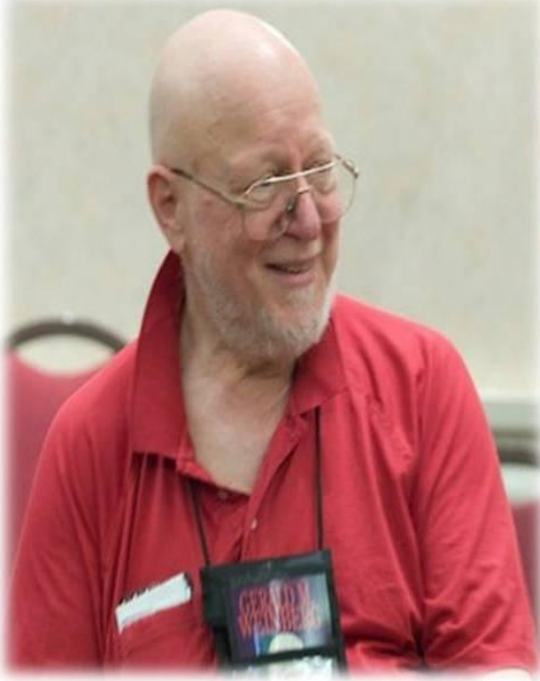
Don't miss the action...



join us on FACEBOOK !



# Tea & Testing



# with Jerry Weinberg

## The Fish-Eye Lens (Part 4)

### Satir's Three Universal Questions

How do I get all this information about context to begin with, so I can strip out all these biases? One way is to use Virginia Satir's Three Universal Questions, modified slightly to take into account the entire organization and all its people:

- How did they get here? (Past)
- How do they feel about being here? (Present)
- What would they like to have happen? (Future)

If you read *The Secrets of Consulting*, you'll recognize that "How did they get here?" is the general question underlying Boulding's Backward Basis:

Things are the way they are because they got that way.





You'll also recognize "How do they feel about being here?" as embodied in Brown's Brilliant Bequest:

Words are often useful, but it always pays to listen to the music (especially your own internal music). "What would they like to have happen?" is often harder to get at, because each organization has "official" aspirations, but these seldom give an accurate picture of what people are really hoping will happen.

Here are some examples of Big Picture questions that help me see the larger context, and which once again make an enormous difference in how I approach an assignment.

### **How did they get here?**

- Can I get a history of the organization? Is it an official history, and if so, how does it differ from the unofficial oral history that I hear from people?
- What's their past experience with consultants? What was the process by which they chose me? Is that typical in this organization?
- How long do people typically stay in this organization? How long do they stay in the same jobs? Where did they come from before they got here?
- Is this a profit-making company, and if so, what is its history of profitability? If it's a non-profit, what is the vision that created them?

### **How do they feel about being here?**

I can get the best impression of this part of the context by simply walking around and meeting people, being friendly and inquisitive about their work and the things I see around their workplaces. Of course, if the people who invited me discourage me from walking around and doing this, that tells me more than a thousand pictures.

- Are the people I meet eager? Happy? Friendly?
- Do they like their surroundings? What kind of evidence do I find in the way they organize their workspaces? Do they feel free to show evidence of their personal life, and if so, what do they show and not show?
- What evidence do they show of professionalism and how they measure it? Do I see certificates as evidence of training, and if so, what training are they showing off?

Do they keep things tidy? Do they keep things so tidy it looks as if they are afraid of citations for disorderly workplace?

- Are people puzzled about what's expected of them in this organization? Do their titles have meaning? Do they use their titles, and if so, how do they use them? As badges of honor? As attempts to force respect?
- Are they sure of themselves? Do they feel empowered to make decisions and be backed up by the organization?

- Is this the right mood for succeeding in this job? If not, what would have to happen to get them in the right mood?

### **What would they like to have happen?**

- Why did they look for a consultant? For reassurance about what they've already decided? The expertise? As a scapegoat? To hold their hands?
- What will success look like, to them?
- How long do they want me here?
- How were people chosen to work with me on this assignment? Did they feel they had a choice, and if so, why did they choose? Why were they chosen? How many people wanted this assignment but were not chosen? Is that a typical choosing method?
- What will success look like, to the organization? Can they give examples of previous situations that were successful? Unsuccessful? Can they explain why they were regarded that way?
- What do they fear? Do they seem to be hiding something from me? How do they react when I ask about that?

### **The Fishy-Eye Lens**

I can overcome many of the biases that prevent my clients from seeing their own environment, but ultimately, I need to take my own pictures of the context. As I search for context information, I always have my Fish-Eye Lens hooked up with my Fishy-Eye Lens, that part of me that looks for examples of The Incongruence Insight:

When words and music don't go together, they point to a missing element.

Since I wrote *The Secrets of Consulting*, I've been gathering juicy examples of incongruent messages about the context in an organization.

- A major technology company simultaneously rolled out two new programs on the same day: (1) a random drug testing program, and (2) an "Individual Dignity Enhancement" program.
- A telecommunications company started a program of "outstanding contribution recognition." Each month, the winners were photographed receiving a plaque from a senior executive, and the photos were prominently displayed at the cafeteria entrance. In the text below each photograph, the outstanding contributor's name was given in plain 12-point text; the executive's was given in 18-point bold.
- An insurance company purchased laptop computers for employees to use while traveling. Fearing the laptops might get stolen, the managers came up with a clever solution: permanently bolt the laptop computers to the employees' desks.
- A consulting company decided they needed to raise employee morale. They conducted an "anonymous survey," but insisted that every employee write an identifying number on the survey form, "for control purposes."



- A freight company reorganized to define roles and clarify goals. Management decided to communicate the changes by ordering each department to build floats for a "Quality Parade."
- A manager at a telecommunications company wanted to reinforce the "team" concept in his department. He held a meeting to tell the assembled "team" that henceforth he will carry a baseball bat with him at all times and each team member will carry a baseball while at work. Some team members found a way to hang the baseball around their necks so they don't have to carry it. Others fantasized about wrestling the bat away from the manager and using it.
- A related case was the manager who kept a bullwhip on the wall in his office. He said it was a souvenir from a leadership course he took, and it reminded him of the major lesson of that course—individual initiative.
- A company decided that instead of raises it will give bonuses if five of seven company goals are met. At the end of the year, the employees are informed that they have met only four of seven goals, so no bonuses. One of the goals they missed was "employee morale."

I characterize these examples as right-left messages. When right-hand messages (words) are contradicted by left-hand messages (actions) guess which one will be heard— which ones make up the real context of the organization.

There's a meta-context, too, for most of these managers couldn't even see the incongruity when it was pointed out to them. That really tells you something about the managerial context of these organizations.



Or perhaps they were lying. Perhaps the incongruity was intentional. In that case, I can apply the Inverse Gilded Rule:

If something's faked, it must need fixing.

To seek out right-left messages quickly, I ask about things like the organization's vision statements, company newsletters and press releases. Then I check for behavior that aligns with these public pronouncements—or doesn't. Then I ask about it.

Either way, my Fishy-Eye Lens has revealed important information about the context in which I'm going to be working.

***to be continued in next issue...***



# 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](mailto:hardpretzel@earthlink.net) or on twitter @JerryWeinberg

**More Secrets of Consulting** is another book by Jerry after his world famous book **Secrets of Consulting**.

This book throws light on many aspects, ways and tools that consultant needs.

“Ultimately, what you will discover as you read this book is that the tools to use are an exceptionally well tuned common sense, a focus on street smarts, a little bit of technical knowledge, and a whole lot of discernment”, says **Michael Larsen**.

**More Secrets** is definitely useful not only to consultants but to anyone for building up his/her own character by implementation of the tools mentioned in day to day life.

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

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

## MORE SECRETS OF CONSULTING



Gerald M. Weinberg

**TTWT Rating:** ★★★★★



A photograph of a green, conical pendulum bob hanging from a thin wire. The bob is positioned over a surface of light-colored sand. In the sand, there is a large, circular, swirling pattern that resembles a sand mandala or a ripple in water. The text "Speaking Tester's Mind" is overlaid on the image in a large, blue, serif font with a white outline.

# Speaking Tester's Mind

- straight from the author's desk



# That Barton Fink feeling...

- Mike Talks

Barton Fink is a classic 1991 movie from the Coen Brothers starring John Turturro and John Goodman. The story is a compelling one with enough parallels to the world of testing that we can all appreciate.

*Barton Fink is a successful play-writer in New York, who is lured by a Hollywood company to come and write movie scripts. He's been headhunted by Capitol Pictures because of his success on Broadway, and they've bought him in on a 5 year contract. In their eyes he can do no wrong, and they're after him to bring some of his magic to their celluloid.*

*Barton tries to adapt to the new format of the moving picture, but finds himself stuck with writers block. He goes around other writers to share ideas and learn how they handle it.*

*Eventually, struck by inspiration and fired up, he gets to work on a masterpiece of scripting ... but the studio bosses hate it, so angered and furious that they feel that Barton Fink is just not getting what they're after.*

*The film ends with Barton subject to a vengeful retaliation by Capitol Pictures which reminded me of the Greek punishment of Sisyphus (who was punished to push the same boulder up a hill for eternity). They will keep him under contract for the next five years, and they will pay him and he can continue to write. But they will NEVER produce one of these scripts. Writing, Barton Fink's very passion, will be something that will only bring him misery.*

*That Barton Fink feeling* is something a lot of testers go through. I was lucky to attend the KWST2, a New Zealand peer conference on ethics that had James Bach as content owner. Around the conference there seemed to be a story repeated again and again of some companies who would bring in testers, particularly "experienced" testers, because they knew they had issues with the way their organisation approached testing.

The problem would then arise that these experienced testers once hired would be under constant pressure from management – *"if you're an experienced tester, where are your scripts, where are your metrics, where are your test case counts"*.

This is the pain a lot of testers are facing. Managers who've been on a project before and think they know testing – *"testers at my last company gave me Quality Centre graphs, so where are yours?"* Testing over the 15 years I've been involved is evolving and moving on – much like the Agile movement, there's a drive that whilst scripting has its value, then the execution of that script had more value. We're looking at our actions and saying "whilst these things have value, these other things have MORE value". But alas management isn't moving along with us.

This is the Barton Fink pain – you are brought in for your experience. *Expressly for your experience*. However management can seek so much control over how you work, that you're not given the freedom to really apply that experience. This is a frustrating experience for the tester, who feels he has to serve two masters – his managers and his professional responsibility to do a good job.

The irony is by going through the motions of compliance with your managers requests you know you are in actual fact doing a second rate job on your primary goal – which is testing the product.

Part of the problem is that organisations often recognise that they have a shortfall in an area, and will even bring in talent to address that area, but will then be so rigid that these people won't be able to make any changes.

One of the issues as highlighted above is that although you bring within your new project all your experience and your scars, they are just that, YOURS. There are two dynamics at work here – your experience and the experience of your team/project/organisation. You are ready to take the next step towards a more workable efficient testing program, but your group is not.

Group learning is definitely frustrating and much slower than individual lessons. But it has to happen at a rate everyone is comfortable with. But it's not all bad news – groups can respond well to leadership and mentorship to learn new things. Are you up to the task? *Here are some things to think about and try out, if you're finding yourself familiar with the Barton Fink feeling ...*

### **Be a subversive not a radical**

As John the Baptist will testify, there is a risk in being a herald of news before others are ready to hear it (*he ended up losing his head because of it*).

The fact is that people who come over too radical and extreme to what's the expected norm are most often ignored as lunatics or not even hired to begin with! Sometimes to get in amongst the sheep you need to dress yourself up as one. That might mean accepting short term someone else's testing agenda, whilst trying to subtly push your own over months, which you know will address testing far better.



It means sometimes giving the project manager some their metrics, whilst also telling them other ways, *better way*, to get the pulse of testing, and show as much as possible why these are more effective. Try wherever possible to use evidence from the projects your team has had experience of to highlight new ways of working.

Remember when I said before, how do groups learn at their own pace? This is the case here. You might have experienced situations and ways of working countless times before in your experience. But for some on the project they don't have the experience – and so the project will have to actually make a few mistakes before you can effectively lead and say “that’s why we should go in this direction”.

Never underestimate the human capability to learn the wrong lessons though. On a previous project we had software delivered to us which fell apart with some very basic tests – we had major critical defects to resolve. *Could what have happened been any simpler?* However, for some reason my project manager was convinced that the reason we were having problems was our testing scripts needed to be more detailed (*and to stop testing and start scripting*), and I could not talk her out of this notion I'm ashamed to say. Whereas I was felt that the fact we'd managed to find and locate so many critical defects said we were doing something right.

### **Transformation isn't a one way street**

I was in a talk by Rob O'Connell from Assurity last year when he said some very poignant words about transformation – it's very easy to say “to get better you all need to do something different”. There is a definite mentality that “I'm doing everything right, it's just all you who need to change”.

Everyone is involved in change, it can never be a single person's agenda – not if you want the changes to last (*otherwise people have a habit of trying to undermine so they can revert to form*). Change has to be a group consensus and vision. Move too fast, too quickly and any issue will because naysayers will be quick enough to denounce any new method of being “*worse than our previous practice*”.

### **Why it's worth the hard yards**

It's a good question – if you feel you're making no progress, *should you stay or look for another organisation elsewhere?*

It's nice to think there's a project out there which has everything right, where your own ideals and the project ideals align nicely. But every project where that's a case will have had to go through some form of transformation – and transformation is always painful.

Maybe this project is your masterpiece – the one you will transform and show leadership doing it?

It's important to remember effective and efficient test teams aren't just grown, you don't just follow a recipe to make them. They have to be forged by the hard work of those involved – teamwork, collaboration, concession, and conflict. These are all emotionally exhausting areas where all parties are giving as well as taking.

## Above all patience

I was talking with a test manager from another company and was aware we were talking about the respective management teams we work with much like a pet we're working on house training.

That might seem a bit belittling of our managers, but it's probably in the right vein. Our motivation to train up our groups is not out of being right, but out of love – because we know it's for the best of our organizations , and we have the knowledge and experience that will give us success.

Much like a new puppy is something you feel absolutely committed to, they also have a habit of making a lot of mess and chewing a lot of things you feel are precious up. But it's not through anger, but through patience, nurture and occasional discipline you train that puppy up to be a faithful companion ...



**Mike Talks** is a test manager in Wellington, and self-confessed movie nut. Always on the lookout to be educated as much as to educate, he is also the author of the LeanPub book 'The Software Minefield'.  
<http://leanpub.com/TheSoftwareMinefield>

[Back To Index](#) 



# Looking for RIGHT job in Software Testing?

visit **Qualityjobsportal.com**

*after all, it's your career we are talking about !*



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](mailto: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](mailto: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.





# *The Credible Tester*

*- Part 2*

**- By Joe DeMeyer**

## **Meeting the Team (sidebar/anecdote)**

In one project, my first team meeting occurred when I could not be in the office. I dialed into the meeting and was introduced as the Test Lead. Occasionally, participating in a meeting by phone can be challenging however attending the first meeting of all team members on a new project was interesting. But I think it had an unintended positive affect.

The PM started going through the project schedule and mentioned a few meetings would occur to provide details. When asked who wanted to attend, several people responded and I asked to attend also. There was a pause and a clarifying question. "Yes", I replied, "I would like to attend that meeting, please." When more meetings were discussed, clarifying questions became invitations. When finally I met other people on the project, they knew me as the tester who wanted to participate in many meetings.

I want to participate in the project at the same level as most other team members – at least at the start of the project.

## **THE VISIBLE TESTER**

Some team members interpret a tester's task as executing tests. They neglect to consider research and planning required for creating those tests. But there is an opportunity in this. Exceed their expectations: execute a test during the first week of your project (your first day if you are familiar with the product). The test can be conformational (the system behaves in this manner today), or educational (the system

behaves in this manner during these conditions – which happen to resemble some of the requirements of the project). Regardless of the test, share what you learn! Table 2 has some suggestions for tests you can execute at the start of a project.

Table 2. Run These Tests Now

Developer Concerns	Meet informally with developers to discuss their concerns about the requirements, technology, or project. Create a test or tests for these concerns and execute them.
Project Requirements	<p>The requirements are a great place to start your testing:</p> <ul style="list-style-type: none"><li>Are requirements complete?</li><li>Is there more than one interpretation?</li><li>Are requirements testable?</li></ul> <p>As a team, think through each requirement in a critical manner to generate test ideas, improve testability, and to execute tests in your mind.</p> <ul style="list-style-type: none"><li>What steps are required to set up for evaluating this requirement?</li><li>How can I make this more testable?</li><li>How do I know when I have a result?</li><li>How do I know the result is expected?</li></ul> <p>Meet informally with the requirements' author to discuss your interpretations and seek to clarify requirements where possible. Also see CONTINUE TO PARTICIPATE below.</p>
Project Manager/Stakeholder Concerns	Be alert for any concern expressed by a Project Manager or Stakeholder. Create a test to evaluate that concern and share the results with them!

During the project's first few weeks, some team members become familiar with my requests for information or meetings to discuss the existing system. There are many responses. "You won't have to know that" is a time worn phrase (you might wonder aloud if they need it to do their work), or "You'll have to test this function like this", and "You'll have to know this complex business process". These are opportunities to engage them in conversations on both the system and their understanding of testing.

As you understand more about the system, talk about tests you can execute that demonstrate the system's behavior.

This builds credibility by showing this person that not only do you understand the system but you know how to demonstrate it works and doesn't work.



## USE YOUR TESTING VOICE

Questions are a natural method for gaining information. However, constant questioning can be burdensome. When you discover some part of a product you want to know more about, make a list of questions. Craft scenarios around those questions so your conversation becomes a discussion rather than an interview.

You can ask questions for two reasons: Ask them when you want to learn, or ask them when you want to confirm. Asking a conformational question helps start and guide a conversation. It provides context. It also provides a platform for you to present solutions or ideas to issues you ask about. Discussing solutions to potential issues and experiences builds credibility in that topic and in you.

## RESEARCH AND SHARE WHAT YOU LEARN

Diving into information about the project and the products is a natural response for a tester new to a project. If you are familiar with the platform or product, start evaluating products and reviewing documentation. If you are new to the platform, ask your Lead Developer and Business Analyst about the products and the proposed changes. This is another opportunity to engage your project team (i.e., build rapport and credibility). Where possible, try to draw analogies between changes to the products and changes that are similar to work you have previously completed.

"This sounds familiar. Does this product interface through the X mode?"

"This sounds familiar. Has the X mode functionality been evaluated using Y methods?"

"A previous project used the X mode to perform that functionality. I might be able to re-use some of those test cases."

While documentation may assist in understanding product changes, a little internet research about the technology could stimulate ideas for testing (develop a skill at finding, absorbing, AND applying new ideas and information to your project). When you find valuable information like this, be sure to share it with both your testing team and your development team. When you share it, drawing a picture or pictures communicates your understanding, what you have learned, and how you applied it to the project.

As a tester, your job is to provide information.

In addition to test results, providing information you learn is also valuable to the project.

## What is Testing?

I can describe testing like active listening – in active listening, you hear information and you describe what you heard in your own words. The speaker and the listener both feel understood. A tester (the listener) learns about a system and describes its behavior based on information generated by their tests.

A tester should be able to describe, in an engaging manner, how those tests confidently provide information so others can make decisions about the system.

The key to credibility is to provide that description early in the project. When everyone understands the tester understands, everyone has more confidence in the tests and the tester.

## BUILD CREDIBILITY AND CONFIDENCE IN OTHERS

In addition to establishing credibility with the project team, you must also establish your credibility with other testers. In a Test Lead role, I recommend sharing your thoughts on the project and its testing challenges often, coaching or mentoring testers on your team, and providing team members with challenging assignments.

A project changes over its course and issues in that project come and go. About once per month, I share what I believe to be important issues in strategy, testing, or the project. There may be others but sharing your concerns helps raise awareness and bring focus to project changes.

Discuss the Test Strategy	What has changed and why What has been added or removed What has been clarified
Discuss Testing Topics	Testing Methods and Strategies Reporting Metrics Automation
Discuss Project Topics	Team Members Project Products

A Test Lead is in a position to provide feedback and to coach or mentor. Helping your testing team become effective testers helps your team, and the individual testing. When you see opportunities for improvement, meet with the tester to discuss them. Share your observations and your suggestions! Similarly, if a tester is unfamiliar with domain or technical topics, share your knowledge or ask to meet with an expert.

If you want a Test Lead role or other types of testing experience, discuss this with your present Test Lead. Ask them if there are opportunities to perform Lead activities, and to guide or mentor you in these activities. One way to learn and practice these skills is on a project. The Test Lead should be able to delegate challenging, low risk tasks without impacting the project. If you are the Test Lead, be sure to inform the Project Manager of this delegation, and help them understand some flexibility is expected.

## CONTINUE TO PARTICIPATE

As the project continues, you will begin test execution. In addition to test results, there are other opportunities to maintain credibility. I recommend participating in requirements, design, and code reviews. Not only are these meetings informative but they can help generate test ideas. However, to make the most of these meetings, ask for and review documentation (requirement specifications, design documents, or code) and be ready with focused and relevant questions that both test and clarify these documents. By asking good clarifying questions, you will start to get invitations to these kinds of meetings often.

### Questions That Test Understanding

At requirement review meetings, a tester can help clarify requirements so they have a single meaning for everyone at the meeting. I recommend asking questions in the form of a test. These kinds of questions help bring out meaning and clarify a requirement for everyone.



In the table below, notice how a small scenario engages everyone in the test, and ends in a clarifying question.

Requirement	Question(s) That Test
The system completes the transaction in 10 minutes.	If the system completes the transaction in 9 minutes or 11 minutes, does it fail to meet this requirement?
The system displays the total cost.	If the total is displayed as 10.0115, does it pass this requirement? 10? \$10.0115?
The Withdrawal transaction on the new system executes the same operation as the Withdrawal transaction on the legacy system.	The legacy system Withdrawal transaction performs poorly under load. Should the new system's performance be the same? When many users withdraw \$20 from multiple locations, a few account balances change by \$21. Should this behavior be maintained?

### Testability Advocacy

Testability takes many forms but I prefer to use it in terms of improving the ability to test a product. When testability issues occur, discuss them with your team. If the product design appears to hinder your ability to evaluate it effectively, discuss your concerns with the developer.

Determine if log files, database queries, or transaction record details (for example, values of variables, values of method arguments, etc.) can be added or used to improve testability. Advocate for the testability of the products!

Design and code reviews are good meetings to learn more about the domain and more about the implementation of the project's products (occasionally, you also learn about changes to requirements since development started). In the design review, evaluate the design for your ability to test the implementation. Ask yourself if there are sufficient logs or user interface cues in the design you can use to effectively evaluate the product's operation (see Testability Advocacy). If not, ask for them.

In the code review, you can listen for audio cues. If there is a lot of discussion around specific parts of the code, you should probably craft a few more tests there. If you read code, you can participate more and suggest methods to improve testability (if you don't read code, I recommend learning). Participate and help prepare the product for testing!

### Become a Credible Tester

Credibility in a tester begins the first day of a project. Use these suggestions to establish and maintain your credibility. Be patient and consistent with your efforts for it requires time and interaction to build credibility. With this credibility, you can ask for more testable products, suggest design enhancements, and get more bugs resolved.

[Back To Index](#)

**Joe DeMeyer** has been working in IT for over 10 years in both development and testing roles.

He enjoys sharing his experience with his peers and the FIRST Lego League team he coaches.

To maintain his skills, Joe designs, builds, and most importantly, tests small payloads and flies them using a weather balloon. Contact Joe at [Dev\\_To\\_Test@yahoo.com](mailto:Dev_To_Test@yahoo.com).



We congratulate **Quality Testing** for crossing  
mile-stone of  
**10000** community members



Do you have any Questions or Feedback on articles that we publish in Tea-time with Testers?

No Problemo! We will publish your Feedback/Comments and also the answers to your Questions that you have for our Authors.

Do write us your Feedback and Questions in below format and send it to [teatimewithtesters@gmail.com](mailto:teatimewithtesters@gmail.com) :

- Your Name
- Your Brief Introduction
- Article Name
- Your Feedback or Questions if any

➤ Your Feedback or Question

Make sure to write **Feedback For < Article Name>** in your subject line.



A photograph of several young students in a classroom, seen from behind, with their hands raised in the air. They are facing a chalkboard that has some faint writing on it. The students are wearing colorful shirts: light blue, red, orange, and green. The entire image is framed by a thick black border.

# In the school of Testing

*for your better learning & sharing experience*

# Software Performance Engineering :



## “TRANSACTION PROCESSING ANALYSIS” – KYC: Part 2

*by Samajeet Mohanty*

This short paper is an extension of my last article on “Workload Analysis”.

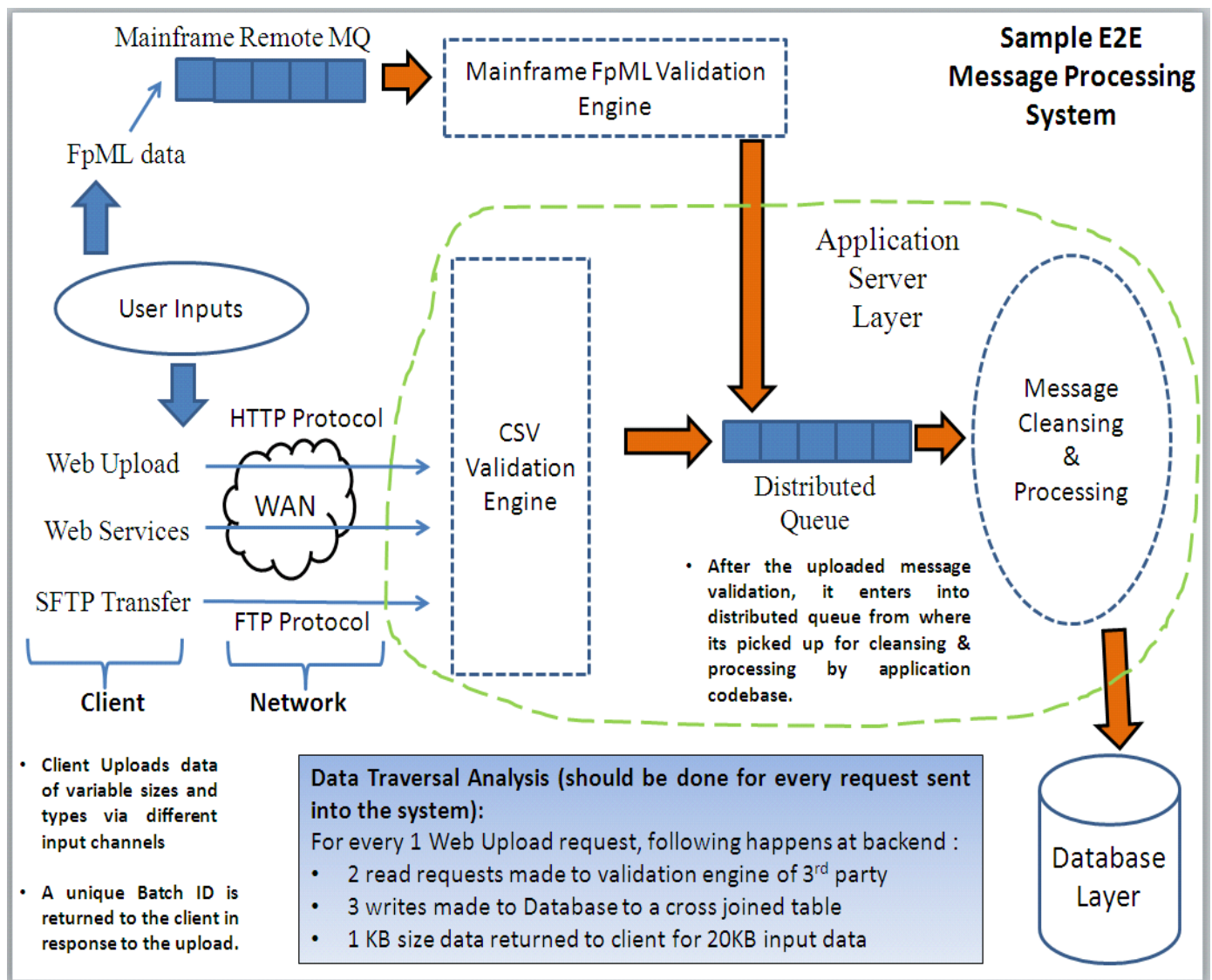
In addition to the workload distribution pattern, it’s vital for a Performance Analyst to understand the topology-flow for all transaction types in-scope for performance testing. By now, you know that a transaction can be broadly classified as Online or Offline type. Online transactions are basically the user interactions with System Under Test (SUT) over the internet or intranet whereas some of the Offline transactions could refer to:

- System background processes
- Scheduled batch job
- BOD / EOD executions
- Any ETL processes

By **topology-flow**, what I mean is to be aware of various infrastructural components a business transaction interacts with from the time it gets into SUT as input till it is delivered as an output to the end user or client.



**Figure 1.0**



Let's look at the depiction example above to have a clear understanding.

- The system is basically a Message Processing Entity
- It accepts different kinds of input data either in TEXT format (CSV file which gets converted to FpML format) or FpML format (XML file) itself
- The input can come from varied sources - Web, FTP, Web

## Services and Mainframe Queue

- Irrespective of input mechanism, all the FpML messages go to the distributed queue residing in the application server for further cleansing and processing.
- Once data is updated into respective tables in the database, a batch Id is returned to user with an acknowledgement message.
- Prior to the test, it should be clearly understood what type and how many calls are made at each infrastructure layer for each type of input request. This data could help in tuning exercise post-test to investigate if either the size and type of data being communicated between say User <- -> App Layer OR App layer <- -> DB layer and so on can be reduced or changed for better performance. This I term as "Data Traversal Analysis" which forms a pretty vital piece during requirements gathering phase.

If the system throughput/performance is not par with SLA, then it becomes very difficult to diagnose the bottleneck at infrastructural level / tier unless we have a good understanding of the transaction navigation through the system. Once we know the component level details, it also becomes pretty clear on where to setup monitoring checkpoints to capture performance measurements for better analysis.

Back To Index

## APPENDIX:

**BOD** → Beginning of Day

**EOD** → End of Day

**ETL** → Extract, Transform and Load

**SUT** → System Under Test



**Samarjeet Mohanty (Samar)** is a, self-proclaimed, practitioner of anything to do with Software Performance Engineering world.

He's quite experienced in Performance Engineering and Testing methodologies of software applications (BFSI) using industry standard tools and techniques.

Apart from being an active participant in technical forum's concerning performance and generic testing QA, Samar likes to interact with creative-minded professionals from all walks of life to better understand their take in the related field.

If interested, connect on:

LinkedIn:

<http://ca.linkedin.com/in/samarjeetm>

Twitter:

<http://twitter.com/SamarjeetM>

# Announcing new batches for Selenium- Basic and Advanced course

## Regular Batch

**Start Date:** 10<sup>th</sup> October 2012

**Course Duration:** 30 Days

**Time:** 9.30 PM IST

**Class Duration:** 1 hour

**Course Fee:** \$ 325 USD

**Syllabus:** [CLICK HERE](#)

REGISTER NOW

## Weekend Batch

(Saturday-Sunday)

**Start Date:** 18<sup>th</sup> October 2012

**Class Duration:** 5 hours/day

**Course Fee:** \$ 325 USD

**Syllabus:** [CLICK HERE](#)

REGISTER NOW

**\$50 USD OFF** for [Quality Testing](#) community members, [Mobile QA Zone](#) community members and [Tea-time with Testers](#) subscribers.

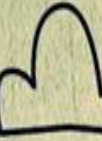
Please Note:

- Only Session No: 1 will be a demo session.
- Recorded videos will be made available.





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**

## 5 simple tips to keep testing simple

### Acknowledgement:

*I'd like to thank my friend [Kobi Halperin](#) for the comment he left on my [last article](#) that prompted me to write this one with some concrete examples of things you can do (as a tester, a test manager, or as any other type of service provider) to make your job simpler and more efficient.*

## A simple approach

I think that a lot of my approach is directly derived from my Agile way of looking at work.

I don't intend to sound like an expert on making things simpler (my wife would even say that I always make simple things complicated!), but I like to think that my approach to managing testing projects comes from experience and boils down to working as straightforward and simply as possible.

Basically I follow a small number of principles to help me organize the work of my team, making sense of the mess caused in today's working environment characterised by the chaotic "right-here right-now" syndrome.

But enough chatter... Here are my 5 simple tips to help you keep your testing simple:

### 1. Divide and conquer

There are almost no real complex tasks, as long as you are willing to look for ways to break them into smaller and simpler components.

Many times I meet QA managers who explain to me how they manage their tests using a small number of (very long) excel sheets or wiki pages. When I ask them why do they work this way they explain to me that they started with small documents that grew over time...

One of the first pieces of advice I give these managers is to divide and conquer. By breaking down their very long and complex testing procedures into smaller and more modular test cases, they can gain flexibility and achieve faster and more accurate coverage.

But this advice is not only good for the size of test cases. If you look into any testing tasks and break it down into smaller testing tasks then you will be able to manage your team more efficiently and provide better visibility to your internal customers.

Let me use a specially hard example that people always use to explain why some tasks cannot be run as modular activities – Load Testing.

If instead of scheduling all the load testing activities towards the end of the test plan (based on the premise that you should tests only once your whole end-to-end system can be tested for the performance and correct response time) you schedule smaller and more focused load testing sessions on specific components during the development process, you will be able to find issues faster and without the need to run complex load scenarios and even more complex analysis operations to find the actual bottlenecks.

My experience shows that by running more focused load sessions you will spend less time testing overall and reach better results than by working based on more traditional – once all is in place – scenarios.





The same goes for any testing activity, break your complex testing operations into smaller tasks and try to run them as soon as possible. This will help you to find the issues fast and help your developers solve them even faster.

## 2. Keep your eye on the “important” ball(s) at all times

Managing any services organization is always about juggling tasks (and yes, testing is a service). You have many things that need to be done and only a small number of resources to do them. No matter how much you try, you will not be able to do everything, specially not at the same time, and never with the level of coverage/thoroughness/attention you intended at first.

Once you understand this it becomes trivial that at any given time you can only focus your attention on a limited number of tasks. So it is better to have a good definition of what are the important tasks that you want to focus on, and make sure all your team is in the same page with you.

A practical idea is to make sure that during your weekly QA meeting (if you don't have a weekly meeting with your team then stop reading this post and set up a recurring meeting NOW, it will be the most valuable thing you do today!) have a whiteboard handy and each week make sure you write down the important tasks or things that should be on everyone's radar.

Give your team the chance to comment on these tasks (or any they think should be there) and make sure you are not missing something important that they know and you don't.

Leave the list in a place that everyone can see – not an email!

You can even start each week's session by going over the list from the previous week and catching up on the events that happened around these tasks.

BTW, regarding the “juggling balls” analogy, an important part of it is to realize that you will not be able to keep all the balls in the air. At one time or another you will reach your limit and some of the other balls just bounce... It's a fact of life and of project management.

## 3. Categorize tasks based on Importance, Complexity and Last Start Date

Directly related to the previous point, is the need to be able to categorize and prioritize your tasks. This is the best way (the only way?) to know what is important and what tasks will be left to fall.

After working with many parameters and weighted algorithms etc, I now use a simple method where each one of my tasks (I am talking about the smaller testing tasks and not the very big tasks your Project Manager always talks about) has 3 individual and unrelated parameters:

I) Importance – What is the business importance or the priority your company gives to this task? Would it be risky if you don't run this test? - High / Medium / Low.

II) Testing Complexity – What is the complexity level of the tests to be run? These parameters incorporate the difficulty to test as well as whether this test can be done by any tester/developer or only by an experienced tester.

III) Last Test Start Date – An important parameter that will reflect when is the latest you can run this test and still be relevant for your project.

These 3 parameters won't give you a calculated index or any way of automatically deducing what tasks should you be doing or when. I personally don't believe in these algorithms mainly because reality is constantly shifting in our type of projects, but they will help you to sort and filter your task backlog and make sure you always know what to focus at what time.

Remember that as a Manager, the task of managing cannot be delegated to any algorithm 😊

#### 4. Learn to say NO



Wow, this is a hard one! Maybe the hardest one in the list. You need to learn how to say to No, and also when to inform your customers that you cannot do something that you previously said you would do.

As I stated before, the reality of our projects is that they are in constant change (I remember someone once saying that the only stable part of our projects is change itself!), and as so you will need to be flexible and also to teach your internal customers that this is the reality of the game.

The math is simple: If you pull the blanket on one side of the bed then someone will be left out on the other side...

Stop committing to tasks you know up front you cannot perform, and start updating your users as soon as you understand you will not be able to perform a task that you previously committed to do.

Timely information is the key to making the correct decisions and compromises.

#### 5. Give full visibility into your work

Hand in hand with prioritizing and saying NO comes the practice of providing visibility to all your work and tasks, as well as to the prioritization and the changes affecting your work.

Some managers think that if they "show all their cards" they lose part of their power, their flexibility and the ability to make their own decisions. This could not be farther from the truth!

By working in full transparency with your customers and peers you will show how professional your work really is. You will also allow them to provide inputs that will help you make better decisions for your team and for the success of your project and your company.

#### Do you have any other ideas or tips?

Transparency and sharing is the key to growing and improving.

If you have any other simplifying ideas share them with us!



**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

# Teach-Testing →

An Ambitious Campaign by Tea-time with Testers



Click here to Cast Your Vote



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](mailto: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*

## **Behave yourself!**

### **A "Descriptive to Prescriptive" approach to test functionality**

One of the common statements that I come across is "Domain knowledge is critical to testing functionality". The belief is that knowledge of the domain acquired from prior experience is very necessary to perform good functional testing. Not that I disagree, because prior experience is always handy. But the key question is 'What if I am encountering that functionality for the first time? Will I be effective?' Wouldn't it be nice to be able to do this using a set of techniques in a scientific manner?

Let us first examine the basics... The objective of functionality testing is to ascertain if the behaviour of the Entity Under Test (EUT) is as intended. So what is behaviour? When we tell a unruly kid, "Behave yourself", what we mean is 'please comply with the acceptable norms i.e. rules'. Hence behaviour of an EUT is about 'obeying some rules'. These 'rules' are really the specification. So to test a functionality of any EUT, we need to first understand the intended rules and then create scenarios that are really various behaviours by combining the rules in different ways and then check the outcome(s).

So how do we figure out the rules? Hmmm... If I had domain knowledge then I would know how this should work, and therefore the rules. Note that this can be dangerous too as this thing sets up a bias, and we may assume certain rules that may be incorrect. Anyways, the issue is, how otherwise? To understand the intended behaviour, first describe what the EUT is supposed to do. Describe the behaviour as a series of steps. Each step accepts/uses data and processes the data according to some condition(s). Examine each step and extract the condition(s) hidden in the step. Voila... We now have the various conditions aka rules.

This technique of extracting conditions (aka rules) is what I call "Descriptive to Prescriptive", that is converting the descriptive specification to a prescriptive specification. "Describing" enables us to understand the behaviour from first principles while "Prescribing" helps us to validate the behaviour. We all know the act of clear and comprehensive description consists of the **5W+1H - Who, What, When, Where, Why & How**. So when you encounter any functionality for the first time, apply 5W1H and describe the behaviour as a series of simple steps.

Now examine each step and identify conditions implicitly or explicitly present. Once the conditions are extracted from each step, re-write each step as a series of rules and what we have now is prescriptive behaviour. From this it is easier to create the test scenarios and then identify the various data sets for each scenario and subsequently generate test cases and therefore for the entire EUT.

This behaviour based approach to functionality testing is implemented by two techniques 'Box Model' and 'Behaviour-Stimuli Approach' in HBT (Hypothesis Based Testing). This lessens the need for domain knowledge as a prerequisite by enabling you to question better to understand behaviour, and then extracting conditions to test effectively.

So whether you test a technical feature or a business flow try this approach. Be the doctor - "Ask the patient to describe so that you can prescribe". Enjoy the happiness that comes out of the cure.

May you contribute to healthy software and a cleaner world.



**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](mailto:ash@stagsoftware.com).



[Back To Index](#)





## 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 below b4 20<sup>th</sup> Oct. 2012 & grab your Title.

Send -> [teatimewithtesters@gmail.com](mailto:teatimewithtesters@gmail.com) with Subject: Testing Puzzle

**Exciting  
PRIZE for 1<sup>st</sup>  
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.

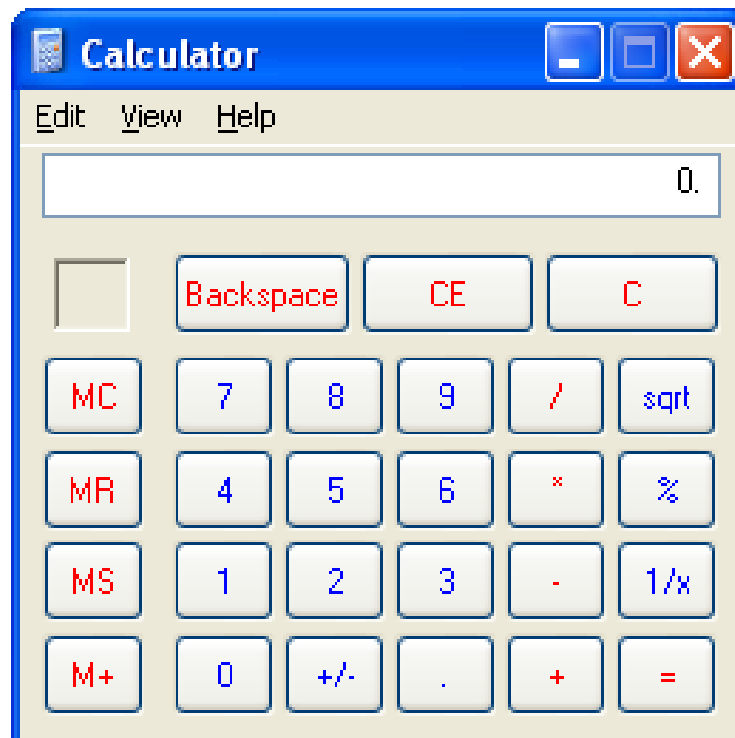
\*CONDITIONS APPLY.

# “The BIG Value”

What is the biggest value that can be represented on the Windows calculator?

Hint: there is binary, hex, decimal etc representation

The answers should have an explanation.



[Back To Index](#)

## 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.

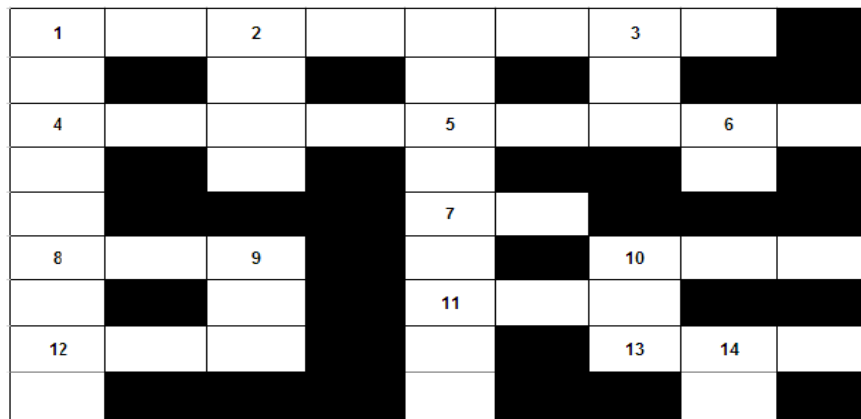
He tweets as @testalways.

You can find some interactive testing puzzles on his website [www.testalways.com](http://www.testalways.com)





# TESTING CROSSWORD



## Horizontal:

1. A software item which is the object of testing (8)
4. It is performed to verify that the completed software package functions according to the expectations defined by the requirements / specifications, first word (8)
6. It is a method for deriving test cases, in short form (2)
7. A test suite that exercises the full functionality of a product but does not test features in detail, in short form (2)
8. It is checking Application Programming Interface of a Software System, in short form (3)
10. An approach to integration testing where the component at the top of the component hierarchy is tested first, with lower level components being simulated by stubs, in short form (3)
11. It is a command-line tool for generating data for Random Testing & Field testing (2)
12. A test case design technique in which the test case suite comprises all combinations of input values for component variable, in short form (2)
13. It is a tool to realistically simulate multiple browsers requesting pages from a web site and a large number of requests with a relatively small number of client machines, in short form (3)

## Vertical:

1. The product of PushToTest (9)
2. \_\_\_\_\_ provides for the implementation of compatible keyword-driven test automation frame work, in short form (4)
3. Middle words in the term LEARN (3)
5. It is a fully functional, commercial-grade performance testing product (7)
6. It provides mock objects for interfaces by generating them on the fly using Java's proxy mechanism, in short form (2)
9. A group of people whose primary responsibility is software testing. It is called \_\_\_\_\_, in short form (3)
10. An online test case management and test-tracking system built with PHP, in short form (3)
14. Testing is concern with the application testing without following any rules or test cases. It is called \_\_\_\_\_ testing, in short form (2)

# Answers for last month's Crossword:

A	P	P	H	A	N	C	E
S		T		U			G
C	U	N	I	T	D	G	G
E		E			E		A
R	I	S	K		V		L
T		T		R	A	M	P
		E		C			H
S	T	R	E	S	S		A

## Answer for last Testing Puzzle:

We can determine the bad input box by just taking one set of values.

Eg: Number the boxes 1,2,3,4 and 5 respectively.

Take test data for 5 input boxes as: 10,20,30,40 and 50 respectively in the 5 numbered boxes.

The last digit displayed in the result box is the bad box number.



*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

Hi Team,

I regularly read this magazine and very happy to see that you conduct interesting competitions too.

I liked August month's editorial as it was quite an eye-opener. Please keep this up and many wishes for your future.

- Neha Verma

Your magazine is very appealing visually. Articles are very interesting.

- Cathy R.



*The KIT Is Koming!*

# TESTKIT CONFERENCE 2012

**Koming!**

**stKIT**

**& Test Automation  
Conference**

**The Date**  
**October 15 - 17, 2012**  
**BWI Airport Marriott**  
**Linthicum, MD**

**The KIT is Koming!**

**TestKIT**

**Testing & Test Automation**

**Conference**

**Save The Date**

**October 15 - 17, 2012**

**BWI Airport Marriott**

**Linthicum, MD**

[www.testkitconference.com](http://www.testkitconference.com)

## Register Now!

**OCTOBER 15 - 17, 2012**  
**BWI AIRPORT MARRIOTT**  
**LINTHICUM, MD**

- Tutorials ■
- Concurrent Presentations ■
- TABOK Certification Training & Exam ■
- Discussion Forums & Networking ■
- 4th ATI Automation Honors ■
- Keynotes ■



[www.testkitconference.com](http://www.testkitconference.com)



# in ne>xt issue

articles by -

A photograph of a metal tag with the text "IT'S ALWAYS TEA - TIME" engraved on it. The tag is attached to a chain and a small metal clock. The background is a dark, textured surface.

IT'S  
ALWAYS  
TEA - TIME

Jerry Weinberg

T Ashok

Joel Montvelisky

Martin Jansson

Michael Bolton

Ben Kelly



# our family

## Founder & Editor:

Lalitkumar Bhamare (Mumbai, India)

Pratikkumar Patel (Mumbai, India)



Lalitkumar



Pratikkumar

## Contribution and Guidance:

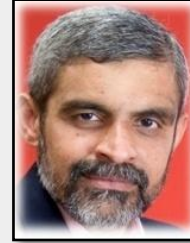
Jerry Weinberg (U.S.A.)

T Ashok (India)

Joel Montvelisky (Israel)



Jerry



T Ashok



Joel

## Editorial | Magazine Design | Logo Design | Web Design:

Lalitkumar Bhamare

Image Credits- weiphoto.com and Harper's Bazaar

## 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



Join our community on



Follow us on



[www.teatimewithtesters.com](http://www.teatimewithtesters.com)

