THE INTERNATIONAL MONTHLY FOR NEXT GENERATION TESTERS

Tea-time with Testers

SEPTEMBER 2014 | YEAR 4 ISSUE VIII

Jerry Weinberg
Managing in a Team Environment

Dean Mackenzie
Egoless Testing

Paul Gerrard
IoE Architecture

Parimala Hariprasad
Testing Lessons from Desing Thinking World

Ashwin Arora
The Art of Localisation Testing

T Ashok
The Promise

Georgios Kogketsof
Road to Test Automation

Over a Cup of Tea with Karen Johnson

testomaton

Quality Assurance via Automation

Software testing startup specializing in test automation services, consulting & training for QA teams on how to build and evolve automation testing suites.

Services

Test System Analysis and work estimation

Review of client's system (either in development or on early stage) to produce a Test Plan document with needed test cases and scenarios for automation testing.

Tests creation and Automation

Development of test cases (in a test plan) and Test Scripts to execute the test cases.

Regression and Test evolution

Development of Regression test suites and New Features suites, including test plans and test scripts or maintenance of existing.

Architecture Consulting

Review of software architecture, components and integration with other systems (if applicable).

Trainings

Depending on the particular need, we can provide training services for: Quality Assurance theory and best practices, Java, Spring, Continuous Integration, Groovy, Selenium and frameworks for QA. For further information please check our web site.

We enjoy putting our experience to your service. Our know-how allows us to provide consultation services for projects at any stage, from small up to super-large. Due to our range of expertise we can assist in software architecture and COTS selection; review of software designs; creation of testing suites and test plans with focus on automation; help building quality assurance teams via our extensive training curriculum.

look us up: www.testomaton.com - twitter: @testomaton

Get ready for your gift… this festival season!

IN THROUGH THE SIDE DOOR

A video talk by Michael Larsen

Stay tuned for more updates…

WWW.TVFORTESTERS.COM

TEA-TIME WITH TESTERS

First Indian testing magazine to reach 115 countries in the world !

Created and Published by:

Tea-time with Testers.
B2-101, Atlanta, Wakad Road
Pune-411057
Maharashtra, India.

Editorial and Advertising Enquiries:

Email: editor@teatimewithtesters.com

Pratik: (+91) 9819013139
Lalit: (+91) 8275562299

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

Editorial

Of Love and Lights…

New season has already started showing its arrival. Air is turning pleasant and atmosphere is getting filled with joy, excitement and love.

There is this amazing thing about festivals. No matter where we are, how we are; they always give us joy, happiness without asking for it. And without expecting anything from anyone in return. One of the other things that I like about festivals is that, they teach us the true joy of giving. And how could we not give our readers something awesome this festivals season? I am glad to offer you TV for Testers and Software Tools Magazine; two new ventures by Tea-time with Testers.

Over the period of time, we realized that there are testers with amazing ideas and skills with them but not everyone can express it well when it comes to writing. So we decided to create an alternative platform where such testers could showcase their ideas to the world, in the form of videos. It would also serve as additional platform to learn from our industry leaders and experts in the form of their video talks, interviews, webinars, tool demos etc. We have already posted some amazing videos on TV for Testers and I am sure that you are going to like then.

First issue of Software Tools Magazine is all set ready to be launched this Diwali. I will let you find more about it in its launch edition itself.

Till then, enjoy this awesome issue of Tea-time with Testers. On behalf of TTwT family, I wish you a very happy and prosperous Diwali in advance. May this season shower you with light of wisdom and success.

We will meet soon.

Yours Sincerely,

Bhamare

- **Lalitkumar Bhamare**
editor@teatimewithtesters.com

THE INTERNATIONAL MONTHLY FOR NEXT GENERATION TESTERS

Tea-time with Testers

AUGUST 2014 | YEAR 4 ISSUE VII

Jerry Weinberg
Managing in Team Environment

Iain McCowatt
STOP ISO29119

Paul Gerrard
IoE- Architecture and Risks

Sanjay Joshi
Automated Accessibility Testing

Georgios Kogketsof
Road to Test Automation

T Ashok
Load testing is not bombarding the system with concurrent users

Over a Cup of Tea with James Christie

© Copyright 2014. Tea-time with Testers. All Rights Reserved.

August 2014| Year 4 Issue VII

Dear Lalit,

I've just got around to reading the recent Teatime With Testers, and what a bumper issue that was.

What I most enjoyed was your editorial talking about ISO29119 from the uniquely Indian folklore perspective of the Mahabharata. Those quotes were superb pieces of wisdom.

I love pretty much any countries folklore, I think it's a way people have used to pass wisdom and "stories that matter" down the generations. I know English folklore best. A good folk tale is at its core a good story, but for those with experience of life, there is an understanding of there being "seeds of great truth" within.

Ian McCowatt’s piece on stop 29119 was superb, and I'm pleased he brought up the witch hunt that's likely to occur in our future, when a piece of software will fail, and a crowd with torches and pitchforks will demand "why didn't you use THE standard", without really understanding what the standard involves.

The interview with James Christie was superb, but I didn't appreciate it being split up like that.

This quote is indeed words to ponder on -> "As an auditor I always disapproved of auditees who seemed to be doing work to cover themselves in a future audit rather than to do the best job they could in the present."

- **Mike Talks**

We love your feedback.

Write to us on editor@teatimewithtesters.com

QuickLook

What's making News?

Meet David Keil, new CEO of QASymphony

We recently had the pleasure to name our new Chief Executive Officer David Keil to join the QASymphony team. Our founder and prior CEO Vu Lam will continue taking an active role in his new position as Chief Product Officer, working with our internal developers to bring you the best tools on the market for creating great software.

David brings years of rich executive leadership experience, having previously served as CEO of Digistrive, Inc. of Atlanta, a provider of e-commerce solutions, as well as being CEO of Integrated Broadband Services, a software and services company. He holds an MBA from The Wharton School and a BS in Applied Mathematics and Economics from Brown University.

What made you decide to join QASymphony?

I had the good fortune to work in leadership roles with several technology companies over the past 25 years. I was hoping to find the right group of people that I felt I could work well with and have mutual respect. When I met with co-founders Josh Lieberman and Vu Lam, it was made clear to me that these individuals are high quality people who I could collaborate with and learn a lot from. After speaking with people in the industry I learned that the company is heading directly into an accelerated market growth dynamic as Agile ramps up and is further embraced by the more mature customer set.

I've had a lot of success in my career with sales and marketing and on the strategic side, and I've been strong on financing. These are the key areas where the company was looking to shore up resources. It seemed to me that my skill set would be valued and highly complementary to QASymphony. There's a high chance for success.

What would you say is your leadership style?

There are three areas I find most important. First and foremost, I believe heavily in accountability. People thrive in that type of atmosphere. Secondly, I care deeply about teamwork. As we continue to supplement the team with other leaders and associates, we'll focus on the dimension of culture and fit and teamwork. Having a consistent team-oriented approach to what we do is something that I will reinforce in my leadership. Thirdly, I feel it's important for people to have fun and enjoy what they do, to feel excited about coming to work each and every day.

How will your past experience guide QASymphony to the next level?

Over the next few months we'll map out a long-term strategic plan. These plans are dynamic and not designed to be rigid because they're open to change. It's important to let the team know where we're headed. I will also put a lot of effort around competitive strategy. I'll leverage my experience around sales leadership in growing a strong and motivated sales team. Another area of focus is in our ability to strengthen the balance sheet and bring outside investment to complement existing investments. We have a promising future and I look forward to injecting some financial sophistication into the company over the short and long term.

What are your thoughts about the market?

I like to spend time with customers since that's where a lot of learning comes from. There seems to be a lot of pent-up demand in the market for a more advanced, more sophisticated solution that allows for the exploratory testing we provide. We hit a segment of the market that has not historically received a lot of attention from newer entrants. Plus many of the legacy providers have not made the big investments that one would expect. We're seeing a lot of passion around agile testing. It's a very exciting and vibrant time for us.

You recently visited Vietnam, where many QASymphony employees live. How was that experience?

I've traveled quite a bit in my career… to South America and Europe but I've not done much traveling in Asia. My first trip to Vietnam was outstanding. It exceeded my expectations on many levels. Vietnam is fascinating because of the energy and vibrancy of its people. They are so optimistic about the future, and have so much energy and passion about their work. The business environment there has improved dramatically and opened up, creating terrific opportunities. The people are incredibly talented with a prideful culture. It was also important to see firsthand how testing offers a market where QASymphony is poised for success.

How do you like to spend your free time when not in the office?

I've been married for 19 years and I have two young children, an 11-year old daughter and a boy who is 14. As proud as I am about what I've accomplished in business, I'm even more proud about my role as a father; that's been the most important job I've had in my life. Also, I'm very passionate about baseball. I've been a longtime New York Mets fan. People like to joke with me because as a fan, it has been quite humbling. But it builds a lot of character.

About QASymphony

QASymphony's test management and agile testing solutions help teams create better software. With QASymphony's tools businesses can accelerate testing to keep up with the pace of today's development to ensure that productivity gains from Agile development are matched with the oversight, visibility and control required to build quality software. Empowering quality software at companies such as Silverpop, BetterCloud, and Zappos, QASymphony enables teams to communicate and collaborate faster, bringing visibility and control back to the development and testing lifecycle. The company is headquartered in Atlanta, GA.

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

# # #

Press Contact:

Victor Cruz

Principal, MediaPR

vcruz@mediapr.net

Back To Index

Photo credits: Amir Eltahan

Karen Johnson is an IT professional with experience in fast paced environments. Sheis experienced in both startup and large corporate environments as well with teams located onshore, offshore and a blend of onsite/offshore. She helps organizations and teams create high quality software.

Over the course of her career Karen has focused on software testing and she is well-known in the software testing community, but in the past eight years working as a solo consultant, she has acquired practical business and organizational acumen.

Karen has written for TTwT on multiple occasions in past. Ever since I have got to know Karen and about her work, I was keen on meeting her personally. Finally, I got to meet Karen at CAST 2014 conference. It was pleasure talking to her and I did enjoy interviewing her in person.

Read on to find out what all we talked about, in this exclusive interview.

- Lalitkumar Bhamare

Over a Cup of Tea with Karen Johnson

Thank you for talking with us today, Karen. We are keen to know your journey in software testing.

The obvious significant milestone for me was when I left my last full-time job eight years ago I have been working as an independent consultant since. But let me go backwards in my career and highlight an aspect of my career that has been the most fulfilling and interesting.

In the beginning of my career without so much a conscience decision to do so but in hindsight it seems clear to me now, I changed jobs in general every three years and each time sought a very different type of software to work with. I think that those radical changes in work helped to feed a hunger to learn that served me well.

In the past eight years, I've been working as a hybrid between a contractor and a consultant. I work on a project/contract basis and each contract is essentially a story of its own. I still look for an array of experiences, preferring situations where this is some aspect that I'm still learning while bringing expertise. Working contractually has allowed me to make changes more rapidly, a situation I enjoy.

We are aware of your work in mobile testing field already. Would you like to share some stories from your experience/experiments in other spaces in field?

The mobile testing "thing" – I need to clarify. In 2009, I started working with mobile because I had a client request to assist a team with a project. Several aspects of that work were cool and ahead of the times such as planning what devices to test based on user stats, multi-lingual SMS text testing and other aspects that testers are still dealing with. As typical for me, I started speaking and writing about my experiences and rapidly became known for mobile testing. I've taught on the subject but what I think it is not understood that I have not been focused on mobile in the past couple of years, I don't consider myself in the mobile space any longer. Interested in mobile – yes! Multiple device testing – yes. But I am not focused on mobile as much as I was. It's funny to me now how quickly we give people attribution as experts without knowing on the basis of what? Or if the so-called expert has done recent work in thearea. We should be more skeptical! For almost three years now, my interests and energies have been in other directions.

Mobile devices bring lots of use case possibilities and scenarios with them. What would be your advice for testers so that they can cope up with these challenges and be good at it?

My advice to testers focused on mobile – don't forget to expand your knowledge in broad based learning and other technologies. Mobile is cool, offers interesting challenges but like all technologies – there will be something else to learn next and if all your background is hyper-focused on one technology, you're not developing yourself. From someone who has been around awhile, believe me – there will be something else to learn.

What things about 'writing' do you think have helped you become great at testing?

Writing is a mind clarifying activity, as is teaching, both activities make me pull my thoughts together, think clearly and see what I can share with other testers. Both activities also help me to see the overall things to learn versus the particulars of the project I'm working on. I have to know the difference because while experience reports are interesting to hear – they are not always broad-based enough to appeal to a reading audience for my writing or a class for an audience when I'm teaching.

A comment I've wanted to make on writing is this: I have a journalism degree, I enjoy writing and so I do, I don't write to become better known or to market myself but from a natural instinct to write. At least a couple of times a year I meet someone who says they would like to write but their reason to do so is to establish or position themselves in some way and they don't enjoy writing. Don't write if you don't enjoy it.

... CONTINUED ON PAGE 60

Tea & Testing with Jerry Weinberg

Managing in a Team Environment

Are you the problem?

When you are unable to get teamwork it's often because of something you are doing. It may be something you are doing consciously, but that has a boomerang effect. Performance appraisals—intended to help each member of the team learn—often have this effect. By evaluating the individual members of the team, no matter how well you do it, you are sending a message that you're not able to trust their own evaluation of each other.

If you feel you have to give performance appraisals, a more effective way is to appraise the team performance and ask the team to appraise the performance of each member. The external performance rating is obtained by multiplying the team's overall rating by the individual's rating as given by the team. This approach says you trust the team to do what's best for meeting the goals you give it, and that they all get credit equally from you for meeting those goals as a team.

Note: Read complete series from June 2014 issue

Most people are aware of the divisive nature of conventional performance appraisals. You may, however, be giving signals in other situations that are quite out of your awareness, as in the following story:

Cora, the senior vice president of Information Services, had two VPs working under her who were supposed to team up to create a strategic plan. Tom and Len argued for eight months over approach T versus approach L.

Cora was, in fact, unconsciously contributing to the endless discussion because she was giving signals that indicated the winner would be next in line for her job. Thus, the matter was not settling T versus L, but Tom versus Len.

Once Cora discovered of what was going on, she told Tom and Len one morning at 8:30, "You two go in the conference room and don't come out until you've settled this. If you don't come out by noon, you're both fired."

The matter was settled by 9:15. I don't recall if T or L won. But of course, it didn't matter since Tom and Len were working as a team.

Using the MOI model for interventions

When choosing interventions, a good guide is the MOI model. This model says that when a team or individual isn't functioning well, there may be a missing ingredient. It could be motivation, organization, or information. Your job is to find out which, and intervene accordingly.

The MOI model will help you avoid egregious mistakes like this one:

The company had embarked on an ambitious campaign to improve quality. As part of that campaign, they formed an all-volunteer group of 25 software engineering advisors. The members of this group were to receive some training, then act as coaches to other members of the organization. In order to be selected, they had to put themselves through a time-consuming and rigorous process involving interviews, essays, and five recommendations. They also had to agree that all work would be over and above their normal workload, so that any classes or coaching time would have to be made up by working unpaid overtime.

The selection had been highly competitive, so the software engineering manager decided to reward those chosen. He hired a famous and expensive motivational speaker to give them a one-hour pep-talk—exactly what they didn't need. After all they had gone through and committed to, they were insulted that the boss thought they needed a motivational push. What they wanted was some organization to their work, and information to equip them to coach others. The meeting, which was certainly well-intentioned, wasted a lot of money, but more important, undermined the relationship between the volunteer team and the boss.

Communicating with the Outside

All new managers have to wrestle with their role as interface between the team and the rest of the organization. One role is serving as conduit for messages and directives from upper management; another is as a conduit for messages traveling in the other direction. If the new manager gets any formal training at all, it generally deals only with the company's policies and procedures concerning administration and personnel. This training suggests that the company, at least, views the manager's most important role as inward communication.

On the other hand, if the new manager has been moved from the ranks of the workers, it's easy to slip into the outward communication role.

Mark Weisz had this to say about a third possible role of the manager with respect to the exterior of the team:

Now that I have someone working for me it's really made me "protective." For example, I try to grab all the interrupts for myself so at least someone is making steady progress. Protecting your work group is absolutely necessary—that's a good instinct to have. Back in grad school, they called it "boundary management," but in less stuffy language it's "air cover" or "The Mother Duck Syndrome." "Grabbing all the interrupts" sort of sounds like hardware, like an interrupt controller. Would that sound good on a resume? "Hi, I'm a professional interrupt controller."

In other words, not only do you send communications outward, but you act to filter as much inward data as possible. Your job is to screen data that pesters the team, while allowing data that genuinely helps it. A good example of using the Mother Duck role effectively is in negotiations between software developers and marketers. The typical marketer may be ten years older than the typical developer. Thus you often have children negotiating with adults, who also happen to be professional negotiators. This bias often pushes the negotiation one way, leading to the common symptom: "schedule macho."

Managers who want to protect their teams would do well to forget about C-coding and develop their negotiation skills and congruence, as suggested by this story told by Mark Manduke when he read a draft of this section:

I was software manager for a team of seven programmers. Several members of my team were foreign born, young, and easily intimidated by the burly marketers and senior management, who would interrupt them to get agreement on "just one more bell and whistle for my special customer." We were having significant problems meeting our schedules, which also seem to make the marketing and management folks grin as they tortured us with deadlines and watched us put in exorbitant overtime.

One day, I called a meeting for the team and explained, "From now on, we will reach a team consensus on our schedules for assigned software tasks. They will be reasonable and reflect what we believe we can do in a forty-hour week. I will negotiate those schedules with Marketing and Engineering management. Once we are all comfortable with a development and delivery schedule, any private attempts to change or add requirements will be politely referred to me. If we can't meet our own estimates, then we will work for any necessary overtime to compensate. But one of my duties to you, the team, will be to act as the single negotiator for any changes. If we all stick together and speak one voice, we will have unchallengeable power. They can't fire all of us."

The result was amazing. This team became the hardest working and most fiercely loyal software compatriots I have ever had the pleasure of working with.

That's the kind of congruence you need to be a successful manager of software teams.

Helpful Hints and Variations

1. If you become too effective as a Mother Duck, team members may come to believe that you are hiding things from them. You will also prevent them from having the information and experience they need to make their own career decisions, such as whether they would like to try their hand at management.

2. Phil Fuhrer suggests a relationship between failure to support teams and a common survival rule:

For many managers the rule: "I must always being in control of my staff or at least look that way," is a major source of incongruence. I think this might come from early childhood where mom or dad controls the family or the teacher controls the class. I also think this rule is reinforced by upper management wanting to feel that middle management is in control. I think this control rule is the chief reason that teams are not really supported.

3. In Dan Starr's experience, customer relations are an important part of setting up a successful team:

Another thing that teams need is a good definition of the real client and problem to be solved, for a team builds enthusiasm and commitment as it grows. This is especially true in an organization where teamwork is not the norm. The team will come to perceive itself as something different and a little special, and it hurts big time if all that energy and work is wasted because you got the problem definition wrong.

Summary

1. Many managers undermine team spirit by their high-and-mighty attitudes. Other managers unconsciously undermine their own efforts to reap the benefits of effective teams. The first group is hopeless, but the second can be helped by learning to become more congruent.

2. Perhaps the most frequent confusion about the manager's role in a team-based organization is between the manager and the team leader. The team leader is responsible for the technical task of the team. The manager is responsible for the nontechnical direction of two or more teams.

3. Seen from inside the team, the manager's role is to unburden the team leader by handling certain nontechnical tasks. Seen from the manager on the outside, the manager's role is to control the team in terms of the higher goals of the organization.

4. The number one way to build teams is to delegate challenging work. Challenge is an emotional reaction, influenced as much by the way the task is assigned as by the task itself. Each temperament, for instance, is challenged in a different way, so an effective leader has to reframe a challenge for each temperament.

5. Delegating is not as easy as it seems to non-managers, for several reasons. It's easy to be misunderstood, to use the wrong communication medium, to make mistakes, and to become defensive when things don't work out and people complain about what you did.

6. Managers make two mistakes concerning the amount of control they exert over internal team affairs: too much or too little. Pattern 1 managers tend to intervene too little, placating the team. Pattern 2 managers tend to overcorrect by intervening too much, blaming the team, often in the form of claiming to be better able to do the job themselves. Both placating and blaming usually stem from the manager's internal needs rather than the team's.

7. Well-functioning teams can be recognized by the behavior of their members. They make sure that everybody participates in decisions. They stay in touch with one another, and everyone feels the chance to contribute. They are united, and talk in terms of "we." They have fun. They rely on each-others' individual strengths and all do what they can that's best for the team. When they speak, they take care to be sure that everyone understands. They each feel strong and useful, but not out of proportion to their competence as a team. Finally, it's easy to monitor them, because they show how they are truly feeling.

8. You manage a team gently, using single incidents only to trigger a state of alertness for other signs. When you do make interventions, you use them as opportunities for team building and team problem solving.

9. It's easy for a manager to envy a well-functioning team. The cure is to emphasize the joys of successful management, and the part successful management plays in the team's success.

10. Attempts to reward a team can often backfire. Work with the team to find out what would truly reward them. Use the MOI model to determine what kind of intervention a team needs at any moment.

11. It's not the manager's job to punish people. It is the manager's job to arrange opportunities for people to learn. To do that, the manager may have to play Mother Duck, protecting the team from inappropriate influences from outside.

12. Performance appraisals can destroy team spirit. Avoid them if you can. If you can't, a more effective way is to ask the team to appraise the performance of each member. An external performance rating is obtained by multiplying the team's overall rating by the individual's rating as given by the team.

Practice

1. Probably the worst intervention a manager can make in a team is giving performance appraisals to the individual members. Explain this statement, using diagrams of effects to support your argument for or against.

2. What are your three top candidates for the worst intervention a manager can make in a team? Explain each.

3. (Bill Pardee) Since so many people abuse the term "team" by using it to describe any group the speaker wishes had a common goal, compose a short explicit description of what would constitute a "congruent team" in your organization.

4. Bill Pardee asks, "If we think teams (and I do) constitute the indivisible atom of effective software development, wouldn't it help to reduce the baggage associated with the classical term, in which so many people call any group a team? Katzenbach and Smith use the term "real team" to make this distinction. Can you propose a better term to describe such teams?

5. In Mark Manduke's negotiations for his "fiercely loyal software team," identify how he balanced self, other, and context. Can you apply this approach to a team you're involved with?

6. (Peter DeJager) When you delegate tasks, do you delegate problems or solutions? Which do you think is a better way to get effective performance from a team?

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 **SoftwareTest Professionals first annual Luminary Award.**

To know more about Gerald and his work, please visit his Official Website here .

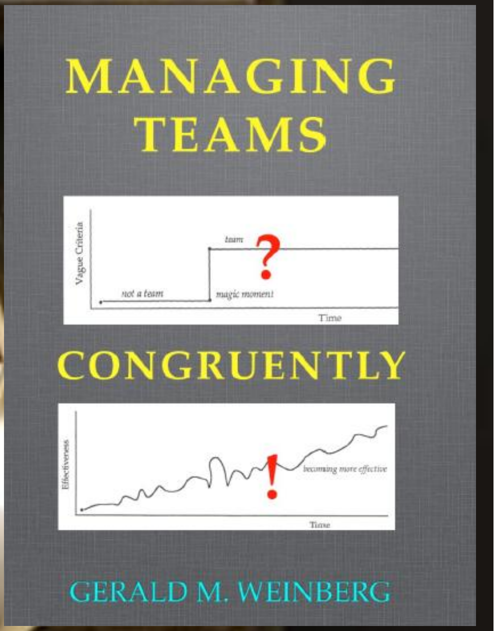
Gerald can be reached at hardpretzel@earthlink.net or on twitter @JerryWeinberg

To be effective, team managers must act congruently. These managers must not only understand the concepts of good software engineering and effective teamwork, but also translate them into their own practices. Effective managers need to know what to do, say what they will do, and act accordingly. Their thoughts and feelings need to match their words and behaviors.

And how should they do that? Jerry has shared this secret in his **MANAGING TEAMS CONGRUENTLY** book.

Its sample can be read online here.

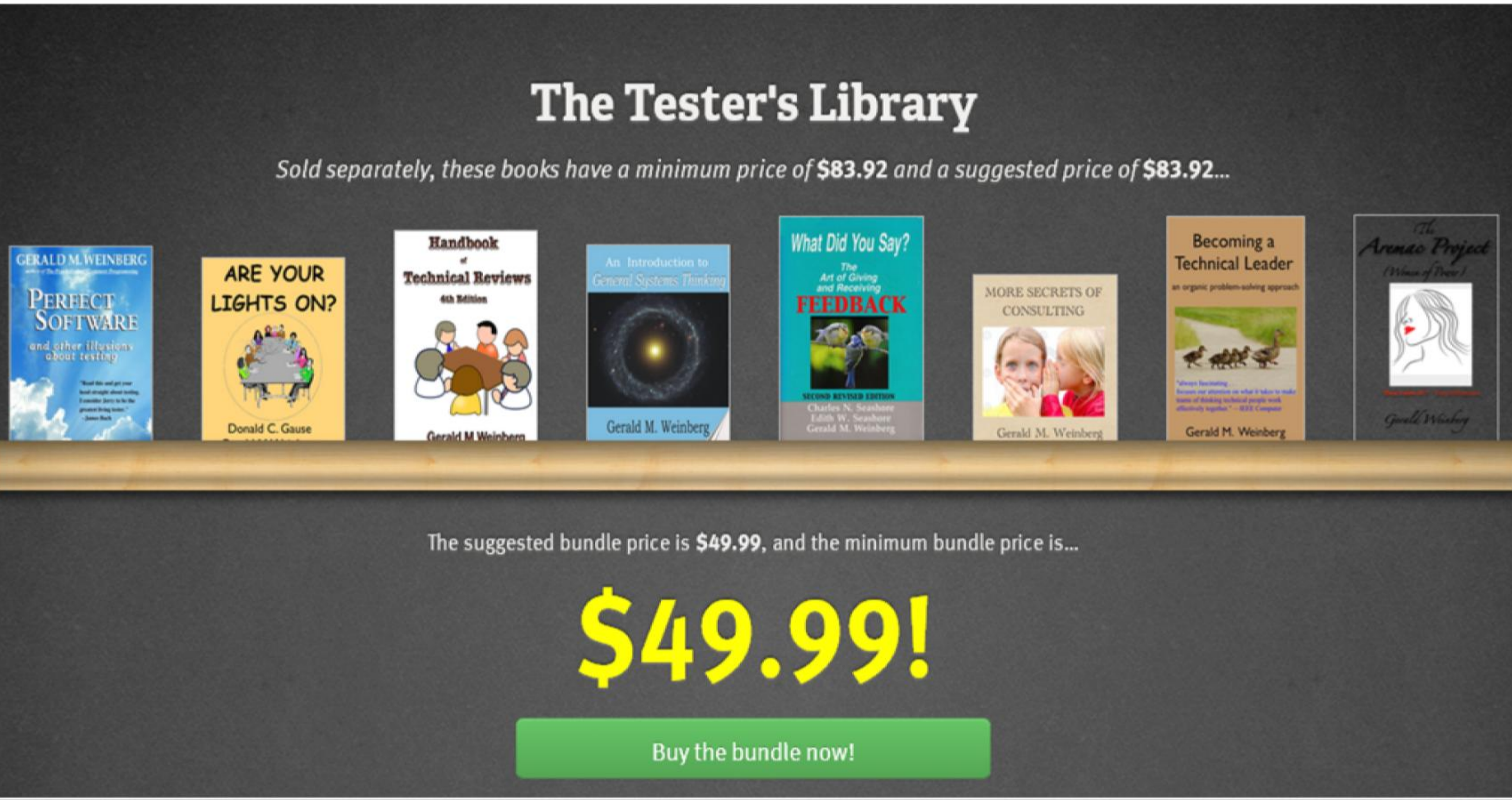
To know more about Jerry's writing on software please click here.



TTWT Rating: ★★★★★

The Bundle of Bliss

Buy Jerry Weinberg’s all testing related books in one bundle and at unbelievable price!



The Tester's Library

Sold separately, these books have a minimum price of **$83.92** *and a suggested price of* **$83.92**...

GERALD M. WEINBERG
PERFECT SOFTWARE and other illusions about testing
ARE YOUR LIGHTS ON?
Donald C. Gause
Handbook of Technical Reviews 4th Edition
Gerald M. Weinberg
An Introduction to General Systems Thinking
Gerald M. Weinberg
What Did You Say? The Art of Giving and Receiving FEEDBACK
SECOND REVISED EDITION
Charles N. Seashore
Edith W. Seashore
Gerald M. Weinberg
MORE SECRETS OF CONSULTING
Gerald M. Weinberg
Becoming a Technical Leader
an organic problem-solving approach
Gerald M. Weinberg
The Aremac Project
Gerald Weinberg

The suggested bundle price is **$49.99**, and the minimum bundle price is...

$49.99!

Buy the bundle now!

The Tester's Library consists of eight five-star books that every software tester should read and re-read. As bound books, this collection would cost over $200. Even as e-books, their price would exceed $80, but in this bundle, their cost is only $49.99.

The 8 books are as follows:

- Perfect Software

- Are Your Lights On?

- Handbook of Technical Reviews (4th ed.)

- An Introduction to General Systems Thinking

- What Did You Say? The Art of Giving and Receiving Feedback

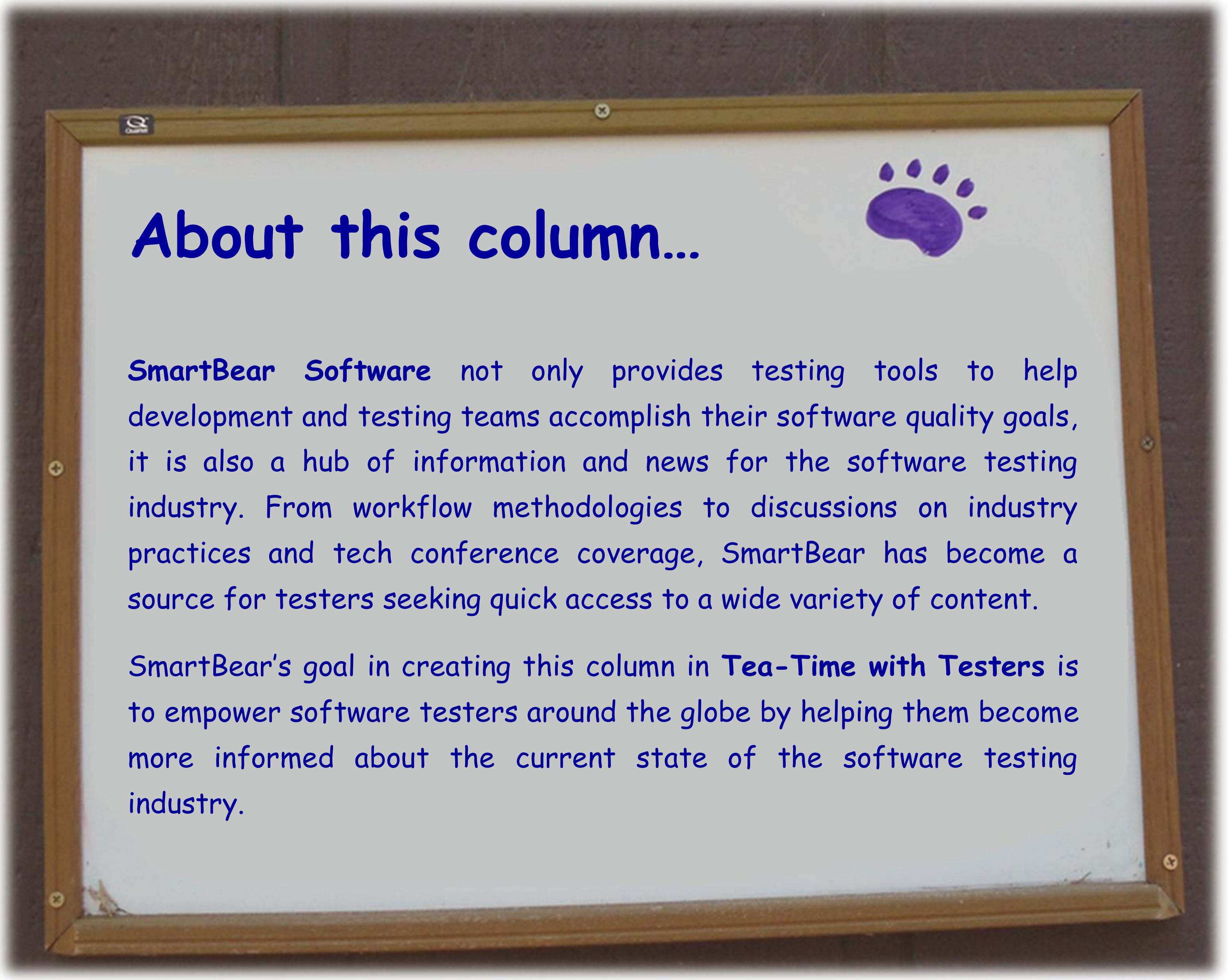
- More Secrets of Consulting

-Becoming a Technical Leader

- The Aremac Project

Know more about this bundle

TEA-TIME WITH SMARTBEAR



About this column…

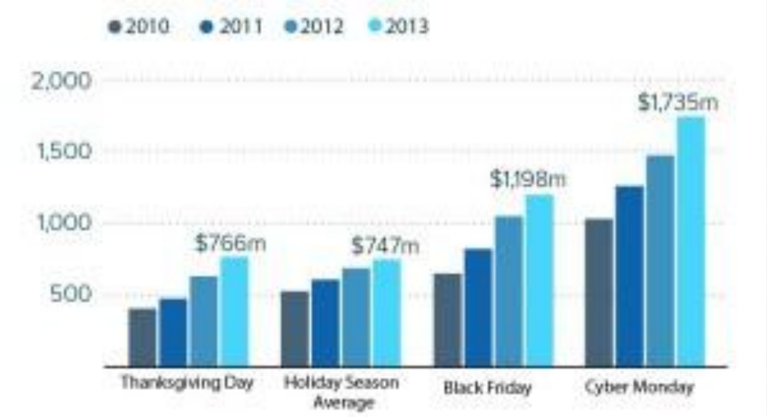
SmartBear Software not only provides testing tools to help development and testing teams accomplish their software quality goals, it is also a hub of information and news for the software testing industry. From workflow methodologies to discussions on industry practices and tech conference coverage, SmartBear has become a source for testers seeking quick access to a wide variety of content.

SmartBear's goal in creating this column in **Tea-Time with Testers** is to empower software testers around the globe by helping them become more informed about the current state of the software testing industry.

This Holiday Season Help Your Website Achieve Peak Performance – Part 1

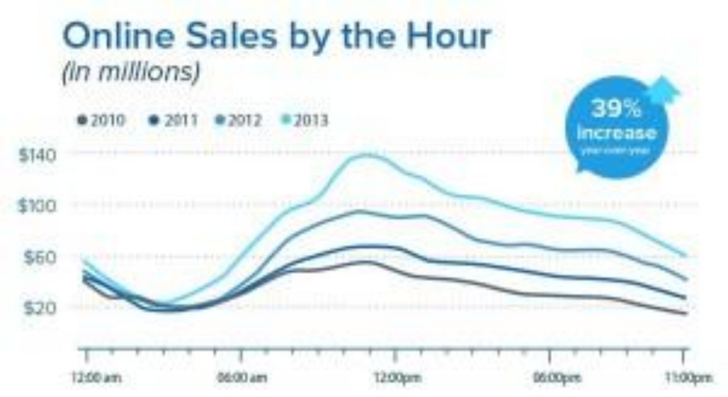
- by **Nikhil Kaul**

This is Part 1 of a multi-part article series intended to provide you with information you need to succeed during the upcoming holiday season. So just sit tight and follow this series for tips you can put to use immediately to help ensure you avoid glitches this holiday season. And when you do make use of those tips and achieve holiday season success, don't forget to share your stories with us.



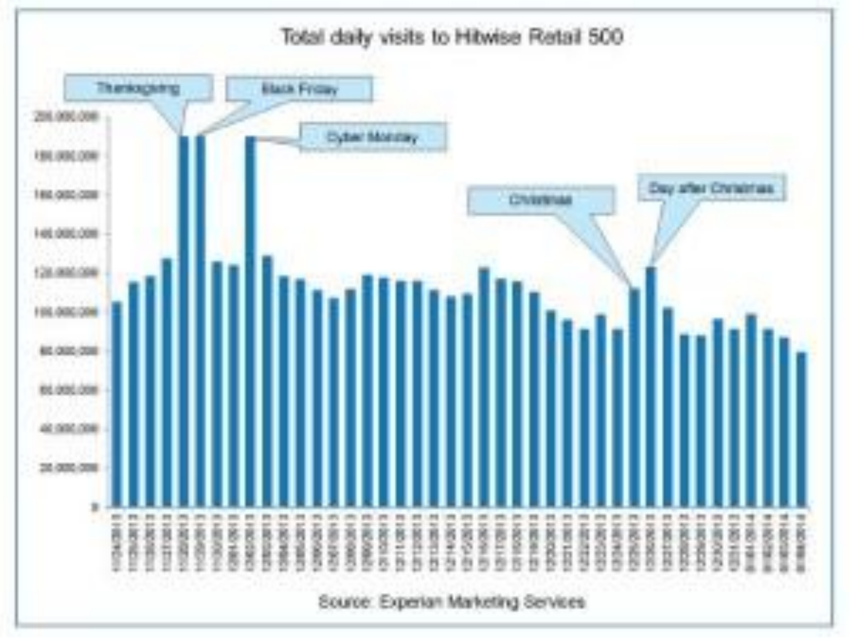
Source: www.statista.com

To truly prepare for the holiday season, you must first understand the enormity of the business opportunity the holiday season represents for any internet retailer. In 2013, Cyber Monday alone presented a revenue potential of $1.74 billion (Yes, billion with a "B"). Black Friday presented the potential of *another* $ 1 billion just from Americans! So while it may have already been obvious to you that the holiday season is prime time for making some extra $'s, I believe these figures give you some sense of scale regarding just how much "some extra $'s" can equate to.



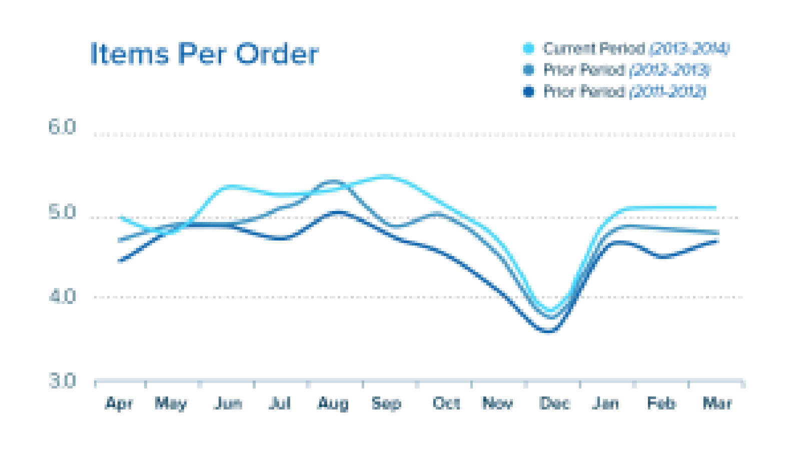
Source: 2013 Holiday and eCommerce Benchmark

If the previous numbers didn't excite you, consider this. Consumer spending not only continues to increase on a year-over-year basis, but also on an hourly basis for the past few years during the holidays. For instance, Thanksgiving online sales per hour grew by 18% year-over-year in 2013. BlackFriday was much more impressive as retailers managed to pull in a 39% year-over-year growth. 18%-39% year- over-year growth is certain to make virtually any business owner or investor salivate! But, to really capitalize on this growth, we first need to understand what is driving this growth.

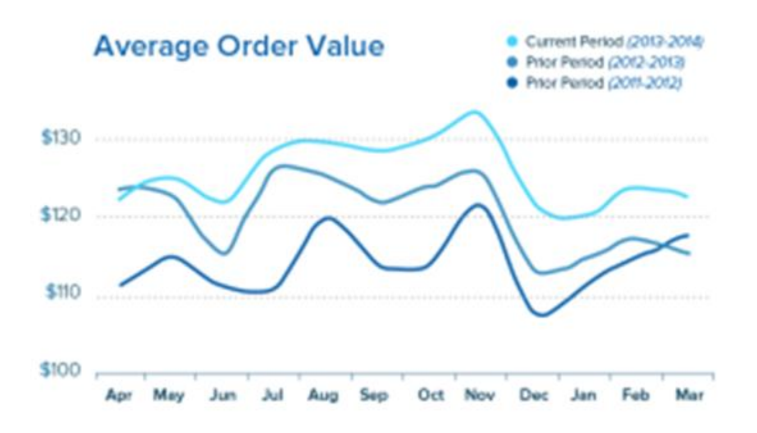

Total daily visits to Hitwise Retail 500

Source: Experian Marketing Services

Obviously this increase in spending has been driven by the increased number of consumers who are moving to online shopping every year, as researching and buying on the web is more convenient than visiting brick-and-mortar stores. For instance, retailers experienced a 20% increase in traffic during Thanksgiving in 2013 over 2012 and on BlackFriday there was an upswing of 17% in internet traffic.


Items Per Order

Source: IBM Seventh Annual Online Retail Holiday Readiness Report



Source: IBM Seventh Annual Online Retail Holiday Readiness Report

Given this sample of the data (and there is a *lot* more supporting data scattered around the web if you’re not yet convinced), the gains in online revenue during the holiday season are not only driven by increased site traffic, but also by new highs achieved in “Average Order Value” and “Items per Order” every year. What this means is that not only increased number of customers are shopping online during the holiday season, but these customers are willing to buy more quantities and spend more while placing an order. Over the next partof this series, we will analyze the challenges e-retailers face while converting these potential customers into paying customers. Also, we will examine numerous steps retailers can take to ensure their sites are adequately prepared for the upcoming holiday season.

An Experience Report by a First Time CAST Attendee

What did I learn / unlearn about testing at CAST?

- The scope and out of scope of testing.
- How to and how not to test at all.
- Is there a school of testing that I belong / subscribe to?
- Do I belong / subscribe to the existing schools of testing at all?
- How to change and quickly adapt to the evolving world of software testing.
- Solving problems and sharing solutions with other community members via conferring, blogging and / or interacting by other communication mediums.
- This list is incomplete, as I am still learning as I jot down these points.

Per Scholas

If you need an honest opinion about CAST, I would say head straight to the Per Scholas participants.

Wowness in a word. Per Scholites knew what was being talked about at this conference and how relevant it is to the current trends in the software testing world, as the education provided to them via the STEP program is first hand and undiluted.

There was an active participation from this group:

- Harrison C Lovell
- J. Winston Tokuhisa
- Jessica Nickel

Humans of New York - And how these people helped me learn.

Keith Klain and **Anna Royzman**

As enthusiastic as ever Keith and Anna, helped me answer a lot of questions throughout my journey to CAST2014. I was glad to meet and spend time with both these AST board members. Thank you for sharing your time. You both made this conference happening via the event promotion prior, during and post the event.

Harrison C. Lovell a Per Scholas graduate, whose thirst for questioning is unquenchable. As he puts it: "I came, I saw, I conquered".

He found automation, philosophy and gaming well paired off with each other at the conference.

There were other talks that emphasized on 'Questioning' as an integral part of testing including the one which Harrison and Michael Larsen delivered - Coyote Teaching.

More about the talk here in Michael Larsen's words: http://www.mkltesthead.com/2014/08/coyote-teaching-watch-how-it-all-came.html

Geordie Keitt from **Doran Jones Inc**

His views on the conference:

- Very well run and logistically smooth. Kudos to all the past and present AST Board members.
- About the number of participants, many including Geordie echoed that the number of participants was / is the right number as compared to the other large conferences.

Ryan Arsenault, a Community Management Associate at uTest

- Introduced me to the ways of working at uTest.
- uTest's vision, mission and focus at uTest?
- What do they do and how they are getting the testers to be socially informed and be active on their discussion forums.

Ryan, for the major part of it spoke about how to get projects at uTest and if you are new to testing and need more information login to http://www.utest.com/

Conferences are a place to meet and learn from testers who are quiet not accessible otherwise and elsewhere. I was glad to interact with **Maria Kademo, Claire Moss, JeanAnn Harrison, Pradeepa Narayanaswamy, Lanessa Hunter** among others with whom I shared practical problems, discussed solutions, approaches used to solve testing problems and gather their take away from this conference.

Thank you all for your involvement. Oh and did I mention Mike Lyles, Michael Larsen who shed light upon other testing conferences happening and live blogging respectively.

A Testing Troupe at CAST2014

Election and results - BOD

The results are announced and we have the new and re-elected BOD's for AST.

President: Michael Larsen

Executive Vice President: Keith Klain

Treasurer: Alessandra Moreira

Secretary: Markus Gartner

Vice President of Education: Justin Rohrman

Executive at Large: Erik Davis and Peter Walen

Congratulations AST BOD's. I personally look forward to all that this team can collectively do for the betterment of the testing community. We already have a great start in the form of STOP 29119 Movement.

I spoke to a few of the BOD's and their vision for CAST2015 and here is an excerpt from the same.

Justin Rohrman

Biggest focus will be on:

- Applicable knowledge (Know it → Use it). So that the participants can help themselves with the practical and applicable knowledge gained.
- Authentic problem solving and that CAST would continue to encourage open debate.

Alessandra Moreira

Being a newly elected BOD of the AST: Ale, wishes to get CAST to reach out to more of its international audiences.

She shared that CAST2014 was different this year because of the buzz and the excitement it generated pre-conference and that CAST is diverse and she was glad to see the testing community maturing. And the part that I liked much about her vision is "Bringing CAST to Asia" which would make CAST affordable to many Asians participants who look forward to be a part of the conference in the near future. I know that IlariHenrik Aegerter is so recommending for Portland as the next CAST venue ;)

For a detailed report on the AST election results, please follow the link below:

http://www.associationforsoftwaretesting.org/2014/08/15/election-results/

Participation from India

From a land of over a billion people, there were a few (countable on fingers) participants.

I wish the participants spread the word and get the community to participate often and in good numbers. (Good numbers = fairly enough to spread the word about the context driven testing community). Many including myself, were unaware of the ongoings in the Software Testing world until very recently. So I urge the participants to go out and get the word spread about this friendly testing community who are willing to learn and share their knowledge with you. I am doing my bit.

Speakers and Volunteers who attended CAST this year were:

Parimala Hariprasad

Dhanasekar Subramaniam

Smita Mishra

Lalitkumar Bhamare

Jyothi Rangaiah

Test Lab - an extension of CAST

An exhibition of testing products and services like TestComplete and SoapUI Pro by SmartBear were on display at the Test Lab.

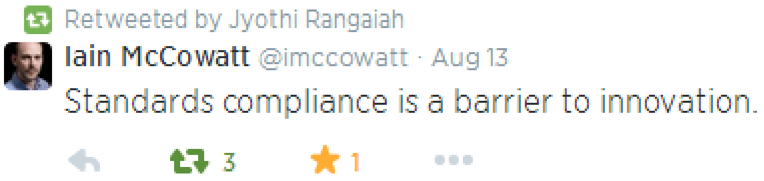
At the Test Lab, I found Mike Lyles encouraging the testers to test the http://mailinator.com/and http://leankit.com/applications.

- We spoke about a host of ideas mostly about the testing community, the participation and the talks at CAST. The relevance of the abstracts prepared with the talk.
- K - card and its usage.
- Traditional versus what new lessons we learnt at CAST.
- The relevance of the talks to the current trends in Software Testing.

Mike shared that he could meet great testers here, who come only to CAST.

Testers took time out of the scheduled talks to come over and log bugs at the Test Lab and win goodies from SmartBear. Thanks SmartBear team for sponsoring and getting involved with CAST in this scale. Spotted Claire Moss enthusiastically participating, I felt she is omnipresent as she was live tweeting the entire event.

This happened at CAST2014



Retweeted by Jyothi Rangaiah

Iain McCowatt @imccowatt · Aug 13

Standards compliance is a barrier to innovation.

Standard 29119 and a petition against it

Standard 29119 non-compliance and what’s in it for me?

Go ahead and read this 28 page document if you wish to.

Guess now you are pretty sure and get to decide for yourself, if this is how you wish to introduce software testing to the future generations? Answer for yourself.

1,127 signatures were registered as of 4th October. If you are a software tester or not yet, you could be a part of this historic moment too. Sign the petition if you choose to by following this link below: http://www.ipetitions.com/petition/stop29119Thanks to Karen Nicole Johnson, Iain McCowatt and the ISST team for their enthusiasm in being the pre-cursors to this movement.

One conference many lessons- whether you subscribe to a particular school of testing or not. This conference appeals to you because of its diversity, the learning it has to offer and the enthusiastic and ever growing community. I urge the people in it to actively participate. The community needs you as much as you need it.

Reading material and related articles

Humongous amounts of information was generated at CAST and a lot of it can be read on participants blogs and the event itself was live telecasted (http://www.ustream.tv/channel/castlive) which can now be watched online by subscribing to the AST news channel on YouTube.

Some references are as follows:

Below is a link to the four page reference reading, book recommendations and related articles collated at CAST.

https://docs.google.com/document/d/1mUcqtkSpJks5aTB1DSIvceDn4JbJ-42LyEb_R2nWiT4/

Here's a Test Engineer's Skills map that I chanced upon while preparing this article and I thought you might want to take a look at:

http://www.ministryoftesting.com/wp-content/uploads/2013/10/test-engineer-skills-map.png

About AST and the Involvement in Testing Community

AST's Grant Program

Read about it here:

http://www.associationforsoftwaretesting.org/2014/05/21/ast-grant-report-notice

Many volunteers (including myself), Per Scholas graduates and the local testing community meetups benefit from this Grant Program. Thank you AST and team for making my wish come true to meet and learn with the world class software testers. I continue to add value and give back to this community in my capacity.

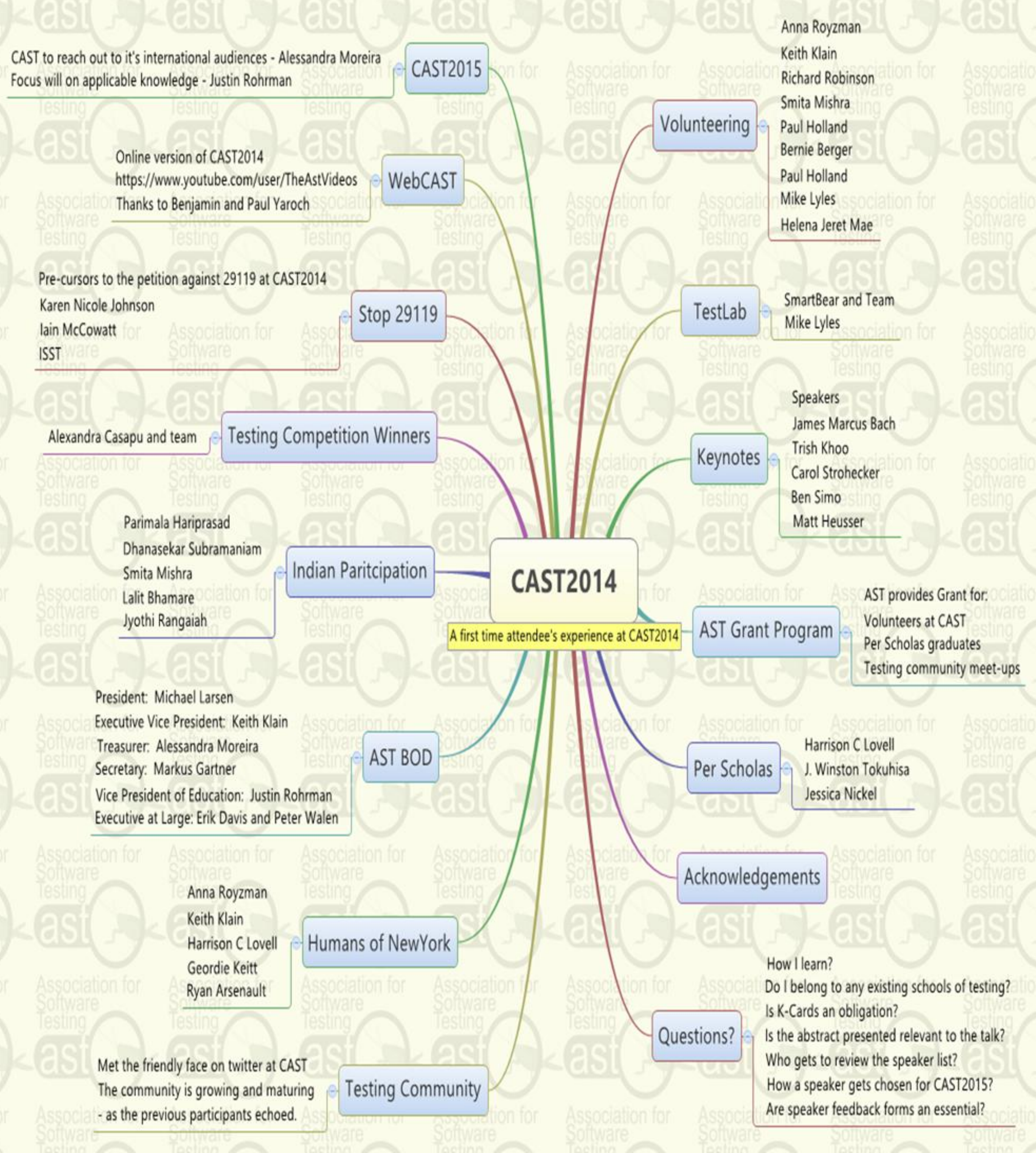
Acknowledgements

Thank you AST Board of Directors, facilitators, sponsors and participants for making CAST2014 a grand success and an event that I look forward to for the coming years.

Thank you Keith Klain and BODs for introducing me to CAST. I have enjoyed my time at CAST meeting people, learning from them and sharing my learning via this write-up.

I look forward to actively participate in the future CAST conferences (live or otherwise) and do my best to get the community grow and I would love to see people share new learnings year after year at CAST and other conferences for software testing.

Here's a mind map version of the above write-up.



CAST2014

A first time attendee's experience at CAST2014

CAST2015
- CAST to reach out to it's international audiences - Alessandra Moreira
- Focus will on applicable knowledge - Justin Rohrman

WebCAST
- Online version of CAST2014
- https://www.youtube.com/user/TheAstVideos
- Thanks to Benjamin and Paul Yaroch

Stop 29119
- Pre-cursors to the petition against 29119 at CAST2014
- Karen Nicole Johnson
- Iain McCowatt
- ISST

Testing Competition Winners
- Alexandra Casapu and team

Indian Paritcipation
- Parimala Hariprasad
- Dhanasekar Subramaniam
- Smita Mishra
- Lalit Bhamare
- Jyothi Rangaiah

AST BOD
- President: Michael Larsen
- Executive Vice President: Keith Klain
- Treasurer: Alessandra Moreira
- Secretary: Markus Gartner
- Vice President of Education: Justin Rohrman
- Executive at Large: Erik Davis and Peter Walen

Humans of NewYork
- Anna Royzman
- Keith Klain
- Harrison C Lovell
- Geordie Keitt
- Ryan Arsenault

Testing Community
- Met the friendly face on twitter at CAST
- The community is growing and maturing
- - as the previous participants echoed.

Volunteering
- Anna Royzman
- Keith Klain
- Richard Robinson
- Smita Mishra
- Paul Holland
- Bernie Berger
- Paul Holland
- Mike Lyles
- Helena Jeret Mae

TestLab
- SmartBear and Team
- Mike Lyles

Keynotes
- Speakers
- James Marcus Bach
- Trish Khoo
- Carol Strohecker
- Ben Simo
- Matt Heusser

AST Grant Program
- AST provides Grant for:
- Volunteers at CAST
- Per Scholas graduates
- Testing community meet-ups

Per Scholas
- Harrison C Lovell
- J. Winston Tokuhisa
- Jessica Nickel

Acknowledgements

Questions?
- How I learn?
- Do I belong to any existing schools of testing?
- Is K-Cards an obligation?
- Is the abstract presented relevant to the talk?
- Who gets to review the speaker list?
- How a speaker gets chosen for CAST2015?
- Are speaker feedback forms an essential?

Cheers and do enjoy the online version of CAST2014 on a computer near you.

Appendix

AST - Association of Software Testing-http://www.associationforsoftwaretesting.org

@ast_news

ISST - International Society for Software Testing- http://www.commonsensetesting.org/

@intsst

K-Cards- http://testingthoughts.com/blog/26

Read the comments too - Making K-Cards accessible to the color blind facilitators, speakers and audiences.

Per Scholas- http://perscholas.org/

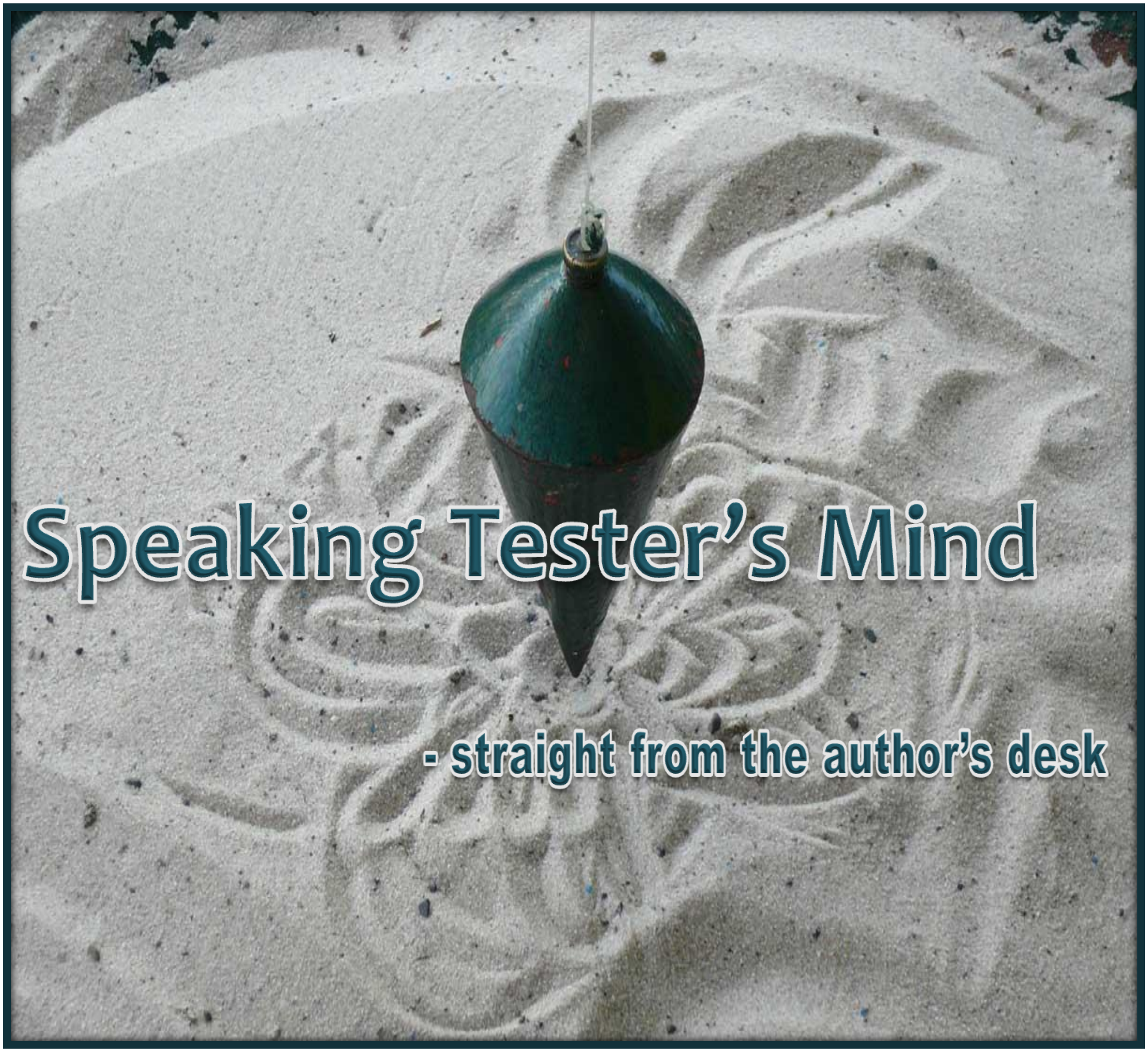
@perscholas

STEP

Software Testing Education Program (STEP) is a free 8-week course written in collaboration with several industry partners and led by Paul Holland.

Jyothi Rangaiah - Software tester at Test Insane, a learner, blogger, building mind map factory at the moment - an open source contribution to the testing community.

Blogs at testbrewer.com and can be reached on twitter @aarjay



Speaking Tester’s Mind

- straight from the author’s desk

Egoless Testing

by Dean Mackenzie

I recently popped my head into the public speaking space to give a lightning talk at the local (i.e. Brisbane) Tester Meetup. It's a rare occurrence, but it was generally well-received, so I thought I would etch my words of wisdom onto virtual paper (for posterity).

About three years ago, I read one of Gerry Weinberg's more well-known books: The Psychology of Computer Programming. Tangent alert: do yourself a favour and take a look at it. While aimed at programmers, a surprising amount of it is still relevant for testers. For a book written over 40 years ago, it was a very prescient and still insightful look at how software development teams work.

Amongst the cornucopia of topics discovered in the book, the concept of egoless programming was especially striking (in the non-violent way). An attempt to address the "problem of the ego in programming", Weinberg cites the example of John von Neumann (one of the earliest programmers), who continually sought others to check his programs for errors, omissions and "clumsiness".

It's an instructive little section, and I immediately began to think of it in terms of testing. Egoless programming appears quite common these days: a Google search shows plenty of sites referring to the concept (particularly the "ten commandments" of egoless programming), and the code review in an Agile sprint appears the formal successor to Neumann's persistent reviews. But "egoless testing", so to speak, is perhaps not as prevalent in our thoughts and deeds as it might possibly be.

There are three scenarios that come quickly to mind where egoless testing may yield large benefits. They are common; every day episodes in our testing world, but in many cases we fail to handle them particularly well.

- Not thinking to review their work with others. Possibly a by-product of being a "lone tester" (though it could happen to any tester), the thought of having someone look over their work just doesn't occur to them. It's not part of the organisational culture, it's not a personal work habit, and so it slips by the way-side.

- A self-imposed expectation to think of everything test-related. People declaring (perhaps explicitly, perhaps not) themselves "testing authorities" and thus free from the need of having grubby hands (especially those non-testers) paw at their work.

- Under too much time pressure to review their work with others. The classic "death march", "no time to chat" scenarios that often plague testers. It's not that they wouldn't like to have their work checked, but they've got to have at least 78 test cases executed by close of business, the test report (in ISO-29119 format) written by tomorrow morning and the requirements review for next sprint done by Friday afternoon.

  Enter egoless testing. If we used the "25 words or less" format that cereal boxes were once so fond of applying to competition entries, it could possibly be summed up as this: actively review your work with others, and do likewise for them. But being the verbose character I am, the following is a slightly more voluminous list of concepts that have been largely distilled from Weinberg and flavoured with a dash of testing perspective.

- Don't identify or confuse yourself with the work. Good testing is difficult, and we won't always get it right. There WILL be omissions, errors and clumsiness in your work and that is completely natural.

- In the research / exploration / test design stages (however you do this), actively seek ideas from others. Even if you think everything is covered, it probably isn't. I'm fond of citing an example where a university student with minimal testing experience gave me some great ideas while testing a banking application.

- Once you have some test ideas or test notes, get them reviewed for errors, avenues for deeper investigation, gaps in your testing or ambiguities.

- Having work reviewed can be either a very formal or very informal process. See below for some ideas on how this might be conducted.

- Don't limit your review panel to other testers. Developers, BAs, product managers and users can all provide major insights into the product and your testing of it.

- Having your work reviewed is going to take time, but the benefits that can come from it will often far outweigh the overhead.

When we do this, we're effectively addressing any ego-driven problems that could adversely impact our testing. But not only is our testing is improved, our skills are also developed as we're exposed to new ideas and different ways of doing things, not to mention shining a light on the blind spots that haunt the edges of our thinking. We also build rapport with the software development team.

When it’s couched in such glowing terms, egoless testing sounds fantastic! So naturally you’re going to want to start practicing it as soon as possible. And when it comes to that, there are a few options available.

- Pair testing. One could argue we’re practicing egoless testing simply by working with someone else, as a feedback loop for each person naturally occurs as the session progresses.
- Conversation and reflective questioning. Talking through a tester’s work not by criticising, but gently probing their thinking and approaches in order to help them “self-discover” any weaknesses or gaps.
- Using a program that has a “review” step (e.g. JIRA). A more formal method, it can nevertheless be useful for prompting team members to review each other’s work regularly (though by making it a procedural step, reviews may devolve into “tick and flick” exercises).
- Walk-throughs of tests and/or test sessions, which could be either retrospective or in advance, for those who still favour more prescriptive testing.

So, maybe you’re not an egoless tester yet. Or maybe you occasionally do it when you have the time. However often you’re doing it, take it up a level. Help others become egoless testers. Find new or better ways of doing it. And why not try this tomorrow?

Dean is a software tester living in sunny Brisbane, Australia. Having fallen into testing around seven years ago, he subsequently stumbled across the context-driven approach several years later.

He enjoys studying new ideas (especially around testing or learning), and finds there just isn’t enough time to look at all of the good ones.

Dean blogs intermittently atwww.yesbroken.com and can be found under @deanamackenzie on Twitter.

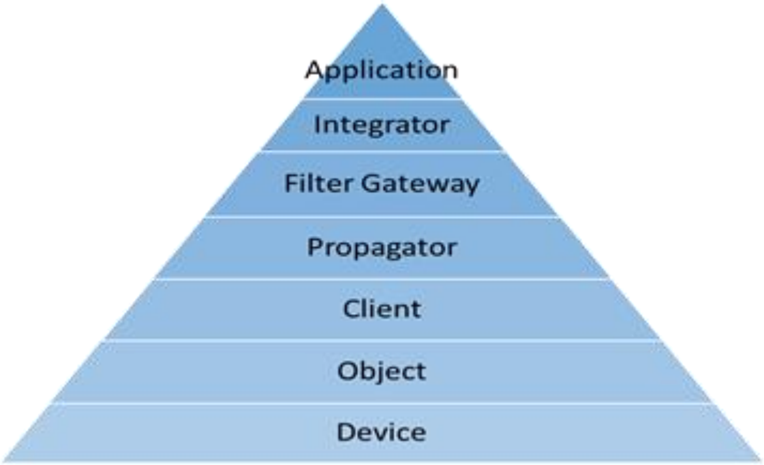
Internet of Everything – Architecture and Risks

by Paul Gerrard

IoE Architecture

A Layered Model of the IoE

In the same way that standards for networking devices at the periphery are evolving, there is no standard or conventional architecture for the IoE. However, some patterns are emerging and as might be expected, the architecture can be split into logical layers.



This seven-layer model is an amalgam of several styles of schematic that have been published. This one is device and product neutral and it is not aligned with any particular technology. In terms of scale and perspective, these layers look best drawn as a hierarchy with the device layer at the base.

In describing the individual layers, I'll present them from the bottom-up.

Level 0 – Device Level

The device itself. The device is usually a sensor or actuator and will often be a component of a larger, physical object. A device could be a motion sensor or a strain-gauge; it could be a light-sensitive diode or an electric motor. It could be a device controller chip (that manages a step motor).

Level 1 – Object Level

One or more devices may be packaged on a single board and or housing. For example, a home security sensor may contain a movement sensor, a temperature sensor, a smoke detector and so on. Alternatively, they may be wired to the object, for example, a set of sensors, distributed on a piece of machinery all wired to a single digital controller.

Level 2 – Client Level

At the client level, one or more objects are connected and may access a network through a software-enabled client. The client manages the collection of data from and distribution of messages to the objects. The client might use one (or more) distinct networking protocols to connect to (one or more) propagator nodes. The client may or may not be connected to the internet depending on the selected networking protocol and whether the device is fixed in a location or moves (in a car or mobile device).

A client might be a dedicated local network node supporting objects in fields (weather station), streets (streetlamps), a household device (a fridge or TV), a home security hub or a numerically controlled machine connected to a factory network. The client software might run on a 'smart' device but it could also be running on a general-purpose PC, laptop, tablet or mobile phone.

Level 3 – Propagator Level

The propagator node or server is connected to the internet (to connect to filter gateways) but exists in the locality of the devices being controlled. Propagator nodes are likely to be software applications running on standard hardware. The propagators will probably use a recognised operating system exposing web or web-like services for use by both clients and the remote servers that perform the gateway and integrator functions.

A propagator might be a server that monitors the temperature, rainfall and soil moisture for the fields that comprise a farm or even a district. It could collect data from and control devices that are based on streetlamps in a street or wider, local area. Propagators usually manage the devices of a select type or scope.

Level 4 – Filter Gateway Level

The filter gateway is a layer that exists primarily to ensure that only meaningful data is channelled to the Integrator functions. The gateway is, by design, connected to the internet and might be in any location. The filter might extract the meaningful or valuable or significant data from the potentially vast quantities of data collected from clients that control possibly millions of devices. A filter might also be configured to route different types of data to different integrator functions. The filter might also act as a firewall to protect the integrator system and applications from denial of service attacks.

Level 5 – Integrator Level

Whereas the propagator functions collect data of a defined scope, an integrator function might combine the climate data collected in farmers' fields with the long-term weather forecasts and farmer's calendars for ploughing, seeding, fertilising and harvesting. The integrator brings together the data from multiple sources and makes an integrated view of collected data available to applications. Integrator functions might also provide the configuration parameters to filter gateways to select the data that is routed to it by the gateway.

At the integrator level, the data might be referred to as 'big data' for the first time. This software is likely to run in a data-centre environment.

One could say this is where the data received from millions of devices is transformed, or patterns in data are derived, to create information that make sense to humans. The integrator functions enable 'information neighbourhoods' where different applications can be limited to (or focused on) a selected data domain.

Level 6 – Application Level

The application is the where the full value and capabilities of the IoE and associated big data are exploited. An application might be a central road traffic control system operated by a city council or the app, embedded in a car's computer that allows drivers to make more informed routing decisions in the journeys. IoE applications might comprise huge systems running in data-centres to free or very cheap apps running on smartphones or tablets that access data through web services.

Implementation of the Architecture

Needless to say, the various layers represent hardware devices or software systems running on devices. In a mobile phone app that monitors your exercise on a road bike, for example, you can see all seven layers in action. The GPS sensor in the phone is at level 0. The data from the sensor is integrated with an internet mapping service and the data presented through a mobile app at level 6.

In most use-cases, devices are packaged into objects that share network-connected client functionality. A single object might combine several devices such as a field weather station that collects temperature, rainfall, barometric pressure and humidity data. In a home security environment, the sensors are packaged and distributed as units through the house. A central hub located near your broadband router might combine the propagator, filter-gateway, integrator and application functions, and so on.

Regardless of implementation, the layered model can help you to understand the architecture of your system.

The Risks of Failure

In this article, I have set out a technical architecture that can be used to understand the function of each component in an IoE implementation. We'll use the layered model to look briefly at the technical risks. But there are also societal or personal risks that are being aired in the media and that we need to pay some attention to.

At this time, I only want to give a flavor of the concerns and I'll delve a little deeper into the risks that we need to pay attention to as testers in the next article in this series.

Given that I am writing for a software testing oriented audience, I will concentrate on the software aspects. However, you should understand that more and more devices will be connected to controllers and actuators that have embedded software and these cannot be separately tested. Further, many devices will be designed and packaged for outdoor use so natural hazards, roaming effects and connectivity loss will all need consideration.

Technical Risks (a non-definitive list)

Device + Object Risks (physical)

- Embedded controller failure
- Power failure or interruption
- Physical relocation and damage (frost, water, heat etc.)

Client Risks (software)

- Loss of connectivity to Propagator
- Device registration failure
- Device ID duplication
- Conflicting or ambiguous device management commands
- Security hacking, loss of control
- Device spoofing
- Denial of service/jamming

Propagator Risks (software)

- Overload from device chirps
- Loss of connectivity to clients
- Loss of connectivity to gateways/integrators
- Security hacking, loss of control
- Device spoofing
- Denial of service/jamming
- Message/command hijacking
- Failure to filter/merge/enrich data correctly

Filter Gateway risks (software/hardware)

- Home gateways shipped with default usernames/passwords for admins
- Software virus infection
- Security hacking, loss of control
- Propagator impersonation
- Denial of service/jamming
- Failure to filter/merge/enrich data correctly

Integrator risks (software)

- Flooding with bogus data
- Performance failure – data ingestion
- Performance failure – reporting overload
- Propagator connection failure
- Sparse data interpolation/interpretation
- Duplicate data load into databases
- Inconsistent distributed data load into redundant systems
- Inconsistent/incorrect/misleading distributed analyses
- Application integration failure

Application risks (software)

- Integrator integration failure
- Security hacking, loss of control
- Propagator impersonation
- Denial of service/jamming
- Of course, all of the familiar application risks of mobile apps, web services and infrastructure are in scope.

Social/Personal Risks

The risks to individuals and potentially society as a whole are only just being articulated. The overwhelming focus of attention from the infrastructure vendors and security professionals is on security risks. But the risks to privacy arising from the unauthorized merge and integration of personal data with data collected on the IoT are significant. Most individuals with insights into the IoT are extremely nervous. The wider consumer market are still somewhat unaware of the possibilities.

Complexity

The IoT is taking systems complexity into unknown territory. In the past, a systems integrator would know exactly what systems are integrated, the connections between them and the data being shared and exchanged. The security access and control policies are under their control. But with mobile systems, the border between systems that were never intended to integrate are being removed or at least blurred. This is not just a security challenge. Unforeseen interactions between devices will never have been tested and the outcomes of these, although benign in principle, remain unpredicted and unknown.

Privacy

For a single individual, their mobile device(s) might well have copies of or at least access to their home security system, their car systems, company services and potentially, their financial transaction history and bank details. As that user traverses the wireless networks of their company, shopping malls, banks, city transportation, refuse and health infrastructure, their devices are also liable to be discovered by less responsible companies or hackers. These intrusive systems might steal or corrupt your data or take control or infect your device with viruses or bots that act with your authority or bank details for fraudulent activity.

Abuse

There is an accelerating trend towards home automation, in-car and workplace automation and these bring benefits to individuals. But governments and international organisations are planning smart cities, distributed healthcare, mobile payments systems, anti-terrorism and law enforcement systems that also benefit society.

As mobile applications have been rolled out the warning signs are there that governments will exploit the big data that is captured on individuals not only to trap criminals, but also to know more about and potentially control its citizens.

Corporate Security

Companies operating highly secure IT systems may find their secure networks are connected increasingly to smart building infrastructure. Even now, an office building might have 1500 detectable sensors installed that capture services data. In their own right, these are harmless. However, many of the trusted, connected devices in the building will be running familiar operating systems. These could be hacked into and provide a trusted gateway into other IT systems. Another scenario would be that a hacker shuts down a security system prior to performing a robbery or terrorist activity.

In contexts such as transportation systems, manufacturing production lines, TV stations, energy generation/distribution and smart cities, the potential for hacking, disruption and chaos is unbounded. Isolated systems like these were not usually vetted by security specialists or IT departments. That will have to change. The problem for the industry is that most security professionals are very much oriented towards traditional IT and networked services. The IoT is a whole new ball game.

Summary

In this article, I have described what's new, technically, about the IoE and how it will scale. The architecture of the IoE is still evolving but I have tried to provide a meaningful architectural model that can help you to make sense of the technology. The model is also useful to identify some of the technical risks we need to focus on and I have listed some of the most obvious ones. Security dominates all of the concerns being articulated by industry commentators, but for non-trivial systems, significant functional and integration complexity challenges must be overcome. It all needs testing.

There are broader societal risks that the IoE brings. I have looked briefly at some of the issues that government, companies and individuals must pay some attention to.

In the next article, I'll look closer into the main risks areas of concern and present a speculative strategy for testing the Internet of Everything.

References – IoE Architecture

There are countless papers, articles, presentations and videos on the future Internet of Everything. I have only listed those that are specifically referenced in the article.

1. The Advent of Smart Systems, Services & The Internet of Things, Harbor Research, http://harborresearch.com/wp-content/uploads/2013/12/Harbor-Research_Diversified-Industrial-Newsletter.pdf
2. Rethinking the Internet of Things, Francis daCosta.
3. Open Mobile Alliance Lightweight Machine to Machine Technical Specification, http://technical.openmobilealliance.org/Technical/release_program/docs/LightweightM2M/V1_0-20131210-C/OMA-TS-LightweightM2M-V1_0-20131210-C.pdf
4. MQTT (MQ Telemetry Transport), http://mqtt.org/

References – Risks of IoE

This is a small selection of the hundreds of risk-related articles and papers that are currently available.

5. The Risks of Using Portable Devices, https://www.us-cert.gov/sites/default/files/publications/RisksOfPortableDevices.pdf
6. Heartbleed Poses Risk to Clients and the Internet of Things, Symantec, http://www.symantec.com/connect/blogs/heartbleed-poses-risk-clients-and-internet-things
7. Cyber Threats to Mobile Phones, http://www.us-cert.gov/sites/default/files/publications/cyber_threats-to_mobile_phones.pdf
8. Internet of Things (IOT): Seven enterprise risks to consider, http://searchsecurity.techtarget.com/tip/Internet-of-Things-IOT-Seven-enterprise-risks-to-consider
9. Overcoming the next generation of information security vulnerabilities, http://searchcompliance.techtarget.com/video/Overcoming-the-next-generation-of-information-security-vulnerabilities
10. Q&A: Address Internet of Things security concerns across an enterprise, http://searchmanufacturingerp.techtarget.com/feature/QA-Address-Internet-of-Things-security-concerns-across-an-enterprise
11. 'Internet of Things' holds promise, but sparks privacy concerns, http://www.dw.de/internet-of-things-holds-promise-but-sparks-privacy-concerns/a-15911207
12. The Internet of Things likely to drive an upheaval for security, http://www.computerworld.com/s/article/9248069/The_Internet_of_Things_likely_to_drive_an_upheaval_for_security
13. How Will Companies Secure the Internet of Things? http://www.fool.com/investing/general/2014/04/17/how-will-companies-secure-the-internet-of-things.aspx
14. Opportunities and Risks of the Internet of Things, http://www.innovasic.com/news/industrial-ethernet/opportunities-and-risks-of-the-internet-of-things/#sthash.IKQOVu3H.dpuf
15. With the Internet of Things, smart buildings pose big risk, http://www.computerworld.com/s/article/9248292/_With_the_Internet_of_Things_smart_buildings_pose_big_risk

Back To Index

Paul Gerrard is a consultant, teacher, author, webmaster, developer, tester, conference speaker, rowing coach and a publisher. He has conducted consulting assignments in all aspects of software testing and quality assurance, specialising in test assurance. He has presented keynote talks and tutorials at testing conferences across Europe, the USA, Australia, South Africa and occasionally won awards for them.

Educated at the universities of Oxford and Imperial College London, in 2010, Paul won the Eurostar European Testing excellence Award and in 2013, won The European Software Testing Awards (TESTA) Lifetime Achievement Award.

In 2002, Paul wrote, with Neil Thompson, "Risk-Based E-Business Testing". In 2009, Paul wrote "The Tester's Pocketbook" and in 2012, with Susan Windsor, Paul co-authored "The Business Story Pocketbook".

He is Principal of Gerrard Consulting Limited and is the host of the UK Test Management Forum and the UK Business Analysis Forum.

Mail:paul@gerrardconsulting.com | Twitter: @paul_gerrard | Web: gerrardconsulting.com

Internet of Everything - What is it and how will it affect you?

Paul Gerrard

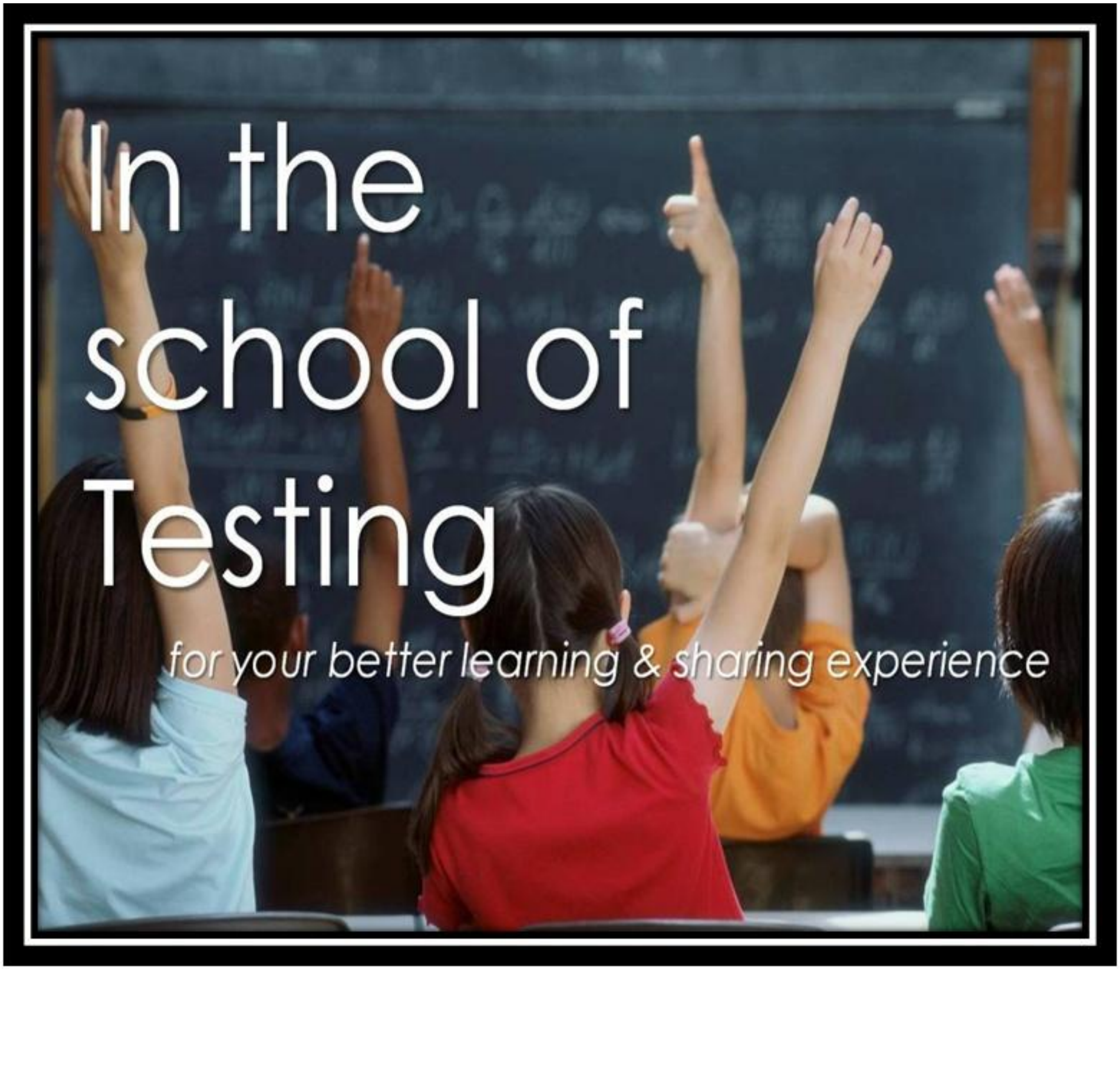
paul@gerrardconsulting.com

gerrard consulting

gerrardconsulting.com

@paul_gerrard

Click to watch it exclusively on TV FOR TESTERS!


In the
school of
Testing
for your better learning & sharing experience

Testing Lessons from the Design Thinking World

by Parimala Hariprasad

CAST is a very special conference for me (apart from Let's Test and Bug DeBug). I have several friends who have attended this conference and told me that I MUST attend this conference, even if it is at my own expenses. Two years ago, I told myself that someday I will present at CAST. My dream has come true this year. My family thinks I am crazy. For me, it's worth my time, effort and money for the wonderful testers I met at this conference.

The Key Highlights of My CAST Talk revolved around below topics.

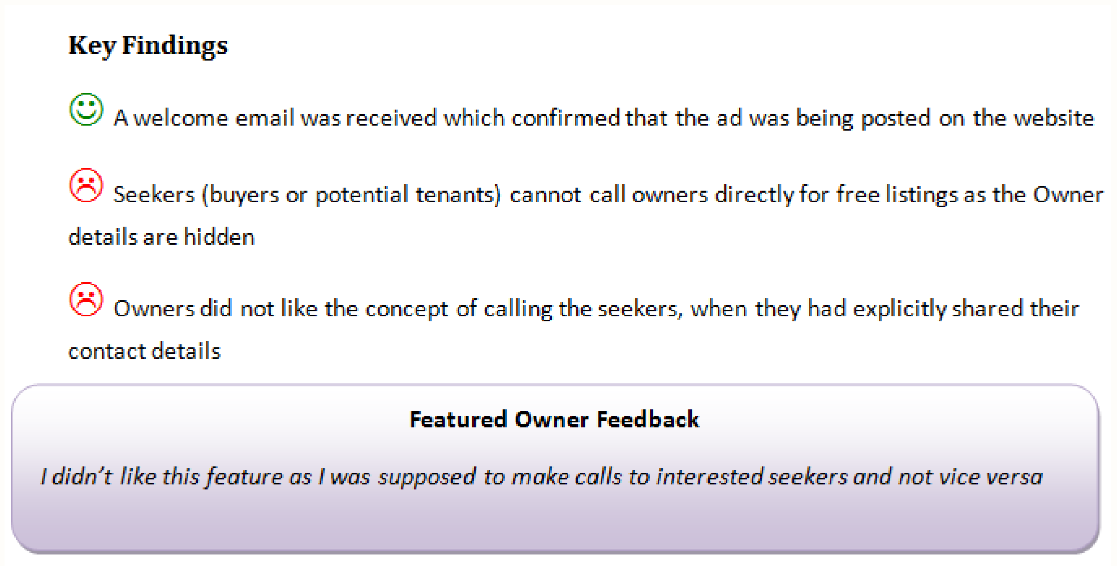
1. Emotions Testing
2. Multi-Sensory Experience
3. Testing for Errors
4. Customer Touch Points

Let me cover each one of them in detail below:

Emotions Testing

Several years ago, Don Norman was on a radio show along with designer Michael Graves. He had just criticized one of Graves' creations, the "Rooster" teapot, as being pretty to look at, but difficult to use—when a listener called in. The caller owned the Rooster. "I love my teapot," he said defensively. "When I wake up in the morning and stumble across the kitchen to make my cup of tea, it always makes me smile." His message seemed to be: "So what if it's a little difficult to use? It’s so pretty it makes me smile, and first thing in the morning, that’s most important.”

Emotions Testing is evaluating the emotional state of the user before, during and after product is used and identifying the pain points thereof. This technique can be used to evaluate the product against different emotions that a user goes through. Test results can be represented on an Emoticon Dashboard as described below:



Key Findings

☺ A welcome email was received which confirmed that the ad was being posted on the website

☹ Seekers (buyers or potential tenants) cannot call owners directly for free listings as the Owner details are hidden

☹ Owners did not like the concept of calling the seekers, when they had explicitly shared their contact details

Featured Owner Feedback

I didn’t like this feature as I was supposed to make calls to interested seekers and not vice versa

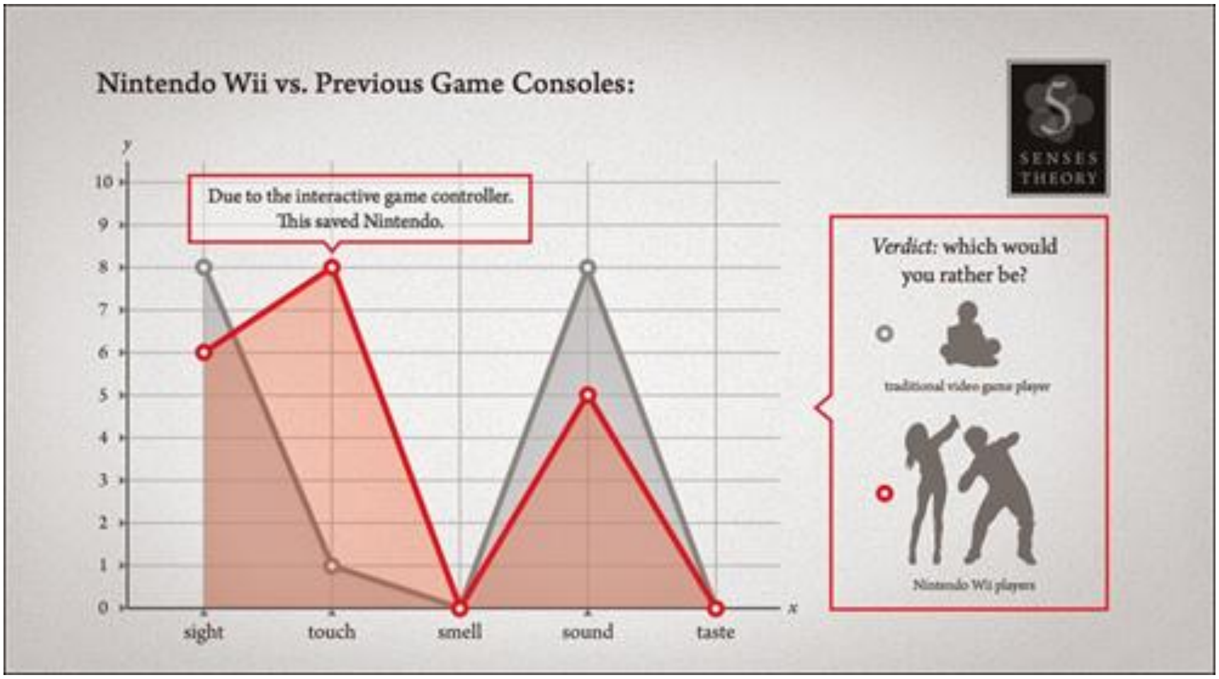
Emoticon Dashboard

What you see in the picture is a small snapshot of Emoticon Dashboard prepared after testing the product for emotions. Happy emoticon denotes good user experience and a sad emoticon denotes bad user experience. At the end of the report, you can feature a hard hitting feedback from the user highlighting the biggest pain point in the product. You can add more emoticons based on Plutchik’s Wheel of Emotions and customize it for your needs. This way you can directly communicate with the stakeholders on how the product fares on emotions testing and facilitate better decisions.

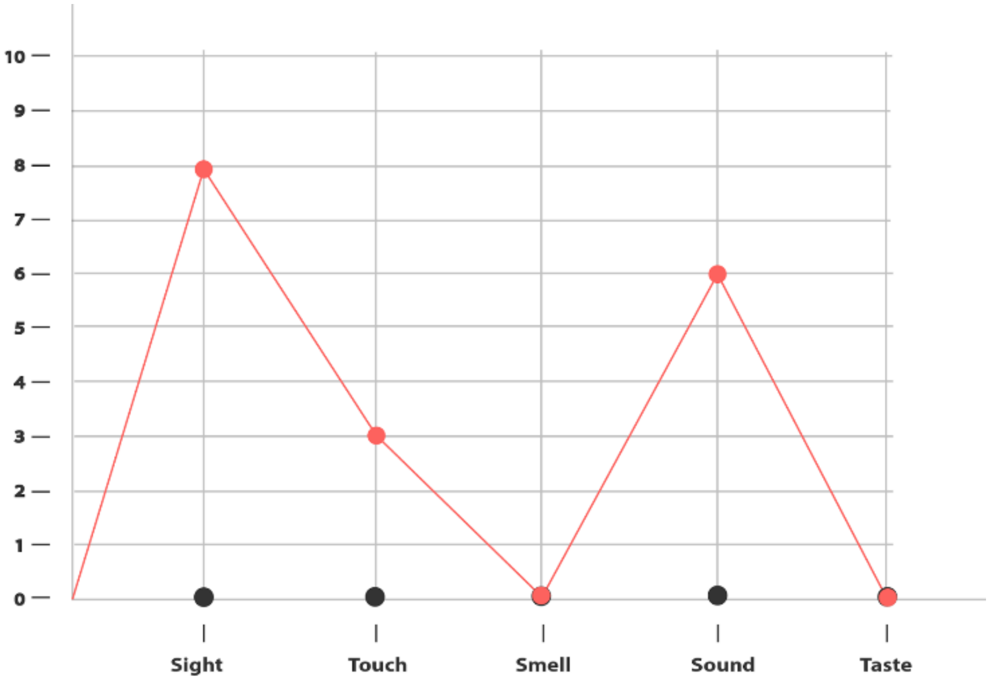
Multi-Sensory Experience Testing

Jinsop Lee, an industrial designer, believed that great design appeals to all five senses. He called this, the Five Senses Theory. Jinsop also gave a Ted talk on this topic a short while ago. According to him, one can grade any experience on all five senses. For e.g., you can grade eating noodles on sight, smell, touch, taste and sound.

Similarly, you can grade your biking experience. Jinsop graded himself on a bunch of adventure experiences like bungee jumping, playing games on two different consoles and others. The five senses graph for Jinsop’s experience on Nintendo Wii against older gaming consoles is displayed below. This clearly tells which gaming console he preferred.

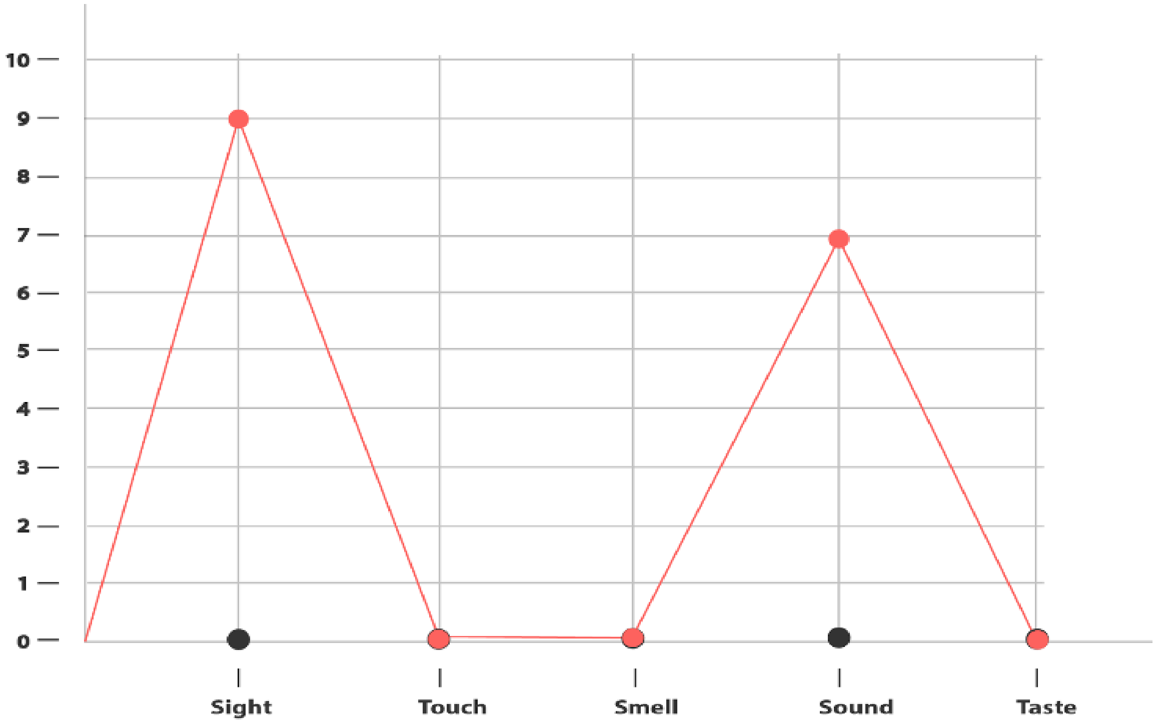


This Five Senses theory can be applied to User Experience Testing too. Users can be asked to review the applications under test and map them on a scale of 1-10 on all five senses. The broader the area covered, the better the experience. Further, this theory can be customized to rate the applications at features level or flow level. The Flow example below describes user's experience when he was using (learning) the app 'X' for the first time.



User Experience when using app X

The Feature example below shows how the user felt about 'Seen' feature on Facebook Chat window.



User Experience on "Seen" feature in Facebook

Several such experiments can be performed with users/testers to understand how different senses respond to different scenarios in applications. Like any other framework driven by individuals, Five senses theory has its limitations too:

1. It varies from person to person as everyone's senses may not work the same way.

2. All senses may not be applicable for all people. For specially-abled people, they may not even be able to hear or see.

3. It is hard to implement on large sets of users

4. For some products, all senses may not be applicable. For example, how do you rate this article for taste using this theory?

Despite its drawbacks, Five Senses Theory is a good technique to understand how products can be designed for a multi-sensory experience.

Error Messages Coverage

While speaking, we can correct ourselves if we stumble or mess up. Products and systems often do not correct themselves because they are only as intelligent as the people who built them. This leads to "slip" which is the most common error – when we intend to do one thing and accidentally do another.

Well-designed products allow us to detect slips through feedbacks. For example, in a delete operation, it is good to ask for a confirmation to verify if the user wants to proceed. If that operation is irrevocable, it is better to warn him of the consequence and take his consent. A general heuristic is to never take away control from the use

Error messages coverage can be refined by adapting some lessons from the design world. This refining can be achieved at multiple levels as listed below:

Errors-based Scenarios Testing

Testers could get a list of all error messages programmed into the product and design scenarios for each and every error message

Negative Testing

Negative testing is not the same as error messages testing. In error messages testing you start with known error handling and test it. This is essentially "positive" testing for error handling code. In negative testing, however, you think differently. Negative testing means to "negate" required conditions. In other words, you consider all the things that the programmer/designer requires for his code, and then systematically block those conditions. Example: the program needs memory, so reduce memory.

Recoverability Testing Matrix

Every error message needs to be tested for Recoverability.

- Visibility to the user of what was done
- Do/Show/Tell them what went wrong
- How the user can reverse unwanted outcome
- If reversibility is not possible, indicate to the user what needs to be done next

Some Examples

Failure Usability Heuristic by Ben Simo

Error Elimination Testing by David Greenlees

Feedback Parser by Santhosh Tuppad

Customer Touch points Testing

A customer touch point describes the interface of a product/service with customers/users before, during and after a transaction (adapted from Wikipedia). Several times, a user is not just unhappy with the product/service, but customer touch points too. As a user, think of yourself. How many times did you stop yourself from visiting that supermarket whose attendant didn’t pay attention to your questions? How did you feel when you were the first to enter a pharmacy, yet the pharmacist served customers who came after you? Supportability factors go a long way in defining customer touch points and how they feel about the product and the organization in general.

Supportability Factors

What are the factors creating product loyalty? Are they in the product itself? Price? Color? Quality? Quantity? Others? Great user experience is a sum total of all the above listed aspects including supportability factors listed below:

- Calls / SMSs
- Email
- Chat
- Service Requests
- Feedback
- Field Visits

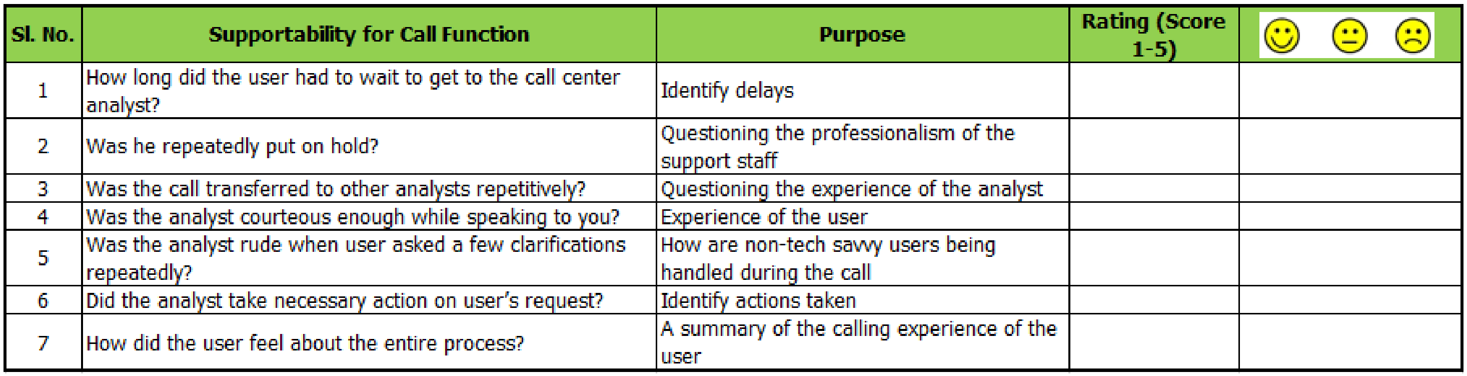
If a user calls customer care and is put on hold for 30 minutes, he would not like it. If he gets 20 messages a day on his phone after buying a product, he would be annoyed. If an email complaint from the user is never acknowledged or responded to, he would never complain again to the organization. He would complain in public over the internet by writing his story of poor experience, with a wider reach and visibility and hence, discouraging other users to buy the same product/service.

If an organization provides a chat channel but a support personnel is not available to support customers, this damages the reputation of the organization. From time to time, users might raise service requests (or tickets) to solve problems on the products they are using. If the tier1 and tier2 analysts don't know anything about the service requests they handle, it becomes a showdown of sorts. How are user feedbacks handled also lends credibility to the organizations' care model for the users. When a field service technician from the organization visits the user's site to fix the product or replace it, user's experience with the technician goes a long way in defining whether the user will remain loyal or not.

How to measure Customer Touch points?

Supportability factors are the key to determine whether customer touch points are good or bad, if the products are making a good impact or not or if the customers/users are sticking around or saying goodbye. Products have to be tested for supportability factors to measure users' experience at several touch points. How do testers test for it? There is a way.

Testers can come up with a survey questionnaire that asks questions about different support factors. Let's take an example of the activation process for a new network connection on a cell phone. User needs to call the call center, provide details to them and get the network activated which usually takes up to 24 hours in India. We could come up with a list of questions like:



Sl. No.	Supportability for Call Function	Purpose	Rating (Score 1-5)	🙂 😐 🙁
1	How long did the user had to wait to get to the call center analyst?	Identify delays		
2	Was he repeatedly put on hold?	Questioning the professionalism of the support staff		
3	Was the call transferred to other analysts repetitively?	Questioning the experience of the analyst		
4	Was the analyst courteous enough while speaking to you?	Experience of the user		
5	Was the analyst rude when user asked a few clarifications repeatedly?	How are non-tech savvy users being handled during the call		
6	Did the analyst take necessary action on user's request?	Identify actions taken		
7	How did the user feel about the entire process?	A summary of the calling experience of the user		

Back To Index

These questions might seem unrelated to testing but they are testing related in reality, because if we don’t test the processes in the organization that serve products and product users, excellent products might fade away in a short time period. Apple became the Apple it is today because of the time and money they spent on every little detail associated with the product – be it the product itself or the packaging, the color of the product, support experience and so on. It is becoming increasingly important to create good customer touch points because users are no longer looking just for products that serve their needs, but also for engaging experiences while using the product.

Lessons I learned from the Design Thinking World

I LEARNED something that brought meaning to my testing style.

1. I learned that, Design helps me test better
2. I learned that I can help the designer by raising questions and imagining a great variety of scenarios for the designer to consider
3. I learned that Design, like Testing, is an open-ended and human process
4. I learned that Design, like Testing, is often considered abstract and impractical, even though it is very real and important
5. I learned that I need to “Step Very Carefully” when giving suggestions to designers about design, because they may feel you are an amateur who is criticizing things you don't understand. Therefore, state your suggestions respectfully, remind them that design matters are their decisions rather than yours, and consider posing them as questions rather than statements. “

Summary

It was exciting to present at CAST. I was particularly happy because many testers in my session appreciated the design side of testing and told me that they would try and implement some of these ideas on their projects. Ben Simo told me that he has been doing something similar at his work and it’s not so easy. That lightened up my spirits because I know how difficult it is to advocate design lessons in the testing world.

The most exciting part was meeting passionate testers from that part of the globe, many of whom I have been interacting on twitter and other social media. The best part of our conversations was when we put challenging problems on the table and discussed how each one of us had solved similar problems. I learned so much from these discussions. I learned that similar challenges exist everywhere in testing and they need to be solved head-on. Conferences like CAST only make difficult challenges look easier to solve. That is why it’s so special!

Parimala Hariprasad spent her youth studying people and philosophy. By the time she got to work, she was able to put those learnings to help train skilled testers. She has worked as a tester for close to 12 years in domains like CRM, Security, e-Commerce and Healthcare. Her expertise lies in test coaching, test delivery excellence and creating great teams which ultimately fired her as the teams became self-sufficient. She has experienced the transition from Web to Mobile and emphasizes the need for Design Thinking in testing. She frequently rants on her blog, Curious Tester (http://curioustester.blogspot.com).

She tweets at @CuriousTester and can be found on Linked In at http://in.linkedin.com/in/parimalahariprasad. She currently serves as Delivery Director at PASS Technologies.

The Art of Localisation Testing

by Ashwin Arora

Every organization aims at developing globally successful quality product within the cost and time parameters. Software testing being a major component takes considerable time during the software development cycle and hence a significant cost is attached to it. Talking further about the global adaptability of software i.e. Internationalization & Localization of product in multiple locales possesses a huge challenge in terms of limiting the time and cost attached to Software testing. 'How much' and 'Too much' localization testing has always been a big question for all the Global software Organizations.

During the course of my engagement in planning the localization testing for our clients, certain approaches have been devised to cut down the time and hence cost. Before we start discussing them, let's try to find out the basic expectations related to the quality of any Localized Product, which are as follows.

- Software should be entirely localized (as applicable) correctly in respective locales
- Cultural Appropriateness should be met
- UI Correctness
- Functionally, the quality of a localized product should exactly match its counterpart product in English

The first three points above are related to Linguistic testing and the last one indicates Functional testing. The associated challenges are as follows –

- Linguistic experts are chosen from experienced translators. It is required to train Linguistic Experts on how to install and use the product

- Network bandwidth and time is consumed while downloading localized build onto the linguistic expert's globally distributed sites
- False alarms may be raised by Linguistic Expert on functional issues as they may have inadequate knowledge about the product functionalities
- Owing to chances of escape defects and false alarms raised can have an impact on the turnaround time

The effective solutions to these challenges is to have separate team of Linguistic and Functional testers; as most often Linguistic testers are language experts but not the qualified Quality Engineers to perform functional testing. Secondly, functional testers could provide Linguistic experts, the product screenshots/screen-casts in English and in the respective languages instead of downloading heavy builds at their ends. This would definitely result in a drastic cut down in the turnaround time and there may be a significant improvement in the localized product quality.

Comparative Analysis:

	Pre-Requisite		Knowledge / Experience			
	Need Product Screenshots	Need Product Build	About Installation	About Product	About Testing	About Linguistic Languages
Linguistic Testing	✓					✓
Functional Testing on Localized Builds		✓	✓	✓	✓	

Now, we will be discussing about the Localization Functional Testing which is the most challenging, costly and debatable affair. Reducing cost without compromising with the quality is a key here as there are multiple locales, which needs to be tested in a limited timeframe. I have mentioned a word 'debatable' above as Quality managers have diverse perspectives in terms of identifying the appropriate coverage for each locale. The conservative & safe approach is to execute all the available test cases on each of the locales or locales/platforms combinations separately; even though as per the Internationalization Engineering, only the associated resources/strings are localized and being fitted into the same code (and off course there could be exception of hard coded strings). Here comes a big question: If English version of the product has been completely tested - Is there a need to re-execute all the functional test cases on multiple localized versions separately? Even if Organization chooses to do this – consider the amount of cost or no. of resources required in order to meet the deadlines. Hence, appropriate & effective optimization or the distribution of testing among locales is a key to reduce the turnaround time and cost without compromising the quality. To summarize - the challenges are as follows:

- Multiple localized versions of products
- There could be multiple O.S platforms and hence numerous locales & platform combinations.
- Windows or MAC (or both are supported)
- There could be additional hardware requirements like Net-books, or specific printers etc.

Here are the approaches that would prove as effective indicators and could be customized as per the needs.

- Leverage test efforts across platforms within each language group
    - Business language group
    - Character encoding language group
- Provide at least one complete pass across all locales
- Provide at least one complete pass across all platforms
- Provide more coverage to locales which have highest business priority
- Maximize coverage on platforms which are widely used (in any locale or region)
- Provide coverage via sanity or Adhoc testing pass on few locales (as required)
- Distribute test cases across locales where character set is same
- Talk to developer and identify the hardcoded or exceptionally handled areas. And, execute the related test cases on all locales
- Check if there is any language specific processing in the code or not
- Try to find out if functional testing on Windows or Mac could be leveraged

Before I conclude, it is worthwhile to note that above mentioned points are only the suggestions and there is no standardized process to perform the Optimization as things may change from one product to another or one Organization to another.

I hope this article would prove useful to you. Thanks for your time.

Ashwin Arora is a Sr. Software Quality Engineering Manager at QA InfoTech Pvt. Ltd. Ashwin joined QA InfoTech in 2005 and currently he is managing multiple QA & testing teams.

He has a good experience in QA project management with client-focused delivery; and, maintaining a track record of releasing high quality products within proposed schedule and budget. He takes a keen interest in test optimization and proposing quality oriented cost optimized test solutions to clients.

Mr. Arora earned a ‘Bachelor of Information Systems degree with Honours’ from Guru Gobind Singh Indraprastha University.

Road to Test Automation

- Part 2

- by Georgios Kogketsof

Code Efficiency the road to a framework creation

In my previous article I focused on how to use an abstraction layering model to create maintainable code for the automation test scripts as shown in Figure 1.

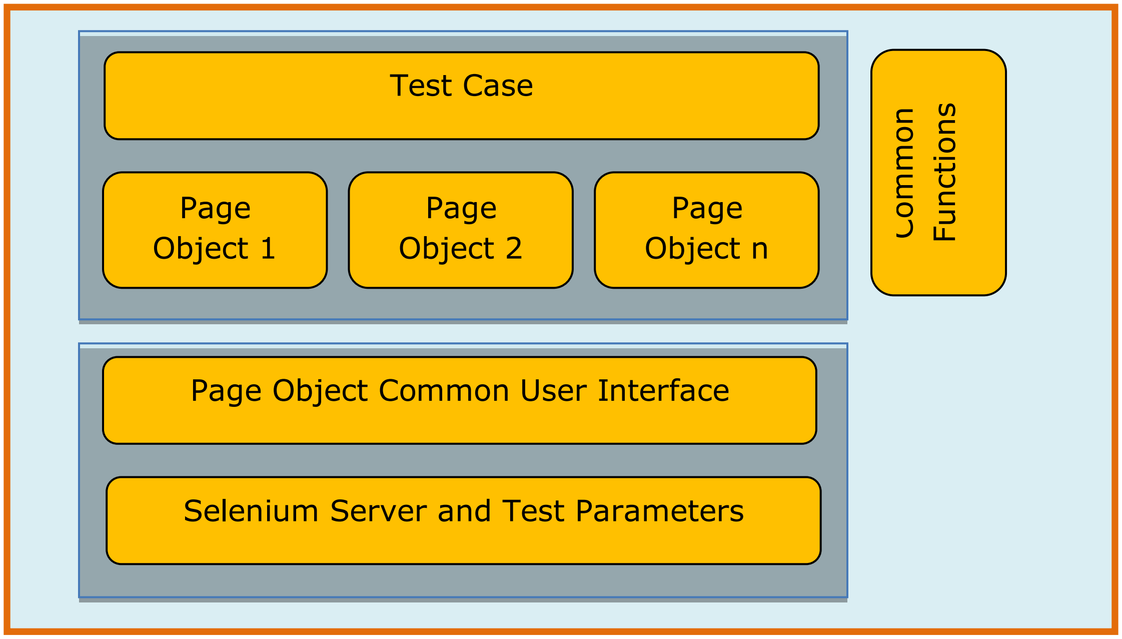

Test Case
Page Object 1
Page Object 2
Page Object n
Common Functions
Page Object Common User Interface
Selenium Server and Test Parameters

FIGURE 1 ABSTRACTION LAYERING MODEL

In this article I will elaborate more on what each layer means and the need for its existence. Moreover, I will try to show how each layer is handled in the context of a test suite and guide towards the creation of a framework.

The proposed model shown previously has two distinct blocks independent from each other. The distinction between the bocks becomes apparent in the context of a framework in which the bottom part remains the same for every test case script and the top changes with each test case.

The 'Selenium / Webdriver Server and Test Parameters' layer is created once in the context of a test suite and used every time we need to create an automated script. Because of the fact the 'Selenium / Webdriver Server and Test Parameters' layer is created only once the implementation of the layer should be in the context of a framework rather of a test suite.

In the 'Selenium / Webdriver Server and Test Parameters' layerwe include code to initialize the selenium session and our test parameters such as logging user name and passwords, timeouts and page urls. The following piece of pseudo-code shows how we can declare a selenium session. Assuming that every test script follows the structure (@BeforeClass, @Tests, @AfterClass) representing the test case pre-conditions, the actual test steps and the test case post-conditions then at the @BeforeClass of every test case we have the following piece of code shown in Figure 2:

```
protected Selenium selenium;

@BeforeClass
public void startSelenium(){
   this.selenium = new DefaultSelenium(host,port.browser,url);
   this.selenium.start();
}
```

FIGURE 2 SELENIUM SESSION PSEUDO CODE

In the context of a test suite for each test script the selenium session should be initialized using the code shown in Figure 2. The repetition of the initialization code could create a maintainability issue repeating the same piece of code in more than one place. With the proposed layering the repetition issue is resolved by placing the initialization code to the 'Selenium Server and Test Parameters' layer thus placing the initialization code in a single location. Access to selenium session is gained by passing *this.selenium*as argument to every function needed.

In pseudo code every function handling selenium commands should be in the form shown in Figure 3.

```
myMethod(this.selenium, this.params)
```

FIGURE 3 PSEUDO CODE

Although this is a perfectly working solution, passing the selenium session and the test parameters to every method is inefficient. To create a more efficient design we need to set some specs to simplify and accelerate the development process:

- The session and the test parameters should be passed as a private copy so there will be no conflict on parallel test execution
- This private copy should reside inside some sort of storage and be accessible by simple set and get methods
- The storage should be accessible from any method within the context of a thread

An easy solution that fits the above specs is to use Threadlocal, so the selenium session and the test parameters are statically stored locally to a thread.

The 'Page objects common user interface' layer is used for interfacing with the page objects layer. In this layer all the common functions that appear in the page objects are included. These functions are actions made by the user (click, input, select, get value etc.) and the mask the appropriate selenium / web driver command along with a logic behind the action. For example the method click in this layer should include the following pseudo-code:

- Wait for element to appear
- Click
- Wait for Ajax to finish
- Wait for a specific action like: table addition, element to appear, next page to load etc.

As with the 'Selenium / Webdriver Server and Test Parameters' layer we can create the 'Page objects common user interface' layer once in the context of a framework and use it as needed in a test suite.

The 'Page Objects' layer is where all the page objects reside. Because of the simplicity of the page objects class, code efficiency is limited to simple development techniques. One of the techniques my test team uses to create page object classes is to define all of my element locations in an enum structure. In the structure included are two methods one for setting static element addresses and one for the dynamic element addresses. A concrete example of a page object class is shown in Figure 4:

```
public class MyPageObject {

publicenumCssLocators {

        BUTTON_APPLY_CHANGE("css=…']"),

        TABLE("css=div[id$=''{0}''] table tbodytr"),


        /** The my locator. */

        private String myLocator;


        /**

        * Instantiates a newcss locators.

        *@paramcssLocator the css locator

        */

        CssLocators(String cssLocator) {

        myLocator = cssLocator;

        }


        /**

        * Gets the css.

        *@return the css

        */

        public String getCss() {

        returnmyLocator;

        }
```

FIGURE 4 PAGE OBJECT CLASS

The difference of this layer with the previous ones is that it cannot be created only once in the context of the framework because the page objects are different for each project under test thus the distinction in the layer blocks shown in Figure .

Implementing the aforementioned layers in the context of a framework require above average development skills. Knowledge of patterns such as builder pattern and techniques such as Threadlocal could be valuable in accelerating the implementation process creating the foundation for an in house framework. In house developed frameworks help the automation testers speed up testing eliminating "down time" to debug automation scripts.

As an automation tester it is crucial to me to have my in house developed framework rather to use a test record and replay software.

To be continued in next issue…

Georgios Kogketsof is a full time test engineer. In his 15 years of testing experience has worked in multinational, multicultural testing projects for major corporations in the defense industry and in digital marketing. George's expertise as a developer and as a tester fused in creating a hybrid model of a test engineer proved to be extremely effective in automation testing. Over the years George and his testing team have developed a methodology for creating fast, maintainable scripts for automation testing web apps. George is proud in his involvement in the creation of the open source testing framework Stevia.

Visit his blog at: http://seleniumtestingworld.blogspot.gr/

Get Stevia at: https://github.com/persado/stevia

CONTINUED FROM PAGE 13...

You have worked within organisations as well as an independent test consultant. What difference does it make on one's perception about quality in either of the roles?

Two advantages in working as a consultant: a fresh perspective and not getting involved in politics.

A fresh perspective is easier to have when you're new and you haven't been through past software releases. Without the background build-up, it's easier to see some issues and obstacles. It's also easier to think of solutions that existing staff may no longer think are possible.

Without having to worry about performance reviews, salary evaluations or long-term relationships (at least not in the same way), it is easier to speak up and not spend as much time and energy worrying about stepping on toes. Also it seems to me my observations are less skewed by "who" as I don't have past history with staff and have no reason to be thinking or perceiving issues in any personal way.

Which job is more challenging and why?

Both employees and consultants face obstacles, I think it's a matter of figuring out which set of obstacles you're best equipped to deal with.

Being a consultant means you're expected to show up and be ready, experienced and able to offer solutions quickly. There is a constant pressure for performance, adding value and showing that you are already able to deal with the issues at hand. If you like faster variety and don't mind working alone and not feeling as much a part of a team, consulting may work well for you. Being an employee means you're expected to be thinking longer range for the team and the company. There is an expectation that you can "play" well with others and that you will find a diplomatic way to handle issues and get along – you will need that over the long haul. While there is an expectation that you should be ready to do the work stuff, there is more understanding that the company may need to vest in your ongoing learning so while there is pressure to perform, you have a bit more breathing room. If you like feeling part of a team, like having co-workers you get to know and consistency in your day, employment may be a better option.

Would you like to tell us more about Professional Tester's Manifesto you have created and what testers can do to change things for better?

At CAST 2014, an interesting turn of events took place. I wrote my thoughts very clearly in a blog post and I would encourage readers to read that post. You can find that blog post here.

As for making a change in our profession, I would encourage testers who asked to be certified by their companies to raise questions, be discerning consumer students. I would advise any testers not to pursue a certification *if* you do not believe in certification because doing so – while possibly "pleasing" an employer helps to fortify the current state of certification – so if you don't believe in it, don't do it.

I was there in your session at CAST 2014 conference. Needless to mention that I learned a lot. How was your overall experience with the conference?

The CAST conference was a good conference; I'm saying that based on the variety of different topics and speakers who presented. I think one thing about the conference that I did not like was seeing so many people who are new in the field not comfortable in approaching or talking to some of the folks (who like myself) have been around a while. It is important to meet new people and to exchange ideas. This is one of the reasons I talked about the pitfalls of being a celebrity tester, you can find the interview here.

Thanks for your time, Karen. We enjoyed talking to you.

Happiness is....

Taking a break and reading about **testing**!!!

Like our FACEBOOK page for more of such happiness

https://www.facebook.com/TtimewidTesters

The Promise

Our focus as testers is on finding issues in the product/application. And it is a good goal to pursue. Step back a little and think... Is there a greater goal? If so what is it?

A product is sold on the promises that marketing makes to its customer base. And a custom application is also sold on the promise of the value it delivers to the business. The greater goal is “The Promise”. And these promises are based the expectation of end users.

When we test a system, we are in fact checking the unwritten promise of the development. That of ascertaining if there are defects that may derail quality and break the promise of good working software. But that alone is not enough. As end users we have expectations that have been built on the promises by marketing folks. And these go beyond the mere good working software. These are about ‘how it enhances my experience and could make my life better’. My life at home or office depending on the product!

So we need to elevate our thinking to a higher level. As one who is responsible for, “Ensure that we deliver on the promise” of ensuring that the expectations are indeed met. That expectations are meaningful and are indeed value adding.

But it is difficult due to the bias. The bias of knowing deeply about the system. When we test a system, we get to know the system in detail and over time "get used to" the kinks. The kinks that may ultimately derail the promise. The kinks that messes up expectations.

The difficulty is in shifting to the view of being an end user and then evaluating the promise. The difficulty in quietly getting acclimatised to challenges in implementation, to becoming numb to the lack of clarity of the pristine promise and therefore not doing a stringent job as expected. Kind of like the "Stockholm syndrome". And this is but natural. So what can we do about this?

How can we get this job done? Let us step back a little and see how we get any job done. We can get a job done by (1)Doing it ourselves or (2) Facilitating to get it done. In this case the 'doing it by ourselves' seems to pose the challenge due to bias. So maybe we should resort to (2). By facilitating to getting this done.

Before I get to the details, let me explain the genesis of this article. In a recent customer engagement with a startup company, the conceiver of the product was the founder and CEO of the company. He was keen that the testing team understands his vision of the product that he has been communicating to his prospective customers and investors so that they can deliver his vision. And he spent a good half day explaining his vision to the team before they commenced testing.

And after a few weeks testing, when the test team had a review with him, he was keenly questioning as to the high level scenarios related to his promise. And I could see that test team was finding this difficult. Difficult to clearly elucidate how they are assessing his promise because some of the key information to evaluate the promise were indeed unclear/missing. And the test team did not know how to fill this. And the test team thought that it was solely their responsibility to find this information. And this information was related to kinds of information used in a query, their usage patterns, data volume and finally perception of patience by the end user.

And then is when it dawned that being a facilitator would help. Facilitate by enabling the product conceiver (in this case the CEO) to make the promise testable. Promises do commence by being vague initially, and it is the job of the test team to crystallise it. By building a straw-man, and then working with key stakeholders to identify the elements (of the promise) and then making it clear.

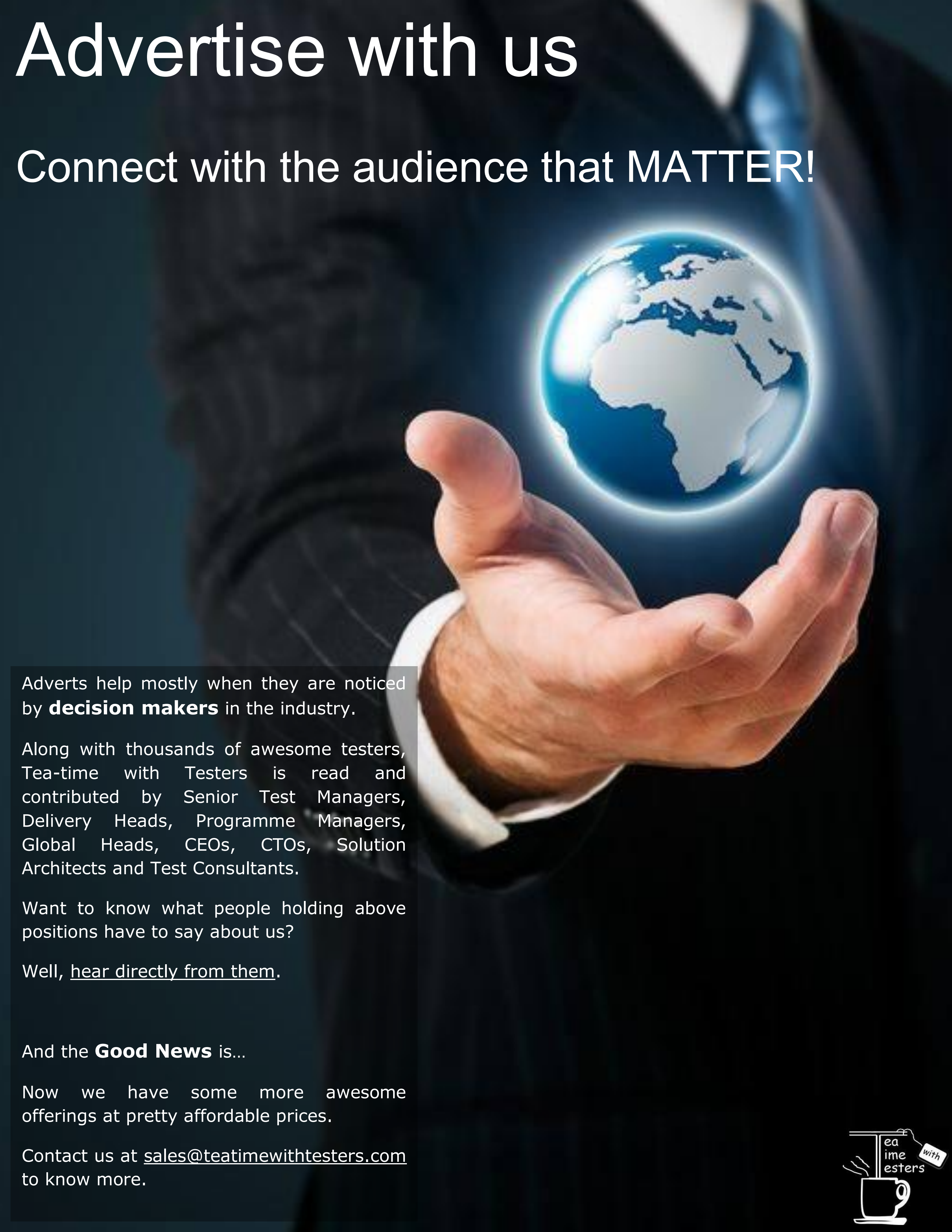
It is not the activity of doing it oneself, but by actively facilitating the thought process. And this is very iterative. In this case, this could be identifying elements of the usage, the situation in which this would be used, the deployment over the next 6-12 months and therefore the potential volume. Being a tester who is very aware of the product and its current limitations, it is necessary to understand this bias and therefore let the other stakeholders figure this out. And play the active role of a facilitator. Of dis-engaging and ensuring that biases do not come in the way. Of elevating and become mature. It is about "The promise".

So next time, do look for defects but keep the larger goal in vision. And remember that it is not always about "doing", but also about "facilitating".

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**



Advertise with us

Connect with the audience that MATTER!

Adverts help mostly when they are noticed by **decision makers** in the industry.

Along with thousands of awesome testers, Tea-time with Testers is read and contributed by Senior Test Managers, Delivery Heads, Programme Managers, Global Heads, CEOs, CTOs, Solution Architects and Test Consultants.

Want to know what people holding above positions have to say about us?

Well, hear directly from them.

And the **Good News** is…

Now we have some more awesome offerings at pretty affordable prices.

Contact us at sales@teatimewithtesters.com to know more.

Every Tester
who reads Tea-time with Testers,
Recommends it to friends and colleagues .
What About You ?

Image : vernhart

in nexxt issue

articles by -

Jerry Weinberg

Wayne Yaddow

T Ashok

Sanjay Gupta

...and others

our family

Founder & Editor:

Lalitkumar Bhamare (Pune, 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 – Anu @ My Dream Canvas

Core Team:

Dr.Meeta Prakash (Bangalore, India)

Unmesh Gundecha (Pune,India)

Dr. Meeta Prakash

Unmesh Gundecha

Online Collaboration:

Shweta Daiv (Pune, India)

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

facebook.

Follow us on

www.teatimewithtesters.com

Give Feedback