

THE INTERNATIONAL MONTHLY FOR NEXT GENERATION TESTERS

Tea-time with Testers

SEPTEMBER 2013 | YEAR 3 ISSUE VIII

Women
in
Testing
special
&

Over a Cup of Tea with Leah Stockley



The Conference for Thinking Testers

December 14-15, 2013 | New Delhi - India

ThinkTest talks

Don't miss the chance to hear from legends like James Bach and other experts in industry...



JAMES BACH



MICHAEL LARSEN



SMITA MISHRA



LALIT BHAMARE



AJAY BALAMURUGADAS



MEETA PRAKASH



T ASHOK



MUKESH SHARMA

Click [HERE](#) for more details

ping

PRESENTED BY



Prudential

a benefit ping-pong party for Per Scholas



October 24 / 7pm - 9pm

SPiN New York / 48 E. 23rd St. NYC

\$125 - General Play / \$300 - Doubles Tournament Play

GENEROUSLY SUPPORTED BY

PIMCO

NASDAQ OMX

BARCLAYS

Capital One



TEA-TIME WITH TESTERS

First Indian Testing Magazine to reach 101 Countries in the world !

Created and Published by:

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

Editorial and Advertising Enquiries:

Email: editor@teatimewithtesters.com
Pratik: (+91) 9819013139
Lalit: (+91) 8275562299

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

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

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

Editorial

By our guest – Leah Stockley



I feel very honoured to be asked to guest edit this amazing bumper edition of TTWT. I would have been for any edition, but the fact this one celebrates the contribution of women to the testing field makes it a little closer to my heart.

To be fair, I feel women have always been fairly well represented in Testing, as opposed to other areas of Technology. A friend once told me it's because we're great at picking holes in other people's work!! Outrageous...! But at its essence perhaps he had a point, although I would prefer to say that women often possess great attention to detail, clearly an essential skill for a good tester. Perhaps our wider peripheral vision also helps...!

But women bring far more to testing than this. They bring empathy, people skills and the ability to communicate, not to mention technical savvy and excellent organisational skills... And let me just clarify that I am not in any way stating that male testers are not capable of these strengths, of course they are. I honestly don't feel the need to differentiate between men and women. To me it is always about merit. If you work for me and have the right skills, attitude and do the job well, you will get the recognition you deserve. I don't care about your background, your gender or whether you have blue spots!

So then I'm surprised to have recently found myself working with an exceptionally talented group of ladies, many of whom have contributed to this edition of TTWT. We meet (virtually, as we are spread across the globe) to discuss how we can encourage more ladies to rise up the ranks in testing, and especially to be represented on stage at conferences. We all agree there are a good proportion of women in testing, but we want to make sure we can set an example, and inspire some of those ladies who may be at the early stages of their career to realise they have the same opportunities for growth as men, and help them build the confidence to seek out those opportunities.

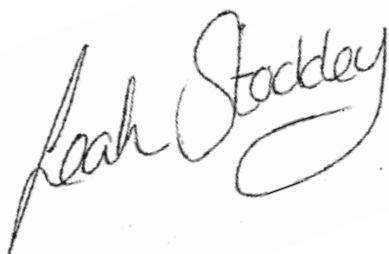
Personally, I've only started building that confidence to stand up and be seen in the last year or so. It started with my first attempt at blogging... the feedback made me realise that people are interested in hearing the experiences of others. That boosted my confidence to consider talking at a testing conference. I did that for the first time in December 2012. There I had mixed feedback on my presentation style. But I took that as learning and it spurred me on, and helped me realise how I could improve the next presentation I made. I had to take that first step in order to be able to improve. Living in Singapore I have struggled to travel to the major conferences, or even the many tester meet-ups I was reading about. So instead of continually complaining that there was nothing for us testers here in Singapore... I set up the Singapore Testing Forum! Now every 2 months, 60+ testers are gathering in a room for a couple of hours to learn more about testing from others. And yes, I've spoken a couple of times myself and happy to say I'm getting much better feedback. 12 months ago I would never have believed it if you told me I would start a blog, start a testing forum in my city and stand up to deliver 3 public presentations before the year was out! It's amazing what a little action can do... and how supportive people are of anyone who tries.

I said earlier that women are well represented in testing, but I have to admit there are global variations. When I look at our team in India, we have a fair representation of female testers, but disappointingly, no female test managers at all! That may well be due to cultural differences that can put a greater domestic burden on women, but as recognised figures such as Dr. Meeta Prakash and Smita Mishra show us; despite this, women can still rise up and have the same success as their male peers. We may face some discrimination (I have recently) but it is important to realise that comes from a weakness and insecurity on the part of the person dishing it out. We can choose to ignore their behaviour, in fact be flattered. If someone is taking time to pay you that negative attention, it's possible you are a threat to them. So stay positive and focussed and you will continue moving forwards in spite of them.

Perhaps the hardest part is, knowing where to start? My advice would be to seek out a mentor. Many companies have a mentor scheme available. If yours doesn't, then take a look online or in your local community. If you are specifically looking to improve your public speaking www.Alineattheladies.org offer a mentor service, but there are many other resources available to you. Find someone you admire, someone who has skills you would love to develop, and approach them to see if they would mentor you. I did this myself recently. I was nervous to ask, as I felt that person would be too busy. But I did my research and approached her with the exact reasons I wanted to be mentored by her. Actually she was flattered that I asked and said yes. My mentor is a woman, but that wasn't deliberate, and I would happily choose a man as a mentor also. It is important for anyone wishing to develop their career to seek a senior sponsor/ mentor. And I don't mean someone who sits on the promotion board, but someone who can help guide and challenge you to define and reach your goals.

So when the TTWt team approached me to be involved with this edition, it felt like the perfect time to say yes. My first talk at the Singapore Test Forum was entitled 'The rise of the Thinking Tester'. I truly believe we are approaching a crossroads where more and more testers are becoming proud of our skills and are gaining the confidence to stand up for our profession. We want to improve our reputation by leaving projects in no doubt of the value we add. And we can do that by supporting each other and making sure we continually learn and get better at our jobs! It is evident from TTWt content that the editorial team also recognises the importance of continuous improvement & knowledge sharing. They dedicate their own personal time to deliver this thought provoking magazine to your inbox every month... let that thought inspire you... and know you can learn from it, regardless of whether you are female, male or of the blue-spotted variety!

Enjoy Reading!



Leah Stockley
Inspiredtester.com





Editorial

What's making News?

Tea & Testing with Jerry Weinberg

Speaking Tester's Mind

The Women in Power... Nah! The Power of Women – 14

Why Software Testing is a Great Career for Women – 16

What does it mean to be a Women in Testing? – 23

What Leadership means to Me.. – 27

All The World's a Stage - 29

Women and Conferences – 31

Shentreprenuer - One in a third billion - 33

In the School of Testing

Learning to Test – 39

Prepare for Promotion Now! – 42

Peer Pressure – 45

Be A Student for Life – 49

Police is your Friend - 53

T ' Talks

Change. Continuous motion. Cadence -67

A cup of Tea with Leah Stockley

Family de Tea-time with Testers



WeTest Weekend Workshops

WeTest Weekend Workshops - Saturday, November 30, 2013

New Zealand has a wide culture of development led weekend events, but until recently there was little out there for testers who want to develop their craft. A major influence within Wellington has been the emergence of WeTest peer conferences, which have provided a great forum for people to network and share their experiences.

However peer conferences can be daunting for those new to them, and indeed numbers tend to be limited. It soon became obvious that there was an appetite for an event which could be inclusive but affordable. The WeTest Weekend workshops - sponsored by consultancy Assurity - are an affordable mixture of an afternoon of discussions and workshops around testing led by testers representing the many IT companies from around Wellington. Their aim is to explore and challenge ideas and techniques whilst providing a forum for networking within the Wellington test community.

For more details, contact - mike_talks@hotmail.com



Tea & Testing



with

Jerry Weinberg

Managing Others – The Manager's Job (Part 2)

Tactics for relieving overload

Here are some actions my clients have taken to help relieve this problem of overloading their most experienced people:

- Block the telephones of the most experienced people so they cannot receive incoming calls.
- Find offices for them out of the main traffic pattern, with doors they can lock from the inside. Provide them with "Do not disturb" signs.
- Put managers near the offices of the experienced people, and post signs directing people to see the manager, rather than the overloaded person. The managers watch for violations of this direction and act to educate the violators.
- Give the experienced people a strong-willed assistant who will reroute all calls and other interruptions. A good assistant can also watch for opportunities to offload tasks. Instruct the assistant, "Your job is to see that this person has maximum uninterrupted time to work."

Perhaps the most important action is to give the inexperienced people things to keep them busy—things that won't require the experts' time—so they'll stay out of the more experienced people's hair and become experienced people themselves.

Listening

A manager is responsible for listening to why it should not be done at all, why it should be done by someone else, or why it should be done in a different way than what you've prescribed.

This is a perfect description of super reasonable *listening without hearing*. This manager knows the answer, and just needs to wait until the other person stops talking. This model says that managers *prescribe* the way things should be done (Pattern 2), rather than *describe* what outcomes are desired (Pattern 3).

The blaming undercurrent to this super reasonable statement shows us that this manager perceives most of these objections to be bogus. Most of them are. Anyone who manages by prescription will hear lots of bogus objections. The objections are not to the particular prescription, but to the *act of prescribing*. If the manager is not in the prescription business, the need for bogus objections seldom arises.

Why would a sensible manager assume such a non-listening, prescriptive position? Not wanting to be a manager in the first place certainly contributes to the need to retain technical direction, but that's not the whole story. Some managers, after all, are able to let go of their technical expertise and actually manage. The super reasonable tone suggests another alternative: excessive ego involvement.

Author Tom Crum relates the parable of an inventor who is so ego-involved that he cannot trust the help of others for business assistance. As a result, ...his invention takes on such importance that his life becomes a secret mission and he is constantly wary of everyone, thinking them potential enemies out to usurp his plans and ideas.

As a result of this inability to share information, the invention is a commercial failure. In contrast, when he learns to trust others as parts of his "team," ...he is extending himself beyond his own talents and becoming richer in ability.

When he perceives his support people as an extension of himself, all information gets shared. Can you apply this parable to computer programmers? A major job of the software engineering manager is to develop openness and trust among all the workers contributing to successful software. What better way to foster openness and trust than by being a model listener?

Following Up

A manager is responsible for following up to see if the job has been done correctly, discovering that it hasn't, and listening to some of the world's worst excuses from the employee who should have done it.

"Excuses" is a blaming manager's term. Following up is done to help the worker do the job, and for this you must listen to reasons, not label them excuses. The reasons may not be correct, but so what? Right or wrong, the reasons always contain the information an effective manager needs to steer the organization. The manager who made the above statement does not understand much about how to extract such information.

Evaluating Quality

A manager is responsible for following up again to see if the job has been done, only to discover that it has been done incorrectly, but deciding that you'd better leave it as it is because it's as good as you're likely to get.

This describes a pure placating position, one that builds resentment and lowers quality.

The Pattern 3 manager does not evaluate quality, but puts in place the processes, not the people, to evaluate quality. Examples of such processes are quality assurance, customer satisfaction surveys, and technical reviews. When the manager *does* evaluate the quality of work, the main purpose of that evaluation is to help the employees develop their own skills. The most important feedback to your employees is actions, not words.

My colleague, Richard Cohen tells the following story:

At the end of a project in 1980, the project manager gave all of us 'Hero' awards and meant it when he said he thought we were all heroes. I had another manager, who I liked a lot better, who felt that he was a failure if we had to work overtime more than a few times a month.

The action that counts most is managing well in order to create the right conditions for work, rather than giving rewards for producing acceptable quality—perhaps while working under suboptimal management conditions.

Personnel Decisions

A manager is responsible for wondering if you ought to get rid of the person who can't seem to get the job done correctly, but deciding that the successor is most likely to be just as bad—maybe even worse.

This description stems directly from the One-Dimensional Selection Model. This manager sees an all-or-none universe, in which there are good people (like the manager, presumably) who should be retained and bad people who should be fired.

This, indeed, is the typical view of newly appointed managers:

When asked to describe what it meant to be a manager, nearly all of the managers began by discussing management's rights and privileges, not its duties. They generally began by stating explicitly that being a manager means being the boss. ... They routinely spoke of only two kinds of people-management decisions: hiring and firing subordinates.

To be continued in next issue...

Biography

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



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

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

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

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

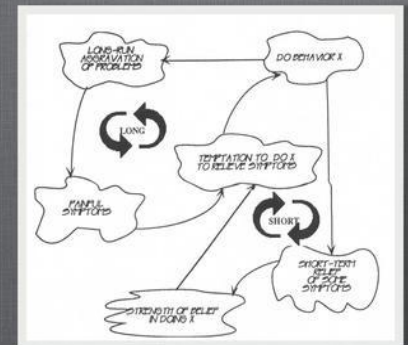
MANAGING YOURSELF AND OTHERS is another famous book written by Jerry.

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

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

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

MANAGING YOURSELF & OTHERS



GERALD M. WEINBERG

TTWT Rating: ★★★★★

A photograph of a green, conical pendulum bob hanging from a thin wire. The bob is positioned over a surface of light-colored sand. In the sand, directly beneath the bob, is a circular pattern of concentric, slightly raised ridges, resembling a ripple in water or a footprint. The entire scene is framed by a dark blue border.

Speaking Tester's Mind

- straight from the author's desk



The Women in Power.... Nah! The Power of Women!!

by Dr.Meeta Prakash

When Lalit discussed with me this idea of having a dedicated issue towards women in testing, I got so excited. This was about a dedicated issue to showcase who are the women and what they can do in challenging environment of today in technology field. He went on to publically request articles by contribution and we were overwhelmed by the kind of mails we received. From there on to here we are publishing this edition with a special editorial by Leah and contributory articles by so many recognized women in testing field.

In general people have a mythical belief that it is difficult for women to be strong on technical side. But believe me, it is just a myth ☺. In fact we are capable of doing much beyond an average male can (yes! I am being gender biased and generic here ☺). There have been focused studies that have proved that women are exceptionally strong in multi-tasking as compared to men. We are also exceptionally good at juggling our various hats that signify roles of a professional / daughter / sibling/ parent/ etc. etc. we seamlessly move in between activities and tasks and take things to closure with an ease across multiple challenging situations at the same time. ...and that's the power of the women!

Coming back to the field of testing, let me first start with myself. My entry into software testing was not by chance but a choice driven by destiny. This led me to tread various paths academically with excellence, ending with a PhD in Software testability. While the world was trying to break 'the testing' as just another activity within various software development model; I was quietly studying and getting trained on the art that some people had already initiated. Yes, I am talking about Jerry Weinberg / Cem Kaner / James Bach / Meyers. These were first few writers who really invoked the interest in me to do what I do today as my soul hearted passion. And YES! I feel proud when I stand up to say – Testing is what makes me going!

While I am enjoying the passion I have faced; I realized some challenges too.

Some of the key challenges I have faced as a woman in Testing (with what I feel about them) are-

Perceptions – Managing perceptions is one of the top most list. Each time you sit facing a man in the opposite chair; you feel they are trying to access somewhere behind the back of their mind if you can actually do what you say you can. This does not let the people-combination give the best and work like a true passion driven team.

Women can't code – In fact with our critical analytical ability we are better than the most. Especially in testing as it requires us to do many a things at same go. In fact, in general I feel that any tester knows more than what a developer might know. Cause the most difficult thing to do in life is to criticize other person's work and find gaps/mistakes/faults.

They do not understand technology – If you belong to same school of thought; please engage with us more often. This is a myth. The projects are run as efficiently as it can by anyone. Quick learning and adaptability come naturally to women.

They will have liabilities – yes, indeed. As we have the ability to do much more than an average individual can. We have the capability to balance the act well with multitasking across our professional and personal lives.

There are multitudes of examples across the industry where we find women giving as well as achieving the distinction; breaking the myths of limiting thought process. **Women in line** is such an initiative that is being driven to help out those people especially women who are trying to outstand and outshine but need a little more support to be able to step out there. This edition is dedicated to all of them and the upcoming ones too. We have some lovely contributory articles compiled and I hope you all will enjoy this edition of Tea-time with Testers, with a difference.



Dr. Meeta Prakash is a passionate tester who strongly believes in leadership by example. She is a PhD in software testability. Has 14+ years of experience in Software Testing and Automation. She has worked on multiple domains and technologies across product and service industry in IT. She has extensive experience in program management, project management, people management and defect handling. She also has experience in the Global Delivery Model of IT Industry, Offshore Development Centre and the Build Operate Transfer Model.

Test techniques and test process improvement are her key interest areas. She is trained by some of the best teachers in software testing industry and is an active member of international testing community.

She blogs at <http://testingthetestable.blogspot.com/> and she can be reached at <http://twitter.com/meetaprakash>

Why Software Testing is a Great Career for Women...



by Lisa Crispin

I was honored when asked to contribute to this issue on women in testing. My only concern is that the women who read this probably already ARE in testing. When I got my first job in software development, there were as many women as men in our department. Diversity in the software industry has declined dramatically since then. How do we reach women so that we can explain how much testing has to offer as an exciting, rewarding career?

For the past several years, I've looked for ways to attract more women to software development. In 2009, I helped Mike Sutton with an Agile Alliance-sponsored effort to introduce awesome women in agile to the world. We conducted some video [interviews](#) with women who were introduced to us: Marlena Compton, Lori Hylan-Cho, Sharon Robson, Daniella de Leon and Ashley Wali, and showed them at the Agile 2009 conference. I personally feel this led to more recognition for the contributions of leading women agile practitioners.

Since then, I haven't participated in any organized efforts to make our profession more diverse. However, I take every opportunity to talk to women who are thinking about a career change, and explain how much opportunity software development, and testing in particular, can offer. I support efforts to get girls and women interested in software, such as [Girls who Code](#), [Programming Diversity](#), and [Nairobi Dev School](#). I wish I had more time to participate in events such as the [Grace Hopper Celebration of Women in Computing](#). I've joined our local [Women Who Code](#) group, but haven't had time yet to attend any meetings. I'll get there, though!

What Women May Not Know about Software Testing

Since you, the reader, are likely already a tester, I depend on you to encourage women to consider testing as a career. If you know someone who's already a tester, but is perhaps in the wrong job right now and feeling discouraged, show them the potential that's out there. Here are some points I always try to help women understand if they are even thinking about a career in testing, or software in general.

It's all about people

I think one reason a lot of women dropped out of the software industry was that they got shoved into a cube where they were expected to mindlessly execute scripted manual tests, or write code as quickly as possible to specifications into which they had no input. The vision of introverted coders heads-down at the keyboard is probably something that turned off a lot of women.

Good news! Software projects succeed when they have lots of good people who work together. Testers play a key role in getting the right people together to discuss how they can delight the customers. For example, if the team is uncertain about how a particular feature should behave, a tester can grab a coder and a business expert to talk it over. In our book *Agile Testing: A Practical Guide for Testers and Agile Teams*, Janet Gregory and I call this the "Power of Three". George Dinwiddie has an even better name for it, "[The Three Amigos](#)".

Especially on teams practicing Agile development, lines continue to blur, and the level of collaboration goes up. Lately, for the first time, the testers on my team have been able to work together with our designers. We look at early mock-ups together, come up with test cases, keep each other informed of changes, and pair to do acceptance testing. My team pairs for all development activities and many testing activities. If you don't like people, don't get into software testing!

One of the aspects of my job I enjoy most is working with people from a variety of backgrounds, with diverse experience. I work on a distributed team, so I learn about what's going on in different parts of the continent (or the world, on my previous team). It helps me open my mind to new possibilities and viewpoints.

It's a creative process

When I was in college, punch cards were still in vogue. There was no way I was going to get any more involved with computers than I had to! To study computer science at my university, you had to major in engineering. Like many people even today, I perceived software development as a left-brained activity, all analytical and requiring math. I did not like math, and math did not like me!

A couple of years out of school, I was hired as a programmer/analyst trainee because of my domain knowledge about business. I discovered that there's no one right way to develop software, and it's more of an art and a craft than a science. In my experience, testing requires lots of creativity. In his conference presentation "[Artful Testing](#)", Zeger van Hese demonstrates the amount of artfulness and

creativity that goes into testing software. Chris McMahon [compares software development with performing](#) music or other creative arts.

I can give plenty of examples of creative activities from my daily job. Exploring a new software feature, thinking about what users will do with it, imagining worst-case scenarios, these things stimulate the creative juices! We're always brainstorming about strategies for testing a new UI, for ways to better capture and communicate requirements, for documentation approaches that work better for ourselves and our customers. We participate in design discussions. Brainstorming and designing means lots of drawing on a whiteboard (real or virtual).

On our team, designers and testers have created different personas representing users. Each persona has a name, a back story, even a picture. This helps us think of testing charters to reflect real-life scenarios for how people will use our software.

Writing code is a creative process, too. My job has often involved automating tests, and test code is, well, code. Writing useful, maintainable automated tests takes lots of imagination and experimentation.

[Testing is fun!](#)

My workday always involves lots of play. We laugh at a funny video, we joke over snacks, we engage in riotous ping-pong matches and video games. There's a page on our wiki where team members occasionally post entertaining photos of teammates.

Most importantly, we enjoy our work. We're allowed to do our best work. If we run into an obstacle, our team will help us overcome it. It's a good feeling to discover an overlooked requirement, or a piece of code that doesn't function quite right, well before we release a new feature to production. Our programmers thank us for helping them deliver the best possible code. Our customers are often delighted with our product, and tell us so.

Technology always moves fast, so there's always something new to learn. This is a blessing and a curse. If you devote time to professional growth and honing your craft, you'll always have marketable skills and good career opportunities. The only problem is that it can be tiring to always have something new to learn! But it's certainly never boring.

[Good testers are rewarded](#)

I work in the Denver, Colorado area, where there is a high demand for testers who have good "T-shaped skills": they're technically aware, they can draw a high-level representation of the system they test, they can communicate with programmers. They know good ways to do effective exploratory testing, they know how to plan testing activities alongside development, they can collaborate with coders to automate tests. They are well-paid, but more importantly, they can get jobs where they have autonomy to manage their own work and collaborate with teammates in other roles.

It's true that you may have to look hard to find the right team to join. Too many companies today still don't understand that a focus on software quality is the only way a development organization can keep up with the changing demands of business. But there are good, smart companies out there who get it.

I think a lot of women who opted out of a software development career were stuck at companies who mandated one "death march" after another. Teams with too much [technical debt](#) try to overcome the problem by making people work 80-hour weeks. I don't know why more women than men on those teams thought to themselves, "this is nuts, I have a life!", but we all need to honor our values and priorities.

If you find a team that values quality, you'll be able to enjoy the rewards of contributing as a tester, while working at a sustainable pace.

How to get started

There are so many online resources, mostly free, to help you get started learning how to test software. Online communities such as Software Testing Club are a good place to start your research, and even have discussion forums on the topic of [self-education](#). The online [BBST Foundations](#) courses have an excellent reputation. You can practice testing alongside experienced testers at [Weekend Testing](#) sessions. If you live in a sizeable metropolitan area, there are probably testing user groups and meet-ups you can join. You can find these and more ideas about [self-education for testers](#) in Markus Gartner's [webinar](#) slides.

In some countries, it can be hard to get an entry-level testing job without some kind of certification. But do your homework before starting a certification program. Too many of them are set up to generate revenue for companies without giving students any useful skills. [Elisabeth Hendrickson](#) has some words of wisdom about certification programs.

Talk to your friends!

Do you know a woman who's just starting out professionally, or is not happy with her current career? Tell her what it's like to be a software tester. Tell her how satisfied you were the last time you found a tricky bug that would have caused harm in production. Walk her through a typical day in your job. Find out what she thinks she knows about being a tester, and explain some of the surprising things about testing we've discussed here.

This doesn't only apply to women, of course. We need all kinds of people, from all different backgrounds, to make our teams stronger and our software better.

I'm told that the trend is starting to reverse itself, that more women are enrolling in software-related courses at the college level, and that numbers of women in software are going up. That's great to hear, because it can be lonely to be the only woman, or one of only two or three in an office! We need to do more to educate women about software testing, shake up mis-informed stereotypical views of who a software tester might be, and encourage them to give testing a try.

And how about you?

How are you feeling about your own career? I've met testers who feel beaten down. There are just so many companies that don't understand the value of building quality into software. It's easy to fall into the trap of thinking this is just how it is.

I've certainly worked in situations where I felt frustrated by a corporate culture that doesn't value quality. I applied patterns from [Fearless Change](#) to try to address pain points. My testing team and I found ways to add value. But sometimes a cultural problem is hard to fight. If you have valuable skills, you can vote with your feet. Look for a company that wants what you have to offer.

Not long after I started my software career, a co-worker gave me great advice: "Job security is all in your mind. If you think you can get a good job, you feel secure". I took those words to heart, and invested energy in keeping my skills up to date and my "network" well-populated. I hope you will, too. And please try to add more awesome women testers to that network!

Lisa Crispin is the co-author, with Janet Gregory, of *Agile Testing: A Practical Guide for Testers and Agile Teams* (Addison-Wesley, 2009), co-author with Tip House of *Extreme Testing* (Addison-Wesley, 2002) and a contributor to *Beautiful Testing* (O'Reilly, 2009). She has worked as a tester on agile teams for the past ten years, and enjoys sharing her experiences via writing, presenting, teaching and participating in agile testing communities around the world.

Lisa was named one of the 13 Women of Influence in testing by *Software Test & Performance* magazine. For more about Lisa's work, visit www.lisacrispin.com.

Lisa can be contacted on Twitter @lisacrispin



[Back To Index](#)



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

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

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

“The Guiding Star”

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



E-mail your question on –

theguidingstar@teatimewithtesters.com

Please Note :

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

Result of our Ideal Reader Contest is here.....

And the
Winners
are...



Christy Henitha

Kays Mak

Wasif Ahmed

Rajesh Krishnan

Moath Zuhud

congratulations!

Winners - please contact us before Oct 15th 2013 to claim your prize.

What does it mean to be a woman in Testing?



by Johanna Rothman

Well, I don't know what it means to be a man in testing! But I can provide some perspective on how I have seen the field change over my 30+ years in software.

Back in the 70's

I was a developer in the 70's. I don't remember seeing a separate organization for testing. We tested our own code, and asked our colleagues to test our code for us. Because I worked in assembly language, the amount of code I could write at any time was small. So the amount I tested was small too. Computers and memory was small. Everything was small. Things were easier, in the bad old days.

Because everything was small, the problems were more manageable. Oh, we still solved hard problems, but we solved them in simpler ways. The architecture was simpler. The UI was simpler. (We didn't have windowing systems. We had command lines.) Systems were easier to develop and test.

I was one of the few women on my team. There were few women managers, but I didn't realize that. I was having fun, writing code, developing systems, breaking my code and my friends' code.

Into the 80's

I started using proprietary message-passing languages, a pre-cursor to object oriented languages in 1980. I learned several, as a developer, still testing my own code. I was working for small companies at the time.

In 1985, I joined Symbolics as a tester. They offered me a job as a tester because I didn't know Lisp.

I quickly learned to shift my thinking from "how do I solve this problem?" to "how do I break this code?" A total mind shift.

At first, my peers at Symbolics didn't have much respect for testers. That's because the first test manager thought we could do capture/replay. He had hired a couple of smart manual testers with no technical skill, and a couple of people like me, who were developers, but had no Lisp experience. Yikes!

Well, you all know how the capture/replay dream has yet to be realized. But we did develop some guidelines for when we should do manual or automated tests.

Since I was a developer by training, I wanted to automate everything. That wasn't cost effective. How can you automate installation, for example? But you can automate much of file system testing. Or network testing. Or disk testing. Yes, we had proprietary hardware, and I tested interfaces. I quickly gained a reputation as the person who would try just about anything, and I was evil in my testing.

Testing Needed Organization

We needed organization for the beta testing at Symbolics, and I managed that part of the project for several releases, while testing. Then I became a program manager and then a functional manager and then a director.

What's the Role of Testing?

When I was the SQA Manager, I participated in an off-site. We were all supposed to define our missions for our teams and then explain them to the rest of the managers. I stood up and said,

The Mission of the SQA Group is to provide information about the product under test.

The CEO was surprised. "What do you mean? You assure the quality of the product! That's what I pay you to do!"

"Nope, you only pay me to provide you information. You want me to assure you quality, you have to give me the developers. I can assure you all the quality you want then."

I did not "get" the developers.

Integrating Everything

I was the Director of Development Software Products when I left that company, which meant I was responsible for the development, testing, and support teams for the development product line.

At Symbolics, the developers and testers had a long history of working closely together, regardless of management. In this role, I started using many of the project and program management tips I use today: rolling wave planning, deliverable-based milestones, small cross-functional teams, monthly deliverables and more.

I guess you could say I tested myself into and out of management.

Onwards and Upwards

I took another management job at a voicemail company. This time, I was the Director of Software Verification and Support, also known as SQA and Continuing Engineering. In that job I had to hire many testers and continuing engineering folks.

If you can spell test, you can test

The 1980's were also the beginning of the decade of what I call the "dumbing-down" of testing. By the time I reached the voicemail company, I encountered many would-be testers with no technical skills.

Capture/replay was still a pipe dream. The product architecture was a distributed system, so we needed people who could write a little bit of code. Sometimes, just from an Excel spreadsheet. I had developers who would gladly write hooks into the code for testers, but they needed to be told *where* to write hooks into the code.

The testers don't need to be able to read the code, but they need to be able to describe where they wanted the hook.

We had a huge problem with multitasking. I learned how to manage the project portfolio out of necessity. If you read *Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects*, the story from the preface is from my time in this company. It's a true story.

I left that company when the new VP of Engineering decided he didn't want a pregnant Director of Software Engineering. Yes, that was illegal 21 years ago, and they paid me for the privilege. We were both happy to see me leave.

What's a Pointer?

I took a job as a contract tester to learn C++ in 1992 when my daughter was born, so I could work when I was awake. I transitioned to the SQA manager position and left after a year to start my consulting business because I didn't fit the culture of the organization.

I still saw too many "second-class testers." I wrote an article about this several years later, called No More Second Class Testers!, <http://www.jrothman.com/2004/01/no-more-second-class-testers/> after I did an assessment and I just couldn't take it anymore.

Testing Requires Creative, Problem Solving Skills

I have had my own unique career. You wouldn't want to have my career path. You want your own.

But through it all, you can see that I have sought ever-increasing challenges in the types of problems I have solved.

Whether you are a woman or a man, you must develop creative problem solving skills. I chose to develop my technical problem solving skills first, then my project management skills. Now, I am focused on my interpersonal skills.

Testers don't all require coding skills. But they do need some technical skills. Without any technical skill, they become useless to the technical team. It doesn't matter if you're agile, waterfall or something else. Software development is a technical endeavor. You need technical skills.

What agile has done for testing

Agile has brought testing out of the backwater and made it cool. Back in the Feb/March 2013 issue, I wrote an article called "How Technical Does a Tester Have to Be?" where I discussed the four

dimensions of technical skill. Only you can decide for yourself where you live on those four dimensions of functional skill, domain expertise, tools and technology, and industry expertise.

Agile has helped all of us ask, "What are our functional skills?" "How do we acquire domain expertise?" It has probably put too much emphasis on using tools and technology, but I expect that. And, agile has lessened the emphasis on industry expertise, which I think is great. In my opinion, there was too much reliance on industry expertise, which could go stale quickly, unless you were in the industry working. And, if you were working as a tester, you were not in the industry.

Ladies, and Gentlemen, my advice, for what it's worth: keep learning. If you maintain your career development, your career will pay you back.



Johanna Rothman is the author of *Hiring Geeks That Fit*, <https://leanpub.com/hiringgeeks>.

See her other books at <http://www.jrothman.com/books/>.

She writes an email newsletter, the Pragmatic Manager, <http://www.jrothman.com/pragmaticmanager/>

[Back To Index](#)

What leadership means to me...



by Anne-Marie Charrett

Occasionally, I get a glimpse of how other people perceive me. Usually, it's a simple sentence offhand remark like, "You're so quiet" or "You don't create waves". This type of remark jars with me, because I don't see myself as quiet or nice or whatever adjectives people use to describe me. In my mind, I'm much more like a Joan of Ark figure, sword waving as I charge into war!

So there is a difference in how others and I perceive myself. I don't know why this is such a revelation to me- after all, thinking logically about it, there is no reason why they should be similar. Perhaps it's because the views are so disparate, it challenges me to think about who I really am. On reflection, I like that. It's nice to have your preconceptions about yourself challenged.

I've pretty much always done what I wanted to do, especially when it comes to work. Even when I've doubted some of the decisions I've made, I like to think I own them and I try to avoid making excuses for why things don't work out the way I like.

I think this is important for any person, but it's essential if you set yourself up as someone in authority, be it a test manager or a testing expert. After all, if you are not able to stand true to yourself, what gives you the right to set yourself up as someone to be followed? Since I was knee high to a grasshopper I've done my own thing. I was brought up to believe in myself. I owe this to my dad, who used to take me out of school to go hunting on horseback at the tender age of eleven. (I always wondered why the nuns didn't catch on to the fact that I had flu every last Friday of the month). He believed that life's lessons were learned on a hunting field, and in many ways he was right. I learned that I could do things even though I was desperately afraid. (Hunting on horseback is fast and frightening!) I learned resilience.

At 13 I decided I wanted to become an engineer, and despite not initially getting the grades I needed, I managed to fulfil this goal by the time I was 23. I also wanted to travel and so I did, travelling through Europe with a backpack at the tender age of 17. My travels ended up taking me to Australia, which I've now learned to call home.

I always found people's reactions to me being an engineer as amusing. They expected me to feel marginalised and intimidated, perhaps because I'm female. I never felt that! I've always expected people to treat me as an equal, and for the most part, that's what I got.

I've always expected a lot of myself and of others, mostly because I desperately didn't want to fail at things. As a junior tester, as long as I was motivated, I tended to walk the extra mile. Management notices this type of behaviour, and ticks it up to leadership material. Personally, it was more that I didn't want to fail, than wanting to lead others.

That's not to say I wasn't ambitious. I was. I wanted recognition for my ability and my skills, and the only way I thought I could achieve that was through management. My ego isn't quite so desperate these days; I've learned that what you think about yourself is more valuable than any external recognition, although that doesn't mean I don't appreciate my time in the spotlight!

So for me, I see myself as resilient, strong and able to achieve just about anything if I want to. I do feel powerful because I've done what I've wanted. When people say I'm quiet and introverted, it seems incongruous with that. It's not really. You can feel powerful and be quiet. It's taken me a while to see that, and to accept and welcome that. I think lots of quiet people believe they are not leaders; that's simply not true. You can be a leader, you can be the leader of you, and you can be a leader of others.

One aspect to leadership that I've struggled with is being a recognised leader. While my ego thrives on those moments in the spotlight, putting myself "out there" doesn't come naturally to me. Fortunately, I've had James Bach, who has inspired me to challenge myself by being more outgoing. As James put it "We need people like you to be leaders in our community" and I agree. What keeps me striving as a leader in this area is the knowledge that I'm helping other testers to grow and achieve what they want. I get personal satisfaction when people are inspired to change something in their testing work. I find this aspect of leadership very rewarding.

In essence? Leadership for me is about service, duty, working hard at what you love and being trustworthy. Yes. I think I can live with that.



Anne-Marie Charrett is a testing coach and trainer with a passion for helping testers discover their testing strengths becoming the testers they aspire to be. Anne-Marie is director of Testing Times, a company that offers coaching, training and consulting on software testing.

She specializes in test team transformation, creating and improving test teams to become powerhouses of testing excellence. She does this by focusing on a tester's skill and empowering them to make the changes themselves, securing long lasting change.

Anne-Marie can be found on twitter at @charrett & blogs at <http://mavericktester.com>.

A background image of red stage curtains, partially drawn, framing the title text.

All the World's a Stage

by Claire Moss

At Dragon Con, a large local science, fiction, fantasy, puppetry, writing, all-the-things conference, I was watching a Star Trek panel where a young, black man came up to the microphone during the time for audience questions. As a fan of the show, he told the two black men in the group what an inspiration they have been to him: strong black men in space. I had never really considered this aspect of Star Trek before, having only had casual contact with the franchise myself. (And, yes, I'm already working on fixing that particular character flaw through regular doses of Star Trek episodes.)

Following closely on the heels of a conversation of feminism and women in software testing, this exchange fascinated me. How could a person so loosely related to another be such an inspiration? And what did this mean for me as a female speaker on software testing?

I have recently been solicited to join a list of female speakers on technology. At first, I was a bit insulted. If you want me up on a stage for my body, then guess again because I'm not that kind of girl! And what does my gender have to do with my worthiness as a speaker anyway?

Having heard the words of this sci-fi fan, I see another possibility. Could my presence on stage be an inspiration to someone who identifies with me rather than another speaker? And if it could, is that cause more worthy than my pride and my concerns about being chosen as the token woman?

Recently, a guy friend had been describing his exploration of feminist thought to me and that began to

open me up to different perspectives about this solicitation. He mentioned that conference organizers are getting pressure to gender balance the speaker roster. I would rather be judged solely on merit. But, if social pressure makes gender an inevitable factor, would I rather know about it or have it happening behind the scenes? And if I'm already meritorious, is gender a reasonable tie-breaker?

Now I'm not one for boasting, so I'm basing the evaluation of my speaking abilities on the feedback I received from the conferences where I've appeared this year. Given that the feedback was very positive, it seems reasonable that someone might select me for another conference in the future if I throw my hat in the ring again. And if all publicity is good publicity, then getting my name out there more would improve my odds. So it seems all signs would point to yes.

And yet I'm left wondering what is most beneficial to not just other women in software testing but the craft itself. Do I seek to inspire another up-and-coming woman to do great things in this field? Do I want to promote myself? Or do I simply want to contribute to excellence in our field agnostic of the gender producing the insights?

What would you do?

[Back To Index](#) 



Claire Moss has always had a passion for writing, which might be a strange trait for a discrete mathematician, but that doesn't stop her from blogging or writing testing articles. After working briefly as a software programmer during college, Claire signed on as a quality engineer after graduation. By now, Claire has been testing software for 10 years.

When you find your calling, you never look back! You might say she's a compulsive empiricist when it comes to software. Claire continues to use her evil powers for good on the job and on her blog.

Contact Claire on - aclairefication.com



Women and Conferences

by Lorinda Brandon



A funny thing happened while I built my career. While I was successful and built a good reputation within the sphere I occupied, in one sense the tech world was passing me by. As I rose up the ranks, I got pummeled and discouraged by the sheer maleness of the industry but I plowed on, believing that I was forging the way for all the other women behind me. And then I took a position as an evangelist, where a lot of my job was to go to tech conferences and either speak or mingle or both. And then's when it hit me. I wasn't forging a path for the other women behind me... because there were no women behind me.

It's not news that we are faced with a distinct shortage of women in the tech industry and an even greater shortage of them in tech conferences. *Many explanations have been offered* about why there are no women attendees or speakers, most of them leveled at incompetence or insensitivity on the part of the conference organizers. I am in the camp that believes the issue is larger than the organizers – it begins with women themselves and the lack of priority they put on being out there, mingling with the rest of their industry.

As I went to one developer conference after another, I started to really feel the isolation of it. It became habit to count the women in the room at every session I went to – sadly, it was an easy count because it never hit the double digits. And I started to understand the magnitude of what it really meant – I was learning a lot at these conferences. I was watching new products launch, old technologies debunked, and concepts being debated by the leaders in the industry. But all of those people talking and most of the people listening, were men. *So I began to tweet about the situation* from the darkened rooms I was sitting in and that began a whole other learning cycle for me. Here's what happened – my laments about the lack of women at these conferences were often retweeted by the men at the conference. Men replied to me with as much concern as I was voicing. And then I started to find myself in conversations with conference organizers, all of whom were obviously discouraged by their inability to bring more women to their conference.

Across the board, the feeling was that they needed more women speakers if they were going to bring in more women attendees. That may be true, although I think it might be easier to tackle it from the ground up – get more women to attend and let them naturally gravitate toward the stage. But they

were all focused on getting more women speakers in an effort to provide role models and a female voice that would appeal to more female attendees. The reality they were facing was that fewer women submit to speak and of those that get accepted, too many of them cancel before the event.

But it's not just about women speakers. First, we need women attendees. Conferences are where you learn about what your industry is talking about, where the thought leaders are going. Conferences are where you meet other people in your industry that you would otherwise never run into – you learn who they are, what they do, share some beers and some war stories... and before you know it, you're LinkedIn, they're following you on Twitter, and your own personal connection to the industry has grown substantially.

It's an old familiar conversation – the boys' network supports its own. Well, sure, any network supports its own. What we're really railing against is that the boys have a network. So, of course, in trying to overcome the power of that, we started forming our own networks when what we really needed to do was to infiltrate theirs. And here's my take – *while there are the occasional blips of bad behavior which act as a catalyst for larger complaints against the gender ratio*, in my experience most men not only would welcome women into their network, they are actively trying to attract women to the network. Do they occasionally make a misstep in their efforts (like a recent dev conference I went to that set up a "Ladies Table" at their lunch)? Yes they do. But they are trying to find the thing that will finally work and that means trying a lot of things that don't work. And women aren't helping. We need to show up. We need to engage.

I mentioned recently to a group of women that in the conferences I go to, the ones that are geared toward software testing have a much higher population of women. That is to be expected, I suppose, because testing has a much more balanced gender ratio than development does. But I am not sure why developer conferences and tester conferences are separate anyway – as time goes on, we are merging those disciplines more and more so why wouldn't we have the same concerns and conversations? Why shouldn't there be some discussions about testing at a developer conference and some development topics at a testing conference?

So, how about this for a radical suggestion? What if we got some of those women in the testing industry to attend developer conferences? I promise, they're not reading code to each other – the discussions are exciting and innovative and if you love software, you'll love being there. Let's help balance the numbers at those conferences and, even better, infiltrate the agenda. We can kill two segregation problems at once – the lack of women and the lack of testers on their stages. I know for a fact that those conference organizers would love a way out of their man caves but they can't find the path out on their own. *Women have to actually show up and help.*



For almost 30 years, **Lorinda Brandon** has worked in various management roles in the high-tech industry, including customer service, quality assurance and engineering. She has worked at some of the leading companies in the industry, including Mashery, SmartBear, RR Donnelley, EMC, Kayak Software, Exit41 and Intuit, among others.

She specializes in rejuvenating product management, quality assurance and engineering teams by re-organizing and expanding staff and refining processes used within organizations.

Follow her on Twitter @lindybrandon.

Shentreprenneur - One in a third billion

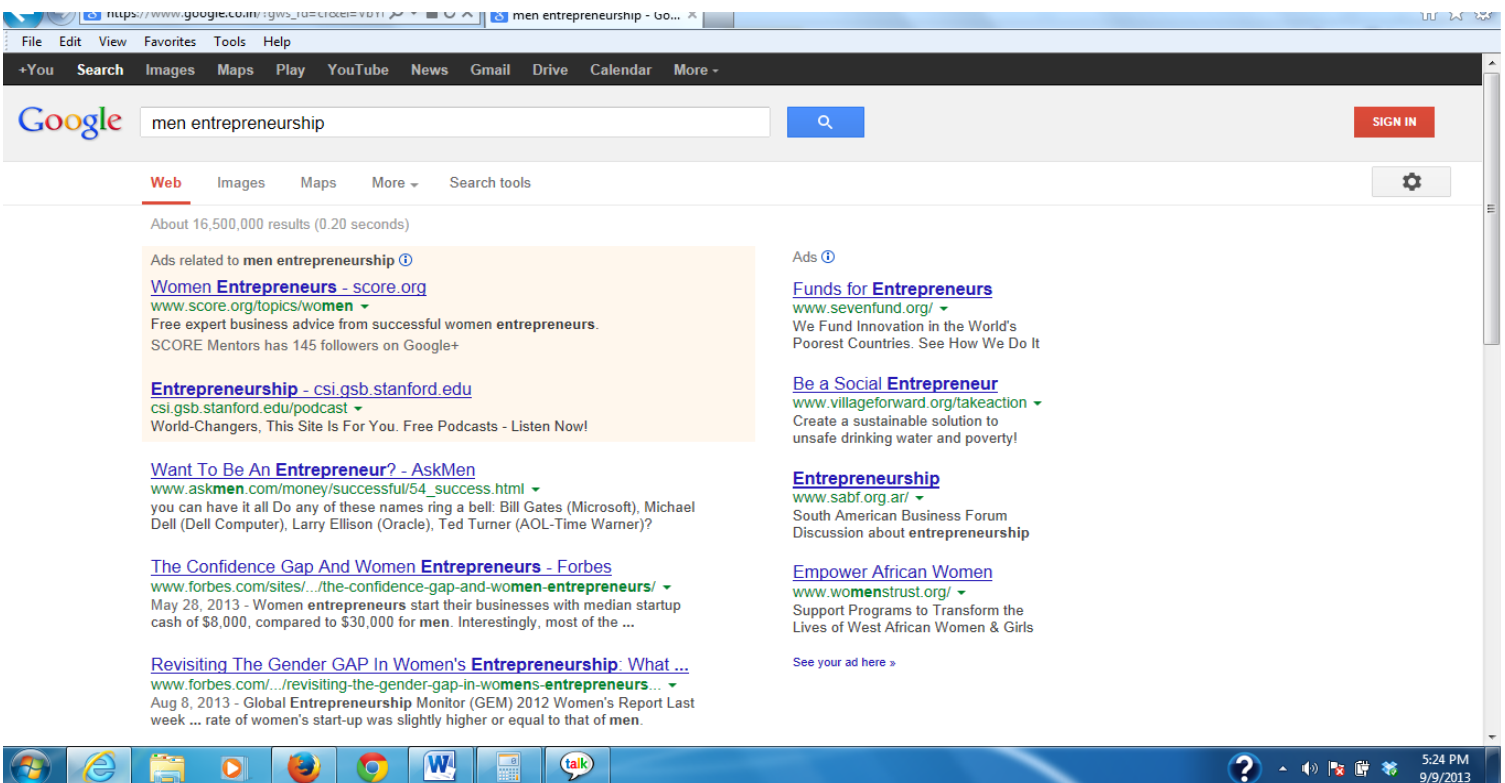
by Smita Mishra

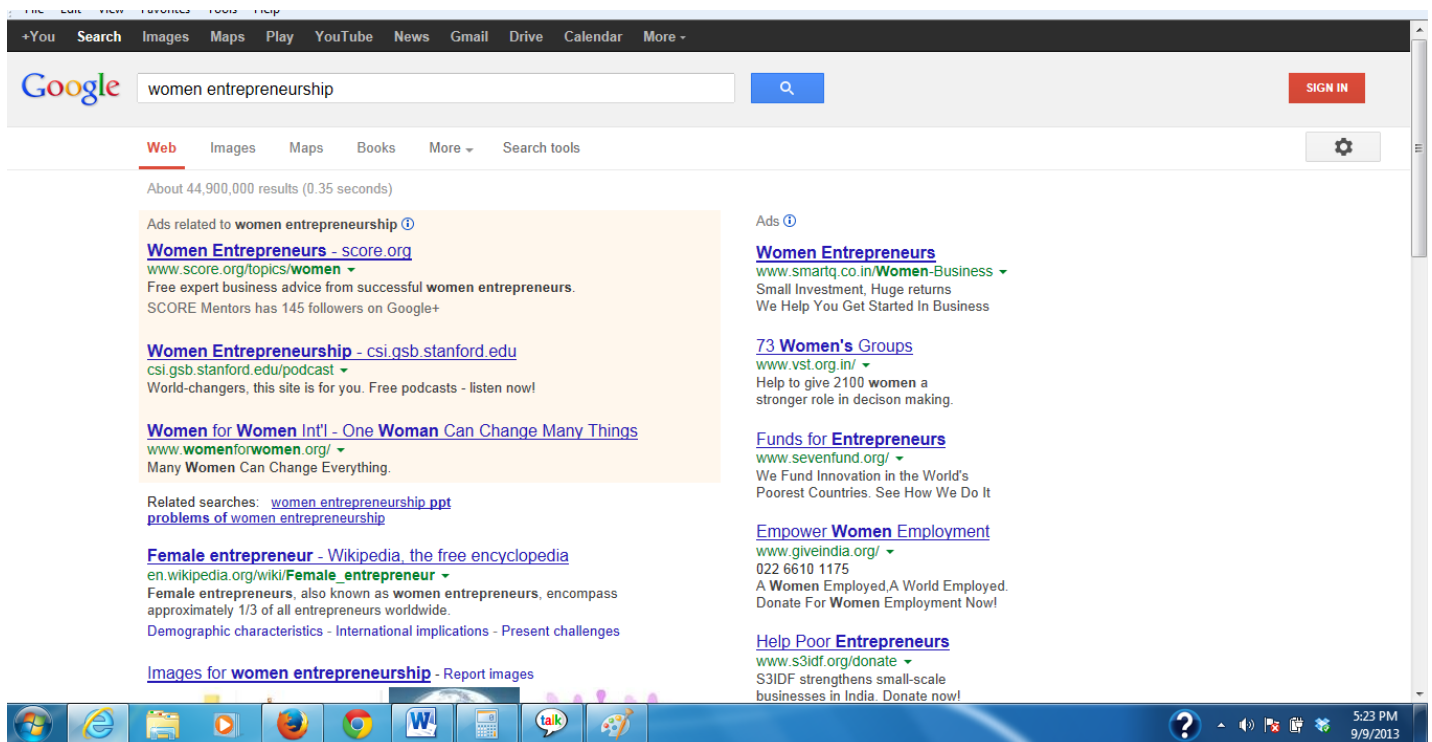


As I write this I am constantly jumping from one thought to another in my mind, trying to find the right words and stories to share with the readers. I am afraid of portraying any incorrect or discouraging scenario for women entrepreneurs. Not because there are no gender biased obstacles but I don't wish for this article to look like some call for a Pink revolution. Because it is not.

Applying converse theorem of mathematics - In today's world - if Google® search doesn't recognize something, it's more than likely that we are talking about innovations or discoveries that shall happen in distant future, since topics with even remote relevance can be found there.

Well, have a look for yourself:





I didn't play tricks here – try this on your machines.

This is making some obvious revelations – there is no such thing as Men Entrepreneurship while there is certainly something to be talked about WOMEN entrepreneurship. And we will talk about it – in just a while. For now – let's start from the start.

An entrepreneur by definition is - A person who organizes and operates a business or businesses, taking on financial risk to do so.

By practice (read - as per Smिता's personal experience) – an entrepreneur is one who is willing to work systematically on an idea to achieve her vision.

I worked as a Test professional for about 10 years – different organizations / teams / clients / business domains / technologies / tools – learnt a lot (I thought) and decided one day to take the plunge.

So when I started as an entrepreneur - I had this idea of offering testing solutions and services to my clients, in the process helping them protect their brands and businesses against current and future risks (just think of scalability – if you didn't get this). I won awards for testing and being the best rated person in the team, every promotion I recommended for my team members always went through. I was always given the best projects and promoted every year. In short, I was doing well for 10+ years in the industry. And so, was pretty confident about my initiative.

Little was I aware of the "entrepreneurship" process though. But I took my leap of faith and started.

I was prepared to go back and forth, evolving the strategies, solutions and services as per the customer's needs and their demands. And I did have to reinvent my approach and solutions and my

services way more than I had expected. However, ***What I was not prepared for was the human side of it!!***

WRT Client: Recently a networking guru – Paritosh Pathak told me – you don't network, you BUILD network. This line has caught my attention and I am using it here to tell you a story. I used to be naïve enough to believe that if I know someone and that someone knows I am good at work (e.g.: ex-clients / ex – bosses, ex – colleagues...etc) they would do business with me, whenever they have a need. But like the line above – you don't just make a sale, you build customers. It took time but I did realize - Not everyone who knows you and has a relevant need is your immediate customer. So, my learning? - stop harassing the clients for work!! How I was perceived as an individual contributor or even as part of small team of a large corporation was very different than how I was being perceived as THE team of the small organization.

WRT Team: We are not in the manufacturing sector where the customer comes in and just buys the product he needs. Services are different and difficult to sell, since a lot of the time the customers don't even know they need it. Products, being tangible are easier to market. The key to building the reputation of the organization and the product lay mainly on the product itself.

Contrary to this, services are difficult to market – you are going to be judged purely on the service provided by any resource of your organization. Of course, there will be moments where your resource will interact independently with the clients. And as clear as you might be with your vision and mission, unless each individual who faces your client independently understands and is aligned one hundred percent to the company vision, this may not always result in a pleasant business outcome. Thus – I realized an organization is defined not only by its services and products or CEO, but the overall culture it stands for. So, I learnt my next big lesson. Unlike the corporate culture - where I was coming from and was myself part of the very few top star performers –as an entrepreneur I will have to ensure that each of my testers is a star performer. I can't just depend on a handful.

What did I do -

Though we say, more the merrier the reality is, beautiful things make sense to business only if they contribute to actual business – otherwise it's difficult to bring attention to such details of organizations that don't contribute to the bottom line. So, as a business woman I wanted to see value in this thought. I knew I am able to motivate an individual; to help them understand my vision but also see them achieve their full potential so others also see them as star performers. However, I realised– If I continue to make just a few star performers - I am limiting my company from achieving its full potential. I saw that each of my employees had that ability, which I was not tapping.

Do I have time to sit with each individual and do this – sometimes – yes, mostly – no. So, what did I do? I worked on the overall culture of the group (not just certain individuals) – reignited the enthusiasm in them; cultivated understanding of the company vision. I made the team agile enough so they could adapt to the evolving solutions, services and products of the company.

As a startup and being 1st generation entrepreneur - I knew we were taking time in shaping up. It would have been a lone fight, but by working on the company culture it transformed the fight into WE. Together we all understood it would be sometime before we reach THAT POINT (where we would know

we are going to stay here for some time) and this helped the team fit better to the business needs. Suddenly things made sense to all. Actually – for few it didn't work out but it was good for both sides to know this and choose accordingly.

WRT ME: I love Testing – always did. Enjoyed being hands on at every stage of work. I wasn't one of those managers who had no clue of what the actual changes of the applications are, didn't really ever believe in lot of people management. So, when I started as an entrepreneur – I had to give significant amount of time to Sales and Marketing too. And, I began to lose motivation as I was getting less and less to test. One fine day, the Director (India head) of my first client – IRIS Technologies Ltd – Mr. Anil Apte, who has been a supporting figure throughout – was talking to me and I opened up to him – saying I can't handle this. I don't enjoy this anymore. I thought that by starting my own firm, I will get more clients directly to me and will be free to pursue my likes. I will do more and more of testing work. However, it's not going in that direction at all. Now, for most part of the day- I am only trying to educate my potential and existing clients about Testing and what we can do for them. The actual testing is being done by other team members. To this – he calmly responded – YOU CHOSE THIS. You should have chosen to be a test consultant – working directly with few clients and not trying to run a company as an entrepreneur.

Well, I agree – he was correct. I had to let 2 of my aspirations fight with each other... and sometimes they still do. But I have found a way to manage. I choose to be an entrepreneur and also actively participate in testing when I can at work or through friends at networking sites – thus fulfilling both my urges. E.g.: Weekend Testing / Testing competitions / Testing classes / Testing conferences.

As far as the women part of the entrepreneurship is concerned – incidentally, I never felt the difference of being a man or woman on the shop floor. I see CEOs of all (testing) companies slog equally hard. I never felt the difference unless I was specifically made to feel like a woman. And trust me – we all have and still will, meet such people – both men and women. However, I do not think I faced it just as a CEO. You are just as likely to face such folks too – without getting into entrepreneurship. Press the IGNORE BUTTON strongly and move ahead. That's all you've got to do.

I come from a patriarchal form of society where, like it or not – women are culturally bought up with a sense of being less – we train our minds to be guilty slaves. If you are away from your kids while working – you are likely to feel guiltier than a man would. If you skip an evening at home to go out with your friends and let someone take care of your kids' homework – you are more likely to feel guilty. It's so acceptable for a male colleague to be paid more or handle bigger responsibilities at work. Every time we enjoy a moment it's like a stolen moment that you need to replace with a moment of hard work and punishment for yourself. Unfortunately – this is something we do to ourselves – no one else does it to us. We get so trained in the starting years in limiting ourselves that after a period of few years – we are on auto pilot.

So, stop limiting yourselves based on what some X person(s) might think of you if you did what you like. Gracefully accept your abilities to compete and succeed. Be assertive. Don't let yourself be the roadblock for your success. You can deal with the rest of the world.

I constantly trained my mind to never feel the bias of being a woman. It was a difficult but refreshing change. Suddenly there was so much more energy to compete, zeal to achieve and hope to succeed.

Thankfully, I have met all the amazing people in my journey so far – helping me to grow. All I had to do was reach out to them. Tell them I am looking for this kind of support and these are my dreams. And they were all so ready to help me – be it Keith Klain / James Bach / Lorinda Brandon/ Leah Stockley / Anna Royzman/ Michael Larsen /Matt Heusser /Navyug Mohnot – still many names to add.

As an entrepreneur – I am still learning. As a woman entrepreneur – it's no different.

Remember - Life is a series of problem-solving opportunities. The problems you face will either defeat you or develop you depending on how you respond to them.

If I was not as lucky as I am – and I met more wrong people than I did and really did hit the glass ceiling I would still not worry – because I am equipped with a diamond head that would help me cut through the bias.

In case you are looking to network with women from techie background and are interested in getting mentored or in mentoring someone in need , please visit a blogsite managed by Lorinda Brandon (@lindybrandon) : www.alineattheladiesroom.org

[Back To Index](#) 



Smita Mishra is the founder and chief Test Consultant at QAZone Infosystems, which is a niche software testing organization. She is a Test professional who has spent over 12 years practicing testing and leading test efforts of varying sizes, cutting across all key domains and technologies. In her current role, she is involved in creating test teams, managing testing for software firms, leading the overall test strategy for them. She is also supporting her key clients in initializing & implementing continual process improvement and has supported them in designing and implementing meaningful metrics.

She is actively involved in learning, practicing and promoting Context Driven Testing. She can be found on Twitter at @smita_qazone.



In the school of Testing

for your better learning & sharing experience



Learning to Test

by Fiona Charles

How should we teach beginning testers their craft? Should we put them through certification courses or other formal training, or should we find ways for them to learn experientially, with guidance from skilled practitioners?

I am firmly of the opinion that software testing is a craft, and that like all serious crafts it should be taught through apprenticeship. That doesn't preclude formal classes, but it puts the primary emphasis on guided hands-on experience, which should then be supplemented by classes that provide a sound theoretical grounding and a wider industry context for the apprentice's particular experience. I admit my biases: I am the child of a skilled craftsman engraver, and I learned to test experientially. I grew up knowing of the pride of craft, so I never saw it as somehow inferior to a profession. When I became a tester it never occurred to me that a job where I rolled up my sleeves and got deep into the logic of a system could be anything other than a craft.

I had extraordinary luck at the beginning of my career. As a student studying English Language and Literature, I fell into a summer job as a technical writer with my university's library automation department, which was in fact a leading company in its field. They had never had a technical writer, and my hiring manager had a theory that the best manuals were written by people who could write but knew nothing about computers. I fit that bill for sure, and though I didn't understand a word he was saying about the job, I just kept nodding and smiling and saying, "Sure! I can do that!" I got to invent that job—with a lot of help from generous colleagues—and they hired me to do it full-time when I graduated.

I learned the most important lessons of my career in that place. I had to be inventive and independent, but I had to seek out help and guidance when I needed it. I also learned from the beginning to ask for feedback on my work. If any reader couldn't immediately understand and follow what I'd written, then it wasn't clear and I needed to rewrite it.

We didn't have any testers there, so I didn't know what testers did or even that they existed. But later, when I was offered a testing job at another company, I just repeated what had proven to be a winning formula: "Sure! I can do that!" I got to invent that job, too.

In both companies, I was working for what were essentially startups: little Canadian companies that made a big splash in their respective markets with innovative and (for the most part) excellent software. I worked in those little companies for a total of 14 years, which gave me invaluable experiences and opportunities to gain depth. After that, I went to work for a smallish Canadian consulting company (which was later bought by IBM), and spent another 13 years working on many varied projects and gaining breadth. I still have close friends from all 3 of those companies.

If a tester at the beginning of her career were to ask me how best to get experience and grow her skills, I'd say this was a good sequence. First, get yourself into a place where you can grow deep skills and build lasting relationships of trust with a group of skilled practitioners. When you're ready to move on, look for a place where you can work with many different teams in different organisational cultures, on systems in many different business domains and varied platforms.

And actively learn everything you can about systems as you learn to test.

I learned to test by testing software for a living, as did everyone in my benighted generation when we had no formal training, no certifications, no testing courses. Oh yeah—and at some point fairly early on, I picked up Glenford Myers "Art of Software Testing". I read other books later and (much later) took some courses. I still do those things.

Did I say we were "benighted"? Nuts!! It was a terrific way to learn. Far from muddling in the dark, I was learning to shine a light on the system. I had already done things very much like this in my previous job as a self-taught technical writer. I was so ignorant when I started that I thought it was a tech writer's job to exercise the software inside and out to see what it did. I never learned to read Assembler in that tech writing job (unfortunately), but when the programmers learned COBOL and PL/6, I learned it with them. Later tech writers who went to tech writing school thought exploring the software and reading the code was the wrong way to get information about a system and they should just ask the programmers. I feel sorry for them, the idiots. They missed a lot, both of fun and of learning. By declining to challenge what the programmers told them and parroting what they were told, they also wrote inferior and inaccurate documentation—as anyone who has tested a system could tell them.

In 1984, I went to work as the sole official tester for a little Canadian company with very big-ticket customers. Our large software application was a manufacturing execution system for semiconductor manufacturers.

I read what I could find about the system and about our customers. There wasn't much. I asked questions. And I explored the system. When I found something that didn't look right, I wandered over to talk with the senior programmer for that part of the system. When he (they were all guys at that point) had a moment, he'd clear a space and get me to sit down with him. I'd show him what I'd done and seen. Sometimes it looked okay to him and he'd explain why.

If it didn't look right to him or I still had questions, we'd go look at the FORTRAN code and we'd talk about what we saw and he'd show me the fix. Pretty soon I took a FORTRAN manual out of the office library and learned to write and read FORTRAN. Then I could read the code before going to the programmer with an anomaly, so I could take more information along.

I had to run batch jobs sometimes, so I read the manual and taught myself DCL (the much friendlier VAX version of JCL). The guys had written a scripting language to test the basics of the system, so I became an expert on that, and not incidentally learned some things about automated testing at a point

when very few people practiced it. When we ported the system to IBM with an ORACLE database, I taught myself SQL.

Eventually, I learned more about test techniques by reading some testing books and trying out the things I read, or at least giving a name to what I already did.

One of my jobs was to organize all the programmers and the customer support people (when we got some) into one big team to test the system for each release. So I got to learn some things about test management, too.

I think this was an excellent way to learn testing. Could you call it “unstructured”? I’d say no for many reasons, not least because I was testing for a living. I had responsibilities and I had the discipline of production releases. I also had the support of knowledgeable programmers and customer support people. We were proud of our product and we worked together to make it the best we could.

Writing this reminded me of my experience learning human, as opposed to programming, languages. I learned French and Latin in school primarily by learning and memorizing grammar: verb forms and paradigms. We read some, and we spoke less—French, that is (we didn’t learn to speak Latin at all, sadly)—but an awful lot of our learning was divorced from the real human use of those languages. I liked grammar and I was good at it. But I never really learned to read either language well, and my spoken French is embarrassingly poor.

Contrast that with how languages are taught now. You learn primarily by speaking and reading—exploring. You learn grammar too, but more by osmosis or because you need to use a particular form. I think if I’d learned this way, I could actually have learned to speak and read French, and at least read Latin.

So how do we think people should learn to test? I don’t think it should be by rote. Nor on paper. (Does anyone learn to play the piano by learning theory first? Theory comes later: after the student has mastered the rudiments of the instrument.)

I think testers should learn testing by working with real software: exploring it, finding bugs, making mistakes and learning from them, working with knowledgeable people.

Fiona Charles teaches organizations to match their software testing to their business risks and opportunities. With 30+ years experience in software development and integration, she has managed testing and consulted on testing on many challenging projects for clients in retail, banking, financial services, health care, telecommunications and emergency services.

Throughout her career Fiona has advocated, designed, implemented, and taught pragmatic and humane practices to deliver software worth having—in even the most difficult project circumstances. Her articles on testing and test management appear frequently and she speaks and conducts experiential workshops at conferences. Fiona edited *The Gift of Time*, and guest-edited —Women of Influence||, the January 2010 special issue of Software Test & Performance magazine.

Fiona is co-founder and host of the Toronto Workshop on Software Testing. She can be contacted on Twitter @FionaCCharles





Prepare for Promotion Now!

by Bernice Ruhland

We often hear how the number of women moving up the corporate ladder becomes smaller at each rung. There are a lot of good conversations discussing the various reasons that factor into this problem. This article is based upon my experience on a few things women can do to let their voices be heard, have more influence, and prepare for senior level positions.

Who are Your Mentors?

Throughout your career you should have different mentors based upon your changing career needs. Sometimes the informal mentoring relationships are the strongest because they naturally develop over time. Also look into any company-sponsored programs because you may connect with a higher-level manager who can help progress your career. Plus social media has opened another mentoring pathway.

Do not be afraid to ask women who have been promoted into senior positions for advice. Often these women are accused of not helping other women. But I wonder how many women have reached out to them? On the flip side, I have witnessed women who become competitive with other women in their organizations, which is never helpful. This is another good reason to include female mentors outside of your company. Do not focused only on women in the technology field, but find women who work in finance and other male-dominated industries. Their experience and advice will be valuable. Many of them have experienced the glass ceiling and have overcome it.

Sometimes women tend to only seek out women as their mentors. I believe it is better to seek out both women and men. I believe both can provide valuable guidance to help progress your career. If you can, have a male mentor who works within your company. He may provide a different perspective on what it takes to succeed and rise up into the ranks. If you can build a trusting relationship, be open about your career aspirations and discuss what might be blocking your opportunities.

How are Your Communication and Presentation Skills?

The ability to communicate your message effectively is important for career progression. I have found that as you communicate to higher levels of management, they do not want the details but a concise answer to their questions. If you are telling them about a problem they do not want to know all the steps you went through to research the problem. They typically want to know the problem and the recommended solution. Whereas working with testers and developers the conversations tend to revolve around the details. To improve your skills take a presentation course, study personality models, or even join Toastmaster an international club designed to help members develop communication and leadership skills. Public speaking is one of the fears I hear a lot from professional women; but it is a fear we must overcome¹.

Whatever option(s) you select, the time and money spent is well worth the investment². I improved my presentation skills through a public speaking course. Initially it was intimidating but using a presentation model helped me prepare and present information focused to the audience. What I learned through this course was important to my career development. I found opportunities to make small work presentations to improve my confidence. Over time I became comfortable making presentations with limited lead-time and improved my ability to communicate with senior management. This allowed me to become more visible in the company, which opened up new opportunities expanding my responsibilities.

Do You Know the Unwritten Rules?

The business world has become relaxed with flexible working hours and casual dress policies. However, there are often unwritten rules associated with senior level positions. A company may have a casual dress code but how does the people at the level you wish to achieve dress? How do they present themselves in meetings and other work settings? There use to be a popular saying: dress for success. It may sound dated - however, it might be important to dress for the level you wish to achieve. Several years ago, I had a woman ask how her male counterparts could take her more seriously in meetings. This was a woman who knew her field, communicated well, and was not afraid to speak up. It was puzzling and frustrating that she could not get her voice heard. I noticed she was wearing a sweatshirt and jeans - which was acceptable under our dress code. I advised her to dress business casual the next time she was going to this meeting and she found that her voice was heard. Of course it is hard to know if she gained more confidence or if she received more respect by her presentation.

I work at a company with a casual dress policy. My own personal preference is to dress in a manner that if I am called into a meeting with the CEO I have a professional appearance. Sometimes I will wear jeans but always my best jeans preferably with a jacket. Some companies are truly casual and business casual may not be appropriate. I am not suggesting that the clothes you wear will automatically gain you respect or a promotion. But depending upon where you work it might be part of the package. Only you can make the determination.

Where is the company going?

Companies are facing tremendous competition and must keep up with technical advances. As testers we understand that quality for our customers changes, which can impact how we approach testing problems. When viewing our careers, we need to consider that our companies are also changing. Do not limit your opportunities based upon what exists today. Consider how your company is adapting to a changing environment to determine if there is a different role for you. Of course it can be more challenging to understand future opportunities within a company than it is to prepare for an existing position. Having the correct connections that can help you understand the company's future direction is

important. Plus demonstrating your skills and ability to accomplish challenging assignments to show you are ready when those opportunities arise. I often have found that these types of positions are "filled" while they are being created. At that point it can be too late to try and bridge your accomplishments on why you are best for the position.

What Professional Women Groups and Online Communities Do You Participate in?

Attending a local Woman in Technology group or participating in the online forums³ are options to network with other tech women. Also consider professional women groups not focused on technology since women across industries tend to face similar problems. The online groups allow you to connect through social media, plus they may sponsor conferences and other events. Some of them provide career development sections through their website including mentoring opportunities.

LinkedIn has several professional women groups and one that I follow: "Connect: Professional Women's Network, Powered by Citi". Many of these "women" groups bring in viewpoints from men; which is a nice benefit. As with any group (women-based or gender neutral) make sure they are action-driven where the conversations are not going down a negative, complaining avenue without any solutions. Time is precious so find groups where you can contribute and gain something for your own career.

References:

- (1) Man Karlsson, S., July - August 2013, Feel the Power of Facing Fears, *Inspirational Woman Magazine*
- (2) Crispin, L., December 2011, Professional Growth: Is it Unrealistic to do it on Your Own Time, *Tea Time With Testers*
- (3) WITI Women in Technology International witi.com; The International Women's Leadership Association theiwl.com; WIT Women in Technology womenintechnology.org



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

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

Bernice can be reached at:

LinkedIn: <http://www.linkedin.com/in/bernicenielruhland>
Twitter: [bruhand2000](https://twitter.com/bruhand2000)
G+ and Facebook: Bernice Niel Ruhland



My career as a tester started many years ago, but the first encounter with my future peers happened in 2007, when I was tasked to find the best education courses for my company's QA Department. I evaluated few tutorials and found them to be profoundly boring, and then attended 3-day *Rapid Software Testing* class. I took it in Toronto with *Michael Bolton*, and then invited *James Bach* to teach it to my department. At the same time, my then coworker *Bernie Berger* invited me to *STiFS* (Software Testing in Financial Services) peer workshop.

To say I was hooked is to say the least. The great stimulation of your brain together with shared passion for the advancement of the craft and curiosity to discover the various aspects of your profession is what I found to be so irresistible. But the main thing was that I felt inspired and motivated. I met famous people who shared my interests and my passion.

I attended my first *CAST* in 2008, and in 2011 had my first 10 min public presentation - again, at *CAST*. When I left the conference that year, I went on Twitter. One of my first tweets said: "... I have this weird feeling that everyone is an old friend... Are we all from the same planet?" Yes, that's how I feel about my peers. We share the same language and same passion. We are curious about same things. We want to talk to each other about our craft. We want to help each other to get better at it. It's a true friendship in a very interesting sense. You want to be true to your friends.

This is what my peers taught me:

Maintain your integrity.

Understand where you stand, and don't bend your principles. Have a voice. Be honest to yourself and to others. *Matt Heusser* is my colleague and a good friend, and he *is doing some amazing things in bringing the community together*. Matt writes on variety of topics covering software testing, agile development, and development management, and invites an army of people to give him feedback on his articles. While Matt is the *recognized expert in software testing*, he is honest about his vulnerabilities. If you listen to his talks or join his projects, you will notice how exposed Matt is about his expertise and where he is lacking it. His stand point wins a lot of respect.

Learn all the time, through all possible channels.

Expand your knowledge beyond your craft. Testers are professional learners. The scientific discoveries, cognitive psychology, knowledge management, focusing and defocusing techniques, pattern recognition, psychological types and methods of communications, you name it.

Lead by example.

James Bach taught a tutorial at CAST 2011 on *Context Driven Test Leadership*. He treated us, the attendees, with utmost respect, and showed us what it means to be a leader. Through his own examples, he demonstrated how to evoke a test leader in you, and to be on a constant lookout for excellence. James also taught us on how to engage in debates about your ideas. If you are a leader, you have to offer your ideas publicly and should be willing to put your ideas to a test. I follow James, he doesn't compromise. It bugs some people, but doesn't bug me. I know I can trust him.

Prepare for a fight - you will be attacked.

You will be challenged by the people who don't want to change the status quo and the people who have their own agenda. Some will have trouble accepting what you have to say, and some will want you to shut up. Develop emotional intelligence. *Learn to argue* and learn how to win.

Serve your community.

CAST is an amazing testing conference which is more inspiring than any other I attended (I have been to many). CAST is run by volunteers. The most senior members of the Context-Driven testing community offer comradeship and mentorship that you find incomparable to any other. They are interested in your growth. I have an honor in personally knowing *Ross Collard*, whose support and advice I truly cherish. Ross is a legend in the world of testing, and yet he finds time and interest in encouraging my endeavors.

Becoming one of them

There will come the time when you will know that you moved up close to your future peers. People who you respect will want to talk more to you, invite you to their peer conferences and events, request your feedback, and start following you on Twitter. That's the time when you take the next step.

Now, you need to master skills in getting to where you need to be. I will use my case as an example. This year I took *BBST*. It's the course which teaches the black box software testing skills; but the way it is organized, it also provides the opportunity to receive and give feedback to other course participants. As a test manager and a mentor of *PerScholas STEP* program, I need to provide feedback to testers. *BBST* allowed mastering my feedback-giving and mentoring skills.

Also, I speak at conferences several times a year. Some people are natural at it; I am not. *WomenInLine*, the group I belong to, paired me with *Mark Tomlinson*, a Performance testing expert and a great speaker. Through few mentoring sessions, Mark exposed me to the science of public speaking, explained power dynamics of speaker/audience interaction, and helped me to construct my upcoming CAST session in the most effective way. Seriously, that made a world of a difference. I felt much more confident in working the room, speaking my mind, and delivering my message in the most powerful way. Based on feedback, I know I made an impact on my audience. And that was the intent.

From where I stand, here is my advice to you:

Contribute.

This year, I hosted *WOPR* and *STiFS* workshops at my company, assisted with Test Retreat and co-organized Test Leadership Camp. I went to PerScholas and spoke to STEP students about my experiences as a test manager. I started AST Quality Leader SIG and organized a panel at CAST 2013 inviting test leaders to talk about their experiences and share their stories. Find how you can contribute to your community.

Learn.

I am attending *Change Artistry* workshop with *Jerry Weinberg* and *Johanna Rothman*, because I want to be taught by the most respected teachers in the world on the skills I need in becoming a change agent. Also, I am taking *BBST Instructors* course. I suggest you get educated by the best in your field.

Promote.

Comment on skilled testing in forums. Join (or start) local groups. Write about testing in blogs. Get on Twitter, find out what others in community buzz about, and share with your followers. Bring what you learn to your work place. Excite others about things that excite you. My friend *JeanAnn Harrison* is an expert in Mobile testing. It's easy to find her at test forums. I think she never sleeps...

Practice.

At your work, try out the things you learn from attending conferences, workshops, seminars, or read about. Share your experiences. You might have an insight that others will be thankful for. Or prove some "best practice" to be wrong in your context.

Ask for feedback.

I invited *Rob Sabourin* to my session at CAST 2013. His feedback was very insightful while complimentary. Robert reinforced the strengths of my talk, and also made some comments on how to coordinate my future sessions with a facilitator to better fit my style. Do ask for feedback from the people you respect.

Earn your reputation.

Participate in *peer workshops*. Look up the workshops in your country/state, and if you don't see one, you can always start your own. Engage into conversations with the people you are aspired to. Demonstrate your skills. Be visible. At some point, you find people who are aspiring to you. Give back, help them grow.

Sharpen the saw.

If you want to see how passionate testers spend their time, watch them after dark at test conferences. If you spot a crowd playing brain games at bars -- make no mistake, these are the members of Context-

Driven testing community. When you have some free time or need to divert from your daily routine, play the *SET* game. Better yet, play it with someone else.

Conclusion

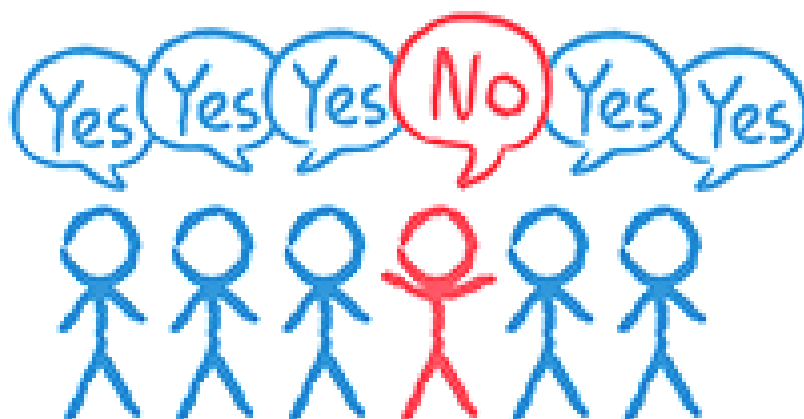
If you want to be the best in the world, you'll want to choose your peers very carefully. After all, who you spend your time with will influence you, and will help define you to others. Make it your goal to learn who they are. Find out who they aspire to, where they hang out, what they are passionate about and what they are committed to. Then -- start conferring. Seek out advice from your future peers. Contribute to the causes they find important. Find mentors in the areas you need to improve.

Become a mentor yourself. Be transparent. Be humble. You have a lot to learn.



Anna Royzman is the test lead in a cross-functional product development team that delivers game-changing software in the financial industry, where “quality” is as important as the “time to market.” With a wealth of experience in the testing and quality assurance field, she has developed unique perspectives on quality leadership during the past decade. Anna organizes discussion panels, leads SIGs, creates workshops, and speaks at conferences to promote the value of skillful testing and the whole team approach to quality.

Anna started AST Quality Leader SIG in 2012, and serves as the SIG Chair.



Be a Student for Life...



by Teri Charles

Tester1: Have you heard of James Bach?

Tester2: Who?

Tester1: How about Michael Bolton, Pradeep Soundararajan, or Elisabeth Hendrickson?

Tester2: Ummm... Who are they?

Tester1: Do you use heuristics in your testing?

Tester2: What does that mean?

Tester1: Does your team use Context Driven Testing?

Tester2: Context driven what?

Are the people and concepts that Tester1 asked about completely foreign to you? If so, you're not alone. But there are resources out there that can help you make sense of them. I would like to open up a whole new world to you, my fellow Testers!

It's very easy for people who know these names and terms to take them for granted. However, there are many, many Testers who have never heard of any of them-- not to mention the dozens of other test leaders, trainings, books, and other resources that are available. And I am not just talking about people new to testing. I am also talking about experienced and senior Testers.

Because you know what? Tester2 was ME not too long ago.

My Journey

I've been a Software Tester for over 10 years. Like a lot of other Testers I have met, I was in the dark for a lot of those years. I didn't know what it really meant to be a Tester. There was no test training out there when I started (or, at least, none that I knew of). I had no idea that people actually wrote books about testing. The only other Testers I knew were the few I worked with. I call those years the "I didn't know what I didn't know" phase of my career.

There's no such thing as the "University of Software Testing", so we come from various backgrounds: Computer Science majors to high school graduates (or dropouts) and everything in between. I went to college to study music, played in a band, and wrote songs and lyrics. Then I took a circuitous path, working in political non-profit organizations and a bookstore, writing screenplays for movies (no, nothing ever produced), and dabbling in documentary filmmaking. Yep, sounds like the perfect journey toward a career in Software Testing, right?

But it was. Because it was **my** journey and it was perfect for **me**.

Every experience on my path let me do the things I love: learning new things, being curious, working through challenges, helping people, and exploring. (Now is my journey starting to sound like a Tester? I thought so!) My curiosity and love of learning were the qualities that convinced my first manager to pluck me out of a group of candidates --some of whom were more technical and more experienced than I was-- for my first testing job.

The Past

I look back now and am a bit surprised that, without a lot of training, I figured out how to do the job pretty well. I found lots of bugs, brought teams together, dived into learning new things, found ways to innovate and improve our processes, and was a well-respected leader and team member. If it hadn't been for my varied experiences and natural curiosity (and probably a little luck), I wouldn't have been nearly as successful as I was.

But here's a secret: I was also unhappy at times. But I didn't know why. And I had no idea what to do about it.

My Aha Moment

Being laid off is part of our industry. If it ever happens to you, don't take it personally. It's business. But if you let it, it can also be a blessing.

My layoff came about a year and a half ago. The blessing that came with it was the opportunity to stop. To just "be". I was able to look within and do some serious thinking. What I realized was that, while there were a lot of things I liked about my work, I also had to admit that I had been unhappy because there was SOMETHING MISSING. But what was it? And how do you find out what is missing... when you don't know what is missing? I just knew there had to be more to this thing called testing. And I knew right then and there that the only way I wanted to go was forward-- and discover what this "more" was.

My Education

So, I did what I usually do. I dived in. And I didn't just dive into the shallow end of the pool; I dove into the deep end. But I was swimming blind. Remember, I didn't know what was missing.

So I started with Google searches. Of course! I started Googling everything to do with testing, test trainings, Testers, software testing, test books, and test blogs. It was amazing! Every link took me to another, then another. One of the first things I stumbled upon was James Bach's book, "Secrets of a Buccaneer Scholar". How lucky that this little miracle fell into my life at the exact time I needed it! His journey was my journey. So many of his experiences and feelings were exactly what I had experienced in the past or what I was going through in that very moment. It was an inspiration for self-learning.

From there, things took off. I couldn't believe that this whole world of testing information existed. I had so much to learn and I didn't know where to start. (I still feel that way even today! So many things to learn!) So I just took a deep breath and chose one thing to start with. And then I chose something else. And on, and on, until I had done these things (among others):

- Discovered and worked with a great testing coach, Anne-Marie Charrett, who challenged me and helped me to push myself. Anne-Marie also introduced me to heuristics and oracles. @charrett
- Took a month-long online testing course with Ajay Balamurugadas. @ajay184f
- Started my Twitter account (@booksrg8) and discovered the most amazing testing community!
- Refreshed my SQL skills.
- Read numerous books, white papers, testing magazines/newsletters, and blogs on testing.
- Completed the 5 day Rapid Testing Intensive class (online) from James and Jon Bach in July 2012. www.satisfice.com
- Participated in Weekend Testing events.
- Created my blog on Testing (www.bouldertester.blogspot.com).
- Joined several Testing organizations.
- Learned about Context Driven Testing. Talk about changing how you approach testing! (context-driven-testing.com).
- Took free online classes from places like Coursera, Udemy, etc.
- Successfully completed the BBST Foundation course (Association for Software Testing).
- Became co-organizer of local testing Meetup in Boulder, Colorado.
- Got involved in a wonderful, local Meetup called GDIBoulder to learn things like HTML/CSS and improve my skills to become a better Tester. @GDIBoulder
- Took the Rapid Software Testing (RST) course with James Bach in Orcas Island, Washington in June 2013. www.satisfice.com

It Never Ends

Here's the thing, though. The list above is just the beginning. It's a journey, remember? There are SO many other things to learn, to read, to create, to delve into, and people to meet. My list is long and keeps growing! Yes, there are times when it's overwhelming. There are times when there are so many

things on my to do list, that I get paralyzed and do nothing from the list for days. But that usually doesn't last very long; pretty soon I remember to take one step at a time and choose the next thing I want to do. You can't beat yourself up if/when that happens to you.

Your Journey

If you don't know this yet, YOU are responsible for your own education. No one else is responsible for it. If you want to be a better Tester, you can't just wish for it. You have to work on it every day. If you don't know where to start, that's okay. But don't let it stop you. And I'll let you in on the one of the most important things you can do.

Ask. Just ask for help.

You will find that we have an amazing and generous community of Testers. Take your first step and reach out. Do you know you want to improve in something but don't know how to start? Is there something new you've heard of but don't understand? Look around you and ask a fellow Tester. If they don't know, suggest you figure it out and learn together. If you meet a new Tester, strike up a conversation and ask your questions. When you learn about a new test guru, reach out to them. You will be pleasantly surprised how helpful these busy people will be. It doesn't mean you'll get the right answer or the wrong answer. It doesn't mean everyone will give you the same answer because everyone's journey is different, and their journey is right for them. So ask a lot of different people. Sift through their responses like a gold miner looking for gold. Find the gold nuggets that feel most true to you and keep them-- then search for more. There's not just one answer.

It really is about choosing to take that first step. I have found that once you get up and have the courage to take the first step, it's easier to take the second, then the third. I am always looking at ways to get better as a Tester. That's the key. It's what motivates me. But the other thing that motivates me is helping others. If I learn something, I want to give it away. It's the main reason I started my blog and most of my posts focus on sharing things I've learned as well as my own journey. In the spirit of helping others take their own first steps, I recently created a presentation for my local test Meetup (bouldertester.blogspot.com/2013/08/testing-resources.html). It's a list of different people, organizations, trainings, books, and resources to help Testers who are looking for some of the same things I was looking for when I started taking my first steps in learning to be a better Tester. If you're looking for answers, maybe it will help you find some of them.

And if it does help, pass it on! Because it is true. The more you give, the more you receive. You will not only help others in their journey, but your own journey will grow in ways you can't imagine!



Teri Charles has been a Software Tester for over 10 years. Her awakening to become a better Tester began a year and a half ago when she discovered things like Context Driven Testing and "Secrets of a Buccaneer Scholar" by James Bach, and it hasn't stopped. She has a passion for continually learning new things and passing on any knowledge that comes her way.

When not actively testing, Teri is also the co-organizer of her local testing Meetup in Boulder, Colorado and the co-founder and volunteer business manager of a 140 member women's community chorus.

Teri's blog can be found at bouldertester.blogspot.com and she regularly tweets at @booksrg8.



Police is your friend

by Abiola Adewoye

Of all the activities that take place during the development of software, testing stands out as a very important step; it is an exercise that evaluates the software with the intent of uncovering errors so that they could be corrected.

However, testers are constantly faced with situations that undermine their importance in the software business. This is caused by the wrong perceptions of their responsibilities by the team members with whom they are involved in software production!

The software developers for example see them as a pain in the neck because they would not let his project *fly* without crossing the t's and dotting the i's. After all, a unit testing of the functionalities should suffice and the end users would be glad to have a functional application - not to bother about aesthetics or following software conventions and standards to the letter!

How can one deny the existence of the inherent conflicts of interest that occur when one solely relies on the programmer who has built an application to test his own work? It will be absurd to expect anybody to prove that his OWN work is faulty or has errors!

The project Manager who should understand better at times also see testers as bottlenecks that impact project delivery.

It is natural for every business owner to invest more resources to items that bring the highest returns to the business. The fact that some software business owners do not invest much on the testing unit is an indication of the low level of importance they ascribe to the profession. These business owners find it more 'profitable' to expend resources over trainings of other members of the production team; they see the Testing profession as a passive unit that contributes little or nothing to the bottom line.

The testing profession is not a passive but a preventive one; a wise software business owner knows better, he clearly understands that the software business is different from every other form of businesses. A simple but sensitive error from poor quality product could be devastating, leading to a life time damage of the brand thus jeopardizing future sales. Learning from mistakes resulting from such poor quality software is not often the best decision because there may not be a room for a second chance. The costs savings from proper software testing are therefore said to be hidden.

A good way to emphasize the importance of testing to the business owners is to monetize these costs and intimate the management of same.

Some of the problems that result from poor quality software are:

- Increased customer support costs, such as the cost of answering phone calls of complaints, the cost of refunds, free upgrades etc.
- Lost sales: we are in the age of free information circulation, it takes no time before an aggrieved customers fills the Internet with bad information about a “buggy” software.
- Reputational or brand damage
- Legal costs, such as lawsuits, fines, and lawyers' fees.
- Inability to maintain or support the products over a long time. A product with uncorrected errors is difficult to maintain or upgrade to catch up with changing technologies.
- Errors are easier and less costly to fix when they are in development stage rather than production when a major application overhaul or downtime may be required.
- Bad software affects business valuation.

In my country, Nigeria - the Police Force has a slogan that states: "Police is your friend" which is written all over their offices and some of the time written on the walls of the cells! Now, this is contradictory because the same cell where offenders are confined has its wall decorated with the statement... How can you be my friend when I am subjected to your tortures?

Not to be taken for its literal translation, the slogan communicates the fact that no matter the action of the police officer(s); be it pleasing or not... such actions are in the best interest of the concerned parties.

So also is the Testing profession to the Application Developer, Project Manager and the Business owners at large. The impact of an independent test group cannot be overestimated in software development, they ensure that the software that are being released into the market are free of errors and thus helps build the users' confidence in the product. For a company that is interested in building a long lasting brand, software quality is not optional; it is the core of the business!

The time has gone when the only defense against programming errors is native intelligence of the programmer; we are now in an age when modern test design techniques are helping us reduce the number of initial errors that are inherent in the code. Different tests methods are beginning to cluster themselves into several distinct approaches and methodology.

Project Managers can make life easier by planning testing into the project schedule.

There is a saying that goes thus: “Treat your customers well, they will always return to you and your product well, they will not return to you” The Testers help the project development team treat their product well and ultimately help the Business owners treat their customers well. What an amazing role of a perfect mid-fielder! He is not the bottle neck some refer him as...

Before you relegate him to the rear seat of the train, remember; the police is not your enemy but your friend!

Abiola Ilupeju started her career as a software developer before she moved to the field of software testing.

She is passionate about providing quality IT solutions to solving complex business problems in a way that ensures that the user requirements are met and the investments of the business in IT initiatives is maximized. She is currently the Quality Assurance Manager in a software developing company in Nigeria. Part of her responsibility as the Quality Assurance Manager is: defining and implementing Software testing strategies as well and communicating quality control information to all relevant departments, outside vendors, or contractors of the organization where she works.

Abiola holds a degree in Computer science with Economics; she is certified in IT service Management (ITIL) and has passed the Certified Internal System Auditor (CISA) examination.



The article was birthed out of the initial challenges she faced while pioneering the Software Testing Unit in the organization she works for. She loves writing and sightseeing. Email: abiolailupeju@gmail.com



Latest from Quality Learning



Selenium WebDriver **Online Training**

Special offer

Buy one seat and get 100% off on another one.

Note - This offer valid till 3rd Oct 2013 only

[Back To Index](#)



A man in a grey suit and blue tie is drinking from a white mug. He has a surprised or stressed expression. To his left is a large, messy stack of yellowed papers. In the center of the image is a large, oversized foot wearing a black sock with colorful horizontal stripes (blue, green, yellow, orange).

Taking
a break?

a click here
will take you there



She is a new and refreshing voice in the world of Software Testing. She is on a mission to educate testers and project teams. She does this by leading and driving improvements across test programmes to deliver efficient, effective test solutions that empower testers and drive quality.

In a short period of time, she has built a reputation as an expert in Context Driven Testing and has cemented her identity amongst the Global Testing Community, not only as an expert in CDT but as an Innovator and Thought Leader.

Her fresh open approach has enabled hundreds of testers to learn and implement new techniques, thus improving not only the individuals testing skills but increasing the awareness of what real testing can deliver.

She has her own blog (www.inspiredtester.com) and speaks at Testing Forums. She has also just set up the new Singapore Testing Forum ... there are just a few of the ways that she is driving change.

Why does she do it? What drives her? What is her mantra for success?

Read in this exclusive interview with **Leah Stockley** and find out what inspires this **Inspired Tester**.....

Over a Cup of Tea with Leah Stockley

Context Driven Testing : Rise of the Thinking Tester

Leah Stockley www.InspiredTester.com



We are curious to know about your career journey. Are you a tester by choice or by chance?

By chance really. I saw a job advert years ago titled 'logical thinker wanted, no experience necessary'. When I applied, the job was for a software tester in a forward thinking bank. They realised by training up those with no preconceptions, they could mould people to be the testers they wanted! That said, my Dad worked in quality control for an engineering firm, so clearly we are both wired with a logical mind and attention to detail... Maybe it's less by chance than I think!

Your blog URL is 'inspiredtester.com'. One can read it as inspired tester and it can also be read as 'inspire-d-tester'. Which interpretation would you prefer and why?

The first, because the name was intended as a declaration about my own state. I wanted to use my blog to share my experience becoming a Context Driven tester. After 15 years testing I had started to lose touch with the elements of testing that originally inspired me to love the job.

That has returned in the last 2-3 years, thanks mainly to others who have made it their mission to advance software testing and to do so publicly. I felt I could also share my experience in the hope it may help inspire others to fall back in love with testing too! Pay it forward...

As compared to programmers, most testers are generally passive learners. How important do you think continuous learning is for testers? What would be your advice on self learning for testers?

I'm not sure that statement is any truer of testers than developers to be fair. There are far too many people who have fallen into a pattern of numbness and stagnation in any given profession. For a while I was close to being one of them!

Whether you apply it to testing or your life overall... Continuous learning is so important... and that has nothing to do with collecting certificates or qualifications! Learning has re-invigorated my life! I now challenge myself to learn something new every day. Sometimes that's about testing; sometimes it's about how people think, how we communicate, technology, health, whatever! The important point is to never fool yourself that you have nothing left to learn, however much of an expert you may be at something. Does an athlete stop training when they win a gold medal? No! The best ones keep training, as they know it's the only way to stay at the top of their game! The same applies to testing ...and life. There is always more to learn. And the best part is ... Learning keeps you feeling young (you appreciate that as the years of experience start to build up behind you!)

We would like to know about your Context Driven awakening. What makes you a Context Driven Tester?

I stumbled across exploratory testing about 4 years ago and used it on my project. By no means did that make me a context driven tester, but that's when the door opened. My true awakening started when I had the privilege of attending RST training by James Bach. He opened my eyes to how much of a better tester I wanted to be! Since then, I've continually tried to learn and improve my testing skills and help others do the same! My aim... To confidently face any testing situation that arises! That doesn't mean I have all the answers right now. It means I have tools in my kit bag (such as heuristics and the ability to listen & question), which allow me to quickly identify how to solve testing problems. Someone with many ideas, few preconceptions and bucket loads of common sense. To me that is a Context Driven Tester.

Could you tell our readers about your experience and experiments around 'Visual Test Strategy'? What benefits do you see over the traditional way of test documentation?

Wow I could take up your whole magazine extolling the benefits of visual test strategies. If you've ever read a traditional test strategy or plan... or even worse, been the poor souls that had to write one and get everyone's sign off, Visual test strategies can bring you hope! They provide so many advantages to our projects, the biggest of which is clear communication of the test approach for a project. Providing information (such as scope, risk, estimates, automation coverage) in a visual form allows these aspects to be presented and discussed in a way that is easy to understand. This allows the project team to make informed decisions about budget and risk.

As for my experience, I've seen many of our test teams adopt this recently. Some to great success, some with mediocre success... I've seen a few key differences that seem to affect the result.

1. How early the visual test strategy was created. Teams who produced it later on (the old 'tick in the box' documentation approach) found it added less value as the fundamental test decisions had already been made

2. Whether or not they shared the VTS with the project team. Regardless of the above, they help teams improve their knowledge and test analysis but when you have such a great communication tool at your disposal, why wouldn't you show your clients. We have only had positive feedback from the project teams.

If you get a magic wand, what thing(s) will you change in software testing field with immediate effect?

The view that maximum automation coverage solves all testing problems. I would like more people to be honest about their experience of failures with automation so companies stop throwing good money after bad! Let me be clear, I'm in no way against using automation to aid testing. It's an essential tool for a tester but only when it's used in the right way. Realistic expectations of its value and limitations have to be communicated accurately. I spent 10 years of my career delivering automated solutions to clients without really questioning what value I was providing. Looking back, I can guarantee most did not deliver the return on investment predicted, due in large parts to the fact they were heavily GUI based and required frequent maintenance, however well designed the frameworks were. Worst of all, they always attempted to replace the efforts of manual testers, instead of complimenting them. Wow I feel a soap box moment... I think I should take this topic to my blog.

Please tell us about your experience with rolling out CDT at your workplace.

From the first dalliance with Exploratory Testing, then the training by James Bach, all the way to now, has been an epic journey! Of course it started a little cautiously... we picked a couple of projects to pilot the techniques & approach on, to make sure we really could change how testing was done in a large regulated environment with no adverse impact. To be honest I always knew it would work, hence I volunteered to drive the roll-out... to me CDT is about common sense testing, it's about doing the right thing at the right time... so how could that ever fail? I think the confusion about 'will CDT work or not' is based on a misunderstanding that it's like Agile is for development. If that were true, of course it wouldn't suit some projects. However, as CDT really means, "determining the best approach for each project/ context", you are always free to pick and choose the approach and techniques that are right for you. The challenge for me has been to ensure our testers & test managers have been trained in some exceptional test analysis skills, so they are even more able to design an efficient approach that meets the projects needs.

Rolling out change in big organisations is not an easy job and maybe that's why organisations don't generally try. According to you, what is the best way to inspire action?

I completely agree... and only by experiencing it do I truly appreciate what a challenge inspiring change on a large scale is. I think the main learning points I have taken from this experience are:

Firstly to realise that most people don't like change. Especially in a pressured environment like ours. If they feel it may involve some element of hard work, they will shy away and rather carry on as they are... even when that involves a lot of complaining about their daily grind! Realising this allowed me to not take it personally when 90% of the people I worked with didn't get the epiphany I did when I first found CDT.

Secondly, realise that even the majority of those who are open to change, want to see some evidence of success first. They don't want to jump until they know it's safe.

Lastly, have patience! My boss always likened it to turning the direction of an ocean going tanker. It takes time and planning!

A great source of learning for me was Simon Sinek's TED video titled "How great leaders inspire change". Not only did his 'Why, What, How' approach help me to formulate the presentations required to convince people CDT would benefit them, but his graph on the 'Rate of adoption' reset my expectations of how long this would take. At the beginning of this year, the CDT rollout became my full time job. At that point I knew I was at an adoption rate of about 10-12% of projects. Pretty low; considering I'd been at this (part-time) for 2 years. But I looked at Sinek's graph and realised I was at tipping point... it gave me new hope to dive into the full time role. I set my expectation that; if by the end of 2013 I had reached 40% of projects then I was doing well. Well, there's still 3 months of the year to go and I know the rollout has inspired 60-70% of projects to adopt some techniques. The biggest driver of this change has been the Visual Test Strategy/ models.

So to sum up a long story... my advice to inspire change.

1. Set yourself realistic targets.
2. Find the 5% of innovators who actively seek out change.
3. Work with them to give everyone else proof/ evidence/ case studies they need to come on board.
4. Work hard on how to communicate the benefits to other people and my last piece of advice, make sure one of your supporters is someone senior who will back you.

The reality is, 2 years ago 90% of the very large team I work in, thought CDT was a fad, my boss and I were crazy and this would all go away! A lot of those same people now come to me for advice on implementation. I've stuck the course and proven that CDT has delivered extra value to every project using it, evidenced by some excellent written feedback from project teams.

There are many women who are testers but very few are seen at senior positions, even today. What would be your advice for lady testers who aspire to gain senior positions in testing?

Actually, when you look at the contributors to this edition of TTWT, we can see there are quite a few women now making a name for themselves as experienced testers. But I agree that's a recent change. Other than Dorothy Graham, there were very few women on the conference circuit until the last year or two. I'm assuming here that when you say Senior, you mean 'well known' but actually there are many senior female Test Managers out there in organisations. I guess women have traditionally taken the burden of raising the family, which can impact careers, but I think that is less and less an issue, as our global society gets more comfortable with equality at work and at home.

So advice for anyone wanting to build seniority... get involved with the testing community. As Teri Charles says in her article, they really are a welcoming, supportive and friendly bunch.

Another piece of advice, pick someone you consider to be Senior and model them. That doesn't mean copy everything they do. You should always maintain yourself and your integrity... but do study them a little. Consider what makes them a senior tester in your eyes? Then match your own skills and experiences against theirs and you will find your plan to achieve that more senior position will quickly become clear. Ultimately, be brave enough to work hard at your goal. Seniority comes from the recognition and respect you get from your peers, not from the years experience you have behind you.

What makes a good tester? Why do women make good testers?

Good question. James Bach delivers a great presentation on what he believes makes an excellent tester. Some people may be surprised at some of the elements he includes: cognitive science, applied epistemology, communication skills and many more. Read his study [here](#).

So I can't give as comprehensive an answer here but personally I would agree a good tester should be many things: inquisitive, technically savvy (not always an expert... but not confounded by it), have attention to detail but most important of all, to be a people person. Unless you can communicate effectively, all the technical skills in the world will not help you be recognised as a great tester. Actually, a test team can be made up of many people whose proficiency levels may vary for each of these skills. This would make a dream team. But an exceptional individual tester needs to balance all of these. I know many brilliant male testers so I'm taking nothing away from them by saying that women are often very good at understanding and communicating with people... and multi tasking. And as Test Managers, they often have less confrontational ways of delivering those hard/ bad messages to project teams. But it still ultimately comes down to the desire to improve your skills in your chosen field.

Inspired Tester according to you is.....

Someone who takes pride in their work, who wants to get better, who wants to learn and will put in personal effort to do so. You may be a lucky tester who gains inspiration from the people around you in your 9-5 work but truly inspired people do not stop building their skills and careers when they leave the office for the day.

What is the biggest challenge in the testing field today, in your opinion?

I truly think it's the perception of testing. Several large multinational companies have made a vast amount of money by dumbing down the profession. In order for them to make this money they have played down the importance of skilled testers in favour of 'processes and tools'. Unfortunately they are the ones who get to speak to the senior executives of large companies and therefore set this perception. Which also, very sadly, seems to have got into the psyche of a lot of testers! It's really time we testers fight back and become more vocal about the importance and value of skilled testing.

The AST (<http://www.associationforsoftwaretesting.org>) and ISST (<http://www.commonstesttesting.org/>) are two organisations trying to achieve this.

Who are the people or books that you take inspiration from?

As you can tell from this interview, James Bach was my first inspiration to improve my testing skills. And the book he co-authored with Cem Kaner & Brett Pettichord (Lessons learned in Software Testing) is a fantastic resource for testers. Most of my inspiration lately comes from blogs and online content. I love that there is something new to read and share every day. That could be about testing or communication, psychology, technology etc. My advice... get a twitter account and pick a few key people to follow or start with a more generic account like Software Testing Club, which shares links to many testing blogs. I also mix that up with Harvard Business Review, Sir Ken Robinson (am very interested in education at the moment) and anyone else who is sharing informative articles that make me think a little differently each day.

Your opinion about Tea-time with Testers? Any message for our readers?

I'm a huge fan of TTWT. The passion you guys demonstrate to deliver such thought provoking content each month is an inspiration to me. I particularly like that you are discerning with your articles. You don't just print the same old testing mantras just to fill some pages. This is evident with the very well recognised and respected authors you get to contribute each month.

And my message to you readers....

...by taking time to read more about your craft you are already more inspired than the average tester! But most important, if you read something that inspires you, be sure to pass it on to someone else who may also benefit. Sharing knowledge is how our work lives get enriched. Pay it forward...



Call for Articles !

Have you got something to say?

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

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

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

Submit your articles, stories, thoughts around software testing.

now its your chance to be heard...!

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

testing intelligence

- *its all about becoming an intelligent tester*



an exclusive series by **Joel Montvelisky**

It's not only about testing!

"It's impossible to run all these tests in the time we've got left!"

How many times have you said this phrase?
I know I've said it tens if not hundreds of times...

Looking back, I must have sounded like Star Trek's Dr. McCoy with his:
"I'm a doctor, not a..."

Funny thing is that also looking back I think that about 80% of the times I said this phrase we were eventually able to release the project on time (or very close to it), either because we managed to run most of the tests we needed to run, or because (believe it or not!) we found a way to release the project without running all the tests we had planned

Do you ALWAYS run all your planned tests?

Unless you are working for NASA, a life-sciences company, an aerospace firm or another type of “regulated industry” company the answer will surely be NO.

I bet that not only did you “not-run” some of the tests you had planned, but also that during the actual test execution phase you realized that you needed to run some additional tests you had not anticipated during the planning stages of your project.

And so the question is, should you feel bad about this?

I mean, if you look at it from a traditional “project planning” perspective you either don’t know how to plan your tests, or you are letting other managers bully you into taking exaggerated risks by not running important parts of your test plan, or both.

Or is there another alternative?

OK, I am being a little harsh here, but it is only to make a point 😊

After all there might be a third option: Maybe the truth is that your project is simply too dynamic, and at the rate things are constantly changing it is extremely hard (or plain out impossible) to plan all your tests up-front!

– If your projects are anything like mine, then I guess you were nodding your head as you read the last sentence. –

The Crystal Ball approach

A wise tester (please don’t ask me who, because I really don’t remember) once said:

“If I could provide the same information about the product by looking at a crystal ball, without running a single step, I would still be doing my job as a tester.”

To this I usually add, that not only would you be doing your job as a tester, you would be doing it more effectively 😊

But what does this has to do with test planning or ever-changing projects?

It has everything to do with it!

Your job as a tester is not only about testing!

As the wise tester said, our objective is not to test but to provide information. Testing is only one of the tools we use (maybe the most powerful tool we possess) in order to gather the data. But as with any tool, you need to use it correctly in order to meet your purpose.



If your projects change so should your test plans

Fighting project change is like fighting the Sun and the winds, a total waste of time and efforts. In the best case you will seem like a grumpy tester, and in the worst case you will be the team buffoon that no one takes seriously anymore and who is constantly stating that they won't be able to release the project on time...

If this is the case then instead of fighting change, plan for change.

There are a number of things you can do in order to cope with change as part of your test planning:

1. Make high-level plans and then low level plans

Instead of making very detailed and low level test plans for all your project (up to 6 or 9 months ahead of time), make high level plans and estimates. Then, as the project progresses, take the most immediate tasks (2 to 4 weeks in front of you) and create lower level plans ONLY for these parts, taking into account the changes and actual status of your project.

2. Exercise mental flexibility

There are many definitions for mental flexibility, but in this context I mean the ability to cope with change and to react to it in a productive and constructive way.

Instead of getting mad at people who will come to you with new features and issues that will make you change your plans, take these changes as constraints that cannot be blame on anyone. Then look at them as your challenges and solve them by thinking what is the best thing you can do based on this new reality facing your project.

3. Plan your tests as modular building blocks

I love Lego, whenever I want to build something there are always countless ways of arranging the small pieces. If half way across my building I am missing a piece I can always improvise and still make something close to what I was aiming to do.

BTW, today I do this mostly with my son, but I still find myself something buildings things without even thinking about it as kind of a mental relaxation exercise.

But back to our changing projects 😊, your approach to testing can be similar to building with Lego. If you plan your tests as modular blocks that can be used in order to create scenarios and achieve different levels of coverage, then you should not have a problem shuffling these blocks in order to answer your current testing needs. If an unforeseen change suddenly appears, then look in your virtual lego-box and find the combination of tests that you can put together to supply answer this challenge. Even if in practice it is more complicated than this (and it is a lot more complicated than this!) having modularity in your tests will provide you with much more flexibility and higher degrees of freedom to help cope with change.

4. Make it a habit to ask your stakeholders what information they need NOW

Back to our "crystal ball", our job is to provide information (testing is only the empirical tool we use) and just like your project is constantly changing so will the information needs of your team change. Make it a healthy habit to regularly ask your project stakeholders what information they require and what is important for them that you test in order to make their decisions.

Providing useless information can sometimes be even more annoying to them than not providing any information at all...

How do you cope with project change?

There are many other things you can do to cope with change. What do you do as part of your daily work to achieve this?

If you have additional ways and ideas I invite you to share with us.



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

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

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

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

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

T ' Talks



T. Ashok exclusively on software testing

Change. Continuous motion. Cadence.

We all know that change is constant. We also know that adapting to change is hard. We resist change.

An adaptive system responds. It responds rapidly to change. In nature, this is a key to survival. And also key to delivering high performance.

Change is challenging and worrisome. We resist because we are typically afraid. Afraid of possible bad outcomes. Afraid the risk it puts us in.

To adapt to change requires information. Information that we can use confidently to respond well.

Let us look at how we respond when changes are done to software. When a change is done to software, we always get into a mode of regression. Worrying about 'will what was working-before continue to work as the way it was'? And then also worrying about 'how might the changes affect other stuff negatively'? And our response is to use our experience to guide us to choose a subset of the larger set of test cases to regress in case they are non-automated. Or choosing a large set of test cases that we think is necessary to be run always and automating them.

Let us go a little deeper. A change is done i.e. code or configuration has been changed to change the behaviour. What is behaviour? A set of conditions that entity should satisfy. So when a change is done, certain new conditions have been added, existing conditions modified or deleted. So our worry is if those parts of behaviour that still are the same i.e. those combinations of conditions that are unchanged function the same way as there were before. So the information that we should elicit therefore is about what conditions are new/modified/deleted. Note that conditions may be of various types - data related (syntax/format, size, rate, type etc), behaviour related (functionality), load-handling related, performance-related, security-related and so on.

The second aspect of the change is the domino effect. What other parts of the system can the change impact? And what kind of impact - functional behaviour, performance, security etc. To know what other parts of the system may be affected, we need to have a good understanding of the linkages between the various requirements/features/system-elements i.e. what requirement/feature is connected to what requirement/feature. Having understood the linkages, the next step is to analyse/qualify the linkage based on the change i.e. what conditions of the linked element would be affected by the change done?

Let us look at this a little differently, from the perspective of validation. What we are keen to understand is the fault propagation ability of a change i.e. how could a fault in the system created due to a change propagate via the linkages and ultimately cause a failure? So correlating the changes done to an entity to the potential types of defects (of that entity) that could be irritated allows us to choose the appropriate test cases to regress that entity. Proceeding further, analysing as to how an irritated potential type of defect could propagate along the linkages i.e. affect the other elements (by irritating its potential types of defects) allows us to select the appropriate test cases for the linked entity. Interaction matrix and Fault Propagation are some of the techniques used in Hypothesis Based Testing (HBT) to stay on top of changes to ensure optimal and effective validation.

Using information about conditions, potential defects types, linkage and fault propagation allows us to respond to changes well and adapt testing constantly.

Constant adaption is wonderful as it results in fluid continuous motion. A beautiful feeling of aliveness. And that is what nature is about. Continuous morphing, resulting in improvements. So as software evolves, enjoy it. Adjust and adapt - the test strategy plan, test cases and tooling.

In cycling/running there is an interesting concept of cadence. In the case of cycling, cadence is about how many times/minute you rotate the pedal. When you move from a plain terrain to a climb (change), the speed will drop. So the natural response is expending more power by pedaling harder to maintain the speed. The challenge is that heart rate rises rapidly and it is a just a matter of time when this becomes difficult. This is where cadence comes handy. Instead of expending higher energy, shifting to lower gears and rotating the legs (pedals) faster results in maintaining the same speed on the incline. So it is typically recommended to spin at a higher cadence to compensate for the change. Correlating to what we do, my take is that cadence is about continuous motion to enable a good response to the change. High cadence implies nimbler movement and using information about the terrains as we ride enables one to respond well. Now I like climbs and look forward to climbs!

In running the concept of cadence is about the number of steps we do in a given time. Rather than run with large distant steps, it is suggested that you take small steps, and take more steps during a minute. Taking small steps and more of these per minute allows one to use energy optimally yet achieve higher speeds.

Summarising the rate of work ("cadence") enables us to respond to "change" well (of course we need good information) enabling a "continuous motion" (that is ultimately a state of bliss).

This being a special issue on "Women in testing", I cannot resist connecting these concepts to this theme. In my opinion women handle changes and adapt very well. They manage multiple things (spouse, kids, family, house, work,...) and each of these 'change' (most often cause problems!) continuously and they seem to juggle these very well. To all of you there, my salute!

So next time when a feature changes, enjoy it. Develop a high cadence process/suite that allows you to respond rapidly and enjoy the continuous motion.



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



[Back To Index](#)



Testing PUZZLES

by Sebi



Claim your **Smart Tester of The Month** Award. Send us your answer for Puzzle b4 20th October 2013 & grab your Title.

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

Puzzle

If you go to the URL (hypothetically, the actual url on the website doesn't do this. but let's say it does for the purpose of this exercise) www.testalways.com/index.php?regex=Li4vLi4vLi4vLi4vLi4vLi4vLi4vLi4vLi4vZXRjL3Bhc3N3ZAAucG5n

you see this content:

```
'root:x:0:0:root:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

```
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
sync:x:5:0:sync:/sbin:/bin/sync
```

```
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```

```
halt:x:7:0:halt:/sbin:/sbin/halt
```

What is the URL that shows this content:

```
'# Do not remove the following line, or various programs
```

```
# that require network functionality will fail.
```

```
127.0.0.1 localhost.localdomain localhost' , based on the previous one?"
```

[Back To Index](#)

Biography



Blindu Eusebiu (a.k.a. Sebi) is a tester for more than 5 years.

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

He tweets as @testalways.

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



Every Tester

who reads Tea-time with Testers,

**Recommends it to friends and
colleagues .**

What About You ?

in ne>xt issue

articles by -

A close-up photograph of a silver-colored metal tag with rounded ends, hanging from a chain. The tag has the words "IT'S ALWAYS TEA-TIME" engraved on it in a stylized, hand-drawn font. Below the tag, a small, round, silver-colored metal tea timer is visible, also hanging from the chain. The background is a dark, textured surface.

IT'S ALWAYS
TEA-TIME

Jerry Weinberg

T Ashok

Jim Holms

Bernice Ruhland

Joel Montvelisky

and some surprises...!!!

our family

Founder & Editor:

Lalitkumar Bhamare (Mumbai, India)

Pratikkumar Patel (Mumbai, India)



Lalitkumar



Pratikkumar

Contribution and Guidance:

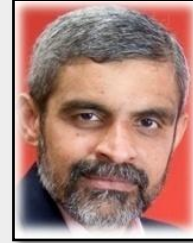
Jerry Weinberg (U.S.A.)

T Ashok (India)

Joel Montvelisky (Israel)



Jerry



T Ashok



Joel

Editorial | Magazine Design | Logo Design | Web Design:

Lalitkumar Bhamare

Core Team:

Dr.Meeta Prakash (Bangalore, India)



Dr. Meeta Prakash

Testing Puzzle & Online Collaboration:

Eusebiu Blindu (Brno , Czech Republic)

Shweta Daiv (Mumbai, India)



Eusebiu



Shweta

Tech -Team:

Chris Philip (Mumbai, India)

Romil Gupta (Pune, India)

Kiran kumar (Mumbai, India)



Kiran Kumar



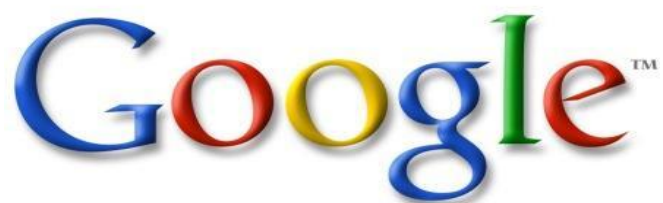
Chris



Romil

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

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



Join our community on



Follow us on



www.teatimewithtesters.com

