

Tea-time with Testers

FEBRUARY '14 | YEAR 4 ISSUE 1

Jerry Weinberg

Make-works and take-credits

Jim Holmes

Working with locators – part 2

Ruchi Chourisia

Invaluable Testers

Dr. Sanjay Gupta

Not just numerical values

T Ashok

It is value, not activities that matter

Jyothi Rangaiah

DISSCOH - Heuristic Approach to Bug Reporting

Joel Montvelisky

Can you test without knowing the names of your users?

Over a Cup of Tea with Jerry Weinberg

Musings over Tea-Time

Anthology of T-Talks

By T Ashok

A black and white photograph showing the silhouette of a person from the chest up, facing right. They are holding a microphone. The background is slightly out of focus, showing some text like 'he' and 'ng'. A horizontal grey band is overlaid across the middle of the image.

COMING SOON

Celebrating THREE years of Tea-time with Testers. Anniversary Special!

Asia Pacific's Largest Software Testing Conference

11th International Software Testing Conference

STeP-IN
SUMMIT 2014

Testing **NOW**

Interaction » Insights » Opportunities

JUNE 25-27, 2014 AT BANGALORE &
JUNE 18 & 20, 2014 AT PUNE & HYDERABAD

Call for Speakers

Submissions are invited for Speaking Opportunities & Best Publications

STeP-IN SUMMIT 2014 Best Speaker

Here's a chance to be recognized as a great speaker / contributor in the Software Testing space.

A great opportunity for:

- Industry Recognition
- Acknowledgement on the conference website, and through social media
- Complimentary passes to STeP-IN conferences for the next two years
- Cash awards
 - Audience Choice Best Speaker Award – INR 10,000/-
 - Jury's Choice Upcoming Star Speaker – INR 10,000/-



**For more details visit: www.stepinsummit.stepinforum.org
email: papers2014@stepinforum.org**

Produced by:

STeP-IN
Forum
www.stepinforum.org

Hosted by:


The Knowledge Corporation
www.qsitglobal.com



TEA-TIME WITH TESTERS

First Indian testing magazine to reach 115 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 or claims made by advertisers in this ezine do not necessarily reflect those of the editors of **Tea-time with Testers**.

Editorial



Three years already!

Looking back to look forward...

Back in 2011, our minds were ignited with an idea of testing magazine. Not that there weren't any magazines already but we wanted to achieve something different. Telling testers about basics, best practices and things they already knew was not going to help. We wanted to create magazine that would give voice to testers and testing community as a whole.

The best practices in industry were not really solving our testing problems and being hands-on testers, we realised the need of platform where testers could share ideas, their learning, their experiments and experiences and could express their opinions, boldly. On other hand, it was also important to let them know what great minds in industry were talking about. And we finally figured out...what we were going to do via Tea-time with Testers.

We started reaching active testers and experts in industry and eventually we got overwhelming response. The reason was obvious; they liked what we were doing and how we were doing it. I can't thank enough to all people who have contributed to this magazine. One after another we launched different initiatives like Bug-Boss Challenge, Blogger of the Month Contest, S.T.O.M. Contest, Teach Testing campaign, Women in Testing special edition and all of these have received wide appreciation in community. I am glad to share that our State of Testing Survey-2013 has been a big success, so much so that our friends from Japan are working on its Japanese translation and they are also conducting a new survey especially for Japanese Testing community. We have been generously supporting various testing conferences world-wide, since our inception.

After 3 years; as I look around myself; I see that things are changing. There is more awareness than it was before. There are more 'thinking testers' than what I had seen 3 years back. There are more conferences, contests and tester meet-ups. And now, more publications talking about context and real testing.

I don't know yet how far we have achieved what we wanted to achieve but if this is what called 'intellectual testing culture' then I can proudly say that Tea-time with Testers does have its small share in building and promoting it.

And for years to come, we will continue to give our contribution towards building this profession.

Yours Sincerely,

A handwritten signature in purple ink that reads "Bhamare".

- **Lalitkumar Bhamare**

editor@teatimewithtesters.com



QuickLook

3 years
down,
forever
to go!



Editorial

What's making News?

Tea & Testing with Jerry Weinberg

Speaking Tester's Mind

Invaluable Testers – 33

A cup of Tea with Jerry Weinberg

In the School of Testing

Working with locators – 38

Not just numerical values – 42

DISSCOH - Heuristic approach to Bug Reporting – 48

Can you test without knowing the names of your users? -60

T' Talks

It is value, not activities that matter – 56

Testing Puzzle – S.T.O.M. Contest

Family de Tea-time with Testers

“Your ideal form of influence is first to help people see their world more clearly, and then to let them decide what to do next”, is what I have learned from him and that’s how he has been helping many others like myself.

“I don’t think he really needs to be introduced but just in case you have been living in a cave and have never heard of him, I think that the best way to start his introduction is to say that he is a special kind of person, a “people changing person”. He has the incredible gift of, in his own words, “making you aware of the things you were not aware of” and thus changing the way you look at things and the way you approach tasks”. That’s how Joel Montvelisky describes him.

I firmly believe that books, blogs and articles he has written are invaluable assets for our industry and they are going to guide us for many years to come.

He has been guiding us right from our early days and we feel humbled for having him as part of our family. Interviewing him was my dream and I am glad that it has finally come true. I must say, this anniversary issue has become even more special with this special interview.

Yes, you guessed it right. Interviewing **Jerry Weinberg** has been a great experience and I am sure, you too will enjoy it.

- Lalitkumar Bhamare

Over a Cup of Tea with Jerry Weinberg



Does Jerry Weinberg need any new introduction? We all have known him for years in various roles like a prolific writer, author of over 80 books, consultant, coach, mentor, philosopher, computer scientist, novelist...and the list continues. If we are allowed to ask, how would you like to introduce yourself? We are eager to know about Jerry Weinberg from Jerry Weinberg himself.

I've always been interested in helping smart people be happy and productive. To that end, I've published books on human behavior, on leadership, and stories about smart people—how they produce quality work and learn to be happy. My books may be found linked from my website: geraldmweinberg.com. I've won many awards for my writing but the "award" I'm most proud of is the book, *The Gift of Time* (Fiona Charles, ed.) written by my student and readers for my 75th birthday. Their stories make me feel that I've been at least partially successful at helping smart people be happy.

Are there any special memories from your childhood that you'd like to share especially some moments that made you fall in love with field of computers?

At age 11, in 1944, I read a story in Time magazine about "giant brains," as computers were sometimes called back then. I fell in love with the idea that these "brains" would help me understand my own brain, which was the source of great happiness for me at the same time it was the source of much mean criticism and punishment. I knew then that I would spend my life with computers. I was right, at least for the next 70 years so far.

And how did Gerald become Jerry?

I always wondered about that, myself. I was never called "Gerald" in my entire life, so at age 50, I gathered enough nerve to ask my parents: "If you were going to call me Jerry, why did you name me Gerald?" Their answer, from each of them independently: "I don't know."

You have written hundreds of articles and over 80 books. We are curious to know about your love for writing. What was the start point and what was your inspiration?

I cannot remember any time I didn't write, or love writing. I used to write little stories for my father before I was old enough for school. This is like asking, "When did you start breathing, and what was your inspiration for wanting air?"

How do you find such ideation, inspiration and energy to bring in more and more for your ever thirsty readers?

As I live in this world and look about me, I see unhappy people everywhere, and I think, "It doesn't have to be this way." I perceive that much of this unhappiness arises from ignorance, so when I think I can clear up some of this ignorance, I write.

In one of your earlier interview, you said, "Testing has barely been born yet. As IT matures, so will testing. Without testing, IT will never mature." According to you, what are the characteristics of mature testing and how do we benchmark it?

When testing is finally mature, many of today's common questions will disappear. For instance, "When do you start testing?" That will no longer be a question because professional developers will understand that testing does start the instant you first think it might be possible to build something. Testing will not be something stuck on at the end of a development project, and professional testers will be part of the development team the entire way, from that first thought to the ultimate retirement of the product. My book, **Perfect Software and other Illusions about Testing**, details many of these questions that will no longer be asked.

You have been involved with the field of computers for over 50 years. As compared to its state in past, how much do you think has software testing evolved? Do you suppose that we are headed in right direction and have reached any significant point that you thought should have been achieved by now?

In many ways, testing has moved backwards in 50+ years. We have better tools now, but they are not used by the majority of projects, so what good are better tools? In general, the standards for quality have deteriorated. For instance, in the past year, I have tried perhaps 50 different apps, and every single one of them has displayed serious and obvious bugs that would never have survived even minimal testing. That said, there are some exemplary testing organizations, but they form a small percentage of all the people who claim to be software builders.

What do you think and believe is still left to discover / unearth in field of testing?

Discovering new things about testing is the wrong goal. Right now, we know enough things, but we don't use them widely enough. If we want to improve, we should be concentrating on actually doing those things we know are good testing practices. I believe **Tea-time with Testers** is one organization that's trying to encourage this.

You have co-founded and hosted AYE (Amplify Your Effectiveness) conference.

What differentiates AYE from other conferences?

All our conferences and workshops are based on experiential learning rather than lecturing. To understand this approach, I recommend my **Experiential Learning** sequence of three books.



Would you like to share some memories from past AYE conferences?

No, that would be lecturing/listening rather than experiencing. You'll have to experience one of our offerings to understand the difference, and the experience is different for each participant. Perhaps you would get some understanding by asking your readers what memories they have from AYE or Problem Solving Leadership (PSL) or some other experience with us.

What is your opinion about standardization of processes and practices, considering it for software testing field in particular?

My greatest problem with standardization in general is that those who set standards do not start and end with the question, "What will this standard do for us?" Without answering that question, we wind up with standards that are not tested—and even worse, are not testable. That's hardly what we want in our testing standards. Fortunately, most "standards" for software processes are not followed, so they don't do as much harm as they could. Perhaps we should be working on guidelines, rather than standards that someone hopes will enable us to test without using our brains.

According to you, how important is domain knowledge for a tester to become an expert tester?

What a tester needs is not domain knowledge, but the ability to pick up meaningful domain knowledge quickly and correctly.

You have also been a consultant in your career, what according to you is the top secret of consulting?

That there is no such thing as a “top” secret in consulting. Good consulting is so complex, so nuanced, that there is no one “top” secret that applies everywhere. (I guess this is a meta-secret—a secret about secrets—but I don’t think there’s a “top” meta-secret, either. Perhaps that’s the top meta-meta-secret, or else it’s meta-secrets all the way down.)

What are those non-computer related skills that helped you become a successful consultant?

It wasn’t so much skills, but my intense drive to learn anything that might help me become better at helping others. I’m not an especially skilled learner, but that has helped me because I tend to understand why people have trouble learning something. I can then use that understanding to help them learn. Some consultants are just too smart to be good at helping others, because they, themselves, never had difficulty learning something, so they are impatient with slow learners. Being a slow learner myself, I don’t have that problem. Perhaps one of my best skills is knowing how to be patient—something I was slow to learn.

Humans created computers and programmed them. Is there anything that human should learn from computers? (or something that you learned?)

Just to pick one thing, I’d say that the most important thing I learned from computers was that I wasn’t nearly as smart or perfect as I used to think. When I program a computer, it does exactly what I said to do—and that shows me how dumb and imperfect I am.

What are the key ingredients to make one a successful test manager?

Oh, my, that’s a whole book. Or two. I’m working on that now, but I have no idea when I’ll be ready to publish. For the moment, the first thing that comes to mind is “humility.” After that, I guess the manager must take on the role for the right reasons—or at least not for the wrong reasons such as money, power, or status.

According to you, how important are test metrics in project release management?

In actual practice, not every important. But if you wish to do release management well, the right test metrics are supremely important. What else should be more important for deciding when and how to release a product? Schedule? Promotion for the test manager? I don’t think so.

Do you think testing can be measured in meaningful way with standard metrics that are being followed in industry today?

NO. First of all, there are no “standard metrics” in the industry today, though there are lots of metrics that their vendors would like you to believe are meaningful and should be standard.

When not reading and writing, what can Jerry Weinberg be found doing?

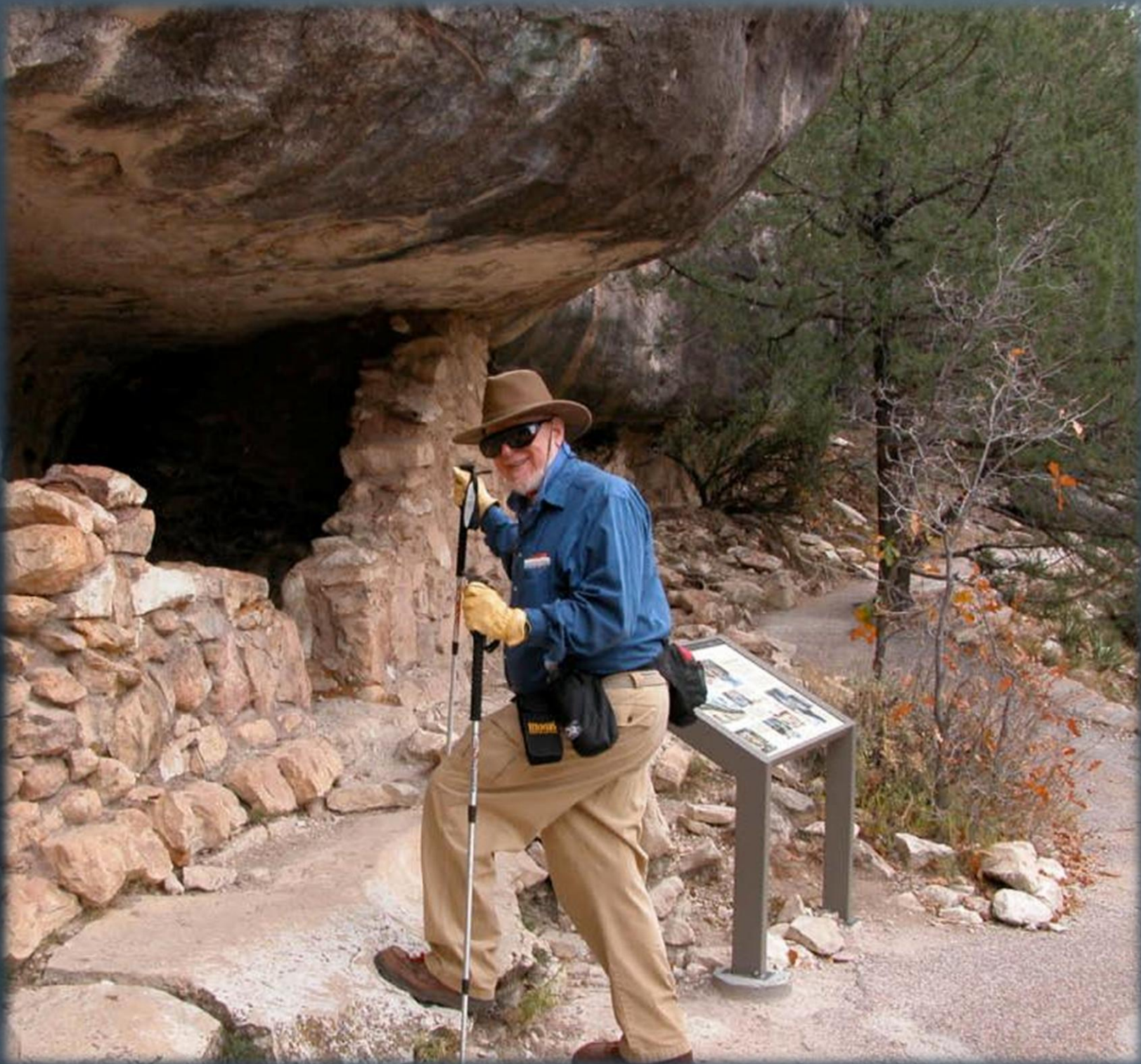
Sleeping. Spending quality time with my wife and my children and my grandchildren and my dogs. Hiking. Helping smart people be happy and productive.

Your message to the testing community would be...

When you’re finished learning, you’re finished as a tester.

Last question, you have given invaluable contribution to development of Tea-time with Testers magazine and have been deeply involved in its content. Team TTwT is ever grateful to you for that. What made you join this family and how do you feel being part of it?

I feel proud to have some small part in what you're attempting to do. I joined the family because I thought you were doing the right things, which is a very rare but essential part of building the testing profession.



What's making News?



QASymphony Unveils Free eBook for Testers

Free eBook outlines techniques for QA departments working on Agile software development



ATLANTA, GA – February 10, 2014 —

QASymphony (www.qasymphony.com), a leading developer of Quality Management solutions for software developers and QA testers, today announced the release of **"What you Need to Know about SW Testing in the Agile Era,"** a free eBook aimed at the testing community. This publication is designed to offer up useful techniques for the modern testers of today. The eBook can be downloaded by clicking [here](#).

Taking into account the Agile revolution in software development, the rise of cloud computing and the growing popularity of mobile devices, this eBook imparts practical advice from industry experts Vu Lam, CEO of

QASymphony, and veteran practitioner Sellers Smith, Director of QA at Silverpop.

It highlights the impact of major changes in the way software is developed, suggesting the right approach to tackle any new testing project.

Broken into four short articles, "What You Need to Know" is a vital read for testing practitioners in the field. It identifies a real need to update and re-imagine roles to cater for the Agile methodology and explains why the right testing methods are essential. Practical steps to achieve the changes required are laid out in the following order:

1. Reimagining the Tester for the Agile Age
2. Finding the Right Mix with your Testing Methods
3. Surprising Truth about Exploratory Testing
4. Building Good Tests: Why, What, How?

"It's time to reevaluate how we approach testing and to blend techniques that are capable of providing the transparency and flexibility that modern software development demands," says Vu Lam, CEO of QASymphony. "We hope to foster a greater spirit of community across the testing profession."

QASymphony is the developer behind qTest, a powerful test management solution designed to serve the needs of testing professionals. The company most recently announced qSnap, a free screen capture and documentation plug-in tool that works across all popular browsers. The tool can be downloaded at <http://www.qasymphony.com/qsnap.html>.

About QASymphony

QASymphony is a leading provider of testing solutions that fit the needs of testing organizations at any level of maturity. Whether you are making the initial move from manual processes and need basic management help or you have processes and tools in place and are looking to enhance productivity, our test management and agile testing solutions can help you test more effectively. With HQ in Atlanta, GA, presence in Dublin, CA and operations in Ho Chi Minh City, Vietnam, QASymphony is a software company built to revolutionize how software is tested, adopted, and supported. Empowering the QA testing teams for companies such as Good Technology, Silverpop, BetterCloud, and Zappos, QASymphony is a software-loving team, united by a common belief that software can be better and better tested.

Website: www.qasymphony.com

Facebook: www.facebook.com/qasymphony

Twitter: www.twitter.com/qasymphony

Press Contacts:

Victor Cruz

Principal, MediaPR

vcruz@mediapr.net

Sarah Lander

VP of Marketing, QASymphony

sarahlander@qasymphony.com



TESTING CUP 2014

2-3 JUNE, WARSAW
POLISH NATIONAL STADIUM

Polish Championship in Software Testing will be held at the Polish National Stadium in Warsaw. The event will take place on June, 2-3.

This is the second attempt to find the best software testers in Poland. The first TestingCup took place in September 2013. More than one hundred testers faced Mr. Buggy - the application created especially for the occasion. The organizers have announced that this year's two-day championship will welcome doubled number of participants.

The championship consists in testing an application on the basis of specification delivered. Pure satisfaction of winning is not the only prize that winners can receive. In 2013 total award was 25 000 PLN (~6 000 EUR).

Everyone can participate no matter how experienced in testing he or she is. The competition is divided into two categories: individual and team. **Registration is now open.**

TestingCup is not only a competition for testers, but also a testing-orientated conference with both Polish and foreign speakers attending. If you want to become a speaker at the event, please contact TestingCup team before 10th of March at testingcup@testingcup.pl.

We encourage you to visit TestingCup website: <http://testingcup.com> and follow us on Facebook: <https://www.facebook.com/TestingCup>.



A million dollar smile ?

Ask our sales team about
our **Smiling Customer** 😊 programme

*Adverts starting from \$100 USD | *Conditions Apply

Tea & Testing



with

Jerry Weinberg

Make-works and take-credits.

Not all bureaucratic problems can be solved with a smile. For one thing, they often come clad in writing, and how can one smile in a memorandum? For instance, imagine you work in an institution which habitually circulates memoranda such as the following;

TO: All Staff and Ships at Sea

FROM: His Eminence, The Dean

SUBJECT: Utilization of Commas in Weekly Punctuation Reports

It has recently come to my attention that the utilization of commas in the weekly report concerning the quantity of punctuation employed by each member of the staff has been confusing to certain members of the Board. According to my analysis, the problem seems to lie in the inability of certain members of the staff to distinguish between commas utilized in the text for purposes of punctuation and commas utilized in the text for purposes of reporting upon the utilization of commas in texts.

I would suggest a solution along these lines, with details to be worked out by a Committee on Commas which I shall presently appoint:

1. Commas utilized for purposes of punctuation should be left as normally used in English or American texts.

2. Commas reporting upon the utilization of commas in other texts, or in their own texts, should be enclosed within single quote marks, like this: ','

2a. As a potential alternative to part 2 of this solution, the commas reporting upon utilization of commas in other texts, or in their own texts, should be enclosed within double quote marks, like this: ", "

I am circulating this memo to the distribution list in order to obtain the fullest possible feedback from the entire staff concerned with this pressing problem, so that we all may input our creative and innovative ideas.

As this phenomenal memorandum circulates, there will be an abundance of smiles, none of which is likely to be given to the face of His Eminence. What is to be done? Can we get a clue from asking, once again:

WHERE DOES THE PROBLEM COME FROM?

Whenever we find a vast circulation of bureaucratic activity, full of sound and fury, signifying nothing, we may be facing a problem that comes from nowhere. Or, more precisely, that comes from the problem itself. A classic example of this sort of self-perpetuating problem is the International Conference.

While writing my book "Are your lights on?", we had the pleasure of reading about yet another International Disarmament Conference taking place in Geneva—a place where the International Conference has been developed to a high art form. Could it be, the Geneva resident speculates, that disarmament problems are so intractable because disarmament conferences are so attractive?

What would happen to the Arms Race if the disarmament conferees began working at 6:30 in the morning, like honest Swiss workingmen and workingwomen do? Or if the chairs were hard wood instead of soft leather? Or if the meals were frozen fish sticks and soggy potato chips at an Akron drive-in, rather than omble chevalier and pommes anglaises at Geneva's Le Senat?

Now don't get us wrong. We have nothing against disarmament. Nor do we demean the distinguished ladies and gentlemen whose arduous labors for the welfare of the poor, tired, huddled masses must occasionally be relieved by a spot of gaiety and pleasure. We merely wish to indicate the bare possibility of the problem-solving process, person, or institution becoming the problem itself.

In the old days, when radicals were really radical, they used to say, "If you're not part of the solution, you're part of the problem." It could well be different: "If you're part of today's solution, then you're part of tomorrow's problem." Come to think of it, what did happen to those old radicals—they definitely were part of the solution.

Speculate with us for a moment, just for speculation's sake. What would happen, after all, if the Nations of the World suddenly disarmed? Would the patisseries in Geneva suffer reversals, and half the bureaucrats find themselves transported—tourist class—to the Akrons of the world?

Not bloody likely!

Recent history is strewn with examples of famous problem-solving institutions whose problems have disappeared. They didn't pack their formal wear and catch the next Swissair flight home. No, indeed! Instead, they sought another problem to solve. The veritable end of polio didn't see the March of Dimes come to a Halt. There were plenty of other worthy diseases to conquer. The end of the War (any war) didn't see the march of the armies come to a halt, either.

True, they became "standing" armies, but that doesn't mean they stood around doing nothing. More often than not, a standing army finds domestic problems urgently claiming their peculiar talents.

In short, the ultimate source of the problem may be nowhere at all. In other words,

IN THE VALLEY OF THE PROBLEM SOLVERS,
THE PROBLEM CREATOR IS KING, OR PRESIDENT, OR DEAN.

Which brings us back to our original problem—how to handle the memo concerning commas?

Your father's father may have told you, "There's two kinds of people in the world, those that do work and those that make work for others to do. Keep away from the make-workers and you'll do all right."



Figure 1. There's two kinds of people in the world, those that do work and those that make work for others to do.

Or, your mother's father may have said, "There's two kinds of people in the world, those that do work and those who take credit. Keep in the first group—there's much less competition there."

Either of these remarkable observations can be used to solve the memorandum problem.

One way is to be physically separated from the people who generate memoranda. Such a separation is most readily accomplished by creating one small group of elegantly decorated offices—preferably on the top floor of the tallest available building—far from the drab work areas. How will you get administrators to move to them? How do you get bees to come to a pansy patch? Flies to a dung heap? Without any fuss, the administrators will wind up in the top of the tower at their Herman Miller desks, leaving the workers in the basement—at their orange crates—doing the work.

In grandfather's day, the age of xerography had not yet dawned. In that simpler time, physical separation was sufficient to keep the administrators feeding their make-work to one another. In this more advanced age, when any fool with the key to the copier can become a widely-read author, separation is not enough. Sooner or later, the workers will have to come to grips with that indispensable problem-creator, the memorandum.

We have received the "Utilization of Commas" memo and don't know what to do. Knowing that the problem comes from nothing, we ask ourselves, "What would grandpa have done?"

Taking a firm grip on our ball-point pen, we scribble across the upper right-hand corner, "A fascinating concept. Let's discuss." and route the original paper back to His Eminence. (Don't clutter your files with a copy—let the Circulation Department be your filing system.)

At least three or four feedback cycles should pass before His Eminence realizes that he can't get a meeting together merely by having his secretary phone everybody for an open time.

When he finally decrees a meeting, be sure to have a dental appointment at the same time. (Keep a cavity in reserve, unfilled.) Then, after the meeting, take the announcement memo and scribble across the upper right-hand corner:

"Sorry I couldn't make it—dentist!!

What about semicolons? Let's discuss."

With any imagination at all, you should free one month from administrative interference each time His Eminence sends you a memo. You use almost no effort, no filing space, no clerical time, and—what is most important to our feathered friends of the forest—no paper other than what is sent to you. By sending the problem right back where it came from, you give all the credit to the dean, but you may get some work done.

Are you afraid to try it? Don't be, for there's really no chance of the make-workers figuring out what you're doing. In fact, they'll love every minute of it.

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](#)

ARE YOUR LIGHTS ON? is one of the famous books Jerry has written together with Donald C. Gause.

ARE YOUR LIGHTS ON? has received great feedback from readers and we strongly recommend you to read it if you want to get 'problem solving' right, of course along with many other interesting insights that this book offers.

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

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

ARE YOUR LIGHTS ON?



Donald C. Gause
Gerald M. Weinberg

TTWT Rating: ★★★★★

TEA-TIME WITH SMARTBEAR



About this column...



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

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

The State of Mobile App Development & Testing

2014



The Rapid Growth of Mobile App Development

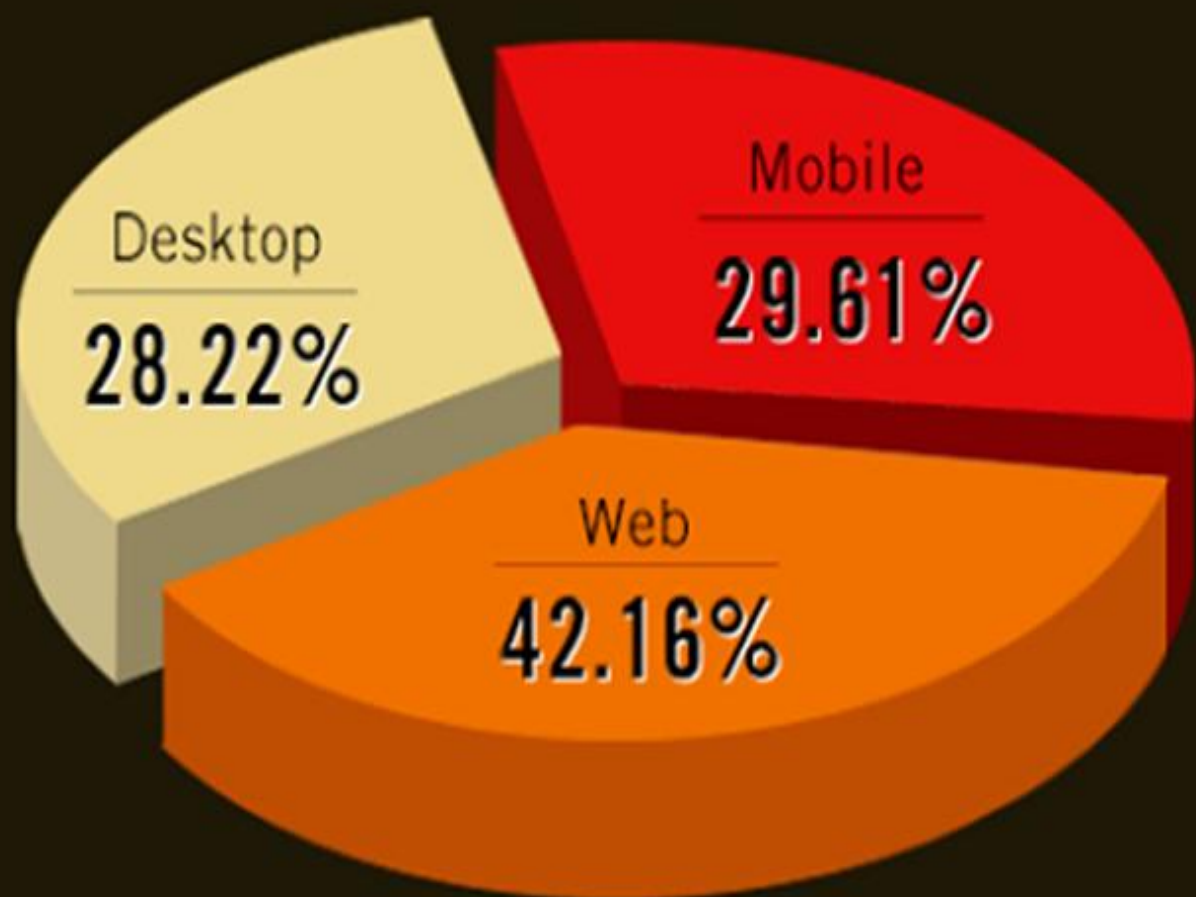
Rapid growth of mobile will continue to transform the software industry in 2014 and beyond.



Nearly

30%

of those Building Apps,
are Building Mobile Apps



What Types of Applications Do You Currently Build?

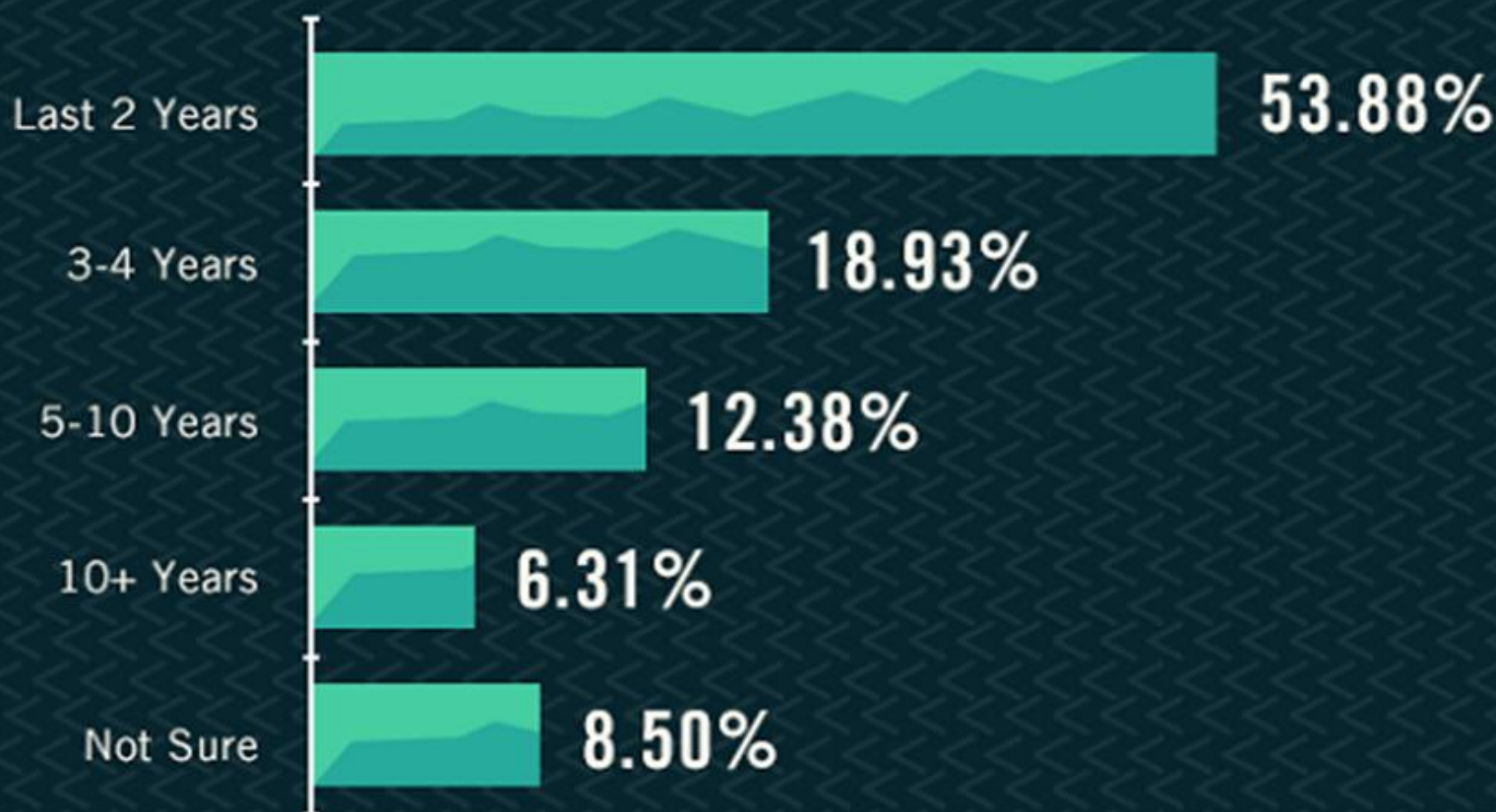
Rapid Entry Into The Mobile Space

More
than

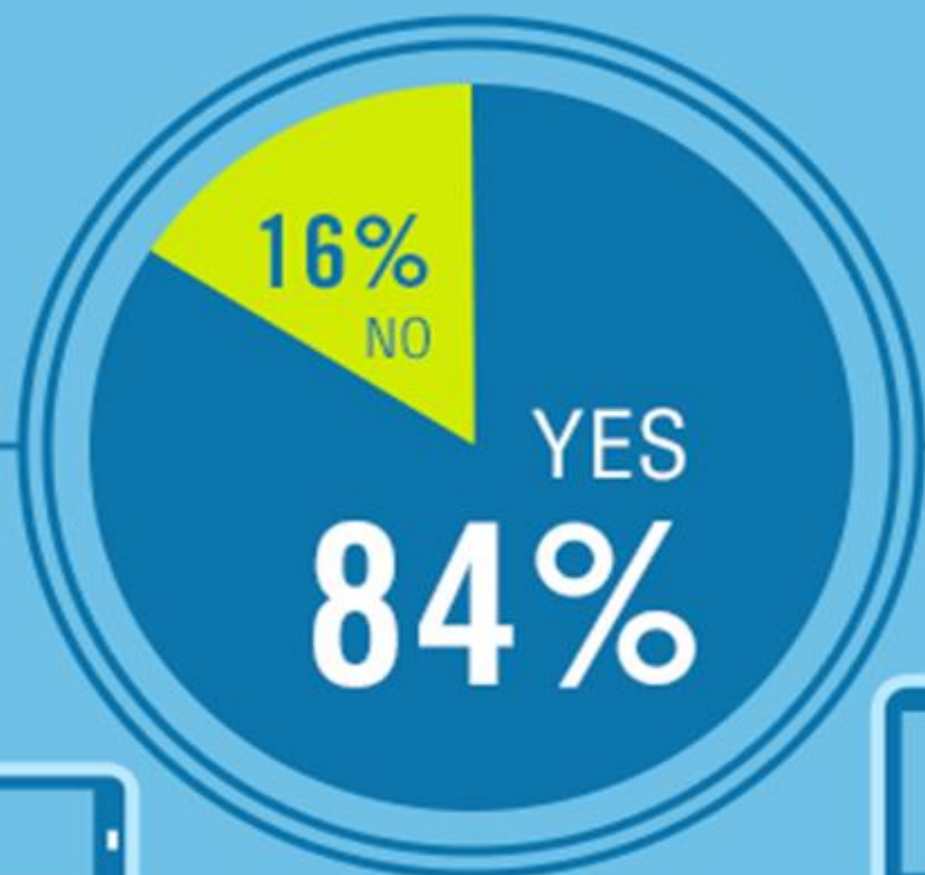
50%

of respondents who are currently building mobile apps **have entered the space within the past two years.**

How Long Has Your Organization Been Building Mobile Applications?



84% OF THOSE WHO ARE NOT CURRENTLY BUILDING MOBILE APPS **PLAN TO ENTER THE SPACE IN THE NEAR FUTURE.**

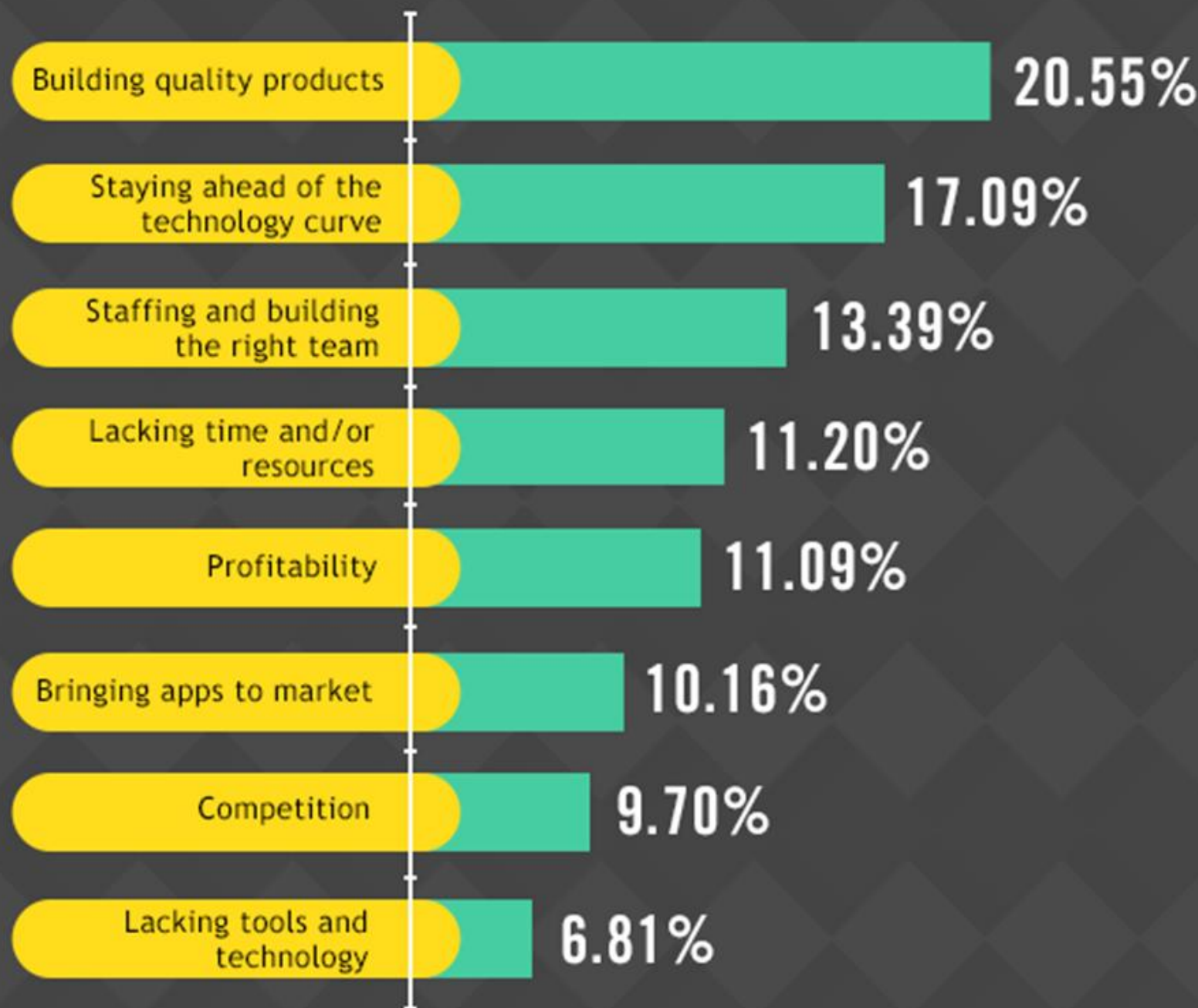


Do you plan to enter the mobile space in the near future?

FOCUS ON MOBILE QUALITY

The Greatest Challenge for Success in Mobile: Quality

What do you think are the greatest challenges for succeeding in the mobile space in 2014?



A BUG CAN COST YOU A CUSTOMER

Nearly **50%** of consumers will delete a mobile app if they encounter just a single bug.

How do you react upon encountering a bug while using a mobile app?



More than a third of consumers (35.03%) would report the bug, either publicly, to the app provider, or to friends or peers.

A circular infographic with a blue-to-teal gradient. The number '61%' is displayed in white. The circle is divided into two segments: a larger teal segment on the right and a smaller blue segment on the left.

61%

of respondents are currently using **3 or more different quality processes** when building mobile apps.

A circular infographic with a blue-to-teal gradient. The number '33%' is displayed in white. The circle is divided into two segments: a larger teal segment on the right and a smaller blue segment on the left.

33%

of respondents are using as many as **4-6 quality processes**.

THE MOST COMMON QUALITY
PROCESSES INCLUDE:

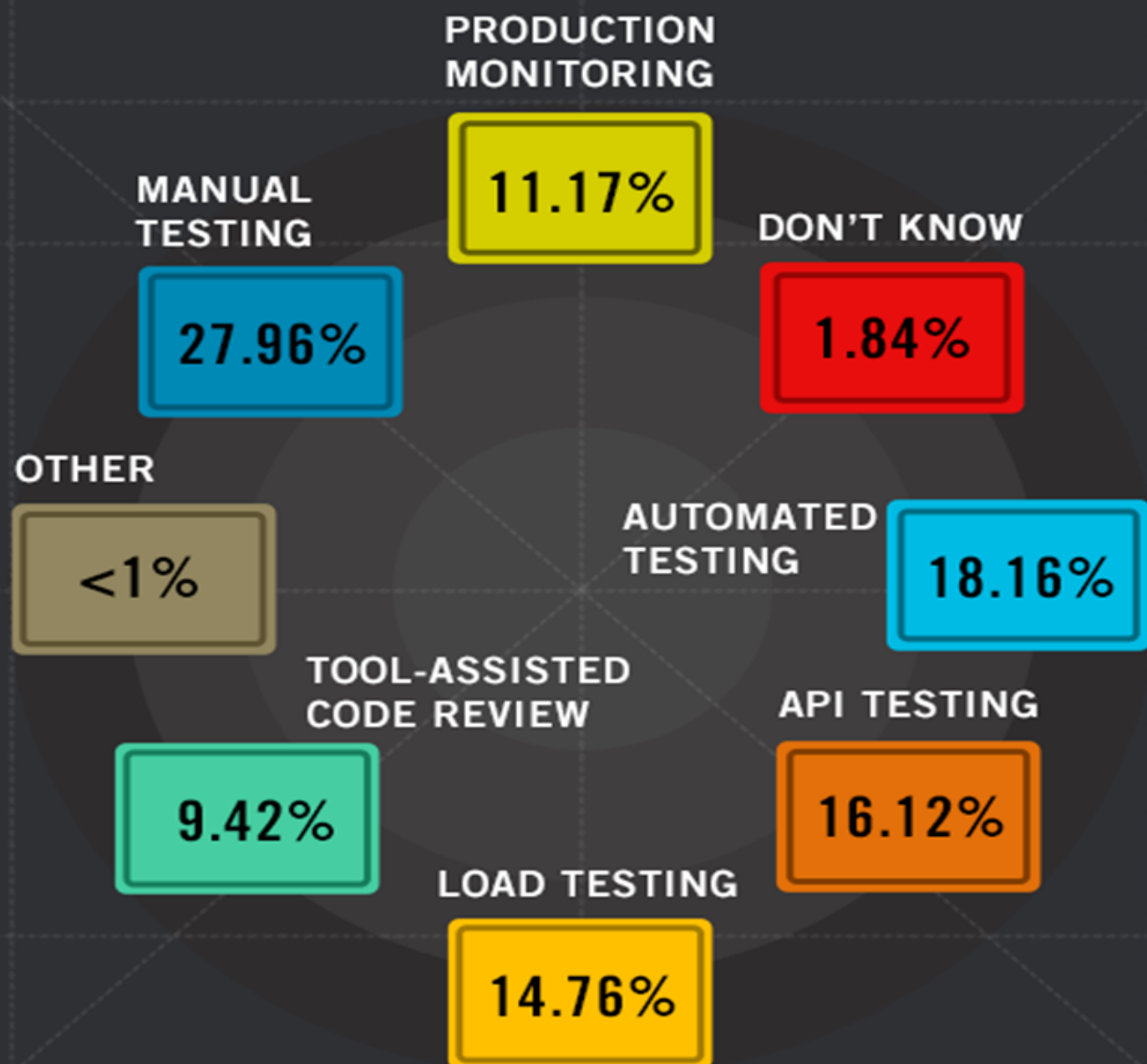
MANUAL
TESTING

AUTOMATED
TESTING

API TESTING AND
LOAD TESTING

PROCESSES USED TO ENSURE MOBILE APP QUALITY

Respondents are using a variety of processes, both in pre-deployment and post-deployment, to ensure quality across the application lifecycle.



While manual testing is the most popular process, automated testing, API testing and load testing account for nearly 50% of utilized quality processes for building mobile apps.

Need for Various Key Quality Processes and Tools to Decrease Risk

In more than 64% of organizations, developers and testers test their mobile apps.

Developers and testers are both prioritizing testing and quality throughout the application lifecycle.

How do you test your mobile apps?

Developers and testers
both test them

64.37%

No, we don't
test them

1.15%

Don't know

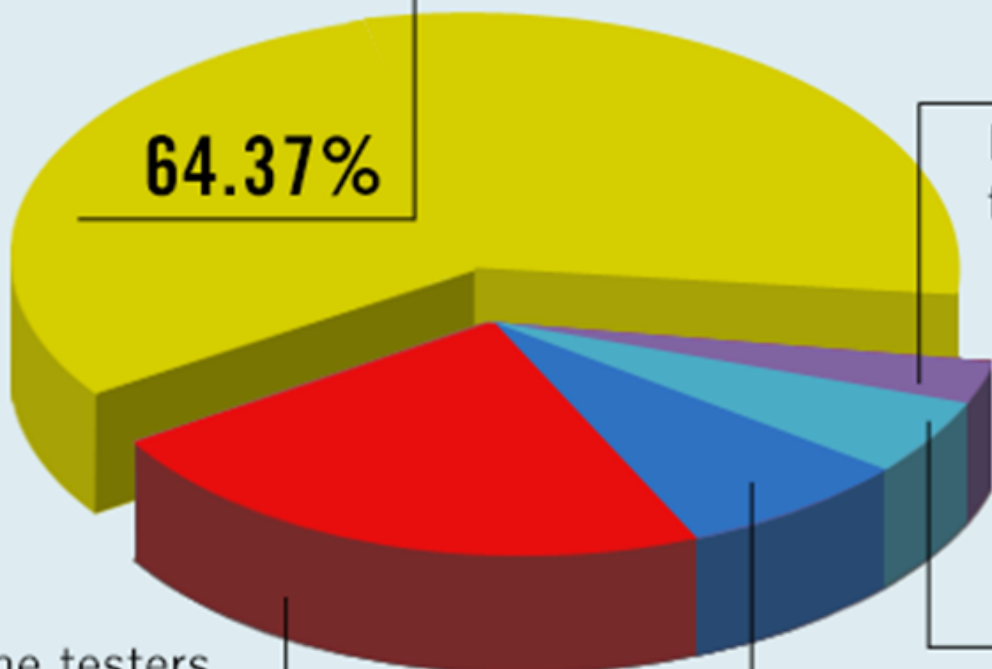
4.89%

The testers
test them

21.55%

The developers
test them

8.05%



Sources:

SmartBear Software (smartbear.com) conducted a global online survey from October through December 2013.

The findings presented are based upon the aggregated responses from over 1,040 software developers, testers and consumers.



TestComplete
by SMARTBEAR

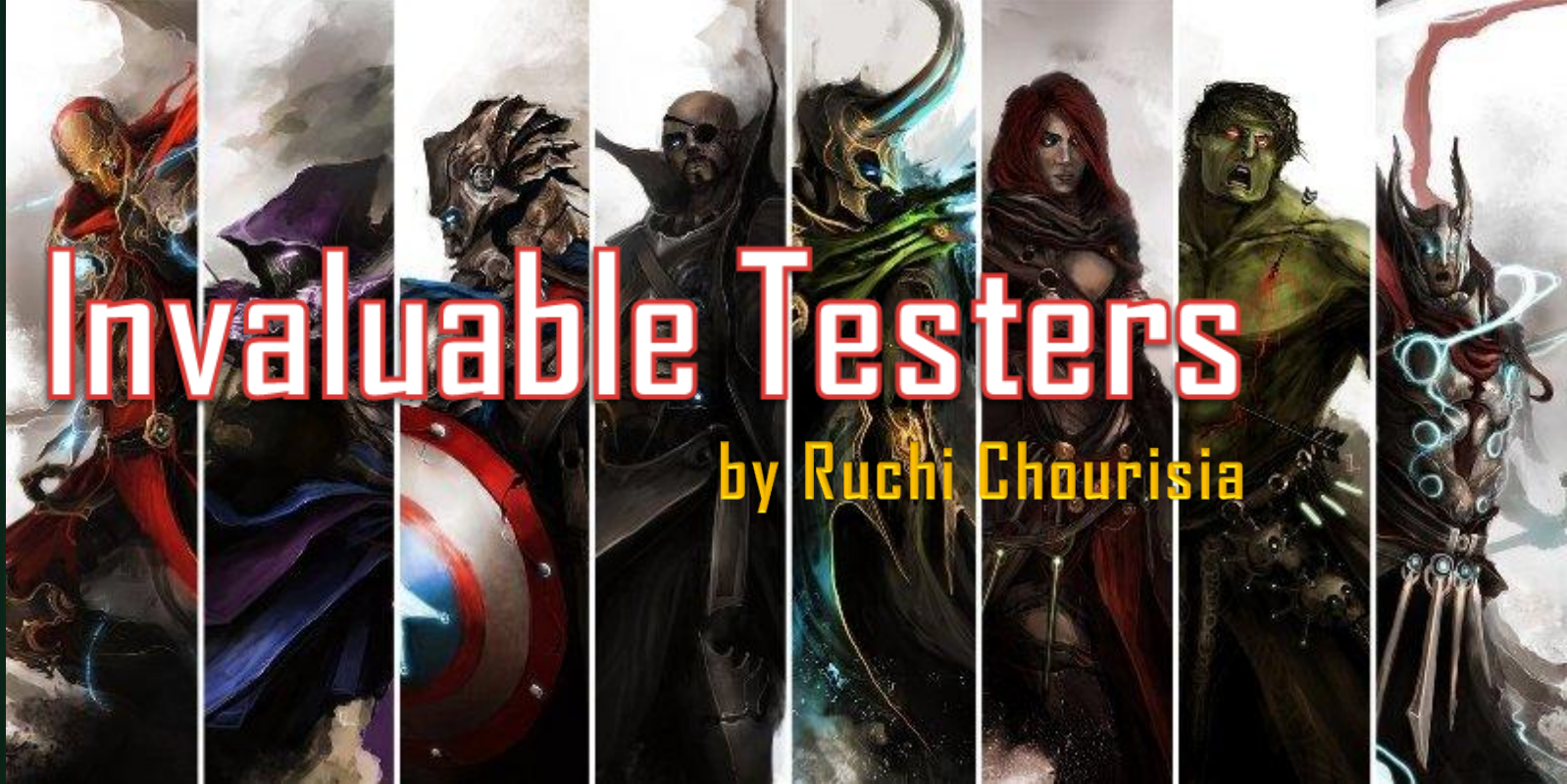


SMARTBEAR

A photograph of a green, teardrop-shaped 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 sand mandala. The entire scene is framed by a dark blue border.

Speaking Tester's Mind

- straight from the author's desk



Invaluable Testers

by Ruchi Chourisia

Quality is one word which has always lived above quantity; it has always found right people appreciating it among masses. As the world became competitive, it was only Quality which segregated the class from the mass.

Very soon the myth that quality is for manufacturing industry was broken and one could hear customers complaining about quality in service/software industry as well. It then became necessary to ensure quality in any kind of business. And very soon quality became in-separable part of any delivery happening in IT industry.

In service industry, quality shaped itself in processes and means. Processes ensured that every step was accounted for and somebody took ownership. Means developed itself into different types of tools and people-profiles who made sure that customer gets what he ordered for. When we talk about quality in IT industry, organisations gradually understood that quality comes with the usage of better components in technologies, services & manpower. So they involved skilled people in development process and used best practices, by which they made sure to deliver best quality product to their customers.

When an end product is delivered by development team, then it is imperative that somebody from outside of development team verifies the delivery. This person is responsible to map the specifications with final product and validates its functioning as requested by the customer. These people ensure that developed product is 'the right product', it fulfils customer's maximum requirements, is reliable and all passes all security checks.

To ensure all these, there was a need for capable people with better sense of observation, creativity, with ability to think out of the box and who could challenge the developed product. These people were then called "software testers".

What are the things that testers 'typically' do?

Testers play a key role in product success. They ensure that product is tested in best possible ways, and confirm its robustness.

Testers try to check each and every feature of the product within its limits and beyond its limits. They test as they are using the product for the first time like any other end user, keeping in mind that products' important features must work. Testing products' functionalities is not the only thing that testers do, they also take responsibility for its usability, installability, capability, reliability and other possible abilities. In short they test products against quality criteria (ref: [HTSM by James Bach](#)). For example: What is the use of Email application on a mobile phone when the mobile phone does not support internet? Or what is the use of a movie ticketing website if the website lacks listing of theatres or movies?

Testers add to product quality by giving valuable suggestions to enhance the usability, design, and speed or add more features if they think it needs to be added for better user experience. If we look from the top level of an IT company, it seems obvious that testing & quality assurance is a major concern, but when you start to analyse the project development life cycle you find that testing is often done in the end of each phase or after complete cycle.

Expectations from testers: High performance in lesser time

There are high expectations from testers when it comes to deliver a robust and a quality product. They are held responsible for any failure after the delivery and share the onus with the development team even though the amount of time spent on product by both development & testing team is not equal or even not justified. It is witnessed in most of the projects that testing team is involved just right before the delivery; it is development team which is involved right from the beginning and have better knowledge of the product. This practise is not good if it is seen in the light of delivering best product because in these types of scenarios, testing team works under lot of pressure and have huge tasks to achieve. Most of which are test related activities and not even actual testing.

Since testers are involved on later stage, management should not keep their expectations high as the limited time given to them would not allow testing of all functionalities with all set of input data and typical test related activities (heavy documentation, execution reports, QC related activities for example). There is a need of using risk based testing techniques to identify important defects quickly and early. Testers may create specific scenarios from the project and business flow understanding. For maximum coverage, they need a proper test bed where they can test every aspect with provided resources and the risks involved around software implementation. And most importantly by understanding what stakeholder's really want. Ironically all these tasks are performed at the time of release and in that short span of time. In such situation tester can only do exploratory testing. Most of the times testers just do smoke or sanity testing of few primary requirements. Due to such testing practices, testers are unable to find major bugs and impact on main functionalities on a later stage. That invites many client issues and bugs during acceptance testing phase. Since the testing was not done rigorously with all scenarios and most importantly regression testing, latent bugs probability increases. All these result in bad impression to client about organization's quality department.

Normally, during project estimation, about 25% time is given for testing out of complete project life cycle, but there are many factors which should be kept in mind when we directly give only 25% time. There are factors like, no. of complex modules, integrated modules, no. of resources/developers in project, their experiences for example- per stats (not obvious) people with less experience will leave more issues or functionalities and people with good experience leave no scope for testers to find silly mistakes in their developed modules, there are certain environmental factors which hinder testing (like

system crash) and resumption takes lots of time which impacts delivery time which in turn leaves very less time for testing.

I personally believe that project's success depends upon team work, including development team, testing team and management team and most importantly Project Manager/Lead.

Test automation: To be or not to be?

This is a very crucial question, as it affects project budget. It is astonishing that testers are not asked when decision of automation is made nor it is estimated according to the project's context. In these days it solely depends upon how much profit is coming from selling automation nonetheless whether project is of maintenance type where only regression is required. Or whether project is of data storage and downloads kind, where load testing becomes important. But these considerations are purposely overlooked as companies want to generate more profit and hence they compromise with the product quality.

Most of the time manual testing is chosen, as organisations with huge turnover only can buy automation tools as the licensing cost of these tools is very high. Hence it is easy to do manual testing with 1 or 2 entry level testers which generally costs very less. I feel that testers should always keep themselves updated and should keep finding newer tools from open-source, tools that would help to achieve specific testing, tools that would do load testing in cloud server, etc.

Most important phase in both manual and automation testing is preparing test cases, which can only be done manually that means a tester sit and create test cases based of his own understating, tool do not understand anything itself, it is the role of tester to create a test case based on test scenario. Tester either creates a best test case consisting of important functionality or simple test case to check GUI. So, if tester lacks project understanding, client's vision then it results in impotent test cases. Therefore, a tester with complete understanding of project is very important in any bug free release and necessary for test case scenario preparation both in manual as well as in automation testing. So, we cannot give importance just to automation tester and leave manual testers behind. Both are crucial for project's success, ultimately both are incorporating their respective skills to make the best quality product.

As I said, a project completion and success is a team work, not a one player game. It is necessary to follow complete software development and software testing life cycle properly the way it has to be implemented. Once the development and testing go hand-in-hand most of the issues would not arise and those found will be fixed within the time frame. It is essential to understand the importance of each and every member of team, ensuring that everybody is assigned a specific task and they perform their role in best possible way. Organisations should keep in mind that if quality is poor whether you release project before time or with extra features it will be of no use to the customer. We never overestimate anybody so we should not underestimate our invaluable testers also.

[Back To Index](#) 



Ruchi Chourisia, is a bio-informatician by academics but has great passion for testing. She has more than 6 years of experience in QA. She is currently working as a Test Lead & Auditor in a privately-owned software development company. She started her career as a programmer and in past she has worked with some renowned government organisations like Indian Council of Medical Research (ICMR), Institute of Genomics and Integrative Biology (IGIB), putting together her talents in the research that was conducted.

Contact Ruchi at chourisia.ruchi@gmail.com or visit her portfolio at: <http://www.linkedin.com/in/ruchich> .

Introducing...

Quality Remarks

- a column to discuss your test management problems with Keith Klain!

So, are you having testing times lately?

Want to bring change in your testing culture and need help?

Have exciting ideas around testing but not sure about implementation?

Or looking for guidance and advice on your testing problems?

Worry not. Keith Klain is here!

In this exclusive column, Tea-time with Testers is offering you an opportunity to interact and get help directly from experts in industry.

Many thanks to Keith Klain for agreeing to offer his guidance through this forum.

How will it work?

It's easy. Just send us your questions on editor@teatimewithtesters.com and we will publish Keith's answers in subsequent issues of Tea-time with Testers.

Make sure to mention **Question for Quality Remarks** in your subject line.

Note: Tea-time with Testers reserves complete right to deny any question without giving any justification.



Keith Klain is the Chief Operating Officer for Doran Jones, a technology consulting firm specializing in software testing and agile development. With 20 years of multinational experience in enterprise-wide testing programs, Keith has built and managed global test teams for financial services and IT consulting firms in the US, UK, and Asia Pacific. Keith is a current member of the board of directors for the Association for Software Testing and was the recipient of the 2013 Software Test Professionals Luminary award.

Visit his blog at www.qualityremarks.com

Follow him on twitter: @KeithKlain

A photograph of a classroom scene where several students are raising their hands. The students are seen from behind, wearing light blue, red, orange, and green shirts. They are in front of a dark chalkboard. The text 'In the school of Testing' is overlaid in white. The entire image is framed by a thick black border.

In the school of Testing

for your better learning & sharing experience



Working with locators - part 2

a series by Jim Holmes

Text Content

There are cases where you'll want to avoid fixed locators all together and instead think of querying objects on the page to retrieve information you need. I find myself doing this regularly when working with table or grid tests.

If you're building a test that checks if you can properly retrieve data from a grid row, then you don't want that test reliant on the specific order or number of the row in the grid. In this case it makes a lot of sense to avoid locators, and instead get rows from the table that match a particular criteria.

Using the table below, imagine a test whose purpose is to verify that clicking the Edit link on Jayne Cobb's row pulls up an editing grid with the proper values pre-populated. I wouldn't want this test to be dependent on the location of Jayne Cobb in the grid, nor would I want the test to fail if the Edit link changes columns.

Region	Company	LastName	FirstName	Id	
Europe	Top Notch Music Academy	Beethoven	Ludwig	9	Edit
New Earth	Blue Sun	Cobb	Jayne	12	Edit
Eastern	Relativity Inc	Einstein	Albert	8	Edit
Midwest	Telerik	Holmes	Jim	1	Edit
Scotland	Bravely Bravely, LLC	Knight	Robin	4	Edit
Scotland	Round Table Hotels	Leodegrance	Guinevere	5	Edit
Midwest	Tip Top Software	McGillicuddy	Katy	3	Edit
Western	Merwin Consulting Ltd	Merwin	Sarah	6	Edit
New Earth	Serenity, Inc.	Reynolds	Malcom	7	Edit

There are a number of approaches for this. I could use an XPath query to return a collection of the table's rows and iterate through that. I could use a different XPath query to return just the row I'm looking for. I could also get all the rows in the table and iterate through them to find the particular row I want. Text within an element can be used for element location; however, as with XPath you'll need to be extremely careful how you go about using this strategy. You'll also need to understand the difference between Text Content and Inner Text, and you'll need to understand how your automation toolset supports this.

Text Content is the text, including whitespace, of the element you're looking at. Only that element, not its children.

Inner Text is the text of the current node plus its children. There is no whitespace between the text of the elements, but text within each element is properly shown. Inner Text is extremely useful when you're doing something like inspecting a menu for particular values. A quick "inner text contains 'foo'" may be just the trick you need.

The trick about Inner Text, or Text Content, is that it's implemented differently depending on your automation driver, framework¹, or tool. For example, WebDriver doesn't actually support grabbing text or inner text for locators. Instead, you've got to get a reference to the element and check its Text property.

This makes finding specific row elements in a table a bit dramatic—you need to iterate through the rows and check each one for the text you're looking for.

```
IWebElement table = browser.FindElement(By.Id(PeopleGrid));  
IWebElement targetRow = null;  
IList<IWebElement> rows = table.FindElements(By.TagName("tr"));  
foreach (var row in rows)
```

¹ I talked about the difference between UI automation *drivers* and *frameworks* in an earlier article.

```

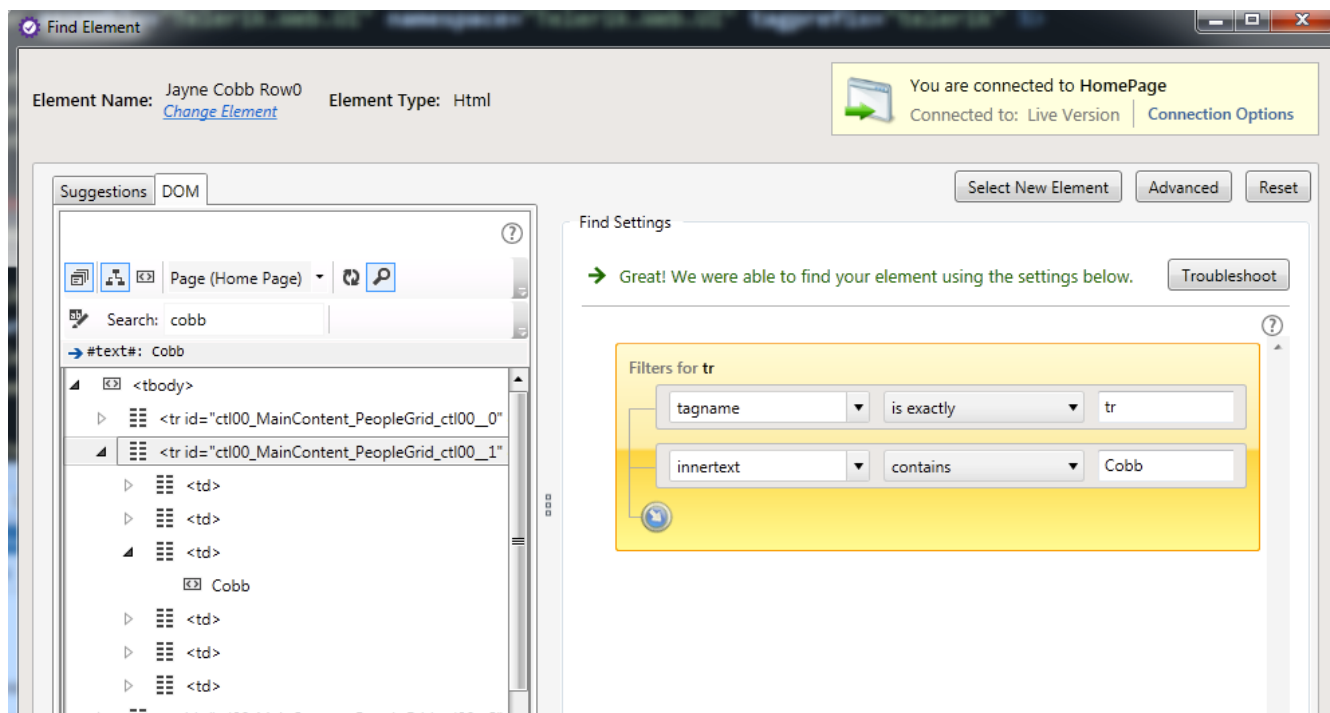
{
    if (row.Text.Contains("Cobb"))
    {
        targetRow = row;
    }
}
- }

```

Frankly, you'd be much better served in this case by using an XPath and its contains function:

```
//table[@id='PeopleGrid']//tr[contains(.,'Cobb')]
```

Some other drivers like Telerik Testing Framework, or tools like QuickTest Professional or Telerik Test Studio allow you to use TextContent or InnerText right in their locators:



Other Attributes

Other attributes may be handy when you've nothing else to fall back on. Image 'src' attributes, link targets, or 'name' attributes may all be good choices for locators.

None of these attribute types are mandated to be unique, but again, avoid thinking about perfect locators and look for a strategy that meets the need at hand.

Combining Strategies

Most automation toolsets enable you to string together strategies to form one locator. This lets you get precise, flexible locators when you're unable to craft one directly for the element you're targeting.

For example, I've often found myself missing an ID value on the element I want; however, there's one close upstream in the DOM. In this case I can use that element's ID and a short XPath to nail down the element I need. I could also use CSS/JQuery selectors in this example, but I think you get the point!

Developers Can Help With Locators!

I'm going to fall back to the regular refrain you frequently hear from me: testing works best when it's a whole-team activity. Automation works especially well when you're able to work closely with your systems' developers. Developers can do awesome things to make your automation work much easier, especially when it comes to helping out with good locators.

Take a situation where you're struggling where IDs for a control are dynamically generated by the platform you're working with. [ASP.NET](#) is particularly bad in this regard, especially since the ID varies by the control's position on the page. If another control is added to the hierarchy then the ID you're relying on will change.

In this case developers can prepend or append a static value to the control, resulting in an ID you can use to lock in on.

Developers can also work to inject key values in dynamically generated IDs, binding those IDs to data drawn for user records, for example. In this case you can now use a locator for a row where previously you would have had to avoid it.

Get Comfy With Your Locators

Locator strategy is the most critical aspect of your system to learn, and you need to learn it right quickly. Understanding how your system's platforms, toolset, and development team stack together to impact your locators will greatly ease your automation tasks. Understanding how working with developers can further ease your automation tasks will make you even more productive!

To be continued in next issue...

Jim Holmes is the Director of Engineering for Test Studio at Telerik. He has over 25 years in the IT field in positions including PC technician, WAN manager, customer relations manager, developer, and yes, tester. Jim has held jobs in the US Air Force, DOD sector, the software consulting domain, and commercial software product sectors. He's been a long-time advocate of test automation and has delivered software on a wide range of platforms.

He co-authored the book *Windows Developer Power Tools* and blogs frequently at <http://FrazzledDad.com>. Jim is also the President of the Board of Directors for the CodeMash conference held in the middle of winter at an indoor waterpark in Sandusky, Ohio.



Not Just Numerical Values

by Dr. Sanjay Gupta



We Are Not Just Numerical Values: Game changers, when discussed as Test Metrics

Data collection during Software testing phase and their intelligent representation ensures the success of the project by taking appropriate actions to track the progress on a right path. Test Metrics representation and sharing them with key stakeholders during the life cycle of the project ensures pro-actively mitigating risks and helps in bringing any missing requirements along with the assessment of the progress and quality of ongoing development and testing.

Teams can use these numbers (data) to track required information, study progress, and make the correct conclusions. One can use built-in and custom reports that are based on the data captured during test case development & execution, defects and their life cycle from test management or defect management tools automatically or collecting them manually. Intelligent interpretation after analyzing these results will help in answering questions such as:

- Are we on right path?
- Is our defect pool growing or shrinking?
- What is our burndown rate*?
- What is the test case design and execution productivity of the team

- Can we measure the Skill set of the team on numerical scale and take appropriate actions?

This article is going to highlight key important test metrics and their interpretations towards ensuring on-time , quality deliverables to our customers.

Productivity as a measurement:

Often, you might have assured to key stakeholders saying we are working towards productivity improvement initiatives. How to measure team productivity and what does it mean for a business? For example, we are engaged in a QA phase and would like to measure the outcome of the team members at global scale. In a manual testing scenario, a good test analyst can write and execute 20 and 25-30 test cases respectively depending on the complexity of the test scenario. A productivity rate for test case design or execution compare to a baseline number is a powerful measurement to showcase the capability and quality of the test team in terms of their functional understanding, capability in writing the effective test cases and speed during the QA phase.

How does a productivity number become a bottleneck in on-time delivery?

Times when we have decided on the number of Test cases to be written and executed in a given time, it is very important to meet the Takt time [***Takt time is a tool for planning and tracking***]. Takt time is the measurement on the required productivity of the team in terms of completing the job in time. For example, if we have 100 test cases to be executed in 5 days. The required productivity should be 20 days per day to complete the job in time. In this case, if one Test Analyst is working, his productivity should be minimum 20 TC/day. Below are the typical productivity metrics to be measure.

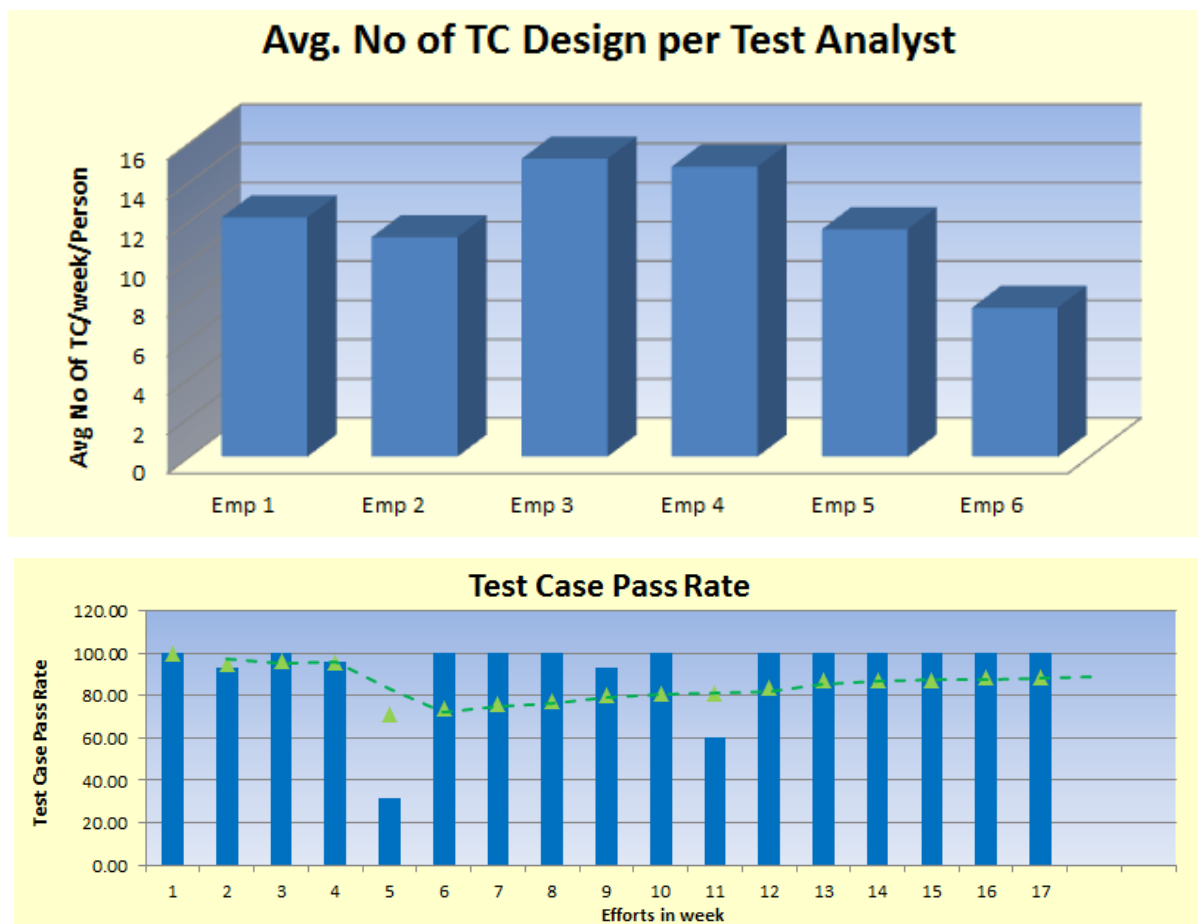


Fig. 1: (A) A typical productivity measurement of team members week wise towards Test Case Design. (B) Test Case Pass Rate can also be important measurement of team productivity during test case execution cycle. This can be correlated to the quality of the test cases and functional understanding of the team.

Defect Count as Information:

Information on number of defects, their state & severity is critical information for both the QA and development team towards delivering bug free application. The project health is also related to the number of defects detected by Test Team in a given QA phase. The criticality of the defects decides on the decision for the application stakeholders to decide on the priority of the defects to be fixed.

Defect rejection rate is a key metrics to be captured in determining the quality and capability of the test analyst. If the QA team is raising many defects which are rejected by development/BA team, it questions the functional understanding of test team.

Defect age is also one more key defect metrics to be reviewed. Defect age is defined as time interval between the open and closed status of a given defect. Defect age associated with criticality of a defect helps you in finding out the reason for the delay and an opportunity to take the appropriate actions.

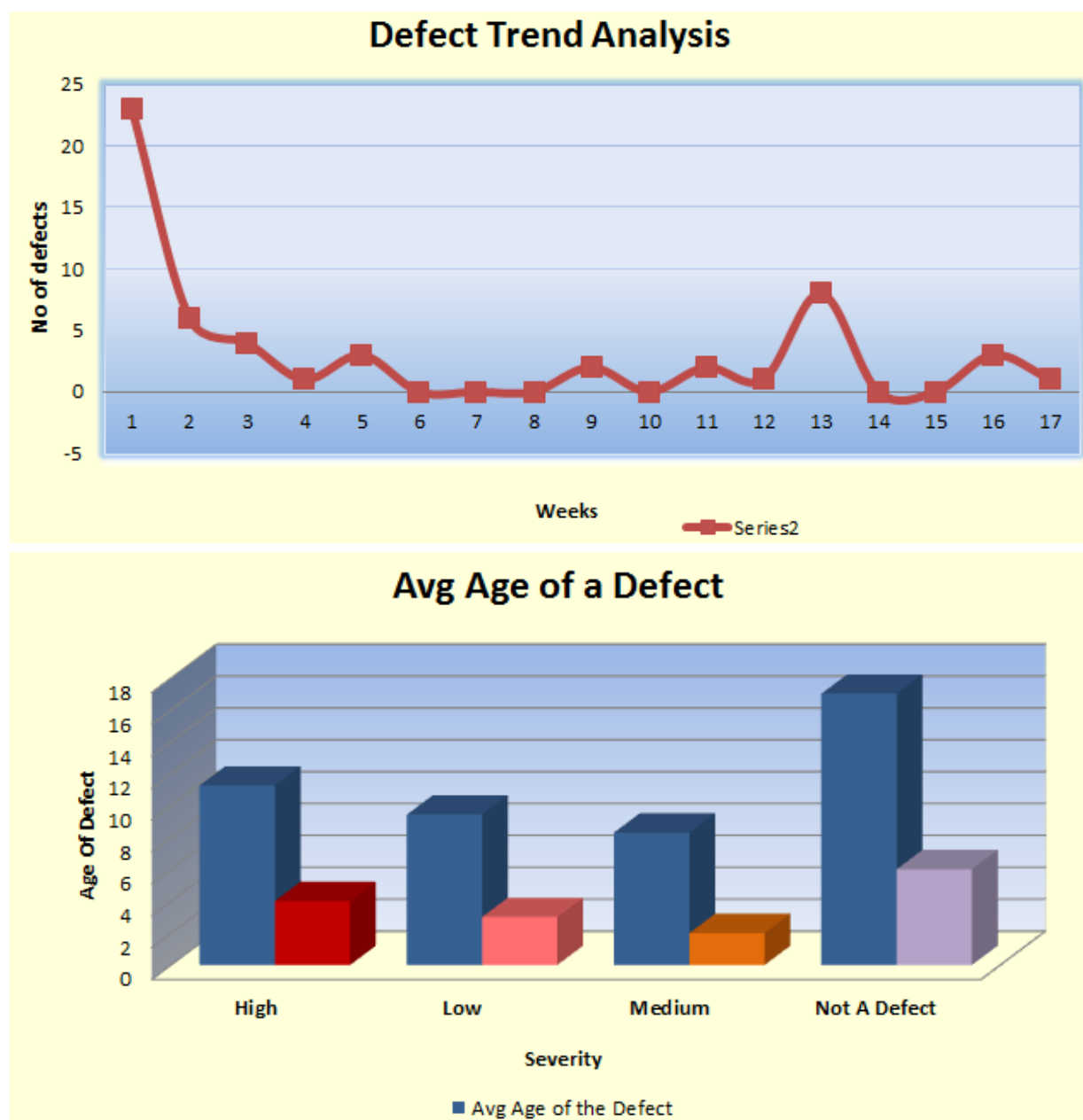


Fig 2: Fig 2.a presents the defect trend in a given test cycle & Fig 2.b represents the defect age of the defects along with their severity.

The defect trend analysis can give a measure on the stability of the application. If the numbers of defects raised are getting low with time, it means that the application is getting stable day by day. It also gives a confidence to decide on releasing the application in production.

Defect trend and defect age gives an opportunity to control the quality of the testing phase. In case when a given defect is open for long time, one gets an opportunity to immediately take an action to find out the reason, why it is open for long time and take the appropriate action.

What can go wrong with defect numbers?

Many times, it is observed that the development team or the application owners get furious about the reported number of defects. In few cases, the same defect raised by different team members increases the number and it consumes a good amount of time before concluding it as similar /repeated defects. A good analysis and functional description written against each defect is a good practice to follow.

Do we have desired Skillset? Can it be measured?

In today's competitive environment, cost, employee attrition rate, reduced time to market and skillful team members are major challenges. Ensuring a right skill set among the team members is a challenging job and an art of the project manager to ensure the best skill set by taking the appropriate actions.

How can we measure it?

A good measurement will be a holistic (3 D) view of skillset:

1. A scale to evaluate individual's skillset w.r.t required skills
2. Overall Team Competency for required skill sets
3. A measurement of Team readiness against a given skill set

Below is a tool, which is capable to evaluate skills on all the above three dimensions. In the given required skill sets as represented in below picture, resource one has overall competency of 0.75, which is a good number as in global standards a skill of 0.7 or above against required set of skills are acceptable.

If you closely look the total overall team competency (in yellow) number, it is 0.57, which says that overall team competency for required skillset is not good on the same scale. The third dimension is the readiness of the team against a given technology or skill. If you see, overall team is good for Java as it has java competency as 0.89, which is good but on the other hand if you look the scores against XML and Data Base knowledge they need an improvement as they are 0.39 & 0.36 respectively. These scores are far below the acceptable global norms.

Functional/System Area	Java	Insurance Domain knowledge	XML	Configuration management	Requirement Gathering	Data Base	.Net	SQL	Total Ranking 0	Total Ranking 1	Total Ranking 2	Total Ranking 3	Total Ranking 4	Individual Total Score	Normalized score	Knowledge Level
Resource1	4	3	2	3		3	4	2	0	0	2	3	2	21	0.75	A
Resource2	4	3	4	2	3	3	2	4	0	0	2	3	3	25	0.78	A
Resource3	4		2	1				4	0	1	1	0	2	11	0.69	B
Resource4	3	3	0	0		0		0	4	0	0	2	0	6	0.25	C
Resource5	4	0	0	2		3		2	2	0	2	1	1	11	0.46	C
Resource6	4	3	0	0				3	2	0	0	2	1	10	0.50	B
Resource7	4	4	3	1		1		3	0	2	0	2	2	16	0.67	B
Resource8	3	3	0	3		0		2	2	0	1	3	0	11	0.46	C
Resource9	2	1		2				2	0	1	3	0	0	7	0.44	C
									10	4	11	16	11			
Total Ranking 0-No Knowledge	0	1	4	2	0	2	0	1	10							
Total Ranking 1 -Conceptual Knowledge	0	1	0	2	0	1	0	0		4						
Total Ranking 2 -Experienced	1	0	1	4	0	1	1	3			11					
Total Ranking 3 - Expert	2	5	1	2	1	3	0	2				16				
Total Ranking 4 - Guru	6	1	1	0	0	0	1	2					11			
Team Total Score	32	20	11	14	3	10	6	22						118	Team Competency	Overall Grade
Normalized score	0.89	0.63	0.39	0.35	0.75	0.36	0.75	0.69							0.57	B

How to improve on competency?

Once you identify the skill area to be improved on, create an action plan like technical, hands-on experience or domain trainings and evaluate the team after the few months of trainings. The improvements in the skills can be seen in the output from the team. For example as decreased in number of defects slipped in production, number of quality defects raised by test team, reduction in number of rejected defects etc.

What Next?

Test Metrics can be represented or tailored in many ways. The same data can be represented in different visual ways. Decide and stick to pre-defined test metrics for a given engagement. For example, you can make a checklist of the test metrics used for water fall model, Agile development or for enhancements type engagements. It will generate a common understanding in all the Test Analysts in an organization and helps in smooth quality delivery. A meaningful pictorial representation of project progress as Test metrics is a powerful tool in assessing the progress and an opportunity to take appropriate actions in advance to mitigate any major risk.

Reference:

[1] Burndown and Burn Rate Reports <http://msdn.microsoft.com/en-us/library/dd380678.aspx>

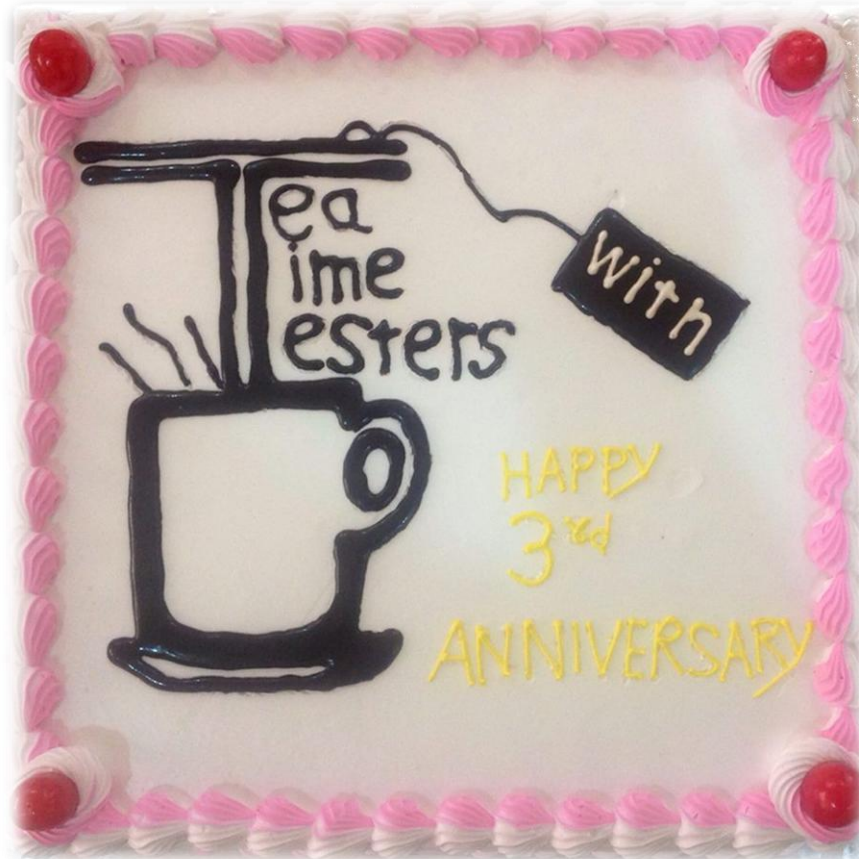
Acknowledgement:

The author would like to thank and gratefully acknowledge the support and encouragement provided by Mr. Dhiraj Sinha, Ramamurthy Ramki and Ashutosh Vaidya .



Dr. Sanjay Gupta in the Application Testing practice is leading the Innovations & Strategic Test Consulting at Dell Services, Bangalore. Dr. Sanjay has over 14 years of IT industry experience in Software Testing, Solution Designing and Application Development. Dr. Sanjay has in-depth knowledge of Life and P&C Insurance domain and rich hands-on experience in Retail, BFSI , Transportation, Media and publishing domain. He has in depth knowledge of tools like Purify, Code Coverage, Quantify, Rational Robot, RSA and Rational Rose and Orthogonal Array base Test Optimizer. He has published over 30 research papers in international conferences and journals/website of international repute in the field of Software Testing, Requirement Gathering and J2EE.

Dr Sanjay received his Doctorate from Indian Institute of Technology, Bombay. He is a Sun certified programmer and Java Trainer. He is Recipients of Marquis Who'sWho in Science and Technology & Who's Who in the world.

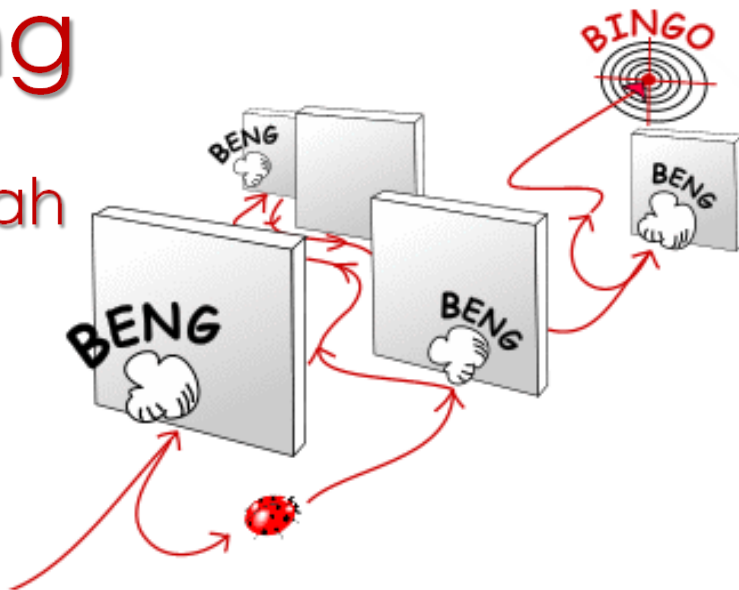


Thank you dear readers for your love and support!

This cake is for you!!!

DISSCOH – Heuristic Approach to Bug Reporting

- by Jyothi Rangaiah



This article focuses on the approach to bug reporting with a closer look at the rejected and deferred bugs.

What is a heuristic?

The word heuristic is derived from the Greek word 'heuriskein' meaning 'to discover'.

A commonsense rule/set of rules intended to **increase the probability** of solving some problem (definition via the WordWeb 5.52).

I was introduced to the heuristic approach of solving software (testing) problems when I first met Pradeep Soundararajan.

Based on my personal experience with regard to bug reporting, I have collated the points and the examples below to ponder upon while reporting bugs.

DISSCOH

D - DISCOMFORT

I - INQUISITORY

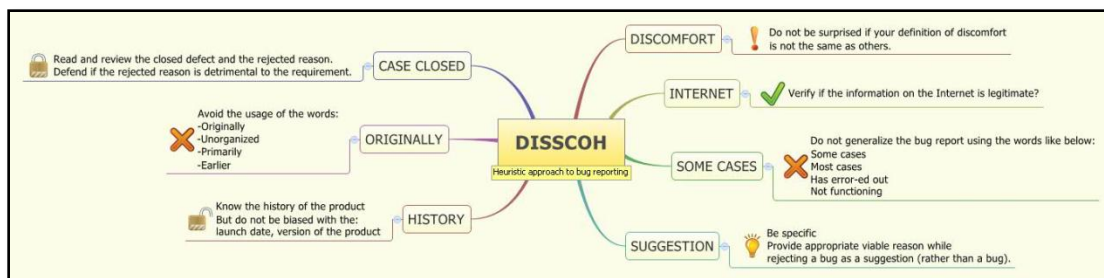
S - SOME CASES

S - SUGGESTION

C - CASE CLOSED

O - ORIGINALLY

H - HISTORY



[Click [here](#) to enlarge]

D – DISCOMFORT

As an example, consider the word 'Error' being misspelt as 'ERORR'. This misspelt word is causing discomfort and requires to be logged as a bug.

It is possible that this may not be seen as discomfort by some people and the bug if logged could get fixed or rejected or deferred.

To understand what a misspelt word could cause, take a look at this picture in the link below.

<http://www.theguardian.com/books/gallery/2012/oct/25/worst-typos-pictures#/?picture=398262579&index=0>

And that's the price Chilean's paid for the misspelt word.

"Question with an intention to gather information"

When we know better we test better.

The information gathered could help us in understanding the impact the bug might cause if left unattended.

Refer to the wishing wand chapter from the book – 'More Secrets of Consulting' authored by Gerald M. Weinberg on questioning with an intention to gather information.

Question who the users, target audience and the customers are. Know that your definition of discomfort can be different from the others.

Question the stakeholders, business owner, product owner, your own self, users and the involved in order to gather information to answer the above questions.

If it seems clear to you that this bug needs to be fixed then log it and move ahead with logging the other bugs, which you think could also add value to the software being developed.

Have you noticed this perspective on how fast/slow a misspelt word on a very popular website/stand alone application versus a not so popular website/stand alone application is handled?

Have you observed if you added value by logging other bugs along with the misspelling, this caused both the bugs being handled versus only the misspelling issue being ignored?

What factors are affecting the decision makers? Have you questioned the decision makers?

If not, it is worth finding answers from the decision makers.

Consider reading the below excerpt from an article by Michael Larsen on his blog www.mklttesthead.com

<http://www.mklttesthead.com/2013/09/leave-your-ego-at-home-99-ways-workshop.html>

"Spelling errors are often items that really get an executive irritated, especially in things like End User License Agreements. I learned this a number of years ago when, because of misspellings and typos in the legal agreement, the net result was the fact that we had created a loophole where we were making a

tacit agreement that was never intended (and that we would be liable if something were to happen if the users did not do the steps necessary).

These areas are not glamorous, and sometimes they can be quite tedious. While that may be true, these are also areas that are considered most likely to impact revenue, and therefore will be given close scrutiny by executives. If we are taking the time necessary to look in these areas and report on what we see, we will probably get more traction and movement on these issues. Why? Because these are the issues that really matter. It may not seem fair or intuitive, but you can almost never go wrong reporting and championing bugs that the CEO has mentioned will be problematic for them if they were to be released."

I – INQUISITORY

Users must verify if the information found on the Internet is authentic. If/when producing the information found on the Internet as a proof for a defect.

Consider the below link here as an example to probing.

http://www.gs1.org/barcodes/support/prefix_list

The information related to the prefix list 985-989 is missing in the link provided and was missing at the time the issue (missing information) was reported to be fixed 6 months ago.

Post the fix, the readers now have access to this information: *Prefixes not listed above are reserved by GS1 Global Office for allocations in non-member countries and for future use.*

This information is relevant and is essential to any retail organization.

Optional further reading:

<http://askleo.com/how-the-internet-is-breaking-journalism-and-what-it-means-to-you>

"Question with an intent to gather the RIGHT information"

Who are the contributors to Wikipedia? Views expressed on Wikipedia or on the Internet could be an individuals perspective and/or an organization adding their own view.

Diligently work on providing evidences and be thorough with investigation. Build your first and subsequent investigation reports with references leading to why/why not the information on the Internet is correct or incorrect.

Also know, when to stop probing.

S - SOME CASES

Consider a bug description like the one below:

On the 'Nokia Lumia 520' phone in some cases the user is unable to slide down to the whole list of the paired devices which are connected via the Blue tooth on that device.

Wait! In most cases the users and/or the programmers (as there is a work around it) might ignore this bug so why fix at all?

"Avoid the usage of words like some cases, most cases from the testing vocabulary"

What are some cases? What are most cases? Who is the judge of such cases? Is there a way to concretely confirm what some and most cases are? If the answer is No – then avoid the usage of some cases and most cases from verbal and non-verbal communication and while reporting bugs.

Instead it helps if some/most cases are expanded to include the details of what do they really mean.

S – SUGGESTION

Consider this bug rejection reason – "Suggestion rather than a bug".

The bug validator conveys that the raised bug is not a bug but a suggestion.

Testers, provide reason for logging the bug with appropriate reasoning.

For example:

A tester could have logged the bug cause he/she did not have all the required/sufficient information to judge the failed test as a suggestion rather than a bug.

It could be a missing requirement in the requirements document or the scope of testing is not specified or is limited. It could be any of these or other reasons. But it is essential that the bug validator too provides reason for marking the bug as a suggestion.

"Ask for and receive very specific rejection reasons"

C - CASE CLOSED

Failures can teach a lifetimes lessons.

"Care to defend the rejected bug if required"

Read the rejected reason when a bug is rejected. If there is no reason for rejection, ask for it over an email. Re-open a closed bug if found detrimental to the current requirement or you foresee it as a future potential candidate (a deferred bug) And remember this when trying to recreate that assumed irreproducible bug.

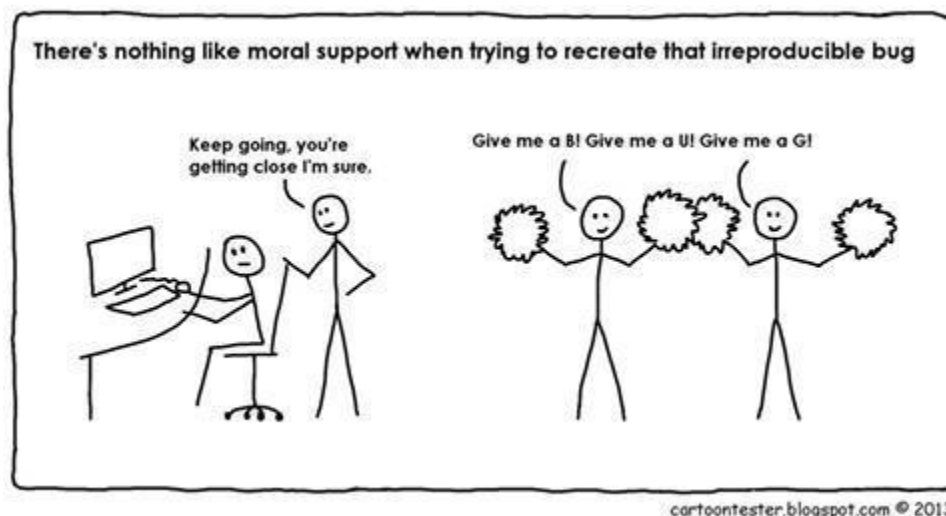


Image courtesy: <http://cartoontester.blogspot.in/> The above cartoon is designed by Andy Glover.

O – ORIGINALLY

Have you heard this before?

'Originally, on my machine the requirement is satisfied and hence this is not a bug'.

"Know your test environments"

- a) Did the programmer/tester mean this only to dodge further questions?
- b) 'Originally' is in consideration to what?
- c) Avoid usage of words like: *Unorganized, primarily, to begin with, earlier and works on my machine* when logging and rejecting a bug.

In agile, requirements don't always stay true to their original description.

Ask what Originally means?

H – HISTORY

Learn the history of the product. While testing, try not to be biased by the launch date, version and previous state of the product.

"Bias and un-bias if required"

"Test the assumptions regularly"

Find the answer to:

- a) Should I be performing sympathetic testing as this product was launched 80 hours/days ago?

Irrespective of any such preconceptions, log the bug.

And:

Remember your right to information. Question whenever required.

If there is no scope to question or there is a block recognize it. Try to unblock.

You think it isn't easy: Get help.

The below point does require to be addressed.

There could be a case when a bug remains in New state or has not been looked into. Follow up on such bugs. Be persistent and log bugs *dispassionately.

Learn about *RIMGEA: A Bug reporting Heuristic

<http://prairietester.blogspot.in/2013/09/rimgea-bug-reporting-heuristic.html>

<http://searchsoftwarequality.techtarget.com/tip/Software-testing-is-improved-by-good-bug-reporting>

How to use DISSCOH when logging a bug or approaching a rejected bug?

I would like to suggest that you try it and check its usage when reporting a bug.

Try to add, edit, delete and share your views with the world.

Acknowledgments:

My gratitude goes to Carsten Feilberg and James Marcus Bach who mentored me on bug logging and bug titling respectively.

Andy Glover (cartoontester.blogspot.com) and Michael Larsen (www.mkltesthead.com) for granting permission to use the cartoon and the excerpt from their blogs respectively.

Other mentions and further optional readings about Heuristics include:

Parimala Hariprasad

<http://curioustester.blogspot.in/2009/10/power-of-heuristics-and-mneumonics.html>

Lynn McKee

<http://www.qualityperspectives.ca/blog/>

Ben Simo

<http://www.questioningsoftware.com/2007/08/failure-usability.html>

Jason Coutu

<http://prairietester.blogspot.in/2013/09/rimgea-bug-reporting-heuristic.html>

Happy and efficient bug logging.



Jyothi is a blissful learner, a software tester and an analytical thinker who is trying to also be a critical thinker. She was introduced to the elite group of testers in 2013 and since then has been in total admiration of the community of testers.

She is learning and unlearning software testing with each test executed, every article/book read and reviewed.

Loves to network and learn from the world of skilled testers and contribute to the testing community.

She blogs at chroniclesoftesting.blogspot.in and can be reached on twitter @aarjay.


Happiness is....

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



Like our FACEBOOK page for more of such happiness

<https://www.facebook.com/TtimewidTesters>



Call for Articles !

Have you got something to say?

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

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

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

Submit your articles, stories, thoughts around software testing.

now its your chance to be heard...!

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

T ' Talks



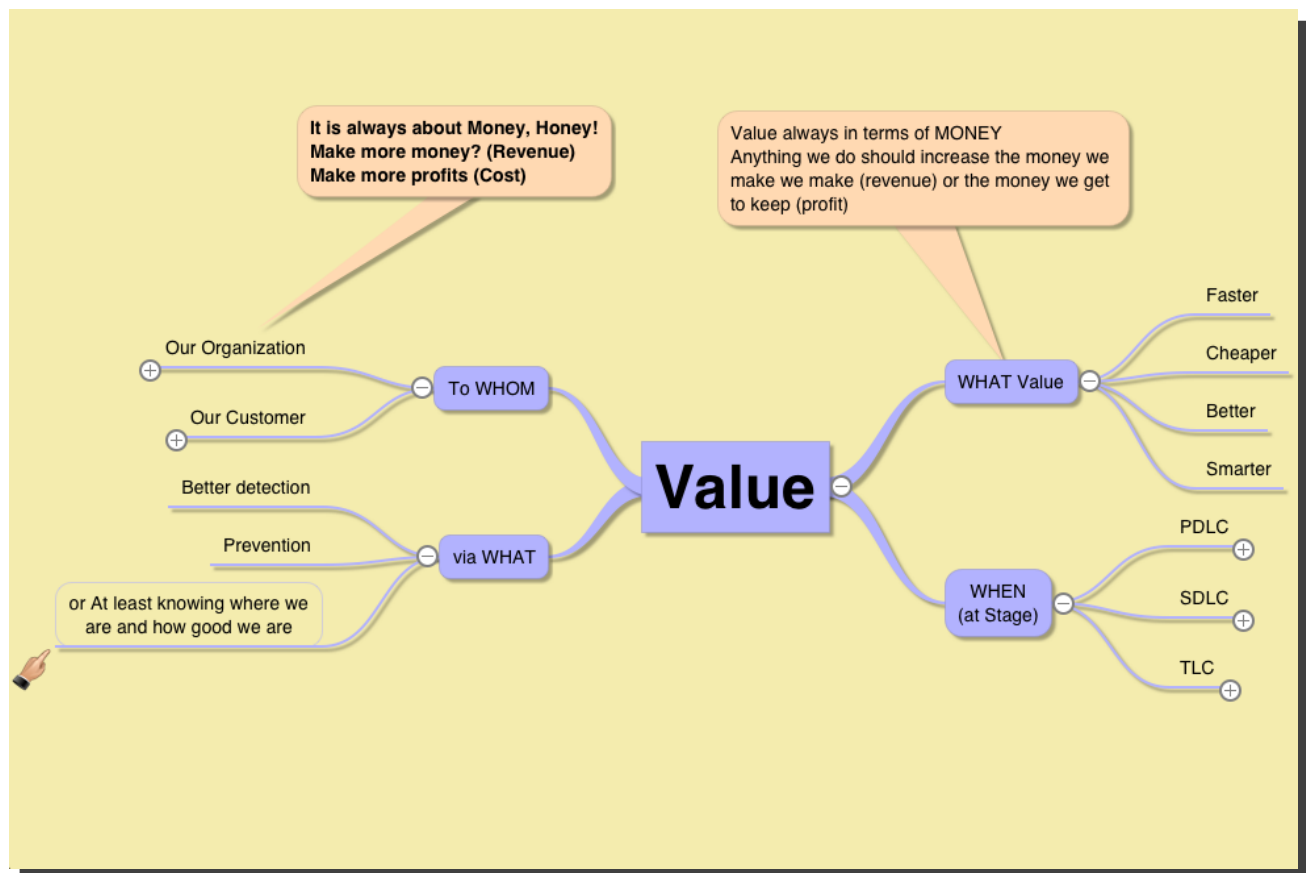
T. Ashok exclusively on software testing

It is value, not activities that matter.

We busy ourselves with activities and feel good doing it. And we do it well. We do it diligently. We expect appreciation, rewards for doing the activities. Is that good enough? Nah! It is really the outcomes of activities that matter. It is about delivering value that really counts.

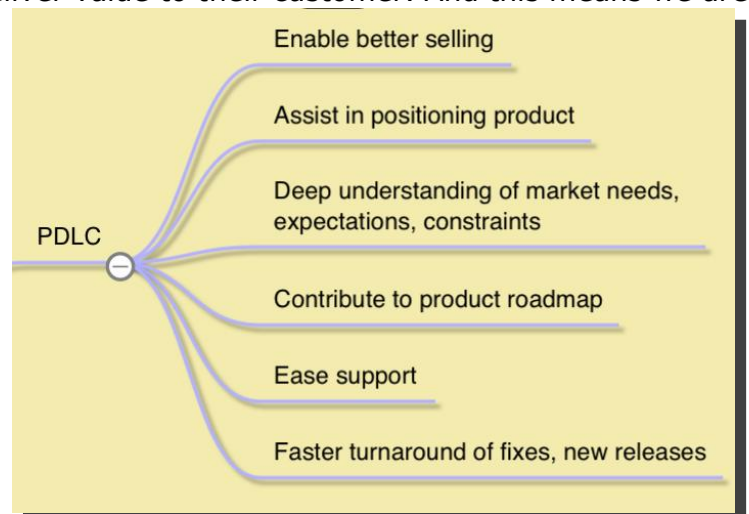
So what is value in our context? What are the various value propositions that we should be offering? And when do these matter? How do we measure value? For whom is the value proposition?

Let us start with what is value. We are in the business of software. And business is about making money. About increasing our revenue. About increasing profits. Ultimately value is associated with money. How does what we are doing enable higher yield? Higher profitability? If what you perceive as value cannot be tied to money, then re-think. Value is really a ratio of the outcome to the cost expended. Better outcomes and the same cost improves the revenue capability and adds to the top line while the same outcome at a lower cost improves profitability adds to the bottom-line. Of course a combination of both would be brilliant i.e. higher revenue and much higher profits.



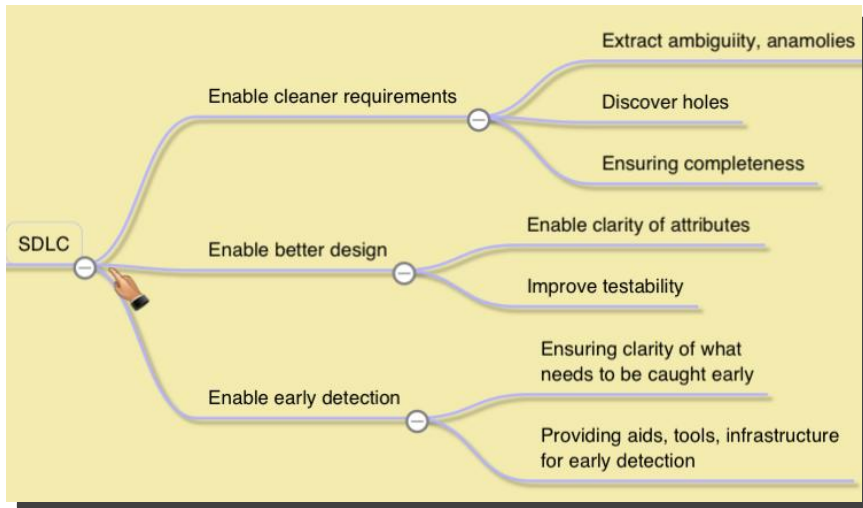
Value is about outcomes being better, faster, done cheaper and smarter. If we deliver better quality, enhancing customer experience and therefore stickiness and win more customers, then our outcome has a direct correlation to the revenue. Sweet! If we are able to test quicker and contribute to get the product out of the door quickly, then we have the ability to sell earlier and once again possibly contribute to increased revenue. Nice! If we optimise, improve productivity and therefore lower the cost of doing, then we increase the profitability. Good. And finally if we get smarter in performing the activities, then we do less work and still accomplish more with less. Better and cheaper. Really sweet!

Let us see at what stages we can deliver value. At the highest level the Product/Application Development Cycle, a deep understanding of the markets, user needs and expectations would ensure that we via the features enable our customer to deliver value to their customer. And this means we are contributing to better positioning and selling. Increased revenue. If release quality is excellent our direct contribution is to the lowered support. Also as test staff with a deep knowledge of the product and usage, we could contribute to support in enabling them get up to speed on the product and also contribute to faster issue resolution. Lower cost. And finally a great set of test assets and skills can result in faster turnaround of fixes/new-releases. Higher



customer satisfaction and therefore more money.

Stepping lower into the SDLC, our value could be how we can enable requirements to be clean - question, question to extract ambiguity, uncover anomalies, holes and ensuring completeness. By enabling sharper clarity on attributes (non-functional requirements) we can enable better design and finally being sensitive to testability aspects, we can engineer in better testability at the design stage.



Now for the next state of SDLC, by ensuring clarity of what issues need be caught at the dev stage and providing aids, tools, infrastructure we add value to the dev stage and to the developers.

Now at the stage of Test Life Cycle (TLC), ensuring great coverage and thereby lowered escapes, coming up with an efficient strategy to ensuring rapid testing, increased productivity contributing to both the top line and bottom line respectively. One of the critical value contributions of the test

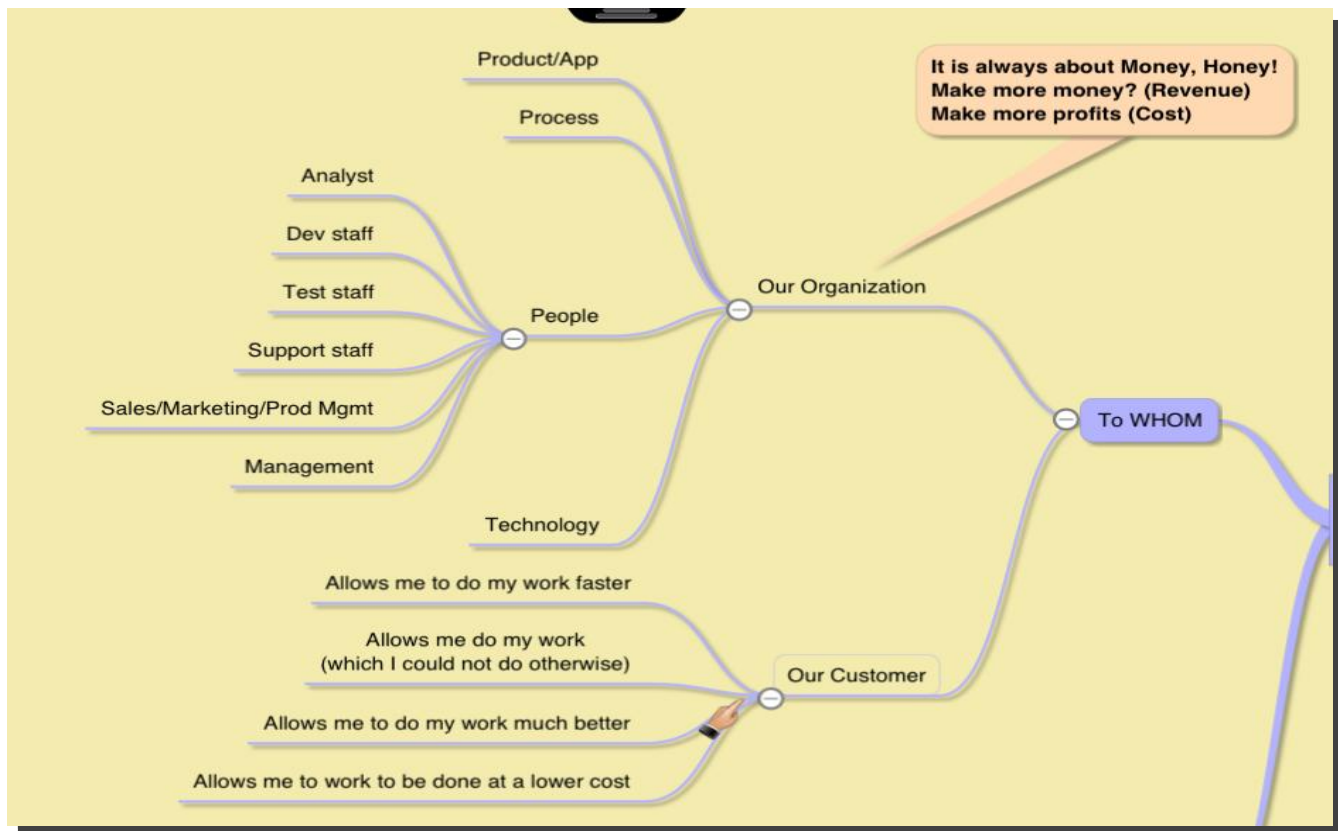
staff is to provide a clear insight of the system in terms of current state of completion and quality and how much more is pending and the risk to meeting the deadline. The latter is especially important to the management to aid in critical decision making. So when you collect metrics and create reports, ask yourself as how is aiding the recipient of the report (normally the management).

That brings the question, via What do we deliver value? We can deliver via preventing defect or by better detection. At the least we should be assessed to and present with crisp clarity, the state of the system and the risk of delivery to enable objective decision making by the stakeholders. Like a doctor who is able to read the symptoms, diagnose and present treatment alternatives!



Value is always related to the recipient. Now who is/are our recipient(s)? (1) Our organisation - How do what we do, deliver outcomes that affect positively the developers, analyst, support staff, product & marketing staff, and finally management. And secondly how are we impacting the building of product in terms of process and technology to support better quality, faster and chapter delivery? (2) Our customer - The ultimate notion of value is when the customer is delivering higher value to his/her

customer due to our product. If the customer is able to do his/her work faster, do things which he/she could not do before, then we are delivering the highest value.



So go beyond just finding bugs. Ask yourself how you are contributing a great customer experience to enable the customer deliver a higher value to his/her customers. Ask yourself how you are impacting the various staff (internal customers) in your organisation to deliver value to their (internal) customers. And you could do this detection/prevention/clear-reporting at any stage via better test assets, process, aids and technology. So if you revel is staying busy, if you are doing very many things, pause. Ask yourself as to ether you know the purpose of what you are doing, the value expected from you. Quite a few times, it is a cold blast on your face. It hits you and you warm up to the idea of value! Ultimately doing activities intensely make your tired and older while delivering value makes other happier and therefore you stay happy and cheerful! Enjoy it.

I bet you want more money for what you. Want a raise, a big one? So in your performance review, go to your boss and tell him/her the value you have delivered and demand more money! Don't tell him/her how diligent/hard you worked, he is bound to tell you - So what, that is your job!

Let us toast to delivering value.
Have fun!



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



testing intelligence

- *its all about becoming an intelligent tester*



an exclusive series by **Joel Montvelisky**

Can you test without knowing the names of your users?

A couple of days ago I saw a short YouTube video that got me thinking about an issue I stumble upon almost daily as a tester.



The video, not even closely related to testing, was posted out of a CNBC program that was itself talking about a documentary made by Al Jazeera on the topic of Abortion Legislature. In this video the host of the program, Rachel Maddow, shows a cut from the documentary where a legislator is asked a simple but extremely good question by the interviewing reporter.

The topic of the documentary is “very loaded” and so I don’t want to go into it. But the comment from the CNBC host really drove a point I’ve been trying to get through to other testers, developers and product owners for a long time now.

Why don’t you start by watching the [short video](#) (1:48 min) to understand what I am talking about.

Done, good!

Now, let’s focus solely on the sarcastic commentary made by Rachel Maddow:

***“Why would a woman want an abortion? I’ve never thought about it!”
Says the man who’s doing his best to ban abortions in Ohio...***

Regardless of what side of the abortion issue you are in, and there are very valid cases for both sides of this topic, you have to agree with the host. How can a legislator decide on issues affecting other people without deeply understanding what are [the reasons behind their choices or actions](#)?

I am not talking about the reasons that push the legislator, I am sure he also has good ones, but in this case the question is very simple and it is focused on the individuals affected most directly by the decision here, the women wanting to do an abortion.

How can he choose for them without even understanding why do they want to perform the abortion in the first place???

Is this related to testing?

YES, and now let me explain to you why.

When I saw this video I was “virtually transported” to the many times when I found myself “defending” (there is no other word for this) the interests of end-users in front of developers or product managers who had not really thought about the reasons behind our users’ actions and decisions.

I don’t intend to compare the two issues, but in both of them we have people making decisions that directly affect others without really understanding the reasons these people have for behaving the way they do.

I am sure you’ve seen this yourself!

When programmers choose a “development solution” based mostly on technical constraints (what is technically “safer”, or the less complicated way to do something), and this forces the end-users to work harder to do achieve simple actions in the product.

Or when product managers decide on workflows or configurations with only partial information about the real needs of the users and without understanding the ways they work with the product during their day-to-day operations.

I remember a case about a decade ago where we developed a product for Customer Support centers. The product was pretty impressive and it had some amazing features in it, but it also had many modal messages that got in the way of our end-users, who could not work in the way they needed with multiple customer tickets opened in parallel at the same time.

When will people realize that quality is not only making sure there are no bugs or crashes in the product?

It is not enough to be “customer focused”

But back to our video. I want to say once more that these are cardinaly different situations, but still we are talking about the same symptoms: Someone makes a decision without understanding the reasons, the real needs, of his constituents or his end users.

Why would she / he / the user want to do this???

- Did we really stop to think about it?
- Do we understand how is this person thinking?
- Do we care what he feels? Do we need to care at all?
- Do we have a good sense of what are the working / living / thinking constraints dictating the needs and actions?



It is NOT ONLY about WHO this person IS, but most importantly about WHY does he/she wants to do what they do!

It is easy to say that you are focused on your users, that all the decisions you make on the product are in order to provide them more features or to make the product more stable. Many developers and product owners will even explain how they are sacrificing things that are important for the company in order to focus more the user.

But at the same time, we are missing the point altogether!

Did we stop to understand if the user agrees? Do we know our users enough to make these choices? Simply put, do we understand how our users FEEL about the features we are providing them?

The fact is that we don't spend any energies on who are users are (their personal profiles!) and why do they do what they do (their professional reality).

If you want to provide a solution that really fits the user it's not enough to know the user, you need to empathize with him.

**When you empathize with someone you not only know what he does,
you can also share the feeling of why he is doing it.**

How do you empathize with your users?

The answer is simple: learn more about them.

Go and meet users. See how they work and what “other things” they need to do as part of their jobs.

Ask them to walk you through their work while they are using your system. Get real-life feedback focusing on the WHY they do things, and not only on WHAT they are doing.

You can even dare to ask them what would they do differently in order to make their work better if they could.

The aim is to get enough information to put yourself in their shoes.

If for some reason you cannot go and meet your users in person – and there are many reasons why you will not be able to do this – try to get in touch with them in other ways. Use Skype or any other virtual meeting system. Even a combination of phone and emails will provide you value. Anything you do to be closer to your users will be better than doing nothing at all.

If you think that our users will not want to “waste” time on this, you are wrong! Your users are eager to share with you their thoughts if you are only willing to listen.

Create personal profiles



After you meet a bunch of users and understand their needs (no need to meet all of them...) create schematic personal user profiles to help you and the rest of your team understand and relate to them better.


These profiles should include professional, personal and background information (if possible add a picture of this user at work with your system). The profiles should include, for example, a name as well as age, education, professional aspirations, work times, noise conditions, devices they use to work.

They should also include the number of children they have, hobbies, social media they use, etc. All things that will help you relate to them and to make a mental picture of their needs and preferences. Once you have these profiles you can evaluate your product, its features and workflows based on their “individual” personal profiles, and help all the team to develop and provide a product that will be better suited to your users.

You should not be surprised when quickly enough the whole team relates to these users by name and when you start referring to them as if they were real people everyone knows and works with on a day-to-day basis.

Do you empathize with your users? Does it help?

If you've tried this and succeeded or even failed I will be happy to hear your experience!



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

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

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

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

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

STATE *of* TESTING

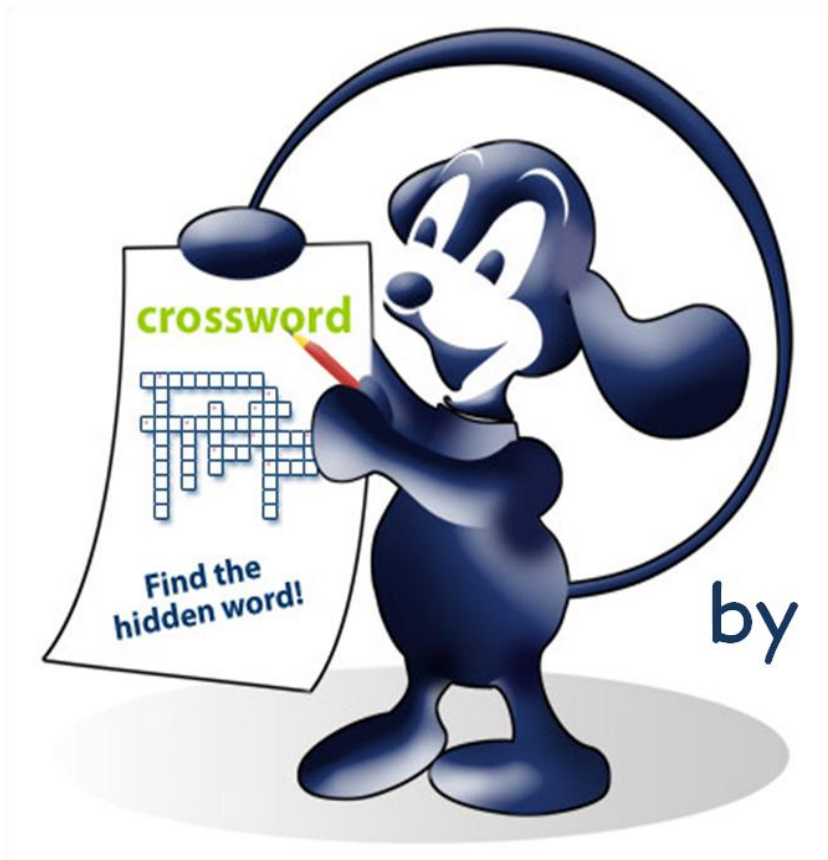
2013

Download Survey Report



Back To Index





by



Claim your **Smart Tester of The Month** Award. Send us your answer for Crossword b4 25th March 2014 & grab your Title.

Send -> editor@teatimewithtesters.com with
Subject: Testing Crossword



TESTING CROSSWORD



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

Horizontal:

1. It easy to write powerful automatic black-box tests for Android applications (8)
5. It is a document detailing a systematic approach to testing a system such as a machine or software (8)
7. An instance of an output, in short form (2)
8. An iterative incremental framework for managing projects commonly used with agile software development (5)
11. Checks for memory leaks or other problems that may occur with prolonged execution, in short form (2)
12. Test to ascertain how the components of a system are performing, given a particular situation (8)

Vertical:

1. An Automated functional testing and regression testing software (8)
2. It is a desktop bug tracking system (5)
3. A program that generates test cases in accordance to a specified strategy, in short form (2)
4. A free and open source web based bug tracking tool (6)
6. It is the testing of a resource or resources multiple times under program control, in short form (2)
8. A test case design technique in which test cases are designed to execute state transitions, in short form (2)
9. It is an architecture style of networked systems (4)
10. Testing the ease with which users can learn and use a product, in short form (2)

Click [here](#) for answers to last month's crossword

Advertise with us

Connect with the audience that MATTER!



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

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

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

Well, hear directly from them.

And the **Good News** is...

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

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





Every Tester

who reads Tea-time with Testers,

**Recommends it to friends and
colleagues .**

What About You ?

in ne>xt issue

articles by -

Jerry Weinberg

Pual Gerrard

T Ashok

Jim Holmes

JeanAnn Harrison

Joel Montvelisky

...and others

our family

Founder & Editor:

Lalitkumar Bhamare (Pune, India)

Pratikkumar Patel (Mumbai, India)



Lalitkumar



Pratikkumar

Contribution and Guidance:

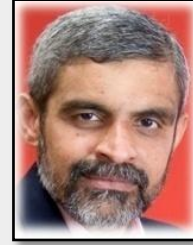
Jerry Weinberg (U.S.A.)

T Ashok (India)

Joel Montvelisky (Israel)



Jerry



T Ashok



Joel

Editorial | Magazine Design | Logo Design | Web Design:

Lalitkumar Bhamare

Core Team:

Dr.Meeta Prakash (Bangalore, India)



Dr. Meeta Prakash

Testing Puzzle & Online Collaboration:

Eusebiu Blindu (Brno , Czech Republic)

Shweta Daiv (Pune, India)



Eusebiu



Shweta

Tech -Team:

Chris Philip (Mumbai, India)

Romil Gupta (Pune, India)

Kiran kumar (Mumbai, India)



Kiran Kumar



Chris



Romil

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

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

Google™

Join our community on

facebook.

Follow us on



www.teatimewithtesters.com

