THE INTERNATIONAL MONTHLY FOR NEXT GENERATION TESTERS

Tea-time with Testers

February 2016 | YEAR 5 ISSUE XII

Articles by –

Jerry Weinberg

John Stevenson

T Ashok

Parimala Hariprasad

Ravikumar BN

Over a Cup of Tea with Maik Nogens

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

Created and Published by:

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

Editorial and Advertising Enquiries:

- editor@teatimewithtesters.com
- sales@teatimewithtesters.com

Lalit: (+49) 15227220745
Pratik: (+49) 15215673149

Editorial

What are we up to?

Thanks to all who wrote me wishing best on completing 5 years of Tea-time with Testers. Your response has always been overwhelming and we thank you once again for your love.

As we are always trying to give our best in new and newer ways, here are couple of things we are working on currently and you can expect to hear more on than soon.

- State of Testing Report 2016 is almost ready and we'll publish it soon. So, if you are interested then keep your eyes and ears open.
- We want to make our 'interviews' more interesting than before and going forward you'll see some interesting initiative from us on that front. 'Unsung Heroes' is going to be our theme once in a while as I think that not every amazing tester I have known personally is known to many in our community at large.

Tea-time with Testers has always made special efforts to make worthy thoughts, ideas, initiative and people reach wider audience and we would continue to do so. There are some other cool things we are currently working at but it's too early to talk about them. Let's let there be some surprises? ☺

And hey, we have a new member joining our editorial team this year. I welcome Dirk Meißner on the board and I'm sure that his contribution will make your reading experience even more fun.

That's all for this time. Enjoy reading this issue, wish you happy Easter in advance and keep rocking!

Sincerely Yours,

[illegible] - **Lalitkumar Bhamare**

editor@teatimewithtesters.com
@Lalitbhamare / @TtimewidTesters

QuickLook

LST News

All About Software Tools

Editorial

What's making News?

Tea & Testing with Jerry Weinberg

Speaking Tester's Mind

Testing Skills: Leading Teams - 11

What Color Do You Choose? - 22

In the School of Testing

Review of WAPT Tool – 30

Testing Debt in Agile Projects - 38

T ' Talks

SEVEN Habits of an Effective Tester– 46

Over a Cup of Tea with Maik Nogens

Family de Tea-time with Testers

1st Annual Test Leadership Congress/QA Expo

Test Masters Academy, the division of A Quality Leadership Institute, founded by Anna Royzman in 2015, announces the Grand Opening Spring Program April 25-28th in New York City.

1st Annual Test Leadership Congress/QA Expo, April 27th. This year's theme is "Test Management Agility". Test Leadership Congress/QA Expo is the first in its kind, as it will bring together the leaders and managers in software testing for an opportunity to share and debate ideas and techniques they find successful. By spending a full day at the conference, attendees will forge new vital contacts, gain inspiration by discovering successful applications of modern trends and practices which could have an immediate impact for their current work in 2016 and their future career success. Opening Keynote by Fiona Charles "The Dying Art of Test Management". Speakers from Spotify, Etsy, Huge, Johnson and Johnson, major financial institutions and more.

Accompanying conference, several well-known educators will teach full day and half day masterclasses in the area of their expertise on April 25th-28th. The classes are selected to appeal to testers and managers looking to improve their professional skills, and to the larger technology community, interested in improving the quality of their products. The tutorials are covering in-depths automation, mobile testing, agile practices, leadership, critical communication skills and even the basics of testing itself. Classes by Richard Bradshaw, Stephen Janaway, Fiona Charles, Kate Falanga, Jess Ingrasselino and Noah Sussman.

A note for the sponsors/QA Expo participants: becoming a sponsor will gain you an exposure and a direct contact with 100+ decision-makers in the Software Testing and Quality Assurance market. By supporting our 1st Annual Test Leadership Congress, you get in on the ground floor in making your name synonymous with test and quality leadership innovation.

1st Annual Test Leadership Congress

"Test Management Agility"

We brought you years of practice, expertise, innovation and thought leadership.

Speakers from Spotify, Etsy, Johnson & Johnson, major financial institutions, and more.

Opening Keynote: Fiona Charles "The Dying Art of Test Management"

Masterclasses

Well-known educators will teach full day and half day masterclasses in the area of their expertise.

The classes are covering in-depths automation, mobile testing, agile practices, leadership and even the basics of testing itself.

TESTMASTERSACADEMY.ORG

My first virtual meeting with Maik happened few years back when we were together judging NRG Global's Testing Competition program lead by Matthew Heusser.

And then we kept on discussing testing via some or the other forum. Maik is working as Senior Test Consultant with Diaz & Hilterscheid and has over 20 years of experience in testing. What I like and appreciate most about Maik is his down to earth nature and his passion for learning. Tell him about something new in testing field and Maik will jump on it like moth on fire ☺.

I finally met Maik here in Hamburg and it's been great discussing testing with him in person. I also appreciate his (and Alina's) efforts behind running testing meet-up for English speaking testers in Hamburg (not because I was invited to speak at inaugural meeting) and it is getting great response already. It's indeed an honor to be knowing the person behind GATE, PotsLightning, STUGHH, ASQF SIG "Agility" and with infectious passion for learning.

And how could I not interview such passionate tester for Tea-time with Testers? Read on to know what Maik thinks about different things around testing….

- Lalitkumar Bhamare

Over A Cup of Tea with Maik Nogens

Thanks for talking with us today, Maik. It's been a pleasure meeting you in person, finally

Thanks Lalit for inviting me and the pleasure is completely mine.

Well, I guess testers who are active in community don't really need your introduction but still, would you like to tell us about your testing journey in short?

My first commercial job in 2000 was of testing software, with no guidance or formal knowledge, but it connected well with my persistence and curiosity, the famous "what if…?" kind of nature.

After working in the Middle East, my passion for testing was finally established and since then I am focusing on testing.

You are one among those highly experienced testers I have known who are never tired of learning new things. I personally find your passion for learning infectious. What motivates you?

I find stagnation and routine often boring and pointless.

While a certain degree of routine can be beneficial, for example being able to exercise test techniques or recalling fitting heuristics, if there is only routine, people can never grow and find out what they are capable of.

You are one of the active and well known figures in German testing community. Please tell us more about the testing community over there…

There is an established test community here around more "traditional" ISTQB scheme, especially in regions where automotive, medicine or banking industries are strong.

On the other hand, there is a new generation which emphasizes a good work life balance and with the advent of the "modern" agile work culture also the testing community is changing.

There are a lot of local user groups and events in the major cities, especially Hamburg and Berlin, which focus on agile topics and their impact on an agile testers work.

With the Berlin based Testing Consultancy Diaz & Hilterscheid there is also a group of professionals, who established a global conference ("Agile Testing Days") to bundle these local activities.

Recently people have come to know you through the 'Software Testing World Cup (STWC)" which we find an interesting initiative. Where did it all begin? And can you tell us about STWC 2016?

I got the inspiration from Matt Heusser's NRG global testing competition held in 2013. In the beginning I just wanted to repeat such event in Germany. After discussing with my boss, Jose Diaz, we decided to make it a bigger event.

In a few weeks I drafted the concept, as it stands today. Based on the Olympic Games idea, there are six continental online rounds and the Finals. Each continental winner team is flown to the Finals round, which is played on-site i.e. at the Agile Testing Days conference in Potsdam, Germany.

It was a huge success in 2014, with over 1000 teams and nearly 4000 testers participating.

With each event we try to improve and the newest experiment is giving back to the community with team feedback from the evaluation team, so that teams have a chance to improve themselves professionally.

You can find out more on www.softwaretestingworldcup.com; also about the upcoming events for this year.

I believe you have fair idea and experience of different schools of testing right from CDT, Miagi-Do, Standards School. What is your general experience when it comes to putting all these principles at work?

While working in a former company I found, that the separated, formalized approach, how testing was perceived and to be executed, was not working for us. With the exposure to Miago-Do and Context-Driven-Testing in particular and the discussions and exchange with the people behind it, I found better ways to establish and execute testing within a given work culture and respect their current needs as well.

Today I strongly believe that the agile way is the way to go. If US Army can implement "boots on the ground" decision making; I think any company can change.

Guess the quote below from Fleet Admiral Ernest J. King; (CINCLANT Serial 053, 21 Jan 1941) summarizes it well -

"… If they are reluctant (afraid) to act because they are accustomed to detailed orders…

If they are not habituated to think, to judge, to decide, and to act for themselves in...

We shall be in sorry case when the time of "active operations" arrives..."

What makes one a successful Agile tester according to you?

That's a tough question. I think, only in hindsight one can spot "success". And it is very subjective, your team can consider your impact successful, while you have the feeling, you could have done more or better. Or vice versa.

For me, currently it means: support your team as best as you can, raise the difficult or "obvious" questions, argue for quality, where it is needed and don't be an obstacle, where it is not needed.

And remember, what worked well in the last project/employer, will not work 100% the same for the next. Be open for change and respect the context.

Would you like to tell us about POTSLIGHTNING that you are associated with?

"PotsLightning" is the name of a non-profit, no-entrance peer event, where passionate testers could meet for a day and talk about testing. The first event was in 2012.

I scheduled the event the day before the Agile Testing Days conference to enable an international exchange of the German testers and international visitors and speakers.

The most interesting topic was in 2014, where testers from Australia to Germany talked about embedded systems, from huge farm machines with GPS to music instruments with software chips.

These kinds of events help to connect the community and also enhance the visibility of the many good testers we have in Germany.

What are the benefits and drawbacks of working as a consulting tester?

Working as a consultant means one has to travel and work with different clients, work cultures and software. This alone is interesting and a great chance to learn and improve. From SAP in banking to web and mobile stacks in social business, the range is great.

It also helps to reflect what works in certain work cultures and what not; the goal is always to assist the client with the best fitting testing for him.

The drawback is that often the consultant has to move on, when the contract is finished. Sometimes the interesting changes are just starting to happen at the client and I would like to see it through.

You have recently started Testing meetups for English speaking testers in Hamburg. How it is going? Any experience to share?

Since 2012 I am running the "STUGHH" (Software Tester User Group in Hamburg). This is a German speaking group. In my current project I am working with a lot of non-German speaking testers and colleagues. A tester friend of mine (Alina Avadani, keep an eye on her, she is an upcoming testing star!) and I had the hypothesis, that the English speaking testers are not on the same online platform as the STUGHH, but that there are still a great number of them in Hamburg.

In 2016 we started the experiment with a new English speaking user group. And we wanted it to be more practical and hands-on from the beginning.

So far it is amazing. We chose the Meetup platform and try a monthly event cycle. Our current experiment is a series of three events around "Microservices". The attendees are also not purely testers, also developer or HR folks show up and participate.

What is your opinion about changes happening in our industry lately and its impact on testers?

Personally I think the testing profession has improved a lot since 2008, when I started to get active in the community. With it also came an increase in visibility and responsibility. A more recent example is the VW emission scandal. It also brings the ethnic of testers more to the forefront.

We saw what Snowden and Assange had as an impact in their areas, respectively.

Where do we testers draw the line? It doesn't always have to be the "loss of life or money" situation, where it seems "obvious" (at least it should). What about data collection, tracking user behavior, data exposure? And what do we do, when we won't be heard inside the business?

What is the next big thing happening for testers in near future, according to you?

I think mobile testing has not reached its full potential and exposure to the business. On top of that, there is also Internet of Things and Virtual Reality, which seem to make it to the mass markets in a few years.

Currently I doubt that there will be significantly new or changed testing basics we need to invent, but the application of our testing knowledge to this new fields will be put to the test (pun intended). I am looking forward to that.

Another change I see and strongly propagate is the influence of testers into the business areas.

I know some former testers, who are now working as product owners, usability engineers and other more business related field. Considering that that is just a personal spot check, I assume the dark figure is even higher.

Having experienced this transition myself over the last 14 month; I feel this is a change, where testers can bring value to the business.

What are some testing books or blogs that you regularly read?

To be honest, the time I invest into reading at the moment is limited. Just now I listen to Appium courses on the Udemy online platform.

I try to keep up to date with a new content publisher page on the XING platform called "Software Quality", where I am also an editor.

The last book, which I finally read last year, was Kahneman's "Thinking, Fast and Slow". Nearly once a week I experience something happening, which connects back to the model and behaviors, he describes.

I find it also quite in line with Gigerenzer's "Gut Feelings: The Intelligence of the Unconscious", which gave me confidence in my early years as a tester, that my gut feelings are often fact-based.

And how can we not ask you about importance of attending conferences for testers? ☺

Conferences are great on so many levels. First, you can learn from the talks and sessions themselves. Second, most conferences offer "beginner" or "lightning talk" slots, where everyone can have a say about a topic close to their heart.

Plus if it is a real "from the people for the people" conference, there is a lot of spirit and energy going between the sessions. People make up their own sessions. Discussions go on all night.

And most presenters are totally approachable and that provides a unique chance to talk, question or exchange with folks, whose blogs, books or talks you read and watch during the rest of the year.

Your advice to our tester friends would be…

When you reach that low, where you question yourself or doubt your approach… reach out to other testers. Get their opinion or feedback. More often than not, you can get positive support and you will see that you are onto the True North. Keep testing and educating yourself!

Last question for today, what do you think about 'Tea-time with Testers'? We would like to get your feedback.

I know and read your magazine since few years now and I have always liked the mix of articles, interviews and news from our field. Keep it up and going!

How Culture Brokers Make Software Successful

Perhaps if I'd been using my fourth-generation Executive Time Management software package, the mistake never would have happened. But I couldn't even get the darn thing loaded, so I put the manual in my bottom drawer and fell back to my zeroth generation paper and pencil calendar. And somehow I managed to schedule, two lunch appointments for the same day.

Jake, who was IT manager at Ninth National Bank, suggested that we all have lunch together. And that, because of my mistake, I should pay for the lunch. My stupidity had left me with no good alternative.

I called Lorene at the IT section of Glittering Mines. She agreed to the lunch *å trois*—at an appropriately expensive French restaurant. It was going to be an expensive mistake.

After lunch, emboldened by a stomach full of lemon sole, I decided to talk about the software difficulties that engendered this situation.

Jake was immediately sympathetic: "You're not the only one who has trouble with so-called fourth generation software. I've spent $75,000 for a 'user-friendly' financial analysis package. It's so unfriendly that all the users cursed me for buying it."

"You obviously bought the wrong package," Lorene said. "For the same price, we got a financial analysis package that really is user-friendly. Our users love it, and they love us for buying it for them."

"What package is it, Lorene? Perhaps Jake could get a trade-in."

"It's called MONEYPENNY—"

"...but that's the stupid package we-bought!" Jake roared. "Your users must be a lot smarter than ours."

Lorene looked puzzled. "If you knew our users, you wouldn't say that. Maybe you got a different version of the system."

"Ours is version 3.12."

"So is ours. I don't understand it."

While Jake and Lorene ran through several other possible differences, I looked for loophole in the lunch bill. I couldn't see any way out of the bill, but I did see an opportunity to earn it back.

"I have an idea," I volunteered. "Perhaps there's something different in the way you two managed the introduction of the package. If you're interested, I can tackle the question as an add-on to my consulting contracts."

After a short negotiation, we agreed that I would interview a few users in each organization, as long as it didn't add anything to their costs. If I found anything interesting, then it night lead to future contracts.

At Ninth National, however, the users taught me nothing except a few new swear words. At Glittering Metals, the users were less emotional, but I didn't learn anything at all.

I decided I must have been mistaken. Something about MONEYPENNY made it more usable at one place than another. I asked Cliff, the last user I interviewed, if he'd let me watch him use MONEYPENNY. I promised to sit next to him and say nothing, regardless of what happened.

It seemed MONEYPENNY was an average "fourth generation" package. It had some nice features, but it didn't meet my human engineering standards. Still, Cliff seemed to be getting along all right... until the screen flashed this ominous message:

ILLEGAL ENTRY... COMMAND REJECTED

Cliff pondered the message for a few moments, then hit a few keys. Another message:

RECURSION ATTEMPTED IN PARAMETER

"What's recursion?" Cliff asked.

"It would be better if I remained an observer," I said, a bit cowardly. Just do whatever you would do naturally."

"Well, at least tell me what parameter is"

"Don't be angry, but I promised not to say anything."

"Okay, then call the Hot Line. Scott or Leatrice should still be there. They usually work late."

"Who are Scott and Leatrice?"

"They're the Hot Line Programmers for MONEYPENNY. Who did you think I would call?"

Being a good consultant, I played it cool. "Oh, the Hot Line Programmers, of course. I just didn't know their names."

Cliff talked to Leatrice, who gave him a magic formula to put the hex on his unwanted recursion. The hex worked, unlocking the screen and sending Cliff merrily on his way. After the session, I asked him if he ever had any trouble with MONEYPENNY.

"Trouble? No, nothing that I couldn't handle. It's a real user-friendly system. "

Cliff never mentioned Scott or Leatrice, but I thought I'd pay a visit to the other end of the Hot Line.

It turned out that the Hot Line is an institution at Glittering Metals, dating back to the dawn of punched card tabulating equipment.

"Programmers serve on the Hot Line mostly when they're between projects," Leatrice told me. "It's good experience—you never know what you'll get into."

"What kinds of things?"

"Everything and anything, I suppose."

"That's not very helpful. Don't you have a job description?"

"Not really. It's kind of a temporary assignment, so you just keep your old job category. I'm a Senior Programmer, but we have Systems Analysts, too. And one Systems Programmer."

Later, Lorene told me that the job of people on the Hot Line is to do whatever is necessary to make users happy. In the old days that might have meant running off an extra copy of some report they had lost and carrying it upstairs to them. Now it frequently meant figuring out how to get their user-friendly systems talking to them again.

"It sounds to me like a Hot Seat, rather than a Hot Line. What kind of people do you put in that job?"

Lorene pondered my question for a few minutes before answering. "I guess I look for generalists—people who can do a little bit of everything. They've got to be technical enough not to be snowed by systems problems, but they've got to be good communicators—good listeners especially—above all else. Not many of our technical people are good listeners."

"What else?"

"Well, they're problem-solvers. They don't get stuck. Somehow they get things going."

Having concluded my investigation, I went back to Ninth National. I explored the question with Jake, but he said they had nothing like the Hot Line. I then asked him if they had any user-friendly packages that had worked well.

"Sure," he said without hesitation. "There's a statistical analysis package that they use in Personnel They've never had a moment's trouble with it—but they certainly don't have a Hot Line."

I arranged to visit Personnel, where I soon discovered that their "Hot Line" was Arlo, a Senior Records Clerk who had learned to use the statistical package when he did a Master's thesis in Personnel Management. Arlo was the one who had suggested the package in the first place, and he was a gold mine of miscellaneous information about how to make it work.

Arlo wasn't part of the IT department, so Jake didn't even know of his existence. All the same, Arlo seemed to be the critical difference between success and failure of the statistical package.

When I returned from the trip, I discussed these events with Dani, my life partner and resident anthropologist. "What you seem to have discovered," she said, "Is what anthropologists call a *cultural broker*."

"Tell me more."

"Cultural brokers exist whenever two different cultures must interact in spite of their different languages, value systems, customs, and other barriers to communication. The cultural broker is a person who happens to have one foot in each culture, so can act as a go-between whenever the need arises."

I was impressed with the aptness of this description. Users often say that IT is a "different culture." IT-ers, on the other hand, certainly feel the same way about their customers and users.

A series of visits to other clients convinced me that behind every successful "user-friendly" system there were cultural brokers. Sometimes they were right out in the open. Sometimes they were so well hidden they almost blended into the wallpaper. But they were always there e. Once I became aware of this hidden function, I discovered cultural brokers under such titles as Customer Service Representative, User Service Specialist, Technical Specialist (with a variety of subtitles), Consultant (again with subtitles), and Systems Engineer.

And, since the advent of the Agile movement, I've found them operating openly as Customer Surrogates. The problem with this title, in my opinion, is that it suggests a one-way communication, ignoring the function of communicating with the customer about the culture of the Agile team. But such a hidden function is not exceptional. Most of the cultural brokers I've found are masquerading under conventional titles, including Programmer, Analyst, Systems Analyst, Systems Programmer, Trainer, Database Administrator, and Business Analyst.

Throughout my consulting career, whenever I find a successful system, I've hunted for the cultural brokers. Though they were called by different names, attached to different departments, and had different job descriptions, there were cultural brokers everywhere software systems were successful.

Here are just a few examples.

A large bank has a department staffed with "Technical Specialists." Organizationally, Information Center reports to the IT manager, but physically it is in a separate office close to the two major user groups.

The Technical Specialists support whatever packages are purchased for users with such activities as training new users, consulting on immediate problems, helping users write specs for add-ons, and negotiating fixes with vendors.

The IT manager says that the title "Technical Specialist" was purposely vague, so as not to prejudice anyone against doing anything necessary to get the users what they want. He says: "Sometimes they come over here and convince us to do things for the users that I can't believe I'm agreeing to. And I hold their pay checks!"

In the actuarial department of one insurance firm, buying software packages is an established practice. Whenever a new package is contemplated, one of the younger actuaries is assigned the job of making the selection, or deciding to build their own. Once a package is selected or built, the chooser becomes the cultural broker for that system—under the informal title of Technical Specialist for the XYZ System.

Glittering Metals had their Hot Line system, in which the cultural brokers were any technical people who happened to be under-loaded. In several other installations, essentially the same system was in place with no formal recognition from management. Users simply learned the name of someone in IT who "knew the answers," and that person's name got passed from user to user.

In another bank, the investment analysts were happily using a new system without any outside help. In fact, they were so happy with the system that they started to swamp the bank's on-line database. A database specialist who was sent upstairs to investigate discovered that their "programming" not only consumed their time but also produced many subtle bugs that invalidated their analyses.

The investment manager requested that the database specialist be permanently assigned to assist the analysts, thus creating a new cultural broker.

In the marketing department of an electronics manufacturer, their informal hot line system changed when the informal cultural broker departed for greener pastures, leaving the users with no effective cultural broker. Without too much trouble, one of the heaviest users of the program generator acquired the cultural broker role. The job moved out of IT entirely, but management hadn't known it was there in the first place, so it didn't cause a ripple at the upper levels.

Some of the marketing users, however, thought that they were probably not employing the full power of the package, because without technical support from IT, they shied away from some of the more "technical" features. In one specific example, someone tried to use a formatting feature but didn't succeed. When the cultural broker couldn't figure out what was wrong, the user decided to use a less convenient format.

The situation in this electronics firm suggested that I might be able to find places where some potentially useful software wasn't being used because no cultural broker was available or forthcoming. It's a bit difficult, though, to attribute causes to events that *didn't* happen, so I had to make some inferences. For instance, I noticed that the larger the group of potential users of a system, the more likely one of the users was to become a cultural broker.

In the case of Ninth National Bank, only three potential users had actually seen a need to try the system. None of the three evidently had sufficient motivation or technical bent to get it going, and it withered away on a shelf.

In a situation like that, the IT department could have increased the chances of success by supplying a cultural broker, at least during the early stages. Or they might have tried to create a larger user community.

Even though a small group of users is less likely to create its own cultural broker, there will always be some individuals who have just the right combination of skill and motivation to succeed with a new system without assistance. Some of these individuals become missionaries for the new system, recruiting other users with such enthusiasm you begin to suspect they're receiving a finder's fee for each convert. Actually, their biggest payoff seems to be acquiring the role of cultural broker, with concomitant respect and attention from their colleagues.

And the future for programmers and analysts? Commercial applications are supposed to "eliminate the need for programmers." But the simple truth is that they're not that well designed or well implemented and won't be for a long time. And, the more important information systems become to their users, the more critical the programmer's role becomes.

No, I think both programmers and analysts will still be around for a long time. But their roles will change. One such role is that of systems programmer. Installing and maintaining individually programmed systems is straining the supply of competent systems programmers. As a result, some applications programmers are moving in an even more technical direction.

Another changing role is system designer. If systems are to become truly user-friendly, we're going to need ten times as much work on their designs before they even reach their users.

But the greatest new demand is for programmers and analysts to move in the other direction—not so much further from the technical side as closer to the users' side.

The culture brokers I have encountered have retained or acquired the basic technical skills of programming, but also have more than the typical programmer's share of "people skills." These culture brokers, whatever they are called, are first of all good listeners.

Secondly, they tend to be generalists—interested in more than one thing and capable of learning new things quickly. And this includes a strong interest in people—they like people and enjoy helping them.

Good culture brokers are good problem solvers they get the job done, regardless of what it takes. If software is going to continue to succeed, we'll need a lot more such people. It would pay us to start finding them and training them straight away.

Back To Index

People Skills—Soft but Difficult

Curious to know why you should get 'People Skills' bundle by Jerry? Then check this out

Biography

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

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

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

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

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

Jerry Weinberg has been observing software development for more than 50 years.

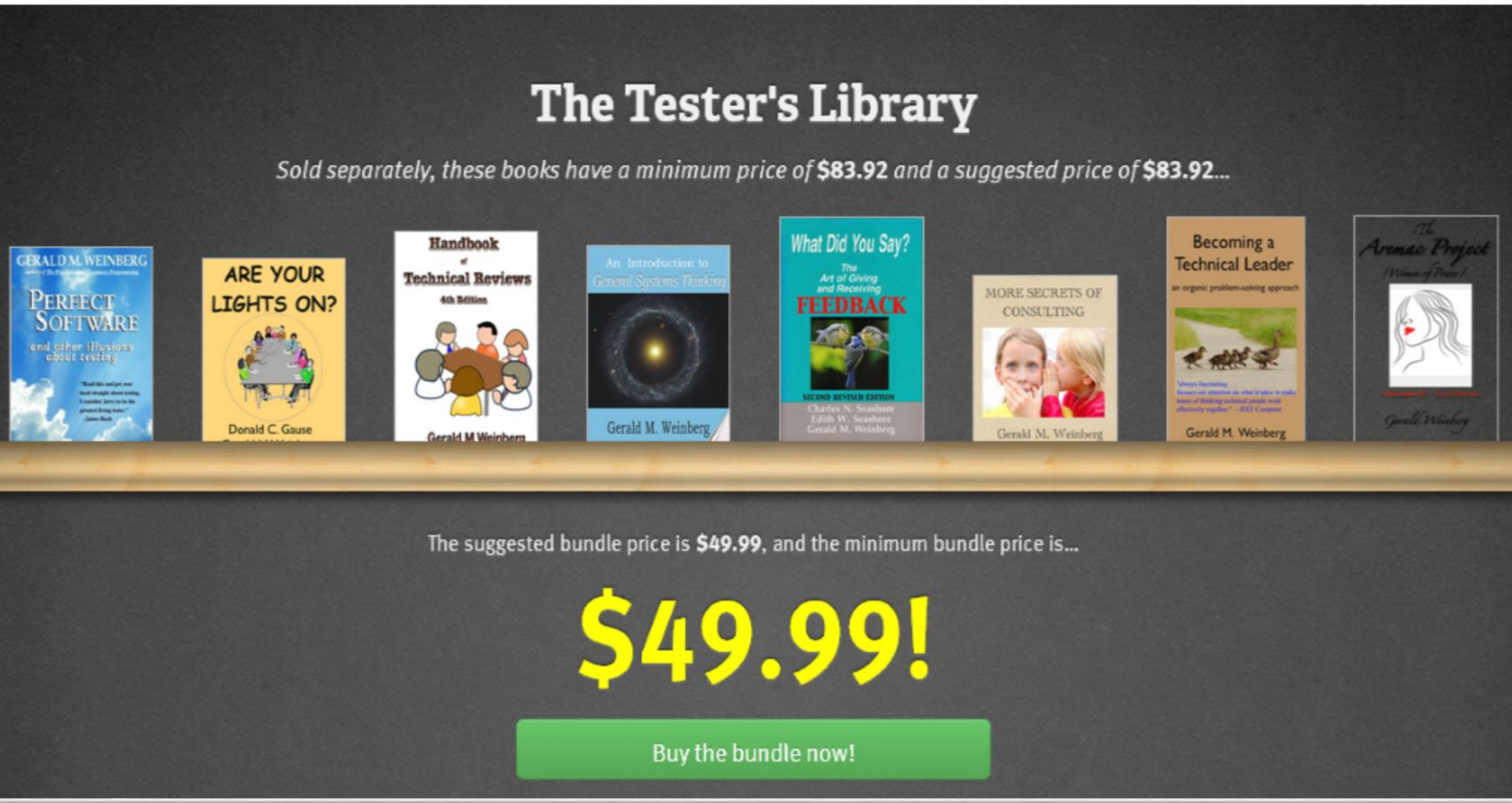
Lately, he's been observing the Agile movement, and he's offering an evolving set of impressions of where it came from, where it is now, and where it's going. If you are doing things Agile way or are curious about it, this book is for you.

Know more about Jerry's writing on software on his website.

TTWT Rating: ★★★★★

The Bundle of Bliss

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



The Tester's Library

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

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

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

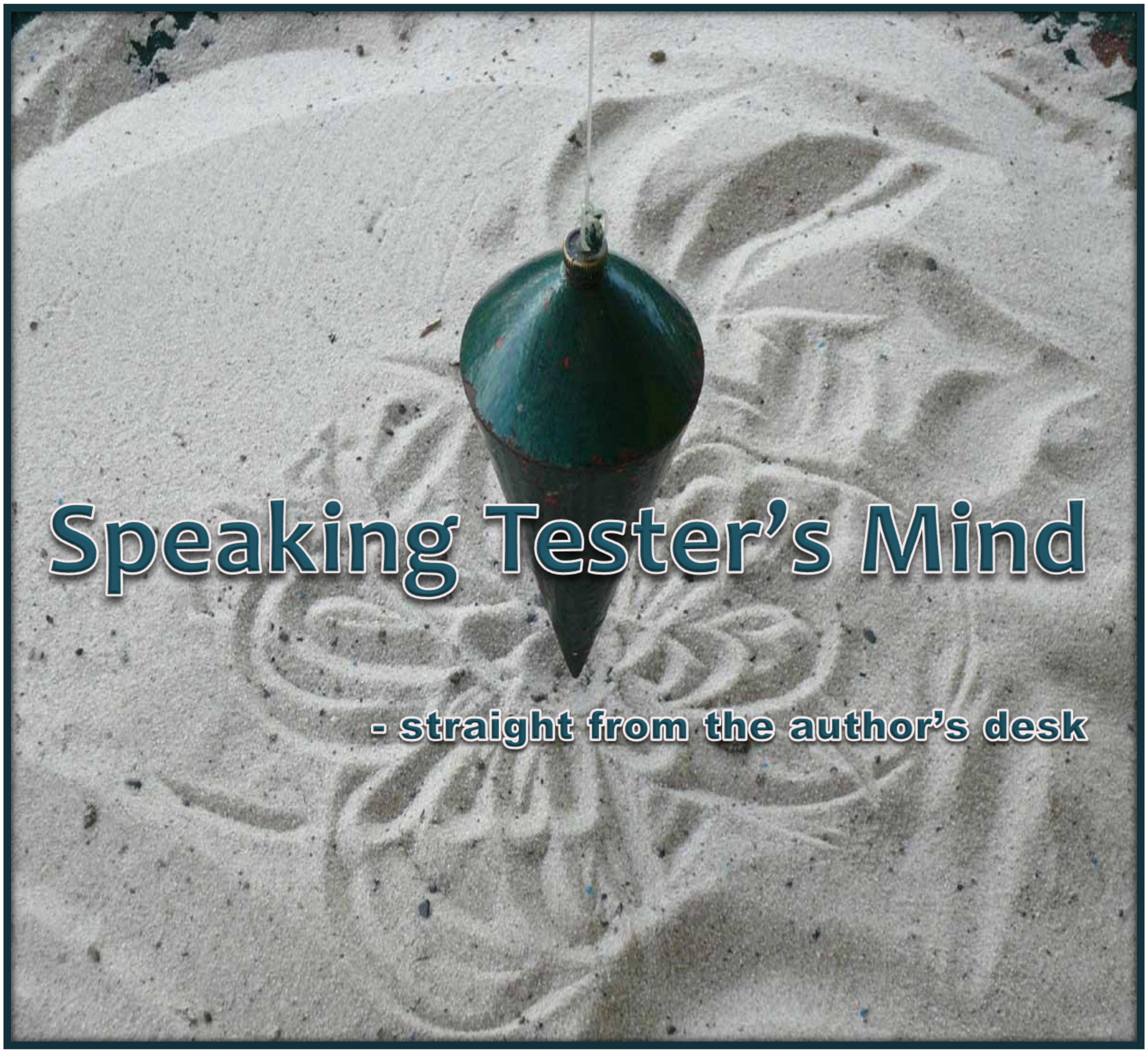
$49.99!

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

The 8 books are as follows:

- Perfect Software
- Are Your Lights On?
- Handbook of Technical Reviews (4th ed.)
- An Introduction to General Systems Thinking
- What Did You Say? The Art of Giving and Receiving Feedback
- More Secrets of Consulting
- Becoming a Technical Leader
- The Aremac Project

Know more about this bundle



Speaking Tester's Mind

- straight from the author's desk

Testing Skills
Part 3: Leading Teams

- by John Stevenson

If you are leading a software development team it is a good idea to provide to provide an environment in which each member of the team feels safe, valued and important. It takes a lot of skill to lead teams without resorting to command and control style of managing the team. Providing the team a safe environment in which it is OK to experiment a little and be able to learn from failing is crucial to encourage creativity and innovation within the team.

When leading a team in this way you can act as the facilitator and ensure that the bad elements of teamwork do not start to impact the team dynamics. You can be the sounding board, the oracle, the voice of reason, the mentor, the confidant, by doing this you encourage the team to flourish which leads to success.

As a leader you also need to give the team a clear direction of what you expect the team to deliver, this can change as time passes however, this should always be transparent and visible to the team. Ask yourself as the leader what do you want from the team? What is your ultimate goal and then explain that to the team along with your preferred priorities. Then step back and watch the team self-form around the problem to provide you with a solution.

The principles behind the agile manifesto has some useful tips for leading and working in teams.

"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."

"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."

"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."

J Richard Hackman provides the following five tips for leading teams.

1. Teams must be real. People have to know who is on the team and who is not. It's the leader's job to make that clear.

2. Teams need a compelling direction. Members need to know, and agree on, what they're supposed to be doing together. Unless a leader articulates a clear direction, there is a real risk that different members will pursue different agendas.

3. Teams need enabling structures. Teams that have poorly designed tasks, the wrong number or mix of members, or fuzzy and unenforced norms of conduct invariably get into trouble.

4. Teams need a supportive organization. The organizational context—including the reward system, the human resource system, and the information system—must facilitate teamwork.

5. Teams need expert coaching. Most executive coaches focus on individual performance, which does not significantly improve teamwork. Teams need coaching as a group in team processes—especially at the beginning, midpoint, and end of a team project.

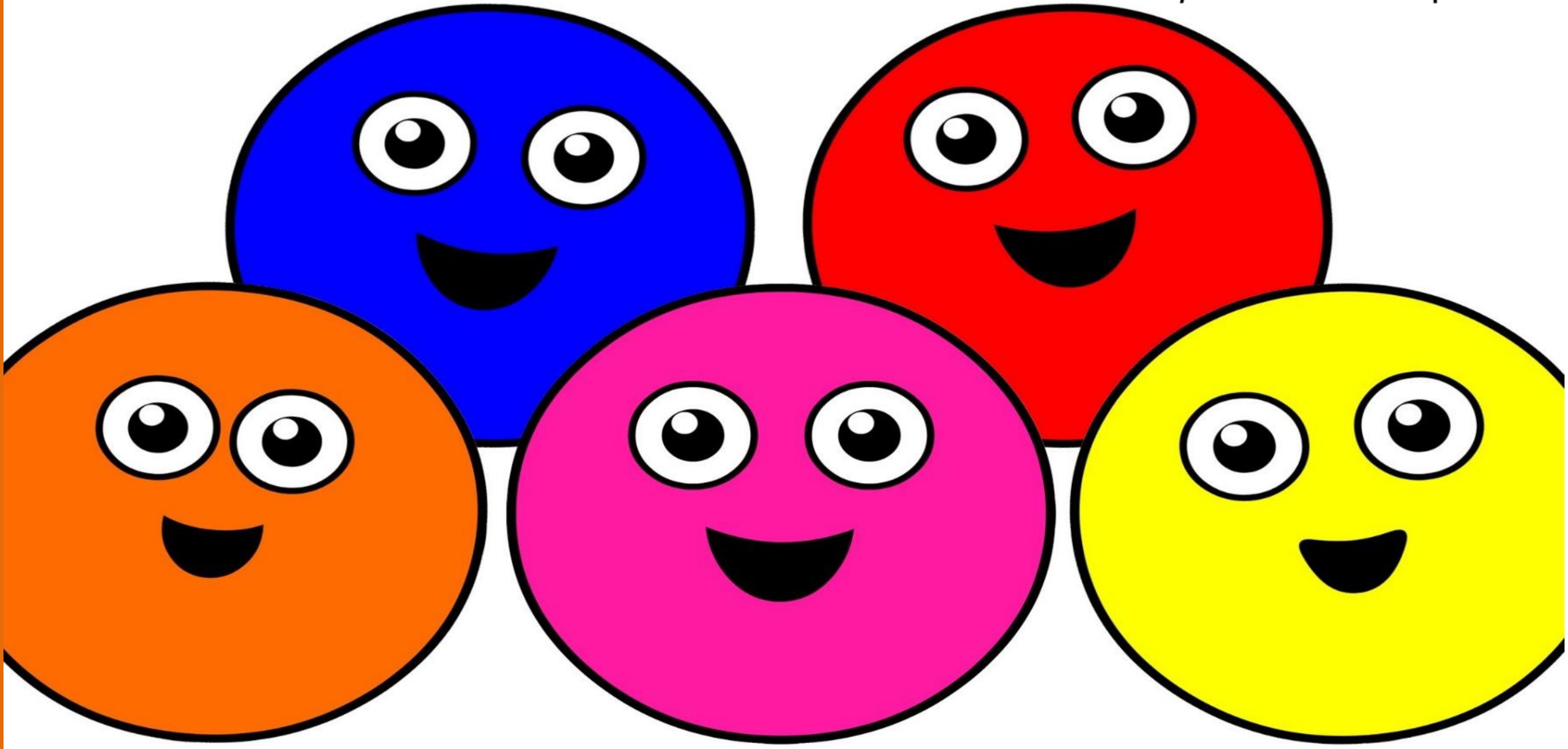
Let's talk about 'Note Taking' as skill in my next article.

John is tester, blogger, tweeter and author who has a passion for the software testing profession. He is keen to see what can be of benefit to software testing from outside the traditional channels and likes to explore different domains and see if there is anything that can be of value to testing. At the same time he likes to understand the connections between other crafts such as anthropology, ethnographic research, design thinking and cognitive science and software testing. He is currently writing a book on this called "The Psychology of software Testing".

John has presented workshops and presentations at various events such as Agile Alliance, CAST, Testbash and Let's Test.

Color, Color, What Color Do You Choose?

- by Parimala Hariprasad



In November last year, Twitter changed its star icon indicating favorites to a lovely heart. While folks at Twitter called them likes, many twitter users started calling them hearts. Akarshan, the product manager who led this effort at Twitter, "We want to make Twitter easier and more rewarding to use, and we know that at times the star could be confusing, especially to newcomers. You might like a lot of things, but not everything can be your favorite. The heart, in contrast, is a universal symbol that resonates across languages, cultures, and time zones. The heart is more expressive, enabling you to convey a range of emotions and easily connect with people. And in our tests, we found that people loved it."

Check out this video to see how the new like/heart feature works.

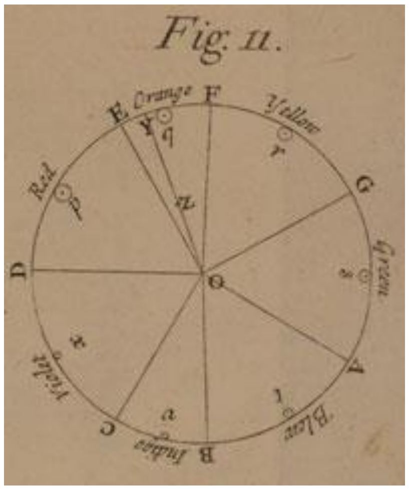
Around same time, Google and HP changed their logos, and key players like Myntra, Flipkart and others changed their color schemes. Colors like 'Red', 'Blue' and 'Yellow' became prominent for several mobile apps. It left me wondering about why we obsess with colors?

What's in a Color?

Color is an engaging and powerful form of communication. Color piques emotional interest in products. Many times, even the best design gets trumped if appropriate colors are not used. For designers, color is about expressing their emotions and bringing in certain emotions in users. For users, color is a key to creating perception about brands. For e.g. the color red might remind us of McDonalds, Pizza Hut or Virgin Atlantic.

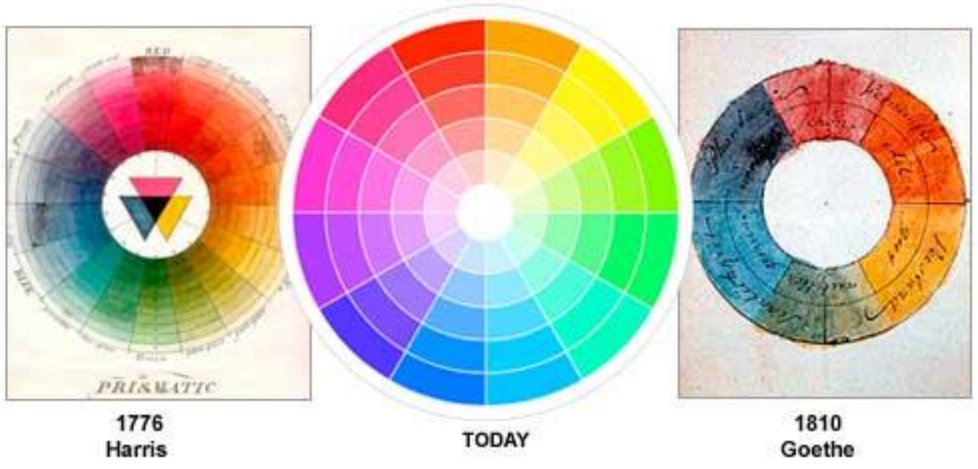
Color Wheel

A **color wheel,** also known as **color circle** is an abstract representation of different colors around a circle that shows relationships between primary colors, secondary colors, tertiary colors etc.



Newton's asymmetric color wheel that correlates colors with musical notes and planetary symbols.

In 1666, Sir Isaac Newton's work with white light led him to the discovery of the visible spectrum of light. So, the first color wheel dates back to Newton's period (displayed in picture above). Newton's experiments led to the theory that red, yellow and blue were the primary colors from which other colors are derived. Since then, several variations of this concept have evolved with time. A short history of the color wheel is represented in the picture below (image Credits: www.colormatters.com).



The Color Theory

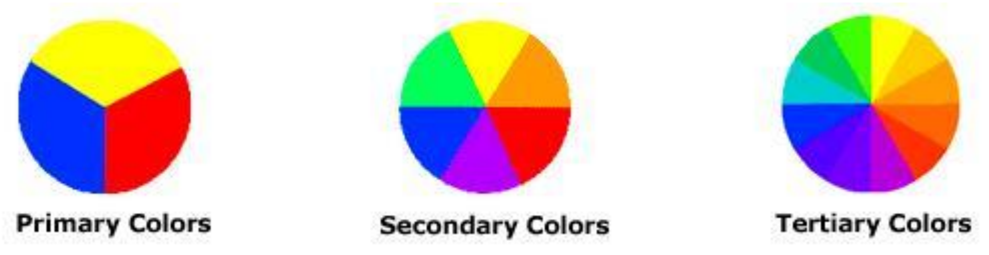


Image Credits: www.colormatters.com

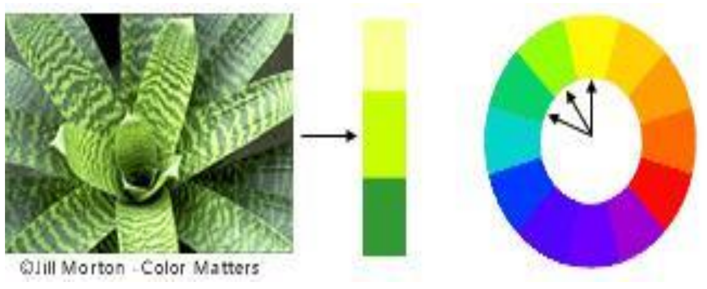
According to color theory, there are three **primary colors**: red, blue and yellow. These cannot be mixed or formed by any combination of other colors. Mixing two primary colors together creates the **secondary colors**: green, orange and purple. **Tertiary colors** are formed by mixing a primary and a secondary color. E.g., blue-green, yellow-green, red-orange, red-purple and so forth. These days, color experts have come up with unique names for secondary and tertiary colors.

Color Harmony

In visual experiences, harmony is something that is pleasing to the eye. It engages the viewer and it creates an inner sense of order, a balance in the visual experience. When something is not harmonious, it's either boring or chaotic. Color harmony delivers visual interest and a sense of order.

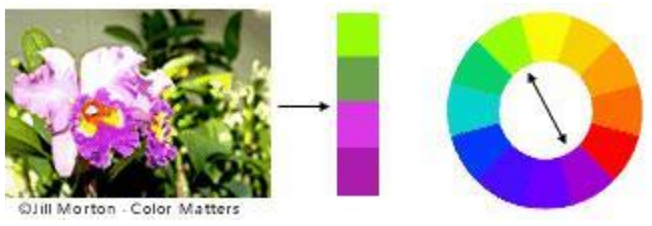
www.colormatters.com has this to say about harmony between colors: There are many theories for harmony. The following illustrations and descriptions present some basic formulae:

1. A color scheme based on analogous colors



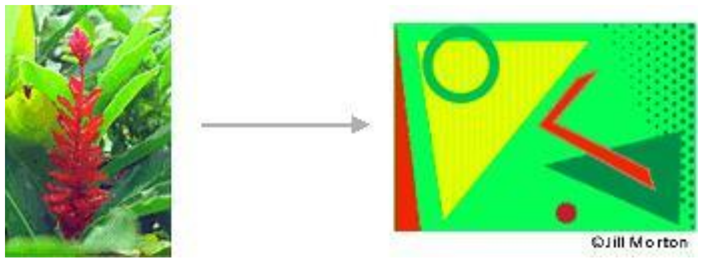
Analogous colors are any three colors which are side by side on a 12 part color wheel, such as yellow-green, yellow, and yellow-orange. Usually one of the three colors predominates. Analogous colors are excellent for creating color chemistry using three or four of them together, such as red, orange, and yellow, or blue, green, and yellow.

2. A color scheme based on complementary colors



Complementary colors are any two colors which are directly opposite each other, such as red and green and red-purple and yellow-green. In the illustration above, there are several variations of yellow-green in the leaves and several variations of red-purple in the orchid. These opposing colors create maximum contrast and maximum stability. Although they have been used by artists for centuries, care should be taken when using complementary colors together. Although they will make each other vivid, used together they can also be difficult to focus on.

3. A color scheme based on nature



Nature provides a perfect departure point for color harmony. In the illustration above, red yellow and green create a harmonious design, regardless of whether this combination fits into a technical formula for color harmony.

The Role of Color in Visual Interface Design

Color plays many roles in Visual Interface design. These are the prominent ones:

<u>Visual Cues</u>

Colors, in their simplest role, provide visual cues to evaluate a situation or perform an action. Consider an example of a traffic signal – Red means 'Stop', Green means 'Go' and so forth. This rule is applied on mobile devices as well for Incoming Calls, where tapping on Red icon disconnects the call while tapping on Green, connects the receiver to the caller.

<u>Relationships between Heterogeneous Objects</u>

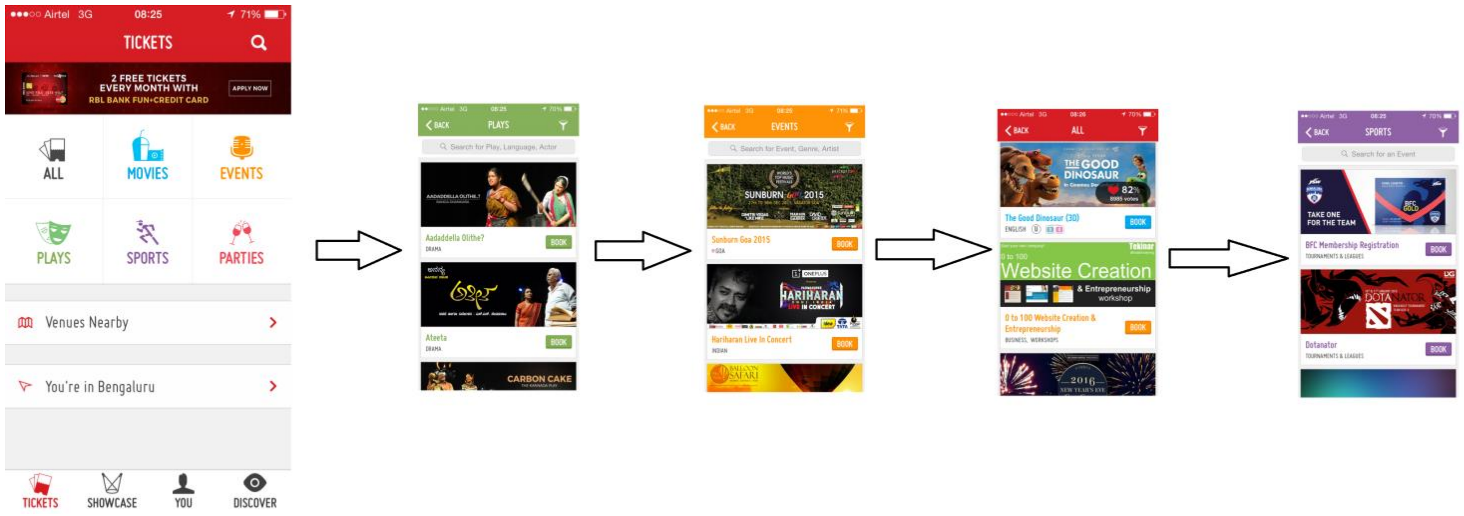
Color can be used to establish relationships between similar and dissimilar objects. For example, buttons on a website or app could all be one color. Tab bar items or navigation links could be in another color and so forth.

<u>Priority / Importance</u>

Important / Critical items can be colored in strong, highly saturated shades to grab attention from users. For example, a highly saturated red button on an otherwise white background with light colored UI elements will attract users easily.

Book My Show app's approach to Colors

Book My Show's overall branding had Red as prominent color, mostly in header section on landing screen and other generic screens.



One striking thing is the method Book My Show has used to created memorability on the app. Movies are represented in sky blue, Events in orange, Plays in green and Sports events in Purple. If a user enters 'Sports' section, purple becomes a prominent color mostly for 'Action' buttons. For a user who has used the app a couple of times, it becomes easier to relate to specific events by associating them with colors. Note that this association works well if there are fewer items - in this case, there are 6 prominent colors. If there are more categories, adding too many colors to differentiate them might become an overhead.

Closing Thoughts

Dan Saffer, in his book '*Designing for Interactions*' states, "Knowing basic principles of the Color Wheel will help designers avoid clashing colors. The painter Josef Albers noted that a color doesn't exist until it meets another color. Designers should look at the edges of colors and see how they work together. White backgrounds tend to darken, and can even deaden, colors. Black backgrounds tend to lighten colors. Try backgrounds that aren't pure white or black, such as light gray. Very pale yellow backgrounds with black type are good for older eyes. Care should be taken to avoid an effect called chromo stereopsis, which occurs when two colors side by side seem to cause both colors to vibrate. Red and blue together for instance, blue text on a red background will often create this effect, irritating viewers' eyes. In general, colored text on a colored background is difficult to execute well."

The Button Color A/B test goes on, to prove how critical role, colors play in deciding the likeability of products. Concepts from color wheel helps designers create interesting palettes by applying underlying principles of color theory. Designers should become familiar with the color wheel which is helpful not only for choosing a color scheme, but also for avoiding color-related problems.

So, the question to you, like the color game we used to play as kids is: "Color, Color, What Color Do You Choose?"

References

- http://www.colormatters.com/color-and-design/basic-color-theory
- Book : Designing for Interactions by Dan Saffer

Parimala Hariprasad spent her youth studying people and philosophy, and she applied her learnings to become a better person. Parimala has worked in domains of CRM, security, e-commerce, and now, travel for over 12 years. She is a UX Alchemist who helps with UX Strategy and assists teams on Interaction Design, Content Strategy, Expert reviews, User testing and Coaching. Parimala has experienced the transition from web to mobile and emphasizes the need for Design Thinking in products.

She frequently rants on Mobile User Experience Design on **her blog**, tweets as **@CuriousTester** and is on **LinkedIn**. Parimala currently serves as Senior UX Architect for Mobile Competency Center at Amadeus Software Labs, Bangalore.

Back To Index

Love to Write?

Your ideas, your voice. Now it's your chance to be heard!

Send your articles to editor@teatimewithtesters.com

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

Review of WAPT - a performance testing tool

While looking for a good performance-testing tool, I stumbled upon WAPT and got immediately attracted by its user-friendly design. This tool gives you freedom to generate a load, which stands pretty close to real life users.

As I went on evaluating this tool for my own requirements, I thought my research would come handy for those who are looking for something similar. And hence this post.

Let's see what WAPT is about in more detail:

WAPT focuses mainly on web application performance testing (HTTP/HTTPS protocol). The tool has two variants, WAPT and WAPT Pro. Both variants offer very user friendly GUI which helps to record and debug test scripts with minimum effort.

WAPT allows you to record any kind of web application using desktop browser or mobile browser.

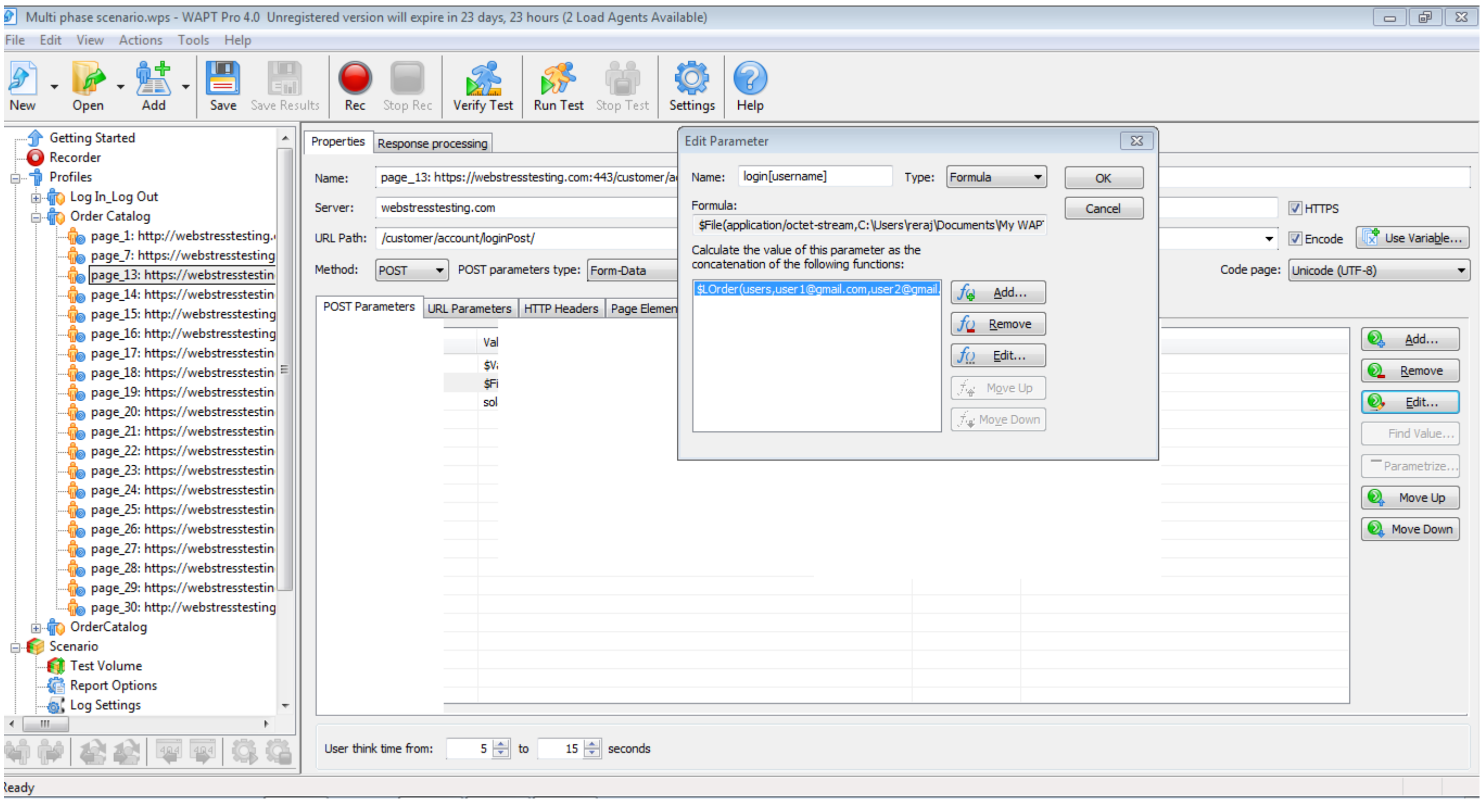
Key features of WAPT:

1. **Recording user profiles or scripts**

   Recording user profiles is made very easy in WAPT (similar to many other performance testing tools). You click on the record button and can go through the user actions. But the interesting part of WAPT comes after recording. Zero coding or very little coding is required and almost all actions are included as functions. Hmm… nice!

2. **Parameterisation**

   Parameterisation is also made easy in WAPT by inbuilt functions that enable the use of a list of values as input. For example, the "**Ordered List"** function returns the values in the list one by one where as "**Random from list"** function will return random lines from the list of values.


Multi phase scenario.wps - WAPT Pro 4.0 Unregistered version will expire in 23 days, 23 hours (2 Load Agents Available)
File Edit View Actions Tools Help
New Open Add Save Save Results Rec Stop Rec Verify Test Run Test Stop Test Settings Help
Getting Started
Recorder
Profiles
Log In_Log Out
Order Catalog
page_1: http://webstresstesting.
page_7: https://webstresstesting
page_13: https://webstresstestin
page_14 … page_30
OrderCatalog
Scenario
Test Volume
Report Options
Log Settings
Properties
Response processing
POST Parameters
URL Parameters
HTTP Headers
Page Elemen
Add... Remove Edit... Find Value... Parametrize... Move Up Move Down
Ready

Name: page_13: https://webstresstesting.com:443/customer/a

Server: webstresstesting.com

URL Path: /customer/account/loginPost/

Method: POST POST parameters type: Form-Data

HTTPS ✓ Encode ✓ Use Variable...

Code page: Unicode (UTF-8)

User think time from: 5 to 15 seconds

Edit Parameter

Name: login[username] Type: Formula

Formula:
$File(application/octet-stream,C:\Users\reraj\Documents\My WAP

Calculate the value of this parameter as the concatenation of the following functions:

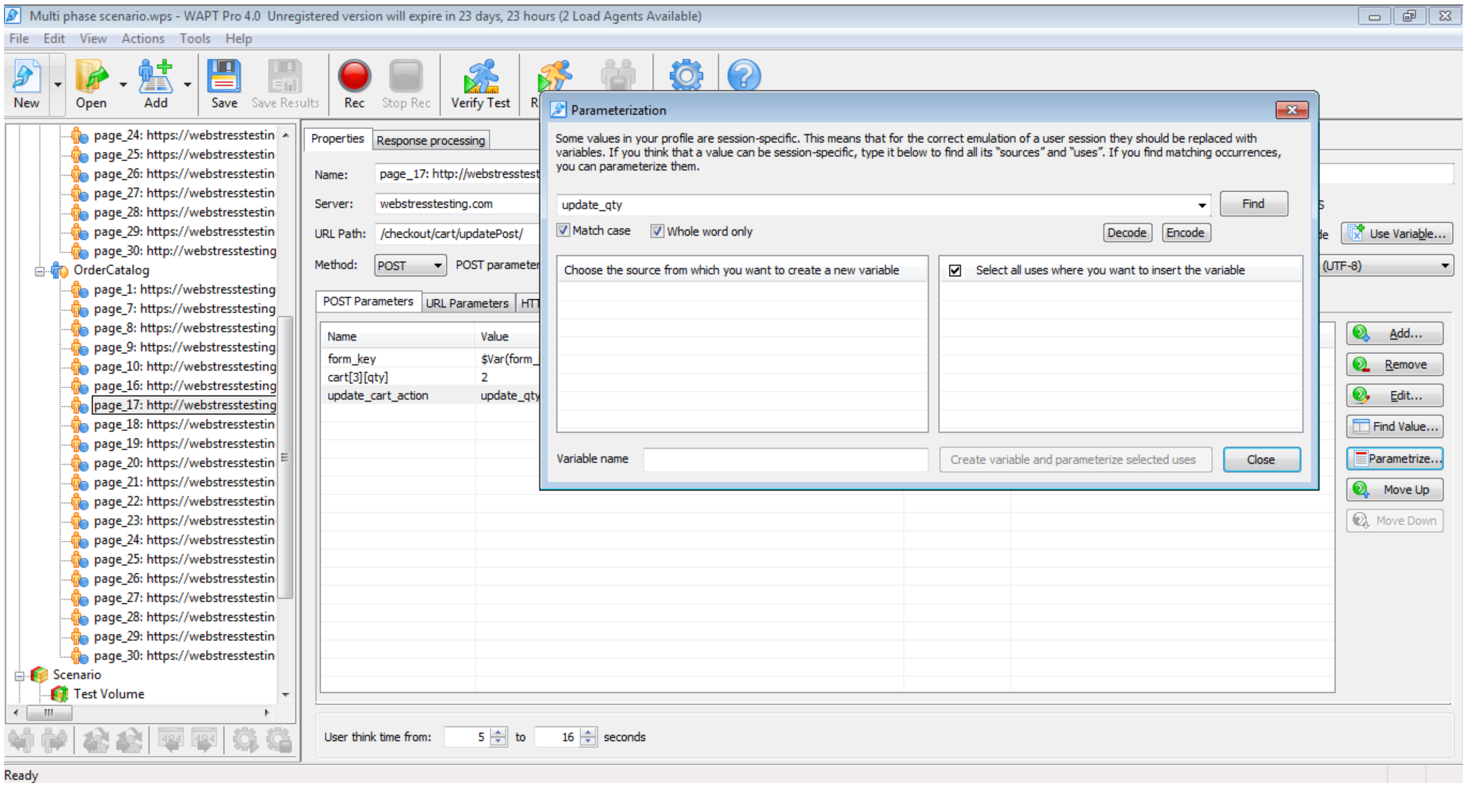
$LOrder(users,user1@gmail.com,user2@gmail.

OK Cancel Add... Remove Edit... Move Up Move Down

3. Correlation

Correlation, which is the most time consuming part for most of the tools ***is just a two-step process in WAPT***. Bingo!

The values posted with the request are listed in the POST parameters and you can make use of parameterise button which will open a new window where you can select the source value to create a variable and the requests where to replace the selected value with the created variable.



Multi phase scenario.wps - WAPT Pro 4.0 Unregistered version will expire in 23 days, 23 hours (2 Load Agents Available)
File Edit View Actions Tools Help
New Open Add Save Save Results Rec Stop Rec Verify Test
Parameterization
Find
Decode Encode
Choose the source from which you want to create a new variable
Select all uses where you want to insert the variable
Create variable and parameterize selected uses
Close
Use Variable...
Add... Remove Edit... Find Value... Parametrize... Move Up Move Down
Properties Response processing
POST Parameters URL Parameters
Scenario
Test Volume
Ready

Some values in your profile are session-specific. This means that for the correct emulation of a user session they should be replaced with variables. If you think that a value can be session-specific, type it below to find all its "sources" and "uses". If you find matching occurrences, you can parameterize them.

update_qty

☑ Match case ☑ Whole word only

Variable name

Name: page_17: http://webstresstest

Server: webstresstesting.com

URL Path: /checkout/cart/updatePost/

Method: POST

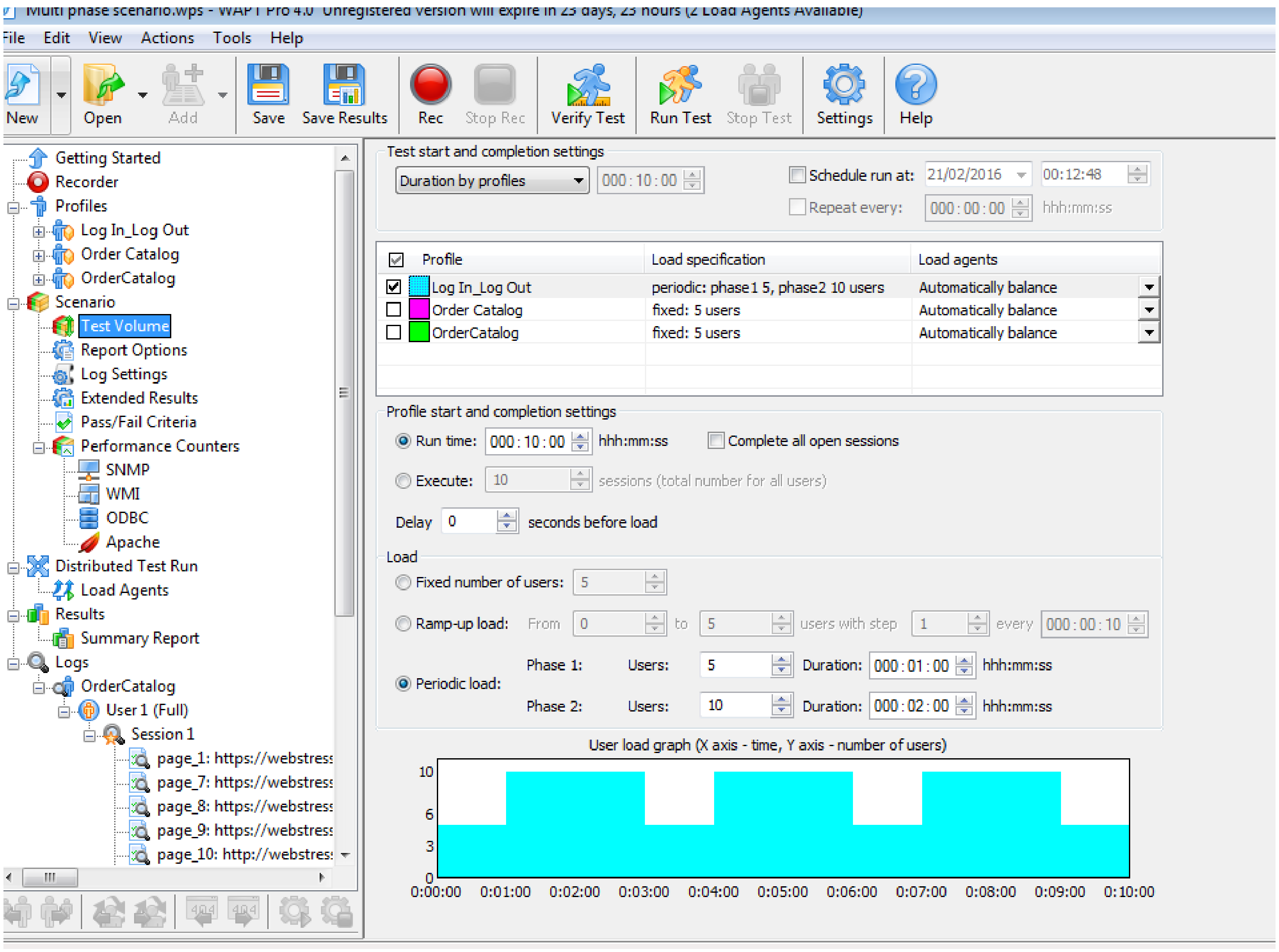
Name	Value
form_key	$Var(form_
cart[3][qty]	2
update_cart_action	update_qty

User think time from: 5 to 16 seconds

4. Creating User Scenario

WAPT has three types of load options that help to make the scenario as similar as possible to the activity of real users.

The option for ***multiple phases of load is one of the outstanding features*** as this enables you to ramp up the load for a certain period of time and then ramp down, ramp up again after an interval. This is useful for endurance testing when you need to check that the server resources can be allocated and freed several times without memory leaks and performance degradation.



Multi phase scenario.wps - WAPT Pro 4.0 Unregistered version will expire in 23 days, 23 hours (2 Load Agents Available)

Test start and completion settings

Duration by profiles 000 : 10 : 00

☐ Schedule run at: 21/02/2016 00:12:48

☐ Repeat every: 000 : 00 : 00 hhh:mm:ss

✓	Profile	Load specification	Load agents
✓	Log In_Log Out	periodic: phase1 5, phase2 10 users	Automatically balance
☐	Order Catalog	fixed: 5 users	Automatically balance
☐	OrderCatalog	fixed: 5 users	Automatically balance

Profile start and completion settings

[● Run time:] 000 : 10 : 00 hhh:mm:ss ☐ Complete all open sessions

○ Execute: 10 sessions (total number for all users)

Delay 0 seconds before load

Load

○ Fixed number of users: 5

○ Ramp-up load: From 0 to 5 users with step 1 every 000 : 00 : 10

[● Periodic load:] Phase 1: Users: 5 Duration: 000 : 01 : 00 hhh:mm:ss
Phase 2: Users: 10 Duration: 000 : 02 : 00 hhh:mm:ss

5. Executing the scripts

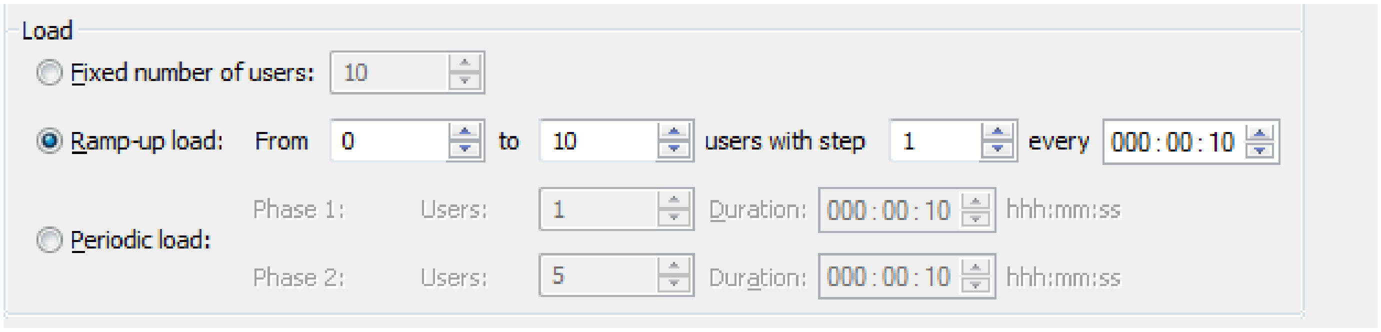
WAPT offers executing the scenarios from different load agents that can be distributed across the local and remote networks.

One of the key features of WAPT claims to be a 64 bit load engine which enables you to generate a load up to 10,000 virtual users (of course the number varies according to the weight of the webpages in the scenario and other factors such as extended test log and statistics).

However, some test options like full logging and saving extended statistics may also affect the engine performance. More complex test sessions with a big number of session-specific parameters consume more CPU time per user as well. So, I recommend to keep this in mind when designing the tests and planning the environment.

WAPT allows you to choose from three different test execution Load models

1. Fixed No of users
2. Ramp up load
3. Periodic load

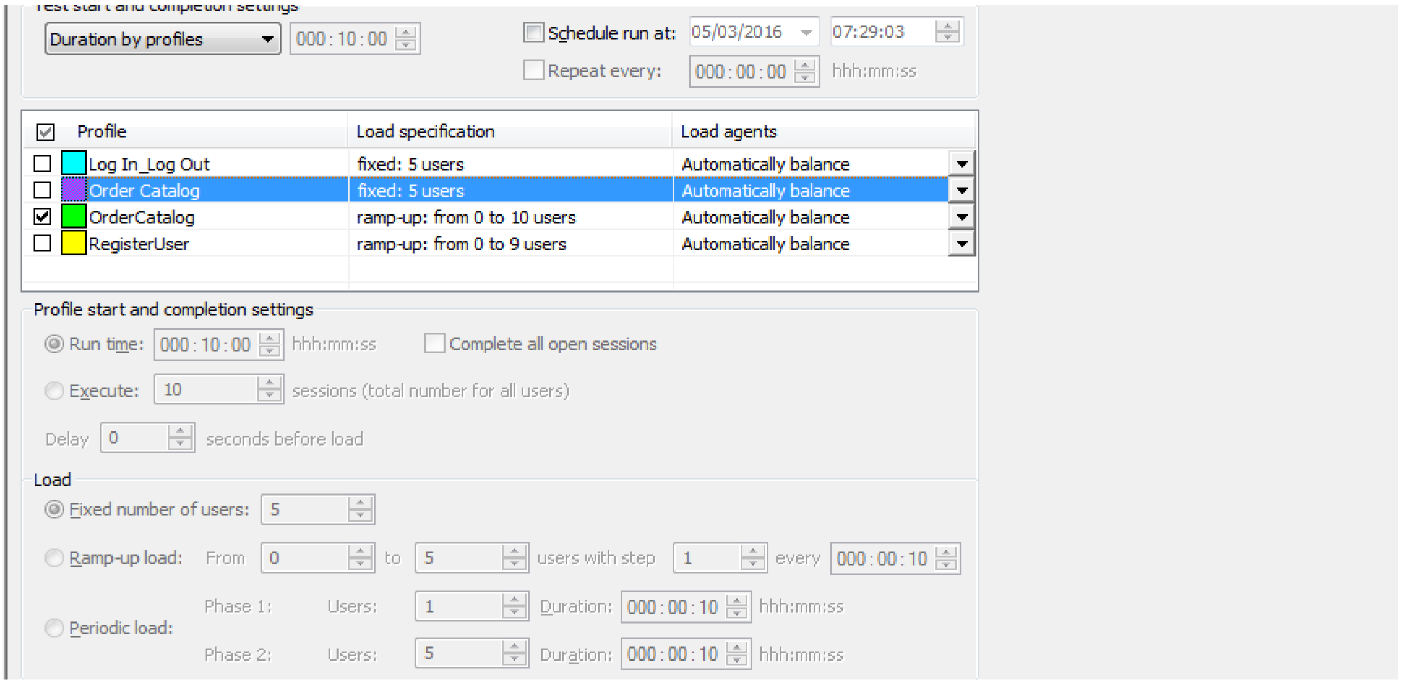

Load

( ) Fixed number of users: 10

(●) Ramp-up load: From 0 to 10 users with step 1 every 000 : 00 : 10

( ) Periodic load:
Phase 1: Users: 1 Duration: 000 : 00 : 10 hhh:mm:ss
Phase 2: Users: 5 Duration: 000 : 00 : 10 hhh:mm:ss

Also, WAPT allows you to configure the load profile for each script and allows executing them from different load agents. This gives you the freedom to load the scenarios in multiple phases.


Test start and completion settings

Duration by profiles 000 : 10 : 00

☐ Schedule run at: 05/03/2016 07:29:03

☐ Repeat every: 000 : 00 : 00 hhh:mm:ss

☑		Profile	Load specification	Load agents
☐	🟦	Log In_Log Out	fixed: 5 users	Automatically balance
☐	🟪	Order Catalog	fixed: 5 users	Automatically balance
☑	🟩	OrderCatalog	ramp-up: from 0 to 10 users	Automatically balance
☐	🟨	RegisterUser	ramp-up: from 0 to 9 users	Automatically balance

Profile start and completion settings

(●) Run time: 000 : 10 : 00 hhh:mm:ss ☐ Complete all open sessions

( ) Execute: 10 sessions (total number for all users)

Delay 0 seconds before load

Load

(●) Fixed number of users: 5

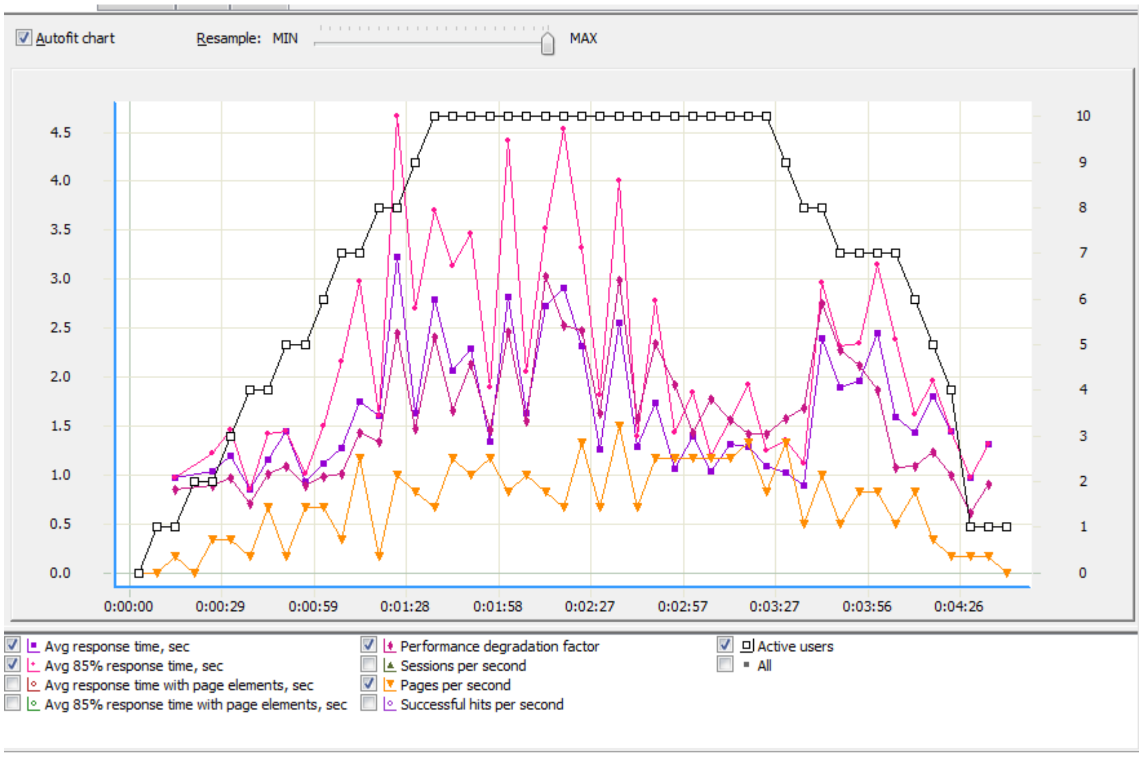
( ) Ramp-up load: From 0 to 5 users with step 1 every 000 : 00 : 10

( ) Periodic load:
Phase 1: Users: 1 Duration: 000 : 00 : 10 hhh:mm:ss
Phase 2: Users: 5 Duration: 000 : 00 : 10 hhh:mm:ss

6. Test Reports

WAPT collects all possible data and produces a very useful performance test report.

1. The report is simple, easy to read and understand.
2. Produces response time report for each profile by dividing the results in multiple time intervals. This helps to understand how the response time affected when peak load was present.
3. Reports load agent utilization, which helps as an indicator to improve the load distribution.
4. Response time graphs for individual page/request are available. It also allows you to correlate different graphs and create new graphs for analysis. For example, if you want to analyse the response time pattern against the active users then it can be done easily by selecting/de selecting the available counters as shown below.



[✓] Autofit chart

[✓] Avg response time, sec
[✓] Avg 85% response time, sec
[ ] Avg response time with page elements, sec
[ ] Avg 85% response time with page elements, sec
[✓] Performance degradation factor
[ ] Sessions per second
[✓] Pages per second
[ ] Successful hits per second
[✓] Active users
[ ] All

5. An additional feature, you can edit the JavaScript code automatically created for report analysis as per your convenience. Now the tool automatically divides the test duration into multiple equal periods and analyses the test result. This can be edited as required.

Some additional modules supported by WAPT that might interest you:

If your website contains specific components like JSON or ASP.net, then you can improve the accuracy of the testing by downloading below additional modules.

- Module for ASP.net testing
- Module for JSON format
- Module for Adobe Flash testing
- Module for Silverlight testing
- Module for GWT testing
- Module for binary formats

What I liked most in WAPT?

- The 64 bit load engine option (available only for the Pro version)
  Usually it requires a lot of investment for load generators, but by introducing 64 bit load engine, WAPT reduces the investment of additional resources for load generation. This is said to be supporting 10,000 virtual users concurrently.
- Additional programming elements like if-then-else, while loop, etc. help to improve the user actions and to create real world work load model. And that in turn allows you to create more robust and dynamic user profiles.
- Very light weight and user friendly GUI.
- Automated correlation of session variables and less programming makes life easier for a performance tester.
- Additionally supported by very detailed logs and supports comparison of request and response against recorded logs.
- WAPT is comparatively less expensive than its direct competitors.

Things I would have loved to see in WAPT:

- Currently, WAPT does not allow setting up dynamic configuration script in proxy settings. Appears that WAPT blindly uses browser proxy settings, but recording fails when browser is configured to access the network using automatic configuration script.

  Test recording and test execution are two different things. You wrote about recording above and about execution below. This is misleading. You can simply say that recording though an automatically configured proxy does not work **SOMETIMES**. It works in most cases. As for the test execution, it cannot be configured to use automatic proxy configuration for the reason outlined in my previous comment. But it can be done through a directly specified proxy.

  It would have been a nice feature but it is equally important to note that executing proxy scripts usually takes 10 to 1000 milliseconds. Doing this for each of 10,000 users concurrently would overload the system quite fast. So, whether you require this feature or not mostly depends on your choices and preferences.

- At the moment WAPT cannot be installed in any operating system other than Windows. It would be nice to have OS support for Linux as well, at least for the load agents/engines.

- Currently you cannot change the load options i.e. number of virtual users during the test execution. This would be a nice feature as this is more helpful when you do stress tests. I guess, WAPT team is already planning to add it in their next product update though.

Summary:

All in all I liked the WAPT tool for the kind of cool features it has and flexibility it offers to a performance tester. If some of the nice to have features that I explained above get incorporated then I'm sure this tool can become preferred tool of many. However, that's my personal feeling and other user's opinion may differ. But overall, the tool is great and I would like to use it for my requirement for now.

Based on my overall experience we (Tea-time with Testers) would rate this tool 4 out of 5.

You may want to check out more about WAPT on their website - http://www.loadtestingtool.com/ and here is their Quick Start guide if you can't wait to try your hands. You can download the trial version from - http://www.loadtestingtool.com/download.shtml

Reghu RJ is a performance test engineer with high of experience in different performance testing tools such as Jmeter, IBM Rational performance tester, NeoLOAD and Load runner.

As an experienced tester and test lead Reghu performs different activities, such as requirement management, test planning, test design and uses different tools to create and execute performance tests. He is also part of Tea-time with Testers magazine's tech team.

Reghu has certification in IBM Rational performance tester and Load Runner.

Testing Debt in Agile Projects

- by Ravi Kumar BN

Introduction

Anything we put off or postpone is considered debt. Technical debt is the work that should have been done on a project but wasn't. It includes those internal things that you choose not to do now, but which will impede future development if left undone. This includes deferred refactoring.

Technical debt doesn't include deferred functionality, except possibly in edge cases where delivered functionality is "*good enough*" for the customer, but doesn't satisfy some standard (e.g., a UI element that isn't fully compliant with some UI standard).

Technical debt seems to occur on Agile teams as well as on non-Agile, Waterfall, V-Model-, whatever-teams for the development part. A disorganized project with lots of duplicate code, no separation of concerns, little accurate documentation and few automated tests is said to have accrued a lot of technical debt. Projects with a lot of technical debt are difficult to build new features for and prone to bugs that are difficult to diagnose and fix.

There are situations where debt builds from how the team handles testing, specifically for testers. Some teams are still under intense pressure to deliver on a fixed date. Regardless of the state of testing or findings from testing or test coverage, there is pressure on testers to "*say it works.*"

In an agile team, technical debt in testing builds up at two levels:

1. How scrum team handles testing: At abstract level, scrum team incurring debt
2. How tester handles testing: At granular level, tester incurring debt

This article discusses on how technical debts builds in testing and the approaches to deal with them in agile projects.

Scrum Teams Incurring Debt - Cliff

In any software development project during the planning stage each phase or team gets their allotted time. When it comes to testing reality, requirements are defined late, added late, the design was late or the code was late, testers get crunched on time so the team won't slip the schedule.

A theoretical burndown chart in an agile project has the same idea. User stories and "*user story points*" get moved from "*In Development*" to "*In Testing*" at a somewhat steady pace over time and are delivered over time – this is ideal.

The troubling phenomenon common to so many teams these days is the cliff. Testers wait and wait and, as the final days of the sprint approach, the bulk of user stories get dumped on them, with the expectation of full validation and testing as the sprint demo and review come up.

There is no way a test team can do an effective job at this point. Most teams in this situation, under pressure from product owners/customers/whomever, make up quick and dirty rules:

- The story is "*done but not tested*." (*ScrumBut*)
- Test it in the next sprint while they wait for new functionality. (*AgileFalls*)
- Break the story into 2 stories, the coding and the testing. The coding is done. (*ScrumBut and AgileFalls*)
- Say it's done and if the Product Owner finds a bug during the demo we can write a new user story on that bug. (*ScrumBut*)

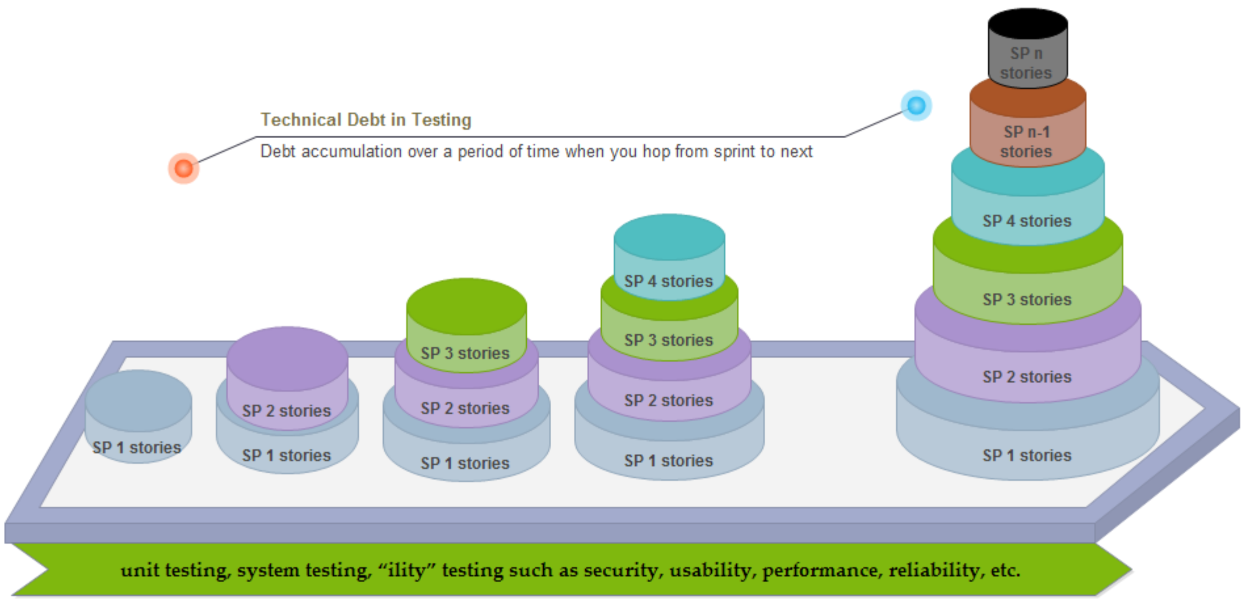
…And many more creative and flawed ways to "*count the story points for velocity*," or say it's done and build more technical debt.

There is so much wrong with these solutions, so much ScrumBut and AgileFalls combined, these situations need their own approaches on how to recognize and remediate these situations.

Tester Incurring Debts

When you are working on an agile project, (or any process using an iterative lifecycle), an interesting phenomenon occurs. Here's how it works:

- in iteration one, you test all the stories as they are developed, and are in synch with development
- in iteration two, you remain in synch testing stories, but when you integrate what has been developed in iteration one with the new code, you now have more to test than just the stories developed in that iteration
- in iteration three, you have the stories to test in that iteration, plus the integration of the features developed in iterations that came before


Technical Debt in Testing
Debt accumulation over a period of time when you hop from sprint to next
SP n stories
SP n-1 stories
SP 4 stories
SP 3 stories
SP 2 stories
SP 1 stories
unit testing, system testing, "ility" testing such as security, usability, performance, reliability, etc.

As you can see, integration testing piles up. Eventually, you have so much integration testing to do as well as story testing, you have to sacrifice one or the other because we are running out of time. To end the iteration (often two to four weeks in length) some sort of testing needs to be cut in this iteration to be looked at later.

When you explore the other kinds of testing you could be doing you will notice that it is sufficiently large list of testing (unit testing, system testing, "ility" testing such as security, usability, performance, reliability, etc.), and becomes very clear that to keep in synch with development, you consciously incur "*testing debt*".

Dealing with Testing Debt

Why do we want to test that much? We can do testing in three broad contexts: the code context (addressed through Test Driven Development - TDD), the system context and the social context. The social context is usually the domain of conventional software testers, and tends to rely on testing through a user interface. At this level, the application becomes much more complex, greater than the sum of its parts. As a result, you have a lot of opportunity for testing techniques to satisfy coverage. You can get pretty good coverage at the code level, but you end up with more test possibilities as you move towards the user interface.

Preventive/Proactive Approach:

Do the following within an iteration, alongside development:

- work as a sounding board with development on emerging designs
- help generate test ideas prior to story development (generative TDD)

- help generate test ideas during story development (elaborative TDD)
- provide initial feedback on a story under development
- test a story that has completed development
- integration test the product developed to date (possibly automated)
- harden sprints to date (possibly automated)

Of note, when you are testing alongside development, you can actually engage in more testing activities than when working in phases (or in a "testing" phase near the end). You are able to complete more testing, but that can require that you use more testers to still meet our timelines. You should always have clear Definition of Done (DoD) and be watchful on DoD adherence and raise alarm when in need.

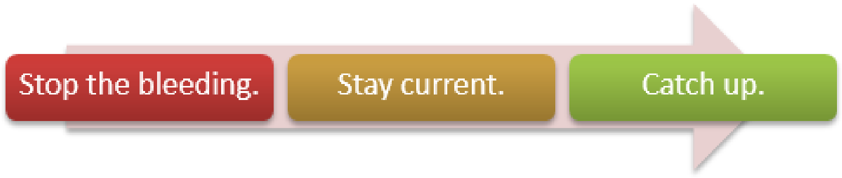
Corrective/Reactive Approach:

You normally do as much integration, system, "ility" testing as you can in each iteration, but when you are running out of time, you incur some testing debt in these areas. As the product is developed more (and there is now much more potential for testing), and as you incur more testing debt throughout a project, you should then look at some options for dealing with it.

- Leave off story testing in favor of integration testing. This is not a recommended option. Instead it is preferred to keep the feedback loop as tight as you can on testing stories that are being developed so you stay in synch with the developers. May be you should have testers in scrum team do user story testing and a separate sprint hardening team to pick up debt clearance.
- Schedule a testing iteration at the end of the development cycle to catch up on the testing debt - to do all the integration, "ility", system testing etc. to test a completed system. But this can cause a huge lag in the feedback loop.
- Bring in more testers to help address the testing debt, and bring on the maximum number you can near the end.
- Plan for risk-based testing to optimize tests based on release objectives and customer base.
- Automation your manual tests, stay current with new tests and then catch up.
-

Reduce Manual Test Technical Debt

When a team that has relied mostly or entirely on manual testing decides to adopt Scrum, it will quickly discover how hard it is to run short sprints when there's a lot of manual testing to be done each sprint. It will also realize that unless it does something drastic, the technical debt will continue to accumulate. Teams in this situation can follow a three-step process to extricate themselves from at least the worst of these problems:



The first priority of a team with technical debt in the form of an over-reliance on manual testing is to stop the bleeding, stop things from getting worse. The best tourniquet is to find ways to automate some of what is being tested manually. To mix metaphors, teams should find the low-hanging fruit: tests that will be easy to automate but save a lot of manual effort. The real low-hanging fruit is often not automating some test execution but automating other testing tasks, like populating databases or automatic navigation to the page where you'll start manual testing. You're not reducing the number of manual tests, but you're reducing the total time it takes to run them.

After the bleeding has been stopped, the situation will no longer be getting worse from sprint to sprint. Manual tests still will be added each sprint but the team is finding enough low-hanging fruit each sprint to offset the time needed to run the new manual tests. At this point, it is time to move to step two: learning to stay current. During this phase, the team focuses on learning how to write and automate tests for whatever new features are added during the sprint. While doing this, no more debt is accumulating, so the situation isn't getting any worse, but it's not yet getting any better either. Learning to add automated tests in the same sprint as the feature will be a new skill for the team. It won't be as hard to learn as the initial skills were during the first phase but will require new discipline.

Eventually the team enters the final phase, which is when it catches up on additional outstanding testing debt. It is not important how quickly the outstanding testing debt is brought down as long as it is indeed coming down. Obviously, one would prefer the debt to come down as fast as possible.

Conclusion

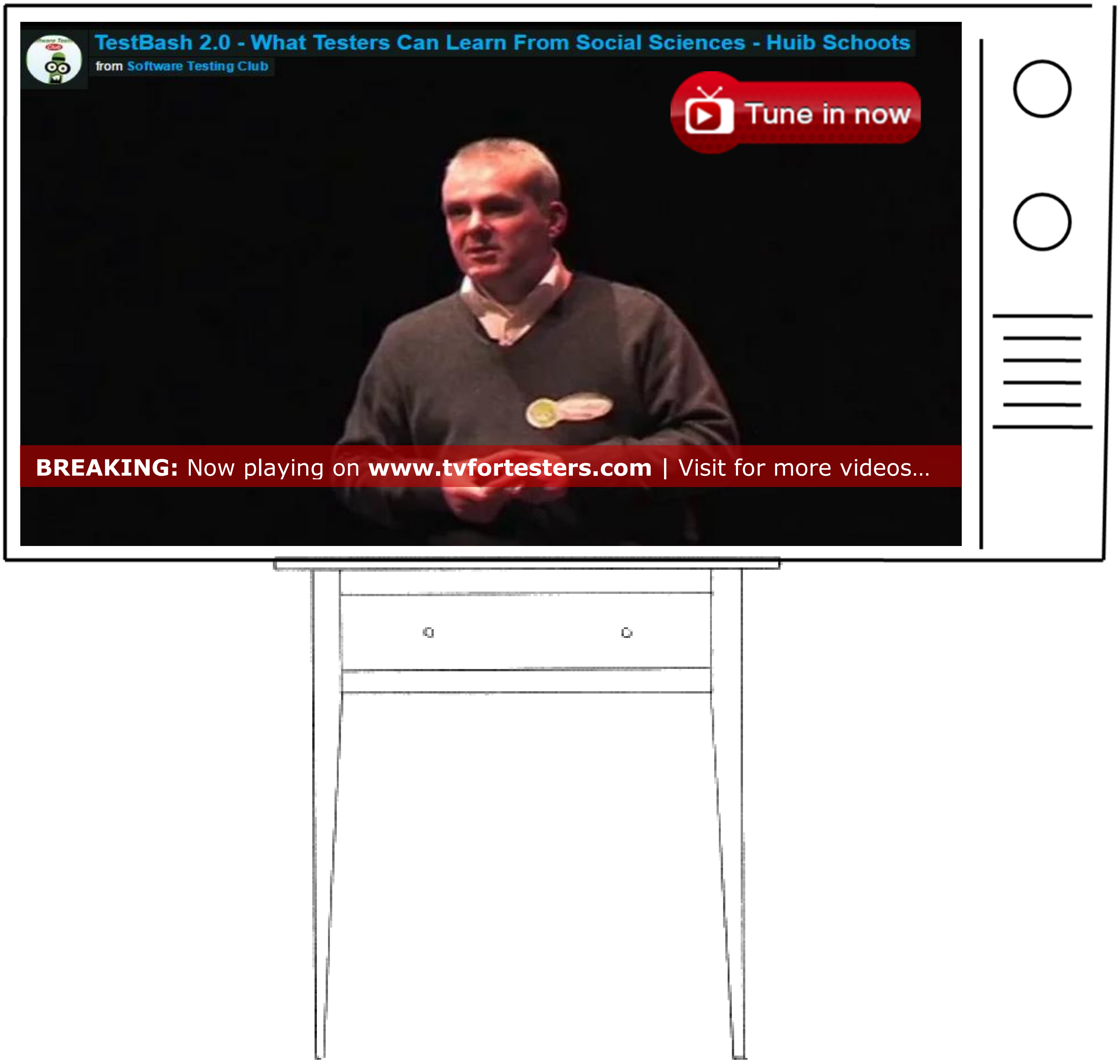
Technical Debt is inevitable but is not all bad. It is important that organizations take cognizance of technical debt and find ways and means to manage it efficiently to prevent a system collapse. It has to be communicated, managed and serviced! When teams are chronically in debt, the issues that brought it up need to be recognized, communicated, and hopefully resolved.

Test teams play a special role in certain types of debt. Test teams should be especially aware to not create more debt by compromised test automation. Test teams can greatly help the team through recognizing and communicating debt issues as they arise. They need to document intelligently, not document everything. They need to use the right approach to deal with testing debt.

Go well, not fast. Care about your code. Take the time to do things right. In software, slow and steady wins the race; and speed kills

Ravi Kumar BN is a Product Verification Manager for Imaging Clinical Applications & Solutions, HealthCare IT, Philips Innovation Center, Bangalore. He is a master's graduate from IIT Kanpur, UP, India, and BE (CSE) from SDM College of Engg & Technology, Dharwad, Karnataka, India. He has 13 years of experience in software quality processes and testing at Honeywell. He is a six sigma, lean and agile testing expert and a six sigma techniques and tools trainer. He is actively involved in building and deploying testing strategies for various platforms such as ERP (Peoplesoft), CRM (Siebel, SFDC), BI (Cognos, OBIEE), emerging technologies such as Mobility, Cloud, Analytics, Voice, Responsive Web Design and Wearables. He has attended design thinking workshop and has expertise in deploying design tools in problem solving and usability testing.

Back To Index


TestBash 2.0 - What Testers Can Learn From Social Sciences - Huib Schoots
from Software Testing Club
Tune in now
BREAKING: Now playing on www.tvfortesters.com | Visit for more videos…

TV for Testers

Your one stop shop for all software testing videos

Sharing is caring! Don't be selfish ☺

Share this issue with your friends and colleagues!

Happiness is....

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

Like our FACEBOOK page for more of such happiness

https://www.facebook.com/TtimewidTesters

SEVEN Habits of an Effective Tester

What is a habit?

According to Wikipedia, a habit is a routine of behavior that is repeated regularly and tends to occur unconsciously. In the current era of fast paced development, we have embraced light weight agile processes, rely on automation to move fast, and are very reliant on the skill of individual to do great work. Good testing habits play a significant part in ensuring that the reliance on the individual tester is not misplaced. These form the core, enabling the tester to be smart, purposeful, and efficient and deliver value. These habits enable the tester to weave the techniques, tools, process into an unconscious set of activities to become a master craftsman, who is immersive, enjoying every moment, producing outcomes that seem like works of art. So what may these habits be? Inspired by Stephen Covey, I have also listed SEVEN habits that make an effective tester.

Habit #1- Being empathetic: Thinking from consumer view

Striving continuously to be inside the head of end users and understand what they want, what they do, what they value, when they do, how much they do, how they do is vital to understanding expectations. In whatever we do, not just the act of testing, be it asking questions, writing reports, making proposals, analysing data, being outside your body and viewing what you do from the consumer of your activity enables you to become very sensitive to others' expectations.

Habit #2- Staying curious, questioning: The enquiring mind

The act of testing is not merely an activity of executing test cases to uncover defects, it is about understanding as to what can happen in different situations and therefore probing, questioning, to hypothesize, experiment, analyse. It is about being active, being alive in the process of inquiry and therefore not a monotonous act of evaluation. Tools aid the latter allowing us to stay curious.

Habit #3- Critique mentality: Looking for issues

I have a good friend to whom if you give anything, he will find issues. Be it a product, a document, a point of view, his natural instinct is to be on the opposite side and look for issues. By the way he is a very successful tester who is a leading a large team now. A mentality that is constantly critiquing anything looking for possible issues, negative thinking that is filled with positive energy to uncover issues!

Habit #4- Thinking clearly, doing diligently: Analytical and disciplined

Effective testers think! Analytical thinking, breaking down complexity, simplifying everything, applying logic, not swayed by emotions enables one to enter the unknown confidently and emerge successful. As testers, our job demands us to constantly explore the unknown, requiring us to dissect any situation logically, and solve these by doing activities in a diligent manner. Clear thinking has to be supplemented by clear execution and this is where diligence plays a significant part. The orderly execution with sharp clarity on what we are doing and why.

Habit #5- Observant, Adapting & Learning: Nimble and smart

Exploring the unknown requires one to be very observant, sensitive, to scan all information, spot anomalies, and course correct to steer towards the goal becoming smarter in the entire journey via immersive learning. Exploring unknown demands multi-dimensional knowledge beyond just product, technology, domain, into various areas like psychology, art, science, finance etc. enabling one to be creative.

Habit #6- Outcome focused, not just do activities: Deliver value

It is not doing about performing activities, staying busy, it is delivering value in what we do. It is not being focused on outcomes of ensuring what value we deliver to our consumers. A goal focused outcome enables activities that we do be sharp and purposeful and therefore the habit where we are constantly asking/questioning what we want to achieve enables better alignment of activities.

Habit #7- Persistence: Never give up

Exploring unknown territory demands persistence. Persistence in doing, persistence in questioning, seeking information, of analysing from all angles, of solving situations that arise of not being fazed by road blocks. Well brute force persistence may be meaningful always, it is necessary to understand the big picture and timeout as appropriate. Habits are formed only with practice and they result in unconscious behaviours that allows us to be extremely efficient.

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

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

He can be reached at **ash@stagsoftware.com**

Back To Index

testomaton

Quality Assurance via Automation

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

Services

Test System Analysis and work estimation

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

Tests creation and Automation

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

Regression and Test evolution

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

Architecture Consulting

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

Trainings

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

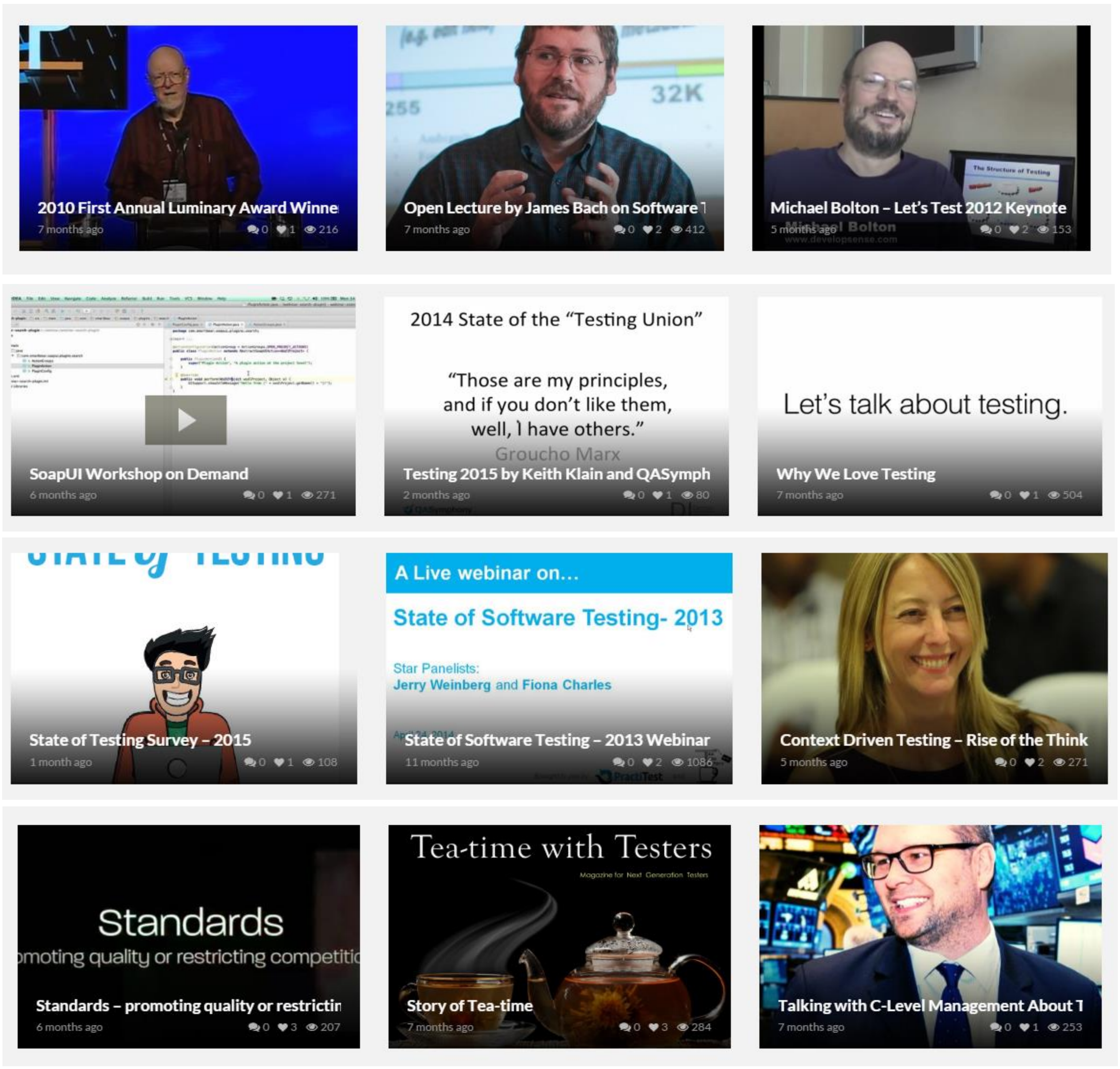
We enjoy putting our experience to your service. Our know-how allows us to provide consultation services for projects at any stage, from small up to super-large. Due to our range of expertise we can assist in software architecture and COTS selection; review of software designs; creation of testing suites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

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

Got tired of reading? No problem! Start watching awesome testing videos…

TV for Testers

Your one stop shop for all software testing videos



2010 First Annual Luminary Award Winne
7 months ago 0 1 216

Open Lecture by James Bach on Software T
7 months ago 0 2 412

Michael Bolton – Let's Test 2012 Keynote
5 months ago 0 2 153

SoapUI Workshop on Demand
6 months ago 0 1 271

Testing 2015 by Keith Klain and QASymph
2 months ago 0 1 80

Why We Love Testing
7 months ago 0 1 504

State of Testing Survey – 2015
1 month ago 0 1 108

State of Software Testing – 2013 Webinar
11 months ago 0 2 1086

Context Driven Testing – Rise of the Think
5 months ago 0 2 271

Standards – promoting quality or restrictin
6 months ago 0 3 207

Story of Tea-time
7 months ago 0 3 284

Talking with C-Level Management About T
7 months ago 0 1 253

2014 State of the "Testing Union"
"Those are my principles, and if you don't like them, well, I have others."
Groucho Marx
Let's talk about testing.
A Live webinar on…
State of Software Testing- 2013
Star Panelists: Jerry Weinberg and Fiona Charles
Standards
promoting quality or restricting competitio
Tea-time with Testers
Magazine for Next Generation Testers

WWW.TVFORTESTERS.COM

www.talesoftesting.com

What does it take to produce monthly issues of a most read testing magazine? What makes those interviews and articles a special choice of our editor? Some stories are not often talked about…otherwise….! Visit to find out about everything that makes you curious about **Tea-time with Testers!**

Advertise with us

Connect with the audience that MATTER!

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

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

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

Well, hear directly from them.

And the **Good News** is…

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

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

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

in next issue

articles by -

Jerry Weinberg

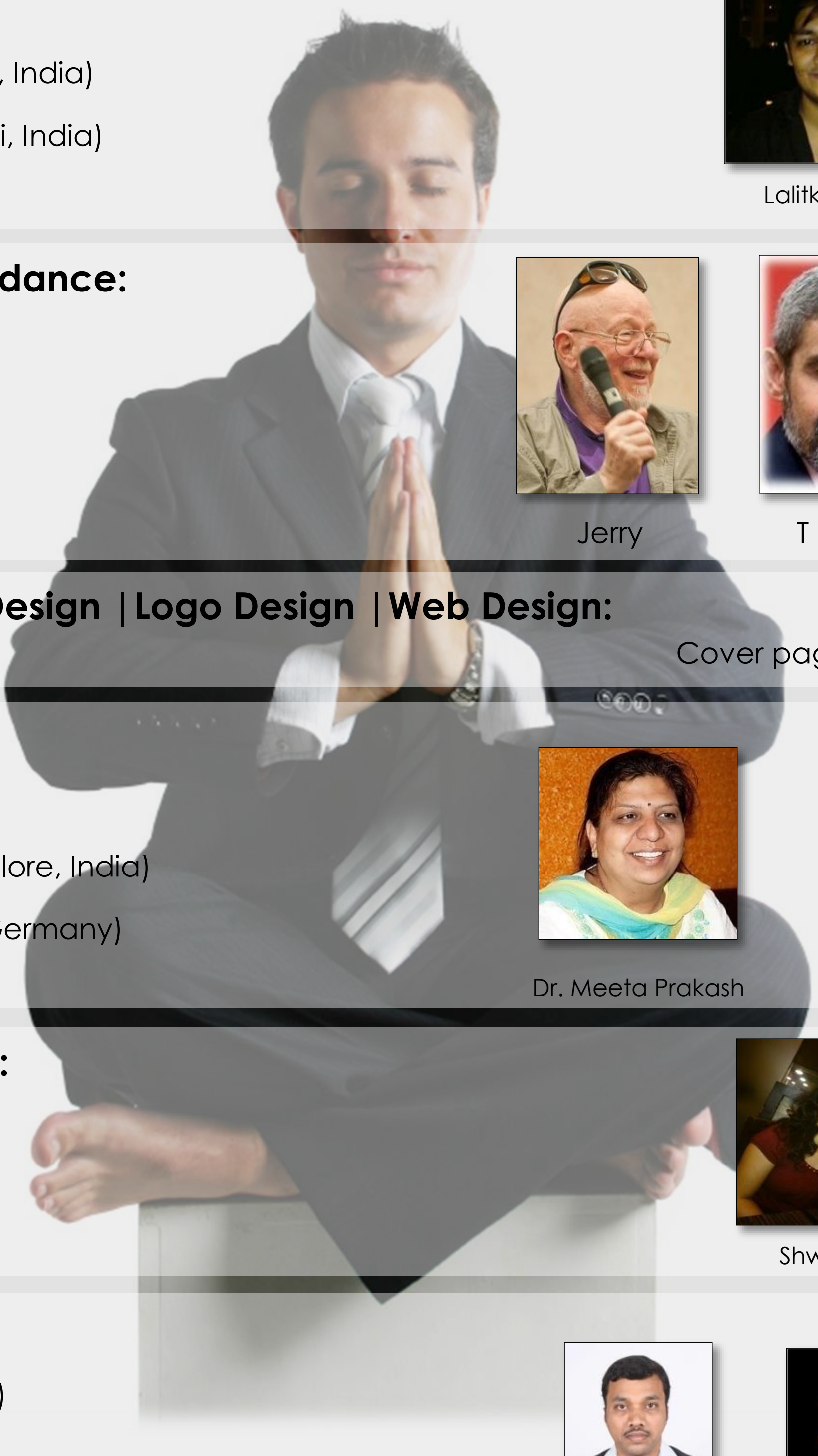
T Ashok

Sakis Ladopoulos

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

Cover page image – DIY Easter Egg

Core Team:

Dr.Meeta Prakash (Bangalore, India)

Dirk Meißner (Hamburg, Germany)

Dr. Meeta Prakash

Dirk Meißner

Online Collaboration:

Shweta Daiv (Pune, India)

Shweta

Tech -Team:

Chris Philip (Mumbai, India)

Romil Gupta (Pune, India)

Kiran kumar (Mumbai, India)

Kiran Kumar

Chris

Romil

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

To get a **FREE** copy,

Subscribe to mailing list.

SUBSCRIBE

Join our community on

facebook

Follow us on - @TtimewidTesters

www.teatimewithtesters.com

Give Feedback