Testing Circus

Volume 6 - Edition 9 - September 2015

Interview with Albert Gareev

Magazine for Software Testers

www.TestingCircus.com

Testing Circus

September 2010 - September 2015

5 Years X 12 Editions = 60th Edition

www.TalentPlusPlus.com

Kick Start Your Career

Talent++

www.TalentPlusPlus.com

Confused with your career decision?

We can HELP

What is where?

Testing Circus

Volume 6 - Edition 9 - September 2015

Testing Circus Team

Founder & Editor – Ajoy Kumar Singha

Team -

- Srinivas Kadiyala
- Dwarika Dhish Mishra
- Pankaj Sharma
- Jaijeet Pandey

Edition Number : 60 (since September 2010)

Editorial Enquiries: team@testingcircus.com
Ads and Promotions: ads@testingcircus.com
Article Submision: article@testingcircus.com

Testing Circus India
Chaturbhuj Niwas, 1st Floor,
Sector 17C, Shukrali,
Gurgaon - 122001
India.

*On the Cover Page - Albert Gareev

From the Keyboard of Editor

Wow! This is the 60th edition of this magazine and marks the completion of 5 years of this little magazine. A very happy and exciting moment for all of us who volunteer for this magazine. My heartiest thanks to Srinivas, Sanath, Dwarika, Pankaj, Jaijeet and numerous others who have helped us in the past. Without your effort Testing Circus would have closed its door long ago.

We may not have crossed Malcolm Gladwell's 10,000 hours rule of success but 5 years is a long journey. This was only possible due to the contributions by our authors around the world. We are also thankful to our advertisers who help us run this tiny magazine.

In this edition, you will love the articles from prominent testers like Pradeep, Geordie and Karen. You will also enjoy Albert's interview and other regular columns.

There is a testing workshop and conference 'TestBash' happening in New York in the first week of November. TestBash NYC has an impressive line of speakers. Testing Circus is a media sponsor to the event. More details available inside this edition.

More news next edition. Till then Happy Testing!

- Ajoy Kumar Singha
@TestingCircus // @AjoySingha

Feedback please! team@testingcircus.com

5 Years of Being Helpful

- Geordie Keitt

Five years ago, when Testing Circus began, it was September 2010. I was working for ProChain Solutions and was wrestling with a difficult problem of usability.

Microsoft Project did not have any tools for copying work on the Pert chart, the 2-d task layout view, and the consultants were having to copy all the tasks representing one stream of work by hand and arrange them similarly so as to represent a similar stream of work by another team. They did this on their own time, often staying up to insane hours so as not to use precious time in front of the client. They never complained, but I knew they were burning out. I had to do something. The only problem was that it was not my job.

A few months earlier I had accompanied a consultant on a large-scale project planning session, where I saw firsthand as these issues arose in the room. Each session cost nearly 40 highly skilled workers many hours out of their day. The session leader, Norman, guided them through the process of defining the desired end state of the project, back through each pre-requisite deliverable or activity, to the point where they were able to begin the project. He guided them to assign resources to each item, to connect the item with other items in logical relationships, and to create aggressive estimates of time to completion. At every stage, the group found other parallel chains of work, similar but not identical, and suggested copying the first chain to make another, and do some simple find / replace on the text to correctly name the new sequence. Norman would patiently explain that Microsoft Project did not have that capability, and it would be more trouble than it was worth to do it in front of them, he had duly noted the duplication, and would update the schedule offline. This seemed to satisfy the team, but I noted the look of fatigue on Norman's face as he said this, and at the next break I asked him what was up.

"To take care of this sort of thing keeps me up until 4AM most nights on the road. We have no tools to copy and paste segments of networks, to merge networks from multiple files while keeping the Pert arrangements, or to send segments of a network to other files. We do all of it by hand, at night, without the client around to tell us if we've done it right."

So I did a subversive thing. I stopped documenting the network build session, which was my charter, and instead started trying to solve a problem for a suffering friend.

I opened up Microsoft Project's Visual Basic window, and started figuring out how to copy tasks. First I figured out how to tell if a task was selected. It turns out that the Project object contains an array of selected tasks. Then I figured out how to make copies of all the selected tasks – you have to loop through the array of selected tasks and make new tasks containing copies of all the attributes of selected tasks.

When I went to the Pert to find the new tasks, though, I didn't see where they were. I saw them in the Gantt, no problem, so I knew they had to exist. I finally found them in the top left corner of the Pert chart – at X = 0, Y = 0. That didn't do me any good, so I tried to copy the X, Y coordinates of the original tasks to the new tasks. Turns out, you can't use the X or Y coordinate settings of a task on both sides of the assignment operator – nice arbitrary rule, Microsoft Visual Basic team! – so I had to copy the data to a number field in the task, then retrieve it from the number field and assign it to the X and Y coordinates.

Again – when I went to the Pert chart, I didn't see where they were! They certainly weren't in the top left corner anymore. But then I realized, they were right there in plain sight, like the purloined letter of Sherlock Holmes. The copies lay exactly on top of the originals. I went in

and added 5 pixels to the X coordinate and subtracted 5 pixels from the Y coordinate, and there they were – all the tasks, offset slightly right and down from the original. All I had to do then was establish the links between them, and I was done, at least for the start.

At the next break I showed Norman my new utility, and gave him a copy.

Someone had actually listened to him. He didn't know what to say.

The next morning Norman thanked me and said that he had gone to bed before midnight during a client visit for the first time ever.

Next of course he hit me up for a raft of upgrades: text replacement, choice of whether or not to include links to predecessors and successors of the selected group, the ability to copy to another file, and of course a button to make it happen. So I spent the day doing that.

He got to sleep before midnight again.

The next day he started using the capability in front of the client. One of them noticed the new capability and asked for a copy of the macros. Norman obliged.

I had delivered production software.

By September all the consultants were using the macros. They had grown into a small suite of tools for manipulating the Pert chart: selecting sets of tasks, inverting selections, hiding all unselected tasks, hiding selected tasks, copying them with various degrees of fidelity, etc. It was now a part of our packet for certification candidates, internal consultants who received extra training on our tools, It even had a name: ProChain Expert Tools.

It was time to start trying to take it to the next level: making an actual add-in out of the macros, with an installer, and integrating it visually into our other offerings. Or even more ambitious, offering it to Microsoft so that they could make these capabilities part of the product.

I wrote up a grand plan for the whiz-bang version of the tools, something that would turn a client's knees to jelly, make them putty in our hands: auto-spacing after adding a new node, anti-paging for plotting on giant rolls of paper, eliminating all the tedious intermediate steps and workarounds that I had coded in my jejune exuberance to make the tools clean and nice to use. I gave the plan to the consultants, who loved the ideas and wanted to get them. I gave the plan to our software development leads, who prioritized it right above fixing the bad error messages put out by Project 2000 when run on Windows ME. I gave the plan to Microsoft, and never heard back.

But you know what? I had coded production software! I'm a tester. We don't usually get to do that. But I followed my instincts, followed my heart, and used code as a tool to eliminate a little suffering. I'm so happy that I did that.

Since then, I've done a lot more testing, but also some more coding. I have produced test frameworks and illuminative scripts of every stripe. Recently I learned enough Python to work on an incredibly interesting AOP project that writes unit test scripts on the fly while software is running (WAT? See my blog on DoranJones.com). And I use writing, presenting, and speaking to share a little knowledge, a little passion, a little joy at the opportunities to help others however I can.

I've worked with Ajoy, the ringleader of this crazy little Circus, and he has the kind of heart I recognize. His knowledge, passion, and joy come through with every article, every tweet, and every issue.

May your testing journey be as long and as awesome as the journey of this little magazine. Happy testing!

About the Author

Geordie Keitt is the Director of Testing, Doran Jones, Inc. He fell into software testing like so many others, by accident. Without a plan or a mentor he skidded around the industry for a few years and made plenty of mistakes. Luckily he met James Bach at a conference in 2000, where James gave a polarizing keynote "The Future of Testing Looks Exactly Like the Present" about how there is no progress in testing, only a very few think innovatively, and the ones who do will pick the meat from everyone else's carcasses one day. Geordie was in the 10% who responded positively - he had found a kindred mind and became a part of the CDT community. He tweets at @geordiekeitt.

Reach out to a larger number of audience.

Write for Our Readers.

Looking for a theme to write?

We have some ideas for you.

Click Here to get all ideas!

www.TestingCircus.com

Where is the Uber for Testing?

- Pradeep Soundararajan

10 years ago, I started answering questions from testers posted on discussion forums in Orkut. Over these 10 years, I see that the questions haven't changed except that Orkut has vanished and some other forums have come up. This is a sign of how our industry has not progressed. There have been changes but not disruption. For instance, I know the role of test manager has vanished in many organizations. Agile disrupted traditional roles in testing (and created new problems). Well, Agile didn't create those problems. Putting people to understand and implement Agile did.

By disruption, I mean, behavior change resulting in more efficient methods /that has scaled/ to impact every part of the world. Disruption can be dangerous because it brings in a sudden change and people have less time to adapt. Uber is a disruption and there are countries across the world where taxi drivers and their unions have protested against it. Where is the Uber for testing?

I thought I was going to disrupt the world with Moolya in 5 years when I started it. I almost began doing that. 3 years on it, I ended up getting sucked into things that are my own karma and right now I am not disrupting anything. I am, at best, trying to do better testing than most other services companies, which isn't a big deal. I realized, it takes too many iterations before I can disrupt. All these years, going through the iterations was very painful and made me go in quest of *"Is there a non-painful way of becoming successful at disrupting?"* The answer is a No and hence Yes. I am on a transition from No to Yes.

The last 5 years of mine is a huge opportunity lost. I am far ahead of where I started from and far away from where I could have been. The knowledge that I know what got me far and what didn't get me far enough is the hope in which I live today to have a chance at disruption.

Non testers disrupting the test space

It is my pleasure to write about Jay Srinivasan and Manish Lachwani who built App Purify that got acquired by Google. Their names are unheard by the testing communities I am involved in. Google took their mobile app test platforms and launched the Mobile Test Cloud which is in Beta now. Both Jay and Manish didn't come from a testing background but saw an opportunity to solve a testing problem and executed it well. Now, their work is reaching millions of app developers through their Google acquisition.

I also want to introduce you to Rentention.ai (earlier known as Beta Glide) founded by fresh college grads from IIT in India. They were working on a start-up that focused on Android Fragmentation testing. They appear to have pivoted now to do what they call as, uninstall intelligence. They received funding and are on course to become the leader in understanding why people uninstall apps - which is pretty useful information to testing and especially to understanding user experience. Someday they will be acquired by a large organization and hence will scale globally.

Testers disrupting the test space through entrepreneurship

I don't see a Uber for testing yet from testers turned entrepreneurs. I see there are people who are getting closer to it though. And that is what people need to do. Get closer to the cliff with the ability to fly and the wind will push them and they will fly.

I am hoping Rahul Verma's Unitee framework would disrupt and be widely adopted and used. From the conversations I have had with Rahul, it sounds disrupting.

Similarly, what Keith Klain, Doran Jones has done with Per Scholas is also a tipping point for people in the US to think how a great opportunity existed in a place like Bronx and they didn't notice it. When Per Scholas, Keith Klain and Doran Jones take this beyond USA, they are definitely making a global one. They are solving a problem of employability and making better testers than what industry produced.

There are people like Henrik Andersson and some of his HoT team members at House of Testing who have put themselves to get closer to disrupting. Based on the conversations I had with Henke in Copenhagen this year at that Japanese restaurant whose name I don't remember as I write this, I see his thinking patterns moving towards the end of the cliff with his wings already strong.

What makes Keith, Rahul and Henrik closer to disruption?

They failed. They continued. They iterate. They continue.

Keith did a mini disruption at Barclays before he decided to do it outside of it. I have personal record of seeing it on the floor. Keith was putting himself in position that others don't put themselves in. By that I mean, he was clearly risking his job. He was bold to do that because he wanted to disrupt. Now, he must have failed and iterated several times before he could do what he has done at Barclays and what he is doing at Doran Jones. The wind will do its job.

Rahul Verma and I have collaborated, fought and settled our wars over a beer at the Prost Pub and now we have a common favorite beer. So, I know him well enough. Rahul is successful with Test Mile. By that I mean, he has learnt what it means to disrupt. That is why, he started to focus his energies on building Unitee. With Unitee he hopes he will bridge the gaps that have kept open by the world for long time.

Henrik has maintained a culture in House of Test. Out of everybody else in the world, I know for sure what it means to do that. I am not as successful as he is in it. His position is great and with degrees of shift, House of Testing, is just what every country needs.

Others?

Oh, there are people like Gojko Adzic. I wouldn't call him a tester because he is sort of everywhere in the cycle of software development. I had an epic argument with him at one of the Oredev conferences where IMHO he said something that appeared to violate my basic understanding of testing and then we haven't had a chance to speak after that. However, over the years, he did things like Mindmup and Bug Magnet Chrome extension that makes perfect sense so I don't care what he thinks of testing as long as he does great tools. If he brings in a wider business layer to what he does he too could bring the Uber of testing.

There is plenty of Intellectual Property lying out there in documents and blog posts that haven't seen the light it needs. My friend Shmuel Gershon almost got there. He developed a session note taking tool and then he didn't venture it as a business and make an Uber out of it. He didn't.

If somebody took one small thing out of a material like Rapid Software Testing or from speaking to smart people like Paul Carvalho, Ben Simo, My Grandpa Robert Sabourin, Julian Harty, Alan Richardson and converted that to a tool that can disrupt, I would rejoice. It just doesn't happen.

Our chances are slim and grim

Now, given the options we have to see a Uber for testing and comparing it to the population of testers in the world, it looks like we are talking about a fraction in a million chance of seeing Uber for testing in this life of ours. That is because, testers don't want to turn entrepreneurs. I have done independent consulting. That is not what I am talking about. The ones that are

turning entrepreneurs are lacking the vision. The ones that have the vision don't have the right teams. The ones that have the right teams have baggage of their own to clear.

It takes 5 years from today to get closer to the cliff with strong wing power with the hope that wind will push. 5 years. That is how far I am. In the interest of the community, I hope others get there before me.

About the Author

To those who have never published an article. Bio section is what the authors write about themselves. To those who have read the article there is a little bit of my bio inside this particular one. If you want to continue reading what I write - look for my Linkedin and Quora.

Editor's note: Pradeep is the Co-founder at Appachhi & Founder Managing Director at Moolya Testing Pvt. Ltd.

Congratulations to Testing Circus on its fifth birthday! Long may the show go on!

- Michael Bolton

IT'S HERE

TESTBASH NYC

I LOVE TESTBASH PRECISELY BECAUSE IT OFFERS A MORE INTIMATE AND DIRECT INTERACTION WITH MY TESTING COLLEAGUES.
- MARK TOMLINSON, PERFBYTES

TESTBASH

On November 5-6th 2015 we will gather to hear, learn from, meet and speak to other people who love software testing.

TestBash is a conference dedicated to the art of software testing. We are an open and friendly community of people who are dedicated to improve the world of software testing. We believe change will happen through learning opportunities and through making friends.

We are a friendly bunch. Come join us. You'll soon feel at home.

Click here to know more about the Workshops and Conference Talks

***Testing Circus** is a Media Sponsor to Test Bash NYC 2015

How to Test Series, Part 5

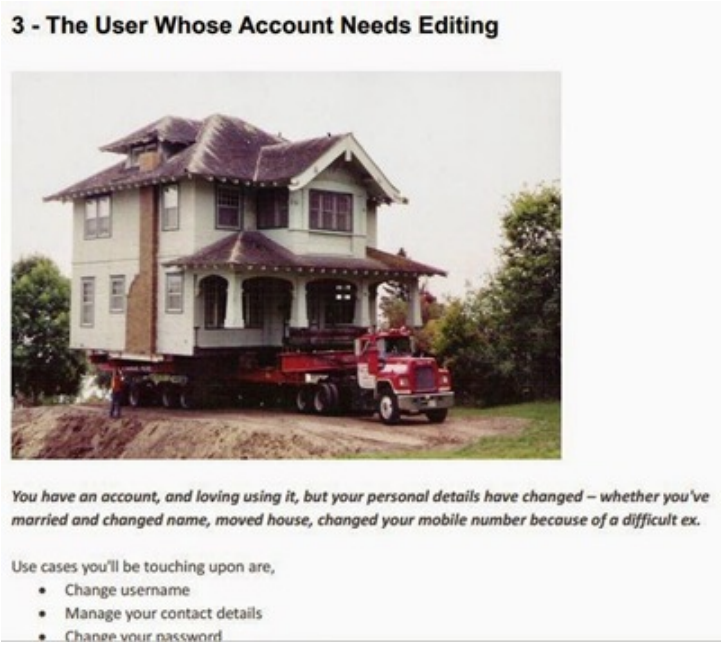
How to Test - Account Self Management

- Mike Talks

Welcome to the penultimate in "how to test" series. Today, we're going to jump in with some oracle ideas about account self-management. Note that some of these ideas as well as *"testing expectations"* could also form *"design ideas"* as well ...

Oracles

At my team we have a series of character cards and one of them is as below, "user who needs their account editing" ...



This is a range of functions which allow you to change some of the details of your account. The bottom line for a company, it really helps if certain very simple functions a user can modify for themselves over ringing a support line for. It's easier for them, and means you're not wasting money on call centre staff to do basic jobs.

If your idea of service is making a customer wait 30 minutes on the phone whilst listening to Cat Stevens for a password reset, and more than likely the end result is your customer deciding they don't really need your system that badly.

Account Hijacking Danger!

There are some account details which if changed can lead to account hijacking. This is where if someone uses a machine in at an internet cafe after you and you're still logged in, they could potentially *"hijack"* your account. We'll talk a bit about those when we encounter areas where it could be dangerous.

Obviously a lot of the below depend on context of *"what service you are delivering"* and the level of rigour you need around it. Let's look at some details you might want to change ...

Changing Name

People change name a lot. The most obvious one is after marriage when it's tradition the bride change her surname to the groom's.

But there can be other reasons as well - for instance I have to admit not being too in awe of the surname *"Talks"*, as my school life pretty much consisted of every teacher saying at the start of the year, *"Michael Talks ... I hope he doesn't"*. I thought I had it bad until at University I met a guy called Nicholas Lunt who'd wanted to be a teacher, but decided otherwise *"because of what my name rhymes with"*.

Now if you're Twitter or Facebook or any social media, this should be a fairly easy thing to do. However I do know some social media make have safeguards - you can't for instance change your name to the name of

someone famous like *"Kate Middleton"*, *"Arnold Schwarzenegger" or "Donald Trump"* without flashing some ID to prove that's really your name. Facebook also has an issue with the surname *"Talks"*, which it thinks is so rare *"it's just a joke"*. Meh, thanks. So my son had issues creating an account in his real name – in the end he created one called "Zulu Warrior", which it allowed. The difference for this would be if you have an online bank. For that given the ability for fraud by just changing your name, you'd want some more rigour and a *"come in and show proof of name change"*.

Changing Date of Birth

As far as I know, there is no way you'd ever want to change your date of birth. Okay maybe you'd want to younger, but there's no legal reason. If your system offers you to, then this is a bit of a no-no. Sure admin might need to correct, but the user?

Changing Gender

We can get a bit schoolboy giggly about this. But having experienced the other side of this, and the difficulty of changing gender through friends like Violet, having the world recognise your new gender if different from birth gender is a big deal. People who do need to be treated with respect and compassion, and not the butt of a joke. The laws in the UK and NZ recognise the right of people to legally change their gender to that which they associate with, regardless of birth gender, so typically your system needs to as well.

Changing Password (Hijacking Danger)

Yes, it makes sense to change passwords occasionally. But to avoid the hijack scenario, it's generally good to ask for the old password first to prevent anyone from doing it. It also makes sense to send an email/text to say the password has changed *(but obviously NOT what the new password is)*, just in case you go *"wait a minute, I did not change the password on this!"*

Changing Email Address / Phone Number (Hijacking Danger)

Your email address and mobile phone number are typically used in *"forgotten password"* scenarios where you say you've forgotten your password, and a temporary one is emailed to you. If an unscrupulous party sets the email to one they control, then they're just a step away from hijacking your account.

Hence it makes sense as for password to have this change password protected, but also for there to be a *"changed email address"* notification sent to the OLD email address. And similar for the mobile phone number.

Hopefully I don't have to explain why it needs to be the old email address to you - if not, have a good think about it!

Similar logic to this applies to changing your mail address.

Heuristics And Test Ideas

This is the last in a series looking at oracles, heuristics and test ideas. Our first exercise quite thoroughly set out the details for testing account creation. The second exercise, regarding logon gave some more room for you to try things out yourself.

This time around, with those examples in your mind, and our expectations above clearly set out, you should be ready to fly this one solo!

Feel free to drop a line to me via **Testing Circus** editors if you want to contact me with a response. Even better, if you have an idea for how to test the account self-management, or any other common feature, write an article about it and submit it to Testing Circus.

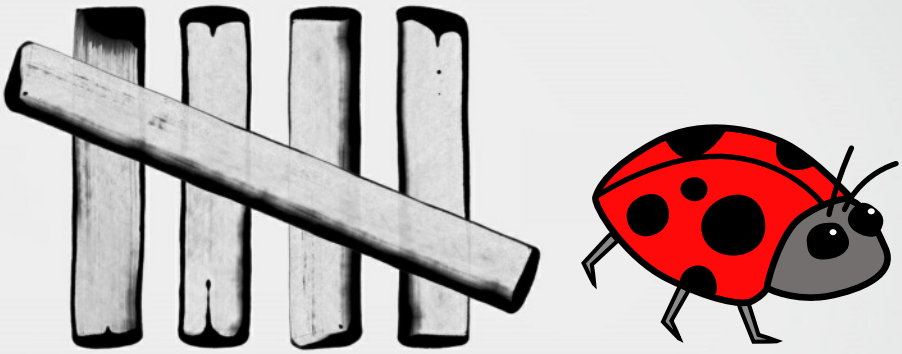
Thanks for the response I've got on this series – I've had a few people who've approached me with good things to say about the articles. Next time we'll wrap everything up by talking about how you should write up a defect when you've found one. What makes a good defect report?

A collection of this series and other testing elements is included in the book How To Test available at https://leanpub.com/howtotest

About the Author

Mike Talks is a certified teacher. Although not certified to swim, he seems to manage quite well without that certification. Which shows there's a gulf between being able to do something, and being certified to do it. He works and tests in New Zealand.

We survived 5 years because of YOU.



Thank YOU.

ALBERT GAREEV

Independent Consultant/Contractor, Practice Lead

Toronto, Canada

Interview with Testers

Albert Gareev is software testing craftsman and practice lead; Toronto based consultant and contractor. With over 20 years of diverse industry experience, Albert has performed testing for challenging projects; successfully implemented complex and large-scale automation. He is a "full stack" technical lead – he combines strategic and tactical vision, efficient team work setting, continual growth of skills and mindset of his team, and own hands-on technical skills.

Albert is founder-organizer of Toronto Testing Meetup (http://www.meetup.com/Weekend-Testing-Toronto-Meetup/), and also a facilitator of the Americas chapter of Weekend Testing.

* Interviewed by **Srinivas Kadiyala**

1. You have been in IT industry for more than 18 years. How did you start your testing career? What changes did you see in IT, especially in software testing?

I started as a software engineer. Back then, all engineers were trained to be self-sufficient. We were trained to learn through practice, invent our own solutions, use any tools and create our own from available resources. In my early career years I developed software and I tested it.

There were many changes in IT. From geek culture it practically became a mass-culture. Almost everything became more technical, even computerized, and elements of programming are now used almost everywhere.

From a single profession, software engineering spawned multiple specialties and roles. Testing is one of them, though I don't see it as an isolated profession. It will always exist as a science and skilled craft, but in different contexts it might be carried on by different IT roles, not necessarily testers by title.

I also think that Testing has reached yet another maturity plateau. Ten years ago there were just a few people who thought of the profession philosophically, questioned it, evolved it, and wrote about it. There were followers around them, recent comers to the field, who read, discussed, argued, and evolved as well. Now many of them are testing directors, managers, and leaders of the practice. I, too, belong to this wave. And, as many, having reached a degree of mastership, I wonder – what's next? Now it's my turn to take it further – how can I do that? Well… below I give some details on how I answer these questions.

2. You have given a talk on "Accessibility Testing" at CAST 2015. What made you focus your interests on research and develop methodologies for Accessibility Testing?

I gave a talk and four hours workshop on accessibility testing heuristics.

I'm an Accessibility advocate because it's part of Diversity and Equality.

Making Accessible Web and IT is important for everyone. Everyone is aging. While some are still

ALBERT GAREEV

lucky to keep fully functional vision, cognition, motorics, and hearing, most of people lose one or another with years. Or just don't have it as sharp as before – and barriers to information control and access arise. We live in information processing era. People talk of not using Facebook for a week as a "greatest challenge"... Now imagine yourself fully deprived from communication channels you're used to.

On the other hand, I'm a testing practice lead. Developing new testing methodologies is in the core of the job. I gladly took on the challenge and journey, and I was happy to share my experiences at the Conference. I'm proud to have this as one of my contributions to the craft of testing and to the community.

3. You have recently started Weekend Testing Toronto, offline and online. What topics it includes to help the local testers from Canada.

I started Toronto Testing Meetup in January, 2015. I'm not very ambitious; I merely wanted to be in a local community of testing professionals where we could share our experiences, experiment, practice, and learn. I couldn't find this kind of a testing workshop in Toronto for years. Inspired by Michael Larsen – "if you really want it – you create it" – I went ahead.

Now, immediately I knew as an organizer that it's not about my interests. If you want people to come out you should have something of interest for your community, and not all of them probably will be advanced and senior testers.

How to find a job in testing or become a tester are the questions being asked quite often. I balance the time between topics of testing craftsmanship, management, automation, and job finding questions. Within the meetup I founded Job Finding Club where testers can get help with resume review, interview practice, skills assessment, and other. More senior members volunteered to help, and I also partnered with recruiters.

And, of course, during our hands-on testing sessions we're sharpening our testing skills – which is of great help to everyone regardless of their level.

4. How do you hire good testers? Anything different than traditional approach?

I have no idea what the "traditional approach" is. The approach varies from company to company.

In most cases when I'm involved, I'm responsible for the technical interview. My own approach is turning the interview into a kind of a workshop with simulation of real job situations to work on them together with the candidate and observe their behavior and performance. A fair trick is to ask candidates what publications on testing do they read, and then, maybe, discuss a recent article from Testing Circus.

Like in software testing, an interview may help discover some problems but it can't guarantee absence of them.

On the other hand, if you spent some time working together with a person, you are much likely to know them better than based on spending 1-2 hours in an artificial setting. Workshops that we regularly conduct at the Toronto Testing Meetup are perfect ground of getting to know testers personally and professionally.

5. Do you think domain knowledge is as important for software testers as cross-functional testing skills?

Are you trying to trick me with this question? :) Domain knowledge helps us to recognize the problem but its testing skills that we use to explore the product. Core testing skills – critical thinking, analysis, synthesis, modelling, questioning – are cross-functional and transferable well beyond testing domain.

ALBERT GAREEV

6. In current trends, there are topics like "Traditional testing is dead". Do you think testers should have automation testing knowledge?

"Traditional testing"? What is that? I suppose, as "traditional testing" you meant an approach when manual test scripts are created by one person ahead of time and executed by others upon arrival of the software a few weeks or months after. With test case numbers, created and executed, as main performance metrics. There are variations, of course, and yet it's a quite typical testing model referred as a "factory" approach or school of testing. This kind of testing approach encourages selection of cheap (and poorly skilled) resources – but is still very attractive for a number of reasons. Many outsourcing companies live off the Factory model and remain quite alive.

However, there are products and industries demanding rapid feedback and a highly effective testing, i.e. – finding problems that matter. To remain relevant, testing model has to adapt or "die" – rather, be absorbed into responsibilities of various IT roles. Note that titles are superficial. I've seen many teams and projects claiming that they do great products without QA – but a lot of thorough testing activities were carried on by their team members. I do think though that having within a professional who is specialized in critical thinking and questioning helps the teams move faster with lower risks.

Generally, knowing how code is written and works may help to be a better tester in some circumstances. But I certainly don't think that automation by itself will be a savior for testers with Factory school mindset. Being a long time automation engineer and test lead I have examples of great testers quickly learning automation tools, and examples of automation scripters seriously struggling to offer a real value for the testing team. Another warning I always give to the teams I work with: be conscious about your time allocation – time you spend coding is the time lost for testing. It might be worth as a personal fun but not necessarily a good investment project wise.

Having said that, for specialized testing – load/performance, security, integration, and other – creation and use of tools extends testers' capabilities beyond "naked eye and hand" and makes it only possible.

7. You have vast experience on different tools and specialties. Do you mentor anyone to help new or aspiring testers?

I help anyone in our Testing Meetup as long as I can, and if I can't I try to help people to connect with someone who can. I view our meetup as an empowering environment for building stronger testing community and I work towards this vision.

8. Do you have any advice for our readers on how they should learn software testing?

Practice, think, read, collaborate, and practice again. There are so many opportunities. Courses like Black Box Software Testing by the AST and Rapid Software Testing by James Bach and team. Chapters of Weekend Testing online and testing meetups around the globe. Testing conferences and their online resources. Software Testing Club. Blogs. And testing magazines that will tell readers about everything I mentioned above.

In July I gave a lightning talk – "Software Testing: Professional Growth Leads and Hacks" – accompanied by a mindmap. Feel free to use it to illustrate my answer.

9. Do you read testing books? If you were to name five books that every tester should read which ones those would be and why?

I read testing books and books that can inspire better testing.

ALBERT GAREEV

As a number one I recommend "Perfect Software and Other Illusions about Testing" by Gerald Weinberg. This is a technical philosophical book about testing, quality, IT processes, and people. Not just a bag of tricks and techniques but insightful answers to why do we need (to bother) to test, what's the nature of software development process, and how people play in it based on their roles.

The second book is also Gerald Weinberg's "Becoming a Technical Leader". Testing is a profession that demands a lifelong learning. This book is an excellent guide.

Although this argued by some, testing is one of the most "social" professions amongst other in IT. We need to find problems important to people, influence people, negotiate, and deliver difficult messages – and do that professionally and effectively. What could be the best recommendation here than Dale Carnegie's "How to Win Friends and Influence People"?

"Testing Computer Software" by Cem Kaner is an old but still solid book that can give a great exposure. I recommend reading it along with later added online references and comments by the author.

On par with books I recommend Internet resources. Testers should read blogs, testing magazines (like yours!), think, debate, and participate in discussions.

10. How do you manage your time especially when you have so much to do?

Rather, I manage my energy, priorities, and focus. As a manager, I also empower and delegate. Of course, spending time and being productive are not the same, whether for a solo player and any team with its leader.

A significant part of productivity is choosing what tasks to take on, doing at the right time, and spending right amount of time working on them. There are also some other tricks that I gladly share:

- Communication! Make sure you understood your boss well and your team understood you well. Get in touch frequently. Check if priorities and needs are changing. Make sure you're not spending your time on something that is actually less needed or not needed at all.
- Work journal and mindmaps. I take notes of all my projects, activities, insights, and ideas. I mindmap my strategies. Writing down my thoughts helps me to free up my mind for focusing on the immediate task. Mind maps are great instrument for efficient context-switching.
- Use powers of your sub-consciousness to process the information "in the background". I'm really just an apprentice in these techniques. My introduction began with the book "The Organized Mind" – a good reading for testers, with guidelines on time management and much more.

11. How do you motivate testers, what are your best advices for those who are struggling with career paths?

I see first part of this question asked often and in all kinds of flavors. This is a good indicator of the problem.

In the particular context I'd try to find out what are the demotivating factors and what can we do about them. Removing obstacles might be more effective than trying to overcome them.

And there are workarounds. From some testers coming to our workshops I heard that they're unhappy about their jobs but the Toronto Testing Meetup is a breath of fresh air re-igniting their passion for testing. How we do that? – Just by providing a friendly collaborative environment and opportunities to learn and practice new test ideas.

12. Name few people you would like to thank, people who helped you directly or indirectly in your career as a software testing professional.

ALBERT GAREEV

I'm very thankful to many people I learnt from throughout my life.

At the occasion of this question, I'd like to thank two people whose impact has been crucial on my professional career in Canada.

My first words of deep appreciation go to Veronica Seeto. Being a seasoned leader in QA and Project Management, she is also a passionate coach. I was lucky to be her mentee through Mentoring Partnership Program. You see, even very strong technical skills can take you only this far, if you are a stranger in the foreign culture. Veronica gave me this much needed exposure and feedback, coached in job finding, and supported my endeavors. An impact of such a connection spreads far. Even though it's been nearly ten years ago, I'm honored to call her my friend and mentor.

And I have a special remark about my family venture. In conditions, when it's just you and your spouse alone, in thousands of kilometers away from your former home and life, family and career success are impossible without strongest mutual support. 30's is a very challenging life period – you must keep up with everything – to study and to love, to work and to take care of the family. In our case, the conditions were even more extreme.

I'm deeply thankful to my wife, Gulnara, for all her endurance and support, for her contribution to my career successes, and I'm proud of her career successes as of my own. I'm also thankful for the two beautiful kids. Without family and children – how much are worth any individual accomplishments?

13. Five things nobody knows about you –

Some less known things about me:

- Earlier in my career there was a period when I extensively performed UAT and troubleshooting of PC games
- I'm no stranger to poetry
- In my childhood I wanted to be a biologist and naturalist
- I designed and developed interpreters for a couple of specialized scripting languages
- I'm a member of The Committee on Standards and Professional Practices (CSPP) of The Association for Software Testing (AST)

14. Usually our last question. Do you read Testing Circus Magazine? If yes, what is your feedback to improve this magazine?

I read the magazine and recommend articles to colleagues at work. I think you're doing a great work critically important for the testing community.

As a possible new feature I'd recommend creation of meta-content: cross-referencing the articles, creating theme pages (like the one you currently have for interviews) on leadership, automation, testing techniques, etc.

Blog URL: http://automation-beyond.com

Twitter URL: https://twitter.com/AGareev

Tracking Verification Activities in Agile Projects

- Ravi Kumar BN

Traditionally testers have always been part of independent QA team. With agile deployment, few testers (called agile testers) would now be part of scrum teams. The primary testing objective for an agile tester is to test each user story meeting acceptance criteria.

Many scrum teams use COTS tools such as Rally or TFS to record and track user stories planned and developed in each sprint cycle. Agile testers would be required to test all new features, RFCs and defect fixes in their scrum teams. Each of these test items should be recorded as user stories by category:

- development user stories (new feature development or change request)
- test-only user stories (fix provided by other scrum teams, dev & test spans across sprint due to complexity / capacity issues, workflow test)
- verification defect user stories (product defect for resolution)

Every user story is required to capture all tasks right from analysis & design, development & code reviews and testing. Hence it is evident that all testing activities for a user story gets recorded as tasks and tracked to closure as part of user story acceptance criteria.

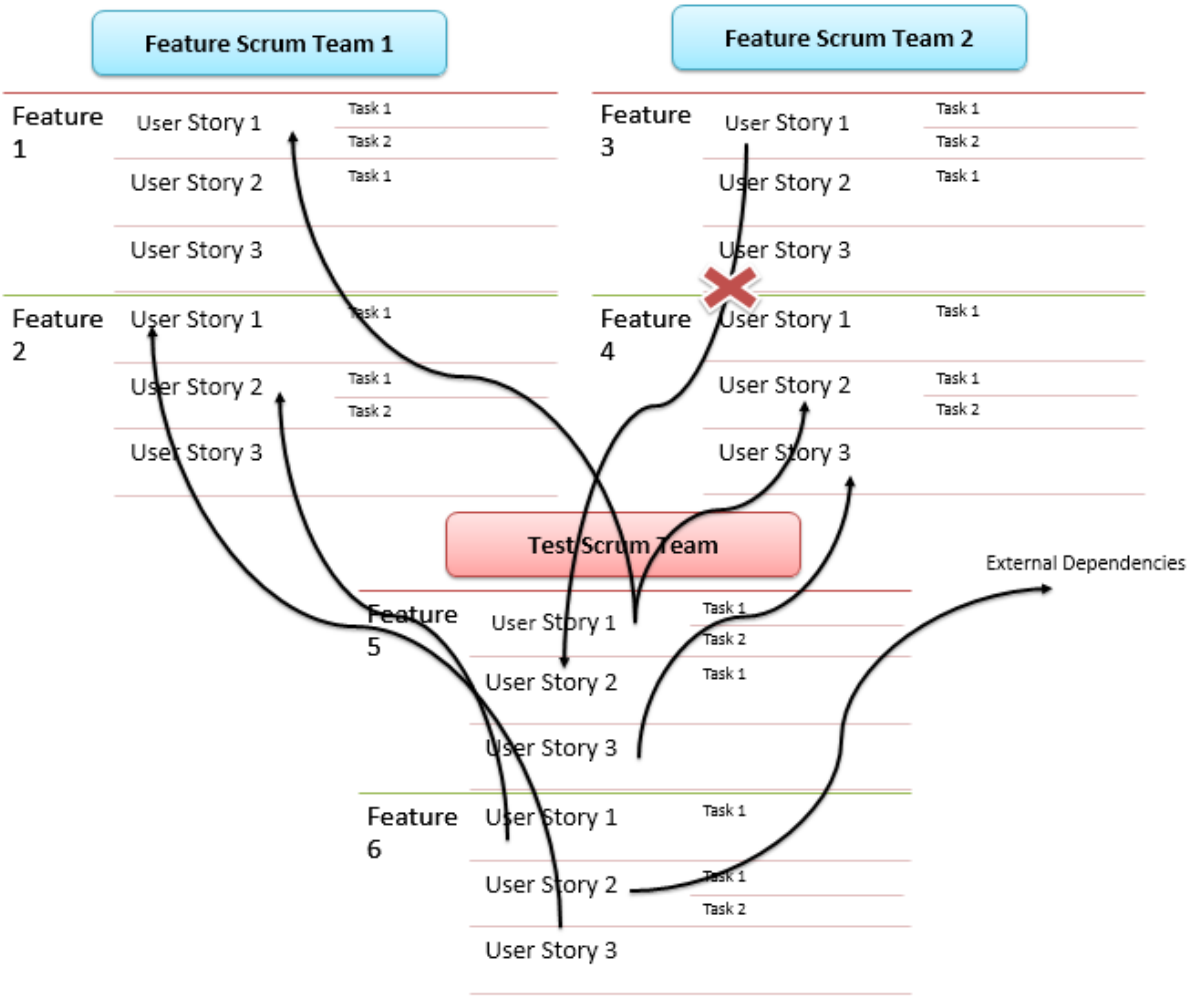
Agile testers are required to identify and create testing tasks for each user story including testing-only user story planned for the sprint in sprint planning meeting. They should plan to create testing tasks covering design, execution and defect verification activities.

System-level solutions are largely built on the efforts of the feature (scrum) teams. One of the sub-team of system team is system test team. This team integrate the work of all feature teams every sprint to test new features and run a regression test on the entire system. This team is generally an independent test or verification team.

The question is '*should verification tasks performed by this independent test team be tracked as user stories (and/or tasks) or not?*' The answer is '**Yes**', due to following two primary reasons:

- To get visibility on verification activities planned/completed for each sprint.
- To understand dependencies for verification activities with other scrum teams.

In a practical scenario, one test team will be integrating and testing features developed by multiple feature (scrum) teams. Hence it doesn't make sense to create verification user stories and related tasks in a particular feature team's sprint backlog nor link them to any specific feature team's user stories. If you did that then verification tasks, if for various dependencies doesn't get done, will stop you from exiting the current sprint even though acceptance criteria is met at the user story level though completion of agile test tasks.



Ideal approach is to create a separate new folder or workspace for test scrum team. Then for each sprint cycle:

- Create one or more verification user stories establishing dependencies (completion of appropriate feature team's user stories including build availability). These dependencies should help the team to know if they are ready to begin verification.
- Create one or more verification tasks under each user story. Remember tasks should not be at test scenario or test case level (can be part of test case management tool such as HP ALM).

While agile test tasks are tied to user story to satisfy acceptance criteria, verification tasks should be tied to features at the program increment level (one or more sprints). By doing this, you can ensure that formal verification is complete before product release meeting the release acceptance criteria.

About the Author

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. He has authored and published few testing articles in QAI conferences and online magazines such as Testing Experience, Testing Circus etc.

IT'S CHECK AUTOMATION, AND NOT TEST AUTOMATION.

WE FOCUS ON TEST COVERAGE & CHECK AUTOMATION COVERAGE.

WANT TO WORK WITH US?

— Write to our Founder at st@testinsane.com —

7 Practical Ways Testers Can Help Developers

- Karen N. Johnson

Bring empathy to the tester/developer relationship
Of all the people on a team, it is the tester and developer that have the most unique relationship. Why? Because testers often have to mentally walk down the same path as developers (in order to test) and in doing so, we gain empathy for the work developers on the team must accomplish. Here are seven practical ways a tester can help improve tester-developer relations and work.

1. Resolve design gaps
Offer to resolve missing information. Sometimes it is not until development is under way that missing or misunderstood design information is realized; volunteer to research and resolve the missing gap of information. As developers are building software, they may come to realize that some aspect of the design is lacking the information they need. When you become the trusted tester on a team, the developers are sometimes willing to share that frustration with you – in that moment, you can volunteer to track down the missing information.
From a personal perspective, I've enjoyed opportunities to dig in and talk with different people from UX designers to product owners to the end users to help resolve not just how something would or should be built but how software will actually be used – that research has helped not just development but my testing as well.

2. Share the gear
Offer to share test equipment. In mobile testing, often it is the team's tester who holds onto not just the devices but has the responsibility for keeping track of devices. Sometimes the tester on the team has access to more devices and computer equipment than the developers on the team. Whether the devices are BYOD (bring your own device) or an actual test lab, offer to share in order to help developers perform as much testing or investigation (what happens on a different operating system, browser or device), as they are willing.

3. Offer to pair test
In recent years, there have been many discussions about pair programing and pair testing. When developers pair with testers, they can pick up insights into the testing mindset resulting in an expanded mindset about what testing is or can be.
Here are a few resources on pair testing that are I've found insightful and practical:

More powerful pair testing @CAST 2013 - Rachel Carson
https://www.youtube.com/watch?v=LoWmfldDOs8

Better Story Testing through Dev-Tester Pairing – Abby Fichtner
http://www.slideshare.net/HackerChick/better-story-testing-through-devtester-pairing?qid=859c6262-f863-4b4c-878c-66eb7df0f6c0&v=qf1&b=&from_search=6

Sherlock Holmes and Pairing @Romanian Testing Conference 2014 - Adi Bolboaca
http://www.slideshare.net/adibolb/sherlock-holmesandpairing-adibolboaca?qid=43361e9d-e407-4c2c-90cd-1511179075d4&v=qf1&b=&from_search=5

4. Test "what if" scenarios, boundary & error conditions

In recent years, we (as an industry) have focused on pair testing but at times, there is an advantage in working solo as a tester to be the person who can test early, test ahead and discover what happens by investigating what happens with the software when we try a certain condition or scenario. We can work with developers by working alone and ahead to discover. Test, learn and then share what you learn with developers as rapidly as practical to create a tight feedback loop.

5. Review auxiliary code

As a tester, you might not have the technical skills to perform code reviews but you might have enough technical knowledge to review other "bits" of auxiliary or supporting code written in HTML, CSS, SQL or stored procedures. Or you might volunteer to run a suite of automated tests in a continuous integration environment. Another option to lend a hand would be to read unit level tests that have been written in order to "complement" manual testing efforts to the automation that may have been built by development.

6. Build test data sets

Offer to build sets of data to help test both positive and negative conditions. You might be able to build different types of test data that can help both you and a developer test positive and negative conditions. For example in a test environment, you might be the person who takes the time to build test accounts with different permissions or different "past history" information to speed up testing. You could offer to share those test accounts or test data with your developers to help accelerate their testing as well.

7. Share test ideas

Share your test ideas throughout the development process. In the beginning of my career, I enjoyed finding flaws in logic and as a result, finding "great" bug finds but over the years, I came to realize that the sooner I shared my ideas, the sooner the developers could code to prevent some scenarios from being built to begin with. I realized, why not voice my test ideas early on to help prevent bugs from ever being created.

About the Author

Karen N. Johnson has worked as a software test consultant for several years. Recently she joined Orbitz Worldwide Inc. as Director of Mobile Quality. While focused on software testing and predominantly working with the testers throughout an organization, Karen helps teams and organizations improve quality overall. Her professional activities include speaking at conferences both in the US and internationally. Karen is a contributing author to the book, Beautiful Testing by O'Reilly publishers. She is the co-founder of the WREST workshop, the Workshop for Regulated Software Testing. She has published numerous articles; she blogs and tweets about her experiences. Find her on Twitter as @karennjohnson (note the two n's) and her website: http://www.karennicolejohnson.com.

Govind PK

Expert Exploratory Tester

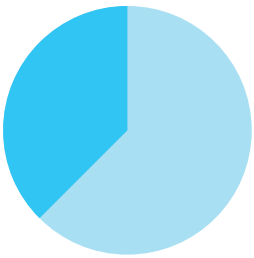
India

236	1653
Following	Followers

Level 7 (870 points)

30 more points needed to became a validater

4 test cycles won



83 bugs reported

Receivable amount in this month

₹19,500/-

What is 99tests?

99tests is a crowdsourced testing platform consisting over 12,000 testers who have logged around 75,000 bugs. Testers are exposed to various types of testing and domains. 99tests is a platform where you can earn, learn and grow by login bugs, connecting with testers across the globe.

Why should I sign up with 99tests?

1. Work whenever you wish to and from any place you want to
2. Choose the products/ software you want to test
3. Collaborate with the testers across the globe, share your skills and grow with them

Meet our community


India


USA


Russia


UK


Philadelphia


Austin

Join our community today!

99tests.com

"Being in the field of software testing for a while, I always wondered whats my stand in the testing skills I possess compared to the other testers. This was more of a positive thought as I was more keen on identifying what I lacked and what I needed to improve."

AWARDED BY

NASSCOM GSF productnation BIG IDEA Small Business TRENDS

The Evolution of Cooperation

Why Agile Approaches Will Prevail

- Michael Niessl

One of my favorite non-fiction books is Robert Axelrod's The Evolution of Cooperation, published more than 30 years ago. The book compares different strategies to win a game called the Prisoner's Dilemma, a classic scenario (explained below) used in game theory. Based on Axelrod's findings, he showed how cooperation can, and most probably will, evolve even if the environment is hostile. Under certain circumstances these findings, prove or at least strongly suggest, that transitions from traditional development methodologies to agile methodologies will succeed even in big organizations, and that agile development approaches will prevail over time.

The Prisoner's Dilemma

The idea of the Prisoner's Dilemma centers around two hypothetical criminal friends. The friends are arrested and faced with imprisonment. Authorities can't present enough evidence to justify a high sentence, so the prosecutor, hungry for success, offers a deal: if one prisoner betrays his friend he will be set free immediately and the other one will be imprisoned for three years. This deal is offered to both friends without giving them a chance to coordinate their actions. If both friends rat out each other, the sentence will be two years for both of them. If none of them speaks, they will both do just a year.

	Friend 2 keeps silent	Friend 2 rats on his friend
Friend 1 keeps silent	1 year sentence for both	Friend 1: 3 years Friend 2: goes free
Friend 1 rats on his friend	Friend 1: goes free Friend 2: 3 years	2 years sentence for both

The interesting fact is that, from a game theory point of view, the wisest albeit counterintuitive decision would be to betray your friend. It would guarantee the best possible outcome independent of your friend's decision. Scientists have a fancy term for that: in game theory it is called a Nash Equilibrium, named after Nobel Prize winner John F. Nash whom you may know from the movie "A Beautiful Mind".

Evolution of Cooperation

In the beginning of the 1980s, Robert Axelrod asked fellow scientists around the world to submit strategies for a contest based on the Prisoner's Dilemma. He changed one significant parameter though. Instead of just playing just one round of the game, the strategies were compared over the course of a series of rounds. In such an Iterated Prisoner's Dilemma, a game strategy can benefit from actions taken in previous rounds – e.g. if one opponent started to cooperate in previous rounds the other opponent can choose to reciprocate that behavior.

Everybody taking part in the contest submitted a strategy. Axelrod would then compare each strategy against another strategy in 200 iterations of the Prisoner's Dilemma. Eventually the scores accumulated by each strategy throughout the tournament led to an overall winner: the best strategy.

It was surprising that despite the rather complex and elaborated strategies (or better, algorithms) used in the tournament, a very simple strategy won in the end. It's called 'Tit for Tat'. The strategy is to cooperate (keeps silent) in the first round and mirror the other player's decision from the previous round for all subsequent moves. After analyzing and publishing the

results, Axelrod called for another tournament to see if there was an even better strategy. Again scientists from different disciplines sent in their strategies, knowing well that 'Tit for Tat' won the first tournament. Guess what happened? Surprisingly 'Tit for Tat' won again.

Based on the findings of these two tournaments, Robert Axelrod derived insights about the nature of cooperation. In addition to three other characteristics (retaliating, forgiving, and simplicity) that will not be discussed further at this point, he found that a successful strategy has to be cooperative in the first place. Axelrod made sure to mention that there are certain criteria to be met to guarantee that such strategies succeed. First and foremost, it's the iterative nature of his setup - with the likelihood of the opponents 'meeting again' - that fosters cooperation.

Axelrod took it a step further, showing that even in a hostile environment, where there is likelihood of opponents "playing dirty" cooperative strategies will not only survive but also succeed.

Small clusters of individuals playing cooperatively will be more successful than opponents playing dirty.

The beautiful thing is that it doesn't work the other way round – small clusters of individuals playing dirty will not succeed in a cooperative environment. I see this as a very romantic and optimistic (but still mathematically sound) way to look into the future.

Agile Transitions

Coming back to the tester's world, let's look at the ongoing changes in the development departments of established enterprises around the world. The last couple of years have seen the rise of agile development methodologies. Organizations have adopted, or plan to adopt, agile methodologies and either struggle to implement it or struggle to scale their agile adoption. While methodologies like Scrum look simple on paper, it quickly becomes difficult when R&D departments realize that the whole organization needs to pull together for a smooth transition towards a real agile behavior – it can't be done just by the Devs on their own.

Over the last decade(s), a lot of time, effort and money have been invested into building test centers of excellence (TCoEs) to centralize all the testing activities of an organization to use synergies, bundle skills, and provide a single point of contact for anything involving testing. These TCoEs have also often been outsourced, meaning that there are businesses depending on running TCoEs for a large share of their revenue.

As Scrum accounts for 90%+ of the methodologies adopted during agile transitions, it is a reasonable simplification to look at just that methodology. A typical Scrum team consists of three roles: the Product Owner, serving as the voice of the customers, the Scrum Master, helping the team to become more effective and efficient following the Scrum processes in delivering product increments, and finally the Development Team. There is no tester role in a Scrum team (there's neither a 'tech writer' nor a 'business analyst' role)! Testers are considered part of the dev teams, and going forward the line will get continuously blurry. Having testers embedded in development teams is the exact opposite of having testers clustered in a TCoE. All the efforts, time, and money taken to build these TCoEs are not helpful for a smooth agile transition.

In addition, levels of expectations have increased and the tester's role is constantly evolving. Scrum (and other agile methodologies such as Extreme Programming) force development teams to work in short iterations. In the past, the R&D departments hacked away for months and months, then tried to put everything together, integrate it, make a product out of their work, and finally handed it over to the test team. At that point time was typically short, deadlines were passing by, and testing had to be cut short and/or release dates postponed. This approach was called 'waterfall'. The short iterations of Scrum, called sprints, address the shortfalls of the waterfall methodology by its very nature. A product is developed (and even delivered, to stakeholders at least) every two to four weeks. Work that is done in such a sprint is not just coding – remember that the testers are embedded into the team. The product increment delivered at the end is designed, developed, tested, documented, and ready to be delivered. It is this approach that, during a transition, puts a lot of pressure on the testers previously clustered in TCoEs and on the organization as a whole. Suddenly testers are forced into test automation – otherwise the simply can't keep the pace. The test is 'shifted left' - it has to start as early as development. Manual testers without too much technical background are suddenly part of a very technical development environment. Testing tools need to be changed in order to cope with these new challenges. Processes need to be adapted or be removed altogether.

What's the advantage? Why would any organization accept so many troubles to go through such a transition? Not surprisingly, it's the usual suspects: shorter time-to-market, higher quality but also quicker feedback from customers, a more flexible reaction to this feedback. In short it is best of both worlds: doing the right things and doing things right. Agile methodologies pay off in many different ways.

Why agile will win

How are the previous two stories, agile transitions and evolution of cooperation connected? Although it may seem to be farfetched, I think the two are closely related. Robert Axelrod's work shows that such transitions can succeed even if, all things considered, it doesn't look very promising in the beginning.

Agile methodologies enforce cooperation. One of the key agile principles is 'Individuals and interactions over processes and tools' – which very simply means that people talk to each other, work together… they cooperate in order to create a better product for their customers faster. Agile methodologies, and thus cooperation, pay off – they reward their proponents. That is one of the key criteria, discovered by Robert Axelrod, needed for cooperation to succeed. The other very important criteria, the 'shadow of the future' as he called it, where you have to meet again for cooperation to evolve (i.e: your actions today will have an impact on how your colleagues treat you in the future) is an intrinsic characteristic of Scrum. A tongue-in-cheek way to look at it is that a string of sprints is nothing more than an Iterated Prisoner's Dilemma. Sprints guarantee that the 'opponents' will meet again and again and their actions will have an impact on future behaviors.

So although maintaining the status quo as a team (e.g. as the TCoE) might appear to be the best solution individually, it is very likely that a solution that is best for the greater good of the whole organization will succeed over time. All it takes is a small group of individuals working together and an environment that fosters cooperation – as Scrum does.

Start the revolution – you will succeed.

About the Author

Michael Niessl started his career as a software tester over a decade ago. Working for Tricentis, a test automation and test management tool vendor, he has had the chance to work at the very edge between development and testing ever since. As an engineering manager he is currently pushing forward the agile transition of Tricentis' development team, facing all the challenges such transitions mean for an organization as a whole.

Little Lessons Learned from Software Failures

- Doris Robinson

In 2004, Electronic Data Systems (EDS) now acquired by HP was criticized by the UK's National Audit Office for its work on IT systems for the UK's Child Support Agency (CSA), which ran seriously over budget causing problems which led to the resignation of the then CSA's head, Doug Smith on 2004-11-27.

The system's rollout had been two years late and following its introduction, the CSA was obliged to write off £1 billion in claims, while £750 million in child support payments from absent parents remained uncollected.

An internal EDS memo that was leaked at the time admitted that the CSA's system was "badly designed, badly tested and badly implemented". UK MPs described it as an "appalling waste of public money" and called for it to be scrapped.

What can we learn from the above?

1. As technology becomes ever more complex and challenging; business processes are constantly evolving, it is critical that both business and IT project teams are singing from the same hymn sheet and dancing to the same drum beats. There should be an overwhelming transparency of what the project objectives are, who are the benefactors and what are the gains or lack thereof if the project is not successfully implemented;
2. Identify/secure a captain/chair that shares the project vision; we know what happens when there are too many cooks in a kitchen - the soup might end up been too salty or without any at the dinner table;
3. Have a clear goal; don't be rigid in changing your approach in order to achieve a favorable result. Only statue remains as is, as the captain, be willing to stir the ship to avoid troubled waters or away from sea rocks at any time of the project life cycle, and always keep the wider team involved and informed on the new direction or approach;
4. Engage, secure the right key stakeholders from the get go and incorporate the act of human resource retention. Allow room for every team member to feel valued and that they have contributed to the greater good of the project;
5. Set up the core technical, business representatives, test and QA personnel needed to see the project through from initiation down to production usage and encourage collaboration among teams;
6. Ensure the project teams responsibility for requirements gathering, design, build, test, quality, business and those signing the project checks are in harmony with the overall project vision, timeline and these collective teams are assembled from the beginning of the initiation stage;
7. Engage already motivated people when assembling your team members, inspire them with what the objectives are and understand that the phrase human resource has a stronger word "**human**" – treat members of your team with respect, duty of care and understand everyone has an individual goal; remember, when you fail to inspire motivated people, they will find it somewhere else;

8. If you have to, carry the various teams along, ensure they are equally kept informed of the project communication and statuses;
9. Manage, see the project as one big family – can't choose them but they chose you in essence no one likes to be left out of appropriate or effective team communication or information;
10. Ensure those that are familiar with the current business process are evenly matched with the strategic process that brought about project demand - in other words ensure the business change or technology change is fit for the business future direction. The solution or service introduced might be successfully designed, tested and implemented but majority of your future business processes or service direction does not match what was implemented;
11. Provide a realistic delivery timeframe, often at times you see it in a project that gets moved every month instead of taking a holistic view or an audit review on the project direction and measures on how to address what is the cause of constantly needing to change the timeframe dates. In practice, you cannot leave the tap on and constantly be mopping the floor and expect the floor to be dry any time soon;
12. Embrace change, projects are designed to change the way we currently do things. So ensure your project is championed by people that are accommodative to change processes and can literally sing from the tree top in ensuring anyone they come across is able to register/identify with what the proposed changes are in a positive light.

About the Author

Doris Robinson is a Testing Consultant with 14 years' experience in IT from working in different capacities; as a project manager, QA, test and release manager.
Doris has delivered testing activities for complex global management information systems and continuous testing planning responsibilities for several companies. She is currently working as the testing Subject Matter Expert for Johnson Matthey, a global sustainable company in the UK, she was previously the Quality Control Manager for one of the world oldest private investment bank C. Hoare & Co in the UK.

Connect with our 4000+ fans in our facebook page.

http://www.facebook.com/TestingCircus

The Simplest Web Automation Tool

Simple

Powerful

Productive

Simple powerful scripting

Smart Object identification - No complex Xpaths

Powerful record and playback -Any Browser Any OS

Fast parallel batch playback

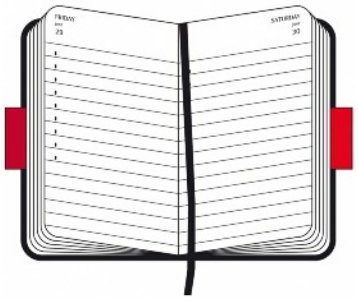
Ajax? No time out issues

Become our fan -
https://twitter.com/_sahi
http://www.facebook.com/sahi.software

Request a free demo by sending us an email at support@sahi.co.in

http://www.sahi.co.in

A Fake Tester's Diary

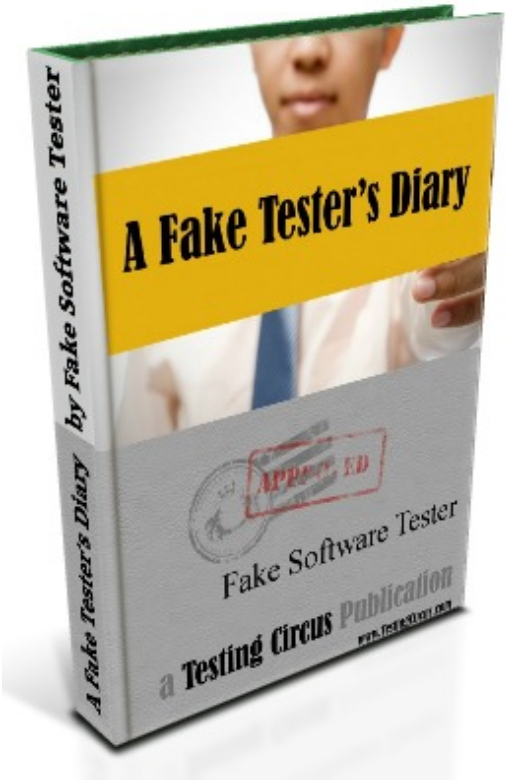


Episode 57

http://www.testingcircus.com/category/a-fake-testers-diary/

The Testing Marathon

5 years of Testing Circus. Woohoo.... and my goal says I have 45 years to go. 50 was my target for Testing Circus. With most circuses going out of business, Testing Circus seems to be one of the few circuses still left in business. It is when you reach a milestone (like 5 years) when you begin retrospecting on yourself and on your life so far; when I look back at the last 5 years and did my retrospective, I realized that there's a bunch of things that others should have done differently. Then, I realized that it helps someone else that much more if I retrospect and tell them what to do. So, I did a retrospective and decided that I need to point hands at myself if I have to improve.



My Retrospective

Slipping is fine as long as you can get back up on your feet. We all will keep falling and learning; it's more important to fail fast and have the capability to get up. For example, in the world of mobile testing, there are so many changes happening that a lot of things are not in your control; give yourself the skill to get back fast when you have problems.

Colleagues are a great place to learn from --- don't think that your colleagues are out to get you; try to learn from them. They are a great source of domain knowledge and testing knowledge. Have periodic time setup with your colleagues and ask them for feedback on your work. Ask them specifics on things that you are doing fine and where you need to improve.

Don't let someone's promotions pull you down --- A normal sight in a team of 10 is where 2 people get promoted and the other 8 don't; more often than not, they let it affect them and they stopped contributing more. If you did not get promoted and your friends got promoted, don't let bogged down; use it to spur you to get better and get a promotion in the next cycle or get a better paying job.

Feel free to talk about your mistakes in front of a group --- We all are very protective about our mistakes; don't be. Feel free to talk about your mistakes in front of a peer group; feel free to talk about what you learnt from them and share your learnings. Don't feel bad if someone critics you and mocks you; it's a learning that you should share with the entire group.

Ignorance cannot be an excuse --- have you ever told someone that you don't know who does stability testing for your projects because they are in a different group? Have you told someone that the systems behaves in a specific way because it's in the requirements document? If the answer to both questions is YES, then it means you have not done due diligence about chasing your ignorance away from yourself.

Career Progress --- Your projects, office politics, personal life taking a demand on you are not reasons for your career to stagnate. Your career is stagnating only because you don't have the skills to take it forward. If you think like this, you will keep re-skilling yourselves or you will get to work in a place where you are paid your worth and you get a business position that you are worthy of.

So happy retrospection !!!

Congrats on your 5th anniversary, Testing Circus is a terrific resource and doing so much to help us forge new frontiers in testing.

- Lisa Crispin

#Testers2Follow

Andy Tinkham

Global Practice Lead for QA & Testing @ C2 IT Solutions, board gamer, photographer, genealogist, co-host of Testing Bias podcast.

https://twitter.com/andytinkham

Next Gen Test Conf

Next Generation Testing Conference (2015 Schedule): Chicago: 17 Sep | Johannesburg: 31 Jul | Manila: 6 Aug | Singapore: 20 Oct | China: Dec | Brazil: Dec

https://twitter.com/NGTConf

PerfBytes

Helping IT professionals to improve performance practices.

https://twitter.com/PerfBytes

SmartBear Software

Software quality tools for the connected world.

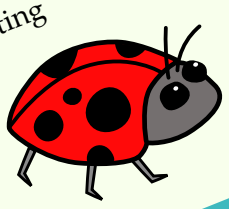
https://twitter.com/SmartBear

http://Twitter.com/TestingCircus

Testing Circus

Testing Circus is a world's leading English language magazine for software testers and test enthusiasts. Monthly editions since September 2010. #testing

Follow us at https://twitter.com/TestingCircus



#Follow Us On Twitter

https://Twitter.com/TestingCircus

200+ testers to follow on Twitter -
http://www.testingcircus.com/testers-in-twitter

Book Reviews

BOOK WORM'S CORNER

Completing 5 years of Testing Circus reading. Happy Birthday to Testing Circus and to all the other animals in Testing Circus. I always hoped that they would gift us some good books, but they always tell us to read it by ourselves. Hopefully, we will get a book sponsor who will donate books to us to chew on.

The 2 books that I chewed on this month are

Perfect Software & Other Illusions of testing

A "Succinct book"; I could appreciate this book only because I was a fan of the author. If you belong to a testing community that is powered by ISTQBs and Testing Certifications, it's impossible to relate to this book at all. This book does not try to influence you to change your mind either; it's more about how the author believes in a certain ideology as the basis for this book. If you are not a fan of Gerald Weinberg, then read his other literature to become his fan and then buy this book; only then you'd understand that both "Perfect Software" & "Perfect book on Software testing" are actually illusions.

Buy it here

Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality

A "Pragmatic" read; not many of us would have solved (or faced) the business problem of automating large enterprise systems. This book teaches you how to do that. A common trend is for people to search for tools, hands-on exercises, tutorials in automation books; if you are one of them, you'd be disappointed. Secondly, don't make the mistake of "thinking agile" while reading this book. It's not about testing in agile mode at all. However, if you ever have to gift a book to a testing team, gift this book and see how they can do wonders to your project. Or if you have a lowly paid tester and need to hire a costly consultant, then buy this book, give it to the lowly paid tester and ask him to make a recommendation; then, hike his pay and hear his ideas. Most likely, he can read this book and solve your business problems.

Buy it here

Love,
WoBo

AppAloud
Talking Wee Bit Louder About Apps

New Release Of TestRail Offers Full JIRA Test Management & More

Gurock announced a new major update of our popular test management application. TestRail 5.0 offers an all-new & best-in-class JIRA integration with full JIRA Server and JIRA Cloud add-ons to directly view and track linked TestRail test results, test cases and reports inside JIRA.

TestRail 5.0 also introduces new live defect reports to directly see the status of all linked issues and test results in one place, new options to lookup the status of linked requirements from external tools as well as detailed section statistics for test runs and milestones. New options have been added to integrate TestRail's statistics with Confluence and other tools, improved Markdown formatting options, new test result history and context tabs and much more.

New features added in TestRail 5.0:

Universal JIRA Server and JIRA Cloud Add-ons

TestRail5.0 offers a universal JIRA test management integration for TestRail with both JIRA Server and JIRA Cloud. It now offers the same rich integration options for both JIRA Server and JIRA Cloud by including a native JIRA Server plugin and Atlassian Connect add-on for cloud instances. This way customers can easily use TestRail together with JIRA in every possible combination, and be sure that the integration always works even after possibly migrating to the cloud in the future.

Linked Test Cases and Results in JIRA

With TestRail's new JIRA add-ons, authenticated TestRail users can now directly see details such as test result comments, attached files, screenshots and the live status for linked results inside JIRA. Additionally, linked test cases (for JIRA requirements and user stories) are accessible from the JIRA issue sidebar for a full two-way requirement, issue and test case integration.

TestRail's new JIRA integration also makes it easy for users to expand linked results to lookup more detailed test result information from JIRA, allowing testers, developers and managers to directly see the related test result comments, attachments and attributes at a glance.

TestRail Metrics & Reports in JIRA

TestRail 5.0 enables to embed and support viewing TestRail's dashboard and project statistics directly in JIRA for authenticated TestRail users. TestRail's integration also automatically remembers the selected TestRail project to automatically map statistics to JIRA as you view them. The new embedded statistics make it much easier to track your testing efforts from JIRA and also directly jump to relevant test runs, milestones and test suites.

Real-Time Defect Reports

TestRail 5.0 introduces new defect reports to directly see the status of all linked issues and defects at a glance. The new Defects tab for test runs, test plans and milestones show the latest status and details of all pushed and linked defects. And because TestRail always pulls the latest information from your issue and bug tracker when you open the report, there's no need to synchronize or update any data manually!

Live Test Run Section Statistics

TestRail 5.0 also adds new features to the test suite and test run pages. To make it easier to see the test results per group (e.g. sections such as feature modules or grouped tests based on test type etc.), TestRail now shows small graphs next to group titles to indicate the distribution of results in that group.

Testers can now also lookup the detailed test results of the selected group. Additionally, TestRail also shows the number of test cases in a group from the test suite/test cases page in addition to the number of tests per group on test run pages.

Reference Lookups & JIRA Requirements

TestRail 5.0 is now adding capabilities to directly lookup the details and status of requirements directly in TestRail without having to leave the application. Similar to test management application TestRail's existing defect lookup functionality, you can now also configure the integration to enable lookups for references and requirements. This works for all existing and future defect plugin integrations, so you can lookup references from JIRA, Redmine, FogBugz, Rally, Axosoft and many other tools.

To make it easier to create new test cases for requirements, user stories and use cases stored in JIRA, the new JIRA integration also includes a button to create new test cases from a JIRA issue. This not only allows you to select the TestRail project (and optionally test suite) TestRail also links the new test case to the JIRA issue automatically.

Improved Markdown Table Formatting

TestRail 5.0 Offers new powerful table formatting options to make it easier to include full tables in your test case details. TestRail also includes a handy table build dialog to easily select and design your table options.

Salesforce Unveils Salesforce IoT Cloud

Salesforce announced Salesforce IoT Cloud. Powered by Thunder, a massively scalable, real-time event processing engine, IoT Cloud connects billions of events with Salesforce, unlocking insights from the connected world that empower anyone to take the right action, for the right customer, at the right time.

"Salesforce is turning the Internet of Things into the Internet of Customers," said Marc Benioff, chairman and chief executive officer, Salesforce. "The IoT Cloud will allow businesses to create real-time 1:1, proactive actions for sales, service, marketing or any other business process, delivering a new kind of customer success."

Connected World, Disconnected Data

The combination of mobile, social, sensor, wearable and cloud technologies has triggered a deluge of data. More than 90 percent of the world's data has been generated over the last two years. And, with the number of connected devices projected to reach 75 billion by 2020, the volume of data available is expected to grow exponentially.

This world of connected devices and digital content presents an enormous opportunity for companies to take advantage of the new data. In a June 2015 report, the McKinsey Global Institute estimates that IoT applications may have a potential economic impact of as much as $11.1 trillion per year by 2025. However, businesses have been unable to capitalize on the vast volume of data from the Internet of Things.

Salesforce IoT Cloud, Powered by Thunder—Connecting to the Internet of Customers

IoT Cloud empowers businesses to connect data from the Internet of Things, as well as any digital content, with customer information, giving context to data and making it actionable—all in real-time. Thunder, built on a massively scalable, modern architecture, can "listen" to the connected world, ingesting billions of events a day, from any source. IoT Cloud's capabilities include:

- Listen to the World at IoT Scale: IoT Cloud connects everything to Salesforce. In addition to the Internet of Things, connecting to phones, wearables, windmills and industrial turbines and other devices, IoT Cloud connects data from websites, social interactions and more to Salesforce. By connecting the billions of real-time events and digital content with Salesforce, the IoT Cloud brings customer context to transactional data.
- Trigger Actions with Real-time Rules: With IoT Cloud, business users can use intuitive, point-and-click tools to define, modify and set rules and logic for events that can trigger actions across Salesforce. A global fleet management company, for example, can enforce passenger safety standards by setting filters for "hard brakes" or "hard accelerations" and defining rules that trigger in-car sensors to log service cases reporting possible instances of erratic driving. Or, a national retailer holding a holiday sale can set rules based on loyalty program status, inventory or sales performance, triggering retail beacons to send discount offers to in-store shoppers in real-time.
- 1:1 Proactive Engagement through Salesforce: IoT Cloud seamlessly works across the Salesforce Customer Success Platform to surface insights and trigger real-time 1:1, personalized actions for sales, service, marketing or any other business process. For example, a thermostat provider can parse through billions of events gathered from weather forecasts, sensors and temperature settings to proactively alert customers on how to manage their HVAC usage within their predefined budget. Or, a vehicle assistance service partnering with an auto brand can send personalized offers on behalf of local dealers based on sensor data that tracks fluid levels and mileage.

Salesforce Ecosystem Expands with IoT Partners

Salesforce is accelerating the adoption of IoT Cloud with initial launch partners ARM, Etherios, Informatica, PTC ThingWorx and Xively LogMeln that provide connectivity between devices and the Internet. The company will continue to expand its ecosystem of IoT partners to enable businesses to connect with their customers in a whole new way.

Salesforce App Cloud—the Unified Platform for Delivering Connected Apps Fast

Thunder is part of the Salesforce App Cloud, an integrated set of platform services that enable businesses to develop connected apps fast. With services for rapid app development, modern user experiences, integration, mobile app dev, identity management, compliance, governance and more, App Cloud is the most comprehensive and agile platform available to CIOs to deliver their app portfolio.

App Cloud also includes an ecosystem of 2.3 million developers, who have built 5.5 million apps, and the AppExchange, the world's largest enterprise app marketplace, which features more than 2,800 ISV apps and 40 Lightning Components. All of this runs on the industry's most trusted enterprise infrastructure that delivers approximately 3.7 billion transactions every business day.

IoT Cloud will be in pilot the first half of 2016 with generally availability later in the year. Pricing will be announced at the time of general availability.

AppAloud
Talking Wee Bit Louder About Apps

Mobile Banking Apps: WHAT Consumers Expect And HOW Testers Can Deliver

As smart phones, tablets and phablets become popular; a Forrester research shows that by 2018, the world will have 214 million mobile banking users. Banks are in a race with each other to attract customers with mobile apps that appeal in terms of functionality and looks. In such a situation, testing is sure to play a crucial role in determining that quality is not compromised in the 'time-to-market' race.

Here we take a look at the key features that customers expect a mobile app should possess and the testing considerations that need to be considered to develop the perfect app.

Key features of mobile banking apps

1. The first and foremost aspect that worries end-users is security. A banking app should allow only secured entry via PIN possibly in combination with knowledge-based answers and visual keys.
2. All apps should allow basic banking functions like checking balances, transaction history; pay bills, cash transfers and other functions
3. Apps can achieve high customer engagement by including functionality like ATM locator, bank locator, and multichannel options to contact the bank, including social channels.
4. Among apps that meet all the above basic features, the app that finds favor among users is one with an eye-catching design and an intuitive layout that is easy to use.
5. Some leading bank apps come with neat features like
    - Money Transfer
    - Temporary passcode generation through which you can withdraw money (if you have forgotten your debit card)
    - Sending alerts when account balance falls below or exceeds a pre-specified figure.
    - Reporting lost/stolen cards.
    - Check deposit

- Contextual and targeted cross-selling

Testing considerations for mobile banking apps

1. Testing has to take into account multitude of device types with varying screen sizes. Use of emulators can help here, but testers should be aware of what can be reliably tested using emulators
2. Performance of an app will vary with speed depending on the network, type of internet connection and data usage. Testing should check the performance under different conditions of usage.
3. Apps that are easy to install and use are popular among consumers. Sufficient usability testing should be done to address these factors adequately.
4. Banking apps should have high fault tolerance: The tester should check what happens when the session is interrupted for any reason.
5. Security Testing: Adequate attention has to be given to checking that the app meets security and compliance requirements in order to protect customer data.
6. Load Testing: Banking apps have crashed time and again due to high traffic. Testing should ensure the app has sufficient load bearing capacity
7. Localization Testing: The app should be equally usable for people of all market and regions. Testing should consider location specific content, date formats, characters, currency conversions, and others.
8. Error messages generated should be helpful and make sense to the user
9. Crowd-sourcing testing: Getting testers or beta users around the world to test the app subjects the app to a broad range of devices, networks, and locations

serious about software quality

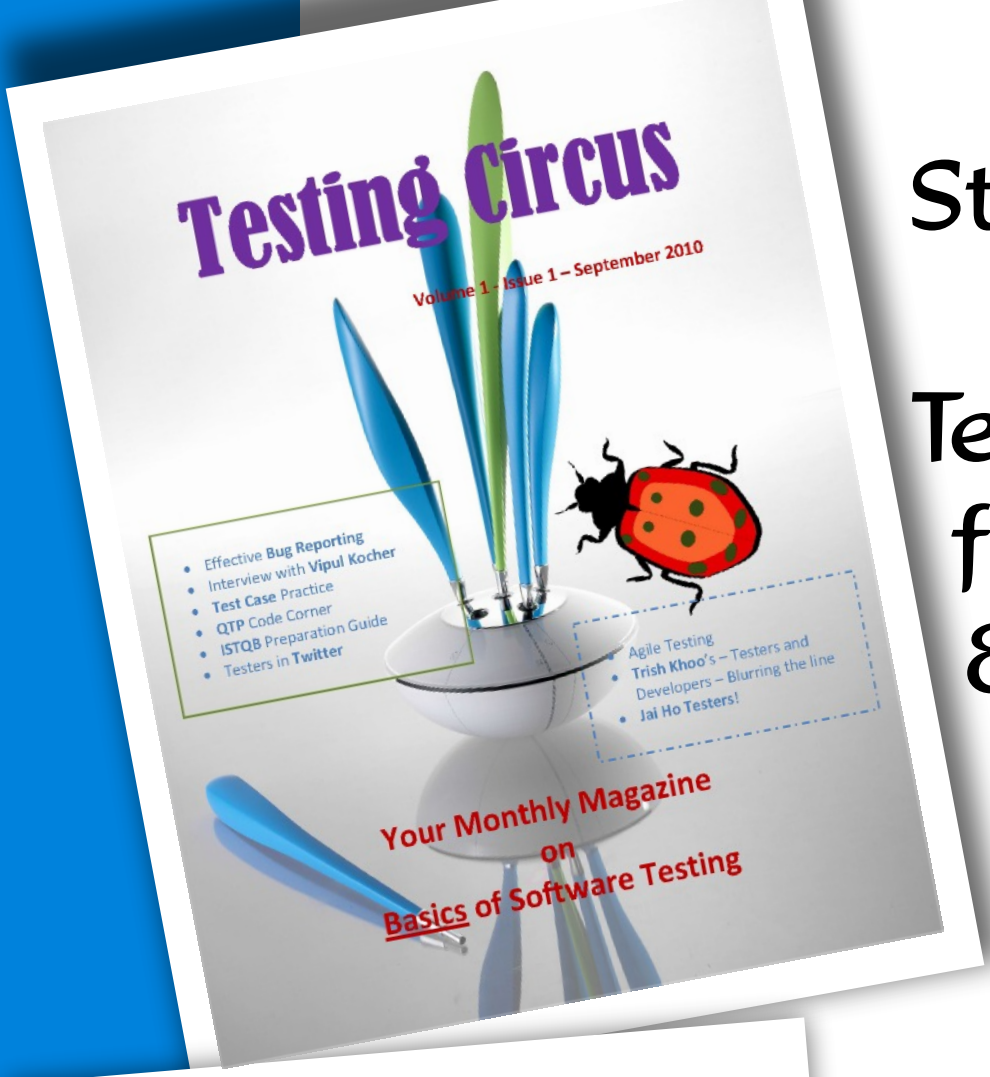
At Doran Jones, our mission is to help technology organizations improve their ability to deliver software and add business value. We believe the best way to do this is through hands-on delivery, working alongside our clients. Let us show you life at the intersection of talent and opportunity.

- **Software Development**
- **Software Testing**
- **Training and Coaching**
- **Recruitment**
- **Urban Onshore Outsourcing**

DJ Doran Jones

as seen in: WIRED

www.doranjones.com



Still relying on reading Testing Circus from tweets & facebook updates?

Subscribe with your email id and get the magazine delivered to your email every month, free!

To Subscribe

Click Here

Testing Circus

www.testingcircus.com