

THE INTERNATIONAL MONTHLY FOR NEXT GENERATION TESTERS

# Tea-time with Testers

October - November 2015 | YEAR 5 ISSUE IX





# TEA-TIME WITH TESTERS

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

Created and Published by:

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

Editorial and Advertising Enquiries:

- [editor@teatimewithtesters.com](mailto:editor@teatimewithtesters.com)
- [sales@teatimewithtesters.com](mailto:sales@teatimewithtesters.com)

Lalit: (+91) 8275562299  
Pratik: (+49)15215673149

This ezine is edited, designed and published by **Tea-time with Testers**. No part of this magazine may be reproduced, transmitted, distributed or copied without prior written permission of original authors of respective articles.

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

# Editorial

## #WeAreWithYouParis and everyone else having testing times

I was around when 26/11 attack happened on Mumbai. World was just recovering from the bitter memories of it and we failed our innocent kids in Pakistan last year. We prayed and hoped it will all end there but recent incident in Paris says that world is going through a phase where either all bad is coming to an end or the good.

I am not an expert on issues related to world peace but it fills my heart with pride when we publish our each issue filled with contributions from testers across the world. If individuals with different thoughts, views and opinions can co-exist and can contribute to the craft of testing, is it really hard to live along with people having different beliefs and to contribute to the betterment of humanity?

I don't think so. All we need to have is a desire to make things work. I wanted to share some interesting things we are working on but then I felt that we all have a bigger problem to care about at this moment. And that we all should do our bit to make things better.

I am doing my bit by dedicating this issue to the victims of Paris attack, to everyone else who is going through tough times and to the world peace, once again.

*May every being be happy, May every being be free from all diseases, May every being feel and appear blissful, May there not be trace of sorrow with any being, Oh God, Bless us, so that we may live together, we be nourished together, May we study the brilliance of your knowledge together leading to the self-unfoldment, without any sense of pride, prejudice or ego.*

*May peace prevail along our bodily and mental level.*

*May peace prevail along all the celestial bodies in the cosmos.*

*May peace prevail along our journey to self-realization.*

*Aum Shanti.*

Sincerely Yours,



- **Lalitkumar Bhamare**  
editor@teatimewithtesters.com  
@Lalitbhamare / @TtimewidTesters



# QuickLook



# All About Software Tools

## Editorial

### What's making News?

### Tea & Testing with Jerry Weinberg

### Speaking Tester's Mind

Testing Skills: Abductive Reasoning – 14

Can misused DevOps kill Testing? – 17

### In the School of Testing

IT-Compliance for Test Data Management – 22

### T' Talks

Leadership Lessons from Maasai Mara – 29

### Family de Tea-time with Testers



## GOOGLE IS TRYING TO FIX BIGGEST PROBLEM WITH ANDROID

Google is negotiating deals with chipmakers to help solve fragmentation, one of the biggest problems with Android, The Information reports. By working directly with chip partners, Google is looking to create hardware that will work seamlessly with Android, just as Apple products do with iOS.

The fragmentation of Android is a real issue for Google, phone makers, and users, who have to deal with outdated hardware, poor security, and a sub-standard experience. According to one estimate, there are over 18,700 distinct versions of Android compared to Apple's five or six.

Google is seeking a more Apple-like approach, combining core functionality of the phone onto one chip. In the negotiations, Google asked chipmakers for a more sophisticated camera unit, improved on-board memory, and more power efficiency.

There could be a snag: The profit margins of Android handset makers, from Samsung to HTC, are already tightly squeezed and so the companies are more than willing to go with what works rather than what excels.

If Google manages to solve this conundrum, however, it could lead to an overall better Android experience for users that would place high-end Samsung, HTC, and Huawei devices in line with the iPhone.

**More news? Visit [Latest Software Testing News!](#)**

# Tea & Testing



with

# Jerry Weinberg

## **The Courage Stick**

### **The Fraidycat Formula**

To learn to look like a great spirit, you must understand how fear influences people to act in certain ways. I call this the Fraidycat Formula:

If your fear of doing A is greater than your fear of doing B, then you do B. For instance,

- If your fear of flossing is greater than your fear of tooth decay, then you don't floss.
- If your fear of writing is greater than your fear of missing the schedule, then you don't finish that report.
- If your fear of marketing is greater than your fear of poverty, then you don't make that phone call to a prospective client.

There are three different strategies for putting the Fraidycat Formula to work for you:

- You can increase the fear of B. Think about the most horrible toothache you had, and the most awful and expensive dental treatment, and you may find yourself flossing regularly.
- You can reduce the fear of A. Stamp the word "DRAFT" on the each page of the report you're writing and plan to give it to your client "for review only"—and you may find you can write easily, without fear.

- You can find a C—a third action you can take to accomplish what you're really after, an action that comes with a different load of fear. Instead of phoning the client, you drop by the restaurant where you know she eats lunch every day, and meet her face-to-face, by "chance." Or you send a letter, or an e-mail, or a telegram, or smoke signals—whatever you fear less than making a phone call.

General George Patton once said that courage is fear holding on a minute longer. Use your Courage Stick to calm you and remind you to use that minute well—finding every available resource. So let's examine some important situations in which a consultant can use that minute well and appear to act courageously.

### **Discussing the Indiscussible**

Over the years I've come to believe that the key moment in a relationship occurs when one or both partners feels there's something that can't be talked about. This could be one thing, or many things, and for a lot of different reasons. When that moment arrives, the one thing that must happen is that the two partners talk about that thing.

As a consultant, one of the most important things I can do to improve relationships is get that indiscussible subject out and on the table—but that seems risky. When I begin to fear this risk, I remind myself of the terrible consequences I've seen when the partners don't discuss the taboo subject.

For instance, I was asked to work with Bill and Sherman, the co-developers of a software product who weren't talking with one another. In this case, the thing that Bill and Sherman needed to talk about was that the Sherman didn't want to talk to Bill. He didn't even want to talk about the fact that he didn't want to talk to Bill, so I decided to approach the subject indirectly, to reduce risk. I went to see Sherman and showed him my Courage Stick, letting him hold it and feel its smoothness.

"It's nice," Sherman said. "What is it?"

"It's my Courage Stick. I brought it with me because I wanted to talk to you about something and I was a little afraid you might not like it."

"You, afraid of me? I've never seen you afraid to say anything."

"Oh, I get scared about lots of things I need to say, but my Courage Stick reminds me that there are also fearful consequences when important things aren't said."

"Like what?"

"Like if I don't tell you what will happen to your company if you and Bill don't talk about some essential subjects. Like how you're going to have a product that sucks, and how everybody is going to infer that Sherman is a lousy software architect."

You see, I didn't know why Sherman was afraid to talk to Bill, but I knew that this would tap into one of Sherman's greatest fears and change the Fraidycat Formula. I never tried to get Sherman to admit that he was afraid of talking with Bill, and after a little more coaxing, I led him by the elbow down to Bill's office. I stayed for a while to act as referee, but soon Sherman's fear was a thing of the past. Later, Bill told me that it took a lot of courage for Sherman to approach him. I didn't bother to correct his impression.

## Not Giving Answers People Want To Hear

In approaching Sherman, I used the fact that it's easy to tell people what they want to hear. We are very skilled, and trained in school and at home, in giving answers other people want to hear. Luria, the Russian psychologist, tells marvelous stories about his interaction with the Russian peasants who didn't have this kind of training.

He tried to give them "intelligence" tests, such as a picture of three people (two adults and a child) to which he asked, "Which one doesn't belong with the others?" The peasant answers, "They all belong. The adults are needed to work, and the child to run errands, otherwise the adults would have to stop working."

These stories remind me of the responses I get from clients when I try to tell them something that doesn't fit their models of "reality." I suggested to one client that he needed to slow down the front-end work of developing software—to take more time discovering what problem he was really trying to solve. He replied with a puzzled look, "Apparently you don't understand the realities of software development. I've measured our development process and we spend 68% of our time just in debugging the code. So where are we supposed to find time to do all this front-end work you're preaching about?"

Clearly, he wants me, the expert, to confirm his erroneous model—a model that ignores the relation between his rush at the beginning and his long testing cycle at the end. And just as clearly, I won't be doing my job as a consultant if I tell him what he wants to hear. In many of my consulting assignments, I have to be the bearer of bad news—news that the regular employees don't have the nerve to report to the boss. You might think that this takes courage, but all it really takes is another calm application of the Fraidycat Formula.

"You're right," I say—starting with what he wants to hear. "It would be silly to add more time to the front-end—unless we can show that it will save time in the place you need it most, the back-end testing. If I can do that without disrupting any existing projects, I know you'll give me a chance to demonstrate that to you?"

Of course, he didn't want to hear that he was wrong, but he especially didn't want to hear that he wasn't the kind of guy to give people a chance. So, I got my demonstration, and he got a new model of how things could be done.

## Not Always Going By The Book

When you're afraid of doing what's right, one of the easier escapes is to do what's conventional. As Loftus' Law states:

Some people manage by the book, even though they don't know who wrote the book or even which book it is.

The Fraidycat Formula explains why this should be so. If your fear of doing A is greater than your fear of going by the book, then you go by the book. And since people are almost never punished for going by the book, the incentive for doing A must be very large, or the fear of doing A must be very small. That's why most "resistance" to A is, in fact, merely a question of safety, or lack of it. This is why startup companies seem more courageous—they don't have the safety of a book to fall back on. And that's why you, as a consultant, can help A to happen by making it less dangerous than going by the book.

Nobody ever needs a consultant to operate "by the book." They only call on you, with your high daily rates, when the book didn't work. So your Fraidycat Formula is different. If you go by the book, they won't need you, and you'll lose the contract, which is a bigger fear than almost anything else you might suggest. This is the psychological basis of Marvin's Fourth Great Secret, as revealed in *The Secrets of Consulting*:

Whatever the client is doing, advise something else.

In other words, for a consultant, this is the cowardly—that is, the safest—thing to do.

### **Dealing with Impossible Demands**

Sad to say, however, there's no completely safe strategy that applies in all cases. If there were, then that strategy would become "the book" for consultants—and then there would be no reason to pay us such high fees. Sometimes, the client puts on so much pressure to conform that the risk of deviation swings the Fraidycat Formula in the other direction.

Johnny, a computer consultant I knew, called me for help. He had recommended against buying a software package, then agreed to help when his client went ahead and bought the package anyway. Now they have the package, and it's not working out—and he was being blamed.

I couldn't help Johnny much at this point, but his Courage Stick could have helped him earlier. He had the courage to tell them what they didn't want to hear—that the package wouldn't work for them—but he lacked the courage to be congruent with his earlier statement when they asked him to help. What he needed to say was, "I'm sorry, but I already told you that the package wouldn't work for you. What kind of consultant would I be if I now backed down from what I said before. But I will be glad to help you find some other solution once you discover that what I said was true, if you still want me."

But Johnny had been afraid to say this, for fear of looking uncooperative. Now that he's seen the result of acting incongruently, the next time his Fraidycat Formula may produce a different result. As Emerson said, "A great part of courage is the courage of having done the thing before." And, he might have added, "and discovering what you really need to be afraid of."

Those of us who have had the experience of yielding to impossible or unreasonable demands have learned that there's no peace of mind if you don't have what it takes to act on your convictions. The stink of cowardice is difficult to wash out of your reputation, and impossible to cleanse from your self-esteem.

### **The Courage to Examine Yourself**

Although courage may not really exist, what appears to be courage produces great quotes. Winston Churchill said, "Courage is the first of human qualities because it is the quality that guarantees all the others."

In terms of my assessment of what's behind the appearance of courage, this would read; "The ability to act on calm and correct assessments of risks and rewards is the first of human qualities because it is the quality that guarantees all the others."

Thought my version doesn't sound as snappy as Winston's, perhaps my expression is more helpful. Most of us don't feel like the heroes invoked by the Churchills of the world, but this ability to use your Courage Stick really does "guarantee" all the other tools. What good is your Golden Key if you know what doors to open, or close, but are too fearful?

What good is a full Wisdom Box if you cannot bring yourself to be guided by its contents? No, it's always better to be a do-something than know-everything.



# 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](mailto:hardpretzel@earthlink.net) or on twitter [@JerryWeinberg](#)

Widely acclaimed as a consultant's consultant, Gerald M. Weinberg builds on his perennial best-seller *The Secrets of Consulting* with all-new laws, rules, and principles. You'll learn how to fight burnout, stay curious, understand your clients, negotiate effectively, and much, much more.

Consultants need more than technical skills—they need self-awareness and a strong set of personal abilities. Weinberg helps computer consultants identify and strengthen each aspect of their performance using a "consultant's tool kit" of seventeen memorable symbols. He devotes a chapter to each of these symbolic tools, from *The Wisdom Box* to *The Fish-Eye Lens* to *The Oxygen Mask*. If you aspire to be successful consultant, **this book** is for you.

Know more about Jerry's writing on software on [his website](#).

## MORE SECRETS OF CONSULTING



Gerald M. Weinberg  
Award-Winning Author

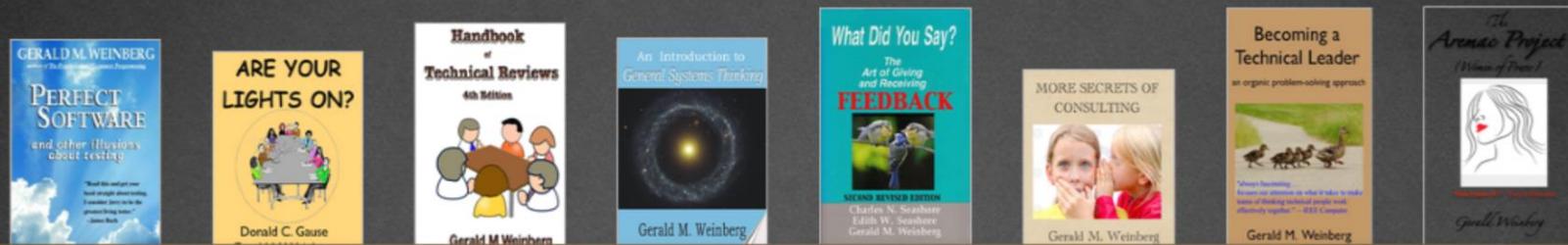
TTWT Rating: ★★★★★

# 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...



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

# \$49.99!

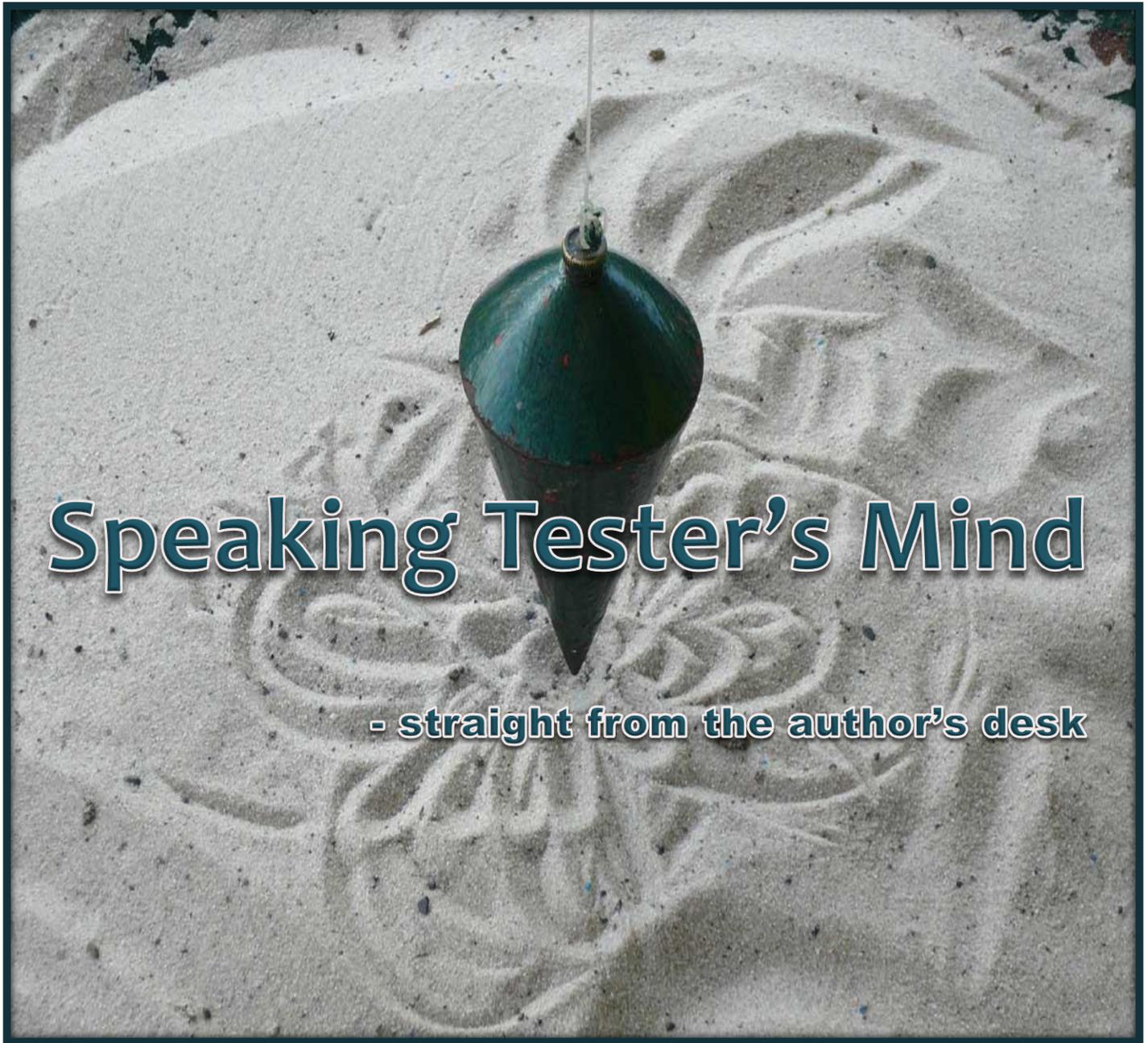
Buy the bundle now!

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

The 8 books are as follows:

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

Know more about this bundle



# Speaking Tester's Mind

- straight from the author's desk



# Testing Skills

## Part 1: Abductive Reasoning

- by John Stevenson

This is the first in a series of hopefully short articles which looks at skills and techniques outside the traditional testing that can be useful to those who practice testing.

Future topics planned include:

- Influencing by listening
- Note Taking
- Leading teams
- Persuasion and how to sell
- Speaking the language of business
- Remote teaching experiential style
- Going beyond the model

If you can think of any others that would be useful please let me know.

I am making these public, since by writing it down and making it public I am committing myself to do it. That is your first tip in this article, if you want to commit to doing something which you keep putting off, writing it down and make it public.

## Abductive Reasoning.

Abduction came about from the work of [Jo Reichert](#), in this work Reichert came up with another cognitive logic process to describe discovery when the researcher encountered surprising findings in the data. He called this "a cognitive logic of discovery". Before this there were two types of reasoning in common use, 'inductive' and 'deductive'.

- **Inductive** - Making generalized conclusions from specific observations
- **Deductive** - Proving or disproving a theory from observations (scientific method)

Abductive reasoning is an important process for those involved in testing. The majority of time when we are testing we discover surprising behavior in the software. This normally makes us rethink our theories of how the software works and as such we begin to re-evaluate our understanding of the software. We create a new rule, or test idea to further investigate the surprising element of what we have just tested. This is key within grounded theory; our thoughts about the software and how it behaves change as we explore the software more. How we report these surprises and the behavior of the software is crucial to the value that testers provide to a project.

*"There are two strategies involved in abduction, both of which require creating the conditions in order for abductive reasoning to take place"\* (Reichertz, 2007: 221).*

*The first is a 'self-induced emergency situation' (Reichertz, 2007: 221). This means that in the face of not knowing what to make of a surprising finding, rather than dwelling on the infinite number of possibilities, the analyst puts pressure on themselves to act by committing to a single meaning.*

*The second strategy is completely antithetical to the first. It involves letting your mind wander without any specific goal in mind, or what Pierce (1931–1935), a key writer on abduction, called 'musement'\* (Reichertz, 2007: 221). "*

[Qualitative Research Methods in Psychology: From core to combined approaches - Nollaig Frost - 2011.](#)

Reichertz makes the following observation about these two strategies.

*"What these two quite antithetical strategies have in common is tricking the thinking patterns of the conscious mind in order to create 'an attitude of preparedness to abandon old convictions and to seek new ones."*

[The SAGE Handbook of Grounded Theory:\(Sage Handbooks\) - Antony Bryant, Kathy Charmaz - 2010.](#)

Testers need to be able to abandon their old convictions and seek out new ones. This is especially important when we are testing software, since our biases and beliefs and previous experiences can influence our decision making. Using some of the methods described in this book can allow us to challenge our thinking about the software and engage in abductive reasoning.

One famous use of abductive reasoning is that used by the fictional detective [Sherlock Holmes](#) by [Sir Arthur Conan Doyle](#). Many people believe, wrongly, that Sherlock Holmes uses deductive reasoning to solve his cases, when in reality he used abductive reasoning.

*"Holmes' method doesn't resemble deductive reasoning at all. Instead, it's much more similar to a form of reasoning known as "Abductive Reasoning"* ["Debunking Sherlock Holmes Myths - Maiza Strange May 2014"](#)

To summarize abductive reasoning is taking your best guess based upon your current knowledge, observations and experiments. These pieces of information may be incomplete but you use your cognitive reasoning processes to form a theory or conclusion. For example:

*"A medical diagnosis is an application of abductive reasoning: given this set of symptoms, what is the diagnosis that would best explain most of them? Likewise, when jurors hear evidence in a criminal case, they must consider whether the prosecution or the defense has the best explanation to cover all the points of evidence. While there may be no certainty about their verdict, since there may exist additional evidence that was not admitted in the case, they make their best guess based on what they know."* ["Deductive, Inductive and Abductive Reasoning - Butte College."](#)

Further reading on abductive reasoning:

- [Abduction, Deduction and Induction in Qualitative Research - Jo Reichertz](#)
- [Abductive Thinking and Sensemaking: The Drivers of Design Synthesis - Jon Kolko](#)
- [Abductive Reasoning - Wikipedia - 2015](#)



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.



# Can misused DevOps kill Testing?

- by Ruslan Desyatnikov

There's been some buzz in recent years within software development circles. A methodology has been thrown around and while some praise it, others sometimes cringe. Maybe you've heard the term once or twice, maybe you've formed an opinion, and maybe you still have questions. Of course we're talking about DevOps. In particular, we're talking about how DevOps may affect Quality Assurance.

The manner in which the DevOps methodology is frequently misused has led to some unintended negative consequences. Done correctly, DevOps can have a staggering positive change in a company's software development lifecycle. But done incorrectly, as it often is, it can have the opposite effect. Misused DevOps methodology can lead to developer burn out, undue stress on QA professionals, and shoddy customer-facing products. The scariest part for those of us in QA is the misguided notion that the DevOps methodology will eliminate the need for dedicated QA.

## What is DevOps?

First, a quick primer on our subject matter. Ask ten software professionals what DevOps is and you're likely to get ten different answers, and this is where the problem starts. At its simplest definition, using a DevOps method means that previously isolated teams such as development, QA, Business, and Operations communicate more frequently, more in depth, and work together in a collaborative manner and share responsibility for the end product. This isn't unlike other methodologies out there, such as Agile. And like similar methodologies, DevOps aims to help teams put software and services into production faster and with higher quality.

With a stress on communication, integration, automation, and collaboration, teams use tools and policies to facilitate faster delivery with increased quality. Many teams aim for continuous improvement via extremely fast release cycles with smaller, incremental changes that pose less risk due to affecting such small parts of the system. There's certainly nothing wrong with that at all, isn't that what we all want?

The split between good DevOps and bad DevOps seems to happen when teams decide just how to achieve the speedy delivery they want.

### **Misuse #1 – The Jack-of-all-trades approach**

Some teams misconstrue DevOps to mean that each member of the team plays multiple roles. The developer is also the DBA and the build manager, for instance; the BA is also the QA person, so on and so forth. This kind of 'jack-of-all-trades' approach is necessary for small startups who don't have the capital to hire larger teams. But it means that corners are cut, individuals are stressed to the max, and each different task that a person branches out to perform is done with decreasing quality. People specialize for a reason – one person cannot know and do everything well for an indefinite period of time. Continuing this kind of process as the company grows and the demand increases will only rapidly burn out team members. Not to mention that continual task switching actually slows people down and costs both time and money.

Choose the right person for a specific job. It's to any team member's advantage to have multiple skillsets – it makes them more employable as an individual and more indispensable as a team member. It's also great for the team and business to have team members who can perform multiple roles when the need arises, but it should not be the preferred state.

### **Misuse #2 – The release-too-fast approach**

Some teams release multiple times a day and such a rapid release schedule inevitably means cutting corners. As those of us who have been in QA for a while know, QA is often a big target for such cuts. It seems as though we've only just gotten to the point of full acceptance of how valuable effective QA is for a team, and now those managers who never really bought into it are using DevOps as an excuse to cut QA again. It's not overt since they do claim that they desire quality, but they just don't allow for the time to actually achieve it. The time constraints DevOps methods can put on teams means that the meticulous manual testing just isn't viable.

This speed at the cost of testing and quality assurance does not result in a quality end product. And, releasing poor quality software – no matter how quickly – is not beneficial for any business. While DevOps methods can aid in speeding up releases, it needs to be tempered with equally efficient quality assurance.

### **Misuse #3 – The automate-everything-approach**

The major push now is to automate, automate, and automate. Unit tests, functional tests, even delivery. Again, in theory there's nothing wrong with automation, but in practice every single aspect of development and QA cannot be automated. Doing so leads to removal of the human element of actually using and exploring a system the way a human is apt to do. Removing that aspect can lead to glaring defects that end users will ultimately find.

Automated tools can merely check code, they can't actually test. They can't problem solve, they can't explore and find different paths, and neither can they determine aesthetics or usability. While testing should always be supported by tools, it still requires a human element. Testers need tools and should know how to use them and use them well, but these tools can never replace the tester despite how badly

a manager may want them to. Meaningful exploratory, human testing will always be necessary. But there will continue to be managers who misconstrue DevOps to mean replacing the testers with an automated tool that simply checks code.

## Final Thoughts

The deeply collaborative, quicker released, and higher quality products that DevOps aims for simply cannot be done without adequate testing. Properly used, DevOps requires more testing, not less, but the testing itself needs to be more efficient. Without a doubt automation is necessary and testers need to adapt their skills to a faster paced world.

While all testers really should have some level of coding skills, we can't and shouldn't assume that the future of all quality assurances lies in testers becoming programmers. Developers aren't the only one expected to multi-task, testers need to as well. It would help to learn some architecture, some coding skills, and the business aspect. The more that a QA professional can do, the more beneficial they are to any team, but they're still needed as testers, not developers.

QA professionals should always keep their skills up to date and keep an eye to future needs in this ever-changing world of technology. But should they be worried their jobs disappearing? No. They should learn to automate, keep their skills fresh, maintain the ability to adapt, and understand that a human tester will always be needed. At least until SkyNet is created.



Ruslan Desyatnikov is Founder and CEO at QA Mentor, Inc. He brings nearly 20 years of Quality Assurance, Quality Control, Process Improvement and Software Testing experience.

He is responsible for growing QA Mentor as a trusted partner for software testing, independent verification and validation, and strategic consulting for client base around the world. Contact Ruslan via LinkedIn <https://www.linkedin.com/in/ruslandesyatnikov>



Love to  
Write?



Write For Us

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

*Send your articles to [editor@teatimewithtesters.com](mailto:editor@teatimewithtesters.com)*

A photograph of a classroom scene. In the foreground, the backs of several students' heads and shoulders are visible as they sit at desks. They are all raising their right hands, with some pointing their index fingers upwards. The students are wearing light blue, red, orange, and green shirts. In the background, a dark chalkboard is visible with some faint, illegible writing on it. The entire scene is framed by a thick black border.

# In the school of Testing

*for your better learning & sharing experience*

# IT – Compliance for Test Data Management

- by Rozalia Noga



## Introduction

*"Life is a continuous exercise in creative problem solving"*

- **Michael J. Gelb**

We now live in a global world and to possess technical devices is a matter of course. Any technical device contains at least one software program or more. Users expect the software to be error free and expect a guarantee of quality. Therefore, **software quality** is an essential factor, because it is not tangible.

Before a software product is released in market, it has to be tested thoroughly to ensure its quality. To achieve the ultimate success criterion for **high software quality** the availability of good **quality - test data** is equally essential.

Today, data is collected, recorded, managed and archived. Data has become an economic "good". Many firms contain an incredible amount of data sets. Such data can rather should be used for software testing. The companies want to protect themselves because confidence in the software product and at the same time in the company plays a large role. To produce high-quality software, companies sometimes need more than an IT department and quality assurance teams. They need a group in the form of **IT - Compliance** which operates and controls the test data management.

This article shows the relationship between the **IT - Compliance and test data management**. It explains and emphasizes the importance of it.

## **Test Data Management in the company**

*"Have no fear of perfection – you 'll never reach it!"*

- **Salvador Dali**

In a lot of big companies worldwide, every day large volume of data is collected and processed, and they are growing steadily. This data is referred as production data or real data and is usually stored in a central location. For quality assurance and the development of data, data is needed to perform software testing. To this end, the production data must be defined, prepared and provided. The production data is original data and can contain personal information, the persons specified may not be disclosed, and they also need to be protected.

This also means that, before the production data can be released for testing purposes, they go through an anonymization process. This process is presented so that certain contents such as the name and the first name must be made unrecognizable to protect the privacy of individuals. There are several types of anonymization procedures that can be applied, however, they must be selected according to objective criteria and according to the company. The departments must explicitly state what data is primarily of importance to them and what data plays a secondary role. The specific test data needed for testing purposes, also depends on the software under test and from the company's industry/domain. Example: a software to be used for hospitals, will be tested with patient data, etc. It's a great challenge to achieve the balance between the usefulness of the data and the identity protection of persons.

### **Example of test data management and IT – Compliance**

Before the testing process starts, some important aspects / questions should be verified. In addition, there are two challenges that should be managed: (a) judiciary regulations and (b) technical requirements. Both should be checked.

In the next step, the original data should be evaluated to determine what kind of data is needed for testing. In further step, a consultation can be done with IT - Compliance to decide how to transform the data in order to protect the privacy. Furthermore it also important to decide what kind of technology can be used for the transformation. The transformation would be anonymization. The aim of anonymization is to change the personal information of individuals so that the person in question cannot be traced back. In the anonymization process the personal data is actually changed. Real data contains a high level of detail concerning the information of individuals and for this reason must be well protected. It is advisable to make proper choice of tools to be used for anonymization. Only then when the original data and the tools have passed through the review process test data should be generated.

### **Test Data Management in "Real Life"**

The daily work brings many interesting challenges and each employee is trying to find a quick solution. At the beginning of an IT project the software testers and developers are given a time indication to implement the project. In the course of the project, it often turns out that the projects are much more complex than the management has agreed on, and above all, they need a lot more time. The developers and software testers are set under pressure by the project manager to gain results quicker.

In busy work life, efforts are made to meet all the regulations. For software testing, the "**data**" is always "**immediately**" required; therefore the use of production data is very tempting as the testing can be done "immediately" with it. If theoretically however production data is used, we must bear in mind that this procedure is illegal and prohibited by law.

The Department which contains the real data must first find the time to prepare the data to release it later. After this, the process of anonymization is carried out and after several complicated processes, the software testing should begin under such contexts. This extra time is very expensive for the project. The management, the software testers and the developers have the same goal; they all want to complete the project successfully. To expedite the deliveries, rules are bypassed unfortunately and we must make efforts to restrict that.

## ***IT-Compliance***

*"If you never change your mind, why have one?"*

- **Edward de Bono**

### **What does IT – Compliance mean?**

*"IT - Compliance defines the observance of legal and regulatory requirements, which consist of laws, regulations, contracts and standards. The rules related to the use of an IT landscape in the company".*

Recently the term IT - Compliance has evolved to a popular term in the industry. **The goal of IT - Compliance** is to identify and ensure that the requirements are met in terms of IT and thus avoid penalties. Furthermore, however, IT - Compliance means the compliance with internal policies, which each company defines for itself. It should be ensured that the imposed rules are followed and violations are detected immediately. The compliance requirements serve to recognize the risks in the company and to specific reduce these.

Unfortunately, it is not sufficient to introduce an IT - Compliance management in a company, as many divisions are affected by different specifications. It must rather be ensured that the most important IT - Enterprise areas are covered, such as the test data management, which is very important for the software development and quality assurance. The targets applied to the test data management, are to minimize the risk of data management and to enhance the safety and the quality of the test data of the software testing.

### ***Reasons for the introduction of IT - Compliance in the test data management***

The introduction of IT - Compliance based on the test data management is not only to comply with legal and regulatory requirements but also to demonstrate the company's transparency and accountability, concerning which test data was used, and especially for which tests that data was used.

The legal and internal corporate requirements are increasing continuously. In order not to lose track, it is necessary to introduce a continuous process which guarantees an overview. Companies need a policy for dealing with test data, which includes the IT - Compliance requirements and the solution to implement the requirements. This ongoing process should be reviewed regularly in order to protect the company and to continue the development process.

Concerning this process many tasks and challenges need to be faced and managed. The question arises: "How should IT - Compliance be implemented in the test data management? **The answer is: "Calling an IT - Compliance Manager"**.

*To be continued in next issue...*

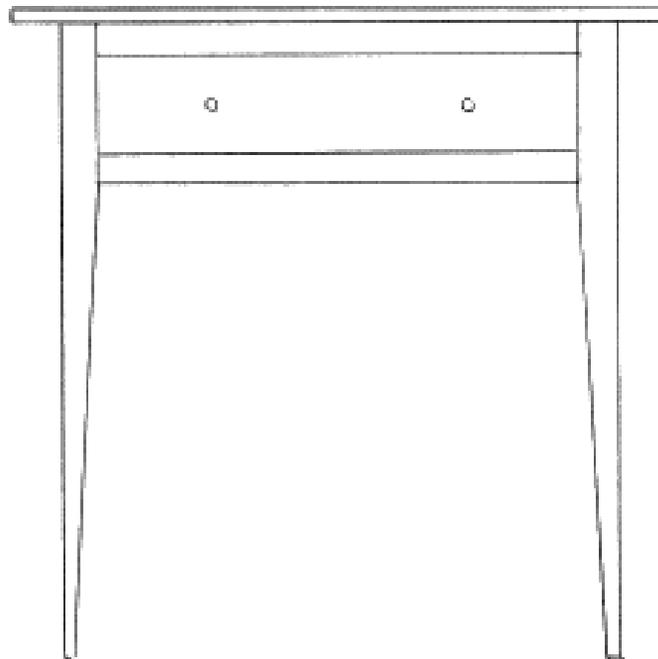


Rozalia Noga works as a Software Test Engineer in scrum environment. As a student she worked for SAP Germany and started there her career as a Software Test Engineer.

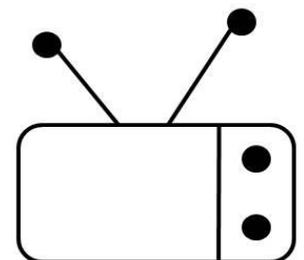
She has testing experience in industry and commerce software, medical software and mobile app testing.

Rozalia is a Certified Tester of the ISTQB Foundation Level (CTFL).

[Back To Index](#) 



# 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>

# T ' Talks



*T. Ashok exclusively on software testing*

## **Leadership lessons from Maasai Mara**

---



A few weeks ago we were holidaying at Maasai Mara, a lovely 1510 sq km game reserve in Kenya. The animals are outside in their natural habitat without a care about the humans who were inside the open top jeeps. Seeing the Big-5 (Lion, Leopard, Rhino, Elephant and Wildebeest) and the others in close quarters was wonderful. Observing their behaviour, mannerisms at close quarters was not only enjoyable, but educational. Here is what I learnt about leadership from these wonderful creatures.

## The Lions

A lion is busy eating the fresh kill of wildebeest whilst a cackle of hyenas are sitting a little distance away patiently waiting for the lion to finish. Once the lion is satiated and moves away to rest, the hungry Hyenas will finish the rest. Their teeth are so powerful that they can crush the bones. In another instance we saw a wake of vultures in fierce fighting to eat the remains of a carcass killed by a lioness. In nature nothing is left to waste with each animal specialised in its contribution to the nature's eco-system cycle. A wildebeest is a big animal and it is a shame to waste anything at all. The Hyenas don't mess with the lion nor do the vultures with any of these animals.



**Leadership lesson: It takes multiple skills to get a job done well. Each facet of work requires varied intelligence and power, understanding this and abiding your time patiently is key to being successful. So in your team, who is the Lion(ess), Hyena, Vulture...? Each one of them is important and there is nothing superior or inferior.**

## The Leopard

A Leopard had dragged up a young waterbuck up the tree, suspended it precisely on a branch with the body resting on the branch and legs hanging down. It then went about methodically tearing up the rear with its canines. Though gruesome and sad, it was interesting to see the leopard periodically adjusting the dead animal on the branch as it progressed methodically with its job of eating.



**Leadership lesson:**

**Leopard is a very powerful animal which with its sheer speed and power can snare an animal. To ultimately accomplish its goal of satisfying its hunger, it must secure the dead animal, which it does by dragging the animal up the tree and then eating it by doing it methodically, tear from the rear. What do we learn this? In addition to having a powerful team, it is necessary to employ powerful techniques and a good process to ensure that the job is well done. Power is best extracted via using good techniques and harnessing the same in a methodical manner.**

## The Giraffe



A herd of giraffes are moving majestically, elegant in their movements as they peacefully chomp on the leaves off the tree top. Giraffe by virtue of its height and its distinctive coat is indeed a majestic animal. They are not shy at all. One of them walked to our vehicle confidently, stopped, looked at us without any fear and then strutted away. The movements were peaceful and seemingly unhurried. On a different note, it is interesting how a giraffe walks- Both the legs on one side go forward and then the next side go forward, very unlike the other animals where front and rear are in sync.

**Leadership lesson: The majestic appearance oozing confidence, fearless attitude, grace and elegance are traits that make a leader successful. And the team will do anything to ensure that the leader stand tall always.**

## The Zebras

The vast African Savannah is filled with Zebras, their skin shiny and beautiful with stripes of black and white. They are an easy prey for the lions. Their excellent sense of sight and hearing coupled with their acute sense of smell enables them to detect danger which they nimbly respond to, by quickly running away. Their legs are so strong that a kick can kill an animal.

**Leadership lesson: To survive/course-correct requires one to continuously measure (aka smell/watch) and respond nimbly to steer to a safe place. Not only is sufficient to know the danger, but it is imperative to have enough power to move away quickly.**



## The Hippos



Hippos relaxing in the water seem to be the most lethargic animals. They seem to spend the day seemingly relaxing in the pool when not eating grass. But do you know that Hippos are the second largest killers of humans?

They are not carnivores, they attack only to protect. Their strong bite can kill humans.

**Leadership lesson: Each one of us possess enormous power. It is dormant most of the time. The key is unleashing it.**

## The Warthog

These animals were the most interesting. Belonging to the same family of pigs, these herbivores were constantly running, seemingly for the sheer joy of it. Enjoying themselves just running across the vast expanse. They stop suddenly only to recommence in a short while. It was sheer joy to see these "pigs with tusks" run despite no threats from other animals.

**Leadership lesson: Explore. Be curious. Enjoy. To unleash your full potential, you need to be unbounded and enjoy the flow.**



## The Wildebeest

Maasai Mara is famous for the crossing where over 1.5M wildebeest cross over from Tanzania to Kenya and vice versa. Our trip was close to the end of 'crossing season'. Keen to see the action up close, we went to the river Mara where the crossing occurs. And there it was, a group of wildebeests on the other side of river deciding on what to do - 'cross or not to cross'. Upstream nearly a kilometer ahead, a crocodile was basking on the rock while a couple of them were gently floating on the water, just their tops visible on the surface, the nemesis for these crossing animals.



The wildebeests are still trying to decide, one of them pushes the other into the river and that guy does not go any further, in fact he turns back. Then they stand still for a few minutes and then another one goes further into the water and then turns back. Suddenly they run along the river, some in the water and some on the shore, coming to a dead stop. After some time. They do nothing but stand still possibly for one of them to decide. None of them take the next step. One of them at the end on the land is I guess fed up. He decides that he has had enough of this indecision and decides to go back into the land and not cross the water. He climbs the mud embankment into the high land and as he climbs, a few follow suit. One by one the remaining ones follow. The decision has been made, they are not going to cross. The animals that were standstill a moment ago due to indecision are rapidly moving away following the first one to the high land.

**Leadership lesson: We all suffer from indecision and do possess herd mentality. A good leader takes decision and confidently goes ahead and the team follows. Not all decisions may indeed be right, but taking a decision is indeed superior to not doing anything.**

I hope you enjoyed the educational tour of the Savannah!

Want to be a successful leader? Let the animals be your guide.

Assemble varied skills to build a great team. Let good techniques and methodical action unleash the power. Watch progress and steer continuously. Be confident. Make decisions. Enable each individual to unleash their full potential. And enjoy the journey.

Hakuna Matata!



**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](mailto: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.



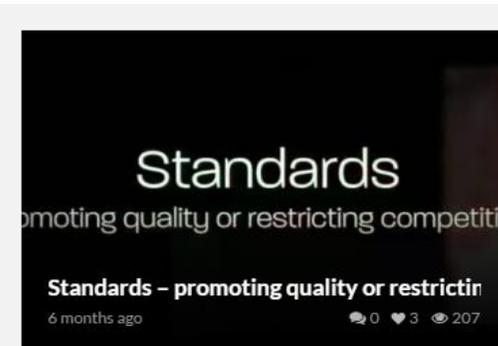
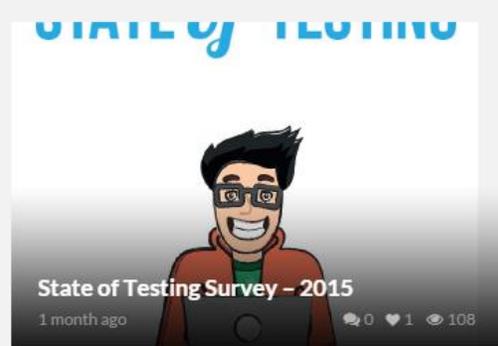
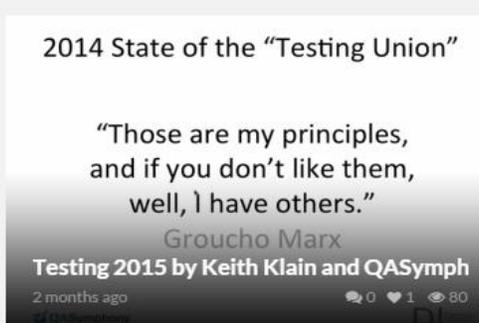
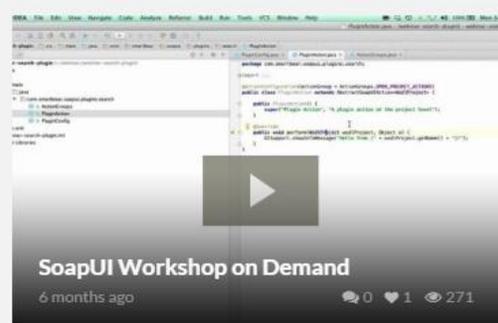
We enjoy putting our experience to your service. Our know-how allows us to provide consultation services for projects at any stage, from small up to super-large. Due to our range of expertise we can assist in software architecture and COTS selection; review of software designs; creation of testing suites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

look us up: [www.testomaton.com](http://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





[www.talesoftesting.com](http://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](mailto:sales@teatimewithtesters.com) to know more.





Every Tester

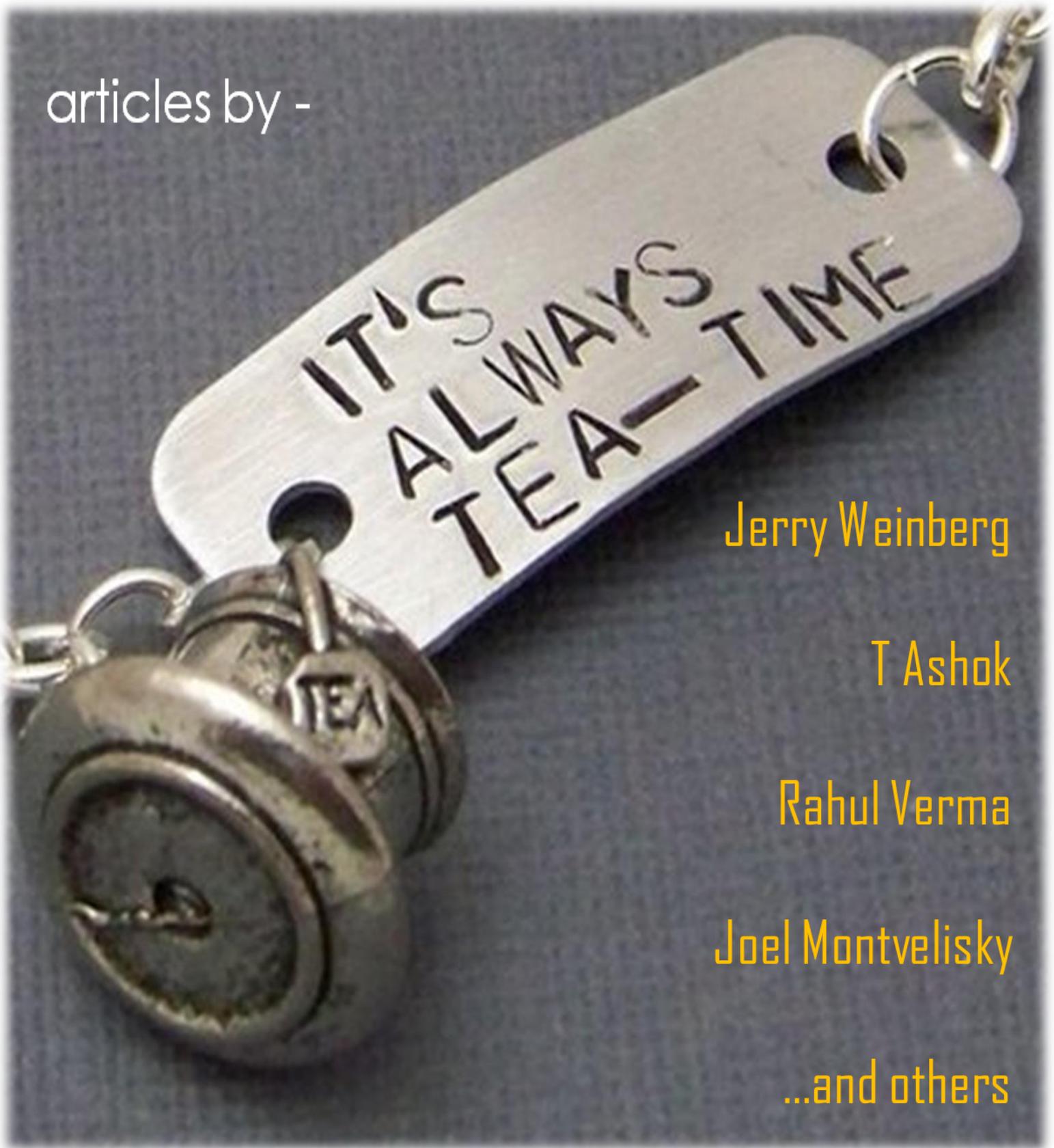
who reads **Tea-time with Testers**,

Recommends it to friends and  
colleagues .

**What About You ?**

# in ne>xt issue

articles by -



IT'S  
ALWAYS  
TEA-TIME

Jerry Weinberg

T Ashok

Rahul Verma

Joel Montvelisky

...and others

# our family

## Founder & Editor:

Lalitkumar Bhamare (Pune, India)

Pratikkumar Patel (Mumbai, India)



Lalitkumar



Pratikkumar

## Contribution and Guidance:

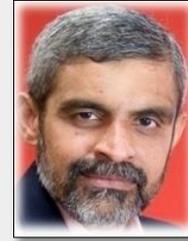
Jerry Weinberg (U.S.A.)

T Ashok (India)

Joel Montvelisky (Israel)



Jerry



T Ashok



Joel

## Editorial | Magazine Design | Logo Design | Web Design:

Lalitkumar Bhamare

Cover page image – HDWallpapersrocks.com

## Core Team:

Dr.Meeta Prakash (Bangalore, India)

Unmesh Gundecha (Pune,India)



Dr. Meeta Prakash



Unmesh Gundecha

## Online Collaboration:

Shweta Daiv (Pune, India)



Shweta

## Tech -Team:

Chris Philip (Mumbai, India)

Romil Gupta (Pune, India)

Kiran kumar (Mumbai, India)



Kiran Kumar



Chris



Romil

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

To get a **FREE** copy,  
Subscribe to mailing list.

**SUBSCRIBE**

Join our community on

**facebook.**

Follow us on - @TtimewidTesters



[www.teatimewithtesters.com](http://www.teatimewithtesters.com)

