

## Editorial

Recently, we have seen so many natural occurrences or processes; be they floods or cyclones, forest fires, droughts, or a raging volcano.

All these natural phenomena are generally called disasters or calamities by us, humans. Yes, life on the planet gets affected. But what could be the reasons? What can we do?

The shifts in the earth's atmosphere, environment, and structure could be possible reasons. Climate change and global warming, we all understand are creating havoc. So many changes are happening due to the decrease in green cover and the over population we see around us.

In the beginning of the twentieth century the world had a population of only around 1.6 billion people. Today we have grown to 6.7 billion, which is an enormous and unprecedented increase. So wherever there are any small changes in nature like earthquakes, tsunami, floods, etc, human beings are badly hit.

Advancement in medical science, to a large extent, has enabled us to take death into our hands. So it is all the more important that we also take birth into our hands. As a nation, we need to achieve a reduction in population by at least one third in the next 50 years. We must also ensure that a minimum of 33% land is under green cover. If these two things happen we can largely control some calamities.

Plant a Sapling and nurture it to become a Tree!

*Editorial Team*

## What's Inside

Editorial	1
News	1
Upcoming Conferences	2
Article	3
FSTB	6
Interview	8

## Product News

TestersDesk.com is a free web-based/online toolkit for the testing community enabling Better Test Design and Faster Test Data.



It features Test Design tools such as Pairwise Test Case Generator that can be used to reduce the test combinations, a Subset Test Case Generator to select subsets of test items of a specified size from a larger set, among others.

From the perspective of Test Data generation, the free toolkit provides an array of tools to generate files (using Size-based File Generator), Common Test Data Generators like Language String Generator and Syntax-based String Generator in which the tester can provide the syntax/format of the test data, and many other tools alike.

The outputs are given out on the screens as well as through CSV/XML files that can be downloaded. Many testers use the outputs in manual testing as well as in automated testing (by integrating the CSV files we give with their data-driven scripts developed in QTP, Sahi, etc).

TestersDesk.com has won several awards and continues to strive in helping the testing community with new tools.

Testers can register for free at <http://www.TestersDesk.com> and use the web-based application as needed.

## News Briefs

VMLogix Partners with Red Hat to Expand Virtualization Platform Support

[http://www.vmlogix.com/index.php?option=com\\_events&task=view\\_detail&agid=40&year=2009&month=11&day=02&Itemid=194](http://www.vmlogix.com/index.php?option=com_events&task=view_detail&agid=40&year=2009&month=11&day=02&Itemid=194)

TrueSight Load Test Adapter Automates and Simplifies HP LoadRunner Web Application Performance Validation

[http://www.coradiant.com/news/271009\\_adapter.htm](http://www.coradiant.com/news/271009_adapter.htm)

Seapine Software Plays Critical Role in DOE Cleanup Project

<http://www.reuters.com/article/pressRelease/idUS135467+27-Oct-2009+BW20091027>

Agitar testing product will write code

<http://www.javaworld.com/javaworld/jw-08-2006/jw-0814-idgns-agitar.html>

## Random Collection: Blogs on Testing

### Test This Blog

Eric Jacobson's Software Testing Blog

<http://www.testthisblog.com/>

### Test Notes

<http://geekswithblogs.net/srkprasad/Default.aspx>

### Collaborative Software Testing

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

### Software Testing Blog

<http://venkatreddyc.wordpress.com/>

### Testing Reflections

QA Blog Aggregator

<http://www.testingreflections.com/>

### Ronpih's weblog : Software Testing Blogs

<http://blogs.msdn.com/ronpih/pages/software-testing-blogs.aspx>

### Testing Geek

<http://www.testinggeek.com/index.php/more-on-testing/125-software-testing-interesting-testing-blogs>

### Software Testing Zone

<http://software-testing-zone.blogspot.com/>

### U Test

<http://blog.uteest.com/>

### QA Software Testing Blog

<http://qasoftwaretesting.com/>

## Upcoming Conferences

- **EuroSTAR 2009**

Nov 30 - Dec 3, 2009

Stockholm, Sweden

<http://www.eurostarconferences.com/?id=2>

- **Agile Development Practices Conference**

Nov 9-13, 2009

Orlando, Florida

<http://www.sqe.com/agiledevpractices/Default.aspx>

- **STAREAST 2010**

April 26-30, 2010

Orlando, FL

<http://www.sqe.com/StarEast/Splash.aspx>

- **QUEST- Quality Engineered Software and Testing Conference Dallas 2010**

April 19-23, 2010

Dallas, Texas

<http://www.qaiquest.org/dallas/>

## File Fuzzing - Employing File Content Corruption to Test Software Security

by **Rahul Verma**, *QA Technical Lead*,  
McAfee Software (India) Pvt Ltd, Bangalore, India

*With an experience of 7 years in the industry, Rahul has explored the areas of security testing, large scale performance testing and database migration projects. Currently, he is a QA Tech Lead in AVERT Labs at McAfee India. He got the Testing Thought Leadership Award at TEST2008 conference for his website [www.testingperspective.com](http://www.testingperspective.com), along with the Best Innovative Paper Award for his paper on design of Fuzzing Frameworks.*



*(Continued from previous issue of this newsletter)*

### 6. File Fuzzing – Putting TIGEMA on Job

Fuzzing is easier understood if we split the process into steps. The fuzzing process can be remembered with TIGEMA mnemonic, wherein:

- T – Target(s)
- I – Input Vectors
- G – Generate
- E – Execute
- M – Monitor
- A – Analyze

Each of the above describes a distinct step in the process of fuzzing. One or more of them might work in conjunction or parallel to each other. Figure 1 gives a visual snapshot of these steps in conjunction with each other. Following sections discuss the steps in detail as applicable to file fuzzing:

#### • Identify Targets

This step can be approached in two ways in file fuzzing:

- o Identify the software to be fuzzed. Identify all input files that it takes. Shortlist the formats to be fuzzed. Fuzz them.
- o Identify the file format to be fuzzed. Identify all softwares that support the identified file format. Shortlist the softwares to be fuzzed. Fuzz them.

The first approach is typically employed when one is dealing in testing the security of the product on which one is working. Security researchers who find vulnerabilities in

## Article...

third party softwares employ both of the above approaches based on context.

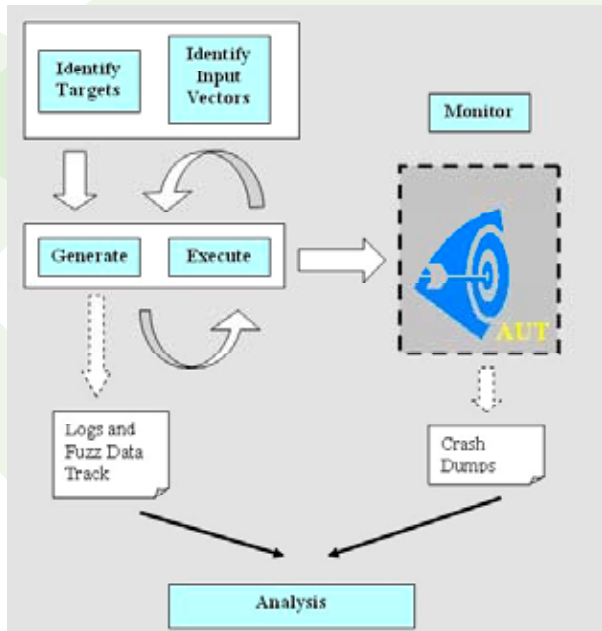


Figure 1: Steps in Fuzzing

#### • Identify Input Vectors (Files)

In case of file fuzzing the type of input vector is a file but there can be multiple files that one wants to fuzz. E.g. an anti-virus would scan almost all existing known formats. So, when fuzzing anti-virus, one would typically fuzz a mixed set of formats. There are situations when you fuzz different file formats for the same software but each of them has separate purpose e.g. a configuration file, a license file, the primary file format (document/media file) etc.

#### • Generate Fuzz Data

This is the step where actual fuzzer development comes into picture. Based on the inputs chosen, you make decisions for the kind of fuzzing you want to employ. This governs the quality and quantity of fuzzed data you will receive for the inputs you have identified.

#### • Execute

At this step you send (publish) the fuzzed data (for an input vector) to the target application. This might be a post generation process or it might run along with the generation process.

In the former, you first generate all the fuzzed data and write to an output file and later send this data one by one to the application. You might require a lot of disk space in this case depending on the kind of fuzzing. If you are fuzzing file formats, if the size of

the file is large, you might end up consuming a lot of disk space (or at the worst running out of disk space). In some cases, this option might not be feasible at all.

So, in case of file fuzzing, the latter approach is followed. You generate fuzz data and send it to the application. In case the application crashes, a copy of the data is retained and next fuzz iteration gets executed; else the data is ignored (or deleted if on disk) and fuzzing process is continued. This way, only that fuzz data which is problematic is retained on the disk (in the form of files/database entries etc.).

- **Monitor**

This is done while you are sending the data to the application. This typically involves a debugger being attached to the application right from the beginning of the test. It might also involve monitoring the resource utilization on the box. If there is a crash, the fuzzer should be able to know about it. The debugger takes a dump of the application in case of a crash for later analysis. The fuzzer then launches the application again, attaches the debugger and proceeds to the next fuzzing step.

The fuzzer should have a component which puts a cut off limit on the running time of the application (called time threshold) and monitor the related process. Time threshold helps in killing an application and proceeding with the next test case as a part of normal fuzzing process.

- **Analyze**

The crash dump and the fuzz data that caused it are taken for analysis at this stage. This is typically taken up by a security researcher and/or development team with knowledge of vulnerability analysis.

The software tester's job at this stage is providing the required data to the mentioned team. Based on interest, a tester can learn basic crash dump analysis and be of further help.

## 1. Approaches for File Fuzzing

There are many factors which govern the way file fuzzing will be implemented. Some of the key factors which one needs to consider are:

- **Specific File Fuzzing Versus General File Fuzzer:**

Fuzzing a specific file format can be done using quick scripting with no time spent on designing reusable components, but when one is looking at fuzzing more than one file formats using the same tool, framework design has to be considered and has to be split into classes/functions that can be employed when executing file fuzzing of various sorts.

- **OS platform:**

The type of OS platform has a large impact on the way the tool is designed because of the fuzzer needs to understand how the OS handles process, what are the debugging options available, how resources can be monitored etc.

- **Data Corruption Method - Generation versus Mutation:**

At a broad level, a fuzzer can produce fuzz data in two ways – generation and mutation. In generation, the complete protocol is generated from scratch based on the knowledge of the protocol built into the fuzzer. This requires lot of ground work to be done by reading relevant manuals and analysis. In case no such published data is available, you will have to resort to reverse engineering skills, which most of the times is quite a complex task. The advantage is that you get complete control over the protocol and can get good code coverage.

Mutation is about capturing good data and then fuzzing various sections of data. For this you use a base good file for mutating. The advantage is that you can get started with fuzzing efforts quickly, but one must take care of internal dependencies of the fields and optimum code coverage.

- **File Format- Ignoring versus Handling Internal Dependencies:**

In many protocols there are fields that are dependent on other fields in turn, e.g. they might include length, checksums etc. If you choose to abide by these conditions, the fuzzing process gets a little trickier than otherwise. A suggested way is to build these checks into the fuzzer you build and carry out tests in both ways – breaking the dependencies and abiding by them. This helps to unearth any false assumptions and also making sure that correct parsers are hit (of course by increasing the number of test cases executed significantly)

- **Blind Fuzzing versus Format-aware Fuzzing**

A blind fuzzer has no knowledge of the underlying protocol. It is assigned the task of blindly corrupting or generating a data packet and sending to the application. This is very easy to build but results in wastage of CPU cycles and time in generating and testing data that is out rightly rejected, sometimes, much before it reaches the target. The protocol aware fuzzer is complex to build but is more reliable and result-oriented.

Blind fuzzers are usually tied to the mutation approach and protocol aware fuzzers are tied to generation approach (or to mutation approach while abiding by the dependencies of the fields).

## Conclusion

All in all, file fuzzing (or fuzzing in general) is a good and easy way to test software for security issues. A software tester can further contribute in the area by brushing up skills

on threat modeling for analyzing various input vectors and associated threats, code coverage to check effectiveness of fuzzing tool, core dump analysis to understanding cause of the crashes captured and vulnerability analysis to associate crashes to a possible vulnerability that could be exploited.

Fuzzing should not be thought of as a replacement for other forms of testing. It should be a new form of testing added to the existing tests being conducted.

### Definitions, Abbreviation and Acronyms

#### Fuzzing

Fuzz Testing or Fuzzing is a software testing technique that provides random data (“fuzz”) to the inputs of a program. If the program fails (for example, by crashing, or by failing built-in code assertions), the defects can be noted.”

#### Black Box Testing

Testing an application with little or no knowledge of the underlying implementation

#### Gray Box Testing

Testing an application with knowledge of logic/high level implementation but not of the exact code

#### Threat Modeling

A method of assessing and documenting the security risks with a software application.

#### Vulnerability

A security exposure in an operating system or other system software or application software component

#### Threat

Possibility that vulnerability may be exploited to cause harm to a system, environment, or personnel.

### References

#### Fuzzing: Brute Force Vulnerability Assessment

A book dedicated to the art of fuzzing by Sutton Michael, Greene Adam, Amini Pedram. Published by Addison-Wesley Professional

#### *Building a Fuzzing Framework: A Primer for Software Testers*

Paper written by Rahul Verma (author of this paper) dealing with building a fuzzing framework. Selected for TEST2008 Conference.

#### *Wikipedia: Fuzz Testing.*

[http://en.wikipedia.org/wiki/Fuzz\\_testing](http://en.wikipedia.org/wiki/Fuzz_testing)

#### *Fuzzing.org: Fuzzing Software.*

<http://www.fuzzing.org/fuzzing-software>

## New Certification

### ITB introduces QAMP certification in India. ITB is the exclusive partner in India (National Authority) for QAMP.

QAMP, a new certificate for advanced vocational training of software engineers, supported by iSQI has successfully been introduced in co-operation with and according to the needs of the IT-Industry.

QAMP is targeted towards quality assurance managers, because they require competence in the entire development process. It tests the practical knowledge and the project experience of the certified employee. Theoretical and practical knowledge both have to be updated annually. In addition to a broad theoretical knowledge, there's a need to successfully pass three certification exams:

- Software Test (FL)
- Specified Module (AL)
- Requirement Engineering

#### Four Steps To QAMP

- Software Testing. (ISTQB Foundation Level)
- Specified Module .ISTQB Advance Level Test Manager/Configuration Manager/Project Manager/Secure Software manager/Software Architecture
- Requirement Engineering (IREB)
- 2 years of practical experience

#### Upcoming IREB Examination is on December 20, 2009

For further information please visit our site [www.istqb.in/ireb](http://www.istqb.in/ireb) or mail us at [ireb@indiantestingboard.com](mailto:ireb@indiantestingboard.com)

## From the ITB Desk

From ITB Desk

6

New Affiliates

### New Affiliates

#### Eon Technologies Pvt. Ltd.

Website: [www.eonglobal.com/](http://www.eonglobal.com/)



### Upcoming ISTQB Examinations

- November 15, 2009 at Pune, Hyderabad, Chennai, Trivandrum, Bangalore, Mumbai, Kolkata, Noida.
- November 22, 2009 at Coimbatore
- November 27, 2009 at Madurai
- November 29, 2009 at Cochin
- December 5, 2009 at Indore
- December 6, 2009 at Baroda
- December 13, 2009 at Chandigarh
- December 18, 2009 at Trinulveli
- December 20, 2009 at Pune, Hyderabad, Chennai, Trivandrum, Bangalore, Mumbai, Kolkata, Noida

For Online registrations, please visit [http://www.istqb.in/enrollment\\_form.php](http://www.istqb.in/enrollment_form.php)

*Last date for payment & registration is 10 days before the exam date*

#### New Examination Centre

- Chandigarh

#### Address:

Zimin Software Pvt Ltd  
SCF 111, Phase-X  
SAS NAGAR, Mohali-160059

**Examination date: December 13, 2009**

**Technical Test Analyst Exam** according to the New Syllabus is being conducted on November 15, 2009 for the first time.

### Message from the FSTB President

The French Software Testing Board (Comite Francais des Tests Logiciels in French) is very active and develops itself through word of mouth, without any publicity. We have reached widely and now are considered as an important actor by the software companies, the government and major industries (aerospace, railroad, automotive, banking, as well as retail, services organizations, consulting firms, etc.). After the success of the first French Software Testing Day in 2008, the French software testing board plans the next French Software Testing Day in March 2010. These events are the only French speaking events in France.

**Bernard Homes**

President, French Software Testing Board (FSTB)

### Titbit...

A clock that has only 9's in it.






Having 9s does not make it special.

What makes it really amazing is the fact that... it has only 3 9s in each digit representation....

# French Software Testing Board (FSTB)

## FSTB Fact sheet

Location	The FSTB is located in France and covers French speaking countries that do not have a specific testing board. Based in the south of France, it covers areas such as Paris, Brittany, Marseille, Toulon.
Founded & operational since	The Board was created in 2004, and is operational since 2005.
Certifications awarded	About 800
Language(s) for exams	Three: French, English, and German
Exam fee	Open exams: 250 euros plus taxes; CFTL members and CFTL accredited training providers may have a discount of 20%
Software Testing market	French market is hit by the current economic crisis, but we expect about 8% increase over the next four years
Population	About 64 Million
Capital	Paris
Flag	Blue-White-Red vertical stripes 
Map showing France	
Map of France w/details	
Other info	This link provides a wealth of interesting data: <a href="https://www.cia.gov/library/publications/the-world-factbook/geos/fr.html">https://www.cia.gov/library/publications/the-world-factbook/geos/fr.html</a>
Contact	<a href="mailto:info@cftl.fr">info@cftl.fr</a>



### Srinivasan Desikan

*srinivasan.desikan@gmail.com*

*Srinivasan Desikan is the author of one of the bestselling book “Software Testing – Principles and Practices” which is being used by more than 50 universities across the world. He has been part of large testing and product-development teams, and has been in the field of software testing since 1989. He has occupied a variety of technical and management positions at Novell Inc., Wipro, Talisma, Siebel and Agile Software. He has received one of the biggest engineering awards “Employee of the Year-1999” at Novell for setting up system test team and the lab from scratch. He currently works as Master Technologist for Testing at Hewlett-Packard (HP). He has vast experience in the areas of test automation, test management, test processes and in setting up test teams from scratch. He has delivered talks on testing at international conferences both in India and abroad, such as QAI-India, ASI STAR-2002 (Melbourne, Australia), PSQT/PSTT-2003 (Washington, USA), CSI, BCIC, IEEE, SPIN (Chennai), STEP-AUTO and STeP-IN. He has more than 20 technical publications with magazines such as CTO Forum, PCQuest, Times of India, Deccan Herald, Express Computer, SEA Software ...etc. With two patent disclosures on testing pending to be published, he is currently doing his research in defect detection and prevention mechanisms for mission-critical software applications based on matured automobile industry experiences (Toyota-way). A post-graduate from Pondicherry Engineering College, he is currently serving as honorary advisor to colleges and Universities in course design, delivery and Instruction methods for software engineering and design. As a honorary advisor to companies he helped many of the testing services organizations in setting up and improving their competency and*

*business during late 90's and early 2000. Passionate about software testing and product development, he spends his weekends giving guest/free lectures at universities, colleges and companies to promote the subject. He is the recipient of “Testing Thought Leadership Award” from Test2008.in for his contributions to both academia and Industry.*

**WIT** -- Srinivasan, please tell us something about yourself, your background, your interests and things that you like to do in your free time?

**SD** -- I come from a middle class family and completed my schooling in a government school. After schooling, I was keen on becoming a Chemistry graduate, and had no idea about other fields such as Computer Science. But with my father's insistence, and the encouragement from my college principal, I enrolled for Computer Science. This decision is important because it was the first time Computer Science was introduced as a subject at graduation level. I secured the 1st place when I graduated in 1986.

I completed my post-graduation from Pondicherry Engineering College, and was one among the many students who formed the first batch to graduate from this college. I was “pushed” into software testing in 1989, while many of my classmates ventured into supposedly more lucrative roles as programmers and analysts.

While software testing was initially never a key interest for me, I learned soon enough that testing does provide a quick passport to the IT industry. This interest of mine in software testing enabled me to perform successfully in different roles such as Test Engineer, Automation Engineer, Test Manager, Test Architect, Worldwide Director, Head of Testing and now Master Technologist in testing.

Coming to the question on free time – free time is like white elephant for many of the people on earth. A day has only 24 hours and a week has only 7 days. I try to keep a calendar, in which 5 days are completely dedicated for the organization I work for, 1 day is for teaching at a college about what I learnt from industry and 1 day is dedicated exclusively for time with family. But I don't want to say that I teach only during my free time. This would imply that teaching is a second priority for me, which is really not the case. Rather I would say that I spend my weekends constructively in colleges for two way benefits. Helps me to understand how the book authored by me is used and helps the college in getting industry perspective from my involvement.

**WIT** -- Could you tell us something about your book “Software Testing – “Principles and Practices”.

## Interview...

What inspired you to write this book? How is this book different or unique?

**SD** -- When I started teaching software testing in colleges in late 1990's, there were not enough books or college curriculum in this area. In particular, there were no books published from India that was affordable to poor students. This gave my friend and I the motivation to publish a book that is rich in content and affordable for everyone. Regular teaching on this subject left us with plenty of materials and audio & video recordings and we decided to convert them in to a book. Today, this book is one of the best sellers in its category and is recognized by more than 47 universities across the world. More than the copies sold, it is really motivating to get hundreds of emails from students, professors and professionals on how helpful this book has been for them.

**WIT** -- Do you have plans to write another book?

**SD** -- Definitely No - at least not in the near future. The job of a writer is to ensure that readers are benefited by the content of the book. I would like to spend my efforts in popularizing the topic of Software Testing to more and more colleges and organizations with the book that I have already published, rather write another book.

In my efforts in the future, if I find any avenues or topics that are not covered by books in the market, then perhaps I would think of writing another book.

**WIT** -- What in your opinion is considered to be "passionate" about software testing, that you spend your weekends giving guest/free lectures at universities, colleges and companies to promote the subject? Which subjects do you concentrate on? What sort of audience normally attends these? Do you think you have been getting enough support from people in this Endeavour?

**SD** -- Being passionate is always contagious factor. My passion for software testing is also that way and I get my constant energy from my colleagues, friends, students and teaching staff. In my opinion, there is lot to learn in this subject and that is possible only if I concentrate full time on it. This is the reason I teach and give lectures only on software testing. My regular audience includes professionals from the IT industry, students and professors. Last year, I visited 40 colleges, 5-6 IT companies and 5-6 conferences which give an idea on the number of invites I get to speak. Of course, I could always accommodate more if I have more time at hand. Wherever I go, I get more than 100% support from all quarters. This just proves that being passionate is contagious.

**WIT** -- What is your advice to people who want to learn testing on their own? How can they do it without spending a lot of money?

**SD** -- There are many books on software testing that can give you the desired start. If you are a student, pick up a testing project and use the lab resources to learn the topic in a practical way. If you are a professional, then make

use of the people around you to enquire about testing related resources that are available in your organization. Focus on practical testing aspects that can be implemented in a small way in your college or organization. Please make use of Web resources, however with a caution. Online information could have been written by people who are less experienced in this field.

Certification is good to test your testing knowledge but always enroll with authentic certification authorities. There are many certification boards which offer affordable certifications by separating the teaching and exam sessions. If you are confident that you can study on your own, and only appear for an exam, then that would make the certification more affordable.

**WIT** -- The co-author of your book is Gopalaswamy Ramesh. Could you tell us something about him?

**SD** -- Both Ramesh and myself come from product organizations and are practitioners of testing. I met him in one of the conference and amazed to find converging goals in us, but with exploding ideas. A combined experience of 45+ years is what we tried to put in our book. Both of us have the passion to teach. He has gone one step ahead, and resigned from his position of Senior Director at Oracle to spend more time in teaching and consulting. Now he teaches project management, software maintenance and software testing at colleges.

**WIT** -- Do you blog?

**SD** -- I don't write blog but I do read blogs that are dedicated to testing. What is depressing is that many of the blogs project individuals, their strong views or business needs. Testing is a subject that needs a different perspective or treatment. I would like to see neutral blogs in the future that are more genuine in exploring the topic of Software Testing.

**WIT** -- is your opinion on the state of software testing today and where do you see it going?

**SD** -- Today, software testing is used to judge the quality of a product and is a penultimate phase in product release. Growing popularity of Agile methodologies and techniques such as test driven development, may take testing to be the frontrunner for defining quality requirements of product in future. Changes happen not only due to technology advancements. The business environment and an unpredictable economy also bring it's own set of changes. Economy of testing, low-cost testing, defect prevention will play more significant roles in the future. However, the biggest challenge in these changes revolves around people. In the past,

knowing just the concepts of testing was good enough to be a test engineer. Today, test engineers are expected to be good in both testing and development as automation and understanding of products, require both skills. With all these changes in technology and the economy, test engineers will be required to understand not only roles & skills such as Development & testing, but also business environment and what constitute product quality for customers.

**WIT** -- What are the things a person who wants to choose testing as a career should do to prepare himself for it?

**SD** - The concepts, principles, and the purpose of testing should be understood first. Testing doesn't exist alone and its main purpose is to ensure quality of product and profitability of the organization. As I said earlier, change is constant and someone should adapt himself/herself to those changes and keep learning. A career in testing is possible only if test engineers protect their jobs during tough times like the current situation. There is adequate growth in competency areas in testing around the world and the jobs are going to be only for the deserving best. This requires constant learning and foreseeing the changes in the world of software testing. Testing is a career which should be chosen by choice and not by chance.

**WIT** - What things should a novice tester do to enhance his/her testing skills for a better growth in the career?

**SD** - There is no place for novices in testing and there is no one called novice tester unless we have novice developers and novice managers. By nature, all people are good testers by their own analytical skills and using those inherent skills in unearthing defects takes them to the first level. Learning concepts from software testing books, through certifications and understanding of how testing is done in other matured industries (such as automobile industry) will create a better understanding on the purpose of testing and can take the testers to next level.

**WIT** -- What are top 5 testing books that you would recommend to people?

**SD** - Being an author of a best selling book it is unfair to ask me to recommend top 5 testing books. I would leave this question to the readers of my book to judge. But personally, I would recommend our book "Software testing – principles and practices" for readers because of our involvement with both academia and industry and whatever we have written is our practical experience on testing and the intent of the book is to share that experience both good and bad.

**WIT** -- If you were asked to name four persons who have made maximum contribution to testing in India, what names would you suggest?

**SD** -- Prakash Mutalik : For founding one of the very first companies dedicated to software testing in India.

Vikram Shah: For showing what management support can do to testing and for bringing one of the very first system test labs in Bangalore for product testing.

Ramesh Gopalaswamy (my co-author): For promoting software testing in academia across India.

Vipul Kochar: For founding the Indian Testing Board and for making software testing certifications familiar across India

**WIT** -- What are the most important lessons that you think every tester should learn, either from your experience or own experience?

**SD** -- Lesson 1: Learn to fail personally yet make the product successful. Organizations and individuals that depend on testing to just find and fix problems many times fail due to increased R&D cost and market window time lost. Test engineers who prevent defects proactively don't get any visibility and hence they are perceived as failed but they are my real heroes for product success.

Lesson 2: Testing on its own produces nothing and always it is part of a bigger purpose to make product successful. Making testing successful alone doesn't make a product successful.

Lesson 3: The job of testers is to ensure testing is eradicated. If any tester repeats the same activity again and again, then it is an indication of decreasing product quality and lack of growth for that individual.

*Disclaimer: All views expressed in this interview are my own and not that of my employer – Srinivasan Desikan*

We thank [WhatIsTesting.com](http://WhatIsTesting.com) for allowing us to publish this interview, which is originally available at:

[www.whatistesting.com/forum/phpBB3/viewtopic.php?f=35&t=228&start=0](http://www.whatistesting.com/forum/phpBB3/viewtopic.php?f=35&t=228&start=0)

## Contact Us

### Resources:

The Art of Software Testing (2004); 2nd Edition  
by Glenford Myers  
Revised & updated by Tom Badgett and Todd Thomas  
with Corey Sandler  
Wiley; ISBN-10: 0471469122

Black Box Testing (1995)  
by Boris Beizer  
Wiley; ISBN-10: 0471120944

Lessons Learned in Software Testing (2001); 1st Edition  
by Cem Kaner, James Bach, and Bret Pettichord  
Wiley; ISBN-10: 0471081124

How to Break Software: A Practical Guide to  
Testing (2002)  
by James Whitaker  
Addison Wesley; ISBN-10: 0201796198

Managing the Testing Process: Practical Tools and  
Techniques for Managing Hardware and Software  
Testing (2002); 2nd edition  
by Rex Black  
Wiley; ISBN-10: 0471223980

Managing Agile Projects (2005)  
by Kevin J. Aguanno  
Multi-Media Publications Inc.  
ISBN-10: 1895186110

## Tell us... Contribute

Feed us back your comments, questions, or suggestions on what you'd like to see in iTEST. Write to [feedback@indiantestingboard.com](mailto:feedback@indiantestingboard.com)

If you're inspired to write, you may even send your contributions to us. The decision to include your contributions will rest solely with the Editorial Team.

Write to us at [contribute@indiantestingboard.com](mailto:contribute@indiantestingboard.com)



### Head Office:

INDIAN TESTING BOARD  
A-24, First Floor, Sector-59  
NOIDA - 201301(U.P)

email: [info@indiantestingboard.com](mailto:info@indiantestingboard.com)

Phone: 0120-4355270

### Branch Office :

INDIAN TESTING BOARD  
1/2-1, 1st MAIN ROAD, 5th CROSS JUNCTION  
PALACE LOOP ROAD,  
VASANTHNAGAR  
BANGALORE - 560052  
Mobile : 09343494617

### Accreditation Managers :

#### Vinay Baid

Cell: +91 9310373004

e-mail: [vinay@indiantestingboard.com](mailto:vinay@indiantestingboard.com)

#### Pravesh Pugalia

Cell: +91 9343494617

e-mail: [pravesh@indiantestingboard.com](mailto:pravesh@indiantestingboard.com)