# From Quality by Chance, To Quality by Design

## 1. What is testing?

- ✓ An integral part of engineering
- ✓ Set of activities performed to verify that desired level of quality exists in the product
- ✓ Remember Murphy?

> *If anything can go wrong, it will…If there is a possibility of several things going wrong, the one that will cause the most damage will be the one to go wrong*
>
> *Courtsey:http://www.edwards.af.mil/history/docs_html/tidbits/murphy's_law.html*

## 2. What is software testing?

- ✓ Integral part of software engineering
- ✓ Set of activities performed to verify that desired level of quality exists in the software
- ✓ An afterthought till recently
- ✓ All quality movements (Unified process, agile methodologies, CMM, ISO etc.) advocate serious software testing efforts
- ✓ In practice, often done as a ritual, and often done, in isolation, without consideration of overall software development effort

> *"Whatever you do, you are not done, until you have verified that you did what you wanted to do"*
>
> *Courtsey: Ivar Jacobson*

- ✓ Testing is not only an effort to verify that the software behaves as defined but also to verify that it does not behave in an undesirable way even in the worst of circumstances
- ✓ Therefore, unlike other activities of software development, this requires traces of pessimism and destructive approach

## 3. What is quality?

- ✓ "In god, we trust": a good quality policy ? No
- ✓ "We do all right things, so the product should also be right": : a good quality policy ? No

> *Odds against error-free performance seem overwhelmingly high; there is only one way of performing a task correctly or at best, very few; opportunity to stray along a multitude of unintended or inappropriate pathways…. many ways to bungle a simple operation*
>
> *Courtsey: James Reason*

- ✓ There are as many definitions of quality as there are quality Gurus
- ✓ From the practical business perspective, what satisfies, or better still delights, the customer can be considered as a working definition

  *"Our job is to give the client not what he wants But what he never even dreamt he wanted"*

  *Courtsey: Sir Denys Lasdun*

- ✓ It is not an abstract notion but rather very specific & concrete, in the context of a specific project, at the specific point in time

  *"Quality is not an act; it is a habit"*

  *Courtsey: Aristotle*

## 4. What is special about software?

- ✓ Software development is inherently complex and creative activity, and involves teamwork

  *"As such, I work from the premise that software development has been, is, and will remain a fundamentally hard profession and no one thing will make a state change in how we develop software"*

  *Courtsey: Grady Booch*

  *"Einstein argued that there must be simplified explanations of nature, because God is not capricious or arbitrary. No such faith comforts the software engineer. Much of the complexity that he must master is arbitrary complexity"*

  *Courtsey: Fredreck Brooks*

  *"Software is not limited by physics, like buildings are. It is limited by imagination, by design, by organization. In short, it is limited by properties of people, not by properties of the world….. We have not met the enemy, and he is us"*

  *Courtsey: Martin Fowler*

## 5. What goes wrong?

- ✓ Software industry and, consequently software engineering, has been, and continues to be, evolving at a drastic pace.
- ✓ An average developer has an arduous task of building expertise on, and delivering solutions using, brand new technologies that he has to work with.
- ✓ This, combined with the pressure to churn out as much code as possible, leaves very little room for adoption of the available process standards, methodologies and tools, which in themselves are evolving.

- ✓ Developer is tasked with responsibility of construction, often measured by lines of code churned out, while tester is tasked with destruction, measured by the number of defects found
- ✓ Consequently, both developers & testers end up at loggerheads and perceive testing as a fault finding activity rather than as an opportunity to ensure quality
- ✓ In such situation, the first casualty is often the formalism and software testing, as an avoidable overhead

*The best tester isn't the one who finds the most bugs or who embarasses the most programmers.The best tester is the one who gets the most bugs fixed*

*Courtsey: Cem Kaner*

## 6. Why bother?

- ✓ Market is getting increasingly competitive
- ✓ Failure is not an option in a competitive world
- ✓ Demand/supply ratio favors the customer
- ✓ Software is no longer a hype but rather a commodity like any other
- ✓ Software business is expected to be like any other business
- ✓ It pays to satisfy (or, better still, delight) the customer
- ✓ Much easier to get repeat work from a satisfied customer than hunting in the open market

*"Succeeding in a changing world is beyond just surviving "*

*Courtsey: Azim Premji*

## 7. What stops, if it is so important?

- ✓ Customer's perspective is often captured in requirement specification which is often limited to the expected functionality only, leaving many things unsaid.
- ✓ Worse still, more often than not, even such documents go outdated as the project pressure mounts.
- ✓ To be effective
  a. Testing must
     i. be performed and managed properly
     ii. include a constructive destruction
     iii. be done with the customer in mind and synchronize well with the rest of software development work.
  b. Judicious mix of human factor, to leverage to make software testing creative, and automation, for consistency and repeatability, is required.
  c. Testers must
     i. understand the software development life cycle
     ii. be trained on the methodology, technology and tools
     iii. have good analytical and communication skills
     iv. have problem solving attitude, rather than problem finding
     v. know basics of automation
- ✓ Impedance to adoption

a. (Lack of) Awareness of choices
b. Fear of unknown
c. (Lack of) Skilled manpower
d. (Lack of) Opportunity for learning
e. (Lack of) Time to experiment amidst daily business pressures
f. (Lack of) Expert guidance, to fall back in case of necessity

## 8. Who we are?

- ✓ Small Company with
  a. Clear focus & vision
  b. expertise in
     i. Software engineering process (CMMI, RUP, Agile, XP)
     ii. Wide range of technologies (J2EE, .Net etc)
     iii. Software testing methodologies, practices & tools (Rational Robot, Test Manager, Functional Tester, Test Real Time, Purify, Quantify, PureCoverage, Rational Quality Architect, WinRunner, JUnit etc)
- ✓ Partnership with IBM Rational, HP Mercury, and Compuware

## 9. What services we offer?

- ✓ Software engineering consulting
- ✓ Business process modeling
- ✓ Requirement management
- ✓ Architecting & design
- ✓ Outsourced software testing

## 10. What is a competency centre?

- ✓ One stop solution for software engineering related solutions for the entire software development life cycle
- ✓ An opportunity to Horne individual skills of employees of organisations and to make oneself be at par with the industry changes.
- ✓ Mitigates the risks and unknowns providing knowledge base, tools, & experts in your friendly neighborhood
- ✓ Proof of Concept facility
  a. See and assess the success, even before you invest your time & money
  b. Opportunity to build skills in the high end of value chain
  c. Bridges gap between theoretical learning and practical implementation
- ✓ Last but not least an Affordable Solution.

*"Experience is the best of school masters! Only the school fee is heavy !! "*

## 11. What makes us different?

- ✓ Availability in your friendly neighborhood
- ✓ Working closely with you, than in isolation

- ✓ Adoption of engineering practices and standards
- ✓ Proven track record
- ✓ Certification
- ✓ Partnership with the best in world

- ✓ Judicious mix of human and automation efforts

## 12. Similar initiatives

- ✓ Technopark Software Engineering Competency Centre at Technopark, Trivandrum (India)

*If one desires a change, one must be that change before that change can take place*"

*Courtsey: Azim Premji*

| Sl. No. | What we do | What we deliver | The benefit |
|---|---|---|---|
| 1 | Requirements study<br>a) Reading through the requirement documents<br>b) Discussion with stakeholders<br>c) "Hands on" on the software | a) Glossary, with explanation<br>b) Use case diagram<br>c) Use case specification, with key scenarios<br>d) Supplementary specification | a) Essential to transparent operation & effective solution<br>b) You get to know and fine tune our understanding of the software<br>c) Serves as a smoke test of the software<br>d) Exposes the showstoppers, from an independent standpoint |
| 2 | Defect analysis | a) Defect patterns<br>b) Fault model | a) An input to further testing efforts.<br>b) This would help to prioritize and bring special focus to certain areas |
| 3 | Test planning & designing | a) Test plan<br>b) Test cases<br>c) Test procedures | This brings transparency by explaining:<br>a) what we are doing<br>b) when<br>c) how |
| 4 | Test implementation & execution | a) Test scripts (recorded or written) for automated runs<br>b) Test logs, for both manual & automated | Reusable assets for testing |

| | | | |
|---|---|---|---|
| | | c) Defect report, stating what is the defect and how it was found/how it could be reproduced | |
| 5 | Defect analysis | a) Defects, with severity<br>b) Defect trend<br>c) Defect density | Get to know weak spots on time |

*"For the want of a nail, a shoe was lost;*
*For the want of a shoe, a horse was lost;*
*For the want of a horse, a rider was lost;*
*For the want of a rider, a battle was lost;*
*For the want of a battle, a kingdom was lost, and*
*All for the want of a horse shoe nail.".*