

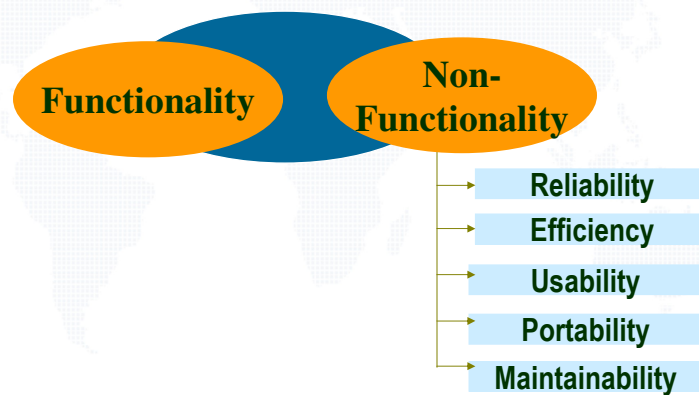
## When to Stop Testing?

**Mr. Arunkumar Khannur**  
ISQT Process & Consulting Services Pvt. Ltd.  
# 732, 1st Floor, 12th Main, 3rd Block,  
Rajajinagar, Bangalore - 560 010, INDIA  
Phone: + 91 - 80 - 23145564 / 65 /  
23142425 / 32989992  
Fax: + 91 - 80 - 23142425  
[www.isqtinternational.com](http://www.isqtinternational.com)



## When to Stop Testing?

- ◆ **Decision is becoming more and more strategic.**
- ◆ **WHY??!!**



## Challenges

- ◆ **Functional Requirements – more or less clear**
  - what the system does and why part
- ◆ **But**
  - **Performance Requirements – what are the target**
    - ◆ **Transactions/second**
    - ◆ **Download times**
  - **Scalability Requirements**
  - **Inter-operability – with which version of platforms, products, utilities and so on....**



## Challenges

- ◆ **Implicit Requirements that come up later in the cycle**
  - I assume we use JVM x.xx.xx.....
  - I assume we are using double encryption algorithms
- ◆ **There are certain requirements for which coding happens after testing!**
  - **Support Red Hat Linux x.xx.x**
  - **Support Solaris y.yy.yy**
    - ◆ **No need to code anything for these – just test, if we get any defects, we will fix**



## Challenges ...(Contd.)

### ◆ In summary

- All requirements are not clear
- For all requirements we do not have inputs
- For different requirements we should talk to different stakeholders
- For those requirements there is no coding, we have no clue what is happening till we get a defect!
- .....



## So.. Decision on When to Stop Testing is...

### ◆ Not Easy

### ◆ How can we address this?

### ◆ 3 Steps

- Strategy
- Quality Profiling
- Deriving Exit Criteria



## Strategy #1 What to Test...

### ◆ Be in the “game”

- Be stakeholder of requirements
- Do some Benchmarks and provide “data” for performance, scalability requirements
- Be prepared to do R&D



## Strategy #1 (Contd.) What to Test...

### ◆ Be “proactive” where required

- Make effective use of “scope” section in your test plan
- Communicate details of the plan & strategy in nutshells rather than full fledged document mails
- Be involved in the project all the time (if not whole of test team, may be the lead/manager)



## Strategy #2 What to Test...

- ◆ **Have suggestions and solutions apart from questions and issues – after all Testing is one of the “stakeholder”**
- ◆ **Take initiative and get things done! – rather than being in the receiving end – move out of the mode - I test whatever is “stated” – be a relay partner**



## Strategy #3 What to Test...Summary

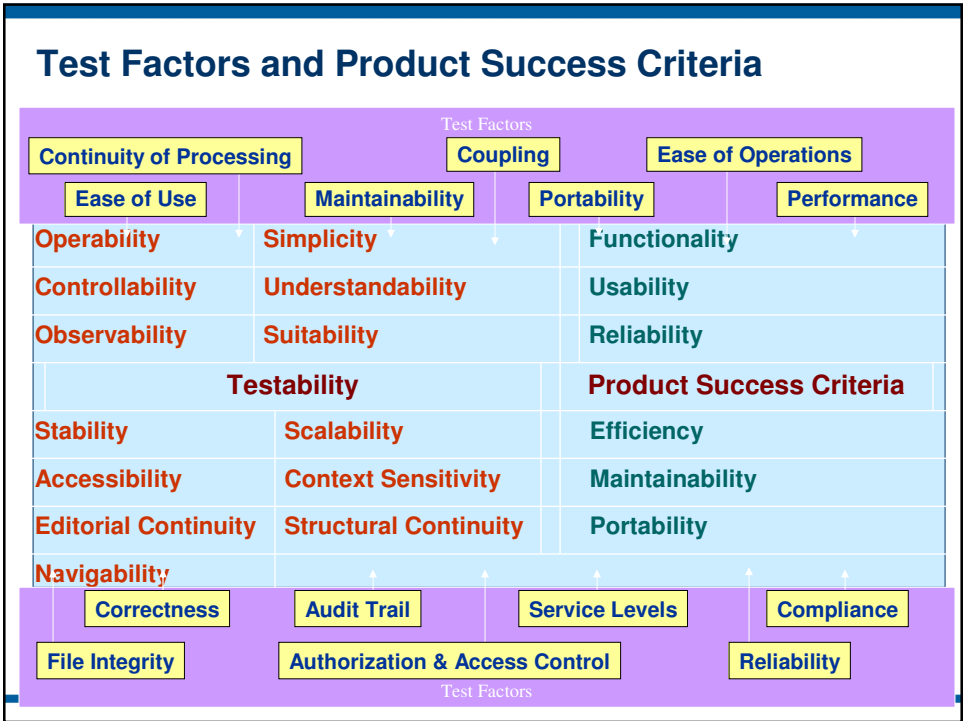
- ◆ **Visualize Requirements using models**
- ◆ **Models are to understand the context**
- ◆ **Express the “perceived” behavior**
- ◆ **Any for of the model can be used**
- ◆ **Be consistent within project**
- ◆ **Picture Vs Words**
- ◆ **Mental Models can also be a tool**
- ◆ **Key is to “gain” product knowledge**



### Strategy #4 How much to Test...

- ◆ **Analyze Specified Requirements to understand**
  - **Test Factors**
  - **Testability**
  - **Product Success Criteria**





## Quality Profiling

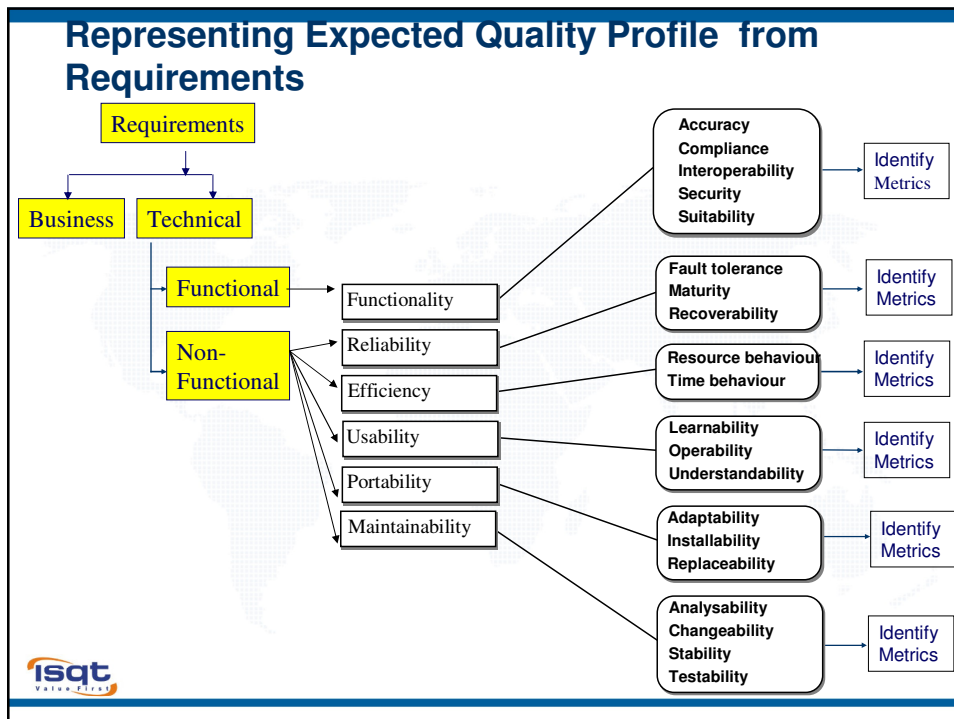
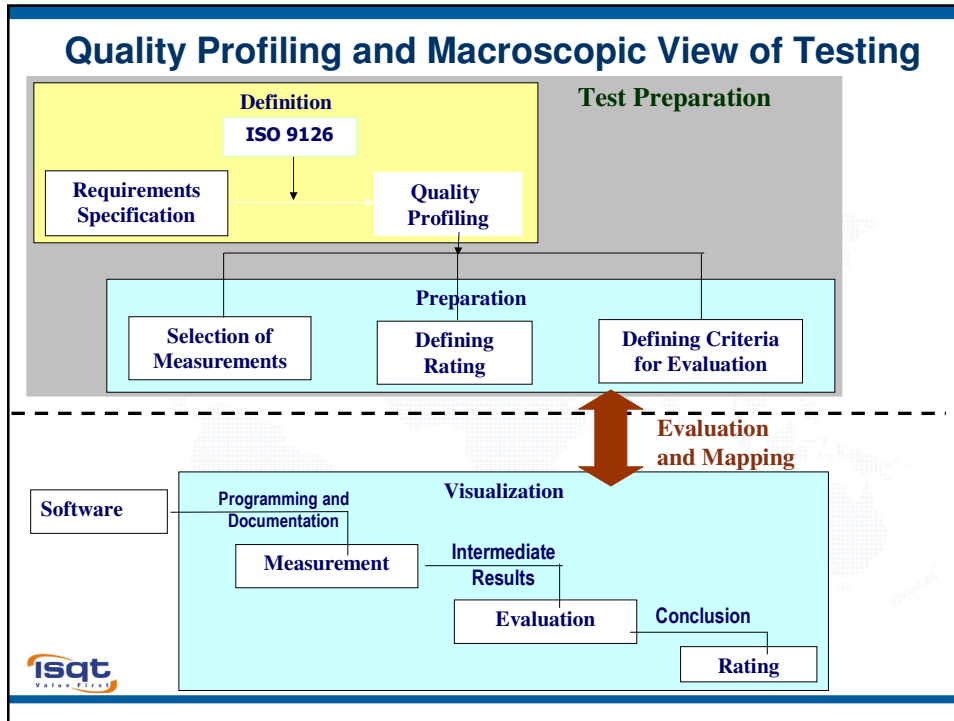
- ◆ **Quality profiles consist of**
  - **The list of specific quality characteristics considered relevant for the software product**
  - **a sort of ranking of relevance of the different quality attributes proposed by a model**



## Representing Expected Quality Profile using ISO/IEC 9126

- ◆ **Study Requirements Specification document**
- ◆ **Identify components in software to be tested**
- ◆ **Identify expected Quality Profile at**
  - **Six Characteristics level**
  - **Sub-characteristics level**
- ◆ **Identify corresponding Measurements**
- ◆ **Represent the results as a expected quality profile table**
- ◆ **Choose Evaluation Techniques**





### Quality Profile Table

Sl.No.	Requirement Specification	FUNCTIONALITY				RELIABILITY				USABILITY				EFFICIENCY		MAINTAINABILITY				PORTABILITY			
		A C C U R A C Y	C O M P L I A N C E	I N T E R O P R	S E C U R I T Y	S U I T A B L E	F A U L T T O L R	M A I N T E N A N C E	R E C O V E R B L E	L E A R N I N G	O P E R A T I O N A L	U N D E R S T A N D	R E S P O N S E	T I M E	A N A L Y S E	C H A N G E	S T A B I L I T Y	T E S T A B L E	A D A P T A B L E	I N S T A L L	C O N F I G U R E	R E P L A C E M E N T	
1	Master Setup Required for the system	H	M	NA	NA	M	NA	NA	M	NA	NA	NA	NA	NA	M	M	M	NA	NA	NA	NA	NA	
2	Transactions to be carried out(3.2)	H	H	NA	M	M	M	M	NA	L	H	L	NA	H	L	L	M	M	NA	L	NA	NA	
3	Reports to be generated from the system(3.3)	H	H	NA	H	M	NA	NA	NA	L	H	H	NA	H	NA	NA	NA	NA	NA	NA	NA	NA	
4	System Capability(3.2.1)	NA	NA	NA	NA	NA	H	H	H	NA	NA	M	H	M	NA	NA	NA	NA	NA	NA	NA	NA	
5	Safety Requirements(3.7)	H	L	NA	H	NA	H	H	H	NA	NA	NA	NA	NA	H	M	M	NA	NA	NA	NA	NA	
6	System Environment Requirements(3.9)	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	H	H	L	L	
7	Computer Resource Requirements(3.10)	NA	H	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	M	NA	NA	H	H	L	
8	Software Quality Factors(3.11)	NA	H	NA	NA	NA	NA	NA	NA	NA	NA	NA	M	H	NA	NA	NA	NA	NA	NA	NA	NA	
9	Training Related Requirements(3.14)	NA	NA	NA	NA	NA	NA	NA	NA	H	M	H	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
10	Packaging Requirements(3.17)	NA	NA	NA	NA	NA	NA	NA	NA	M	H	H	NA	NA	NA	NA	M	NA	NA	H	NA	M	

### Identify corresponding Measurements: Example

- ◆ **The metrics used to evaluate the maintainability could be the following:**
- ◆ **Analyzability:**
  - cyclomatic number
  - number of statements
  - comments rate
  - calling proof
- ◆ **Changeability:**
  - number of jump
  - number of nested levels
  - average size of statement
  - number of variables

- ◆ **Stability:**
  - number of parameters referenced
  - number of global variables
  - number of parameters changed
  - number of called relationships
- ◆ **Testability:**
  - number of non-cyclic path
  - number of nested levels
  - cyclomatic number
  - number of call-paths

**The actual implementation of each metric depends on the programming language.**

## Exit Criteria

- ◆ **Standards for work product quality, which block the promotion of incomplete or defective work products to subsequent stages of the software development process**



## Exit criteria...

- ◆ **The purpose of exit criteria is to define when to stop testing, such as at the end of a test level or when a set of tests has a specific goal.**
- ◆ **Typically exit criteria may consist of:**
  - **Thoroughness measures, such as coverage of code, functionality or risk.**
  - **Estimates of defect density or reliability measures.**
  - **Cost.**
  - **Residual risks, such as defects not fixed or lack of test coverage in certain areas.**
  - **Schedules such as those based on time to market.**

# Thank You

ISQT Process & Consulting Services Pvt. Ltd.  
# 732, 1st Floor, 12th Main, 3rd Block, Rajajinagar,  
Bangalore - 560 010, INDIA  
Phone: + 91 - 80 - 23145564 / 65 / 23142425 / 32989992  
Fax: + 91 - 80 - 23142425  
[www.isqtinternational.com](http://www.isqtinternational.com)



ISQT process & consulting services pvt. ltd.  
Value First