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

When to Stop Testing?

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

Functionality
- Reliability
- Efficiency
- Usability
- Portability
- Maintainability

Non-Functionality
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

When To Stop Testing

Arunkumar

Khannur - ISQT Process and Consulting

Services Pvt Ltd
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

### Test Factors and Product Success Criteria

<table>
<thead>
<tr>
<th>Test Factors</th>
<th>Continuity of Processing</th>
<th>Coupling</th>
<th>Ease of Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Use</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintainability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operability</td>
<td>Simplicity</td>
<td>Functionality</td>
<td></td>
</tr>
<tr>
<td>Controllability</td>
<td>Understandability</td>
<td>Usability</td>
<td></td>
</tr>
<tr>
<td>Observability</td>
<td>Suitability</td>
<td>Reliability</td>
<td></td>
</tr>
<tr>
<td>Testability</td>
<td>Product Success Criteria</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stability</td>
<td>Scalability</td>
<td>Efficiency</td>
<td></td>
</tr>
<tr>
<td>Accessibility</td>
<td>Context Sensitivity</td>
<td>Maintainability</td>
<td></td>
</tr>
<tr>
<td>Editorial Continuity</td>
<td>Structural Continuity</td>
<td>Portability</td>
<td></td>
</tr>
<tr>
<td>Navigability</td>
<td>Correctness</td>
<td>Audit Trail</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Service Levels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>File Integrity</td>
<td>Authorization &amp; Access Control</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compliance</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reliability</td>
<td></td>
</tr>
</tbody>
</table>
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 an expected quality profile table
- Choose Evaluation Techniques
Quality Profiling and Macroscopic View of Testing

**Definition**

ISO 9126

**Test Preparation**

Requirements Specification

Quality Profiling

Selection of Measurements

Defining Rating

Defining Criteria for Evaluation

Evaluation and Mapping

Software Programming and Documentation

Visualization Intermediate Results

Conclusion

Representing Expected Quality Profile from Requirements

**Requirements**

Business Technical

Functional Non-Functional

Functionality Reliability Efficiency Usability Portability Maintainability

Accuracy Compliance Interoperability Security Suitability

Fault tolerance Maturity Recoverability

Resource behaviour Time behaviour

Learnability Operability Understandability

Adaptability Installability Replaceability

Analyzability Changeability Stability Testability

Identify Metrics

When To Stop Testing Arunkumar Khannur - ISQT Process and Consulting Services Pvt Ltd
### Quality Profile Table

<table>
<thead>
<tr>
<th>Sl.No.</th>
<th>Requirement Specification</th>
<th>Functionality</th>
<th>Reliability</th>
<th>Usability</th>
<th>Efficiency</th>
<th>Maintainability</th>
<th>Portability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Master Setup Required for the system</td>
<td>F</td>
<td>C</td>
<td>U</td>
<td>L</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>2</td>
<td>Transactions to be carried</td>
<td>H</td>
<td>C</td>
<td>U</td>
<td>L</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>3</td>
<td>Reports to be generated from</td>
<td>H</td>
<td>C</td>
<td>U</td>
<td>L</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>4</td>
<td>System Capability(3.1)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5</td>
<td>System Environment(3.7)</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>6</td>
<td>Software Quality Factors(3.11)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>7</td>
<td>Training Related Requirements(3.14)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>8</td>
<td>Packaging Requirements(3.17)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

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

### 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

When To Stop Testing  
Arunkumar  
Khunnur - ISQT Process and Consulting  
Services Pvt Ltd
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.
When To Stop Testing                  Arunkumar
Khannur - ISQT Process and Consulting
Services Pvt Ltd