

SOFTWARE QUALITY ASSURANCE

John Abbott College JPC

Higher-Order Testing

**M. E. Kabay, PhD, CISSP
Director of Education, NCSA
President, JINBU Corp**

Copyright © 1997 JINBU Corp. All rights reserved

Higher-Order Testing

- Beyond Module Testing
- Integration Testing
- Function Testing
- System Testing
- Acceptance Testing
- Installation Testing
- Test Planning and Control
- Test Completion Criteria

Beyond Module Testing

- **Practical programs must represent real-world needs**
- **Programs must do what their users expect and demand**
- **SDLC: System Development Life Cycle**
 - **requirements: why program is needed**
 - **objectives: what and how well**
 - **external specifications: representation of program to its users**
 - **how program is constructed**

Beyond Module Testing

- **Software errors arise from miscommunication**
- **JAD (Joint Application Development) and RAD (Rapid Application Development)**
 - **emphasize constant correction**
 - **by constant communication with users**
- **Specific testing phases emphasize corrections to specific phases of SDLC**

Integration Testing

- **Establish that interconnections among modules function as required**
- **Implicit in module testing as previously discussed**

Function Testing

Find discrepancies between program and external specification

- **What the program does as a black box**
- **User-eye view**

System Testing

- **Definition**
- **Facility Testing**
- **Stress Testing**
- **Volume Testing**
- **Usability Testing**
- **Security Testing**
- **Performance Testing**
- **Storage Testing**
- **Configuration Testing**
- **Compatibility / Conversion Testing**
- **Installability Testing**
- **Reliability Testing**
- **Recovery Testing**
- **Serviceability Testing**
- **Documentation Testing**
- **Procedure Testing**

System Testing

Definition

- Compare program to original objectives
- Cannot base tests on external specifications
 - are attempting to verify conformity between what the ext specs *represent* and actual behaviour of program
 - therefore work with user documentation + program objectives + program
- Must have written, measurable objectives for program

System Testing

Facility Testing

- **Refers to the documented features or functions**
- **Scan objectives sentence by sentence**
- **Look for failure to comply**
- **Can usually be done without computer**

System Testing

Stress Testing

- **Show that program cannot handle sudden increase in load, demand, input**
- **Applies to programs that must have minimum throughput or response time**
- **Distinct from volume testing**
 - **volume testing looks at total continuous load to process**
 - **stress testing looks at effects of sudden imposition of load**
- **E.g., if specs stipulate ability to handle up to 200 concurrent sessions, try suddenly going from 50 users to 200**

System Testing

Volume Testing

- **Show that system cannot handle maximum required amounts of input**
- **E.g., if program must be able to handle 200 Gb files, test with 200 Gb files and more**
- **E.g., if specifications stipulate ability to process 10,000 orders in a batch, test with 10,001 orders**

System Testing

Usability Testing

- **Show that normal users will fail to accomplish their documented goals using program**
- **Clumsy design**
- **Unsuitable language**
- **Meaningless error messages**
- **Inconsistencies in functions from screen to screen**
- **Inadequate checks on input**
- **Useless options**
- **Difficult data entry**

System Testing

Security Testing

- **Try to violate rules of confidentiality, integrity and availability**
- **“Tiger Teams” specialize in attacking security**
- **Be sure to obtain authorization for such attacks!**
- **Use known attacks as described in security literature**
- **CERT, CIAC Advisories**
- **USENET postings**
- **National Computer Security Association**

System Testing

Performance Testing

- **Show that program is unable to meet specified throughput or response-time requirements**
- **Example of failure:**
 - **10 second Service Level Agreement for all transactions**
 - **43 minute response time**
 - **absence of volume testing allowed poor design to pass**
- **Often combined with volume and stress testing**

System Testing

Storage Testing

- **Inability to meet requirements for working correctly with specified storage**
 - cannot work with minimum RAM
 - minimum required disk space exceeds capacity
- **Incompatibilities with**
 - RAM management software
 - virtual memory
 - encryption software
 - compression software

System Testing

Configuration Testing

Look for inability to function with specified

- Hardware--RAM, ROM, CPU, peripherals**
- Software--operating system, TSRs, drivers**
- Program parameters--directories, max, min users / files / records**
- Network operating systems**
 - versions**
 - parameters**

System Testing

Compatibility/Conversion Testing

- **Failure of conversions from older systems**
- **Inability to use existing data files**
- **Conflicts with other legacy systems**
- **Incompatibility with older operating systems**
- **Inability to function on older hardware**
- **Conflicts with older networks**

System Testing

Installability Testing

- **Difficulties during installation of new product**
- **Integration with operating systems**
- **Special installation software**

System Testing

Reliability Testing

- **Difficult to demonstrate long MTBF**
- **Monitor rate of discovery of new errors**
- **Use mathematical models to estimate reliability**

System Testing

Recovery Testing

- **Show that system cannot recover according to specifications**
- **Try deliberate “sabotage”**
- **Measure MTTR**
- **Look for permanent data damage**

System Testing

Other types of tests

- **Serviceability Testing**
- **Documentation Testing**
- **Procedure Testing**

Acceptance Testing

- **Performed by client organization**
- **Conformity to contract**
- **Includes installation testing**
 - **methods of showing that installation failed**
 - **check files, directories, code libraries, databases**

Test Planning and Control

- **Too many organizations act as if there will be no errors found**
 - **No resources for response**
 - **Inadequate time allocated for repair**
- **Need extensive planning**
- **Regression testing**
 - **Especially important**
 - **Tests after every change or set of changes**

Test Completion Criteria

- **Bad idea**
 - Stop when time runs out
 - Stop when no more errors found
- **Better way**
 - Complete specific methodology
 - But not possible for all phases
 - Subjective
 - Focusses on test method, not goal
- **Best approach**
 - Find specific number of errors...
 - ... and try for a few more when you find those

Test Completion Criteria

How to know when enough errors found?

- **Statistical calculations**
- **Mark-recapture of known errors**
- **Parallel testing by independent teams**

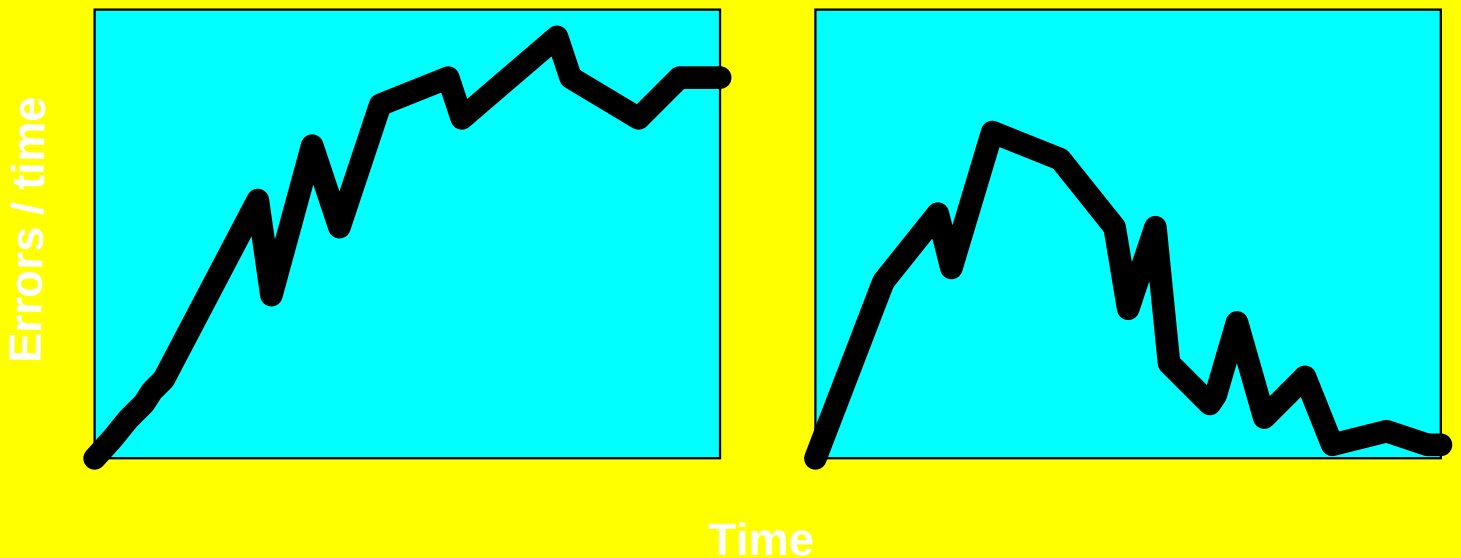
What if there are too few errors in reality?

- **Need judgement of test quality**
- **Independent evaluation**

Test Completion Criteria

Graph errors found per unit time

- Continue searching if success rate high
- Consider stopping when success rate falls



Independent Test Agency

- **Hire different organizations for development and testing**
- **Develop own separate department**
- **Has worked well in practice**
 - **high motivation in testing**
 - **healthy competition**
 - **development of specialized testing skills**

