Heuristics of Software Testability

by James Bach, Satisfice, Inc.

Controllability

The better we can control it, the more the testing can be automated and optimized.

- A scriptable interface or test harness is available.
- Software and hardware states and variables can be controlled directly by the test engineer.
- Software modules, objects, or functional layers can be tested independently.

Observability

What you see is what can be tested.

- Past system states and variables are visible or queriable (e.g., transaction logs).
- Distinct output is generated for each input.
- System states and variables are visible or queriable during execution.
- All factors affecting the output are visible.
- Incorrect output is easily identified.
- Internal errors are automatically detected and reported through self-testing mechanisms.

Availability

To test it, we have to get at it.

- The system has few bugs (bugs add analysis and reporting overhead to the test process).
- No bugs block the execution of tests.
- Product evolves in functional stages (allows simultaneous development and testing).
- Source code is accessible.

Simplicity

The simpler it is, the less there is to test.

- The design is self-consistent.
- Functional simplicity (e.g., the feature set is the minimum necessary to meet requirements)
- Structural simplicity (e.g., modules are cohesive and loosely coupled)
- Code simplicity (e.g. the code is not so convoluded that an outside inspector can't effectively review it)

Stability

The fewer the changes, the fewer the disruptions to testing.

- Changes to the software are infrequent.
- Changes to the software are controlled and communicated.
- Changes to the software do not invalidate automated tests.

Information

The more information we have, the smarter we will test.

- The design is similar to other products we already know.
- The technology on which the product is based is well understood.
- Dependencies between internal, external and shared components are well understood.
- The purpose of the software is well understood.
- The users of the software are well understood.
- The environment in which the software will be used is well understood.
- Technical documentation is accessible, accurate, well organized, specific and detailed.
- Software requirements are well understood.