Source Instrumentation

Return to Glossary

Source Instrumentation is the process by which developers and domain experts selectively instrument the Source Code under test for the purpose of writing test scenarios against the executing application. Implementing tests that leverage **Source Instrumentation** is called **Expectation Testing**. This validation technique is very useful for verifying proper code sequencing based on the software's internal design.

Unlike traditional unit testing that drives testing based on input parameters and isolating functionality, expectation testing is executed within a fully functional software build running on a real target platform. Expectation tests are not dependent on input parameters, but often leverage the same types of input/output controls used by functional and black-box testing.

Another unique feature of expectation testing is that domain expertise is not required to implement a test. Developers and domain experts use instrumentation to export design knowledge of the software to the entire team. Furthermore, there is no stubbing required, no special logic to generate input parameters, and no advanced knowledge required of how the application software is coded.

To enable effective test coverage, developers and domain experts are required to insert instrumentation at key locations to gain insight and testability. Here are some general suggestions for Source Code areas to consider instrumenting:

- critical function entry/exit points
- state transitions
- critical or interesting data transitions (using optional payload to convey data values)
- callback routines
- data persistence
- error conditions

Source: https://www.stridewiki.com/index.php?title=Source_Instrumentation_Overview

From:
https://www.omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.a_glossary:s:source_instrumentation&rev=1642588130

Last update: **2022/01/19 05:28**