Tool Qualification-Kit for Testwell CTC++ Test Coverage Analyser
Content

- Overview
- Qualification Method
- Example: First Qualification Steps
  - Installation
  - Qualification Planning
  - Text Execution & Analysis
- Availability and feedback
Testwell CTC Qualification Kit

- Conforms to ISO 26262, DO-330, DO-178C, IEC 61508, EN 50128
- Based on a qualification model for Testwell CTC++
  - Flexible, i.e. fit for many purposes
  - Process integration: considers development process
  - Minimal effort: integrate existing verification activities
  - Extensible: User can add tests and mitigations
- Covers:
  - Language C,
  - Coverage: Statement, conditions, MC/DC,
  - report generation: text, xml, html
- Developed together with VALIDAS
- Q-Kit contains
  - User Manual
  - Qualification Support Tool
  - Test cases for Testwell CTC++
  - Test Automation Unit to run tests
  - V&V Report
  - 8 hours of qualification support from Validas AG
- Price: please contact us
Content

- Overview
- Qualification Method
- Example: First Qualification Steps
  - Installation
  - Qualification Planning
  - Text Execution & Analysis
- Availability and feedback
Standards (ISO 26262, IEC 61508, DO 178, EN 50128) require a safe usage of all SW development tools
- Tool classification: Determine impact & qualification needs
- Critical tools: Qualification
- Documentation & guidelines

We consider the following processes
- Tool Chain Analysis: Classification of all tools (TCA)
- Tool Qualification: Qualification of one tool, e.g. by application of a qualification kit with a qualification support tool (QST)
- Qualification Kit Creation (TCA)
  - Create support for classification
  - Create support for qualification
The verification objectives of the tool operational verification and validation process consist of the detection and reporting of errors that may have been introduced during the tool development processes.
Qualification by Validation

Qualification Kit User Manual

1. Test Suite Spec
   3. V&V Plan
      4. V&V Report
      6. Test Plan
      7. Test Report
      8. Tool Qualification Report

Potential Tool Errors:
- Remaining
- Detectable, Avoidable
- Errors in unused features

Tool Classification Report

Tool Safety Manual

Error Mitigations

Test Automation Unit (TAU)

Legend:
- N: Sequence
- : Requires

Verifysoft TECHNOLOGY
1. **Structure Modeling**
   - Use-Cases,
   - Relevant tool features,
   - Artifacts

2. **Analysis Modeling**
   - Potential errors
   - Available checks & restrictions

3. **Qualification Modeling**
   - Qualification cost optimization
   - Qualification planning
   - Qualification tests
     - Identification
     - Assignment to
       - Features
       - Errors

---

**Tool Qualification Model**

**Qualification Documents**

- Tool Impact
- Tool Classification Report
- Tool Safety Manual
- Tool Qualification Plan
- Tool Qualification Report
- Test Plan
- Test Report
- V&V Report

---

1. Application of Qualification Kit
2. + 3. Creation of Qualification Kit
Content

- Overview
- Qualification Method
- Example: First Qualification Steps
  - Installation
  - Qualification Planning
  - Text Execution & Analysis
- Availability and feedback
Installation

- Install Graphviz (document generation) from http://www.graphviz.org
- Install Python 3.2 or higher (test execution) from http://www.python.org/download/releases/3.2/ and add it to your search PATH
- Install CTC
- Install Qualification Kit
  - Get QKit_Qual-win32.win32.zip
  - Unpack it, e.g. into C:\Program Files\Validas\QKit

Contains documentation in plugins\QST_ToolModel_1.0.0.201310180835\Files\QKit\Documentation

This is the Qualification Support Tool
Content

- Overview
- Qualification Method
- **Example: First Qualification Steps**
  - Installation
  - Qualification Planning
  - Text Execution & Analysis
- Availability and feedback
Start QST

- Details can be found in documentation (QKit User Manual)
- Double click on

![Validas Qualification Support Tool.exe](image)

- Click on Start Qualification

![Qualification Support Tool](image)

- Select an (empty/non-existent) Directory for Qualification, e.g. C:\Projects\Qualification\ExampleQualification
Select Qualification Standard

Select Standard for Testwell CTC++

There are more safety standards supported for qualification. Please select one

Standard for Qualification

- ISO26262
- IEC61508
- EN50128
- DO178C
- DO330

Save  < Back  Next >  Finish  Cancel
Select Variant

Select Variant for Testwell CTC++

There are more variants ready for qualification. Please select one

Variant for Qualification

- [ ] HostTest
- [x] TargetTest Bitcov
- [ ] TargetTest Hota

Save  < Back  Next >  Finish  Cancel
Select Tool

Select Tool for Qualification from Testwell CTC++

There are more tools ready for qualification. Please select at least one from Testwell CTC++

Tool for Qualification

- CTC (Criteria3)

Use Cases of CTC (Criteria3):

- Target with Bitcov

Add Use Case  Remove Use Case

Save  < Back  Next >  Finish  Cancel
Configure Use Case (Default)
Configure Use Case

- Use Case has Features
- Features have potential Errors and known bugs
- Errors have Mitigations
  - Checks
  - Restrictions
- Status: Next disabled, i.e. not enough mitigations selected
Configure Use Case: Error Mitigation

- **Select Error:** available mitigations get highlighted

- **Choose mitigation:** mitigated errors get marked as “(mitigated)"

Green Errors are testable and do not need mitigations
Currently no tests available
Use Case: Known Bug Mitigation

- Mitigate the Known Bugs (like potential errors)
- When all untestable potential errors are mitigated, color changes to **yellow** and “Next” Button is enabled
Optional: Planning of Roles

- Select roles in the Qualification Project (left side)
- Assign persons names (right sides) by entering their names
Optional: Planning of Steps

- Select step in the Qualification Project (left side)
- Assign dates (right sides) by entering their values and roles

Steps are hierarchic and can be planned with inheritance, i.e. planning super steps instead of detailed steps.
Optional: Planning Artifacts

- Select Artifacts in the Qualification Project (left side)
- Assign files (right sides) by entering their paths

- Note: the planning can also be done as documentation of the project before extending the Tool Qualification Plan to the Tool Qualification Report
Optional: QST Summary

- Shows the status and the generated documents

**Qualification Summary for Qualification of CTC**

Summarizes the qualification that can be started now by creating the qualification documents of CTC in Target with Bitcov

- **Target Directory**: C:\Programme\Qualification\CTCEexample
- **Names of Use Cases**: Target with Bitcov
- **Number of Features**: 18
- **Number of selected Checks**: 6
- **Number of selected Restrictions**: 2
- **Number of Tests**: 0

<table>
<thead>
<tr>
<th>Name</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool Classification Report</td>
<td>C:\Programme\Qualification\CTCEexample\Validation\Documentation\TCR.docx</td>
</tr>
<tr>
<td>Tool Qualification Plan</td>
<td>C:\Programme\Qualification\CTCEexample\Validation\Documentation\TQP.docx</td>
</tr>
<tr>
<td>Tool Safety Manual</td>
<td>C:\Programme\Qualification\CTCEexample\Validation\Documentation\TSM.docx</td>
</tr>
<tr>
<td>Tool Test Plan</td>
<td>C:\Programme\Qualification\CTCEexample\Validation\TestExecution.txt</td>
</tr>
<tr>
<td>Toolchain Model</td>
<td>C:\Programme\Qualification\CTCEexample\QKit\Model\Model.tca</td>
</tr>
</tbody>
</table>

Pressing Finish starts generation of documents
Generated Documents

- Qualification Documents

- Set the Word variable Client with your company/project name

- Update the document (Strg+a, F9)

Test plan in ExampleQualification/Validation/TestExecution.txt
Content

- Overview
- Qualification Method
- Example: First Qualification Steps
  - Installation
  - Qualification Planning
  - Text Execution & Analysis
- Availability and feedback
Test Selection

- Some features are testable (green)
Test Execution: Configuration

- Test Execution Directory: ExampleQualification/Validation/TestRun contains the test script runTest.bat

- Edit the **java path** inside runTest.bat to a java runtime environment (jre), e.g. the one delivered with the QST tool

This is the command to start the test execution:
Test Execution: Run Tests

- Open a command shell and change into the TestRun directory by
  cd %QST_QUALIFICATION_TARGET%\Validation\TestRun

Start the tests by
  .\runTests.bat ..\TestExecution.txt ..\ToolConfig\tool_config.py

Please note that the setting of the QST_QUALIFICATION_TARGET directory does not impact the Windows explorer settings, hence restart it or start the command scripts (e.g. for coverage measurement) from the console.
Test Execution: Progress

- Console shows progress of test execution:
  - Date, times, TAU version, results,...

```
c:\\Programme\Qualification\ExampleQualification\Validation\TestRun\.\runTests.bat ..\TestExecution.txt ..\ToolConfig\tool_config.py
[INTERTAU_VERSION]=[Version 1.13, $Rev: 3430 $]
[INTRATAU_VERSION]=[Version 0.1, $Rev: 554 $]
[TAU_VERSION]=[0.1]
starting tests
09.02.2014
19:45
configuration

[QUALIFICATION_DIR]=c:\Programme\Qualification\ExampleQualification\Validation\TestRun\.\.
[JAVA_HOME]=c:\Program Files\Java\jre7]
[PYTHONPATH]=c:\Programme\Qualification\ExampleQualification\Validation\TestRun\.\.

[TEST_PLAN]=c:\Programme\Qualification\ExampleQualification\Validation\TestExecution.txt
[CONFIG_FILE]=c:\Programme\Qualification\ExampleQualification\Validation\ToolConfig\tool_config.py

Executing job command: "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\.\.
ctc -Im gcc test.c -I "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\.\.
cmpc -o test.exe

Executing job command: test.exe > test.out

Executing job command: ctcpost MON.sym MON.dat -p profile.txt

[SUCCESS]: testDiffOutFiles
[FAILURE]: testDiffReportFiles: False is not true : Generated and reference coverage files differ.

FINISHED TEST RUN
Stop Time = [2014-02-09 19:45:18.741000]
Duration = [0:00:02.335000]```
Test Output: Logfile and Report

- Created in TestRun directory:
  - runTests.log: Logfile
  - TestReport/junit-noframes.html

- Build-Folder (after test execution):
  - runTst.bat: Script for re-execution of this single test
## Test Report: Overview

### Unit Test Results

- **Tool:** CTC
- **TAU Version:** 0.1
- **Qualification directory:** c:\Program Files\Qualification\ExampleQualification\Validation\TestRun\..
- **Java home:** C:\Program Files\Java\jdk
- **Pythonpath:** c:\Program Files\Qualification\ExampleQualification\Validation\TestRun\..
- **Test plan:** c:\Program Files\Qualification\ExampleQualification\Validation\TestExecution.txt
- **Config File:** c:\Program Files\Qualification\ExampleQualification\Validation\ToolConfig\tool_conf.py
- **User:** slotosch
- **Date:** Sun Feb 09 19:45:20 CET 2014

### Summary

<table>
<thead>
<tr>
<th>Tests</th>
<th>Failures</th>
<th>Errors</th>
<th>Success rate</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>50.00%</td>
<td>2.274</td>
</tr>
</tbody>
</table>

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

### Package

<table>
<thead>
<tr>
<th>Name</th>
<th>Tests</th>
<th>Errors</th>
<th>Failures</th>
<th>Skipped[0/1]</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.274</td>
</tr>
</tbody>
</table>

### Test Case General

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Type</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>testDiffOutFiles</td>
<td>Success</td>
<td></td>
<td>0.001</td>
</tr>
<tr>
<td>testDiffReportFiles</td>
<td>Failure</td>
<td>N/A</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Errors:
- File 'C:\Python31\Lib\unittest\case.py', line 570, in _executeTestPart
  function()
- File 'C:\Program Files\Qualification\ExampleQualification\Validation\TestRun\General\Test.py', line 60, in testDiffReportFiles
  self.assertIsTrue(true, 'Generated and reference coverage files differ.
File: 'C:\Python31\Lib\unittest\case.py', line 506, in assert True
  raise self.failureException(msg)
Test Logfile: TestRun/runTests.log

Contains test results and commands to reproduce test

STARTED TEST RUN

Configuration:
TAU Version: Version 1.13, $Rev: 3430$
QUALIFICATION_DIR: c:\Programme\Qualification\ExampleQualification\Validation\TestRun\...
JAVA_HOME: C:\Program files\Java\jre7
PYTHONPATH: c:\Programme\Qualification\ExampleQualification\Validation\TestRun\...
\Kit\TAU\InterTAU; C:\Programme\Qualification\ExampleQualification\Validation\TestRun\..
TEST_PLAN: c:\Programme\Qualification\ExampleQualification\Validation\TestExecution.txt
CONFIG_FILE: c:\Programme\Qualification\ExampleQualification\Validation\ToolConfig\tool_config.py

Parsing the test plan: c:\Programme\Qualification\ExampleQualification\Validation\TestExecution.txt...
Finished parsing the test plan successfully.
Removing .xml files from TestData...

Collecting all test files from C:\Programme\Qualification\ExampleQualification\\Kit\Testsuite\General
Adding test file C:\Programme\Qualification\ExampleQualification\\Kit\Testsuite\General\Tst.py

TESTSUITE_DIR: C:\Programme\Qualification\ExampleQualification\\Kit\Testsuite
TESTRUN_DIR: C:\Programme\Qualification\ExampleQualification\Validation\TestRun

Deleting preexisting test run folder C:\Programme\Qualification\ExampleQualification\Validation\TestRun\General
Copying test files from : C:\Programme\Qualification\ExampleQualification\\Kit\Testsuite\General to test folder
Creating file C:\Programme\Qualification\ExampleQualification\Validation\TestRun\General\runTst.bat

Loaded 2 tests from C:\Programme\Qualification\ExampleQualification\Validation\TestRun\General\Tst.py
Running test package: General.Tst

Executing job command: "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\...
\Kit\TAU\InterTAU\bin\tau-run --log-level=## --log-file=STDOUT:STDOUT: "b" --log-level=## --log-file=STDERR:STDOUT: None"
Executing job command: ctc -i m gcc test.c -I "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\.
STDOUT: STDOUT: b"
Test Analysis: One Test Failed

Logfile message explains

Executing job command: "c:\Programme\Qualification\ExampleQualification\Validation\TestRun..\..\QK"
STDOUT: STDOUT: b''
STDERR: STDERR: None
Executing job command: ctc -i m gcc test.c -I "c:\Programme\Qualification\ExampleQualification\Valic
STDOUT: STDOUT: b''
STDERR: STDERR: None
Executing job command: test.exe > test.out
STDOUT: STDOUT: b''
STDERR: STDERR: None
Executing job command: ctcpo MON.sym MON.dat -p profile.txt
STDOUT: STDOUT: b''
STDERR: STDERR: None

SUCCESS]: testDiffOutFiles

[FAILURE]: testDiffReportFiles: False is not true : Generated and reference coverage files differ.

From Tst.py:

def testDiffReportFiles(self):
    boolOk=checkutils.diff(strFile1=T_Test.here+r'\profile.txt',
                          strFile2=T_Test.here+r'\expected_profile-im.txt')
    self.assertTrue(boolOk, 'Generated and reference coverage files differ.')
Reports are Different

Different dates

Different number of executions

More Information (different report options?)
Content

- Overview
- Qualification Method
- Example: First Qualification Steps
  - Installation
  - Qualification Planning
  - Text Execution & Analysis
- Availability and feedback
Trac & Wiki System

- [https://opentrac.teststatt.de/tracs/verifysoft/wiki](https://opentrac.teststatt.de/tracs/verifysoft/wiki)
- Documentation
- Download of Releases & Nightly Builds
- Issue Reporting & Tracking (of the Q-Kit, not CTC)

Available Reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>Active Tickets</td>
</tr>
<tr>
<td>{2}</td>
<td>Active Tickets by Version</td>
</tr>
<tr>
<td>{3}</td>
<td>Active Tickets by Milestone</td>
</tr>
<tr>
<td>{4}</td>
<td>Accepted, Active Tickets by Owner</td>
</tr>
<tr>
<td>{5}</td>
<td>Accepted, Active Tickets by Owner (Full Description)</td>
</tr>
<tr>
<td>{6}</td>
<td>All Tickets By Milestone (Including closed)</td>
</tr>
<tr>
<td>{7}</td>
<td>My Tickets</td>
</tr>
<tr>
<td>{8}</td>
<td>Active Tickets, Mine first</td>
</tr>
</tbody>
</table>

- Q-Kit users get access to it