

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**

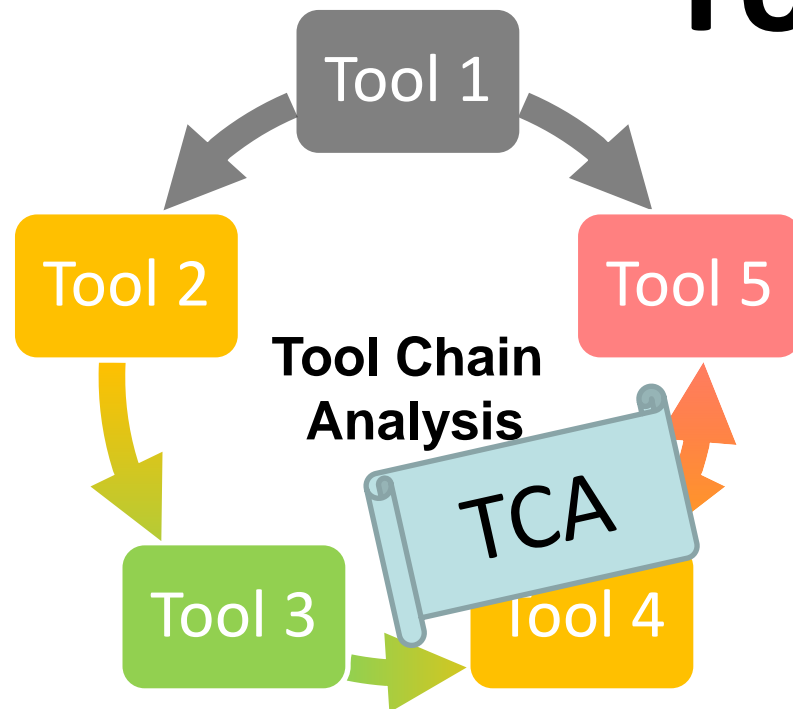


Content

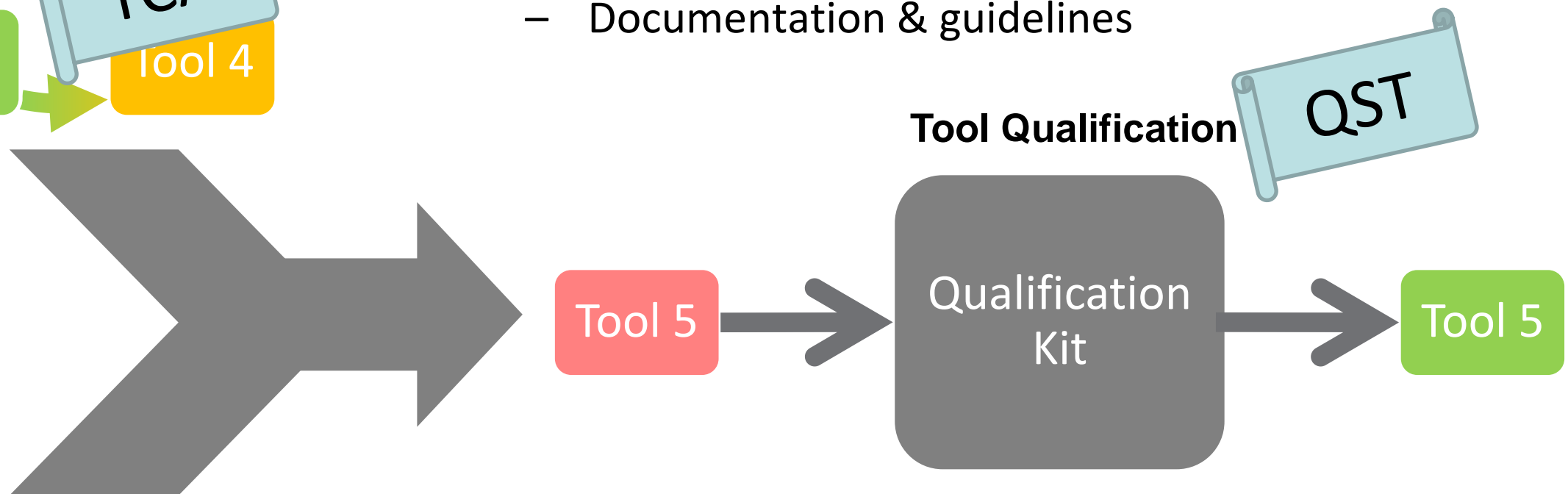
- ▶ Overview
- ▶ **Qualification Method**
- ▶ Example: First Qualification Steps
 - Installation
 - Qualification Planning
 - Text Execution & Analysis
- ▶ Availability and feedback



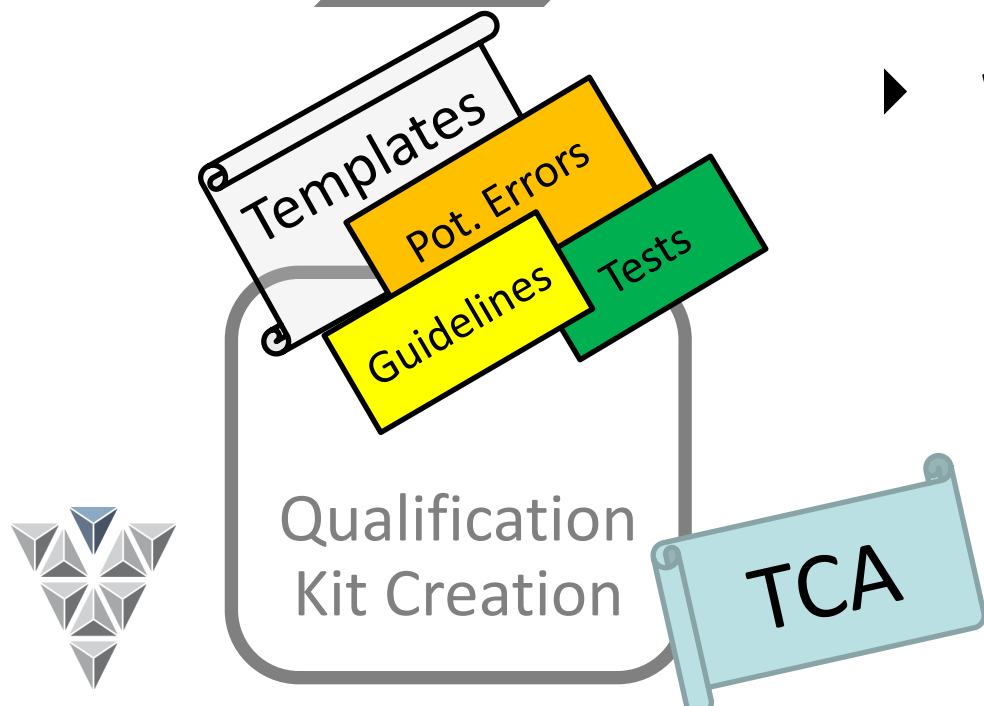
Tool Qualification



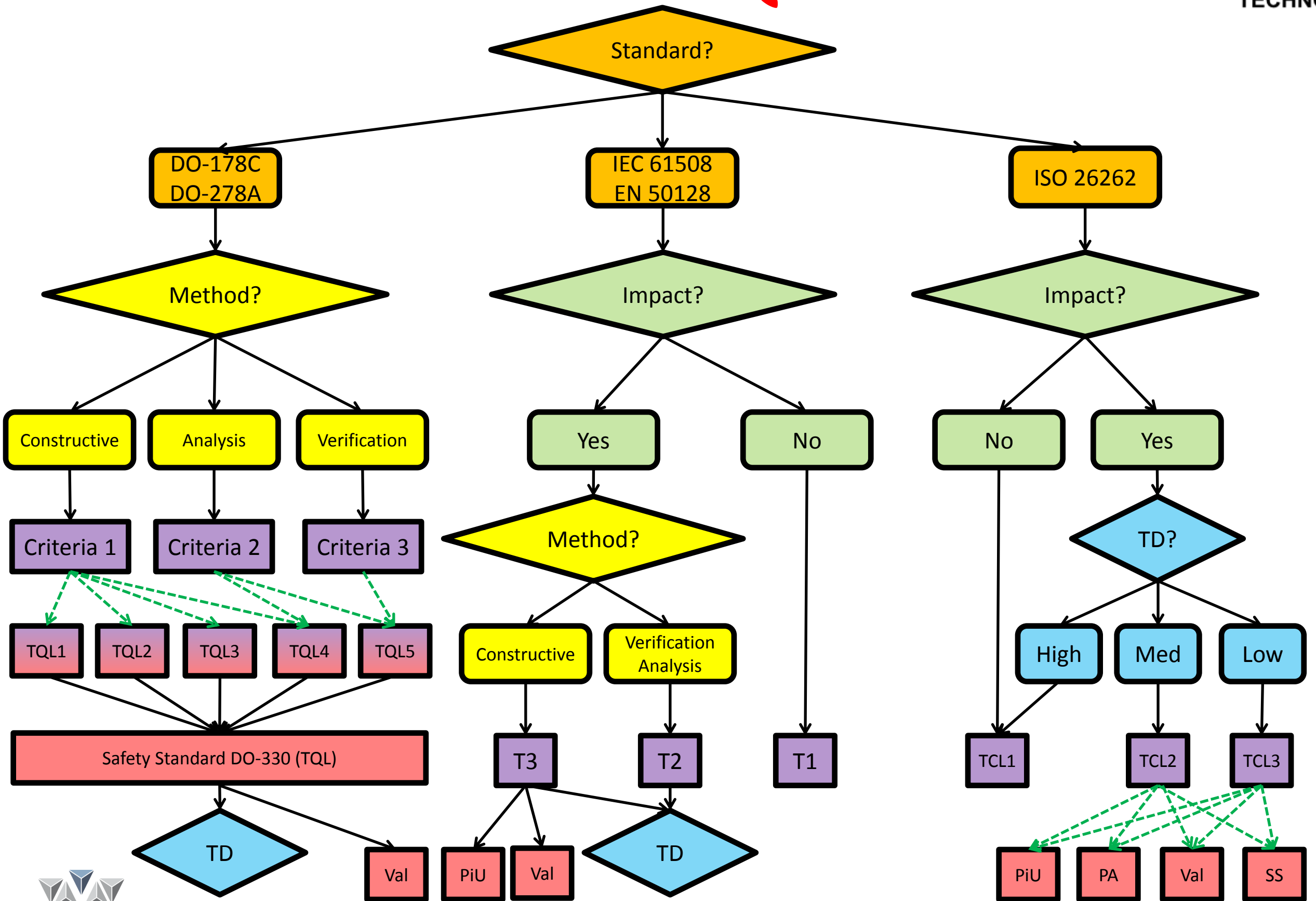
- ▶ Standards (ISO 26262, IEC 61508, DO 178, EN 50128) require a safe usage of all SW development tools
 - Tool classification: Determine impact & qualification needs
 - Critical tools: Qualification
 - Documentation & guidelines



- ▶ 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

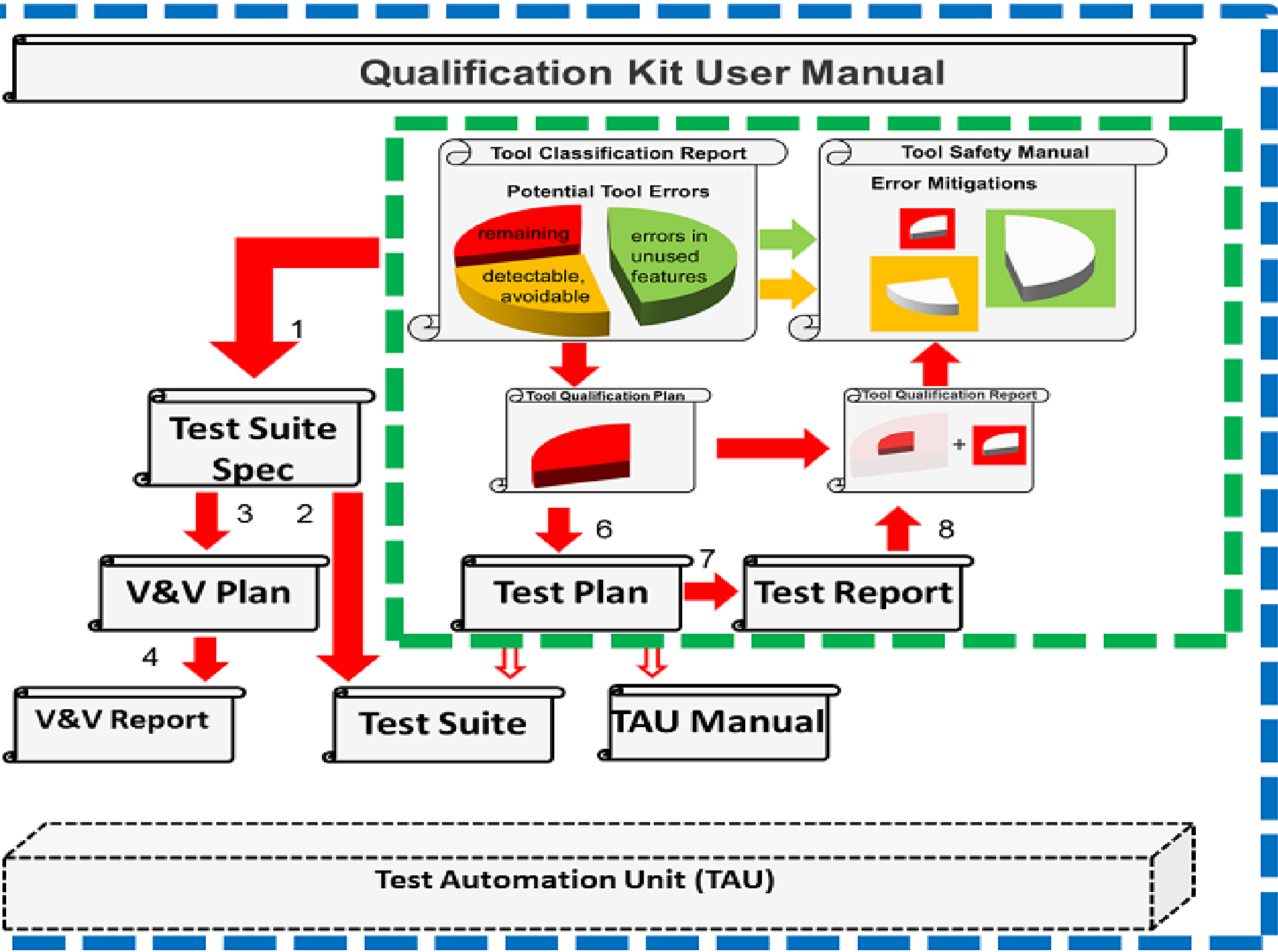


Tool Classification & Qualification Verifysoft TECHNOLOGY



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



Model-Based Tool Qualification

1. Structure Modeling

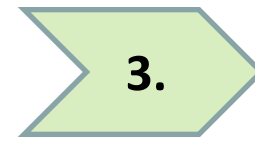
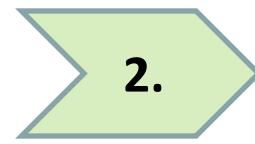
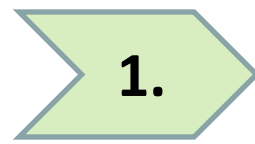
- Use-Cases,
- Relevant tool features,
- Artifacts

2. Analysis Modeling

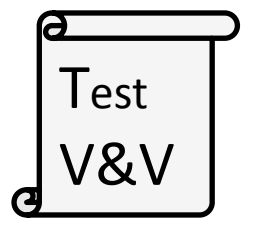
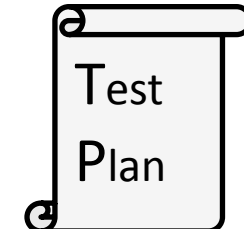
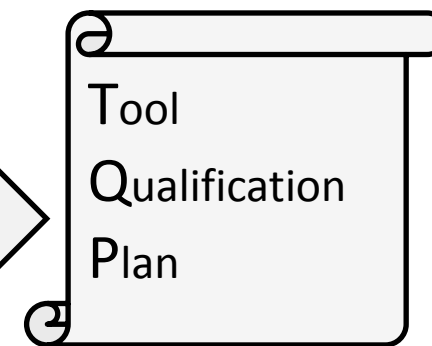
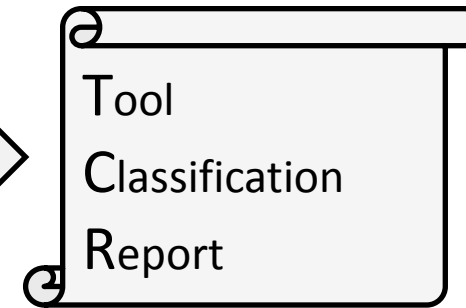
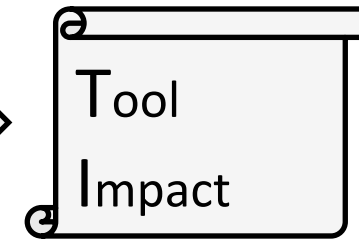
- Potential errors
- Available checks & restrictions

3. Qualification Modeling

- Qualification cost optimization
- Qualification planning
- Qualification tests
 - Identification
 - Assignment to
 - Features
 - Errors

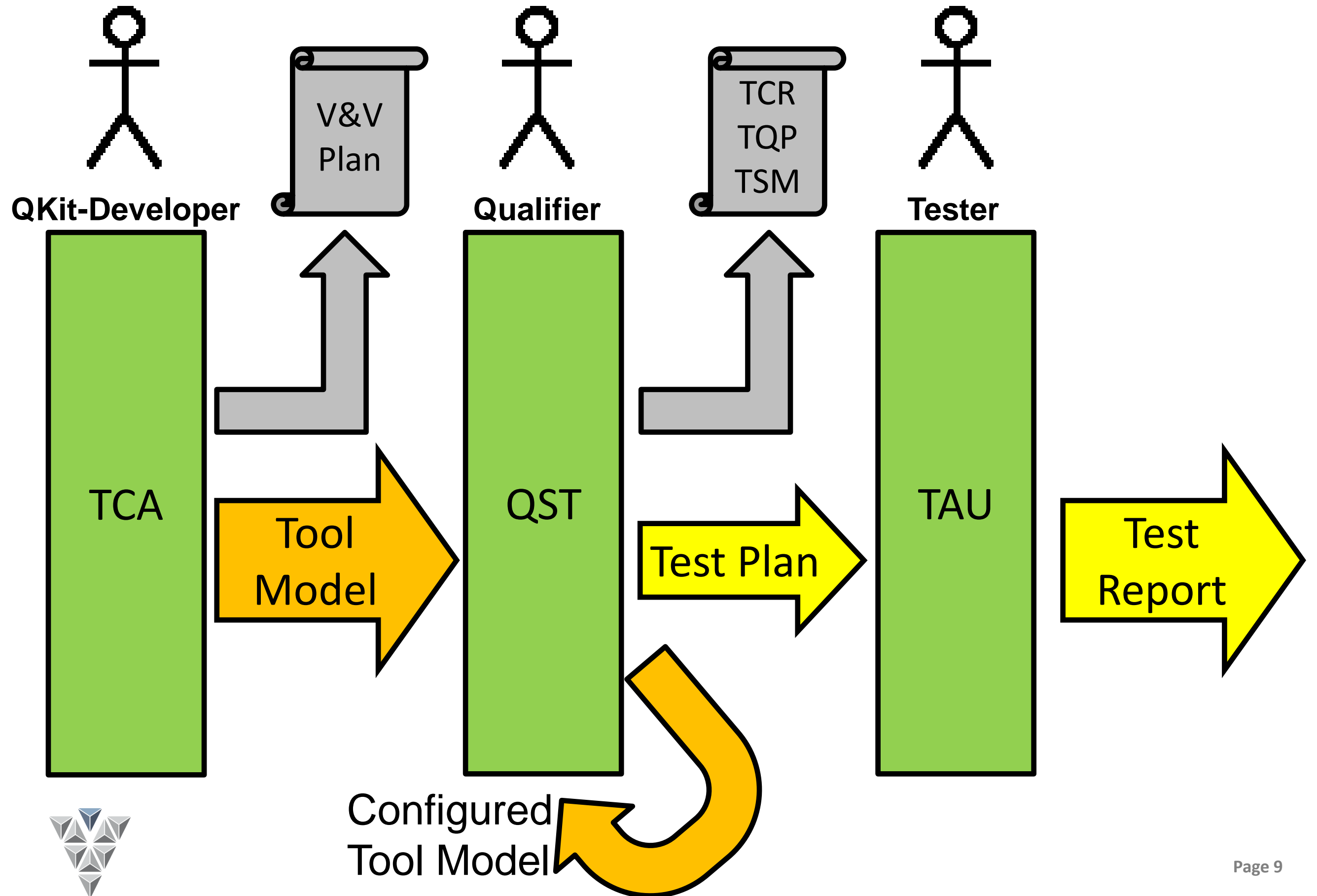


Qualification Documents



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

Tools & Generated Artifacts



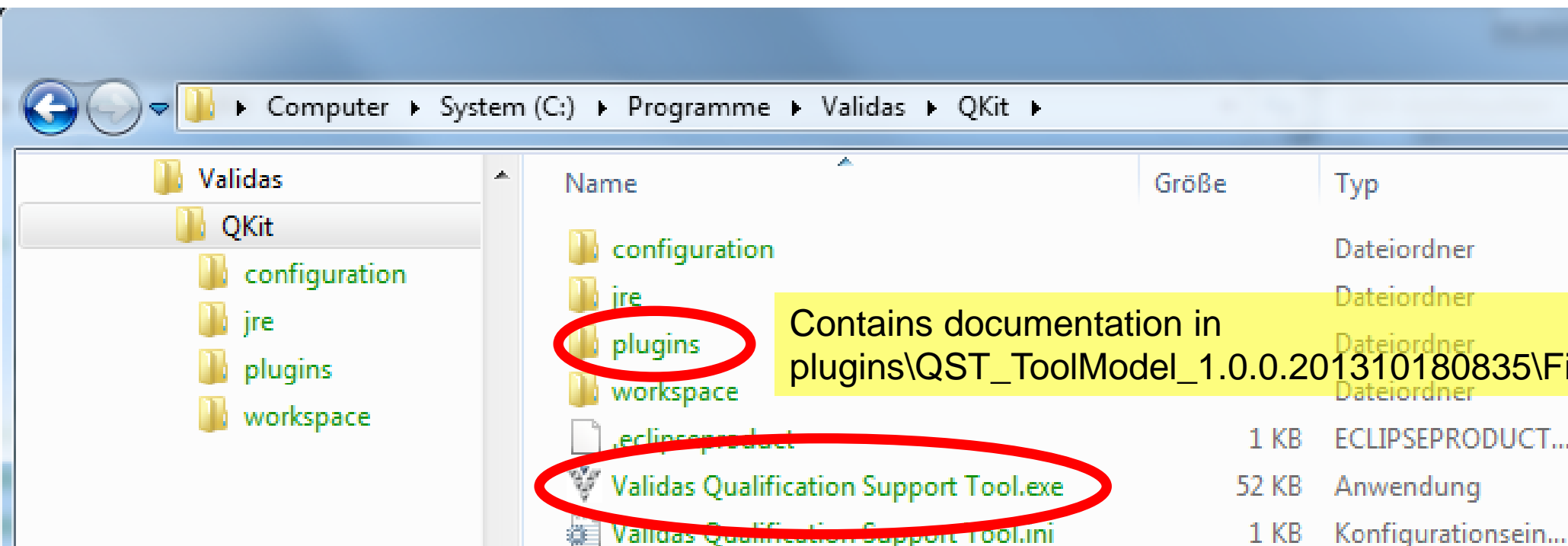
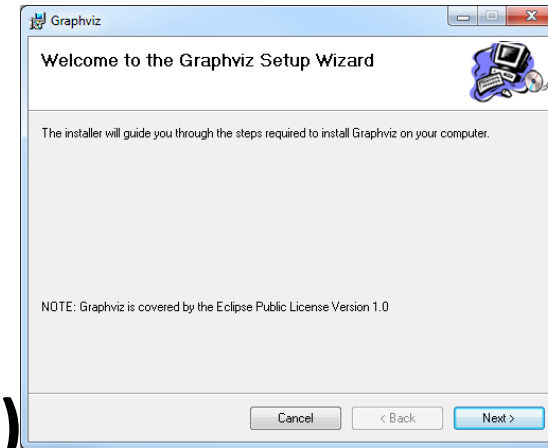
Content

- ▶ 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

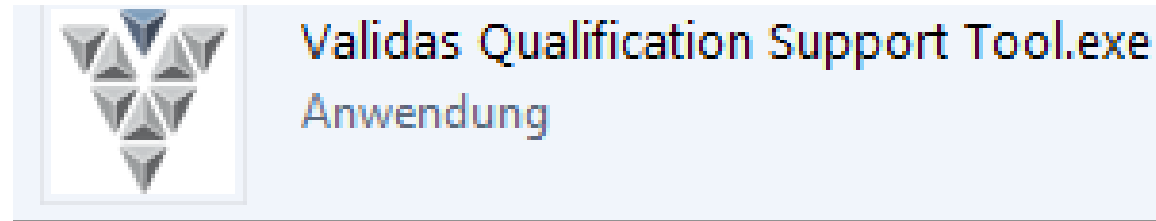
- ▶ Overview
- ▶ Qualification Method
- ▶ **Example: First Qualification Steps**
 - Installation
 - **Qualification Planning**
 - Text Execution & Analysis
- ▶ Availability and feedback



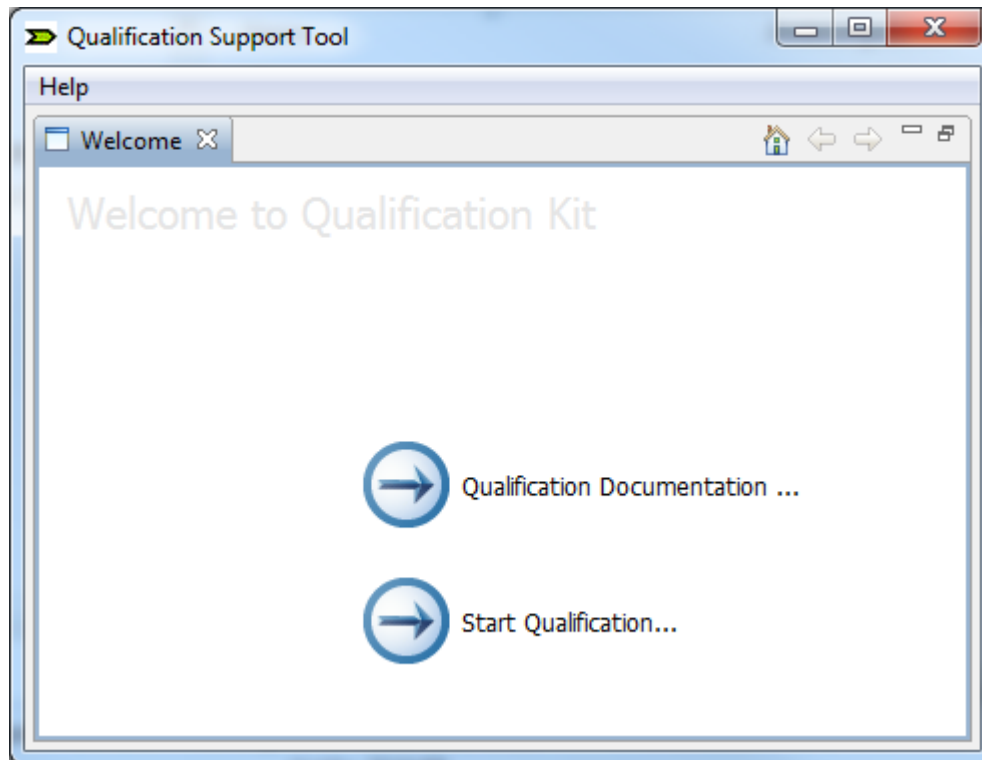
Start QST

- ▶ Details can be found in documentation (QKit User Manual)

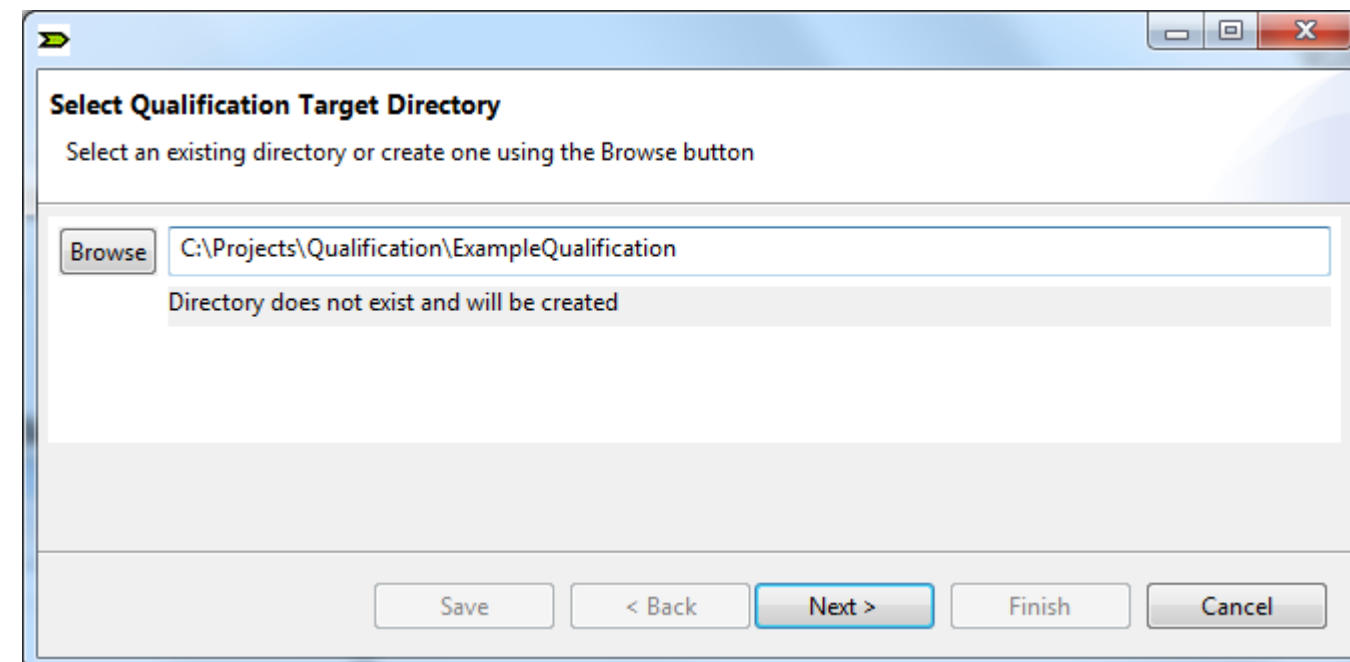
- ▶ Double click on



- ▶ Click on Start Qualification



- ▶ 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

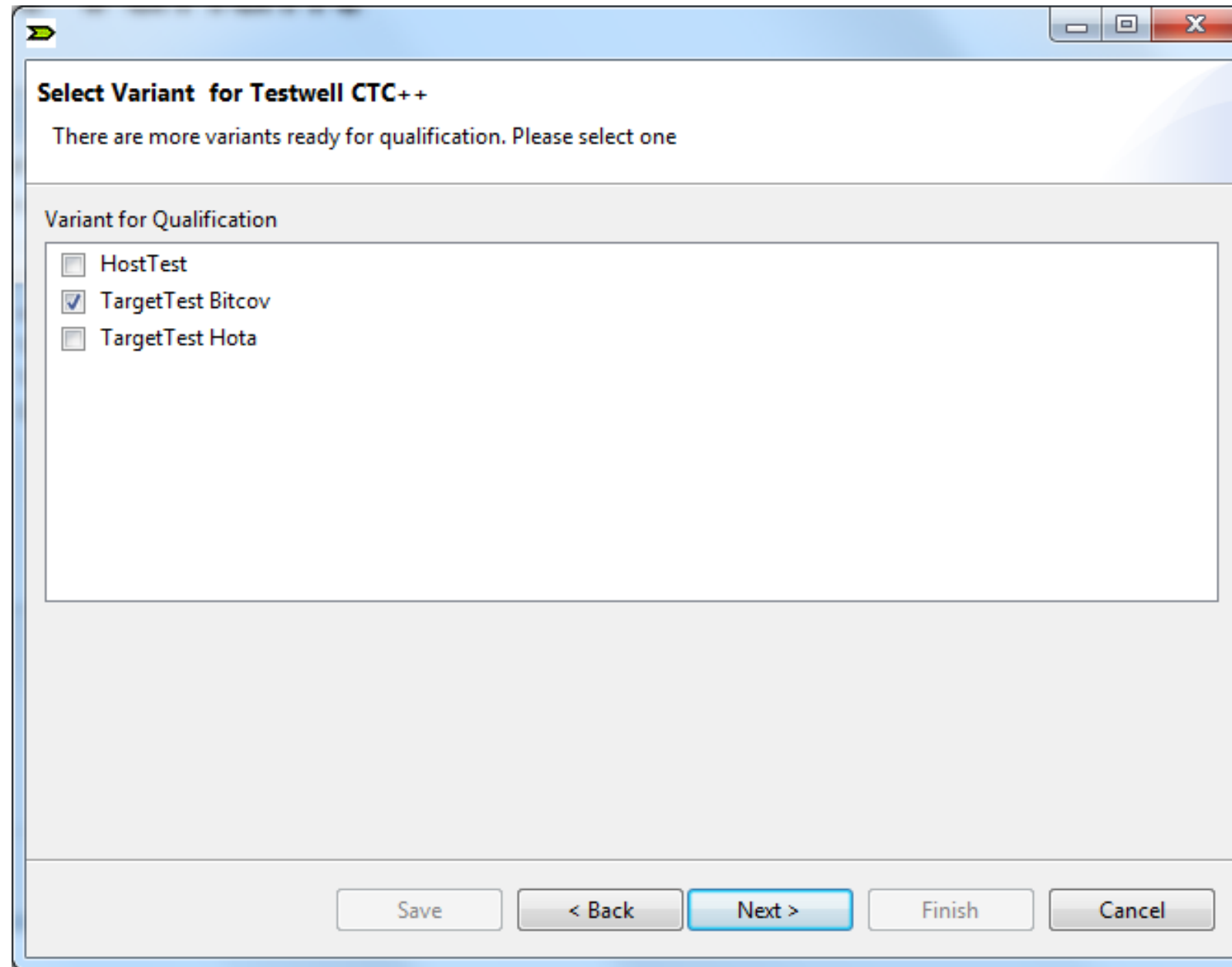
Standard for Qualification

- ISO26262
- IEC61508
- EN50128
- DO178C
- DO330

Save < Back Next > Finish Cancel



Select Variant



The screenshot shows a Windows-style dialog box titled "Select Variant for Testwell CTC++". The dialog has a title bar with a green arrow icon on the left and standard minimize, maximize, and close buttons on the right. The main content area contains the text "There are more variants ready for qualification. Please select one" and a section titled "Variant for Qualification" with three checkboxes: "HostTest", "TargetTest Bitcov" (which is checked), and "TargetTest Hota". At the bottom of the dialog, there are five buttons: "Save", "< Back", "Next >" (highlighted in blue), "Finish", and "Cancel".

Select Variant for Testwell CTC++

There are more variants ready for qualification. Please select one

Variant for Qualification

- HostTest
- 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)

Use Case Definition
Define your Use Case of "CTC" for qualification: Select Features and Mitigations

Use Case Target with Bitcov with 29 Features and 55 Errors: 0 Tested, 55 Mitigated, 0 Untested. Known Bugs: 6/6 mitigated

Identification
Name: Target with Bitcov

Features

- CTC
 - Counters (Selection is mitigatable)
 - Counter Size 1 Bit (mitigatable)
 - Coverage (Selection is mitigatable)
 - Condition (mitigatable)
 - Decision (mitigatable)
 - Function (mitigatable)
 - MC/DC (mitigatable)
 - Multicondition (mitigatable)
 - Statement (mitigatable)
 - ctc.exe (Selection is mitigatable)
 - CodeExclusion (mitigatable)
 - Instrumentation (Selection is mitigatable)
 - i d (mitigatable)
 - i f (mitigatable)
 - i m (mitigatable)
 - ctc2dat.exe (mitigatable)
 - ctc2html.bat (mitigatable)
 - ctc2static.bat (mitigatable)
 - ctcpost.exe (Selection is mitigatable)
 - fc (mitigatable)
 - fd (mitigatable)
 - ff (mitigatable)
 - fmcdc (mitigatable)
 - p (mitigatable)
 - t (mitigatable)
 - u (mitigatable)
 - x (mitigatable)
 - FileExclusion (mitigatable)

Potential Errors

- CTC
 - Incomplete Instrumentation of ? Operator (mitigated)
 - Instrumentation only in Control Structures (mitigated)
 - Non Compilable Code (mitigated)
 - Non Compilable String Code (mitigated)
 - Tool Crash of Ctcpost (mitigated)
 - Uninstrumented Global Code (mitigated)
 - Uninstrumented or Non Compiling Code for ^ (mitigated)
 - Counters (mitigated)
 - Counter Size 1 Bit (mitigated)
 - Coverage (mitigated)
 - Inferred Feature Error Condition Coverage Too High in Condition (mitigated)
 - Inferred Feature Error Condition Coverage Too Low in Condition (mitigated)
 - Inferred Feature Error Decision Coverage Too High in Decision (mitigated)
 - Inferred Feature Error Decision Coverage Too Low in Decision (mitigated)
 - Inferred Feature Error Function Coverage Too High in Function (mitigated)
 - Inferred Feature Error Function Coverage Too Low in Function (mitigated)
 - Inferred Feature Error MC/DC Coverage Too High in MC/DC (mitigated)
 - Inferred Feature Error MC/DC Coverage Too Low in MC/DC (mitigated)
 - Inferred Feature Error MultiCondition Coverage Too High in Multicondition (mitigated)
 - Inferred Feature Error MultiCondition Coverage Too Low in Multicondition (mitigated)
 - Inferred Feature Error Statement Coverage Too High in Statement (mitigated)
 - Inferred Feature Error Statement Coverage Too Low in Statement (mitigated)
 - Inferred Feature Error Wrong Behaviour in Statement (mitigated)
 - Inferred Feature Error Wrong Behaviour in Condition (mitigated)
 - Inferred Feature Error Wrong Behaviour in Decision (mitigated)
 - Inferred Feature Error Wrong Behaviour in MC/DC (mitigated)
 - Inferred Feature Error Wrong Behaviour in Multicondition (mitigated)
 - Inferred Feature Error Wrong Behaviour in Function (mitigated)

Mitigations

- CTC
 - Safety Guidelines
 - SG_ctc
 - SG_CodeExclusion
 - SG_CounterLocation
 - SG_CoverageView
 - SG_CoverageViewCondition
 - SG_CoverageViewDecision
 - SG_CoverageViewFunction
 - SG_CoverageViewMCDC
 - SG_FileSelection
 - SG_General_CoverageAnalysis
 - SG_General_KnownBugs
 - SG_InstrVerification
 - SG_ReportUntested
 - SG_ReportXML
 - SG_StatementCoverage
 - SG_ctc2dat
 - SG_ctc2html.bat
 - SG_ctc_twice
 - SG_dmp2txt
 - Test Bed
 - Safety Guidelines
 - SG_Test_Compare

Buttons: Save, < Back, Next >, Finish, Cancel

Configure Use Case

- ▶ Use Case has Features
- ▶ Features have pot. Errors and known bugs
- ▶ Errors have Mitigations

- Checks
- Restrictions

- ▶ **Status: Next disabled, i.e. not enough mitigations selected**

Use Case Definition
Define your Use Case of "CTC" for qualification: Select Features and Mitigations

Use Case Target with Bitcov with 18 Features and 42 Errors: 0 Tested, 40 Mitigated, 2 Untested. Known Bugs: 8/8 mitigated

Identification
Name: Target with Bitcov

Features

- CTC
 - Counters (Selection is mitigatable)
 - Counter Size 1 Bit (mitigatable)
 - Coverage (Selection is mitigatable)
 - Condition (mitigatable)
 - Decision (mitigatable)
 - Function (mitigatable)
 - MC/DC (mitigatable)
 - Multicondition (mitigatable)
 - Statement (mitigatable)
 - ctc.exe (Selection is mitigatable)
 - CodeExclusion (mitigatable)
 - Instrumentation (Selection is mitigatable)
 - InstrOptions (Selection is mitigatable)
 - i d (mitigatable)
 - i f (mitigatable)
 - i m (mitigatable)
 - ctc2dat.exe (mitigatable)
 - ctc2html.bat (mitigatable)
 - ctcpost.exe (Selection is mitigatable)
 - fc (mitigatable)
 - fd (mitigatable)
 - ff (mitigatable)
 - fmcdc (mitigatable)

Potential Errors

- CTC
 - Incomplete Instrumentation of ? Operator (mitigated)
 - Instrumentation only in Control Structures (mitigated)
 - Non Compilable Code (mitigated)
 - Non Compilable String Code (mitigated)
 - Tool Crash of Ctcpost (mitigated)
 - Uninstrumented Anonymous Function in C++ (mitigated)
 - Uninstrumented Global Code (mitigated)
 - Uninstrumented Member Function in C++ (mitigated)
 - Uninstrumented or Non Compiling Code for ^ (mitigated)
 - Counters (mitigated)
 - Counter Size 1 Bit (mitigated)
 - Coverage (mitigated)
 - Function (mitigated)
 - MC/DC (mitigated)
 - ctc.exe (mitigated)
 - Instrumentation (mitigated)
 - ctc2dat.exe (mitigated)
 - Coverage too Low due to ctc2dat (mitigated)
 - MC/DC Coverage too High due to ctc2dat (mitigated)
 - Other Coverage too High due to ctc2dat (mitigated)
 - ctc2html.bat (mitigated)
 - Coverage too Low from ctc2html (mitigated)
 - MC/DC Coverage too High from ctc2html (mitigated)

Mitigations

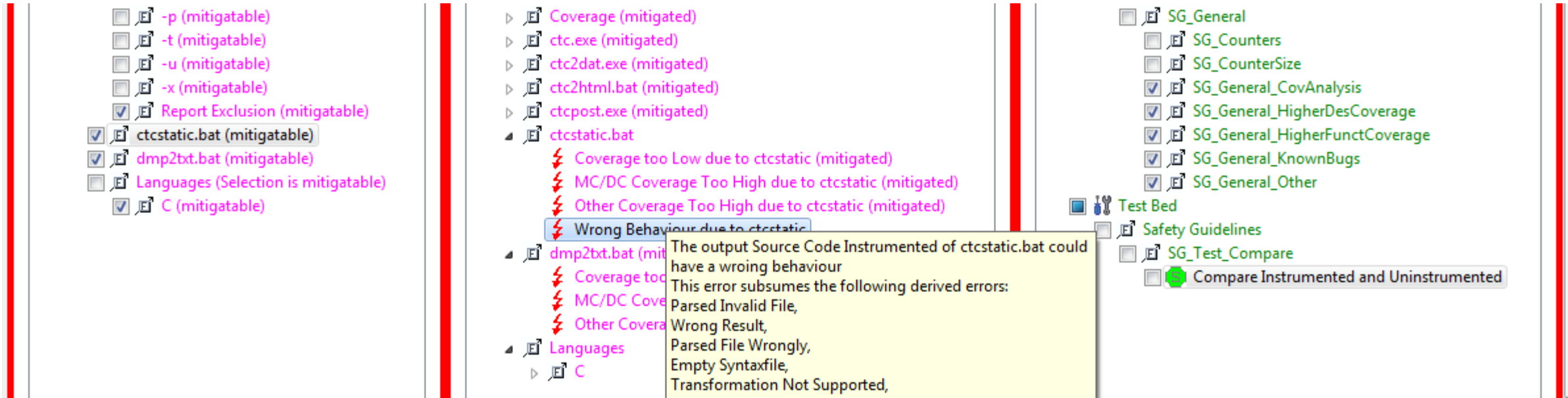
- CTC
 - Safety Guidelines
 - SG_ctc
 - SG_ctc2dat
 - SG_ctc2dat_FileListCheck
 - SG_ctc2dat_SyntaxCheck
 - SG_ctc2dat_Verification
 - SG_ctc2html.bat
 - SG_ctc_twice
 - SG_ctcpost
 - SG_dmp2txt.bat
 - SG_General
 - SG_Counters
 - SG_CounterSize
 - SG_General_CovAnalysis
 - SG_General_HigherDesCoverage
 - SG_General_HigherFunctCoverage
 - SG_General_KnownBugs
 - SG_General_Other
- Test Bed
 - Safety Guidelines
 - SG_Test_Compare

Buttons: Save, < Back, Next >, Finish, Cancel



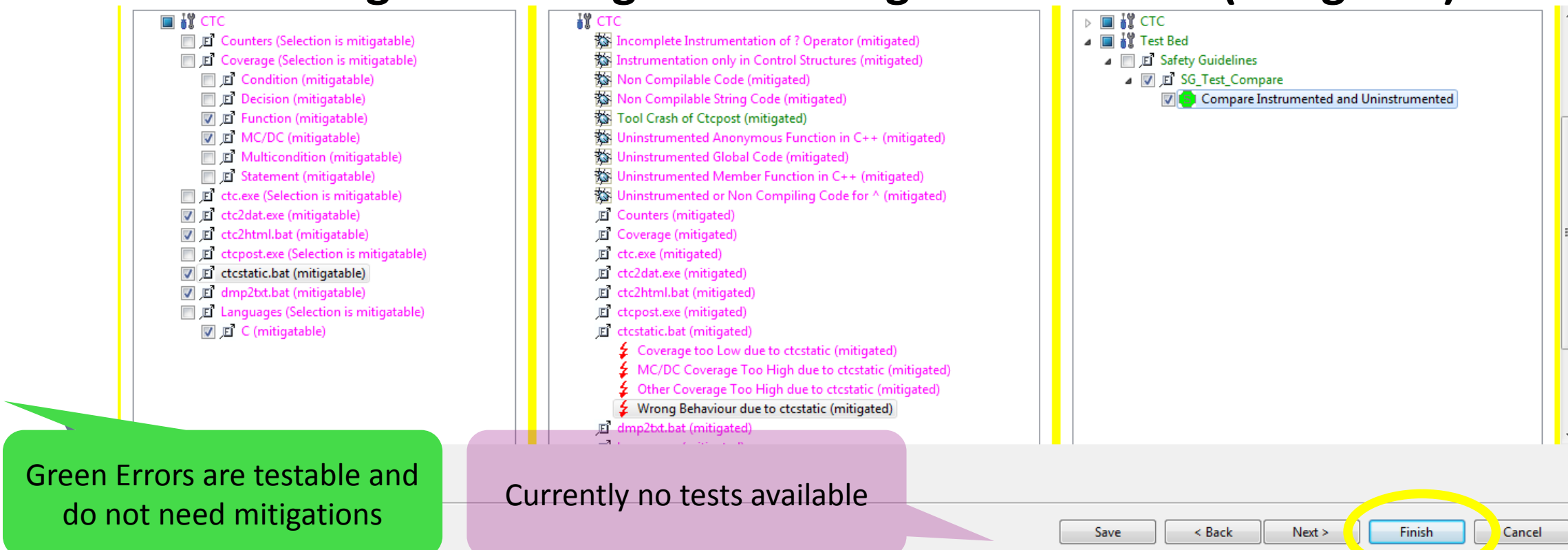
Configure Use Case: Error Mitigation

▶ Select Error: available mitigations get highlighted



The screenshot shows a configuration window with a tree view on the left and a list of errors on the right. The tree view includes categories like 'Report Exclusion', 'ctcstatic.bat', 'dmp2txt.bat', 'Languages', and 'C'. The error list on the right shows various errors, with 'Wrong Behaviour due to ctcstatic' selected. A tooltip for this error explains that the output source code instrumented of ctcstatic.bat could have a wrong behaviour and lists derived errors: 'Parsed Invalid File', 'Wrong Result', 'Parsed File Wrongly', 'Empty Syntaxfile', and 'Transformation Not Supported'. On the right side, there are checkboxes for 'SG_General' and 'SG_Test_Compare', with 'Compare Instrumented and Uninstrumented' checked.

▶ Choose mitigation: mitigated errors get marked as “(mitigated)”



The screenshot shows the same configuration window as above, but now all errors in the list are marked as '(mitigated)'. A green callout bubble on the left says 'Green Errors are testable and do not need mitigations'. A purple callout bubble in the center says 'Currently no tests available'. At the bottom, there are buttons for 'Save', '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Finish' button is highlighted with a yellow circle.

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

The screenshot shows the 'Use Case Definition' window in the Verifysoft software. The window title is 'Use Case Definition' and the subtitle is 'Define your Use Case of "CTC" for qualification: Select Features and Mitigations'. The main content area is titled 'Use Case Target with Bitcov with 15 Features and 37 Errors: 0 Tested, 37 Mitigated, 0 Untested. Known Bugs: 8/8 mitigated'. The window is divided into three main sections: 'Identification', 'Features', and 'Potential Errors'. The 'Identification' section shows the name 'Target with Bitcov'. The 'Features' section lists 15 features, including 'CTC', 'Counters', 'Coverage', 'Condition', 'Decision', 'Function', 'MC/DC', 'Multicondition', 'Statement', 'ctc.exe', 'CodeExclusion', 'InstrOptions', 'Instrumentation', 'ctc2dat.exe', 'ctc2html.bat', 'ctcpost.exe', 'ctcstatic.bat', 'dmp2txt.bat', 'Languages', and 'C'. The 'Potential Errors' section lists 37 errors, all of which are marked as 'mitigated'. The 'Mitigations' section shows a tree view of mitigations, including 'Safety Guidelines', 'SG_ctc', 'SG_ctc2dat', 'SG_ctc2html.bat', 'SG_ctc_twice', 'SG_ctcpost', 'SG_dmp2txt.bat', 'SG_General', 'SG_Counters', 'SG_CounterSize', 'SG_General_CovAnalysis', 'Analyze Code Coverage', 'SG_General_HigherDesCoverage', 'SG_General_HigherFunctCoverage', 'SG_General_KnownBugs', 'Known Bugs Check', 'SG_General_Other', and 'SG_Gen_OtherCoverage'. The 'Next >' button is highlighted with a yellow circle.



Optional: Planning of Roles

- ▶ Select roles in the Qualification Project (left side)
- ▶ Assign persons names (right sides) by entering their names

Manage Qualification Project

Manage the Qualification Project

The screenshot displays the 'Manage Qualification Project' interface. On the left, a tree view under 'Qualification Plan' shows a list of roles: 'Qualification Expert (Oscar Slotosch)', 'Tool User', 'User Safety Expert (My FuSi)', 'Tool Provider (Olavi Poutanen)', and 'Qualifier'. Below these are organizational phases and documents. On the right, the 'Qualification Role Settings' panel is active, showing 'Assigned Person' set to 'My FuSi'. The 'Inheritance' section is empty, and the 'Description' section contains text about the safety expert's role.

Manage Qualification Project

Qualification Plan

- CTC
 - Qualification Kit Application
 - Qualification Expert (Oscar Slotosch)
 - Tool User
 - User Safety Expert (My FuSi)
 - Tool Provider (Olavi Poutanen)
 - Qualifier
 - 1: Organization Phase
 - 2: Classification and Preparation
 - Tool Classification Report
 - Tool Safety Manual
 - Feature List
 - Tool Definition
 - Model
 - Qualification Kit
 - QKit Manual

Qualification Role Settings

Assigned Person: My FuSi

Inheritance

Description

The safety expert ensures that the qualification fit's to the requirements of the user. This is done by selecting / confirming the mitigations for the used tool features and by reviewing the qualification documents.



Optional: Planning of Steps

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

Qualification Plan

The screenshot displays the 'Qualification Plan' interface. On the left, a tree view shows the project structure under 'CTC' and 'Qualification Kit Application'. The selected step is '2-4: Generate Preliminary Documents'. On the right, the 'Qualification Step Settings' panel is visible, showing fields for 'Planned Date' (28 November 2013), 'Finished Date', and 'Qualification Role' (Qualifier). A calendar widget is open, showing the date 28.11.2013. Below the settings, the 'Inheritance' section shows 'Planned Date: No inheritance needed', 'Finished Date: No date set in super steps', and 'Qualification Role: No inheritance needed'. The 'Description' section contains the text: 'Based on features and mitigations the QST tool can generate the documents'.

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

The screenshot displays the 'Qualification Plan' interface. On the left, a tree view under 'CTC' shows a 'Qualification Kit Application' with several sub-items: 'Qualification Expert (Oscar Slotosch)', 'Tool User', 'User Safety Expert (My FuSi)', 'Tool Provider (Olavi Poutanen)', 'Qualifier', '1: Organization Phase', '2: Classification and Preparation', 'Tool Classification Report', 'Tool Safety Manual', 'Feature List', 'Tool Definition' (highlighted), 'Model', 'Qualification Kit', and 'QKit Manual'. On the right, a settings panel for 'Tool Definition' is visible, with a 'Path' field containing '<Qualification>/Validation/Documentation/ToolDef.pdf'. Below the path field are sections for 'Inheritance' and 'Description'. The description reads: 'Defines the version, configuration and option of the tool. It is contained in the model of the tool model in the following places:'.

- ▶ **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\CTCExample
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

Name:	Path:
Tool Classification Report	C:\Programme\Qualification\CTCExample\Validation\Documentation\TCR.docx
Tool Qualification Plan	C:\Programme\Qualification\CTCExample\Validation\Documentation\TQP.docx
Tool Safety Manual	C:\Programme\Qualification\CTCExample\Validation\Documentation\TSM.docx
Tool Test Plan	C:\Programme\Qualification\CTCExample\Validation\TestExecution.txt
Toolchain Model	C:\Programme\Qualification\CTCExample\QKit\Model\Model.tca

Save < Back Next > Finish Cancel

Progress Information

 Generating images

Generating image 17 of 35 [Tool_linker_use_link_linker_a_ErrorView]

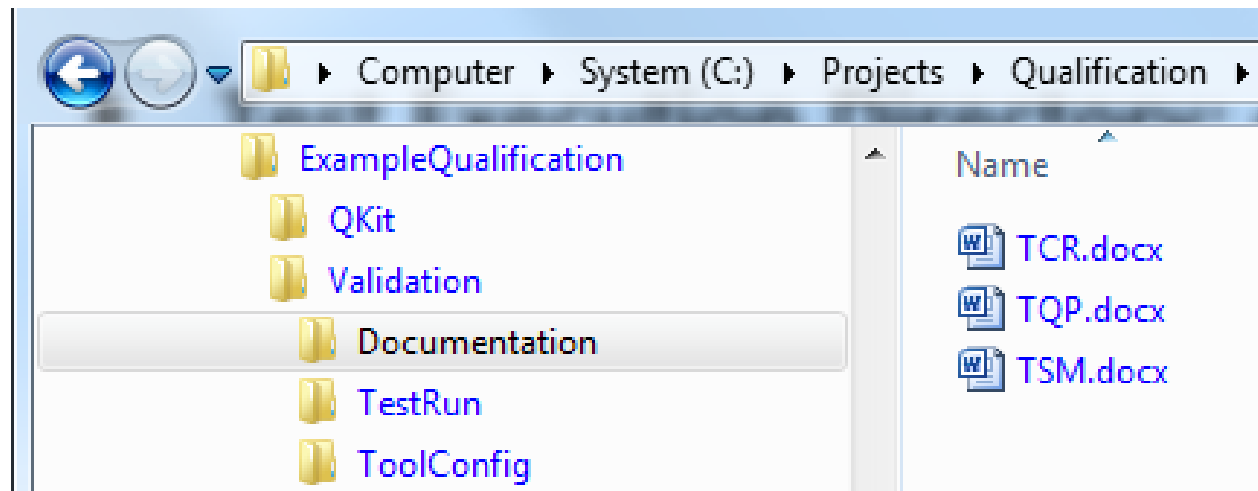
Cancel

► Pressing Finish starts generation of documents



Generated Documents

► Qualification Documents



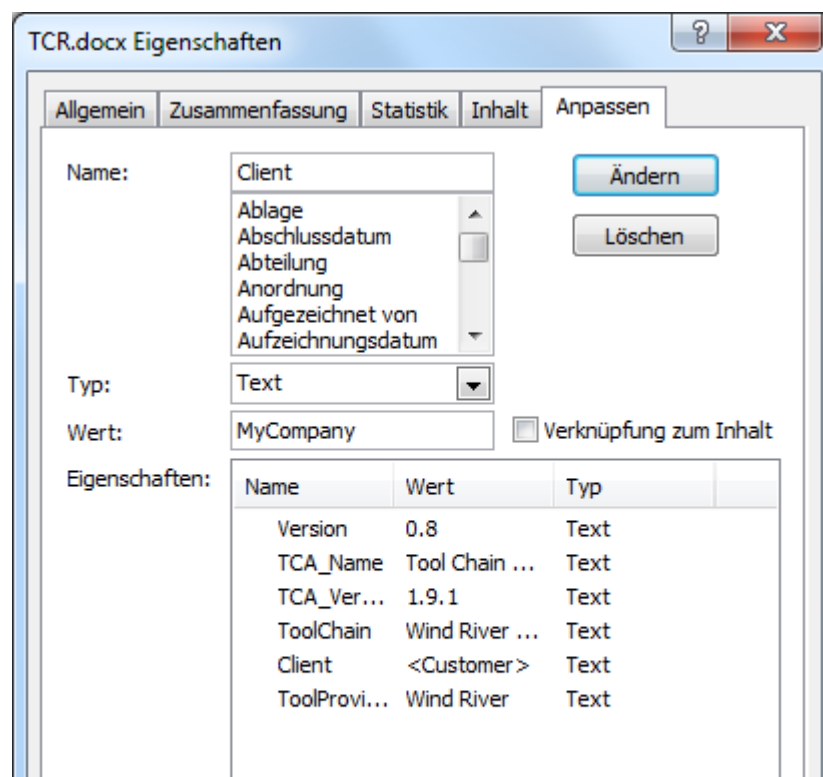
► Set the Word variable Client with your company/project name

► Update the document (Strg+a,F9)

A screenshot of a Notepad++ window showing a test plan document. The title bar is "C:\Program Files\Qualification\ExampleQualification\Validation\TestExecution.txt - Notepad++". The document content is as follows:

```
1 #
2 # This is a test plan generated from the Tool Chain Analyzer
3 #
4 # It covers 1 tool and 1 use case in 1 directories:
5 # - CTC
6 # - Host of CTC
7 #
8 TS_ROOTDIRS
9 <Testsuite>=C:\Programme\Qualification\ExampleQualification\QKit\Testsuite
10 #
11 TESTRUN_ROOTDIR
12 C:\Programme\Qualification\ExampleQualification\Validation\TestRun
13 #
14 TESTS
15 #
16 # DIRECTORY_NUMBER: 1
17 # DIRECTORY_PATH: <Testsuite>\General
18 #
19 <Testsuite>\General
20 #
21 # the directory contains the
22 # - 1. General
23 #
24 # TEST_NUMBER: 1 / 1
25 # TEST_ID: CTC.Testsuite-Gen
26 # TEST_NAME: General
27 # TESTED_FEATURE: ctc.exe
28 # TESTED_USECASE: Host
29 #
30 # TEST_DESCRIPTION
31 # Some general test cases
32 # COMMENT: Currently used to test the TAU
33 # END_TEST_DESCRIPTION
34 #
35 #
36 #
37
```

Test plan in
ExampleQualification/
Validation/TestExecution



Content

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



Test Selection

- ▶ Some features are testable (green)

Use Case Definition
Define your Use Case of "CTC" for qualification: Select Features and Mitigations

Features

- CTC
 - Counters (Selection is mitigatable)
 - Counter Size unsigned long (mitigatable)
 - Safe Counters (mitigatable)
 - Coverage (Selection is mitigatable)
 - Condition (mitigatable)
 - Decision (mitigatable)
 - Function (mitigatable)
 - MC/DC (mitigatable)
 - Multicondition (mitigatable)
 - Statement (testable)
 - ctc.exe (Selection is mitigatable)
 - CodeExclusion (mitigatable)
 - Instrumentation (Selection is mitigatable)
 - i d (mitigatable)
 - i f (mitigatable)
 - i m (mitigatable)
 - ctc2html.bat (mitigatable)
 - ctcpost.exe (Selection is mitigatable)
 - fc (mitigatable)
 - fd (mitigatable)
 - ff (mitigatable)
 - fmcdc (mitigatable)
 - p (mitigatable)
 - t (mitigatable)
 - u (mitigatable)
 - x (mitigatable)

Potential Errors

- CTC
 - Incomplete Instrumentation of ? Operator (mitigated)
 - Instrumentation only in Control Structures (mitigated)
 - Non Compilable Code (mitigated)
 - Non Compilable String Code (mitigated)
 - Tool Crash of Ctcpost (mitigated)
 - Uninstrumented Global Code (mitigated)
 - Uninstrumented or Non Compiling Code for ^ (mitigated)
 - Counters (mitigated)
 - Counter Size unsigned long (mitigated)
 - Coverage (mitigated)
 - ctc.exe (mitigated)
 - Coverage too Low (testable)
 - Wrong Behaviour (testable)
 - Default error for wrong output Executable of ctc.exe
 - Tested by: General
 - CodeExclusion (mitigated)
 - Instrumentation (mitigated)
 - Inferred Feature Error Coverage too Low in ctc.exe (mitigated)
 - Instrumentation (mitigated)
 - Inferred Feature Error Wrong Behaviour in ctc.exe (mitigated)
 - Wrong Instrumentation (mitigated)
 - i d (mitigated)
 - i f (mitigated)
 - Inferred Feature Error Coverage too Low in ctc.exe (mitigated)
 - Function Coverage too High (mitigated)
 - Function Instrumentation not Working (mitigated)
 - Inferred Feature Error Instrumentation Option Not Working in Instrumentation (mitigated)
 - Inferred Feature Error Wrong Behaviour in ctc.exe (mitigated)

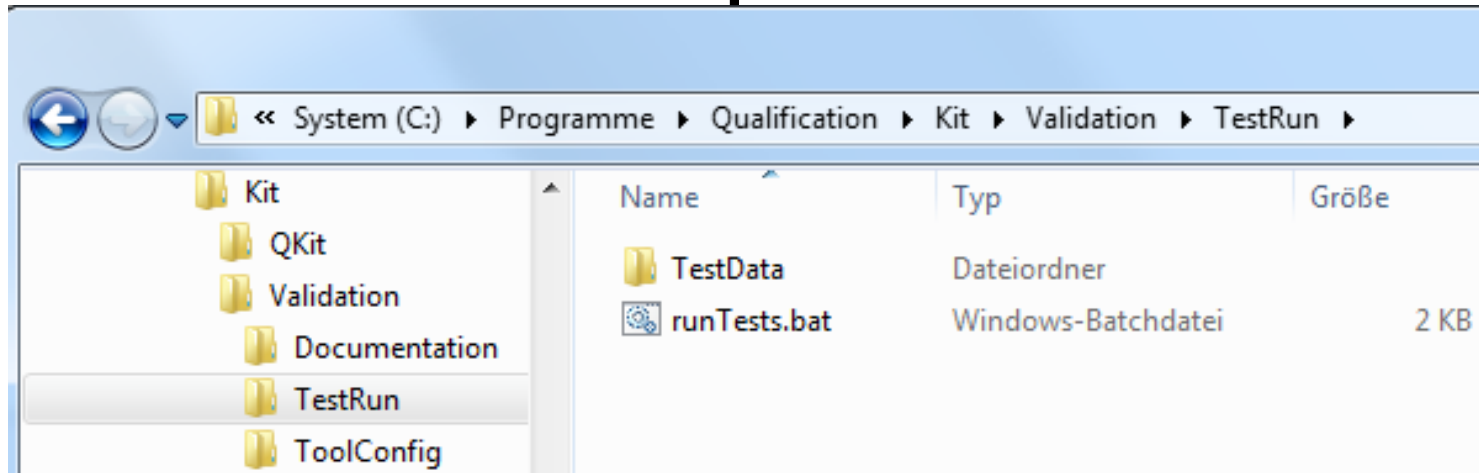
Mitigations

- Safety Guidelines
 - SG_ctc
 - SG_CodeExclusion
 - SG_CounterLocation
 - SG_CoverageView
 - SG_CoverageViewCondition
 - SG_CoverageViewDecision
 - SG_CoverageViewFunction
 - SG_CoverageViewMCDC
 - SG_FileSelection
 - SG_General_CoverageAnalysis
 - Analyze Code Coverage
 - SG_General_KnownBugs
 - Known Bugs Check
 - Instrumentation Verification
 - SG_ReportUntested
 - SG_ReportXML
 - SG_SafeCounters
 - SG_StatementCoverage
 - SG_ctc2html.bat
 - SG_ctc_twice
 - Execute Code Twice (With/Without Instrumentation)
- Safety Guidelines
 - SG_Test_Compars

Buttons: Save, < Back, Next >, Finish, Cancel

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

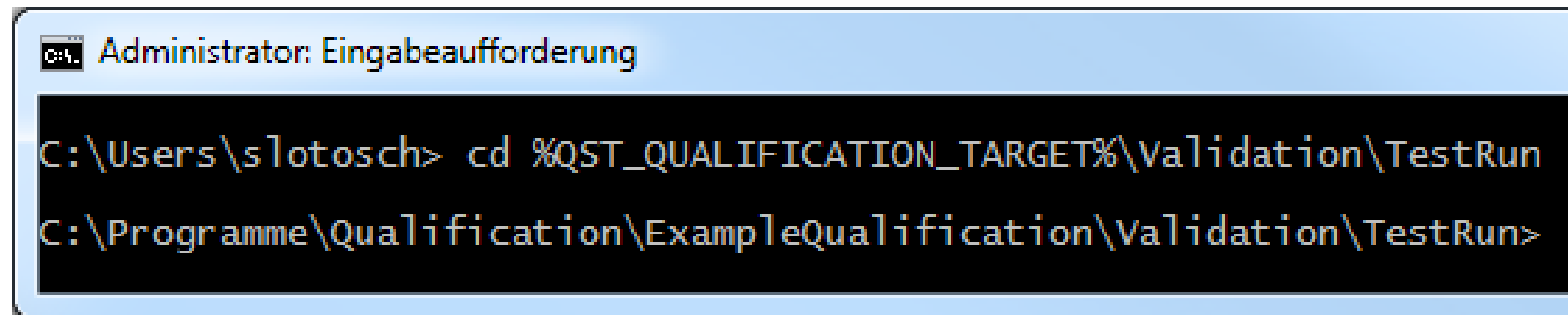
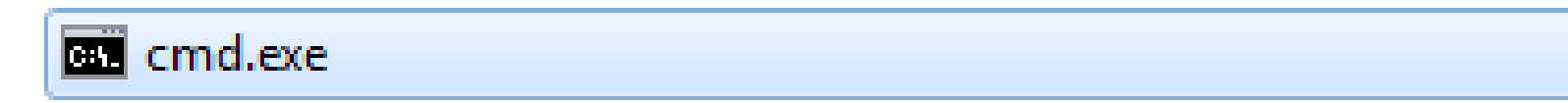
```
*C:\Program Files\Qualification\Kit\Validation\TestRun\runTests.bat - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
TestExecution.txt runTests.bat
1 @echo off
2 REM
3 REM This is the main script to run the tests
4 REM
5 REM first configure it by setting the path variables
6 REM then start it such that creates a logfile, e.g.
7 REM
8 REM .\runTests.bat ..\TestExecution.txt ..\ToolConfig\tool_config.py
9 REM
10 REM
11 REM configurable section
12 REM
13 REM usually it should suffice to configure JAVA_HOME
14 REM
15 SET QUALIFICATION_DIR=%~dp0\..\
16 SET JAVA_HOME=C:\Program Files\Validas\QKit\jre
17 SET PYTHONPATH=%QUALIFICATION_DIR%\QKit\TAU\IntraTAU;%QUALIFICATION_DIR%\QKit\TAU\IntraTAU;%~dp2
18 SET TEST_PLAN=%~dpnx1
19 SET CONFIG_FILE=%~dpnx2
```

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`

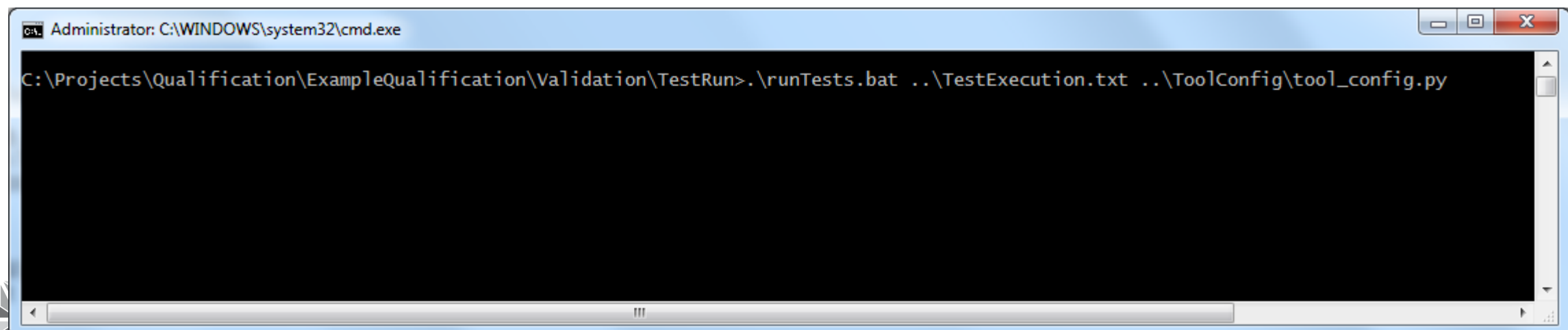
Programme (1)



```
C:\Users\s1otosch> cd %QST_QUALIFICATION_TARGET%\Validation\TestRun  
C:\Programme\Qualification\ExampleQualification\Validation\TestRun>
```

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

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



```
Administrator: C:\WINDOWS\system32\cmd.exe  
C:\Projects\Qualification\ExampleQualification\Validation\TestRun>.\runTests.bat ..\TestExecution.txt ..\ToolConfig\tool_config.py
```



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\..\..\QKit\TAU\InterTAU;c:\Programme
\Qualification\ExampleQualification\Validation\TestRun\..\..\QKit\TAU\IntraTAU;c:\Programme\Qualification\ExampleQual
ification\Validation\ToolConfig\]
[TEST_PLAN]=[c:\Programme\Qualification\ExampleQualification\Validation\TestExecution.txt]
[CONFIG_FILE]=[c:\Programme\Qualification\ExampleQualification\Validation\ToolConfig\tool_config.py]
[TOOL_NAME]=[ctc]
Executing job command: "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\..\QKit\TAU\IntraTAU"\t
crexexpand.bat model.c > test.c
impSTDOUT: STDOUT: b''
STDERR: STDERR: None
Executing job command: ctc -i m gcc test.c -I "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\
..\QKit\TAU\IntraTAU" "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\..\QKit\TAU\IntraTAU"\tr
acer.c -o test.exe
STDOUT: STDOUT: b''
STDERR: STDERR: None
Executing job command: test.exe > test.out
STDOUT: STDOUT: b''
STDERR: STDERR: None
Executing job command: ctcpst 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.

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

General	09.02.2014 19:45	Dateiordner
TestData	09.02.2014 19:45	Dateiordner
TestReport	09.02.2014 19:45	Dateiordner
runTests.bat		
runTests.log		
runTests.py		

Build-Folder (after test execution):

Name	Änderungsdatum	Typ
__pycache__	09.02.2014 19:45	Dateiordner
expected_profile-id.txt	09.02.2014 18:55	TXT-Datei
expected_profile-if.txt	09.02.2014 18:55	TXT-Datei
expected_profile-im.txt	09.02.2014 18:55	TXT-Datei
expected_trace.txt	09.02.2014 18:55	TXT-Datei
model.c	09.02.2014 18:55	C-Datei
MON.dat	09.02.2014 19:45	DAT-Datei
MON.sym	09.02.2014 19:45	SYM-Datei
profile.txt	09.02.2014 19:45	TXT-Datei
readme.txt	09.02.2014 18:55	TXT-Datei
runTst.bat	09.02.2014 19:45	Windows-Batchda...
test.c	09.02.2014 19:45	C-Datei
test.exe	09.02.2014 19:45	Anwendung
test.o	09.02.2014 19:45	O-Datei
test.out	09.02.2014 19:45	OUT-Datei
tool_config.py	09.02.2014 19:45	Python File
tracer.o	09.02.2014 19:45	O-Datei
Tst.py	09.02.2014 18:55	Python File

runTst.bat: Script for re-execution of this single test



Test Report: Overview

Unit Test Results + Test Report: Overview

file:///C:/Program Files/Qualification/ExampleQualification/Validation/TestRun/TestReport/junit-noframes.html

Google

Unit Test Results

Designed for use with [JUnit](#) and [Ant](#).

Tool: CTC
TAU Version: 0.1
Qualification directory: c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\.
Java home: C:\Program Files\Java\jre7
Pythonpath: c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\..\QKit\TAU\InterTAU;c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\..\QKit\TAU\IntraTAU;c:\Programme\Qualification\ExampleQualification\Validation\ToolConfig\
Test plan: c:\Programme\Qualification\ExampleQualification\Validation\TestExecution.txt
Config file: c:\Programme\Qualification\ExampleQualification\Validation\ToolConfig\tool_config.py
User: slotosch
Date: Sun Feb 09 19:45:20 CET 2014

Summary

Tests	Failures	Errors	Success rate	Time
2	1	0	50.00%	2.274

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

Package

Name	Tests	Errors	Failures	Skipped[0/1]	Time(s)
General	2	0	1	0	2.274

[Back to top](#)

TestCase General

Name	Status	Type	Time(s)
testDiffOutFiles	Success		0.001
testDiffReportFiles	Failure	N/A <pre>False is not true : Generated and reference coverage files differ. File "C:\Python32\lib\unittest\case.py", line 370, in _executeTestPart function() File "C:\Programme\Qualification\ExampleQualification\Validation\TestRun\General\Tst.py", line 40, in testDiffReportFiles self.assertTrue(boolOk, 'Generated and reference coverage files differ.') File "C:\Python32\lib\unittest\case.py", line 506, in assertTrue raise self.failureException(msg)</pre>	0.002

Test Logfile: TestRun/runTests.log

► Contains test results and commands to reproduce test

STARTED TEST RUN

Start Time = [2014-02-09 19:45:16.406000]

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\..\..\QKit\TAU\InterTAU;c:\Pro

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\QKit\Testsuite\General

Adding test file C:\Programme\Qualification\ExampleQualification\QKit\Testsuite\General\Tst.py

TESTSUITE_DIR: C:\Programme\Qualification\ExampleQualification\QKit\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\QKit\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\..\..\QKit\TAU\Int

STDOUT: STDOUT: b''

STDERR: STDERR: None

Executing job command: ctc -i m gcc test.c -I "c:\Programme\Qualification\ExampleQualification\Validation\Tes

STDOUT: STDOUT: b''

Test Analysis: One Test Failed

► Logfile message explains

```
Executing job command: "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\..\QKi
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: ctcpst 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

profile.txt <-> expected_profile-im.txt - KDiff3

Datei Bearbeiten Ordner Navigation Vergleichsansicht Zusammenführen Fenster Einstellungen Hilfe

A: C:\Program Files\Qualification\ExampleQualification\Validation\TestRun\General\profile.txt
Oberste Zeile 1 Kodierung: System Zeilenende-Kodierung: DOS

B: C:\Program Files\Qualification\ExampleQualification\Validation\TestRun\General\expected_profile-im.txt
Oberste Zeile 1 Kodierung: System Zeilenende-Kodierung: DOS

* CTC++, Test Coverage Analyzer for C/C++, Version 7.1.1 *
* *
* EXECUTION PROFILE LISTING *
* *
* Copyright (c) 1993-2013 Testwell Oy *

License_notice: This is a limited period evaluation copy license.

Symbol_file(s)_used : MON.sym_ (Sun_Feb_09_19:45:17_2014)
Data_file(s)_used : MON.dat_ (Sun_Feb_09_20:00:27_2014)
Listing_produced_at : Sun_Feb_09_20:00:27_2014

Coverage view : As instrumented

MONITORED SOURCE FILE : test.c
INSTRUMENTATION MODE : multicondition

HITS/TRUE	FALSE	LINE DESCRIPTION
4		2 FUNCTION main()
4	0 -	6 if (tr2 (6 , 2 , 1))
4	0 -	7 if (tr2 (7 , 4 , 0) (tr2 (7 , 5 , 1) tr2 (7 ,
0	-	7 1: T (_ _)
4	-	7 2: F (T _)
0	-	7 3: F (F T)
0	0 -	7 4: F (F F)
16	0 -	8 for (;i < tr2 (8 , 8 , 5);)
4	12	10 if (i == tr2 (10 , 10 , 3))
4		10 break
		13 else
		14 L:
4		16 return 1
		17 }

***TER 60 % (9/ 15) of FUNCTION main()
94 % (16/ 17) statement

***TER 60 % (9/ 15) of SOURCE FILE test.c
94 % (16/ 17) statement

MONITORED SOURCE FILE : c:\Programme\Qualification\ExampleQualification\Validation\TestRun
INSTRUMENTATION MODE : multicondition

HITS/TRUE	FALSE	LINE DESCRIPTION
1		2 FUNCTION main()
1	0 -	6 if (tr2 (6 , 2 , 1))
1	0 -	7 if (tr2 (7 , 4 , 0) (tr2 (7 , 5 , 1) tr2 (7 ,
0	-	7 1: T (_ _)
1	-	7 2: F (T _)
0	-	7 3: F (F T)
0	0 -	7 4: F (F F)
4	0 -	8 for (;i < tr2 (8 , 8 , 5);)
1	3	10 if (i == tr2 (10 , 10 , 3))
1		10 break
		13 else
		14 L:
1		16 return 1
		17 }

***TER 60 % (9/ 15) of FUNCTION main()
94 % (16/ 17) statement

***TER 60 % (9/ 15) of SOURCE FILE test.c
94 % (16/ 17) statement

MONITORED SOURCE FILE : c:\Programme\Qualification\ExampleQualification\Validation\TestRun
INSTRUMENTATION MODE : multicondition

HITS/TRUE FALSE LINE DESCRIPTION

Anzahl der verbleibenden ungelösten Konflikte: 7 (0 davon Leerraum-Konflikte)

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>
- ▶ Documentation
- ▶ Download of Releases & Nightly Builds
- ▶ Issue Reporting & Tracking (of the Q-Kit, not CTC)

Available Reports

Report	Title
{1}	Active Tickets
{2}	Active Tickets by Version
{3}	Active Tickets by Milestone
{4}	Accepted, Active Tickets by Owner
{5}	Accepted, Active Tickets by Owner (Full Description)
{6}	All Tickets By Milestone (Including closed)
{7}	My Tickets
{8}	Active Tickets, Mine first

- ▶ Q-Kit users get access to it

Verifysoft - Mozilla Firefox

https://opentrac.teststatt.de/tracs/verifysoft/wiki

Trac

logged in as slotosch@validas.de | Logout | Preferences | Help/Guide | About Trac

Wiki | Timeline | Roadmap | Browse Source | View Tickets | New Ticket | Search

wiki: WikiStart | Start Page | Index | History | Last modified 8 weeks ago

Welcome to the Trac of the Verifysoft Testwell CTC++ Qualification Kit

The CTC++ Qualification Kit is available here. This trac system is used as development tool for exchanging data and tracking tasks and findings. Please help us to improve the quality by submitting findings and proposals for improvement.

Starting Points

- Documentation (see [DocumentationPage](#))
- QKit (see [KitDownload](#))
- Development (see [DevelopmentProcess](#))
- Tickets (see <https://opentrac.teststatt.de/tracs/verifysoft/report>)

Edit this page | Attach file | Rename page | Delete this version | Delete page

Download in other formats:
Plain Text

trac POWERED | Powered by Trac 0.12.5 By Edgewall Software. | Visit the Trac open source project at <http://trac.edgewall.org/>

