Qualification Kit for Testwell CTC++

Tools which are used for the development of safety critical software need to be classified (Tool Classification) in order to determine whether a qualification of the tool are necessary. This classification considers the impact of the software tool on the project.
Tools which can have an impact on the project, and in which a malfunction is not immediately apparent, the safety standards require proof of trustworthiness. This proof is done by a qualification of the tool (Tool Qualification).
A tool must always be qualified within the concrete development environment of the tool user.
To simplify the qualification of Testwell CTC ++ in safety-critical projects, Verifysoft Technology offers a tool Qualification Kit for Testwell CTC++. The Qualification Kit is suitable for the standards EN 50128, ISO 26262 (automotive), DO-178C (aeronautics) and IEC 61508 (railway).

Figure 1: The Tool Qualification Kit for Testwell CTC++ covers several standards

Please note that software verification tools like Testwell CTC++ need to be qualified and not certified. Only the safety critical software which is used eg. in the aircraft or the car has to be certified.

Tool Classification

As mentionned above, the tools used in the development of safety-critical software must first be classified.
Test coverage tools like Testwell CTC++ are verification tools. Although those tools are unable to insert errors in your code, they can fail to detect errors which are already introduced in your code (for example by premature termination of the testing due to the display of a to high test coverage).
Result of the classification is to decide whether and to what extent a qualification of the tools for the specific project is required.
The classification process of software tools is slightly different for the standards DO-178C, IEC 61508/EN 50128 and ISO 26262 (c.f. figure 2).

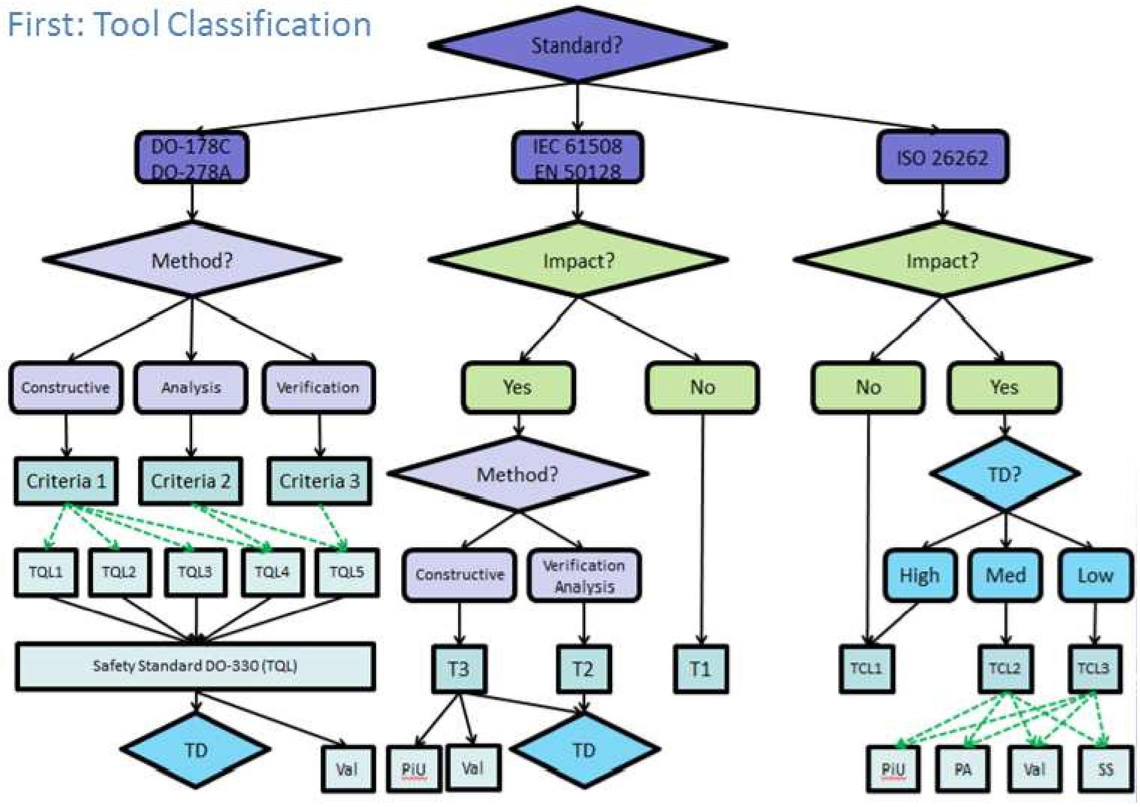


Figure 2: Tool Classification for different standards (Source: Validas AG, Munich)

The following description is about the automotive standard ISO 26262 (c.f. right part of figure 2). First is checked whether the tool has an influence (impact) on the software.

If the tool has no impact on the software, the tool is classified TCL 1 (Tool Confidence Level 1) and the qualification is not needed.

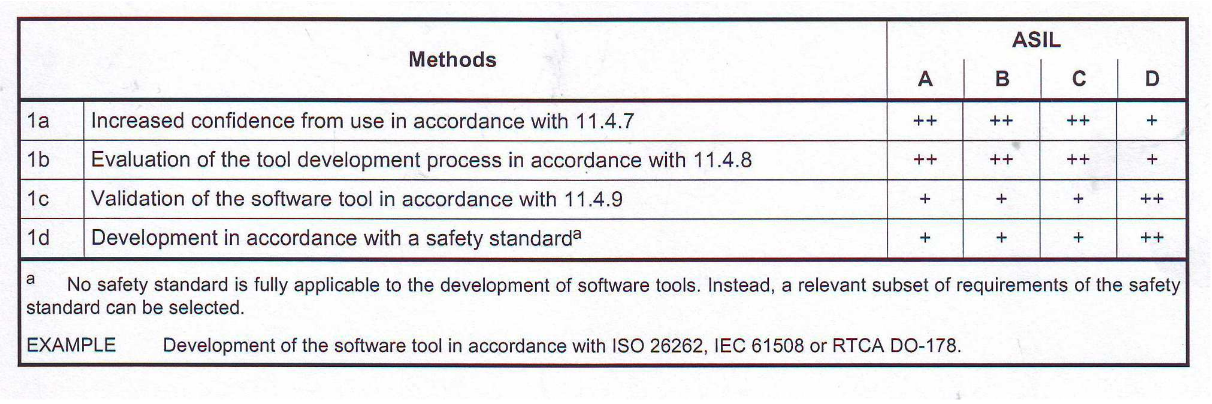
If the tool can have an impact on the software, the probability to detect such an impact need to be evaluated (TD). If the probability to detect a misfunction is high, the tool is classified TCL1 (no qualification needed). For example if you use two different coverage tools and if you compare the results, the probability to detect misfunctions is high. By this usage of redundant tools the tool qualification can be avoided .

If the probability of the detection of misfunctions is medium, the tool confidence level is 2 (TCL 2). For a low probability to detect misfunctions the tool confidence level is 3.

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Table 1: Determination of the tool confidence level (TCL)

While for the tool confidence level 1 no tool qualification is required, the software tool must be qualified for tool confidence levels 2 according to the following table:



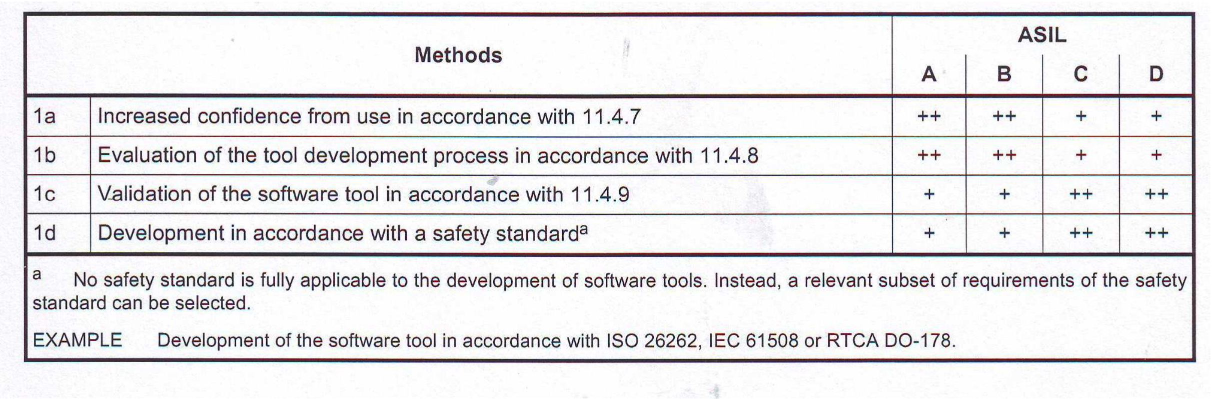
Methods		ASIL			
		A	B	C	D
1a	Increased confidence from use in accordance with 11.4.7	++	++	++	+
1b	Evaluation of the tool development process in accordance with 11.4.8	++	++	++	+
1c	Validation of the software tool in accordance with 11.4.9	+	+	+	++
1d	Development in accordance with a safety standard[a]	+	+	+	++

[a] No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.

EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178.

Table 2: ISO 26262: Qualification of software tools classified TCL2

For tool confidence level 3 the ISO 26262 requires the following:



Methods		ASIL			
		A	B	C	D
1a	Increased confidence from use in accordance with 11.4.7	++	++	+	+
1b	Evaluation of the tool development process in accordance with 11.4.8	++	++	+	+
1c	Validation of the software tool in accordance with 11.4.9	+	+	++	++
1d	Development in accordance with a safety standard[a]	+	+	++	++

[a] No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.

EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178.

Table 3: ISO 26262: Qualification of software tools classified TCL3

Tool Qualification by Validation

The qualification method is tool validation by testing the safety relevant parts oft he tool. The standard ISO 26262 describes the validation of a software tool in chapter 11.4.9:
“The validation measures shall demonstrate that the software tool complies with its specified requirements (use cases of the tool). The malfunctions and their corresponding erroneous outputs of the software tool occurring during validation shall be analysed together with information on their possible consequences and with measures to avoid or detect them. In addition the reaction of the software tool to anomalous operating conditions shall be examined.”

Tool Qualification Kit for Testwell CTC++

The Tool Qualification Kit for Testwell CTC++, which have been developped by Verifysoft Technology in cooperation with Validas AG (Munich/Germany), reduces dramatically the effort fort he tool qualification. The kit supports the user in the qualification of safety critical software development according to the standards ISO 26262, DO-178C, IEC 61508, and EN 50128.
The Qualification Kit for Testwell CTC++ validates that statement, decision, and MC/DC coverage is properly analysed and shown for he programming language C by Testwell CTC++ in your development and project environment.
The kit consists of a Qualification Support Tool (QST), a test suite with many test cases for the different use cases of Testwell CTC++, a Test Automation Unit (TAU) to execute the test cases, a Validation & Verification Report (V&V Report) in order to prove the quality oft he qualification kit, and an user manual.

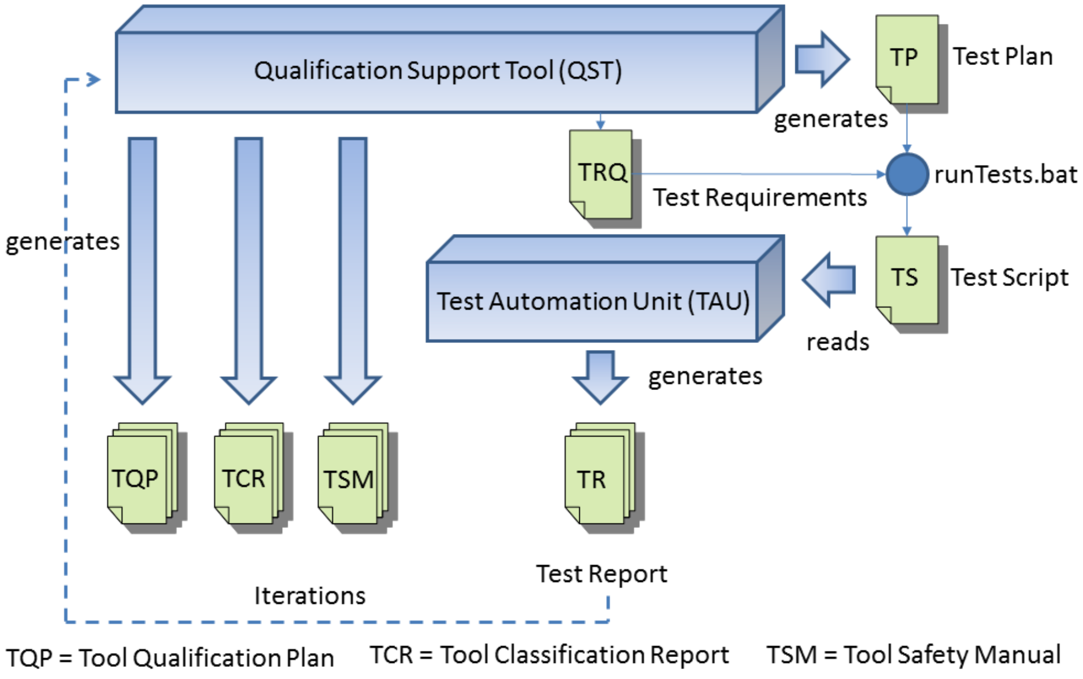
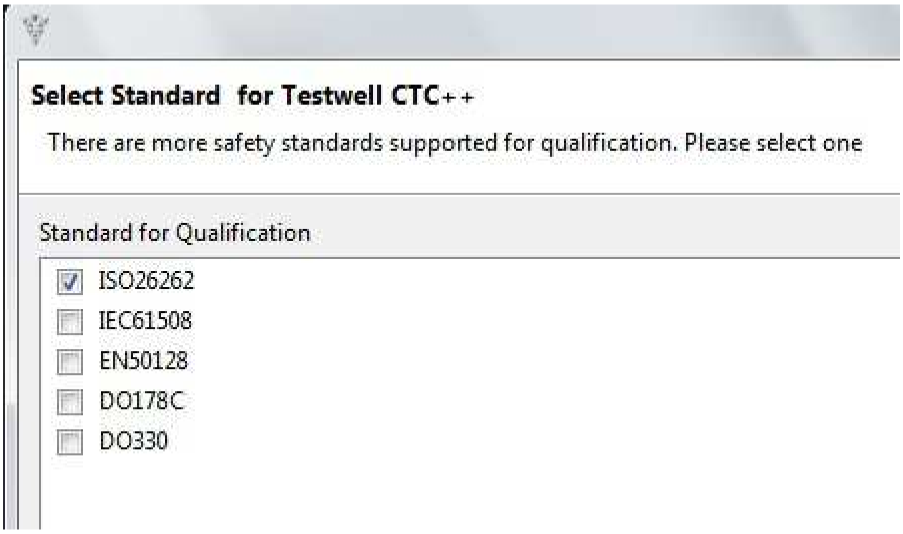


Figure 3: Tool Qualification Kit for Testwell CTC++

Application of the Tool Qualification Kit

The Qualification Support Tool guides the user through the qualification process. First the relevant standard is choosen (c.f. figure 4).



Select Standard for Testwell CTC++

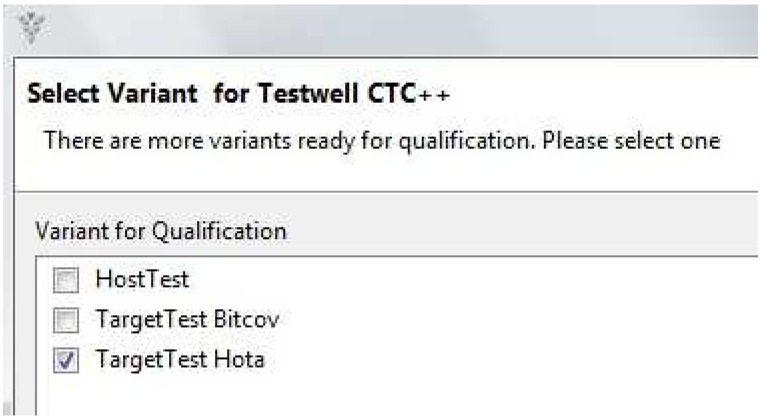
There are more safety standards supported for qualification. Please select one

Standard for Qualification

- ☑ ISO26262
- □ IEC61508
- □ EN50128
- □ DO178C
- □ DO330

Figure 4: Choice of standard within the Qualification Kit

In the next step the used variant of Testwell CTC++ is choosen (host only, CTC++ with host-target add-on for mesuring on target code coverage, or CTC++ with bitcov-add-on for measuring code coverage on very small targets).



Select Variant for Testwell CTC++

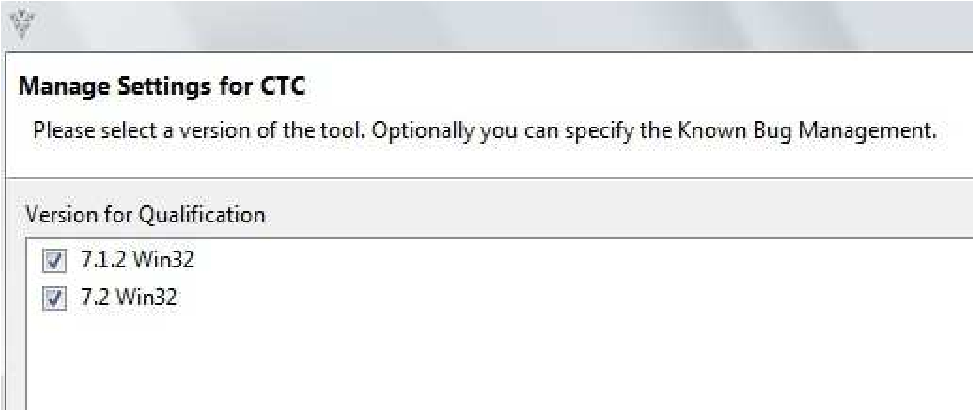
There are more variants ready for qualification. Please select one

Variant for Qualification

- □ HostTest
- □ TargetTest Bitcov
- ☑ TargetTest Hota

Figure 5:Choice of Testwell CTC++ variant

In a further step the version of Testwell CTC++ is choosen (c.f. figure 6).



Manage Settings for CTC

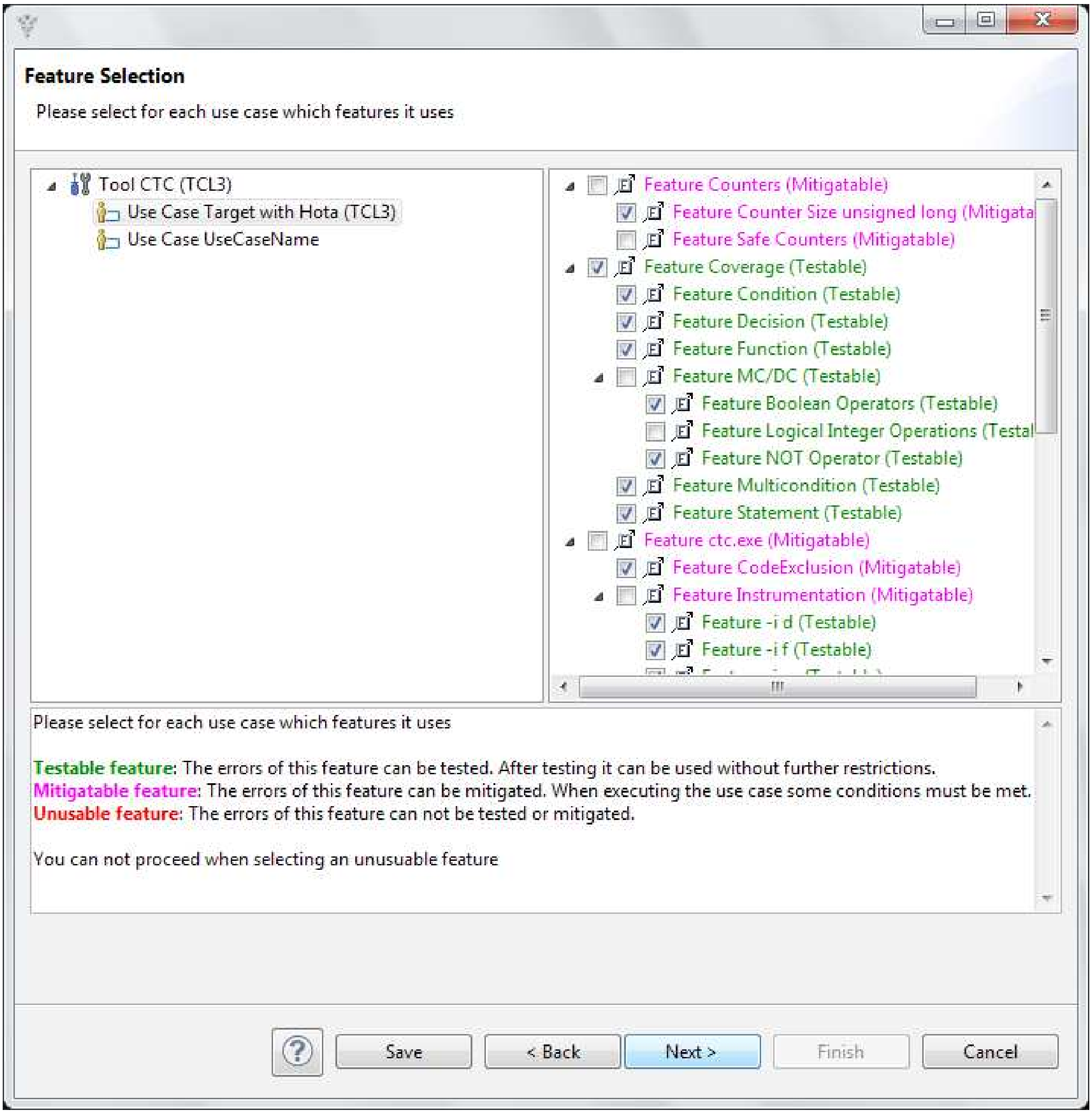
Please select a version of the tool. Optionally you can specify the Known Bug Management.

Version for Qualification

☑ 7.1.2 Win32

☑ 7.2 Win32

Figure 6: Choice of Testwell CTC++ version



Feature Selection

Please select for each use case which features it uses

- Tool CTC (TCL3)
  - Use Case Target with Hota (TCL3)
  - Use Case UseCaseName

- ☐ Feature Counters (Mitigatable)
  - ☑ Feature Counter Size unsigned long (Mitigata
  - ☐ Feature Safe Counters (Mitigatable)
- ☑ Feature Coverage (Testable)
  - ☑ Feature Condition (Testable)
  - ☑ Feature Decision (Testable)
  - ☑ Feature Function (Testable)
  - ☐ Feature MC/DC (Testable)
    - ☑ Feature Boolean Operators (Testable)
    - ☐ Feature Logical Integer Operations (Testal
    - ☑ Feature NOT Operator (Testable)
  - ☑ Feature Multicondition (Testable)
  - ☑ Feature Statement (Testable)
- ☐ Feature ctc.exe (Mitigatable)
  - ☑ Feature CodeExclusion (Mitigatable)
  - ☐ Feature Instrumentation (Mitigatable)
    - ☑ Feature -i d (Testable)
    - ☑ Feature -i f (Testable)

Please select for each use case which features it uses

Testable feature: The errors of this feature can be tested. After testing it can be used without further restrictions.
Mitigatable feature: The errors of this feature can be mitigated. When executing the use case some conditions must be met.
Unusable feature: The errors of this feature can not be tested or mitigated.

You can not proceed when selecting an unusable feature

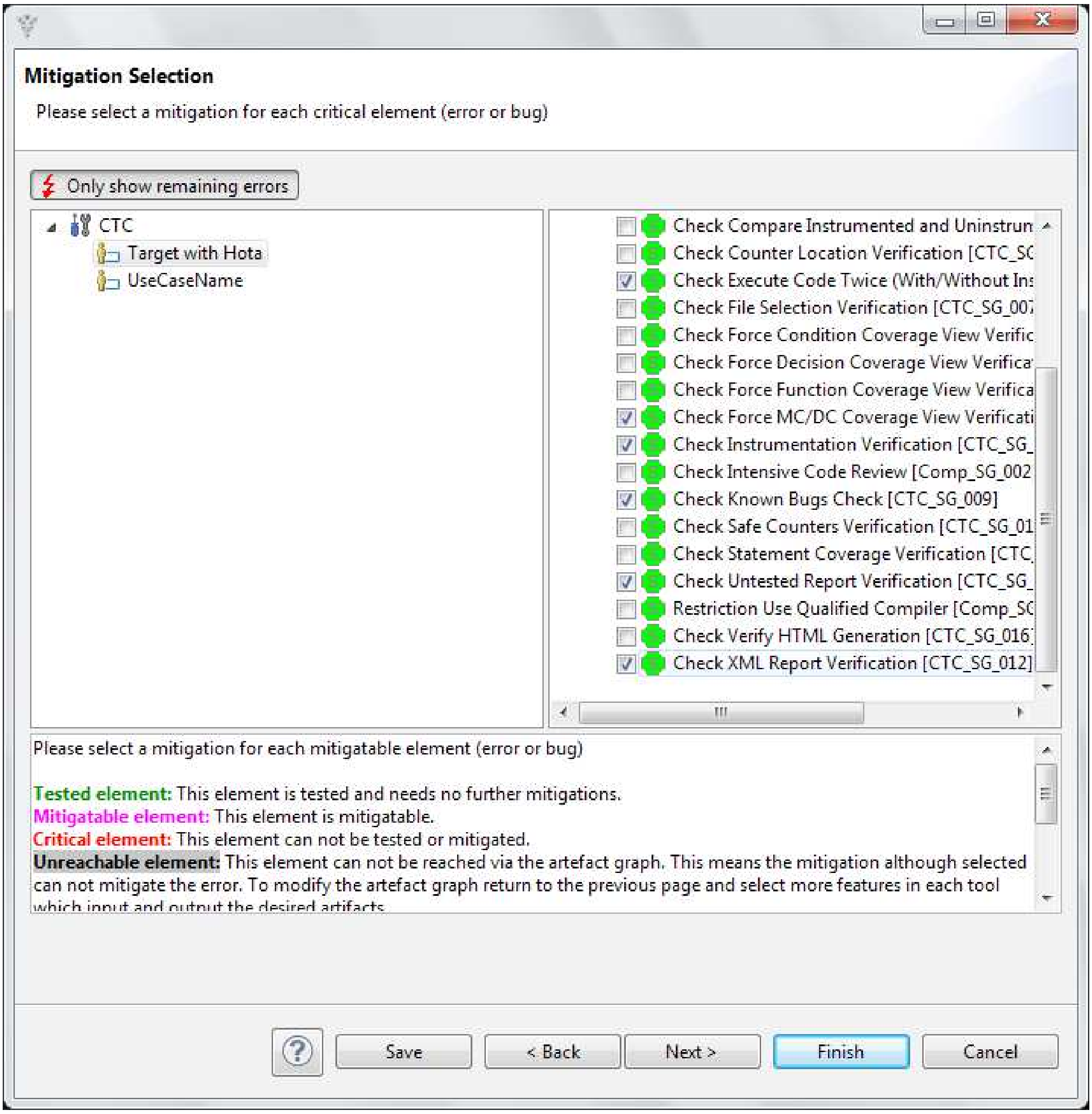
Figure 7: Choice of used features

The user then selects the used features of Testwell CTC++ (c.f. figure 7). Only the features which are used in your project need to be qualified. According to the standard and the ASIL level the used features and mitigations are already preselected.

The usage of features, which have not been validated during the tool qualification is prohibited. The Qualification Support Tool generates a notice within the Tool Safety Manual.

Features which are testable have green color. Pink colored features can be used with some constraints (mitigations).

Mitigations and test cases are preselected in the next step (c.f. figure 8). If needed other cases than the preselected ones can be choosen.



Mitigation Selection

Please select a mitigation for each critical element (error or bug)

Only show remaining errors

- CTC
  - Target with Hota
  - UseCaseName

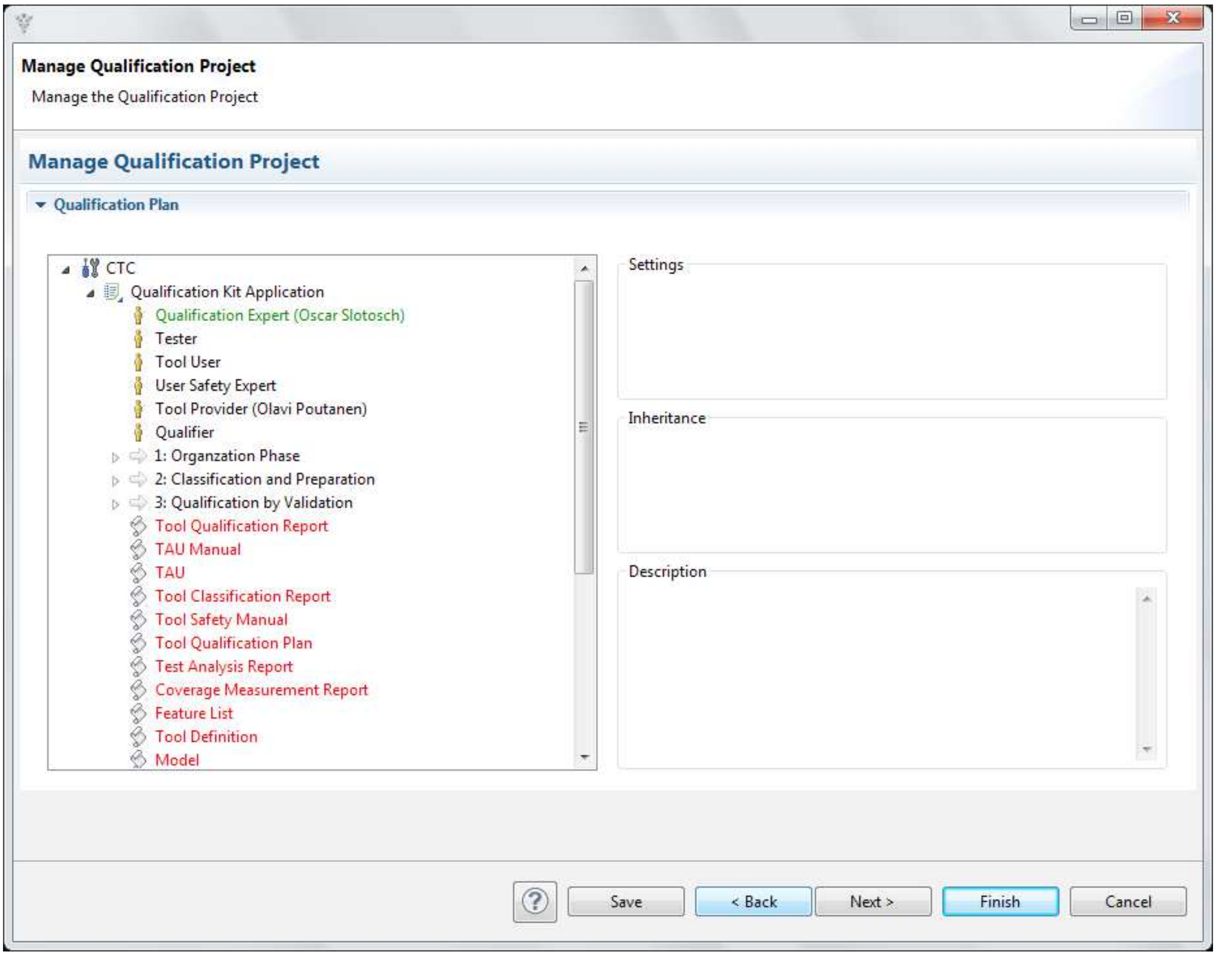
- ☐ Check Compare Instrumented and Uninstrun
- ☐ Check Counter Location Verification [CTC_SG
- ☑ Check Execute Code Twice (With/Without Ins
- ☐ Check File Selection Verification [CTC_SG_007
- ☐ Check Force Condition Coverage View Verific
- ☐ Check Force Decision Coverage View Verifica
- ☐ Check Force Function Coverage View Verifica
- ☑ Check Force MC/DC Coverage View Verificati
- ☑ Check Instrumentation Verification [CTC_SG_
- ☐ Check Intensive Code Review [Comp_SG_002
- ☑ Check Known Bugs Check [CTC_SG_009]
- ☐ Check Safe Counters Verification [CTC_SG_01
- ☐ Check Statement Coverage Verification [CTC
- ☑ Check Untested Report Verification [CTC_SG_
- ☐ Restriction Use Qualified Compiler [Comp_SG
- ☐ Check Verify HTML Generation [CTC_SG_016]
- ☑ Check XML Report Verification [CTC_SG_012]

Please select a mitigation for each mitigatable element (error or bug)

Tested element: This element is tested and needs no further mitigations.
Mitigatable element: This element is mitigatable.
Critical element: This element can not be tested or mitigated.
Unreachable element: This element can not be reached via the artefact graph. This means the mitigation although selected can not mitigate the error. To modify the artefact graph return to the previous page and select more features in each tool which input and output the desired artifacts

Figure 8: Choice of mitigations

The next window serves fort he organisation of the qualificaton process (c.f. figure 9). You can determine roles (who does what and when?).



Manage Qualification Project

Manage the Qualification Project

Manage Qualification Project

Qualification Plan

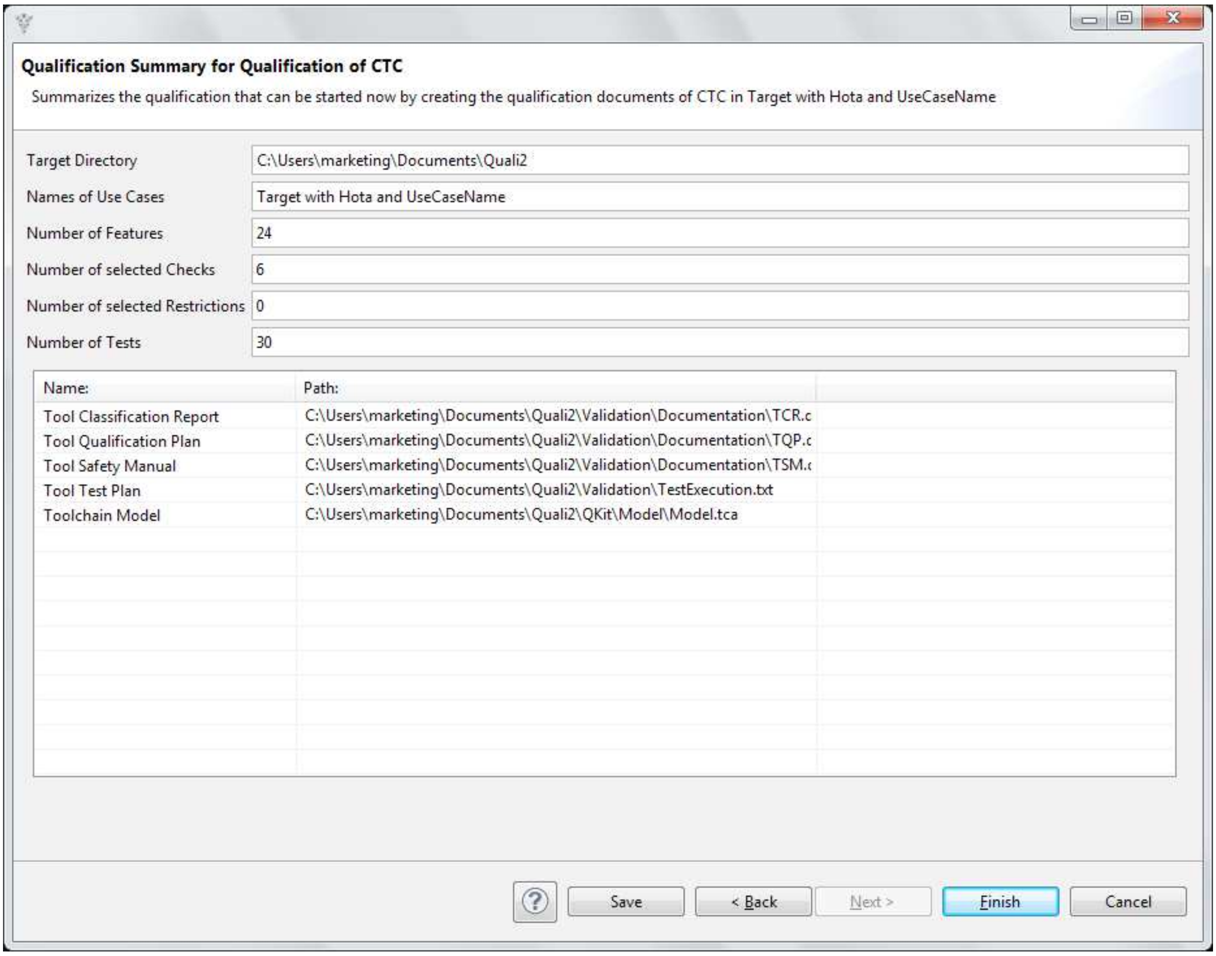
- CTC
  - Qualification Kit Application
    - Qualification Expert (Oscar Slotosch)
    - Tester
    - Tool User
    - User Safety Expert
    - Tool Provider (Olavi Poutanen)
    - Qualifier
    - 1: Organzation Phase
    - 2: Classification and Preparation
    - 3: Qualification by Validation
    - Tool Qualification Report
    - TAU Manual
    - TAU
    - Tool Classification Report
    - Tool Safety Manual
    - Tool Qualification Plan
    - Test Analysis Report
    - Coverage Measurement Report
    - Feature List
    - Tool Definition
    - Model

Figure 9: Planning of the qualification

The next view shows a summary with the number of features, selected checks, number of tests, and the path to the generated documents (c.f. figure 10).

After pressing the „Finish“ button, various documents like the Tool Classification Report (TCR.docx), the Tool Qualification Plan (TQP.docx), and the Tool Safety Manual (TSM.docs) are generated. Also files like Test Run Files, Test Execution.txt and tool-config-files are generated. Some of the generated documents and files are shown in figure 11.

By executing the testrun.bat, a test script is generated. This test script is read and executed by the Test Automation Unit (c.f. figure 12). The generated documents are updated. You have the proof that Testwell CTC++ works properly within your project if all tests have been passed successfully.



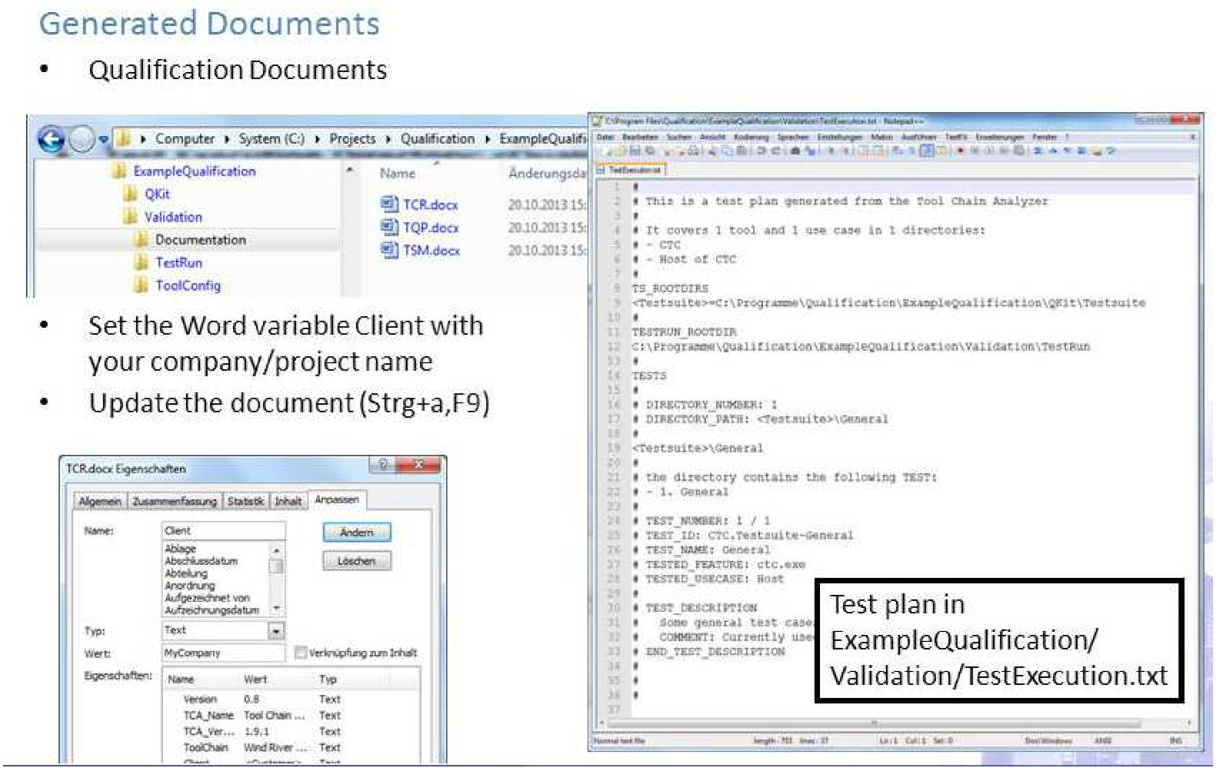
Qualification Summary for Qualification of CTC

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

Target Directory	C:\Users\marketing\Documents\Quali2
Names of Use Cases	Target with Hota and UseCaseName
Number of Features	24
Number of selected Checks	6
Number of selected Restrictions	0
Number of Tests	30

Name:	Path:
Tool Classification Report	C:\Users\marketing\Documents\Quali2\Validation\Documentation\TCR.c
Tool Qualification Plan	C:\Users\marketing\Documents\Quali2\Validation\Documentation\TQP.c
Tool Safety Manual	C:\Users\marketing\Documents\Quali2\Validation\Documentation\TSM.c
Tool Test Plan	C:\Users\marketing\Documents\Quali2\Validation\TestExecution.txt
Toolchain Model	C:\Users\marketing\Documents\Quali2\QKit\Model\Model.tca

Figure 10: Summary of the qualification



Generated Documents

- Qualification Documents

Computer ▸ System (C:) ▸ Projects ▸ Qualification ▸ ExampleQualifi
ExampleQualification
QKit
Validation
Documentation
TestRun
ToolConfig

Name	Änderungsda
TCR.docx	20.10.2013 15:
TQP.docx	20.10.2013 15:
TSM.docx	20.10.2013 15:

- Set the Word variable Client with your company/project name
- Update the document (Strg+a,F9)

TCR.docx Eigenschaften

Name: Client

Typ: Text

Wert: MyCompany

□ Verknüpfung zum Inhalt

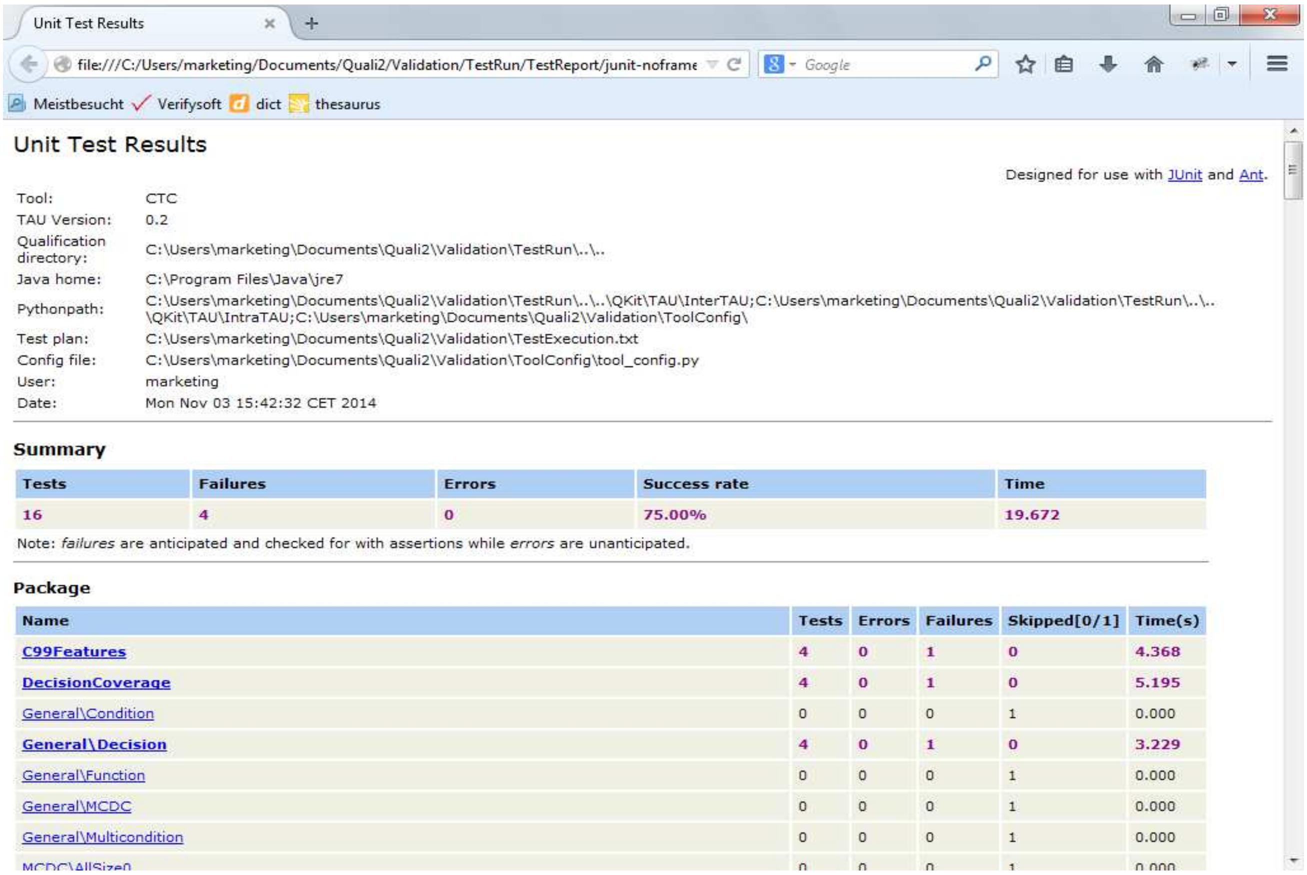
Eigenschaften:

Name	Wert	Typ
Version	0.8	Text
TCA_Name	Tool Chain ...	Text
TCA_Ver...	1.9.1	Text
ToolChain	Wind River ...	Text

```
#
# This is a test plan generated from the Tool Chain Analyzer
#
# It covers 1 tool and 1 use case in 1 directories:
# - CTC
# - Host of CTC
#
TS_ROOTDIRS
<Testsuite>=C:\Programme\Qualification\ExampleQualification\QKit\Testsuite
#
TESTRUN_ROOTDIR
C:\Programme\Qualification\ExampleQualification\Validation\TestRun
#
TESTS
#
# DIRECTORY_NUMBER: 1
# DIRECTORY_PATH: <Testsuite>\General
#
<Testsuite>\General
#
# the directory contains the following TEST:
# - 1. General
#
# TEST_NUMBER: 1 / 1
# TEST_ID: CTC.Testsuite-General
# TEST_NAME: General
# TESTED_FEATURE: ctc.exe
# TESTED_USECASE: Host
#
# TEST_DESCRIPTION
#   Some general test case
#   COMMENT: Currently use
# END_TEST_DESCRIPTION
#
#
#
```

Test plan in ExampleQualification/ Validation/TestExecution.txt

Figure 11: Documents generated by the Qualification Support Tool



Unit Test Results
file:///C:/Users/marketing/Documents/Quali2/Validation/TestRun/TestReport/junit-noframe
Google
Meistbesucht
Verifysoft
dict
thesaurus

Unit Test Results

Designed for use with JUnit and Ant.

Tool:	CTC
TAU Version:	0.2
Qualification directory:	C:\Users\marketing\Documents\Quali2\Validation\TestRun\..\..
Java home:	C:\Program Files\Java\jre7
Pythonpath:	C:\Users\marketing\Documents\Quali2\Validation\TestRun\..\..\QKit\TAU\InterTAU;C:\Users\marketing\Documents\Quali2\Validation\TestRun\..\..\QKit\TAU\IntraTAU;C:\Users\marketing\Documents\Quali2\Validation\ToolConfig\
Test plan:	C:\Users\marketing\Documents\Quali2\Validation\TestExecution.txt
Config file:	C:\Users\marketing\Documents\Quali2\Validation\ToolConfig\tool_config.py
User:	marketing
Date:	Mon Nov 03 15:42:32 CET 2014

Summary

Tests	Failures	Errors	Success rate	Time
16	4	0	75.00%	19.672

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

Package

Name	Tests	Errors	Failures	Skipped[0/1]	Time(s)
C99Features	**4**	**0**	**1**	**0**	**4.368**
DecisionCoverage	**4**	**0**	**1**	**0**	**5.195**
General\Condition	0	0	0	1	0.000
General\Decision	**4**	**0**	**1**	**0**	**3.229**
General\Function	0	0	0	1	0.000
General\MCDC	0	0	0	1	0.000
General\Multicondition	0	0	0	1	0.000
MCDC\AllSize0	0	0	0	1	0.000

Figure 12: Results of the test run

For further information please contact:
Verifysoft Technology GmbH
In der Spöck 10-12
77656 Offenburg
Deutschland

www.verifysoft.com
Phone +49 781 127 8118-0

Last modified: 17 November 2014