Verifysoft TECHNOLOGY

# 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

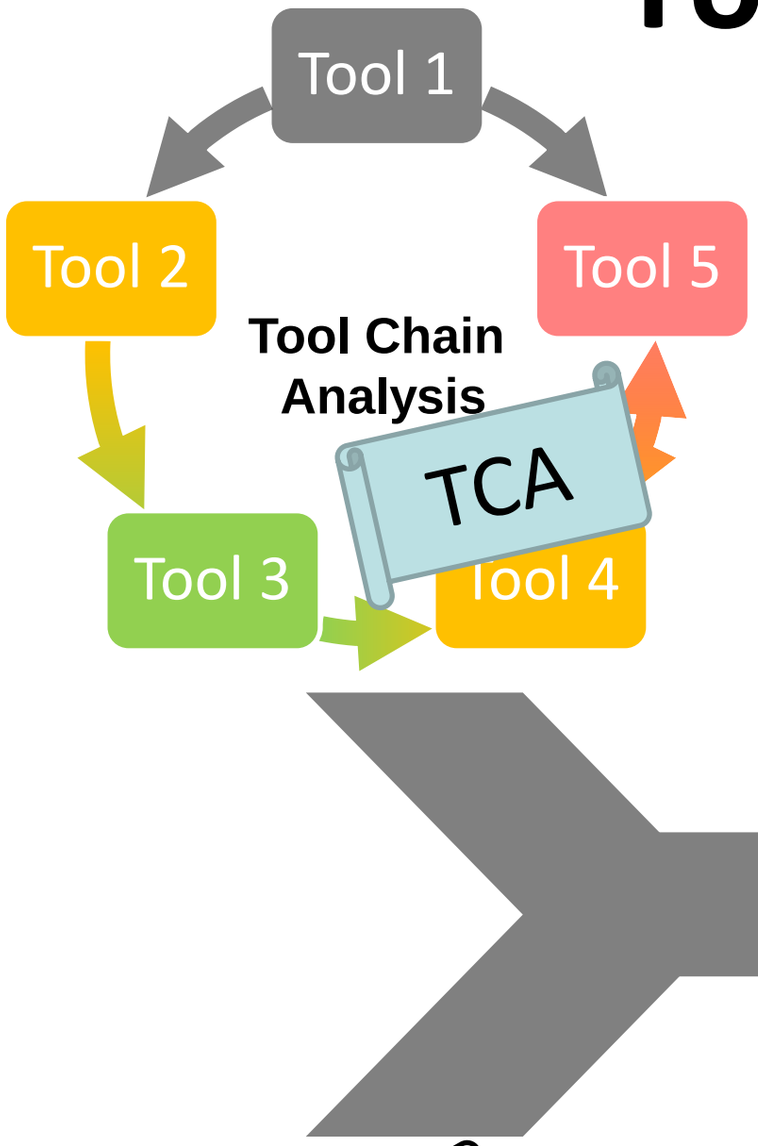
# Testwell CTC Qualification Kit

- **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**

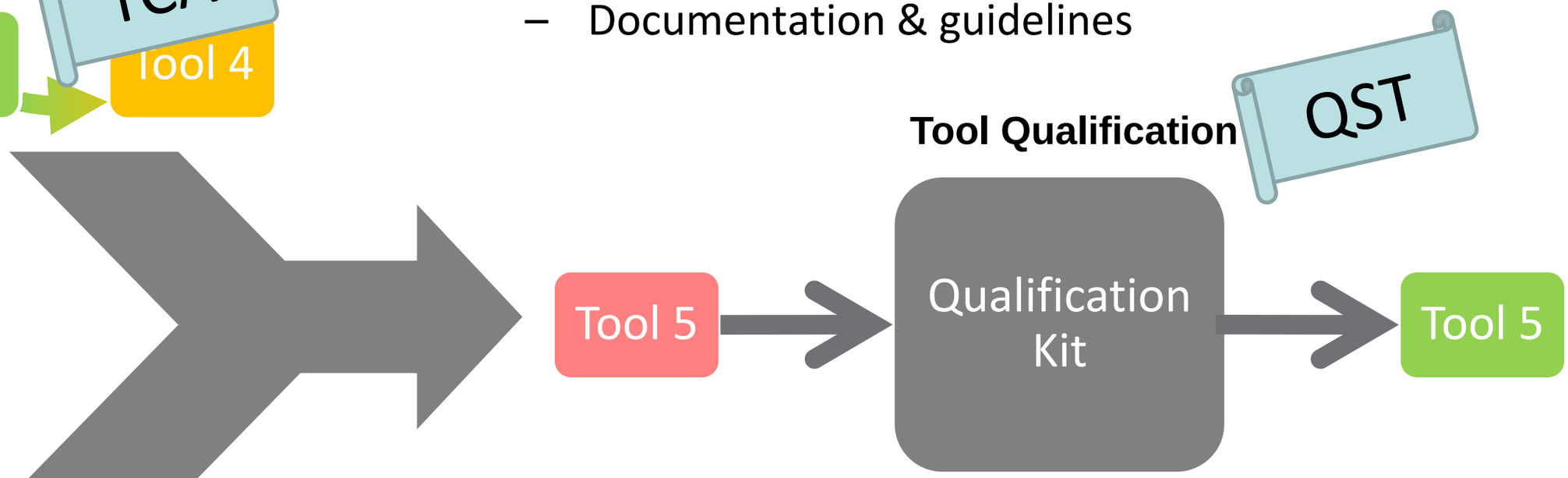
# Content

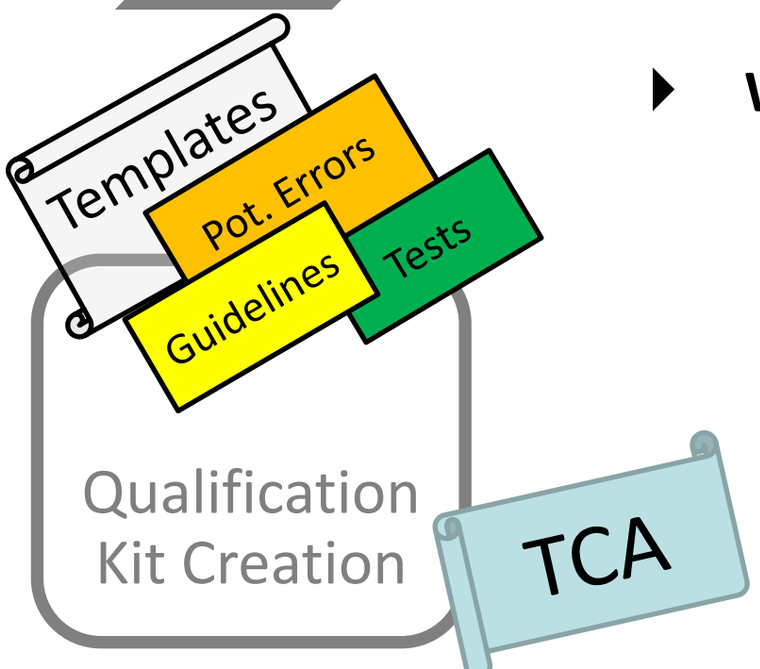
- Overview
- **Qualification Method**
- Example: First Qualification Steps
  - Installation
  - Qualification Planning
  - Text Execution & Analysis
- Availability and feedback

# 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

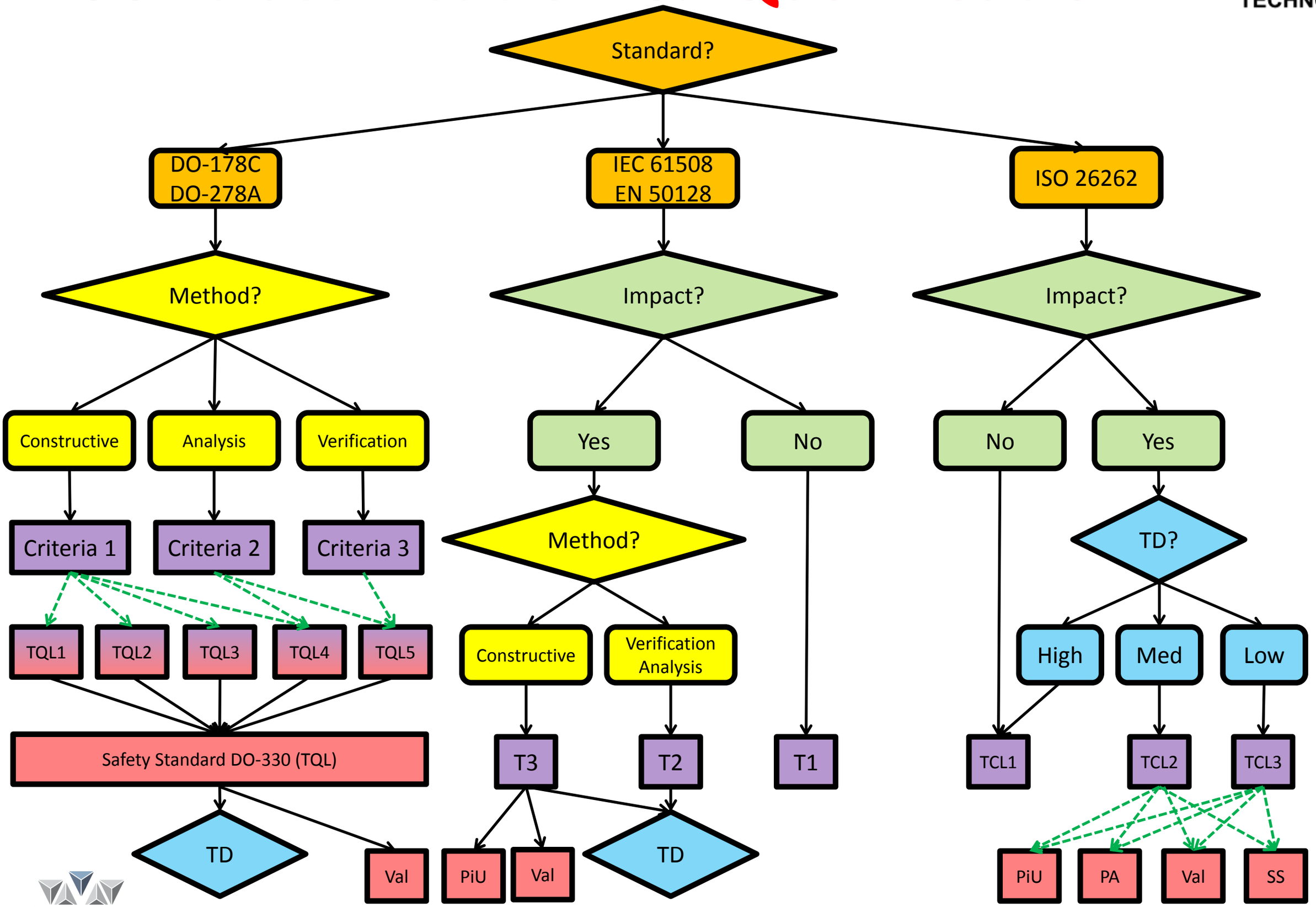



- **We consider the following processes**
  - **Tool Chain Analysis**: Classification of all tools (TCA)
  - **Tool Qualification**: Qualification of one tool, e.g. by application of a qualification kit with a qualification support tool (QST)
  - **Qualification Kit Creation (TCA)**
    - **Create support for classification**
    - **Create support for qualification**

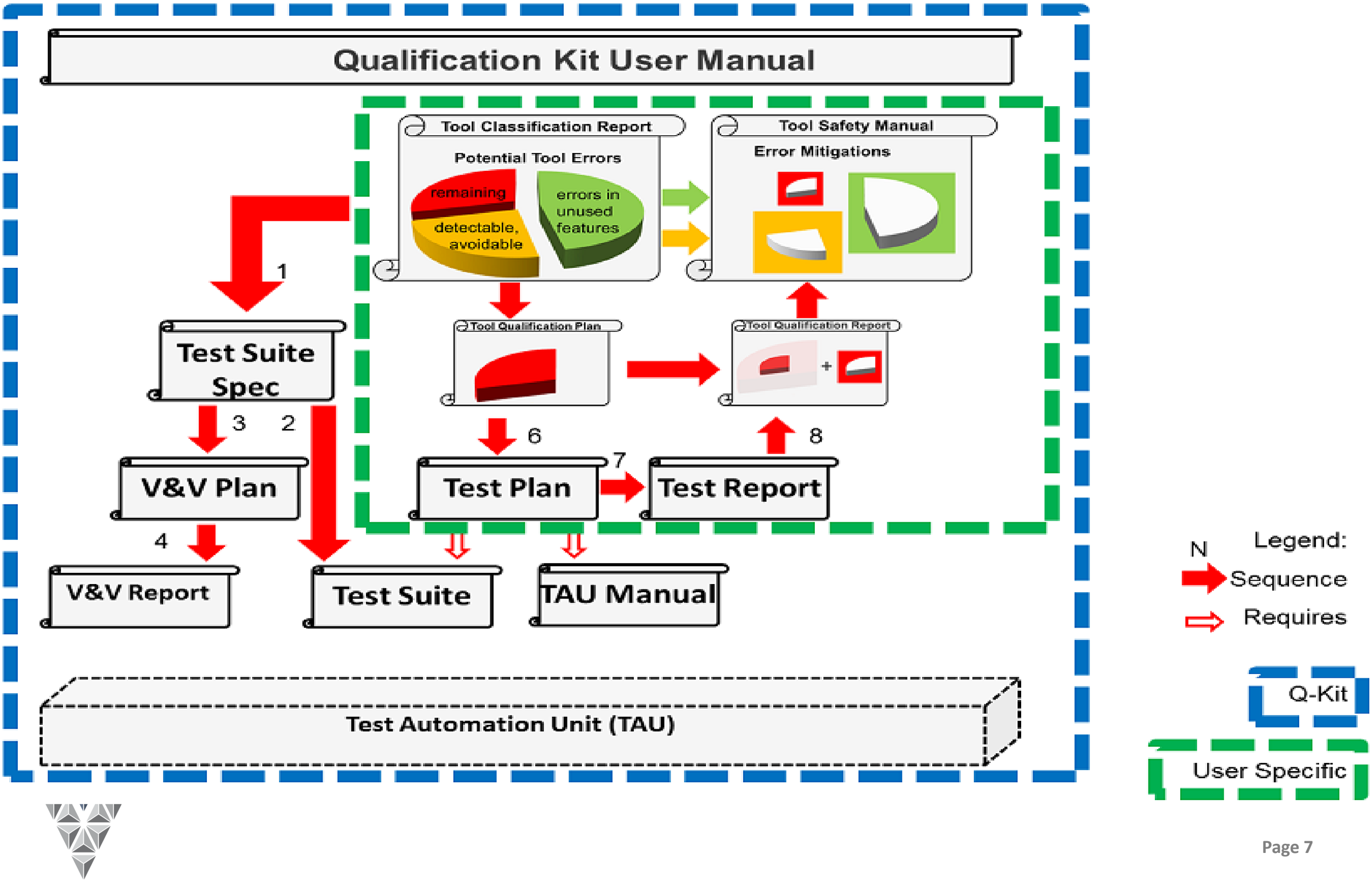
# Tool Classification & Qualification

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.

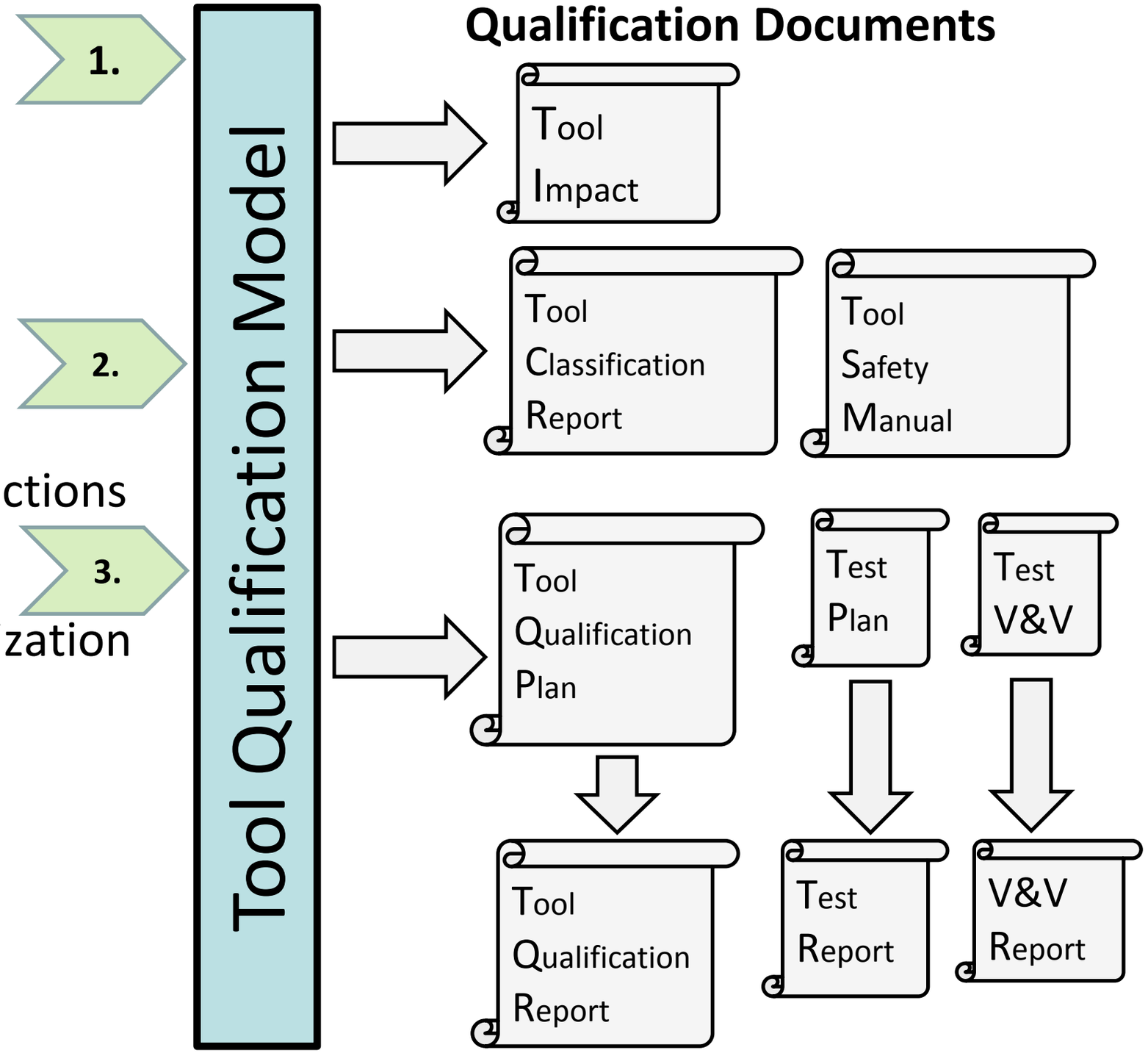
# Qualification by Validation



# Model-Based Tool Qualification

verifysoft TECHNOLOGY

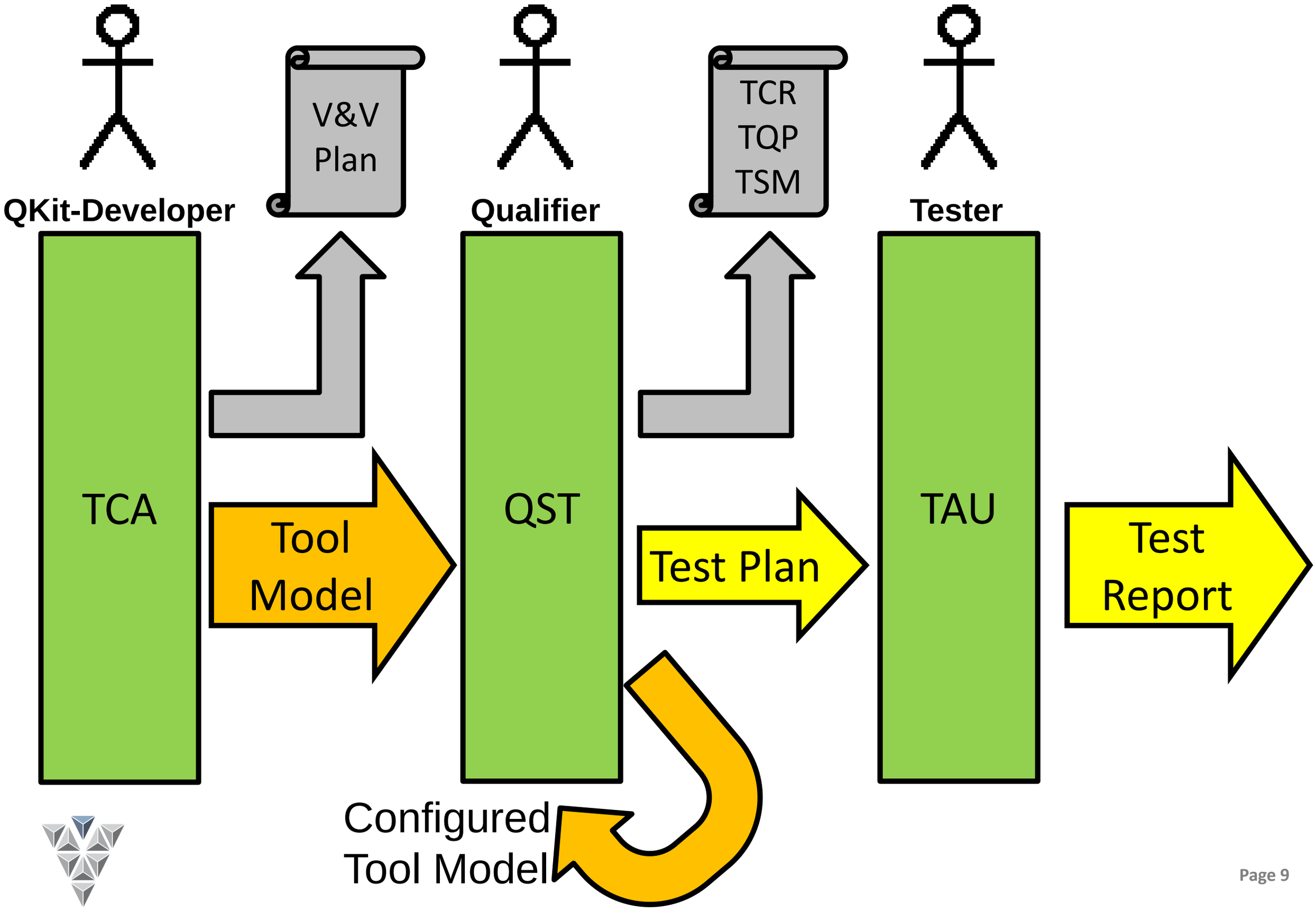
1. **Structure Modeling**
   - Use-Cases,
   - Relevant tool features,
   - Artifacts
2. **Analysis Modeling**
   - Potential errors
   - Available checks & restrictions
3. **Qualification Modeling**
   - Qualification cost optimization
   - Qualification planning
   - Qualification tests
     - Identification
     - Assignment to
       - Features
       - Errors




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

# Tools & Generated Artifacts

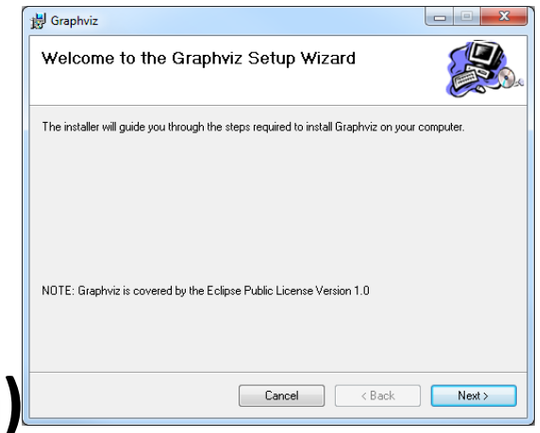
Verifysoft TECHNOLOGY



# Content

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

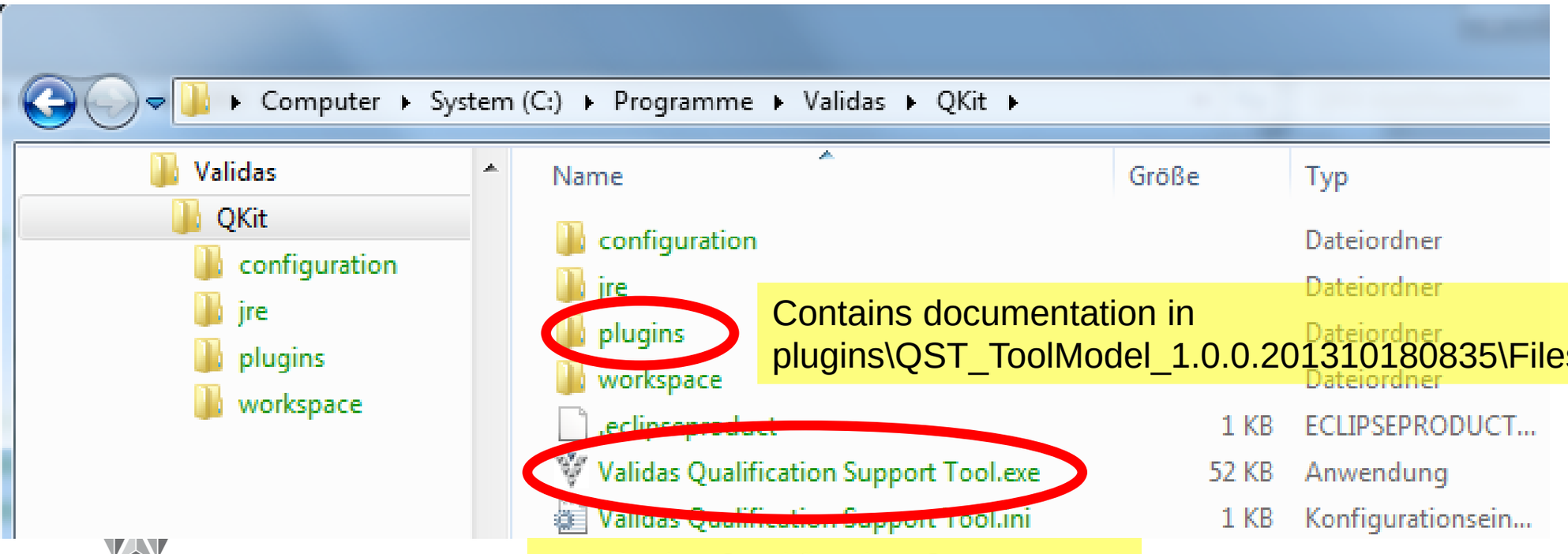
# Installation





- 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

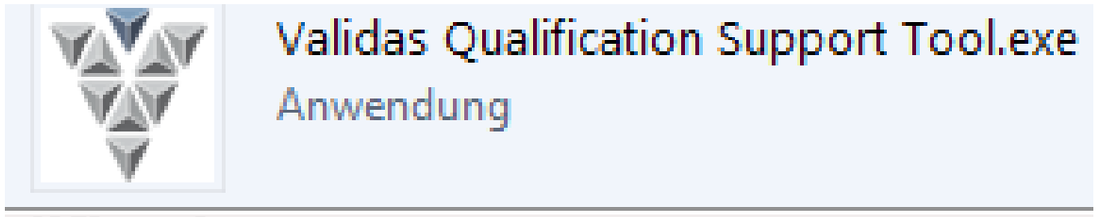
This is the Qualification Support Tool

# Content

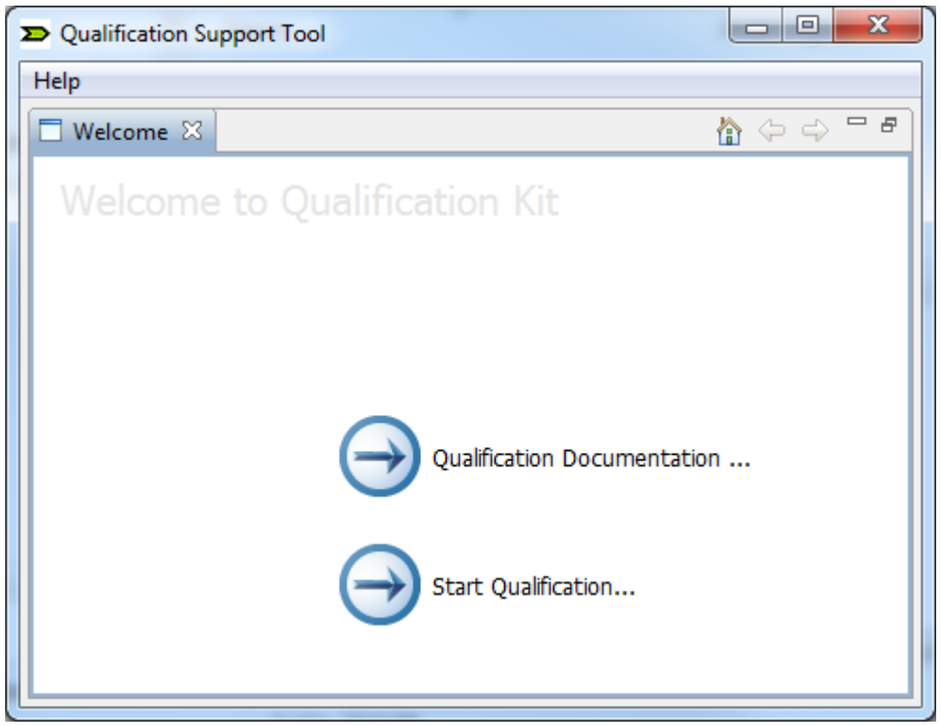
- Overview
- Qualification Method
- **Example: First Qualification Steps**
  - Installation
  - **Qualification Planning**
  - Text Execution & Analysis
- Availability and feedback

# Start QST

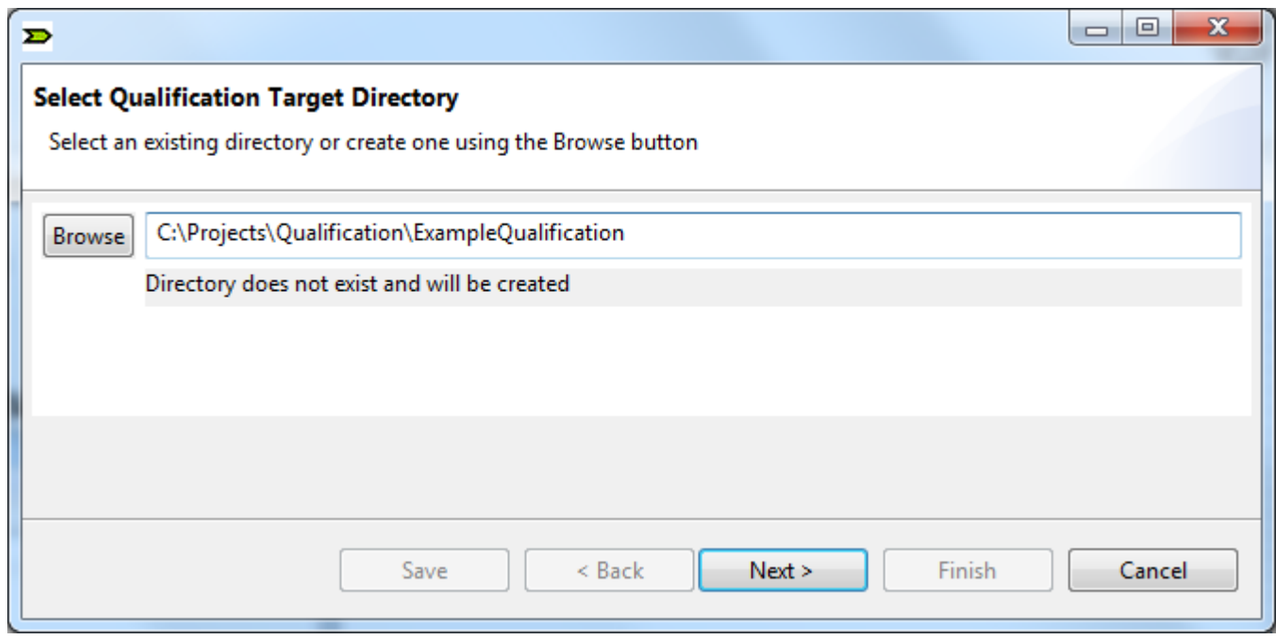
- Details can be found in documentation (QKit User Manual)
- Double click on



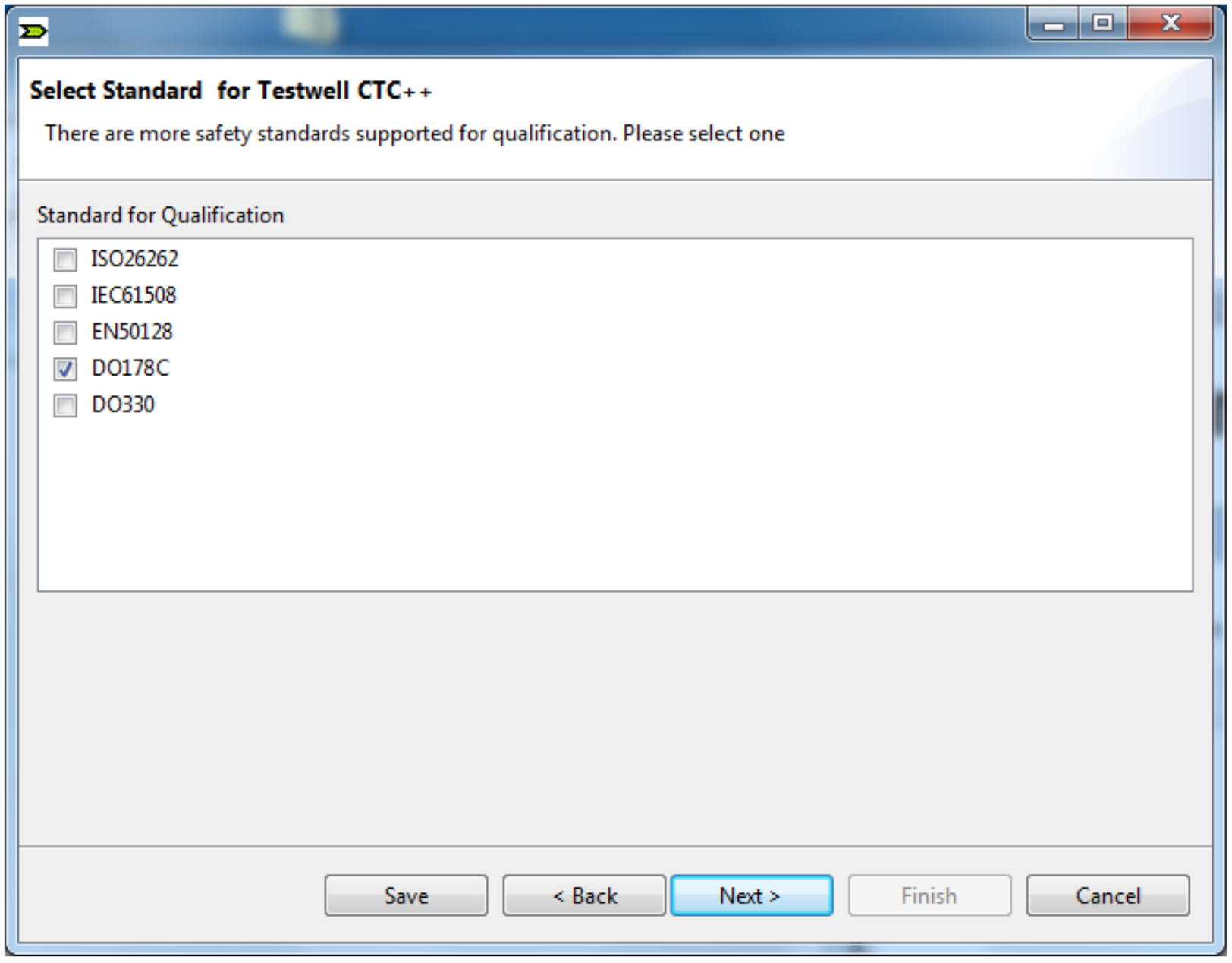

- Click on Start Qualification



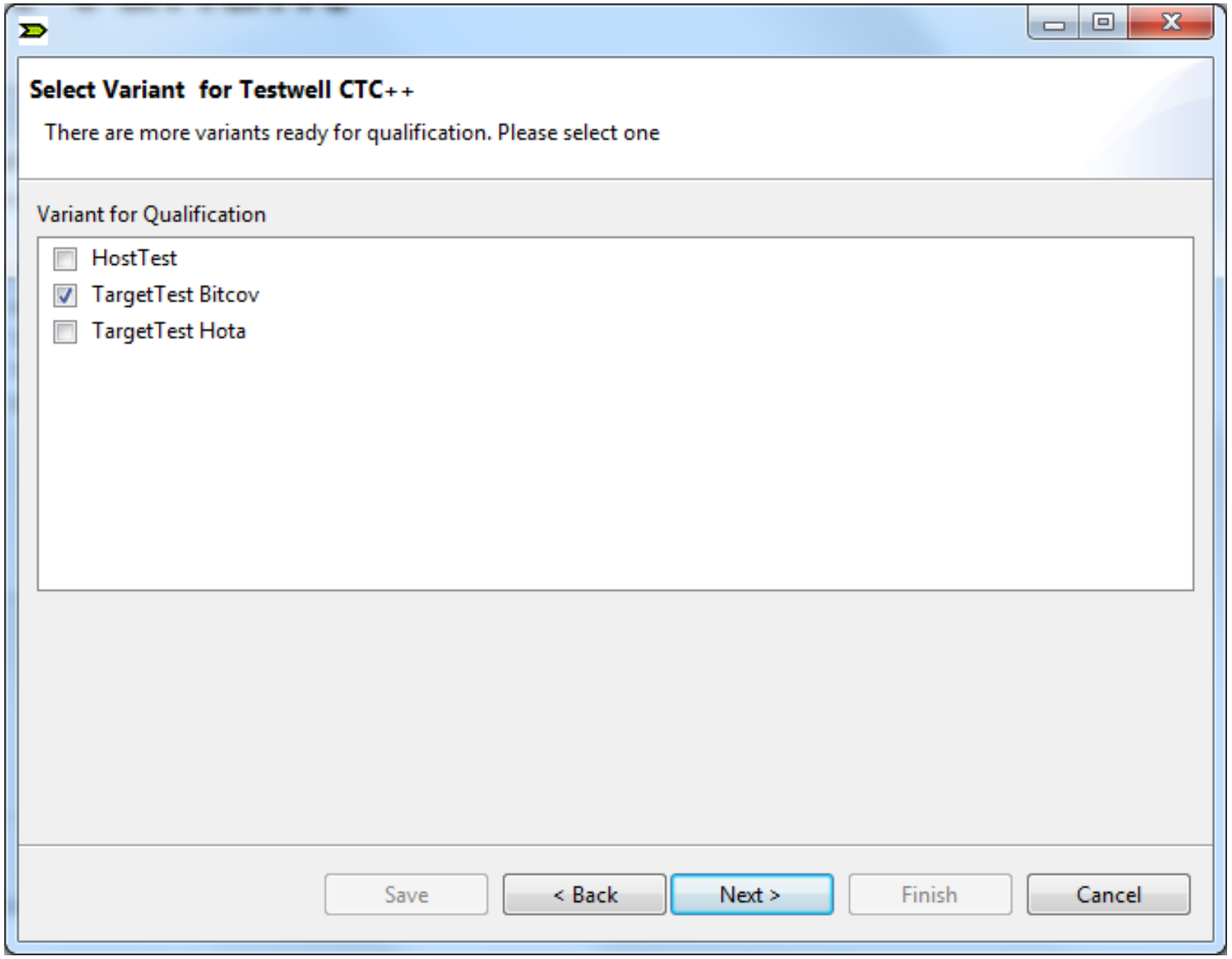

- Select an (empty/non-existent) Directory for Qualification, e.g. C:\Projects\Qualification\ ExampleQualification



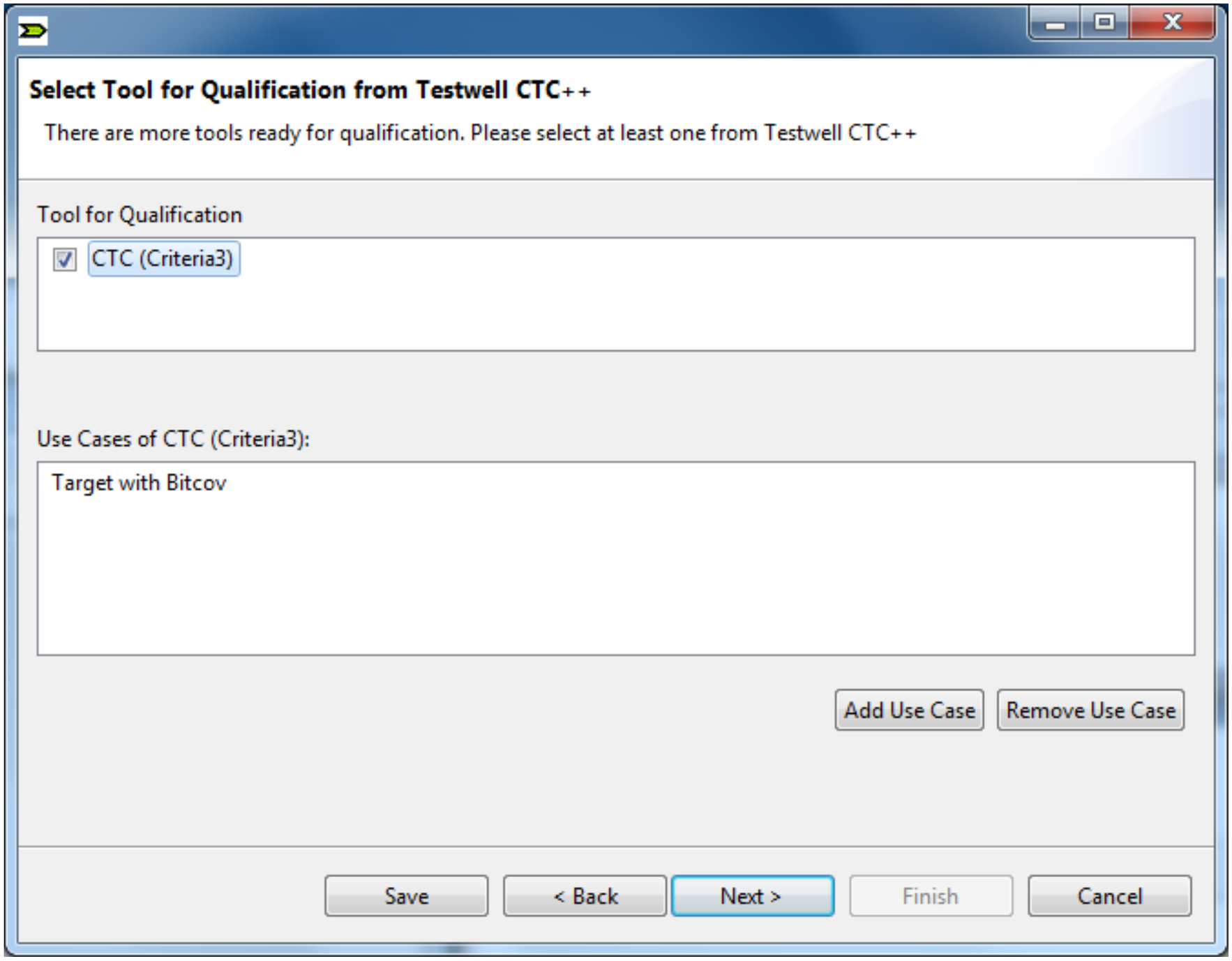
# Select Qualification Standard



# Select Variant



# 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)

Verifysoft TECHNOLOGY

**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)*

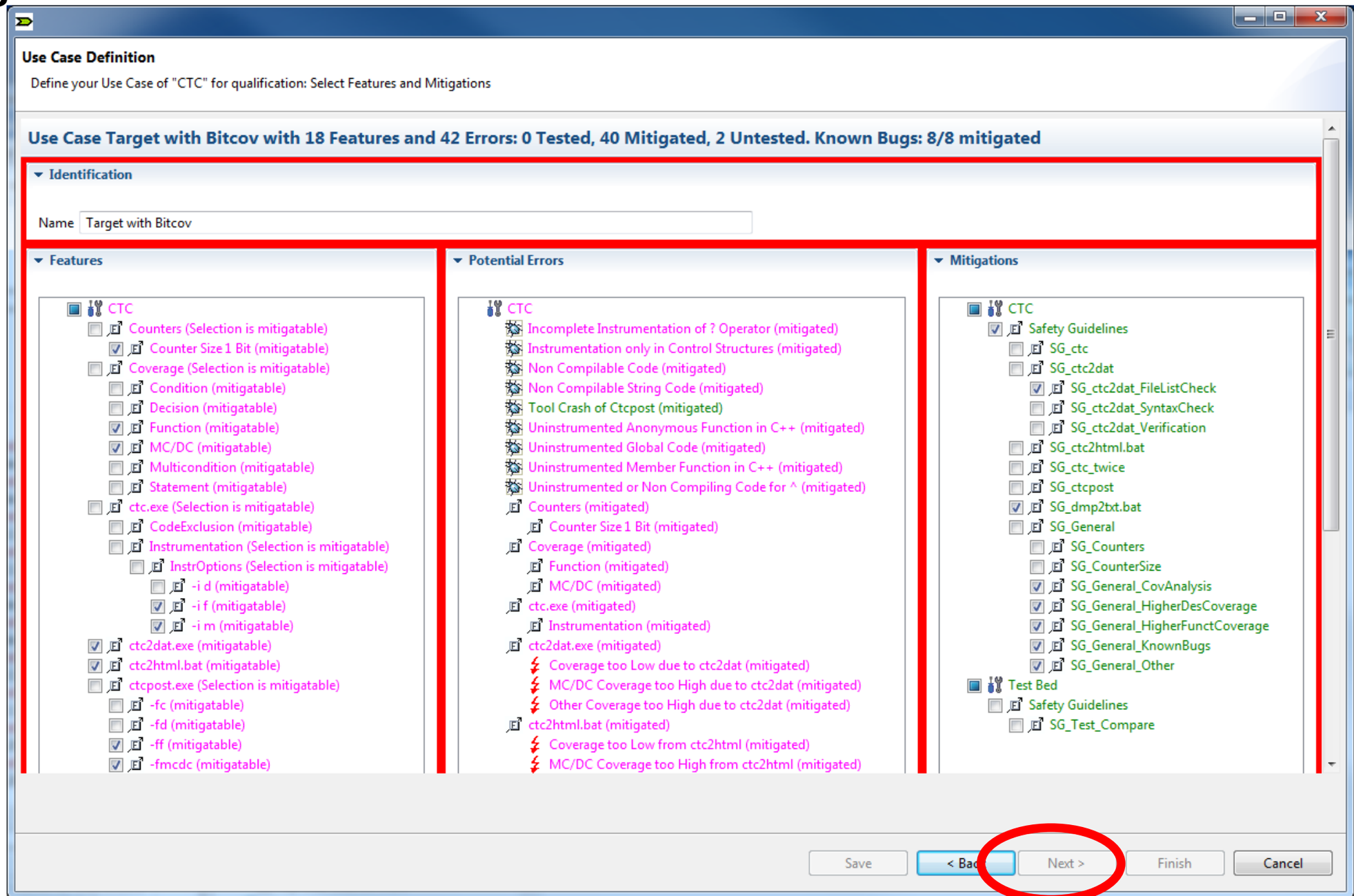
**Mitigations**

- 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

Save | < Back | Next > | Finish | Cancel

# Configure Use Case

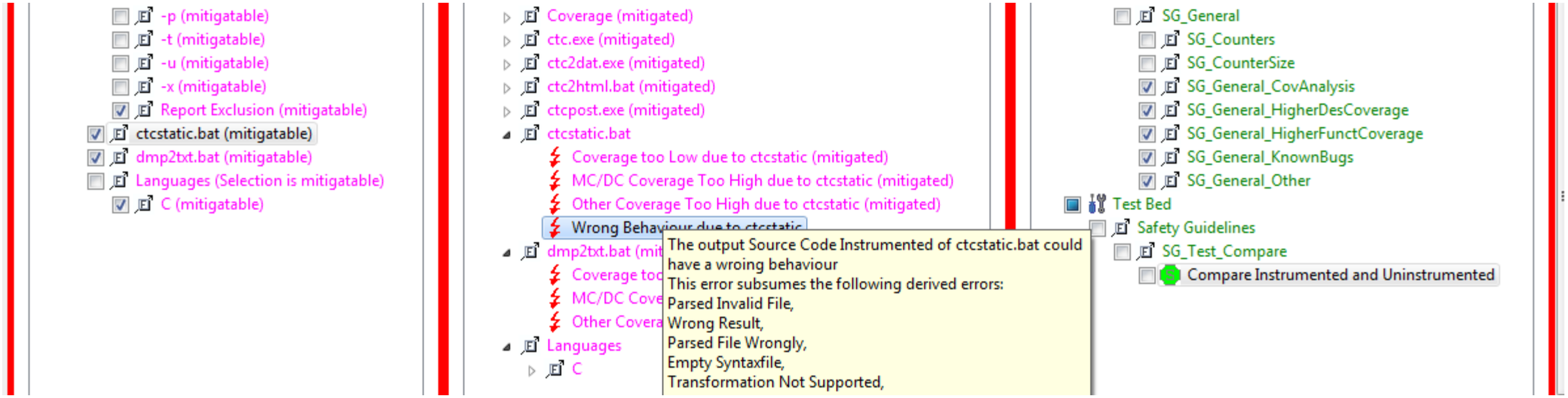
- **Use Case has Features**
- **Features have pot. Errors and known bugs**
- **Errors have Mitigations**
  - Checks
  - Restrictions
- **Status: Next disabled, i.e. not enough mitigations selected**



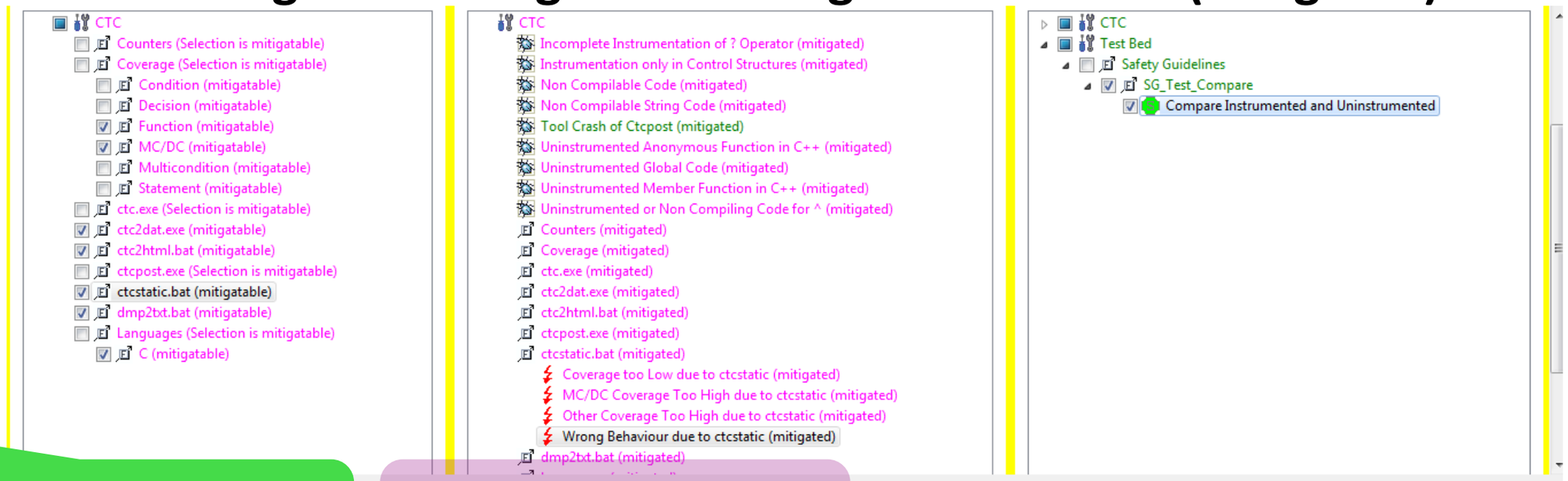
# Configure Use Case: Error Mitigation



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




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

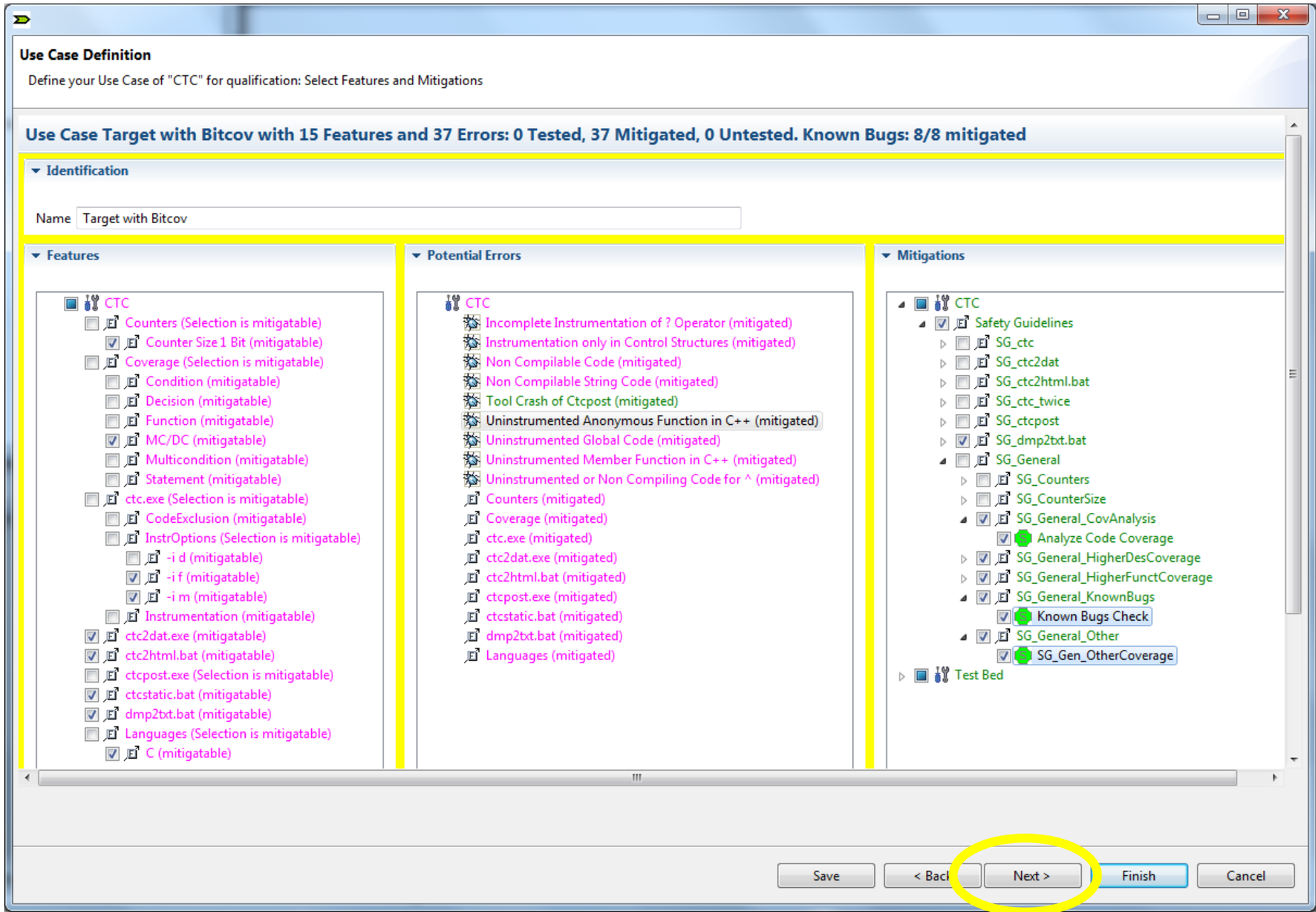



Green Errors are testable and do not need mitigations

Currently no tests available

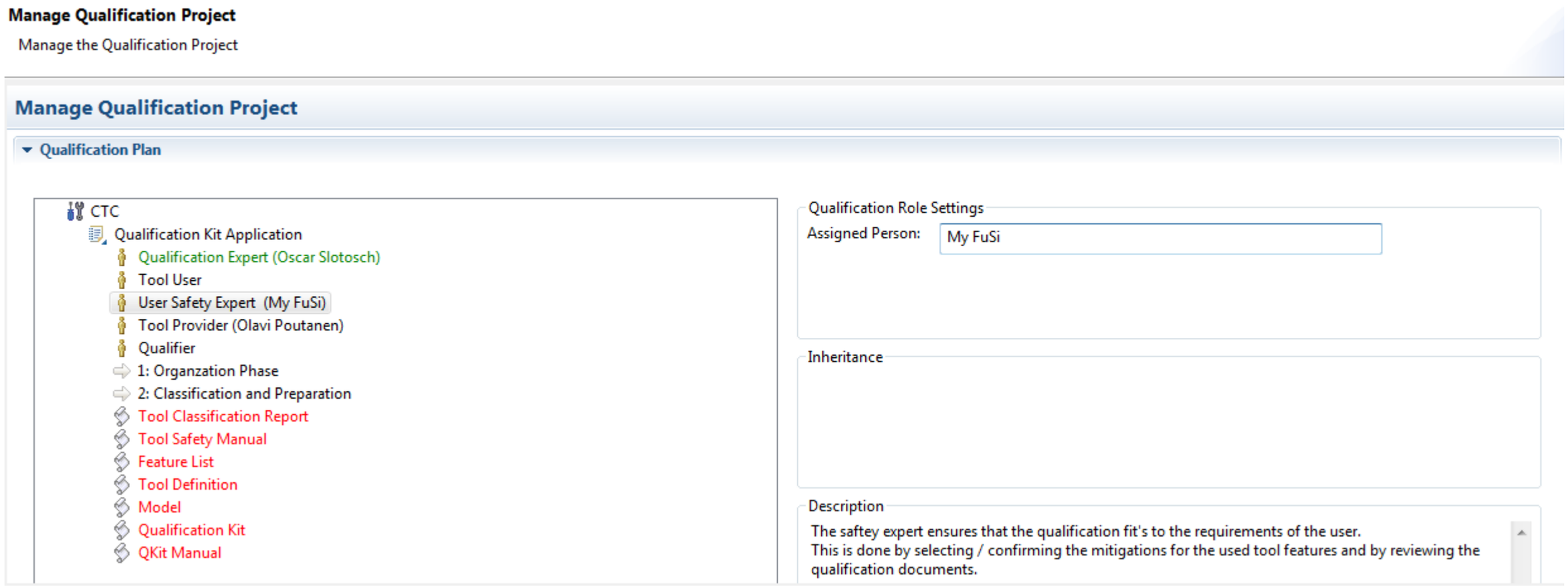
# Use Case: Known Bug Mitigation

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



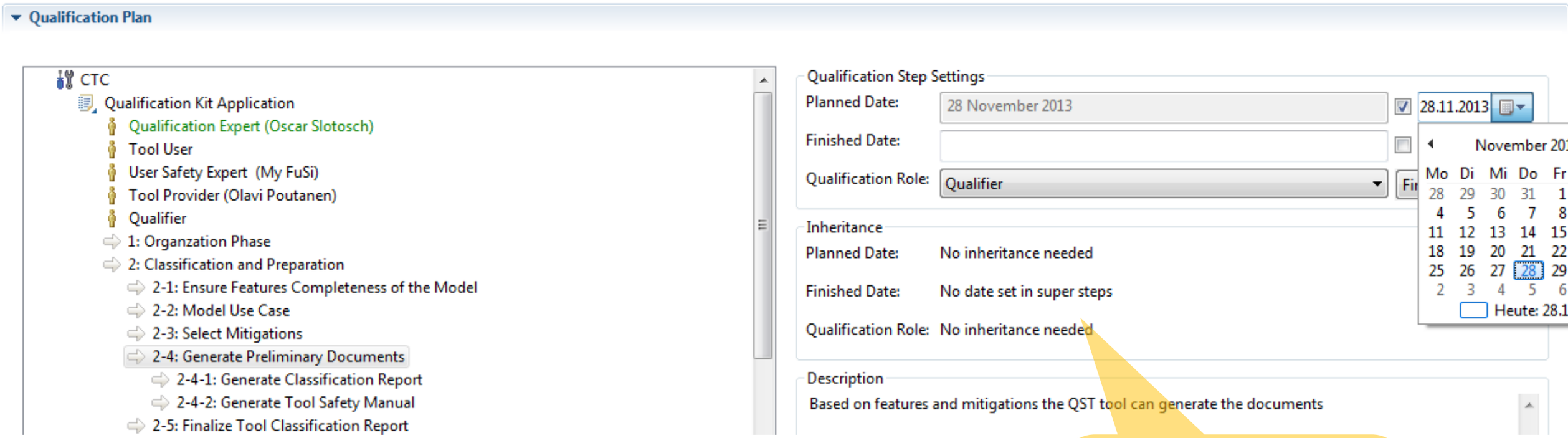
# Optional: Planning of Roles

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



# Optional: Planning of Steps

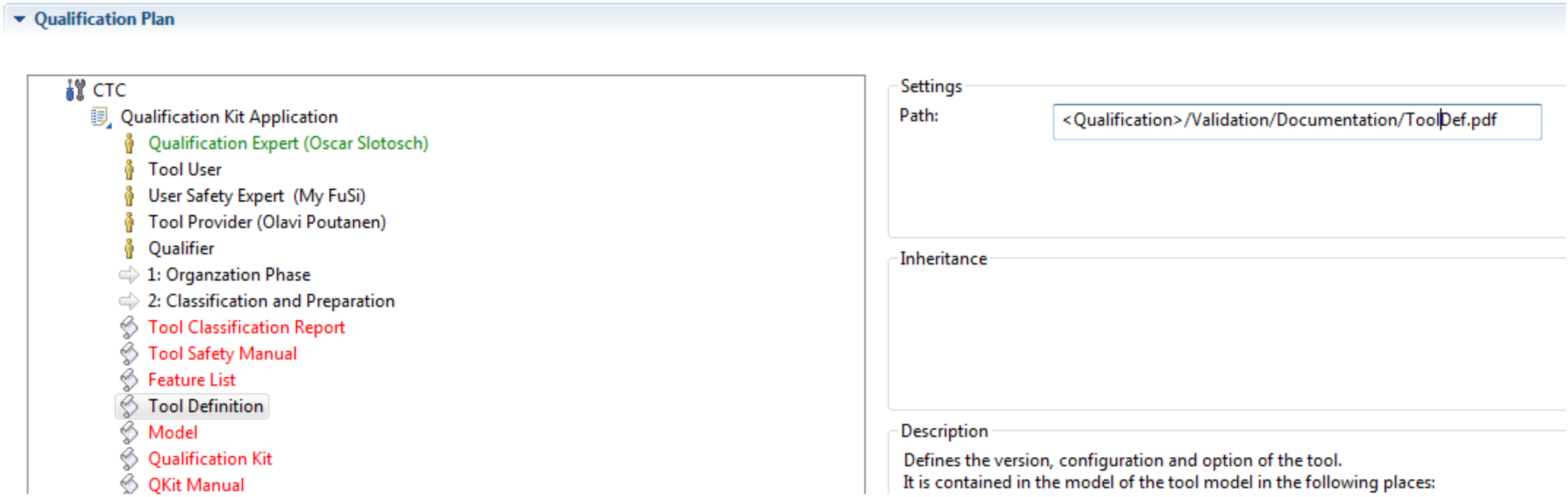
- Select step in the Qualification Project (left side)
- Assign dates (right sides) by entering their values and roles




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**

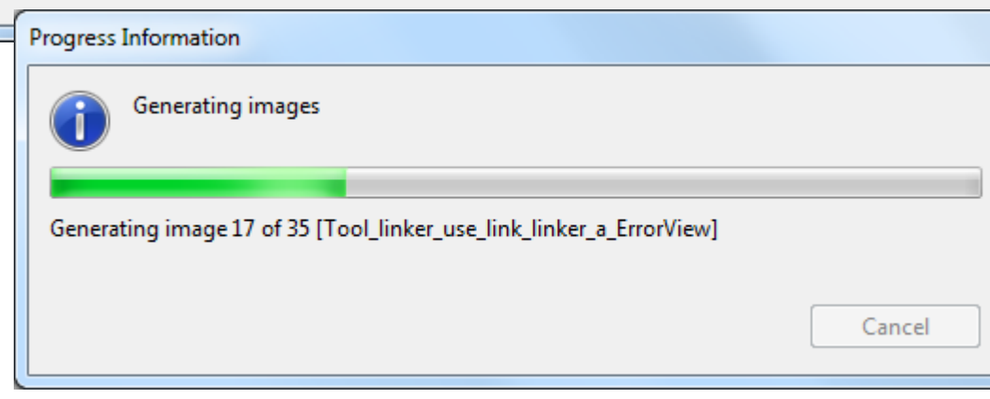
Summarizes the qualification that can be started now by creating the qualification documents of CTC in Target with Bitcov

| | |
|---|---|
| Target Directory | C:\Programme\Qualification\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 |

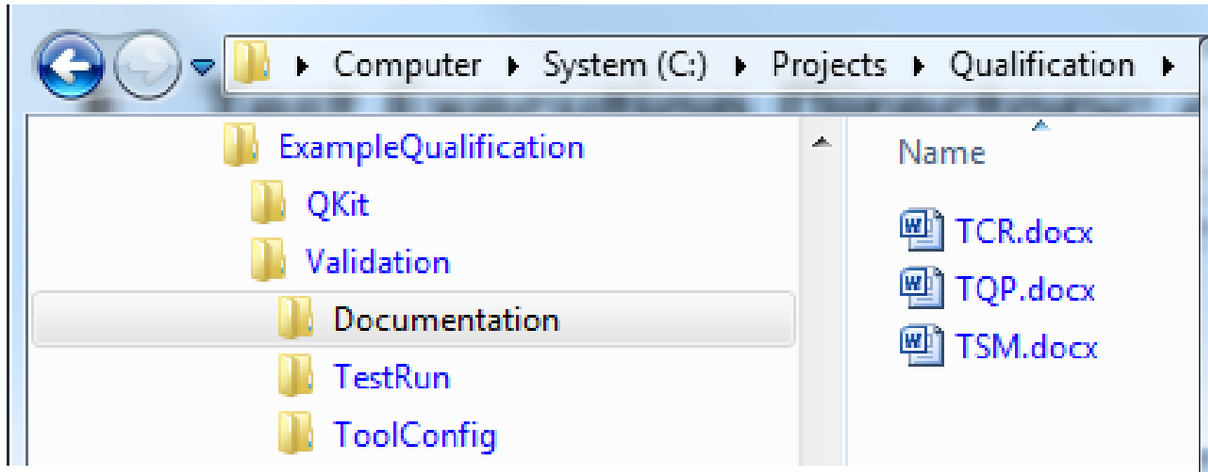
Save | < Back | Next > | Finish | Cancel

- Pressing Finish starts generation of documents

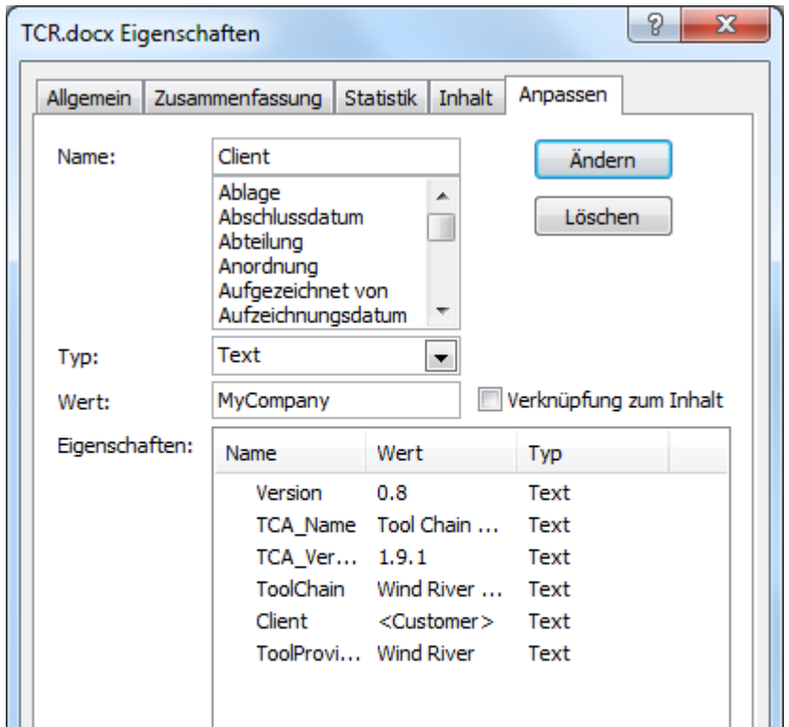


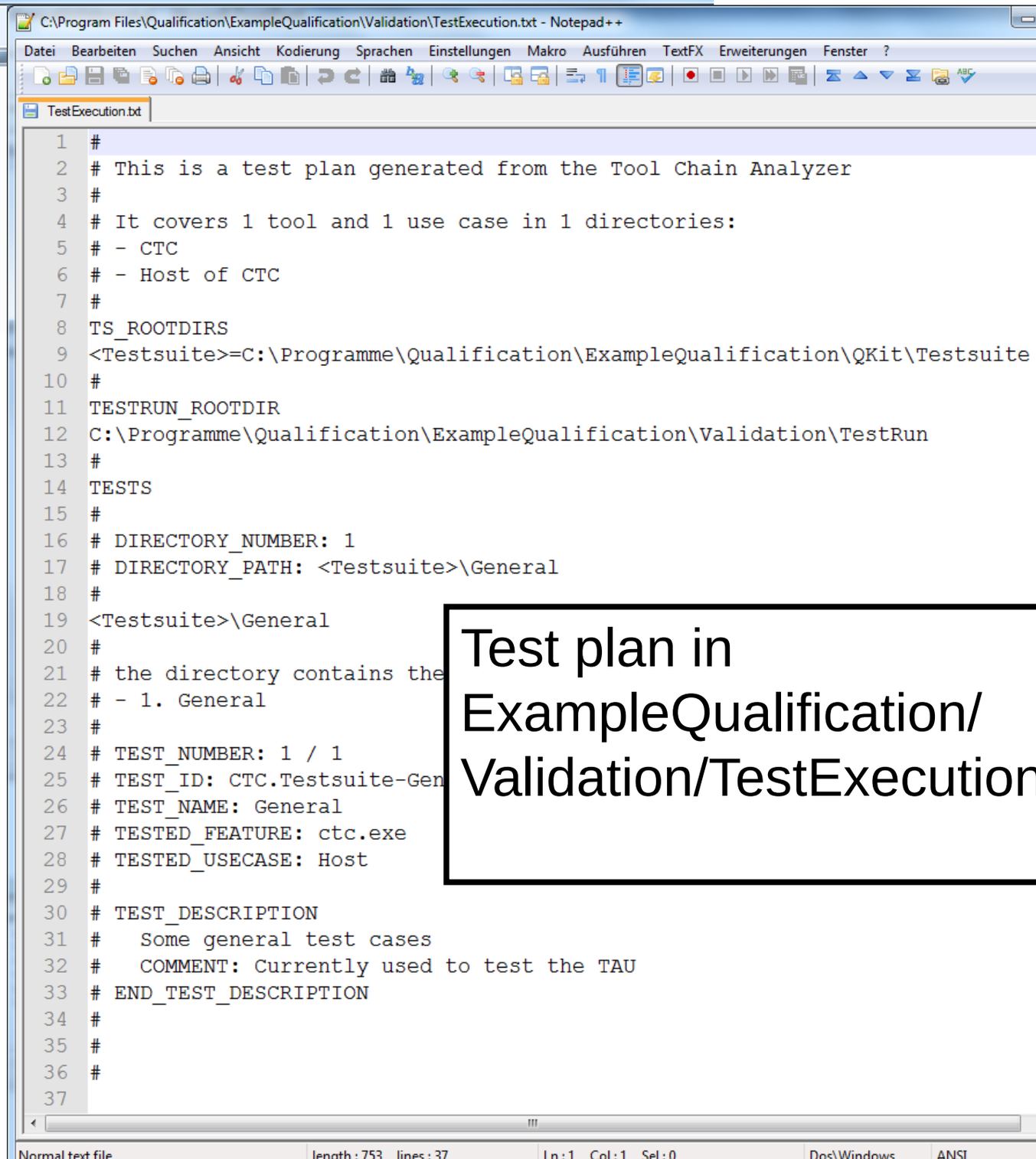
# Generated Documents

- **Qualification Documents**




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



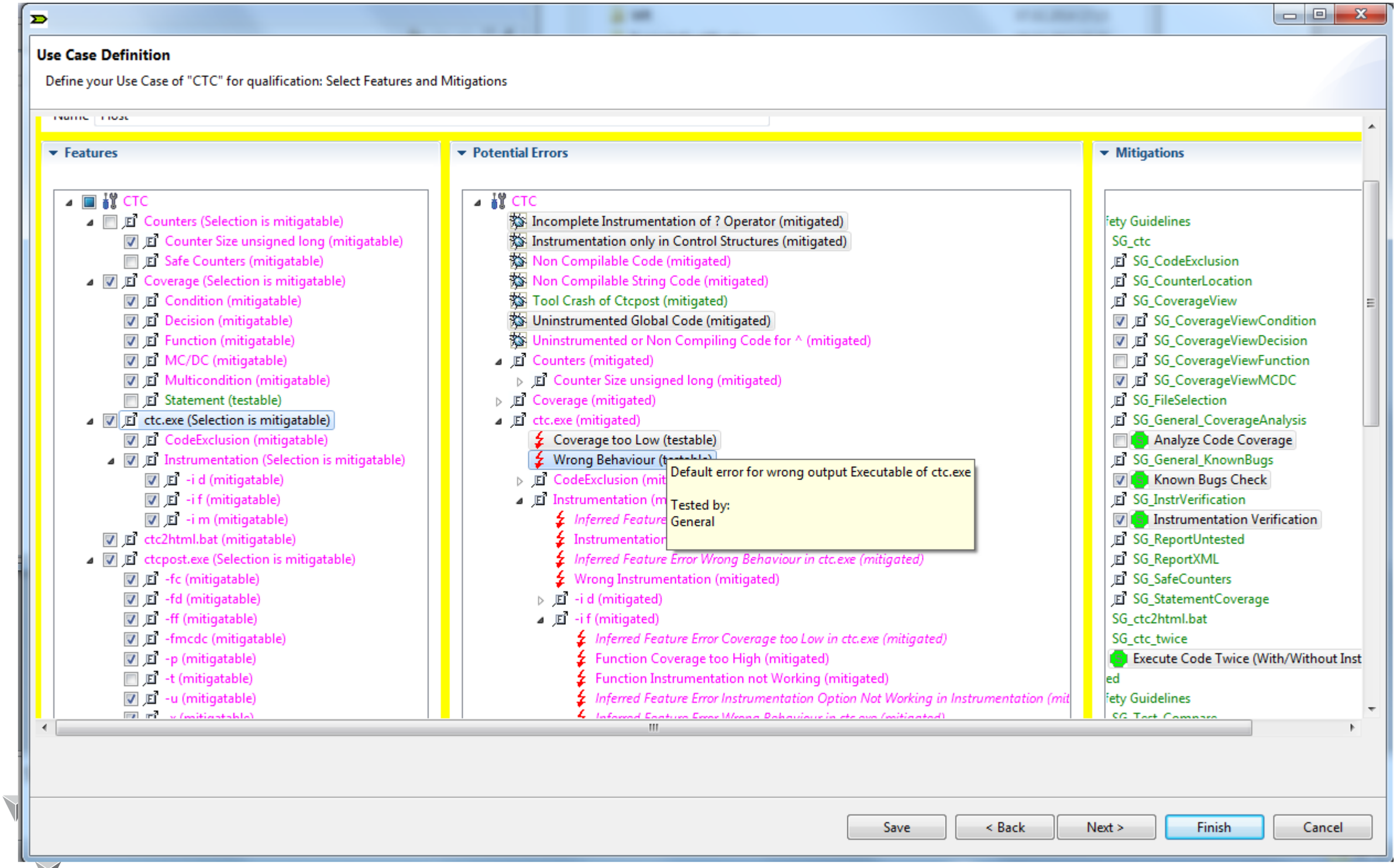

Test plan in ExampleQualification/ Validation/TestExecution

# Content

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

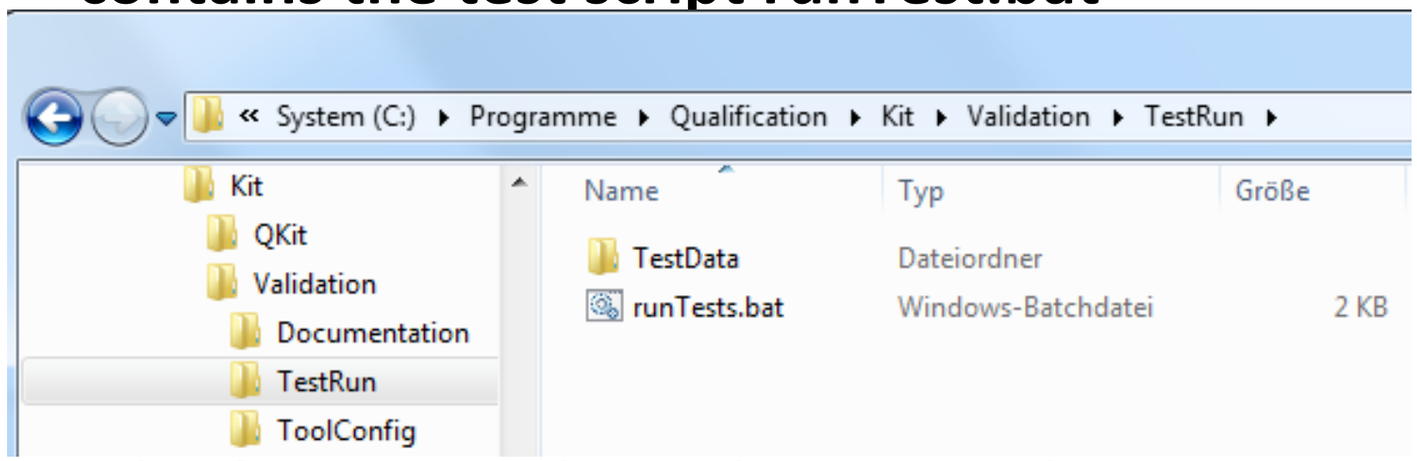
# Test Selection

- Some features are testable (green)

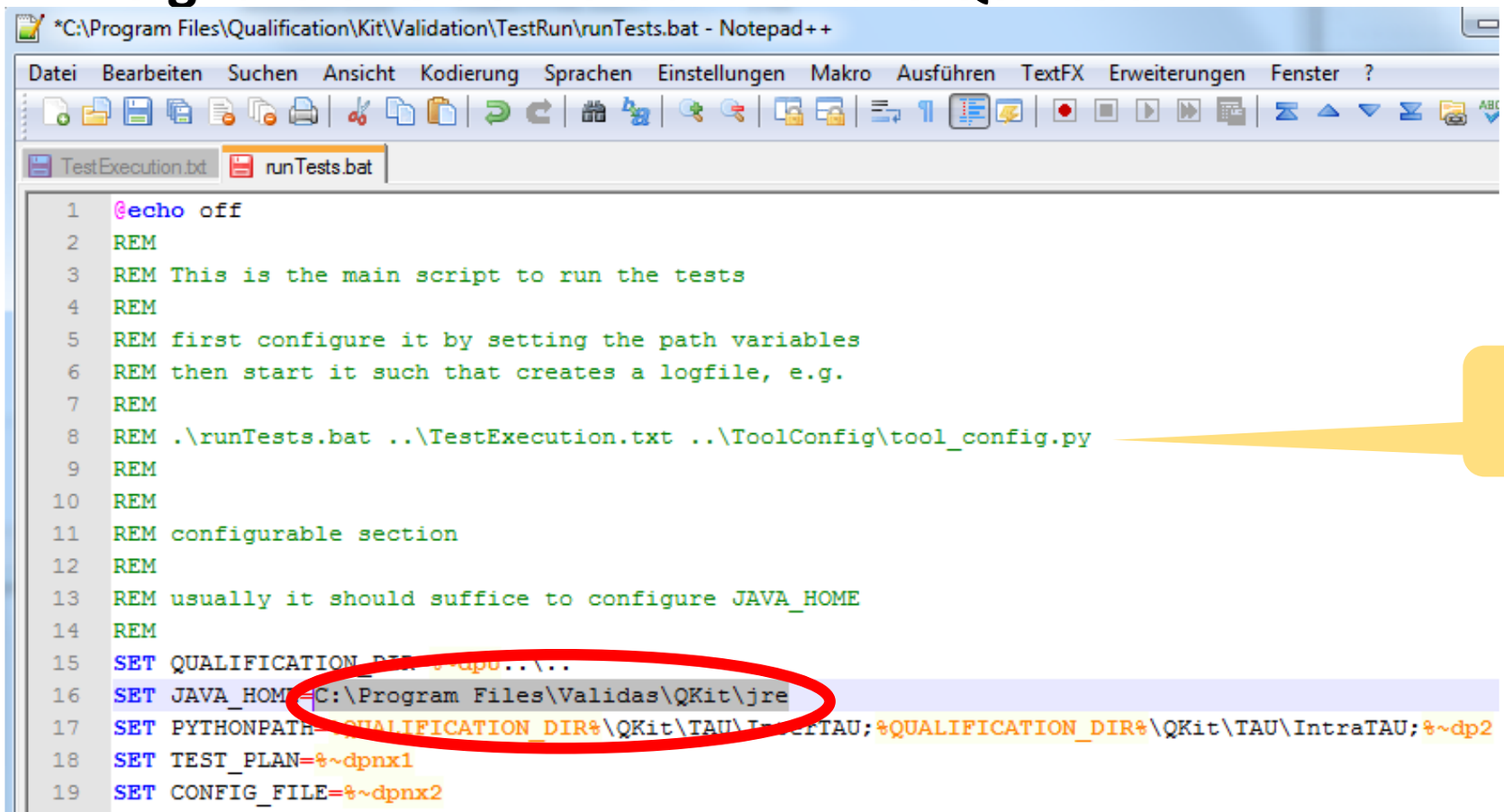


# Test Execution: Configuration

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



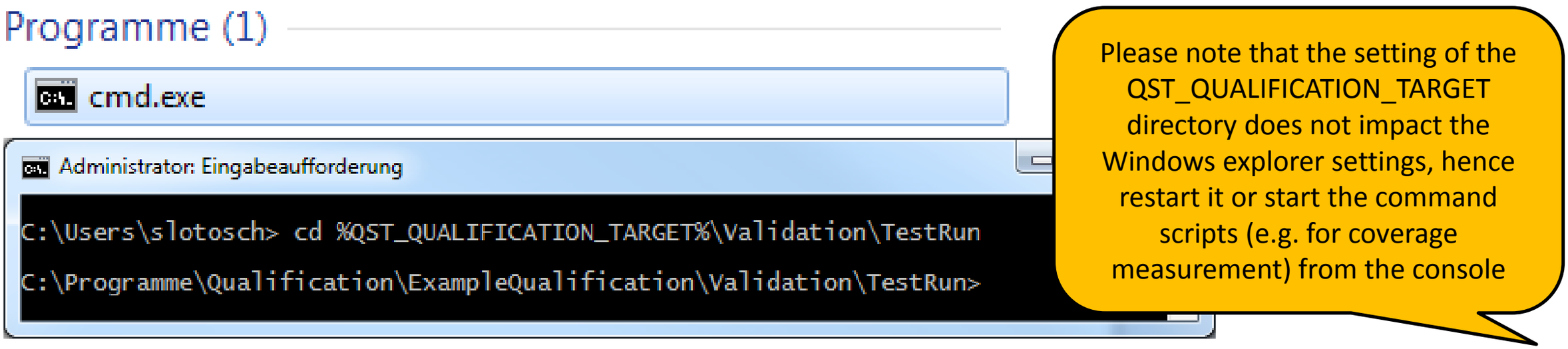

- **Edit the java path inside runTest.bat to a java runtime environment (jre), e.g. the one delivered with the QST tool**




This is the command to start the test execution

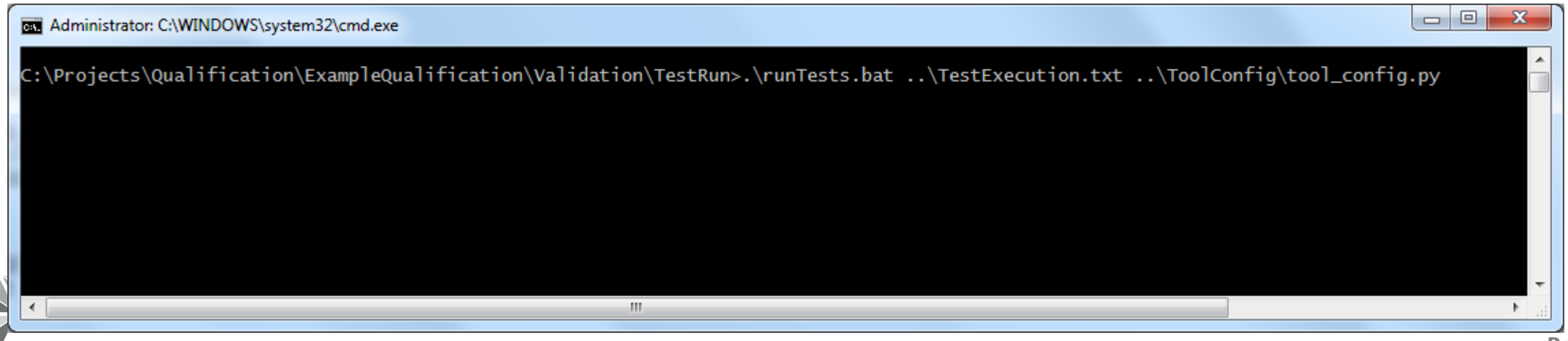
# Test Execution: Run Tests

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




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**



# Test Execution: Progress

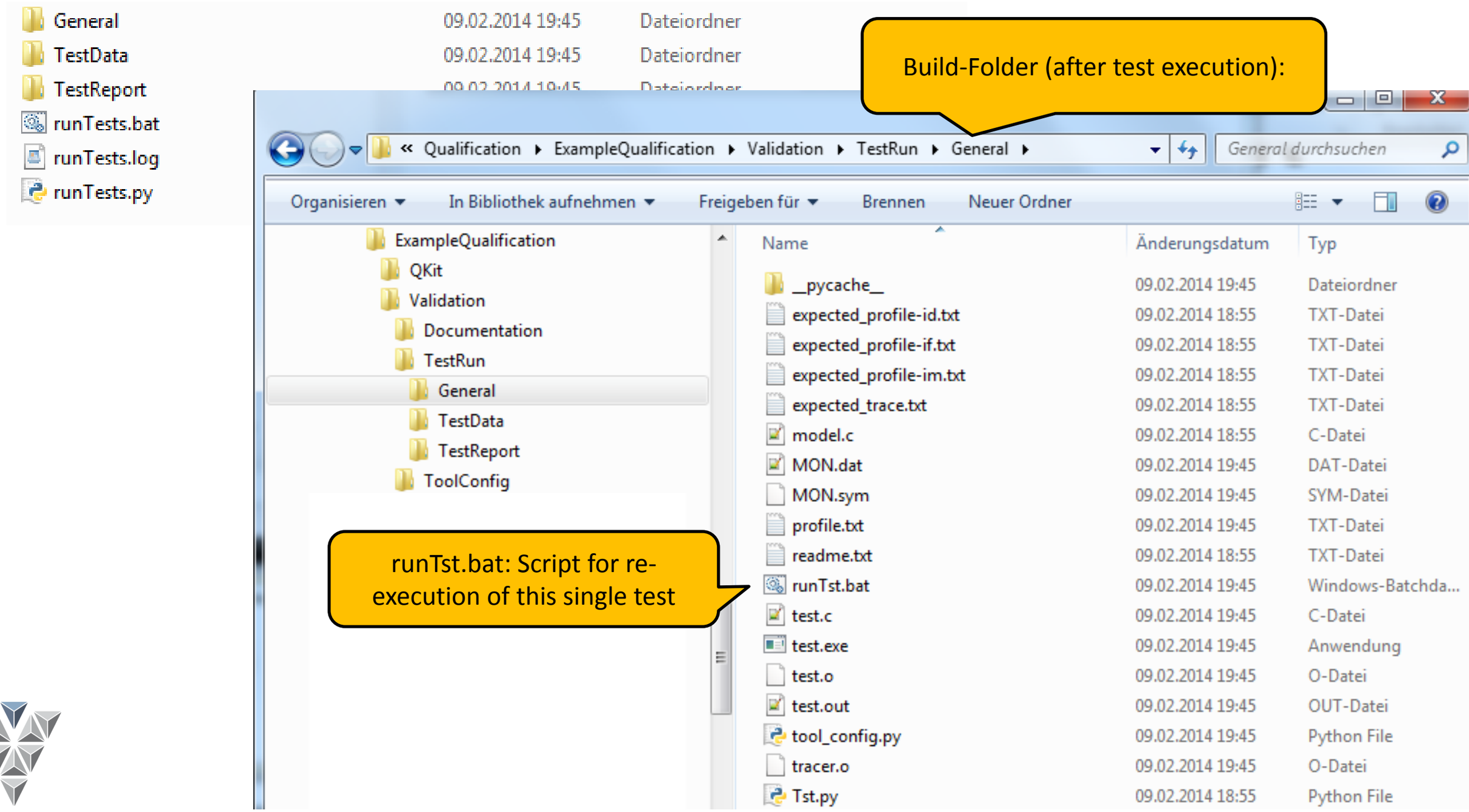
- **Console shows progress of test execution:**
  - Date, times, TAU version, results,…

```
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]
cre
cre
imp
Executing job command: "c:\Programme\Qualification\ExampleQualification\Validation\TestRun\..\..\QKit\TAU\IntraTAU"\t
rexpand.bat model.c > test.c
STDOUT: 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: ctcpost 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



# Test Report: Overview

Verifysoft TECHNOLOGY

Unit Test Results

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

## 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**<br>False is not true : Generated and reference coverage files differ.<br>File "C:\Python32\lib\unittest\case.py", line 370, in _executeTestPart<br>function()<br>File "C:\Programme\Qualification\ExampleQualification\Validation\TestRun\General\Tst.py", line 40, in testDiffReportFiles<br>self.assertTrue(boolOk, 'Generated and reference coverage files differ.')<br>File "C:\Python32\lib\unittest\case.py", line 506, in assertTrue<br>raise self.failureException(msg) | **0.002** |



Properties »

# Test Logfile: TestRun/runTests.log

Verifysoft TECHNOLOGY

- 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\Gener
Copying test files from : C:\Programme\Qualification\ExampleQualification\QKit\Testsuite\General to test fold
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

Verifysoft TECHNOLOGY

- **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: ctcpost 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

Verifysoft TECHNOLOGY

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

```
*******************************************************************************
*           CTC++, Test Coverage Analyzer for C/C++, Version 7.1.1            *
*                                                                             *
*                           EXECUTION PR                                      *
*                                                                             *
*                      Copyright (c) 19                                       *
*******************************************

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  : multicondit

 HITS/TRUE      FALSE     LINE DESCR
===============================================================================

        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 -     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\TestRu
INSTRUMENTATION MODE  : multicondition

 HITS/TRUE      FALSE     LINE DESCRIPTION
```

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                    *
*******************************************************************************


Symbol_file(s)_used___:_host-im.sym_(Tue_Jan_21_08:50:45_2014)
Data_file(s)_used_____:_host-im.dat_(Tue_Jan_21_08:55:51_2014)
Listing_produced_at    : Tue_Jan_21_09:01:23 2014

Coverage view          : As instrumented


MONITORED SOURCE FILE : test.c
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 -     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
-------------------------------------------------------------------------------
```

Different dates

Different number of executions

More Information
(different report options?)

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

# 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