How-To Qualify Eclipse-Based Tools

Version 1.0

Abstract

This document describes how to qualify Eclipse-based tool according to the safety standard DO-330 (Tool Qualification Considerations), see [DO-330].

The DO-330 has requirements to develop tools. Tool qualification for Eclipse is modular with the same modular structure as the tools have, i.e. every plugin can be qualified. The development of qualifiable Eclipse-based tools is a specialized, model-based way to satisfy the DO-330 requirements by using some extensions of the plugin mechanism. Other processes could also satisfy the requirements, but the proposed process has been successfully traced against all requirements in the DO-330. The qualification model contains all qualification relevant data from the complete tool life cycle, especially for requirements, testing and verification.

This document explains the tool qualification process and shows the compliance to the DO-330.

Furthermore some critical functions of Eclipse are identified that need a qualification.

Contents

1 Document History

The history describes the changes in the document.

Version	Date	Author	Changes
0.1	2012.04.10	Oscar Slotosch	First Draft with structure and general introduction
0.2	2012.04.11	Oscar Slotosch	Added references to TDP, TVP
0.3	2012.04.16	Oscar Slotosch	Finished tracing of planning
0.4	2012.04.19	Oscar Slotosch	Feedback from internal Review
0.5	2012.04.30	Oscar Slotosch	More tables and CM
0.6	2012.05.07	Oscar Slotosch	Completed tracing to QA, Liaison process
0.7	2012.05.14	Oscar Slotosch	Restructured Sections and improved overview
0.8	2012.05.22	Oscar Slotosch	Added Tracing to verification process and additional considerations
0.9	2012.05.25	Oscar Slotosch	Added tracing review feedback from Natacha
1.0	2012.06.30	Oscar Slotosch	Reviewed and improved

2 Definitions

This section contains the used abbreviations and some important definitions:

- CA: Certification Authority
- CC: Control Category, see section 4.6 in [TDP]
- CM: Configuration Management, see section 4.6 in [TDP]
- COTS: Commercial Off-The-Shelf
- LLR: Low Level Requirement: every atomic requirement that has one direct association to an implementation element, e.g. a method, variable or ecore class model.
- PSAC: Plan for Software Aspects of Certification
- SAS: Software Accomplishment Summary
- SECI: Software Life Cycle Environment Configuration Index
- TAS: Tool Accomplishment Summary
- TCA: Tool Chain Analyzer: Tool to automatically determine the confidence level required for a tool, based on a formal model in accordance with ISO 26262.
- TCI: Tool Configuration Index
- TCL: Tool Confidence Level
- TCM: Tool Configuration Management
- TDP: Tool Development Plan
- TOR: Tool Operational Requirement: Requirements during the operation of the tool (Use Cases)
- Tool: A qualifiable Eclipse-based tool is a plugin containing the product definition that contains references to the required plugins.
- TOVV: Tool Operational Verification and Validation
- TQA: Tool Quality Assurance
- TQL: Tool Qualification Level
- TQP: Tool Qualification Plan
- TQR: Tool Qualification Report
- TR: The Tool Requirements are used to develop and verify the tool/plugin. They are linked to the TOR and refined by derived TR. Tool Architecture requirements are special cases of TRs.

3 Tool Qualification Process Overview

The tool qualification need depends on the context in which the tool is used. Therefore the tool qualification process is a three phase process

1) Determine the qualification need in the tool operational process (“Planning Phase”). In case of “qualification need” a qualification plan shall be made, see section 5.
2) Qualification
    a. Develop the tool according to the required process (“Development Phase”), see [TDP] and
    b. Verify that the tool works correctly in the tool operational process (“Integration Phase”). This is done according to the qualification plan and documented in the qualification report, see section 6. In the case there is a cooperation with an authority, there shall be a liaison process that satisfies the requirements described in section 4 and
3) Use the tool for production of software in safety relevant systems (“Production Phase”)

The tool qualification process is a part of the software qualification process that is usually carried out between the tool user and an authority that qualifies the software.

The main interface of tool qualification between tool user and tool provider is the amount of required confidence for the used functions. The user determines his confidence requirements: none (0) / low (1) / medium (2) / high (3) and the tool provider has to justify this confidence by providing a sufficiently qualified tool including the evidence for the tool qualification. A high confidence requirement (e.g. tool confidence level 3) expresses that the users trust the tool without verifying all the results. This can be justified by providing a rigorous development with a high tool qualification level (e.g. TQL-1) and vice versa.

The compliance with the DO-330 is achieved by the tool qualification process. The process consists of a general part which is independent from the qualified tool (as this document or the generic tool development plan) and tool specific parts like the requirements, design, and other documents.

The development process is based on a formalized model for the development items. The description of the model is contained in the tool development plan document [TDP] and the tool verification plan [TVP]. The model has three main purposes:

- Verification support: many verification steps can be done automatically based on this model, for example the TQL computation, the transition criteria, the tracing between the artifacts etc.
- Documentation support: from the model the most tool specific documents can be generated, e.g. a requirement specification or a verification plan.
- According to the determined TQL Eclipse can also tailor the process according to the control categories from the DO-330.

Therefore the model eases the qualification significantly but creates requirements to the new qualification supporting components, which are classified according to DO-330-4.4.e as TQL-5.

This document serves as a guide how to qualify Eclipse-based tools by describing the qualification process and gives a general overview on the documents. Furthermore the compliance of the approach to the DO-330 is documented using a tracing between the

requirements from DO-330 and the qualification approach for Eclipse-based tools. This is done using a bi-directional tracing within the tables in section 8 that describe the fulfillment of the requirements with pointers to the documents. The documents contain links to the table, e.g. satisfies: DO-330.5.2.1.a.
The following documents are part of the qualification process

- Generic documents:
  - Tool Development Plan, see [TDP]
  - Tool Verification Plan, see [TVP]
  - Tool Qualification Report Template
- Specific documents for <ToolName>
  - Tool Requirements for <ToolName>, see example in [TCA_TR]
  - Tool Design for <ToolName>, see example in [TCA_TD]
  - Tool Test Specification for <ToolName>, see example in [TCA_TTS]
  - Tool Qualification Report for <ToolName>

The generic documents describe how to derive specific documents for Eclipse-based tools. Adaptation of them is possible but requires re-executing the DO-330 compliance check provided within this document (section 6). The specific documents depend on the developed tool and the content of the DO-330 model of this tool. The specific documents can be generated from the model. To facilitate this process examples for the specific documents are given. More details on the content of these documents can be found in the section 8.7.

Since the tool qualification process is different, depending on the required TQL some actions do not need to be executed for every TQL. This is considered especially during the verification activities and is respected by the corresponding TQLs.

The qualification is based on the same architecture as the architecture of Eclipse-based tools. Every Eclipse plugin project will have a TQL and the corresponding tool qualification model with the required information in it as described within the Eclipse tool qualification process. Therefore the qualification process relays on the correctness of these infrastructure that has to be qualified. The following mechanisms are used and need a qualification:

- Eclipse-Plugin-Framework including (according to DO-330-4.4.e TQL-5 suffices for qualification tools, however for other applications higher level might be required):
  - DO-330-Modeling (EMF),
  - Determination of the tool confidence level,
  - Verification-Checks on the DO-330 model, especially including the checks of the transition criteria and the missing elements for the next qualification stage,
  - Support for the TQLs and CC handling to omit some steps including and verification against the CCs in the tables in appendix of [DO-330].
- Change-Management for the DO-330 model
  - Git,
  - Gerrit and
  - EDF.
- Code-Coverage Measurement,
- Plugin-Separation Mechanism,
- Some OSGI-Features and the Eclipse classloader for integrating the tool (see DO-330-6.1.4.3.2.c) and
- Generator of required qualification documents from the DO-330 model.

The criticality of all used elements and their required functions will be determined based on "Alpha" qualification of the required plugins. The classification and qualification planning of these plugins is subject to further discussions within the Eclipse community.

4 Qualification Liaison Process

The qualification liaison process demonstrates the certification authority that all used tools in the development tool chain comply with their requirements in the development environment. For every used tool the tool operation requirements (TORs) have to be determined from the tool chain definition. The tool chain definition consists of all artifacts that are used and the mapping to the use cases of the tools to describe the processes the tools support (satisfies: DO-330-10.1.16.b). Every use case of a tool is a TOR and has assigned input and output artifacts. This is modeled in the analysis part of the DO-330 model (see section 4.2.2 in [TDP]) and allows to determine the qualification need automatically (see section 4.2.3 in [TDP]) and the results are the tool specific aspects in the PSAC that can be generated from the model, see an example in [TCA_TQP]. After the qualification of the tools the results are the tool-specific aspects of the SAS.
The TQL is determined with an alternative method according to (DO-330-FAQ-D.3) by considering all use cases (TORs) their potential errors and mitigations to determine the impact of the development process. This is done to be conformant to other standards (ISO 26262) and to have more flexibility e.g. in the use of diverse tools to reduce the criticality of tools (satisfies: DO-330-11.5.a, DO-330-11.5.b, DO-330-11.5.b.1, DO-330-11.5.b.2, and DO-330-11.5.b.3).
Figure 1 shows how the compliance of the tool with the TORs is satisfied. Since all elements and relations are part of the formalized DO-330 model the verification can be proved automatically from Eclipse verified.

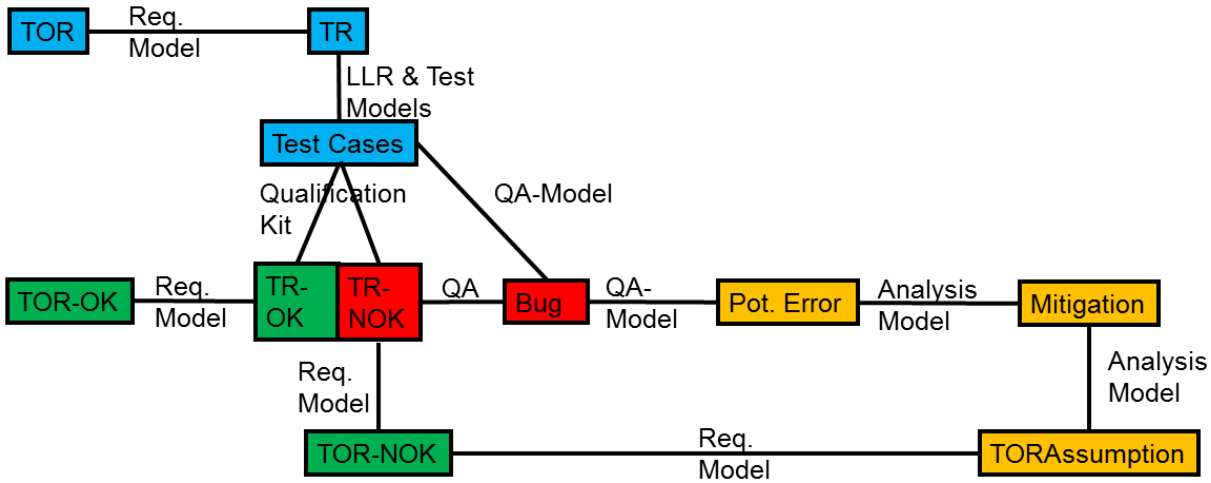

Figure 1: Evidence for Operational Requirements Compliance

Figure 1 explains how the TORs are satisfied. Every TOR has assigned TRs in the requirements model. This is ensured from the verification of the transition criteria for the qualification release. Every tool requirement has assigned test cases (directly or indirectly via the LLRs that have test cases). The set of test cases that trace to the TORs are the qualification kit for the TORs and show the conformance of the tool in the operational environment of the user if they execute correctly. If all tests of a TR pass, this TR is satisfied in the operational environment ("TR-OK"), and if all TRs that are required for a TOR are satisfied also the TOR

is satisfied (“TOR-OK”). More interesting is the case of known problems or new problems in new operational environments. In this case some test of the qualification kit fail (“TR-NOK”), and the tracing allows to identify the violated TRs and TORs (“TOR-NOK”). The QA process ensures that the found problems are reported (in case the user has failing tests during qualification, he is responsible for reporting them) and assigned to potential errors from the analysis model (Satisfies: DO-330-11.4.2.a). The analysis model also contains mitigations for potential errors (checks/restrictions). In case there are no mitigations for this error the “default mitigation” would be not to rely on the TR, e.g. by not using the function of the tool. The found mitigations are passed to the user in form of TORAssumption elements that are assigned to the TORs they belong to. An example for this situation is a known bug in a compiler optimization that restricts the user to use only optimizations until a certain level where the error does not occur.
After a successful application of the qualifications in all tools the qualification reports have to be integrated into the tool specific information in the software accomplishment summary for the developed product (satisfies: DO-330-10.1.16.c) and into the tool-specific information in software life cycle environment configuration index, satisfies: DO-330-10.1.17.b.
The **stability** of a tool can be determined by analyzing the effect of changes requiring to fix some reported bugs. If the changes have little effect the tool is stable. The qualification kit includes all required information to do such analyses (satisfies: DO-330-11.4.2.d). This shall also include the environment information i.e. a tool can have different stabilities in different environments (satisfies: DO-330-11.4.2.e).
Reuse of qualification is possible (if certification authority agrees), since all relevant data are contained in the qualification kit and the kit has only to be applied in the new environment (provided that the environment is not equivalent) and the environment requirements have to be verified again. If the tool version is changed or new relevant bugs are reported in the tool life cycle data the tool needs requalification of the effected parts as if they were new, but existing elements from previous qualifications can be reused. If bigger changes are made (process, environment, requirements, design, implementation), than the corresponding elements have to be re-qualified. The impact analysis is done from Eclipse in the same way as the transition criteria are verified during determination of the qualification state in section 5.3.4 in [TDP].
Satisfies: DO-330-11.2.1.a, DO-330-11.2.1.b, DO-330-11.2.1.c, DO-330-11.2.1.d, DO-330-11.2.1.e, DO-330-11.2.1.f, DO-330-11.2.1.g, DO-330-11.2.2.a, DO-330-11.2.2.b, DO-330-11.2.2.c, DO-330-11.2.3.a, DO-330-11.2.3.b, DO-330-11.2.3.c, and DO-330-11.2.3.e.
A special case of reuse is the qualification of **COTS** tools/plugins that are unchanged integrated into the developed tool. In this case the DO-330-11.3 requires all objectives from the COTS tool that are required from other tools. A proposal is made how the objectives can be split up between the user of the tool (see table DO-330-11-1) and the developer of the tool (see table DO-330-11-2). Both together cover all objectives. Since the Eclipse development of qualifiable plugins follows the same hierarchic principle between users and providers of plugins, there COTS qualification is no special case but the standard case for Eclipse and there is no difference between a “developer-TOR” and a TOR that is used for the determination of the qualification need. Hence the Eclipse development process for qualifiable plugins (and their verification) satisfies all items in DO-330-11-3.

5 Tool Qualification Plan

The tool qualification plan describes the application of the qualification kit in the environment of the user. The tool qualification plan is a tool specific document which is required for every

qualifiable tool or qualifiable plugin. The TQP needs to be adapted from the tool users to the configuration(s) and environments to be qualified.
The tool qualification plan can be generated from the DO-330 model if the qualification stage Beta-Release ("Feature Complete") has been reached. See section 5.2 and 5.3.2 in [TDP] for the tool life cycle and transition criteria to this state.
The tool qualification plan should contain the following information from the model.

- Identification of the tool from Project.ID in section 4.1.1 in [TDP] (satisfies: DO-330-10.1.2.a)
- Configuration of tool from the tools product file (for products) or the project settings (for plugins) as used in the integration of the product, see section 6.2 in [TDP] (satisfies: DO-330-10.1.2.a)
- Qualification Considerations (satisfies: DO-330-10.1.2.b). Note that this information can be completely generated from the DO-330 model (analysis part), see section 4.2.2 in [TDP] with
    - Required Risk Level (from the Product information) that corresponds to the domain tool criticality, e.g. TCL, Risc Class, Criticality
    - Computed TQL according to the analysis of the potential errors and their mitigation possibilities (including evidence for the TQL) and
    - List of TORs from the DO-330 model in section 4.3.1 of in [TDP] and for each TOR
        - TOR Name,
        - TOR Description,
        - Assigned potential errors with:
            - Error Name,
            - Error Description,
            - Max. Error mitigation probability (LOW / MEDIUM / HIGH),
            - List of mitigations (checks / restriction) for the error with:
                - Name,
                - Description,
                - Probability and
                - Artifact by which the mitigation detects / prevents the error and
        - Inputs / Output Artifacts of the TOR,
- Functional overview with the tool operational requirements (including the architecture requirements) and the required plugins/external components from the DO-330 model in section 4.3.2 in [TDP] (satisfies: DO-330-10.1.2.c, DO-330-6.2.2.a.2),
- Description of the operational and verification environment of the tool (from the tools product and preferences section, may be with references to the tool configuration section above) (satisfies: DO-330-10.1.2.d),
- The tool life cycle data are contained in the qualification build of the tool and can be reviewed any time from the certification authority, see section 6.2 in [TDP] (satisfies: DO-330-10.1.2.e). This includes especially also the open problem reports and their associated test cases, since they are expected to fail during qualification (satisfies: DO-330-11.4.2.g.1, DO-330-11.4.2.g.2, DO-330-11.4.2.g.3),
- The hint to the tool life cycle description in section 5 in [TDP] and the qualification activities (satisfies: DO-330-10.1.2.f):
    - Execution of all tests (including of coverage measurement), analysis the produced test reports for deviations comparing with the tools quality report,
    - Verification of the tools qualification stage "Qualification Release" using Eclipse see section 5.3 in [TDP],

    - Verification of the required plugins and
    - Ensure that the found restrictions and constraints are satisfied by the users of the tool, for example by creating a tool's safety manual,
- The expected qualification data is (satisfies: DO-330-10.1.2.g):
    - Test report(s) including the required reports for plugins,
    - Model verification report including the required reports for plugins,
    - Safety manual (if constraints have to be respected) containing the descriptions of all assumptions (TORAssumption with affected TORs) that have to be fulfilled due to the known problems and the problems found during qualification (satisfies DO-330-9.d) and
    - A signed version of the tolls quality report from the validator,
- Additional considerations, for example if non-Eclipse tools have been used with different qualification methods (satisfies: DO-330-10.1.2.h) and
- Organization responsibilities: Tool provider and tool validator from the plugins (including the required plugins) (satisfies: DO-330-10.1.2.i, DO-330-10.1.2.j)

An example for a tool qualification plan can be found in [TCA_TQP].
The tool qualification plan needs to be reviewed, especially to ensure that the TORs from the tool fit to the process in which the tool shall be used (satisfies: DO-330-6.2.1.a). This includes also the operational environment (satisfies: DO-330-6.1.3.1.c).

6 Tool Qualification Report

The tool qualification report contains the tools qualification for the use in the specified processes and specified environment. It has the following structure and can be mainly generated from the DO-330 model of qualified tool and the results of the application of the tool qualification plan (i.e. the verification that the tool also works in the user's environment, satisfies: DO-330-6.2.1.b, DO-330-6.2.2.c.4). It integrates the information from the tool accomplishment summary (DO-330-10.1.15 and DO-330-10.1.16 and DO-330-10.1.17).
Especially the known open problems are expected to cause errors during the qualification. They shall be analyzed in the qualification process (satisfies: DO-330-11.4.2.g)
Note that the TQP can be updated if changes are necessary or the changes to the TQP can be recorded in the qualification report.
It contains the following information

- Tool identification with the ID from the project,
- Tool overview with differences (e.g. newer versions, new problems...) compared with the TQP (satisfies: DO-330-10.1.15.a, DO-330-10.1.16.f). New differences shall be added to the QA system as problem reports (satisfies: DO-330-10.3.4.d),
- Tool qualification considerations with TQL and operational environment (satisfies: DO-330-10.1.15.b),
- Tool life cycle with differences compared with the TQP (satisfies: DO-330-10.1.15.c), especially the test results (satisfies: DO-330-6.2.2.c.2) and their review in case of deviation from the expected results (satisfies: DO-330-6.2.2.c.3) and
- Tool life cycle data (if differences in the TQP).

7 Tool Operation Phase

After the qualification of the tool, the tool can be used for its purpose. The following thinks have to be ensured:

1) Correct installation, documented in a report with the version of the tool, the required plugins versions, hardware configuration, the operating system configuration and installed options, etc. This satisfies: DO-330-10.3.2.b, DO-330-10.3.2.a. Note this includes also the wrapper components for external components used (satisfies: DO-330-10.3.2.c), if the tool is not executed with the default start as described in section 6.2 of [TDP],
2) Correct usage according to the assumptions, especially due to the mitigation of potential and real errors and
3) Checking for new errors in the tool as described in the QA process.

8 Traceability to DO-330

In this section we provide the traceability from the [DO-330] standard to the Eclipse approach for tool qualification. This is achieved by a bi-directional tracing from the Eclipse documents to the DO-330 requirements in this chapter and vice versa from this chapter to the Eclipse Documents. The considered Eclipse documents are the generic documents that are used for every qualifiable tool/plugin:

- This guide Howto Qualify Eclipse-based Tool (the other sections),
- The Tool Development Plan [TDP] and
- The Tool Verification Plan [TVP].

8.1 General Considerations

The DO-330 contains the following sections

1) Introduction
2) Purpose of Tool Qualification
3) Characteristics of Tool Qualification
4) Tool Qualification Planning Process
5) Tool Development Life Cycle and Process
6) Tool Verification Process
7) Tool Configuration Management Process
8) Tool Quality Assurance Process
9) Tool Qualification Liaison Process
10) Tool Qualification Data
11) Addition Considerations for Tool Qualification

a) (Annex) Tool Qualification Objectives
b) (Annex) Acronyms and Glossary Items

a) (Appendix) Membership List
b) (Appendix) Example of Determination of Applicable Tool Qualification Levels
c) (Appendix) Frequently Asked Questions Related to Tool Qualification for all Domains

We provide the tracing to the important sections 4) to 11). Annex a) shall be considered during implementation in Eclipse, since this allows to simplify/omit some steps. We stick to the terms and definitions of annex b and we use information from the appendices where necessary to detail the approach to Eclipse.
For every of these chapters there is a table with the following structure (see Table 1):

Identifier	Keyword	Satisfaction Comment
DO-330-4.1	Qualification Need	Satisfied by satisfying sub-items

Table 1: Tracing Table to DO-330

Table 1 contains an example. The denotation of the content is the following:

- Identifier: This is a unique identifier of the requirement in the DO-330. Since the DO-330 has no sufficient detailed requirement identifiers, we use the DO-330 section and structuring numbers, sometimes followed by some postfix as an additional identifier. To identify the requirements in other documents, we use the prefix DO-330- for all requirements.
- Keyword: The keyword illustrate the identifier.
- Satisfaction Comment: This comment explains how the requirement is satisfied by the Eclipse tool qualification approach. The comment explains the satisfaction of the corresponding requirement, with references to the evidences.

The referenced documents contain the opposite link to the tables to motivate the description in the form of "satisfies: DO-330-4.1".

Tracing to Tool Qualification Planning Process Section

The tool qualification planning is done using a tool chain analysis model with an error-impact analysis, similar to the one proposed by [ISO26262] (see part 8, chapter 11) and the [DO-330] (see FAQ D.2). From this analysis the TORs are determined.

Identifier	Keyword	Satisfaction Comment
DO-330-4.1	Qualification Need	Satisfied by satisfying sub-items
DO-330-4.1.a	Identification	The identification (plugin/product name) of Eclipse products and plugins is reused
DO-330-4.1.b	Intended Use	Is done in the TORs model for the main plugin of the tool model (see section 4.3.1 in [TDP])
DO-330-4.1.c	Qualification Need	See Tool-Analysis part in the model (see section 4.2.2 and 4.2.3 in [TDP])
DO-330-4.1.d	TQLs	See Determination in section 4.2.4 in [TDP]
DO-330-4.1.e	Stakeholders	See “Provider” in MANIFEST.MF and “Validator” in project model in section 4.1.1 of [TDP] and the Validator in the verification data model (see section 4.2.2 in [TVP])
DO-330-4.1.f	Tool Environment Definition	The Environment is defined using the TORContext requirements, see section 4.3.1.4 in [TDP]
DO-330-4.2	Life cycle	See sections 5 and 5.1 in [TDP]
DO-330-4.2.a	Planning process	See section 5.1.1 in [TDP]
DO-330-4.2.b	Development process	See section 5.1.2 in [TDP]
DO-330-4.2.c	Integral process	See section 5.1.3 in [TDP]
DO-330-4.2.1	Transition criteria	See section 5.3 in [TDP]
DO-330-4.2.2	Process input criteria	See section 5.3 in [TDP]
DO-330-4.3	Tool planning objectives	See subsections
DO-330-4.3.a	Life cycle processes	See section 5.1 in [TDP]
DO-330-4.3.b	Tool life cycle: Integration of processes	See sections 5.2 and 5.3 in [TDP]
DO-330-4.3.c	Tool development environments for cycle	See section 6 in [TDP]
DO-330-4.3.d	Additional considerations	See Sections 3.2, 3.3 and 3.4 in [TDP] and additional checks in Table 9.
DO-330-4.3.e	Tool development standards	Standards are defined in section 4 of [TDP]
DO-330-4.3.f	Section 10 compliance of plans	Is checked in Table 8.
DO-330-4.3.g	Development of plans	See extension section 3.1 in [TDP]
DO-330-4.4	Planning activities	See subsections
DO-330-4.4.a	Planning documents	See subsections
DO-330-4.4.a.1	Qualification plan	See section 4.2 in [TDP]
DO-330-4.4.a.2	Development plan	See [TDP], especially sections 5 and 6 and Table 8.
DO-330-4.4.a.3	Verification plan	See [TVP] and Table 8.
DO-330-4.4.a.4	CM plan	Is checked in Table 8.

DO-330-4.4.a.5	QA plan	Is checked in Table 8.
DO-330-4.4.b	Development standards	See section 4 of [TDP]
DO-330-4.4.c	Verification environment(s)	See section 6: Verification Environment in [TVP]
DO-330-4.4.d	Review planning against DO-330	DO-330 review done in this document
DO-330-4.4.e	Assessment of helper tools	See determination of confidence level in [TDP], section 4.2.3

Table 2: Tracing Table to Tool Qualification Planning Process

8.2 Tracing to Tool Development Life Cycle and Process Section

Identifier	Keyword	Satisfaction Comment
DO-330-5.1	TOR Definition Process	See subsections
DO-330-5.1.1	TOR Objectives	TORs are modeled, see section 4.3.1 in [TDP]
DO-330-5.1.2	TOR Activities	See subsections and section 5.1.2 in [TDP]
DO-330-5.1.2.a	Conformance to 10.3.1	See 10.3.1
DO-330-5.1.2.b	TOR verification & consistency	TOR functions and outputs are verified against their implementing TRs, see their ControlState, as described in section 4.3.1.2 and 4.3.1.3 in [TDP].
DO-330-5.1.2.c	TOR details	Is done in the ToolAnalysis, see section 4.2, especially 4.2.2 in [TDP]
DO-330-5.2	Tool Development Process	See subsections
DO-330-5.2.1	TR Process	See subsections
DO-330-5.2.1.1	TR Objectives	See subsections
DO-330-5.2.1.1.a	TR Development	See TR model in 4.3.2 in [TDP]
DO-330-5.2.1.1.b	Derived TRs definition	See definition of derived TR in section 4.3.2.2 in [TDP]
DO-330-5.2.1.2	TR Activities	See subsections and section 5.1.2 in [TDP]
DO-330-5.2.1.2.a	TR satisfiability	TRs are modeled (see section 4.3.2 in in [TDP]) and refined by test cases (see 4.1.2 in [TVP]) and verified, hence they are verifiable.
DO-330-5.2.1.2.b	Functional and Interface	See TRFunctional (section 4.3.2.5) and TRInterface (section 4.3.2.7) in [TDP]
DO-330-5.2.1.2.c	Conformance to process in TDP	Requirements modeling process is described in section 6.1 of [TDP], other creation possibilities of the model are not possible
DO-330-5.2.1.2.d	Conformance to standard in TDP	TR verification (see section 7.1 in [TVP]) ensures the conformance to the plan
DO-330-5.2.1.2.e	TR verifiability and consistency	TRs are verified against their implementing LLRs; see their ControlState, as described in section 4.3.2.2. in [TDP] and verified in 7.3.2.2 of [TVP]
DO-330-5.2.1.2.f	Failure Modes	Failure Modes shall be modeled as special operation modes and error messages, see sections 4.3.2.4 and 4.3.2.8 in [TDP]
DO-330-5.2.1.2.g	TR tracing to TORs	This is satisfied due to the definition of derived TR in section 4.3.2.2 in [TDP]
DO-330-5.2.1.2.h	Derived TR compliance	Is ensured with a special purpose review described in section 7.3.2.3 in [TVP]
DO-330-5.2.1.2.i	Instructions, error messages constraints	See the model elements TRUserInstruction (4.3.2.3), TRExpecedErrorMessage (4.3.2.8) and TROther (4.3.2.11) in [TDP]
DO-330-5.2.1.2.j	Identify unused tools functions	This is done using the tool analysis model (section 4.2 in [TDP]) that shows which functions are used and unused
DO-330-5.2.1.2.k	Detail of TRs	TRs are implemented by LLRs and verified by test cases, see section 4.3.2.2 in [TDP]
DO-330-5.2.2	Tool Design Process	See subsections
DO-330-5.2.2.1	Design Objectives	See subsections
DO-330-5.2.2.1.a	Architecture Development	See architecture model in section 4.4.1 in [TDP]

DO-330-5.2.2.1.b	LLR Development	See LLR model in section 4.4.2 in [TDP]
DO-330-5.2.2.1.c	Derived LLRs	See section 4.3.2.2 in [TDP]
DO-330-5.2.2.2	Design Activities	See subsections and section 5.1.2 in [TDP]
DO-330-5.2.2.2.a	Define Architecture	See section 4.4.1 in [TDP]
DO-330-5.2.2.2.b	Protection for multi-function tools	Is part of the eclipse plugin architecture concept using exported and imported packages, see 4.4.1.4 and 4.4.1.5 in [TDP]
DO-330-5.2.2.2.c	TRs <->LLRs, derived LLRs	See tracing in the LLR model and definition of derived LLRs in section 4.4.2.2 of [TDP]
DO-330-5.2.2.2.d	Derived LLR compliance	Is ensured with a special purpose review described in section 7.3.2.3 in [TVP]
DO-330-5.2.2.2.e	Standard compliance	The compliance to the standards is enforced by using the model described in section 4 in [TDP]. This is verified in section 7.1 of [TVP]
DO-330-5.2.2.2.f	Consistency and verifiability of design	Design (Architecture + LLRs) is traced against their implementing code, and verified; see their ControlState, as described in section 4.3.1.2. in [TDP] and 7.3.2.2 of [TVP]
DO-330-5.2.2.2.g	Interfaces to external tools	Shall be modeled using TRInterface as described in section 4.3.2.7 in [TDP]
DO-330-5.2.3	Tool Coding Process	See subsections
DO-330-5.2.3.1	Objectives	See subsection
DO-330-5.2.3.1.a	Source code from LLRs	See section 6.1 in [TDP]
DO-330-5.2.3.2	Activities	See subsections
DO-330-5.2.3.2.a	Conformance to LLRs and architecture	See section 6.1 in [TDP] and 7.3.2.2 in [TVP]
DO-330-5.2.3.2.b	Conformance to Standards	See section 6.1 in [TDP]
DO-330-5.2.3.2.c	Tracing Code <-> LLRs	Is achieved by the definition of LLRs, section 4.4.2.2 in [TDP] and the development requirements to define the LLRs, see section 6.1 in [TDP]
DO-330-5.2.3.2.d	Detection of inadequate or incorrect inputs	Can be done from Eclipse, see section 6.1 of [TDP]
DO-330-5.2.4	Tool Integration Process	See subsections
DO-330-5.2.4.1	Integration Objectives	See subsections
DO-330-5.2.4.1.a	Executable Generation	Eclipse is used to execute the tool, see section 6.1 in [TDP]
DO-330-5.2.4.1.b	Executable Verification	Eclipse is used to verify the tool, see section 5 in [TVP]
DO-330-5.2.4.2	Integration Activities	See subsections
DO-330-5.2.4.2.a	Compilation	Eclipse is used to compile, see section 6.1 in [TDP]
DO-330-5.2.4.2.b	Detection of inadequate or incorrect inputs	Can be done from Eclipse, see section 6.1 of [TDP]
DO-330-5.2.4.2.c	Installation in verification environment	Not applicable, since development and verification environment are identical (Eclipse)
DO-330-5.2.5	Development Process Traceability	See subsections
DO-330-5.2.5.a	Trace Data TOR<->TR	See subsections
DO-330-5.2.5.a.1	TOR complete	Is modeled in in section 4.3.1.2 of [TDP] and verified by static check in section 7.1.2 in [TVP]

DO-330-5.2.5.a.2	Identify derived TRs	Derived TRs are identified as described in section 4.3.2.2 in [TDP]
DO-330-5.2.5.b	Trace Data TR<->LLR	See subsections
DO-330-5.2.5.b.1	TR complete	Is modeled in in section 4.3.2.2 of [TDP] and verified by static check in section 7.1.7 in [TVP]
DO-330-5.2.5.b.2	Identify derived LLRs, design decisions	Derived LLRs are identified as described in section 4.4.2.2 in [TDP]
DO-330-5.2.5.c	Trace Data LLR <-> Code	See subsections
DO-330-5.2.5.c.1	No undocumented functions	See section 5.1.2 of [TDP]
DO-330-5.2.5.c.2	LLRs completely implemented	See section 5.1.2 of [TDP]
DO-330-5.3	Tool Operational Integration Process	See subsections
DO-330-5.3.1	Objectives	See subsection
DO-330-5.3.1.a	Exe installation	Se the instruction in the user manual, described in section 4.3.2.3. Installation is required for qualification.
DO-330-5.3.2	Activities	See subsections
DO-330-5.3.2.a	Installation	Installation to qualify it, see section 5.1.6 in [TDP]
DO-330-5.3.2.b	Error Detection	Execution of qualification kit, see section 5.1.6 in [TDP]
DO-330-5.3.2.c	Installation Report	Is the qualification report, see section 5.1.6 in [TDP]

Table 3: Tracing Table to Tool Development and Life Cycle Process

8.3 Tracing to Tool Verification Process Section

Identifier	Keyword	Satisfaction Comment
DO-330-6.1	Tool Verification Process	See subsections
DO-330-6.1.1	Purpose of Tool Verification	See subsections
DO-330-6.1.1.a	Implementation of TORs	Included in TOR verifiability check in section 7.1.3 in [TVP].
DO-330-6.1.1.b	Implementation of TRs	Checked by Implementation of TRs in section 7.1.8 in [TVP].
DO-330-6.1.1.c	Source implements requirements	Checked Source Code Tracing in section 7.1.9 in [TVP].
DO-330-6.1.1.d	Executable verification	Is done using dynamic methods (section 7.2 in [TVP]), including coverage measurement, see section 7.2.1 in [TVP].
DO-330-6.1.2	Overview of Tool Verification Activities	See sections 7.1.4, 7.1.5, 7.1.6, 7.1.7 and 7.3.2.1 in [TVP] and subsections.
DO-330-6.1.2.a	Tracing	Tracing is modeled and constraints can be checked model-based.
DO-330-6.1.2.a.1	Traceability TR to TOR	Included in TOR verifiability check in section 7.1.3 in [TVP].
DO-330-6.1.2.a.2	Traceability LLR to TR	Checked by Implementation of TRs in section 7.1.8 in [TVP].
DO-330-6.1.2.a.3	Traceability Tool Source Code to LLR	Checked Source Code Tracing in section 7.1.9 in [TVP].
DO-330-6.1.2.b	Requirements based testing	Is ensured by the Test case tracing check in section 7.1.10 in [TVP].
DO-330-6.1.2.c	Test implementation	Is ensured by the Test case tracing check in section 7.1.10 in [TVP].
DO-330-6.1.2.d	Other verification methods	Every requirements has a verification data element, however the assigned Verification methods (see section 6.2.3. in [TVP]) can be user defined, such that a flexible extension with user defined methods is provided.
DO-330-6.1.2.e	Error reporting	Occurred errors are added to the bug tracking system presented in section 7 and 4.7.2 in [TDP].
DO-330-6.1.3	Reviews and Analyses	See subsections.
DO-330-6.1.3.1	TRs	See subsections.
DO-330-6.1.3.1.a	Compliance with TORs	Ensured from the correct refinement reviews (section 7.3.2.2 in [TVP]) and the derived requirements checks in section 7.3.2.3 in [TVP].
DO-330-6.1.3.1.b	Accuracy and consistency	TRs are implemented by LLRs and verified by test cases, see correct test review in section 7.3.2.4 in [TVP].
DO-330-6.1.3.1.c	Compatibility with environment	Is considered in the qualification plan review as described in section 5.
DO-330-6.1.3.1.d	Failure modes and errors	Since failure modes and errors are TRs, they will be reviewed as part of the correct refinement review in section 7.3.2.2 in [TVP].
DO-330-6.1.3.1.e	User Information	Since user instructions (TRUserInstruction) and expected error messages

		(TRExpecedErrorMessage) and assumptions (TORAssumption) are special requirements, they will be reviewed as part of the correct refinement review in section 7.3.2.2 in [TVP].
DO-330-6.1.3.1.f	Verifiability	Is reviewed by correct test review in section 7.3.2.4 in [TVP].
DO-330-6.1.3.1.g	Conformance to standards	Is ensured by modeling requirements and the transition criteria checking in 7.1.11 in [TVP].
DO-330-6.1.3.1.h	Traceability	Is ensured by modeling requirements and the transition criteria checking in 7.1.11 in [TVP].
DO-330-6.1.3.1.i	Algorithm aspects	Are considered in the correct refinement review in section 7.3.2.2 in [TVP].
DO-330-6.1.3.2	LLRs	See subsections
DO-330-6.1.3.2.a	Compliance with the TRs	Ensured from the correct refinement reviews (section 7.3.2.2 in [TVP]) and the derived requirements checks in section 7.3.2.3 in [TVP].
DO-330-6.1.3.2.b	Accuracy and consistency	If the LLRs are implemented by exactly one Code, they are accurate and cannot contradict. Correctness is reviewed in section 7.3.2.2 in [TVP].
DO-330-6.1.3.2.c	Verifiability	Is reviewed by correct test review in section 7.3.2.4 in [TVP].
DO-330-6.1.3.2.d	Conformance to standards	Is ensured by modeling requirements and the transition criteria checking in 7.1.11 in [TVP].
DO-330-6.1.3.2.e	Traceability	Is ensured by modeling requirements and the transition criteria checking in 7.1.11 in [TVP].
DO-330-6.1.3.2.f	Algorithm aspects	Are considered in the correct refinement review in section 7.3.2.2 in [TVP].
DO-330-6.1.3.3	Tool Architecture	See subsections.
DO-330-6.1.3.3.a	Compatibility with TRs	Tool Architecture requirements are derived requirements and hence covered by the derived requirements compliance review in section 7.3.2.3 in [TVP].
DO-330-6.1.3.3.b	Consistency	Eclipse Architectures (see section 4.4.1 in [TDP]) are consistent by construction with respect to data and control flow.
DO-330-6.1.3.3.c	Conformance to standards	Is ensured by modeling requirements and the transition criteria checking in 7.1.11 in [TVP].
DO-330-6.1.3.3.d	Protection	The Eclipse (OSGI mechanism) ensures that only exported packages (see section 4.4.1.5 in [TDP]) can be used. Others are protected.
DO-330-6.1.3.3.e	External components interface	Shall be modeled using TRInterface as described in section 4.3.2.7 in [TDP].
DO-330-6.1.3.4	Source Code	See subsections.
DO-330-6.1.3.4.a	Compliance with the LLRs	Correct Code review in section 7.3.2.5 in [TVP].
DO-330-6.1.3.4.b	Compliance with the Tool Architecture	Since the tool architecture is based on code model (see section 6.1. in [TDP]) it complies per construction to the code.
DO-330-6.1.3.4.c	Verifiability	Source code is verified using tests (see section 7.2 in [TVP]), non-verifiable part would fail to pass or require an detailed explanation.
DO-330-6.1.3.4.d	Conformance to	Is ensured by modeling requirements and the

	standards	transition criteria checking in 7.1.11 in [TVP].
DO-330-6.1.3.4.e	Traceability	Is ensured by modeling requirements and the transition criteria checking in 7.1.11 in [TVP].
DO-330-6.1.3.4.f	Accuracy and consistency	Partly done by code review (see section 7.3.2.5 in [TVP]) and partly done in Eclipse, see section 6.1 in [TDP].
DO-330-6.1.3.5	Integration Output	See subsections.
DO-330-6.1.3.5.a	Ensure that outputs are complete and correct	See the following items DO-330-6.1.3.5.1 to .3.
DO-330-6.1.3.5.1	Compiler warnings	Are indicated from Eclipse in the editors, see section 6.1 in [TDP].
DO-330-6.1.3.5.2	Missing components	Are identified during build process, see section 6.2 in [TDP].
DO-330-6.1.3.5.3	Incorrect interfaces	Are indicated from Eclipse in the editors, see section 6.1 in [TDP].
DO-330-6.1.4	Tool Testing	See subsections.
DO-330-6.1.4.1	Tool testing objectives	See subsections.
DO-330-6.1.4.1.a	Executable complies with TRs	Every TR has Test cases, see section 4.3.2.2 in [TDP].
DO-330-6.1.4.1.b	Executable is robust with TRs	Every TR has robustness requirements that have test cases, see section 4.3.2.2 in [TDP] and every robustness requirement belong to TRs, see section 4.3.2.9 and verified by transition criteria 5.3.2 in [TDP].
DO-330-6.1.4.1.c	Executable complies with LLRs	LLRs that are testable; see section 4.4.2.2 in [TDP].
DO-330-6.1.4.1.d	Executable is robust with LLRs	Every LLR has assigned TRs that have robustness TRs that have refined LLRs see section 4.3.2.2 in [TDP], hence the robustness LLRs can be inferred from the model and can be tested, see section 4.4.2.2 and verified by transition criteria 5.3.2 in [TDP].
DO-330-6.1.4.2	Tool Testing Activities	See subsections.
DO-330-6.1.4.2.a	Development of test cases and procedures	See subsections and section 6.1.4 in [TVP].
DO-330-6.1.4.2.a.1	Requirements-based tests	Either LLRs or TR are assigned to tests, see section 6.1.2 in [TVP].
DO-330-6.1.4.2.a.2	Test procedures for test cases	Test cases have references to implementation elements (see section 6.1.2 in [TVP]), details for test procedures are in section in 7.2 in [TVP].
DO-330-6.1.4.2.a.3	Test cases trace to procedures	Test procedure and implementations are traceable to test cases; see section 6.1.2 in [TVP].
DO-330-6.1.4.2.b	Normal range tests	See subsections.
DO-330-6.1.4.2.b.1	Valid and boundary numbers	supported methods, see section 6.1.4 in [TVP]
DO-330-6.1.4.2.b.2	Normal transitions	supported method, see section 6.1.4 in [TVP]
DO-330-6.1.4.2.b.3	Logic combinations	supported method, see section 6.1.4 in [TVP]
DO-330-6.1.4.2.c	Robustness tests	Robustness tests are test of robustness requirements that have to be present for, see section 4.3.2.9 of [TDP]

DO-330-6.1.4.2.d	Additional robustness tests	See subsections
DO-330-6.1.4.2.d.1	Abnormal inputs	See TRRobustness in section 4.3.2.9 in [TDP]
DO-330-6.1.4.2.d.2	Abnormal behavior detection	See TRRobustness in section 4.3.2.9 in [TDP]
DO-330-6.1.4.2.d.3	Prevention of invalid output	See TRRobustness in section 4.3.2.9 in [TDP]
DO-330-6.1.4.2.e	Requirements coverage	Is ensured by transition criteria, see section 5.3.3 in [TDP]
DO-330-6.1.4.3	Analysis of requirements-based Testing	See subsections
DO-330-6.1.4.3.1	Objectives	Objectives are satisfied by the following analyses and resolutions
DO-330-6.1.4.3.2	Analysis	See subsections
DO-330-6.1.4.3.2.a	Structural Coverage	100% Coverage is checked in transition criteria for qualification release, see section 5.3.4 in [TDP]; exceptions can be explained manually.
DO-330-6.1.4.3.2.b	External components	External components are embedded using wrappers, similar to COTS tools, see section 3.4 in [TDP]
DO-330-6.1.4.3.2.c	Data & Control Coupling	Is done using the Eclipse classloader that has to be qualified, see section 3
DO-330-6.1.4.3.3	Resolution	As described in the code voverage section in 7.2.1 in [TVP]
DO-330-6.1.4.4	Reviews and analyses of test cases, procedures and results	See subsections
DO-330-6.1.4.4.a	Test cases	Is done in the correct tests review in section 7.3.2.4 in [TVP]
DO-330-6.1.4.4.b	Test procedures	Is done in the correct tests review in section 7.3.2.4 in [TVP]
DO-330-6.1.4.4.c	Test results	Is done in the correct tests review in section 7.3.2.4 in [TVP]
DO-330-6.1.5	Tool Verification Process Traceability	Trace is done in the DO-330 model, see subsections
DO-330-6.1.5.a	TORs, TRs<->test cases LLRs<->test cases	See test part of DO-330 model in section 6.1
DO-330-6.1.5.b	Trace data test cases<-> test procedures	See test part of DO-330 model in section 6.1
DO-330-6.1.5.c	Trace data test procedures<->test results	See test part of DO-330 model in section 6.1
DO-330-6.2	Tool operational verification and validation process	See subsections
DO-330-6.2.1	Objectives	See subsections
DO-330-6.2.1.a	TORs are OK	TORs are qualified according to qualification plan (see section 5).
DO-330-6.2.1.b	Tool complies with TORs	The qualification plan is executed and ensured this and documents this within the tool qualification report, see section 6.
DO-330-6.2.1.aa	TORs fit to PSAC	PSAC is generated from the analysis model and therefore TORs fit to the PSAC, see section

		4.2.2 in [TDP]
DO-330-6.2.1.bb	Tool fits into process	Verification is also done according to the analysis model, see section 4.2.2 in [TDP]
DO-330-6.2.2	Activities	TORs reviewed (see status of TORS in section 4.3.1.2 in [TDP]) and the activities in the following subsections.
DO-330-6.2.2.a	TOR verification	See subsections
DO-330-6.2.2.a.1	Requirements are OK	TORs reviewed, see status of TORS in section 4.3.1.2 in [TDP]
DO-330-6.2.2.a.2	Environment compatibility	See qualification plan in section 5
DO-330-6.2.2.a.3	Robustness requirements defined	See TRRobustness in section 4.3.2.9 in [TDP]
DO-330-6.2.2.b	Identification and demonstration of relevant processes	Done in the analysis part, see section 4.2.2 in [TDP]
DO-330-6.2.2.c	V&V in operational requirement	See subsections
DO-330-6.2.2.c.1	TORs tests	Are modeled in the DO-330 model TORS and Testcases are directly linked, see section 4.3.1.2 in [TDP], and section 6.1.2 in [TVP]
DO-330-6.2.2.c.2	Execution of tests	Is done using qualification and reported in tool qualification report, see section 6
DO-330-6.2.2.c.3	Analysis of test results	Is part of tool qualification plan, see section 6
DO-330-6.2.2.c.4	TORs validation	TORs are under version control (section 6.3.1.2 in [TDP]) and are reviewed for consistency. Afterwards they are executed within the qualification kit and reported in the qualification reported, see section 6

Table 4: Tracing Table to Tool Verification Process

8.4 Tracing to Tool Configuration Management Process Section

Identifier	Keyword	Satisfaction Comment
DO-330-7.1	process objectives	See subsections
DO-330-7.1.a	Configuration item	Objective is detailed in DO-330-7.2.1
DO-330-7.1.b	Baselines	Objective is detailed in DO-330-7.2.2
DO-330-7.1.c	Problem reporting process	Objective is detailed in DO-330-7.2.3
DO-330-7.1.d	Change control	Objective is detailed in DO-330-7.2.4
DO-330-7.1.e	Change review	Objective is detailed in DO-330-7.2.5
DO-330-7.1.f	Status accounting	Objective is detailed in DO-330-7.2.6
DO-330-7.1.g	Archival and retrieval	Objective is detailed in DO-330-7.2.7
DO-330-7.1.h	Other tools	See section 4.6.1 in [TDP]
DO-330-7.2	TCL process activities	See subsections
DO-330-7.2.1	Configuration identification	See subsections
DO-330-7.2.1.a	Life cycle data	See section 4.6.1 in [TDP]
DO-330-7.2.1.b	Item selection	See section 4.6.1 in [TDP]
DO-330-7.2.1.c	Prior implementation	See section 4.6.1 in [TDP] that defined the items generally, especially before the implementation
DO-330-7.2.1.d	Prior to references to it	Nightly build ensures that, see section 7.4 in [TVP]
DO-330-7.2.2	Baselines and traceability	See subsections
DO-330-7.2.2.a	Configuration items	See section 4.6.2 in [TDP]
DO-330-7.2.2.b	Tool product baseline	See section 4.6.2 in [TDP]
DO-330-7.2.2.c	Change protection	Annotated tags protect baselines against changes, see section 6.3.2 in [TDP]
DO-330-7.2.2.d	Change control activities for derived baselines	See section 4.6.2 in [TDP]
DO-330-7.2.2.e	Traceability of baselines	See section 4.6.2 and 4.6.3 in [TDP]
DO-330-7.2.2.f	Traceability of configuration item	See section 4.6.2 and 4.6.3 in [TDP]
DO-330-7.2.2.g	Output/process traceability	See section 4.6.2 and 4.6.3 in [TDP]
DO-330-7.2.3	Problem reporting, tracking and corrective action	See subsections
DO-330-7.2.3.a	Tool problem report	See section 4.6.4 in [TDP] and nightly build in section 7.4 in [TVP]
DO-330-7.2.3.b	Problem Configuration	See section 4.6.4 in [TDP] and nightly build in section 7.4 in [TVP]
DO-330-7.2.3.c	Invoke Change Control	Reporting to Bugzilla ensures this see section 4.6.6 and 7.4 in [TVP]
DO-330-7.2.4	Change control	See subsections
DO-330-7.2.4.a	Integrity	See section 4.6.5 and 6.3.5 in [TDP]
DO-330-7.2.4.b	Change to a configuration item	See section 4.6.5 and 6.3.5 in [TDP]
DO-330-7.2.4.c	Changes to baselines and configuration items	See section 4.6.5 and 6.3.5 in [TDP]
DO-330-7.2.4.d	Consistent changes	Is ensured from the transition criteria in section 5.3 in [TDP]

DO-330-7.2.4.e	Update tool life cycle data	Is ensured from the transition criteria in section 5.3 in [TDP] which also includes tool life cycle data
DO-330-7.2.5	Change review	See subsections
DO-330-7.2.5.a	Identification of configuration items	See section 4.6.6 and 4.6.7 in [TDP]
DO-330-7.2.5.b	Impact of change on TORs	See section 4.6.6 and 4.6.7 in [TDP]
DO-330-7.2.5.c	Access content	See section 4.6.7 in [TDP]
DO-330-7.2.5.d	Feedback	See section 4.6.7 in [TDP]
DO-330-7.2.5.e	Software process feedback	See section 4.6.7 in [TDP]
DO-330-7.2.6	Configuration status accounting	See section 4.6.8 in [TDP]
DO-330-7.2.6.a	Configuration Item Reports	See section 4.6.8 in [TDP]
DO-330-7.2.6.b	Maintained Data	See section 4.6.8 in [TDP]
DO-330-7.2.7	Archive, retrieval and release	See subsections
DO-330-7.2.7.a	Retrievability	See section 4.6.9 in [TDP]
DO-330-7.2.7.b	Integrity Procedures	See subsections
DO-330-7.2.7.b.1	No Unauthorized Changes	See section 4.6.9 in [TDP]
DO-330-7.2.7.b.2	Selecting Storage	Distributed version control as done by Git minimizes the risk and the nightly build in section 7.4 in [TVP] checks every night.
DO-330-7.2.7.b.3	Preventing Loss & Corruption	See section 4.6.9 in [TDP]
DO-330-7.2.7.b.4	Duplication	See section 4.6.9 in [TDP]
DO-330-7.2.7.c	Duplication verification	Nightly build, see section 7.4 in [TVP] duplicates the repository and verifies it.
DO-330-7.2.7.d	Configuration Items	Items are identified (Git id) and pushed before they are used from others according to the Git usage; see section 4.6.9 in [TDP].
DO-330-7.2.7.e	Data Retention	See section 4.6.10 in [TDP]
DO-330-7.3	Data control categories	See subsections
DO-330-7.3.a	Control category 1 (CC1)	See section 4.6 in [TDP]
DO-330-7.3.b	Control category 2 (CC2)	See section 4.6 in [TDP]
DO-330-7.4	Tool life cycle environment	See subsections
DO-330-7.4.a	Configuration identification	All tools to produce the executable are under version control, see section 4.6.1 in [TDP]
DO-330-7.4.b	TCM process for controlling the tools	See DevelopmentTools in section 4.6.1 in [TDP]
DO-330-7.Tab1.1	Configuration Identification	See section 4.6.1 in [TDP]
DO-330-7.Tab1.2	Baselines	See section 4.6.2 in [TDP]
DO-330-7.Tab1.3	Traceability	See section 4.6.3 in [TDP]
DO-330-7.Tab1.4	Problem Reporting	See section 4.6.4 in [TDP]
DO-330-7.Tab1.5	Change Control: Integrity and Identification	See section 4.6.5 in [TDP]
DO-330-7.Tab1.6	Change Control: Tracking	See section 4.6.6 in [TDP]
DO-330-7.Tab1.7	Change Review	See section 4.6.7 in [TDP]
DO-330-7.Tab1.8	Configuration Status	See section 4.6.8 in [TDP]

	Accounting	
DO-330-7.Tab1.9	Retrieval	See section 4.6.9 in [TDP]
DO-330-7.Tab1.10	Protection Against Unauthorized Changes	See section 4.6.10 in [TDP]
DO-330-7.Tab1.11	Media Selection, Refreshing, Duplication	See section 4.6.10 in [TDP]
DO-330-7.Tab1.12	Release	See section 4.6.10 in [TDP]
DO-330-7.Tab1.13	Data Retention	See section 4.6.10 in [TDP]

Table 5: Tracing Table to Configuration Management Process

8.5 Tracing to Tool Quality Assurance Process Section

Identifier	Keyword	Satisfaction Comment
DO-330-8.1	process objectives	See subsections
DO-330-8.1.a	Review of TDP	See section 3.1 in [TDP]
DO-330-8.1.b	Process compliance	See section 5.1.5 in [TDP]
DO-330-8.1.c	Satisfaction of transition criteria	See section 5.1.5 in [TDP]
DO-330-8.1.d	Conduct and conformity review	See DO-330-8.3 and subsections
DO-330-8.2	TQA process activities	See subsections
DO-330-8.2.a	Independency of QA	Qualification stage is automatically computed, hence sufficiently independent, see section 5.3 in [TDP]
DO-330-8.2.b	Tool plans	See section 7.1 in [TVP]
DO-330-8.2.c	Compliance of processes to standards	Is ensured by transition criteria see section 5.3 in [TDP], verified and documented in the tool quality report as described in section 5.1.5 in TDP
DO-330-8.2.d	TQA Audit	One audit suffices to create the tool quality report as described in section 5.1.5, even if it recommended to verify the development environment installation earlier.
DO-330-8.2.d.1	Tool Plans	The tool plans are contained in the DO-330 model, see section 4 of [TDP]
DO-330-8.2.d.2	Deviations	Are detected during the transition criteria check see section 5.3 of [TDP]
DO-330-8.2.d.3	Recording	Deviations are recorded in the tool quality report as described in section 5.1.5 of [TDP]
DO-330-8.2.d.4	Tool Environment	This is also documented in the tool quality report as described in section 5.1.5 of [TDP]
DO-330-8.2.d.5	Problem Reporting	Problem reporting is in the DO-330 model; see section 4.7 of [TDP] and verified in the tool quality report in section 5.1.5 of [TDP].
DO-330-8.2.e	Transition criteria	Transition criteria verification is part of the tool quality report, see section 5.1.5 of [TDP]
DO-330-8.2.f	Control categories conformance	Is ensured from Eclipse, see section 6.3 in [TDP]
DO-330-8.2.g	Tool conformity review	See entry DO-330-8.3 in this table
DO-330-8.2.h	QA data and review	See data table DO-330-10.1.4 and the tool conformity review, entry DO-330-8.3 in this table
DO-330-8.2.i	Tool quality reports of required plugins	See the checklist for the tool quality report in section 5.1.5 in [TDP]
DO-330-8.3	Tool conformity review	See subsections
DO-330-8.3.a	Tool life cycle data complete	The tool life cycle data is contained in the DO-330 model, which verification is part of the tool quality report in section 5.1.5 of [TDP]
DO-330-8.3.b	Evidence for plan conformant production of tool life cycle data	The plans in section 4 of [TDP] describe the production of the model which is part of qualification build see section 6.2 and verified in section 5.3 of [TDP]
DO-330-8.3.c	Tool problem reports	The required mapping from known problems to

	evaluation	potential errors requires to analyze the problems is described in sections 4.7.2 and 6.4 of [TDP]
DO-330-8.3.d	Tool requirement deviations	The open problem reports are contained in the tool quality report, see section 5.1.5 of [TDP]
DO-330-8.3.e	Reproducibility	Is ensured by the nightly build which is done from the archived version, see section 7.4 in [TVP], the report/logfile verification is part of the quality report in section 5.1.5 of [TDP]
DO-330-8.3.f	Considerations of previous known issues	The known issues of previous versions are considered in the tool quality report, see section 5.1.5 in [TDP]
DO-330-8.3.g	Previously developed certified versions	Due to the high degree of automation, currently there is no special support for changes between baselines and therefore this is not applicable.

Table 6: Tracing Table to Tool Quality Assurance Process

8.6 Tracing to Tool Qualification Liaison Process Section

Identifier	Keyword	Satisfaction Comment
DO-330-9.0.a	Submitted data	See subsections
DO-330-9.0.a.1	PSAC	See DO-330-10.1.1 in Table 8
DO-330-9.0.a.2	TQP	See DO-330-10.1.2 in Table 8
DO-330-9.0.a.3	TCI	See DO-330-10.1.1 in Table 8
DO-330-9.0.a.4	TAS	See DO-330-10.1.11 in Table 8
DO-330-9.0.a.5	SECI	See DO-330-10.1.17 in Table 8
DO-330-9.0.a.6	SAS	See DO-330-10.1.16 in Table 8
DO-330-9.0.b	PSAC and TQP	See above
DO-330-9.0.c	Life cycle data	Is contained in the DO-330 model, which is part of the qualification kit, see section 6.2 in [TDP]
DO-330-9.0.d	Known Problems	Is contained in the tool qualification report, see sections and in this document
DO-330-9.0.e	SECI	See DO-330-10.1.17 in Table 8

Table 7: Tracing Table to Tool Qualification Liaison Process

8.7 Tracing to Tool Qualification Data Section

Identifier	Keyword	Satisfaction Comment
DO-330-10.1.1	Tool Specific information in PSAC	The information is in the DO-model contained, a document could be generated, see section [TDP] and subsections
DO-330-10.1.1.a	Identification and Use Cases	Modeled in Project and TORs, see sections 4.1.1, 4.2.2 and 4.3.1 in [TDP]
DO-330-10.1.1.b	Details of use in process	The artifacts in the analysis model provide the link to the automated process, see sections 4.2.2.2, 4.2.2.5 and 4.2.2.6 in [TDP]
DO-330-10.1.1.c	Technology maturity	Maturity be inferred from the number of bugs and their affected elements, which are modeled in section 4.7.6 in [TDP]
DO-330-10.1.1.d	Proposed TQL	See TQL in Project model in 4.1.1 and its derivation in section 4.2.4 and 4.2.3 in [TDP]
DO-330-10.1.1.e	Source Code	Code is part of the Qualification Build, see section 6.2 in [TDP].
DO-330-10.1.1.f	Stakeholders and Roles	See “Provider” in MANIFEST.MF and “Validator” in project model in section 4.1.1 of [TDP] and the Validator in the verification data model (see section 4.2.2 in [TVP])
DO-330-10.1.1.g	Process Operational	See Table 3 (5.1 and 5.3) and Table 4 (6.2)
DO-330-10.1.1.h	Tool Operational Environment desc.	See TORContext model in section 4.3.1.4 in [TVP]
DO-330-10.1.1.i	Qualification reuse	Only possible as described in section 3.3 and 3.4 in [TVP]
DO-330-10.1.1.j	Reference to TQP	TQP is generated from the same model, hence it is trivial
DO-330-10.1.2	Tool Qualification Plan	The information is in the DO-model contained, a document could be generated as described in section 5. See subsections
DO-330-10.1.2.a	Identification, configuration	Identification is modeled, see section 4.1.1 in [TDP] and section 5
DO-330-10.1.2.b	Proposed TQL, compliance	See TQL in Project model in 4.1.1 and it’s derivation in section 4.2.4 and 4.2.3 in [TDP] and section 5
DO-330-10.1.2.c	Functional Overview & Architecture	Functional overview is modeled with Function (see 4.2.2.3 and 4.3.1.3 in [TDP]) and their I/O artifacts (see 4.2.2.5 and 4.2.2.6 in [TDP]). Architecture is modeled including references in sections 4.4.1 and 4.4.1.4 in [TDP] and section 5
DO-330-10.1.2.d	Environment(s) description.	See TORContext model in section 4.3.1.4 in [TVP] and section 6 in [TVP] and section 5

DO-330-10.1.2.e	Visibility	See section 5.4 in [TVP] and section 5
DO-330-10.1.2.f	Life cycle & qualification	See section 5 and 5.1.6 in [TDP] and section 5
DO-330-10.1.2.g	Qualification Data	See section 4.2 in [TVP] and section 5
DO-330-10.1.2.h	Additional Consideration	See section 5
DO-330-10.1.2.i	Responsibilities	Every plugin in a tool has a responsible person, called “provider”. This can be an organization or internal responsibilities or both, see section 6.1 in [TDP] and section 5
DO-330-10.1.2.j	Suppliers	Every plugin (also those of suppliers) have the provider information, see section 6.1 in [TDP] and section 5
DO-330-10.1.3	Tool Development Plan	See subsections
DO-330-10.1.3.a	Standards: Requirement, Design, Code	See sections 4.3, 4.4 and 4.5 in [TDP].
DO-330-10.1.3.b	Tool life cycle	See section 5 in [TDP].
DO-330-10.1.3.c	Development Environment	See section 6 in [TDP].
DO-330-10.1.3.c.1	Requirements	See section 6.1 in [TDP].
DO-330-10.1.3.c.2	Design	See section 6.1 in [TDP].
DO-330-10.1.3.c.3	Coding	See section 6.1 in [TDP].
DO-330-10.1.3.c.4	Compilers	See section 6.1 in [TDP].
DO-330-10.1.3.c.5	Operating Systems	See section 6.1 in [TDP].
DO-330-10.1.4	Tool Verification Plan	See subsections
DO-330-10.1.4.a	Description of methods for verification independence	See section 4 in [TVP]
DO-330-10.1.4.b	Verification methods	See Section 7 in [TVP]
DO-330-10.1.4.b.1	Review methods	The reviews are done using Gerrit; see section 6.3.6 in [TDP].
DO-330-10.1.4.b.2	Analysis methods	Traceability and coverage (except code coverage which is part of the [TVP], section 7.2.1) is checked in the transition criteria in section 5.3 of [TDP].
DO-330-10.1.4.b.3	Testing methods	Testing is done using CodeCover, see section 7.2.1 in [TVP]
DO-330-10.1.4.c	Transition criteria	See section 8 in [TVP]
DO-330-10.1.4.d	Protection considerations	See section 9 in [TVP]
DO-330-10.1.4.e	Reverification methods	See section 10 in [TVP]
DO-330-10.1.4.f	Previously developed components	See section 11 in [TVP]
DO-330-10.1.5	Tool Configuration Management Plan	See section 4.6 in [TDP]
DO-330-10.1.5.a	Environment	See section 6.3 in [TDP]
DO-330-10.1.5.b	Activities	See subsections
DO-330-10.1.5.b.1	Configuration identification	See section 4.6.1 in [TDP]
DO-330-10.1.5.b.2	Baseline and traceability	See section 4.6.2 and 4.6.3 in [TDP]
DO-330-10.1.5.b.3	Problem reporting	See section 4.6.4 in [TDP]
DO-330-10.1.5.b.4	Change control	See section 4.6.5 in [TDP]
DO-330-10.1.5.b.5	Change review	See section 4.6.7 in [TDP]

DO-330-10.1.5.b.6	Configuration status accounting	See section 4.6.8 in [TDP]
DO-330-10.1.5.b.7	Archive, retrieval, and release	See section 4.6.9 in [TDP]
DO-330-10.1.5.b.8	Tool life cycle environment controls	See section 4.6.1 in [TDP]
DO-330-10.1.5.b.9	Tool life cycle data controls	The data is in the DO-330 model, which is under configuration control, se section 4.6.1 in [TDP]
DO-330-10.1.5.c	Transition criteria	See section 4.6 in [TDP]
DO-330-10.1.5.d	Tool Configuration Management data	See section 4.6 in [TDP] and DO-330-10.1.10 and DO-330-10.1.11 in this table
DO-330-10.1.5.e	Supplier control	Required plugins must also have a TCM; this is checked by examination of the tool's quality reports in section 5.1.5 in [TDP].
DO-330-10.1.6	Tool quality assurance plan	See subsections
DO-330-10.1.6.a	TQA environment	See section 6.4 in [TDP]
DO-330-10.1.6.b	TQA authority	Automated tools are used widely; see section 5.1.5 in [TDP], hence independent authority is best supported.
DO-330-10.1.6.c	TQA activities	See subsections
DO-330-10.1.6.c.1	TQA methods	Model-based method as explained in section 4.7 in [TDP], supplemented by a manual quality report see 5.1.5 in [TDP]
DO-330-10.1.6.c.2	Problem reporting activities	Transition criteria see section 5.3 in [TDP] enforce activities as described in the DO-330 model section 4.7 [TDP] and the environment, see section 6.4 in [TDP]
DO-330-10.1.6.c.3	Tool conformity review activity	Tool quality report is done according to the checklist in section 5.1.5 of [TDP].
DO-330-10.1.6.d	Transition criteria	See section 4.7 in [TDP].
DO-330-10.1.6.e	Timing	See section 4.7 in [TDP].
DO-330-10.1.6.f	TQA records	The model (see section 4.7 in [TDP]) and the tool quality report (see section 5.1.5 in [TDP]) are the TQA records.
DO-330-10.1.6.g	Supplier oversight	The tool quality report (see section 5.1.5 in [TDP]) reviews the required plugins quality report and documents the results.
DO-330-10.1.7	Tool Requirements Standards	See subsections
DO-330-10.1.7.a	Used methods for developing TRs	TRs are developed within the DO-330 model; see section 4.3.2 in [TDP].
DO-330-10.1.7.b	Used Notations to express requirements	TRs are modeled with different model elements, see section 4.3.2.2.
DO-330-10.1.7.c	Constraints	Only constraint is to fill the DO-330 model as described in section 4.3.2.2 in [TDP].
DO-330-10.1.7.d	Derived TRs analysis method	Is described in section 4.3.2.2 in [TDP].
DO-330-10.1.8	Tool Design Standards	See subsections
DO-330-10.1.8.a	Design Description method(s)	It is described in section 4.4 in [TDP].
DO-330-10.1.8.b	Naming conventions	See section 6.1 in [TDP].

DO-330-10.1.8.c	Constraints	See section 4.4 in [TDP].
DO-330-10.1.8.d	Complexity restrictions	Neo hard restrictions; see section 4.4 in [TDP].
DO-330-10.1.9	Tool Code Standards	See subsections
DO-330-10.1.9.a	Programming language(s)	See section 4.5 in [TDP].
DO-330-10.1.9.b	Tool Source Code presentation standards	See section 6.1 in [TDP].
DO-330-10.1.9.c	Naming conventions	See section 6.1 in [TDP].
DO-330-10.1.9.d	Imposed conditions and constraints	See section 6.1 in [TDP].
DO-330-10.1.9.e	Constraints on used tools	Eclipse is used, see section 6.1 in [TDP]
DO-330-10.1.10	Tool Life Cycle Environment Configuration Index	See subsections
DO-330-10.1.10.a	Identification of the tool development environment	Is done within the Eclipse configuration “Overview” tab in the project configuration, see section 6.1 in [TDP].
DO-330-10.1.10.b	Identification of development tools	Compiler and classpath are also in the projects settings (section 6.1 in [TDP]) Furthermore there is a list of dependencies (imported packages), see section 4.4.1.4 in [TDP].
DO-330-10.1.10.c	Identification of the tool verification environment	Eclipse and CodeCover is used for verification, see section 5 in [TVP]
DO-330-10.1.10.d	Identification of qualified tools	Qualified required tools are identified from QA in the tool quality report; see section 5.1.5 in [TDP].
DO-330-10.1.11	Tool Configuration Index	See subsections
DO-330-10.1.11.a	Tool Product	See section 4.6.1 in [TDP].
DO-330-10.1.11.b	Tool Executable	Executable is part of the product, see also section 4.6.1 in [TDP].
DO-330-10.1.11.c	Source code	Qualification build contains also source code; see section 4.6.1 in [TDP].
DO-330-10.1.11.d	Previously qualified components	Every plugin is under control, especially qualified ones, se section 4.6.1 in [TDP].
DO-330-10.1.11.e	Tool Life Cycle Data	Is in the DO-330 model, which is part of the qualification build; see section 6.2 in [TDP].
DO-330-10.1.11.f	Reference to environment configuration	Environment configuration (MANIFEST. MF, plugin.xml) is part of the qualification build; see section 6.2 in [TDP].
DO-330-10.1.11.g	media	Git has distributed storage, see section 6.3 in [TDP].
DO-330-10.1.11.h	Rebuild	Rebuild can be done using the plugin.xml, see section 6.2 in [TDP], which can also be retrieved, see section 6.3.3 in [TDP].
DO-330-10.1.12	Tool Problem Reports	See subsections
DO-330-10.1.12.a	Identification of configuration item or process	Identification of the plugin is provided in problem reports using test cases from the process in which the tool is applied; see bug and relation to test cases in section 4.7.2 in [TDP].
DO-330-10.1.12.b	configuration item(s) to be	The affected elements of a bug are

	modified	modelled using references to TRs, LLRs and Code. See sections 4.7.2 and 4.7.6 of [TDP].
DO-330-10.1.12.c	Problem description	Is part of the problem reports, see sections 4.7.2 of [TDP].
DO-330-10.1.12.d	Description of the corrective action	Every branch shall be associated to a problem of TOR, see section 6.3.1 of [TDP].
DO-330-10.1.13	Tool Configuration Management Records	CM records are in the qualification build, produced from gitweb, see sections 6.2 and 6.3.4 in [TDP].
DO-330-10.1.14	Tool Quality Assurance Records	Problems and links to mitigations are contained in the DO-330 model, which is part of the qualification build; see section 6.2 in [TDP] .
DO-330-10.1.15	Tool Accomplishment Summary	The tool qualification report demonstrates that the tool accomplishes its requirements in the user's environment. A structure is given in section 6
DO-330-10.1.15.a	Tool Overview	See section 6
DO-330-10.1.15.b	Qualification considerations	See section 6
DO-330-10.1.15.c	Tool life cycle	See section 6
DO-330-10.1.15.d	Tool life cycle data	See section 6
DO-330-10.1.15.e	Additional considerations	See section 6
DO-330-10.1.15.f	Supplier Oversight	See section 6
DO-330-10.1.15.g	Tool identification	See section 6
DO-330-10-1.15.h	Change history	See section 6
DO-330-10.1.15.i	Tool status	See section 6
DO-330-10.1.15.j	Compliance statement	See section 6
DO-330-10.1.16	Tool-Specific Information in Software Accomplishment Summary	See subsections
DO-330-10.1.16.a	Identification of the tool	From the Project information, see section 4.1.1 in [TDP]
DO-330-10.1.16.b	Qualification need	From the tool chain model that shows how the artifacts (see section in [TDP]) are processed from the tools, see section 4.2.2.6
DO-330.10.1.16.c	Reference to Tool Accomplishment Summary	Delivering the qualification reports with the SAS are part of the liaison process, see section 4
DO-330-10.1.16.d	Compliance with plans	Is documented in the qualification report, see section 5.1.5 in [TDP], which is part of the qualification build, see section 6.2 in [TDP]
DO-330-10.1.16.e	Problem Analysis	Is part of the QA process, see section 5.1.5 in[TDP] which results are part of the qualification build, see section 6.2 in [TDP]
DO-330-10.1.16.f	Tool use differences	Is part of the qualification report (TAS), see section 6

DO-330-10.1.16.aa	Identification of the tool	Same as above DO-330-10.1.16.a
DO-330-10.1.16.bb	Qualification need	Same as above DO-330-10.1.16.b
DO-330-10.1.16.cc	Reference to Qualification Data	Same as above DO-330-10.1.16.c
DO-330-10.1.17	Tool-Specific Information in SECI	See subsections
DO-330-10.1.17.a	Identification of the tool (TQL 1,2,3,4)	From the Project information, see section 4.1.1 in [TDP]
DO-330-10.1.17.b	Reference to TCI and TAS (TQL 1,2,3,4)	References to TCI and TAS are part of the liaison process, see section 4
DO-330-10.1.17.aa	Identification of the tool (TQL 5)	As above in DO-330-10.1.17.a
DO-330-10.1.17.bb	Reference to the qualification data (TQL 5)	TAS as above in DO-330-10.1.17.b
DO-330-10.2	Tool Qualification Data	See subsections
DO-330-10.2.1	Tool Requirement	See subsection
DO-330-10.2.1.a	Functions, Features and Modes	Done using TRFunction in section 4.3.2.5 in [TDP] and TROpMode in section 4.3.2.4 in [TDP]
DO-330-10.2.1.b	User instructions, error messages constraints	Done using TRUserInstruction in section 4.3.2.3 in [TDP] and TrExpectedErrorMessage in section 4.3.2.8 in [TDP]
DO-330-10.2.1.c	Customizing	Done using TRCustomizing in section 4.3.2.6 in [TDP]
DO-330-10.2.1.d	Detailed functional Requirements	TRs as modeled as ToolRequirement see section 4.3.2.2 in [TDP] satisfying DO-330-5.2.1.2.k
DO-330-10.2.1.e	Specific operational customizations requirements	Done using TRCustomizing in section 4.3.2.6 in [TDP]
DO-330-10.2.1.f	Failure modes	Done using TROpMode in section 4.3.2.4 in [TDP]
DO-330-10.2.1.g	Abnormal operation	Done using TRRobustness in section 4.3.2.9 in [TDP]
DO-330-10.2.1.h	Interfaces between tools	Done using TRInterface in section 4.3.2.7 in [TDP]
DO-330-10.2.2	Tool Design Description	See subsections
DO-330-10.2.2.a	Architecture Description	The architecture is modeled within Eclipse using the known architecture elements, which hare grouped into ArchitectureRequirement, see section 4.4.1.1 in [TDP]
DO-330-10.2.2.b	Requirements Allocation	The LLRs are mapped to TRs (see section 4.4.2.2 [TDP]). Some of the TRs are Architecture requirements others not, hence the allocation is defined.
DO-330-10.2.2.c	Input/output description	The Eclipse architecture uses packages, modeled as ArchPackage (see section 4.4.1.3 in [TDP]), that can be imported (ArchImportedPackage in section 4.4.1.4 in [TDP]) and exported (ArchExportedPackage in section 4.4.1.5

		in [TDP]).
DO-330-10.2.2.d	Data & Control Flow in Design	Is modelled using ArchExtensions in section 4.4.1.7 in [TDP].
DO-330-10.2.2.e	Scheduling	The Eclipse-based tools are event-oriented and hence scheduling is defined with the properties, see ArchProperty in section 4.4.1.11 in [TDP] and ArchExtension in section 4.4.1.7 in in [TDP].
DO-330-10.2.2.f	Protection	In Eclipse there are unexported packages (see ArchUnexportedPackage in section 4.4.1.6 in in [TDP]) that are protected and the visibility mechanism of the Java implementations, see section 4.4.3.9 in in [TDP].
DO-330-10.2.2.g	Component description, baselines	The Eclipse components are plugins they are always described with version numbers in the Project Information (see section 4.1.1 in [TDP].). This information is used in the architecture (see sections 4.4.1.3, 4.4.1.4 and 4.4.1.5 in [TDP]) to determine the imported required plugins.
DO-330-10.2.2.h	Traceable LLRs	LLRs are modeled traceable to TRs, see section 4.4.2.2 in [TDP].
DO-330-10.2.2.i	Derived LLRs	Are described as LLRs without TRs, see section 4.4.2.2 in [TDP]
DO-330-10.2.3	Tool Source Code	See subsections
DO-330-10.2.3.a	Code	In src directory, see section 6.1 in [TDP]
DO-330-10.2.3.b	Linking instructions	Linking corresponds to the combination of the plugins into a product as described in section 6.2 in [TDP].
DO-330-10.2.3.c	Compiling instructions with tool identification	Tools are identified within the product description see section 6.2 in [TDP]. The Eclipse environment also contains the classpath for building the tool, see section 6.1 in [TDP]
DO-330-10.2.4	Tool Executable Object Code	Product configurations generate executables as described in section 6.2 in [TDP]
DO-330-10.2.5	Tool Verification Cases and Procedures	See subsections
DO-330-10.2.5.a	Review and analysis procedures	Verification methods are modeled as VerificationMethod in section 6.2.3 and described in section 7 in [TVP]. Reviews, see section 7.3.2 in [TVP], are a special case.
DO-330-10.2.5.b	Test cases	Are modeled using the TestCase elements, see section 6.1.2 in [TVP].
DO-330-10.2.5.c	Test procedures	Are modeled using the TestCase elements, see section 6.1.3 in [TVP].
DO-330-10.2.6	Tool Verification Results	See subsections
DO-330-10.2.6.a	Pass/fail indication	Is in the tool VerificationData elements in

		section 6.2.2 in [TVP]
DO-330-10.2.6.b	Verified item's configuration identification	The verification data refers to different verified items (TOR, TR, LLR) that all have an assigned control status, see section 4.1.5 in [TDP]
DO-330-10.2.6.c	Test results including coverage	The verification data contains a field Results of the tests, that also contains the coverage results, see section 6.2.2 in [TVP]
DO-330-10.2.6.d	Report found problems	Failed test cases cause problem reports, see section 7.2 in [TVP]. If other configuration items cannot be reviewed, the status is not changed to reviewed, see section 7.3 in [TVP].
DO-330-10.2.7	Trace Data	See subsections
DO-330-10.2.7.a	TORs <-> TRs	Is in the DO-330 model, see sections 4.3.1.2 and 4.3.2.2 in [TDP]
DO-330-10.2.7.b	TRs <-> LLRs	Is in the DO-330 model, see sections 4.3.2.2 and 4.4.2.2 in [TDP]
DO-330-10.2.7.c	LLRs <-> Source	Is in the DO-330 model, see sections 4.4.2.2 and 4.4.3.1 in [TDP]
DO-330-10.2.7.d	TRs <-> LLRs, and their associated test cases	See above in DO-330-10.2.7.c and section 4.3.2.2 in [TDP]
DO-330-10.2.7.e	Test cases <-> test procedures	Is in the DO-330 model, see sections 6.1.2 and 6.1.3 in [TVP]
DO-330-10.2.7.f	Test procedure <-> test results	Test Procedure has relation to verification data with the test results see sections 6.2.2 and 6.2.3 in [TVP]
DO-330-10.3	Tool Qualification Data produced during operation processes	See subsections
DO-330-10.3.1	TORs	See subsections
DO-330-10.3.1.a	Description of context of used tool	Is modeled by TORContext, see section 4.3.1.4 in [TDP]
DO-330-10.3.1.b	Description of operational environment(s)	Is modeled by TORContext, see section 4.3.1.4 in [TDP]
DO-330-10.3.1.c	Description of input files	Is modeled by TORFormat, see section 4.3.1.5 and in the analysis part in section 4.2.2.6 in [TDP]
DO-330-10.3.1.d	Description of output files	Is modeled by TORFormat, see section 4.3.1.5 5 and in the analysis part in section 4.2.2.6 in [TDP]
DO-330-10.3.1.e	Requirements for tool functions and technical features	TRs are identified using mapping from TORs to TRs in section 4.3.1.2 in [TDP]
DO-330-10.3.1.f	Abnormal behaviour	Is modeled as TRRobustness requirements, see section 4.3.2.9 in [TDP]
DO-330-10.3.1.g	User Manuals	Are modeled using TRUserInstruction elements, see 4.3.2.3 in [TDP]
DO-330-10.3.1.h	Operational Use	TORs can be adopter using TOROther to the need of the user, see section 4.3.1, especially 4.3.1.7 in [TDP].

DO-330-10.3.1.i	Performance requirements	Are modeled using TRPerformance elements, see 4.3.2.10 in [TDP]
DO-330-10.3.2	Tool installation report	See subsections
DO-330-10.3.2.a	Identification of the environment	Installation report created at the beginning of the operational phase see section 7
DO-330-10.3.2.b	Identification of tool version	Also installation report in section 7
DO-330-10.3.2.c	Identification of external components	Also installation report in section 7
DO-330-10.3.2.d	Identification of the mean to execute the tool	If execution deviates from the default, described in section 6.2 in [TDP], it is documented in the installation report in section 7
DO-330-10.3.3	Verification and Validation cases and Procedures	The tool operational requirements verification is modelled in the TOR elements in [TDP], the satisfaction of the sub items is described in the following subsections
DO-330-10.3.3.a	Review and analysis procedures	The methods are modeled in the same way as the other verification methods, see section 7 in [TVP]
DO-330-10.3.3.b	Test cases	See section 6.1.2 in [TVP]
DO-330-10.3.3.c	Test procedures	See section 6.1.3 in [TVP]
DO-330-10.3.4	Operational V&V Results	The results are stored in the same way in the DO-330 model as the TR verification. See subsections
DO-330-10.3.4.a	Passed/failed indication	Is modeled in verification data assigned to the TORs in section 6.2.2 in [TVP]
DO-330-10.3.4.b	Identification of configuration item	Configuration items are linked in the DO-330 model of the verification data, see section
DO-330-10.3.4.c	Result description	In the Results attribute of verification data, see section 6.2.2 in [TVP]
DO-330-10.3.4.d	Discrepancies to be locked in problem reports	New Discrepancies are logged as problem reports, see qualification report in section 6.

Table 8: Tracing Table to Tool Qualification Data

8.8 Tracing to Additional Considerations for Tool Qualification Section

Identifier	Keyword	Satisfaction Comment
DO-330-11.1	Multi-function tools	See subsections
DO-330-11.1.a	Function Analysis	Same as the analysis of tools but for different plugins, see section 3.2 in [TDP]
DO-330-11.1.b	Protection against lower qualified functions	Protection is achieved from the plugin mechanisms and described in section 3.2 in [TDP]
DO-330-11.1.c	Separate qualification	Is supported using separate plugins, see section 3.2 in [TDP]
DO-330-11.1.d	Protection and Independence	Is described in section 3.2 in [TDP]
DO-330-11.1.e	Integration of unqualified functions	See section 3.2 in [TDP]
DO-330-11.2	Previously qualified tools	See subsections
DO-330-11.2.1	Reuse of previously qualified tools	See subsections
DO-330-11.2.1.a	Approved by CA	See liaison process in section 4
DO-330-11.2.1.b	Same or lower TQL	See liaison process in section 4
DO-330-11.2.1.c	Unchanged life cycle data	See liaison process in section 4
DO-330-11.2.1.d	Equivalent environment	See liaison process in section 4
DO-330-11.2.1.e	Same TORs	See liaison process in section 4
DO-330-11.2.1.f	Access to the qualification data	See liaison process in section 4
DO-330-11.2.1.g	Same version used	See liaison process in section 4
DO-330-11.2.2	Changed environment	See subsections
DO-330-11.2.2.a	New environment is equivalent to verification environment	See liaison process in section 4
DO-330-11.2.2.b	Tool operational environment requirements fit to new environment	See liaison process in section 4
DO-330-11.2.2.c	Environment requirements complies with life cycle	See liaison process in section 4
DO-330-11.2.3	Changes to previously qualified tools	See section 5.3.4 in [TDP] and the following subsections
DO-330-11.2.3.a	Tool operational requirements	See liaison process in section 4
DO-330-11.2.3.b	Tool requirements	See liaison process in section 4
DO-330-11.2.3.c	Tool design description	See liaison process in section 4
DO-330-11.2.3.d	Tool source code	See liaison process in section 4
DO-330-11.2.3.e	Environment and development process	See liaison process in section 4
DO-330-11.3	Qualifying COTS Tools	No differences to the Eclipse process for qualification of plugins, see section 4
DO-330-11.4	Service history	See subsections
DO-330-11.4.1	Reasons for using tool service history	See subsections
DO-330-11.4.1.a	Available data	The relevant data is available in the DO-330 model, see section in 4.7 [TDP]
DO-330-11.4.1.b	Problem reporting	Problem reporting is part of the DO-330 model see section in 4.7 [TDP]and is included in the qualification build (see section 6.2 in [TDP]), together with the

		CM data, see section in 4.7 [TDP]
DO-330-11.4.1.c	Environment	See the environment DO-330 model of Problem reports in section 4.7.2 in [TDP]
DO-330-11.4.1.d	Operation	See the component in DO-330 model of Problem reports in section 4.7.2 in [TDP]
DO-330-11.4.1.e	Stability and maturity	Can be inferred from the number of bugs and their affected elements, which are modeled in section 4.7.6 in [TDP]
DO-330-11.4.1.f	Application service	See the date DO-330 model of Problem reports in section 4.7.2 in [TDP]
DO-330-11.4.2	Tool service history activities	See subsections
DO-330-11.4.2.a	Service history for TOR compliance	The service history is part of the TOR compliance argumentation in the liaison process, see section 4
DO-330-11.4.2.b	Environment support	Can be deduced from the bug history projection for given environment, see section 4.7.2 in [TDP]
DO-330-11.4.2.c	Problem reporting is established	See subsections and QA process in section 4.7 in [TDP]
DO-330-11.4.2.c.1	Representative data available	Data is in the QA part of the DO-330 model, see section 4.7 in [TDP]
DO-330-11.4.2.c.2	Tool problems reported, resulting actions recorded	Actions are recorded in the CM and part of the Qualification Kit, see sections 4.7.6 and 6.2 in [TDP]
DO-330-11.4.2.c.3	Tool problems were analyzed	Every problem report is associated with a test case to reproduce it, see section 4.7.2 in [TDP]
DO-330-11.4.2.d	Identification of configuration changes	Can be used in the qualification liaison process, see section 4
DO-330-11.4.2.e	Analysis of environment	Can be used in the qualification liaison process, see section 4
DO-330-11.4.2.f	Subset tool qualification	Is done by splitting plugins based on service history, see section 4.7.2 in [TDP]
DO-330-11.4.2.g	PSAC or TQP references to service data	Is done in TQP/TQR as described in sections 5, 6 and the subsections
DO-330-11.4.2.g.1	Summary of service history data	Relevant bugs are in the TQP/TQR, see section 5 and 6
DO-330-11.4.2.g.2	Rationale for use of tool's service history	The tool service history is explained in the liaison process in section 4
DO-330-11.4.2.g.3	Analysis of the relevance of the tools history	Relevant test cases are part of the tool qualification I n sections 5 and 6
DO-330-11.4.2.g.4	Availability of problem reporting	Are part of the DO-330 model, see section 4.7.2 in [TDP]
DO-330-11.4.2.g.5	Tool change history	CM data is part of the qualification build, see section 6.2 in [TDP]
DO-330-11.5	Alternative methods for tool qualification	See subsections
DO-330-11.5.a	Alternative method justification	See liaison process and the automatic TQL determination base on this alternative process like ISO 26262 as described in section 4
DO-330-11.5.b	PSAC / TQP contents	See subsections

DO-330-11.5.b.1	Impact on process	More flexible, see section 4
DO-330-11.5.b.2	Impact on life cycle data	Contained in DO-330 model, see section 4.2.2 in [TDP]
DO-330-11.5.b.3	Rationale for use of the method	Flexibility and automatisation of determination, see section 4

Table 9: Tracing Table to Additional Considerations for Tool Qualification

9 References

[DO-330] RTCA DO-330 Software Tool Qualification, December 13, 2011.

[ISO26262] ISO 26262: Road Vehicles - Functional Safety, November 2011.

[TCA_TR] Validas (Draft): Tool Requirements for the Tool Chain Analyzer, Version 0.2

[TCA_TD] Validas (Draft): Tool Design for the Tool Chain Analyzer, Version 0.2

[TCA_TQP] Validas (Draft): Tool Qualification Plan for the Tool Chain Analyzer, Version 0.2

[TCA_TTS] Validas (Draft): Tool Test Specification for the Tool Chain Analyzer, Version 0.2

[TDP] Eclipse (Proposal): Tool Development Plan for every Qualifiable Eclipse Plugin, Version 1.0

[TVP] Eclipse (Proposal): Tool Verification Plan for every Qualifiable Eclipse Plugin, Version 1.0