



Oscar Slotosch, Validas AG

Qualification of Eclipse-based Tools Vision and Roadmap-Status

Tool Qualification (Summary)

- Standards require tool qualification: ISO 26262, IEC 61508, DO, EN 50128
- Qualification process:
 - Classify all used tools (Impact, Use-Cases, Artifacts)
 - Qualify critical tools
 - Use tools

Qualification Methods ISO 26262

Table 4 — Qualification of software tools classified TCL3

	Methods		ASIL			
			в	С	D	
1a	Increased confidence from use in accordance with 11.4.7	++	++	+	+	
1b	Evaluation of the tool development process in accordance with 11.4.8	++	++	+	+	
1c	Validation of the software tool in accordance with 11.4.9	+	+	++	++	
1d	Development in accordance with a safety standard ^a	+	+	++	++	

- Qualification Method DO-330 Development in accordance with a safety standard:
 - Processes Requirements
 - Required Documents
 - Required Verification
 - Required Qualification Process

 Tool Life Cycle Processes

 Tool Qualification Planning Process - Section 4

 Tool Development Processes - Section 5

 Integral Processes

 Tool Verification Process - Section 6

 Tool Configuration Management Process - Section 7

 Tool Quality Assurance Process - Section 8

 Certification Liaison Process to qualify the Tools - Section 9

 Tool Qualification Data - Section 10

 Additional Considerations for Tool Qualification-Section 11

. . .

Qualification of Eclipse-Based Tools

- **Determine Tool Use-Cases (required features / "TORs")** 1.
- **Determine potential errors in features** 2.
- **Determine possible checks & mitigations** 3.
- **Determine qualification needs (remaining critical features)** 4.
 - Identify critical features / contributions (Actions, ...) Α.
 - Identify the plugins of critical features / contributions Β.

All Extensions

Define extensions for this plug-in in the following section. type filter text Image: org.eclipse.core.runtime.products iante di la construita de la construita ToolChainAnalyzer.editor.objectContributionToolExport (objectContribution) 🗄 🛛 🗶 Export (menu) 👬 Tool (XML) (action) 😑 🔀 ToolChainAnalyzer.editor.objectContributionToolChainExport (objectContribution) 🗄 🕅 🕅 Export (menu) Default Errors (XML) (action) X Excel Calls (action) Excel Error (action) 🔎 Excel Feature (action) 🖗 Excel Review (action) 🤴 Excel Tool-Artifact Matrix (action) 🔗 Excel Tool Attribute (action) S GraphViz (DOT) (action) IcolChainAnalyzer.editor.objectContributionToolChainImport (objectContribution) ToolChainAnalyzer.editor.objectContributionError (objectContribution) ToolChainAnalyzer.editor.objectContributionUseCase (objectContribution) ToolChainAnalyzer.editor.objectContributionTool (objectContribution) 🗄 🕅 ToolChainAnalyzer.editor.objectContributionArtifact (objectContribution)

Sependencies

Required Plug-ins

Specify the list of plug-ins required for the operation of this plug-in.

org.eclipse.core.runtime 🚏 ToolChainAnalyzer.edit 🕸 org.eclipse.emf.ecore.xmi 🏶 ora.eclipse.emf.edit.ui org.eclipse.core.resources (3.5.0) Examples (1.5.0) Documentation (1.5.0) 🀌 org.eclipse.ui.ide (3.6.0) 🐌 org.eclipse.ui borg.eclipse.ui.forms (3.5.0) de.validas.tca.util (1.0.0) 🍫 de.validas.coverage (1.0.0)

- Qualify critical features in plugin (not complete plugins!) including derivation of 5. assumptions on required plugins / features
- **Check Qualification requirements of required plugins / features** 6.

Validas AG

Qualification according to DO-330



• Tools

- Tool Operation Requirements (TORs): "The critical features" in the plugin
- Tool Requirements (TRs): Features and design of plugin (mapped to TORs)
- ...
- Complete safety development process including V&V

COTS-Tools (unchanged tools e.g. standard eclipse plugins)

- DO-330 requires "almost complete" DO-330 development
- Main Changes:
 - T-0: Tool Operational Process
 - T-10: Tool Qualification Liasion Process
- Eclipse: Qualification has to be applicable for each plugin
 - Eclipse defaults and
 - tool specific
- Qualification of plugins will make assumptions on Validas required plugins

	Annex A Table	Objectives Applicable to Tool Developer
+	T-0	Objectives 2, 4 and 5: Applicable
ι		Others: Not applicable
	T-1	All: Applicable
	T-2	All: Applicable
	T-3	All: Applicable
	T-4 to T-7	All: Applicable
	T-8	All: Applicable
	T-9	All: Applicable
	T-10	All: Not applicable

Qualification Kit for Eclipse-Plugins



Requirements

- Tool Operational Requirements (TORs)
- Tool Requirements (TRs) including architecture requirements
- Low-Level-Requirements (LLRs)

Tracing

- TORs <-> TRs
- TRs <-> LLRs
- LLRs <-> Code
- TRs <-> Tests
- LLRs <-> Tests

Process

- Definitions
- Documentation of development

Verification

- Tests, Reviews, Tracing

Validas AG Process conformance

Requirements structure covering DO-330 aspects



Tool Automatization possibilities in Eclipse (critical features)

- Requirements Modeling
- Tracing data using @tags/templates
- Completeness checking
- Test execution (including coverage)
- Many generation steps
 - LLRs from Java-doc
 - Architecture Requirements from EMF
 - Tests from EMF
- Generation of DO-330 documents ...

Required Documentation

Generic Documents

- For development of all plugins
- Once created and validated
- E.g. Tool Development Plan, Tool Verification Plan

Tool/Plugin specific documents

- Can be generated from
 - the presented model
 - The chosen architecture (packages, EMF, xText,..)
 - The code-documentation
- Need to be verified
- Examples
 - Requirements for <Tool>
 - Design for <Tool>
 - Test-Specification for <Tool>
 -

Both need to be checked for DO-330 compliance



Tool Development Plan

_						
1[Document History					
2 Definitions						
3 General Information (can be removed if the document is						
4 4 Eclipse Development Standards						
4.1 Tool Requirements						
4 4.1.1 Project						
	4.1.1.1 Documentation					
	4.1.1.2 Definition					
	4.1.1.3 TQL					
 4.1.2 TORs 4.1.3 TRs 						
					4.1.3.1 ToolRequirements	
	4.1.3.1.1 TRUserInstruction					
	4.1.3.1.2 TROpMode					
	4.1.3.1.3 TRFunction					
	4.1.3.1.4 TRCustomizing					
	4.1.3.1.5 TRInterface					
	4.1.3.1.6 TRExpectedErrorMessage					
	4.1.3.1.7 TRRobustness					
	4.1.3.1.8 TRPerformance					
	4.1.3.1.9 TROther					
_	4.1.4 ControlStatus					
4	4.2 Tool Design					
	▷ 4.2.1 ArchitectureRequirement					
4.2.2 LLRs						
	4.2.2.1 LowLevelRequirement					
	4.2.3 Code					
_	4.2.4 Visibility					
	4.3 Tool Code Standards					
5 T	ool Life Cycle					
6 T	ool Development Environment					
Re	ferences					

Status 3rd April 2012



- Created EMF model "DO-330" (including generic editor tool!) to generate
 - Requirements for <Tool/Plugin>
 - Design for < Tool/Plugin >
 - Test-Specification for <Tool/Plugin >
- Created Instance of the DO-330 model for the Validas Tool "TCA"
- Manually created documents (drafts to show structure of documents)
 - Requirements for TCA
 - Design for TCA
 - Test-Specification for TCA
- Selected Test Environment (CodeCover) for executing tests with MC/DC coverage measurement
- Implemented one reference test for TCA with 100% MC/DC coverage
- Started working on generic Tool Development Plan document
- Started checking DO-330 compliance
- Next steps:
 - Extend processes and general documents: Verification, Quality Assurance, Life Cycle, ...
 - Extend model & documentation examples
 - Continue DO-330 compliance checking
- Validas AG Integration into Eclipse Foundation (processes, documents, ...): Eclipse Project

Roadmap - Status April 2012



- 1. Identify goals & requirements for tool qualification in Eclipse
- 2. Propose process / project
- 3. Demonstrate tool qualification & improve proposal
- 4. Establish proposal: Qualify (selected) plugins





Status April 2012

Thank You!







Arnulfstraße 27 80335 München www.validas.de info@validas.de