



**Oscar Slotosch, Validas AG** 

# Testing for Tool Qualification of Eclipse Tools



#### Validas AG

- Tool Qualification of Eclipse-Based Tools
  - Requirements: DO-330
  - Roadmap
  - Model-based approach
- Test Requirements
- Summary

# Validas AG About Us



- We are a technology consultancy for quality assurance of embedded systems
- Our core competences are model based development, model based testing, test automation, tool qualification
- We develop innovative software engineering methods, implement them in form of tools and processes and we support our customers in their application.
- We are members of AUTOSAR and Eclipse Foundation



## Clients





# Validas Expertise



#### Tool qualification

- Application of validation suites
- Creation and guidance for creation of validation suites
  - Customer specific
  - Tool specific
- Method development: qualification of Eclipse-based tools

#### Avoidance of tool qualification

- Tool chain analysis: method and tool (free research prototype)
- Tailoring / focusing of validation suites
- Customer support for tool providers
- Customizing and integration of tools

World-wide leading expertise





Validas AG

#### Tool Qualification of Eclipse-Based Tools

- Requirements: DO-330
- Roadmap
- Model-based approach
- Test Requirements
- Summary

# **Tool Qualification**

- Tool Qualification := qualification of tools
- 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 according to ISO 26262

Table 4 — Qualification of software tools classified TCL3

	Mathada	ASIL					
	MethodsABCIncreased confidence from use in accordance with 11.4.7+++++Evaluation of the tool development process in accordance with 11.4.8+++++Validation of the software tool in accordance with 11.4.9++++++Development in accordance with a safety standard <sup>a</sup> ++++++	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 <sup>a</sup>	+	+	++	++		

#### Some tools provide qualification kits for confidence with evidence into

- Correctness of functions by testing them "validation"
- Development process by documentation

	DO-330 Required Processes
Γool	ife Cycle Processes
	Tool Qualification Planning Process - Section 4
	Tool Development Processes - Section 5
	ntegral 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-



# Avoidance of Tool Qualification

- Analyze process how tools are used
- Identify potential errors
- Identify possible mitigations (checks & restrictions)
- Compute "Confidence Level" / qualification need
  - High error detection => low confidence level
- Document results
  - Confidence Level of Tools (in the analyzed process)
  - Potential errors to exclude with qualification
- Tool Chain Analyzer Tool to support avoidance of tool qualification
  - Provided by Validas AG
  - Developed in research project
  - free download: <u>www.validas.de/TCA170.zip</u>

recon



makerca is bocuckampierca		L Properties 23	📑 🛅 🏶 🕸 🔅				
Resource Set		Property	Value				
file:/C:/Program%20Files/TCA150/plugins/Examples_1.5.0/Make.tca	~	Name	Embedded Tools Development				
Tool Chain Embedded Tools Development (TCL3)		Description	Small development system for embedded SW.				
<ul> <li>Default Error Attributes</li> </ul>		Impact	1 true				
🚦 Tool GCC (TCL2)		Is Assumption	1 false				
📇 Use Case GCC:PC Compile (TCL2)		Comment	12				
Use Case GCC:Target Compile (TCL2)	=	Long Description	12				
E Feature GCC:Compile (TCL2)		Tool Attributes					
Feature GCC:Link (TCL2)		Inputs					
[1] Feature GCC:Tool Application Guide (TCL1)		Outputs					
Tool Make (TCL1)		Inputs Outputs					
Tool Review (TCL1)		Called Tools					
Tool Script (TCL1)		Calling Tools					
Tool SVN (TCL1)		Deactivated	Life false				
Tool Test Tool (TCL3)		ASIL	t≡ A				
S Artifact Coverage Report:SVNFile		Use Assumptions	A B				
S Artifact Executable		Show Only Assumptions	C				
S Artifact Library:SVNFile		Ignore Artifacts	D				
S Artifact Logrile:SVIVEIle		Default Assumption value For	r New Ele				
S Artifact Makefile:SVNFile	*						
Parent List Tree Table Tree with Columns							
Dataflow 🖾			2				
elected element: 🗗 Feature GCC:Compile (TCL2)							
puts:		Outputs:					
Artifact Logfile:SVNFile		Artifact Logfile:SVNFile					
Artifact Source Code:SVNFile		Artifact Object Code					
<i>oy</i>		Feature GCC:Link (TCL2)					
		Artifact	Executable				
		🗼 👘 Use	Case Test Tool:Debug (TCL3)				
		⊳ ⊡ Feat	ure Test Tool:Run Test (TCL3)				
		Artifact Logfile:SVNFile					
		Artifact Mapfile					

# **Eclipse Roadmap to Tool Qualification**



#### wiki.eclipse.org/Auto IWG WP5

- 1. Goals: DO-330
- 2. Concept: model-based tool qualification
- 3. Demonstrate & implement with an Eclipse Project: QPP (Qualifiable Plugin Projects)
- 4. Qualify (selected) plugins





# **Tool Qualification Model for Eclipse**



- Currently Eclipse does not support qualification
- Extended meta-model (and documentation) covers 100% (~ 450 Requirements) of DO-330

#### **Current Metamodel**

#### **New Extended Metamodel**





- Validas AG
- Tool Qualification of Eclipse-Based Tools
  - Requirements: DO-330
  - Roadmap
  - Model-based approach
- Test Requirements
- Summary

### **Test & Verification Model**



Relates test to requirements (TOR, TR, LLR) & implementation



### Test Coverage: CodeCover

}



Run As	tun As			1 CodeCover Measurement For JUnit					
			<ul> <li>j<sup>*</sup><sub>U</sub> 2 JUnit Plug-in Test</li> <li>J<sub>U</sub> 3 JUnit Test</li> </ul>				Alt+Shift+X, P Alt+Shift+X, T		
		Ju							
			Run C	Configurati					
Show methods with Term Coverage 🔹 🗸	90,5 %								
Name	Statement	Br	anch	Loop	Term	?-Operator	Synchronized		
😑 ToolChainAnalyzer	<b>—</b> 6,1 %	-	5,3 %	<b>0,0</b> %	4,6 %	-	-		
🖶 metamodel	<b>6</b> ,1 %		5,3 %	<b>0,0</b> %	4,6 %	-	-		
🖶 de	<b>—</b> 6,1 %	-	5,3 %	<b>0,0</b> %	4,6 %	-	-		
🖶 validas	<b>—</b> 6,1 %		5,3 %	<b>0,0</b> %	4,6 %	-	-		
🖶 iso26262	<b>6</b> ,1 %		5,3 %	<b>0,0</b> %	4,6 %	-	-		
🖶 checks	<b>—</b> 6,1 %	-	5,3 %	<b>0,0</b> %	4,6 %	-	-		
G AssumptionModel	<b>—</b> 6,1 %		5,3 %	0,0 %	4,6 %	-	-		
😳 getIsAssumption	<b>— 100,0 %</b>		100,0 %	-	100,0 %	-	-		
	*/ public static b if (item in Error u return	oole stan ce = uce.    	an getl ceof Er (Error islsAss (uce.ge (uce.ge	SAssumptic ror) { ) item; sumption() tRestrict: tCheck()	on(Object i    (uce.ge ion() != nu != null &&	tem) { tUseCase() 11 && getI getIsAssum	<pre>!= null &amp;&amp; getIsAssumption(uce.getUs sAssumption(uce.getRestriction())) ption(uce.getCheck()));</pre>	<mark>eCa</mark>	
	}	_							
	II (item in Check c	scan	(Check)	item:					
	return	ck.i	sIsAssu	mption()	getIsAss	sumption(ck	.getUseCase());		
	}								
	if (item in	stan	ceof Qu	alificati	on) {				
	Qualifi	cati	on qual	= (Quali:	fication) i	tem;			
	return	qual	.isIsAs	sumption()	)    (qual.	getUseCase	() != null && getIsAssumption(qual.get)	tUs	
			(qual.g	() () ()	!= null &&	getIsAssum	<pre>ption(qual.getTool()));</pre>		
	} if (item in	stan	ceof Re	striction	) {				
	Restric return	res.	res = isIsAss	(Restrict: sumption()	ion) item;    getIsAs	sumption(r	es.getUseCase());		
	}								
Validas AG	II (ICEM IN	uc a	= (UseC	ase) item					
	return	uc i	sTalaan	motion()	n II getIsles	umption/wa	.getTool()):		
Validas AG	UseCase	uc uc.i	= (UseC sIsAssu	(ase) item, (mption()	;    getIsAss	sumption (uc	<pre>.getTool());</pre>		

# Modeling Test Cases (Prototype)



	I Kesource Set			<b>D</b>	
				Property	Value
Delete these te	a 👰 file:/C:/git/qpp/org.eclipse.do	o330.mo	del/plugin.do330	Bugs	
Relate them to	a 💠 Project org.eclipse.do330.	model		Conditions	
	A TO Rs			Expected Outputs	L클
	TRs			ID	Hello Eclipse
– Reduirements	a 💠 Test Cases			Implementation	
	Test Case State Test	st 1		Inputs	LE
	Test Case Test Case	e GUI Sta	rt	Low Level Requirements	
<ul> <li>Implementation</li> </ul>	Test Case Hello Ecl	ipse	New Chline	Method	REQUIREMENT_BASED
	> 🚸 Bugs		New Sibling	Purpose	REQUIREMENT_BASED
		2	Undo Set	Status	EQUIVALENCE_CLASS
	Selection Parent List Tree Table	Tree 🔬	Pada	Test Procedures	BOUNDARY_VALUE
A THE THE REPORT OF			Reuo	<b>Tool Operational Requirements</b>	LOGICAL COMBINATION
Tool Requirements Test Case Hello Eclipse		of	Cut	Tool Requirements	FAULT_INJECTION
- Filter Available Choices		E.	Conv		INTERFACE
Chaire Bettern (* es 2)			copy		PERFORMANCE
Choice Pattern (" or ?)			Paste		ERROR_GUESSING
Choices Feature		×	Delete		BACK_TO_BACK
Arch Ecore Diagram do330.ecorediag     Arch Ecore do330.ecore     Add     Add	evelopment Tool Import		Validate	lello Eclipse	
Arch Genmodel do330.genmodel     Remove			Control		
Arch Property Bug Description Multiline					
♦ TR Function Development Tool Import			Load Resource		
♦ TR Interface Gerrit Interface			Defeash	EQUIREMENT_BASED	
Down			Refresh		
			Show Properties View	)pen	
	OK Cancel ts				

#### Execute them

In could also be generated by model-based test generation (which is not required from DO-330)



- Validas AG
- Tool Qualification of Eclipse-Based Tools
  - Requirements: DO-330
  - Roadmap
  - Model-based approach
- Test Requirements
- Summary

## Summary



- DO-330: First Standard for Tool Qualification
- Eclipse Roadmap to model-based tool qualification
- Validas AG: Testing for Tool Qualification
- Tool qualification avoidance of tool qualification
- Tool qualification requires verification and testing
- Test goals: avoidance of critical errors
- Concrete test methods & tools selectable
- CodeCover to measure code coverage of tests

#### First **Tool Qualification** Symposium

#### Presentations

- Tool user & tool provider
- Qualification requirements & qualification kits
- Experiences from different domains & different industries
- Practical experiences & practical support
- Keynote speech from F. Pothon
  - Tool Qualification Considerations and Certification
     Credits of Qualified Code Generators
  - Location: Munich Airport
- Registration and further information:
  - <u>http://toolqualification2013.eventbrite.com</u>
- Organization: Validas AG
  - tqs@validas.de





# **Thank You!**







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