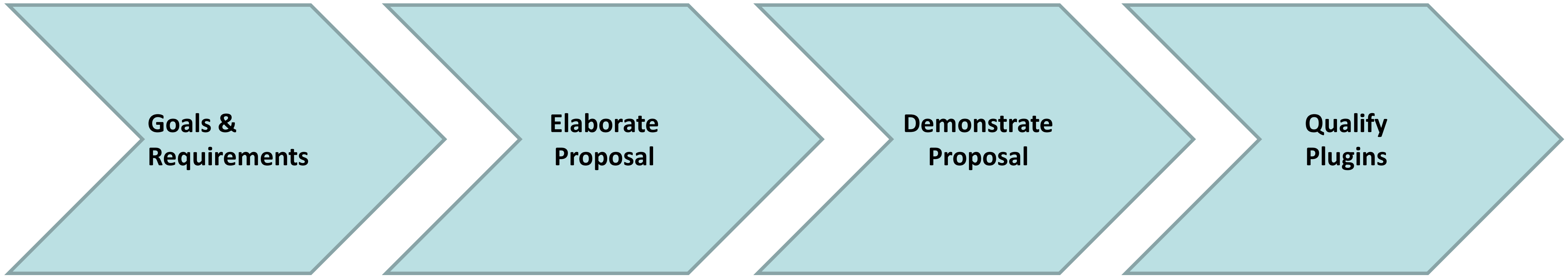
Oscar Slotosch, Validas AG

Proposal for a Roadmap towards Development of Qualifyable Eclipse Tools

Content

Roadmap

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



- Is this a Eclipse project? Not a typical ☺
- Is this an Industrial Working Group process?

Content

- Roadmap
- **Requirements for Tool Qualification (Standards)**
- Proposals for Goals for Eclipse
- Proposals for some steps towards Tool Qualification
- Summary

Tool Qualification (Summary)

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

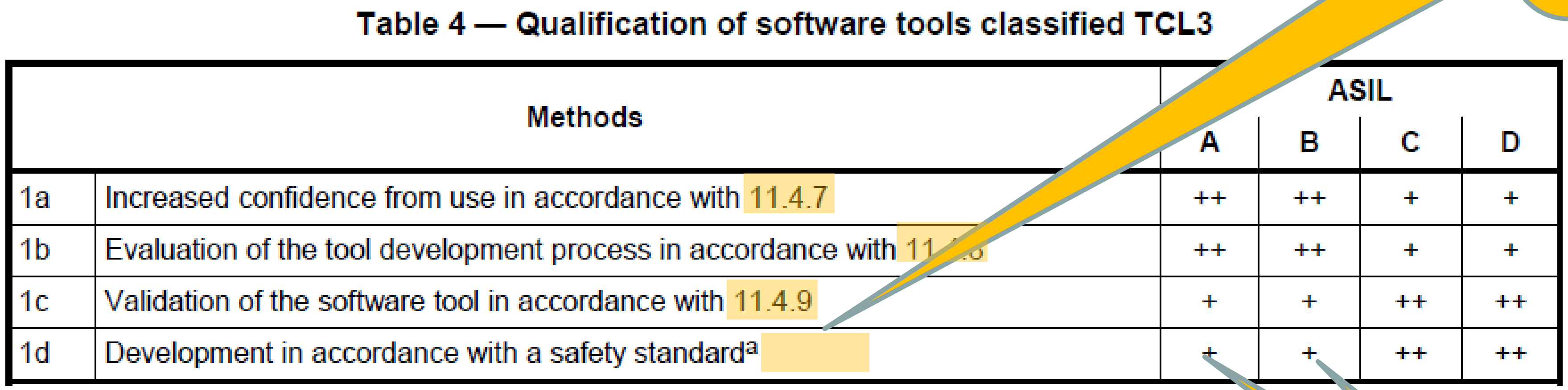


Table 4 — Qualification of software tools classified TCL3

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

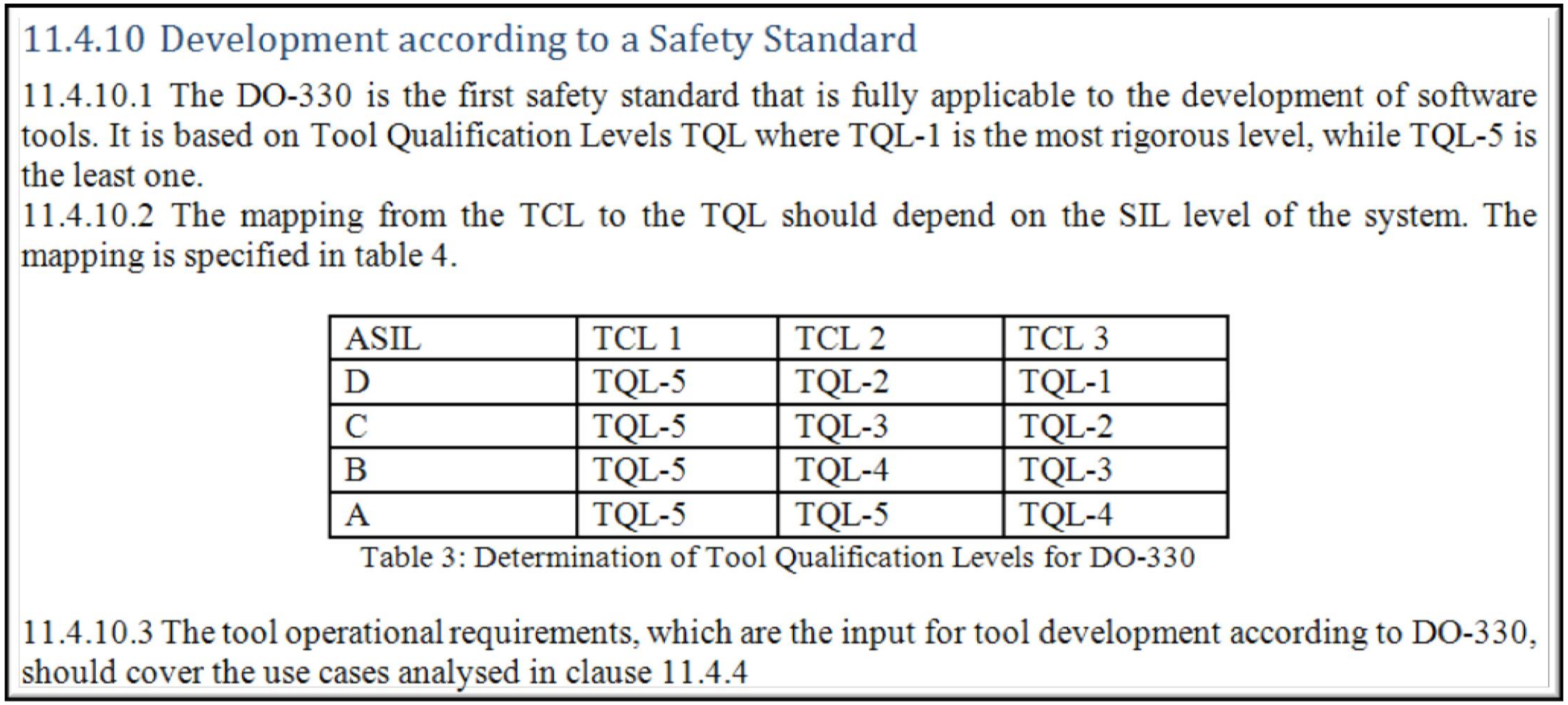
Here is a hole were the new DO-330 standard fits in

- **Some tools provide qualification kits for confidence with evidence into**
  - Correctness of functions by testing them “validatio
  - Development process by documentation
  - ….

Since DO-330 is scalable, here could also be a ++

Extension of the ISO 26262?

- Possible extension / integration of DO-330 into ISO 26262 could look like:



11.4.10 Development according to a Safety Standard

11.4.10.1 The DO-330 is the first safety standard that is fully applicable to the development of software tools. It is based on Tool Qualification Levels TQL where TQL-1 is the most rigorous level, while TQL-5 is the least one.

11.4.10.2 The mapping from the TCL to the TQL should depend on the SIL level of the system. The mapping is specified in table 4.

ASIL	TCL 1	TCL 2	TCL 3
D	TQL-5	TQL-2	TQL-1
C	TQL-5	TQL-3	TQL-2
B	TQL-5	TQL-4	TQL-3
A	TQL-5	TQL-5	TQL-4

Table 3: Determination of Tool Qualification Levels for DO-330

11.4.10.3 The tool operational requirements, which are the input for tool development according to DO-330, should cover the use cases analysed in clause 11.4.4

- Similar chapters exist in DO-178C and DO-254

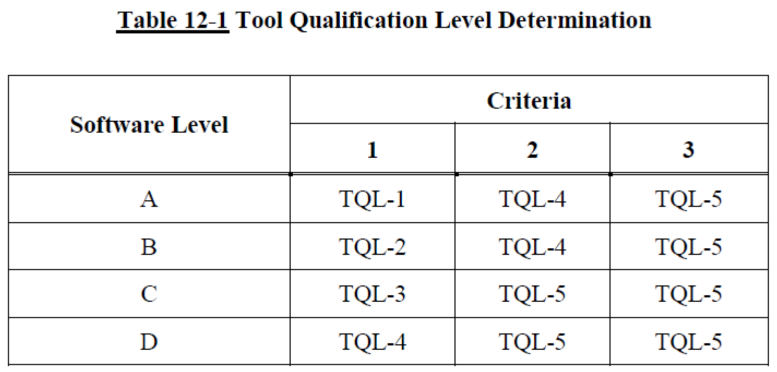


Table 12-1 Tool Qualification Level Determination

Software Level	Criteria		
	1	2	3
A	TQL-1	TQL-4	TQL-5
B	TQL-2	TQL-4	TQL-5
C	TQL-3	TQL-5	TQL-5
D	TQL-4	TQL-5	TQL-5

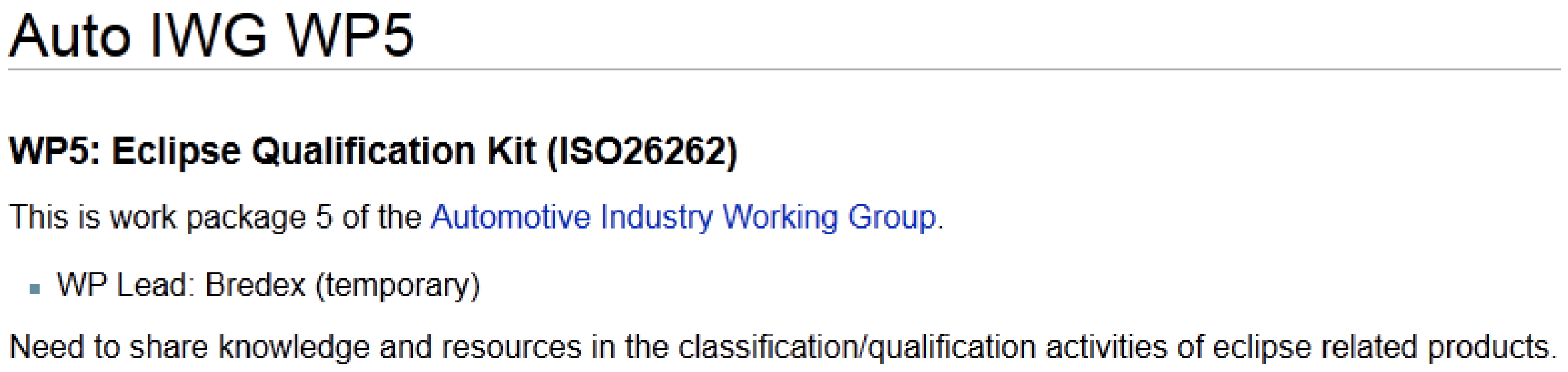
- Extension is not necessary to apply DO-330 in ISO 26262 but could clarify

Content

Goals for Eclipse IWG

- **Exchange & share knowledge**
  - Motivate developers & community to provide qualifyable plugins
- **Provide classification support to users of Eclipse tools**
- **Support the development of qualifyable tools ("Qualification Kits")**
  - Validation
  - Safety-Standard (DO-330)
- **Apply this to reference tools ARTOP, EMF,… ?**

- **Current status (web-page):**



Auto IWG WP5

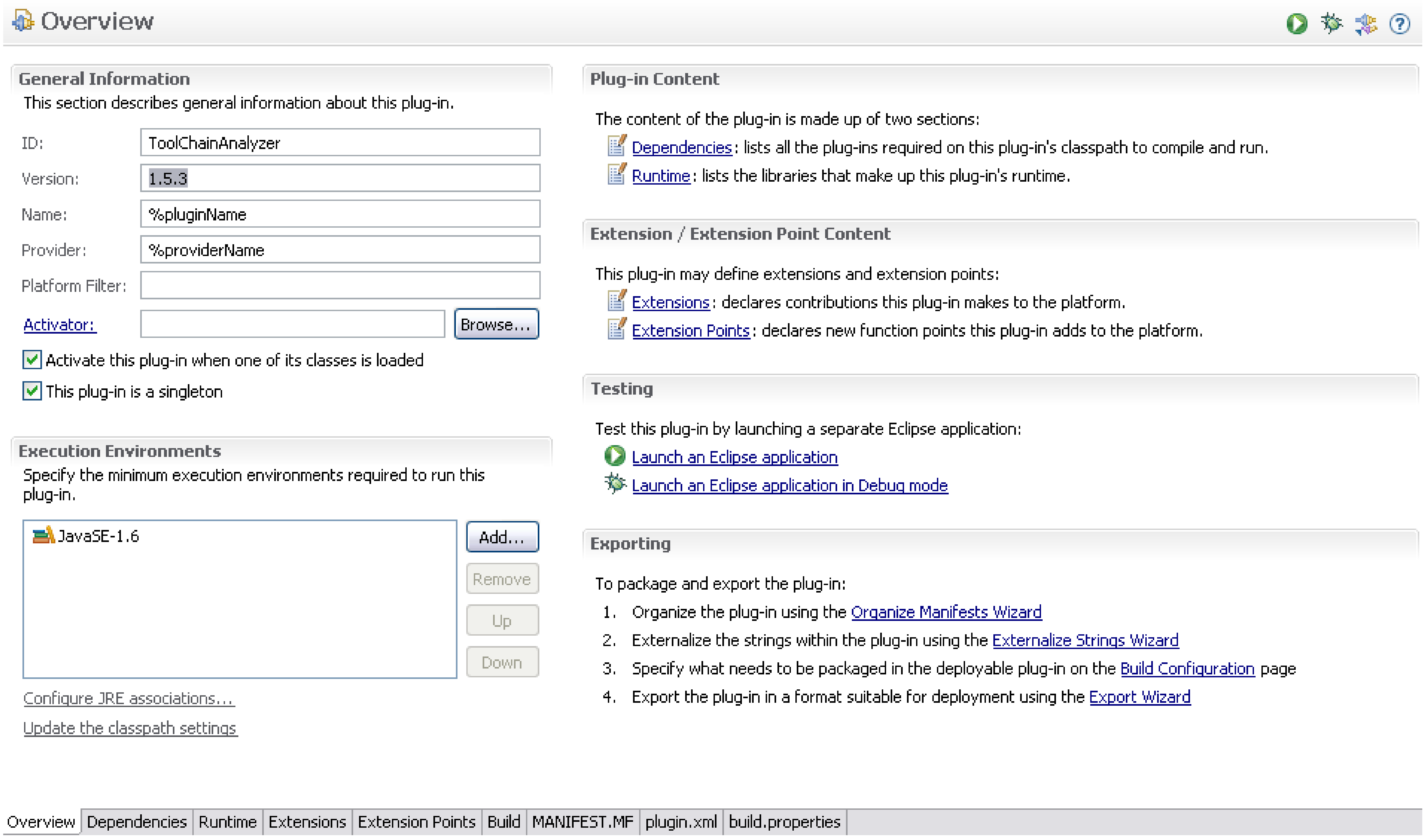
WP5: Eclipse Qualification Kit (ISO26262)

This is work package 5 of the Automotive Industry Working Group.

- WP Lead: Bredex (temporary)

Need to share knowledge and resources in the classification/qualification activities of eclipse related products.

Current Eclipse Metadata



Overview

General Information

This section describes general information about this plug-in.

ID:	ToolChainAnalyzer
Version:	1.5.3
Name:	%pluginName
Provider:	%providerName
Platform Filter:	
Activator:	

☑ Activate this plug-in when one of its classes is loaded

☑ This plug-in is a singleton

Execution Environments

Specify the minimum execution environments required to run this plug-in.

JavaSE-1.6

Configure JRE associations...

Update the classpath settings

Plug-in Content

The content of the plug-in is made up of two sections:

- Dependencies: lists all the plug-ins required on this plug-in's classpath to compile and run.
- Runtime: lists the libraries that make up this plug-in's runtime.

Extension / Extension Point Content

This plug-in may define extensions and extension points:

- Extensions: declares contributions this plug-in makes to the platform.
- Extension Points: declares new function points this plug-in adds to the platform.

Testing

Test this plug-in by launching a separate Eclipse application:

- Launch an Eclipse application
- Launch an Eclipse application in Debug mode

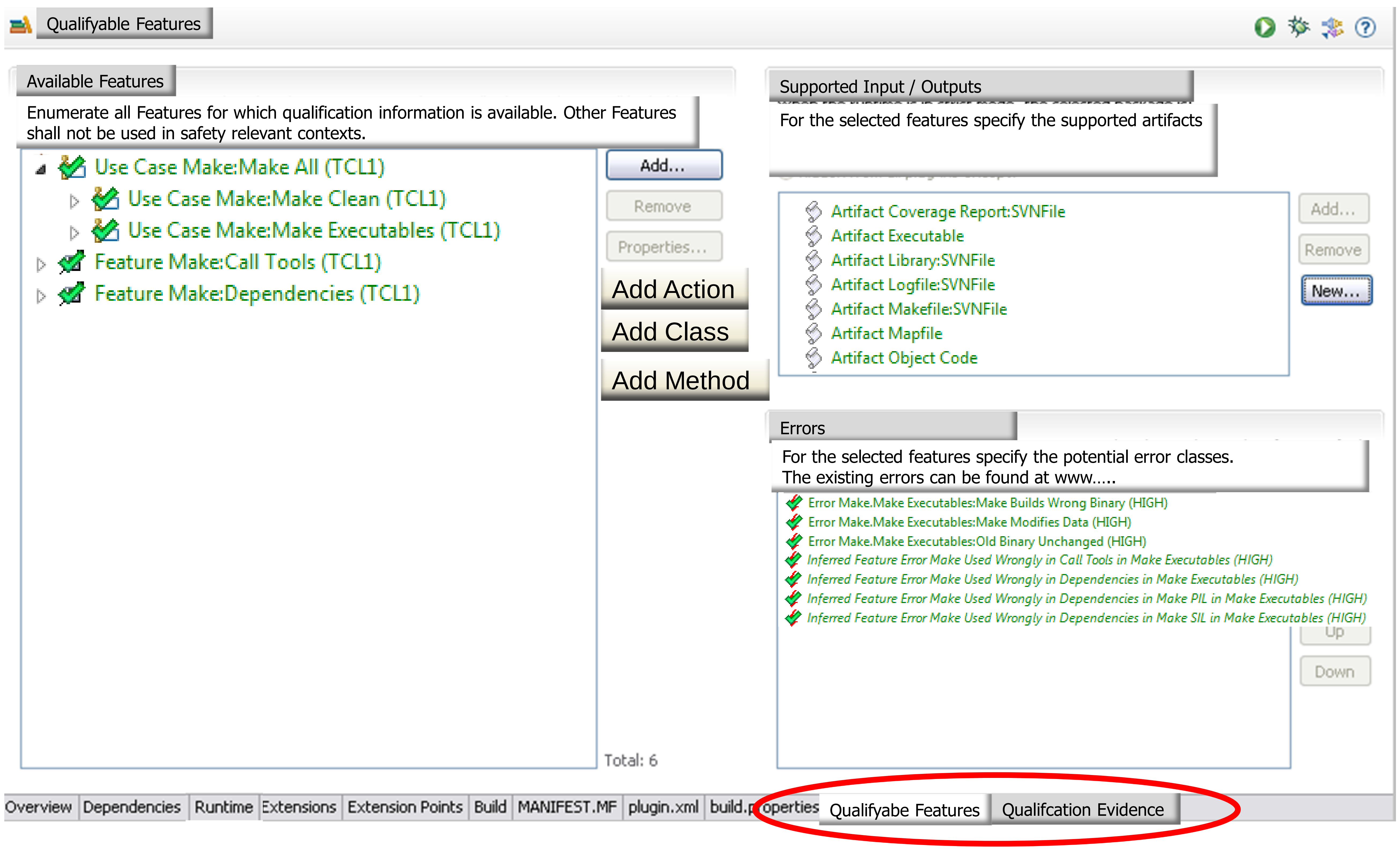
Exporting

To package and export the plug-in:

1. Organize the plug-in using the Organize Manifests Wizard
2. Externalize the strings within the plug-in using the Externalize Strings Wizard
3. Specify what needs to be packaged in the deployable plug-in on the Build Configuration page
4. Export the plug-in in a format suitable for deployment using the Export Wizard

Browse...
Add...
Remove
Up
Down
Overview | Dependencies | Runtime | Extensions | Extension Points | Build | MANIFEST.MF | plugin.xml | build.properties

Vision: Eclipse Classification Data



Qualifyable Features

Available Features

Enumerate all Features for which qualification information is available. Other Features shall not be used in safety relevant contexts.

- Use Case Make:Make All (TCL1)
  - Use Case Make:Make Clean (TCL1)
  - Use Case Make:Make Executables (TCL1)
- Feature Make:Call Tools (TCL1)
- Feature Make:Dependencies (TCL1)

Supported Input / Outputs

For the selected features specify the supported artifacts

- Artifact Coverage Report:SVNFile
- Artifact Executable
- Artifact Library:SVNFile
- Artifact Logfile:SVNFile
- Artifact Makefile:SVNFile
- Artifact Mapfile
- Artifact Object Code

Errors

For the selected features specify the potential error classes.
The existing errors can be found at www.....

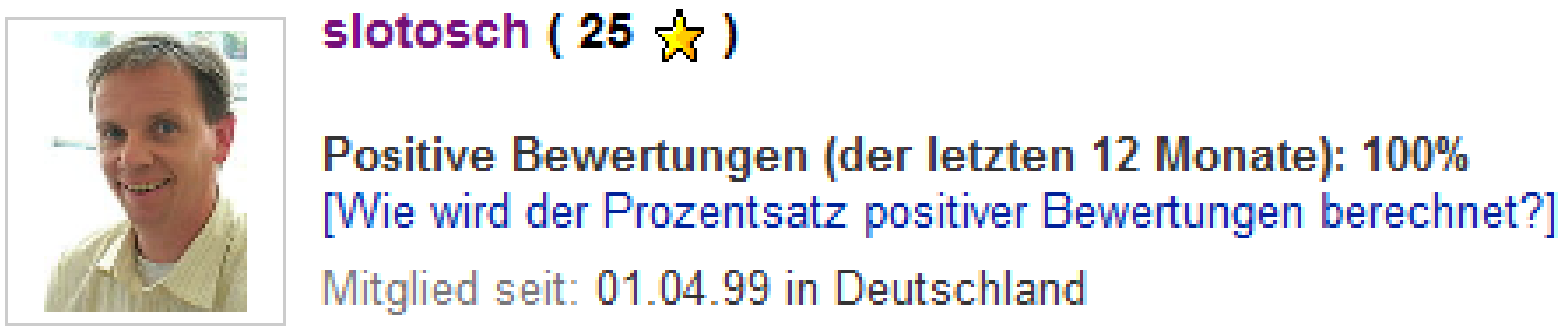
- Error Make.Make Executables:Make Builds Wrong Binary (HIGH)
- Error Make.Make Executables:Make Modifies Data (HIGH)
- Error Make.Make Executables:Old Binary Unchanged (HIGH)
- *Inferred Feature Error Make Used Wrongly in Call Tools in Make Executables (HIGH)*
- *Inferred Feature Error Make Used Wrongly in Dependencies in Make Executables (HIGH)*
- *Inferred Feature Error Make Used Wrongly in Dependencies in Make PIL in Make Executables (HIGH)*
- *Inferred Feature Error Make Used Wrongly in Dependencies in Make SIL in Make Executables (HIGH)*

Up
Down
Overview | Dependencies | Runtime | Extensions | Extension Points | Build | MANIFEST.MF | plugin.xml | build.properties | Qualifyabe Features | Qualifcation Evidence

Proposed Role: Eclipse Validator

There is much (different) work to do such that we need a new kind of worker: The Validator

- **Should provide confidence**
- **Should be more formalized than a committer**
- **Should have qualifications e.g. by filling out questionnaires on**
  - Eclipse qualification process
  - DO-330
- **Should have responsibilities (answer to questions)**
- **Should earn "credits" for each successful validation action**
  - Executed reviews
  - Formulated requirements
  - Created use/test cases
  - Feedback
  - …
- **Comparable: Confidence in ebay:**



slotosch (25 ☆)

Positive Bewertungen (der letzten 12 Monate): 100%
[Wie wird der Prozentsatz positiver Bewertungen berechnet?]

Mitglied seit: 01.04.99 in Deutschland

Content

Proposals

Elaborate Process → Demonstrate Process

Following activities are necessary to achieve goals:

- **Agree on focus, e.g. “Metadata extension for qualification information”**
- **Provide classification support to users of plugins**
  - Use case
  - Potential errors
  - Possible mitigations for errors
  - TCL inference
- **Provide qualification support**
  - Create checklist for DO-330 requirements (depending on the TQL)
    - Qualification data (general, plugin specific, user adaptable)
    - Requirements (general, development, operational)
  - Check Eclipse against the checklist, create
    - Mapping of Eclipse -> DO-330
    - Identify gaps: missing data/requirements
  - Provide model (EMF?) for the missing data
- **Demonstrate it: Small example e.g. EclipseCon**
- **Validate it: bigger example**

Content

- Roadmap
- Requirements for Tool Qualification (Standards)
- Proposals for Goals for Eclipse
- Proposals for some steps towards Tool Qualification
- **Summary**

Summary

- **Roadmap towards development of qualifyable Eclipse tools & plugins**
  - Classification
  - Qualification
  - Usage
- **Applicable to all relevant standards (ISO 26262, IEC 61508, DO-178C, EN 50128,..)**
- **Metadata extension for qualification information of plugins**
- **Much work to do**
  - Checklist
  - Gaps & Mapping
  - Extension of Eclipse processes, metadata, community
  - Improve eclipse plugins where needed
- **Proposed new role for that work: Eclipse Validator**
- **Validas will contribute**

Thank You!

VALIDAS

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