Enabling Development of Qualifiable Eclipse-based Tools: Vision and Concept
Content

- **Tool Qualification Requirements from Standards**
- **Tool Qualification Roadmap**
  - Vision
  - DO-330
  - Concept
    - Model-based Tool Qualification
    - Examples
      - Processes
      - Documents
    - Status: May 2012
- **Current Demonstration Examples**
- **Summary**
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

<table>
<thead>
<tr>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1a</td>
<td>++</td>
</tr>
<tr>
<td>1b</td>
<td>++</td>
</tr>
<tr>
<td>1c</td>
<td>+</td>
</tr>
<tr>
<td>1d</td>
<td>+</td>
</tr>
</tbody>
</table>

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

- Tool Qualification Requirements from Standards
- Tool Qualification Roadmap
  - Vision
  - DO-330
  - Concept
    - Model-based Tool Qualification
    - Examples
      - Processes
      - Documents
    - Status: May 2012
- Current Demonstration Examples
- Summary
Vision: Eclipse Development Process

- Currently Eclipse does not support qualification
- There is a road towards tool qualification for Eclipse, see [http://wiki.eclipse.org/Auto_IWG_WP5](http://wiki.eclipse.org/Auto_IWG_WP5)
- DO-330 is a safety standard for tools

Current Process

- Architecture
- Code
- Integration

New Extended Process

- Operational Requirements (Use Cases)
- Requirements
- Qualification

Eclipse Project: Qualifiable Plugin Projects (QPP)

- How-To Qualify Tools according DO-330
- Tool Development Plan
- Tool Verification Plan

- Requirements-Specification
- Design-Specification
- Test-Specification
- Tool Analysis (TCL/PSAC)
Vision: Eclipse Classification Data

Available Features

- 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

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

Errors

- 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 HIL in Make Executables (HIGH)
- Inferred Feature Error Make Used Wrongly in Dependencies in Make SIL in Make Executables (HIGH)

Add Action
Add Class
Add Method
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
3rd Build: Qualification Kit

- Currently: 2 Builds available in Eclipse
  - Source Build
  - Binary Build
- Missing: Qualifiable Build Configuration with plugin specific
  - Qualification information (TQL, DO-330 Model)
  - Test Cases / Coverage
  - Verification results
  - Documents
  - Involved Validators
  - ...
Content

- Tool Qualification Requirements from Standards
- Tool Qualification Roadmap
  - Vision
  - DO-330
  - Concept
    - Model-based Tool Qualification
    - Examples
      - Processes
      - Documents
    - Status: May 2012
- Current Demonstration Examples
- Summary
DO-330: Software Tool Qualification Considerations

- Is a safety standard applicable to all domains
- Has Tool Qualification Levels (TQL)s: TQL-1 (High), TQL-5 (Low)
- TQL-Level has to be defined from domain standards

Table 2-1 Tool Qualification Level Determination

<table>
<thead>
<tr>
<th>Software Level</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>TQL-1</td>
</tr>
<tr>
<td>B</td>
<td>TQL-2</td>
</tr>
<tr>
<td>C</td>
<td>TQL-3</td>
</tr>
<tr>
<td>D</td>
<td>TQL-4</td>
</tr>
</tbody>
</table>

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

Requires
- Processes,
- Activities and
- Documents
Content

- Tool Qualification Requirements from Standards
- Tool Qualification Roadmap
  - Vision
  - DO-330
  - Concept
    - Model-based Tool Qualification
    - Roadmap Processes
    - Examples
      - Processes
      - Documents
    - Status: May 2012
- Current Demonstration Examples
- Summary
Concept for Eclipse Project QPP

- Prepares Eclipse Project for Qualifiable Plugin Projects (QPP)
- Uses a separate EMF-Model (DO-330-model) for prototyping
- Covers the complete DO-330 (bi-directional tracing)
  - How-To-Qualify-Document (with DO-IDs)
  - Generic Documents
    - Tool Development Plan
    - Tool Verification Plan
    - ...

- Is developed within WP5: Tool Qualification in Automotive Industrial Working Group, see http://wiki.eclipse.org/Auto_IWG_WP5

- Roadmap:
  - Goal: DO-330
  - Every two weeks: new steps (process for DO-330)
  - Presented and discussed in Telcos
Roadmap to the Concept/Project QPP

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

- Tool Qualification Requirements from Standards
- Tool Qualification Roadmap
  - Vision
  - DO-330
  - Eclipse Project: Qualifiable Plugin Projects (QPP)
  - Concept
    - Model Based Qualification
    - Examples
      - Processes
      - Documents
    - Status: May 2012
- Current Demonstration Examples
- Summary
Model for Tool-Requirements

- EMF-Metamodel for Tool Requirements
Planning: Analysis Model for PSAC
Test Coverage and Measurement

```java
public class AssumptionModel {

    /**
     * returns the IsAssumption for an item, ensures that assumption settings
     * are "inherited" from Tool->UseCase->Error->...
     */
    public static boolean getIsAssumption(Object item) {
        if (item instanceof Error) {
            Error err = (Error) item;
            return err.isUserAssumption() || (err.getUseCase() != null && getIsAssumption(err.getUseCase()))
                    || (err.getRestriction() != null && getIsAssumption(err.getRestriction()))
                    || (err.getCheck() != null && getIsAssumption(err.getCheck()));
        }

        if (item instanceof Check) {
            Check ck = (Check) item;
            return ck.isUserAssumption() || getIsAssumption(ck.getUseCase());
        }

        if (item instanceof Qualification) {
            Qualification qual = (Qualification) item;
            return qual.isUserAssumption() || (qual.getUseCase() != null && getIsAssumption(qual.getUseCase()))
                    || (qual.getTool() != null && getIsAssumption(qual.getTool()));
        }

        if (item instanceof Restriction) {
            Restriction res = (Restriction) item;
            return res.isUserAssumption() || getIsAssumption(res.getUseCase());
        }

        if (item instanceof UseCase) {
            UseCase uc = (UseCase) item;
            return uc.isUserAssumption() || getIsAssumption(uc.getTool());
        }
    }
}
```
Tool Life Cycle for Qualifiable Plugins

- Combines the following processes:
  - Planning (TORs)
  - Development (TR, LLRs)
  - Integration (Verification)
  - Configuration Management
  - Quality Assurance

- Fits to existing processes (Project process, Release Process) by extending them with a “Qualification Stage”

- The following stages are defined (and can be determined automatically from the DO-330 model) such that every release has a well-defined qualification stage:
  - **Unqualified-Pre-Alpha Release (“Undefined”):** unknown qualification state
  - **Qualification Alpha-Release (“Analyzed”):** The TORs are defined and TQL is determined
  - **Qualification Beta-Release (“Feature-Complete”):** All requirements (TORs and TRs) are described and have traces to LLRs and Code
  - **Qualification Release Candidate (“Verification Defined”):** All required verification steps are defined. No open bugs of the category “Blocker” are available.
  - **Qualification Release:** (“Successfully Verified”) Verification has been successfully executed and are documented within the qualification kit

- Transition Criteria are formally defined, based on the DO-330 model
Configuration Management

- Configuration Items are all elements within the Qualifiable Eclipse Project
  - Sources
  - Architecture
  - DO-330-model
    - Requirements (TORs, TRs,
    - Tracing
    - ....

- Two Control Categories: CC1, CC2. Item’s CC depends on TQL

<table>
<thead>
<tr>
<th>Tool Operational Requirements Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>5.1.1.a</td>
</tr>
<tr>
<td>5.1.2.a</td>
</tr>
<tr>
<td>Tool Operational Requirements</td>
</tr>
</tbody>
</table>

- Definition of Control Categories (DO-330):

Table 7-1 TCM Process Activites Associated with CC1 and CC2 Data

<table>
<thead>
<tr>
<th>TCM Process Activity</th>
<th>Reference</th>
<th>CC1</th>
<th>CC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Identification</td>
<td>7.2.1</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Baselines</td>
<td>7.2.2.a</td>
<td>7.2.2.b</td>
<td>7.2.2.c</td>
</tr>
<tr>
<td>Traceability</td>
<td>7.2.2.f</td>
<td>7.2.2.g</td>
<td>•</td>
</tr>
<tr>
<td>Change Review</td>
<td>7.2.5</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

Example: TORs changes have to be reviewed for TQL-1 to TQL-4 but not for TQL-5

Plugin Extension has to know this (Transition Criteria!)
11.1.d (Additional Considerations for Multi-Function Tools) states:
if multi-function tools both produce and verify the same output
– Protection shall be used (plugins) AND
– (for TQL-1 and TQL-2) Independence

Example

Proposed Eclipse Tool Development Plan:
3.2 Multi-Function Tools:
If common dependencies are used their TQL should be higher than the TQLs of the using parts (satisfies: DO-330-11.1.d)
Qualification Liaison Process

- For all tools with qualification need
- Demonstrate that the tools conform to their requirements ("TOR"), even if qualification shows errors
Content

- Tool Qualification Requirements from Standards
- Tool Qualification Roadmap
  - Vision
  - DO-330
  - Concept
    - Model Based Qualification
    - Examples
      - Processes
      - Documents
    - Status: May 2012
- Current Demonstration Examples
- Summary
1. **Goals:** DO-330
2. **Concept:** Eclipse Project QPP
3. **Demonstrate & implement** QPP
4. **Qualify (selected) plugins**

### Status May 2012

- **Goals & Requirements:** Identified
- **Elaborate Concept for QPP:** Status: Feasible and 100% of DO-330 covered
- **Demonstrate & Implement QPP:** Ready to start
- **Qualify Plugins with QPP:** Ready to start (Prototyping)

**Summary:** Qualification is feasible and qualification (based on current prototype) could be started now
Content

- Tool Qualification Requirements from Standards
- Tool Qualification Roadmap
  - Vision
  - DO-330
  - Concept
    - Model Based Qualification
    - Examples
      - Processes
      - Documents
    - Status: May 2012

- Current Demonstration Examples

- Summary
Demonstration / Examples

- Tool/Process Analysis & TCL Determination in Progress (work in progress)

- VALIDAS Tool Chain Analyzer: DO-330 Modeling of Analysis, TCL, Requirements, Design, Tests including document templates (for generation)

- Virtual Vehicle Real-time system configuration tool for AUTOSAR: DO-330 model and implementation examples

Generate Documents
Content

- Tool Qualification Requirements from Standards
- Tool Qualification Roadmap
  - Vision
  - DO-330
  - Concept
    - Model Based Qualification
    - Examples
      - Processes
      - Documents
    - Status: May 2012
- Current Demonstration Examples
- Summary
Extended Eclipse (QPP) will support qualification including
- Classification: Tool Analysis -> Planning Process
- Qualification: Process & Model for qualifiable plugin projects
- Usage: Fulfill assumptions and apply qualification kits

Applicable to all relevant standards (ISO 26262, IEC 61508, DO-178C, EN 50128,..)

Metadata extension for qualification information of plugins: DO-330 model

Much work in progress
- Tracing to “How-To-Qualify” document
- Modeling: gaps to current meta-information
- Create documentations (TDP,TVP,TQP,TQR..)

First, second, third, fourth, fifth steps performed

Proposed new role for that work: Eclipse Validator

Many areas of DO-330 already covered
Thank You!

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