virtual vehicle

VALIDAS

Oscar Slotosch & Mario Driussi

Tool Develoment According to a Safety Standard

Content

- **Motivation**
- DO-330 Requirements
- DO-330 Qualification Model
- Demonstrator
- Eclipse Roadmap
- QPP
- Summary

Motivation

- **Modern software development: More**
  - Tools
  - Risks
  - Confidence Needs
  - Tool Qualification
- **Different standards with different tool requirements**
  - ISO 26262: Tool Confidence Levels: TCL 1, TCL 2, TCL 3
  - IEC 61508: Tool Classes: T1, T2, T3
  - DO-178C: Criteria: 1, 2, 3
- **Different Qualification Methods**

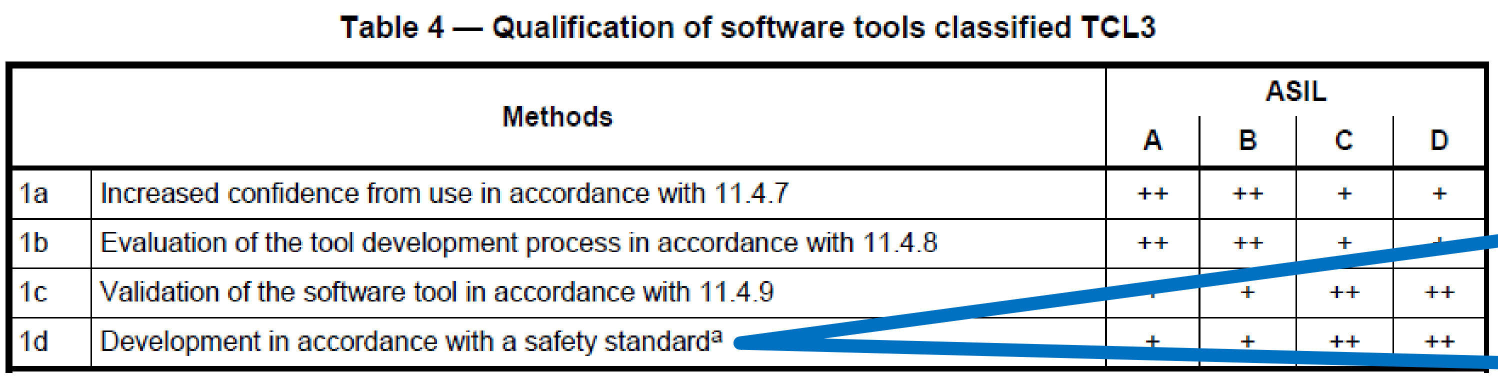


Table 4 — Qualification of software tools classified TCL3

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]	+	+	++	++

a No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.
EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178.

- **Challenges**
  - Technical: Qualify Eclipse platform
  - Organizational: Combine different communities
  - Economical: tool qualification for open source software “Pay per Qualification”?

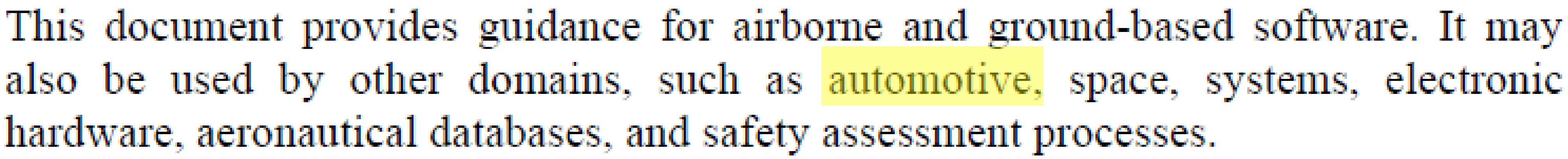
Three weeks later DO-330 was published

Content

- Motivation
- **DO-330 Requirements**
- DO-330 Qualification Model
- Demonstrator
- Eclipse Roadmap
- QPP
- Summary

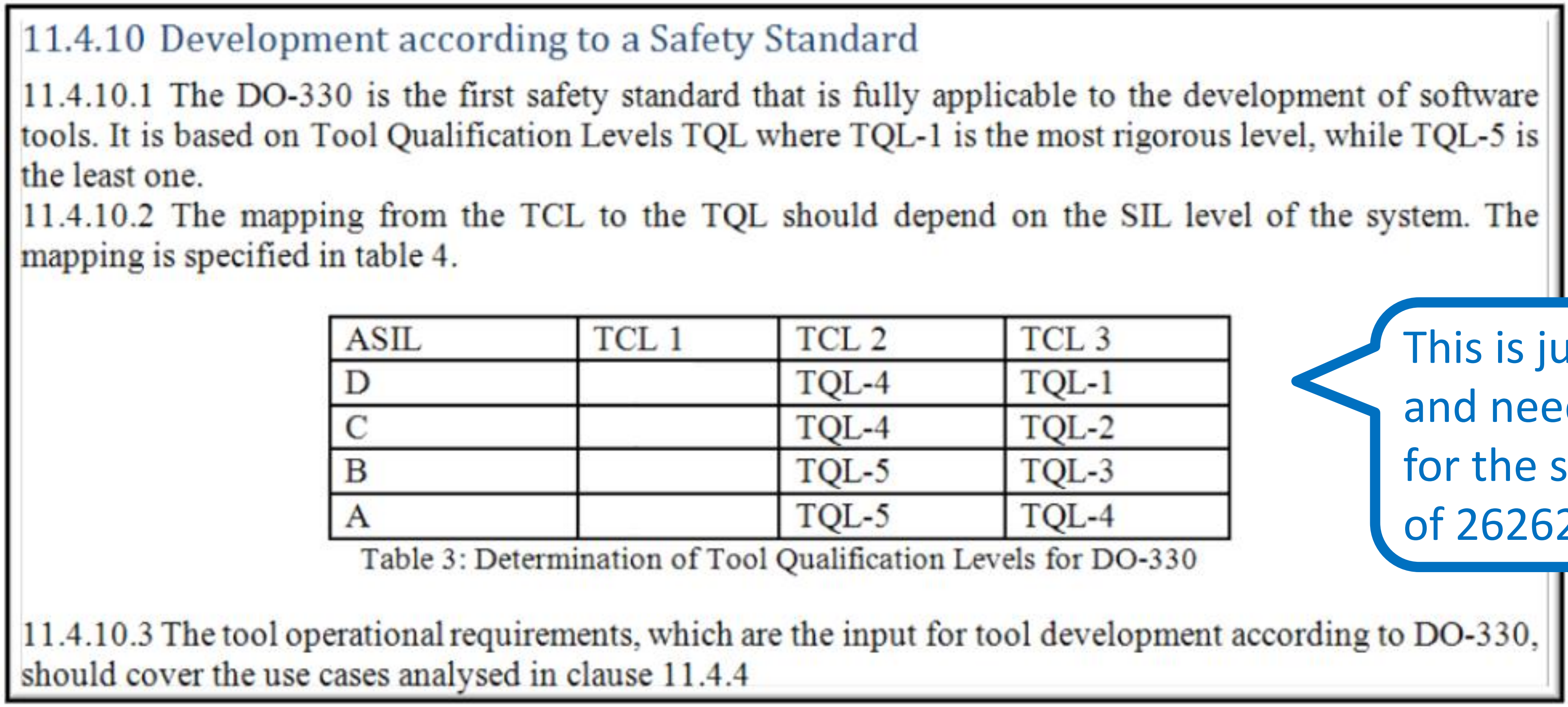
DO-330 & Application Domains

DO-330-1.2.c:



This document provides guidance for airborne and ground-based software. It may also be used by other domains, such as automotive, space, systems, electronic hardware, aeronautical databases, and safety assessment processes.

- **DO-330 defines "Tool Qualification Level" (TQL) from 1 (HIGH) to 5 (LOW)**
- **Integration of DO-330 into ISO 26262 could look like (similar for IEC61508,..):**



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-4	TQL-1
C		TQL-4	TQL-2
B		TQL-5	TQL-3
A		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

This is just a proposal, and needs confirmation for the second edition of 26262

- **Similar chapters exist in DO-178C and DO-278A**

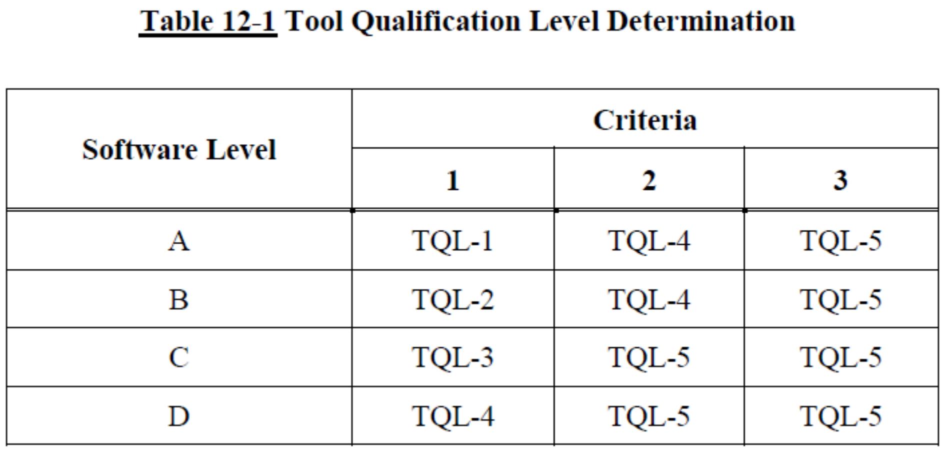
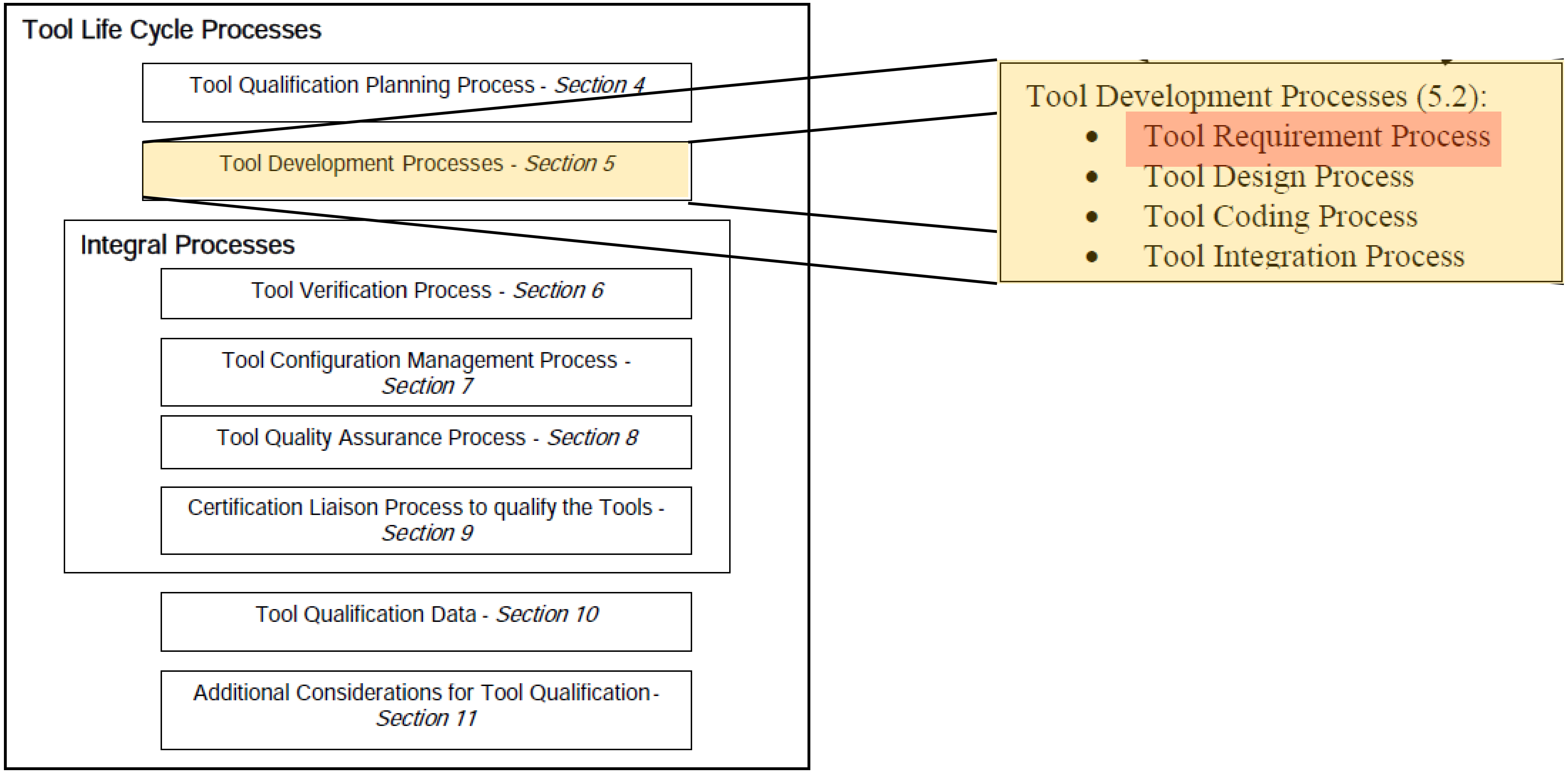


Table 12-1 Tool Qualification Level Determination

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

DO-330 Structure (Example)

- **Structure of DO-330**



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*

Tool Development Processes (5.2):

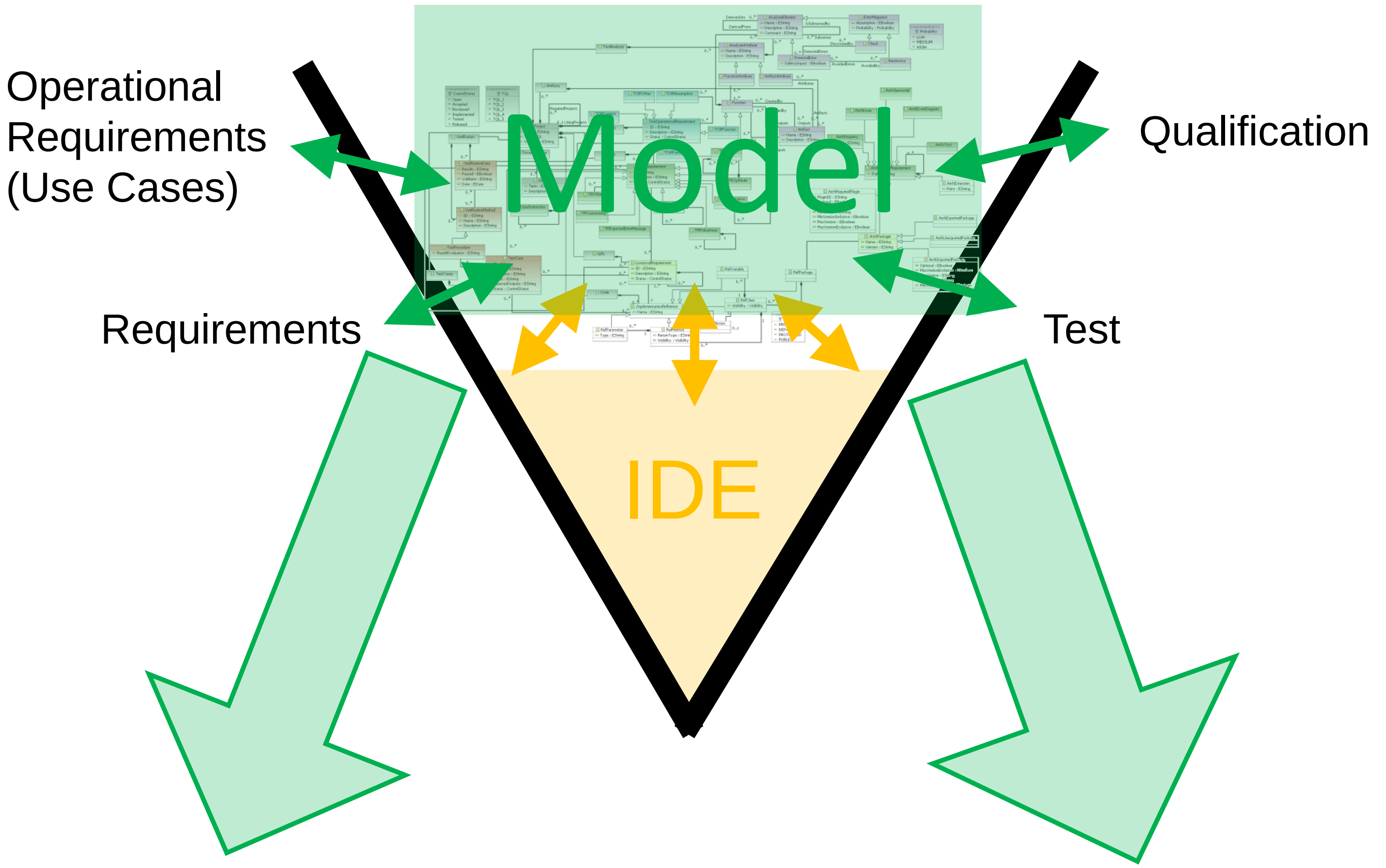
- Tool Requirement Process
- Tool Design Process
- Tool Coding Process
- Tool Integration Process

Content

- Motivation
- DO-330 Requirements
- **DO-330 Qualification Model**
- Demonstrator
- Eclipse Roadmap
- QPP
- Summary

Model-Based Tool Development

- **Model supports developer**
  - Analyses
  - Consistency
  - Completeness
- **Documentation of the model**
  - How-To Qualify model-based tools according DO-330
  - Tool Development Plan
  - Tool Verification Plan
- **Compliance to DO-330**
  - Bidirectional tracing between
    - Model documentation
    - DO-330
  - Satisfies all 450 DO-330 requirements



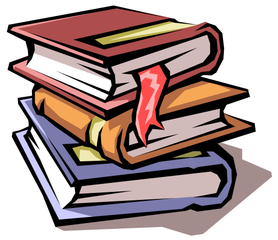
User Guidance

Analyses to determine

- Criticality
- TQL
- Development State
- Open Issues
- Missing Links & Tests
- Maturity
- …

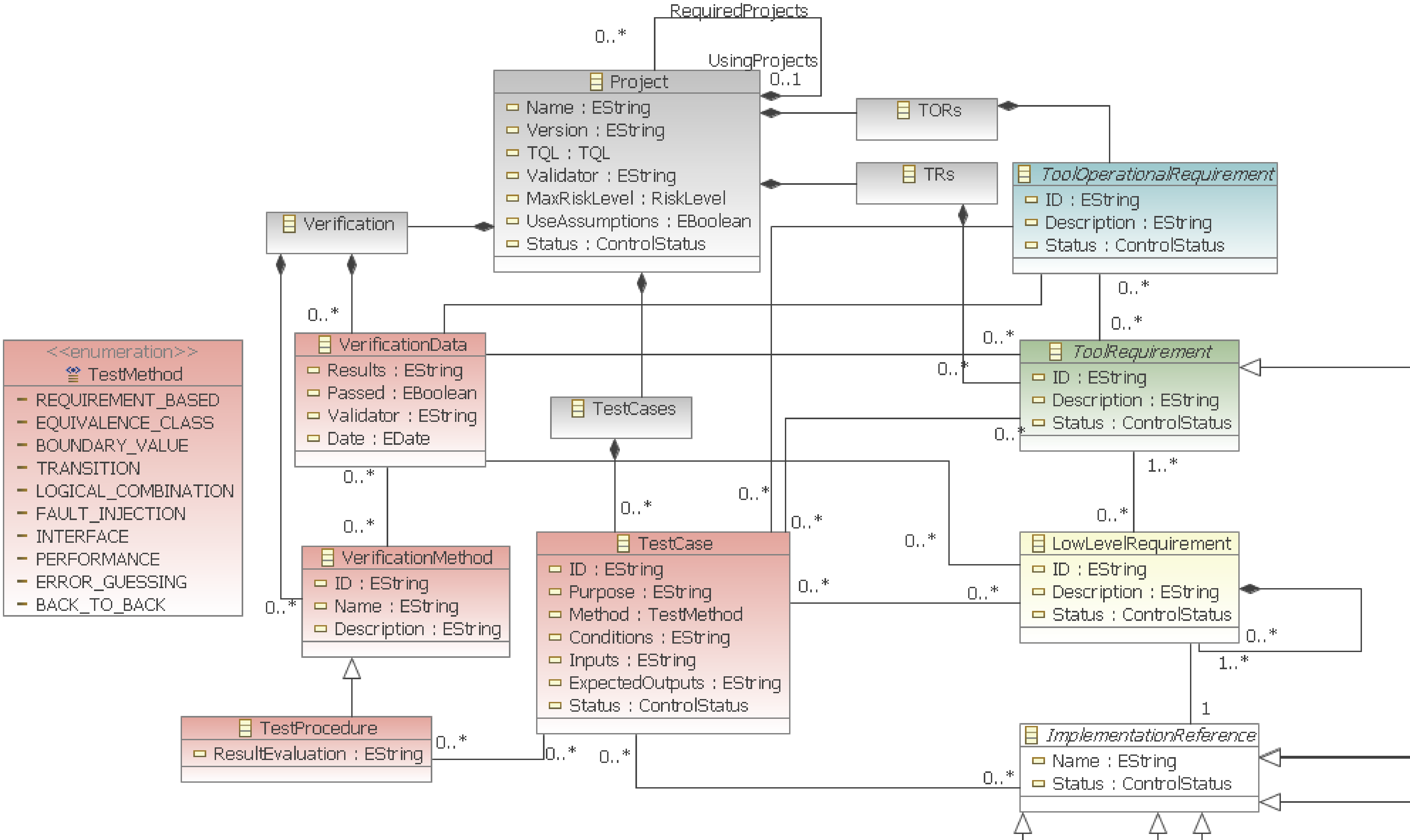
Generated Documents

- Requirements-Specification
- Design-Specification
- Test-Specification
- Tool Analysis (TCL/PSAC)
- …



Example: Test & Verification Model

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



RequiredProjects
0..*
UsingProjects
0..1

Project
Name : EString
Version : EString
TQL : TQL
Validator : EString
MaxRiskLevel : RiskLevel
UseAssumptions : EBoolean
Status : ControlStatus

TORs

TRs

ToolOperationalRequirement
ID : EString
Description : EString
Status : ControlStatus

Verification

VerificationData
Results : EString
Passed : EBoolean
Validator : EString
Date : EDate

TestCases

ToolRequirement
ID : EString
Description : EString
Status : ControlStatus

<<enumeration>>
TestMethod
REQUIREMENT_BASED
EQUIVALENCE_CLASS
BOUNDARY_VALUE
TRANSITION
LOGICAL_COMBINATION
FAULT_INJECTION
INTERFACE
PERFORMANCE
ERROR_GUESSING
BACK_TO_BACK

VerificationMethod
ID : EString
Name : EString
Description : EString

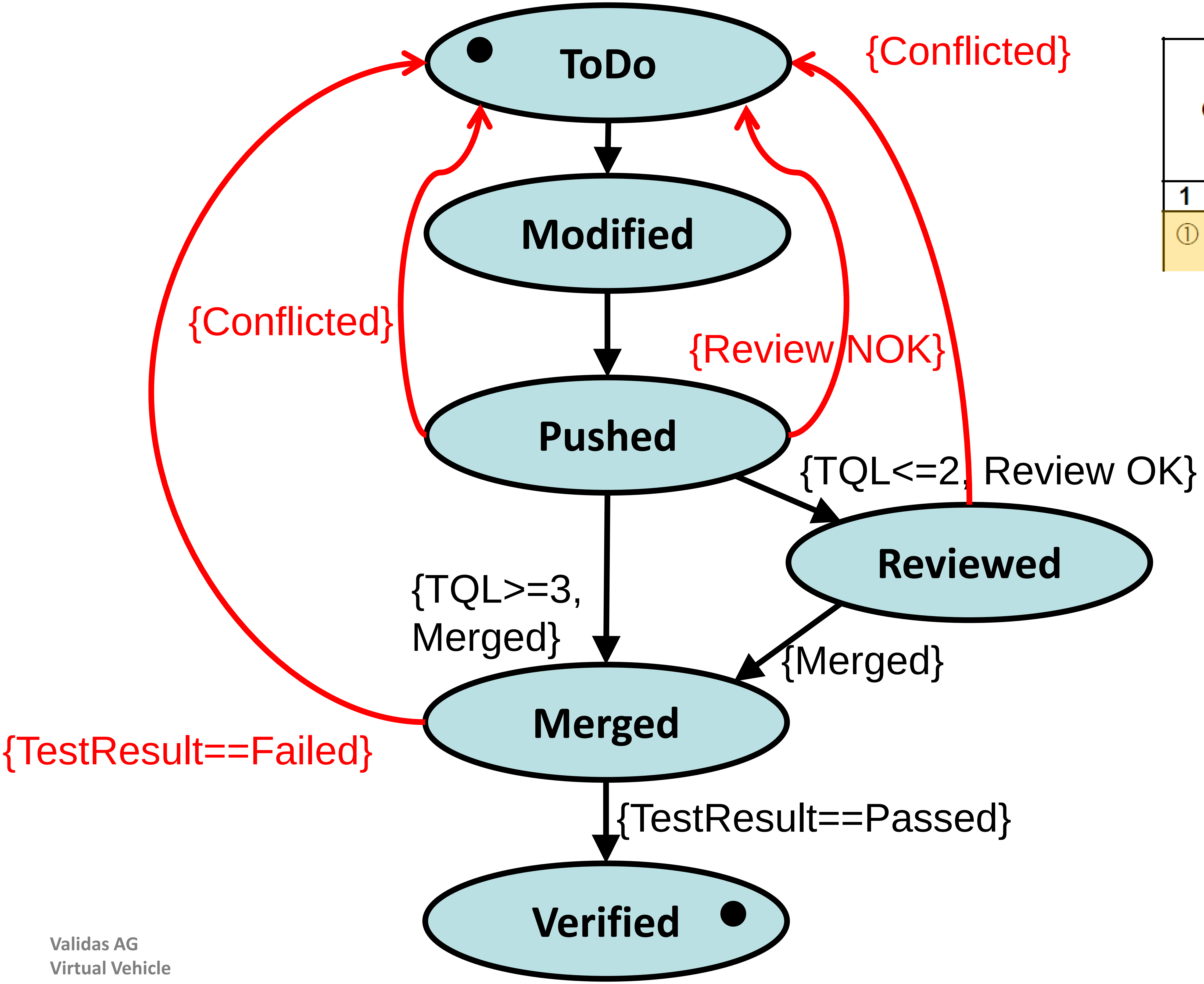
TestCase
ID : EString
Purpose : EString
Method : TestMethod
Conditions : EString
Inputs : EString
ExpectedOutputs : EString
Status : ControlStatus

LowLevelRequirement
ID : EString
Description : EString
Status : ControlStatus

TestProcedure
ResultEvaluation : EString

ImplementationReference
Name : EString
Status : ControlStatus

Control Status of Items

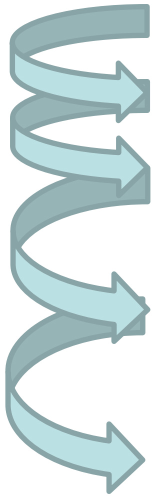


Control Category by TQL				
1	2	3	4	5
①	①	②	②	

TQL-5:
Status is
Private
(unchecked)

Tool Life Cycle “Maturity” for Tools

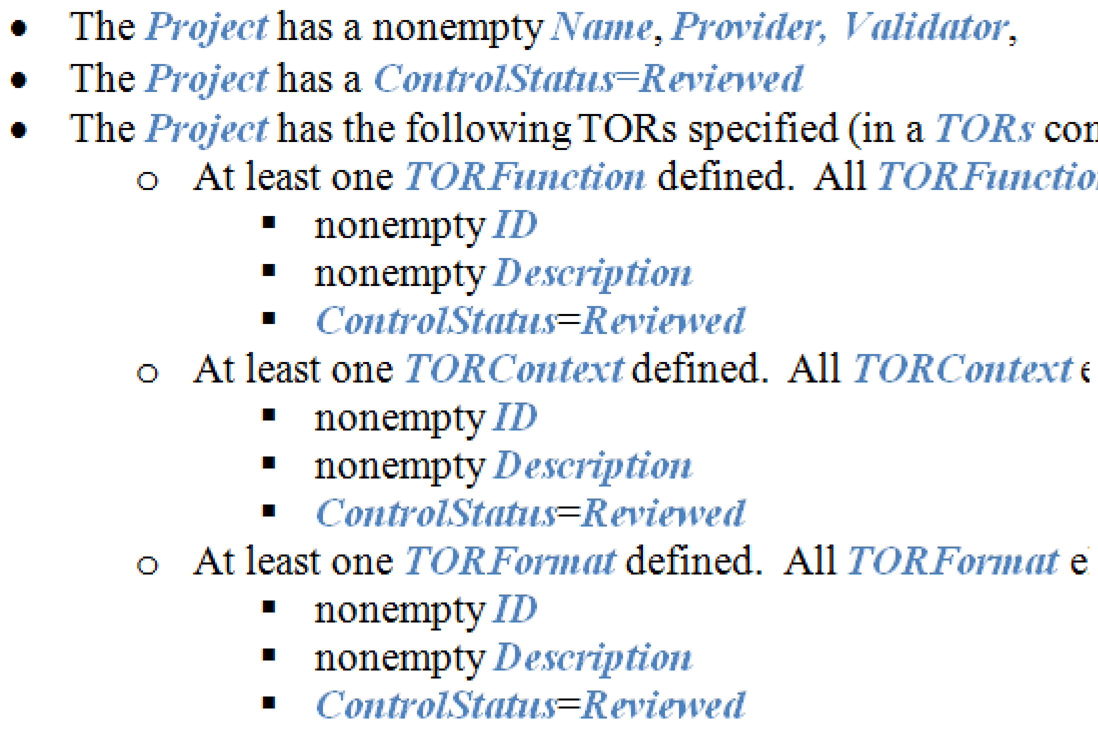
- **Combines the following DO-330 processes:**
  - Planning (TORs)
  - Development (TR, LLRs)
  - Integration (Verification)
  - Configuration Management
  - Quality Assurance
- **Fits to existing development processes (Project process, Release Process) by extending them with a “Qualification Stage”**
- **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**



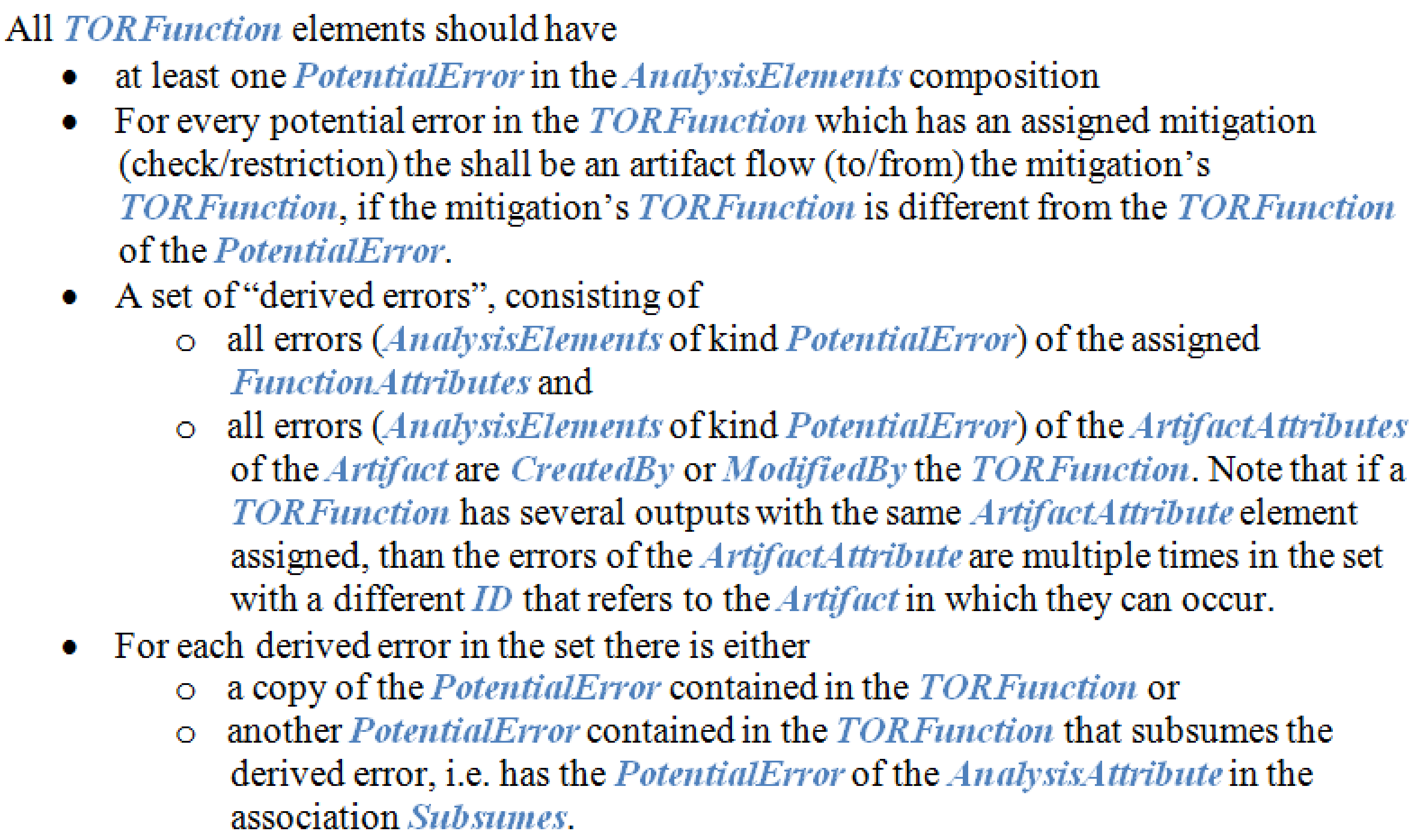
  - **Unqualified-Pre-Alpha Release** (“**Undefined**”): unknown qualification state
  - **Qualification Alpha-Release** (“**Analyzed**”): The TORs are defined and TQL is determined
  - **Qualification Beta-Release** (“**Feature-Complete**”): All requirements (TORs and TRs) are described and have traces 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**

Tool Life Cycle Transition Criteria

- Defined in the “Tool Development Plan”
- Required by DO-330-4.2.1, DO-330-4.2.2, DO-330-4.3.b
- Quite formal definition (can be checked automatically) based on the DO-330 model of the tool
- Example (truncated): Transition to Qualification Alpha State (“Analyzed”)



- The *Project* has a nonempty *Name*, *Provider, Validator*,
- The *Project* has a *ControlStatus=Reviewed*
- The *Project* has the following TORs specified (in a *TORs* container):
    - At least one *TORFunction* defined. All *TORFunction* elements have
        - nonempty *ID*
        - nonempty *Description*
        - *ControlStatus*=*Reviewed*
    - At least one *TORContext* defined. All *TORContext* e
        - nonempty *ID*
        - nonempty *Description*
        - *ControlStatus*=*Reviewed*
    - At least one *TORFormat* defined. All *TORFormat* e
        - nonempty *ID*
        - nonempty *Description*
        - *ControlStatus*=*Reviewed*



All *TORFunction* elements should have

- at least one *PotentialError* in the *AnalysisElements* composition
- For every potential error in the *TORFunction* which has an assigned mitigation (check/restriction) the shall be an artifact flow (to/from) the mitigation’s *TORFunction*, if the mitigation’s *TORFunction* is different from the *TORFunction* of the *PotentialError*.
- A set of “derived errors”, consisting of
    - all errors (*AnalysisElements* of kind *PotentialError*) of the assigned *FunctionAttributes* and
    - all errors (*AnalysisElements* of kind *PotentialError*) of the *ArtifactAttributes* of the *Artifact* are *CreatedBy* or *ModifiedBy* the *TORFunction*. Note that if a *TORFunction* has several outputs with the same *ArtifactAttribute* element assigned, than the errors of the *ArtifactAttribute* are multiple times in the set with a different *ID* that refers to the *Artifact* in which they can occur.
- For each derived error in the set there is either
    - a copy of the *PotentialError* contained in the *TORFunction* or
    - another *PotentialError* contained in the *TORFunction* that subsumes the derived error, i.e. has the *PotentialError* of the *AnalysisAttribute* in the association *Subsumes*.

Content

- Motivation
- DO-330 Requirements
- DO-330 Qualification Model
- **Demonstrator**
- Eclipse Roadmap
- QPP
- Summary

Goals: Eat your own Dog Food

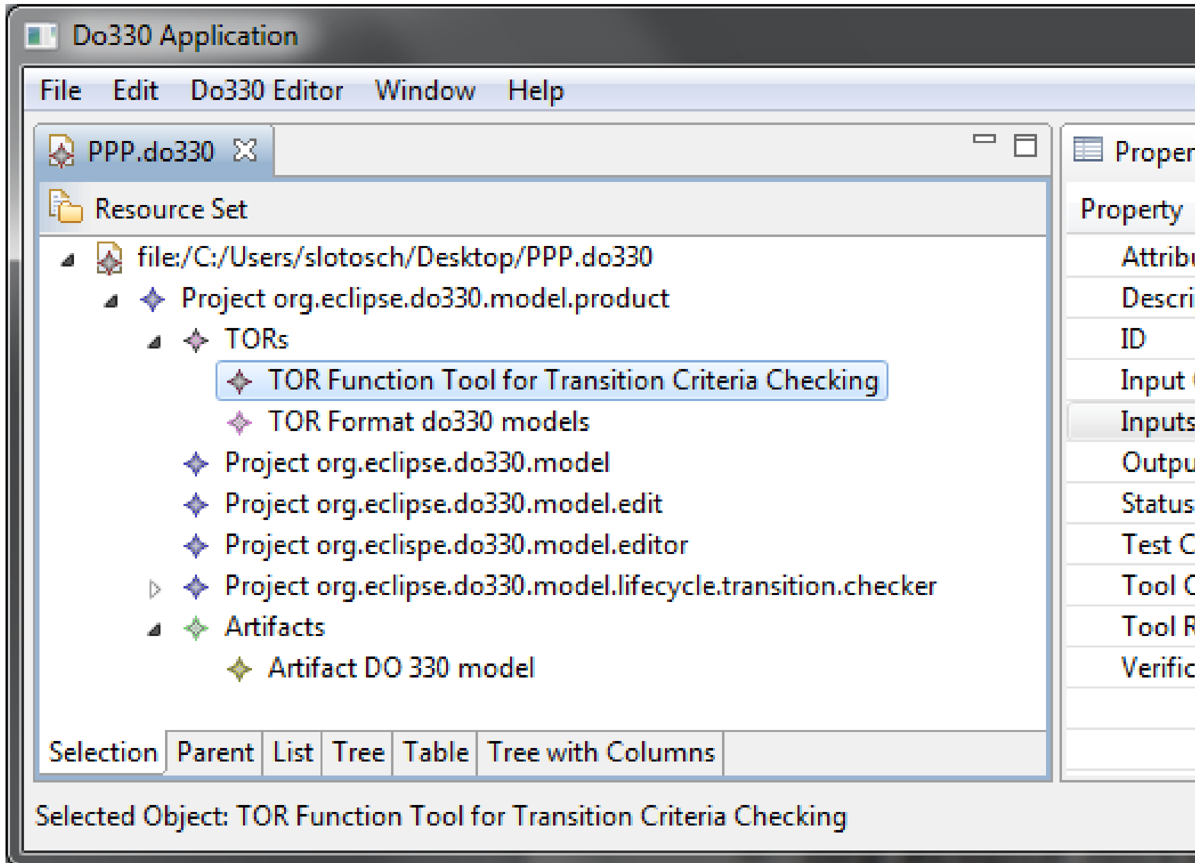
- **Demonstrate the concept**
- **Refine the concept**
- **Start a prototype for DO-330 qualification**
  - Can be used to qualify any tool according to DO-330
  - Can be integrated into Eclipse (QPP)
- **First use case (TORFunction):**
  - Compute the qualification state of a product based on the model as described in Tool Development Plan (Life cycle process)
- **First tool functions (TRFunction)**
  - Validator for the model
  - Derived tool functions: Edit, Load & Save models
- **Steps (monitor effort):**
  - Build a team (“Tool Provider”, “Validators”,…)
  - Set up the project (Eclipse, git, gerrit, bugzilla, DO-330 model)
  - Implement the tool
  - Qualify the tool
- **Milestones: see later slides**

This makes it applicable also to other tools

Well-defined (and small) problem

First Milestones

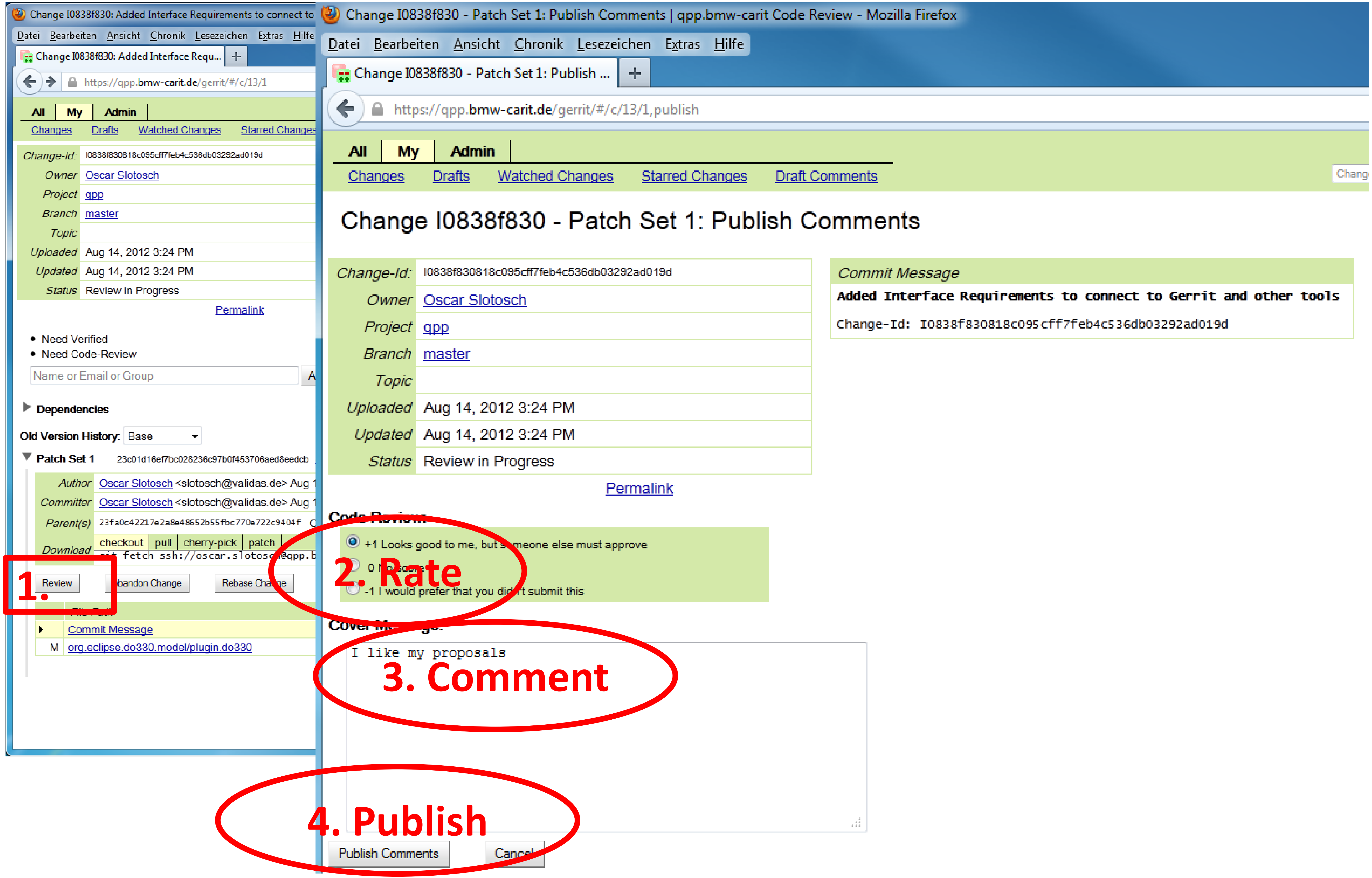
- ✓ **M1: Initial team and process (status reports as part of WP5 telcos) defined**
  - Tool Providers: BMW-CarIT, Validas
  - Validators: Validas, BMW-CarIT
- ✓ **M2: Set up the repository with the following plugins**
  - “**model**”: org.eclipse.do330.model: the do-330 model
  - “**edit**”: org.eclipse.do330.model.edit: the generated edit
  - “**editor**”: org.eclipse.do330.model.editor: the generated editor
  - “**checker**”: org.eclipse.do330.model.lifecycle.transition.checker: checker
  - “**checker.ui**”: org.eclipse.do330.model.lifecycle.transition.checker.ui: the checker’s UI
  - “**product**”: org.eclipse.do330.model.product: product for the prototype
- ✓ **M3: Create DO-330 model files for plugins**
- ✓ **M4: Create TORs for each plugin in the DO-330 model**
  - Review them and model this using “VerificationData” elements
- ▶ **M5: Determine TQLs for each plugin**
- ▶ **M6: Reach Qualification Alpha State for all plugin models (manual check)**



- file:/C:/Users/slotosch/Desktop/PPP.do330
  - Project org.eclipse.do330.model.product
    - TORs
      - TOR Function Tool for Transition Criteria Checking
      - TOR Format do330 models
    - Project org.eclipse.do330.model
    - Project org.eclipse.do330.model.edit
    - Project org.eclispe.do330.model.editor
    - Project org.eclipse.do330.model.lifecycle.transition.checker
    - Artifacts
      - Artifact DO 330 model

Selected Object: TOR Function Tool for Transition Criteria Checking

Example: Review using Gerrit



Change I0838f830: Added Interface Requirements to connect to

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Change I0838f830: Added Interface Requ...
All My Admin
Changes Drafts Watched Changes Starred Changes

https://qpp.bmw-carit.de/gerrit/#/c/13/1

Change-Id:	I0838f830818c095cff7feb4c536db03292ad019d
Owner	Oscar Slotosch
Project	qpp
Branch	master
Topic	
Uploaded	Aug 14, 2012 3:24 PM
Updated	Aug 14, 2012 3:24 PM
Status	Review in Progress

Permalink

- Need Verified
- Need Code-Review

Name or Email or Group

▶ Dependencies

Old Version History: Base

▼ Patch Set 1 23c01d16ef7bc028236c97b0f453706aed8eedcb

Author	Oscar Slotosch <slotosch@validas.de> Aug 1
Committer	Oscar Slotosch <slotosch@validas.de> Aug 1
Parent(s)	23fa0c42217e2a8e48652b55fbc770e722c9404f
Download	checkout | pull | cherry-pick | patch git fetch ssh://oscar.slotosch@qpp.b

1. Review

	File Path
▶	Commit Message
M	org.eclipse.do330.model/plugin.do330

Change I0838f830 - Patch Set 1: Publish Comments | qpp.bmw-carit Code Review - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Change I0838f830 - Patch Set 1: Publish ...
All My Admin
Changes Drafts Watched Changes Starred Changes Draft Comments

https://qpp.bmw-carit.de/gerrit/#/c/13/1,publish

Change I0838f830 - Patch Set 1: Publish Comments

Change-Id:	I0838f830818c095cff7feb4c536db03292ad019d
Owner	Oscar Slotosch
Project	qpp
Branch	master
Topic	
Uploaded	Aug 14, 2012 3:24 PM
Updated	Aug 14, 2012 3:24 PM
Status	Review in Progress

Permalink

Commit Message

```
Added Interface Requirements to connect to Gerrit and other tools

Change-Id: I0838f830818c095cff7feb4c536db03292ad019d
```

Code Review:

● +1 Looks good to me, but someone else must approve

○ 0 No score

○ -1 I would prefer that you didn't submit this

2. Rate

Cover Message:

```
I like my proposals
```

3. Comment

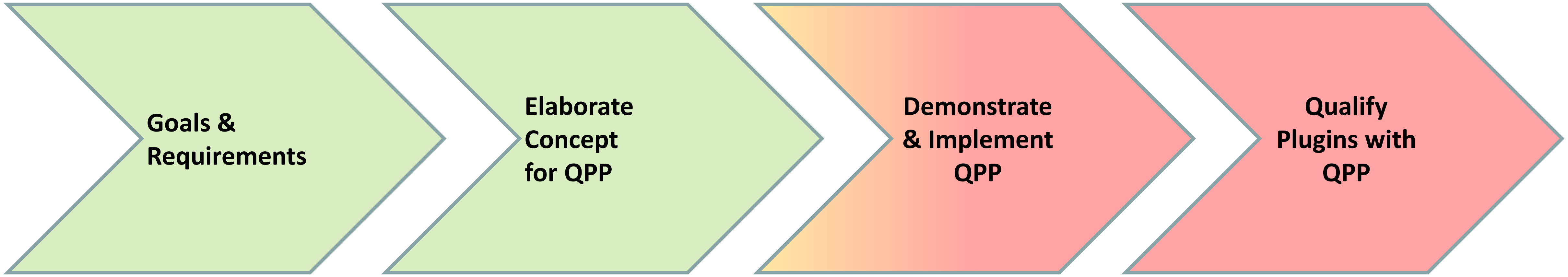
4. Publish

Content

- Motivation
- DO-330 Requirements
- DO-330 Qualification Model
- Demonstrator
- **Eclipse Roadmap**
- QPP
- Summary

Roadmap & Status

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



Identified

Status: Feasible and 100% of DO-330 covered

Started / Stalled

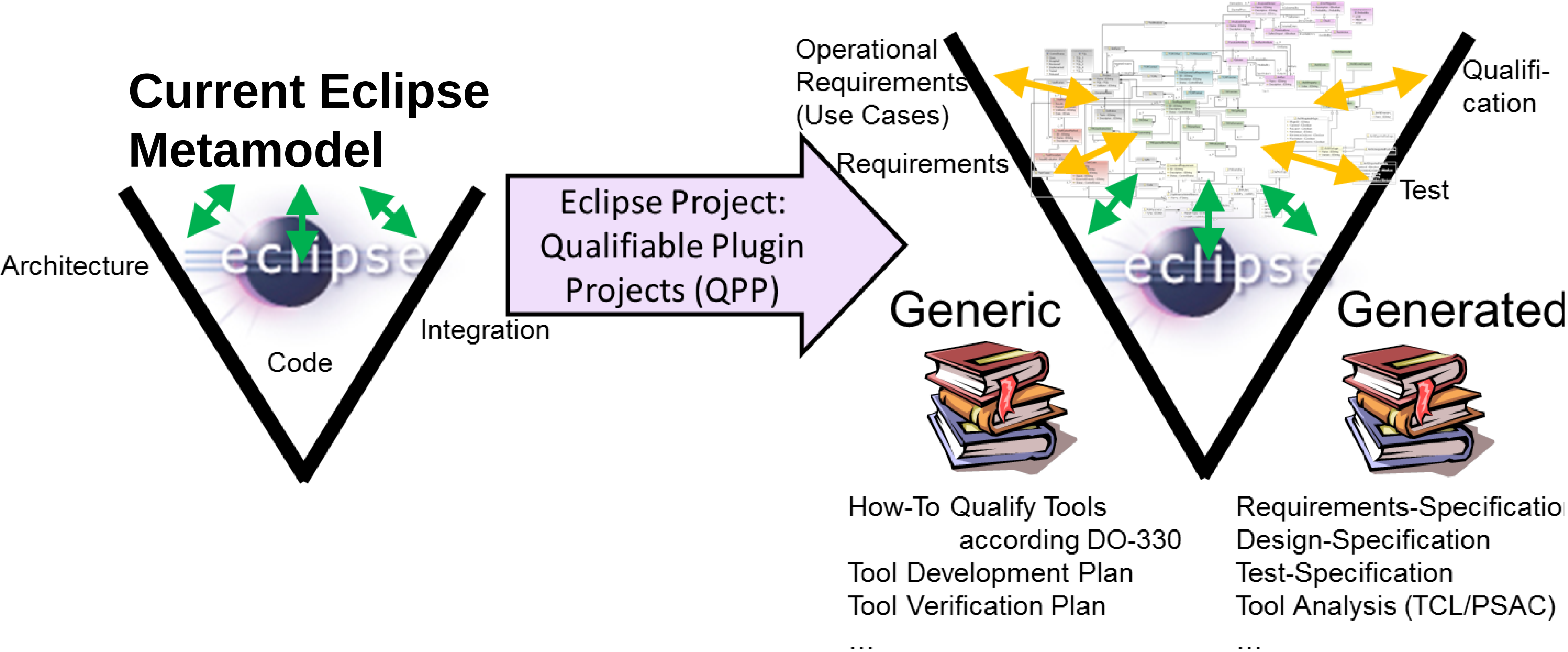
Ready to start (Prototyping)

- **Summary: Qualification is feasible and qualification (based on current prototype) has been started (Demonstrator)**

Development with Eclipse

- **Currently Eclipse does not support qualification**
- **There is a road towards tool qualification for Eclipse, see http://wiki.eclipse.org/Auto_IWG_WP5**
- **DO-330 has been selected as standard for Eclipse from**
  - Automotive IWG
  - Polarsys WG

New Extended Metamodel



Current Eclipse Metamodel

Architecture

Code

Integration

Eclipse Project: Qualifiable Plugin Projects (QPP)

Operational Requirements (Use Cases)

Requirements

Qualifi-cation

Test

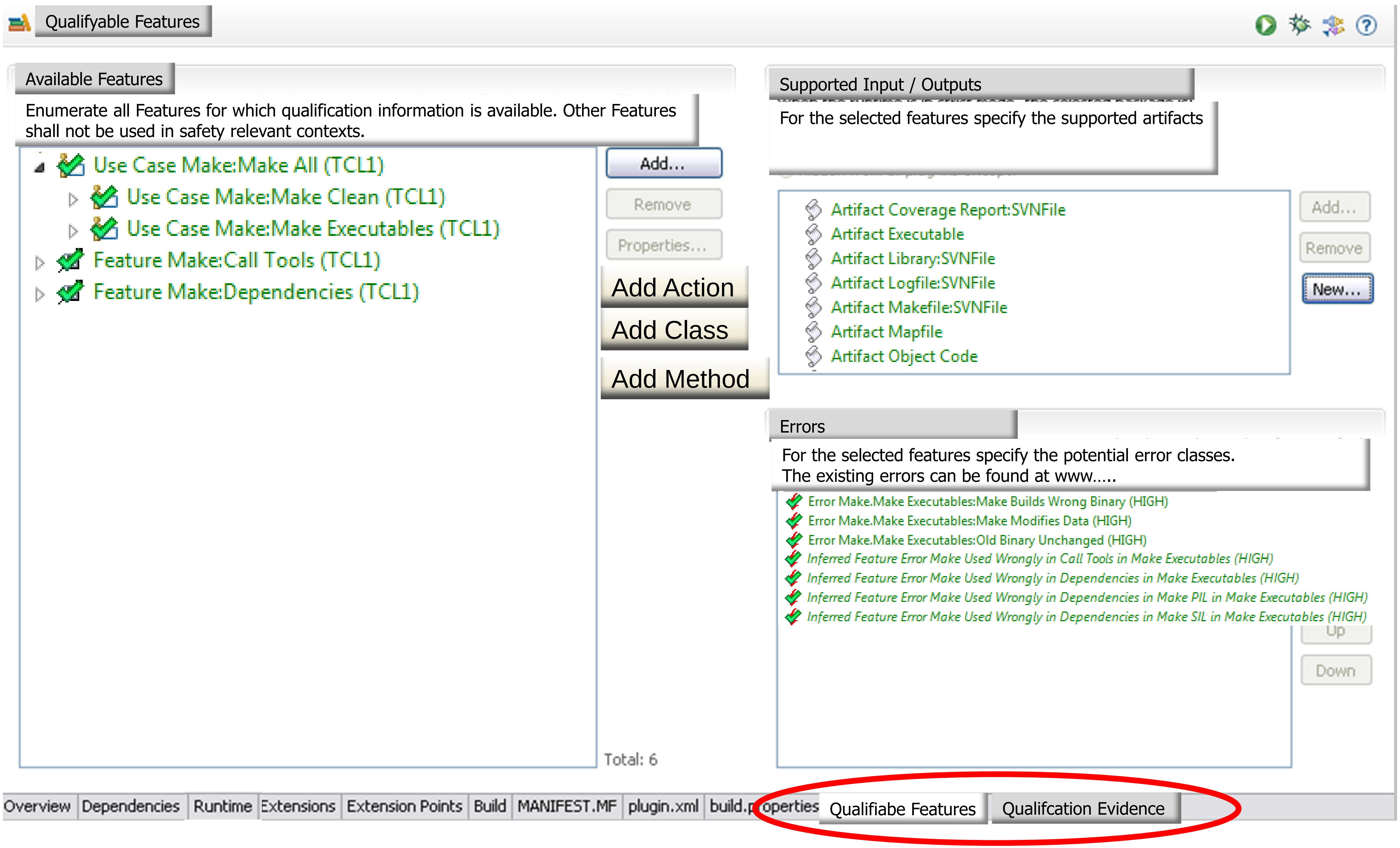
Generic

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

Generated

Requirements-Specificatio
Design-Specification
Test-Specification
Tool Analysis (TCL/PSAC)
...

Vision: Eclipse Qualification Data



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)*

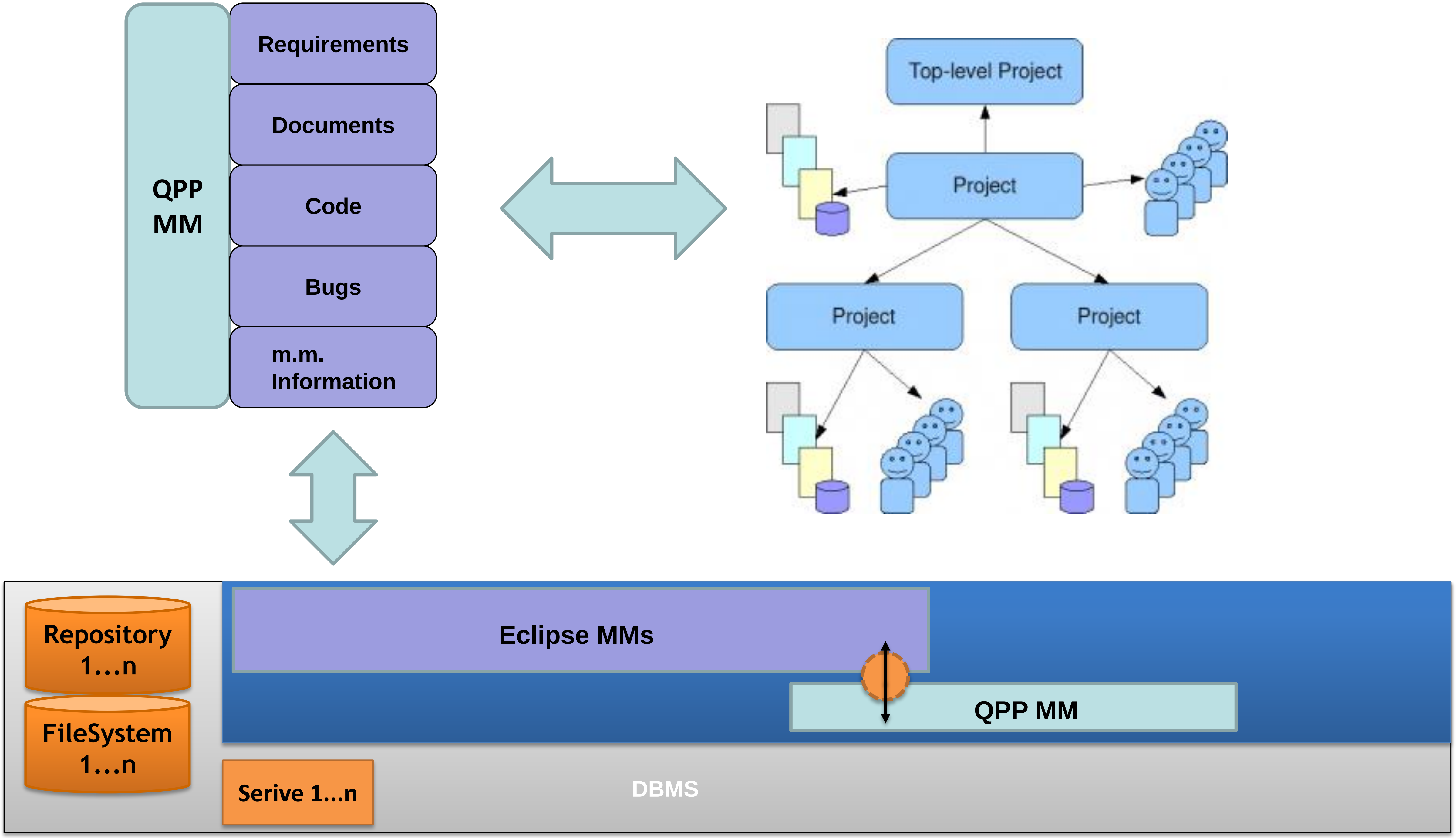
Content

- Motivation
- DO-330 Requirements
- DO-330 Qualification Model
- Demonstrator
- Eclipse Roadmap
- **QPP**
- Summary

QPP Challenges

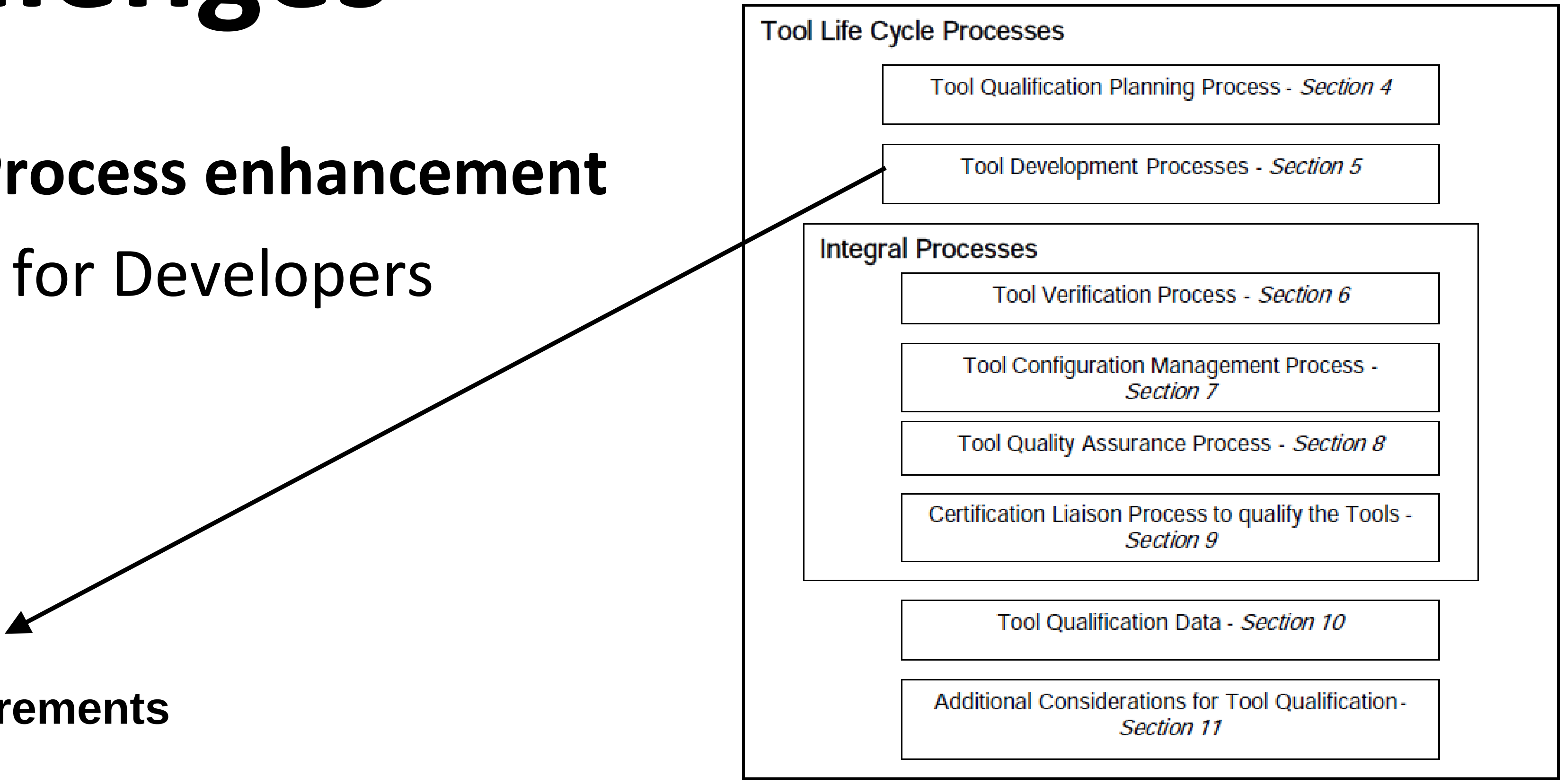
- **“Infrastructure - Connection between models”**

Eclipse Development Process



QPP Challenges

- **Workflow / Process enhancement**
  - Guidance for Developers



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*

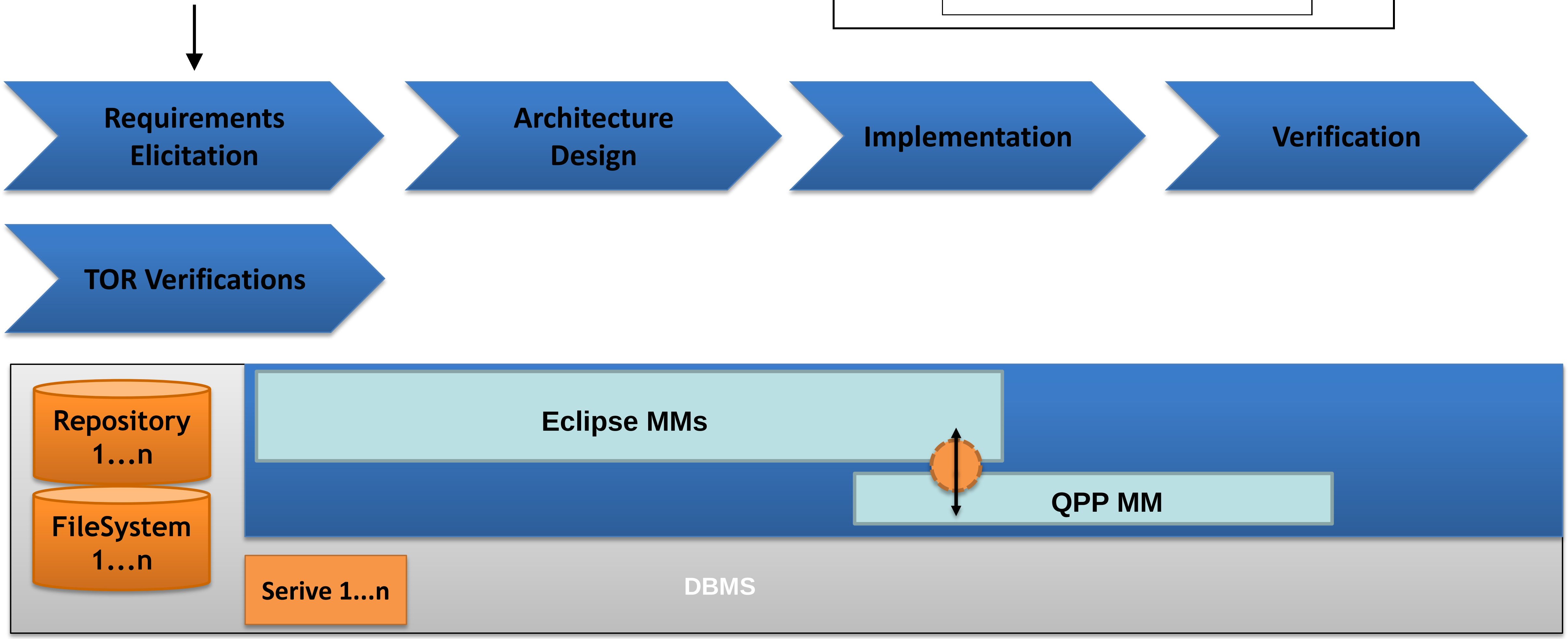
Tool Quality Assurance Process - *Section 8*

Certification Liaison Process to qualify the Tools - *Section 9*

Tool Qualification Data - *Section 10*

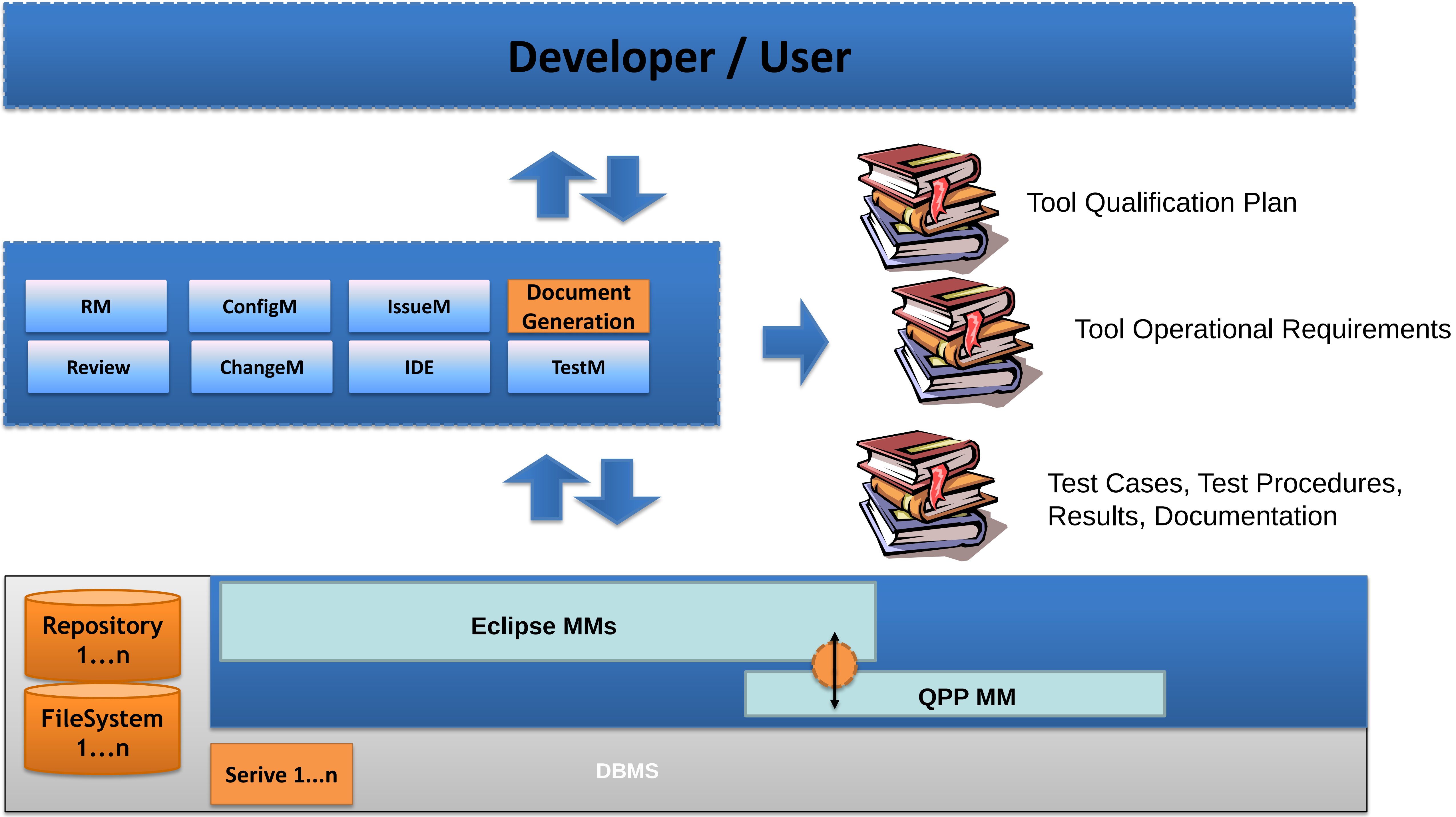
Additional Considerations for Tool Qualification - *Section 11*

Tool Operational Requirements



QPP Challenges

- **Traceability – Interfaces**



Qualifiable Plugin Process

- **Instead of Qualifiable Plugin Project**
  - No implementation project
  - No deadlines / due dates
- **Work on the roadmap “step by step”**
  - Process refinement & DO-330 compliance => research?
  - Examples / Case studies: Driven by pilot users
  - Implementation / Integration: Driven by need
- **Coordination of the steps**
  - Eclipse industrial working groups, e.g. AutoIWG WP5 Tool Qualification
  - Virtual Vehicle & Validas

Content

- Motivation
- DO-330 Requirements
- DO-330 Qualification Model
- Demonstrator
- Eclipse Roadmap
- QPP
- **Summary**

Summary

- **DO-330 is a cross-domain tool qualification standard**
- **Qualification benefits of model-based tool development**
- **Demonstrated the visionary, model-based development**
- **Eclipse-Roadmap towards qualifiable plugin projects (QPP)**
- **Challenges:**
  - ✓ Technical: roadmap concept & demonstrator
  - ✓ Organizational: Cooperation between industrial working groups
  - € Economical (open source):
    - Proposal: Pay per qualification kit application
    - Step by step: qualification infrastructure financing

Thank You!

VALIDAS

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