QUALITY ENGINEERING

OCLLib, OCLUnit, OCLDoc:
Pragmatic Extensions for the Object Constraint Language by Examples

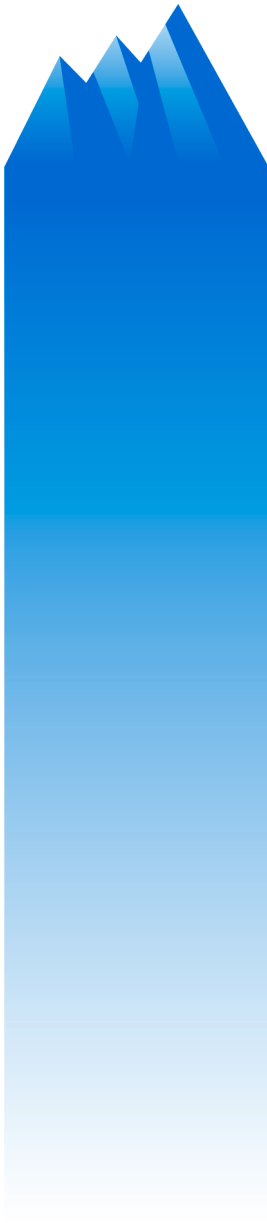
Joanna Chimiak-Opoka

University of Innsbruck, Austria

DresdenOCL – Quo Vadis?, 15th October 2009, Dresden, Germany

Agenda

- Introduction
  - Evolution of OCL
  - Challenges
- Extended Development Process
  - Usage Scenario
  - Tool Support
  - Extensions: OCLLib, OCLUnit, OCLDoc
  - Evaluation
- Summary
  - Ongoing and Future Work



Evolution of OCL

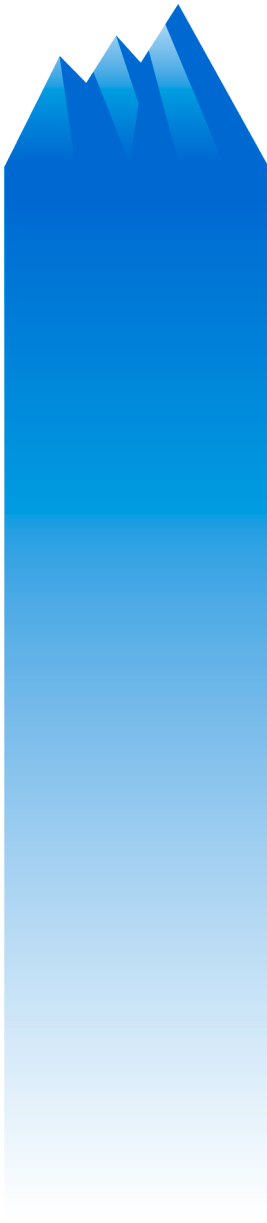
QUALITY ENGINEERING

after over 10 years of standardization and tool support

Object Constraint Language		definition aspects according to the **language engineering**		
		syntax	**semantics**	**pragmatics**
usage in the context of the **model engi-neering**	**constraints** *e.g. in the model quality assurance*			
	queries *e.g. in the model quality assessment*			

the focus of our approach

Challenges



(C1) Error–free OCL development

- reduce the number of syntax and **semantic errors**

(C2) Easy to understand OCL expressions

- correct expressions (C1) **understandable by developers and users**

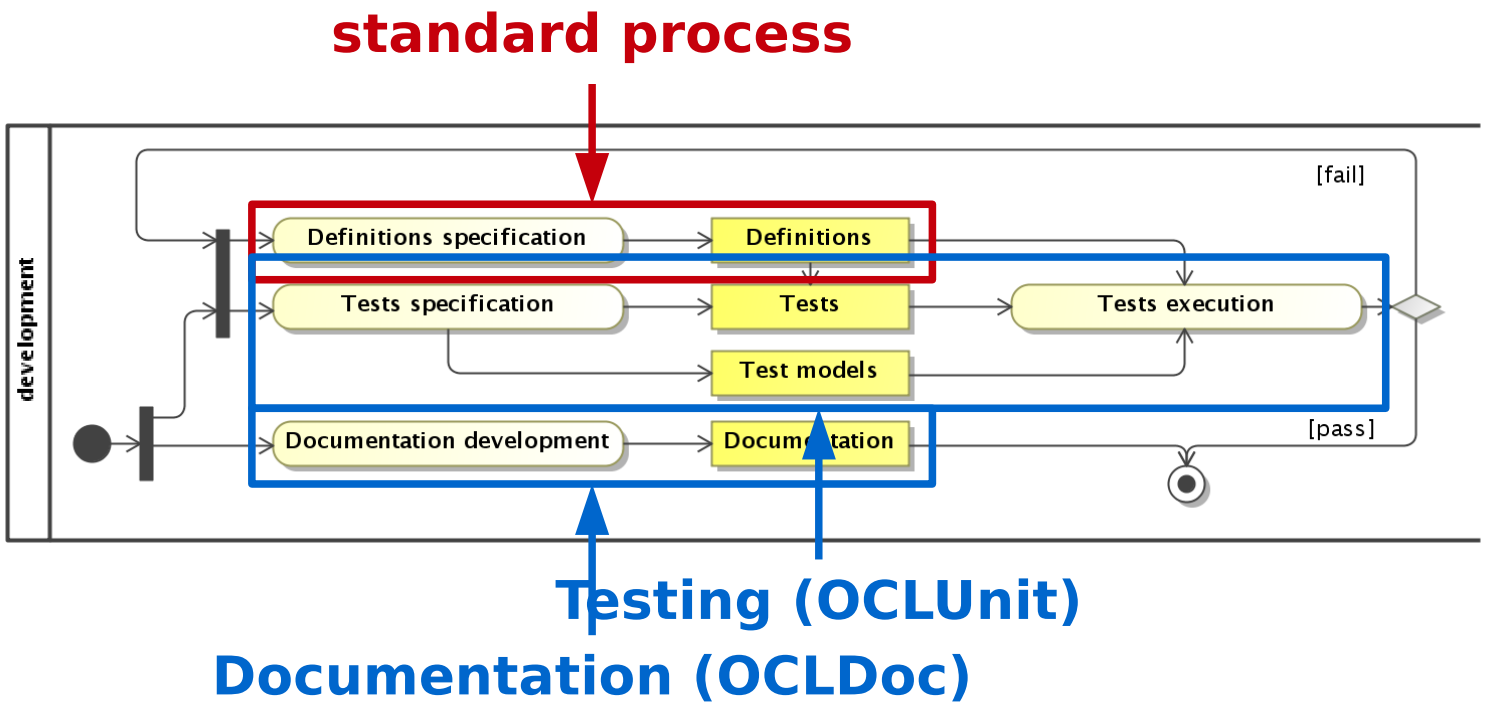
(C3) Easy and efficient OCL development

- complexity leveled by learning (C2) and development support
- stored and shared **experience knowledge**, e.g. correct OCL expressions (C1)

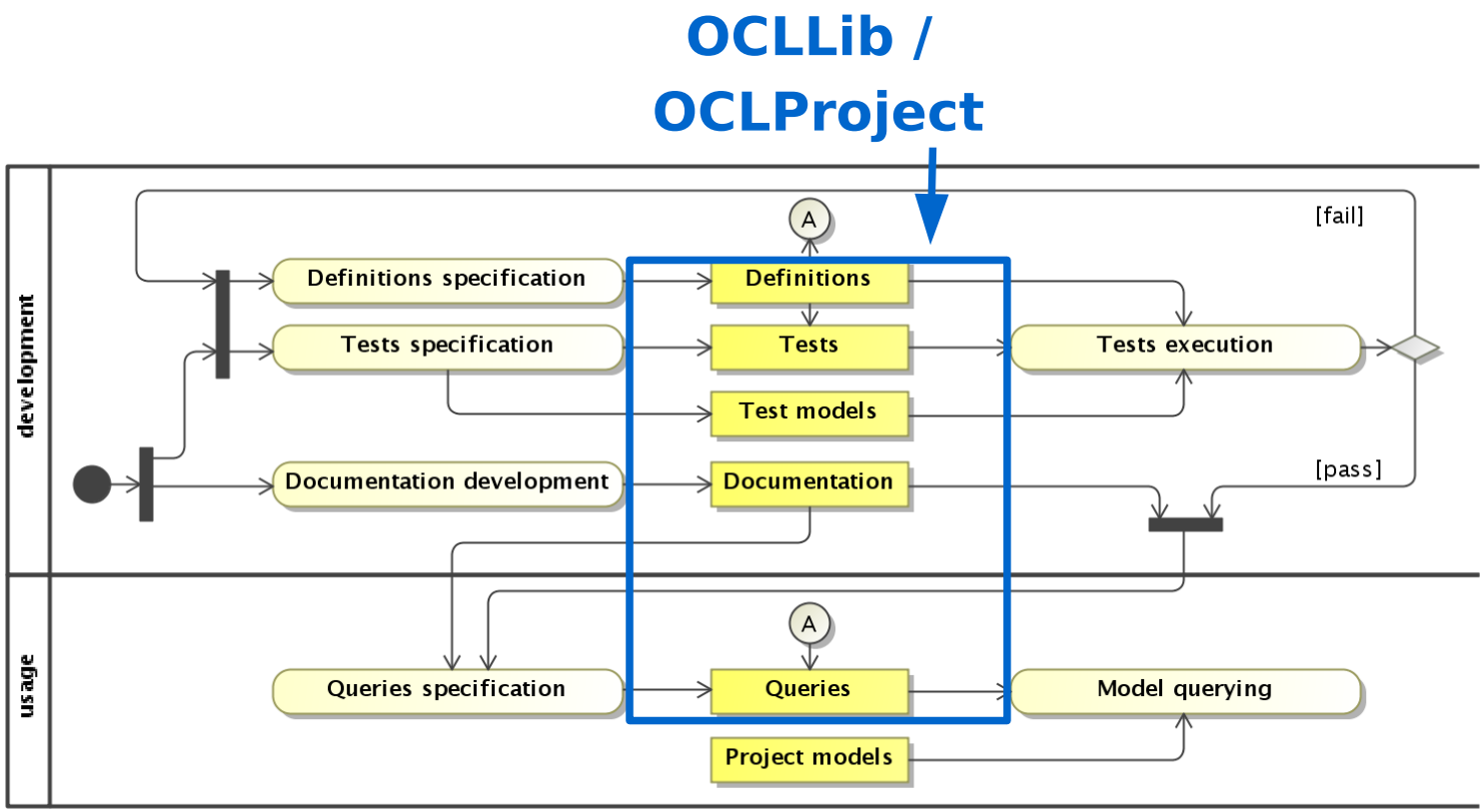
(C4) Easy to evolve OCL expressions

- the meaning of some parts of the specification can be **forgotten** → hard to evolve or refactor to cover new requirements
- introducing **new** parts or **updating** existing ones → undesired impact of other parts of a specification

Extended Development Process



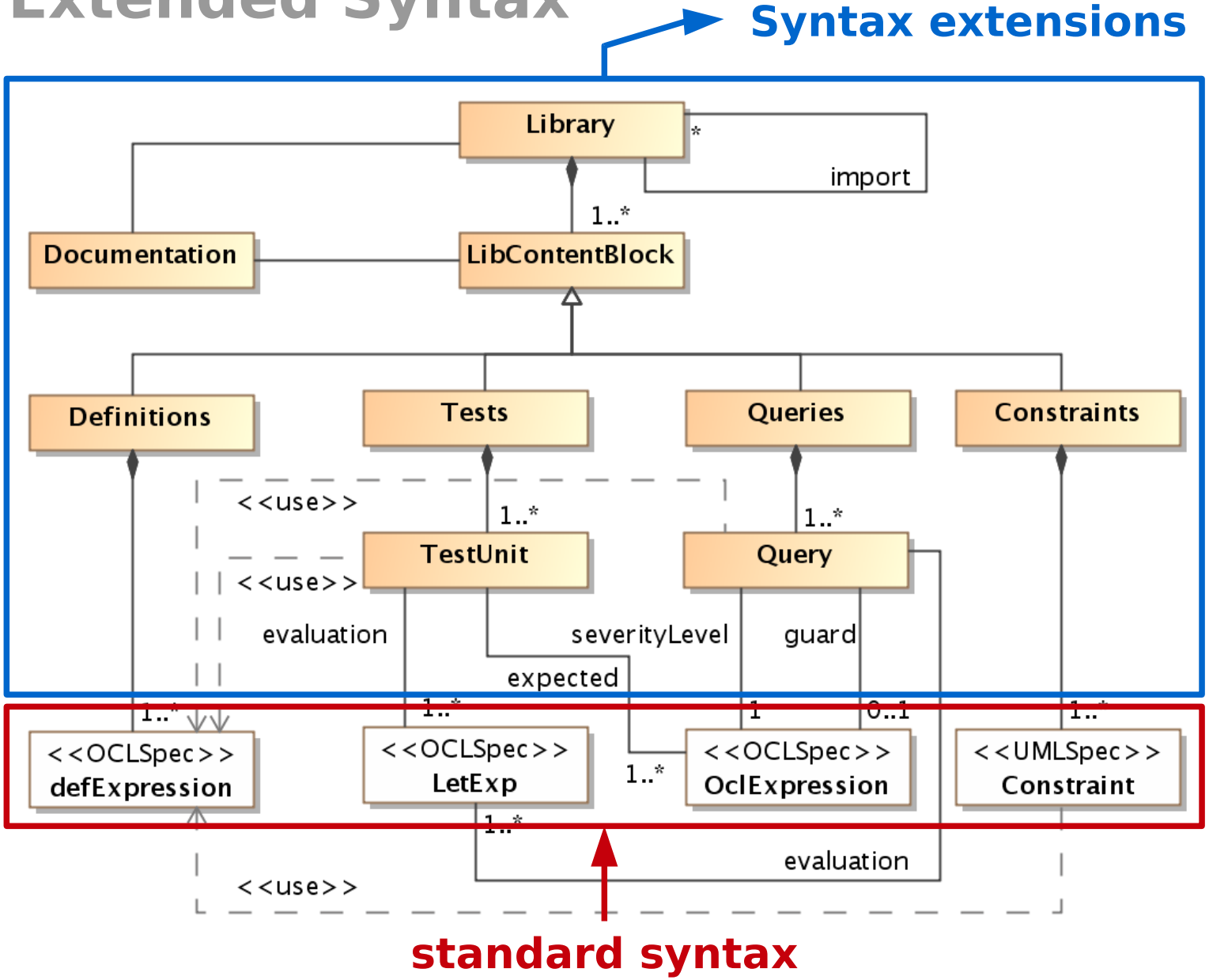
Development & Usage Process



Usage:

- within definitions, constraints, **queries**
- with **project models** not test models

Extended Syntax

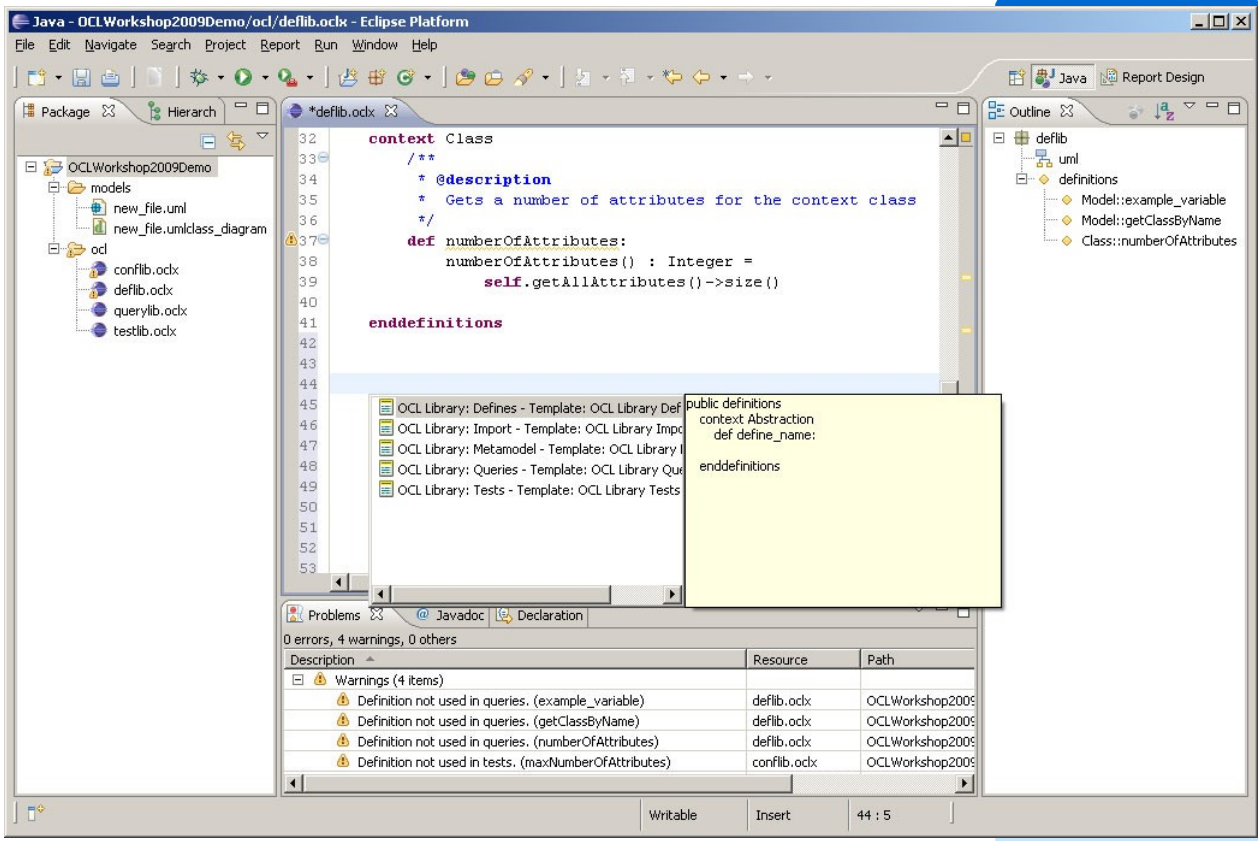


OCLEditor

QUALITY ENGINEERING

implementation

- based on EclipseOCL
- of all proposed extensions:
  - OCLLib, OCLUnit, OCLDoc
- of selected features typical for an IDE



binaries, demos and examples are available online at http://squam.info/ocleditor/

OCLLib:
Collection of OCL Expressions

QUALITY ENGINEERING

Aims and properties:

- to provide a collection of useful and reusable **OCL expressions** (C3)
- related to an underlying **metamodel** (MOF or MOF based)
- **modularized** and **parametrizable**

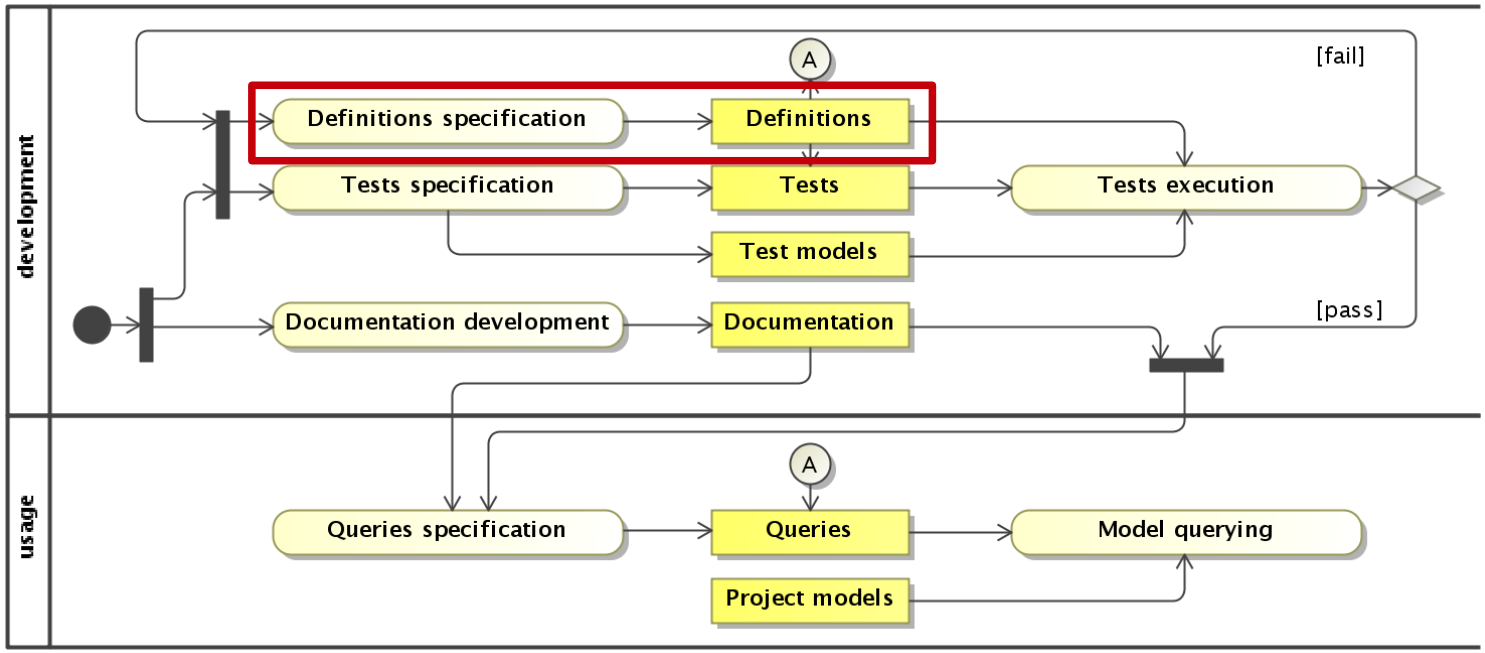
Content:

- definitions, constraints, queries, tests (OCLUnit) grouped into blocks
- documentation comments (OCLDoc)

Example:

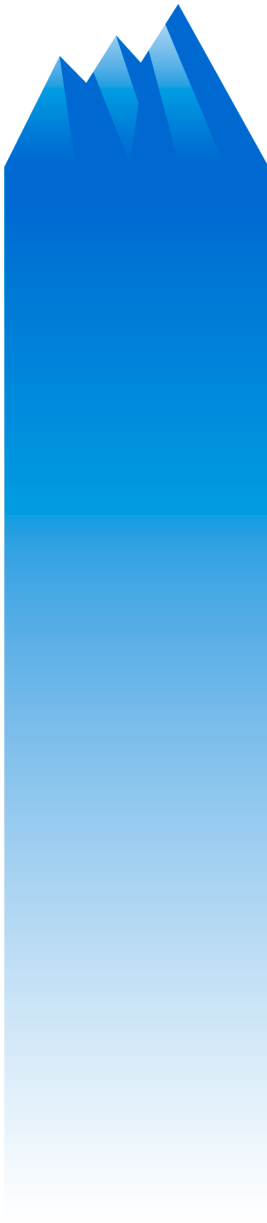
- library 1: a collection of **metrics**
- library 2: upper bounds for metrics
- library 1 **imports** library 2 as a **configuration** (an internal usage)
- libraries used by other **tools** (an external usage)

DEMO: libraries and definitions



OCLUnit:
Testing of OCL Expressions

QUALITY ENGINEERING



Aims and properties:

- to **reduce bugs** and increase confidence in correctness (C1)
- bugs need only to be found once if they are again introduced due to code changes (C4) they can be **automatically detected** with prior defined tests
- a piece of code is **usable** (C3) for anyone (else) only if it passes all available tests
- a test case is a simple scenario with a known result, and can be used to **understand** (C2) code being tested

Content:

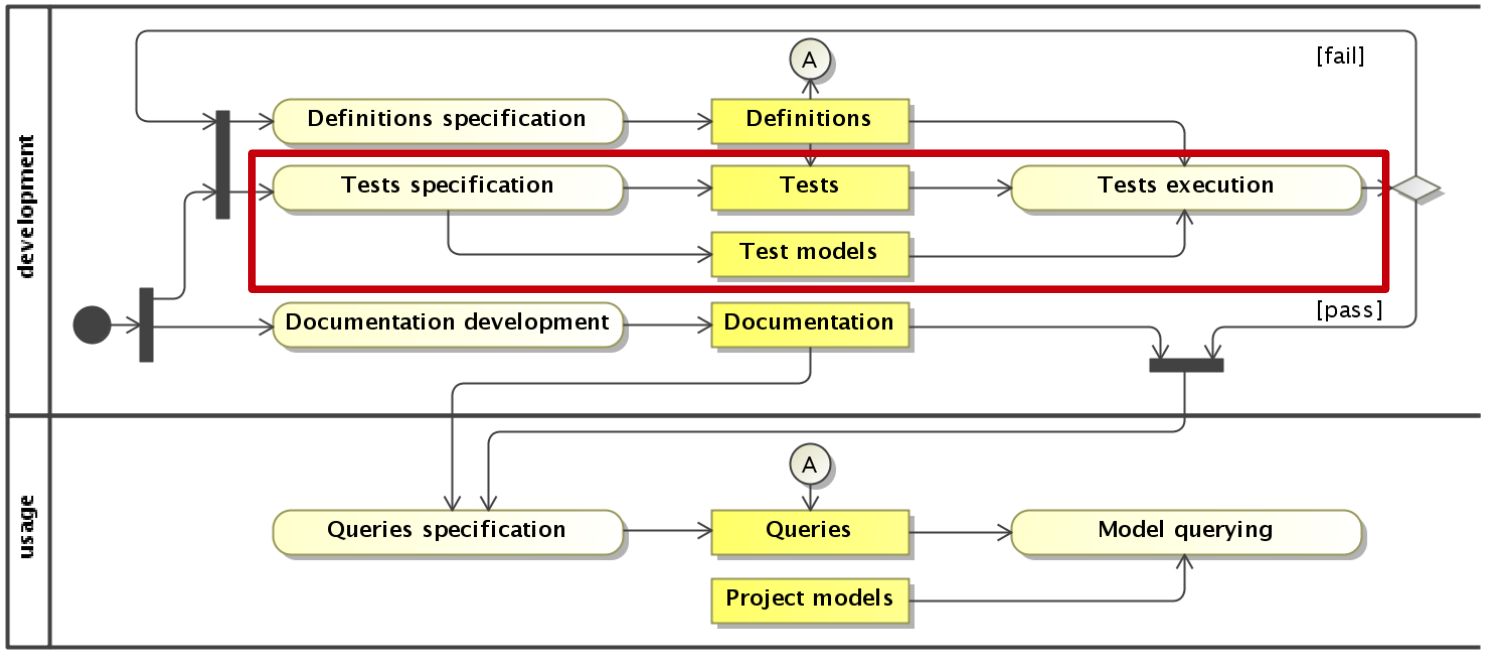
- Test units (as test cases)
- Test models (as test data)

Example:

- for every metric in in library 1:

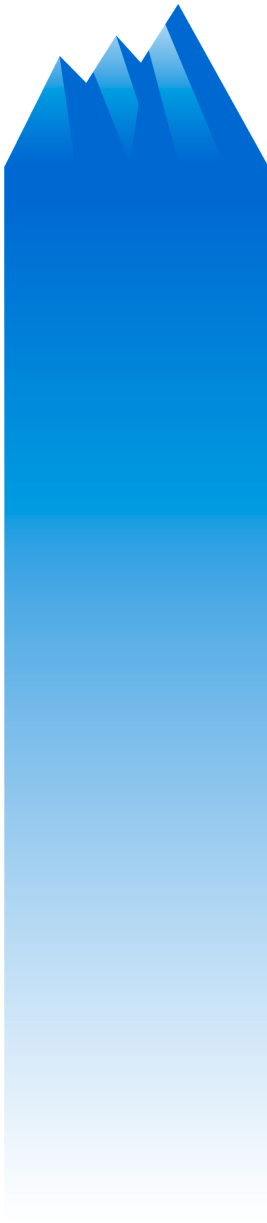
  test units are defined with **expected values** for given **test models**

DEMO: tests


development
usage
A
[fail]
Definitions specification
Definitions
Tests specification
Tests
Tests execution
Test models
Documentation development
Documentation
[pass]
A
Queries specification
Queries
Model querying
Project models

OCLDoc:
Documentation of the OCL Expressions

QUALITY ENGINEERING



Aims and properties:

- a mean to **knowledge transfer** and **communication**
- to reduce the **maintenance** burden
- to improve **productivity** by enhancing **reusability**
- **up–to–date** as it is **generated** out of source code comments

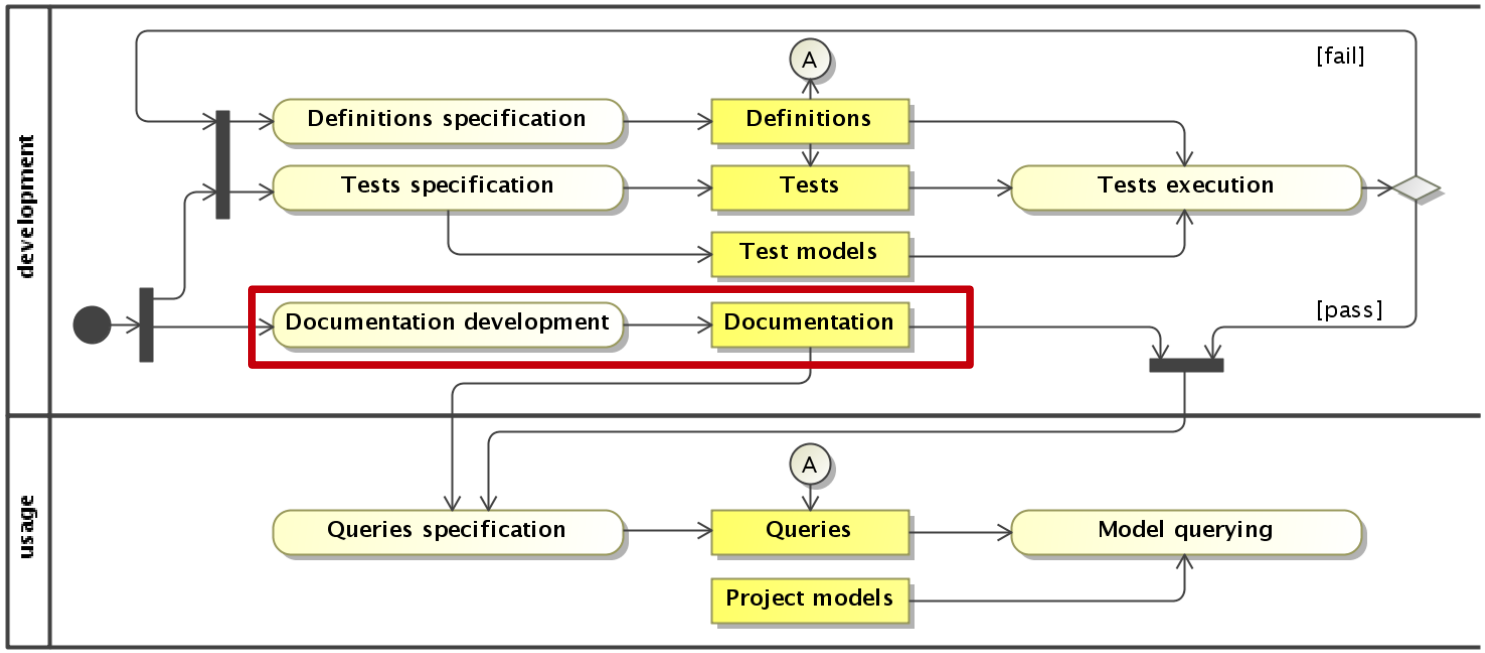
Content:

- tagged comments

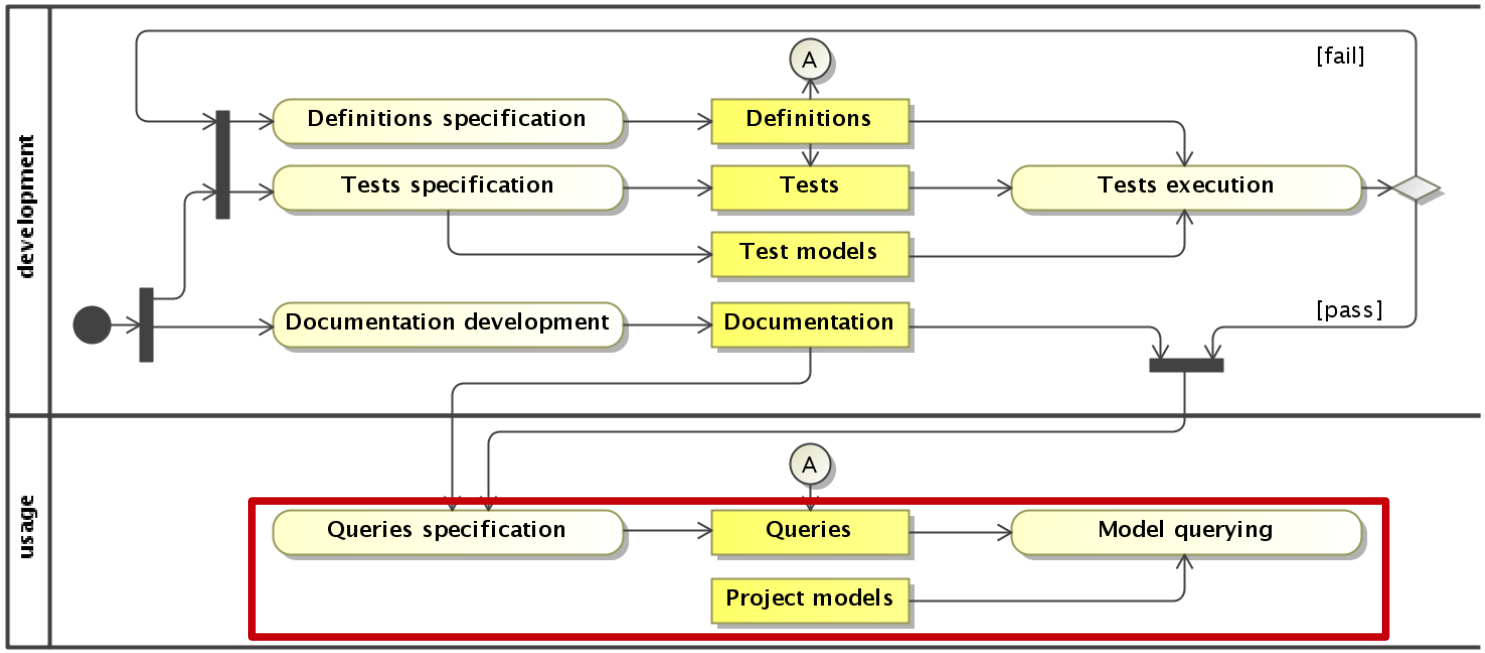
Example:

- a documentation comment is added to every metric in library 1 and provides information about
  - **its author,**
  - **parameters and**
  - **a description with its informal definition**

DEMO: documentation



DEMO: queries

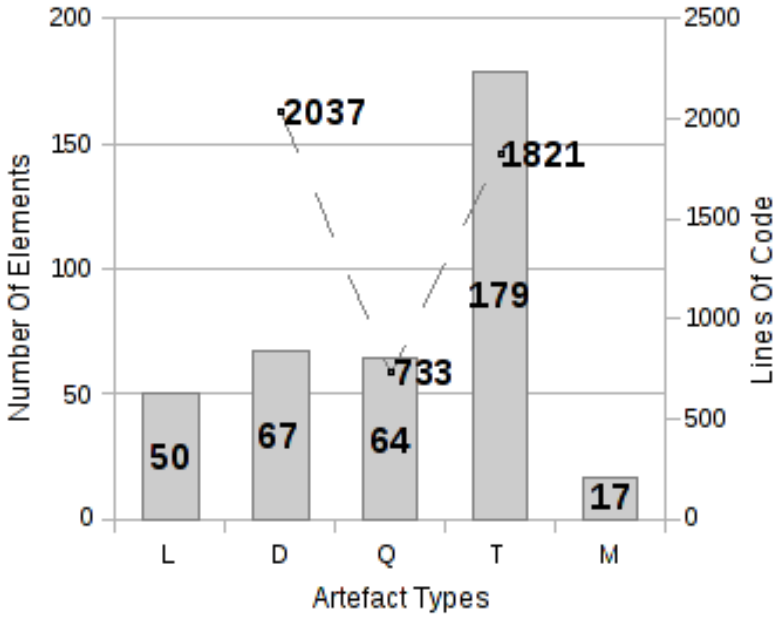


Evaluation

were successfully used in a number of didactic and research projects

the largest project:

- 2 weeks
- 10 students
- **50 libraries**
  with UML metrics
- **4.5 kLOC**
  excluding comments



Summary

the introduced extensions

- **partially address the challenges of OCL development**
- **are implemented in our OCLEditor**
  http://squam.info/ocleditor/
- **were evaluated in a number of project**

Ongoing and Future Work

QUALITY ENGINEERING

To address the challenges

- **currently**
  - we implement **tracing** and **debugging** (C1–C2)
- **in the future we plan**
  - to integrate concepts of **patterns**
  - to collect and evaluate **guidelines** for an efficient OCL library development (C3)
  - to include add **impact analysis**, **regression testing** and **refactoring** support (C4)

QUALITY ENGINEERING

Thank you for your attention!
Feel free to ask questions...
now or later

http://joanna.opoki.com/
http://squam.info/