Research Reports in Software Engineering and Management 2008:01

Towards a Unified Quality Model for Models

A Working Session at the 2nd workshop on Quality in Modeling

Lars Pareto (Ed.)

IT University of Göteborg
CHALMERS | GÖTEBORGS UNIVERSITET
Department of Applied IT

REPORT NO. 2008-01

Towards a Unified Quality Model for Models

A Working Session at the 2nd workshop on Quality in Modeling

Lars Pareto
Christian Lange
Parastoo Mohagheghi
Vegard Dehlen
Miroslaw Staron
Cédric Bouhours
Frank Weil
Cecilia Bastarrica
Sebastián Rivas
Pedro O. Rossel
Ludwik Kuzniarz

Edited by:
Lars Pareto

Department of Applied Information Technology
IT UNIVERSITY OF GÖTEBORG
GÖTEBORG UNIVERSITY AND CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2008
ISSN: 1654-4870

Table of Contents

Introduction

Software quality models, such as ISO9126 [1], are frequently used in large industrial projects: quality models give guidance in what requirements to collect, which architectural qualities to consider, and what to test; they support cost estimation, measurement of project progress, and release-time software approval. (For accessible introductions to software quality models and their use in software engineering, refer to Kitchenham, Pfleeger, and Fenton [2, 3].)

Traditional software quality models are, quite naturally, biased towards code. With the ongoing transition from code centric to model based software engineering, there is an increasing need to extend available software quality models to accommodate for model-specific qualities. The graphical gestalt of models, their varying formality, and their manifold of uses bring out quality attributes highly relevant for modeling but absent (or peripheral) in traditional software quality models. By extending available quality models with these attributes, we help practicing software engineers in their daily work.

Several quality models focusing on model-specific quality attributes have been defined. (Recent overviews may be found in the works of Mohagheghi & Dehlen [4] and Lange [5].) As any other models, these vary in scope and level of detail, with some areas treated less thoroughly than others. This is, perhaps, unavoidable due to the nature of the problem.

But models also vary in their use of quality framework: some use elaborate frameworks taking a multitude of perspectives into account; other use traditional frameworks based on a hierarchical breakdown of quality. This variation adds unnecessary friction to the use of these results in industry, as software engineers in industry rarely have the time to assess new frameworks, less to combine and implement them (with all this means).

To address this problem, the 2nd workshop on Quality in Modeling (which this year had a focus on quality assessment and assurance from an industrial perspective [6]), devoted a ½ day working part to the definition of a *common quality model for models and modelling*. The session's purpose was to elicit quality attributes recognized by researchers in model quality, and to populate an industry-established quality framework with these attributes. By consolidating several active researchers´ understanding on model quality into a common, established framework, a clear picture of what aspects of qualities we understand (and don't understand), and which we know (and don't know) how to measure would hopefully emerge. Such a picture would be useful to industry as inventory of available techniques and useful to researchers as an inventory of open questions in model quality.

This report describes the outcome of this activity. Sec, 2 motivates the working part, and describes its organisation; Sec. 3-5 describe the outcome of the working part, which includes an initial unified model; Section 6 concludes.

Working part organization

Background

A common critique of workshops is that they are too conference like, with too much room for presentations, and too little room for problem solving and active involvement of participants. To this end the organizers of the 2nd Workshop on Model quality [6] (co-located with MoDELS 2007) divided the workshop in two parts: a ½ day presentation part devoted to presentation and discussion of reviewed research contributions; a ½ day working part devoted to group-work around some focus topic appealing to the audience.

Among several topics considered, the organizers settled for the definition of *a unified quality model for models* as focus topic. One of the authors had recently faced the challenge of positioning the results of an industrial case study [7], in such a way that the studied organization could incorporate them in their daily work, and was exploring the quality framework of ISO9126 (which was well-established in the studied organization) [8] for the purpose. The approach was presented at the 5th Nordic Workshop on Model Driven Engineering [9], at which the idea to further explore it in the model quality workshop's working part arose. As the approach was aligned with the accepted papers, addressed an industrial need, and appeared suitable for group-work, it was chosen as focus topic for the working part.

Organisation

General challenges in organizing group work are to communicate the meeting's purpose, to make efficient use of the time together during the meeting, to keep the workload of participants before and after the meeting within bounds, and to communicate the outcome; in case of collaborative research, authorship aspects also need to be taken into account. With these challenges in mind, the following plan for the working part was set up:

1. **Invitation.** Registered workshop participants were to be invited to participate in the working part, informed about the process, and asked to register as active participants.
2. **Questions and Framework.** Topic specific questions and an inspirational framework based on ISO9126 were to be sent out to active participants 10 days in advance of the actual working session.
3. **Contributions.** Participants were to be asked to send contributions to the working parts one day in advance to allow composition of the contributions into a common model. (Composition was to be done by the organizers the night before the workshop). This last-minute composition would allow workshop participants to prepare while travelling, which many regard as an advantage.
4. **Presentations of contributions.** All contributions were to be presented by their authors and discussed during a 5-15 minute slot (depending in the no. of participants); the whole workshop audience was to be encouraged to take active part in the discussions.
5. **Presentation of common model.** The common model (put together in advance) was to be presented by the organizers.
6. **Discussion and group work.** The common model was to be discussed by the audience, and edited on the fly. Hopefully some consensus would be reached.
7. **Technical report.** The unified model was to be presented as a technical report. The purpose of this report should be to document the outcome of the meeting, and to acknowledge all contributions up to and including the meeting; it should not present research carried on after the meeting.

Working part preparations

The preparation of the workshops working part was, at times, chaotic. Registration for the working part and communication of the working part's format did not work out as intended. When questions were to be distributed, the organization committee did not have access to the email-addresses of the workshop. There were unsynchronized changes to the working part's format. As a consequence, the questions were only sent out timely to the three participants, and just before the workshop to a few more participants known to appear.

Although this resulted in fewer and slightly different contributions than planned for, the plan was followed on the large whole, as described below.

Questions and Framework

The questions and the inspirational framework were sent to registered participants on Sep 21st In brief, the participants were asked to address the following questions:

Q1 What qualities of models and modeling matter?
Q2 How do they relate?
Q3 How can they be measured?

Each question was accompanied with instructions that detailed the intention with the question and provided formats for answering the question. In the instructions for Q1, participants were asked to produce a list of quality attributes and to classify these as belonging to either of the following general areas of quality (based on the quality framework of ISO9126[1]):

- *Project quality* relating to how well an organization executes the software process that involves modeling.
- *Process quality* relating to how well the software development process supports modeling (i.e. how well does the process state who should use which models when for what?)
- *Product quality* relating to "technical" properties of the model itself; these may be "white box" properties or "black box" properties.
- *Quality in use* relating to how well users of models can achieve their goals in some particular contexts of use.

The instructions for Q2, asked for two kinds of relationships between the qualities in Q1: *similarity* (a grouping of attributes into similar groups) and *dependence* (a graph that illustrates which qualities affect which).

The instructions for Q3 asked whether or not the author was aware of metrics for the qualities in Q1.

To facilitate easy combination, presentation, and editing of answers, templates for answers, to be edited with PowerPoint, were handed out, along with a request to preferably answer by the use of these templates.

[1] For historical reasons, the framework deviated slightly from ISO9126 in that it distinguishes process from project qualities. We do not regard this as significantly affecting the outcome of the study.

The questions and the framework sent out is given in Appendix A for reference.

Contributed quality models

Seven contributions by eleven researchers were submitted to the workshop's working part:

Contribution 1	Vegard Dehlen, Parastoo Mohagheghi	Vegard.Dehlen@sintef.no Parastoo.Mohagheghi@sintef.nomoha
Contribution 2	Cédric Bouhours	bouhours@irit.fr
Contribution 3	Miroslaw Staron	miroslaw.staron@ituniv.se
Contribution 4	Frank Weil	Frank.Weil@motorola.com
Contribution 5	Cecilia Bastarrica, Sebastián Rivas, Pedro O. Rossel	cecilia@dcc.uchile.cl prossel@spock.ucm.cl
Contribution 6	Christian Lange	c.f.j.lange@tue.nl
Contribution 7	Lars Pareto	lars.pareto@ituniv.se

Table 1 Contributions to the unified quality model

As expected (and desired), contributions varied in scope, and perspective. Some were general, other focused on specific uses of modeling, e.g., in transformations and for design documentation; some provided metrics, other pointed our areas of model quality without known metrics; some followed the framework, other provided perspectives on quality that did not entirely fit it.

As common in definitions of quality, terminology differed. In particular, what some referred to as quality attributes, other referred to as characteristics. This is perhaps unavoidable as virtually all qualities can be further subdivided. However, the submissions indicate that the view of quality model concepts differed.

Contributions also varied in nature. Some were rearrangements of past research results within the bounds of the given framework, other brought poorly understood quality areas much in need for research, e.g., the need for abstraction metrics, and the need for a notion of unified modeling elements.

The contributions (modulo compaction and minor touch-up) are given in Appendix B.

Common format for quality models

The idea to use a template for combining answers was only partly successful: some authors adhered to it while—as the instructions clearly allowed—used their own formats. Many added discussions on quality that did not fit the format.

To combine the models, and to presentation of a common result, the organizers felt a need to put all models on a normal form. For this, the following form was used.

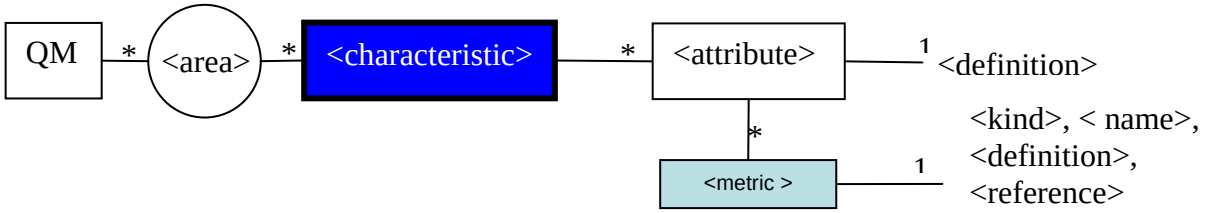


Figure 1 Graphical normal form for quality attributes

Following ISO9126, our quality model consists of a set of *quality attributes* belonging to some *general area of quality*; similar quality attributes are groups into a *quality characteristic*. Each attribute is associated with a definition specifying its essence in natural language, and one or more metrics specifying how it may be measured (thereby also defining it in greater detail). Metrics may be of many different kinds: counts, ratios, and so on [3]. The kind of metric used is indicated using the following notation: # a size, **a/b** a ratio, **1/0** a binary, and **%** a degree of fulfilment metric; an **X** indicates that it is not known whether the quality is measurable. (The kinds arose by quick classification of defined metrics, for the purpose of the presentation, and should preferably be replaced with metric kinds established in measurement theory.)

The following diagram is an example of a quality presented on this form:

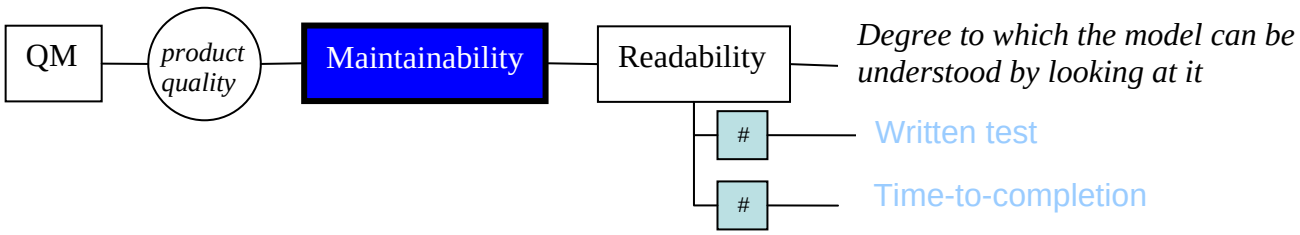


Figure 2 Example of quality attribute on normal form

In the quality model's general area of *product quality*, the characteristic *maintainability* has a quality attribute *readability* defined as "Degree to which, etc.". Readability may be measured using a written test; it may also be measured by time-to-completion for some specific task.

The contributions, were put onto this form before the workshop, and later extended with definitions from the underlying works of the authors. The outcome is given in Appendix C.

Working session

Many workshop participants were actively contributing during the working session; no record was taken, but most contributors should be found among the participant list of the workshop, which included Kerstin Altmanninger, Cecilia Bastarrica, Cédric Bouhours, Robert Canavan, Joanna Chimiak-Opoka, Philippe Dhaussy, Gregor Engels, Ludwik Kuzniarz, Christian Lange, Robert Lario, Martin Monperrus, Wiktor Nowakowski, Lars Pareto, Steffen Prochnow, Gianna Reggio, Sebastián Rivas, Pedro O. Rossel, Miroslaw Staron, Steven Varr, Vegard Dehlen, Daniel Völkel, Xulin Zhao, Frank Weil, and Stephan Weissleder.

Working session introduction

The working session started with an introduction to the purpose and process of the working part, its background, and the choice of ISO9126 as quality framework. (Appendix D).

The question of what would happen to the outcome of the discussion was brought up. There was consensus that outcome should be published in a technical report with all contributors as co-authors, shortly after the meeting. Everyone would be free to take that research ideas onwards, if they so wanted, together with other workshop participants, or on their own, but this would neither be part of the working part, nor documented in the technical report. It was also agreed that the report should be sent out to everyone for review before going into print.

Presentations and discussion of contributions

The seven contributes were (with one exception) presented by their authors and discussed by the whole workshop audience. (Contribution 3 was presented by the organizers, because the author could not attend.)

The following topics were discussed:

- **UME**. The need for a Unified Model Element (see contribution 3) was acknowledged. As far as the audience was aware, this is a new idea.

- **Functionality** The meaning of functionality (in contribution 3) and how it related to completeness with respect to purpose was discussed. An alternate definition would be *completeness with respect to behaviour*.

- **Traceability relate to needs.** What good traceability is much depends on the context of use.

- **Abstraction Metrics**. The need for research in abstraction metrics (see contribution 4) was acknowledged. A common problem in industry is that supposedly abstract models contain too much implementation detail. It is desirable to detect this, and metrics for degrees of abstraction would allow this.

  The following idea was discussed. Suppose a code generation model M_C contains 10 000 elements. Suppose, further, the existence of some abstraction transformations α that capture what good designers do not include in their analysis models: applied

to M_C the transformation α would give a model at the desired level of abstraction. Now by measuring the size of $\alpha(M_C)$ we would know what to expect from the model written by flesh-and-blood designers: if the size of this model is, say, 20 000 and the size of $\alpha(M_C)$ is 1000, then the flesh-and-blood designers have not written an model that is abstract enough.

A problem with this approach is, of course, that it assumes that the code generation model (or the code) is available before the design model. Now, in an industrial setting, this is not much of a problem, as one may *estimate* $\alpha(M_C)$ by application of α to the code generation model of some older product of similar size.

- **Quality Types**. The Model Qualities defined in Q1 of contribution 6 (System, Semantic, Syntactic, Pragmatic, Social, Communicative) did not appear to be model qualities, neither quality attributes in the sense of ISO9126. But what are they? The term *quality types* seemed to make sense to the workshop participants (at least during the meeting).
- **Model Qualities vs. Product Qualities.** The distributed framework used the term product qualities (which stems from ISO9126). Many disliked this term and suggested the term *model qualities* to be used instead. Consensus was reached to use the latter in the common model.
- **Model infrastructure quality**. Many of the categories from contribution 7 were different in nature: they were *model infrastructure qualities* rather that *model qualities.* Consensus was reached that a unified model should make this distinction at the level of general areas of quality.
- **UME_{SIZE}, $UME_{BUSINESSVAL}$, UME_{RISK}**
- **Productivity: product; quality; cycle time. Steerability.**

Presentation of normalised models

The contributions on normal form were presented quickly. As each contribution had been thoroughly discussed during the presentation, there was not much to say about the change into normal form. Rather, time was spent on the more interesting task of discussion similarities and differences between the concepts in the common pool of quality attributes names.

Similarity of qualities

Inevitably, many models defined the similar qualities, albeit with different names. This problem was addressed in a group discussion with the particular aim of finding similar (and identical) and qualities. The following steps were used:

a) Before the working session, each model quality attribute was colour coded (to keep track of its origin) and placed on a common slide (see Appendix E, topmost part).
b) After the presentation of the common model, this slide was collectively rearranged in a group discussion, in which the whole workshop participated. Starting with the general area of product quality, similarly named qualities, or qualities with similar meaning were grouped. After 20 minutes or so, time was up.
c) The organizers agreed to tidy up the product quality part of the model after the workshop.

The slide, as it appeared by the end of the discussion is given in Appendix E, lowermost part, for reference.

Unified Model

The tidying up of the model was done several weeks after the workshop, and involved the following activities:

d) Incorporating changes to terminology agreed during the discussions.
e) Introduction of some quality attributes that had been left out in the preparation of the common slide. (The omission was partly accidental, and due to the different interpretation of the quality frameworks in the submissions: what some had classified as a quality in use, other classified as a model quality.)
f) Further grouping of qualities with respect to the underlying definitions of the quality attributes.

After these steps, the following groups had emerged:

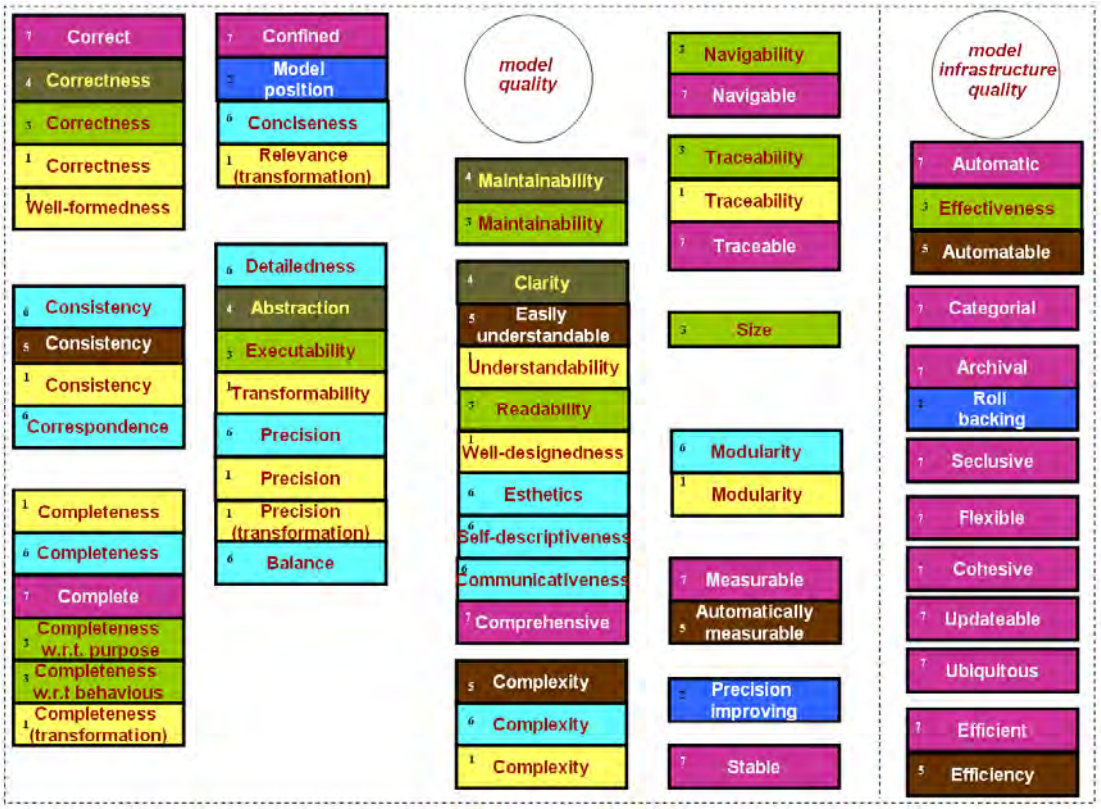


Figure 3 Model Quality attributes grouped with respect to similarity

Here, adjacency means either that the names refer to the same concept (e.g., Clarity and Easily Understandable) or that some of the attributes/characteristics are entailed within in the other (e.g., Well-formedness is one kind of Correctness).

Model and model infrastructure qualities

Coalescing groups, and choosing one of the names used to describe group, one arrives at the following top-level characteristics of model quality (modulo choice of name):

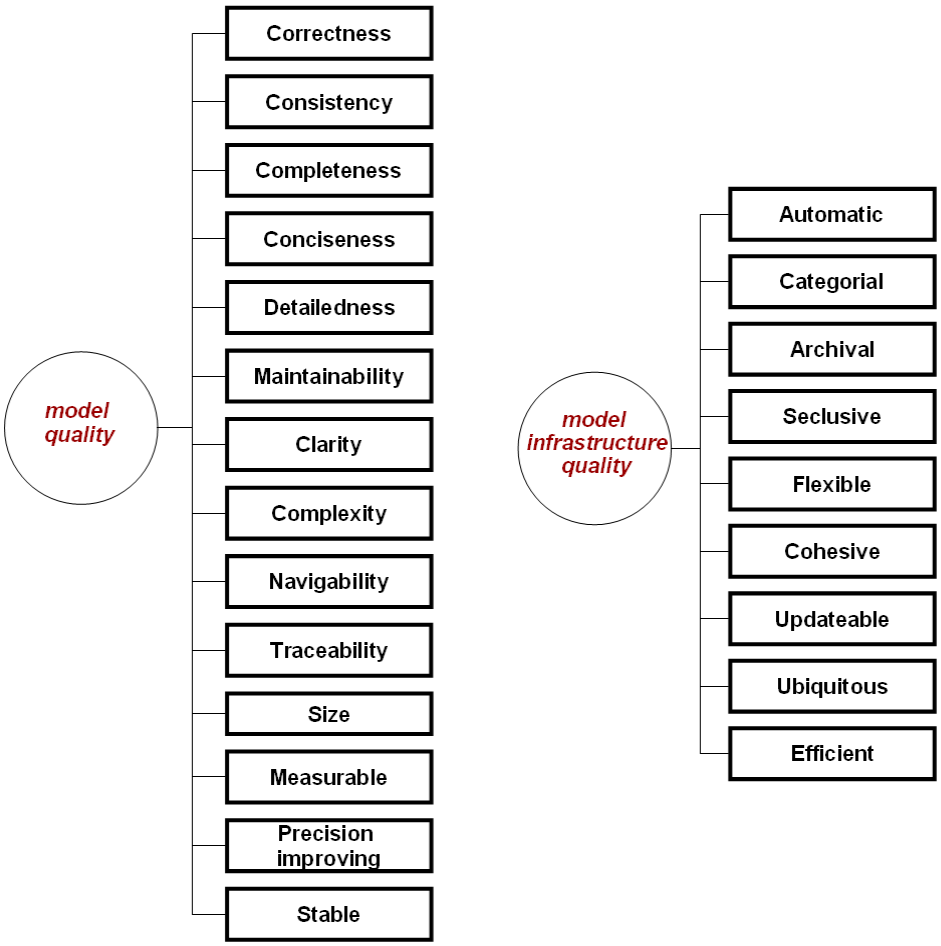


Figure 4 Qualities of Models and Model Infrastructure

These characteristics in turn include sub-characteristics, that capture variations of the characterised quality, e.g., well-formedness is a sub-characteristic of correctness. We refrain from defining these, as this organisation was not discussed during the workshop.

Coalescing of definitions and metrics

The coalescing of attributes in each group leads to competing attribute definitions and competing metrics for the quality captured by the group. For instance, after coalescing *correctness*, we obtain the following conflicting definitions and metrics:

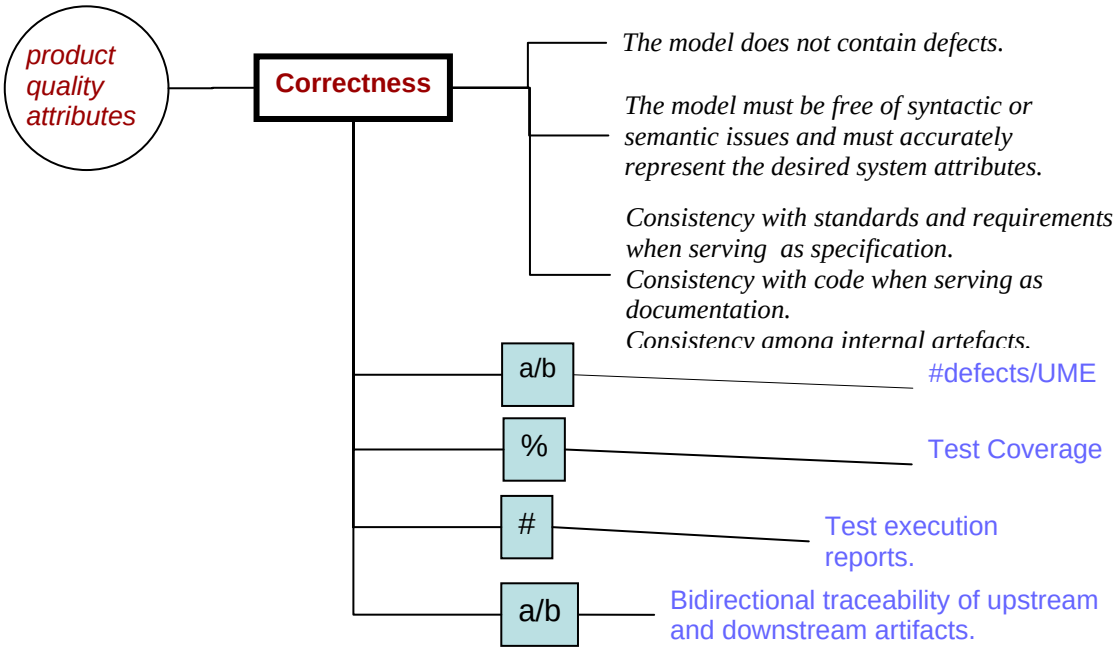


Figure 5 Conflicting definitions of Quality Attributes

Resolution of these conflicts may be done by i) the introduction of sub-characteristics, or ii) generalisation of the definitions to capture the essence of all. Doing this is however beyond the scope of this report.

Process and Project Qualities

The plan of the working part was to unify not only model qualities, but also process and project qualities. The questions and the framework asked for quality attributes in these areas, and several contributions defined such. Unfortunately, workshop time ran out before these qualities were discussed and organised. By putting them together (without further analysis) we obtain the following of project- and process quality attributes:

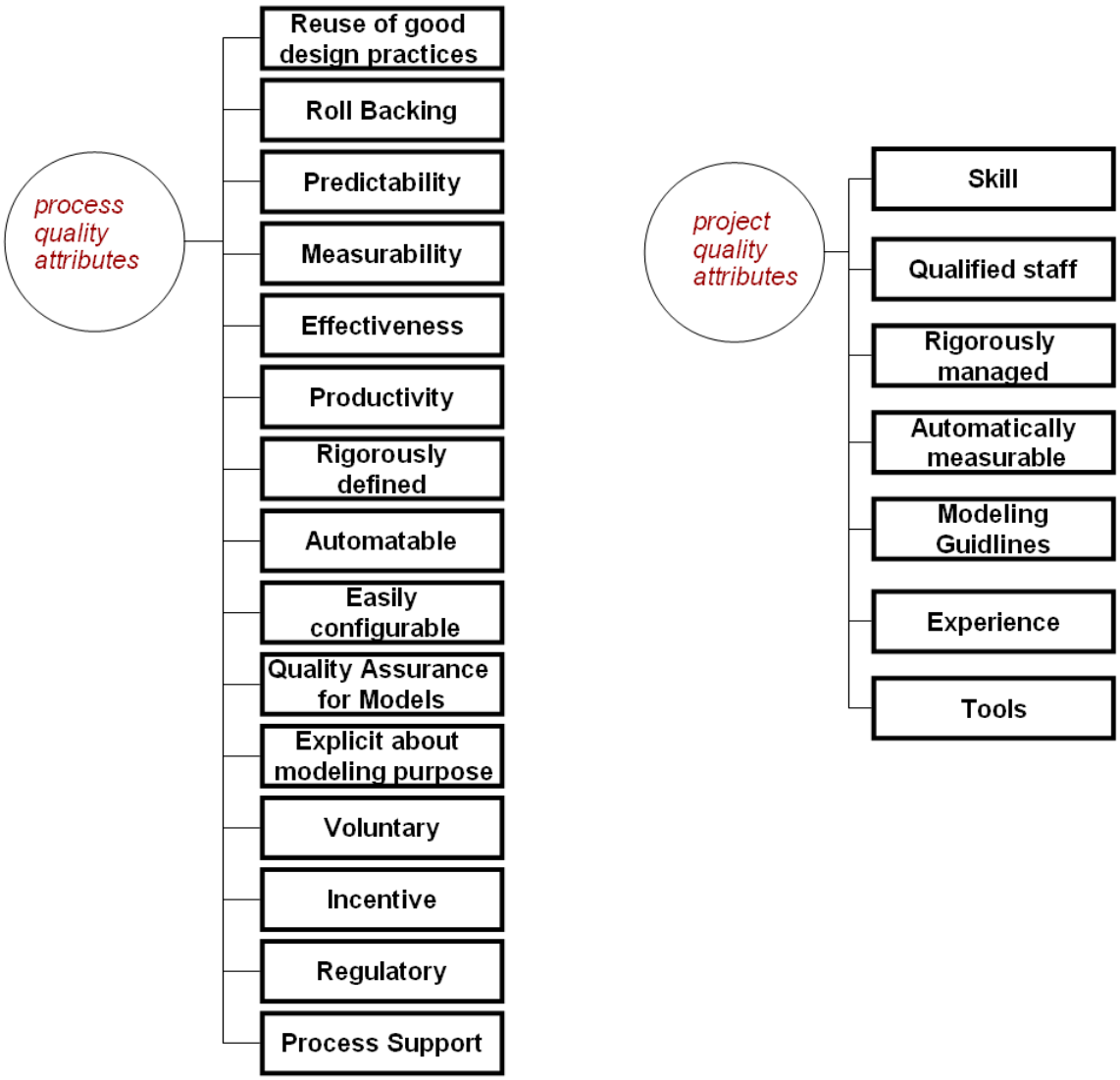


Figure 6 Model related Processes and Projects qualities

Grouping and coalescing these qualities would follow the same steps as for model qualities. This, however, is future work.

Conclusions

Lessons learned about the organization of working parts

The organization of the working part brought some experiences worth noting.

On the positive side:

- the working part format brought many researchers in model quality together to actively consider each others works, and to assess the novelty of research ideas;
- researchers new to the field of software quality were given an introduction to quality models and an overview of key quality attributes in modeling;
- the active group work worked well, and discussions were fruitful;
- the activity as a whole resulted in some preliminary research output (i.e., this technical report) further described below.

On the negative side, there were many organizational slips:

- to require separate registration for the working part was not a good idea; (every workshop participant should be invited to contribute at the time of registration;)
- the session was overly ambitious and the scope of the questions sent out too big; (question Q1 alone would have given enough material for a fruitful working part;)
- questions should be sent out earlier; (10 days in advance allowed to little time for preparation;)
- more than one day should be set off to comprehend and edit the submissions (which may vary more than expected);
- the concepts of any frameworks to be used (in our case the quality framework of ISO9126) need to be described in detail, or the variation in the contributions may become difficult to manage.

On the large whole, the organizers are content with the outcome of the working part and positive to the use of group-work for collective research. Assuming that proper attention will be paid to planning and communication, we may recommend the use of the format for other workshops too.

Research output

During the working part, several open questions were identified, namely

- How do we measure abstraction?
- What is good definition of a unified modeling element (UME)?
- What's the role of quality types in a quality model?
  (Quality types are not part of ISO9126.)
- How do identified qualities relate?

The following progress was also made:

- an inventory or 32 qualities of models and modeling, virtually all in need for further research, was produced,
- quality attributes defined by nine researches in model quality were brought into the quality framework of ISO9126,
- we have showed that the quality framework of ISO9126 lends itself to well to the characterization of qualities recognized by researchers in model quality,
- some promising firsts steps towards a common quality model for models and modeling have been taken.

The working part's purpose, i.e.. to elicit quality attributes recognized by researchers in model quality, and to populate an industry-established quality framework with these attributes, has thus been met.

References

1. *ISO/IEC 9126: Information technology–software product evaluation–quality characteristics and guidelines for their use*. 1991, Int. Org. for Stand. (ISO).
2. Kitchenham, B. and S.L. Pfleeger, *Software quality: the elusive target.* IEEE Software 1996. **13**(1): p. 12-21.
3. Fenton, N.E. and S.L. Pfleeger, *Software metrics; a rigorous & practical approach*. 1997: PWS Publishing Company.
4. Mohagheghi, P. and V. Dehlen, *An overview of quality frameworks in model-driven engineering and observations on transformation quality*, in *Proceedings of the 2nd Workshop on Quality in Modeling*. 2007: Nashville, TN, USA.
5. Lange, C.F.J., *Assessing and Improving the Quality of Modeling*. 2007, Technishce Universiteit Eindhoven: Eindhoven.
6. *Proceedings of the 2nd Workshop on Quality in Modeling*. 2007. Nashville, USA.
7. Pareto, L. and U. Boquist, *A quality model for design documentation in model-centric projects*, in *Proceedings of the 3rd international workshop on Software quality assurance* 2006 ACM: Portland, Oregon
8. Pareto, L. *Quality in Modeling; Towards a quality model for models*. in *Nordic Workshop on Model Driven Engineering NW-MODE 2007*. 2007. Blekinge Institute of Technology.
9. *Workshop proceedings of the 5th Nordic Workshop on Model Driven Engineering*. 2007. Ronneby Sweden.

2nd Workshop on

Quality in Modeling

Co-located with

MoDELS 2007

the ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems

Nashville, TN, USA

October 02, 2007

Questions for the workshop's working part

The source (.ppt) for this document may be obtained by
- download from www.ituniv.se/~lapar/wpq.ppt
- email-request to lars.pareto@ituniv.se
- USB stick (carried by Lars Pareto)

Questions

The purpose of the Quality workshop's working part is to establish a *common quality model* for software models and software modeling practice.

Because quality means different things to different people, we would like *your view* on model quality in the form of answers to the following three questions

Q1 What qualities of models and modeling matter?

Q2 How do they relate?

Q3 How can they be measured?

The purpose being is to establish a common model, we kindly ask you to structure your answers using the guidelines given below. To make the editing process smooth, we also ask you to (preferably) submit your answers in the form of a *powerpoint presentation*, using the diagrams in this document as a starting point. This allows the desired common quality model to be defined by an overlay of the individual contributions of the workshop participants.

Your contribution
- should be the basis for a ~10 minute presentation held by you during the workshop's working part,
- will be combined with all other contributions to make a common quality model. (The common model is edited and presented by the workshop organizers.)
- will be published, along with the common model , in the working parts technical report. (There will be an opportunity to detail on your model after the workshop, in a voluntary 2p extended abstract.

Send your contribution to: lars.pareto@ituniv.se, lku@bth.se

Deadline : 2007-10-01 (at the very latest)

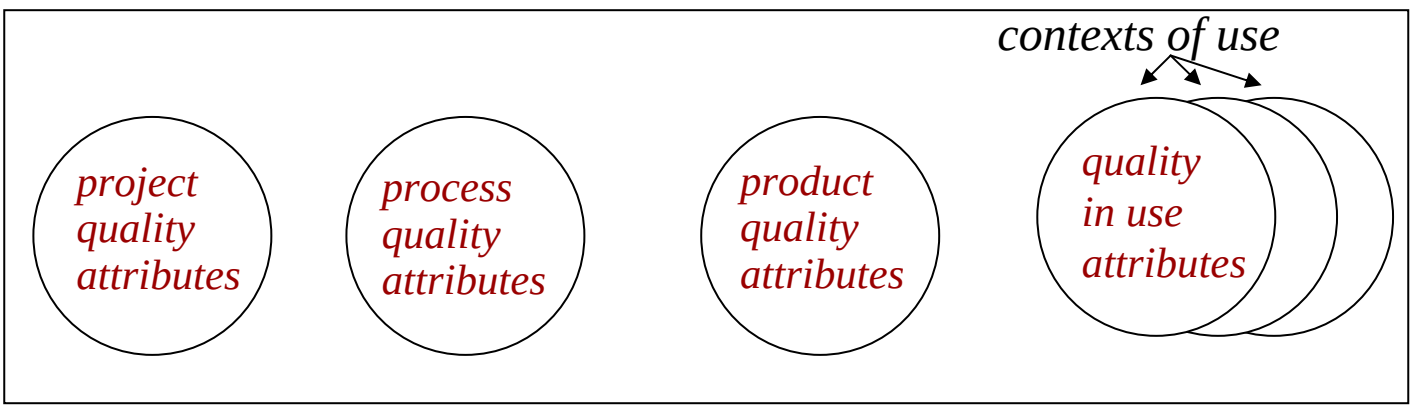
Q1: What qualities of models and modeling matter?

We seek a list of quality attributes that you think are important for some model-related software engineering activity. Quality attribute definitions should consist of a *name* and *a definition*.

Examples

1. *Consistency (Design Model and Code Generation Model)* **"**The analysis models are syntactically consistent with those used for code generation."
2. *Automatically measurable* "Metrics can be automatically computed for the model"
3. *Completeness w.r.t design guidelines.* "The model contains all artifacts described by the guidelines."

The scope of the qualities sought are project-, process-, and product qualities as well as qualities in use, as defined by the following adaptation of the ISO9126 quality framework:



Project quality attributes relate to how well an organization executes the software process that involves modeling.

Process quality attributes relate to how well the software development process supports modeling (i.e. how well does the process state who should use which models when for what?)

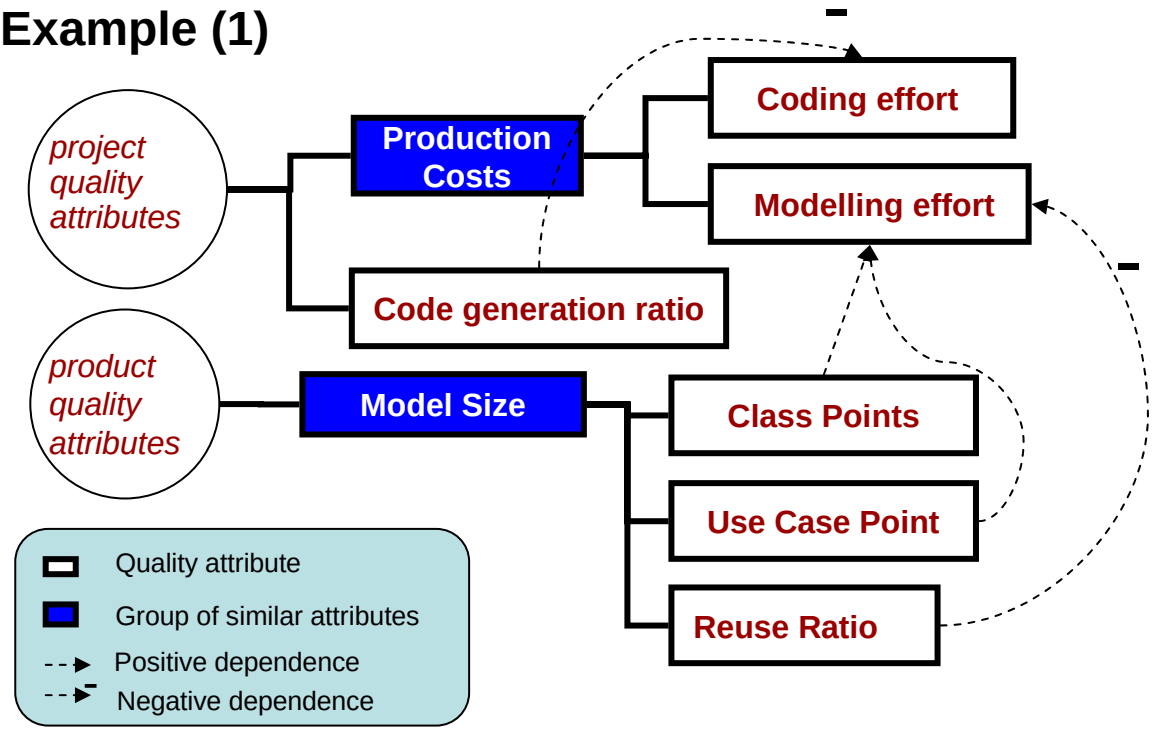
Product quality attributes relate to "technical" properties of the model itself; these may be "white box" properties or "black box" properties.

Quality in use attributes Relate to how well users of models can achieve their goals in some particular contexts of use.

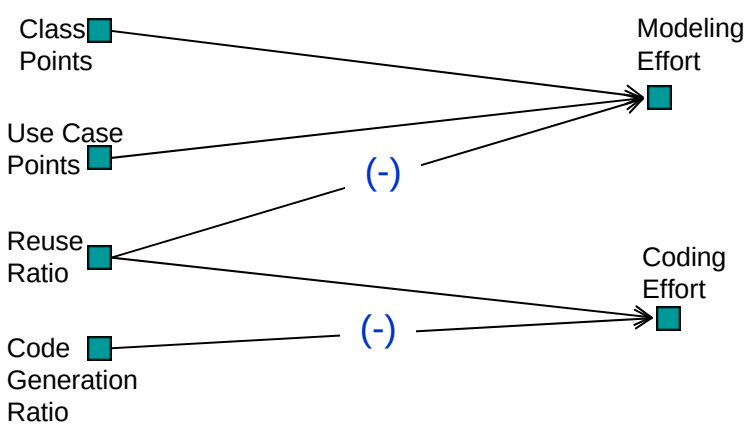
Q2: How do qualities relate?

We are interested in two relationships between quality attributes: *similarity* and *dependence*. What we seek, here, is (1) a grouping of "similar" quality attributes with characterizing definitions of these sets, and (2) a dependency graph between your quality attributes, showing how the quality attributes affect each other.

Example (1)



Example (2)



Q3: How can model qualities be measured?

We are interested in the feasibility of measuring these qualities in real projects, and seek a common body references into the research literature, experience reports, or simply your thoughts on how the qualities may be measured.

Remark: The main purpose of this question is to validate the answers to question Q1: difficulties in identifying or envisioning metrics often indicate that the qualities are too general and should be subdivided into measurable parts. A secondary purpose is to compile a catalogue of methods, techniques, and terminology useful in reasoning about the quality of models and modeling.

Q1: What qualities of models and modeling matter?

Several general quality characteristics for models have been defined:

1. *Complexity*
2. *Completeness*
3. *Correctness*
4. *Understandability*
5. *Modularity*
6. *Precision*
7. *Consistency*
8. …

Two quality characteristics for models in Model-Driven Engineering, based on [Solheim06].

1. *Transformability*
    1. *Completeness*. "The model contains all the necessary elements and relations from the domain"
    2. *Well-formedness*. "The model complies with its metamodel, and also with its specified language profile, if appropriate."
    3. *Precision*. "The model is sufficiently accurate and detailed for a particular automatic transformation. "
    4. *Relevance*. "The model contains only the elements and relationships necessary for a particular transformation. "
2. *Maintainability*
    1. *Traceability.* "The model's elements can be traced backward to their origin (requirements), and forward to their result (another model or program code). "
    2. *Well-designedness.* "The model has a tidy design, making it understandable by humans and transformable to an understandable and tidy result. "

Q2: How do qualities relate? N/A

Q3: How can model qualities be measured?

Quality characteristics for models in Model-Driven Engineering, based on [Solheim06].

1. *Transformability*
    1. *Completeness*. Suggested measurement unit: percentage.
    2. *Well-formedness*. Suggested measurement unit: percentage.
    3. *Precision*. Suggested evaluation: yes/no.
    4. *Relevance*. "Suggested measurement unit: percentage"

- *Maintainability*
    - *Traceability.* Suggested metric: trace coverage, the proportion of traceable model elements relative to the total number of model elements
    - *Well-designedness.* Suggested metric: The quality model of [Marinescu02], preferably extended with other diagrams than class diagrams.

Conclusions:

-We lack metrics and ways to measure many quality characteristics.

- We have many metrics related to size and complexity, but the usefulness of these are not necessarily evident.

Contribution 2 Cédric Bouhours

- In the first slide we present some quality attribute with their definitions.
- In the second slide, we try to explain how to measure them.

Q1: What qualities of models and modeling matter?

- *Process quality attributes*
- ✓ *Reuse of good design practices : "ability for a process development to urge designers to reuse expert knowledge at each activity of design stage."*
- ✓ *Roll backing : "ability for a process to roll back an activity thanks to traces"*
- *Product quality attributes*
- ✓ *Precision improving : "ability for a designer to precise his intent in his model."*
- ✓ *Model position : "models contain only elements available in the phase"*

Q3: How can model qualities be measured?

- ✓ *Reuse of good design practices :*
- ✓ *Number of alternative model detected and validated.*

- ✓ *Roll backing :*
- ✓ *Not measurable*

- ✓ *Precision improving :*
- ✓ *Number of dedicated stereotypes and notes*

- ✓ *Model position :*
- ✓ *Depending on phases. For example, in business model, it is the ratio between common terms in model and terms in requirement.*

Appendix B

Contribution 3 Miroslaw Staron

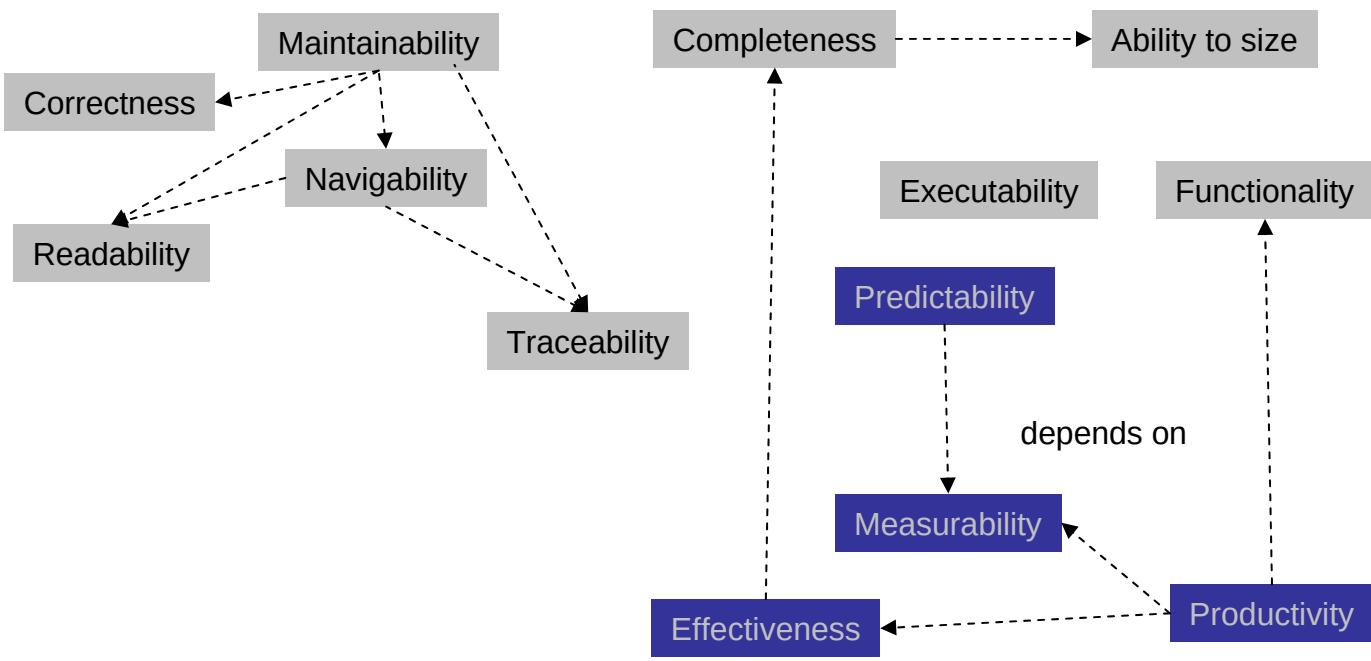
Q1: What qualities of models and modeling matter?

- Qualities of models
  - Completeness w.r.t. purpose: the model is complete – contains all information needed for a purpose; for example the architectural model contains full specifications of all interfaces and protocols
  - Correctness: the model does not contain defects
  - Maintainability: one is able to modify the model without much effort
  - Navigability: one is able to easily navigate through a model, for example during inspections
  - Traceability: information in the model can be linked to other information sources – e.g. requirements, source code, test cases
  - Readability: the model is easy to read
  - Functionality: the model contains the description of all functions of the product
  - Executability: ability to be executed
  - Ability to measure size: one can measure the size of the model in some atomic units (e.g. unified model element)

- Qualities of modeling
  - Predictability: one is able to predict how much time a modeling task will take you; for example developing an architectural class diagram will take 30 +/- 5 days
  - Measurability: one is able to measure the delta of your work when doing modeling; for example how do we measure how much modelling we did during a day – e.g. added 20 attributes to the model, run 10 test cases to test them, fixed 3 defects
  - Effectiveness: the things in the models will be (automatically) in the final product
  - Productivity: the modeling process should be productive

  (It has to be more productive than competing coding process,
  or these in ni point in using it– shorter release times

Q2: How do qualities relate?



I do not think I have similarity (can't really see the notion of distance) now

Contribution 3 Miroslaw Staron

Q3: How can model qualities be measured?

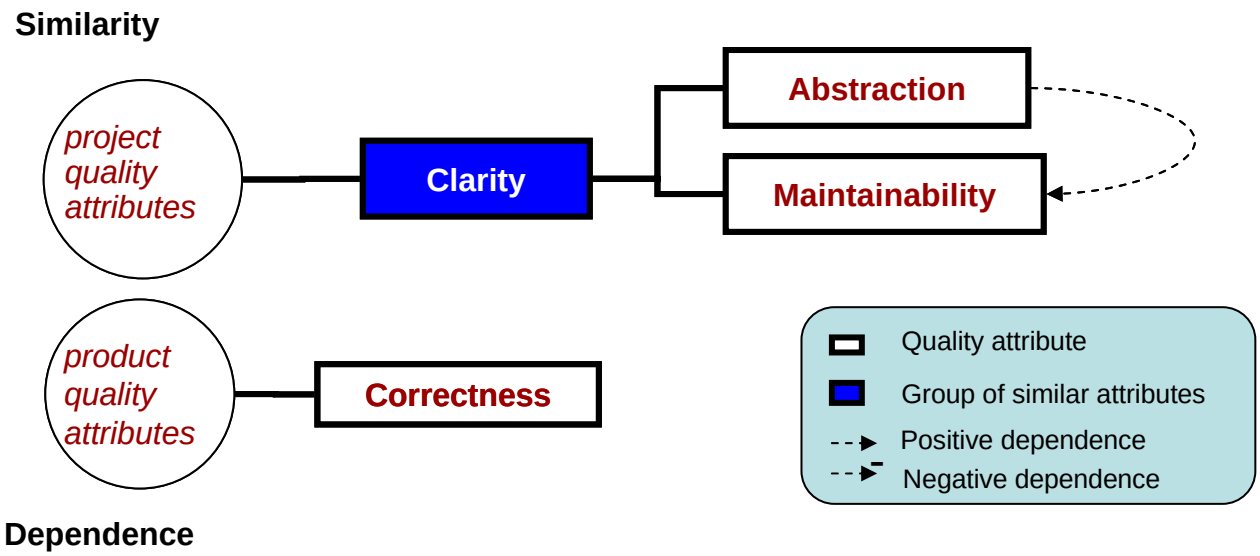
- *UME – Unified Model Element*
  - *A unit which measures an elementary and atomic unit of model – e.g. equvalent to an attribute or a state; then class = 20 UME, state: 20 UME, one NCSLOC = 0.5 UME etc.*
- Qualities of models
  - Completeness w.r.t. purpose: -- not really measurable now --
  - Correctness: n-of-defects/UME
  - Maintainability: effort/change/UME
  - Navigability: average(time-to-find-element/UME) for a sample of elements (n>30)
  - Traceability: n-of-treaceability-links/UME
  - Readability: -- measured empirically: time-to-understand/UME
  - Functionality: features-modelled/features-in-SRS
  - Executability: binary: yes/no at component lvel (or percentage for RoseRT)
  - Ability to measure size: UME
- Qualities of modeling
  - Predictability: 1 – (MMRE-of-predictions)
  - Measurability: UME/time-unit is possible to compute
  - Effectiveness: 1 – (number-of-elements-not-traceable-to-code/all-elements [in UME])
  - Productivity: UME/time-unit

Contribution 4 Frank Weil

Q1: What qualities of models and modeling matter?

1. *Correctness* **"**The model must be free of syntactic or semantic issues and must accurately represent the desired system attributes."
2. *Abstraction* "The model must be free from assumptions about the final implementation."
3. *Maintainability. "*The model must be structured to allow continued evolution by modelers different from the creator."

Q2: How do qualities relate?



Q3: How can model qualities be measured?

Correctness: Test coverage. Test execution reports. Bidirectional traceability of upstream and downstream artifacts.

Abstraction: *I do not know how to measure this.*

Maintainability: This is definitely a roll-up category related to traditional measures such as coupling and cohesion. There are, however, other factors. For example, use of literals in the model should be avoided, with the possible exception of some uses of 1, 0, "", etc.

Q1: What qualities of models and modeling matter?

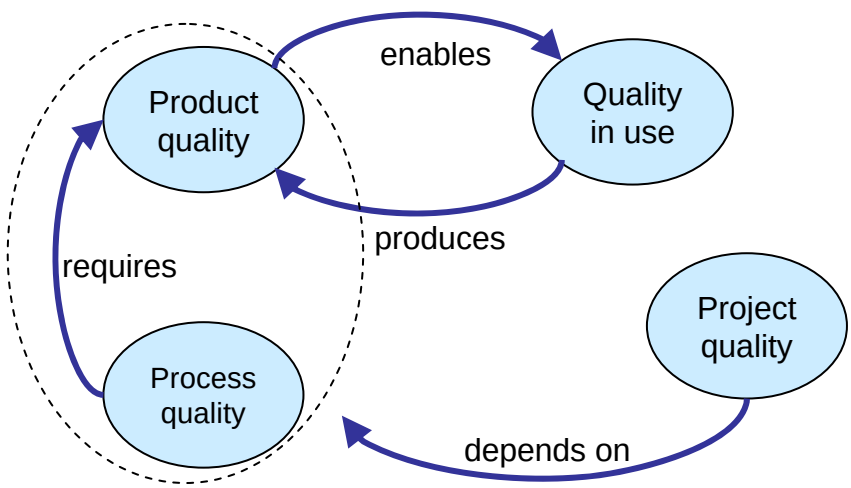
- **Product qualities**
  - Consistency
    - There are no contradictions
  - Automatically measurable
    - Qualities should be objectively quantifiable
  - Complexity
    - Models should be as simple as possible
  - Easily understandable
    - Model meaning should be intuitive

- **Qualities in use**
  - Easily understandable
    - Understandability helps model developers to produce better models
  - Efficiency
    - Lower time and cost of development and evolution

- **Process qualities**
  - Rigorously defined
    - Activities, roles, artifacts, tasks must be clearly defined
  - Automatable
    - Formality enables automatic model transformation
  - Easily configurable
    - Changes should be possible and not too difficult to introduce

- **Project qualities**
  - Qualified staff
    - People knowledgeable in models and the modeling process
  - Rigorously managed
    - The process must be strictly followed using high quality products
  - Automatically measurable
    - Quality can be enhanced only if metrics are available

Q2: How do model qualities relate?



Q3: How can model qualities be measured?

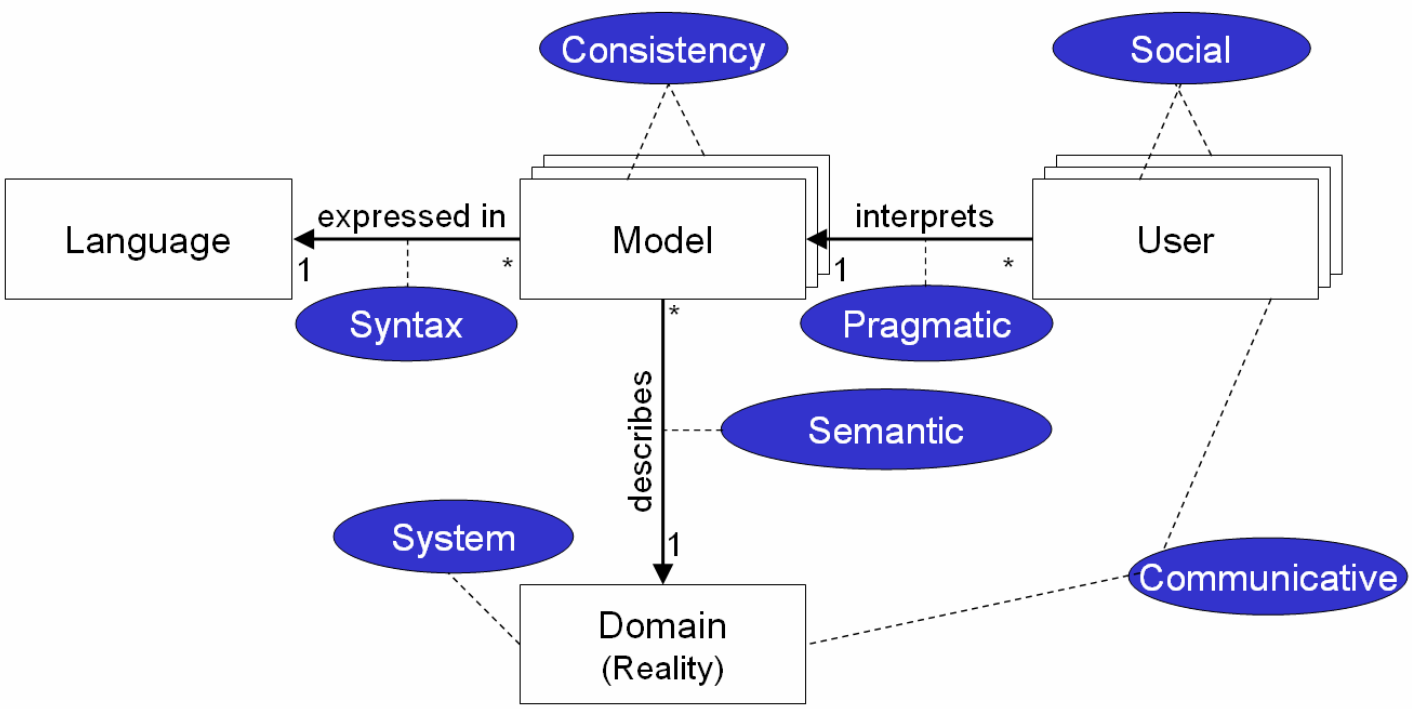
- **Product quality**
  - Consistency and automatically measurable
    - MCC-SPL for UML models

- We are quite interested in having an answer, but unfortunately we do not have one yet.

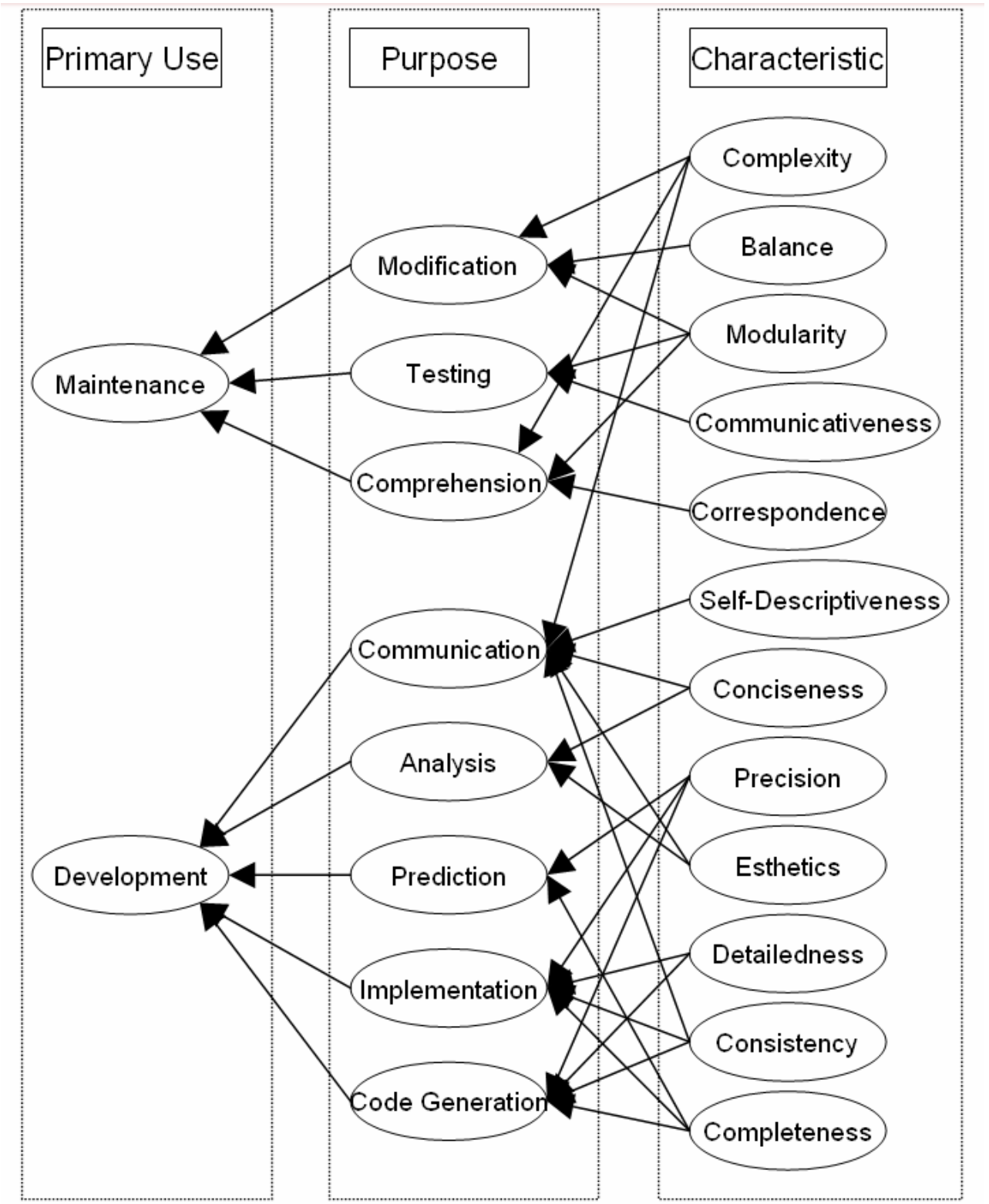
Q1: What qualities of models and modeling matter?

- Project Qualities
  - Modeling Guidelines !!
  - Skill
  - Experience
  - Tools
  - …
- Process Qualities
  - Quality Assurance for Models
  - Being explicit about the purpose of modeling
- Model Qualities (rather than Product Qualities)
  - System
  - Semantic
  - Syntactic
  - Pragmatic
  - Social
  - Communicative

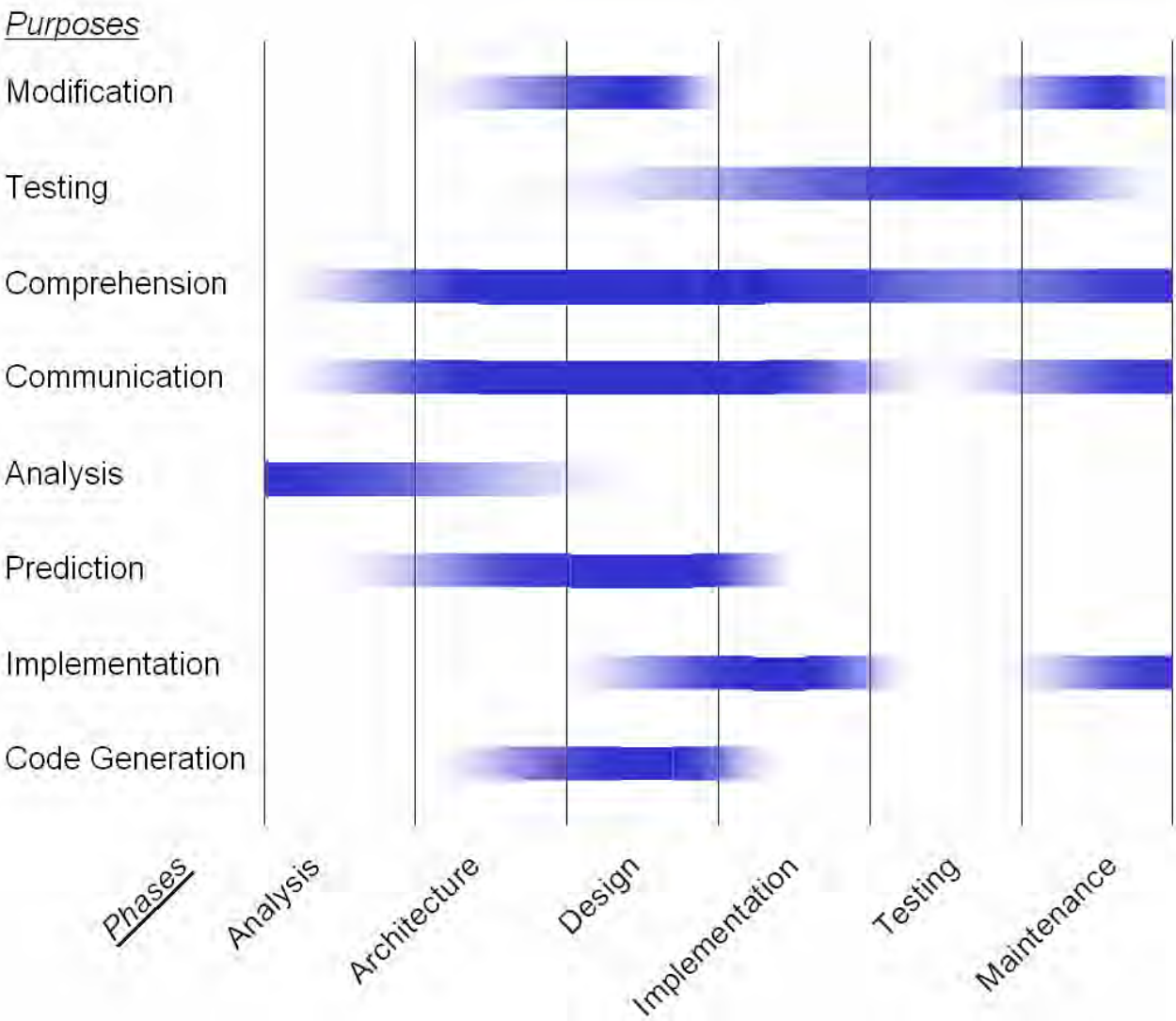
Q2: How do model qualities relate?



Q2: How do model qualities relate?



Q2: How do model qualities relate?



Q3: How can model qualities be measured?

Model Qualities

- System → Metrics, Experts, SAAM / ATAM, Simulation...
- Semantic → human assessment *
- Syntactic → rule checking (automated or inspection)
- Pragmatic → human assessment *
- Social → human assessment *
- Communicative → human assessment *

* = Is it possible to measure these qualities based on automatically measurable qualities?

- → empirical validations

Contribution 7 Lars Pareto

Appendix B

Q1: What qualities of models and modeling matter?

Archival	*Selected information from the model's history is retained*
Regulatory	*Well defined what the model should contain and not contain.*
Flexible	*Allows temporary, conscious violations of key qualities.*
Efficient	*Retrievals of, navigations in, searches in, and updates of models is fast.*
Confined	*That only what's prescribed by guidelines is in the model.*
Automatic	*Supports (semi-) automatic generation of model parts, from other sources.*
Measurable	*Supports generation of metrics and conformance checks.*
Voluntary	*Does not force designers to work in a certain way.*
Correct	*Consistency with standards and requirements when serving as specification.* *Consistency with code when serving as documentation.* *Consistency among internal artefacts*
Incentive	*Supports recognition of individual efforts.*
Ubiquitous	*Can be worked with on the same locations one can work with code.*
Traceable	*Artefacts depending of each other are linked.*
Cohesive	*Related parts retrievable together*
Complete	*Whether all components are documented.* *Whether all artefacts prescribed by guidelines are provided.*
Seclusive	*Allows artefacts to be protected from public view, and excluded from metrics.*
Comprehensive	*Users able to understand model (without being annoyed).*
Navigable	*Supports navigation from several perspectives, in particular string search.*
Usable	*Support user tasks it is intended for.*
Updateable	*Possible to update all artefacts and distribute the updates.*
Categorial	Supports classifications systems, e.g., - initial, pending, approved, retired, etc. and - artefact kind (statechart, seq. diagram, etc.)
Supported by process	*Enclosing processes specify when and how the model's parts should be constructed and maintained.*
Stable	*Structural and Categorial changes infrequent.*

Appendix B

Q2: How do qualities relate? *Similarity*

Obvious superior qualities

- Key qualities
  - Correct
  - Complete
  - Confined

Checking and transformation support.

- Syntactic qualities
  - Automatic
  - Traceable
  - Cohesive
  - Categorial

Effectiveness when read.

- Readability qualities
  - Comprehensive
  - Archival
  - Stable

Capability to support tasks.

- Usability qualities
  - Navigable
  - Updateable
  - Usable
  - Ubiquitous
  - Efficient

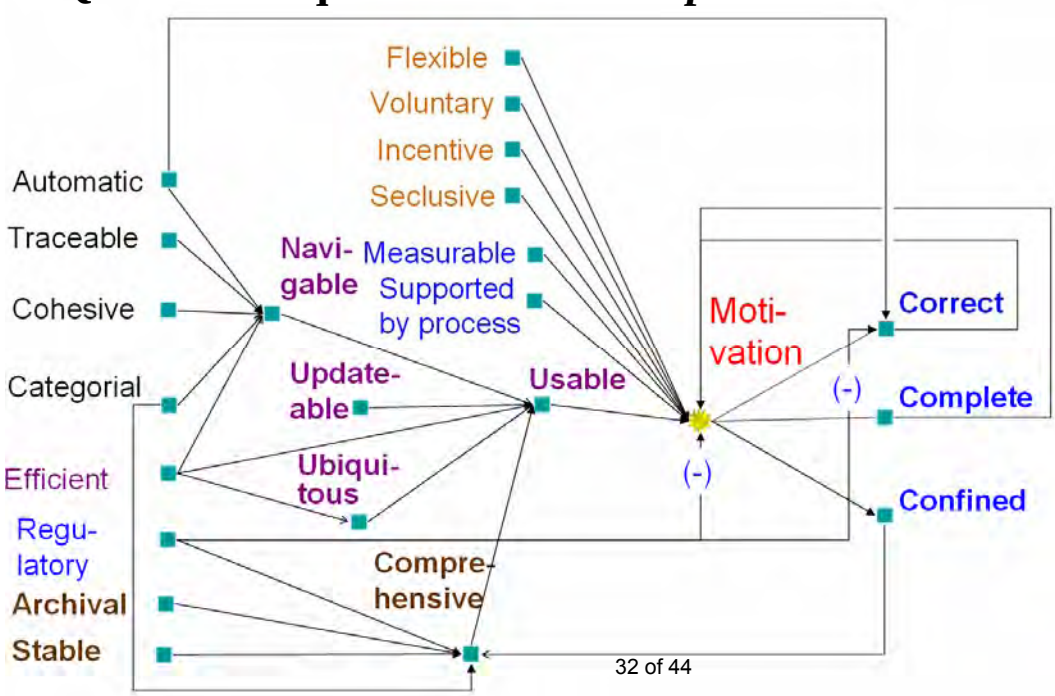
Affect designer motivation.

- Motivational qualities
  - Flexible
  - Voluntary
  - Incentive
  - Seclusive

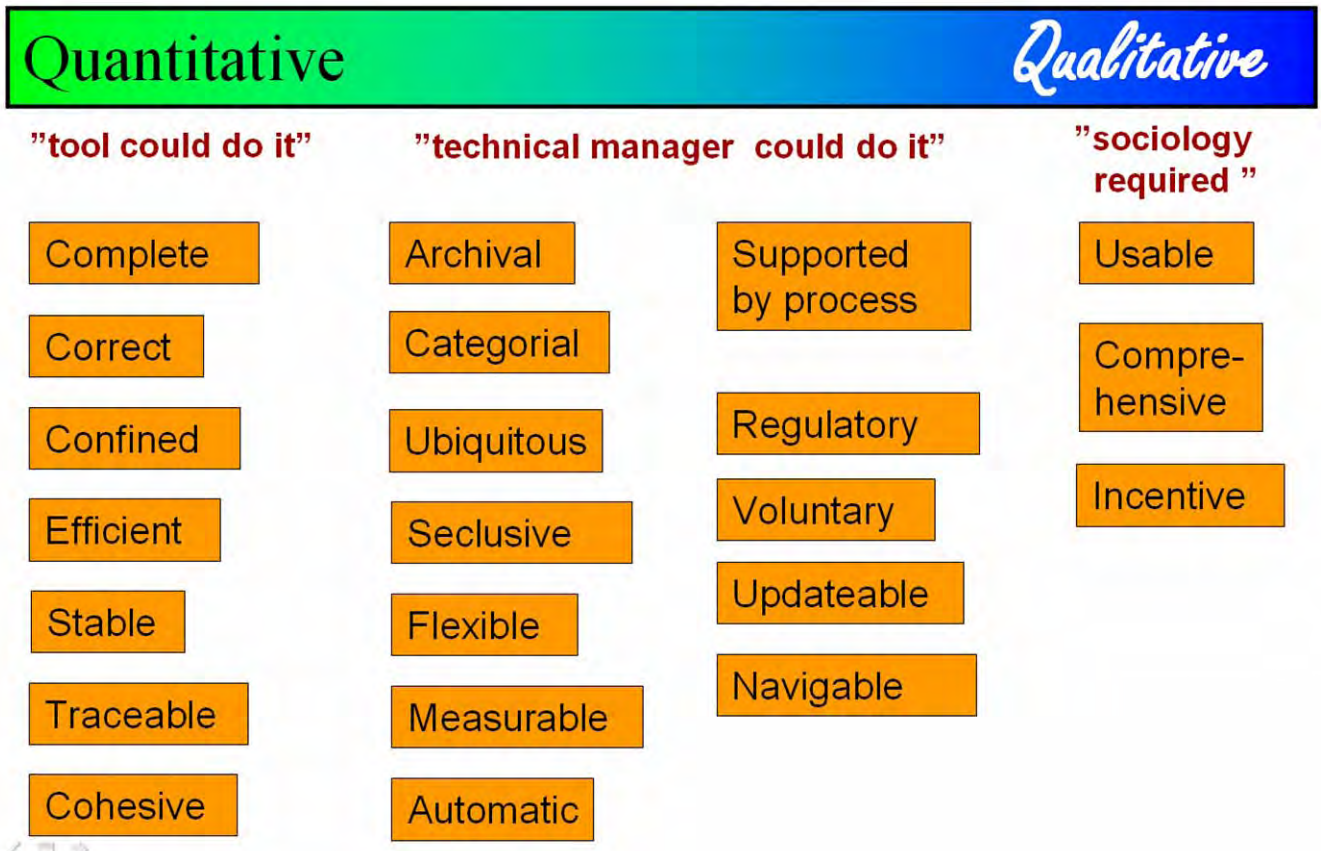
Affect project management.

- Manageability qualities
  - Regulatory
  - Measurable
  - Supported by process

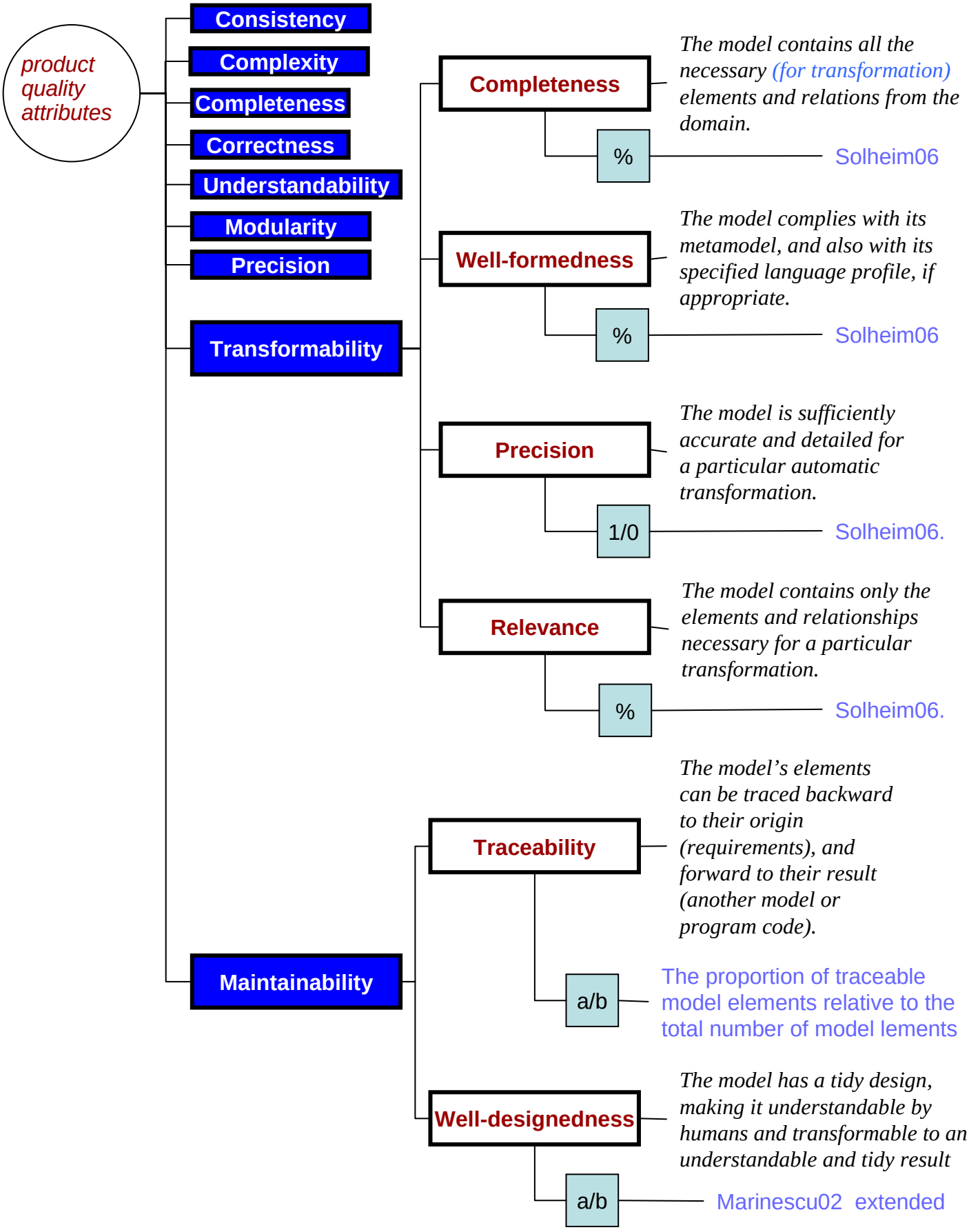
Q2: How do qualities relate? *Dependence*



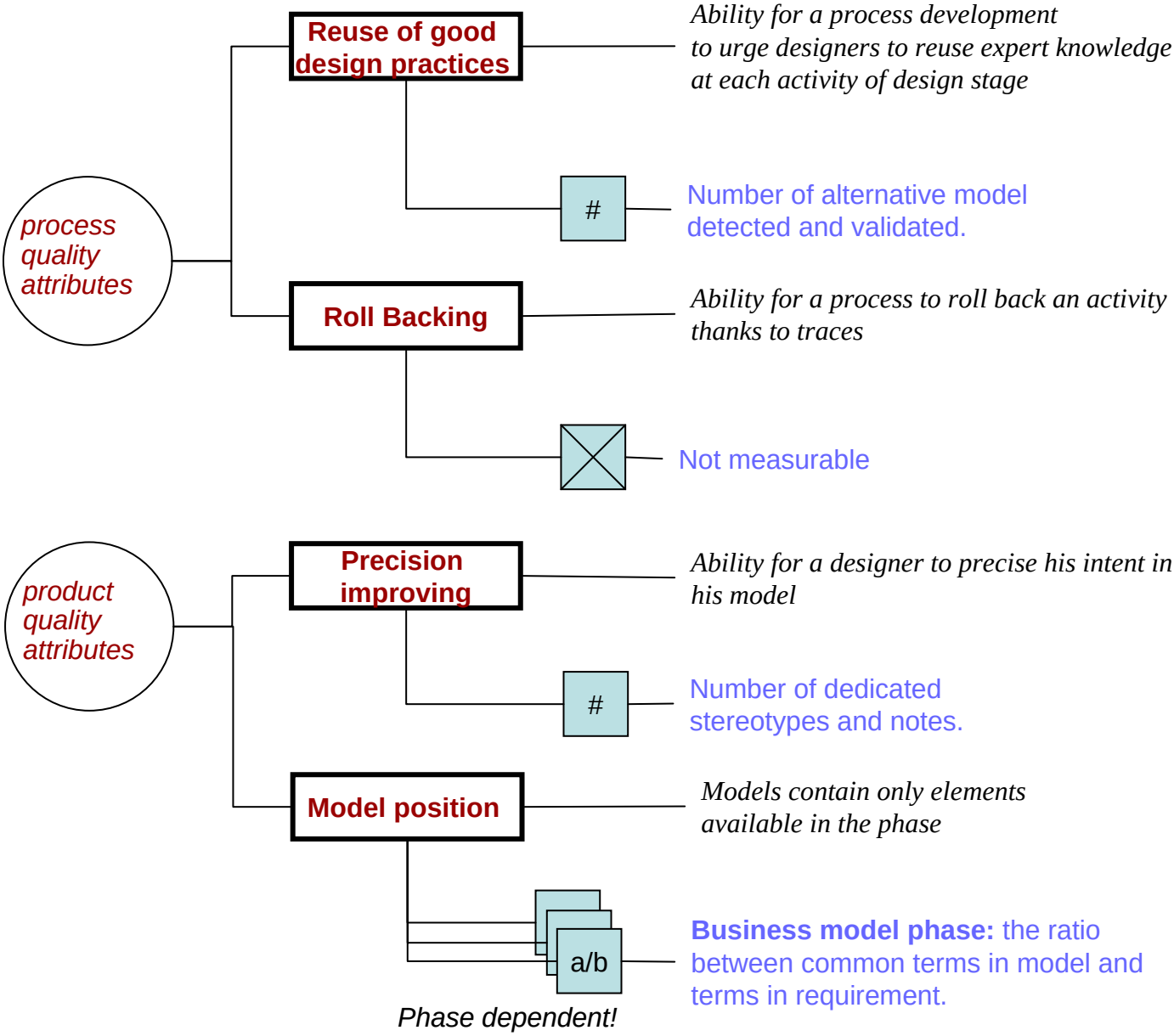
Q3: How can model qualities be measured



Appendix C

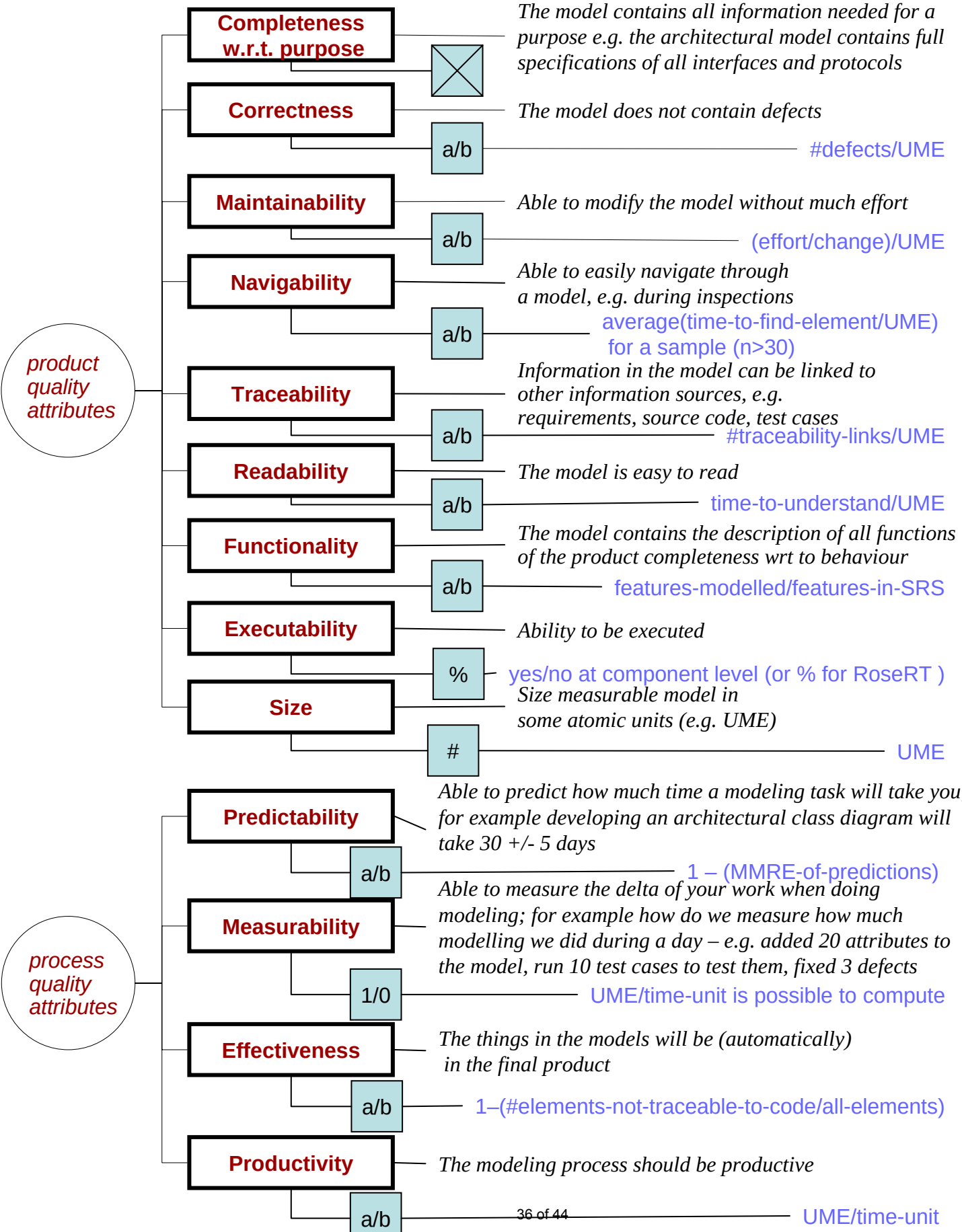

product quality attributes
Consistency
Complexity
Completeness
Correctness
Understandability
Modularity
Precision
Transformability
Maintainability
Completeness
The model contains all the necessary (for transformation) elements and relations from the domain.
%
Solheim06
Well-formedness
The model complies with its metamodel, and also with its specified language profile, if appropriate.
%
Solheim06
Precision
The model is sufficiently accurate and detailed for a particular automatic transformation.
1/0
Solheim06.
Relevance
The model contains only the elements and relationships necessary for a particular transformation.
%
Solheim06.
Traceability
The model's elements can be traced backward to their origin (requirements), and forward to their result (another model or program code).
a/b
The proportion of traceable model elements relative to the total number of model lements
Well-designedness
The model has a tidy design, making it understandable by humans and transformable to an understandable and tidy result
a/b
Marinescu02 extended

Appendix C

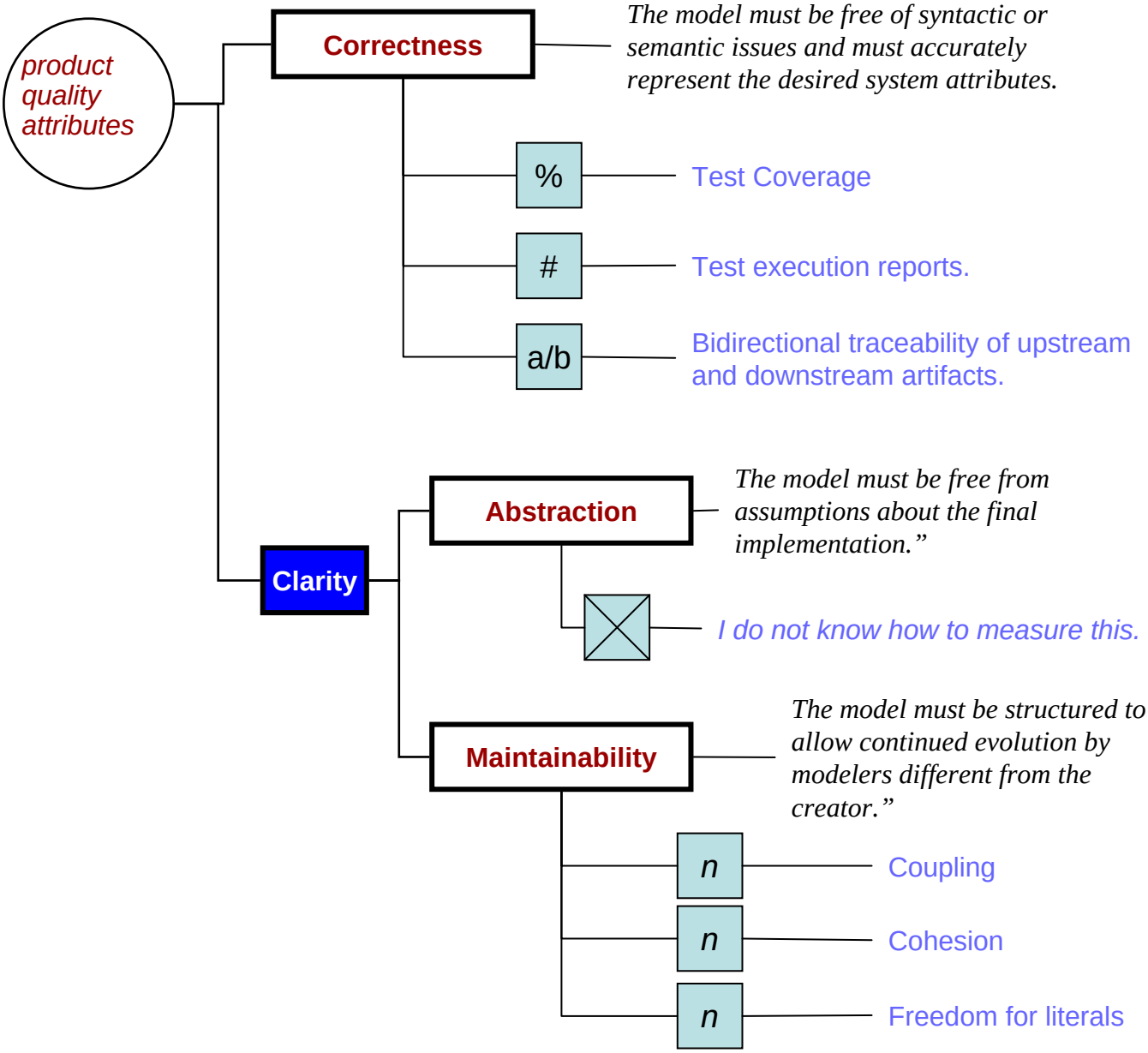

process quality attributes
Reuse of good design practices
Ability for a process development to urge designers to reuse expert knowledge at each activity of design stage
#
Number of alternative model detected and validated.
Roll Backing
Ability for a process to roll back an activity thanks to traces
Not measurable
product quality attributes
Precision improving
Ability for a designer to precise his intent in his model
#
Number of dedicated stereotypes and notes.
Model position
Models contain only elements available in the phase
a/b
Business model phase: the ratio between common terms in model and terms in requirement.
Phase dependent!

Contribution 3 on normal form

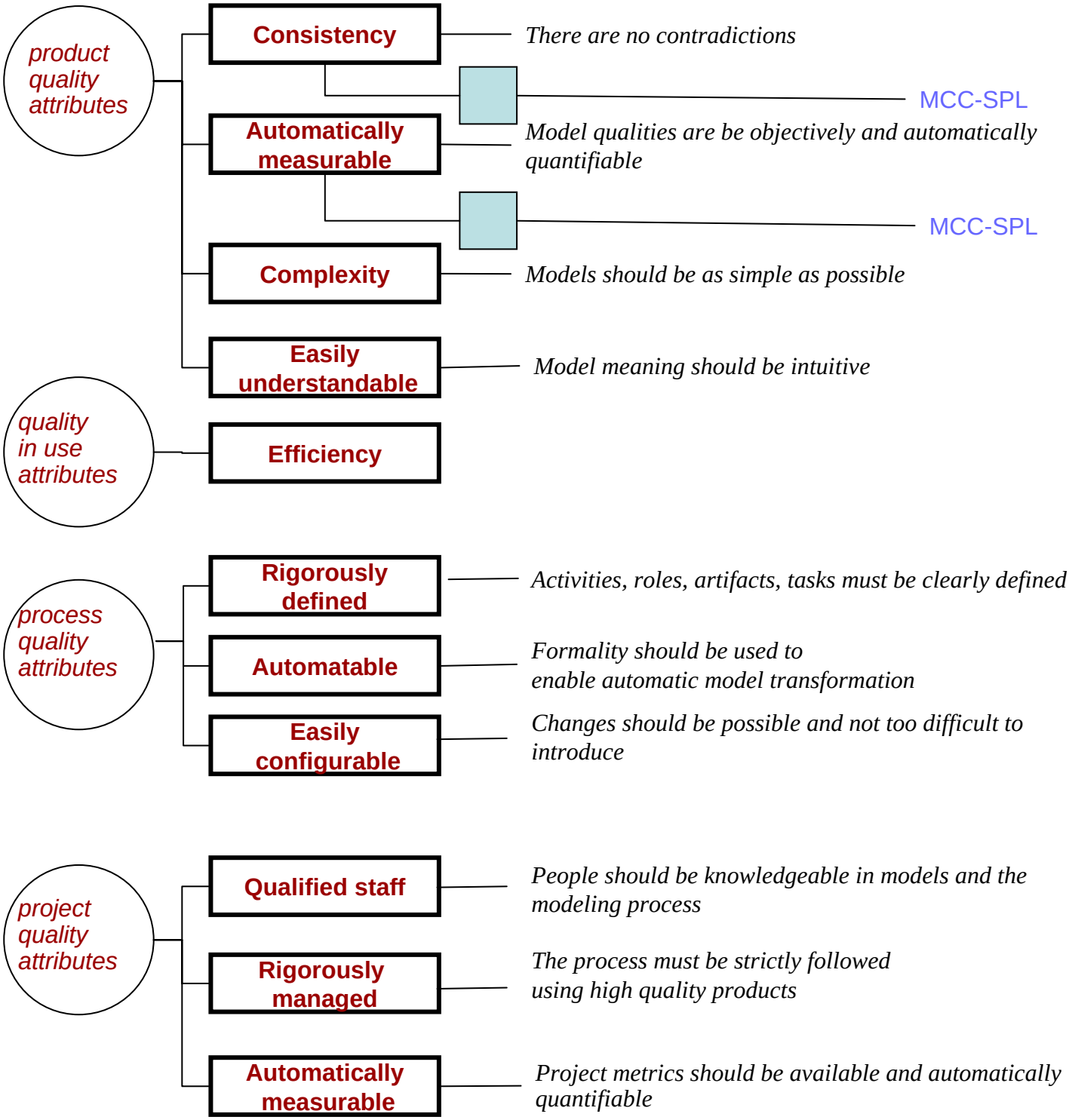
Appendix C



Appendix C

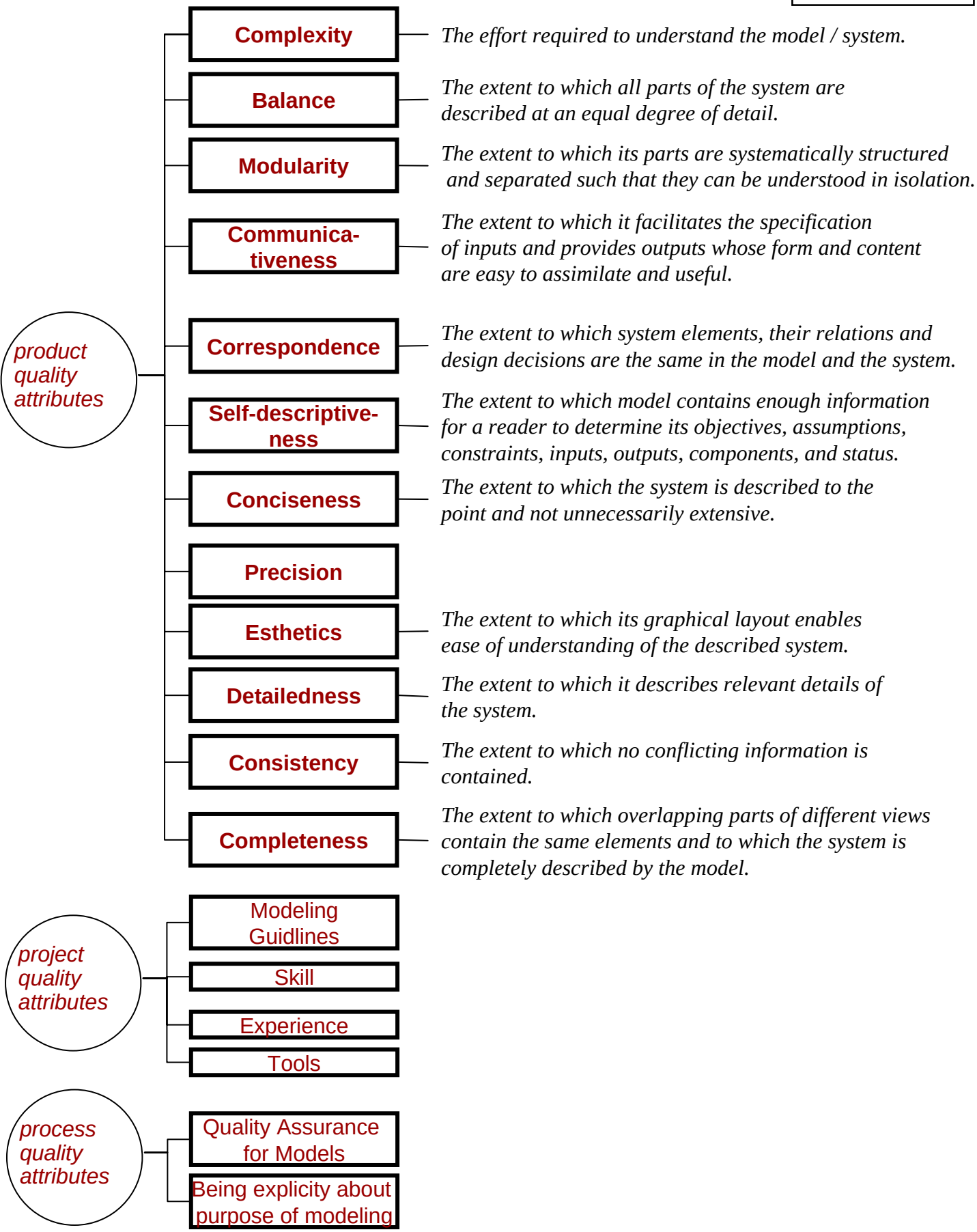

product quality attributes
Correctness
The model must be free of syntactic or semantic issues and must accurately represent the desired system attributes.
%
Test Coverage
#
Test execution reports.
a/b
Bidirectional traceability of upstream and downstream artifacts.
Clarity
Abstraction
The model must be free from assumptions about the final implementation."
I do not know how to measure this.
Maintainability
The model must be structured to allow continued evolution by modelers different from the creator."
n
Coupling
n
Cohesion
n
Freedom for literals

Appendix C

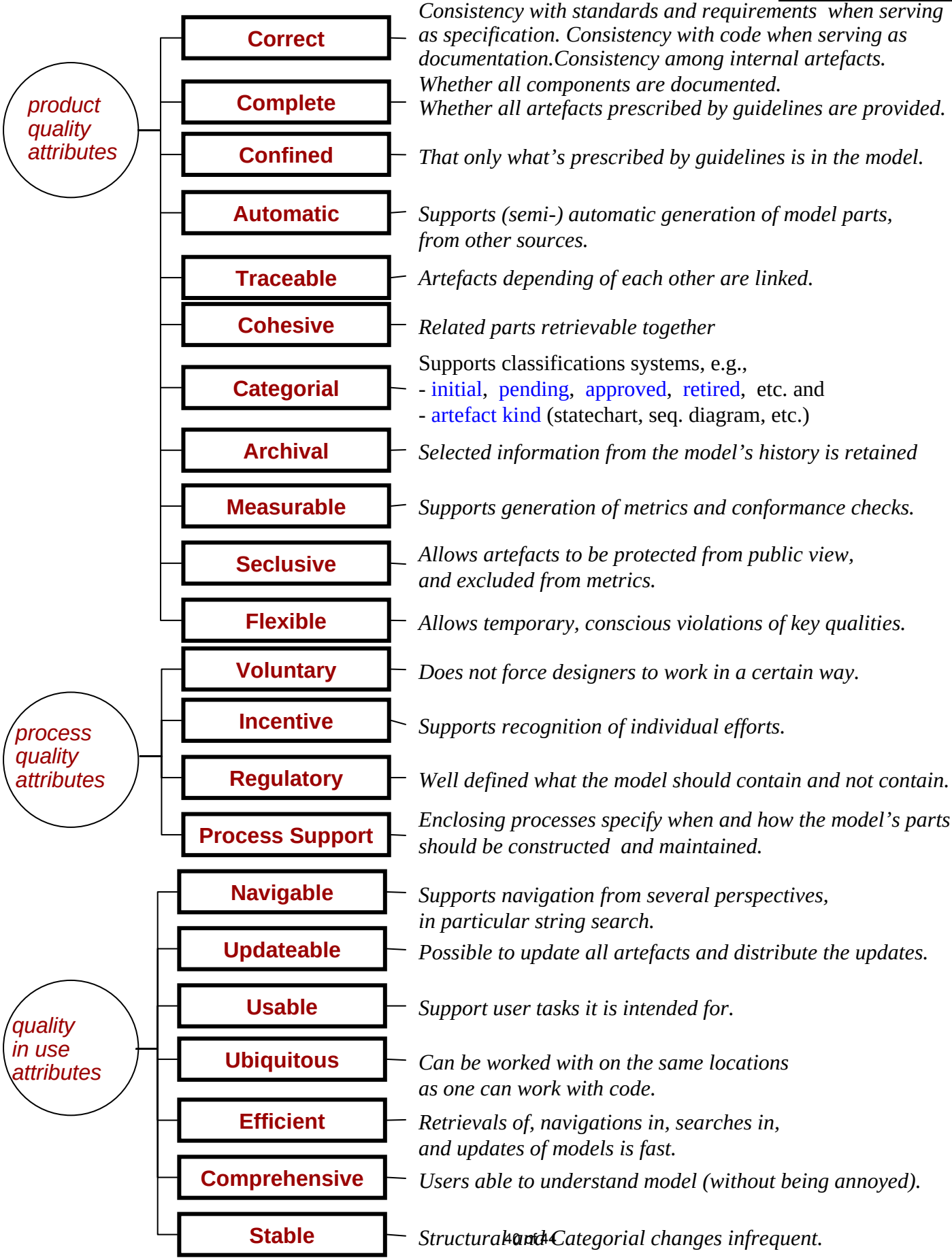

product quality attributes
Consistency
There are no contradictions
MCC-SPL
Automatically measurable
Model qualities are be objectively and automatically quantifiable
MCC-SPL
Complexity
Models should be as simple as possible
Easily understandable
Model meaning should be intuitive
quality in use attributes
Efficiency
process quality attributes
Rigorously defined
Activities, roles, artifacts, tasks must be clearly defined
Automatable
Formality should be used to enable automatic model transformation
Easily configurable
Changes should be possible and not too difficult to introduce
project quality attributes
Qualified staff
People should be knowledgeable in models and the modeling process
Rigorously managed
The process must be strictly followed using high quality products
Automatically measurable
Project metrics should be available and automatically quantifiable

Contribution 6 on normal form

Appendix C



Contribution 7 on normal form



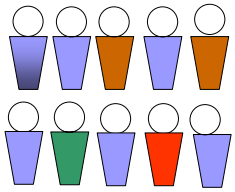
Purpose & Process

Purpose is to establish a *common quality model* for software models and software modeling practice.

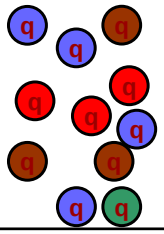
What we are interested in:

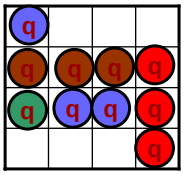
Q1 What qualities of models and modeling matter?
Q2 How do they relate?
Q3 How can they be measured?

Modeling Researchers | **Inspirational framework** | **Qualities** | **Common quality model Techreport**



Available on paper, email, and USB-stick.





Background

Software Manager

"We want to improve our state machine based design practices"

April 2005

Researcher in Applied Model Checking

"State Machine Quality: That's us."

$M, s \models p$

1st Meeting
2nd Meeting

4 senior architects

"We're not modeling at that level of detail."
"Behavioural correctness is not an issue."
"Our main issue is to incorporate NFR:s in our models?"
"This would be something for the people writing the standards we use."
"Protocol verification could be something."

There is more to model quality than $M, s \models p$

Background Part II

Software Designer

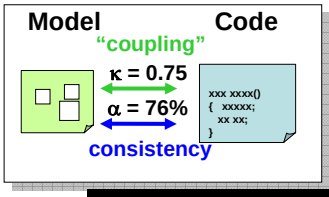
Dec 2005

"Why is it that our design models so often are inconsistent with our code, even though our processes are designed to guarantee consistency?

Our process states that design and code should be consistent…

...but in practice this rarely works."

Idea: Metrics Program!



Interviews

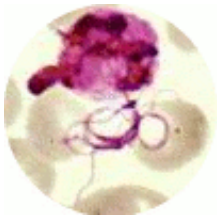
Findings

1. *Ten past attempts, including a consistency checking tool.*
2. *The root cause was **poor usability** **it was not** the lack of a metric.*

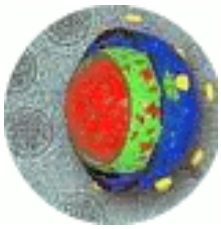
There is more to model quality than artefacts.

Background Part III

We spent ½ year searching for factors that influenced the quality of model artefacts.



X4000



X400

X40

Appendix C

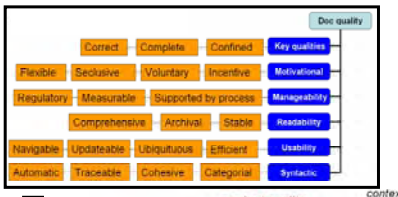
Artefact level
- Process descriptions.
- Design guidelines.
- Review checklists.
- Product metrics.
- Workflows.

People level
- Unmet usability requirments.
- Designer motivation.

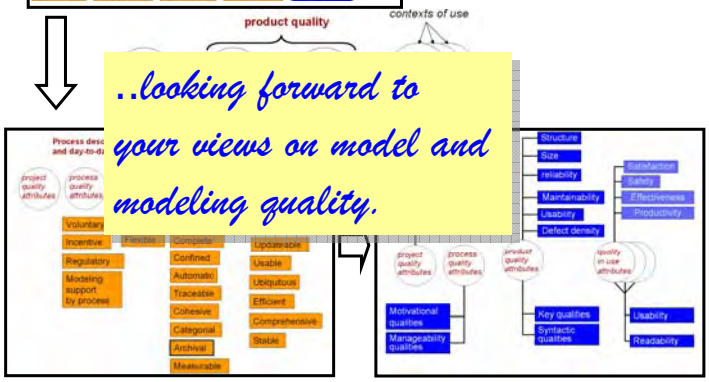
Organisation level
- Incitements.
- Higher level processes.

The Model quality problem solution space is bewildering

Background – part IV



We wrote up the findings in our SOQUA'06 Paper
Pareto & Boquist, *Quality Model for Design Documentation in Model Centric Projects*



..looking forward to your views on model and modeling quality.

Later insight: found qualities fit nicely into the ISO926 quality framework

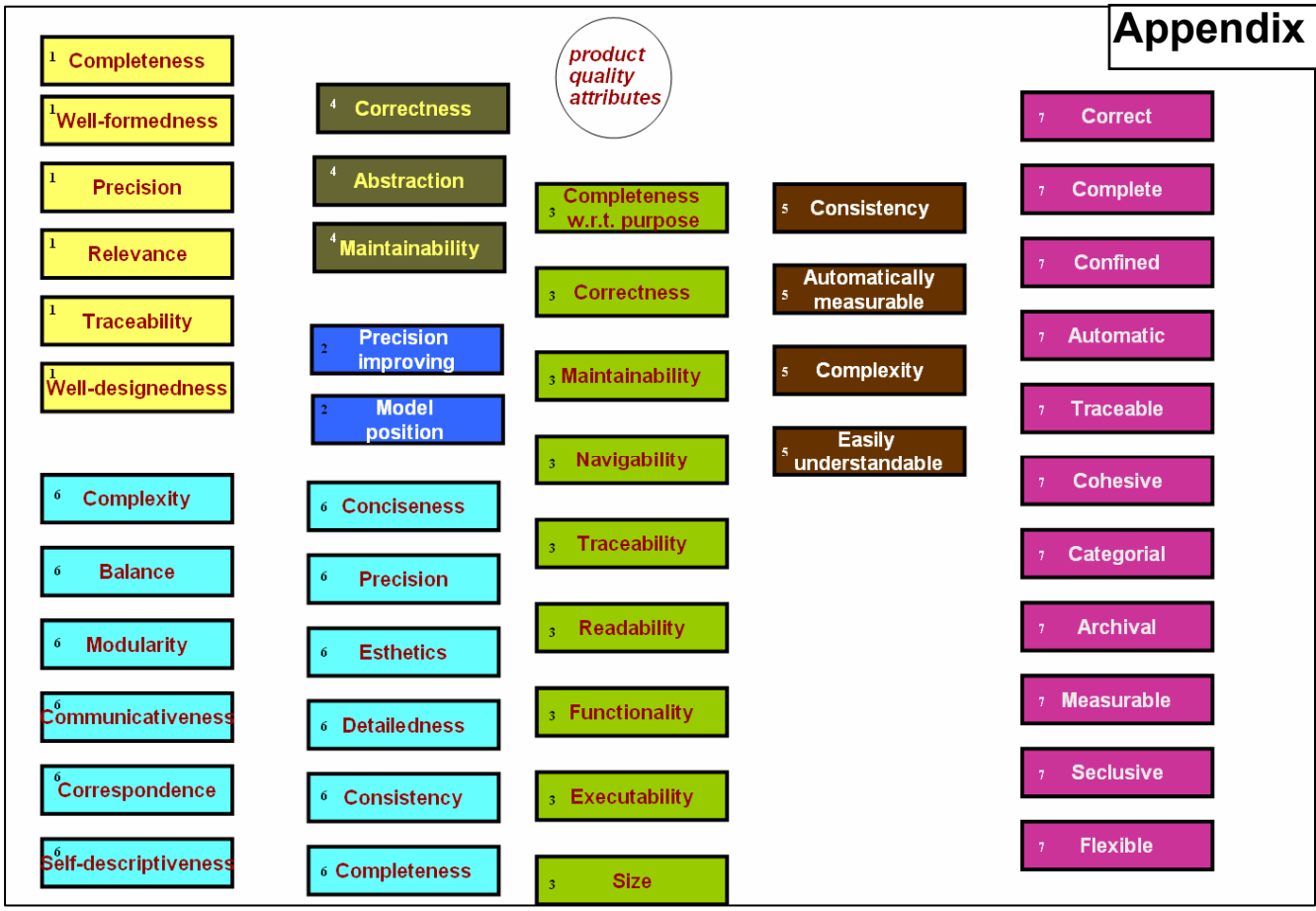
...as do classic qualities of software metrics research.

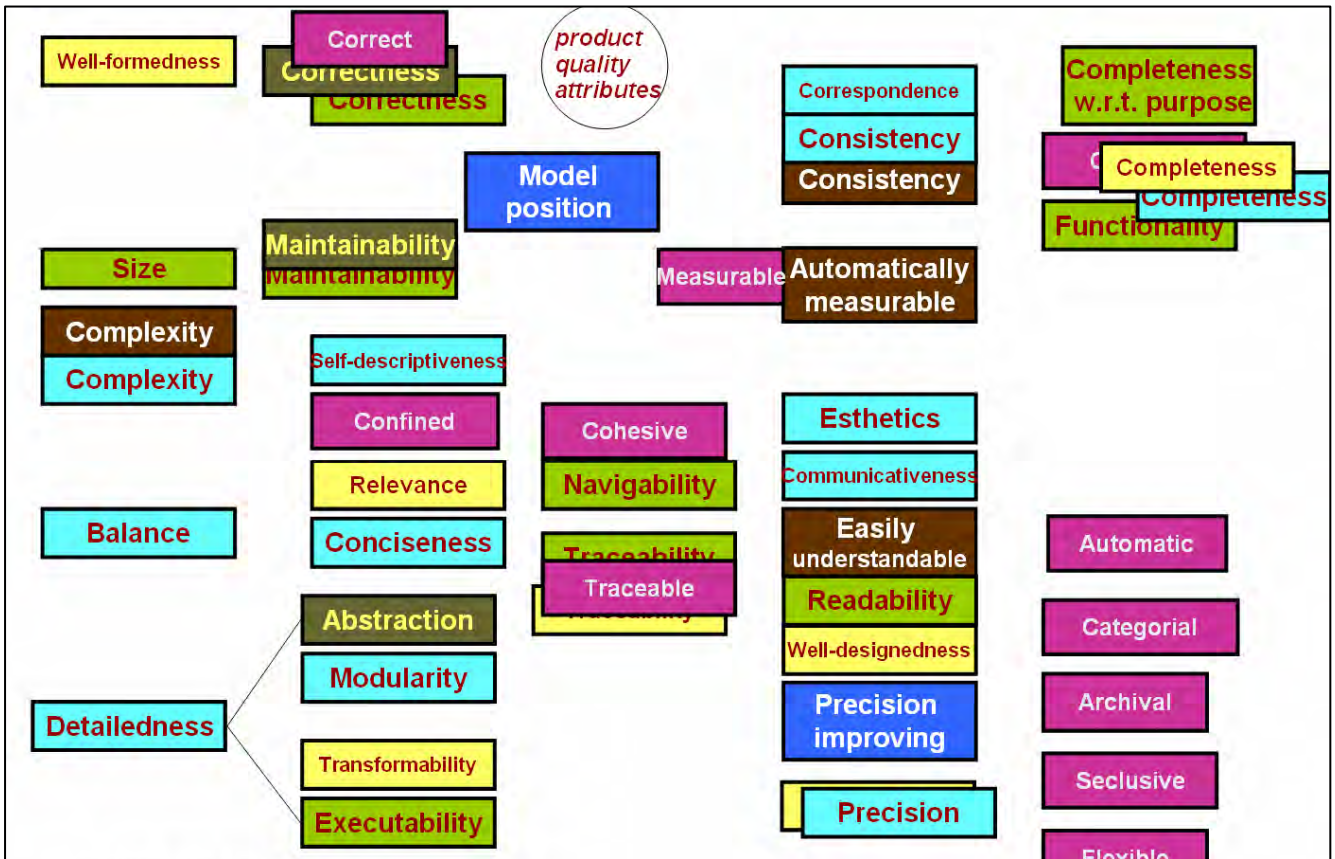
And this is where we are...

Schedule

14:00 – 14:10	**Background**	Lars Pareto
14:10 – 14:20	Contribution 1	Dehlen Vegard [Vegard.Dehlen@sintef.no]
14:20 – 14:30	Contribution 2	Cédric BOUHOURS [bouhours@irit.fr]
14:30 – 14:40	Contribution 3	Miroslaw Staron [miroslaw.staron@ituniv.se]
14:40 – 15:00	Contribution 4	Frank Weil [Frank.Weil@motorola.com]
15:00 – 15:10	Contribution 5	Cecilia Bastarrica [cecilia@dcc.uchile.cl]
15:10 – 15:20	Contribution 6	Lange, C.F.J. [c.f.j.lange@tue.nl]
14:20 – 14:30	Contribution 7	Lars Pareto [lars.pareto@ituniv.se]
14:30 – 15:00	Break	
15:10 – 15:30	Common Model	Lars Pareto
15:30 – 16:30	Discussions	

Appendix E


product quality attributes
1 Completeness
1 Well-formedness
1 Precision
1 Relevance
1 Traceability
1 Well-designedness
6 Complexity
6 Balance
6 Modularity
6 Communicativeness
6 Correspondence
6 Self-descriptiveness
4 Correctness
4 Abstraction
4 Maintainability
2 Precision improving
2 Model position
6 Conciseness
6 Precision
6 Esthetics
6 Detailedness
6 Consistency
6 Completeness
3 Completeness w.r.t. purpose
3 Correctness
3 Maintainability
3 Navigability
3 Traceability
3 Readability
3 Functionality
3 Executability
3 Size
5 Consistency
5 Automatically measurable
5 Complexity
5 Easily understandable
7 Correct
7 Complete
7 Confined
7 Automatic
7 Traceable
7 Cohesive
7 Categorial
7 Archival
7 Measurable
7 Seclusive
7 Flexible


product quality attributes
Well-formedness
Correct
Correctness
Correctness
Model position
Maintainability
Maintainability
Size
Complexity
Complexity
Balance
Detailedness
Abstraction
Modularity
Transformability
Executability
Self-descriptiveness
Confined
Relevance
Conciseness
Cohesive
Navigability
Traceable
Correspondence
Consistency
Consistency
Measurable
Automatically measurable
Esthetics
Communicativeness
Easily understandable
Readability
Well-designedness
Precision improving
Precision
Completeness w.r.t. purpose
Completeness
Completeness
Functionality
Automatic
Categorial
Archival
Seclusive
Flexible
Workshop result: SIC