Maturity Assessment: Modelling the Eclipse Quality

Boris Baldassari
Flavien Huynh

Plan

- Introduction
- Quality Models
- Eclipse Quality Requirements
- Eclipse Quality Model
- Results communication
- Conclusion

Introduction

Polarsys:

- Better control over a project's maturity.
- Clearer quality requirements and evaluation.
- Hardened and enforced Eclipse quality model.

Maisqual:

- Research project focusing on data mining in software engineering.
- Joint project between the INRIA laboratory and Squoring Technologies.

Quality Models

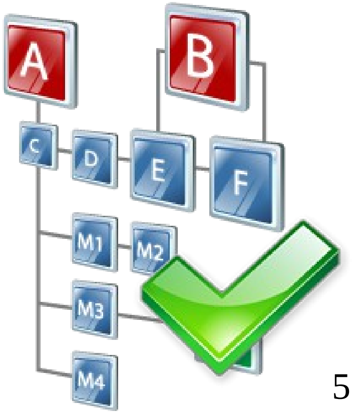
The need for Quality Models

Quality estimation varies with the **subject** (what is evaluated), the **user** (who is evaluating), and the **objective** (what is expected).

Several standards objectively define or measure quality in software engineering.

They usually address:

- **Product Quality** (ISO 9126, ISO SQuaRE, HIS, SQALE, …)
- **Process Quality** (ISO 15504, ISO 9001, CMM)



Eclipse
Quality Requirements

Eclipse Quality Requirements

At present, a project entering the Eclipse ecosystem:

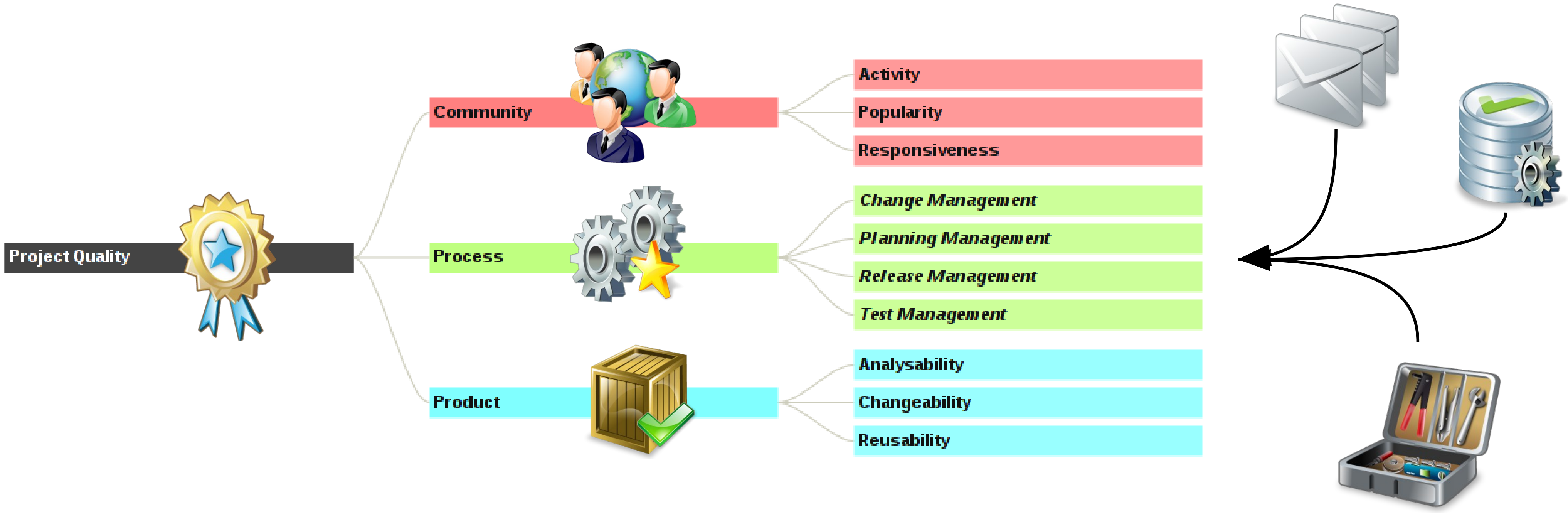
- **Must** comply with strict rules at process level (lifecycle phases),
- **Has to** develop established community channels (ML, Forums …),
- **Can** follow quality recommendations (Checkstyle rulesets, …).

Quality can benefit from an **official** and **comprehensive** definition,

so that any project, new or existing, can **assess** or **monitor** its quality.

Eclipse
Quality Model

Eclipse Quality Model: The Big Picture



Eclipse Quality Model: Rationale

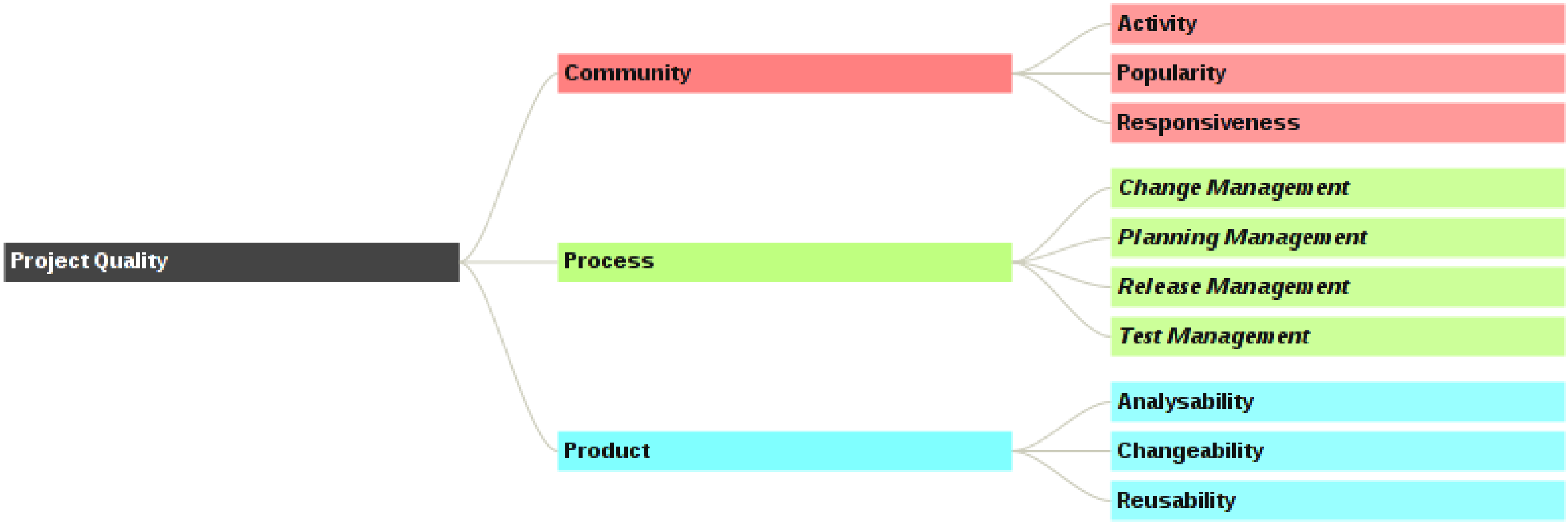
To fit with Eclipse requirements, we propose a model

to assess a project's quality.

This model:

- **Hierarchically** explores the **Community, Process** and **Product** levels.
- Can **assess quality** for any project, in an **automated** and **repetitive** way.
- Is based on **data gathered** from **several repositories** (ML, Forums, SCM, Process, …).
- Is implemented within **Squore**.

Eclipse Quality Model: Top-down approach



Project Quality

Community

Activity

Popularity

Responsiveness

Process

Change Management

Planning Management

Release Management

Test Management

Product

Analysability

Changeability

Reusability

Eclipse Quality Model: Community Quality

Community quality is broken down into 3 sub-characteristics:

- **Activity** is the amount of work achieved in a period of time:
  - number of commits,
  - volume of emails exchanged.
- **Popularity** is the amount and diversity of actors (developers and users):
  - number of committers,
  - number of authors in mailing lists.
- **Responsiveness** is how fast people can get help and answers:
  - number of answers,
  - average time of answers.

Eclipse Quality Model: Process Quality

Process quality definition is a work in progress:

- Certification has specific constraints that need to be further established.
- As sub-characteristics have not all been defined yet, more measures will be needed to evaluate them.

Sub-characteristics identified until now are:

- **Change Management**
- **Release Management**
- **Planning Management**
- **Test Management**

Eclipse Quality Model: Product Quality

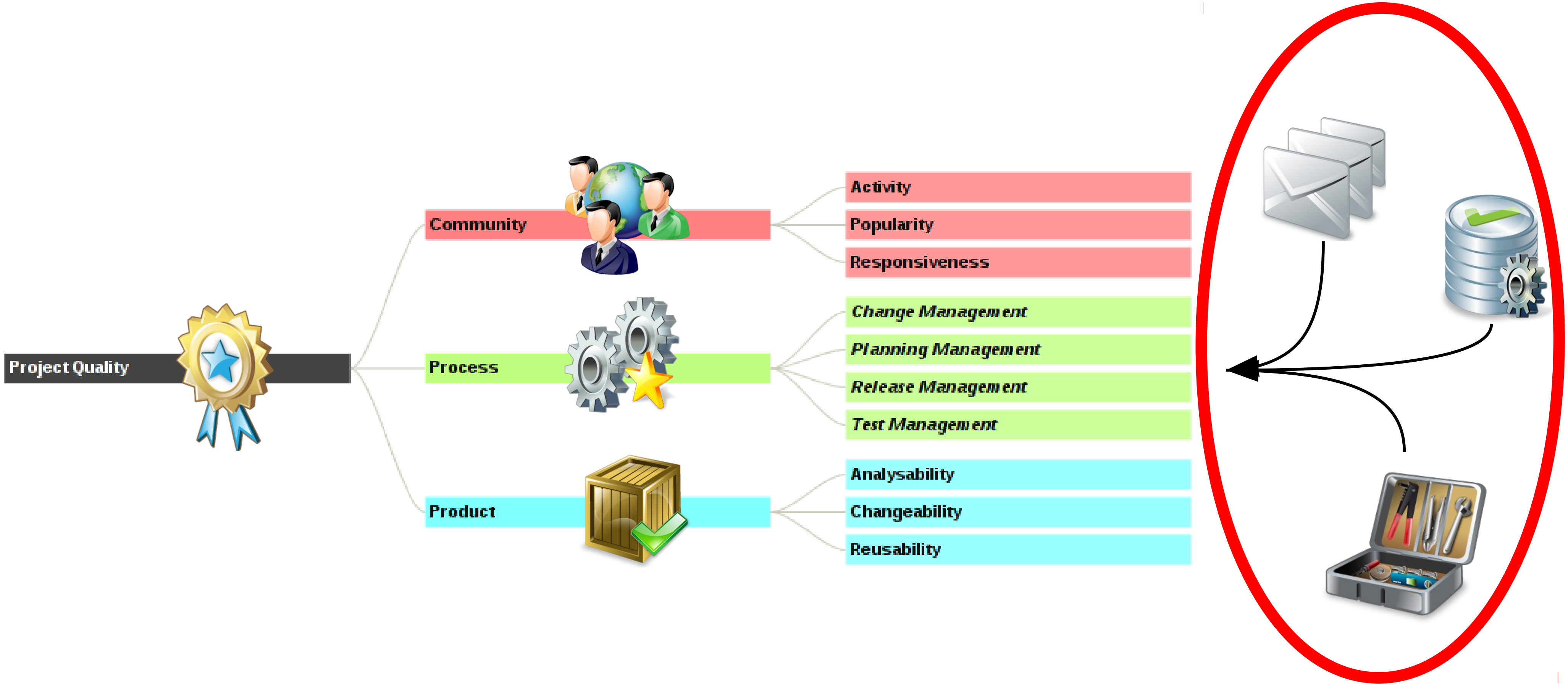
Product quality is associated to Maintainability, per **ISO 9126** definition.

The following characteristics have been mapped to Eclipse Requirements:

- **Reusability:** degree to which an asset can be used in more than one system, or in building other assets.
- **Analysability:** degree of effectiveness and efficiency to assess the impact of an intended change.
- **Changeability:** degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing quality.

The model takes into account measures such as **bad practices** (acquired, almost acquired, lost), i**ntrinsic characteristics** (data-flow complexity, size, nesting ...)

Eclipse Quality Model: The Big Picture



Eclipse Quality Model: Data Providers (1)

The “Data Provider” term designates a module which goal is to gather data from an external source, process it and generate useful measures to provide the model with.

Several Data Providers are used by the Eclipse Model, more can be added to help better compute its attributes.

Mailing lists / forums Data Provider:

- number of posts, of authors, of distinct threads,
- number of answers, median time to answer.

Metrics are computed for last week, last month, and last 3 months.

Eclipse Quality Model: Data Providers (2)

SCM (Subversion) metadata Data Provider:

- number of commits (File & Application levels),
- number of committers (File & Application levels),
- number of committed files (Application level),
- ratio of fix-related commits (File & Application levels).

Metrics are computed for last week, last month, and last 3 months

Eclipse Quality Model: Data Providers (3)

Process Data Provider:

- The Eclipse foundation has initiated a repository to automatically retrieve process information:
  - **number of milestones**,
  - **number of reviews**,
  - **number of themes** (work item categories),
  - **number of requirements** (Bugzilla change requests),
  - **IP logs**.

Eclipse Quality Model: Data Providers (4)

Squore analyser Data Provider:

This tool is able to produce

- **measures**: e.g. McCabe, Halstead metrics, nesting ...
- **bad practices**: e.g. missing default, no assignment in conditions ...
- **cloning information**.

CheckStyle Data Provider:

Produces bad practices detection (e.g. coding and naming conventions).

For the Eclipse Quality model, the Sun Coding Conventions Bundle is used.

Eclipse Quality Model: Data Providers (5)

Other Product Data Providers:

As for Squore or Checkstyle, other Data Providers have been developed to interface the Eclipse Quality model with multiple analysis tools:

- PMD
- FindBugs
- Sonar
- ...

Results communication

Presenting analysis results

For maximum efficiency, we will:

- Publish the detailed **quality model**, from quality characteristics and sub-characteristics to metrics used.
- Provide **pragmatic advice** for quality improvement and good practices adoption.
- Publish the results in a **centralised dashboard**: developers and users should have all relevant information at a glance.

Conclusion

Conclusion

The Maisqual Quality model intends to offer:

- **Clear definitions** of characteristics and sub-characteristics of quality in the Eclipse Way – Top-down approach.
- An **objective and transparent rating** built from publicly available metrics – Bottom-up approach.
- Easy and **automatic retrieval of data** for continuous inspection and improvement of quality.

Conclusion

This is only the beginning of the journey:

- Discuss and get a general agreement on quality requirements with Eclipse and Polarsys actors.
- Add more data sources, e.g. bug tracking system, website and download statistics...
- Quality is everyone's concern and responsibility.
  Make it a real concern for Eclipse projects.

Thank you for your interest!

More information on:
http://maisqual.squoring.com/wiki/index.php/Eclipse

Boris.baldassari@squoring.com
Flavien.huynh@squoring.com

References

- Garvin, David A. Managing quality: “The strategic and competitive edge.” Free Pr, 1988.
- Haaland, K., Groven, A., Regnesentral, N., Glott, R., & Tannenberg, A. (2010). “Free/Libre Open Source Quality Models - a comparison between two approaches.” Software Engineering and Advanced Applications, Euromicro Conference, 0(180054), 439–446.
- Ing, Marc-Alexis Côté M., and Elli Georgiadou. “Software Quality Model Requirements for Software Quality Engineering.”
- Jamwal, R., Jamwal, D., & Padha, D. (n.d.). “Comparative Analysis of Different Software Quality Models.”
- Kan, S. H. (2002). “Metrics and Models in Software Quality Engineering (2nd ed.).” Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Kitchenham, B., & Pfleeger, S. (1996). “Software quality: the elusive target.” Software, IEEE, 13(1), 12–21.

References

- Eclipse Quality: http://wiki.eclipse.org/Eclipse_Quality
- Eclipse Development Conventions and Guidelines: http://wiki.eclipse.org/Development_Conventions_and_Guidelines
- Eclipse Development resources: http://wiki.eclipse.org/Development_Resources
- Polarsys official website: http://www.polarsys.org
- Maisqual research project: http://maisqual.squoring.com