

Towards a Metric for Open Source Software Quality

Siraj A. Shaikh

Cranfield University

s.shaikh@cranfield.ac.uk

Antonio Cerone

United Nations University

antonio@iist.unu.edu

Raymond's View

The *high level* of quality of free software is partly due to the high degree of peer review and user involvement

[Raymond 1999]

Raymond's View

The *high level* of quality of free software is partly due to the high degree of peer review and user involvement

[Raymond 1999]

This **hypothesis**

- is based on **anecdotal rather than empirical evidence**

Raymond's View

The *high level* of quality of free software is partly due to the high degree of peer review and user involvement

[Raymond 1999]

This **hypothesis**

- is based on **anecdotal rather than empirical evidence**
- needs to be validated and tested rigorously **through empirical data analysis**

Traditional Notions of Quality

the degree to which a system, component, or process meets (1) specified requirements, (2) customers or user needs or expectations

[IEEE 1999]

Traditional Notions of Quality

the degree to which a system, component, or process meets (1) specified requirements, (2) customers or user needs or expectations

[IEEE 1999]

conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software

[Pressman 2000]

Metric for OSS Quality

This work attempts to

- identify **notions of quality** for open source software

Metric for OSS Quality

This work attempts to

- identify **notions of quality** for open source software
- understand the **factors** contributing to quality of such software

Metric for OSS Quality

This work attempts to

- identify **notions of quality** for open source software
- understand the **factors** contributing to quality of such software
- discover **relationships** among factors

Metric for OSS Quality

This work attempts to

- identify **notions of quality** for open source software
- understand the **factors** contributing to quality of such software
- discover **relationships** among factors
- model these relationships to **derive a metric** for quality

Open Source Methodology

based on

Open Source Methodology

based on

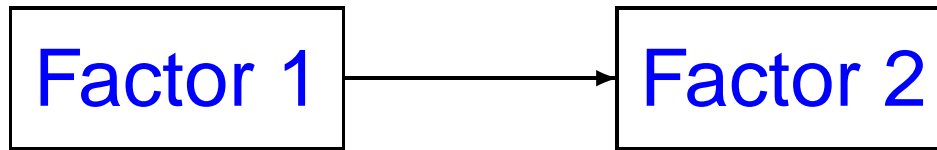
- frequent beta release
- decentralised contribution to a code base
- accessibility (to code) as open as possible
- extensive peer review
- ignoring the traditional software models (?)

Three Notions of Quality

Quality by

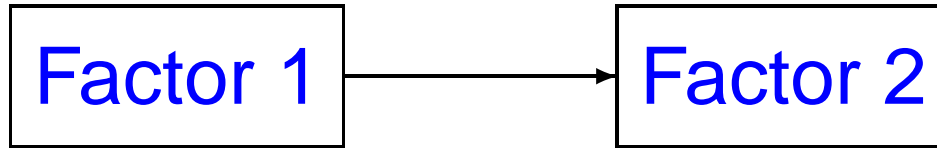
- **Access**
 - open access, extensive peer review
- **Development**
 - frequent beta release, decentralised contribution
- **Design**
 - good design and documentation
 - traditional notions of quality (?)

Relationships between Factors

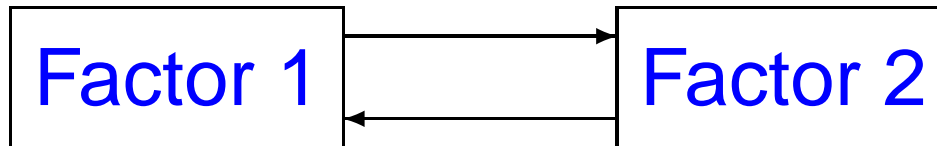


Quality provided by **Factor 2** is dependent on quality provided by **Factor 1**

Relationships between Factors



Quality provided by **Factor 2** is dependent on quality provided by **Factor 1**



Quality provided by **both factors** complement each other

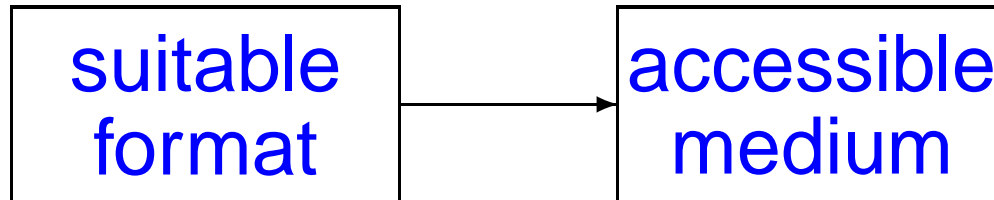
Quality by Access

Quality by Access

suitable
format

- attractive for developers
- workable for development and review
- appropriate for free distribution

Quality by Access



- the Internet

by means of

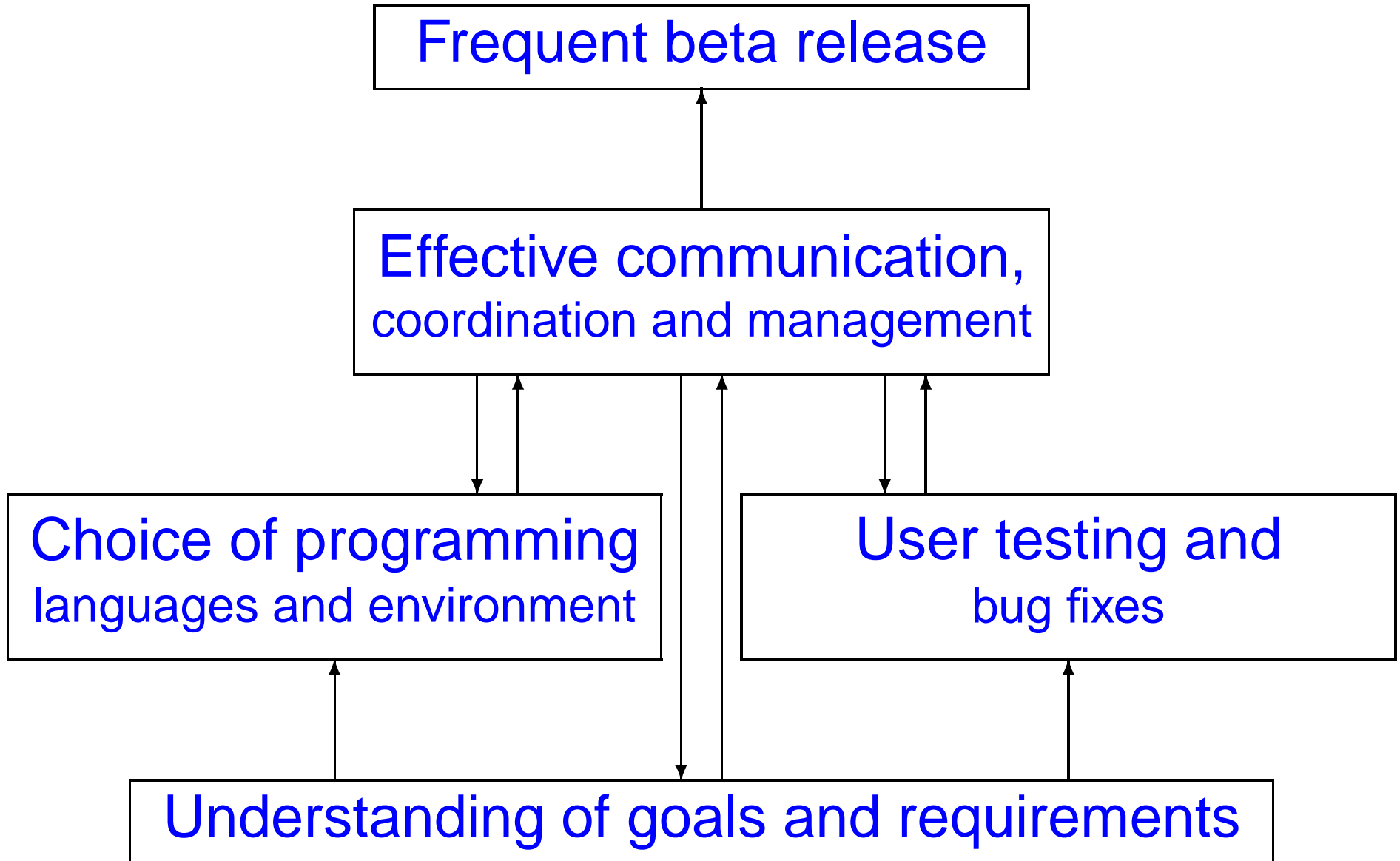
- project web server
- bug tracking systems
- version control systems
- mailing lists
- discussion forum

Quality by Access

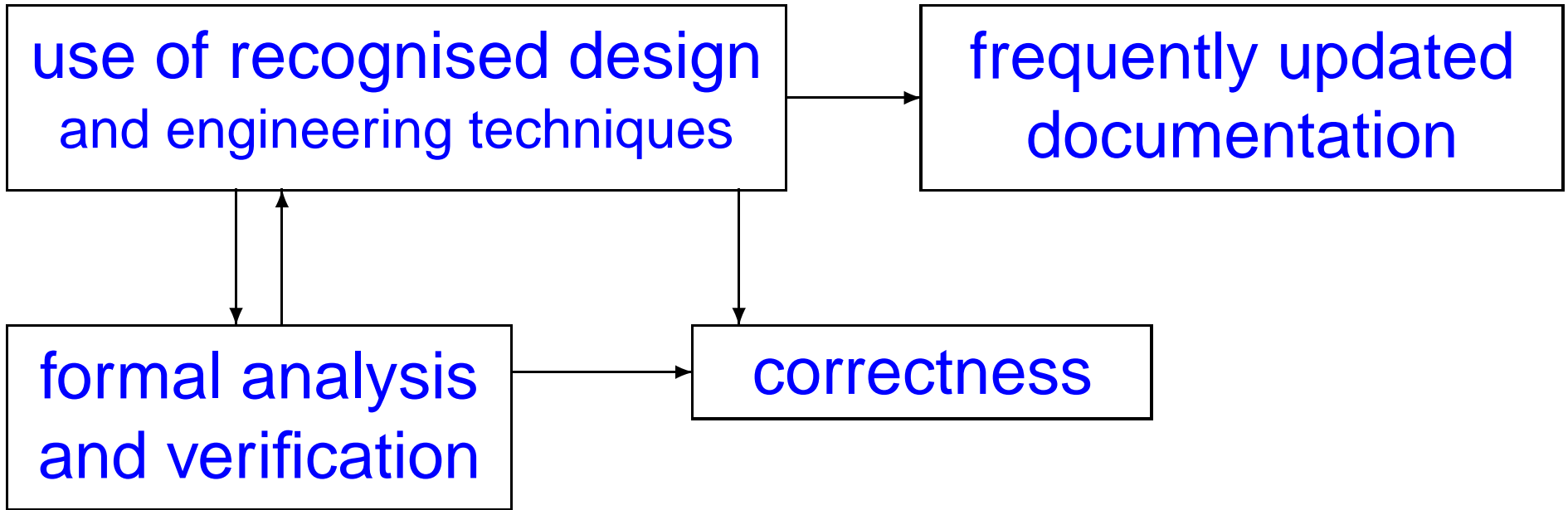


- maximise outgoing information flow

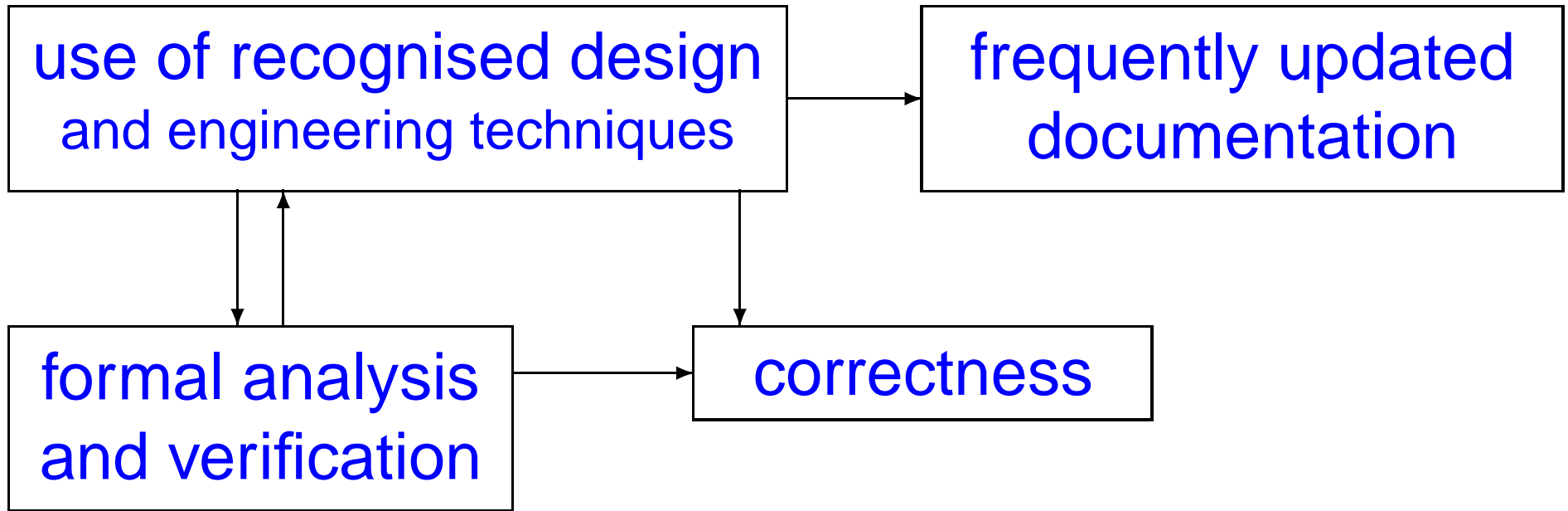
Quality by Development



Quality by Design



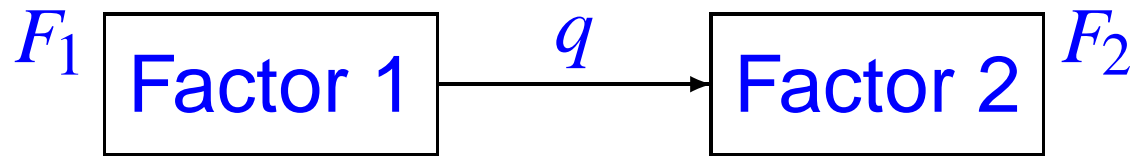
Quality by Design



Factors for traditional software quality

- good software engineering practices
- formal techniques for design and development
- up-to-date documentation

Deriving a Metric for Quality



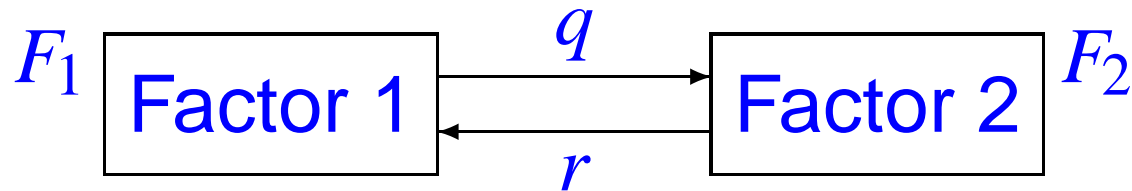
Every factor is assigned an index value between 0.1 and 1.0, greater than 0 and not greater than 1.0. The value corresponds to the quality provided by the factors for a given software project, where the higher the index value the higher the quality provided by a factor.

For the above relation, we calculate the product of the indices, such that

$$F_1 \times (F_2 + (F_1 \times q))$$

where q represents the proportion of quality provided by Factor 2 enhanced due to the quality provided by Factor 1, with $0 \leq q \leq 1$.

Deriving a Metric for Quality (2)



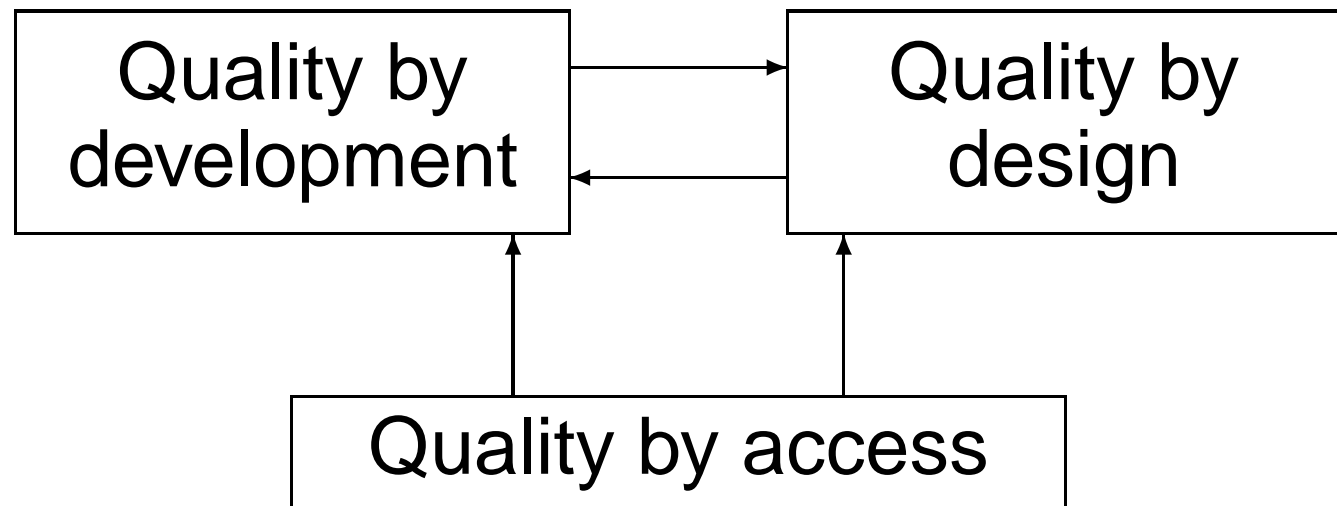
For a relation where the quality provided by both factors complements each other, we calculate the overall metric as follows

$$(F_1 + (F_2 \times r)) \times (F_2 + (F_1 \times q))$$

where r represents the proportion of quality provided by Factor 1 enhanced due to the quality provided by Factor 2 and q is the corresponding proportion of quality enhanced for Factor 2 due to the quality provided by Factor 1, with $0 \leq r, q \leq 1$.

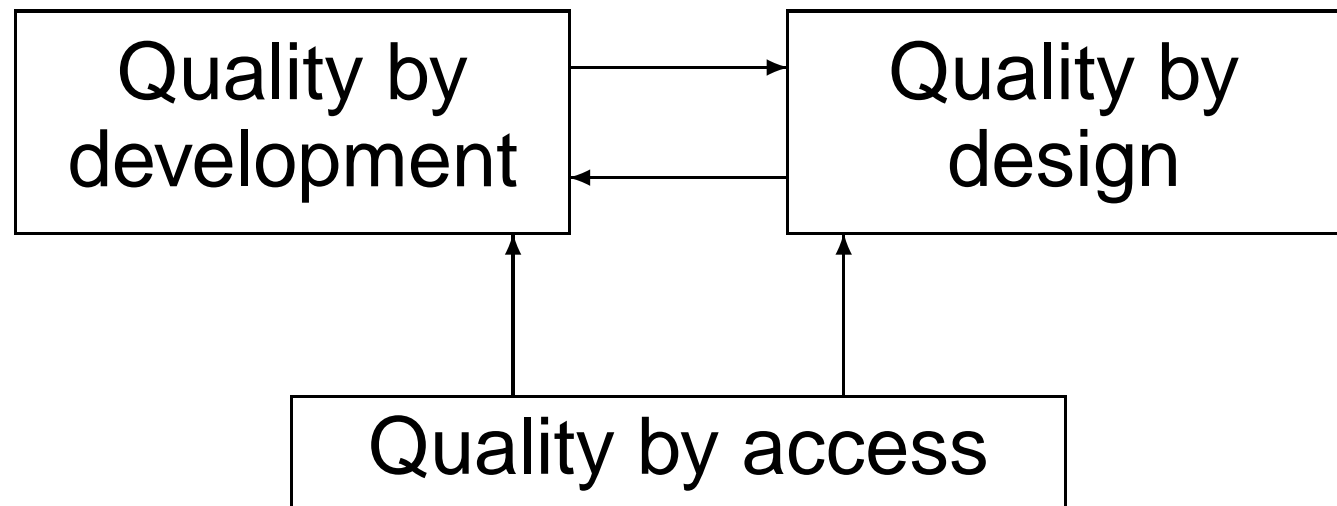
Ongoing Work

A metric is derived for the entire model
where



Ongoing Work

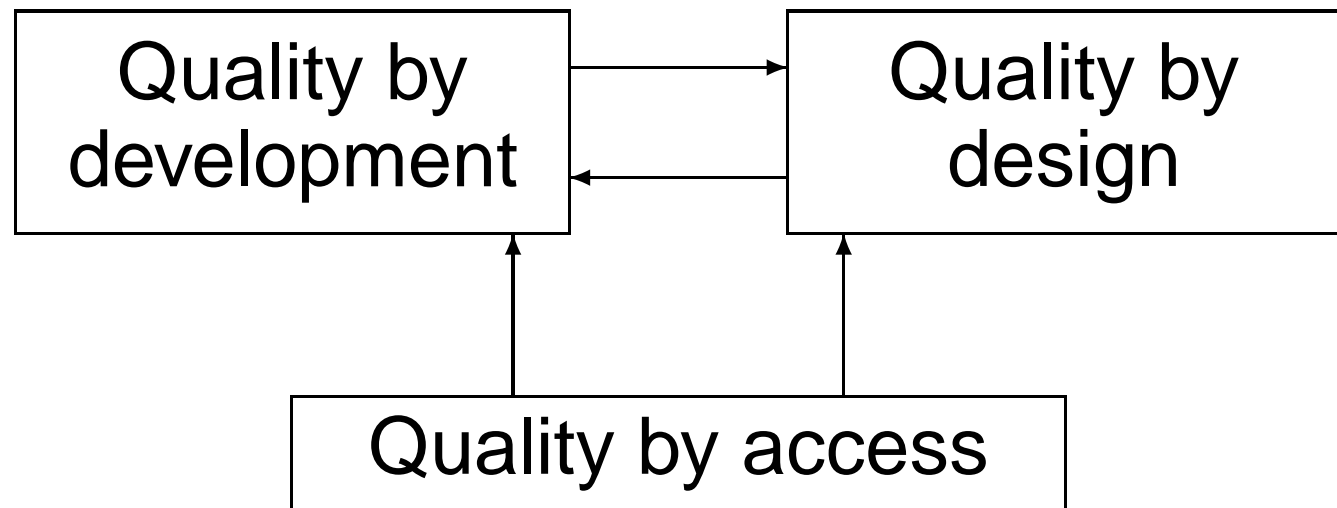
A metric is derived for the entire model where



- Assess quality for existing open source projects

Ongoing Work

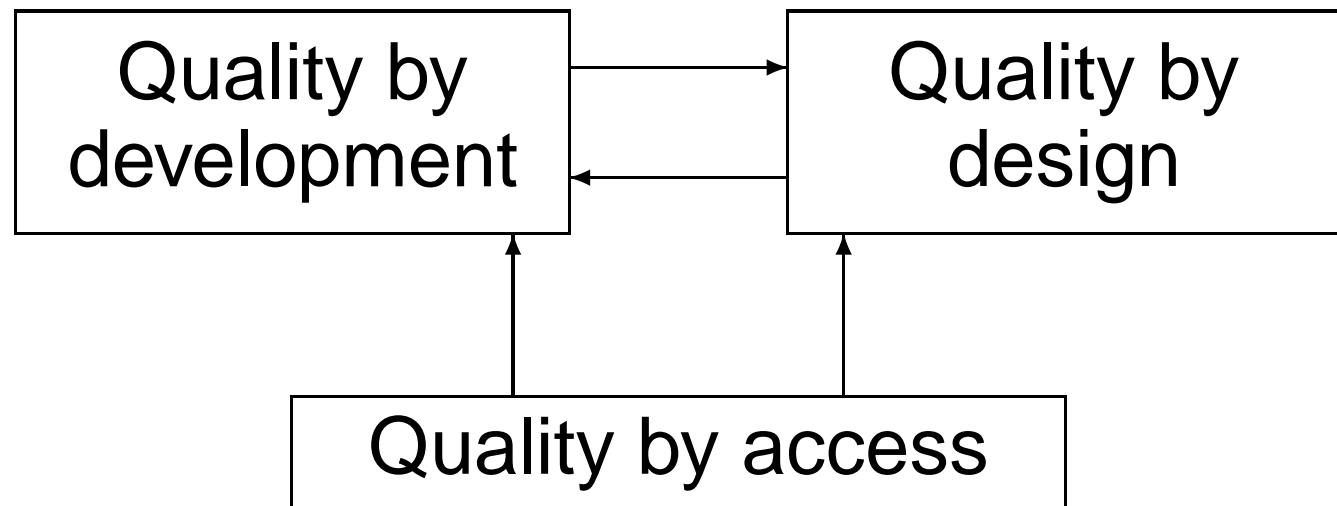
A metric is derived for the entire model where



- Assess quality for existing open source projects
- Refine quality factors and their relationships further

Ongoing Work

A metric is derived for the entire model where



- Assess quality for existing open source projects
- Refine quality factors and their relationships further
- Identify factors to be added/modified/deleted

Conclusion

- Quality assessment that covers both open source and traditional attributes
⇒ room for more factors
- A generic model
⇒ flexible relationships between factors

Conclusion

- Quality assessment that covers both open source and traditional attributes
⇒ room for more factors
- A generic model
⇒ flexible relationships between factors

Challenges

- How do we decide on a quality index?
- Should we consider domain specific attributes?

References

- [IEEE 1999]
IEEE STD 610.12–1990
IEEE Standard Glossary of Software Engineering Terminology
- [Raymond 1999]
E. S. Raymond
The Cathedral and the Bazaar
O'Reilly and Associates, 1999
- [Pressman 2000]
S. R. Pressman
Software Engineering — A Practitioner's Approach
McGraw-Hill International, 2000