Software

Nature & Qualities

Himadri Barman

Centre for Management Studies, Dibrugarh University

Reference:

Ghezzi Carlo, Jajayeri Mehdi, Mandrioli Dino, *Fundamentals of Software Engineering*, c1991 Prentice-Hall Inc.

S/W Development – An Engineering Activity

How it differs?

- Malleable – Modify the product itself – as opposed to its design – *rather* easily
- Software creation is human intensive: it requires mostly engineering rather than manufacturing. Here manufacturing is just the *trivial* process of duplication

The s/w production process deals with design & implementation, rather than manufacturing

Classification of S/W Qualities

Two different approaches:

- ✓ External v/s Internal
- ✓ Product v/s Process

The external qualities are visible to the users of the system; the internal qualities are those that concern the developers of the system

The internal quality of reliability is necessary for achieving the external quality of reliability – **Distinction not sharp**

We use a process or a set of processes to produce the s/w product. Both are inter-related. *Process quality* ***enhances*** *product reliability*.

Configuration Management

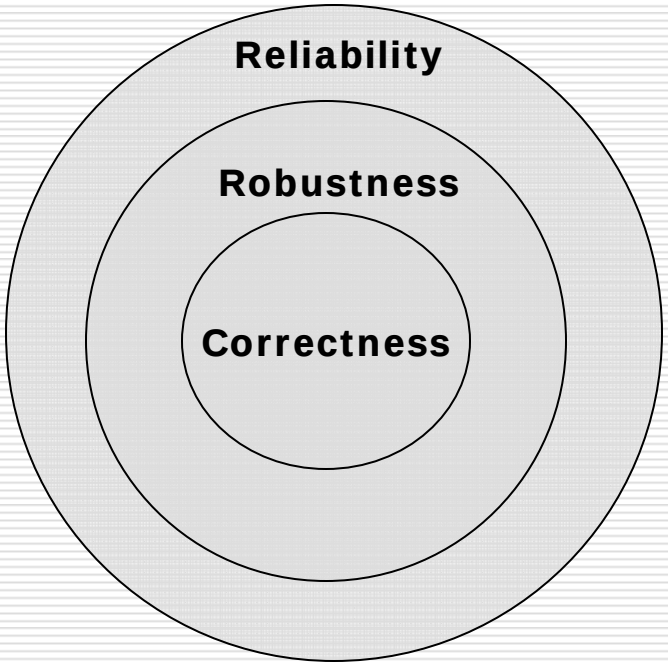
Part of the s/w production process that is concerned with maintaining and controlling the relationship between all the related pieces of the various versions of the product

We select a set of qualities called *representative* qualities to determine and assure the quality of a software product.

What are these?

Correctness , Reliability & Robustness

Process **Product** **Internal** **External**



Ideal Case

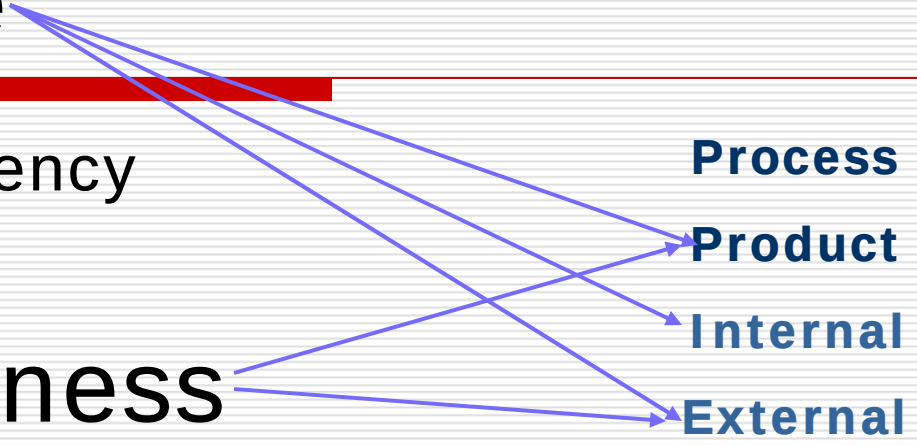
- Satisfies/behaves according to the specifications
- A mathematical property establishing an absolute quantity
- An equivalence relation between the product and its specifications

Performance

- ✓Equated with efficiency

User Friendliness

- ✓Subjective – differs in different systems, for e.g., in an embedded system, user friendliness is reflected in the ease with which the systems can be configured and adapted to the h/w environment



Verifiability

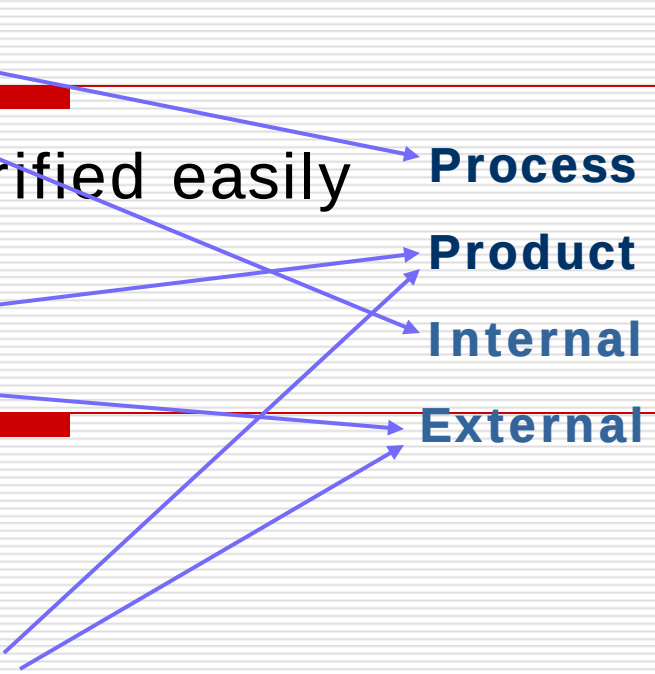
- ✓ If its properties can be verified easily

Reusability

- ✓ Characterizes maturity

Understandability

- ✓ Characterizes user friendliness (external view point)



Productivity

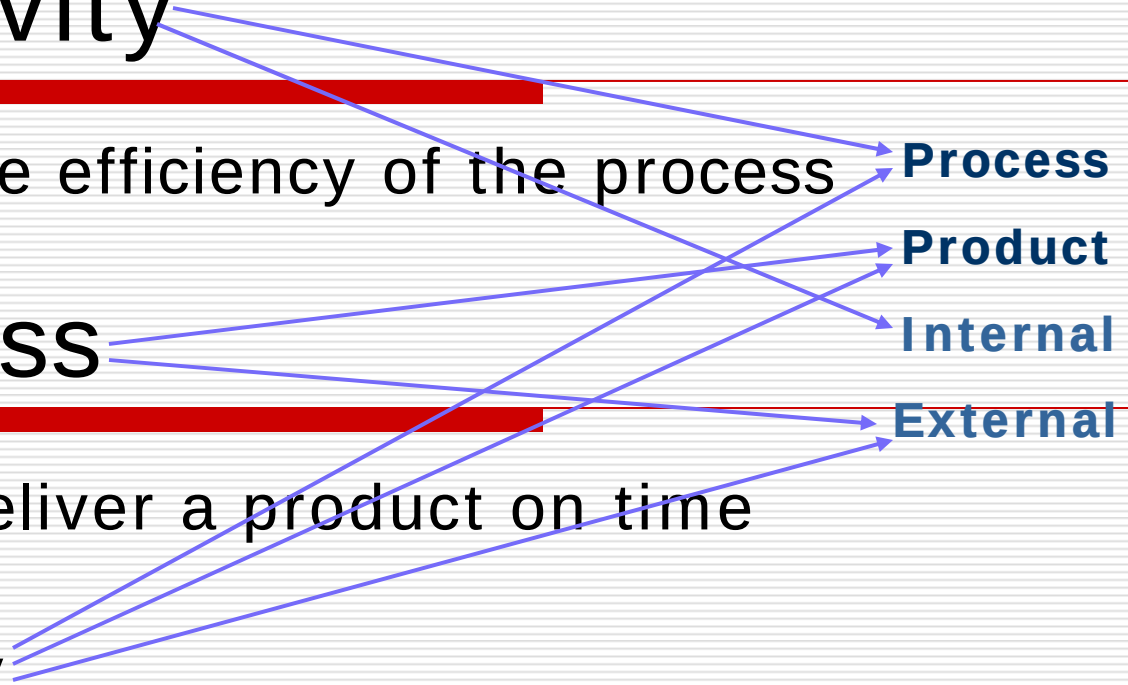
- ✓ Measures the efficiency of the process

Timeliness

- ✓ Ability to deliver a product on time

Visibility

- ✓ Glasnost (transparency or openness) – all steps and its current status are documented clearly

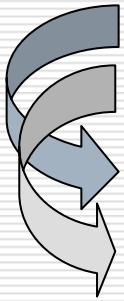


Maintainability

Process
Product
Internal
External

- ✓ S/W maintenance is divided into three categories
  - •Corrective – fixing bugs
  - •Adaptive
  - •Perceptive

sources of change



adjusting the application to changes in the environment

involves changing the s/w to improve some of its qualities – add new functions, make it easier to use, etc.

Maintainability has two measurable dimensions

Reparability – if a s/w system allows the correction of its defects with a limited amount of work

Evolvability – version releases

Portability

- Process
- Product
- Internal
- External

- ✓ s/w is portable if it can run in different environments (hardware platforms) or different O/S s
- ✓ o/s provides portability across h/w platforms

Interoperability

- ✓ the ability of a system to coexist and cooperate with other systems

How to achieve?

- *Correctness*

  Experimental Approach (testing)

  Analytical Approach (formal verification)

  - using HLL
  - using standard algorithms
  - using libraries of standard modules
- *Reliability*
  - extensive testing
- *Robustness*
  - coding taking into consideration all possible errors
  - extensive testing

How to achieve? (Contd...)

- *Performance*

  three approaches

  - measurement
  - analysis
  - simulation

- *User friendliness*

  - standardization of the human interface

- *Portability*

  - use techniques that allow the s/w to determine the capability of the h/w and adapt to them

How to achieve? (Contd...)

- *Repairability*
  - *modular architecture*
  - use of proper tools
- *Evolvability*
  - use of proper tools
- *Reusability*
  - use of proper tools
- *Verifiability (Understandability)*
  - modular design
  - disciplined coding practices
  - use of an appropriate programming language

How to achieve? (Contd...)

- *Interoperability*
  - *standardization of Interface*
- *Productivity*
  - work to plan
  - leaving scope for contingencies
- *Timeliness*
  - incremental delivery of the product (must be combined with other software qualities)
- *Visibility*
  - process steps be documented
  - current status of intermediate products, such as requirements specifications & design specifications be maintained accurately