SOFTWARE QUALITY

CPTS 583

Quality factors and tradeoffs

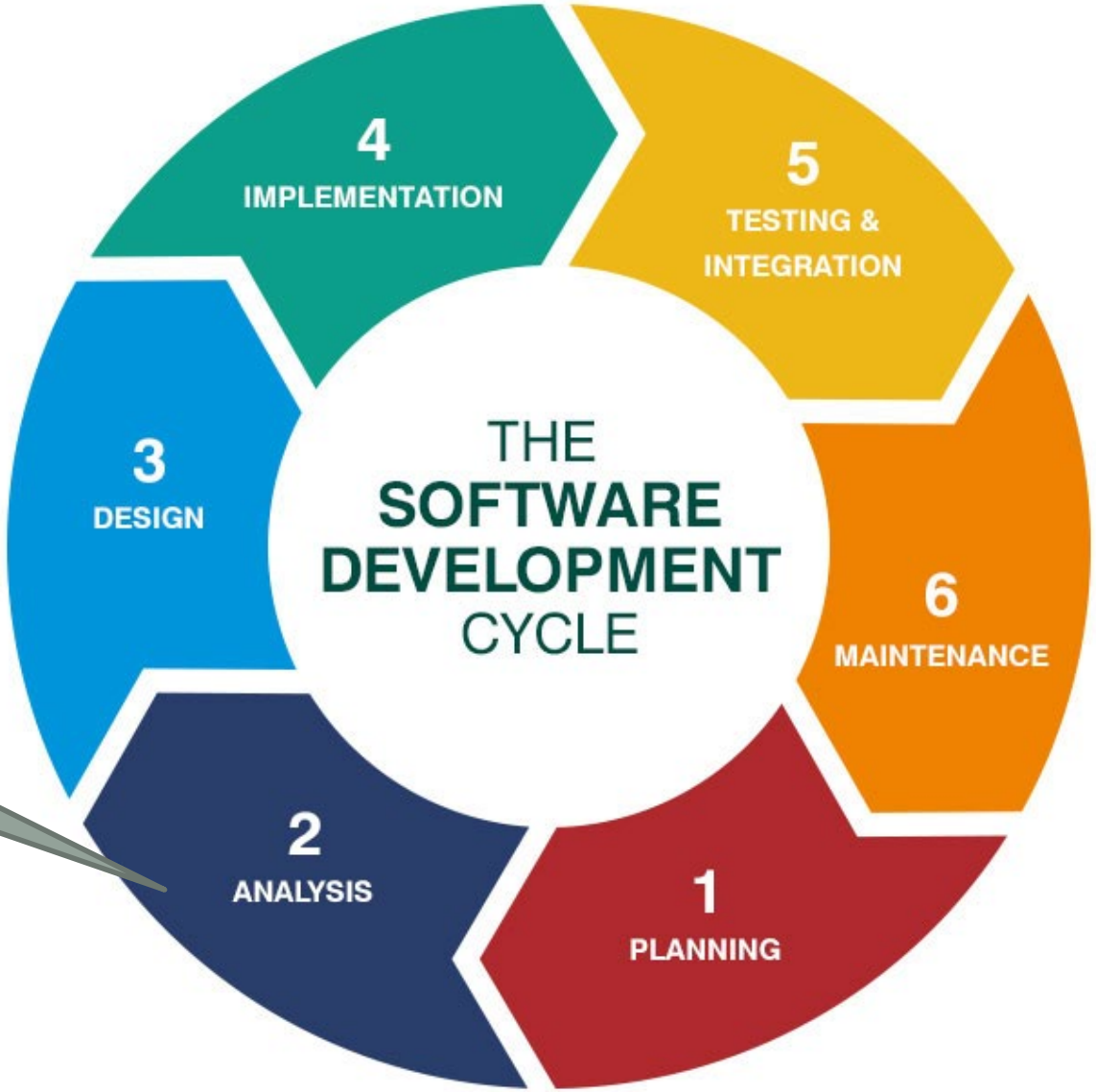
Outline

- Quality factors: what & why
  - Characterizing software quality requirements
- The McCall quality factor model
  - Quality factors
- Tradeoffs among quality factors
  - Conflicts between factors
  - Trading off techniques

Quality factors: background

Just specifying functionalities needed?

Software requirements



Quality factors: what & why

Correctness

- "Our new sales information system seems okay, the invoices are correct, the inventory records are correct, the discounts granted to our clients exactly follow our very complicated discount policy, **BUT** our new sales information system frequently fails, usually at least twice a day, each time for twenty minutes or more. Yesterday it took an hour and half before we could get back to work . . . . Imagine how embarrassing it is for store managers . . . . Softbest, the software house that developed our computerized sales system, claims no responsibility . . . ."

Other quality aspects

Quality factors: what & why

- Need for a comprehensive quality requirements
  - Requirements quality → Software quality
- Quality **factors**
  - Comprehensive list of aspects of the quality

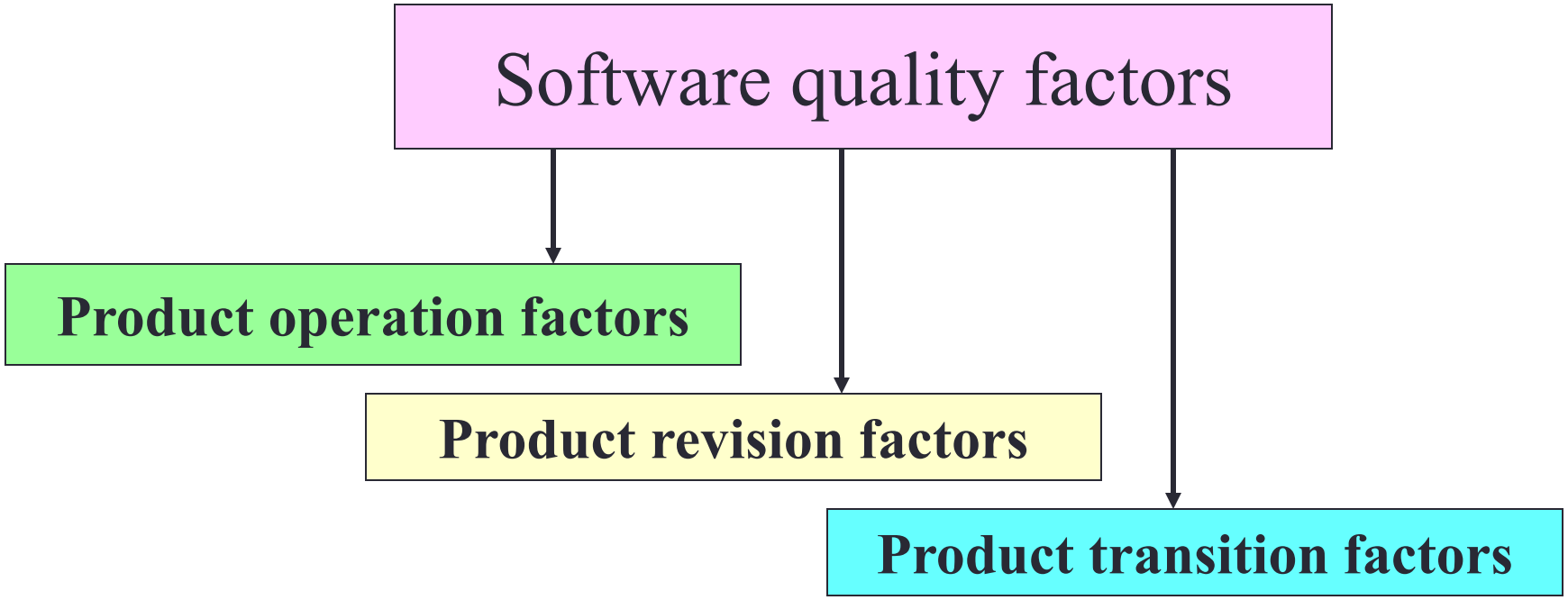
Review: Quality frameworks

- ISO 9126 quality characteristics:
  - Functionality: what is needed?
  - Reliability: function correctly.
  - Usability: effort to use.
  - Efficiency: resource needed.
  - Maintainability: correct/improve/adapt.
  - Portability: one environment to another.

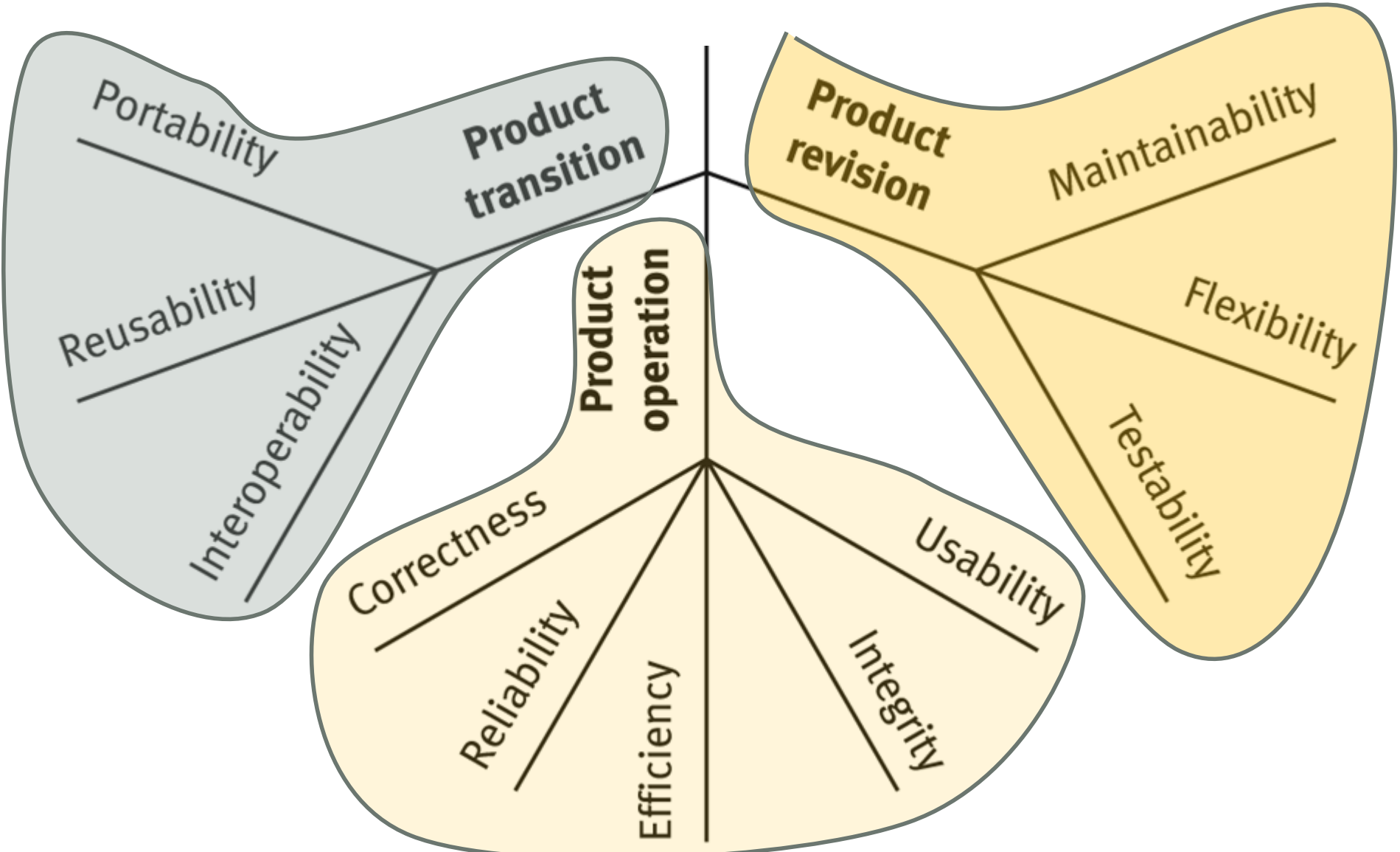
Quality frameworks

- Other quality frameworks/mega-models
  - ▷ McCall: factors, criteria, and metrics
  - ▷ Basili: GQM (goal-question-metric)
  - ▷ SEI/CMM: process focus/levels
  - ▷ Dromey: component reflects Q-attributes
  - ▷ Defect-based view: common in industry
    - cost of defect: by Boehm, NIST, etc.

The McCall quality factor model



The McCall quality factor model



Operation quality factors

- Outputs fulfill specification

-- outputs are accurate

-- outputs are complete

- **Correctness**
- Reliability
- Efficiency
- Integrity
- Usability

In the requirements of an online shopping system:

"the system will show all items put in the carts…"

"the system must compute exact amount of payment…"

Operation quality factors

- Providing service against failure

--- failure rate of the entire system

- Correctness
- **Reliability**
- Efficiency
- Integrity
- Usability

In the requirements of earthquake detection system:
"the system can have a failure rate no more than 0.1%"

Operation quality factors

- System performance
  - In terms of hardware demands

--- processing time

--- storage

--- communication

- Correctness
- Reliability
- **Efficiency**
- Integrity
- Usability

In the requirements of the server side of PayPal.com:

"the system spends for a single deposit transaction no more than 2second CPU time, at most 1KB disk storage, and less than 1Mbps network bandwidth."

Operation quality factors

- Security

--- identity authentication

--- access control

- Correctness
- Reliability
- Efficiency
- **Integrity**
- Usability

In the requirements of the browser side of PayPal.com:

"the system needs a registered user to log in in order to use any functionalities; and only users have an authenticated bank account can make transactions."

Operation quality factors

- Ease of use

--- human/staff resources needed for using the system

- Correctness
- Reliability
- Efficiency
- Integrity
- **Usability**

In the requirements of a hotel booking website:

"the website only takes less than two days for a new staff member to learn how to use reservation functions"

Revision quality factors

- Efforts for failure diagnosis

--- failure cause identification
--- correcting failure
--- verifying failure correction

- **Maintainability**
- Flexibility
- Testability

In the requirements of a graph drawing toolkit

"every method/function in the programs must have error handling constructs (e.g., try … catch ..)"

"all class and method names must be indicative of the purpose of the classes/methods."

Revision quality factors

- Efforts for adaptation

--- to various use scenarios

--- to a variety of customers

- Maintainability
- **Flexibility**
- Testability

In the requirements of a PDF reader

"users can either open a PDF file directly, or drag and drop a file from file explorer to open it"

Revision quality factors

- Efforts to facilitate testing

--- how easy to be tested

- Maintainability
- Flexibility
- **Testability**

In the requirements of an auto control system:

"the system should have standard test data to compare for the self-diagnostic module to do routine check at car startup"

Transition quality factor

- How portable the system is

--- various hardware platforms
--- various operating systems

- **Portability**
- Reusability
- Interoperability

In the requirements of Skype:
"it should run in Windows as well as in Linux"
"it should work well on any MacBook Pro laptops."

Transition quality factor

- Reusable for future projects

--- to save development costs

--- to have better quality

- Portability
- **Reusability**
- Interoperability

In the requirements of a shooting game:

"the graphics engine will be reusable for building another shooting game"

"the joystick controller may be used by other games that use joysticks for user inputs"

Transition quality factor

- Interact with other software

--- collaborating systems

--- firmware

- Portability
- Reusability
- **Interoperability**

In the requirements of an online food ordering service:

"the system should support migrating transactions records to a SATA hard disk array with firmware 3.0+"

"the system will work with MySQL and Oracle databases"

McCall quality factors

Quality Categories	Quality Factors	Broad Objectives
Product operation	Correctness	Does it do what the customer wants?
	Reliability	Does it do it accurately all of the time?
	Efficiency	Does it quickly solve the intended problem?
	Integrity	Is it secure?
	Usability	Can I run it?
Product revision	Maintainability	Can it be fixed?
	Testability	Can it be tested?
	Flexibility	Can it be changed?
Product transition	Portability	Can it be used on another machine?
	Reusability	Can parts of it be reused?
	Interoperability	Can it interface with another system?

Quality factors in requirements

- Examples

Correctness

- Employees salaries should not be late
- Employees salaries should be calculated accurately and must be ready five days before the end of the month

Reliability

- The system should be working as much time as possible
- The system should not be in failure status during working hours (9 to 4). Total time of failure status should not exceed 20 minutes per month.

Quality factors in requirements

- Examples

Efficiency

- The GPS application should use as little as possible of mobile phone battery
- The GPS application should not use more than 10% of battery power in two hours time

Integrity

- Students should be allowed to access their final marks
- Students should be allowed to view their final marks. They should not be able to make any changes

Quality factors in requirements

- Examples

Usability

- The billing system should be easy to use
- Billing staff should be able to learn the most important five functions of the billing system in 3 working hours.

Trading off between factors

- 11 factors
- All to be considered

Efficiency

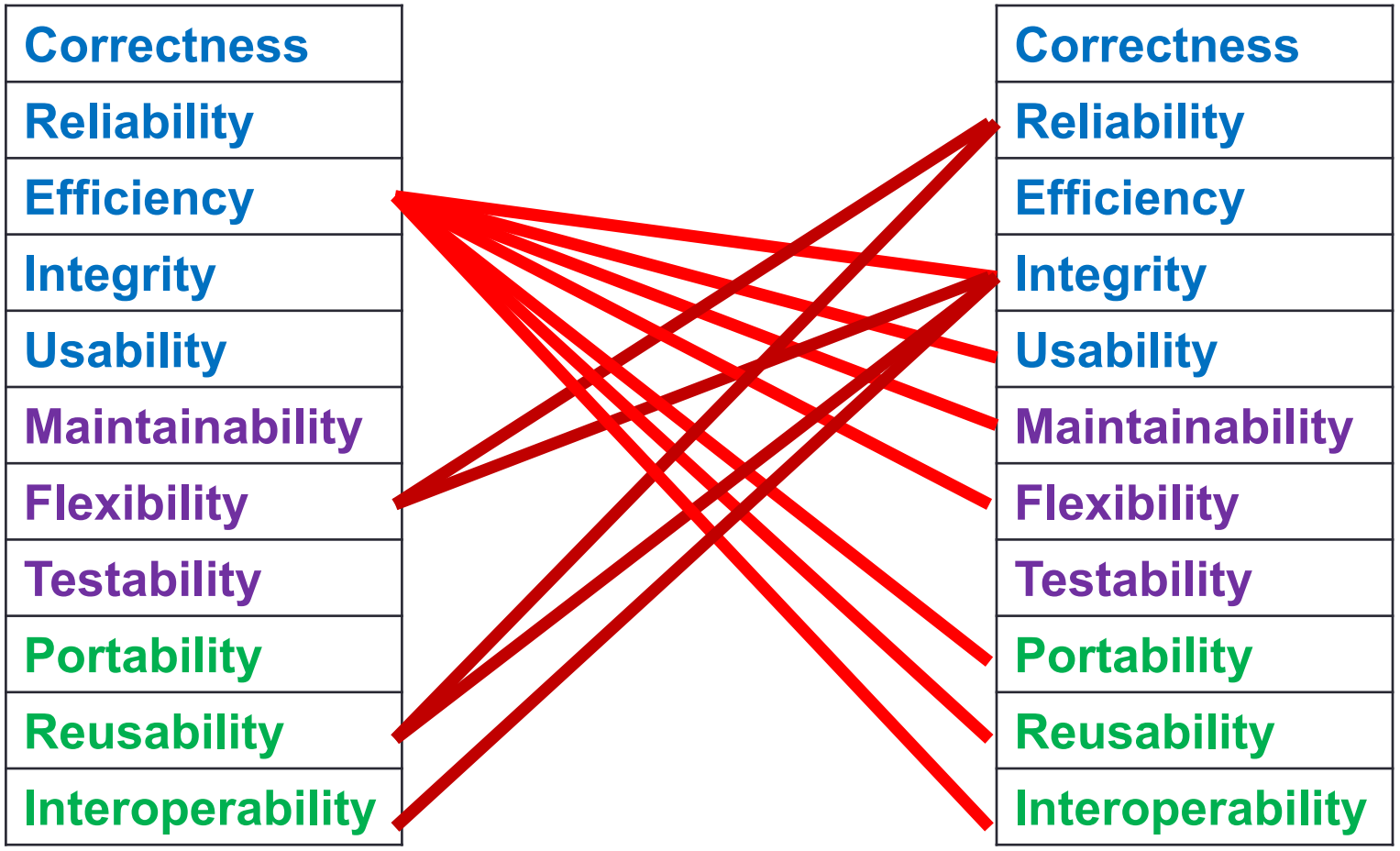
Usability

Efficiency

Usability

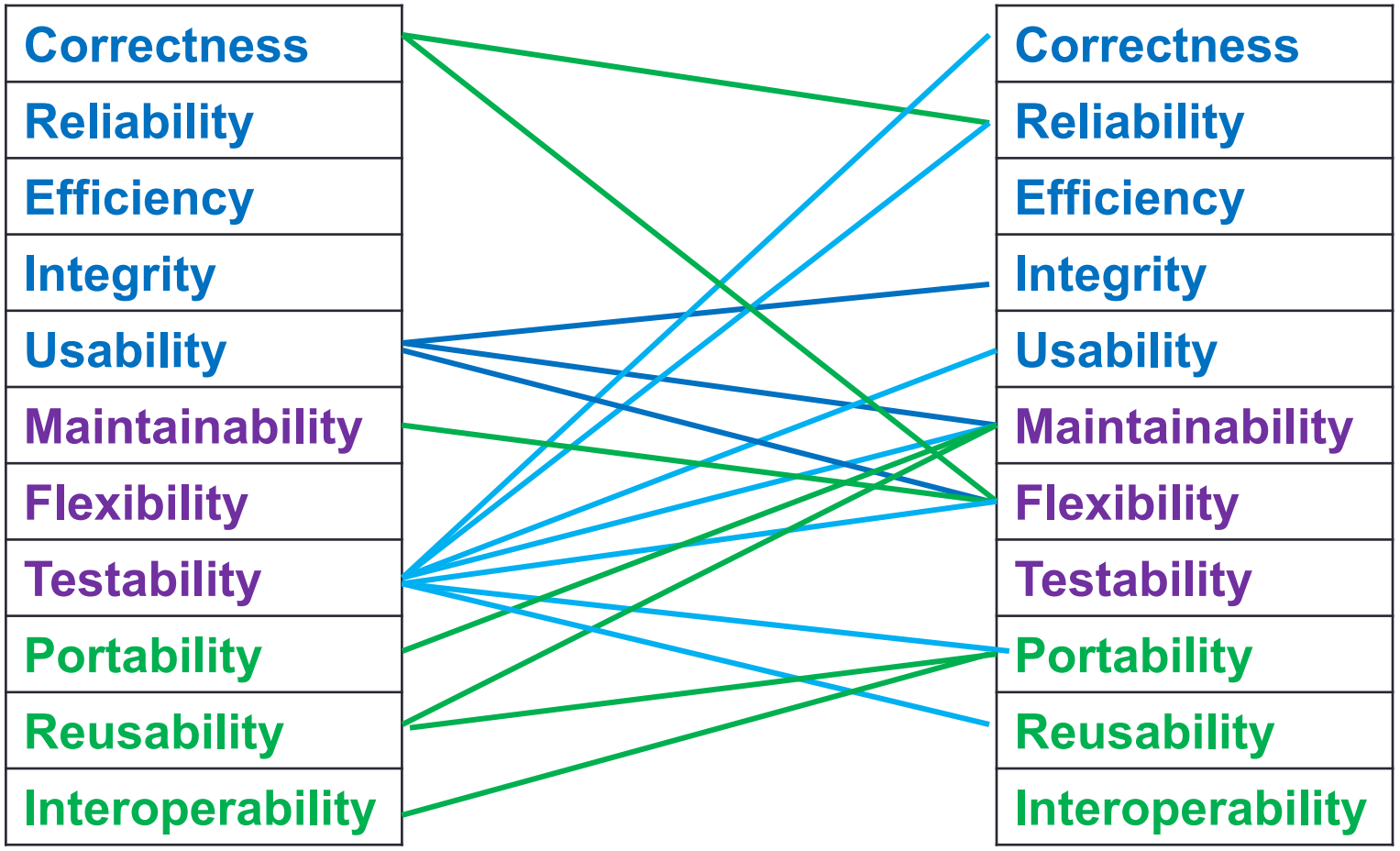
Trading off between factors

- Factors with  (**negative** mutual) influence



Trading off between factors

- Factors with  (**positive** mutual) influence



Trading off between factors

- Tradeoffs between conflicting factors

In practice:

--- prioritization

--- compromise

State of the art:

--- modeling and tracking

--- creating facts & knowledge

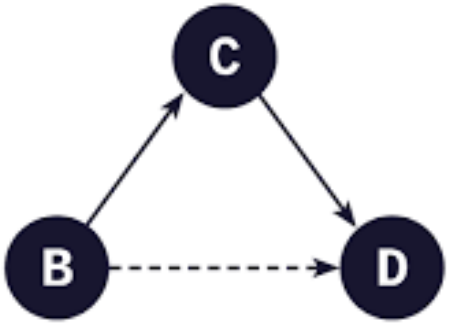
Trading off between factors

- Prioritize & compromise

Transitive influence

--- prioritizing one factor benefits other factors

--- compromising one factor implies compromising other factors



Trading off between factors

Examples:

- Flight software that flies on a single mission satellite will not be concerned with reusability but may be very concerned with reliability.

- A child game software may be more concerned with usability through rich GUIs, than with efficiency.

Trading off between factors

- Case study

Software: to build a networked system, containing components for managing network connections.

Problem: three factors to trade off, portability, performance, and maintainability.

Trading off:

--- create prototypes of the system

--- measure performance

--- run static-analysis tool to quantify maintainability

--- run on both Windows 7 and Ubuntu 14.05

--- decide to trade performance (efficiency) for portability and maintainability, because the worse performance is acceptable in practice

Summary

- Quality factors: what & why
- McCall quality factor model
  - Operation quality
  - Revision quality
  - Transition quality
- Trade offs
  - Influence relationships
  - Dealing with conflicting quality factors