

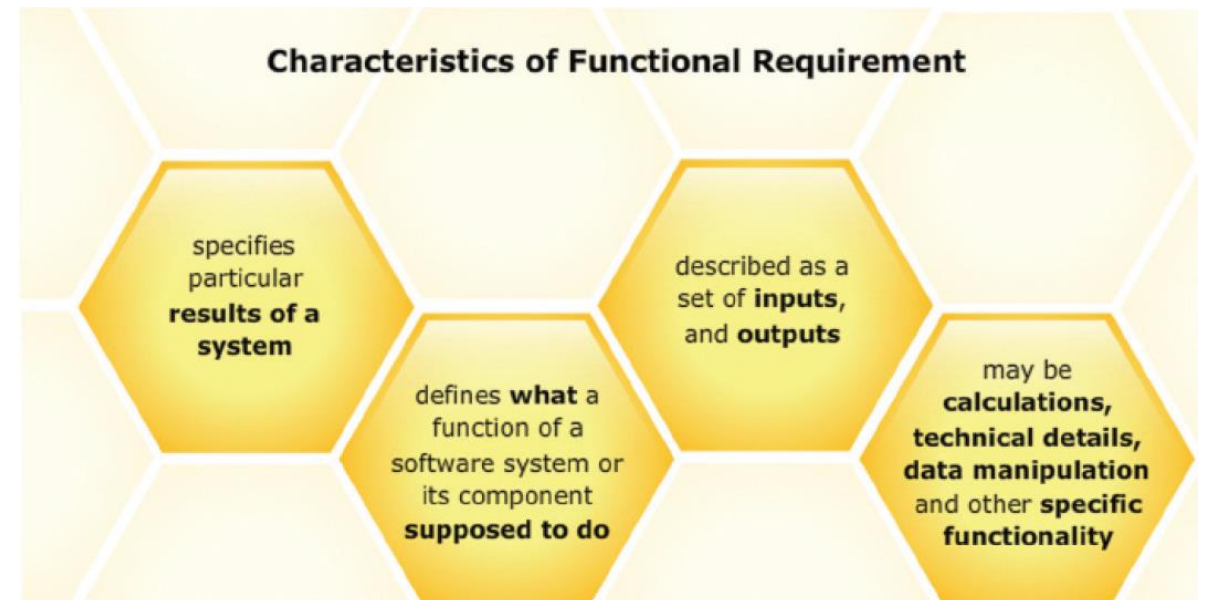
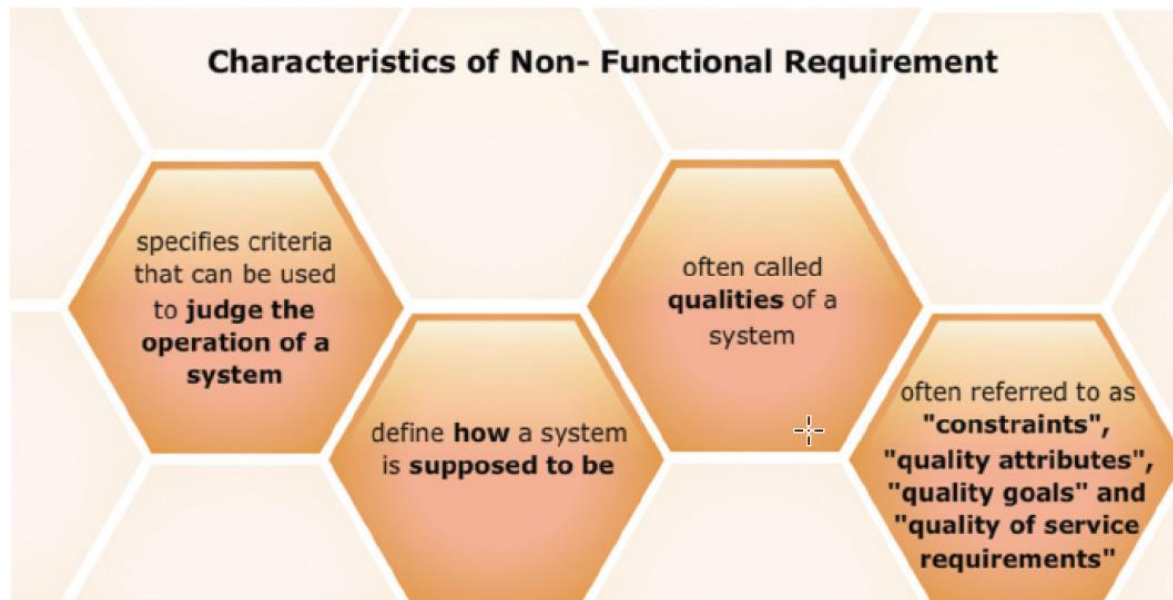
Functional and Non-Functional Requirements

Unit 6

Requirement Engineering

6.0 Introduction

What are Functional and Non-Functional Requirements?



6.1 Functional requirements

- Functional requirements are also called functional specifications or functional requirement specifications.
- It defines the capabilities and functions that a system must be able to perform.
- Functional requirements should include four descriptions:
 1. **Descriptions of data to be entered into the system.** For example: Employee ID, address.
 2. **Descriptions of operations performed by each screen.** For example: Main menu, registration form.
 3. **Descriptions of work-flows performed by the system.** For example: Users can only log-in into the system as a member after he/she has registered.
 4. **Descriptions of system reports or other outputs.** For example: Maintenance reports, data entry reports.

6.2 Functional Requirements: Definition and Concepts

- Functional requirements are documented requirements about the functions of an intended system.
- Readers should understand the system from the document, but no particular technical knowledge should be required to understand it.
- Requirements outlined in the functional requirements will be tested in the
 - a) Operational Qualifications,
 - b) Installation Qualifications, and
 - c) Performance Qualifications.

-
- Functional requirements should include four requirements:



6.2.1 Operational Qualifications

- Operational Qualifications are collections of test cases to check the operational flow of a system. They are usually performed before the system is released.
- Tests that could be performed in Operational Qualifications include:
 - 1) Interface Requirements
 - 2) Business Requirements
 - 3) Regulatory Requirements
 - 4) Security Requirements

6.2.1.1 Interface Requirements

- Field which accepts numeric data entry, dates, or texts must be clearly described. The functionality of every command buttons, text boxes, option buttons, and check boxes in every menus, forms, and reports must also be clearly described.

6.2.1.2 Business Requirements

- Data must be entered before a request can be approved. Only then, the request will be moved to the approval workflow. For example: Every field of personal details form of a system must be filled in by the user before it can be recorded by the system.

6.2.1.3 Regulatory Requirements

- The database will have a functional audit trail, the system will limit access to authorised users, and the spreadsheet can secure data with electronic signatures.

6.2.1.4 Security Requirements

- Member of the data entry group can enter requests but not approve or delete requests, members of the managers group can enter or approve a request, but not delete requests, and members of the administrators group cannot enter or approve requests, but can delete requests.

6.2.2 Installation Qualifications

- Installation Qualifications are collections of test cases used to verify the proper installation of a system.
- Installation Qualifications must be performed before completing Operational Qualification or Performance Qualification.
- Installation Qualifications tests include:
 - Tests that will check whether the operating system has the appropriate hardware.
 - Tests that will check whether all files required to run the system are present.
 - Tests that will check whether all documentation required to train system personnel has been approved.

6.2.3 Performance Qualifications

- Performance Qualifications are a collection of test cases used to verify that a system performs as expected under real-world conditions.
- These tests are sometimes conducted as the system being released.
- A performance qualification might check:
 - Whether a system can handle multiple users without significant system lag.
 - Whether when the system contains large quantities of data, queries are returned in a certain period of time.
 - Whether that independent work-flows do not affect each other when used concurrently.

6.3 Design Specifications

- Design Specifications are also called Software Design Specifications. It describes how a system performs the requirements outlined in the Functional Requirements.
- Design Specifications should include five requirements:
 - a) The specific inputs, including data type, to be entered into the system
 - b) Calculations or codes used to accomplish defined requirements
 - c) Outputs that was generated from the system
 - d) Explanations of technical measures to ensure system security
 - e) Identification of how the system meets the regulatory requirements

6.4 User Requirements Specification

- User Requirements Specification describes what users require from the system.
- User requirement specifications are written before the system is created.
- User Requirements Specification should include elements as below:
 - Screen that can accept production information.
 - System that can produce a summary report.
 - Many users can use the system concurrently without delays.
 - Screen that can print onscreen data to the printer.

6.5 Non-Functional Requirements

- Non-functional requirements are the requirements which are not specifically related to the functionality of the system being developed.
- They place restrictions and external constraints that the system must meet.
- They include safety, security, usability, reliability and performance requirements.
- Classifications of Non-Functional Requirement:
 - Product Requirements
 - Process Requirements
 - External Requirements

6.6 Non-Functional Requirements: Definition and Concepts

- Non-functional requirements define the overall attributes of the intended system.
- They place guidelines on how the user requirements are to be met.
- Certain non-functional requirements may constrain the development process rather than the product.
- They include requirements on standards, methods and implementation languages to be followed.
- Customers generally give these constraints to achieve good product quality and to maintain compatibility with methods used for system design and implementation.

6.7 Examples of Non-Functional Requirements

1. System Constraints
2. User Objectives
3. Competitive Analysis of Features
4. Development Organisation Constraints
5. Development Organisation Objectives
6. Competitors And Industry Trends

6.7.1 System Constraints

- The developer will be looking for different elements of the environment into which the system may serve as constraints on the system.
- This can be done through the installation of infrastructure such as hardware and operating system platforms, or legal applications.

6.7.2 User Objectives

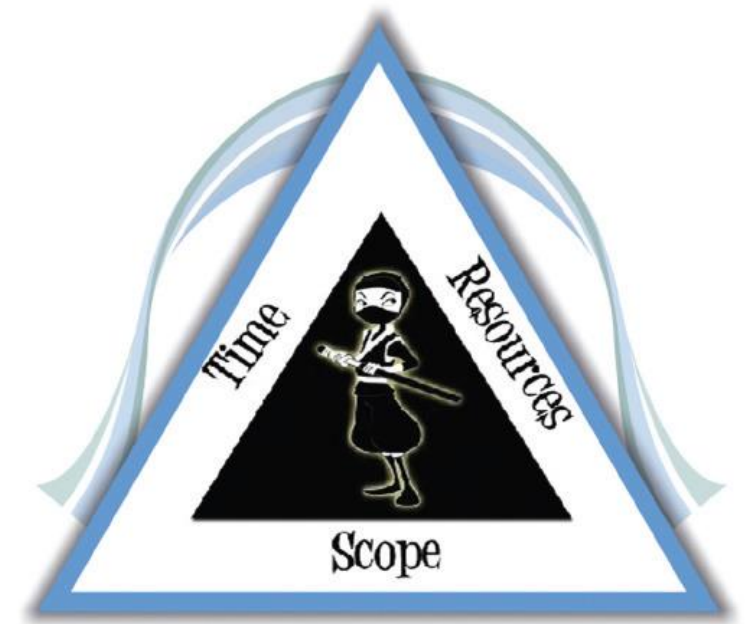
- In establishing the run-time qualities for a system, it is important to identify all the categories of users that will interact with the system, and understand what quality attributes they care about.
- A quality attribute such as performance may surface for one user as a concern, and another as a value, so it is useful to direct elicitation of both values and concerns for any user.

6.7.3 Competitive Analysis of Features

- Run-time qualities are often associated with the product features.
- Features are generally thought of as the characteristics of the product that establish its competitiveness, frequently distinguishing the product functions with at least one quality attribute.
- For example, many web-based catalogue services have on-line payment options. To allay market concern, the electronic payment feature should include transaction security as an essential attribute

6.7.4 Development Organisation Constraints

- In product development, constraints placed by upper levels of management typically take the form of required time-to-market of the application or release or fixed development resources.



6.7.5 Development Organisation Objectives

- Objectives, values and concerns may relate to the business performance, schedules, productivity and effectiveness and work-life balance of the Stakeholders in the development organisation.
- For example, strategic management establishes the product portfolio plan, including planned releases (which products in what timeframe).

6.7.6 Competitors And Industry Trends

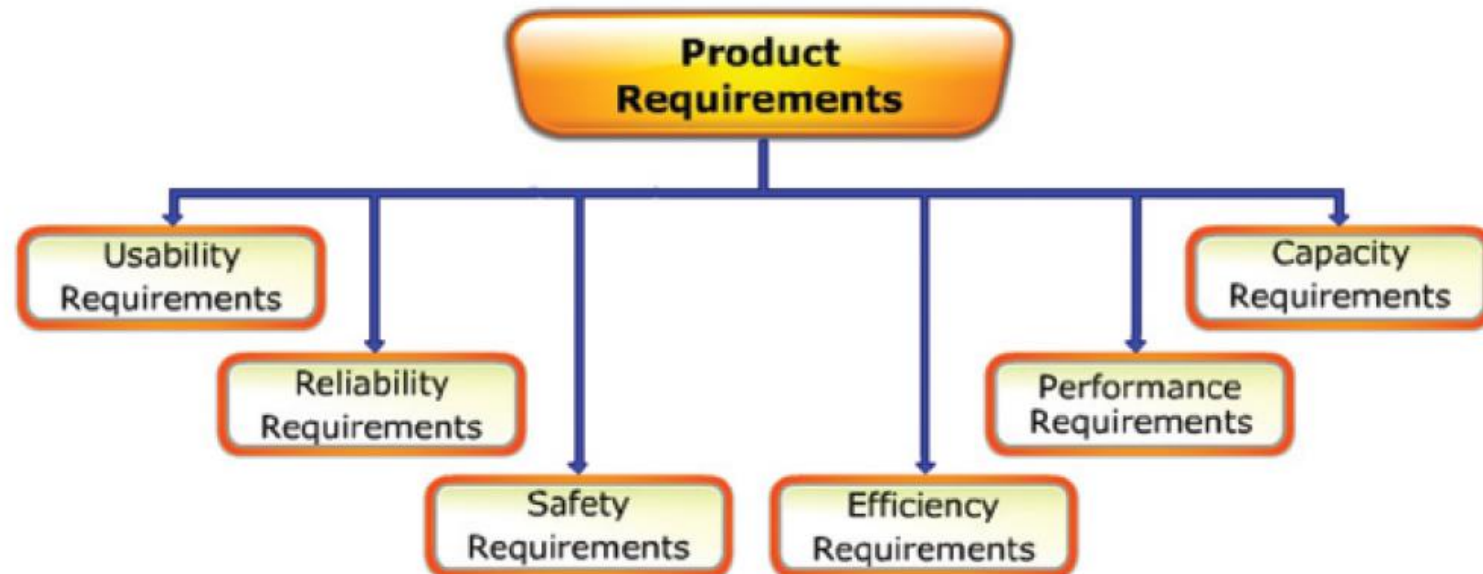
- Benchmarks of competitors' processes and industry trends may drive the organisation to adopt more aggressive productivity objectives which may in turn translate into development-time qualities such as evolvability and reuse.
- Once you have worked with stakeholders to gather their requirements, these need to be documented in such a way that the architects, designers and implementers can all understand them and create a system that fulfils all the requirements.

6.8 Classifications of Non-Functional Requirement

- Product Requirements
- Process Requirements
- External Requirements

6.8.1 Product Requirements

- Product requirements specify the desired characteristics that a system or subsystem must have.
- Some of them are easily quantified, while some are not.
- Product requirements consists of six requirements:





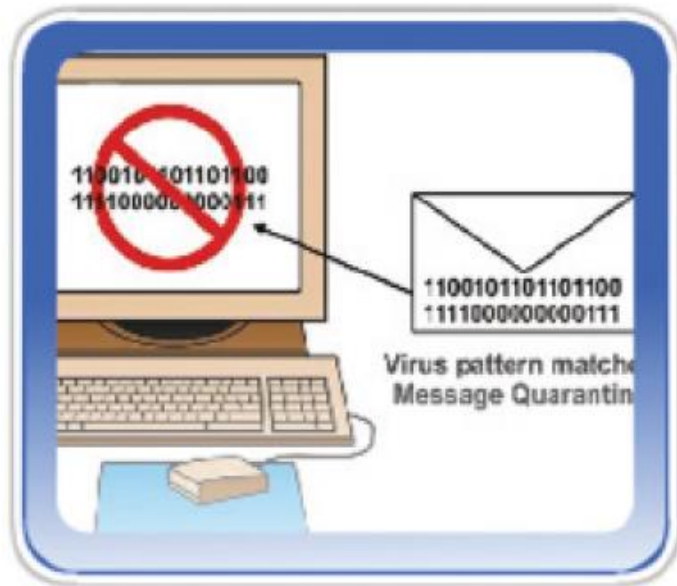
Usability requirements

The requirements on how useful and how the system could be used. It describes if the looks of a system meet what the system's supposed to give.

Reliability requirements

The requirements on how much the system could be trusted. The system should at least be able to block human intruders from getting inside.





Safety requirements

The requirements on how safe the system should be. The system should not be corrupting other systems within the organisation.

Efficiency requirements

The requirements on how efficient the system could be. The system must be able to increase the productivity of its users and the way that jobs can be settled easier.





Performance requirements

The requirements on how the system should be performing. It must be able to do jobs faster.

Capacity requirements

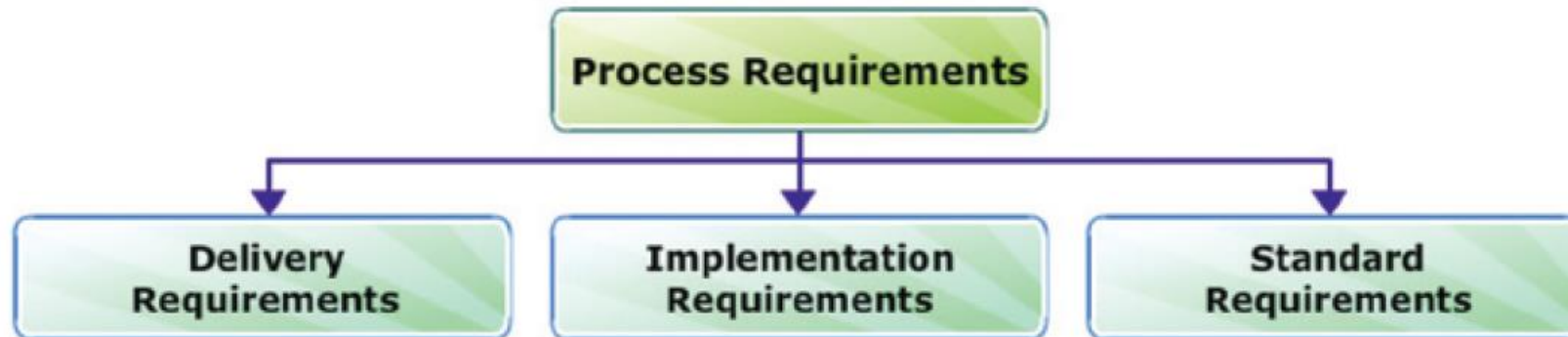
The requirements on how big the system should be. The system must be able to save data of certain amounts.



6.8.2 Process Requirements

- Process requirements are constraints placed upon the development process.
- They may be included because the customer wants to have some power over the development process.
- They include requirements on development standards, methods, and management reports.

- Process requirements consists of three requirements:





Delivery requirements

The requirements that must be taken care of while delivering the system. Certain time period must be fulfilled for the delivery.



Implementation requirements

The requirements that must be taken care of while implementing the system. Certain guarantees must be included. For example, if the system is not working as required, it can be returned.

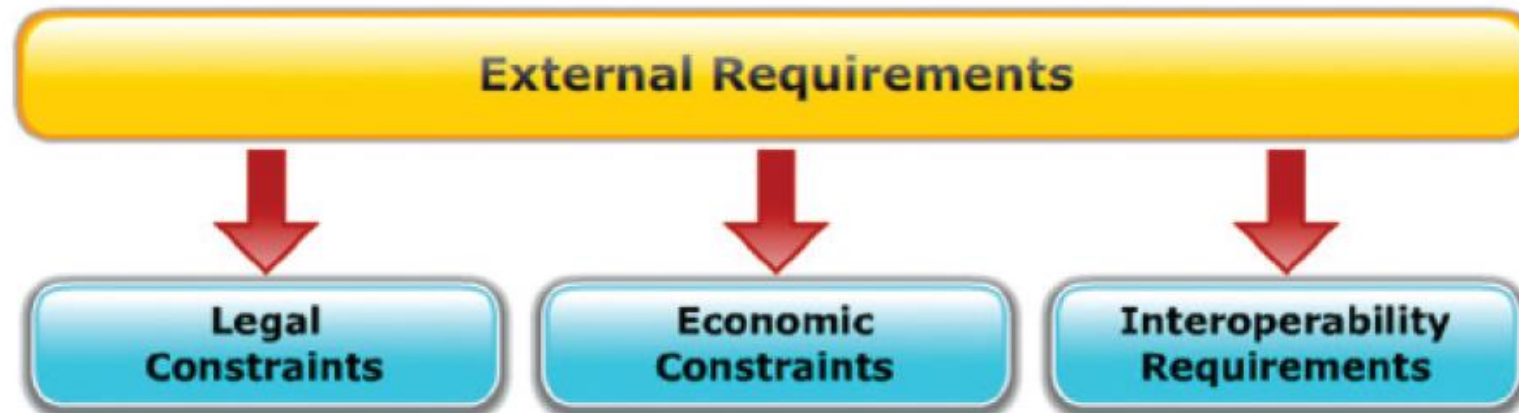


Standards requirements

Standards that have to be taken care of while developing the system. International and domestic standards for system developments are various and it must be fulfilled before the system can be activated.

6.8.3 External Requirements

- External requirements are requirements which may be placed on both the product and the process.
- They may be based on application domain, organizational considerations, or data protection regulations.
- External requirements consists of three requirements:



1. Legal constraints

The legal constraints to be taken care of while the system is developed. For example, the system must be patented before it can be sold if no copyright issues are wanted.

2. Economic constraints

The economic constraints that have to be taken care of while the system is developed. For example, lack of money for budgeting of the development.

3. Interoperability requirements

The inside operation requirements that must be taken care of. For example, lack of manpower to develop the system.

6.9 Approaches to NFRs

1. Product vs. Process?

1. Product-oriented Approaches

1. Focus on system (or software) quality
2. Aim is to have a way of measuring the product once it's built

2. Process-oriented Approaches

1. Focus on how NFRs can be used in the design process
2. Aim is to have a way of making appropriate design decisions

2. Quantitative vs. Qualitative?

1. Quantitative Approaches

1. Find measurable scales for the quality attributes
2. Calculate degree to which a design meets the quality targets

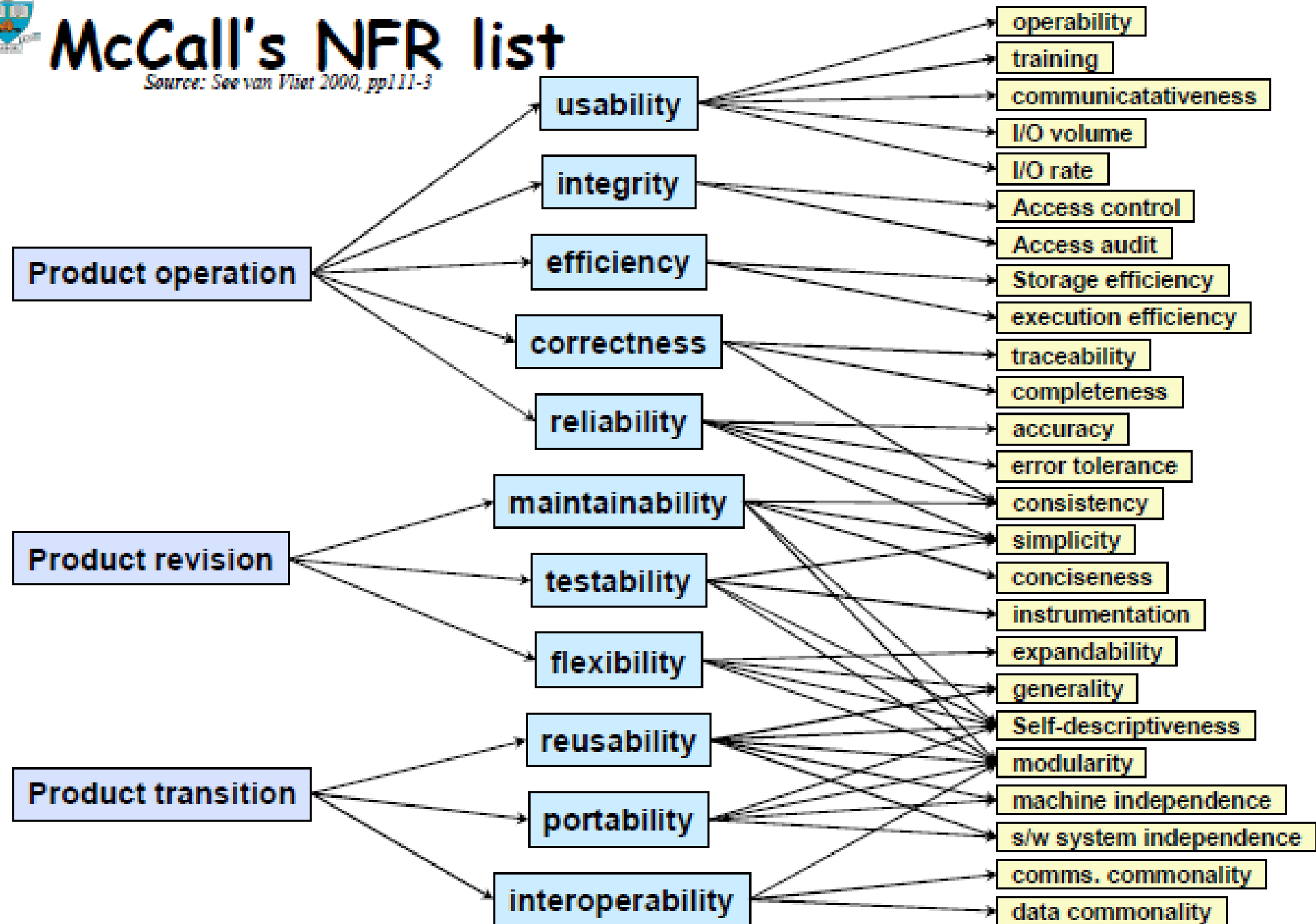
2. Qualitative Approaches

1. Study various relationships between quality goals
2. Reason about trade-offs etc.



McCall's NFR list

Source: See van Vliet 2000, pp111-3

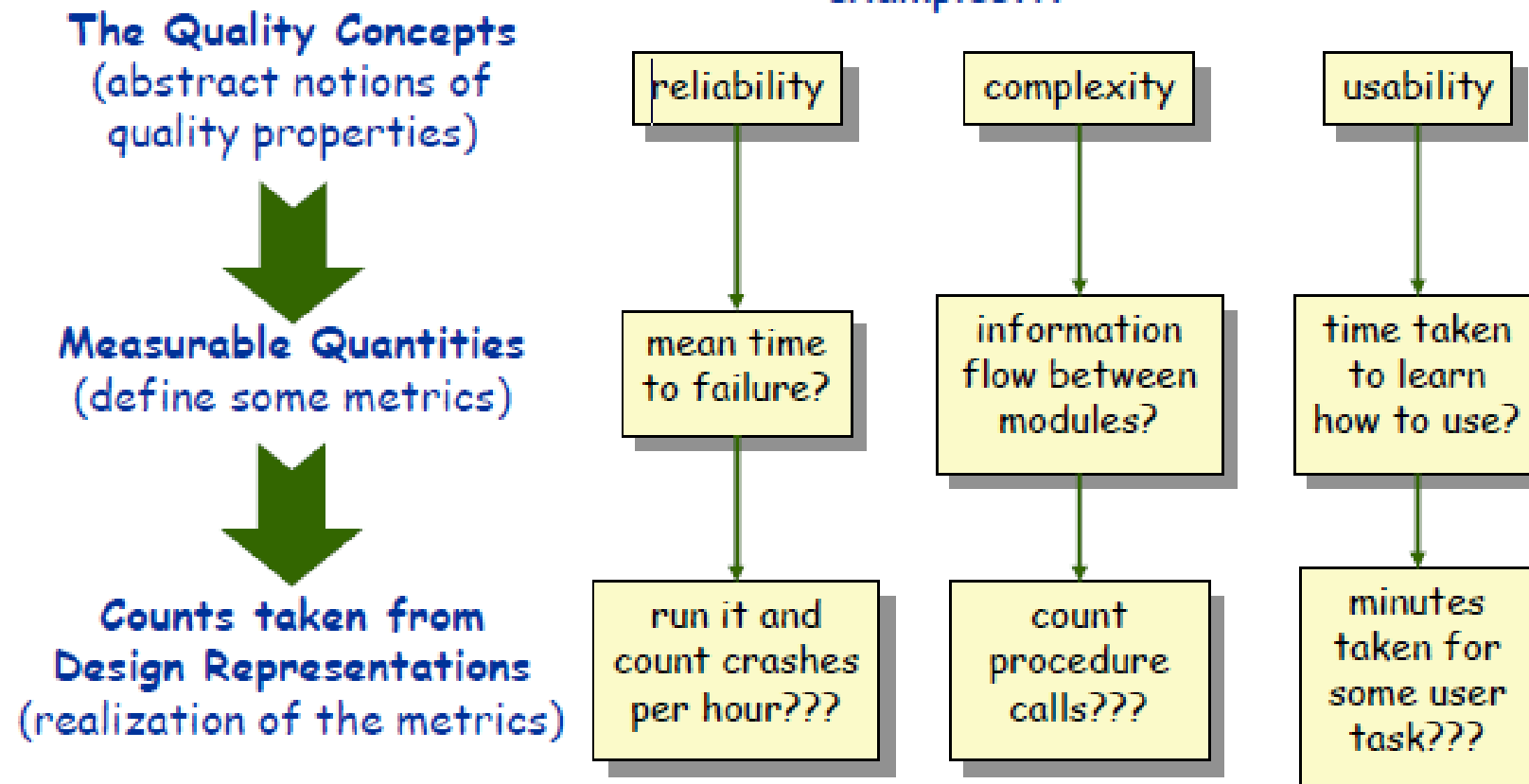


7.0 Making Requirements Measurable

Source: Shneiderman, 1997, p. 100-101

- We have to turn our vague ideas about quality into measurables

examples...



7.1 Example Metrics

Quality	Metric
Speed	transactions/sec response time screen refresh time
Size	Kbytes number of RAM chips
Ease of Use	training time number of help frames
Reliability	mean-time-to-failure, probability of unavailability rate of failure, availability
Robustness	time to restart after failure percentage of events causing failure
Portability	percentage of target-dependent statements number of target systems

Unit 6

The END