Software Requirements

OOAD

What is Requirement?

- ***Requirements*** - descriptions of the services provided by the system and its operational constraints.

- Ranges from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.

Types of Requirements

- **User requirements**
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints.
  - Written for customers
- **System requirements**
  - A structured document setting out detailed descriptions of the system services.
  - Written as a contract between client and contractor
- **Software specification**
  - A detailed software description which can serve as a basis for a design or implementation.
  - Written for developers

Functional Requirements And Non-functional Requirements

Based on the specification of the requirement, the requirement are classified into two categories:

- Functional Requirements
- Non-functional Requirements

Functional Requirement

- Any requirement which specifies **what** the system should do.
  - **Functional user requirements** - **high-level** statements of what the system should do.
  - **Functional system requirements** describes the system services **in detail**.
- **Not concerned with how** the software does things, i.e., they must be free of design considerations

Functional Requirement Examples

- Functional requirements for a university library system (LIBSYS), used by students and faculty to order books and documents from other libraries:
    - The user shall be able to search in the set of databases (input: id, title, authors).
    - The system shall provide appropriate viewers for the user (staff, students) to read documents in databases.
    - The system shall provide the university hot information/events to all users.

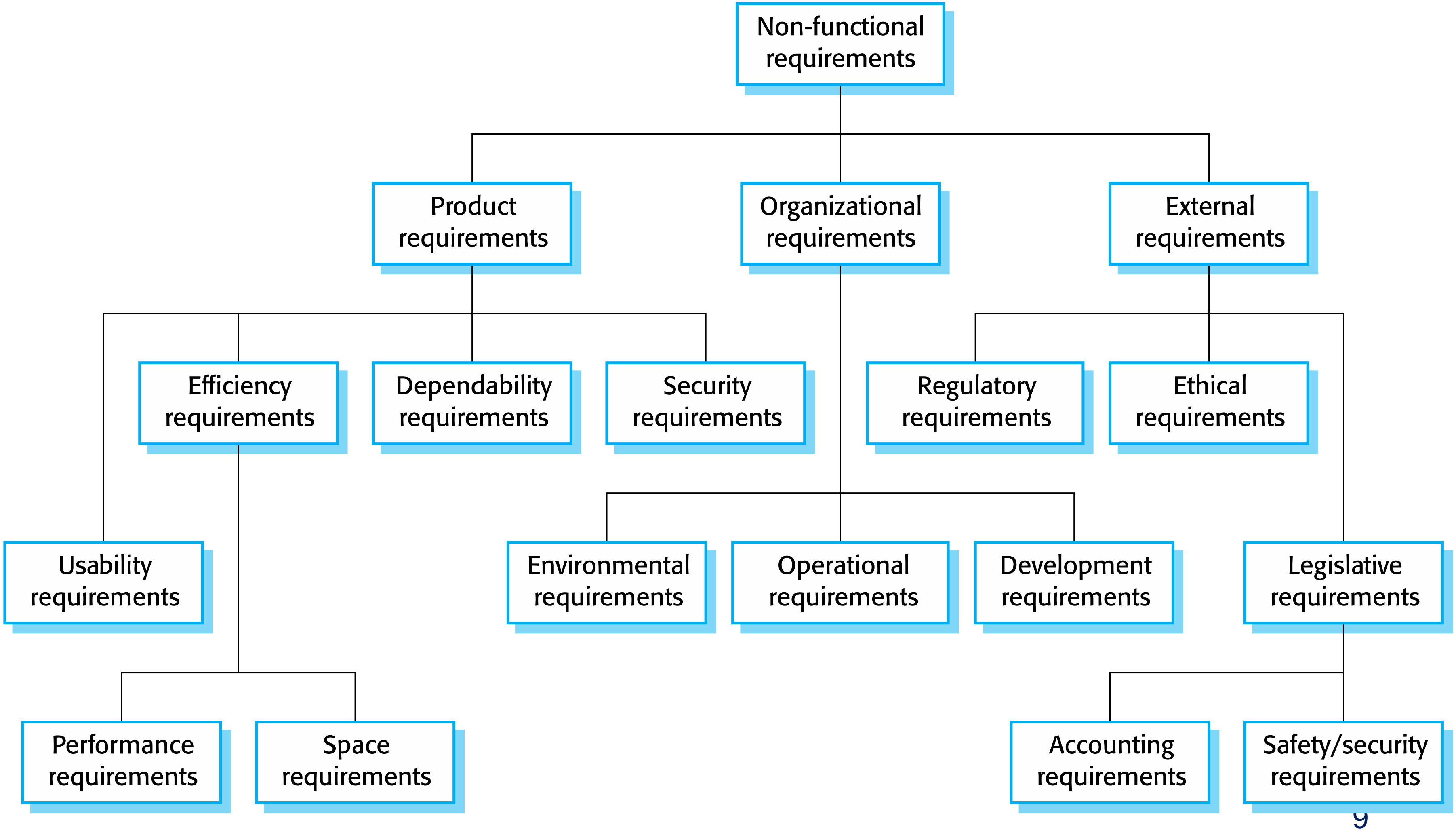
Non-functional Requirements

- A requirement specifies **how** the system performs a certain function.
- Not directly concerned with the specific functions.
- They may relate to emergent system properties such as reliability, response time and store occupancy.

Non-functional classifications

- **Product requirements**
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- **Organizational requirements**
  - Requirements which are a consequence of organizational policies and procedures e.g. process standards used, implementation requirements, etc.
- **External requirements**
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Types of Non-functional Requirements



Non-functional Requirement Examples

- **Product Requirement**
  - The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.
- **Organizational Requirement**
  - The system shall be delivered before this new year.
- **External Requirement**
  - The system shall not disclose any personal information about system users apart from their name and library reference number to the library staff who use the system.

Non-functional Requirement Examples

- **Operational Requirements**
  - The system will operate in Windows environment.
  - The system should be able to connect to printers wirelessly.
  - The system should automatically back up at the end of each day.
- **Performance Requirements**
  - The system will store a new appointment in 2 seconds or less.
  - The system will retrieve the daily appointment schedule in 2 seconds or less.
- **Security Requirements**
  - Only doctors can set their availability.
  - Only a manager can produce a schedule.

Requirements Interaction

- Conflicts between different Non-functional requirements are common in complex systems
  - Example: Spacecraft system
    - To minimize weight, the number of separate chips in the system should be minimized
    - To minimize power consumption, lower power chips should be used
    - **However**, using low power chips may mean that more chips have to be used. Which is the most critical requirement?

User Requirements

- User requirements will often be high-level, unclear and incomplete. They are more like high-level goals, or business goals, rather than software requirements needed by developers.

- Is described to be understandable by users without detailed technical knowledge.

- Can be written or defined using natural language, tables and diagrams.

User Requirements Examples

- **ScreenA** accepts production information, including Lot, Product Number, and Date.
- **SystemB** produces the Lab Summary Report.
- Twenty users can use **SystemC** concurrently without noticeable system delays.
- **ScreenD** can print on-screen data to the printer.

System Requirement

- A detailed description of **what** the system should do.
- An expanded version of the **User requirement** with **more technical detail.**
  - Serve as a basis for designing the system
  - May be used as part of the system contract
- **User requirements** talk about the problem domain, the world of the user. They describe what effects need to be achieved.
- **System requirements** talk about the solution domain, the world of the software logic. They describe what the software must do (which sometimes can oppose to the effects in the user's world that this may or may not achieve).
- **User requirements** can be like wishes. An easy way to tell the difference between a **User requirements** and a **System requirements** is that there is enough detail in the **System requirements** to make it testable.

System Requirements Examples

- **User requirement**
  - We need to be able to spell check documents.

- **System requirement**
  - The system needs to be able to spell check documents and provide autocorrect facilities. There will be support for the following languages, English, French and German will plug in support for other languages.

References

1. **Dennis A., Haley W. B., Tegarden D.** "System Analysis & Design - An Object-Oriented Approach with UML", 5e – 2015
2. **Sarnath R., Brahma D.** "Object-Oriented Analysis, Design and Implementation", 2e – 2015