Software Requirement (Sommerville) Chapter 6

Software requirements provide services of a system and its operational constraints

Analysing, documenting and checking these system services and operational constraints is requirement engineering.

Q What's the difference between user requirements and systems requirements? (as per this author)

User requirements and system requirements both state (writing + diagrams) about services of the system and operational constraints.

But user requirements are at a higher-level of abstraction compared to system requirements.

Thus, system requirements must ~~be~~ have precise ~~&~~ details about system services and constraints etc

User reqs. are read by client mangers, system end-users etc

System reqs are read by system architects, software developers etc

- Requirements need to be written in different level of abstraction (detail) depending on who is reading them (manager, customer, developers etc)

Q What is the difference between functional and non-functional requirements?

functional requirements are ~~&~~ provide insight in to services provided by the system where as non-functional requirements are constraints on services/functions

Constraints include timing constraints, development process and standards. (emergent perporties like performance, reliability, saftey, security and usability)

~~User~~ Requirement

Non-functional requires often apply to system as a whole rather than specific service/function.

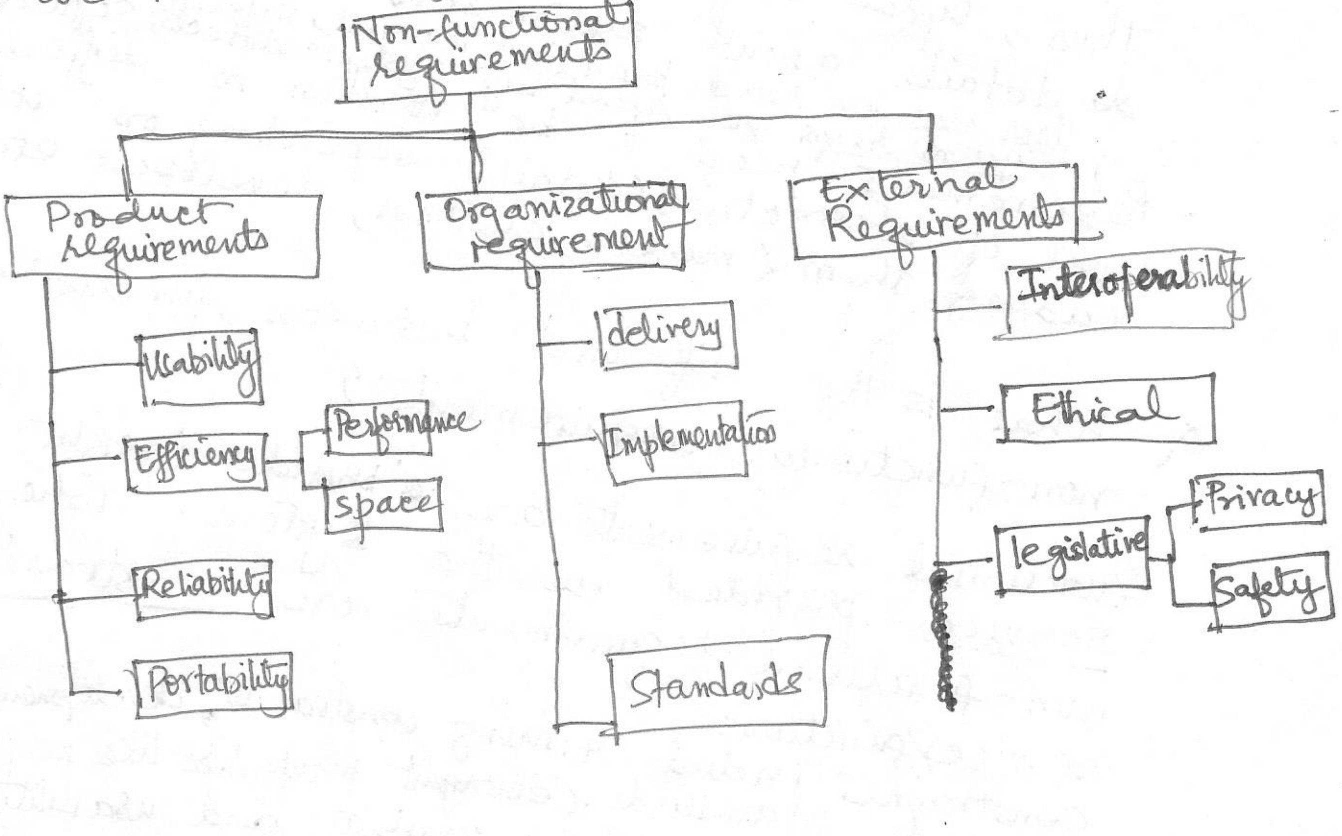
- Domain requirements: Requirements due to a specific domain that reflect characteristic and constraints of a specific domain (functional and non-functional)

  eg: Access rights to students, professors, directors, deans etc to data in the field of education (domain) in a university student-faculty registration system.

What are Characteristics of good requirements?

- unambiguous: All requirements should be clearly written (should not have more than one meaning)
- Complete: All services should be defined
- Consistent: Requirements should not contradict one another

What are the categories of Non-function requirements?



What are some metrics used to measure non-functional requirements?

Req./Property	Measure
Speed	Processes transactions/sec screen refresh time
Size	K bytes Number of RAMchips
Ease of Use	Training time Number of help frames
Reliability	Mean time of failure Probability of unavailability Rate of failure occurance
Robustness	Time to restart after failure % of events causing failure Probability of data corruptios on failure
Portability	% of target dependent statements Number of target systems

What are potential problems with user requirements?

- Lack of clarity
- Requirement confusion: functional, non-functional reqs. , system goals and design may be mixed
- Requirements amalgamation: Several requirements may be stated into one requirement.

How does one minimize misunderstanding about requirements?

- Use a standard format to write requirements that describes it. It should also provide rationale for a requirement
- A reference to more detail if required
- Use consistent language to distinguish between mandatory and desirable Requirements
- Use text highlighting.
- ~~#~~ No to use computer jargon to keep language simple as possible.

How to specify System requirements ?

System requires are user requirements in great detail.
Since it is difficult to write ~~spec~~ detailed reqs.
that would involve domain language in natural language
it leaves scope for multiple interpretation

System reqs. may be expressed in

- Structured natural language
- Design description language
- Graphical notations
- Mathematical specifications

What are the three types of interface specifications ?

A new system must sometimes operate in an existing system that is already present in the environment.

The existing systems interfaces must be clearly defined and included in documentation (may be as Appendix)

The three types of interfaces are

- Procedural interfaces : services offered by existing system
- Data structures : these are passed from one system to anothr
- Representations of data (such as ordering of bits)

eg:
```
interface PrintServer {
    void print (Printer p, PrintDoC d) ;
    void displayPrintQueue (Printer p);
    void cancelPrintJob (Printer p, PrintDoc d);
    void switchPrinter (Printer p1, Printer p2, PrintDoc d)
}
```

The Software Requirement Document.

The Software Req. Document also called SRS is the official document the specifies what developess need to implement.

Audience for requirements document.

system customers → Check to see if it meets their needs

Managess → plan the bid for system and plan the process

system engineers → to understand what system to develop

system test engineers → Develop validation tests

system maintenance engineers → Understand system and relationship between parts

Most ~~widly~~ An widely used standard for SRS is IEEE/ANSI 830-1998

- Preface
- Intro.
- Glossary
- User Reqs.
- System Arch.
- System req.
- System models
- System evolution
- Appendices
- Index.