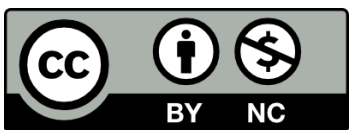
2 REQUIREMENTS

course “software requirements and architecture”

Sep 2023



contents

contents

definition

- Colloquially, a requirement is anything that someone desires.
- Requirements are seen as properties that the systems (yet in project) shall possess when built.
- The requirements express the users necessities and restrictions that are placed on the system.
- From the point of view of the engineer, a requirement can also be defined as something that needs to be conceived.
- According to the IEEE 610.12-1990 standard, a requirement is:
  1. a condition or a capacity that someone needs to solve a problem or to achieve an objective;
  2. a condition or a capacity that must be verified or possessed by a system or by a system component to satisfy a contract, standard, specification, or other formally imposed documents;
  3. a documented representation of a condition or capacity, as in (1) or (2).

classification

- The IEEE 610.12-1990 standard definition divides requirements into two alternatives:
  1. user needs
  2. system capacities
- These two alternatives are strongly associated with two different requirement types that are designated as user requirements and system requirements.
- A requirement is a capacity that a system must possess, to satisfy the users necessities.

classification

- Requirements can be divided into:
  1. functional requirements
  2. non-functional requirements
- Classifying a requirement as either functional or non-functional depends on the point of view of the observer.
- A requirement that for a stakeholder can be perceived as being functional, can be considered as non-functional for a different one.
- The form of a skyscraper is a non-functional requirement for the civil engineer.
- It can be a functional requirement for the urban architect.

Image courtesy of KEKO64 at FreeDigitalPhotos.net

candidate requirement

- The word ‘requirement’ can also mean an indispensable condition or demand.
- The requirements within the scope of an engineering project are negotiable.
- A candidate requirement is a requirement that was identified by some elicitation technique.
- Their incorporation in the system depends on the agreements that are established in the negotiation process.
- The use of the term ‘candidate’ aims to emphasise the possibility of the requirement not being considered.

contents

definition

- A functional requirement describes a functionality to be made available to the users of the system.
- It characterises partially the system behaviour as an answer to the stimulus that it is subject to.
- This type of requirements should not mention any technological issue.
- Ideally, functional requirements must be independent of design and implementation aspects.
- Thus, one increases the technological alternatives that can be explored during the project

set of requirements

- Collectively, the set of functional requirements must be complete and coherent.
- The set is:
  - **coherent**, if there are no contradictions among its elements;
  - **complete**, if it considers all the necessities that the client wishes to see satisfied.
- These two characteristics are hard to ensure for highly-complex systems.
- Defining complete requirements is the most difficult attribute to achieve or evaluate.

implicit requirements

- The requirements that are obvious are frequently forgotten and as such are not documented, neither negotiated.
- When those (obvious) requirements should be undoubtedly implemented in the final system, the analyst must guarantee that they are documented and correctly handled.
- Since the client may not be aware about these requirements, they are commonly referred to as implicit requirements.
- An implicit requirement is a requirement included by the development team, based on the domain knowledge that it possesses, in spite of not having been explicitly requested by the stakeholders.
- An explicit requirement refers to a requirement that was requested by the clients and that is represented in the documentation.

implicit requirements

Image of the author

contents

definition

- A non-functional requirement corresponds to a set of restrictions imposed on the system to be developed.
- It establishes how attractive, fast, or reliable the system is.
- It includes time constraints, restrictions in the development process, or adoption of standards.
- Example technological restriction: "the product must be developed in C++".
- The relevance of a requirement of this type should be discussed and agreed upon between the clients and the development team.
- This avoids taking design and implementation decisions prematurely and hastily.
- alternative terms: quality requirement or quality attribute

independence from the functional requirements

- A non-functional requirement does not change the essence of the system functionalities.
- In sports, the colour of the ball does not affect its functionality.
- The functional requirements remain the same, regardless of the non-functional requirements associated with the system.
- Some projects originate in the non-functional aspects.
- If the current system is slow, then this characteristic may lead to the creation of a faster system that does exactly the same.

Image courtesy of satit_srihin at FreeDigitalPhotos.net

different chairs

different automobiles

independence from the functional requirement

- Non-functional requirements are applicable to the system as a whole and not just to some of its parts.
- Generally, non-functional requirements cannot be modularised.
- The non-functional requirements are frequently emergent properties of the system at hand.
- An emergent property of a system is a property that can be associated with the system as a whole, but not individually to each of its components.
- Reliability is a good example of an emergent property.
- It is not sufficient that all components are reliable, for the respective system to be also reliable.
- The size of a software application is an example of a property that is not emergent.

emergent property

The emergent functional property of transporting persons is only provided by the bicycle as a whole.

Image courtesy of Naypong at FreeDigitalPhotos.net

independence from the functional requirement

- If the system is designed only based on the functional requirements, it may exist as a monolithic entity.
- Non-functional requirements are crucial to decide the system architecture.
- The fulfilment of a non-functional requirement cannot be achieved in an isolated way.
- One cannot maximise a given non-functional requirement without sacrificing some other non-functional requirements.
- The selected level for the satisfaction of a given non-functional requirement affects the satisfaction of other ones.
- A system optimised for performance can see a reduction in its characteristics associated with maintainability.
- Adaptability, for instance, contributes positively to portability.

classification of non-functional requirements

Sommerville [2010] suggests three categories:

- **product requirements**: characterise aspects of the behaviour of the system itself (reliability, performance, efficiency, portability, usability, testability, and readability).
- **organisational requirements**: come from strategies and procedures established in the context of the manufacturing process of the system or the client organisation (process standards and implementation requirements).
- **external requirements**: have origin in external factors to the system and the development process (interoperability, legal, and ethical requirements).

classification of non-functional requirements

Robertson and Robertson [2006] propose eight types:

1. **appearance**: the aspect and the aesthetics of the system;
2. **usability**: the easiness of utilisation of the system and everything that permits a more friendly user experience;
3. **performance**: aspects of speed, real-time, storage capacity, and execution correction;
4. **operational**: characteristics about what the system must do to work correctly in the environment where it is inserted;
5. **maintenance and support**: attributes that allow the system to be repaired or improved and new functionalities to be added;
6. **security**: issues related to access, confidentiality, protection, and integrity of the data;
7. **cultural and political**: factors related to the stakeholders culture and habits;
8. **legal**: laws that apply to the system so that it can operate.

usability

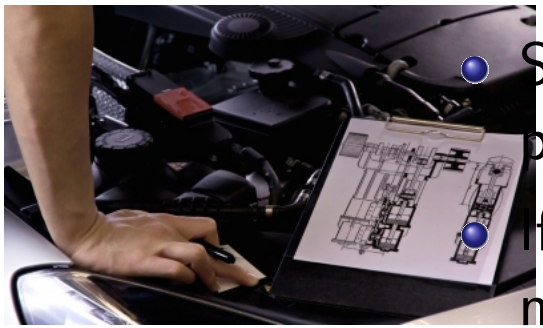
- Usability is a critical aspect for the success of many systems.
  - Ease of use is related to the efficiency of utilising a system and with the mechanisms that reduce the errors made by the users.
  - Personalisation is associated with the capacity of adapting the system to the tastes of the users.

- Ease of learning is concerned with the way users are trained to use the system.
- Accessibility indicates how easy it is to use the system, for people who are somehow physically or mentally disabled.

The product shall be easy to use for illiterate persons.
The product shall be intuitive to use for children with 4 years old.
The product shall be usable by visually impaired persons.

Image courtesy of cooldesign at FreeDigitalPhotos.net

maintenance and support



- System maintenance is divided into four types: preventive, corrective, perfective, and adaptive.
- If maintenance is important, one must consider maintenance requirements in the analysis phase.
- Modifiability is related to maintenance and is dependent on how easy it is to locate the components that must be changed.
- It is preferable that a given change has impact on a reduced number of components.
- In terms of support, it is important to know what kind of support and training the users are expected to be provided.
- Videos can be made to explain how to operate a given product.

The source code of the product programs should contain comments.
The product must be prepared to be translated to any language.

Image courtesy of ponsulak at FreeDigitalPhotos.net

legal

- Any system, regardless of the technology, has to respect the established laws.
- One should consult lawyers, legal advisors and jurists, for knowing whether any law or rule is being broken by the system.

- In Portugal, "portarias" n.º 363/2010 and n.º 22-A/2012 regulate the procedure for certification of software products that deal with invoices, by defining a set of technical rules.

The product shall be aligned with the PMBoK guide.
The product shall be certified by the Taxing Authority.
The product shall fulfil with the copyright.

Image courtesy of Kittisak at FreeDigitalPhotos.net

legal

"In 2031, lawyers will be commonly a part of most development teams."

Grady Booch (1955–), software engineer

contents

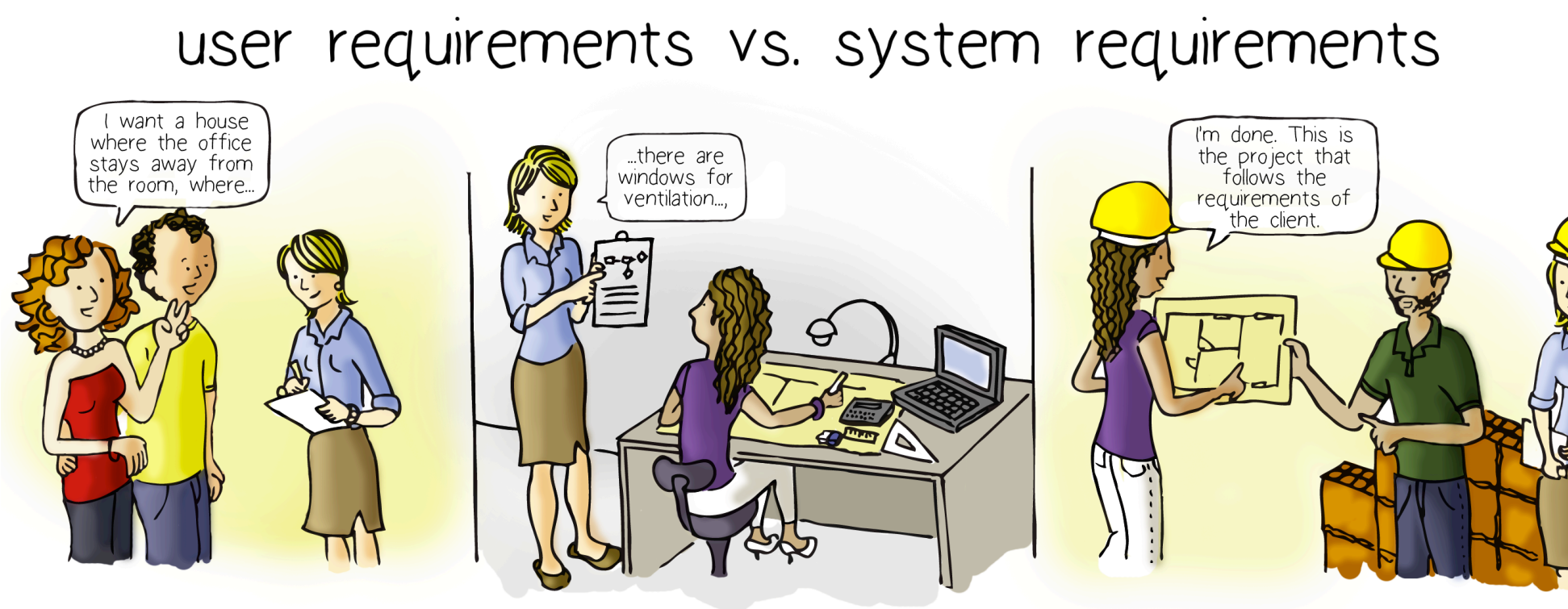
User requirement

- A user requirement represents:
  1. a functionality that the system is expected to provide to its users;
  2. a restriction that is applicable to the operation of that system.
- These requirements are related to the problem domain.
- They are normally expressed without great mathematical rigour, using natural language and informal diagrams.
- This permits stakeholders to read, analyse, and discuss those requirements.

system requirement

- A system requirement constitutes a more detailed specification of a requirement, being generally a formal model of the system.
- These requirements are oriented towards the solution domain, helping the engineers in the system design and construction.
- They are documented in a more technical language than the one adopted for the user requirements.
- It is desirable that they are independent from design and implementation pre-decisions.

relation between user and system requirements



problem domain

- The requirements result from necessities that exist in the problem domain to be addressed.
- User requirements should be described with the problem domain terminology.
- Engineers should avoid to speak with the terminology of the technological domain of the system under development.
- Premature inclusion of solution domain issues should be avoided.
- It is only possible to develop an adequate solution, if the problem to be solved is well characterised.
- Teams tend to develop solutions for problems that are poorly formulated.

Summary

- Requirements represent the necessities of the users and the constraints that are applied to a system and that must be considered throughout the development.
- The requirements can be classified, according to a first criterion, as either functional or non-functional.
- Non-functional requirements are divided in 8 different types: appearance, usability, performance, operational, maintenance and support, security, cultural and political, and legal.
- The requirements can be designated as either user or system requirements.
- User requirements are related to the problem domain and are usually expressed in a natural language.
- A system requirement is oriented towards the solution domain and is a detailed specification of a requirement, generally in the form of a formal model of the system.

bibliography

- Fernandes JM and Machado RJ; *Requirements in engineering projects*, Springer, Lecture Notes in Management and Industrial Engineering series, ISBN 978-3-319-18596-5, 2016. [chapter 3]

  `http://www.springer.com/978-3-319-18596-5`