

## Scrivere i requisiti

- Linguaggio naturale, integrato da diagrammi e tabelle, rappresenta il modo più diretto per descrivere i requisiti
- Questo è universalmente comprensibile, ma possono sorgere tre tipi di problemi:
  - *Perdita di precisione*. Questa è una caratteristica difficile da preservare senza rendere i documenti difficili da leggere
  - *Confusione nei requisiti*. I requisiti funzionali tendono ad essere 'mischiati' con quelli non funzionali
  - *Amalgamazione dei requisiti*. Molti requisiti diversi possono essere espressi insieme

1

## Il documento dei Requisiti

Il documento dei requisiti deve contenere tutte le informazioni relativamente a:

- Funzionalità
- Prestazioni
- Vincoli di progetto
- Interfaccia esterna

Esso deve contenere le seguenti sezioni:

- Introduzione
- Glossario
- Modelli del Sistema
- Definizione dei Requisiti Funzionali
- Definizione dei Requisiti non Funzionali
- Evoluzione del Sistema
- Specifica dei Requisiti

*Sommerville*

2

## La Struttura del documento dei Requisiti ...

- **Introduzione**
  - Descrive quali necessità il sistema risponde e come esso risponde agli obiettivi di business
- **Glossario**
  - Definisce i termini tecnici usati
- **I modelli del Sistema**
  - Definisce i modelli che mostrano i componenti del sistema e le reciproche relazioni
- **Definizione dei Requisiti Funzionali**
  - Descrive i servizi che devono essere forniti

*Sommerville*

3

## ... la Struttura del documento dei Requisiti

- **Definizione dei Requisiti non Funzionali**
  - Definisce i vincoli esistenti sul sistema e sul processo di sviluppo
- **Evoluzione del Sistema**
  - Definisce le assunzioni fondamentali su cui il Sistema è basato ed anticipa i possibili cambiamenti
- **Specifica dei Requisiti**
  - Fornisce la specifica dettagliata dei requisiti funzionali
- **Appendici – eventuali documenti di approfondimento**
  - Descrizione della piattaforma hardware del Sistema
  - Requisiti del Database
- **Indice**

*Sommerville*

4

**IEEE/ANSI 830-1993**

---

- 1 Introduction
- 2 General Description
- 3 Specific Requirements
- 4 Appendices
- Index

*IEEE/ANSI 830-1993*

5

**IEEE STD 830 1993  
Software Requirement  
Specification**

---

6

**Definizioni (IEEE Std 610.12-1990) ...**

---

**Requirement**

- 1) A condition or a capability needed by a user to solve a problem or to achieve an objective.
- 2) A condition or a capability that must be met or possessed by a system or a system component to satisfy a contract, standard, specification, or other formally imposed documents.
- 3) A documented representation of a condition or capability as in (1) or (2).

**Requirement Standard**

A standard that describes the characteristics of a requirement specification.

7

**... definizioni (IEEE Std 610.12-1990) ...**

---

**Requirements Analysis**

- 1) The process of studying user needs to arrive at a definition of a system, hardware or software requirements.
- 2) The process of studying and refining system, hardware or software requirements.

**Requirement Phase**

The period of time in the software life cycle during which the requirements for a product are defined and documented.

**Requirements Review**

A process or a meeting during which the requirements for a system, hardware item, or software item are presented to project personnel, managers, users, customers, or other interested parties for comment or approval.

8

... definizioni (IEEE Std 610.12-1990) ...

**Contract**

A legally binding document agreed upon by the customer and supplier. This includes the technical, organizational, cost and schedule requirements of a product.

**Customer**

The person, or persons, who pay for the product and usually (but not necessarily) decides the requirements. In the context of this document the customer and the supplier may be members of the same organization.

**Supplier**

The person, or persons, who produce a product for a customer. In the context of this document the customer and the supplier may be members of the same organization.

11

... definizioni (IEEE Std 610.12-1990) ...

**User**

The person, or persons, who operate or interact directly with the system. The user(s) and the customer(s) are not the same person (s).

**Requirement Specification**

A document that specifies the requirements for a system or component. Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.

**Requirement Specification Language**

A specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document hardware or software requirements.

10

... definizioni (IEEE Std 610.12-1990)

- **Functionality:** ... what the software is supposed to do ...
- **Performance:** ... the speed, availability, response time ...
- **Design Constraints:** ... policies for data base integrity, resource limits, operating environments ...
- **Attributes:** ... portability, correctness, maintainability, security ...
- **External Interfaces:** ... interaction with people, hardware, other software and other hardware ...

11

IEEE Standard su SRS (IEEE Std 830-1993)

Lo standard IEEE 830-1993 è espressamente dedicato al Software Requirements Specification (SRS).

THE SRS must correctly define all of the software requirements, but no more.

Il punto di partenza è costituito dalla definizione di sette attributi di qualità cui deve rispondere un SRS:

- 1) UNAMBIGUOUS
- 2) CORRECT
- 3) COMPLETE
- 4) VERIFIABLE
- 5) CONSISTENT
- 6) MODIFIABLE
- 7) TRACEABLE
- 8) RANKED (for importance and/or stability)

12

### IEEE Std. 830-1993 – NON AMBIGUO

An SRS is UNAMBIGUOUS if, and only if, every requirement stated therein has only one interpretation. In particular,

- 1) As a minimum, this requires that each characteristic of the final product is described using a single unique term.
- 2) In cases where a term used in a particular context could have multiple meanings, the term must be included in a glossary where its meaning is made more specific.

Ogni requisito ha una sola interpretazione possibile.

Un SRS è importante per tutte le altre fasi del ciclo di vita del software ...  
Esso dovrebbe essere non ambiguo sia per chi lo definisce che per chi lo usa, anche se il differente background rende questo task difficile

13

### IEEE Std. 830-1993 - CORRETTO

An SRS is CORRECT if, and only if, every requirement stated therein is one that the software shall meet.

Non esiste nessun tool o procedura che assicura la correttezza

L'SRS dovrebbe essere confrontato con qualche specifica superiore, come la specifica dei requisiti del sistema, altra documentazione di progetto, standard applicabili, ecc...

Ogni requisito rappresenta fedelmente nel sistema finale qualcosa che è stato richiesto

14

### IEEE Std. 830-1993 – COMPLETO ...

An SRS is COMPLETE if it posses the following qualities:

- 1) Inclusion of all significant requirements, whether relating to functionality, performance, design constraints, attributes or external interfaces.
- 2) Definition of the responses of the software to all reliable classes of input data in all realizable classes of situations. Note that it is important to specify the responses to valid and invalid input values.
- 3) Full labelling and referencing of all figures, tables, and diagrams in the SRS and definition of all terms and units of the measure.

Contiene i requisiti di tutte le funzionalità che il sistema si suppone esegua, e specifica, per tutte le possibili classi di input, la risposta del sistema

La completezza è spesso ottenibile solo incrementalmente

15

### ... IEEE Std. 830-1993 - COMPLETO

Any SRS that uses the phrase

... TBD (To Be Determined) ...

is not a complete SRS. If it is necessary, it should be accompanied by:

- 1) A description of the conditions causing the TBD (for example, why an answer is not known) so that the situation can be solved.
- 2) A description of what must be done to eliminate the TBD.

Un documento di progetto basato su un SRS che contiene TBDs, dovrebbe:

- Identificare la versione o lo specifico numero di release del SRS associato con il particolare documento
- Escludere noiose commissioni dipendenti da sezioni dell'SRS che sono ancora identificate come TBDs.

16

### IEEE Std. 830-1993 – VERIFICABILE ...

An SRS is VERIFIABLE if and only if every requirement stated therein is verifiable. A requirement is verifiable if and only if there exists some finite cost-effective process with which a person or machine can check that the software product meets the requirement.

If a method cannot be devised to determine whether the software meets a particular requirement then that requirement be removed or revised.

Se un requisito non è esprimibile in termini verificabili nel momento in cui l'SRS viene definito, dovrebbe essere definito un momento del ciclo di vita del software entro cui il requisito deve essere presentato in una forma verificabile

17

### ... IEEE Std. 830-1993 - VERIFICABILE

Un requisito non verificabile contiene espressioni del tipo 'works well', 'good human interface', and 'shall usually happen'.

Questi requisiti non possono essere verificati perchè è teoricamente impossibile verificare termini come *good*, *well* o *usually*.

Questa caratteristica richiede la non ambiguità dei requisiti  
Un documento di SRS viene convalidato

18

### IEEE Std. 830-1993 - CONSISTENTE

Consistency refers to internal consistency.

An SRS is CONSISTENT if and only if no set of individual requirements described in it conflict. There are three types of likely conflicts in an SRS:

- 1) two or more requirements might describe the same real world object but use different terms for that objects.
- 2) the specified characteristic of the real world object might conflict.
- 3) there might be a logical or temporal conflict between two specified actions.

non esistono requisiti che sono in conflitto con altri

19

### IEEE Std. 830-1993 - MODIFICABILE

An SRS is MODIFIABLE if and only if its structure and style are such that any necessary changes to the requirements can be made easily, completely and consistently. Modifiability generally requires an SRS to:

- have a coherent and easy-to-use organization, with a table of contents, an index, and explicit cross-referencing;
- not to be redundant;
  - whenever redundancy is necessary, the SRS should include explicit cross-referencies to make it modifiable.
- Express each requirement separately, rather than intermixed with other requirements.

20

### IEEE Std. 830-1993 - TRACCIABILE

An SRS is TRACEABLE if the origin of each of its requirements is clear and if it facilitates the referencing of the requirement in future development or enhancement documentation. Two types of traceability are recommended:

- 1) Backward Traceability: depends upon each requirement explicitly referencing its source in previous documents.
  - requisito collegabile a qualche elemento del progetto e del codice
- 2) Forward Traceability: depends upon each requirement in the SRS having a unique name or reference number.
  - dal progetto e dal codice è possibile risalire al requisito corrispondente

Quando un requisito nell'SRS rappresenta un derivato di un altro requisito, dovrebbe essere fornita sia la tracciabilità forward che la backward

21

### IEEE Std. 830-1993 - CLASSIFICABILE

An SRS is RANKED for importance and/or stability if each requirement in it has an identifier to indicate either the importance or stability of that particular requirement.

Typically, all of the requirements that relate to a software product are not equally important. Some requirements may be essential, while others may be desirable.

Each requirement in the SRS should be identified to make these differences clear and explicit

22