Top of Form

Bottom of Form

**[SER415](https://quizlet.com/347889132/ser415-flash-cards" \t "_blank) Final Exam materials(cumulative)**

**8 quality  
measures for requirements**

1. Correct  
2. Unambiguous  
3. Complete  
4. Consistent  
5. Prioritized  
6. Verifiable  
7. Modifiable  
8. Traceable

**Correctness**

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

**Unambiguous:**

A requirement is unambiguous "if and only if it  
has only one interpretation"

**Completeness**

A set of requirements is complete "if and only if it  
describes all significant requirements of concern to the user,  
including requirements associated with functionality,  
performance, design constraints, attributes, or external  
interfaces"

**Consistency**

A requirement set is consistent "if and only if  
no subset of individual requirements described within it  
are in conflict with one another"

**Prioritized**

Requirements ranked by importance  
and stability

**Verifiability**

A requirement is verifiable "if and only if there exists  
a finite, cost-effective process with which a person or machine  
can determine that the developed software system does  
indeed meet the requirement

**Modifiable**

SRS modifiable "if and only if its structure and  
style are such that any changes to the requirements can be  
made easily, completely, and consistently, while retaining the  
existing structure and style of the set

**Traceable**

A requirement is traceable "if and only if the origin  
of each of its component requirements is clear, and there is a  
mechanism that makes it feasible to refer to that requirement  
in future development efforts".

**Other Quality Measures**

1) Clear  
2) Concise  
3)Cohesive  
4)Feasible  
5) Managed

Top of Form

Bottom of Form

**TERM**

Requirements

**DEFINITION**

Descriptions of the system SERVICES and CONSTRAINTS that are generated during the requirements engineering process

**TERM**

Phases of Requirements Engineering

**DEFINITION**

Elicitation

Analysis

Validation

Change Management

**TERM**

Requirements Elicitation is sometimes called \_\_\_\_\_\_\_ or \_\_\_\_\_\_\_

**DEFINITION**

discovery or gathering

**TERM**

Stakeholders

**DEFINITION**

End-users, managers, engineers involved in maintainance, domain experts, trade unions, etc

**TERM**

Requirements Analysis

**DEFINITION**

Translating requirements expressed as needs into software products.

Provide a model to bridge the chasm between business stakeholders and implementers (e.g. design docs)

Architecture

Higher level design

**TERM**

Requirements Validation

**DEFINITION**

Demonstrating that the requirements defined the system the customer really wants

**TERM**

Objectives with Inception

**DEFINITION**

1. Understand what to build

2. Identify key requirements

3. Determine at least one potential solution

4. Understand costs, schedule, and risk

5. Understand what process to follow and tools to use

**TERM**

Objectives with Elaboration

**DEFINITION**

1. Get a more detailed understanding or requirements

2. Design, Implement, validate and baseline the architecture

3. Mitigate risks, produce more accurate schedule & cost estimates

4. Deployment and Development Environments

**TERM**

Needs

**DEFINITION**

Problem or opportunity that must be addressed

**TERM**

Features

**DEFINITION**

A service the system provides

Identifiable but not implementable

WHAT not HOW

**TERM**

5 Heuristics in Problem Analysis

**DEFINITION**

1. Gain agreement on the problem definition

2. Understand the Root Causes

3. Identify Stakeholders and End Users

4. Define the Solution System Boundary

5. Identify Constraints

**TERM**

Functional Requirements

**DEFINITION**

What the system does

**TERM**

Non-functional Requirements

**DEFINITION**

How well the system does its thing

Stipulations or constraints on the system

**TERM**

Types of non-functional requirements

**DEFINITION**

Product

Organizational

External

**TERM**

Product requirement

**DEFINITION**

The reqs we often think of.

Reqs which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, usability, etc.

**TERM**

Organizational requirement

**DEFINITION**

Internal Stipulations

Reqs which are a consequence of org policies and procedures.

**TERM**

External requirement

**DEFINITION**

External Stipulations

Reqs which arise from factors external to the system and its development process

**TERM**

User Requirements

**DEFINITION**

Written for (and often with) customers

Natural language

Should describe functional and non-functional requirements so that they are understandable by system users who don't have detailed technical knowledge

**TERM**

System Requirements

**DEFINITION**

More detailed specifications

A structured doc setting out detailed descriptions of the system services

**TERM**

Requirements Elicitation Techniques

**DEFINITION**

-Individual Interviews

-Group Meetings

-Prototyping

-Questionnaires

-Observation

-Research

**TERM**

Individual Interviews

**DEFINITION**

2-way communication process

Time sensitive

Could be user, buyer, or expert

**TERM**

Group Interviews

**DEFINITION**

N-way communication

Groups of customers, cross-functional teams, buyers, experts, focus groups etc.

Semi-structured

Cons: Group think

**TERM**

Prototyping

**DEFINITION**

A structure for individual or group exploration

Participants are end users

Cons: Could push yourself into a corner early on

**TERM**

Questionnaires

**DEFINITION**

1-way communication

Possibly anonymous

Cons: False answers, Answer options that are too limiting, answer options that are too broad

**TERM**

Observation

**DEFINITION**

Watch real people in the domain

Ethnography

Cons: Observing can cause behavior to change

**TERM**

Research

**DEFINITION**

0-way communication

Finding and reading written info and artifacts

**TERM**

Use Case

**DEFINITION**

Describes sequences of events between an actor and a system that yield a result of value to the actor

A template for a collection of related scenarios

**TERM**

Use Case Specification (Parts)

**DEFINITION**

Objective

Primary Actor

Trigger

Secondary Actor(s)

Pre/Post Conditions

Scenarios (Success/Failure)

**TERM**

Actor

**DEFINITION**

Defines a coherent set of roles that users of an entity can play when interacting with the entity

Stick figure

**TERM**

A use case should focus on the users \_\_\_\_\_\_\_\_

**DEFINITION**

GOAL

(you should avoid functional decomposition)

**TERM**

Steps to create a use case

**DEFINITION**

1. Identify and Describe the Actors

2. Identify the Use Cases and write a brief description

3. Identify Actor to Use Case relationships

4. Outline the Individual Use Cases

5. Refine the Use Cases

6. Verify & Validate the Use Cases

**TERM**

<<include>>

**DEFINITION**

A stereotype of a dependency

A -> B

The behavior of B is ALWAYS included into A

**TERM**

<<extend>>

**DEFINITION**

A stereotype of a dependency

A <- B

A possible extension, behavior of B may be incorporated into A

**TERM**

Generalization

**DEFINITION**

B inherits the behavior and communication relationships of A and is allowed to override and extend

B is generally a standalone basic use case

Actors may apply Generalization as well

A <- B

e.g. Student <- Graduate Student

**TERM**

Analysis Modeling Techniques

**DEFINITION**

Data/object Models

Behavioral Models

Flow Models

**TERM**

Data/object Models

**DEFINITION**

\* Entity-Relationship (ER)

OOA&D

Data Dictionaries

**TERM**

Behavioral Models

**DEFINITION**

Use Case Models

State Machines

**TERM**

Flow Models

**DEFINITION**

Process/workflow Models

\* Dataflow Diagrams (DFD)

\* Sequence Diagrams

\* Activity Diagrams

**TERM**

Other Requirements Quality Measures

**DEFINITION**

Clear

Concise

Cohesive

Feasible

Managed

**TERM**

IEEE 29148

**DEFINITION**

A newer, longer doc (~100 pages).

Focused on definitions

**TERM**

SWEBOK

**DEFINITION**

Software Engineering Body of Knowledge

Focused on descriptions

**TERM**

DO-178C

**DEFINITION**

Avionics Software Standard

Based on consequences of failure

Full bidirectional traceability

**TERM**

Requirements Maturity Levels

**DEFINITION**

0 - Chaos! No Requirements

1 - Written requirements

2 - Organized

3 - Structured

4 - Traced

5 - Integrated

**TERM**

Change Request Management

**DEFINITION**

Single Channel for Approval

Requirements to

Design to

Code to

Test to

Maintenance

**TERM**

Why is Traceability so important?

**DEFINITION**

Quality

-Can we determine that the req is validated/verified?

Impact Analysis

-What other reqs are impacted?

-What people are affected?

-What downstream artifacts are affected?

**TERM**

5 Step CHANGE MANAGEMENT Process

**DEFINITION**

1. Plan for change

2. Baseline the reqs

3. Change Control Board (CCB)

4. Use a Change Control System

5. Maintain Traceability

**TERM**

Types of Decomposition

**DEFINITION**

Flow-down

Refinement

Completion

**TERM**

Flow-down Decomposition

**DEFINITION**

- Assigning requirements to appropriate subsystems

-An Architectural effort

**TERM**

Refinement Decomposition

**DEFINITION**

-Ensure reqs reach level of specificity where implementation can easily follow

-A Requirements effort

**TERM**

Completion Decomposition

**DEFINITION**

-Adding reqs to complete missing back traces from code to reqs

-Design or even implementation effort

**TERM**

Requirements specify \_\_\_\_ to build, not \_\_\_\_ to build it

**DEFINITION**

what/how

**TERM**

SMT-LIB

**DEFINITION**

well recognized standard for specifying formal constraints to be solved by an automated constraint solver