Requirements Engineering

2024/2025
Course 1

Questionnaire

Link: https://forms.office.com/e/AUyStbq32U

Previous years' answers – Online store

Online store - functionalities

productsView product   products details   product-add product   addressRemove product

visualise products   buy products   product- filter products

review products   item   **product** cart   product to cartCheckout

view products

etcadd/remove product   Delete product   Search product   filter products

product from cart7   cartDelete product   productAdd product

Online store - end-users

storeOwners of the store

type of store   Store client   access   people who are interested

general store   **administrators**   **store**   **people**   store manager

store owner   **Customer**

specific store   item   products   admins type of user

stocks- store

people who own the items   specialized people

Online store - how?

choices and experiences   experience with these projects

online shops   functionalities   possible   previous experience

online bookstore   **Personal experience**   Personal experienceNavigation

Personal choices   online stores application   Familiar with the domain

domainpersonal experience   experience with other platforms

Previous years' answers – University admission

University admission - functionalities

filter admissions

admissionConfirm admission
view status
admission feeView
students
create

document upload
chat

application

admission status
User
login

admission fee

multiple admission
faculties
registration
upload documents

admission answer
application status
result of admission

University admission -end users

students and teachers
TeachersStudentsSoon to be students

school students
high school
admission graduates- students

people
student
teachers of the university

Soon to be students
secretary
university
Admin

teacher
Future students

administrators
applications

users/Students
charge of students
university staff

University admission - how?

experience with such sites
experience with AcademicInfo

Personal experience

personal experinces
admission
Familiar with domain

experience with this kind

Previous years' answers - Hydro power plant

power flowHave
power generatedturn
power output- watch
power grid
power plantStart
View power
power outputView
power plant
power station
status
water
outputView power
managing
hydro power
power output
water flow
Power level
power management
Power calculator
Hydro Power Plant - functionalities
power plantCheck

Electronic engineers
people that are curious
plant workers
plant employees
plant operators
plant experts
power plantany
power plants
work
plant personnel
User
domain
plantany worker
specialized work
Managers
Engineers
hydro power
work of force
People who have plants
administrators of the power
Hydro power plant - end users

Research online
Permissions in the database
Hydro power plant- how?
end-users
specific needs
Personal
experience in the domain
users
development team
guess
Familiar with the domain
common sense
hydro power
needs of this field
needs
power plant useful for the users
business
needs and requirements
relatives familiar
thinking logically

Previous years' answers – Wind power plant

power productionView
Power Calculator hydro power group wind
power stationShutdown
temperatureCheck power

power generatedGenerate **power plant** View power dataMeasure Power
power outputCheck

power production power station wind speed **wind power**
wind turbines

water power power outputView
power generatedturn

Wind power plant - functionalities wind speedCheck

---

specialized work **Wind power plant - end users**
plant workers Workers from the power plant operators
Regular user **people** **wind power** **Engineers**

plant employees **power plants** plant personnel

Managers regular people

wind plant**workers** **Administrator** Employees at the company

People from the domain user/visitor

people with expertise

---

development team **Wind power plant - how?**
Same common sense end-users
Deduction application
power plant

thinking logically Intuition **guess** Imagination Personal experience
t

Assumptions useful

Familiar with the domain hydro similarities
Asumption people

Previous years' answers - Credit due

Credits Due- functionalities

best songs
information about the songs
Search for a song
songsFilter songs
plays of a song   song metadata
songs as possible
multiple songs   artist   update   song metadata5
similar songs
new song   data for songs
playlist   duplicate songs
metadataEdit song
connections between songs
songsGroup songs

Credits Due - end-users

passionate about music   Record Company
enthusiastsRegular people
music creators
software company   Musicians
admin music and administrators
Music producersMusic   users   music people   Music producers
music industry   company   Employees   artists   regular people
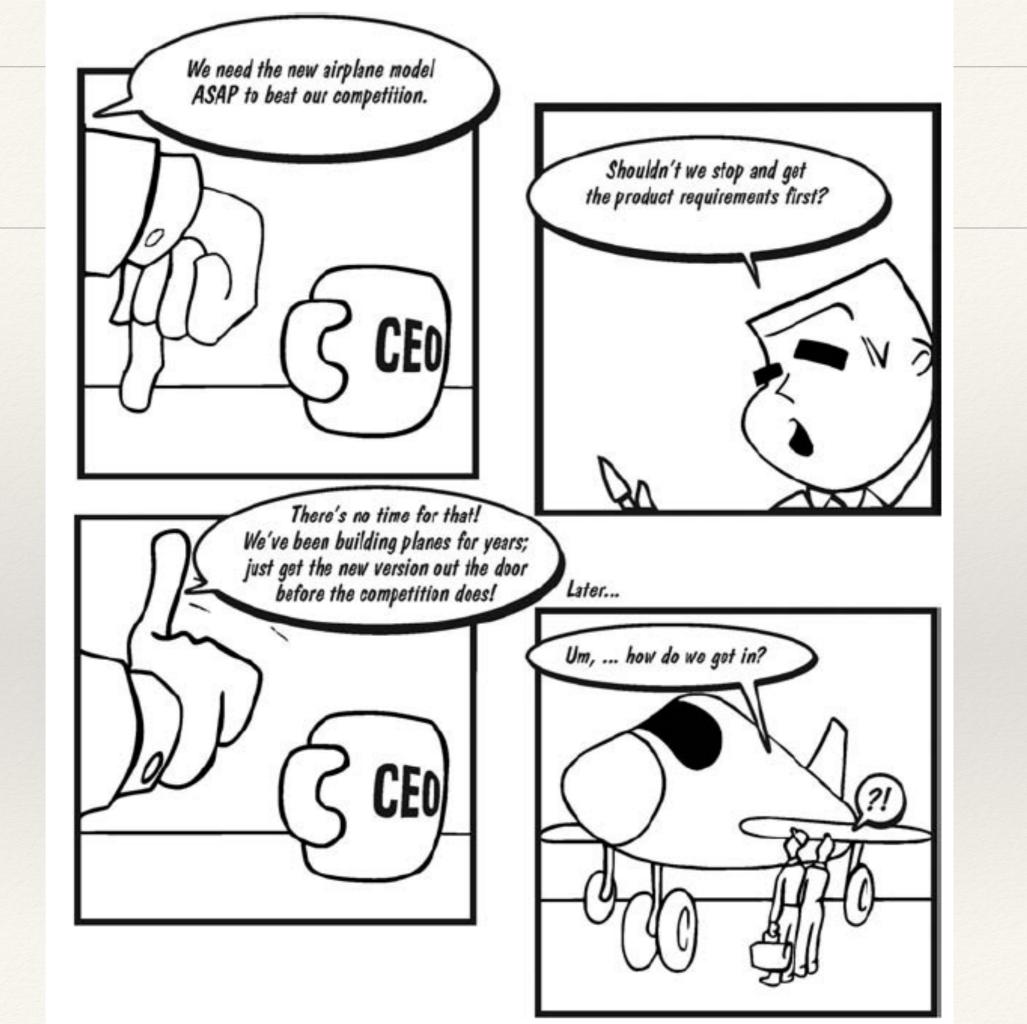Music enthusiastsPeople
People using the application

Credits Due - how?

Familiarity with the domain   similar applications   Asumption
Not sure   music   Intuition   imagination youtube experience
No time   Personal experience   music apps
type of functionalities
Familia with the domain   guess   application   domain regular
thinking   Familiar with the domain   experience with such platforms

"The hardest single **part of building a software** system is deciding precisely what to build." *

*Frederick P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering*.

Addison-Wesley, Reading, MA, 1995.

Requirements Engineering (RE)

❖ It addresses the critical problem of **identifying** and **designing** the **right software** for the customer.

❖ It is considered as **one of the most crucial stages** in software design and development.

❖ Software requirements are the critical determinants of software quality.

Requirements Engineering

❖ Studies (Standish group report - US projects) - Chaos

❖ 1995 Report

  ❖ 52.7% of projects cost 189% of their original budget estimates (only 42% of the original features were implemented).

  ❖ 16.1% of all US software projects are developed on-schedule, on-budget and with all originally planned features,

  ❖ 31.1% of projects are terminated before completion.

  ❖ Main problems: poor requirements, low user involvement and unclear objectives

❖ 1999 Report

  ❖ three of the top ten reasons for project failure: lack of user involvement, unstable requirements and poor project management.

Requirement Engineering

- 2001 Report :
  - unstable requirements and poor project management still among the primary reasons for project failure

- 1996 Europe survey:
  - 3800 organizations from over 17 countries in Europe
  - Most problems are in the area of requirements specifications (50%) and requirements management (50%)

- 2002 UK analysis
  - requirements problems accounted for 48% of all software problems encountered
- 2014, 2015 Chaos Reports

Content

- Introduction: Basic concepts, terminology
- Requirements Elicitation
- Types of Requirements
- Requirements Specification
- Requirements Prioritization
- Quality Assurance
- Requirements Management
- Requirements Negotiation

Bibliography

1. A. Aurum, C. Wohlin –*Engineering and Managing Software Requirements,* Springer, 2005

2. B. Berenbach, D. Paulish a.o. – *Software & Systems Requirements Engineering: In practice,* McGraww Hill, 2009

3. E.Hull, K. Jackson, J. Dick – *Requirements Engineering,* Springer, 2005

4. R. Young – *The Requirement Engineering Handbook,* Artech House, 2004

5. *Business Motivation Model* (BMM) http://www.omg.org/spec/BMM/

6. S. Robertson, M. Robertson, *Mastering the Requirements Process*, Addison Wesley, 3rd edition, 2013

7. Karl Wiegers, Joy Beatty, *Software Requirements (3rd Edition),* Microsoft Press; 2013

Course – Final mark

- Written exam: **60%**
- Project: **40%**
- **Both marks must be at least 5!**

Project

❖ You are responsible for the **requirements engineering of a software system**.

❖ You **may work in teams** (at most 4 students).

❖ You must also develop a prototype using the requirements obtained.

❖ **Deadlines**:

  ❖ **Requirements elicitation** (E): **week 6** (November 5, 2024)
  ❖ **Requirements specification** (RS): **week 11** (December 10, 2024)
  ❖ **Design and prototype** (DP): **week 14** (January 14, 2025)

❖ **Project evaluation**: 30% E, 40% RS, 30% DP

❖ 3 points penalty for each missed deadline

❖ The project cannot be delivered during the examination session.

❖ Projects delivered in the **re-exam session** will be graded **at most 5.**

Project Theme

- **Students' Internship** (help undergraduate students find a company for their mandatory internship).

- **Law 258/2007 (students' internship) - during studies**

- Law 176/2018 (companies' internship)

- Each member of a team has to discuss with at least 3 different stakeholders (end-users).

- Only one stakeholder should have computer science background.

Course 1 outline

- ❖ Definition
- ❖ Basic concepts
- ❖ Activities
- ❖ Artifacts

RE Definition (1)

❖ "**Requirements engineering** involves **all lifecycle activities** devoted to identification of user requirements, analysis of the requirements to derive additional requirements, documentation of the requirements as a specification, and validation of the documented requirements **against user needs**, as well as processes that support these activities." *

❖ It is a domain neutral discipline: software, hardware, and electromechanical systems.

* U.S. Department of Defense, Software Technology Strategy, December 1991.

RE Definition (2)

❖ The "science and discipline concerned with establishing and documenting software requirements." *

* Thayer, R. and Dorfman, M., Software Requirements Engineering, 2nd ed., IEEE Computer Society Press, 1997.

❖ RE deals with all phases of a project or product life cycle from innovation to obsolescence (innovation → development → release→ maintenance→ obsolescence).

Requirements

❖ Requirements are descriptions of how a software product should perform.

❖ A requirement typically refers to some aspect of a new or enhanced product or service.

❖ Definition*

(1) A condition or capability needed by a user to solve a problem or achieve an objective,

(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

A documented representation of a condition or capability as in (1) or (2).

*[IEEE-STD 610.12, *Standard Glossary of Software Engineering Terminology*, Institute of Electrical and Electronics Engineers, 1990]

Requirements

❖ A **requirement** is a necessary attribute in a system, a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a customer or user.

❖ A **requirement** is *a collection of needs arising from the user and various other stakeholders* (general organization, community, government bodies and industry standards), all of which must be met.

❖ The **requirements should be independent of design**, showing "**what**" the system should do, rather than "how" it should be done.

❖ Stated vs real requirements:

  ❖ **Stated requirements** are those provided by a customer at the beginning of a system or software development effort.

  ❖ **Real requirements** are those that reflect the verified needs of users for a particular system or capability.

Stakeholders

- Are "… those **participants in the development process** together with any **other individuals, groups or organizations whose actions can influence or be influenced by the development and use of the system** whether directly or indirectly" *.

- **Typical stakeholders**
    - business owners
    - product managers
    - **various types of users and administrators from the client side**,
    - software team members from the software development side.

*Pouloudi A, Whitley EA (1997) *Stakeholder identification in inter-organizational systems: Gaining insights for drug use management systems*. European Journal of Information Systems, vol 6: pp. 1--14

Stakeholders

- **Online store**
  - Stakeholders: …

- **Admission platform**:
  - Stakeholders: …

- **Publications Management System**:
  - Stakeholders:…

Requirement characteristics

❖ A **good requirement** should be [IEEE Standard 830, IEEE Recommended Practice for Software Requirements Specifications, 1998]:

  ❖ *Feasible*
  ❖ *Correct* (valid)
  ❖ *Unambiguous*
  ❖ *Verifiable*
  ❖ *Modifiable*
  ❖ *Consistent*
  ❖ *Traceable*

❖ Other project- or product-specific characteristics:

  ❖ compliance with specific regulations,
  ❖ meeting electrical safety requirements,
  ❖ etc.

Feasible

❖ A **requirement is feasible** if *an implementation of it on the planned platform is possible within the constraints of the program or project.*

❖ A **requirement is feasible** if and only if it can be accomplished given the resources, budget, skills, schedule, and technology available to the project team.

❖ E.g. a requirement to handle 10,000 transactions per second
  ❖ it might be feasible given current technologies (Oracle, MSServer, MySql)
  ❖ it might not be feasible with the selected platform or database manager (Sqlite).

Valid / Correct

❖ A **requirement is valid** if and only if the requirement is one that the system shall (must) meet.

❖ Determination of validity is normally accomplished by review with the stakeholders who will be directly responsible for the success or failure of the product.

❖ "must" vs "nice to have" requirements:
  ❖ **must**: they are actually needed to make the project or product a success.
  ❖ nice to have: wishful thinking. They add cost without adding value. They can delay project completion.

Remark:

The IEEE Standard 830 uses the term "correct" (without error).

A "valid" requirement may be exactly what the customer wants, but it may still contain errors or be an inappropriate solution.

Unambiguous

❖ A **requirement is unambiguous** if it has only one interpretation.

❖ Natural language tends toward ambiguity.

❖ E.g.: "The data complex shall withstand a catastrophe (fire, flood)."

  ❖ "The data complex shall withstand *a catastrophe of type fire or flood*".

  ❖ "The data complex shall withstand *any catastrophe, two examples being fire and flood.*"

  ❖ Rephrased: "The data complex shall be capable of withstanding a severe fire. It shall also be capable of withstanding a flood."

Verifiable

❖ A **requirement is verifiable** if the finished product or system can be tested to ensure that it meets the requirement.

❖ Product features are almost always abstract and thus not verifiable.

❖ E.g. "The car shall have power brakes." (it does not have sufficient details)

    ❖ "The car shall come to a full stop from 60 miles per hour within 5 seconds".

Modifiable

❖ The characteristic **modifiable** refers to two or more interrelated requirements or a complete requirements specification.

❖ A **requirements specification is modifiable** if its structure and style are such that any changes to a requirement can be made easily, completely, and consistently while retaining the structure and style.

❖ The **requirements specification** should have a coherent, easy-to follow organization with no redundancy (e.g., the same text appearing more than once). It *should keep requirements distinct* rather than intermixed.

Rule:

Information in a set of requirements should be in one and only one place so that a change to a requirement does not require cascading changes to other requirements.

Consistent

- Consistency is a relationship among at least two requirements.

- A **requirement is consistent** if it does not contradict or is not in conflict with any external corporate documents or standards or other product or project requirements.

- Contradiction occurs when the set of external documents, standards, and other requirements result in ambiguity or a product is no longer feasible to build.

- E.g. conflicting requirements
    - **Company standard** requires **all user interfaces** to have **a logo in upper right corner of the screen**.
    - **A user interface requirement** specifies that **the logo must be at the bottom center of the screen**.

Traceable

❖ ***Requirements traceability*** is *the ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases.*

❖ A **requirement is traceable** if the source of the requirement can be identified, any product components that implement the requirement can easily be identified, and any test cases for checking that the requirement has been implemented can easily be identified.

Requirements Specification

- A **requirements specification** is *a filtered set of requirements.*
- "Given two requirements specifications, how would you quantitatively determine that one is better than the other?"
- Characteristics:
    - *feasible* -- if building the product specified is feasible given the state of technology, the budget, and time.
    - *unambiguous* -- if there is no pair-wise ambiguity in the specification.
    - *valid* -- if every requirement in it is valid.
    - *verifiable* -- if every requirement in it is verifiable.
    - *modifiable* -- if there is no redundancy, and changes to requirements are easily and consistently made; e.g., a change to one requirement does not require cascading changes to other requirements.
    - *consistent* -- if the requirement set is internally consistent.
    - *traceable* -- if every requirement in it can be traced back to its source and forward to test cases.
    - *concise* -- if the removal of any requirement changes the definition of the product or system.

Completeness

❖ A **requirements specification is complete** if *it includes all relevant correct requirements, and sufficient information is available for the product to be built.*

❖ A **requirements specification is complete** if it includes the following elements [IEEE Standard 830]:

  ❖ *Definition of the responses of the system or product to all realizable classes of input data in all realizable classes of situations.* It is important to specify the responses to both valid and invalid input values and to use them in test cases.

  ❖ *Full labels and references to all figures, tables, and diagrams in the specification and definitions of all terms and units of measure.*

  ❖ *Quantification of the nonfunctional requirements.* Testable, agreed-on criteria must be established for each nonfunctional requirement.