SACRAMENTO STATE

California State University, Sacramento Computer Science Department

CSc 131 : Computer Software Engineering

Fall 2022

Lecture # 2

Software Engineering, System Engineering & Requirement Engineering

System Engineering

- Before SW can be engineered, the **system and its environment** must be understood.

- To accomplish this :
  - **Overall objectives** of the system must be determined.
  - The **role** of HW, SW, people, procedures must be identified.
  - **Operational requirements** must be elicited, analyzed, specified, modeled, validated, and managed.

SACRAMENTO STATE
Redefine the Possible

System Engineering – Two Views

Business Process Engineering

- Focuses on utilizing IT effectively.

Product Engineering

- Focuses on converting the customer's needs into a working/functional product.

System Engineering

- Concentrate not only on the software but rather on the **system as a whole and its elements**.

- SE occurs as a result of a process called **system engineering.**

System Modeling

- Model the system
  - Easier to **assess** the system.
  - Easier to **evaluate** system components in relationship to one another.

  - More on system modeling later….

System Context Diagram (SCD)

- It establishes the **information boundary** between the system being implemented and environment in which the system operate.
- It defines all **external producers** of information.
- It defines all **external consumers** of information.

System Context Diagram (SCD)

- The objective of the system context diagram is to focus attention on external factors and events that should be considered in developing a complete set of systems requirements and constraints.
- A system context diagram represents all external entities that may interact with a system.

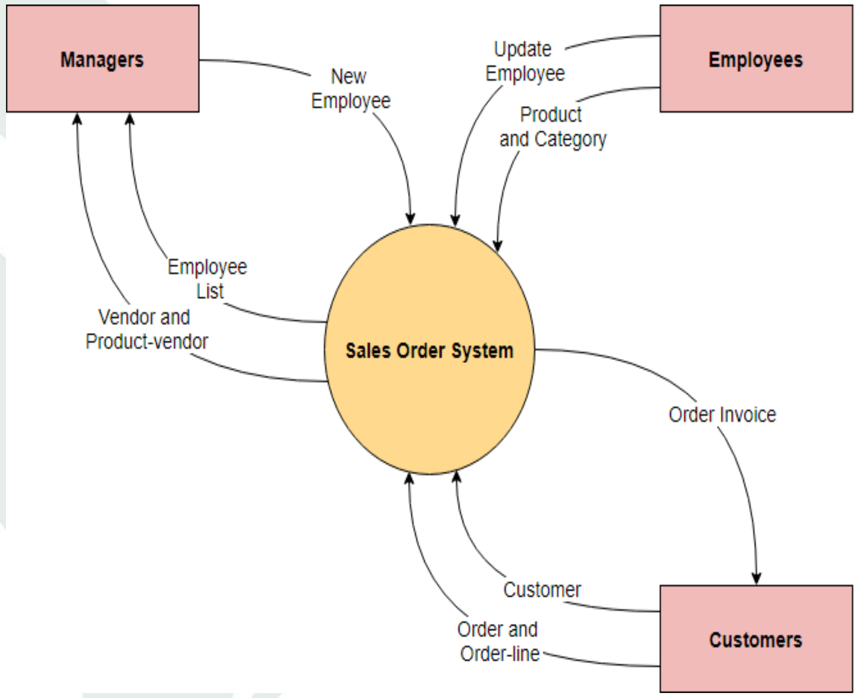


Fig : 1[1]

Software Development Lifecycle (SDLC)

Phases:

- Requirement Engineering
- Design
- Implementation (Coding)
- Testing
- Deployment
- Maintenance

Software Development Lifecycle (SDLC)

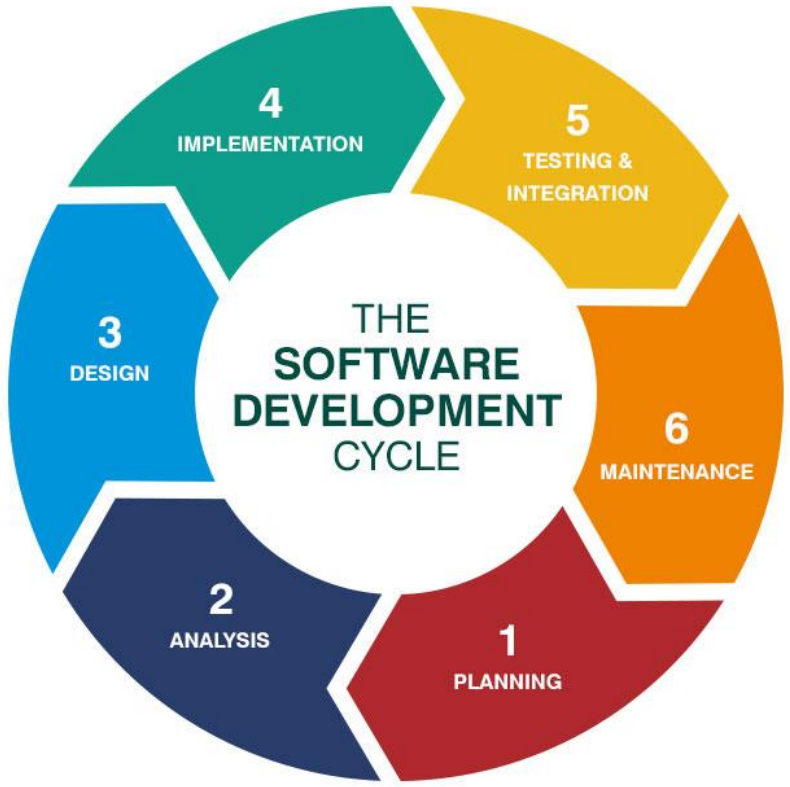


Fig : 2[2]

Requirements Engineering

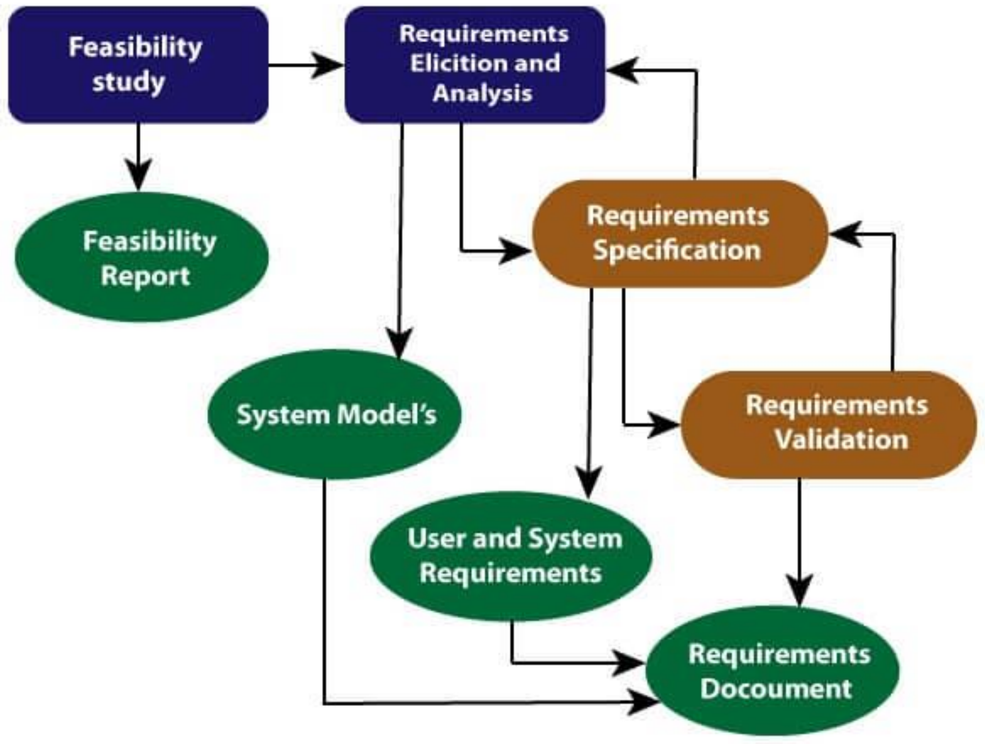
The outcome of the system engineering process is the specification of computer-based system at the different levels.

Requirements Engineering

- It focuses on assessing if the system is useful to the business (feasibility study), discovering requirements (elicitation and analysis), converting these requirements into some standard format (specification), and checking that the requirements define the system that the customer wants (validation).[3]

- In practice, requirements engineering isn't sequential process, it's an iterative process in which activities are interleaved.

Requirements Engineering Process



Requirement Engineering Process

Fig : 3[4]

Definitions

"A feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose"

- A **requirement** is a property or behavior that something must exhibit—it is something demanded or necessary.
- A **specification** is a precise description of something.
- A **requirements specification** is a precise description of properties or behaviors that something must have.

Requirement Types?

- Types of Requirements
  - **Business** Need / Requirements
  - **System Req.:**
    - **Functional** requirements
    - **Non-functional** requirements
  - **Domain** requirements
  - Inverse requirements

Functional vs. Non-Functional Requirements

Functional requirements

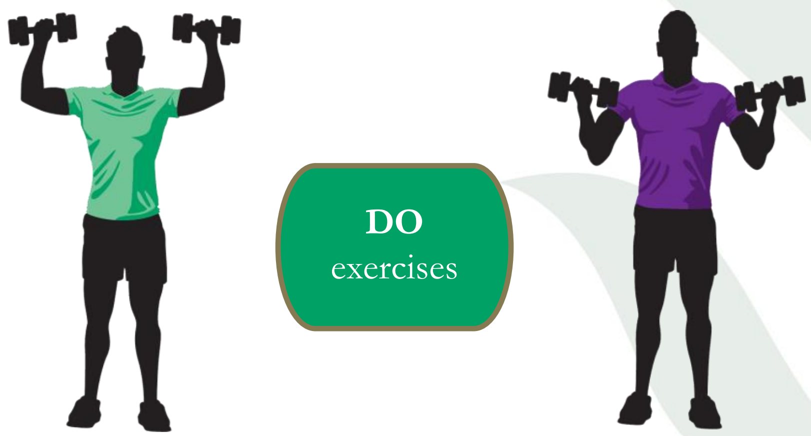
What the product must **DO**

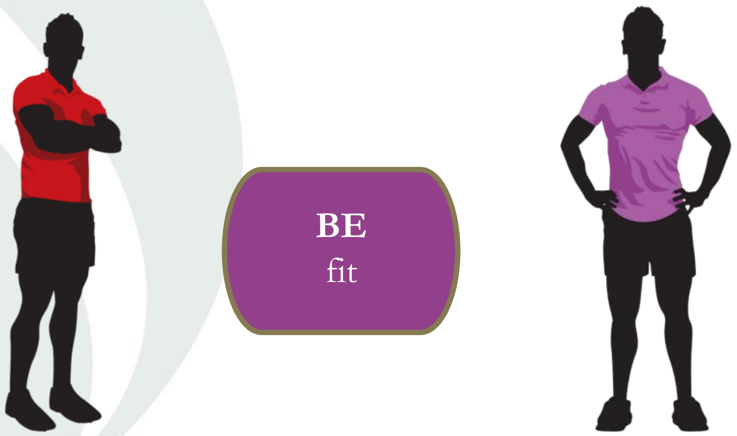
How a software product function. The behavior that maps inputs to outputs.

Non-functional requirements

What the product must **BE**

Properties that a software product must have.





SACRAMENTO STATE
Redefine the Possible

Attributes of a Good Requirement Statement

Complete – contains all information needed to implement it.

Clear – no ambiguity, not possible for different stakeholders to interpret differently.

Correct – is truthful and accurate. You don't want the implementation based on incorrect criteria.

Consistent – does not conflict with other requirements.

Testable – will be possible to verify that the implementation fulfilled the requirement.

Evaluate the Requirements (Individually and as a whole)

- The system shall be available in both English and Korean.
- Each user of the system shall be assigned a unique id upon account creation.
- The system shall store the full name, age and date of birth for each user.
- The system shall be able to generate a report listing every user's id, full name and age.
- All reports generated by the system shall list the user's email.

Complete ?

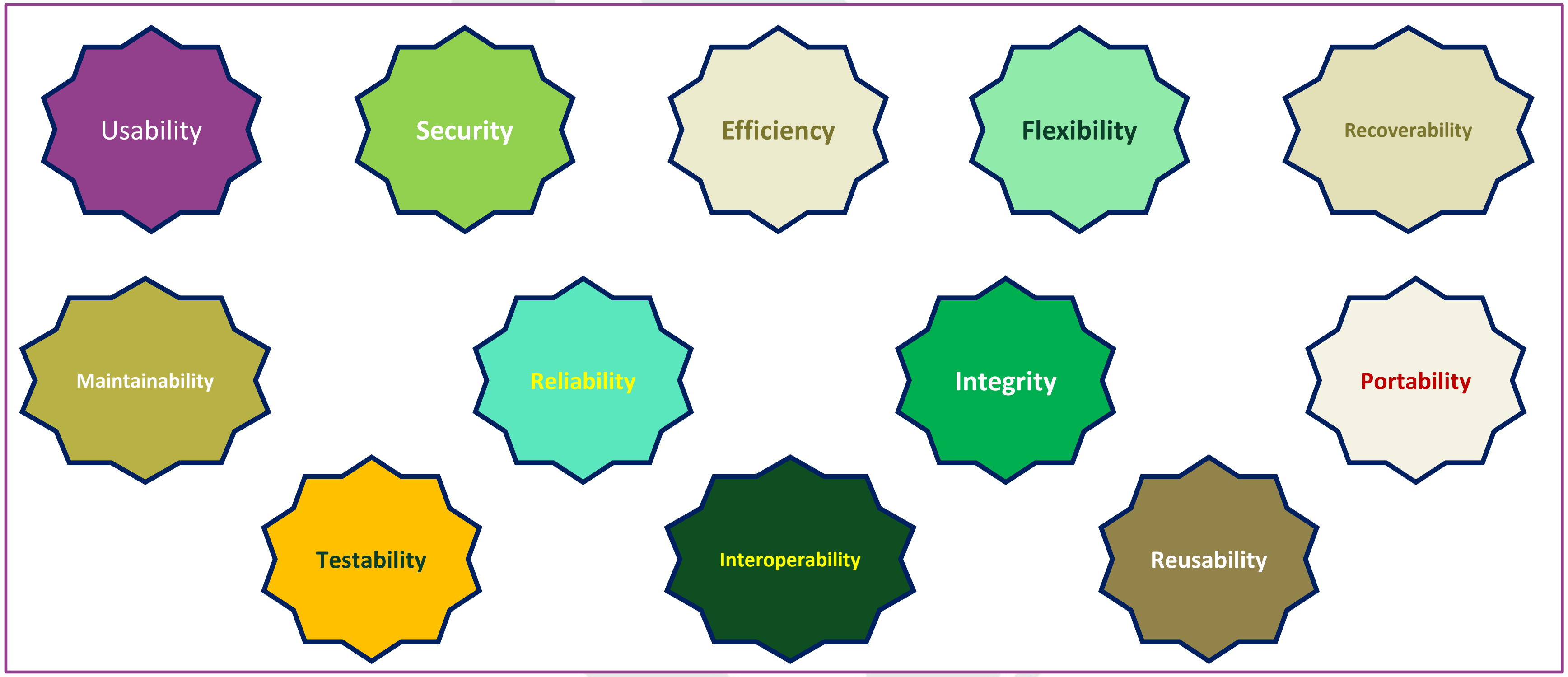
Clear ?

Correct ?

Consistent ?

Testable?

Many NFRs


Usability
Security
Efficiency
Flexibility
Recoverability
Maintainability
Reliability
Integrity
Portability
Testability
Interoperability
Reusability

Functional Requirements vs. NFRs

Functional

System must accept cash deposits

System must accept check deposits

System must accept PIN

System must verify PIN

System must allow cash withdrawals

DO

Non-functional

System must be **fast**

System must be **secure**

System must be **reliable**

System must be easy to **maintain**

System must be easy to **use**

BE

SACRAMENTO STATE
Redefine the Possible

Domain Requirements

- Domain requirements reflect the environment in which the system operates in –

- Domain requirements are important because they often reflect fundamentals of the application domain.

Inverse Requirements

Inverse Requirements is about " what the system shall not do."

Example: The system shall not use red color in the user interface, whenever it is asking for inputs from the end-user.

Requirements According to Whom?

A product stakeholder is anyone affected by a **product** or its **development**.

Marketers
Customers
Regulators
Competitors
Developers
Clients
Users

Requirements According to Whom?

Stakeholders have a major say in the requirements for a product

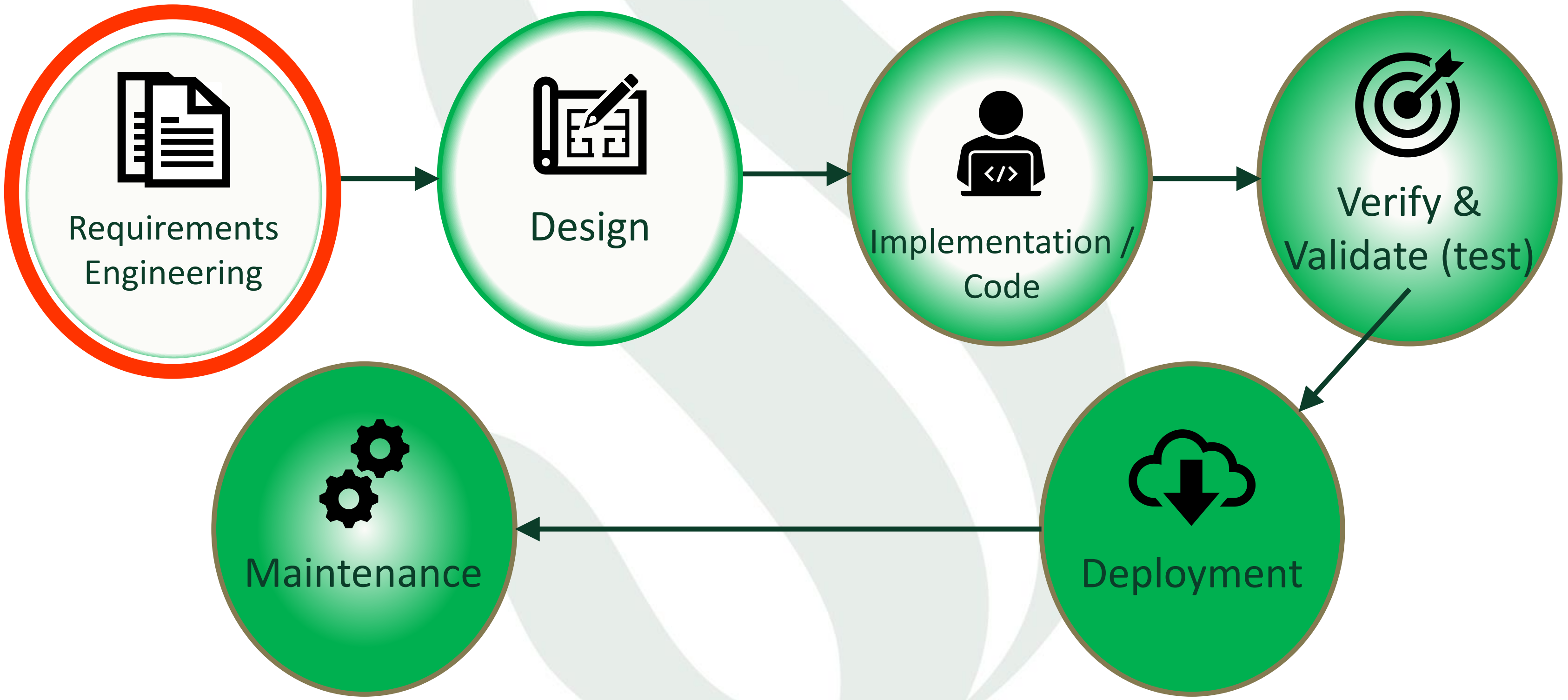
But keep in mind ..

- Conflicting desires and ideas.
- Incomplete and incorrect.
- Too abstract.

- Distinguish between stakeholder needs and requirements.
  - Stakeholder needs are features or properties that stakeholders desire in a product
  - Requirements are product features or properties agreed by all.

SACRAMENTO STATE
Redefine the Possible

SDLC – Requirement Engineering



Requirements Engineering

Requirements engineering process can be described in **seven** steps:

- **Inception**
- Requirement **Elicitation**
- Requirement **Elaboration**
- Requirement **Analysis & Negotiation**
- Requirement **Specification**
- Requirement **Validation**
- Requirement **Management**

Requirements Elicitation

Why requirement elicitation is difficult?

- **Problem of scope**
  - The boundary of the system is ill-defined
- **Problem of understanding**
  - Customers are not sure of what is needed
- **Problem of volatility**
  - Requirements change over time.

Requirements Elicitation

- **Techniques**
  - Interview / Meeting
  - Survey / Questionnaire
  - Observations
  - Temporary Assignment
  - Business Plans
  - Review Internal / External Documents
  - Review Software

Requirements Elicitation Techniques

Interviews

Observation

Focus groups

Workshops (brainstorming)

Prototypes

Document studies

Competitive product studies

Waterfall/Traditional Models

Elicitation is done up-front to create the complete Requirements Specification.

Agile Models

Needs are elicited continuously during development.

Elaboration

- The information obtained in the elicitation step is expanded and refined.
- Develop a **refined model**.

Requirements Analysis & Negotiation

- Once requirements have been gathered then ..

- **Categorize** requirements.
- **Organize** requirements into related subsets.
- Establish requirements **relationships**.
- **Examine** requirements consistency.
- **Rank** requirements based on the need of customers.

Requirements Analysis

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified.

It involves all the tasks that are conducted to identify the needs of different stakeholders.

It is to analyze, document, validate and manage software or system requirements.[

SACRAMENTO STATE
Redefine the Possible

Requirements Specification

- System Specification is the **final product** of system and requirements engineering.
  - It serves as the **foundation** for HW, SW engineering.
  - It describes the **function and performance** of a system/product and its constraints.

Requirements Specification…

- A specification can be:
  - A **written** document
  - A **graphical** model
  - A **formal mathematical** model
  - A **prototype**
  - Any combination of the above …

Requirements Specifications

Write complete, simple sentences in the active voice.

Define terms clearly and use them consistently.

Use the same words for the same thing—avoid synonyms.

Group related material into sections.

Use tables, lists, indentation, white space, and other formatting aids to present and organize material clearly and concisely.

Requirements Specifications

Express all specifications using the words “must” or “shall.”

Should be testable or verifiable.

Should be easy to understand and to change (during this phase).

Each specification should state only a single need or requirement – Atomic.

Each one should have a unique identifier.

Even when done carefully, specifications may still be vague, ambiguous, or hard to understand.

Specifications Example

Nonfunctional Requirements by Type

1. The product must be accessible as a WWW page.
    1.1. The main file for the product must be written in HTML. (S1, S2, S3, S4, S5, S6, S7, S8)
    1.2. The form/presentation of the user interface must be written in CSS.
    1.3. The algorithms required by the product must be written in Javascript.

Functional Requirements by Type

1. Startup Requirements
    1.1. Startup Study Guide Requirements
        1.1.1. The product must be able to start without a study guide.
        1.1.2. The product must be able to open a particular study guide at startup. (U3)
            1.1.2.1. It must be possible to identify the guide in the URL of the main file.
    1.2 It must be possible to include start-up option in the URL of the main file.
    1.2.1. It must be possible to specify the default server as a start-up option.
    1.2.2. It must be possible to specify the directory as a start-up option.

SRS Document

SRS is a document that describes what the software will do and details its functionality - to fulfill all stakeholders' business needs.

SRS – Structure

- A typical SRS often includes:
  - Software purpose
  - Software functions and description
  - Software specific requirements
  - Software assumptions and constraints
  - User characteristics

Why SRS Document

- SRS establishes the **basis for the entire project.**
- SRS ensures requirements are discovered, captured and ultimately delivered

SRS vs. Sys_Rs

Software Requirements Specification vs. System Requirements Specification

- SRS provides **in-depth descriptions** of the software to be built
- Sys_RS captures **needed information** for the overall system requirements.
- **SRS provides much detail than a system requirements specification.**

Requirements Validation

Why requirements validation?

- To Ensure the following:
  - All system requirements have been stated clearly.
  - Inconsistencies, errors, omissions have been detected and corrected.
  - Product conforms to the standards.

- Mainly done by **Formal Technical Review Team**.

Verifying and Validating Requirements

The high cost of "getting it wrong" means we should test the requirements.

Verification means Are we building the product right?

Validation means Are we building the right product?

Test the requirement for "goodness" (clear, concise, complete, etc.)

Main technique – reviews

Keep in mind any assumptions. They may not be shared amongst stakeholders.

Requirements Management

It is a set of activities that support the project team to:

- Identify, control, and track requirements and changes to requirements at any time.

- **Why requirement management??**
  - Because requirements change…

Requirements Review?

- Are the requirements **complete**?
- Are the requirements **concise**?
- Are the requirements **correct**?
- Are the requirements **consistent**?
- Are the requirements **modular**?
- Can they accommodate **change**?
- Are the requirements **realistic**?
- Are the requirements **needed** by the customer?
- Are the requirements **traceable**?

Requirements Engineering in Agile

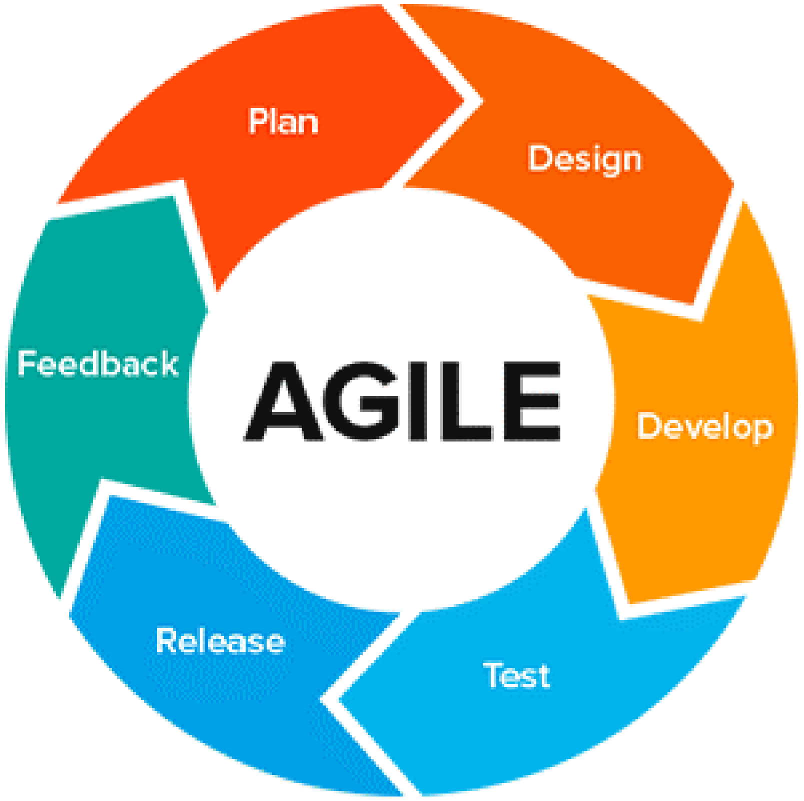


Fig : 5[9]

SACRAMENTO STATE
Redefine the Possible

Requirements Engineering in Agile How it is done

Engineers work with the customers and end- users to find out the application domain, services that system should provide, the required performance of the system, and so on.[8]

Requirements are:

- Categorized and organized into related subsets,
- Consistency of the requirements is checked
- Requirements are prioritized
- User stories are written and acceptance criteria is finalized

Requirements Engineering in Agile
PO Role

Product Owner (PO) is the key in this phase

"**voice of the customer**" throughout the cycle.

The PO must constantly monitor the Product Backlog and be able to identify this scenario and act in consequence.

Requirements Engineering in Agile Scrum Master Role

Scrum Master supports the PO to maintain the Product Backlog in a way that ensures that the needed work is well understood…

Ensures process is followed and removes blocker and impediments

Requirements Engineering in Agile Team Development Role

Development Team focus is to fully understand and flesh out each potential item from the Product Backlog.

Take on the role of analyst to understand scope and expectations ..

Requirements Engineering in Agile

- **Advantages and benefits working with agile model**

❖ Customer satisfaction by rapid, continuous delivery of useful software.

❖ People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.

❖ Working software is delivered frequently (weeks rather than months).

❖ Continuous attention to technical excellence and good design.

References

1. https://online.visual-paradigm.com/knowledge/system-context-diagram/what-is-system-context-diagram/
2. https://bigwater.consulting/2019/04/08/software-development-life-cycle-sdlc/
3. https://medium.com/omarelgabrys-blog/requirements-engineering-introduction-part-1-6d49001526d3
4. https://www.javatpoint.com/software-engineering-requirement-engineering
5. https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/Requirements/DomainReq.html
6. https://reqtest.com/requirements-blog/requirements-analysis/
7. https://www.dataart.com/blog/an-introduction-to-agile-requirements-engineering#:~:text=Requirements%20Engineering%20(RE)%20can%20be,is%20building%20the%20right%20product.&text=Verification%20%26%20Validation%2C%20which%20checks%20the,in%20accordance%20with%20scope%20control.
8. http://jultika.oulu.fi/files/nbnfioulu-201705091721.pdf
9. https://www.wearemarketing.com/blog/what-is-the-agile-methodology-and-what-benefits-does-it-have-for-your-company.html

SACRAMENTO STATE
Redefine the Possible

Questions ..?

Next topic: Requirement Engineering & Analysis Modeling