Code Academy

مـــكيـــن
Makeen

HANDLING USER REQUIREMENTS: FROM GATHERING TO VALIDATION

MARAM ALBARHOOMI
FATEMA ALSUDAIRI
AFRAH ALMAMRI


I meant 3 bedrooms,
a balcony, and a arden!

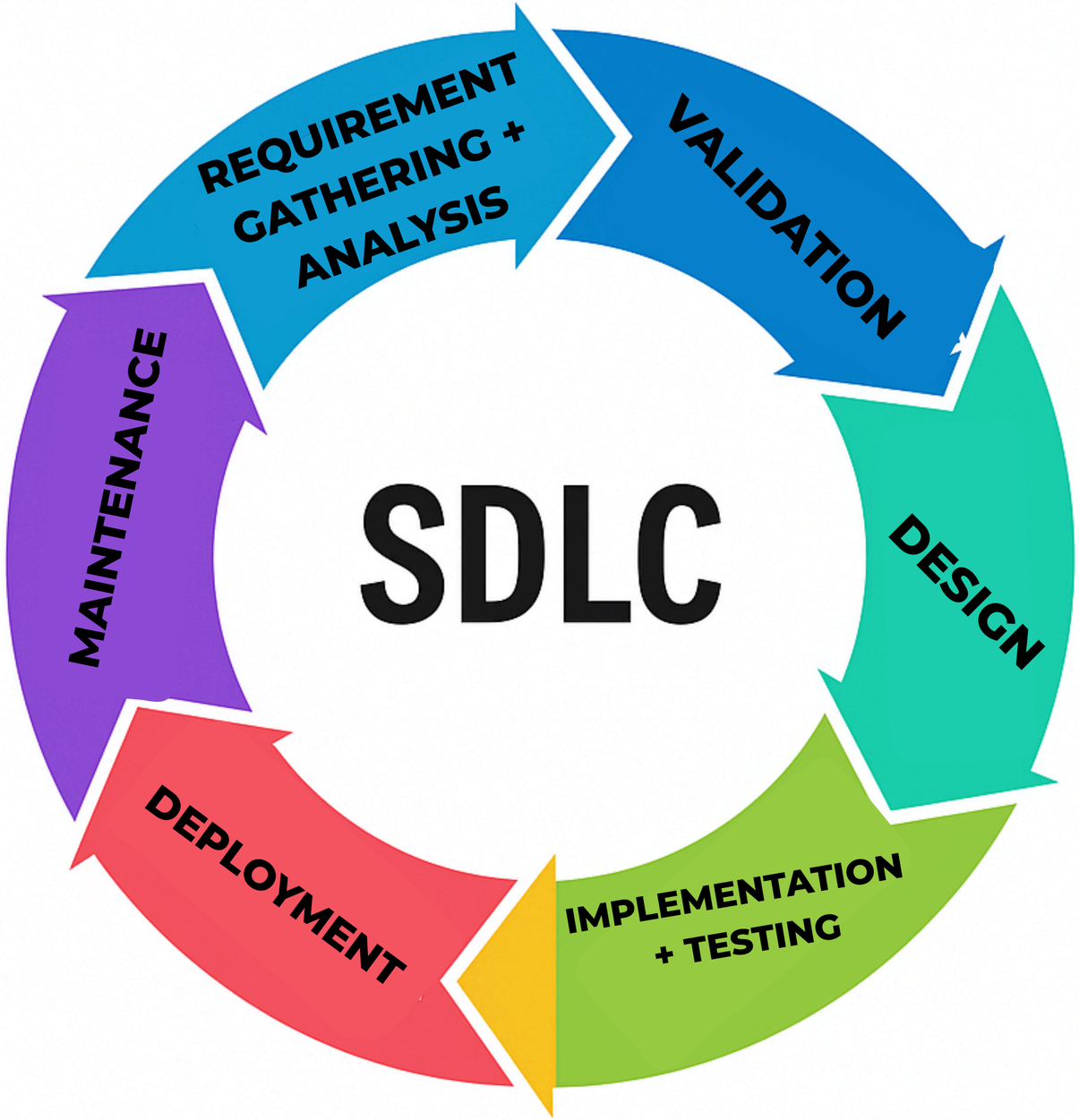
INTRODUCTION

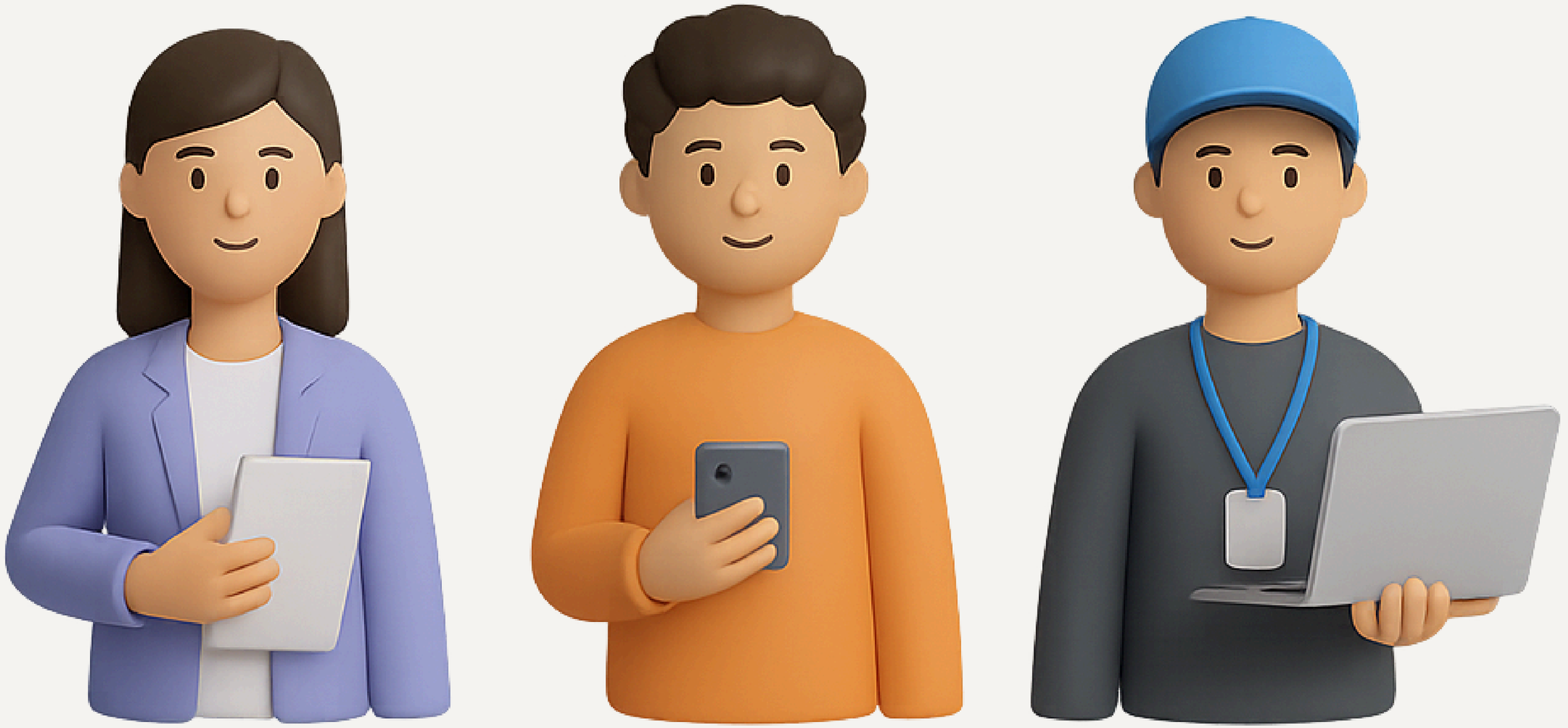
Have you ever asked for something and got something completely different?

yes , In software development, this happens when user requirements are unclear.

Today we’ll explore how to gather and validate user requirements step by step.

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)





WHAT ARE USER REQUIREMENTS?

A USER REQUIREMENT IS A STATEMENT THAT DESCRIBES WHAT THE END USER NEEDS OR EXPECTS FROM A SYSTEM, PRODUCT, OR SERVICE. IT IS USUALLY WRITTEN IN CLEAR, NON-TECHNICAL LANGUAGE TO ENSURE THAT IT’S EASILY UNDERSTOOD BY USERS AND STAKEHOLDERS.

GATHERING REQUIREMENTS

1) interviews with users

2) Surveys and questionnaires

3) Observations

4) Workshops and brainstorming sessions

5) Ask the right questions to avoid confusion later

QUESTIONS TO ASK (EXAMPLE)

What is the main goal of this system?

Who are the main users?

What features are absolutely necessary?

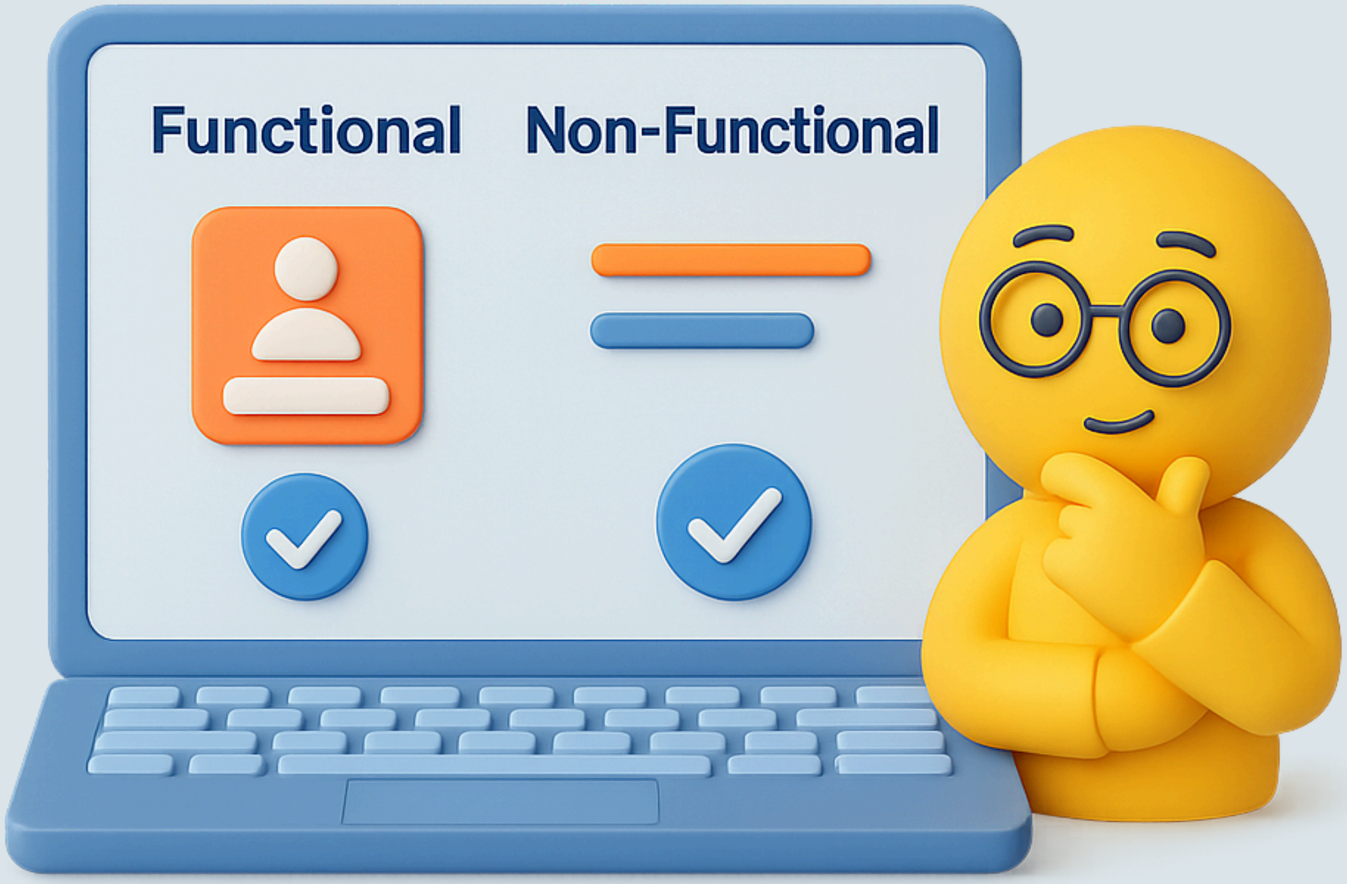
Are there similar systems you like/dislike?

What devices should it work on?

ANALYSIS & DOCUMENTATION

After gathering: analyze and organize

- Remove duplicates and contradictions
- Group by priority (Must have / Nice to have)
- Create a Software Requirements Specification (SRS)


Functional
Non-Functional

FUNCTIONAL REQUIREMENTS

1. ENTER STUDENT DATA
2. ENTER DESCRIPTIVE ASSESSMENTS
3. CREATE A DESCRIPTIVE REPORT FOR EACH STUDENT
   - THE SYSTEM GENERATES A DESCRIPTIVE REPORT FOR EACH STUDENT CONTAINING
4. LINK ASSESSMENTS TO THE CLASS, SUBJECT, AND TEACHER
5. DISPLAY REPORTS TO TEACHERS AND ADMINISTRATION

NON-FUNCTIONAL REQUIREMENTS

1. SECURITY
2. USABILITY
3. RELIABILITY
4. PERFORMANCE
5. MAINTAINABILITY

VALIDATION OF REQUIREMENTS

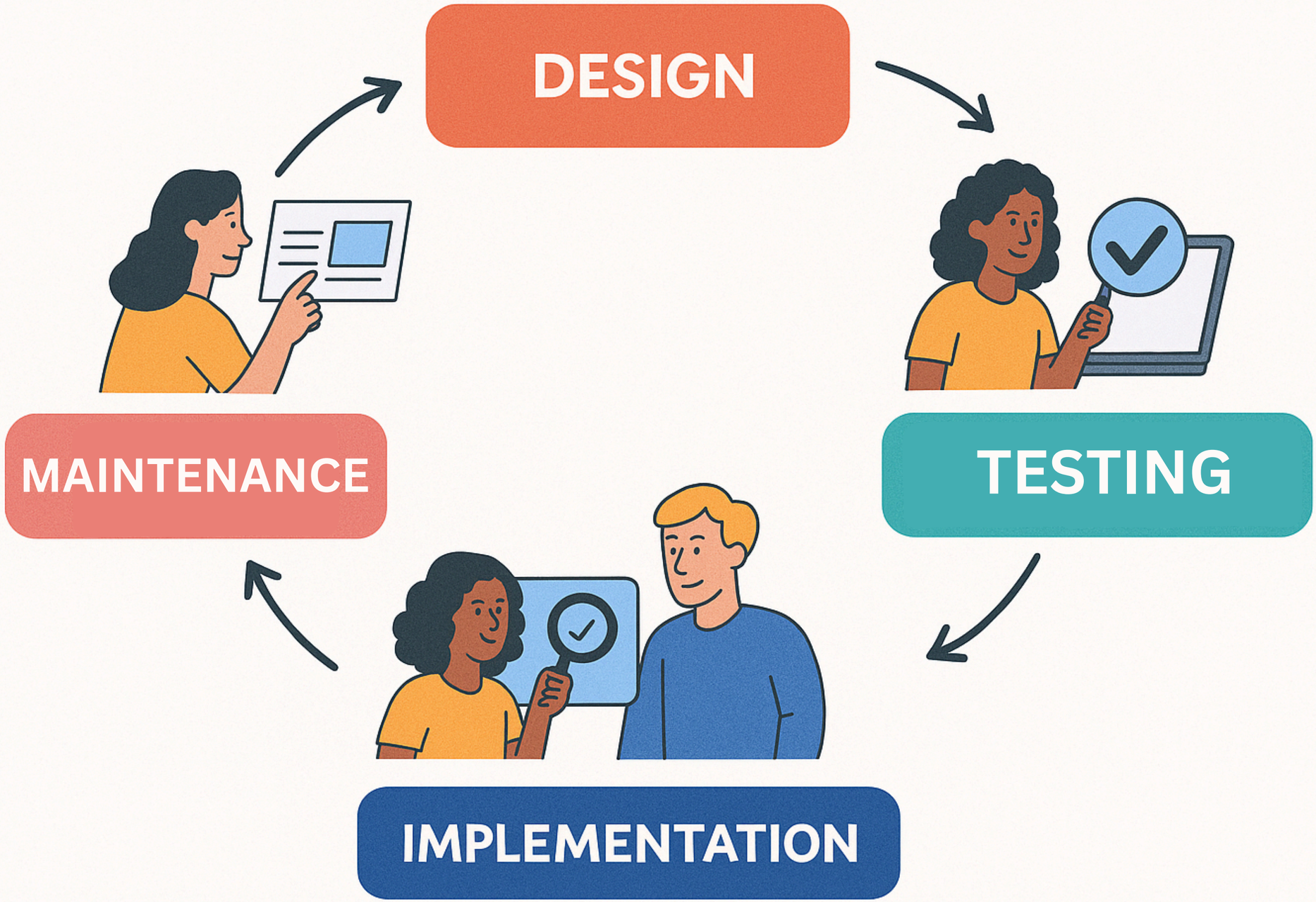
once requirements are gathered and documented, it's important to validate them.

this ensures that what was collected matches what the user actually needs.

HOW TO VALIDATE REQUIREMENTS:

- Meet with the user/client: Review each requirement together.
- Use wireframes: Simple visual sketches of how the system might look.
- Create mockups or prototypes: Interactive demos that simulate how the system will behave.
- Feedback sessions: Let users comment, request changes, or approve features.
-

VALIDATION VS VERFICATION


DESIGN
TESTING
IMPLEMENTATION
MAINTENANCE

REQUIREMENTS ARE VERY IMPORTANT, BUT THE STORY DOESN'T END WITH THEIR COLLECTION AND ANALYSIS...

ON THE CONTRARY, THIS IS WHERE THE SYSTEM IMPLEMENTATION JOURNEY BEGINS.

AFTER WE UNDERSTAND EXACTLY WHAT THE USER NEEDS,
WE MOVE TO THE DESIGN , PROGRAMMING, TEST THE SYSTEM, DELIVER THE SYSTEM TO THE CUSTOMER.

FINALLY, WE FOLLOW UP ON ADDITIONAL GUARANTEES, BECAUSE USER NEEDS NEVER STOP.

COMMON CHALLENGES:

❗ **Users don’t always know what they want.**

🔄 **Requirements may change frequently during the project.**

❌ **Miscommunication between stakeholders and developers.**

📄 **Requirements might be incomplete, unclear, or conflicting.**

⏳ **Time pressure leads to skipping validation or proper documentation.**

HOW TO OVERCOME THEM:

👂 **Practice active listening and ask clarifying questions.**

🧩 **Break down large requirements into smaller, manageable ones.**

🗂️ **Use visual tools like flowcharts.**

🔁 **Keep communication open with regular updates and reviews.**

In The End :

"IF YOU UNDERSTAND THE USER FROM THE START, YOU'RE BUILDING THE RIGHT PROJECT FROM THE START."

Reference:

https://ieeexplore.ieee.org/document/1007048

https:/www.tutorialspoint.com/sdlc/index.htm

https:/www.lucidchart.com/blog/validating-requirements

https:/www.freepik.com%20&%20

https:/www.canva.com

https:/iconscout.com

… thank you …