DIPLOMA WALLAH

(Your One Stop Hub For Diploma Resources)

SOFTWARE ENGINEERING

PRINCIPLES AND PRACTICES

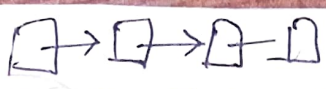
Complete Notes Based on Full Syllabus

- Diploma Engineering

4th Semester

www.diplomawallah.in

Notes prepared by **Sangam**



Sangam
Diploma wallah

unit-05

Design Thinking

Design Thinking is a problem-solving approach that focuses on understanding user's needs, refining problems, and creating innovative solns through creativity and iterative testing.

It is widely used in S/w engineering product development, UX/UI design, and business innovation, app development etc.

There are 5 key Stages:-

1. Empathize (understand the user)
- Gain a deep understanding of the people you are designing for by learning about their experiences, needs and problems.

Activities:
→ Conduct interviews.
→ Observe users in their environment →
- collect stories and emotions.

Ex If we building an app for online learning talk to students about their pain points -
eg- poor internet, distraction, lack of motivation etc.

2. Define - Identify the Real problem.
- Clearly State the problem statement based on insight gathered in the empathize stage.

Activities:-
- Analyze your observation.
- Group similar issues.
- Write user-centered problem statement (called POV- point of view)

Example:-
Instead of saying, "we need a better UI", define the problem like:
"Student in rular areas need a way to access learning materials without heavy internet usage".

3. Ideate - Generate Creative Solutions.
→ come up with as many ideas as possible - think outside the box.
Activities:-
- Brainstorming as many ideas possible.
- mind mapping: Think creative without judgement.
- "How might we..." questions.

Example:-
- offline-first learning app
- Downloadable video bundles.
- voice-based lessons.
- whatsApp chatbot tutor.

4. prototype - Create simple models.
→ Build quick and low-cost versions of the ideas to test and understand what works.
Activities:
- Sketching screens.
- Making paper models.
- wireframe or basic s/w demos.

Examples:-
Making a wireframe of the offline learning app using figma or draw it on paper.
Add main screen: home, lesson, quiz.

5. Test - Get Feedback and Improve
→ let real users try your prototype, get feedback and Iterate to improve the soln.
Activities
- User testing session
- Observe their reactions.
- Ask questions.

Example:-
Give the app prototype to a few students and ask them to perform tasks. Note what confuses them and what they like.

Summary
→ Design thinking is a human-centered approach to problem-solving.
Instead of focusing on what a product or service is it focus on what users truely need.

It encourage creativity, empathy and testing Ideas before final implementation.

① Empathize - understand the user problem & need.
→ observe & engage with real people.

Requirement Engineering

- A requirement is a description of what a system should do or a condition it must satisfy.
- It represent the needs and expectation of user and Stakeholders.
- Act as a bridge b/w clients (who want the system) and developers (who build it).

Importance

- Good requirements = Successful project.
- Wrong or unclear requirements = project failure.
- Clear requirement help in:
  - planning & scheduling
  - Reducing misunderstanding
  - lowering cost of fixing errors.
  - delivering a usable product.

Ex - If a client says "Attendance system required" but does not specify whether it is manual upload or biometric, the final system may be unusable.

Types of Requirement -

1. Functional Requirement (what the system should do)
   - features & operation

   Ex- user login, online payment, report generation.

2. Non-functional Requirement — (How the System Should do):
- Quality attributes
- Ex- Security, performance, usability, response time etc.

3. Domain Requirement:-
- Rules of the particular domain.
- Ex- Banking s/w must follow RBI Security guidlines.

4. Sources of Requirements
- client/customers:- Main Source.
- End-user - actual system users.
- Existing system - current proces/s/w.
- Domain experts - industry specialists.
- Regulatory authorities - laws & compliance rule.

Ex- For a hospital management system, Sources include doctors, nurses, patient & administration.

* Requirement Engineering process
The Requirement Engineering process is a systematic approach to gathering, analysing, documenting and managing requirements for a s/w system.

It ensure the final product meets user needs and buisness goals.

1. Requirement Elicitation (Gathering)
- Collecting requirement from stakeholders.
- Interview, questionnaries, observation, document analysis, brainstorming, prototyping.
→ Done by Client, end-user, managers, domain expert.
→ Goal for to identify what user really need.

2. Requirement Analysis & Negotiation.
- Studying gathered requirement to check feasibility, completness & consistency.
- Detect conflicts (eg. one want X and another Y).
- Goal to remove contradictions, resolve conflicts and prepare a refined requirement set.

3. Requirement Specification
- Writing requirement in a structured document.
- Main document is Software Requirement Specification.
- SRS Contain :- functional req, non-functional req., constraints (technical, legal etc.)
- Its Goal to provide clear blueprint for developers and testers.

4. Requirement validation
- Ensuring the documented requirement are correct and agreed upon.
- Reviews, ~~walkthrough~~ inspection, prototyping, test-case generation.

In this phase Question asked:-
- Are requirement complete?
- Are they consistent?
- Do they match user needs?

→ Goal to prevent errors early → cheaper to fix at this stage than later.

5. Requirement Management
- Handling changes to requirement during the project.
- Requirements often change due to:
  - new business needs.
  - Market Changes
  - user feedback.
- This phase goal is to keep requirement updated & ensure the system evolve with user needs.

→ Elicitation → Collect needs
→ Analysis & Negotiation → Resolve conflicts, set priorities.
→ Specification → write in SRS.
→ Validation → verify with stakeholders.
→ Management → Handle changes continuously.

This project start at the beginig of the project & continues until its end, because requirement can change over time.

Feasibility Study

Iska mtlb hai: Check karna ki project practical hai ya nhi. Matlab jo software ya System Client mang raha hai, kya usko ham bana sakte hain, afford kar sakte hain aur legally sahi hai ya nhi.

Types

1. Technical Feasibility

- Checks whether the required technology, tools, hardware & software are available.

* we have a question that do we have a technical capability to build it?

Ex- If a company wants an AI based Chatbot but has no AI experts then not technically feasible.

2. Economic Feasibility (Cost / Benefit Analysis)

- Check whether the project is financially affordable and whether benefits outweigh costs.

Question:- Is it good if we invest?

Ex- A Startup has a budget of 70000, but development cost is 800000 → not economically feasible.

3. Operational Feasibility
Check if the end-users will accept and effectively use the system.

Ex - If hospital staff cannot easily use a new hospital management system → operational feasibility fails.

4. legal Feasibility
- Ensure the system compiles with laws, rules, and regulation.
- Example:- Banking software must follow RBI or government security rules.

5. Schedule Feasibity (Time Feasibility)
- Check whether the project can be completed within the given time frame.

Ex - A project requires 12 months of work but the deadline is 6 months
- Schedule feasibility fails.

* Requirement modeling
process of representing system requirements using diagram, models, or structured formats.
Types:
1. Scenario Based Model (UML Diagram)
2. Data models. (DFD)
3. Flow-oriented models. (DFD)
4. Behavioral Models. (UML)
5. Class Model (object-oriented model) (UML)