

Week 2

L1 Req. Gathering & Analysis

- impt. of req. → deve. understand what customers want, an agreement is formed on req., inc. in cost & item^{no} if req. are unclear.
- 1^o user - freq. user of the system
- 2^o user - do not directly use the sys., use sys. thru an intermediary
- 3^o user - do not use at all. Affected by software, influenced by the purchase.
- analysis of req. is essential to identify ambiguities, inconsistencies, & incomp. req.

L2 Identifying Users & Requirements

- first, identify all 3 types of users, how to gather
- Questionnaires - series of ques. designed to elicit specific info. from users. Good for getting ans. to specific questions from a large grp. of people.
- Interviews - asking a set of ques., face-to-face 3 types - structured, unstruc., & semi-struc., helps to explore people's issues, elicits scenarios
- Focus Groups - Interviews gives 1 percepⁿ, get a grp. of stakeholders to discuss issues & req., gains consensus, highlights areas of conflict.
- Naturalistic Observatioⁿ - Spending time with stakeholders as they do daily tasks, observing their work in natural settings. Shadowing stakeholders, make notes, asks ques., observe.
- Documentation - procedures & rules for a task, steps involved in an activity, regularⁿ governing a task.
- data gathering is expensive, time-consuming. Be pragmatic, make compromises.

L3 Functional & Non-Functional Req.

- Func. → captures a functionality req. by the users from the sys. $f: I \rightarrow O$
- Non func. → specifies how the system should behave.
- ↳ Reliability - is the extent to which a program behaves the same way over time in the same operating envt.
- ↳ Robustness - to which a prog. can recover from errors or unexpected input.
- ↳ Performance, Portability, Security, etc

L4 Software Req. Specification

- Waterfall → req. gathering is done by system analyst, organize into SRS (topic) document.
 1. Broad outline & descripⁿ of the sys.
 2. func. & non-func. req.
- Adv. → forms an agreement b/w. customers & dev., reduces future reworks, provides a basis for estimating cost & schedules, facilitates future ext.
- Disadv. → lot of documentation. Good only if the req. are fixed. ⇒ Agile

L5 Behaviour Driven Design (BDD) - User Stories

- BDD → asks questions abt. the behaviour of the app. before & during dev., req. are conti. refined to meet user expectaⁿ, based on user stories.
- User stories - short, informal, plain lang. descripⁿ of what a user wants to do within a product which is of value to them, done in 1 sprint.

M	T	W	T	F	S	S
Page No.:						
Date:						YOUVA

- benefits - lighter, help plan & prioritize dev., focus on behaviour as "implementation" of the app, communication b/w users & dev. team.
- SMART story - specific, measurable, achievable, relevant & timely
- drawbacks - contd. contact with users not possible, not scale to large proj., safety critical applications