## Requirements Engineering

Activity starts with planning – identify goals, constraints, stakeholders; direction of project

Why it is hard?

* It is creative/problem solving
* Customer don’t know what they want
* Large system has different stakeholder needs and priorities
* Developer not sure how to build (like junior developer)
* Real requirement creep – scope of project always beyond boundaries and growing

Why important?

* Poor requirements capture = Big problem (=$$)
* Misunderstanding = Wrong ideas of need = Chaos (=$$)
  + = Software project failure (=$$$$$$$!

## Non and Functional Requirement (DEF)

FE define what the system should do from user perspective (not include how system should do and tech details), focus on actor and system, what system must support

NF define user restriction or limitation on system, not directly related with function of system. E.g. usability/ reliability/ security Safety

## Other Requirement

Business Requirements: what must be delivered or accomplished to provide value

Product Requirements: about product to accomplish business requirement (like multiple language)

Process Requirements: developing team must follow the constraints (standard, industry regulation)

## Activities of RE

### Elicitation: find requirement

Include: Process/ Essence/ Communication /Scope

Requirement vs specification

* Required mean non-negotiable, but almost every project change with negotiated re.
* Specifications about development iterative/evolving process
* Project scope: description of software purpose/built, prioritizing deliverables, time/resource available.

Challenges

* Customer don’t know what they want
* Communicate: different language
* Knowledge/cognitive limit
* Behavior issue: conflicting needs
* Tech problem: change management/too stick with methodology

Sources

* Goal
  + Motivate but vaguely formulated
  + SE need focus on value and cost of goals
  + A feasibility study is a low-cost way of doing this
* Domain knowledge
  + Industry-specific concepts
  + Business process
  + Regulatory and compliance requirements
* Stakeholder and business rule
  + Define/constrain aspect of structure/behavior of business itself
    - E.g: student cannot register in next semester’s courses if there remain some unpaid tuition fees
* Operational environment
  + Requirement goes up from environment when software running
    - timing constraints in real-time software or performance constraints in a business environment
* Organisational environment
  + Software support business process and these processes constrain by structure/policy/culture of company

Methods

* Interviews
  + Collect information from person.
  + Success identify:
    - Workflow
    - Factors influence operation system
    - Element that make up system (documents, procedures, policies)
  + Steps: prepare, planning & scheduling, opening & closing, conducting, Following up
  + Facilitated meeting: summative effect, more insight to SR
    - Ad
      * Brainstorm and refine difficult to bring to surface.
      * Conflicting requirement founded
      * More consistent set of requirement
    - DS
      * Handle carefully/avoid poor group dynamic
      * Time consuming

Prototypes

* Tool for clarify requirement
* Provide user context, better understanding
* Mock up design, beta version of software product
* Low fidelity used, avoid stakeholder limit design or incidental character
  + With the high-fidelity prototype, stakeholders might get caught up in minor details like the color scheme, font choices, or placement of buttons. They might fixate on these elements and form strong opinions, which could limit the design team's flexibility to explore alternative options.
* Dis: cause implementation too early, rough prototype become product

Scenarios and Use Case

* Used In plan methodologies
* Provide content to elicitation of user requirement
* For framework for question about ‘what if’/ ‘how is this done’

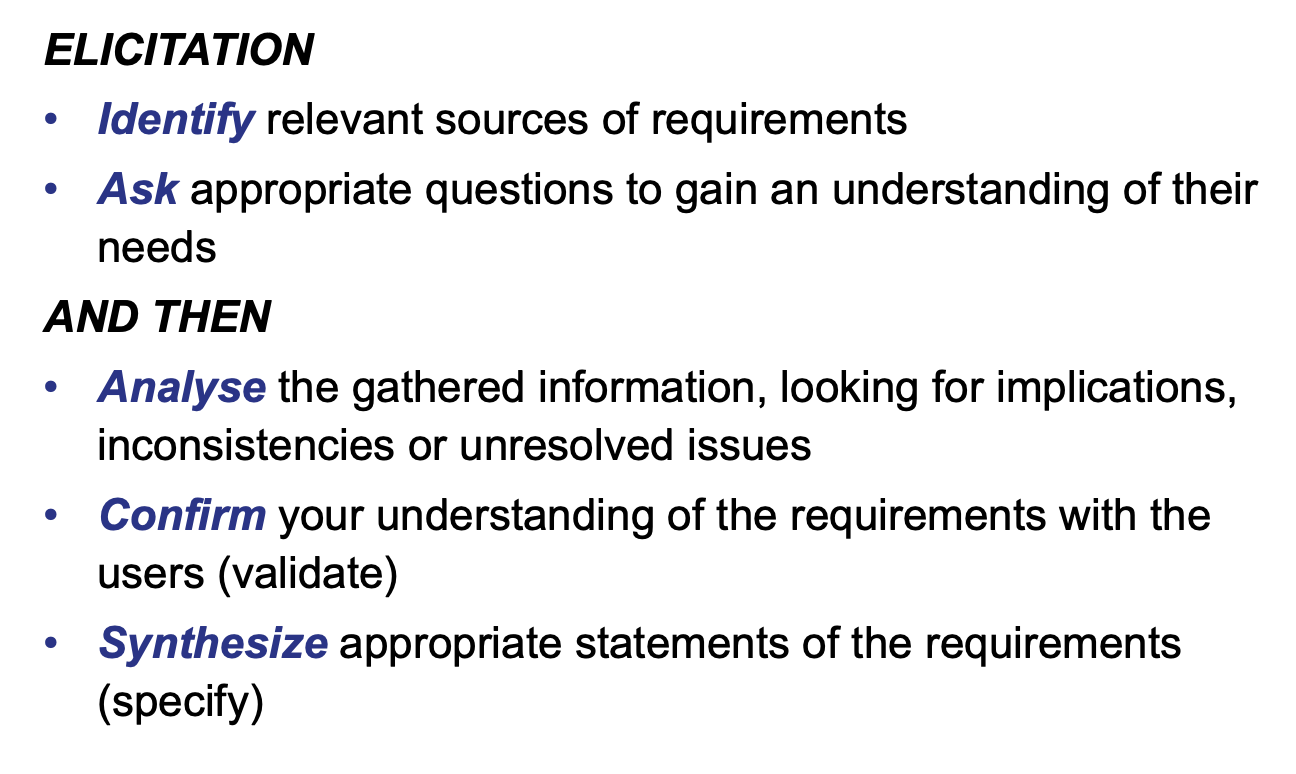
Observation

* Observe worker actual tasks
* Help reflect actual rather than formal process
* ADV: find many tasks and processes that too subtle/complex for actor
* DIS: expensive ()
  + Observer should be outsider, not final user, not suitable discover organizational/domain requirement

### Analysis: ensure requirement correct, complete and unambiguous

### Specification: document them

### Validation: check system fit client needs



Summary:

Elicitation Guidelines

* Sensitive to organisational/political
* Identify stakeholders
* Record requirement sources
* Define operate environment
* Use business concern to drive requirement elicitation
* Looking for Domain constraints
* Scenarios: elicit requirement
* Operational processes
* Reuse requirement