
REQUIREMENTS
ELICITATION

# Challenges of Requirements Elicitation

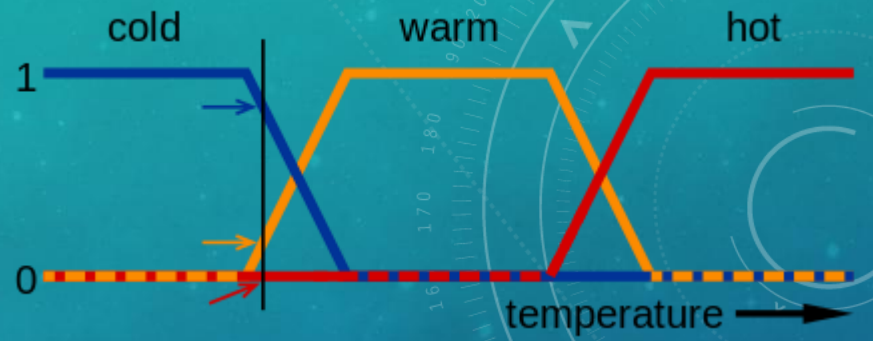
# REQUIREMENT ELICITATION

Elicit means evoke or draw out answer (or fact) from someone.
Collect means gather things together.

**Concept**

- It is the process of identifying the needs and constraints of the various stakeholders for a software system by proactively taking actions
  - Not the same as "collecting requirements."
  - Not a simple matter of transcribing exactly what users say.
- It is a collaborative and analytical process that includes activities to activities to draw out , discover, extract, and define requirements
- Collaborative
  - engage end users
  - use the vocabulary of the business domain instead of forcing customers to understand technical jargon (rather than assuming that all participants share the same definitions)
  - "walk through" the processes with users to make decisions about their work, and extract the underlying logic
  - Try to understand something about clients
  - Design questions to draw out responses

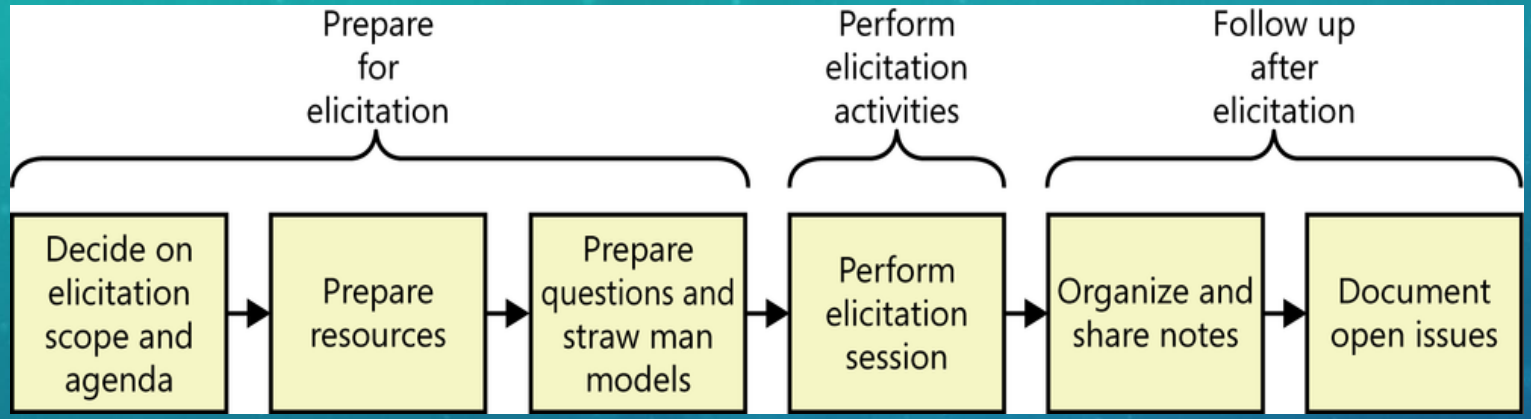
# REQUIREMENT ELICITATION




- Analytical
  - Reasons for analysing or processing
    - What they say may not be what they mean
    - Ambiguity (one word with multiple meanings) and vagueness (not clearly defined)
    - Inconsistency
    - incompleteness
    - Conflicting
  - Gather together ”data” (words/phrases/sentences/business stories/etc.)
  - Discover hidden or missed data
  - Extract information from data collected
  - Define requirement as the process of turning information to knowledge


Exercise: The device tester shall allow the user to easily connect addition components, including a pulse generator, a voltmeter a capacitance metre and custom probe cards. This is a bad description of requirement because of ambiguous words “easily” and “including”. How can we make it a good description?

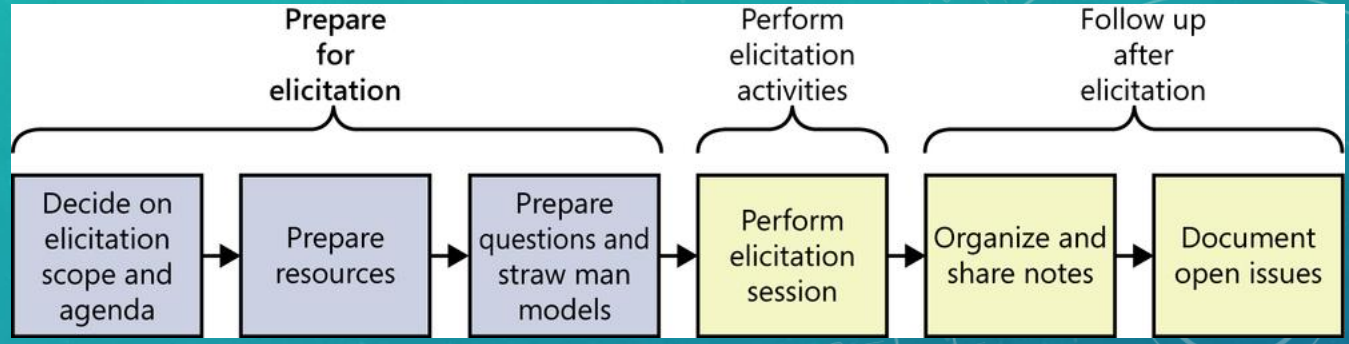
# ELICITATION PROCESS




**"Make your mind" before starting**

- Objectives and expected outcomes (what do you want to achieve?)
- Resources (people, documents, systems, etc.) (what are constraints?)
- Timeline (when to do?)
- Techniques/methods (same or different techniques for each stakeholder group, single or combined techniques) (how to do?)

# PREPARING ELICITATION




**Preparation**

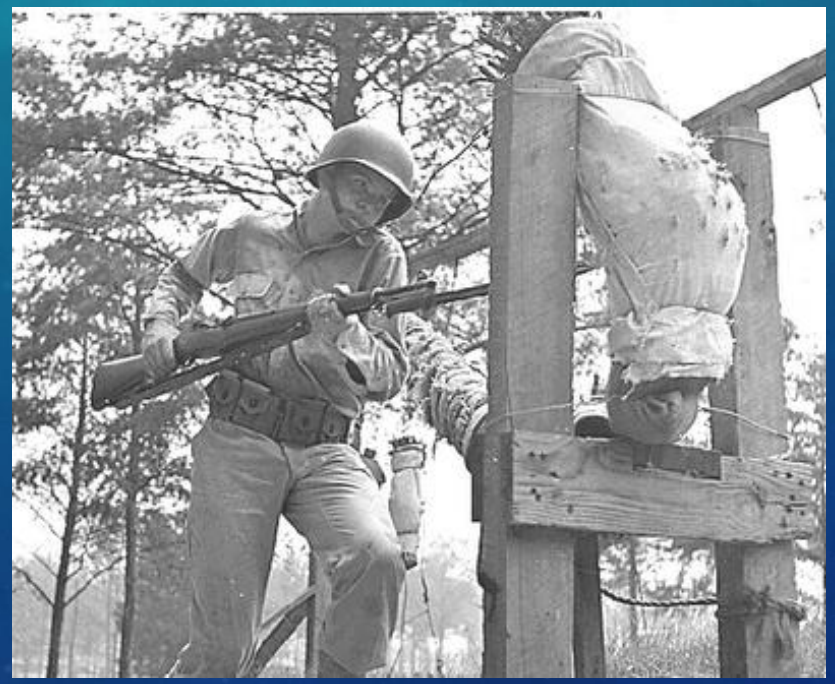
- "Sessions" – you work on a session-basis if you divide stakeholders into several groups and elicit requirements for a group at a time.
- Learn about stakeholders to define groups and hence sessions – Have a background study on all relevant stakeholders, including culture, geographic location, role, and all tiny details and classify them into groups
- Session scope and agenda – decide step by step what to do according to information expected to have and time
- Prepare resources – physical resources such as room, computer, etc. and documents needed, as well as training (yourself) for online tools
- Prepare questions (related to objectives and outcomes)
  - Imagine yourself learning the user's job, or actually do the job under the user's supervision. What tasks would you perform? What questions would you have?
  - Straw man models – brainstorming-like approach to discussing any deficiencies that may lead to problems., "What else could …," "What happens when …," "Would you ever need to …" (You draft a proposal for others to criticise and then you develop a better one).

# PREPARING ELICITATION

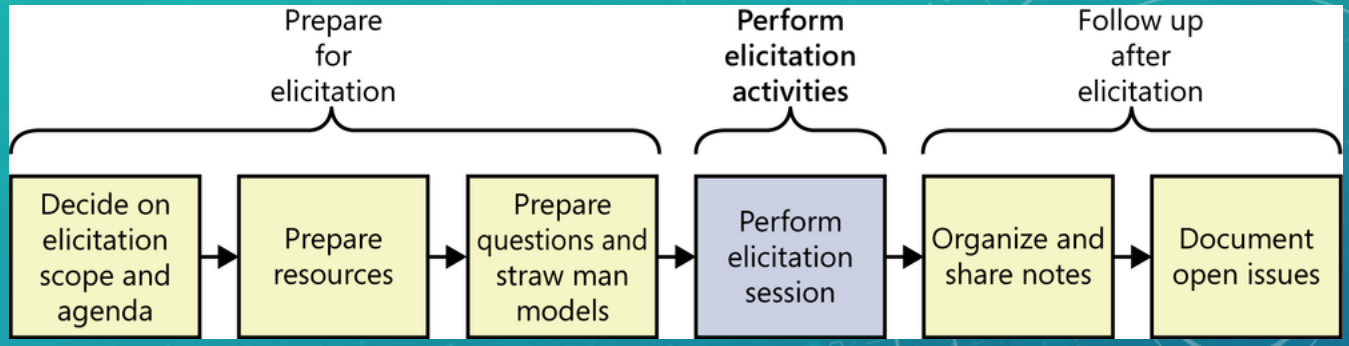
Straw-man approach is actually similar to “抛砖引玉” in Chinese。

Straw-man proposal

- You create an imperfect proposal
- Everyone criticises (or questions or gives constructive comments on) it (so your proposal is a strawman, i.e. a target so the rest to attack)
- You then collect all criticism and build up a better proposal (for being criticised/attacked further)



# PERFORM ELICITATION




**Perform**

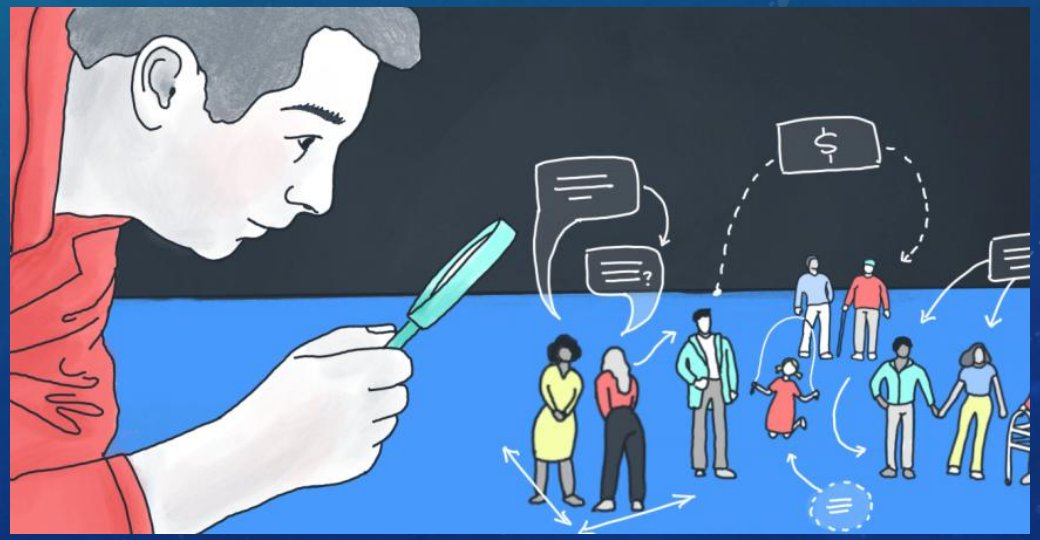
- Elicitation techniques/methods
- Interview – the most commonly used tech for collection and discovering
  - Establish rapport – a close and harmonious relationship in which the people or groups concerned understand each other's feelings or ideas and communicate well
  - Stay in scope – keep questions focusing on objectives
  - Prepare questions beforehand
  - Iterative rather than linear (some questions may not be easy to answer)
  - Suggest ideas – suggest interviewees alternative ways to represent themselves if the BA identifies uncertainties in their sentences
  - Listen actively – find uncertainties for further questions to clarification

# PERFORM ELICITATION

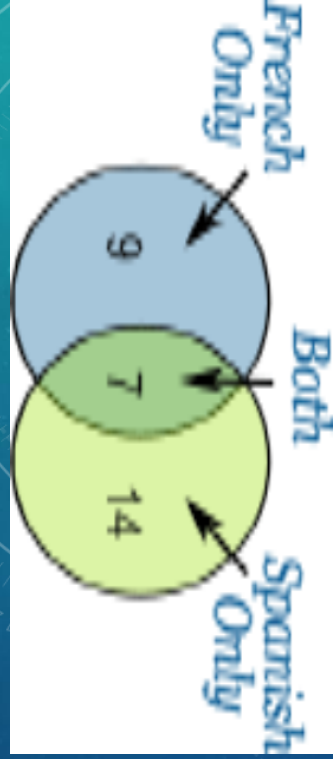
- Workshops encourage stakeholder collaboration in defining requirements
  - Establish and enforce ground rules – starting/ending time, returning from breaks, switch off mobile phones, hold one conversation at a time, etc.
  - Fill all the team roles – note taking, time keeping, ground rules implementing, all being heard, scope managing, etc.
  - Plan an agenda – who leads what and when
  - Stay in scope – keep right level of abstraction to avoid spending too much time and efforts in details. Referring business requirements to avoid project creed.
  - Use flipchart to capture random information for later consideration
  - Timebox discussion – allocate and keep time for each topic
  - Keep the team small but include the right stakeholders
  - Keep everyone engage (organisation skills)

# PERFORM ELICITATION

- Focus groups -- A focus group is a representative group of users who generates input and ideas on a product's functional and quality requirements. Focus groups are useful for exploring users' attitudes, impressions, preferences, and needs
  - Deal with conflicts
  - select the focus group members carefully
  - keep them on topic, but without influencing the opinions being expressed
- Observations (when users confused themselves, less cooperative, not be able to represent clearly, ... )
  - Pros: allows the BA to
    - validate information collected from other sources
    - identify new topics for interviews
    - see problems with the current system
    - identify ways that the new system can better support the workflow
  - Cons: time consuming



# PERFORM ELICITATION




- Questionnaires are a way to survey large groups of users to understand their needs
- Tips of well-written questions
  - Provide answer options that cover the full set of possible responses.
  - Make answer choices both mutually exclusive (no overlaps in numerical ranges) and exhaustive (list all possible choices and/or have a write-in spot for a choice you didn't think of).
  - Don't phrase a question in a way that implies a "correct" answer.
  - If you use scales, use them consistently throughout the questionnaire.
  - Use closed questions with two or more specific choices if you want to use the questionnaire results for statistical analysis. (Also save users' time.) Open-ended questions allows users to respond any way they want, so it's hard to look for commonalities in the results.
  - Consider consulting with an expert in questionnaire design and administration to ensure that you ask the right questions of the right people.
  - Always test a questionnaire before distributing it. It's frustrating to discover too late that a question was phrased ambiguously or to realize that an important question was omitted.
  - Don't ask too many questions or people won't respond.

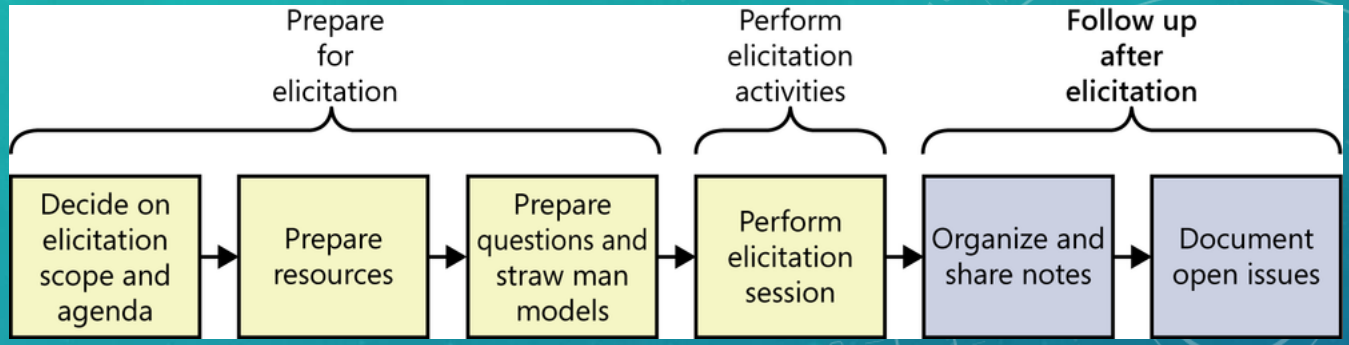
# PERFORM ELICITATION

- Legacy system analysis
  - System interface analysis -- revealing functional requirements regarding the exchange of data and services between systems
  - User interface analysis -- study existing systems to discover user and functional requirements
  - Document analysis -- examining any existing documentation for potential software requirements
- Techniques by project characteristic

  (See the diagram)
- Some tips
  - Educate stakeholders
  - Take good note
  - Exploit the physical space

| | Interviews | Workshops | Focus groups | Observations | Questionnaires | System interface analysis | User interface analysis | Document analysis |
|---|---|---|---|---|---|---|---|---|
| Mass-market software | x | | x | | x | | | |
| Internal corporate software | x | x | x | x | | x | | x |
| Replacing existing system | x | x | | x | | x | x | x |
| Enhancing existing system | x | x | | | | x | x | x |
| New application | x | x | | | | x | | |
| Packaged software implementation | x | x | | x | | x | | x |
| Embedded systems | x | x | | | | x | | x |
| Geographically distributed stakeholders | x | x | | | x | | | |

# FOLLOW UP ACTIVITIES



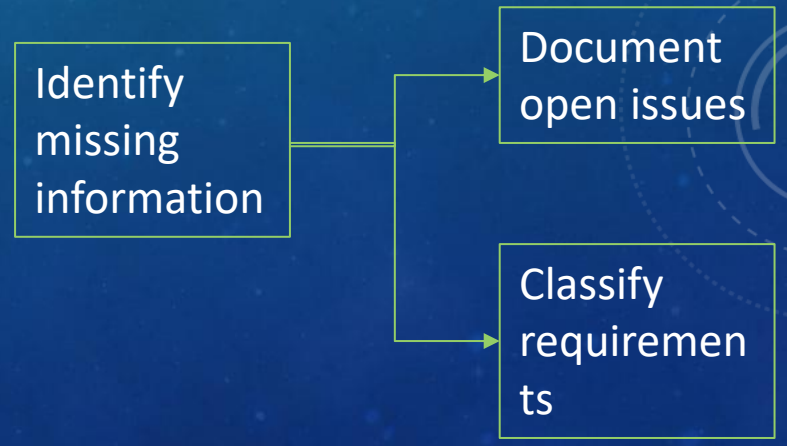

**Follow up**

- First block aims to identify missing information.
  - Organising notes – to find out any missed information because you did not write down (during interview, for example, due to lack of time), according to your memory, also to classify data/information and arrange them in logical ways
  - Sharing notes – to interviewees to confirm what you noted is what you said
- Second aims to decide any area you need to further explore, i.e. to organise further events to clarify and to know more.
- NOTE: the key is to keep the true stories as customers said, so do not edit the notes according your understanding. If you want to edit, keep the original notes in a “safe” place as you may need to refer to it later.

# CLASSIFICATION OF USERS' POINTS

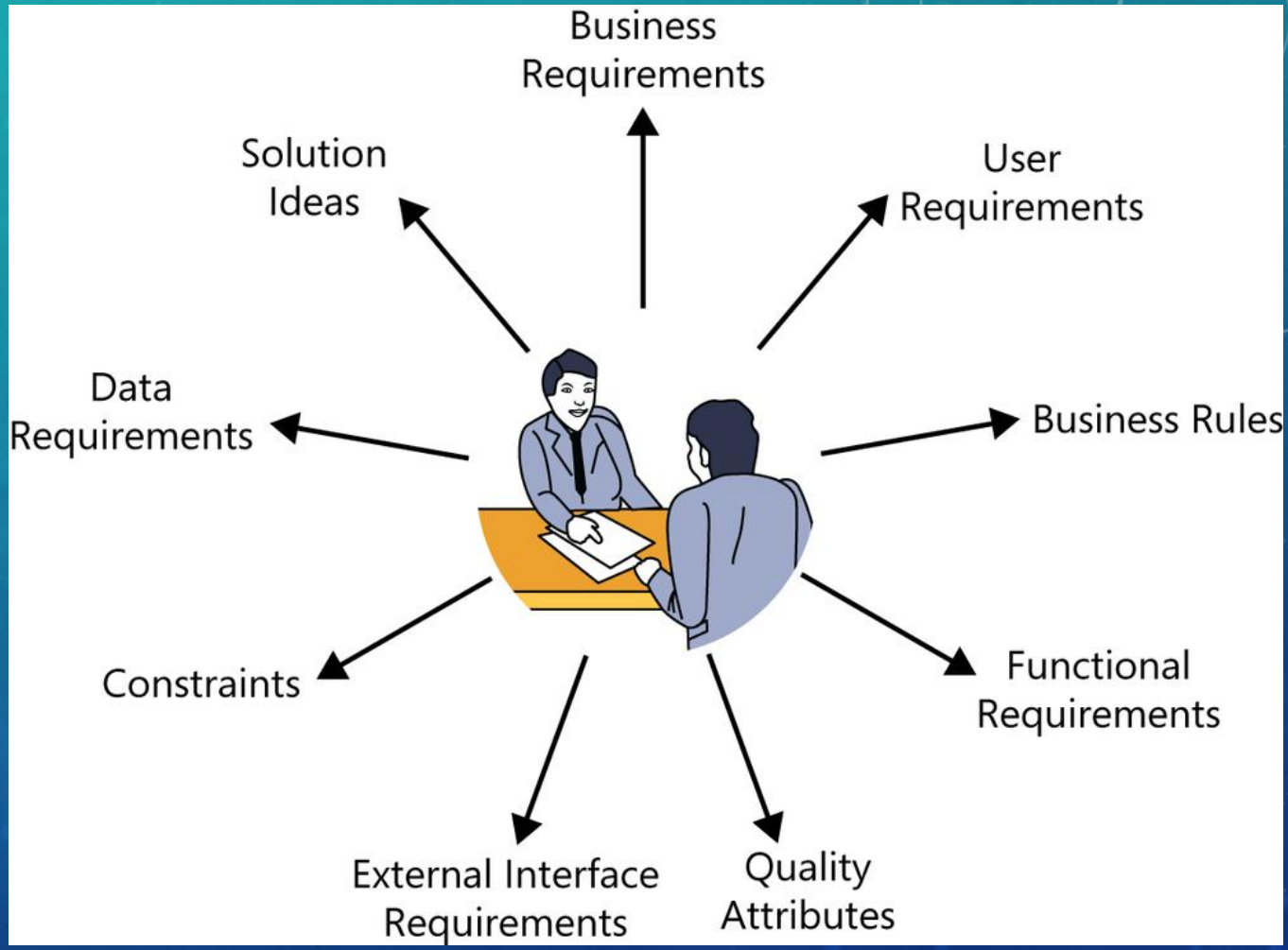
**Classifying customers' input**

- Filter out the followings first:
  - A project requirement that is not related to the software development, such as the need to train users on the new system.
  - A project constraint, such as a cost or schedule restriction (as opposed to the design or implementation constraints).
  - An assumption or a dependency.
  - Additional information of a historical, context-setting, or descriptive nature.
  - Extraneous information that does not add value.



# CLASSIFICATION OF USERS' POINTS

- Then classify the rest into the following nine categories:
  - Business requirements Anything that describes the financial, marketplace, or other business benefit that either customers or the developing organization wish to gain from the product is a business requirement
    - Increase market share in region X by Y percent within Z months.
    - Save X per year on electricity now wasted by inefficient units.



# CLASSIFICATION OF USERS' POINTS

- User requirements are the user goals or business tasks that users need to perform, most typically represented as use cases, scenarios, or user stories
  - I need to print a mailing label for a package.
  - As the lead machine operator, I need to calibrate the pump controller first thing every morning.
  - (Remember unambiguity, completeness, level of details)
- Business rules describe only activities only certain users can perform
  - A unconditional offer requires the applicant to present ATAS and to pay 60% of his tuition fee in advance.
  - Time-off approvals must comply with the company's HR vacation policy.
- Functional requirements describe the observable behaviours the system will exhibit under certain conditions and the actions the system will let users take
  - If anyone entered the laser lab, the laser on warning light should come on.
  - The user must be able to sort the project list in forward and reverse alphabetical order.

# CLASSIFICATION OF USERS' POINTS

- Quality attributes describe how well the system does something
  - The mobile software must respond quickly to touch commands.
  - The shopping cart mechanism has to be simple to use so my new customers don't abandon the purchase.
- External interface requirements describe the connections between your system and the rest of the universe.
  - The manufacturing execution system must control the wafer sorter.
  - The mobile app should send the check image to the bank after I photograph the check I'm depositing.
- Constraints legitimately restrict the options available to the developer
- Data requirements Customers are presenting a data requirement whenever they describe the format, data type, allowed values, or default value for a data element; the composition of a complex business data structure; or a report to be generated
  - Date is presented with the format of dd/mm/yyyy.
  - An order consists of the customer's identity, shipping information, and one or more products, each of which includes the product number, number of units, unit price, and total price

# CLASSIFICATION OF USERS' POINTS

- Solution ideas: Many "requirements" from users are really solution ideas. Someone who describes a specific way to interact with the system to perform some action is suggesting a solution. Examples:
  - Then I select the drop-off station where I want to have my order received from a drop-down list.
  - The smart phone has to allow the user to swipe with a finger to navigate between screens.

# COMPLETION

```
do{
…
…
}
while (…)
```

- When one or more of the followings are reached
  - The users can’t think of any more use cases or user stories. Users tend to identify user requirements in sequence of decreasing importance.
  - Users propose new scenarios, but they don’t lead to any new functional requirements. A “new” use case might really be an alternative flow for a use case you’ve already captured.
  - Users repeat issues they already covered in previous discussions.
  - Suggested new features, user requirements, or functional requirements are all deemed to be out of scope.
  - Proposed new requirements are all low priority.
  - The users are proposing capabilities that might be included “sometime in the lifetime of the product” rather than “in the specific product we’re talking about right now.”
  - Developers and testers who review the requirements for an area raise few questions.