Stakeholder analysis
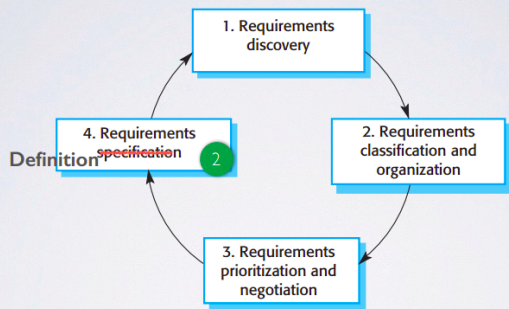
Determine **all the people** that will use the system. A *stakeholder* in an organisation is any group or individual who can affect or is affected by the achievement of the organisation's objectives.

- Work with initial briefs
- Problem analysis
- Interviews discussions

Must identify

- Primary stakeholder
- Secondary stakeholder
- Tertiary Stakeholder

There are a lot of ways to classify/ analyse stakeholders.

- Importance/ Priority
- Impact of their needs
- You can list example people as well as what you want from them

Personas

Personas are fictional characters created to represent the different user types that might use a site, brand, or product in a similar way. Persona should clearly differentiate a stakeholder. **Identify** motivations, expectations, goals, knowledge. **Usage** : Help you to put yourself in the shoes of.. . Try to have a small number (one page at most). Persona is not a report on real people from the client company.

- Choose a generic representative name/photo/etc
- Make 2 or 3 that demonstrate **key user types**
- It might be that several stakeholders can be represented by one persona.

Benefits

- By thinking about the needs of a fictional persona, designers may be better able to infer what a real person might need.
- Prevents "The Elastic User" design(while making product decisions different stakeholders may define the 'user' according to their convenience)
- Defining personas helps the team have a shared understanding of the real users in terms of their goals, capabilities, and contexts
- Personas also help prevent "self-referential design" (when developer unconsciously project their own mental models on the product design)
- Provide reality check
- Prioritization of requirements
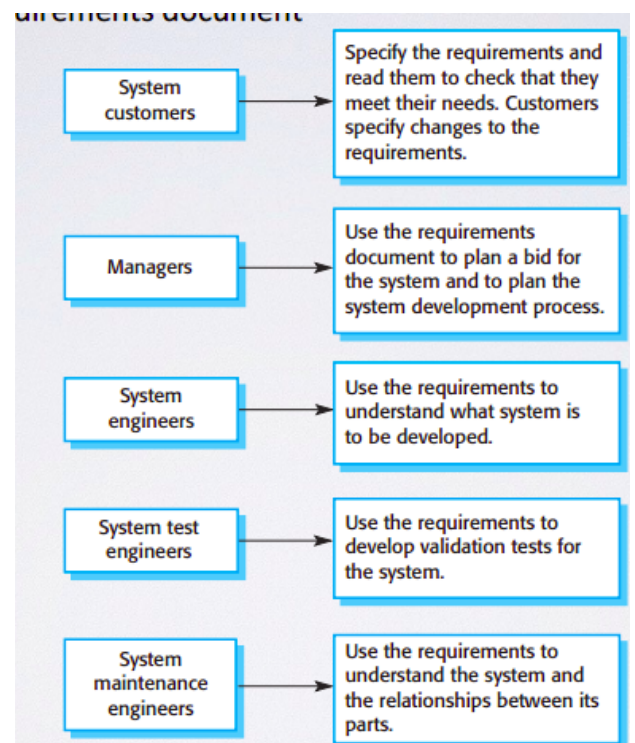
User case diagrams

Need to elaborate the tasks that each Stakeholder will do. The most common method for this is a **use case** , it represents the **people** who use the system **and the tasks** they have to perform. Can be referred as 'actors'.
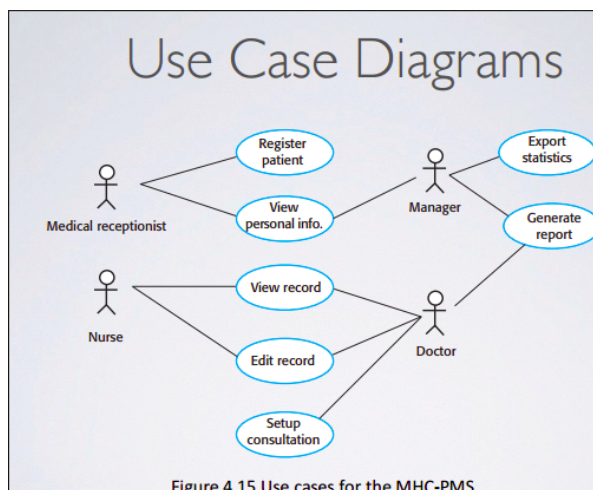

Figure 4.15 Use cases for the MHC-PMS

Not just tasks that a user does, but a set of relationships between people and task (may be directional). User case model both the **people** and the **artefacts, tools or resources** involved in a task's accomplishment.

Include Extend Generalise

Extend is used when a use case adds steps to another first class use case.

Include is used to extract use case fragments that are duplicated in multiple use cases. The included use case cannot stand alone and the original use case is not complete without the included one. This should be used sparingly an only in cases where the duplication is significant and exists by design (rather than by coincidence).

For example, the flow of events that occurs at the beginning of every ATM use case (when the user puts in their ATM card, enters their PIN, and is shown the main menu) would be a good candidate for an include.
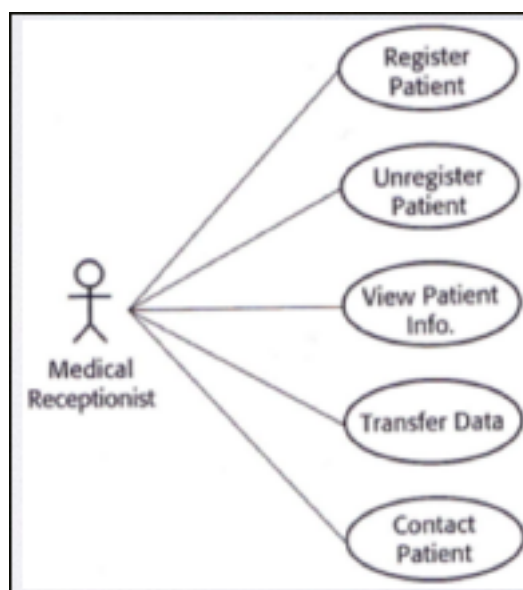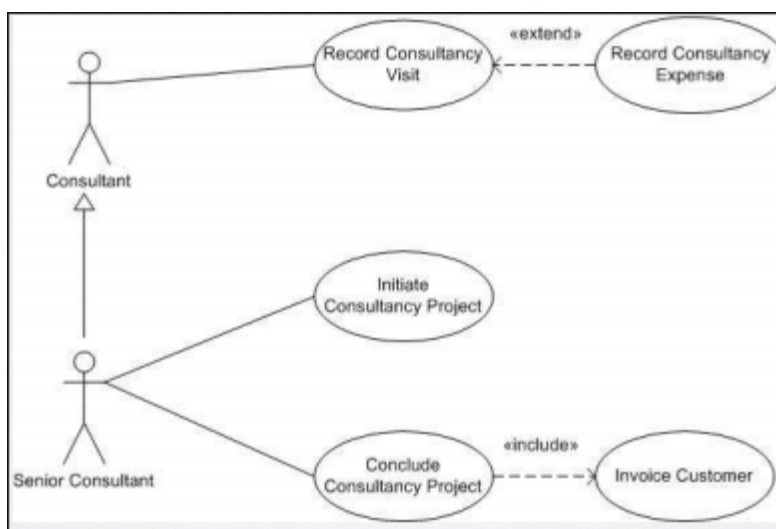
User stories

A One sentence statement of a clear requirement for a user. Should include
- *a role*
- *a goal/desire/action*
- *an effect/outcome*

It is designed to be a lightweight form of user requirements. It is much **quicker** to create and **easier** to mark as done.
- **Pros** : Concise and clear, very little maintenance, clear requirements, breaks down project into simple task, can rank importance.
- **Cons** Difficult to use in BIG projects, loose detail and formality. Doesn't describe **process, tasks, context**

Investigation methods

- Surveys
- Interviews & Focus groups
- Observations
- Others

Thing that you might want to investigate

- All types of people that will use the system
- All tasks people need to achieve, and why
- What they use to do the task, and what happens after
- All the existing technology they have to use
- All the other technology they have to use

Investigation techniques

You have to be asking the right questions as well as follow-up ones. Might want to start asking why because the aim is not to find out what, but to understand why, how when, where.

Identify your investigation goals

Should be easier if you've done personas,use cases. You always need to have one or more **goals** of what you want to find out.

Why are you speaking with this stakeholder. A **goal** helps you frame questions : helps to keep direction and focus; identifies the **main topics** you will discuss

Plan a set of optimal questions from the goal

For each **topic** think of **all the different questions** you could ask, or the ideas you have, this way you can decide what you want to know easier. Then use this to select **good questions**.

Your aim is to get good deep/rich coverage of the questions and **understand as much as possible**. Having clear investigation techniques helps you to develop a strategy by being able to choose one that will maximise your discovery

Surveys

Good for :
- Contracting lots of people
- Finding majority's opinion
- Seeing , how different stakeholders answer same questions

Bad for :
- Actually discovering/understanding specific topic.

Presumes that you already understand the problem , know the right questions to ask, and that the questions make it possible to answer them. Need not to forget, that you **can't ask follow-up** questions. **A common mistake**: choose surveys 'because they are easy' ,put out some poorly planned questions, and not test-run them first.

Open ended questions can often get 1-liners in them. Need to structure questions to avoid them : Provide topics/criteria to answer questions to guide a longer response.

Closed Questions

- Ordered response – likert-scale 1-7 estimation
  - Allow people to mark the amount of certain feeling
  - Can get average responses
- Unordered multi-choice responses
  - Tick things that apply
  - Can count totals for different answers

Common problems

- Questions with 2 parts
- Negation within the question
- Leading questions
- Hot highly-debatable questions, or opinions in answers can draw unnecessary focus
- Over-asking and lengthy surveys

Interviews

Gives freedom to ask follow up questions
- Structured (when studying 'variation' to a 'constant'
- Semi-structured
  - Most common
  - Allows for pre-determined topics and follow-ups
- Exploratory (Often pre-study early-researched)
- Get people to **do things** draw things out,show you around, do a task in front of you,etc. -**not just talk theoretically**

Effective interviews

1. They are open-minded, avoid pre-conceived ideas about the requirements, and are willing to listen
2. They prompt the interviewee to get discussion going using a springboard question, a requirements proposal, or by working together on a prototype system. Just asking people what they want is unlikely to result in useful information.

Focus groups

Pros:

- More people at once
- Faster coverage
- Discuss differences and opposing views

Cons

- Possible conflict
- Have to avoid a dominant speaker
- Takes longer per question
  8-12 people for breadth, 5-7 for dept, **must** do more than 1-2.Ideally 1 leader + 1 note taker

Good interviews are not random "off the cuff", they come with a plan from your goals. You have to prepare an interview plan (at least around 15 topics that you want to cover). Must not forget that a plan is a **to do list** not a script, they might answer the questions before you ask them

Observations

You learn so much more by seeing rather than just talking.
- People are good at telling you what and what they think
- But they are are seeing their perspective, as one user type
- They only have the "tip" of the problem, not a view of solution
- They are not good at viewing themselves
- People are not good at realising everything that is important to tell you

Problems :
- People behave differently when being observed
- Behave in desired/optimal way not normal way

More advanced techniques

- Contextual inquiry
- Ethnography
- Technology Tours

Contextual inquiry

- Mixed method of observation & interviewing
- One on one interviews in **the context of the activity**
- Focused on what people do on a day-to-day basis
- Focused on **process** and **people**
- Asking people to explain their work step-by-step

Technology tours

Helps to understand the possibilities for designing technologies for the home. A "contextually grounded" approach. Understand the possibilities for design in context, like interviewing with objects, but focus in tech context.

Key questions :

1. What technology is present in each room
2. Where is it placed
3. Who uses the technology
4. What activities It supports

Tours should focus on things that 'could be technology' in the process. So if a whiteboard is used to coordinate agile programmers : you should be finding out about that too; how is it related to the process; how could you do this in your software.

Ethnography

There is no better way to learn than :

- Observing an environment
- asking about what's happening
- **getting yourself involved**

Traditionally it involves 'becoming part of the culture' for extended periods of time, but this is typically **unrealistic** for software projects. **But** you still learn most by : getting yourself involved even for a little while.(rapid ethnography is better than not getting involved)

Focused Ethnography

- Not just generally getting involved with the general activities
- Specifically having a go at key activities as identified in interviews etc.
- Need to ask people to reconstruct a normal process, step by step-by-step
- **Always** record the activities somehow

Requirements discovery

Is a continual process. You might

- Discover more users ('actors' in User Case Diagrams)
- Refine/improve personas
- Discover new use cases
- Write additional user stories