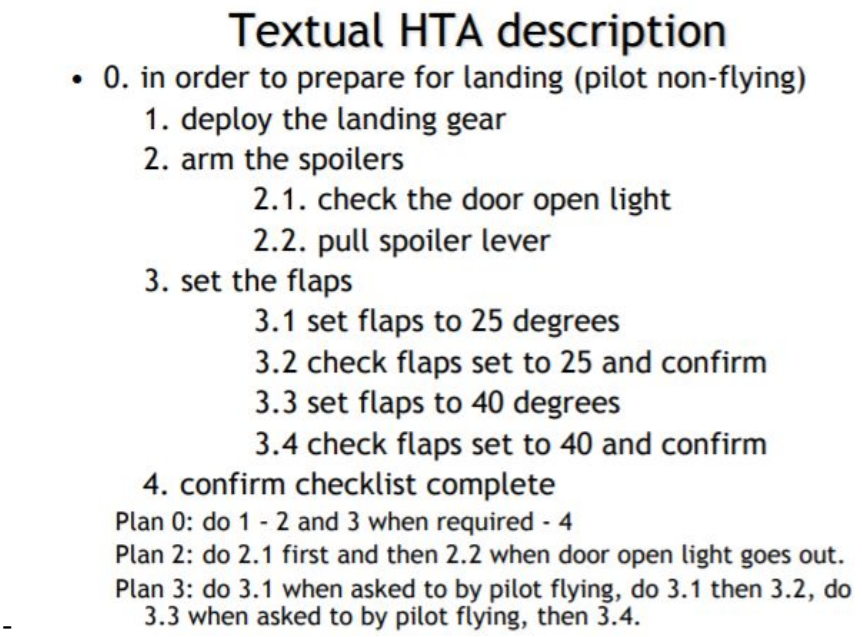
Requirements Analysis

- **Requirements**
  - Understand the users, task, and context
  - Produce a set of requirements through: data gathering, data analysis, and expression as ‘requirements’
  - **Iterative**
  - **Establishing Requirements**
    - What do users want? What do they need?
    - Requirements need clarification, refinement, completion, re-scoping
    - Requirements arise from understanding users’ needs, can be justified and related to data
  - **Types of Requirements**
    - **Functional**
      - What should it do, what data needs to be stored, how will it store it
    - **Non-functional**
      - How well will the system do it, memory size, response time, usability
    - **Environment/context of use:** physical, social, organisational
  - Who are the users?
    - Characteristics (technical competency), System use (frequency)
      - Novice: prompted interactions, step by step
      - Casual/infrequent: clear instructions (menus)
    - **Personas**
      - Capture user characteristics - not real people, but inspired by real users’ characteristics
      - Shouldn’t be idealistic, bring to life with context, name, background
      - Develop multiple personas
      - ADV: goals and needs of user becomes common point of focus, concentrate on manageable set of personas knowing it captures the needs of many, quick to develop
- **Data Gathering**
  - **Interviews**
    - Props - e.g. prototypes, or sample scenarios can be used
    - Good for exploring issues, but time consuming
  - **Focus Groups**
    - Group interviews
    - Good at gaining consensus view, but can be dominated by individuals
  - **Questionnaires**
    - Often used with other techniques, gives quantitative or qualitative data
    - Good for answering specific questions in large groups
  - **Researching similar products**
    - Good for prompting requirements
  - **Direct Observation**
    - Gain insight into stakeholders’ tasks

        - Good for gaining understanding (esp. context), but requires time and commitment, can result in huge amounts of data
    - **Indirect observation**
        - Not often used
        - Good for logging current tasks
    - **Studying Documentation**
        - Good source of data about steps involved, not to be used in isolation but provides key words for interviewing
        - Good for understanding legislation, but doesn't involve actual stakeholders
        - **Note on stakeholders - 'real stakeholders':** need real users input, not managers input
    - **Problems in data gathering:**
        - Domain knowledge is hard to gather - sometimes hard for people to articulate (how do you walk)
        - Political issues with organisation, balancing functional & usability demands
    - **Guidelines**
        - Focus on needs, involve all stakeholders, more than one representative from each group, multiple data gathering techniques
        - Use props and a pilot session, know what data is vital, consider how to record data
- **Task Descriptions**
    - **Scenarios**
        - Informal narrative story, simple, natural, not generalisable
        - Focus on situation/detailed context
        - **Includes: actors, background info, assumptions about environment, goals, sequences of actions**
        - **Linear narrative, easy to understand but doesn't capture edge cases - need multiple**
        - Explore: whats their goal, what can they see, what they do, what they think, what happens when, what is happening inside the system
    - **Use Cases**
        - Assume interaction with a system, assume detailed understanding of the interaction
    - **Essential Use Cases**
        - Abstract from details, do not make the same assumptions
    - **Task Models**
        - Detailed breakdown of the steps involved, avoid committing to implementation details
- **Task Analysis**
    - Goal, motivation, method - (what,why,how)
    - Many approaches to this
    - **Task Decomposition:** split into ordered subtasks
    - **Knowledge based techniques:** what they know, how it is organised

- **General Method:** observe, collect list of words/actions, organise using notation or diagrams
- **Hierarchical Task Analysis (HTA)**
    - Break task into sub-task, sub-sub task etc. and group as plans which specify how tasks may be performed in practice
    - Focuses on physical/observable actions - starts with users goal which is examined and the main tasks are identified
    - **Task decomposition**
        - Describe the actions, structure them in a subtask hierarchy, describe the order of subtasks
    - **HTA - text and diagrams show hierarchy, plans describe order**
    - 

    - **Tasks and goals**
        - Users have goal directed units of activity & tasks are complex
        - To describe tasks: generate list of tasks, group into higher level tasks, decompose lowest level until simple enough
    - **Improving a HTA**
        - Paired actions: if something is turned off, did we turn it on
        - Restructure: add new higher level task grouping
        - Balance: could we simplify it
        - Generalise: could we apply this to more scenarios
        - **HTA flow control**
            - Fixed sequence (x then y then z), optional tasks (if x then y), waiting for events, cycles (while x do y), time-sharing (parallel), discretionary (optional)
    - **What feedback is given to the user to support the most complex task?**
- **Knowledge Based Analyses**
    - To capture knowledge needed to perform a task
    - Focus on: objects used in task, and the actions performed
    - Assess amount of common knowledge between tasks

        - Create Taxonomies for grouping, identify common issues throughout the taxonomies
        - **Expert Knowledge**
            - Expertise for diagnosing & predicting - situational awareness, perceptual skills, developing and knowing when to apply tricks of the trade, improvising, recognizing anomalies, compensating for equipment limitations
        - **Affordances**
            - Refers to the properties of objects - what sorts of operations and manipulations can be done to a particular object
            - E.g. icons ‘afford’ clicking
            - Important in GUI design: **perceived affordance**
- **Uses of task analysis**
    - Lift focus from system to use, suggest candidates for automation, uncover the mental model of users
    - Taxonomies used to suggest layout, action list suggests interface objects
    - Task analysis is never ‘completed’ - iterative
- **Summary**
    - Requirements are crucial
    - Different types of requirements, each is significant for interaction design
    - Most common techniques: questionnaires, interviews, focus groups, direct observation, studying docs, researching similar products
    - Personas can be used to capture users’ needs and preferences
    - Scenarios, use cases, and existential use cases can be used to articulate existing and envisioned practices
    - Task analysis such as HTA help investigate existing systems/practices