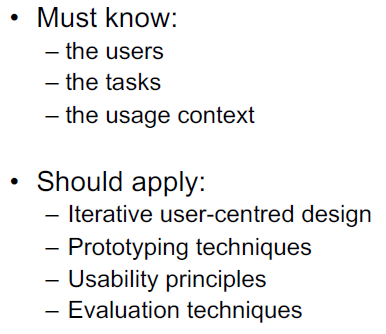
Interação Pessoa-Máquina

### Human-Computer Interaction (HCI)

A HCI envolve o projeto, implementação e avaliação de sistemas interativos no contexto da tarefa do usuário.



### Usabitily

Eficácia, eficiência e satisfação com que usuários especificados podem atingir metas especificadas em um determinado ambiente.

A usabilidade é medida em relação a determinados usuários (selecionado para ser o mais representativo possível dos usuários pretendidos) e determinadas tarefas.

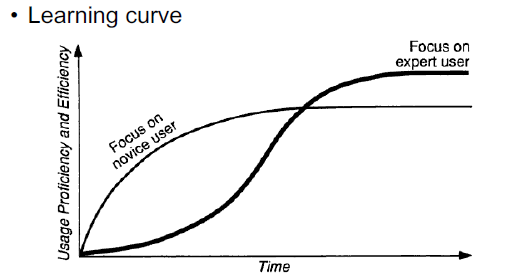
**Especificações:**

* **funcionais:** crucial para garantir a funcionalidade do sistema
* **usabilidade:** crucial para garantir a usabilidade do sistema

## Learnability

* facilidade com a qual novos usuários podem iniciar uma interação efetiva e alcançar o máximo desempenho.

**Testar fazendo:** Escolha alguns usuários que não usaram o sistema antes e meça o tempo que eles levam para atingir um nível especificado de proficiência em usá-lo.



## Efficiency

* uma vez que o usuário aprendeu a usar o sistema, um alto nível de produtividade deve ser possível.

Os usuários são considerados experientes (expert users):

* + - se eles dizem isso
    - se forem usuários por mais de um determinado período de tempo.

Meça continuamente o desempenho do usuário (ex: em termos de número de segundos para realizar uma tarefa específica) até que ele pare de aumentar, quando se considera que o usuário atingiu o nível de desempenho estável.

A experiência também pode ser definida pelo número de horas gastas usando o sistema.

## Memorability

* deve ser fácil de lembrar.

## Teste de usuário padrão com usuários casuais que estiveram fora do sistema por um certo tempo.

**Testar fazendo:** depois que os usuários terminarem uma sessão de teste, peça que eles expliquem o efeito de vários comandos ou o nome de um comando que faz uma determinada coisa

## Errors

* deve ter uma baixa taxa de erro.

Os usuários devem cometer o mínimo de erros possível ao usar um sistema de computador.

A taxa de erro é medida contando o número de tais ações feitas pelo usuário durante a execução de uma determinada tarefa.

## Satisfaction

* deve ser agradável de usar.

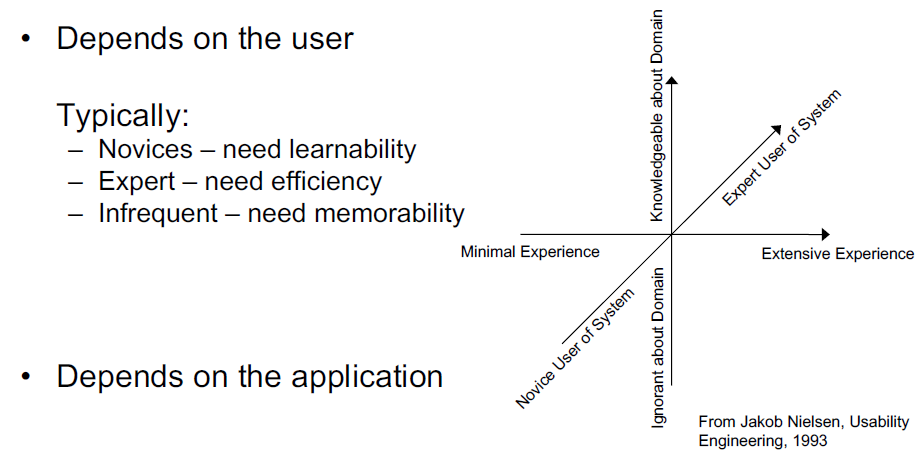
Medidas psicofisiológicas (dilatação da pupila, pressão arterial, frequência cardíaca)

Basta perguntar aos usuários sua opinião subjetiva (média de múltiplas respostas)

Questionários podem ser usados, como os de:

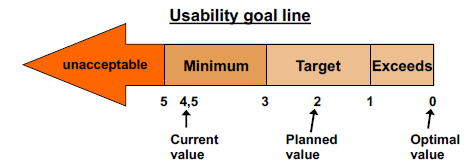
* SUS – Escala de Usabilidade do Sistema
* USO - Utilidade, Satisfação e Facilidade de uso
* QUIS - Questionário de Satisfação da Interface do Usuário
* UEQ – Questionário de experiência do usuário

## Trade-offs



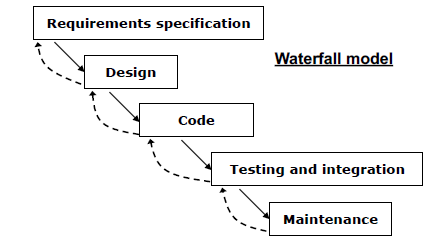
* Considerations other than usability may lead to designs violating some usability principle.
* Make priorities clear on the basis of users and task analysis

## Goal setting



* As metas de usabilidade são razoavelmente fáceis de definir para novas versões de sistemas existentes ou para sistemas que têm um concorrente claramente definido no mercado
* Para novos sistemas completos sem concorrência, as metas de usabilidade são muito mais difíceis de definir

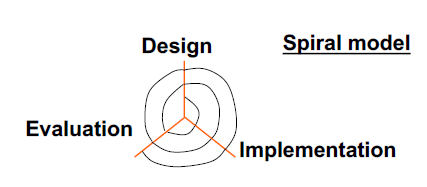
### The Design Process



O modelo em cascata não é apropriado para design de interface de sistemas interativos:

* Os usuários participam apenas dos requisitos especificação e teste.
* A detecção tardia de erros causa custos retificações duradouras.
* Sem suporte para processos realmente iterativos.

# Iterative design



Várias iterações - Custo, precisão e exatidão aumentam a cada iteração.

A primeira iteração pode ser feita em papel: baixo custo

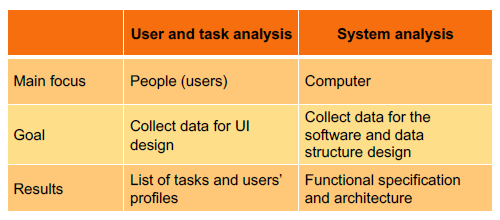
### User and task analysis

O design é baseado nos usuários:

* precisa,
* habilidades,
* contexto,
* trabalhar,
* tarefa.

Coletar dados sobre os usuários e tarefas (características e necessidades) representar os dados, a fim de facilitar a interpretação e orientar o projeto.

**Diferença:**



**To create an application you need to:**

###### User analysis

**Know the user**

* + age, gender, ethnicity
  + Education
  + Physical abilities
  + General computer experience
  + Domain experience
  + Application experience
  + Work environment and social context
  + Communication patterns

**Techniques**

* Questionnaires
* Interview
* Observation

**Obstacles**

* Some users are hard to reach
* Users speak another language

###### Task description

* What users do?
* Why they do it?
* How they do it?
* What they must know?
* What tools they use?

###### Task analysis

Study of the way people perform tasks with existing systems.

**The general method for Task Analysis is:**

* + Observe / Ask
  + collect unstructured lists of words and actions
  + organize using notation or diagrams

**Users can help you learn:**

* + What is related to their job performance
  + What instruments do they use
  + What they actually do

Top-down approach: start with the overall goal of the system and decompose it hierarchically into tasks

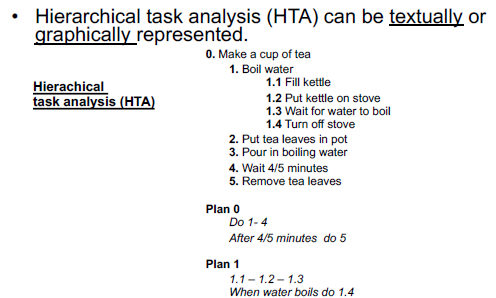
* What needs to be done?
  + Goal
* What must be done to make it possible?
  + Pre-conditions
    - Tasks on which this task depends
    - Information the user needs to know
* What steps are involved in doing the task?
  + Sub-tasks
  + Sub-tasks may be decomposed recursively.

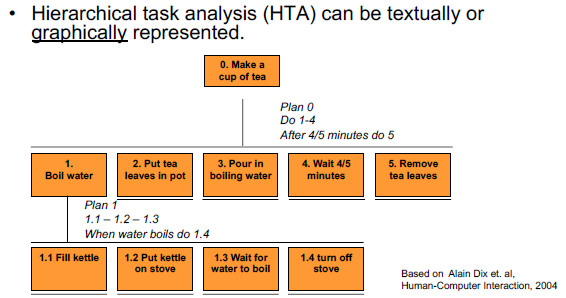
**Collecting information techniques:**

* + - * Direct observation of users performing tasks
      * Interviews with users
      * Contextual inquiry
      * Participatory design
      * Expert advice
      * Documentation analysis
      * Logging

**Example**

* + - Where is the task performed?
      * Supermarket exit, standing up
    - How often is the task performed?
      * once a week
    - What are its time or resource constraints?
      * 3 minutes
    - How is the task learned?
      * Try it
      * Watching others
      * Assistant demo
    - What can go wrong? (exceptions, errors, emergencies)
* Bar code is missing or unreadable
  + - Who else is involved in the task?





**Types of tasks**

* + Fixed sequence (plan 2)
  + Optional tasks (add sugar as task 6)
  + Waiting for events (plan 0 e 1)
  + Cycles (plan 0)
  + Time-sharing (task 1 and 2 can be done at the same time)

###### Observation

Look for the weaknesses in the current system

* Goals not accomplished, wasted time, user irritation

###### Interviews with the users

Estruturado, não estruturado, ou semi-estruturado.

**Plan your questions:**

* + - * How do you perform task X?
      * Why do you perform task X?
      * When (what conditions) do you perform task X?
      * What do you do before you perform task x?
      * What information do you need for…?
      * Who are the persons you need to communicate for…?
      * What do you use for…?
      * What happen after performing…?
      * What is the result of…?
      * What are the consequences of not doing…?

###### Contextual inquiry

Combines interviewing and observation in the user’s actual work environment, discussing actual work

products.

###### Participatory design

Instead of guessing, designers should have access to a pool of representative users. Include representative users directly in the design team.

###### Expert advise

Experts describe tasks as they should be executed not necessary, how they are actually executed.

###### Task scenarios

Based on narratives that describe:

* + Actors
  + Objectives
  + Tools
  + Thoughts/Actions/events (sequence) to achieve the goals
    - A narrative that describes the action that you ask the participant to take on the tested interface.
    - Need to provide context so users engage with the interface and pretend to perform the tasks as if they were at home or in the office.
    - Do not give clues nor describe the steps (not mean being vague)
    - Avoid terms used in the interface

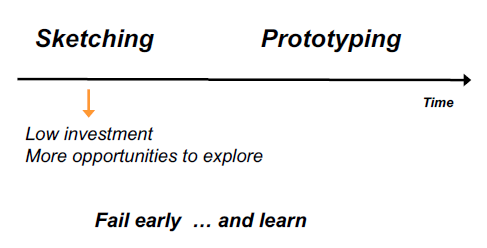
###### Documentation analysis

Describe how it should be done instead of how it is done.

### Sketching

**Attributes of sketches:**

* + - Quickly / Timely
    - Inexpensive / Disposable
    - Plentiful
    - Clear vocabulary
    - Minimal detail
    - Appropriate degree of refinement
    - Suggests and explore rather than confirm
    - Ambiguity



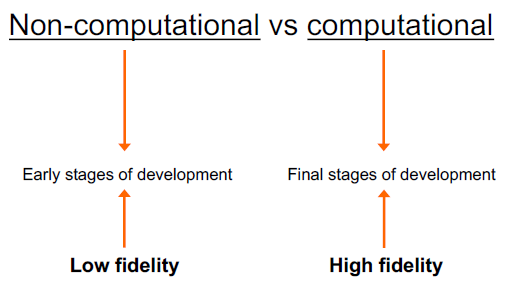
### Prototyping

* Faster development, earlier feedback
* Cheap
* Makes parallel design easier
* Easy to modify and throw away
* The activity of building prototypes encourages reflection in design
* User-centered design

## Fidelity

– Low: omits details, uses cheap materials.

– High: more like the final product.



# 3 approaches to prototyping:

## “Throw-away”

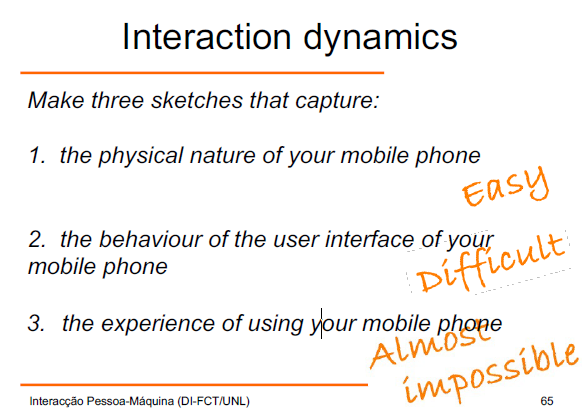
The prototype is built and tested. The knowledge gained by this exercise is used to develop the final product, but the prototype is thrown away.

## Incremental

The final product is built as separate components, one at a time

## Evolutionary

The prototype is not discarded and serves as the basis for the next design iteration. The system is seen as evolving from a very limited initial version to its final release.



# Computational prototypes

* Results

– The same as from the paper prototypes + :

* Efficiency

– Control size, location,…

* Screen layout

– Is it clear, overwhelming, complicated?

– Interface components are easy to find and distinguish?

* Colours, fonts, icons, ...

– Appropriate?

* Interactive feedback

– Do users notice the status bar messages, cursor changes or other interface feedback?