

# Interface Humano-Computador

---

Prof. Jesse Nery

# Identificação de necessidades

- O desenho de novas tecnologias tem, geralmente, como ponto de partida, a identificação de problemas e necessidades.
- Uma das maneiras de identificar essas necessidades é através de coleta de dados.

# Identificação de necessidades

- Observação dos usuários.
- Entrevistas ou Coleta da opinião de usuários.
- Registro de uso ou logs.
- Coleta da opinião de especialistas

# Observação participante

- Na observação participante, o pesquisador vai até o ambiente em que está o usuário final e o observa realizando suas atividades.
- Muitas vezes, os usuários não têm real noção do que fazem, mas um observador participante, ao entrar no ambiente do usuário, tem a possibilidade de compreender o que o usuário faz e porquê faz.

# Observação participante

- Um exemplo de como a observação participante pode ajudar no processo de desenho-redesenho se deu na Qualcomm.
- Uma versão antiga de um dispositivo para caminhões era subutilizado.
- Após a observação do uso deste dispositivo, os projetistas de *interface* descobriram que o fato dos caminhoneiros terem dedos grandes e usarem luvas tornava difícil para eles usarem o aplicativo.
- Então, a partir dessa descoberta, os projetistas elaboraram uma nova *interface* com tela sensível ao toque, além de reorganizarem os elementos de informação para que as respostas mais comuns aparecessem mais, diminuindo a necessidade de digitação.

# Observação participante

Um observador deve se fazer as seguintes perguntas:

- o que as pessoas fazem agora?
- quais são seus valores e objetivos?
- como é o ambiente em que o usuário está envolvido?
- quais as diferenças entre os usuários?

# Observação participante

Durante a observação, o observador deve estar atento a:

- todas as coisas e objetos em volta;
- possíveis alternativas;
- erros de projeto.

# Observação participante

- As vantagens da observação têm muito a ver com a nossa maneira de aprender.
- Na Xerox PARC, foi descoberto que o técnico em manutenção de impressoras via atendimento por telefone era o técnico mais experiente.
- O que a equipe de observação não esperava era que o segundo mais eficiente fosse não o segundo a mais tempo lá, mas o que sentava ao lado do mais eficiente.
- Chegou-se a conclusão que estar ao lado de um especialista dá, ao observador, uma série de conhecimentos informais que não estão em manual algum.

# Observação participante

- Nem sempre o que as pessoas dizem é que realmente desejam.
- O Walmart realizou uma pesquisa através de grupos focais e perguntou se os clientes gostariam de corredores com menos itens. A resposta foi positiva.
- Entretanto, ao implementar essa prática, o Walmart perdeu 1 bilhão de dólares em vendas.
- Assim, não deve-se apenas estar atento ao que as pessoas dizem, mas ao que fazem.

# Entrevistas

- Além de observar as pessoas, também é interessante entrevistá-las.
- O primeiro passo é escolher quem será entrevistado.
- Pode parecer trivial, mas escolher corretamente quem será entrevistado é fundamental para que a entrevista dê bons resultados.

# Entrevistas

São bons candidatos a participantes, pessoas que:

- representam o público-alvo do sistema;
- utilizam sistemas similares;
- não-usuários, porém interessados no sistema.

# Entrevistas

- É preciso estar curioso.
- Cuidado ao criar as perguntas.

# Entrevistas

- Algumas perguntas induzem às respostas.
- Por exemplo, “receber informações em tempo real é importante para você?”
- Não é difícil especular que a quase totalidade das pessoas dirá **sim**. Por que alguém diria **não**?

- Seria mais útil fazer uma observação participativa ou mudar a pergunta para “o que você faz com as informações recebidas em tempo real? Descreva em detalhes.”
- A pergunta torna-se mais ampla, mas permite, também, aprender mais sobre o usuário.

# Entrevistas

- Outro tipo de pergunta que deve ser evitada é “o que você gostaria em uma ferramenta?”
- O entrevistado pode não saber o que responder ou dar uma resposta que não atende as suas reais necessidades ou que não permitam novas possibilidades.
- Não se deve perguntar o que elas fariam ou gostariam em uma cenário hipotético.
- Elas provavelmente terão dificuldades em dizer ou darão respostas falsas.

# Entrevistas

- É sabido também que as pessoas mentem em relação a quantidade de vezes que executam uma determinada atividade.
- “Quantas vezes você se exercita por semana?” ← pouco confiável
- “Quantas vezes você se exercitou na última semana?” ← muito mais confiável

- Perguntar o quanto uma pessoa gostou de algo em uma escala numérica também é complicado.
- Qual a diferença entre um **5** e um **6**? O que significa um **7**?

# Entrevistas

- Outro tipo de pergunta a ser evitada são é aquelas em que a resposta é **sim** ou **não**.
- Especialmente no início de uma entrevista, o ideal são perguntas abertas.

- Outro ponto é dar tempo ao entrevistado responder.
- Dando um pouco mais de tempo, ele pensará e pode até dar uma segunda resposta, mais completa e interessante.

# Entrevistas

Em resumo:

- não usar perguntas que induzem a certas respostas;
- ter cuidado ao usar perguntas que dependem da quantidade de vezes que alguém realizou uma atividade;
- definir bem os critérios das escalas numéricas ou não utilizá-las;
- evitar perguntas binárias;
- dar tempo ao entrevistado responder.

# Identificação de necessidades

- Tanto a observação quanto as entrevistas possuem pontos fracos.
- Principalmente quando o foco é em comportamentos que acontecem durante um longo período, são esporádicos ou ambos.

- Uma estratégia é o uso de diários, onde o respondente deverá preencher, ao executar o comportamento a ser estudado.
- Um diário pode ser de diversas formas, desde em papel, foto, vídeo, áudio ou um aplicativo para celular.

- O uso de diários é mais escalável que a observação direta.
- Quanto mais fácil for ao participante preencher o diário, melhor tendem a ser as respostas.
- Os diários podem levar a um problema de engajamento, pois os participante, após algum tempo, tendem a parar de preencher o diário, sendo necessário relembrá-los.

# Processo de *software*

## Definição

*(Processo de software) - (Sommerville, 2013) “conjunto de atividades relacionadas que levam à produção de um produto de software.”*

# Processo de *software*

- O processo de *software* define a abordagem a ser utilizada segundo a Engenharia de *Software*.
- A Engenharia de *Software* não engloba apenas o processo, mas também as ferramentas utilizadas.

# Atividades do processo de *software*

São atividades do processo de *software*:

- especificação;
- projeto e implementação;
- validação;
- evolução.

Cada uma destas atividades é composta por outras inúmeras atividades. Algumas como documentação e gerenciamento de configuração de *software* servem de apoio às demais atividades.

## Atividades do processo de *software*

As atividades componentes do processo (exceto evolução) podem ser traduzidas para os princípios apontados por George Polya no livro *How to solve it*:

- especificação → compreender o problema;
- projeto → planejar uma solução;
- implementação → executar o plano;
- validação → examinar o resultado para ter precisão.

# Requisitos de *software*

## Definição

*(Requisitos de software) - conjunto de funcionalidades, restrições e características de um software.*

# Requisitos de *software*

Dois grandes desafios da engenharia de requisitos:

- lidar com os interesses dos diversos *stakeholders*;
- garantir que os requisitos sejam descritos de maneira clara e sem ambiguidades.

# Requisitos de *software*



Como o cliente explicou...



Como o líder de projeto entendeu...



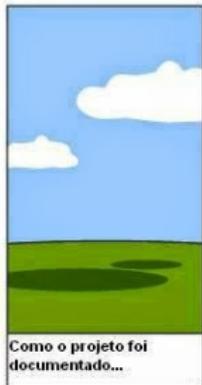
Como o analista projetou...



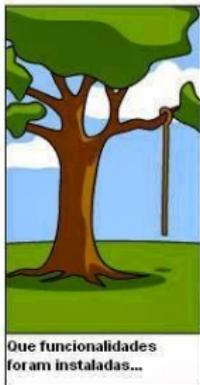
Como o programador construiu...



Como o Consultor de Negócios descreveu...



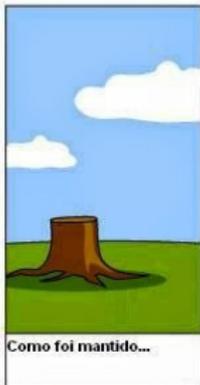
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

# Requisitos de *software*

Tradicionalmente os requisitos podem ser classificados em:

- funcionais;
- não-funcionais.

# Requisitos funcionais

Exemplos de requisitos funcionais:

- O gerente da loja deve ser capaz de acessar os relatórios de estoque mês-a-mês.
- O vendedor deve poder realizar a troca de produto mediante inserção do código da nota fiscal.
- O cliente deve receber, do sistema, um e-mail confirmando sua compra.
- O cliente deve poder acompanhar o *status* da sua encomenda.

# Requisitos não-funcionais

Requisitos não-funcionais:

- podem afetar a arquitetura geral do sistema;
- podem gerar requisitos funcionais ou restringir outros requisitos.

# Requisitos não-funcionais

- Recomenda-se que os requisitos não-funcionais sejam descritos quantitativamente.
- Por exemplo:
  - quantidade de transações por segundo;
  - tamanho ocupado na memória;
  - taxa de ocorrência de falhas.

# Requisitos não-funcionais

Exemplos de requisitos não-funcionais:

- O sistema deve rodar em plataforma web.
- O cliente deve realizar, no máximo, 3 cliques para conseguir acessar o produto que busca.
- A interface do sistema deve estar de acordo com as normas sugeridas pela W3C.

# Engenharia de requisitos

- Não há uma receita simples para escrita dos requisitos, porém busca-se sempre escrevê-los de maneira inequívoca.
- Pode-se utilizar, ao invés da linguagem natural pura, modelos, diagramas, tabelas ou qualquer outra ferramenta.

# Elicitação e análise de requisitos

- Etapa anterior à especificação de requisitos.
- Fortemente centrada nas pessoas.
- Composta pelas seguintes atividades:
  - descoberta de requisitos;
  - classificação e organização de requisitos;
  - priorização e negociação de requisitos;
  - especificação de requisitos.

# Elicitação e análise de requisitos

A elicitação dos requisitos possui alguns pontos chaves:

- os *stakeholders* costumam não saber o que, de fato, querem ou o que é viável;
- os engenheiros de requisitos devem procurar conhecer o trabalho dos *stakeholders* para que se tornem mais próximos da linguagem dos mesmos;
- o engenheiro de requisitos deve estar atento a fatores políticos e *stakeholders* que exijam requisitos para aumentar sua influência;
- o ambiente econômico e social pode acrescentar, modificar e até tornar requisitos obsoletos.

# Validação de requisitos

- A validação de requisitos ocorre em paralelo à análise de requisitos.
- Sua importância se dá em reduzir o retrabalho devido a erros na elicitação e análise de requisitos.

# Verificações de requisitos

As verificações de documento de requisitos que compõem a validação de requisitos são:

- verificações de validade;
- verificações de consistência;
- verificações de completude;
- verificações de realismo;
- verificabilidade (capacidade de teste).

# Validação de requisitos

Dentre as técnicas de validação mais comuns, tem-se:

- revisões de requisitos;
- prototipação;
- geração de casos de teste.

# Gerenciamento de requisitos

- As mudanças nos *softwares* são inevitáveis.
- O próprio uso do *software* pode levar à identificação de novos requisitos.
- Portanto, o gerenciamento de requisitos é necessário em todas as etapas do processo de desenvolvimento.

# Gerenciamento de requisitos

O processo decisório do gerenciamento de requisitos envolve:

- identificação de requisitos;
- processo de gerenciamento de mudanças;
- políticas de rastreabilidade;
- ferramentas de apoio.

# Gerenciamento de mudanças de requisitos

O gerenciamento de mudanças de requisitos envolve três estágios:

- análise de problema e especificação de mudanças;
- análise de mudanças e custos;
- implementação de mudanças.