

Levantamento de Requisitos de IHC

Conceitos Gerais

1. Introdução

- Fazer o levantamento de requisitos é uma das atividades das mais complexas de um projeto de software.
- Esta atividade, geralmente, é realizada por um profissional experiente, que posso induzir os usuários a expressarem suas reais necessidades.
- Em se tratando dos requisitos de IHC, uma série de técnicas devem ser aplicados visando garantir que a interface possa otimizar o desempenho na utilização do software que será desenvolvido.

2. Tipos de Requisitos

- O sucesso de um sistema interativo começa pela correta definição dos requisitos. No contexto de IHC, os requisitos envolvem não apenas funcionalidades, mas também aspectos qualitativos (usabilidade, acessibilidade, desempenho e UX).
- Desta forma, podemos definir requisitos como as especificações formais e informais que definem o que o sistema deve fazer e como deve se comportar.
- Podemos dividir os requisitos em 3 categorias: **Requisitos funcionais, não-funcionais e de interface.**

3. Requisitos Funcionais

- Os requisitos funcionais descrevem uma função que o sistema deve realizar. Ou seja, responde à pergunta: "O que o sistema precisa fazer?".
- Esses requisitos funcionais descrevem basicamente o que um sistema deve ter, por exemplo: controle de estoque, emissão de nota fiscal, fazer um pix, etc.
- Existem diversas técnicas para fazer o levantamento destes requisitos, sendo as mais usadas a entrevista e a observação do fluxo de trabalho.
- Mas esta atividade deve ser feita por analistas experientes, pois extrair os requisitos corretos é uma atividade complexa.

3. Requisitos Funcionais

- Por exemplo, vamos simular uma conversa entre cliente e analista:

Cliente: "Quero um sistema para controlar estoque."

Analista: "Certo. Vou fazer cadastro de produtos, entrada e saída."

Depois da implantação:

"Mas precisamos controlar lotes!"

"Também temos produtos com validade."

"Temos três depósitos."

"O estoque não pode ficar negativo."

"Alguns produtos são vendidos por kg e outros por unidade."

"Precisamos reservar produtos para pedidos."

"A devolução precisa retornar ao estoque."

Resultado: o sistema funciona, mas não atende ao negócio.

3. Requisitos Funcionais

- Foi o programador ou o analista que errou?

Neste caso, provavelmente o processo de levantamento e validação de requisitos que foi feito equivocadamente.

O programador pode ter implementado exatamente aquilo que foi especificado.

Logo, o erro foi do analista.

3. Requisitos Funcionais

- Existem alguns casos famosos de falhas em projetos em virtude de erros nos requisitos, tais como:

1 - FBI Virtual Case File: O FBI tentou desenvolver um grande sistema para permitir que seus agentes compartilhassem e gerenciassem informações de investigações. O projeto recebeu forte investimento, mas devido a falhas graves no levantamento de requisitos (mudança constante, opiniões diferentes, etc.), ele foi abandonado.

2 - HealthCare.gov: O site americano de contratação de seguros de saúde foi lançado em 2013 com problemas graves. No lançamento os usuários tinham dificuldades para criar contas, páginas com erros, falta de disponibilidade e interface complexa.

3. Requisitos Funcionais

3 - Mars Climate Orbiter (NASA): Uma sonda foi perdida ao chegar a Marte, em 1999. Uma parte do software trabalhava com uma unidade de medida, enquanto a outra parte outra unidade. Erro que custou anos de pesquisa e quase meio bilhão de dólares. Falha no levantamento de requisitos (determinar as unidades).

4 - Aeroporto de Denver: Criar software que distribuiria malas por 42 quilômetros de trilhos. O projeto atrasou a inauguração do aeroporto em 16 meses (prejuízo de US\$ 2 bilhões).

Os desenvolvedores ignoraram anomalias do mundo real (malas que se prendem ou caem da esteira), aviões que mudam de portão na última hora.

Os desenvolvedores sofreram com a "síndrome do ir fazendo" e não falaram com usuários finais. Isso gerou muito retrabalho com exigências de mudança no software.

4. Requisitos Não-Funcionais

- Os requisitos não-funcionais dizem respeito as funções do sistema, mas sobre as qualidades que ele deve apresentar.
- Por exemplo: desempenho, usabilidade, confiabilidade, estética, segurança, etc. Segundo Pressman, esses requisitos são críticos no contexto de IHC, pois definem a UX do usuário.
- Um sistema pode estar funcionalmente correto, mas falhar no contexto de aceitação dos usuários por uma série de requisitos não funcionais (tempo de resposta, complexidade, etc.)
- Por exemplo, um sistema bancário que deve fazer 10mil pix por segundo, faz apenas 100 por segundo. Ele cumpre os requisitos funcionais, mas não cumpre os não-funcionais.

4. Requisitos Não-Funcionais

- Alguns cases famosos de falha nos requisitos não-funcionais:

1 – Ariane 5: Em 1996, o foguete Ariane 5 explodiu cerca de 40 segundos após o lançamento. O problema foi uma conversão de um valor de 64 bits para um inteiro de 16 bits (overflow). Um sistema de \$400 milhões deveria ter tolerância a falhas.

2 - Healthcare.gov: Além das funcionalidades não estarem adequadas, este sistema de seguro saúde dos EUA também era extremamente instável.

3 – Mastercard: Operadora ficou indisponível no dia 24 de dezembro no Brasil todo em 2012, gerando prejuízo bilionário.

5. Requisitos de Interface

- No contexto de IHC, podemos definir os requisitos de interface como a forma que o sistema deve se comunicar e interagir com seus usuários ou com outros sistemas.

- Como exemplo:

O sistema deve permitir navegação pelo teclado.

O botão "Salvar" deve estar disponível na tela de cadastro.

O sistema deve apresentar mensagens de erro compreensíveis.

O usuário deve conseguir cancelar uma operação antes da confirmação.

Os Campos obrigatórios devem ser identificados.

A página deve carregar em até 3 segundos.

5. Requisitos de Interface

- Segundo Pressman (2016), os requisitos devem ser claros, mensuráveis e rastreáveis. Além disso, deve existir uma hierarquia de prioridades entre eles.
- Se os requisitos estiverem formulados de forma incorreta, todo o projeto estará comprometido. Segundo Sommerville, requisitos mal definidos são a principal causa de fracassos em projetos de software.
- Segundo Nielsen, requisitos de interface mal especificados são a principal causa de fracasso comercial de produtos de software.

6. Métodos de Coleta de Dados

- A coleta de dados é uma etapa fundamental no processo de levantamento de requisitos em IHC. O analista deve ter a sabedoria necessária para escolher a melhor forma de levantar estas informações.
- Identificar com precisão as necessidades do usuário é uma tarefa muito complexa. Muitas vezes, nem mesmo o usuário sabe suas reais necessidades no contexto de um software.
- Os métodos mais usados são entrevistas, questionários, observação e análise de sistemas antigos ou logs.

7. Entrevistas

- Método mais comum para coleta de dados em IHC. A entrevista pode ser individual ou em grupo, presencial ou remota e roteirizada ou livre.
- A vantagem da entrevista é que permite aprofundar em alguns temas, pegar dados e opiniões subjetivas, além de aumentar a empatia com o usuário.
- A desvantagem é o tempo gasto (usuário para de trabalhar) e possíveis distorções ou vies nas respostas.
- Se a entrevista não for bem conduzida, muitos requisitos podem ficar de fora ou serem interpretados erroneamente.

7. Entrevistas

- Vamos pensar em um roteiro para a entrevista:

1 – Escolha do público alvo:

O primeiro passo é escolher corretamente os usuários que irão participar da entrevista.

Deve-se pegar público o mais heterogêneo possível, de diversos setores, de diferentes cargos, diferentes formações e, principalmente, diferentes níveis de experiência no uso de sistemas.

Se apenas um grupo homogêneo de usuários for entrevistado, corre o risco das informações ficarem enviesadas.

7. Entrevistas

2 - Comece entendendo o trabalho atual: Antes de falar sobre telas e funcionalidades, pergunte:

"Como você realiza essa atividade hoje?"

"Pode me mostrar como você faz?"

"Quais são as etapas desse processo?"

"Quem participa?"

"Quais informações você precisa?"

"De onde vêm essas informações?"

"Para onde elas vão depois?"

Peça para o usuário contar um caso real, não apenas explicar como deveria funcionar. Exemplo: Em vez de perguntar como funciona o cadastro de clientes, peça para ele cadastrar um cliente.

7. Entrevistas

3 - Não aceite a primeira resposta como requisito:

Vamos supor que o usuário pede um botão para exportar para o Excel. Mas talvez este não seja o requisito.

O objetivo é gerar uma planilha e enviar para o gerente comparar com outras planilhas. Desta forma, na verdade o requisito é comparar dados, e não gerar um arquivo Excel.

Talvez, o usuário precise de uma rotina que analise e compare dados automaticamente, mas ainda não percebeu isso.

7. Entrevistas

4 - Use os "5 porquês": Quando o usuário apresentar uma necessidade, pergunte por quê várias vezes?

Exemplo:

"Preciso receber um e-mail quando o estoque estiver baixo."

Por quê?

"Porque preciso verificar o estoque."

Por quê?

"Porque não posso deixar faltar produto."

Por quê?

"Porque a produção para."

Agora descobrimos que o objetivo de negócio é mais importante que simplesmente "mandar e-mail".

7. Entrevistas

5 – Pergunte sobre as exceções: Essa é uma das partes mais importantes do levantamento.

Não pergunte apenas: "O que acontece normalmente?"

Pergunte:

"E quando dá errado?"

"O que acontece quando falta informação?"

"E se o cliente cancelar?"

"E se o pagamento for recusado?"

"E se o sistema estiver fora do ar?"

"Quem resolve?"

"Existe alguma situação em que o procedimento seja diferente?"

7. Entrevistas

6 – Não use linguagem excessivamente técnica se não for necessário

7 – Pergunte "E se...?": "E se o cliente não tiver cpf?", "E se o produto não tiver no estoque?"

8 – Pergunte o que o sistema nunca deve permitir: "Nunca deve ter estoque negativo?", "Nunca pode excluir uma venda faturada"

9 – Desenhe possíveis telas no papel durante a entrevista, para que o usuário possa visualizar.

10 – No final da entrevista, faça uma validação, resumindo com suas palavras o que foi entendido da conversa. E pergunte: "É isso mesmo?"

8. Questionários

- Os questionários são instrumentos estruturados compostos por perguntas abertas e/ou fechadas. Ele permite que um grande número de usuários possa contribuir de forma mais rápida para o projeto.
- Ele é rápido de aplicar, e permite uma fácil comparação dos dados e utilização de técnicas estatísticas.
- Mas no geral, possui baixa profundidade e risco de interpretações equivocadas.

9. Observação

- A técnica de observação (também conhecida como etnografia ou shadowing) é uma das formas mais eficazes de descobrir requisitos ocultos, pois permite ao analista ver o que os usuários realmente fazem, em vez de apenas ouvir o que eles dizem que fazem.
- A observação direta do comportamento do usuário no contexto real de uso é uma das técnicas mais poderosas para levantamento de requisitos.
- Observando os usuários é possível descobrir dificuldades não verbalizadas, estratégias de adaptação e problemas de usabilidade.
- A observação pode sofrer com o **Efeito Hawthorne**, que é a tendência humana de mudar o comportamento (trabalhar mais rápido, seguir regras que normalmente ignora ou evitar atalhos cotidianos) simplesmente por saber que está sendo observado.

9. Observação

- Temos alguns tipos de observação:

1 - Observação passiva (shadowing): Apenas ouvinte, registra as ações sem interferir ou perguntar. Com isso captura o fluxo real de trabalho.

2 - Observação Ativa: Pode interagir e perguntar ao usuário.

3 - Etnografia digital: imersão por tempo prolongado.

4 - Observação participativa: O analista toma o lugar do usuário para entender melhor a rotina

9. Observação

- Vejamos um roteiro para planejamento de uma observação:

1. Preparação (Antes) Definir Escopo:

Focar em um processo específico (ex: Fechamento de Caixa), não no setor inteiro.

Selecionar Perfis: Observar usuários experientes (revelam atalhos) e iniciantes (revelam dores).

Escolha Técnica: Passiva (silenciosa) ou Ativa (perguntas durante o fluxo).

2. Execução (Durante)

Mitigar Efeito Hawthorne: Alinhar que o foco é avaliar o sistema, nunca o desempenho do usuário.

Mapear Gambiarras: Registrar o uso de post-its, planilhas paralelas e anotações físicas.

Identificar Gargalos: Anotar pausas longas, dúvidas, erros e consultas a colegas.

9. Observação

3. Validação & Tradução (Depois)

Debriefing: Entrevistar o usuário logo após a sessão para esclarecer decisões.

Transformação em RF: Traduzir dores observadas em Requisitos Funcionais estruturados.

- É muito comum o analista somar as técnicas de entrevista com a técnica de observação. Com isso, é possível mitigar ao máximo a possibilidade de erros no levantamento de requisitos.

10. Métricas Digitais

- Quando já existe algum tipo de software legado na empresa, é possível se obter diversas informações relacionadas aos requisitos com este sistema.
- É possível fazer uma análise de log, verificar tempo que o usuário gasta para fazer determinadas atividades, analisar o fluxo de navegação, taxa de erros, etc.
- Se for um sistema WEB, pode-se utilizar ferramentas como Google Analytics para uma análise do comportamento do usuário.
- Outra opção interessante é fazer uma entrevista ou observação com os usuários relacionada ao sistema anterior. Com isso é possível observar os pontos positivos e negativos do mesmo, auxiliando no projeto do próximo

11. Considerações Éticas

- A coleta de dados com o usuário sempre deve seguir princípios éticos. A tabela abaixo resume alguns:

Questão ética

Exemplo no levantamento

Privacidade

"Vamos registrar todos os sites acessados pelos funcionários."

Transparência

O usuário não sabe que seus dados serão utilizados para outra finalidade

Consentimento

Coletar informações pessoais sem explicar por quê

Segurança

O cliente pede uma função que expõe dados sensíveis

Manipulação

Interface que dificulte o cancelamento de um serviço

Inclusão

Levantar requisitos apenas com usuários "padrão", ignorando pessoas com deficiência

Conflito de interesses

O analista prioriza o que é melhor para quem paga o projeto, e não para quem utilizará o sistema

11. Considerações Éticas

- Vamos pensar em um exemplo, durante a entrevista, o gerente solicita que o software monitore tudo o que o usuário faz (sites, acessos, arquivos, etc.).
- O analista que trabalha dentro de um contexto ético deve levantar algumas questões:
 - 1 - O funcionário sabe que está sendo monitorado?
 - 2 - Quais dados serão coletados?
 - 3 - Por quanto tempo serão armazenados?
 - 4 - Quem poderá acessá-los?
 - 5 - As mensagens pessoais serão registradas?
 - 6 - O sistema poderá ser utilizado para demissão automática?
 - 7 - O funcionário poderá consultar seus próprios dados?
 - 8 - Existe uma justificativa para coletar cada informação?