

Engenharia de Requisitos de Software

Técnicas de Elicitação de Requisitos



1ª Edição

Copyright © 2022, Afya.

Nenhuma parte deste material poderá ser reproduzida, transmitida e gravada, por qualquer meio eletrônico, mecânico, por fotocópia e outros, sem a prévia autorização, por escrito, da Afya.

Sumário

Técnicas de Elicitação de Requisitos

Para Início de Conversa...	3
Objetivo	3
1. Conceitos e Definições	4
2. Dificuldades da Elicitação de Requisitos	4
3. Técnicas de Elicitação de Requisitos	6
Referências	13

Para Início de Conversa...

A elicitação de requisitos envolve o levantamento das operações que o sistema deve oferecer e como deve oferecê-las. Desta forma, serão apresentadas as definições, dificuldades e técnicas de elicitação de requisitos de software.

Objetivo

- Apresentar definições, dificuldades e técnicas de elicitação de requisitos de software.

1. Conceitos e Definições

Independentemente do processo utilizado por cada organização, em geral, a primeira etapa de qualquer projeto de desenvolvimento de software é a eliciação de requisitos (PFLEEGER, 2004). Elicitar significa investigar, descobrir, tornar explícito, obter o máximo de informações para o conhecimento do objeto em questão. Elicitar requisitos significa identificar os fatos que compõem os requisitos do sistema, de forma a prover o mais correto e mais completo entendimento do que é demandado daquele software. De forma geral, a eliciação de requisitos envolve o levantamento das operações que o sistema deve oferecer e como deve oferecê-las.

2. Dificuldades da Elicitação de Requisitos

Apesar de ser uma etapa comum a qualquer projeto de desenvolvimento de software, a eliciação de requisitos traz consigo uma série de dificuldades que devem ser continuamente tratadas pelos analistas de requisitos. Tais dificuldades são representadas pelas síndromes descritas a seguir.

Síndrome “Sim, mas...”

A síndrome “Sim, mas...” (ou, em inglês, “Yes, but...”) representa basicamente a reação do usuário ao conhecer e avaliar o produto,

mas descobrir que os itens implementados não atendem às suas expectativas. Trata-se de uma reação natural, em razão da própria natureza do software como um processo intelectual intangível, e deve ser considerada no desenvolvimento de sistemas. Portanto, deve-se aplicar técnicas que permitam a antecipação dessas reações, permitindo a melhoria do produto final (LEFFINGWELL; WIDRIG, 2003).

Síndrome “Ruínas não Descobertas”

A síndrome “Ruínas não descobertas” representa a dificuldade de se perceber todos os requisitos de um sistema. Quanto mais se descobre, sente-se que há mais por descobrir e, mesmo com os melhores procedimentos de eliciação e especificação, uma série de problemas ainda persiste (LEFFINGWELL; WIDRIG, 2003).

⚠ Importante

É importante conhecer e identificar todos os *stakeholders* do sistema, pois, de vários deles, pode-se descobrir uma série de requisitos até então desconhecidos. Também é preciso ter a noção de que a missão de descobrir os requisitos do sistema é uma tarefa sem fim. Em certo ponto, devemos reconhecer que já foi descoberto o suficiente para o projeto em questão.

Síndrome “Usuário x Desenvolvedor”

A síndrome “Usuário x desenvolvedor” representa a dificuldade de comunicação existente entre o usuário e o desenvolvedor. Usuários e desenvolvedores, na maioria das vezes, são de mundos completamente diferentes, tendo, cada um, seu próprio linguajar. O desenvolvedor precisa aprender a se comunicar de forma mais adequada e satisfatória com o usuário, para melhorar o entendimento dos requisitos do sistema e produzir um produto com maior qualidade, que atenda às necessidades do usuário (LEFFINGWELL; WIDRIG, 2003).

Agora, reflita: E o usuário? Precisa aprender alguma coisa também ou a responsabilidade é toda da equipe de desenvolvimento?

A maior parte dos problemas relacionados à elicitação de requisitos tem origem nos ruídos dos canais de comunicação entre o cliente e a equipe de desenvolvimento. A Figura 1 representa as possíveis consequências em um projeto quando as pessoas envolvidas não se comunicam adequadamente.

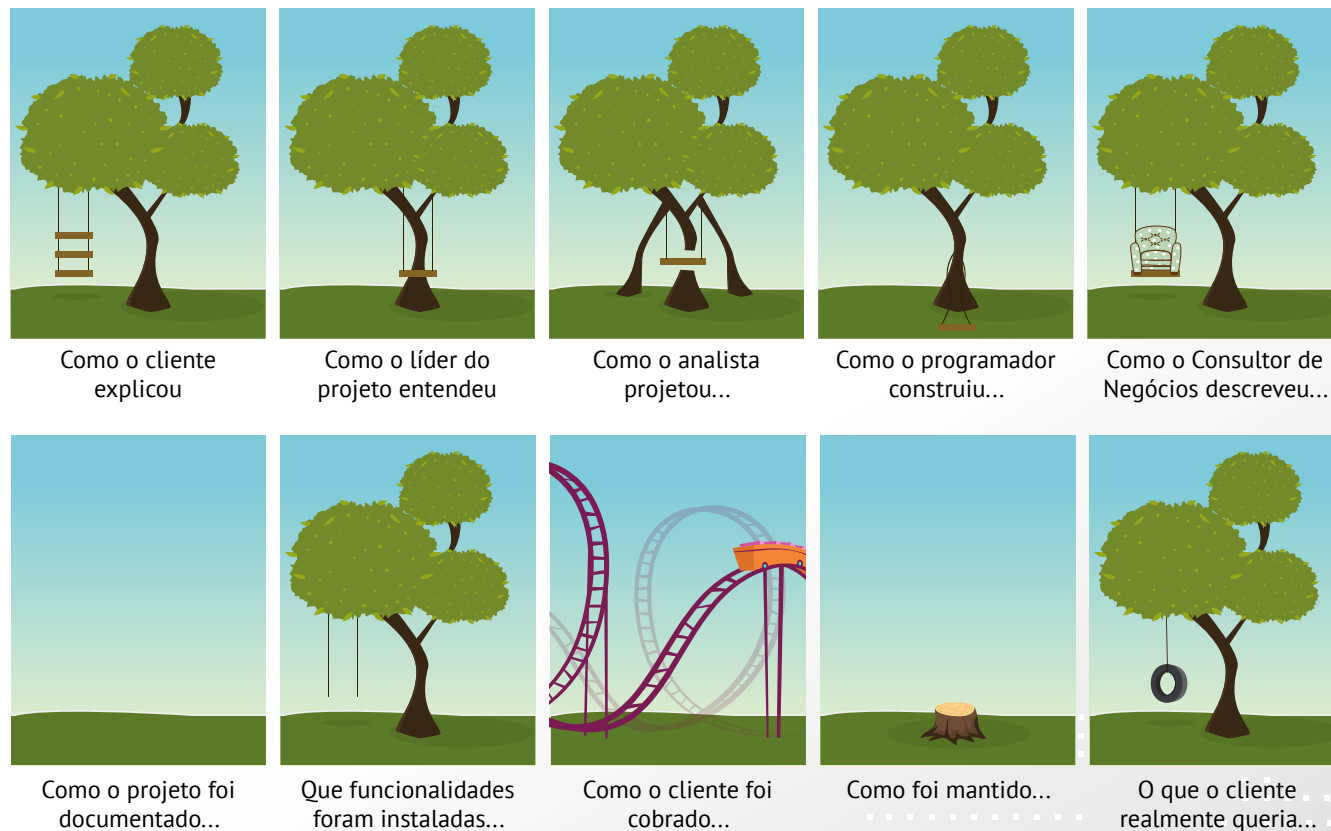


Figura 1: Problemas de comunicação e suas possíveis consequências em projetos. Fonte: Elaborado pelo autor

3. Técnicas de Elicitação de Requisitos

Clientes e usuários ficam insatisfeitos com os produtos de software que lhes são entregues, por uma série de motivos. Um dos mais frequentes para que ocorra esse baixo grau de satisfação é a aplicação inadequada de técnicas de elicitação de requisitos. A princípio, não existe “a melhor” técnica de elicitação ou uma técnica que se aplique a qualquer contexto. Cada técnica possui vantagens e desvantagens. Os problemas inerentes a uma técnica de elicitação específica, eventualmente, podem ser minimizados pela combinação com outras técnicas.

! Importante

O analista de requisitos precisa dominar as técnicas disponíveis e, de acordo com o contexto do projeto, combiná-las da maneira mais apropriada.

.....

A literatura técnica de Engenharia de Software relata inúmeros métodos de elicitação de requisitos, dentre as quais destacam-se as seguintes:

- Entrevistas;
- Questionários;
- Observação;

- *Role Playing*;
- Garimpo de documentos;
- *Brainstorming*;
- *Workshops*;
- Participação ativa dos clientes;
- Prototipação;
- Cenário;
- Reuso de requisitos.

Entrevistas

Entrevista é a técnica mais usual de elicitação de requisitos. Nela, o analista de requisitos discute o sistema com diferentes partes interessadas (*stakeholders*) e obtém um entendimento dos requisitos.

Para realizar uma entrevista, deve-se elaborar um roteiro e permitir que os entrevistados se preparem para responder às perguntas. O entrevistador deve perguntar, ouvir atentamente e recapitular a resposta com o entrevistado, com o objetivo de garantir o entendimento correto da informação que lhe é passada.



Vantagem:

- Contato direto com o usuário e validação imediata.

Desvantagens:

- Grande parte dos usuários entrevistados não consegue explicar sua rotina de trabalho de forma objetiva;
- Em muitos casos, o usuário não se lembra de todos os detalhes relevantes e necessários para a atividade, assumindo que o analista entendeu tudo somente pelas perguntas e respostas;
- Pode exigir grande esforço por parte do entrevistador e por parte do entrevistado, dependendo da quantidade de questões e de entrevistados.

Questionários

A aplicação de questionários é indicada quando há diversos grupos de *stakeholders*, que podem estar localizados em diferentes lugares, o que inviabilizaria ou dificultaria a realização de entrevistas. Dessa forma, o analista de requisitos deve elaborar perguntas e submetê-las, por meio de formulários físicos ou digitais, aos *stakeholders* de interesse. Tais perguntas podem ser fechadas (múltipla escolha), abertas (resposta livre) ou uma combinação de ambas.



Figura 2 - Elaboração de questionário. Fonte: Dreamstime

Cabe ressaltar que a elaboração de questionários exige do analista de requisitos algum conhecimento sobre o domínio do problema, tendo em vista que ele deve elaborar perguntas.

Vantagens:

- Padronização das perguntas e relativa facilidade para analisar as respostas (quando forem de múltipla escolha);
- Possibilitam análises estatísticas;
- Passível de aplicação com grande quantidade de stakeholders.

Desvantagens:

- Limitação do universo de respostas e pouca interação (dependendo das perguntas formuladas, alguns requisitos podem não ser identificados).

Observação

Outra técnica útil ao analista de requisitos é a observação de atividades no momento e nos locais em que são realizadas. Trata-se de um estudo etnográfico, tendo em vista que ela analisa o comportamento das pessoas. Nessa técnica, o analista de requisitos acompanha o trabalho dos usuários em determinadas atividades e documenta os processos e procedimentos executados. Durante a observação, o analista de requisitos pode questionar os usuários sobre procedimentos que não ficaram claros, a fim de melhor compreender o domínio.

Vantagens:

- Técnica especialmente útil para usuários com baixa capacidade de abstração ou que não tenham muito tempo disponível.

Desvantagens:

- Pode inibir o usuário no desempenho de suas atividades e não considera suas experiências;
- Também limita a visão do analista, por estar considerando apenas a forma de trabalho dos usuários, sem considerar outros *stakeholders*.

Role Playing

Role Playing é uma técnica semelhante à de observação, no entanto, no *role playing*, o analista de requisitos realiza (na prática) as atividades do usuário. É interessante que essa técnica seja utilizada de forma complementar às técnicas de entrevista e observação.

Vantagem:

- Técnica especialmente útil para usuários com baixa capacidade de abstração ou que não tenham muito tempo disponível.

Desvantagens:

- Ao utilizar essa técnica, já se deve ter alguma noção do funcionamento da atividade;
- Não considera as experiências vividas pelos usuários.

Garimpo de Documentos

O garimpo de documentos consiste na leitura de documentos existentes na organização que tenham relação com o sistema a ser desenvolvido. Os principais objetivos dessa técnica são compreender o negócio do cliente e capturar o vocabulário do domínio.

Vantagem:

- Facilidade de acesso e volume de informações.

Desvantagens:

- Dispersão das informações e volume de trabalho;
- Apesar de ser importante garimpar documentos, nem todas as informações necessárias estão presentes em documentos, muitas vezes eles não estão atualizados, tornando fundamental a experiência do usuário.

Brainstorming

Em uma sessão de *brainstorming* (tempestade de ideias, em inglês) as partes interessadas (stakeholders) são reunidas em um local, um ambiente que encoraje a participação, permitindo que todas as ideias sejam declaradas em voz alta (para que os demais sejam influenciados) e escritas (para que não sejam perdidas).

! Importante

Em uma sessão de *brainstorming* não pode haver censura. Deve-se tomar cuidado para que os empregados subordinados não forneçam apenas as respostas que agradem aos seus superiores. Portanto, quando *stakeholders* hierarquicamente distintos estiverem presentes, pode-se pedir a cada

participante para anotar, anonimamente, suas ideias no papel e repassá-las ao mediador.

.....

Em geral, uma sessão de *brainstorming* obedece ao seguinte processo:

Geração de Ideias

- Cada pessoa escreve sua própria ideia;
- Ideias vão sendo divulgadas pelo facilitador;
- Novas ideias são criadas a partir das já divulgadas;
- Processo termina quando ninguém tem mais ideias a contribuir.

Redução

- Seleção de ideias que devem ficar;
- Agrupamento das ideias em categorias.

Priorização

- Voto cumulativo (cada um recebe um número X de pontos e os distribui pelas ideias);
- Classificação em “Crítico”, “Importante” e “Útil”.

Vantagens:

- Estimula a criatividade e inovação;

- Possibilita a todos os grupos de *stakeholders* conhecerem as necessidades dos demais grupos;
- Mais eficaz quando cada *stakeholder* detém uma parte do conhecimento sobre algum aspecto do problema.

Desvantagens:

- O grupo, muitas vezes, estará mais interessado em chegar a um acordo do que a um bom resultado;
- A natureza divergente do *brainstorming* encoraja o desenvolvimento de ideias espontâneas, mas não estrutura o refinamento de ideias;
- Precisa de pessoas relativamente qualificadas para resolver o problema;
- A posição social de alguns participantes pode exercer pressões, influenciando o rumo da discussão.

Workshops

Os *workshops* são reuniões planejadas, com apoio de facilitadores, para a discussão de temas em um curto, mas intenso, período de tempo. Geralmente, são reuniões dinâmicas e descontraídas. O principal objetivo da realização dos *workshops* é estabelecer um consenso rápido sobre aspectos de alto nível da aplicação.

Tipicamente, os *workshops* são realizados no início do projeto, especialmente para definir a visão (contexto, objetivos, problemas, soluções).

Vantagem:

- Respaldo de público diversificado e integração entre os grupos.

Desvantagens:

- Custo e logística complicada;
- Quando mal administrado, pode comprometer o processo.

Participação Ativa dos Clientes

A participação ativa dos clientes consiste na incorporação de representantes do cliente e de usuários ao grupo de Engenharia de Requisitos. Nessa técnica, representantes do cliente e usuários precisam aprender as linguagens de modelagem utilizadas para ler as descrições e criticá-las.

Essa técnica proporciona uma integração dos usuários com os analistas que, geralmente, facilita a compreensão do domínio, dada a possibilidade de feedback imediato quando dúvidas surgem.

Vantagem:

- Envolvimento dos clientes e usuários na especificação do sistema.

Desvantagens:

- Requer treinamento dos clientes;
- Pode haver ingerência sobre o processo.

Prototipação

Um protótipo é uma versão inicial de um sistema que poderá ser usado para experimentação. Protótipos são úteis para elicitação de requisitos, porque os usuários podem experimentar o sistema e mostrar os seus pontos fortes e fracos.

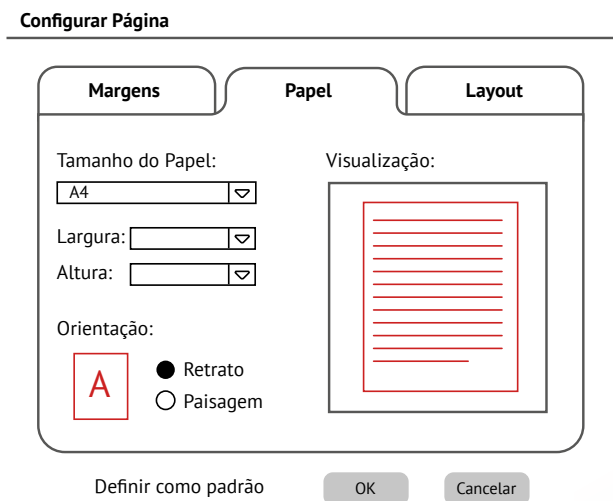


Figura 3: Exemplo de protótipo de uma interface gráfica. Fonte: Elaborado pelo autor

Há dois tipos de protótipos: os protótipos descartáveis, que são descartados após a elicitação de requisitos; e os protótipos evolutivos, que, de forma iterativa, vão incorporando novas funcionalidades e, a partir de um determinado momento, tornam-se o sistema final.

Vantagens:

- O cliente tem algo concreto para criticar;
- Estabelece a viabilidade e utilidade antes que altos custos de desenvolvimento tenham sido realizados.

Desvantagens:

- Custo de treinamento e desenvolvimento;
- Pode não ser possível prototipar algum requisito crítico (incompletude).
- A construção de protótipos presume um conhecimento prévio, mostrando aos interessados possíveis soluções para o problema;
- Muitas vezes, os usuários tendem a se preocupar mais com a interface do que com as funcionalidades;
- Outros problemas seriam o usuário assumir que o protótipo já é a versão final, e a equipe de desenvolvimento admitir fatos não confirmados com os *stakeholders*.

Cenários

Cenários são histórias que explicam como um sistema poderá ser usado. Eles devem incluir:

- Uma descrição do estado do sistema antes de começar o cenário;
- O fluxo normal de eventos do cenário;
- Exceções ao fluxo normal de eventos;
- Uma descrição do estado do sistema ao final do cenário.

A Figura 3 representa um cenário de um sistema de venda de livros representado como uma história de usuário.

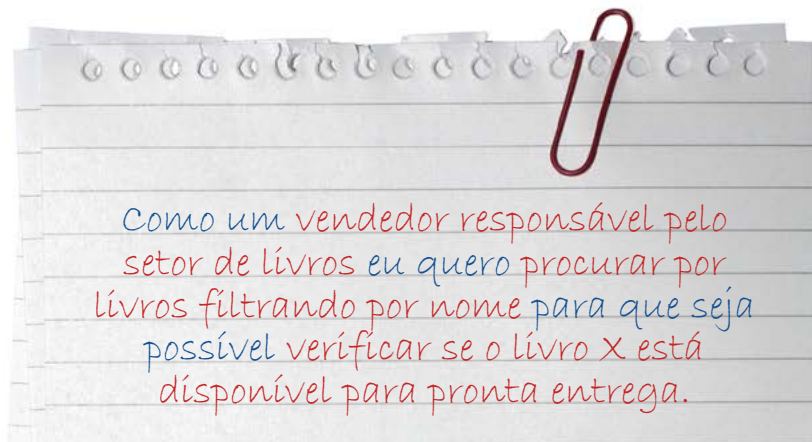


Figura 4: Exemplo de cenário representado na forma de uma história de usuário.
Fonte: Elaborado pelo autor

Vantagem:

- Os cenários são situações menos abstratas para o usuário, promovendo a interação com os analistas.

Desvantagem:

- Cenários complexos podem demandar tempo e aumentar as expectativas do usuário.

Reúso de Requisitos

O reúso consiste em aproveitar requisitos que foram desenvolvidos para um sistema e usá-los em sistemas diferentes. O reúso de requisitos economiza tempo e esforço, pois requisitos reutilizados já foram analisados e validados em outros sistemas. No entanto, há de se considerar as possíveis diferenças de requisitos em sistemas distintos.

Vantagem:

- Relativo ganho de produtividade e qualidade (itens já validados).

Desvantagens:

- Dificuldade de se promover reutilização sem modificação;
- Nem todos os sistemas de mesmo foco possuem as mesmas características (funcionamento); somente o usuário pode determinar o que é necessário.

A elicitação de requisitos é uma atividade complexa que não se resume somente a perguntar o que os *stakeholders* desejam, mas em analisar o domínio da aplicação, os processos de negócio e a cultura da organização na qual o sistema está inserido (SOMMERVILLE, 2011). Portanto, o analista de requisitos necessita identificar as possíveis fontes de requisitos e realizar a extração das informações, de forma cuidadosa e sistemática.

A escolha das técnicas de elicitação deve levar em conta o conhecimento sobre o domínio de aplicação, a quantidade e a disponibilidade de diferentes grupos de *stakeholders*, a existência de documentos ou sistemas legados, entre outros aspectos.

Referências

LEFFINGWELL, D.; WIDRIG, D. **Managing Software requirements:** a use case approach. Addison-Wesley, 2003.

PFLIEGER, S. L. **Engenharia de software:** teoria e prática. 2 ed. São Paulo: Pearson, 2004.

SOMMERVILLE, I. **Engenharia de software.** 9 ed. São Paulo: Pearson, 201