

Aula 09

Site: MoodleWIFI
Curso: Analise de sistemas
Livro: Aula 09
Impresso por: RIANE RUBIO
Data: Friday, 12 Apr 2019, 19:56

Sumário

1. Introdução
2. Especificação de requisitos
3. Rastreabilidade
4. Elicitação de requisitos
5. Levantamento de requisitos
6. Questionário
7. Prototipação
8. Especificação de requisitos

1. Introdução

Objetivos

As fases de **elicitação** (*obtenção de informações detalhadas sobre o que se pretende fazer*) e especificação de requisitos, análise e projeto têm uma importância enorme no contexto de um processo de desenvolvimento de software devido ao fato de identificarem o que deve ser desenvolvido (*requisitos e análise*) e estabelecer como desenvolver (*projeto*).

Portanto, os conceitos de elicitação e especificação de requisitos, análise e projeto de sistemas são, de modo a fornecer uma visão geral destas atividades no contexto do desenvolvimento de software.



2. Especificação de requisitos

Elicitação e especificação de requisitos

Para que um projeto de desenvolvimento de software seja considerado de sucesso, uma das premissas é que o produto gerado atenda o que o cliente deseja.

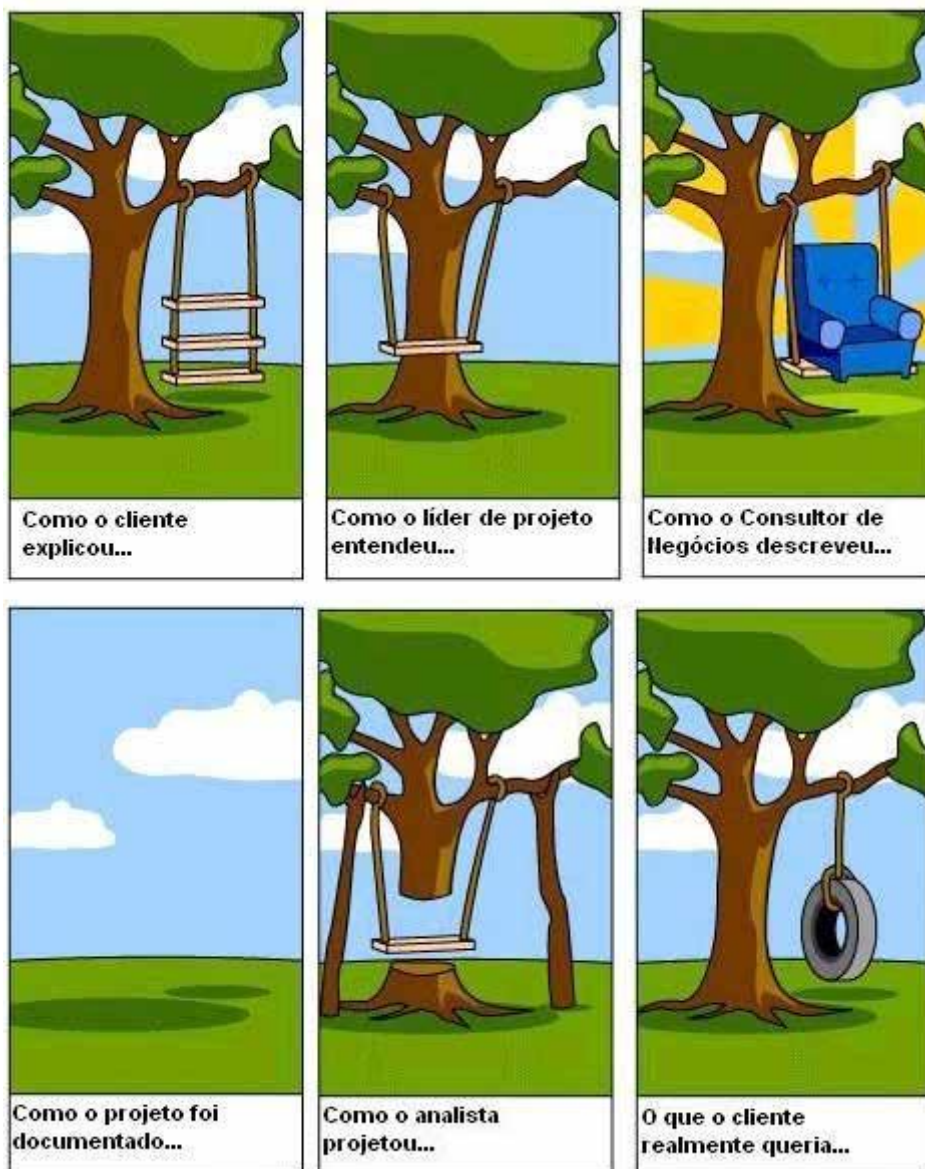
Na grande maioria dos casos o cliente não sabe ao certo o que deseja e por este motivo a descrição das funcionalidades esperadas por parte do cliente pode mudar no decorrer do projeto.

Portanto, há a necessidade de documentação das necessidades do cliente.

Quando estas descrições não são escritas ou são escritas e não são validadas com o cliente, é comum que no momento da entrega o cliente expresse que o produto não é o que ele havia solicitado anteriormente.

Este é um dos grandes erros que os desenvolvedores individuais ou micro empresas de desenvolvimento de software que não se preocupam com requisitos estão sujeitas.

Veja a seguinte imagem que explicita de forma lúdica o que a ausência do uso das técnicas de Engenharia de software causa nas fases iniciais do desenvolvimento de software.



Fases Iniciais do Desenvolvimento sem Técnicas de Engenharia de Software.

3. Rastreabilidade

Como comprovar que o produto desenvolvido está em conformidade com o que foi solicitado?

A solução para esta situação é utilizar descrições do sistema, onde as necessidades são documentadas e validadas pelo cliente.

Deste modo o escopo do software é delimitado, alinhado e acordado entre o cliente e fornecedor (*equipe de desenvolvimento*) do software.

Rastreabilidade

Rastreabilidade é o termo utilizado para expressar um mapeamento que é feito entre os artefatos que se relacionam.

Geralmente este mapeamento é feito através de planilhas e aos pares (Requisitos Funcionais x Casos de Uso).

Requisitos é o nome dado a este conjunto de documentos que descrevem os serviços a serem oferecidos pelo sistema e restrições operacionais de modo a satisfazer as necessidades do cliente.

Outro fator que demonstra a importância de requisitos é a necessidade de manter rastreabilidade entre os artefatos, tendo em vista que os demais artefatos são evoluções do documento de requisitos no decorrer do projeto.

Em linhas gerais, o desenvolvimento dos artefatos ocorre da seguinte forma no decorrer do projeto:

1. O documento de requisitos é criado a partir do levantamento realizado;
2. Especificações de casos de uso são geradas a partir do documento de requisitos;
3. Diagramas da UML são criados para cada especificação de caso de uso;
4. O código é desenvolvido a partir dos diagramas gerados e da especificação de caso de uso (ou história do usuário);
5. As planilhas de teste são criadas tendo como base a especificação de caso de uso (*ou história do usuário*);
6. Quando o cliente vai realizar a validação do software desenvolvido, os requisitos que foram aceitos por ele são utilizados como parâmetro para validar o sistema;
7. Finalmente, quando o software será evoluído, o documento de requisitos é a raiz das mudanças a serem implementadas.

A seguir serão detalhadas as principais técnicas e tecnologias envolvidas na Elicitação (= Levantamento, compreensão) e na Especificação (=Documentação, escrita) de Requisitos.

Especificação de caso de uso é uma forma de descrever o sistema de modo detalhado.

Histórias do usuário possuem formato semelhante aos casos de uso, porém estas estão relacionadas à **metodologias ágeis**.

4. Elicitação de requisitos

Segundo **Sommerville** (2007), nesta atividade os profissionais de informática trabalham diretamente com os clientes e usuários finais do sistema para aprender sobre o domínio da aplicação, quais funcionalidades o sistema deve fornecer, o desempenho esperado e restrições de infraestrutura (hardware, rede...).

Esta atividade pode envolver profissionais com vários perfis dentro da organização para que se tenha uma visão abrangente das necessidades de cada perfil.

Em um sistema a ser desenvolvido para a área de telemarketing, por exemplo, os gerentes, atendentes, diretores, setor de recursos humanos, porteiros e profissionais do setor de Tecnologia da Informação da empresa podem estar envolvidos.

Estes perfis podem variar de acordo com a aplicação a ser desenvolvida, em um sistema de transporte um motorista pode ser um envolvido.

Dada esta diversidade, os envolvidos são denominados através do termo técnico **Stakeholder**. *(público estratégico, descreve uma pessoa ou grupo que tem interesse em uma empresa, negócio ou indústria, podendo ou não ter feito um investimento neles. Em inglês **stake** significa **interesse**, participação, risco. **Holder** significa **aquele que possui**)*

Além dos **Stakeholders**, outras fontes de requisitos são:

- Ambiente físico,
- Documentação (Formulários de cadastro de alunos, por exemplo),
- Dados existentes,
- Recursos existentes,
- Sistemas legados e Segurança.

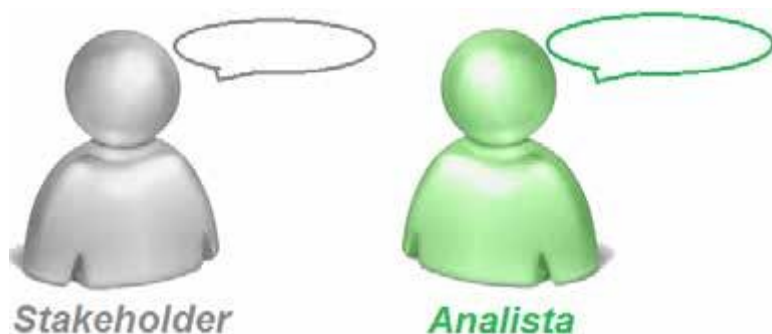
Extrair os requisitos do domínio do problema nem sempre é uma tarefa trivial.

Vários aspectos podem contribuir para isto, dentre eles podemos elencar a disponibilidade dos clientes e clareza do que é realmente necessário ao negócio.

Deste modo é necessário que seja utilizado um conjunto de técnicas de elicitação, a depender das dificuldades enfrentadas.

A seguir, algumas técnicas de levantamento de requisitos e suas descrições:

- **Entrevista:** A equipe de análise de sistemas reúne-se com o(s) stakeholder(s) para uma conversa sobre as necessidades e expectativas em relação ao sistema a ser desenvolvido;



Entrevista entre um analista e um stakeholder.

5. Levantamento de requisitos

- **Observação:** Como o próprio termo sugere, esta técnica consiste na observação do ambiente onde a aplicação irá funcionar a fim de captar informações acerca do funcionamento dos processos da empresa;
- **Demonstração de Tarefa:** Para algumas tarefas específicas a observação isolada pode não ser suficiente.

Muitas vezes é necessário que uma determinada tarefa seja mostrada detalhadamente e repetidas vezes.

Diante deste cenário, a técnica de demonstração de tarefa é utilizada, os stakeholders fazem a demonstração das tarefas aos analistas de sistemas;



Demonstração de Tarefa.

- **Estudo de Documentos:** Documentos em papel, como formulários de cadastro e relatórios já utilizados auxiliam bastante a atividade de elicitação de requisitos, uma vez que estes podem revelar um conjunto de dados esperado para uma funcionalidade;
- **Substituir o Usuário (Role playing):** Quando o domínio da aplicação é cheio de tarefas específicas e complexas, somente substituindo o funcionário que as realiza é que o analista de sistemas consegue ter uma visão abrangente.

Ao substituir o usuário, o analista repete os passos que o mesmo faria, sendo auxiliado pelo usuário em relação às dúvidas que possam surgir.



Analista substituindo o Usuário.

6. Questionário

- **Questionário:** Questionário consiste em redigir um conjunto de perguntas que devem ser respondidas pelos usuários.
Os questionários podem ser feitos para tirar dúvidas existentes em relação aos requisitos já elicitados ou na descoberta de requisitos novos;
- **Brainstorming** (ou *tempestade de idéias*): Para o desenvolvimento desta técnica é necessário que diferentes perfis de usuário participem.

A execução desta técnica segue os seguintes passos:

1. Um tempo máximo de reunião é estabelecido e os participantes tem um momento para refletir sobre o tema abordado;
 2. Em seguida, os participantes vão ditando as idéias e todas elas são escritas de modo que todos possam ver;
 3. Encerrada a sessão, as repetições de idéias são retiradas e idéias que geraram dúvidas são melhor explicadas;
 4. As idéias são agrupadas e combinadas;
 5. As melhores idéias escolhidas coletivamente são analisadas, melhoradas e aproveitadas.
- Esta técnica é importante para alinhar a visão de todos os participantes em relação ao sistema que será desenvolvido. Na Figura é ilustrada a realização de um brainstorming, com a presença do analista e dos stakeholders.

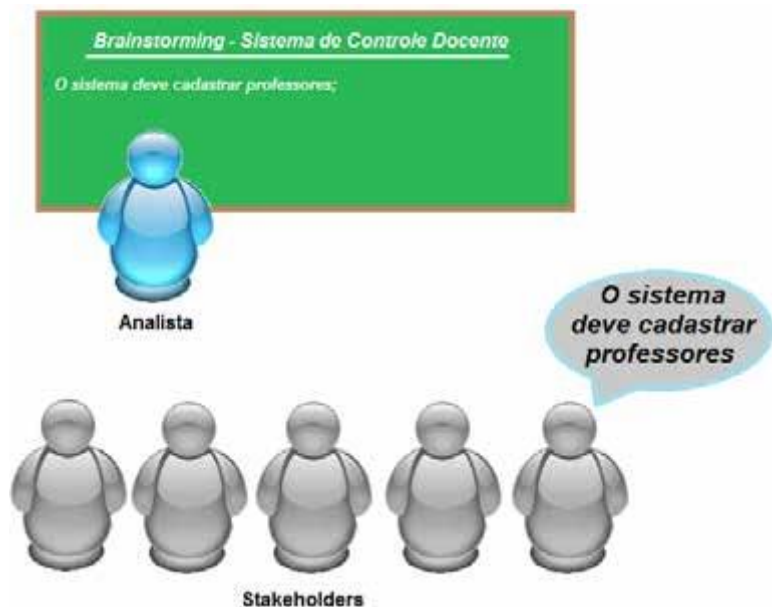
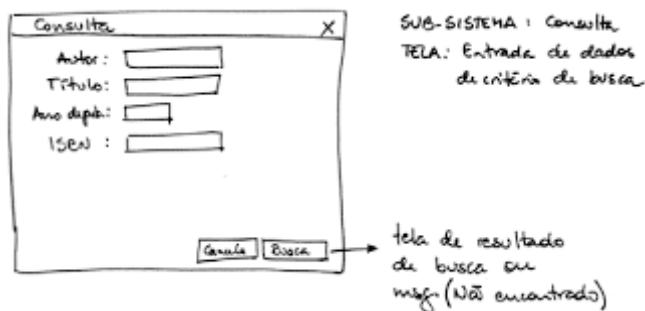


Ilustração da realização de um brainstorming.

7. Prototipação

- **Prototipação:** Consiste na criação de um esboço inacabado do sistema. Geralmente o esboço demonstra a interface de uma determinada funcionalidade do sistema para que o usuário possa ter uma melhor visualização do que será implementado. Este esboço representa a implementação inacabada do sistema, muitas vezes somente a parte de interface gráfica com os formulários com botões e campos de texto; no entanto, sem acesso a banco de dados, tratamento de campos ou processamento de cálculos;



- **Workshops ou Oficinas de Requisitos:** Reúnem todos os envolvidos durante um período curto, mas intensivo e focado. Neste período várias técnicas mencionadas podem ser aplicadas sequencialmente. Uma vez que as técnicas foram empregadas e a elicitação dos requisitos foi dada como encerrada, surge uma nova necessidade: Escrever aquilo que foi identificado para fique documentado e para que possa ser validado pelo cliente. Assim sendo a especificação de requisitos é utilizada com tal finalidade.

8. Especificação de requisitos

Os requisitos podem ser classificados em **funcionais**, **não funcionais** e **requisitos de domínio**.

Segundo **Sommerville** (2007), os requisitos Funcionais de um sistema descrevem o que o sistema deve fazer.

Esses requisitos dependem do tipo do software que está sendo desenvolvido, dos usuários a que o software se destina e da abordagem geral considerada pela organização ao redigir os requisitos.

Um exemplo de requisitos deste tipo seria: *“O sistema deve realizar o cadastro de clientes”*

Já os requisitos não funcionais, como o próprio nome sugere, são aqueles que não estão diretamente relacionados às funções específicas fornecidas pelo sistema.

Eles podem estar relacionados às propriedades emergentes, como confiabilidade, tempo de resposta e espaço de armazenamento.

Um problema comum aos requisitos não funcionais é que eles podem ser difíceis de verificar.

Um exemplo de requisitos deste tipo seria:

- “O sistema deve estar preparado para a realização de até 300.000 transações simultâneas”

Os **requisitos de domínio**, por sua vez, refletem necessidades relacionadas ao domínio da aplicação e incluem uma terminologia específica de domínio ou fazem referência a conceitos do domínio.

Os analistas muitas vezes encontram dificuldade em compreender e de relacioná-los com outros requisitos do sistema.

Requisitos de domínio podem estar relacionados com velocidade de execução, confiabilidade, interoperabilidade com outros sistemas e leis que estejam relacionadas ao domínio da aplicação.

Um exemplo de requisito deste tipo seria: *“O IMC deverá ser calculado através do valor do peso (em kg) dividido altura (em metros)”*.

Os requisitos são necessários para a criação de diversos artefatos do sistema como diagramas, código, protótipo de interface e planilhas de teste.

Dada esta importância, os requisitos devem ser escritos de forma precisa e isto muitas vezes é difícil devido às ambiguidades que frequentemente ocorrem ao se escrever na língua portuguesa.

Veja a seguinte frase: *“O professor falou com o aluno parado na sala”*, a ambiguidade neste caso, se dá em relação a quem estava parado: o professor ou o aluno.

Assim como na frase, comumente a escrita de requisitos é permeada de ambiguidades, que devem ser eliminadas para evitar interpretação errada e criação de artefatos de forma errada.

Além da ausência de ambiguidade, duas propriedades desejáveis aos requisitos são a **completeza** e **consistência**.

Completeza significa que tudo que o usuário necessita que seja implementado esteja definido e consistência significa que os requisitos não devem ter definições contraditórias.

O uso de **jargões computacionais** também figura no rol de vícios frequentemente cometidos que devem ser evitados em requisitos.

Um exemplo deste erro seria: “A consulta aos clientes deve ser feita através de um transfer object que deve ser transitado entre as camadas”.

Esta informação não é do escopo de requisitos, mas deve aparecer em outro documento (O documento de arquitetura – Este não está no escopo desta aula).

A seguir temos alguns requisitos do sistema e em seguida a classificação de cada requisito que aparece neste documento como Funcional, Não Funcional ou de Domínio.

Um exemplo de definição contraditória seria em determinado momento do documento de requisitos aparecer a informação que *“Inicialmente nenhum usuário deve vir pré-cadastrado no sistema”*.

Enquanto que em outro trecho deste documento aparece a informação que *“O primeiro acesso ao sistema deve ser feito com o perfil de administrador, este deve vir pré-cadastrado”*.