



Apostila de Exercícios

Aula Bonus do programa "SAP Fiori para Analistas Funcionais"

Cristiano Marques cristiano.marques@gmail.com

Novembro / 2020



■ ÍNDICE

INFORMAÇÕES SOBRE TÉCNICAS UTILIZADAS			3
		NTREVISTAS	
		Entrevistas Tutoriais	
	1.2.	ENTREVISTAS INFORMAIS OU NÃO ESTRUTURADAS	.4
	1.3.	ENTREVISTAS ESTRUTURADAS	.4
	2.	PROTOTIPAÇÃO	. 5
		StoryBoards	
		SIMULAÇÃO	
		CENÁRIOS	
		CASOS DE USO	
	5.	JAD (JOINT APPLICATION DESIGN)	. 7
	6.	WALKTHROUGHT ESTRUTURADO	3.



INFORMAÇÕES SOBRE TÉCNICAS UTILIZADAS

No processo de elicitação de requisitos devem-se utilizar algumas das diferentes técnicas para descobrir as informações. Estas técnicas auxiliam no processo de controle da qualidade do *software*, uma vez que sua utilização provê requisitos bem definidos, completos e corretos.

Para minimizar os problemas relacionados à elicitação, várias técnicas foram desenvolvidas. A escolha da técnica apropriada para elicitar requisitos depende do tempo e dos recursos disponíveis, assim como do tipo de informação necessária. Técnicas de elicitação de grupo – compreendem técnicas de dinâmica de grupo com o objetivo de entender de forma mais detalhada as necessidades dos usuários.

A seguir estão relacionadas algumas das técnicas sugeridas, tais como: entrevistas, prototipação, elaboração de cenários, casos de uso, JAD, walkthrought estruturado, entre outras. Cada uma delas, individualmente, não é auto-suficiente, necessitando-se da utilização conjunta de técnicas para produzir um resultado completo e eficiente.



1. ENTREVISTAS

As entrevistas são questões relacionadas às partes interessadas e/ou suas necessidades no contexto do problema a ser resolvido, que são formuladas, e as respectivas respostas, anotadas.

Esta técnica é bastante útil quando partes interessadas possuem muitos conhecimentos subjetivos e, ao mesmo tempo, estão dispostos a serem entrevistadas. Para que o processo seja mais eficiente, o entrevistador deve ser experiente.

A realização de entrevistas é uma técnica simples e direta de interação entre entrevistado, normalmente o cliente, usuário ou especialistas de negócio, e entrevistador, que é a pessoa responsável pela elicitação de requisitos, buscando revelar conceitos, objetos e a organização do domínio do problema, além de buscar soluções ou projeções de soluções.

As entrevistas podem ser abertas, nas quais as necessidades e o conhecimento do domínio de aplicação são discutidos abertamente, sem um conjunto pré-definido de perguntas, ou fechadas, onde o entrevistador prepara, previamente, um conjunto de perguntas que serão feitas. Os tipos de entrevistas mais utilizadas são:

1.1. Entrevistas Tutoriais

 Neste tipo de entrevista, o usuário é convidado a falar sobre a problemática. É dele a tarefa de conduzir a conversa. É como se ele explicasse, a partir do seu ponto de vista, como funciona o dia-a-dia do negócio. Há pouco direcionamento do entrevistador em relação ao tema a ser seguido. Suas intervenções são necessárias apenas para garantir a compreensão do tema.

1.2. Entrevistas Informais ou não Estruturadas

 Neste caso, o entrevistador age espontaneamente, sem obedecer nenhum tipo de organização. O entrevistado responde ao entrevistador apenas aquilo que lhe é perguntado. Esta abordagem oferece maior flexibilidade ao entrevistado, que pode aprofundar-se e direcionar a conversa conforme o seu andamento. Geralmente é utilizada no início do processo de elicitação.

1.3. Entrevistas Estruturadas

• As entrevistas são preparadas com antecedência, quando se define previamente o tema a ser conversado e qual o aprofundamento que será dado. É interessante ao longo de um processo de elicitação para dirimir dúvidas específicas.



2. PROTOTIPAÇÃO

A prototipação é utilizada de forma a aumentar a confiabilidade dos sistemas desenvolvidos, facilitando a simulação e modelação inicial a baixo custo das idéias definidas nas fases de desenho de alto nível. É a implementação parcial de um sistema de forma rápida, para obter feedback útil ao processo de requisitos. É utilizada para elicitar requisitos quando há um alto grau de incerteza, ou quando é necessário um rápido feedback dos usuários.

Consiste na produção de uma versão inicial do sistema, que possa estar disponível aos usuários na fase inicial do processo de desenvolvimento. Muitas vezes, somente os requisitos que causam maior dificuldade são incorporados aos protótipos, uma vez que estes devem estar prontos na fase inicial do processo de desenvolvimento do sistema. Os protótipos também podem ser produzidos como uma seqüência de telas em papel, descrevendo a funcionalidade de cada elemento com o qual o usuário poderá interagir.

O protótipo permite que os usuários experimentem e descubram o que eles realmente necessitam para suportar o trabalho deles, estabelecendo a viabilidade e utilidade antes que altos custos de desenvolvimento tenham sido realizados. Pode ser usado para teste do sistema e desenvolvimento da documentação, forçando um estudo detalhado dos requisitos que revela inconsistências e omissões.

Um dos tipos de prototipação utilizado na elicitação de requisitos é o que chamamos de descartável, este é útil para ajudar a elicitação e desenvolvimento dos requisitos, sendo que os requisitos a serem prototipados devem ser aqueles que causam mais dificuldades para os clientes e são os mais difíceis de se entender. Requisitos que são bem entendidos não precisam ser implementados pelo protótipo.

A técnica de prototipação é muito efetiva para elicitação com usuário final, permitindo que o usuário revise casos de uso e sistemas de forma visual. Diversos níveis de prototipação podem ser utilizados, tais como:

2.1. StoryBoards

• Forma mais simples de prototipação, pois se trata de descrição gráfica semelhante ao que se pretende desenvolver no sistema real. Não é necessário qualquer tipo de programação para serem concebidos e eficazes, em sua forma mais simples basta papel e caneta, uma simulação do sistema é desenvolvida em papel e usada para experimentação do sistema, embora existam pacotes de software que permitam automatizar certos aspectos do processo. Mostra os pontos-chave da interface em determinados momentos, permitindo o usuário ter a noção de como o sistema irá se comportar e julgar se a solução encontrada necessita ou não ser melhorada. Mais barato, sem compromisso visual e enfoca dinâmica e navegação;

2.2. Simulação

Técnica que não exige suporte computacional especial, permitindo que apenas parte da funcionalidade esteja desenvolvida, sendo a restante providenciada através de intervenção humana, ou seja, um elemento da equipe faz a vez do sistema interagindo com o utilizador uma pessoa simula as respostas do sistema em resposta a alguma entrada do usuário. Esta interação permite que em tempo real se altere a configuração do sistema face a sugestões ou erros encontrados. Figuras com compromisso visual que permitem navegação, mas tem nível reduzido de interação com usuário.



3. CENÁRIOS

Cenários representam uma descrição técnica e detalhada dos eventos, que permite a compreensão de seu funcionamento, elementos de informação e humanos requeridos, e objetivo pretendido, portanto são bastante úteis na elicitação de requisitos, pois são descrições de como o sistema é usado na prática.

É importante ter uma visão crítica do sistema em definição: não basta estruturar o ambiente e definir o cenário com as informações colhidas, é preciso analisá-lo de forma crítica, questionando a validade de cada parte da estrutura e seu funcionamento, estabelecendo prioridades, eliminando procedimentos redundantes ou dispersivos e passos sem propósito, sem aplicação produtiva no processo interno da empresa.

Esta técnica consiste na produção de cenários, que são exemplos de seções de interação entre o usuário final e o sistema em uma única situação. Os usuários finais do sistema simulam suas interações utilizando cenários. Eles expõem as pessoas responsáveis pela elicitação de requisitos as suas atividades e quais são as informações que eles necessitam, para que seja feita a descrição do cenário.

Os cenários facilitam o trabalho dos *stakeholders* para relatar exemplos de tarefas reais, das quais poderão ser abstraídas as descrições das funções que o sistema a ser desenvolvido, ou alterado, deve prover; além de auxiliar no entendimento dos requisitos e de revelar possíveis facilidades que o sistema poderia necessitar. Uma simples construção de cenários envolve o tratamento de detalhes que não poderiam ser observados de outra forma. É, contudo, uma tarefa que consome tempo e, para sistemas complexos é necessário um grande número de cenários. Têm por vantagem que, uma vez desenvolvidos, podem ser reutilizados através do próprio sistema ou em outros sistemas no qual se aplique o mesmo contexto.

Os cenários podem ser descritos de forma que o usuário define seus objetivos e as etapas, enumerando as atividades que o *software* deve abranger. Inicia-se, então, a fase exploratória, com tarefas, elementos de decisão e testes, os quais são relacionados e podem ser divididos. Nesta fase, deve-se incluir a participação do empresário apenas como mais um usuário do sistema, pois o conjunto deve funcionar de forma harmônica. Na fase de comprometimento, olhem-se visões e dados a serem trabalhados por cada elemento ou grupo. E na fase de sugestão, faz-se a apropriação dos elementos colhidos, divisão do trabalho, desenvolvimento e testes de aplicação. Surgem, assim, novos pontos de vista, novos elementos e novos testes, até o aprimoramento dos pontos a serem realmente inseridos como requisitos.

A estrutura para uma definição de cenário pode ser composta dos seguintes itens:

- **Título:** identifica o cenário;
- **Objetivo:** estabelece a finalidade de um cenário. O cenário deve descrever o modo em que este objetivo deve ser alcançado;
- **Contexto:** descreve o estado inicial de um cenário, suas pré-condições, o local (físico) e tempo;
- Recursos: identifica os objetos passivos com os quais lidam os atores;
- Atores: pessoa ou estrutura organizacional que tem um papel no cenário;
- **Episódios:** cada episódio representa uma ação realizada por um ator onde participam outros atores utilizando recursos disponíveis. Um episódio também pode se referir ao outro cenário. Episódios podem conter restrições e exceções. Uma restrição é qualquer imposição que restrinja um episódio de um cenário. Uma exceção é o tratamento para uma situação excepcional ou de erro.



4. CASOS DE USO

Os Casos de Uso são como um cenário que descreve como o *software* deve ser usado numa determinada situação. Na medida em que os requisitos são elicitados é possível criar um conjunto de cenários que identifique uma linha de uso para o sistema a ser construído. Para criar um caso de uso, o analista deve, primeiramente, identificar os diferentes atores, que podem ser pessoas ou dispositivos, que usarão o sistema.

Especifica o comportamento de um sistema ou de parte dele. É uma descrição de um conjunto de seqüências de ações, incluindo ações alternativas, realizadas pelo sistema, para produzir um resultado esperado por um ator. Podem ser aplicados para captar o comportamento pretendido pelo ator do sistema que está sendo desenvolvido, sem que seja necessário especificar como este comportamento será implementado.

O conceito de *use-case* é uma das principais características da notação UML para a descrição de modelos orientados a objetos. Ele baseia-se na representação de atores que interagem com o sistema. Atores são, portanto, qualquer entidade externa ao sistema. Geralmente os atores representam as pessoas que utilizam o sistema, mas também podem representar outros processos não-humanos.

Um *use-case* identifica que tarefas ou funções principais são desempenhadas por determinado ator, que informação do sistema o ator vai adquirir, produzir ou modificar, que informação o ator deseja do sistema. Demonstra também se o ator precisará informar ao sistema sobre alterações ocorridas no ambiente externo e se o ator deseja ser informado sobre alterações inesperadas.

5. JAD (JOINT APPLICATION DESIGN)

JAD é um conjunto de sessões intensivas e mediadas entre usuários e analistas de um sistema, com o objetivo de explicitar os seus requisitos.

Esta técnica é talvez a mais poderosa para elicitar requisitos, por ser uma técnica de reunião que tem por objetivo acelerar e consolidar o desenvolvimento de aplicações. Sua aplicação permite a criação de sistemas mais eficazes em menor tempo, tendo como um grande benefício à aderência que pode ter a vários métodos de desenvolvimento de software atualmente empregados.

Pode ser definida como uma técnica orientada ao trabalho em grupo, que oferece um ambiente para clientes e engenheiros de requisitos trabalharem como um time, compartilhando informações e idéias sobre um determinado tema ou assunto. Faz uso de sessões de *brainstorming* mediada por um facilitador, que assumirá um papel chave neste processo, de conduzir as discussões e mediar conflitos durante a sessão. Na medida em que as discussões progridem os requisitos.

Sessão conjunta de usuários e equipe de projeto para definição de modelos de casos de uso e especificações de casos de uso críticos. É estruturada com moderador, equipe de definição (tipicamente usuários e analistas) e escribas.

Os componentes básicos de uma sessão JAD são:

- **Patrocinador:** é o dono do sistema. Deve ser influente o bastante na companhia para ser capaz de tomar decisões e prover os recursos necessários para manter o projeto.
- **Líder:** geralmente o líder do grupo de desenvolvimento da aplicação e é capaz de responder perguntas sobre o escopo do projeto, tempo, assuntos de coordenação e recursos. Devem participar das sessões desde que não inibam os participantes.
- **Facilitador:** comanda a reunião e direciona o grupo a seguir a agenda estabelecida. É responsável por identificar os assuntos que podem ser resolvidos como parte da



reunião e aqueles que devem ser colocados no final da reunião para ter uma investigação e resolução. Ele guia os participantes e não contribui com informações.

- **Participantes:** clientes da área de negócios afetada direta ou indiretamente pelo projeto, que são experts em suas funções e possam tomar decisões sobre seus trabalhos. Eles são a fonte de entrada na reunião.
- **Observadores:** geralmente membros do time de desenvolvimento da aplicação do projeto. Observam silenciosamente os procedimentos.

A sessão JAD deve ter objetivos bem definidos, agenda detalhada e modelos, além de um documento final contendo todas as decisões tomadas pelo grupo.

As cinco fases das sessões JAD são:

- Definição do projeto JAD;
- · Pesquisa nos requisitos do usuário;
- Preparação para a sessão JAD;
- Conduzir e intermediar a sessão JAD;
- Demonstrar e obter aprovação do documento final que incorpora todas as decisões tomadas.

Pode-se definir cinco papéis principais como participantes das sessões JAD, que são eles:

• Condutor / Analista de sistemas / Patrocinador / Usuários / Documentador.

Ainda, organizar o JAD em quatro sessões:

- **Reunião Inicial:** ocorre a abertura dos trabalhos, a apresentação dos participantes e a definição de aspectos globais do projeto, especificando a finalidade e escopo do projeto, os objetivos específicos a serem alcançados com o sistema, a problemática atual, limitações do sistema, as pessoas que deverão ser envolvidas nas sessões seguintes e é criado um cronograma de trabalho.
- Levantamento de dados e análise: o analista de sistemas responsável pelo projeto deve efetuar o levantamento de dados e a análise das definições da reunião inicial, descrevendo os processos e elaborando diagramas de fluxos de dados para um melhor entendimento.
- **Planejamento para a Sessão de Design:** é prudente que o analista de sistemas realize uma reunião com o condutor, na qual o analista apresenta ao condutor a proposta a ser debatida na Sessão de *Design*, para que ele tome conhecimento do sistema e apresente críticas e sugestões.
- Sessão de Design: é quando a aplicação é efetivamente trabalhada. Realiza-se uma reunião entre os profissionais da área de informática, os profissionais do setor responsável pela aplicação, o condutor do JAD e o documentador. Representantes de outras áreas muito provavelmente serão convidados a participar, além de consultores externos, se necessário. Esta reunião divide-se em 5 etapas, quais sejam:
 - 1) Apresentações, definição de objetivos, levantamento de problemas e limites;
 - 2) Análise dos processos, dados e interfaces do sistema;
 - 3) Análise do novo fluxo e necessidades administrativas sugeridas;
 - 4) Elaboração de cronograma, aprovação do novo fluxo e avaliação da reunião;
 - 5) Elaboração do documento final.

6. WALKTHROUGHT ESTRUTURADO

• **Técnica de validação** de requisitos com o objetivo de obter aceitação de requisitos. Pode ser empregada para casos de uso, protótipos e combinações dos dois. É estruturada com



moderador, revisor, (tipicamente usuários) e escribas. O processo consiste em sessões de leitura e revisão dos requisitos para identificação de problemas e melhorias.

- **Técnica de análise estática** na qual um projetista ou programador apresenta aos membros do grupo de desenvolvimento e outros profissionais interessados uma parte de documentação ou código, e os participantes fazem perguntas e comentários sobre possíveis erros, violação de padrões de desenvolvimento ou sobre outros problemas. Varredura efetuada por uma equipe no código fonte em busca de erros, analisando questões de eficiência para melhorar a qualidade do software.
- **Técnica de revisão em grupo** de um software ou de alguns de seus componentes. Consiste na formação de uma equipe, com papéis predefinidos para cada um de seus membros, liderada pelo projetista do software. O objetivo é analisar o material para questionamentos e comentários.