

Capítulo I. Usuários e Requisitos

*A hundred objective measurements didn't sum the worth of a garden;
only the delight of its users did that.
Only the use made it mean something.*

Lois McMaster Bujold, A Civil Campaign, 1999

Stakeholders
Requisitos Funcionais
Requisitos Não Funcionais
Descrevendo Requisitos
Elicitação de Requisitos
Entrevistas e JAD

A principal tarefa de um analista é descobrir o que o sistema deve fazer e como deve se comportar segundo as expectativas de seus **usuários** e outros interessados. Usualmente, chamamos esses desejos dos usuários de **requisitos**¹.

O problema de encontrar os requisitos é que, no início do desenvolvimento, ninguém realmente sabe o que um sistema desejado deve exatamente fazer ao ficar pronto, inclusive o cliente. Por isso, encontrar os requisitos é uma tarefa investigativa, que envolve descobrir e validar os requisitos encontrados.

Além disso, com o tempo, os requisitos mudam. É importante, então, manter os requisitos sempre atualizados

Dessa forma, descobrir e registrar os requisitos de um sistema, além de mante-los sempre atualizados, é um trabalho bastante difícil, investigativo, criativo e contínuo.

I.1 Stakeholders e Usuários

Os sistemas que desenvolvemos devem ser utilizados. Na verdade, uma medida importante da qualidade de um sistema de informação e do sucesso do projeto que o construiu é quanto do sistema é realmente usado pelos seus usuários, ou seja, quanto do que foi feito é realmente útil.

Logo, para descobrir o que deve ser feito (o quê), devemos primeiro entender para quem o sistema se destina (quem).

Usuários são todos aqueles que usam de alguma forma o sistema com algum objetivo. O nome pode ser entendido de forma restrita, indicando apenas aqueles usuários finais, isto é, que realmente usam o sistema dentro do escopo do seu objetivo, ou de forma ampla indicando todos aqueles que usam o sistema ou algum de seus produtos, de alguma forma, o que inclui também os desenvolvedores, programadores, instaladores, operadores, etc.

¹ Em inglês, *requirement*, o que levou alguns tradutores a usar o termo requerimentos, considerado inadequado.

Stakeholders, ou **partes interessadas**, são todos aqueles com algum interesse no sistema, afetando ou sendo afetados por seus resultados. A palavra é uma composição dos termos *stake*, interesse ou aposta, e *holder*, possuidor. Fica claro que um *stakeholder* é uma pessoa que possui algum interesse no sistema, em especial um interesse que envolve algum risco. Atualmente a tradução mais comum para esse termo é “parte interessada”, mas esse termo não tem todo o significado do termo em inglês, que adotaremos na falta de um melhor em português. O grupo de *stakeholders* é bem maior que o grupo de usuários, pois envolve não só estes, mas também financiadores e outros que podem até mesmo nunca ter contato direto com o sistema, mas de alguma forma são afetados por ele.

Abordagens simplificadas permitem identificar imediatamente três tipos de *stakeholders*: **desenvolvedores**, **compradores** e **usuários finais**. Uma investigação mais profunda pode achar muitos outros interessados, como: gerentes dos usuários finais, auditores, responsáveis pela operação, responsáveis pela manutenção, usuários de sistemas que enviam ou recebem dados para o sistema específico, etc. É interessante que seja feito um esforço para levantar todos os *stakeholders* de um sistema e mapear, de alguma forma, seus interesses e interações com o mesmo.

I.1.1 Interesses e Objetivos

Usuários possuem um ou mais **objetivos** ao usar um sistema, i.e., ele usa o sistema para obter uma resposta planejada. Usuários e *stakeholders* têm **interesses** no sistema, isto é, eles esperam que o sistema afete o negócio de alguma forma.

É importante atentar para a diferença que existe entre usar o sistema para fazer uma ação(o que), o objetivo do uso, com o motivo ou motivação do sistema para um *stakeholder* (por que), o interesse.

Estudar os interesses nos permite saber melhor como realizar os objetivos. Por exemplo, um cliente de web site de vendas de produtos tem como objetivos, ao longo de uma compra, várias ações, como escolher um produto, saber seu preço, pagar o produto. Porém ao longo de todo esse tempo, seu único interesse é receber o produto. Mais do que comprar, o usuário quer recebe-lo em casa. Então toda a dinâmica do site, ou do software e logística do sistema por trás dele, pode ser construída voltada para esse interesse. Mesmo que quiséssemos atender um interesse menor, como “comprar o produto”, já poderíamos imaginar formas diferentes de implementar as ações correspondentes. Um exemplo disso no mundo real é o “one-click buy”, ou seja, comprar um produto com só um click do mouse e ter certeza de recebe-lo, ideia que foi inclusive patenteada pela Amazon.

Uma prática útil e que pode facilitar a compreensão do ambiente onde os sistema será implantado é definir uma tabela indicando que objetivos cada usuário e que interesse cada *stakeholder* tem no sistema. Lembre-se, porém, que todo usuário é *stakeholder*. Isso pode ser feito, inclusive, desde o início do projeto, mesmo que de forma preliminar.

A tabela a seguir é um exemplo típico. Outro exemplo está no diagrama do tipo ThinkMap após a tabela.

Objetivos e Interesses de Stakeholders			
Agente ou Interessado	Objetivo	Interesse	Prioridade
Cliente	Fazer pedido	Receber o produto pedido	1
Cliente	Obter status do pedido	Prever o prazo de chegada do pedido	2
Gerente	Obter lista de pedidos diária	Saber a produção diária	1

Tabela 1. Lista de objetivos e interesses

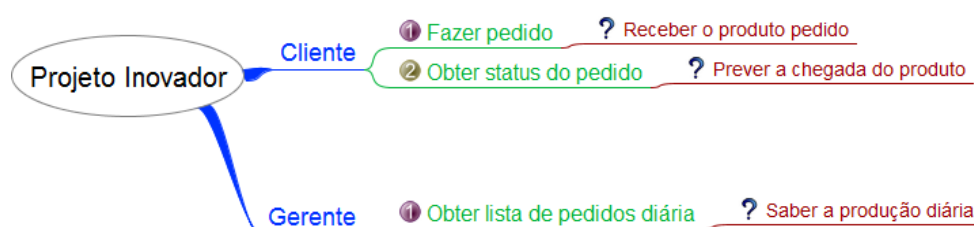


Figura 1. Um mind map com a informação de objetivos e interesses de stakeholders.

I.2 Perspectivas dos Usuários

Quando estamos fazendo a análise de um sistema, por meio de entrevistas ou reuniões, interagimos com pessoas com visões e descrições diferentes do que é o sistema. Um cuidado importante que devemos ter é quanto à posição da descrição que está sendo feita em relação ao sistema.

Mais tarde veremos que as técnicas de análise de sistema estão interessadas, no início, em informações e comandos que partem do ambiente, de fora, para dentro do sistema, ou no sentido inverso. Basicamente, grande parte da análise é dedicada a estabelecer as fronteiras do sistema e que informações cruzam essa fronteira em uma direção ou outra. Desta forma, devemos descrever todo o sistema com essa perspectiva.

Nossos clientes, porém, não têm sua perspectiva limitada pelo nosso modelo, afinal eles conhecem seu negócio e não precisam de um método como o nosso, onde as limitações têm como objetivo clarificar a compreensão do sistema pelo analista.

Assim, muitas vezes um entrevistado descreve o sistema como se fosse uma entidade mágica, outros descrevem o sistema como se fizessem parte dele, outros descrevem apenas as saídas do sistema, etc.

Devemos estar preparados para isso e garantir uma modelagem condizente, que atenda todos os pedidos do usuário. Devemos também fazer, sutilmente, o usuário compreender a nossa forma de descrever o sistema, pelas suas fronteiras. Muitos usuários

entendem facilmente esses conceitos quando traduzidos para português mais simples, como “entrada de dados” e “entrada de comandos”.

A figura a seguir mostra uma forma um pouco antiga, mas muito utilizada até hoje para definir as fronteiras de um sistema: um Diagrama de Fluxo de Dados, em seu modo contextual.

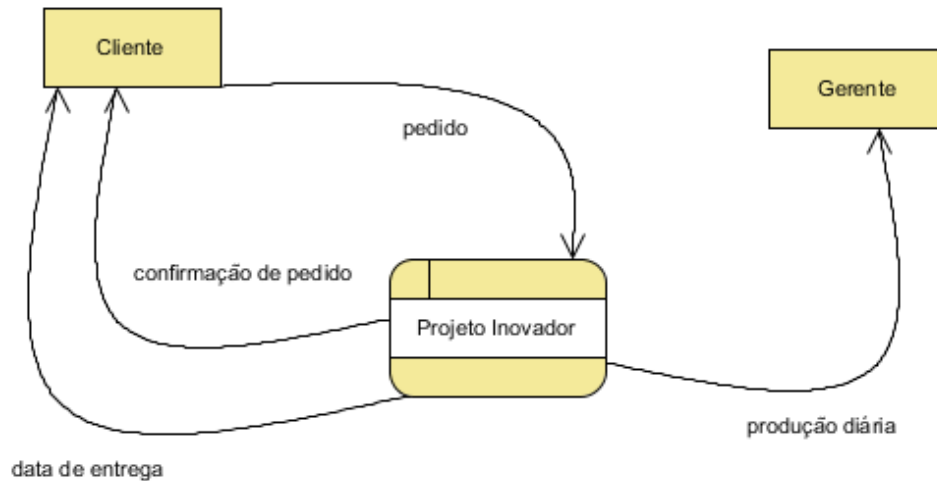


Figura 2. Um diagram do tipo DFD é uma das técnicas usadas para delimitar a fronteira de um sistema e as informações que as cruzam.

As perspectivas básicas que encontramos em entrevistas e reuniões são as seguintes:

- **Entrevistado onisciente:** descreve o sistema como “o sistema”, indicando coisas que ele “deve fazer”. Vê tanto o sistema como os seus usuários de uma perspectiva externa, aparentemente conhecendo os mecanismos tanto por dentro quanto por fora. Normalmente é a posição da alta gerência e de quem contratou o sistema. É comum que não conheça os procedimentos internos do sistema como acontecem realmente, mas apenas de forma geral ou como aconteciam no passado. Exige funcionalidade do sistema, principalmente para atender o nível gerencial.
- **Entrevistado usuário:** descreve o sistema como se o estivesse usando diretamente, muitas vezes já usando o sistema atual. Exige funções do sistema, principalmente para atender o seu nível de atuação (gerencial ou operacional). Pode apresentar alguma desconfiança, pois o novo sistema pode exigir novos conhecimentos. Conhece a entrada e a saída do sistema, mas não necessariamente os procedimentos internos.
- **Entrevistado parte do sistema:** descreve o sistema visto por dentro. Muitas vezes é quem vai ter o trabalho substituído, em todo ou em parte, pelo sistema, o que pode causar desconfiança e até mesmo franca hostilidade. Conhece os procedimentos na forma como são realizados e as exceções que podem acontecer.

Não podemos deixar de entender que alguns usuários fornecem uma perspectiva mista, principalmente quando envolvidos com diferentes partes do sistema.

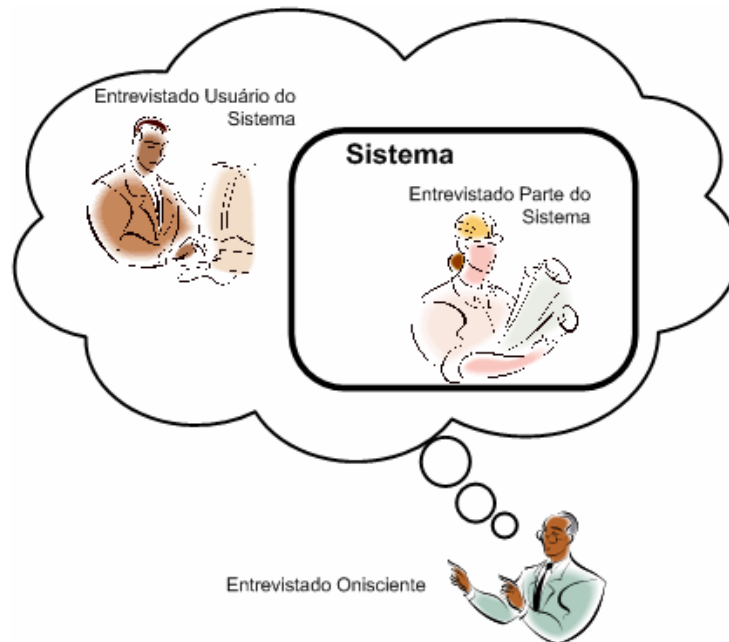


Figura 3. Os tipos de entrevistados em relação ao sistema

Na prática do desenvolvimento de software, percebemos que na grande maioria dos casos o usuário tem dificuldade de expressar suas necessidades. Isso pode acontecer por vários motivos, como o desconhecimento, o foco nas técnicas de solução (que são responsabilidade do analista e não do usuário), a dificuldade de encontrar uma linguagem comum com o desenvolvedor e muitos outros.

O que devemos notar é que apesar do analista ter que atender aos usuários, não tem que atender exatamente ao que eles dizem, mas sim ao que realmente precisam. O trabalho de análise é um trabalho investigativo, realizado junto com os usuários. Discutiremos esse assunto um pouco mais ao falar da elicitação de requisitos.

Não podemos nunca esquecer, porém, que um dos principais fatores de sucesso de um projeto de software é a participação do usuário. Então, não basta entrevista-los, mas devemos conquista-los para a “equipe” do sistema. Por isso é tão importante entender as motivações e os interesses, e não apenas os objetivos.

I.3 O que é um Requisito

Segundo a norma IEEE Std 610.12-1990, **um requisito é**

- **Uma condição ou capacidade necessária para um usuário resolver um problema ou atingir um objetivo**
- **Uma condição ou capacidade que precisa ser possuída ou alcançada por um sistema ou componente de uma sistema para satisfazer um**

contrato, padrão, especificação ou outro documento formalmente imposto

- **Uma representação documental de uma condição ou capacidade no sentido das duas definições anteriores.**

Qualquer que seja o sistema, existem várias formas de entendê-lo. Similarmente, existem vários tipos de requisitos, que são aplicáveis ou não, dependendo da visão necessária naquele instante.

- Um **requisito do usuário** é algum comportamento ou característica que o usuário deseja do software ou o sistema como um todo; o que o usuário quer. São escritos pelo próprio usuário ou levantados por um analista de sistemas que consulta o usuário.
- Um **requisito do sistema** é algum comportamento ou característica exigido do sistema como um todo, incluindo hardware e software. O comportamento desejado do sistema. São normalmente levantados por engenheiros ou analistas de sistemas, refinando os requisitos dos usuários e os transformando em termos de engenharia.
- Um **requisito do software** é algum comportamento ou característica que é exigido do software. São normalmente levantados por analistas de sistemas. O objetivo da análise é capturar todos os requisitos para o software sendo desenvolvido e apenas aqueles requisitos verdadeiros.

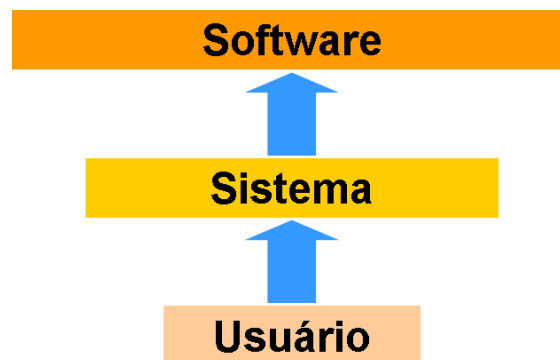


Figura 4. Requisitos formam uma pirâmide invertida, sendo que o nível inferior serve de base para a criação do nível superior, que contém sempre mais requisitos.

Exemplos de requisitos são:

- O sistema deverá imprimir a nota fiscal de venda ao consumidor referente à venda sendo registrada.
- O sistema deverá permitir ao usuário calcular diferentes orçamentos para uma mesma proposta, baseados em formas diferentes de pagamento.
- O sistema deverá avisar que a rede está fora do ar em 20 ± 4 segundos após a rede sair do ar.

- O sistema deverá permitir agendar uma consulta, reservando a data e o horário da sala e do profissional de acordo com as disponibilidades da clínica e o desejo do paciente.

I.3.1 Características de um Bom Requisito

Um requisito deve ter as seguintes características [B8]:

- **Necessário**, significando que, se retirado, haverá uma deficiência no sistema, isto é, ele não atenderá plenamente as expectativas do usuário.
 - Em especial, não devem existir requisitos do tipo “seria interessante ter”. Ou o requisito é necessário ou é dispensável.
 - Deve ser levado em conta que cada requisito aumenta a complexidade e o custo do projeto, logo, não podem ser introduzidos de forma espúria.
- **Não-ambíguo**, possuindo uma e apenas uma interpretação. Nesse caso é muito importante prestar atenção na linguagem sendo utilizada.
- **Verificável**, não sendo vago ou geral e sendo quantificado de uma maneira que permita a verificação de uma das seguintes formas: inspeção, análise, demonstração ou teste.
- **Conciso**, de forma que cada requisito defina apenas um requisito que deve ser feito e apenas o que deve ser feito, de maneira clara e simples.
 - Um requisito, para ser conciso, não inclui explicações, motivações, definições ou descrições do seu uso.
 - Estes textos podem ser mantidos em outros documentos, apontados pelo requisito (de preferência sob o controle de um sistema de gerência de requisitos).
- **Alcançável**, ou seja, realizável a um custo definido por uma ou mais partes do sistema
- **Completo**, ou seja, não precisa ser explicado ou aumentado, garantindo capacidade suficiente do sistema.
- **Consistente**, não contradizendo ou mesmo duplicando outro requisito e utilizando os termos da mesma forma que outros requisitos,
- **Ordenável**, por estabilidade ou importância (ou ambos).
- **Aceito**, pelos usuários e desenvolvedores.

Além desses requisitos, quando um requisito for funcional, deverá ser também **Independente da Implementação**, ou seja, o requisito define o que deve ser feito, mas não como.

I.3.2 Efeitos de Requisitos Errados

A importância de obter requisitos corretos pode ser compreendida rapidamente se imaginarmos que um requisito, quanto mais básico e importante ele é, mais partes do sistema vai afetar.

Basicamente, cada requisito do usuário vai gerar um ou mais requisitos do sistema, que vão, por sua vez, gerar outros requisitos mais detalhes. O efeito, com a evolução do projeto, é que cada requisito inicial representa não só um compromisso, mas um fator crítico em uma sequência de decisões que são tomadas ao longo do projeto.

Assim, os erros de requisito afetam drasticamente todo o projeto. Dados empíricos indicam que um problema nos requisitos, se detectado nas fases finais do projeto, pode custar 100 vezes mais para ser corrigido do que se for detectados na fase de elicitação de requisitos. Mesmo que essa detecção se faça mais cedo, em uma fase intermediária, esse custo ainda seria 10 vezes maior.

Um requisito que não caiba em uma das características de qualidade citadas será propenso a defeitos e aumentará o risco do projeto. Se não for necessário, gastaremos esforços em sua definição e implementação que seriam mais bem gastos tratando de outros requisitos. Além disso, cada requisito adicional aumenta o risco do projeto como um todo, ainda mais se a adição for dispensável. Se for ambíguo, corremos alto risco de implementá-lo errado, desagradando o cliente. Se não for verificável, não poderemos testar de maneira inequívoca se está corretamente atendido, possibilitando discussões sobre sua realização. Se não for conciso, estamos aumentando a complexidade do documento de especificação, permitindo interpretações erradas. Se não for independente da implementação, corre o risco de estar erroneamente definido ou se tornar obsoleto, ou ainda errado, caso a tecnologia evolua. Se não for alcançável, o projeto não terá como terminar adequadamente. Se não for completo, faltará ao projeto alguma coisa que o usuário necessita. Se não for consistente, o projeto chegará a encruzilhadas ou talvez se torne impossível. Se não for ordenável, não saberemos em que momento realizá-lo, e, finalmente, se não for aceito tanto pelos usuários quanto pelos desenvolvedores, se tornará uma discussão permanente no projeto.

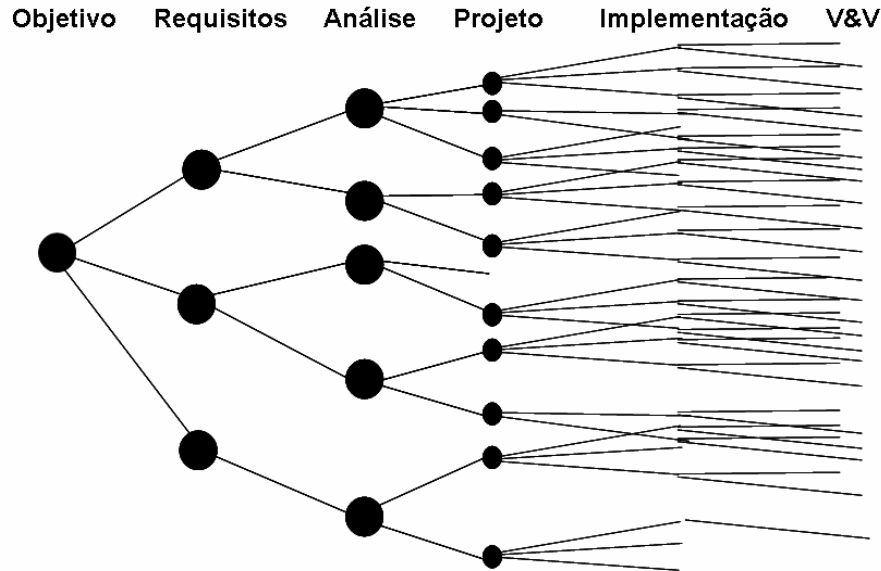


Figura 5. O efeito de um requisito original vai se espalhando pelo projeto.

I.3.3 Requisitos Mudam com o Tempo

Também é importante perceber que os requisitos de um software mudam com o tempo. Essas mudanças ocorrem porque o ambiente em que o software reside também muda. Os países alteram suas leis, as organizações alteram suas práticas, a tecnologia evolui, os usuários exigem novas funcionalidades até então não imaginadas ou desnecessárias.

Caper Jones², em 2005, afirmou que a taxa de mudanças de requisitos para um software comercial chega a 3,5% ao mês, e para um software para Web, chega a 4,0% ao mês. Considerando o tempo médio de execução desses e outros tipos projetos, ele chega a valores médios de 2,58% de mudanças ao mês, 14 meses em média de execução e 32,33% de mudança total nos requisitos. É importante notar, porém, que todas essas afirmações são baseadas em cálculos com Pontos de Função para manutenção, onde pequenas alterações podem causar grandes efeitos.

Esse fato tem efeitos em nossa prática. O primeiro é que devemos estar preparados para a mudança, pois ela vai acontecer. O segundo é que quanto maior o tempo de duração do projeto, maior a quantidade de mudanças de requisitos, o que aumenta ainda mais o risco, que já é afetado pelo projeto ser longo e provavelmente complexo.

Assim, não só é importante conhecer os requisitos, mas também conhecer qual a sua estabilidade. Requisitos pouco estáveis devem ser tratados de forma diferente dos requisitos mais estáveis³. Requisitos mais críticos devem ser tratados de forma diferente dos requisitos opcionais.

² <http://www.stsc.hill.af.mil/crosstalk/2005/04/0504Jones.html>

³ Mesmo que muitas vezes requisitos supostamente estáveis mudem, mas não há como nos prepararmos de forma economicamente viável para essa alteração.

Tipo de Software	Taxa Mensal de Mudança de Requisitos
Sistemas de Informação	1.5%
Sistemas Militares	2.0%
Software Básico	2.0%
Sistemas Comerciais	3.5%

Tabela 2. Algumas taxas mensais de mudança de requisitos, segundo Caper Jones.

I.4 Requisitos e Necessidades

Requisitos são originários de necessidades dos usuários e *stakeholders*. Enquanto os requisitos vivem no “mundo das soluções”, as necessidades vivem no “mundo dos problemas”. Os requisitos implementam as características observadas do sistema, que atendem as necessidades (Figura 6).

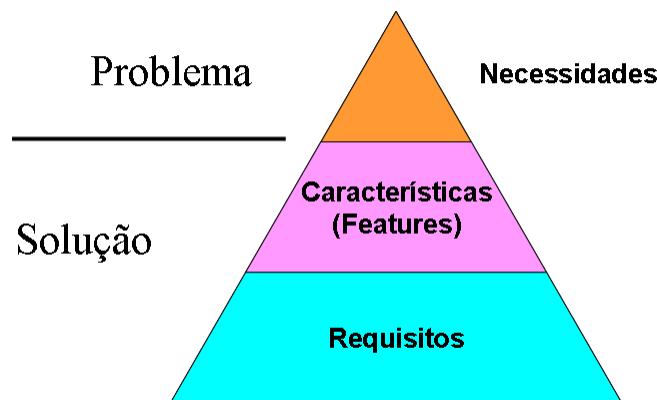


Figura 6. Enquanto as necessidades surgem no mundo dos problemas, características e requisitos definem a solução desejada.

Um problema é uma diferença entre uma situação percebida e uma situação desejada. Mais tarde vamos analisar vários tipos de problema. No momento, nos basta a noção que o problema incomoda o usuário (ou *stakeholder*) ao ponto dele considerar necessário investir alguma quantia de forma a evitar que o problema aconteça.

É comum ouvirmos o ditado “Em time que está ganhando, não se mexe”. Quando somos chamados para desenvolver um sistema, devemos imaginar imediatamente que, se alguém deseja mexer em sua organização, então é porque não está ganhando, pelo menos da maneira que supõe possível. Faz pouco sentido imaginar que alguém iria fazer um investimento de risco, e sempre há risco no desenvolvimento de software, sem que haja uma necessidade clara. No próximo capítulo veremos que um sistema precisa, além de um

objetivo funcional, metas de negócio, que visam determinar como será a solução para problemas específicos.

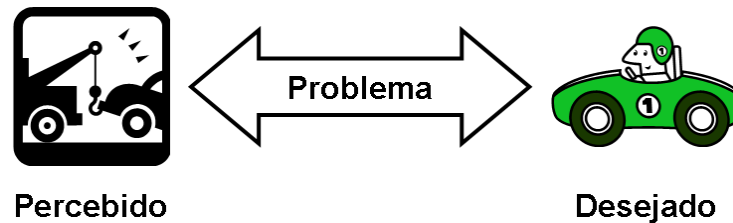


Figura 7. O problema é uma diferença entre uma situação percebida e uma desejada.

Muitas vezes o analista se depara com a necessidade de, antes de definir requisitos, definir propriamente qual o problema a ser tratado. Para isso, deve fazer com que os clientes cheguem a um acordo sobre qual é o verdadeiro problema, ou o problema mais importante. Faz pouco sentido construir um sistema se o verdadeiro problema de negócios não for endereçado pela solução planejada.

Para analisar o problema, é necessário que os usuários:

- Concordem com a definição do problema
- Entendam as causas do problema
 - O problema por trás do problema pode ser mais importante, sendo que o problema visto inicialmente é apenas um sintoma
- Identifiquem todos os stakeholders e usuários
- Definam a fronteira do sistema de solução
- Identifiquem as restrições impostas à solução

Para estabelecer o problema, é interessante criar uma sentença apropriada, como a seguinte:

- O problema de <descrição do problema>, afeta <stakeholders afetados> e resulta em <impacto nos stakeholders>. A solução <indicar a solução> Trará os benefícios de <lista dos principais benefícios>.

Um sistema pequeno certamente produzirá a solução para uma pequena quantidade de problemas, um sistema grande certamente atingirá uma grande quantidade de problemas. O ponto em questão é que deve existir alguma não conformidade, seja ela atual ou futura, para valer a pena o investimento, e o risco, de tentar um sistema novo.

1.5 Tipos de Requisitos

Vários autores dividem os requisitos de formas diferentes, porém quase todos concordam que existem dois tipos básicos de requisitos: os requisitos funcionais e os requisitos não funcionais.

I.5.1 Requisitos Funcionais e de Informação

Um **requisito funcional** representa algo que o sistema deve fazer, ou seja, uma função esperada do sistema que agregue algum valor a seus usuários. Exemplos típicos incluem a emissão de relatórios e a realização e manutenção de cadastros.

Um requisito funcional normalmente é escrito de uma forma específica, indicando que o sistema deve fazer algo, como em:

- O Sistema deve emitir o histórico escolar.

Também é possível que um requisito funcional seja uma especificação mais detalhada, impondo limites ou exigências a outro requisito, com em:

- O CR será calculado a partir da média ponderada das notas em cada curso, sendo que o peso de cada nota será a quantidade de créditos acreditada ao curso.

Existem muitas formas de levantar os requisitos funcionais de um sistema. Como veremos mais tarde, **os eventos essenciais definem todos os requisitos funcionais do sistema**, dado que a função dele é responder a todos os eventos. Apesar de não ser fácil levantar corretamente os requisitos funcionais, a metodologia essencial fornece um arcabouço eficiente para essa tarefa, que é perfeitamente adequado para sistemas de informação.

Requisitos de Informação representam a informação que o cliente deseja obter do sistema. Requisitos de informação também são atendidos por eventos. Muitas vezes o cliente expressa requisitos de informação de forma funcional. Outras vezes o cliente está preocupado em conseguir uma informação, mas não sabe como fazê-lo na forma de um requisito funcional. Em todo caso, sempre nos preocuparemos em levantar todos os requisitos de informação, pois eles representam as respostas fundamentais do sistema.

I.5.2 Requisitos não funcionais

Requisitos não funcionais falam da forma como os requisitos funcionais devem ser alcançados. Eles definem propriedades e restrições do sistema. Muitos requisitos não funcionais são também requisitos de qualidade, como exigências de desempenho e robustez. Outros são restrições ou exigências de uso de uma ou outra tecnologia.

Porém, muitas vezes não é só difícil descobrir quais são os requisitos não-funcionais, mas também produzir uma especificação do sistema que possa ser cumprida em custo razoável e prazo hábil de forma a atender os usuários. Um exemplo típico seriam dois requisitos não funcionais que, genericamente, são opostos: velocidade e transportabilidade. Ora, para fazer um software muito veloz você precisa adaptá-lo especificamente para o ambiente onde ele está funcionando. Para fazer um software transportável, você precisa implementá-lo de forma a funcionar no maior número possível de ambientes. Normalmente esses dois requisitos se relacionam de forma inversa e para implementá-los simultaneamente é necessário um grande investimento de recursos.

Existem muitas formas de se dividir os requisitos não funcionais, a figura a seguir apresenta uma dessas formas, apenas para deixar clara a quantidade de fatores que podem envolver um requisito não funcional, mas sem considerá-la melhor ou pior que outras.

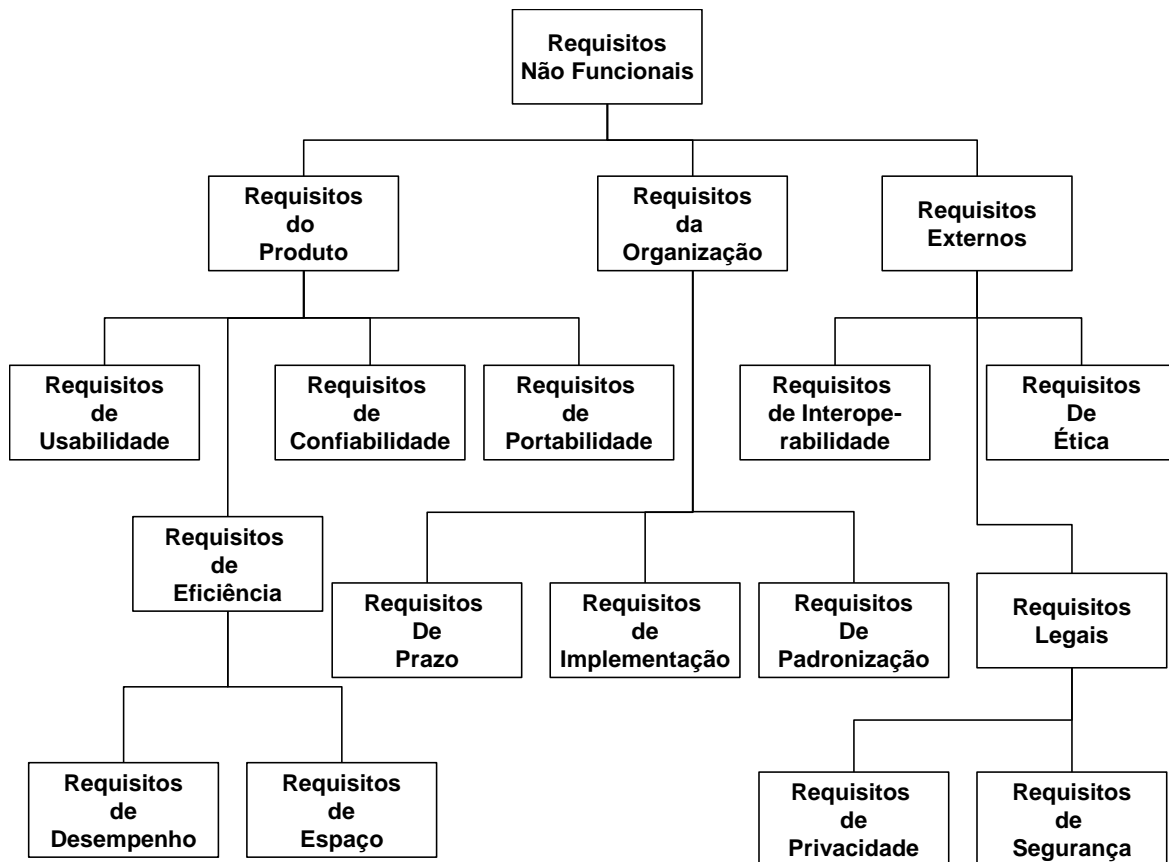


Figura 8. Tipos de requisitos funcionais, adaptado de Ian Sommerville, Software Engineering.

Esse texto tem como foco requisitos funcionais e requisitos de informação. Ainda não existe uma forma plenamente aceita de tratar requisitos não funcionais, mas o leitor interessado poderá encontrar diferentes métodos, a maioria incipiente, na literatura de Engenharia de Software.

Alguns requisitos não funcionais, quando negados, geram um “anti-requisito” que nunca seria pedido por um cliente. Por exemplo, um requisito não funcional comum é que o software seja “confiável”. Ninguém pediria para ser construído um software “não confiável”, porém diferentes clientes estão interessados em diferentes níveis de confiabilidade e diferentes formas de certificar esse nível. Deve ser levado em conta que quanto mais esforço é colocado para se alcançar um requisito, maior é o custo agregado ao projeto e isso servirá como referência para o cliente escolher o grau de confiabilidade que deseja. Por isso um requisito deve ser **verificável**. O requisito “robusto”, por exemplo, pode ser medido por meio do MTBF⁴ do sistema.

1.5.3 Outros tipos de Requisitos

James e Susan Robertson [B9] identificam mais três tipos de requisitos: restrições de projeto, temas de projeto e impulsionadores de projeto.

⁴ Tempo médio entre falhas, do inglês *Medium Time Between Failures*.

- As **restrições de projeto** são os limites impostos ao sistema para que funcione no seu ambiente operacional. Estas restrições tanto podem ser técnicas, envolvendo necessidades ou disponibilidades de fatores como hardware, software, rede e interoperabilidade com outros sistemas, quanto podem ser ligadas ao negócio. Restrições podem ser vistas como requisitos não funcionais de certa forma especiais, pois são limitantes.
- Os **impulsionadores de projeto** são as forças do negócio que fazem o projeto acontecer. Podem ser considerados requisitos na medida em que são os geradores originais dos requisitos funcionais e não funcionais. Tanto o objetivo inicial do sistema como todos os nele interessados (*stakeholders*) são impulsionadores.
- **Fatores Críticos de Sucesso** servem para completar o quadro geral de todos os fatores que influenciarão o sucesso ou o fracasso do projeto.

I.6 Descrevendo Requisitos

Normalmente as especificações de requisitos são escritas em linguagem natural (inglês ou português, por exemplo). O problema é que a forma como falamos e normalmente escrevemos é bastante ambígua. Isso exige que adotemos algumas técnicas básicas, principalmente um formato padronizado, um estilo de linguagem e uma organização que facilite a manipulação do conjunto de requisitos.

Algumas dicas para escrever requisitos são [B10]:

- Use sentenças e parágrafos curtos.
- Use a voz ativa.
- Use verbos no futuro⁵.
- Use os termos de forma consistente e mantenha um glossário.
- Para cada requisito, avalie se a partir de sua definição é possível determinar se ele está pronto ou não.
- Garanta que todos os requisitos são verificáveis imaginando (e possivelmente documentando) uma forma de fazê-lo.
- Verifique requisitos agregados (termos como “e” e “ou” são uma boa indicação) e divida-os.
- Mantenha um nível de detalhe único em todos os requisitos.

I.6.1 Exemplos de Requisitos

A seguir listamos alguns exemplos de requisitos bem descritos, seguindo o estilo de exemplos originais de Karl Wiegers⁶:

⁵ Alguns autores sugerem que o verbo esteja no presente. Na verdade, isso pode causar alguma confusão no leitor de um sistema a ser feito, pois a leitura dará a idéia que o sistema já faz. Além disso, pode haver a confusão entre o sistema a ser feito com o sistema a ser substituído.

⁶ Software Requirements Specification for Cafeteria Ordering System, Release 1.0

- O sistema deverá permitir que um paciente marque uma consulta.
- O sistema deverá confirmar que a consulta foi aceita pelo paciente.
- O sistema deverá permitir que um condômino solicite a segunda via de sua conta condominial.
- O sistema deverá permitir um aviso ao condômino que não pagar sua no prazo correto.
- O sistema deverá permitir que um fornecedor cadastre um produto no catálogo.
- O sistema deverá informar ao sistema de estoques que um produto foi vendido.

Todas as sentenças acima usam, na verdade, uma expressão do tipo “deverá”, tradução do inglês “shall”, que é uma forma tradicional do americano definir uma ordem.

Talvez as seguintes descrições sejam mais adequadas a nossa língua:

- O sistema permitirá que um paciente marque uma consulta.
- O sistema confirmará ao paciente que a consulta foi marcada.
- O sistema permitirá que um condômino solicite a segunda via de sua conta condominial.
- O sistema permitirá um aviso ao condômino que não pagar sua no prazo correto.
- O sistema permitirá que um fornecedor cadastre um produto no catálogo.
- O sistema informará ao sistema de estoques que um produto foi vendido.

Também devemos notar que as entradas do usuário são descritas como “deverá permitir”. Alguns autores, como o próprio Wiegers, admitem que uma especificação de requisito exija algo do usuário, como em:

- O paciente deverá especificar qual a especialidade médica que deseja.

Essa abordagem não é adequada, pois está fazendo uma exigência ao usuário e não ao sistema. Uma abordagem mais apropriada seria:

- O sistema exigirá que o usuário especifique a especialidade médica que deseja.

Desta forma deixamos claro que:

1. O requisito é do sistema
2. Cabe ao sistema exigir do usuário a resposta
3. Não cabe ao usuário saber o que fazer, mas sim ao sistema saber o que o usuário tem que fazer.
4. A especificação de requisitos do software não requer nada ao usuário.

Em conclusão, uma **especificação de requisitos só deve exigir funcionalidade do sistema sendo definido.**

Entendemos que muitas vezes um sistema faz exigências ao ambiente. Essas exigências podem algo como “o usuário deve falar inglês” ou “o computador deve ser do compatível com chips XYZ/2001”. Todas as exigências que o sistema faz devem ser documentadas. Uma localização específica do documento para isso é a seção de suposições ou dependências, mas demandas que o sistema faz não podem ser vistas **realmente** como um requisito do sistema, apesar de possivelmente definir características do sistema.

I.6.2 Documentando os requisitos

Requisitos de projeto podem ser descritos de várias formas. O método *Volere* propõe as seguintes informações [B11]:

- **Número identificador,**
 - para facilitar a discussão, identificamos todos os requisitos unicamente.
- **Tipo**
 - Classificando-o como funcional, não funcional,...
- **Evento que o atende**
- **Descrição**
 - Sentença que descreve o evento
- **Justificativa**
- **Fonte do requisito**
 - A pessoa ou o grupo que o originou
- **Critério de aceitação**
 - Uma medida que possa ser usada para garantir que o requisito foi alcançado.
- **Satisfação do usuário**
 - Um grau, de 1 (nenhum interesse) a 5 (extremamente satisfeito), por exemplo, indicando a satisfação do cliente se esse requisito for alcançado.
- **Insatisfação do usuário**
 - Um grau, de 1 (nenhum interesse) a 5 (extremamente insatisfeito), por exemplo, indicando a satisfação do cliente se esse requisito não for alcançado.
- **Dependências**
 - Referências a outros requisitos que dependem de alguma forma desse requisito

- **Conflitos**
 - Referência aos requisitos que de alguma forma conflitam com esse
- **Material de apoio**
- **Listagem de material de apoio para atender esse requisito**
- **Histórico**
- **Documentação da criação e das mudanças efetuadas**

O método *Volere* ainda propõe que os requisitos sejam levantados em cartões padronizados. O uso de cartões facilita a discussão, auxiliar a documentação, cria um foco (se discute apenas o que está no cartão sendo tratado) em entrevistas e reuniões, e permite a ordenação manual. Maiores detalhes sobre o método podem ser encontrados na página Web da Atlantyc Systems Guild⁷.

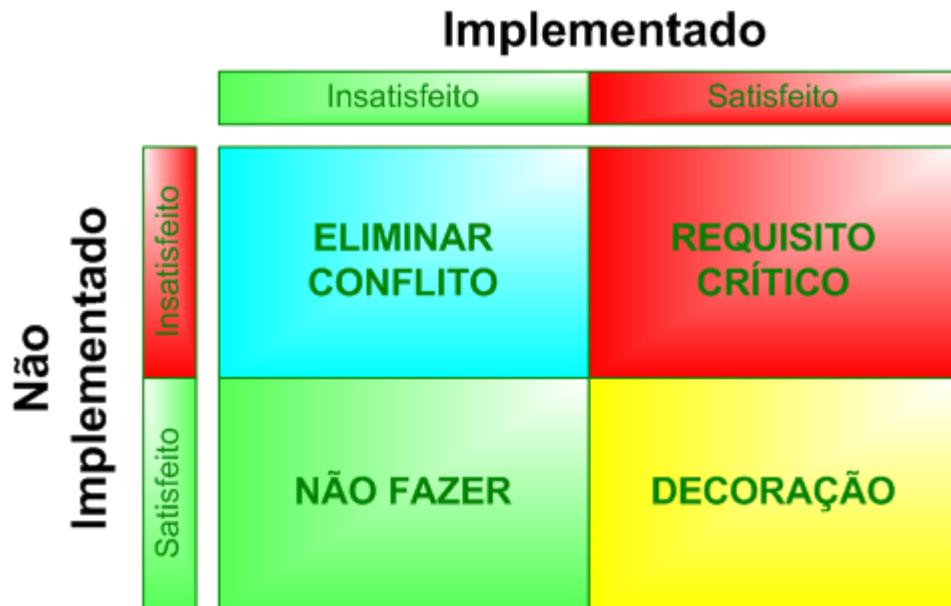


Figura 9. Algumas necessidades relativas as satisfação ou insatisfação do usuário quanto a implementação ao não do requisito.

I.6.3 Dependência de Requisitos

É importante notar que os requisitos não são independentes uns dos outros. Muitos requisitos só podem ser implementados se outros requisitos forem implementados antes. Por exemplo, é impossível fazer um relatório de vendas sem que se cadastrem as vendas previamente no sistema. Uma das atividades mais importantes da gerência de requisitos é manter esse relacionamento de dependência, que influenciará em todo desenvolvimento e Processo do sistema.

⁷ <http://www.volere.co.uk/template.htm>

Para isso existem algumas soluções possíveis. Uma delas é manter uma tabela de dependência de requisitos, outra manter um banco de dados de requisitos que inclua relações de dependência. Existem alguns produtos no mercado especializados na gerência de requisitos.

I.6.4 Priorizando Requisitos

Existe uma tendência grande de o sistema crescer muito durante a análise. Principalmente se entrevistamos um grande número de pessoas, existe uma facilidade natural das pessoas para propor novas funcionalidades para um sistema que ainda não existe, por imaginarem alguma utilidade nessas funções propostas.

Assim, muitas vezes nos vemos envolvidos com uma quantidade de requisitos tão grande que é óbvio que o sistema a ser feito não poderá ser entregue no prazo ou pelo custo combinado, ou que se pensava em combinar.

Nesse caso, algumas técnicas podem ser utilizadas para caracterizar o que deve ser realmente feito ou, pelo menos, em que ordem as coisas devem ser feitas.

A primeira técnica disponível é associar a cada requisito do sistema uma importância. Uma escala de três ou cinco valores é adequada para isso, como em: “Imprescindível para o sucesso do sistema”, “Funcionalidade Importante, mas podemos esperar algum tempo”, “Ajudaria ter, mas é possível viver sem essa funcionalidade”, “Benefícios mínimos”, “Desnecessário”.

A segunda técnica disponível é planejar o sistema para ser entregue em várias versões, mesmo que nem todas as versões estejam incluídas nesse contrato, e pedir para o cliente determinar que funcionalidades devem aparecer em cada versão. Nesse caso pode ser interessante dar um peso ou custo para cada requisito, de modo que o cliente possa “controlar seus gastos”.

Uma terceira técnica disponível é dar uma moeda virtual para o cliente, por exemplo, 1000 dinheiros, e pedir para ele distribuir quanto pagaria por cada função, priorizando no desenvolvimento aquelas funções que o cliente decidir pagar mais.

Todas essas técnicas, porém, ficam dependentes de uma outra priorização importante dos requisitos: a priorização por dependência.

Devem ser levados em conta os vários fatores que influenciam nessa determinação de prioridades, entre eles os citados por [B13]:

- Diminuir o custo da implementação
- Valor para o comprador
- Tempo para implementar
- Facilidade técnica de implementar
- Facilidade do negócio para implementar
- Valor para o negócio
- Obrigação por alguma autoridade externa

I.6.5 Questionando os requisitos

Em vários momentos do projeto, é importante tratar a lista de requisitos como uma lista a ser abertamente questionada. Para isso devemos analisar as características desejadas dos requisitos (Seção I.3.1) e verificar, para cada requisito, se elas são verdade. Além disso, devemos também analisar de que forma os requisitos respondem as perguntas 5W2H (Who, When, Where, What, Which, How, How much). Se o requisito passar por todos os questionamentos, temos um requisito muito bom. Se falhar em alguns, pode ser que não seja o momento ainda no projeto ou que a pergunta não seja pertinente, porém deve ser analisado cada caso.

Os maiores problemas sempre serão encontrados na ambigüidade dos requisitos. Qualquer ambigüidade é um risco, porém no início do projeto a ambigüidade é necessária, pois decisões importantes ainda não foram tomadas. Cabe ao analista saber “em que pé está” o projeto e qual o nível de detalhe adequado aos requisitos.

I.7 A Especificação de Requisitos

Uma especificação de requisitos de sistema é um documento que reúne os requisitos definidos e aceitos por todos os interessados naquele sistema.

Os assuntos básicos que devem ser tratados em uma especificação de requisitos são, segundo um padrão IEEE são⁸:

- Funcionalidade, ou o que o software tem que fazer?
- Interfaces externas, com quem ou com que sistemas o software interage?
- Desempenho, qual a velocidade, disponibilidade, tempo de resposta das várias funções, etc.?
- Atributos (de qualidade), quais são as considerações de portabilidade, correção, manutenibilidade, segurança, etc.?
- Restrições de projeto impostas a implementação, como padrões a serem seguindo, políticas de integridade da base de dados, limites dos recursos disponíveis, ambiente de operação, etc.?

Segundo o mesmo padrão, um sumário adequado a uma especificação de requisitos seria:

Sumário

1. Introdução

1.1 Objetivo

1.2 Escopo

1.3 Definições, acrônimos e abreviações

1.4 Referencias

⁸ IEEE, IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications

1.5 Visão Geral
2. Descrição Geral
2.1 Perspectiva do Produto
2.1.1 Interfaces do Sistema
2.1.2 Interfaces do Usuário
2.1.3 Interfaces de Hardware
2.1.4 Interfaces de Software
2.1.5 Interfaces de Comunicação
2.1.6 Memória
2.1.7 Operações
2.7.8 Adaptações necessários no ambiente
2.2 Funções do Produto
2.3 Características do Usuário
2.4 Restrições
2.5 Suposições e Dependências
3. Requisitos Específicos
Apêndices
Índices

Figura 10. Um roteiro para a especificação de requisitos segundo a IEEE⁹.

Entre os apêndices, é interessante colocar o modelo de dados e quaisquer outros modelos que possam ser necessários.

I.8 Métodos de Elicitação de Requisitos

A elicitação de requisitos é o levantamento, registro e validação [B14] das expectativas dos diversos interessados no sistema, seguido da consolidação e validação dessas expectativas em requisitos formais. Contemplar essas diferentes visões implica projetar interesses divergentes e conciliá-los.

Para levantar requisitos é necessária a interação entre aqueles que conhecem os requisitos ou a necessidades dos usuários e stakeholders e os desenvolvedores. É um processo iterativo de comunicação, onde, por aproximações sucessivas, o desenvolvedor constrói um modelo aceito pelos usuários.

⁹ idem

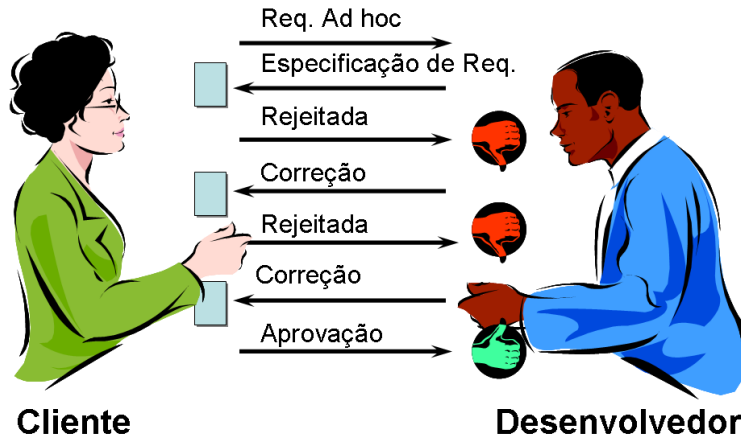


Figura 11 A elicitação de requisitos é um processo iterativo onde, por aproximações sucessivas, o desenvolvedor consegue a aprovação de uma especificação de requisitos.

O mesmo acontece na investigação que um analista tem que fazer com o cliente em busca do requisito. Inicialmente o cliente será ambíguo, generalista e paulatinamente, com a ajuda do analista, deverá chegar a uma especificação precisa do que deseja.

Uma receita geral para o levantamento de requisitos pode ser dada em 5 passos:

1. Identificar quem fornecerá os requisitos
2. Levantar lista de desejos para estas pessoas
3. Refinar cada lista de desejos até que seja auto-consistente
4. Integrar as listas de desejos de forma que sejam consistentes
5. Determinar os requisitos não funcionais.

Apesar de tudo, não é uma tarefa fácil levantar os requisitos. Muitos problemas podem acontecer. É comum que *stakeholders* não saibam exatamente o que querem ou não saibam articular propriamente suas idéias, especialmente quando o desenvolvedor não possui o linguajar típico da área de aplicação (jargão). Muitas vezes, também, os desenvolvedores pensam entender melhor do problema que seus clientes, propondo supostas “melhorias” que afetam custo e aumento o risco dos sistemas propostos.

1.8.1 Processo de elicitação de requisitos

A elicitação de requisitos pode ser modelada em um processo como o proposto por Christel e Kang, apresentado nas figuras a seguir.

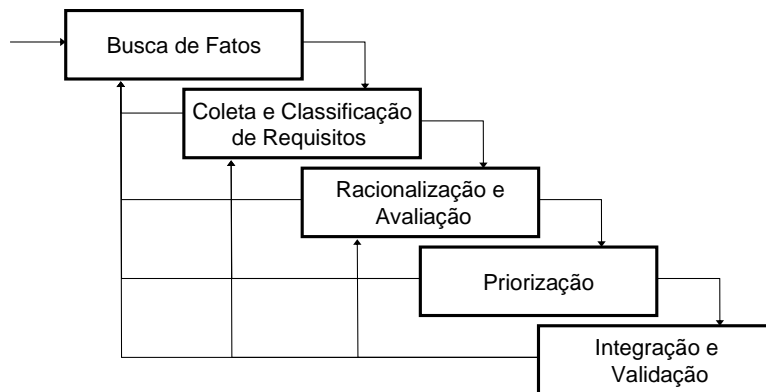


Figura 12. O processo de elicitação de requisitos, adaptado de Christel e Lang [B14].

Tarefas da Elicitação de Requisitos		
Atividade	Tarefas orientadas ao usuário	Tarefas orientadas ao desenvolvedor
Busca de Fatos	Identificar grupos relevantes através de múltiplos níveis da organização. Determinar os contextos operacional e do problema, incluindo a definição dos modos operacionais, objetivos e cenários de missão como apropriados. Identificar sistemas similares. Realizar análise de contexto.	Identificar especialistas do domínio da aplicação e de desenvolvimento. Identificar modelo de domínio e modelo de arquitetura. Conduzir pesquisas tecnológicas, para mais tarde fazer estudo de viabilidade e análise de risco. Identificar custos e restrições à implementação impostas pelo patrocinador.
Coleta e Classificação dos Requisitos	Levantar a lista de desejos de cada grupo.	Classificar a lista de acordo com funcionais, não funcionais, restrições de ambiente, restrições de projeto e ainda de acordo com as partições definidas pelo modelo de domínio e pelo paradigma de desenvolvimento.
Racionalização e Avaliação	Responder questões da forma "Por que você precisa de X?", a partir de raciocínio abstrato. Isso auxilia a transformar o raciocínio das questões sobre "como?" para as questões sobre "o quê?".	Realizar uma análise de riscos, investigando técnicas, custos, prazos e incluindo análise de custos e benefícios e viabilidade baseado na disponibilidade da tecnologia.
Priorização	Determinar funcionalidades críticas para a missão.	Priorizar requisitos baseados em custo e dependência. Estudar como o sistema pode ser implementado de forma incremental, investigando modelos arquiteturais apropriados.
Integração e Validação	Resolver a maior quantidade possível de pontos em aberto. Validar que os requisitos estão concordando com os objetivos iniciais. Obter autorização e verificação para passar ao próximo passo de desenvolvimento (e.g. a demonstração e a validação).	Resolver conflitos e verificar consistência.

Tabela 3. Tarefas do processo de elicitação de requisitos, adaptado de Christel e Kang [B14].

I.9 Métodos Simples de Elicitação de Requisitos

Existem muitas técnicas avançadas de elicitação de requisitos que não cabem no contexto desse livro. Aqui trataremos de duas técnicas básicas que coleta de informações: a entrevista e reuniões de JAD.

I.9.1 A entrevista

Entre as técnicas mais importantes de elicitação de requisitos está a entrevista. Ela está constantemente presente dentro de outras técnicas porque, quase sempre, a Elicitação de Requisitos – em algum ponto - exige comunicação direta com os usuários e outros interessados e a forma básica de comunicação, seja de que forma esteja disfarçada, é a entrevista.

A entrevista procura explicitar o pensamento do entrevistado a respeito das suas relações com seu universo em determinada instância, tanto como indivíduo quanto como profissional, revelando o conhecimento do entrevistado sobre as pessoas, objetos, fatos e procedimentos com os quais interage.

Os objetivos são os mais diversos, por exemplo: estabelecer expectativas do consumidor, verificar níveis de satisfação e necessidades atuais e futuras; estudar tendências na satisfação do cliente ao longo do tempo ou avaliar programas em andamento.

O primeiro passo de uma entrevista é determinar, entre outros aspectos: seus propósitos ou objetivos; a informação necessária, e de quem obter, para alcançar esses objetivos e os recursos disponíveis para implementar e manter o processo de pesquisa. Outros fatores merecem destaque: precisão na determinação da amostra; abrangência e relevância do conteúdo da pesquisa e, essencial, a validação. Os resultados da entrevista são sumariados em um relatório interpretativo que inclui relato sobre os achados e as recomendações mais importantes. A análise pode ser qualitativa ou quantitativa. Normalmente, as entrevistas abertas estão no primeiro caso, enquanto que os questionários são analisados quantitativamente.

A responsabilidade do entrevistador também é grande. Normalmente, além das entrevistas, é ele quem integra as diferentes interpretações, objetivos, metas, estilos de comunicação, e o uso de terminologia. Há também outras tarefas complexas como decidir a oportunidade ou não de incluir certo tipo de informação no conjunto inicial de requisitos. Finalmente, entrevistar e integrar toma tempo. Como os requisitos são voláteis, quanto mais longo o processo mais facilmente os requisitos deixam de atender as necessidades e expectativas dos interessados. Todos esses encargos recomendam que o analista conheça tanto as técnicas de desenvolvimento quanto o domínio no qual se insere.

Existem vários tipos de entrevistas. Durante o processo de análise, elas normalmente seguem o seguinte processo:

1. Introdução inicial
2. Familiarização com o ambiente
3. Busca de fatos
4. Verificação de informações conseguidas de outras formas
5. Confirmação de informações conseguidas com os candidatos

6. Acompanhamento, amplificação ou clarificação de entrevistas anteriores.
7. Outra grande variável da entrevista é o seu objetivo, entre outros:
8. Levantar informações sobre a organização
9. Levantar informações sobre uma função de negócio
10. Levantar informações sobre um processo ou atividade
11. Descobrir problemas
12. Verificar fatos previamente levantados
13. Fornecer informação
14. Obter dicas para entrevistas futuras

Há várias formas de entrevista, entre elas: entrevista por questionário; entrevista aberta; entrevista estruturada.

I.9.1.1 Entrevista por Questionário

O questionário é muito usado como técnica de entrevista, principalmente em pesquisas de mercado e opinião. Exige preparação elaborada. Alguns aspectos particulares do processo merecem destaque: emprego de vocabulário adequado para o público entrevistado; inclusão de todos os conteúdos relevantes e de todas as possibilidades de respostas; cuidado com os itens redundantes ou ambíguos, contendo mais de uma idéia ou não relacionados com o propósito da pesquisa; redação clara; execução de testes de validade e confiabilidade da pesquisa.

Há uma tensão não resolvida entre o uso do questionário como um evento interativo ou como instrumento neutro de medida. Por um lado, como entrevista, é visto como uma interação. Por outro lado, no interesse de torná-lo um instrumento, muitos recursos da interação existentes na conversação não são permitidos, suprimindo recursos de medida de incertezas de relevância e interpretação.

Dificuldade importante é o fato das palavras possuírem significados diferentes para pessoas diferentes em diferentes contextos. Em interações normais essas questões de interpretação são negociadas entre os participantes, mas em entrevistas com questionários o treinamento e o método utilizados proíbem essa negociação. Além disso, há necessidade do uso de técnicas específicas – nem sempre do conhecimento dos projetistas - para a construção e aplicação de questionários. A menor ambigüidade é uma das principais vantagens da entrevista via questionário.

Para gerar bons itens de questionário, devemos:

- Evitar palavras ambíguas ou vagas que tenham significados diferentes para pessoas diferentes;
- Redigir itens específicos, claros e concisos e descartar palavras supérfluas;
- Incluir apenas uma idéia por item;
- Evitar itens com categorias de respostas desbalanceadas;
- Evitar itens com dupla negação;

- Evitar palavras especializadas, jargões, abreviaturas e anacronismos;
- Redigir itens relevantes para a sua pesquisa;
- Evitar itens demográficos que identifiquem os entrevistados

I.9.1.2 Entrevista Aberta

Esse tipo de entrevista evita muitos dos problemas dos questionários, porém também cria outros. O entrevistador formula uma questão e permite que o entrevistado responda como quiser. O entrevistador pode pedir mais detalhes, mas não determina os termos da entrevista. Permanecem, entretanto, as questões: as perguntas podem ser respondidas? A resposta faz parte do repertório normal do discurso do entrevistado? Há muitas coisas que as pessoas sabem fazer, mas tem dificuldade de descrever, como há também o conhecimento tácito, que é de difícil elicitación.

Os benefícios das entrevistas abertas são: a ausência de restrições, a possibilidade de trabalhar uma visão ampla e geral de áreas específicas e a expressão livre do entrevistado.

Há desvantagens também. A tarefa de entrevistar é difícil e desgastante. O entrevistador e o entrevistado precisam reconhecer a necessidade de mútua colaboração ou o resultado não será o desejado. Há falta de procedimentos padronizados para estruturar as informações recebidas durante as entrevistas. A análise da informação obtida não é trivial. É difícil ouvir e registrar simultaneamente; principalmente, porque há fatos que só tomam importância depois de outros fatos serem conhecidos, e aí ele já não foi registrado. Daí a importância da gravação e da respectiva transcrição; fica mais fácil selecionar e registrar o que é relevante e validar com o entrevistado.

São exigências para o relacionamento entre os participantes de uma entrevista: respeito ao conhecimento e habilidade do especialista; percepção de expressões não verbais; sensibilidade às diferenças culturais; cordialidade e cooperação.

I.9.1.3 Entrevista Estruturada

A entrevista estruturada extrai informações sobre perguntas específicas. Nesse tipo de entrevista, é importante entrevistar a pessoa certa. É uma boa técnica para ser usada após uma pesquisa com questionário, quando é possível selecionar, entre as respostas, as partes interessadas com maior potencial de geração de outras informações. Suas vantagens são: Respostas diretas, com menos ambigüidade, informação detalhada. Sua desvantagem básica é que as questões relevantes precisam ser identificadas com antecedência.

I.9.1.4 O processo da entrevista

O processo de entrevista não se resume ao ato específico da entrevista. Na verdade ele começa muito antes e acaba muito depois. O processo normal da entrevista inclui:

1. Determinação da necessidade da entrevista
2. Especificação do objetivo da entrevista
3. Seleção do entrevistado
4. Marcação da entrevista

5. Preparação das questões ou do roteiro
6. A entrevista propriamente dita
7. Documentação da entrevista, incluindo os fatos e a informação conseguida durante a entrevista.
8. Revisão da transcrição da entrevista com o entrevistado
9. Correção da transcrição
10. Aceitação por parte do entrevistado
11. Arquivamento

I.9.1.5 Preparando a entrevista

A preparação é uma necessidade básica da entrevista. Não só precisamos preparar a entrevista propriamente dita, mas também preparar a nós mesmos, como entrevistadores, e ao entrevistado.

Uma entrevista deve ter um **objetivo**. As perguntas ou o roteiro devem ser coerentes. Para isso é importante a determinação desse objetivo,. O entrevistado deve ter noção clara da finalidade da entrevista e perceber sua utilidade. Isso se faz por meio de palestras, textos de divulgação e, principalmente, se explicando ao entrevistado, no início da entrevista, seu objetivo e importância.

Muitas vezes esse objetivo não é específico, principalmente na fase inicial do projeto. Mas deve ser claro, isso é, quando expressado deve permitir que entrevistador e entrevistado compreendam o motivo da entrevista. Assim, no início do projeto os objetivos podem ser: “Conhecer o ambiente de trabalho”, “Levantar expectativas iniciais dos usuários”. Já com o passar do tempo do projeto o objetivo se torna mais detalhado, como em: “Levantar os documentos utilizados no processo de compra” ou “Avaliar as telas relativas ao cadastro de bens”.

A escolha do entrevistado é o segundo aspecto importante. Devem ser escolhidas as pessoas que permitam obter no final das entrevistas uma visão clara e o mais completa possível do problema, das diversas formas de analisá-lo e solucioná-lo. Nunca se deve tratar um problema a partir de um único nível funcional, nem de uma única visão organizacional, pois estaríamos correndo o sério risco de obter uma visão distorcida. Devemos lembrar que o sistema afetará todos os níveis funcionais e departamentos da instituição.

Dependendo do tipo de entrevista, será necessário um roteiro ou um questionário. No início da análise os roteiros levam a execução de entrevistas abertas, no final geralmente temos entrevistas por questionários. Entrevistas estruturadas são preparadas principalmente para esclarecimento de processos e atividades.

Todos os roteiros e questionários devem seguir um modelo padrão, incluindo a apresentação e a conclusão da entrevista. Quanto maior o número de entrevistadores, maior a importância de seguir um padrão.

Outros aspectos fundamentais a serem preparados são:

- A linguagem

- A coerência das perguntas
- A programação dos horários

É importante estar preparado para a linguagem a ser usada na entrevista. Nisso influenciam vários fatores, como nível cultural do entrevistado, terminologia do trabalho, jargão da área, etc. Devemos evitar ao máximo usar os nossos termos técnicos e aproveitar ao máximo a oportunidade de aprender os termos técnicos do entrevistado. Se necessário, ler um pequeno texto esclarecedor sobre a área e, sempre, ler o glossário do projeto. O entrevistador deve sempre esclarecer com o entrevistado todas as dúvidas quanto ao vocabulário utilizado no ambiente onde o sistema será implantado.

Marque a entrevista com antecedência, com confirmação de data, hora, duração e local por todas as partes. As seguintes regras devem ser observadas quanto ao horário;

- As entrevistas devem ter 30, 60 ou 90 minutos e, no máximo, duas horas.
- As entrevistas iniciais podem ser mais longas, enquanto as entrevistas finais devem ser mais rápidas.
- Evite horários perto da hora do almoço ou no final de expediente, ou em uma tarde de sexta-feira ou véspera de feriado.
- Obtenha o telefone do entrevistado, para poder avisá-lo de sua ausência em caso de urgência.
- Chegue sempre 10 minutos adiantado e esteja preparado para esperar e para ter que encerrar a entrevista mais cedo, principalmente com a alta gerência.
- Se possível, caso a entrevista seja mais curta que o combinado, marque imediatamente a sua continuação.
- Quanto ao material necessário para uma entrevista, além do roteiro:
- Prepare e teste o equipamento, principalmente um gravador. Atualmente existem bons gravadores digitais a preços razoáveis no mercado.
- Tenha pelo menos 2 horas de gravação e um jogo de pilhas extras.
- Tenha um caderno de anotações (é melhor que um bloco) reservado para o projeto. Canetas de várias cores, lápis, borrachas.

I.9.2 Realizando a entrevista

O objetivo normal de uma entrevista é conseguir informações do entrevistado, para isso devemos fazer não só que o usuário fale, mas também que ele pense. É importante para o entrevistador não assumir nada e não fazer pré-julgamentos, caso contrário correrá o risco de fazer uma entrevista “viciada”.

O entrevistador deve manter o controle o assunto da entrevista. Não deixe o entrevistador mudar de assunto ou tergiversar, mantendo suas perguntas direcionadas para o objetivo da entrevista.

As duas principais armas do entrevistador são a pergunta e o silêncio. Para perguntar devemos ter consciência do tipo de pergunta que escolhemos. Se quisermos que

o usuário explique algo, então devemos utilizar uma pergunta aberta. Isso é muito comum em entrevistas abertas no início da análise.

O importante é fazer o usuário pensar, para isso, o entrevistador deve evitar perguntas que contenham a própria resposta ou as que podem ser respondidas apenas com um sim ou não. As perguntas fechadas devem ser utilizadas para tirar dúvidas do entrevistador. Use questões começando com “quem”, “qual”, “quando”, “onde”, “porque” e “como” sempre que possível. Tente completar o ciclo (quem – qual – quando – onde – porque – como) para todos os assuntos.

Em dúvida, pergunta novamente de outra forma. O entrevistador deve pedir que processos complicados sejam explicados mais de uma vez, preferencialmente sob perspectivas diferentes. Desenhá-los em quadros brancos ou em papel pode ser uma boa solução para eliminar qualquer dúvida.

É importante estabelecer exemplos concretos para o que está sendo descrito pelo usuário. Também, em caso de uma dúvida, é melhor descrever um exemplo concreto (o que aconteceria se ...) do que uma dúvida abstrata. O entrevistador deve estar consciente que é muito difícil encontrar um entrevistado capaz de raciocinar plenamente de forma abstrata sobre um problema. Mesmo nesse caso, normalmente a forma abstrata se resume ao “caso perfeito”, sendo que as exceções são melhores explicadas com exemplos.

Não tenha pressa, não responda pelo entrevistado. Não se preocupe com a demora para responder ou o silêncio. O silêncio, inclusive, é uma boa tática para fazer o entrevistado continuar falando. Deixe o entrevistado pensar, olhe para ele curiosamente.

Antes de mudar de assunto, verifique sua compreensão, explicando de forma resumida o que acabou de ouvir. Isso permite ao entrevistado pensar e dar uma clarificação se necessário

Esteja atento para a ausência de críticas por parte do candidato. Isso pode ser causado pela falta de confiança do entrevistado em você ou porque o problema é constrangedor demais para ser tratado. O analista deve constatar esse fato no processo de análise, mas não durante a entrevista.

As interrupções causadas por fatores externos (telefone, pessoas que entram e que saem, etc.) podem ser importantes em um processo de entrevistas. Se uma entrevista for prejudicada por muitas interrupções, isso deve ficar no relatório da entrevista.

No relatório, também devemos separar o que é fato do que é opinião.

I.9.2.1 O Comportamento do Entrevistador

Esteja atento ao próprio comportamento. Lembre-se que não importa sua intenção ao fazer ou deixar de fazer algo, mas a interpretação que o entrevistado dará ou que fizer ou não fizer.

No passado era comum que consultores sempre se vestissem de terno, até mesmo apenas ternos escuros. A maioria das empresas hoje utiliza um código de vestimenta mais informal. A regra mais atual é que o entrevistador ou consultor tome cuidado para não provocar um grande desnível entre a sua roupa e a roupa do entrevistado ou cliente, se adaptando as normas de vestimenta do cliente (ou do mercado ao qual o cliente pertence).

Fisicamente, não faça movimentos desnecessários como bater o lápis na mesa, mexer as chaves no bolso, sacudir ou bater os pés, etc. Movimentos automáticos e cacoetes distraem o entrevistado e, além disso, são interpretados como falta de atenção. Não fume, mas também não evite que seu entrevistado fume. Não constranja o entrevistado comentando sobre os males do fumo. Não peça bebidas ou alimentos, como café, mas pode aceitar o que for oferecido. Se necessário, pode pedir água¹⁰.

Estabeleça um horário para a entrevista e o cumpra rigidamente. Devido aos constantes problemas de trânsito da cidade, e a necessidade de se identificar para seguranças e secretárias, o entrevistador deve sempre planejar chegar ao local com uma folga de tempo, algo em torno de 15 minutos.

Mantenha o interesse. Tome notas, mas não seja obsessivo, principalmente não interrompa o candidato para manter suas notas atualizadas. Se autorizado, grave a entrevista e a reveja mais tarde se necessário. Escute ativamente sem interromper. O entrevistado é que deve falar a maior parte do tempo.

Utilize um tom educado e cortês. Não seja engraçado, sarcástico ou depreciativo. Não faça comentários pejorativos ou preconceituosos. Não faça comentários sobre política e religião, ou outro tema controverso. Seja cordial, mas sem deixar de ser profissional. Pergunte e responda com cortesia e honestidade. Não dê opiniões particulares, mesmo quando pedido. A entrevista é o momento de levantar informações, não de emití-las.

Não dê a um entrevistado informações passadas por outros entrevistados. Educadamente, responda que não cabe a você decidir ou opinar.

Evite, de toda a forma, confrontar o entrevistado. Não torne a entrevista um interrogatório. Evite discutir, mesmo que não concorde com o usuário. Em caso de discussão, defina claramente o motivo do desacordo, seja ele motivado por fato ou por opinião. Utilize perguntas para restabelecer a comunicação em caso de desacordo. Se necessário, peça desculpas.

Se necessário elucidar algo que foi mal explicado, lembre-se que “você não entendeu” e que precisa de maiores detalhes. Se os detalhes não estiverem disponíveis, dê ao entrevistado a chance de enviá-los mais tarde.

Basicamente o entrevistador deve ser muito educado.

I.9.2.2 Roteiro Básico

1. Apresente-se ao entrevistado: “Olá, muito prazer, eu sou fulano-de-tal, responsável por parte do projeto XYZ”. Apresente seu cartão de visitas se for o primeiro encontro.
2. Informe ao entrevistado o motivo da entrevista e porque foi selecionado: “Estou aqui para levantar o funcionamento da sua área, e seu nome foi escolhido por ser o funcionário mais experiente” ou “Estou aqui para levantar o funcionamento da atividade X, que é de sua responsabilidade”.

¹⁰ As regras de etiqueta consideram a água a única coisa que pode ser pedida em qualquer situação.

3. Deixe clara a idéia que o conhecimento e as opiniões do entrevistado são importantes e serão úteis no processo de análise
4. Diga o que vai acontecer com a informação levantada
5. Garanta que o entrevistado lerá a transcrição da entrevista e terá a oportunidade de corrigi-la, garanta que nada será passado a outras pessoas sem a revisão e verificação do entrevistado.
6. Determine os assuntos confidenciais ou restritos a serem tratados na entrevista
7. Deixe claro que não haverá conseqüências negativas em função do resultado da entrevista
8. Solicite permissão para gravar a entrevista
9. Se autorizado, inicie a gravação com um texto de apresentação: “Entrevista realizada no dia X...”.
10. Faça a entrevista até faltarem 5 ou 10 minutos para o tempo determinado
11. Avise ao entrevistado que o tempo está acabando e pergunte se gostaria de adicionar alguma informação
12. Solicite ao candidato que responda as perguntas de conclusão
13. Conclua a entrevista de forma positiva, relatando brevemente o resultado positivo alcançado.
14. Se necessário, marque outra entrevista.
15. Entregue ao candidato o formulário de avaliação de entrevista e o envelope correspondente. Ensine-o a enviar a avaliação preenchida.
16. Despeça-se educadamente, agradecendo a atenção e o tempo dispensado.

Muitas vezes a entrevista é precedida por um bate-papo informal de apresentação. Tente manter essa conversação em um tempo mínimo razoável.

I.9.2.3 Documentando a Entrevista

A entrevista deve ser documentada logo após sua realização. Ao documentá-la rapidamente, estará garantindo que recuperará mais informação.

A documentação da entrevista deve fornecer a seguinte informação.

- A data, hora e local da entrevista.
- Nome do entrevistador
- Cargo do entrevistador
- Nome do entrevistado
- Função do entrevistado e a descrição desse cargo
- Se necessário, informações de background do entrevistado, como experiência no cargo ou com computadores.

- Organograma do entrevistado (superior imediato, colegas do mesmo nível, subordinados).
- O objetivo da entrevista
- Nomes e títulos de todos os outros presentes na entrevista
- Uma descrição completa dos fatos descritos e opiniões do entrevistado
- Opcionalmente, uma transcrição da entrevista, possivelmente expurgada das falas que não tinham relação com o assunto da entrevista.
- Todas as conclusões tiradas dos fatos e opiniões como apresentados
- Todos os problemas de negócio levantados durante a entrevista
- Exemplos de todos os relatórios, diagramas, documentos, etc., discutidos durante a entrevista.
- Todos os desenhos e diagramas feitos a partir ou durante a entrevista
- Qualquer comentário relevante feito pelo entrevistado
- Todos os números relevantes (quantidades, volume de dados, etc.) coletados durante a entrevista.

É importante notar que o relatório da entrevista deve ser aceito pelo entrevistado. É normal o entrevistado remover alguma coisa ou colocar algo a mais. O analista deve ficar atento aos motivos do usuário em fazer modificações.

Se houver discussão quanto à interpretação de algo e o analista achar essencial manter sua versão no relatório, deve também permitir que o entrevistado coloque sua versão.

I.9.2.4 As perguntas de conclusão

Ao final da entrevista, é importante realizar uma avaliação da percepção do entrevistado sobre a entrevista que acabou de ser realizada. Para isso é necessário que seja respondido um formulário, contendo perguntas como:

- Você acha que essa entrevista cobriu tudo que era necessário?
- Você acha que foram feitas as perguntas certas?
- Você acha que era a pessoa mais certa para responder essas perguntas?

I.9.3 JAD

Outra técnica importante de elicitación de requisitos, que merece um tratamento separado, é o JAD.

Muitas vezes, quando um grupo de informática entrega um sistema de informação aos seus clientes escuta a frase: “não era isso que eu queria!” Isto acontece porque os desenvolvedores não conseguiram levantar com os usuários suas verdadeiras necessidades.

Este problema de comunicação pode ter diversas causas: linguagem especializada de ambas as partes, desconhecimento da área de atuação pelos desenvolvedores,

preocupações com a tecnologia empregada ao invés das necessidades dos usuários, etc. Na fase inicial do projeto, a dificuldade de comunicação entre clientes e equipe de desenvolvimento é a principal causa de defeitos que serão encontrados no produto final.

Para enfrentar os problemas de comunicação entre desenvolvedores e usuários, foram desenvolvidos, ao final da década de 1970, vários métodos onde o desenvolvimento de sistemas é baseado na dinâmica de grupo.

Na forma tradicional de desenvolver sistemas os analistas entrevistam os usuários, um após outro, tomando notas que são mais tarde consolidadas e então validadas com o usuário, finalmente se transformando em um documento de análise.

O JAD, Joint Application Design, ou Método de Projeto Interativo, substitui as entrevistas individuais por reuniões de grupo, onde participam representantes dos usuários e os representantes dos desenvolvedores.

Quando o método é aplicado de forma correta, os resultados alcançados ultrapassam os objetivos imediatos da obtenção de informação dos usuários e cria um ambiente de alta sinergia na equipe, onde todos se sentem comprometidos com as soluções encontradas. Esse comprometimento permite que todos se considerem “proprietários” e “colaboradores” no desenvolvimento do sistema.

I.9.3.1 O Objetivo do Método

O objetivo do método é extrair informações de alta qualidade dos usuários, em curto espaço de tempo, através de reuniões estruturadas para alcançar decisões por consenso.

I.9.3.2 Os Componentes

O **líder de sessão** tem como tarefa número um conduzir o grupo para soluções de consenso. Esse líder de sessão age como **facilitador**, um servidor neutro dentro do grupo que não avalia nem contribui com idéias. A responsabilidade do facilitador é sugerir métodos e procedimentos que ajudem o grupo a concentrar energia em tarefas específicas, garantindo a todos os membros do grupo condições de participar.

O **documentador** é um auxiliar imparcial do líder de sessão, responsável pelo registro das decisões e especificações produzidas. Apenas as informações relevantes são documentadas, segundo orientação do líder de sessão.

O **patrocinador** detém a autoridade formal sobre as áreas afetadas pelo sistema, estabelecendo diretrizes e objetivos do projeto.

A **equipe** é responsável pelo conteúdo da sessão, representando as áreas envolvidas no projeto.

I.9.3.3 A Dinâmica

A base de trabalho é a equipe presente na reunião. Devemos combinar algumas “regras de jogo”, de modo a alcançar o máximo de produtividade. É natural que em um grupo de 15 pessoas surjam discussões, conversas paralelas, interrupções, etc. Em respeito ao tempo precioso dos participantes vamos é necessário estabelecer um código de cooperação.

I.9.3.4 O Ambiente

O Ambiente físico da reunião é fundamental para a produtividade dos trabalhos. Os seguintes aspectos devem ser considerados:

Os participantes devem estar organizados de forma a poderem se olhar e olhar para o líder de sessão. A melhor arrumação é a em forma de ‘U’.

Não devem acontecer interrupções aos participantes.

Todos devem cumprir a agenda, principalmente o início e o fim da reunião.

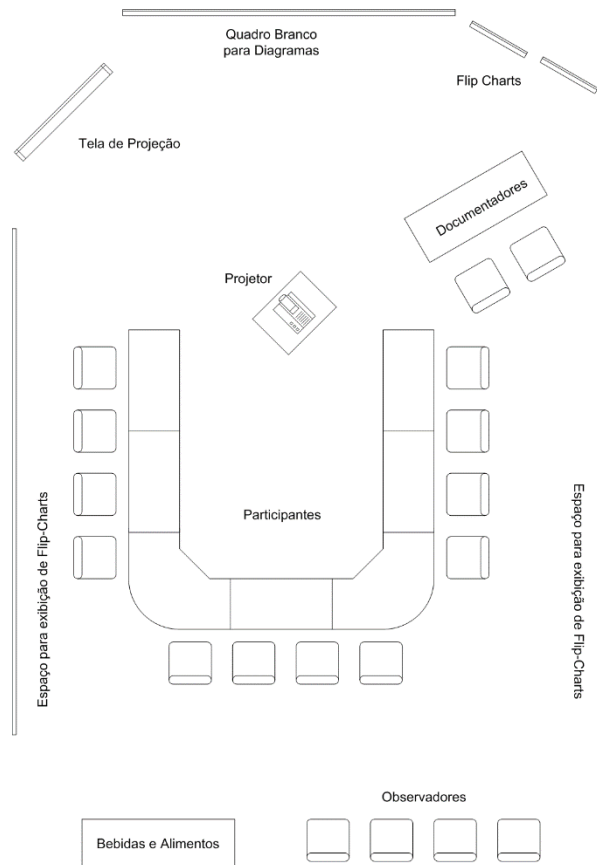


Figura 13. Uma sala de JAD bem planejada

I.9.3.5 O Consenso

A forma mais produtiva de decisão do grupo é aquela obtida por consenso. Consenso não é a unanimidade de opiniões, mas, sim, a situação em que cada membro concorda que a solução encontrada é a melhor para o grupo e que é possível conviver com ela sem ferir convicções ou valores essenciais.

I.10 Exercícios

I.10.1 Projeto 1: Livraria ABC

Baseado em todos os textos disponíveis sobre a Livraria ABC, faça:

1. Uma lista de requisitos funcionais preliminares

2. Uma lista de requisitos não-funcionais preliminares
3. Uma lista de requisitos de informação preliminares

I.10.2 Projeto de Curso

Para o seu projeto de curso, faça uma lista com:

1. Requisitos funcionais preliminares
2. Requisitos de informação preliminares
3. Requisitos não funcionais preliminares
4. Documente cada requisito dessa lista de acordo os descritores de requisitos mostrados nesse capítulo.
5. Para o seu projeto de curso, prepare:
6. Um roteiro de uma entrevista inicial do projeto
7. Um questionário a ser feito com os usuários atuais do serviço com a finalidade de descobrir sua qualidade