

Curso de Sistemas de Informação
Disciplina: Análise e Projeto de Sistemas
Professor: José Maurício S. Pinheiro

AULA 2 – Engenharia de Requisitos: Conceitos Básicos

A tarefa de construir sistemas de informação é, em última análise, é um processo de solução de problemas. Muitos dos sistemas de computadores que elaboramos são substituições ou novas implementações de sistemas não-computadorizados que já existem; além disso, a maioria dos sistemas computadorizados interage ou tem uma interface com vários sistemas existentes (alguns podem ser computadorizados ou não). Para que nosso sistema computadorizado seja bem-sucedido, precisamos conhecer como o sistema atual se comporta.

1. Análise de Sistemas

A principal tarefa do Analista de Sistemas é descobrir o que um sistema deverá fazer. Ao conjunto de necessidades a serem atendidas dá-se o nome de requisitos do sistema. Portanto, descobrir os requisitos do sistema é uma tarefa de análise e investigação e faz parte do objetivo da análise a captura de todos os requisitos para o software que será desenvolvido.

A análise de sistemas consiste nos métodos e técnicas de investigação e especificação da solução de problemas, a partir dos requisitos levantados, para criação e implementação de software em algum meio que o suporte. Alguns conceitos:

1.1. Análise

Análise, derivado do grego *analýein* - desatar, soltar, significa dissolução de um conjunto em suas partes. Em sentido amplo, empregam-se os termos “análise” e “analisar” como sinônimos de exame e examinar, pesquisa e pesquisar, verificação e verificar.

1.2. Processo

Série de fenômenos sucessivos com relação de causa e efeito; por exemplo, uma empresa é uma série de causas (matérias primas, recursos humanos, tecnologia, etc.) que geram um efeito (produtos).

1.3. Programa

Escrito em que se dão os pormenores de um espetáculo, de uma cerimônia, das condições de um concurso, dos procedimentos para execução de uma tarefa.

1.4. Análise de Sistemas

Representa o estudo detalhado de uma área de trabalho (processo), que antecede uma ação que, quase sempre, implica no desenvolvimento de um conjunto de programas integrados (sistema) destinado à execução controle e acompanhamento do processo.

1.5. Requisitos

A palavra “requisitos” pode ter diferentes semânticas. Três em particular são mais comuns:

1. Conjunto de informações necessárias para a construção de um software. Esse conjunto é composto de informações que procuram significar desejos dos que demandam o software e de informações sobre fatos do contexto onde o software irá operar;
2. Modelos escritos em linguagens artificiais que procuram organizar algumas das informações necessárias para a construção de um software;
3. Frases que representam diretivas explícitas a serem levadas adiante pelo software (demanda). Chamamos essas frases de: Requisitos Propriamente Ditos.

Requisitos são as bases para todo projeto, definindo o que as partes interessadas de um novo sistema necessitam e também o que o sistema deve fazer para satisfazer as suas necessidades. Os requisitos guiam as atividades do projeto e normalmente são expressos em linguagem natural para que todos possam obter o entendimento (Fig. 1).

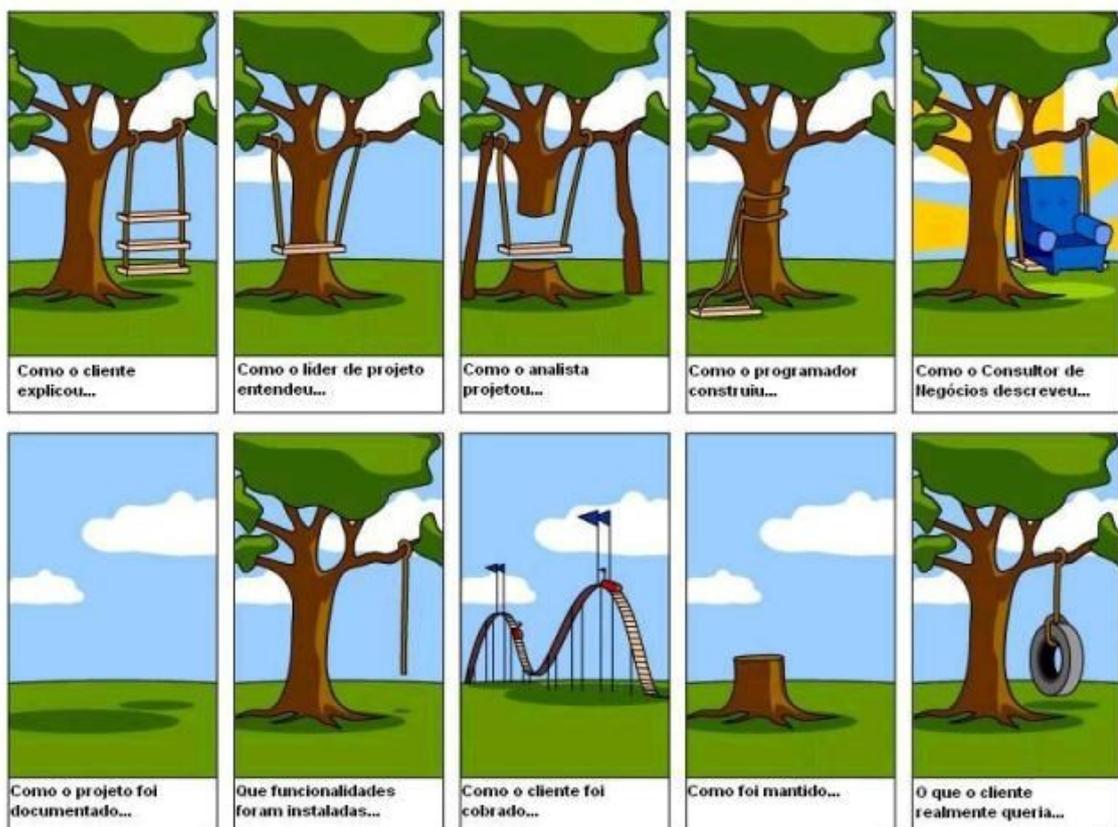


Figura 1 - Diferentes visões de um requisito

2. Engenharia de Requisitos

Depois de haver entendido claramente o que o usuário espera que seja feito, o analista de sistemas deverá levantar informações que envolve a área de desenvolvimento do sistema. É a fase precedente do início de análise, chamada de análise de requisitos do sistema ou levantamento de eventos e dados iniciais.

Um requisito pode ser definido como um desejo, necessidade, restrição ou expectativa de um cliente com relação a um produto. Ou seja, um cliente, ao pensar em um produto, possui muitos aspectos em sua mente. Então, um requisito pode ser resumido como algo desejado por um usuário, com relação a um produto.

Requisitos são a base para estimativas, modelagem, projeto, implementação, testes e até mesmo para a manutenção. Assim, os requisitos estão presentes ao longo de todo o ciclo de vida de um software. Esses aspectos necessitam ser capturados, definidos, organizados, verificados e validados para então chegarmos a uma Especificação de Requisitos. Seu principal objetivo é prover um enunciado completo, claro e preciso dos requisitos de um produto de software.

Os requisitos envolvem atividades de desenvolvimento através do Levantamento e Análise e Documentação de Requisitos, gerência através da Gerência de Requisitos e por fim o controle da qualidade através da Verificação, Validação e Garantia da Qualidade de Requisitos. Todas essas atividades que são relacionadas a requisitos é o Processo de Engenharia de Requisitos. Portanto, a engenharia de requisitos é o processo pelo qual os requisitos de um produto de software são coletados, analisados, documentados e gerenciados ao longo de todo o ciclo de vida do software (Fig.2).

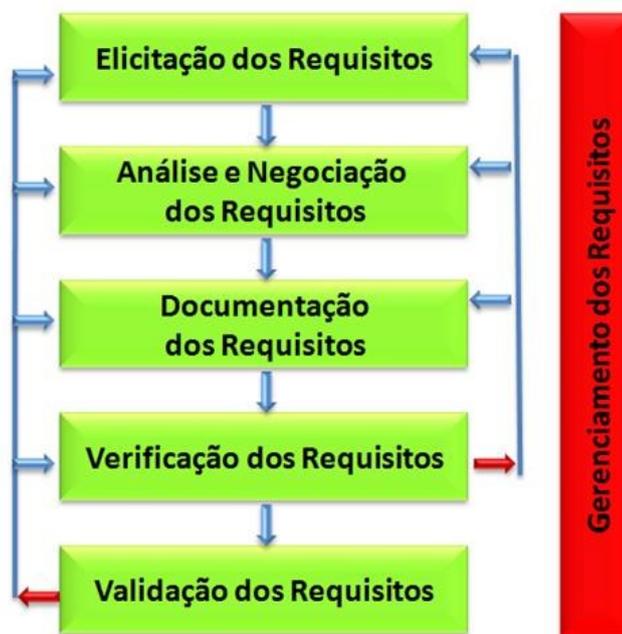


Figura 2 – Fases da Engenharia de requisitos

2.1. Fluxo de Requisitos

É importante detalhar de forma precisa o que é o Fluxo de Requisitos. Esse fluxo reúne as atividades que visam a obter o enunciado completo, claro e preciso dos requisitos de um produto de software. Os requisitos devem ser levantados pela equipe do projeto em conjunto com representantes do cliente, usuários-chave e até contando com a presença de especialistas da área de aplicação, uma vez que o

projeto pode envolver conhecimentos não triviais que exijam a presença de profissionais altamente especializados com a área de negócio do produto a ser construído. O resultado principal do fluxo dos requisitos é o documento de Especificação de Requisitos.

2.2. Qualidade dos Requisitos

Os requisitos de um software correspondem a uma parte importante do desenvolvimento. Por conta disso, é necessário que eles possuam diversas características de qualidade, permitindo assim seu uso de forma adequada e eficiente:

- **Correção** - Uma especificação de requisitos é correta se todo requisito presente nela realmente é um requisito do produto a ser construído. Não existe ferramenta que garanta a correção de uma especificação de requisitos. Para verificá-la, deve-se checar a coerência da especificação dos requisitos do software com outros documentos da aplicação, como manuais, protocolos, regras de negócio etc. Uma outra forma de se tentar garantir a correção é solicitar a aprovação formal da especificação de requisitos por parte do cliente, sem a qual o projeto não poderá prosseguir.
- **Precisão** - Uma especificação de requisitos é precisa se todo requisito presente possui apenas uma única interpretação, aceita tanto pelos desenvolvedores quanto pelos usuários-chave. Em particular, uma especificação de requisitos deve ser compreensível para todo o seu público alvo, e deve ser suficiente para a especificação dos testes de aceitação do produto. Recomenda-se a inclusão no glossário da especificação de requisitos de todos os termos contidos no documento que possam causar ambiguidades de entendimento. é fundamental para uma especificação de requisitos possuir um glossário com todos os termos técnicos e unidades de medida, facilitando assim seu entendimento por todos.
- **Completa** - Uma especificação de requisitos é completa se reflete todas as decisões de especificação que foram tomadas, não contendo cláusulas de pendências. Uma especificação de requisitos completa deveria conter todos os requisitos significativos relativos a funcionalidade, desempenho, restrições de desenho, atributos e interfaces externas, além de definir as respostas do software para todas as entradas possíveis, válidas e inválidas, em todas as situações possíveis.
- **Consistência** - Uma especificação de requisitos é consistente se não há conflitos entre nenhum dos subconjuntos de requisitos presentes, tais como conflitos com o mundo real, por exemplo, formatos de relatórios ou cores de sinalização; conflito lógico ou temporal entre ações, quando, por exemplo, um requisito diz que a ação A deve ser realizada antes da ação B, e outro diz o contrário; e uso de termos diferentes para designar o mesmo objeto do mundo real, como, por exemplo, “lembrete” versus “aviso”.
- **Priorização** - Uma especificação de requisitos é priorizada se cada requisito é classificado de acordo com a respectiva importância e estabilidade. A estabilidade estima a probabilidade de que o requisito venha a ser alterado no decorrer do projeto, com base na experiência de projetos correlatos. A priorização classifica o requisito de acordo com graus de importância atribuídos pelos clientes. A priorização é algo importante pois normalmente os custos e prazos podem ser bastante limitados, sendo importante descrever o que é mais importante e deve ser tratado primeiro.

- **Verificabilidade** - Uma especificação de requisitos é verificável se todos os seus requisitos são verificáveis. Um requisito é verificável se existir um processo finito, com custo compensador, que possa ser executado por uma pessoa ou máquina e que mostre a conformidade do produto final com o requisito.
- **Modificabilidade** - Uma especificação de requisitos é modificável se sua estrutura e estilo permitirem a mudança de qualquer requisito, de forma fácil, completa e consistente. A modificabilidade geralmente requer que haja uma organização coerente, com índices e referências cruzadas; ausência de redundância entre requisitos; definição separada de cada requisito. Essa característica está diretamente relacionada ao uso de padrões e convenções, de forma que o trabalho seja feito utilizando-se formatos pré-definidos e adequados ao uso.
- **Rastreabilidade** - Uma especificação de requisitos é rastreável se permite a fácil determinação dos itens que antecederam o surgimento do item e os itens que foram gerados por conta da existência do item. Isso normalmente está associado a dois tipos de rastreabilidade:
 - Rastreabilidade para trás, na qual deve ser possível localizar a origem de cada requisito. Deve-se sempre saber por que existe cada requisito e quem ou o que o originou. Isso é importante para que se possa avaliar o impacto da mudança daquele requisito, e dirimir dúvidas de interpretação;
 - Rastreabilidade para a frente, na qual deve ser possível localizar quais os resultados do desenvolvimento que serão afetados por cada requisito. Isso é importante para garantir que os itens de análise, desenho, código e testes abranjam todos os requisitos e para localizar os itens que serão afetados por uma mudança nos requisitos.

2.3. Atividades do Fluxo de Requisitos

O Fluxo de Requisitos é composto por várias atividades, que devem ser executadas dentro de um processo de desenvolvimento. Diversos processos de software possuem atividades de requisitos em suas definições. No entanto, a análise de um processo de software específico pode conter diversas diferenças para as atividades apresentadas (Fig. 3).

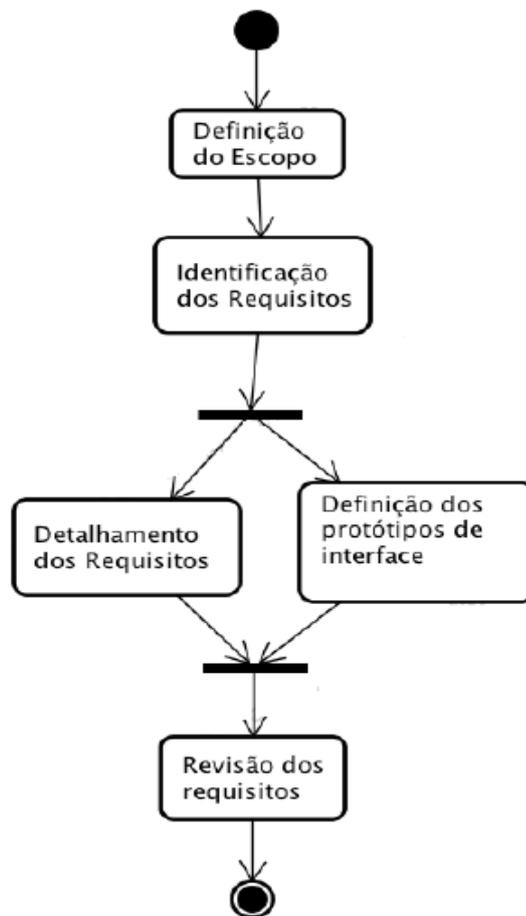


Figura 3 - Atividades do Fluxo de Requisitos

O primeiro elemento na figura (uma circunferência preta) é conhecido como atividade inicial. Ele representa o ponto de partida do fluxo. Os demais elementos (retângulos com os lados arredondados) representam atividades. Existem duas barras de sincronização na figura, que representam que as atividades posteriores só iniciam quando a todas as atividades com ligação à sincronização tenham encerradas. Assim, a revisão dos requisitos apenas tem início quando o detalhamento dos requisitos e detalhamento dos protótipos de interface tenham sido concluídas.

- **Definição do Escopo** - atividade onde o escopo do projeto vai ser identificado. De forma geral, a partir do escopo deve ser possível identificar o que faz parte do projeto e o que não faz parte.
- **Identificação dos Requisitos** - atividade relacionada à obtenção de tudo o que os clientes desejam com relação ao produto. Isso inclui a definição do comportamento esperado, bem como outros aspectos que deverão ser identificados.
- **Detalhamento dos Requisitos** - atividade em que os desenvolvedores, com a ajuda dos clientes, iniciam a desdobrar os requisitos em funções do produto, de forma que o atendimento seja completo.
- **Definição dos Protótipos de Interface** - atividade em que versões iniciais das interfaces do produto são criadas, com o intuito maior de deixar claro o que se deseja, reduzindo assim problemas com requisitos questionáveis ou difíceis de serem entendidos.

- **Revisão dos Requisitos** - atividade em que é feita uma revisão geral do trabalho realizado, com o intuito de remover problemas com relação aos requisitos identificados e todos os seus desdobramentos executados.

É importante ressaltar que o fluxo de requisitos é um fluxo com grande participação do cliente. Ele é quem define praticamente tudo o que será produzido nessa fase do desenvolvimento. Assim, quanto mais mecanismos utilizarmos para facilitar a comunicação com o cliente, melhores resultados obteremos com sua execução.

2.3.1. Definição do Escopo

O Escopo de um Projeto é fundamental para o seu sucesso. Sua definição deve ser feita com a participação dos principais envolvidos. Uma boa forma de evitar problemas com os clientes de um projeto é definir bem o escopo e, para evitar falsas expectativas, detalhar o que não faz parte dele. Por isso é tão importante termos uma seção com os Limites do Produto. De forma geral, o escopo de um projeto pode ser simplesmente um texto, que define o que deve fazer parte do projeto.

Na identificação do escopo, quatro aspectos devem ser observados:

- Entendimento do Domínio da Aplicação - onde se entende, de uma maneira geral, a área na qual o sistema será aplicado;
- Entendimento do Problema - onde entendemos os detalhes do problema específico a ser resolvido com o auxílio do sistema a ser desenvolvido;
- Entendimento do Negócio - onde entendemos como o sistema afetará a organização e como contribuirá para que os objetivos do negócio e os objetivos gerais da organização sejam atingidos;
- Entendimento das Necessidades e das Restrições dos Interessados - onde entende-se as demandas de apoio para a realização do trabalho de cada um dos interessados no sistema, entende-se os processos de trabalho a serem apoiados pelo sistema e o papel de eventuais sistemas existentes na execução e condução dos processos de trabalho.

Em muitos momentos, o escopo do projeto deverá ser reanalisado, pois ele é que define o que está incluído e o que não está incluído no projeto em questão. Um escopo mal estruturado poderá levar a falhas de cronograma e de orçamento, uma vez que tarefas acima do previsto podem ser consideradas, ao invés daquilo o que realmente deveria ter sido incluído.

Como mencionado, é importante definir o que está fora do escopo do projeto. Isso normalmente ajuda aos clientes entenderem de forma mais precisa o que está e o que não está incluído no projeto. A definição do que não está incluído de forma explícita evita falsas expectativas. Isso normalmente favorece o processo de comunicação com o cliente.

2.3.2. Identificação dos Requisitos

Nessa atividade levam-se em conta as necessidades dos usuários e clientes, informações de domínio, sistemas já existentes na organização, regulamentos vigentes, leis etc. O objetivo nessa fase é entendermos a organização como um todo, seus processos, necessidades, possibilidades de melhorias e restrições existente.

Para a identificação dos requisitos, diferentes técnicas podem ser empregadas para ajudar o levantamento desses requisitos: entrevistas, questionários, observação do

ambiente e dos indivíduos nas suas tarefas cotidianas na organização, análise de documentos existentes na organização, cenário de interação entre o usuário final e o sistema onde o usuário pode simular a sua interação com o sistema explicando para o analista o que ele está fazendo e de que informações ele precisa para realizar a tarefa, prototipagem onde uma versão preliminar do sistema, muitas vezes não operacional e descartável, é apresentada ao usuário para capturar informações específicas sobre seus requisitos de informação, observação reações, dinâmica de grupo, e diversas outras técnicas que também podem ser empregadas.

A Identificação do Requisitos é a atividade que prescreve a criação de uma lista dos requisitos para a aplicação a ser desenvolvida. Cada requisito nada mais é que uma descrição textual de algo que deveria ser considerada durante o desenvolvimento de software. Os requisitos podem ser divididos em dois tipos: requisitos funcionais, associados a comportamento, e requisitos não funcionais, ligados a características que o comportamento deve possuir.

2.3.2.1. Requisitos Funcionais

Os requisitos funcionais estão diretamente ligados ao comportamento que a aplicação deve conter. Por exemplo, em um sistema de controle de uma biblioteca, o Empréstimo de Livro exige que o usuário a tomar um livro emprestado esteja cadastrado, ativo, que o livro desejado esteja disponível, não reservado, não seja cativo e que o usuário não esteja com cinco livros emprestados, considerando que esse seja o número máximo de empréstimos simultâneos permitido. Tudo isso que foi comentado são detalhes do comportamento que o software deve apresentar.

2.3.2.2. Requisitos Não Funcionais

Os requisitos não funcionais não expressam comportamento, mas características que o comportamento deve ter. Por exemplo, supondo o mesmo requisito, *Empréstimo de Livro*, podemos exigir como requisito de desempenho que ele funcione com até 100 usuários simultâneos, gerando respostas em até 8 segundos. Outro requisito não funcional associado ao empréstimo seria que ele fosse simples de usar, permitindo que uma pessoa conseguisse utilizá-lo apenas lendo o manual associado. Todas características desejadas ao comportamento especificado para o sistema da biblioteca.

2.3.2.3. Lista de Requisitos

A lista dos requisitos é a expressão que resume aquilo que os clientes desejam. É importante a participação dos Analistas de Requisitos para que seja possível organizar esses pedidos e estruturar o texto que os descreve de forma que seja possível analisar e desdobrar esses pedidos em funções do produto.

A Tabela 1 exibe a lista de requisitos para a ferramenta de gestão de requisitos proposta para o sistema da biblioteca:

Tabela 1 - Exemplo de definição de requisitos

ID	Requisito	Descrição
RF1	Requisitos	O sistema deve permitir cadastrar e controlar todos os aspectos relacionados aos requisitos de um projeto, permitindo visualizar isso e
		acompanhar sua evolução, incluindo as pessoas que trabalharam no projeto, os analistas e gerente do projeto.
RF2	Casos de uso	O sistema deve possibilitar a especificação dos casos de uso, registrando sua descrição, atores, protótipos de tela associados, relacionando aos requisitos que deram origem ao caso de uso.
RF3	Revisão	Deve ser possível realizar uma revisão dos requisitos e casos de uso, utilizando critérios definidos pelos próprios gerentes de projetos, de forma independente dos demais projetos.
RF4	Acompanhamento dos clientes	Deve ser possível que os clientes possam acompanhar a evolução do projeto a qualquer momento, consultando tudo o que foi feito.
RF5	Liberação de acesso por projeto	Deve ser possível liberar o acesso ao sistema por projeto, indicando o seu gerente. Assim, para se ter acesso ao sistema, deverá ser adquirida uma licença para um projeto. A partir disso o gerente ficará responsável por definir quem deverá usar o sistema, seja para trabalhar na especificação de requisitos, como um dos Engenheiros de Requisitos, seja como cliente consultando o resultado do trabalho realizado.
RF6	Geração da documentação	Deve ser possível gerar a especificação na forma de um documento não editável, contendo todos os dados registrados do projeto.

A Tabela 2 exibe a lista de requisitos não funcionais identificados para o produto. São definições que retratam o que se deseja com relação a certas características ligadas ao comportamento do software. São exemplos disso a definição do ambiente (Web), o uso de uma tecnologia específica (MySQL) ou a definição de valores para atributos de qualidade como a quantidade de acessos simultâneos e o tempo máximo de resposta.

É importante ressaltar que requisitos não funcionais necessitam da definição de valores que permitam sua verificação. Dizer que o sistema deve ser rápido não ajuda muito aos testadores que devem verificar se o produto atende à especificação de

requisitos. Mas especificar um tempo máximo de resposta em um contexto pré-definido ajuda bastante.

Tabela 2 - Exemplo de requisitos não funcionais

ID	Requisito	Descrição
RNF1	Ambiente	O sistema deve funcionar em ambiente Web, sendo compatível com os principais navegadores do momento: Internet Explorer, Firefox, Safari e Chroma.
RNF2	Linguagem	O sistema deve ser desenvolvido utilizando-se a linguagem Java, com as tecnologias JSF e Hibernate.
RNF3	Banco de dados	O sistema deve utilizar o banco de dados MySQL.
RNF4	Desempenho	O sistema deve ser construído para funcionar com 100 usuários simultâneos, com respostas de até 5s quando utilizado em rede local.
RNF5	Segurança	O sistema deve restringir o acesso por meio de senhas. Deve-se ter um controle no registro de senha, de forma a impedir o uso de senhas consideradas fáceis.
RNF6	Usabilidade	Um usuário com conhecimento básico em informática e conhecimento nos conceitos de requisitos deveria ser capaz de operar o sistema com um curso de 30 minutos.

2.3.3. Detalhamento dos Requisitos

O Detalhamento dos Requisitos é a atividade em que cada requisito identificado deve ser desdobrado em funções do produto. Cada função do produto pode estar ligada a um único requisito, assim como pode estar relacionada a mais de um requisito. O detalhamento é uma atividade de modelagem. Vale ressaltar que essa modelagem é conceitual, pois estamos preocupados com o domínio do problema e não com soluções técnicas. Portanto, os modelos de análise são elaborados a fim de obtermos uma compreensão maior do sistema a ser desenvolvido e para especificá-lo.

O desdobramento de requisitos gera a identificação dos Casos de Uso. Esse é um termo comum na Engenharia de Requisitos e que precisa ser entendido. Um Caso de Uso é uma fatia de funcionalidade do sistema, que representa algo de valor para seus usuários e que não apresenta nem lacunas nem superposição com outros casos de uso. Casos de uso podem ser considerados as funções que um produto deve oferecer.

Os casos de uso são acionados por atores. Um ator é a representação de um papel no sistema. Atores podem representar um grupo de usuários, um outro sistema, com o qual o sistema sendo especificado deve interagir. Um caso de uso normalmente interage com apenas um ator, mas pode interagir com mais de um. Atores podem ser identificados através dos seguintes critérios:

- Quem está interessado em certo requisito;
- Quem se beneficiará diretamente do produto;
- Quem usará informação do produto;
- Quem fornecerá informação ao produto;
- Quem removerá informação do produto;
- Quem dará suporte e manutenção ao produto;
- Quais os recursos externos usados pelo produto;
- Quais os papéis desempenhados por cada usuário;
- Quais os grupos de usuários que desempenham o mesmo papel;
- Quais os sistemas legados com os quais o produto deve interagir;
- O tempo, quando casos de uso são disparados periodicamente, de forma automática.

Atores são usados para representar também sistemas externos. Estes podem incluir sistemas legados, produtos comerciais de software e outros componentes de um sistema maior. Podem incluir recursos de hardware e comunicação que devam receber um tratamento específico por parte do produto (por exemplo, dispositivos de hardware que normalmente não fazem parte do ambiente operacional). Um ator especial é o Tempo, usado para acionar casos de uso que são disparados periodicamente, de forma automática.

Cada diagrama de casos de uso especifica os relacionamentos entre casos de uso e atores. Os relacionamentos indicam a existência de comunicação entre atores e casos de uso. Um caso de uso pode estar associado a mais de um ator, quando a sua execução requer a participação de diferentes atores.

Normalmente, a comunicação será representada como ligação sem direção. Nesses casos, é convenção que a iniciativa de comunicação parte do ator. Quando a iniciativa parte do caso de uso (por exemplo, alarmes, mensagens, dados enviados para outros sistemas etc.), a comunicação deve ser direcionada para o ator.

A Figura 4 exibe um exemplo de como é representado um caso de uso e atores em UML. Observar que o relacionamento possui setas com indicação de direção. Isso mostra o fluxo de dados no relacionamento.

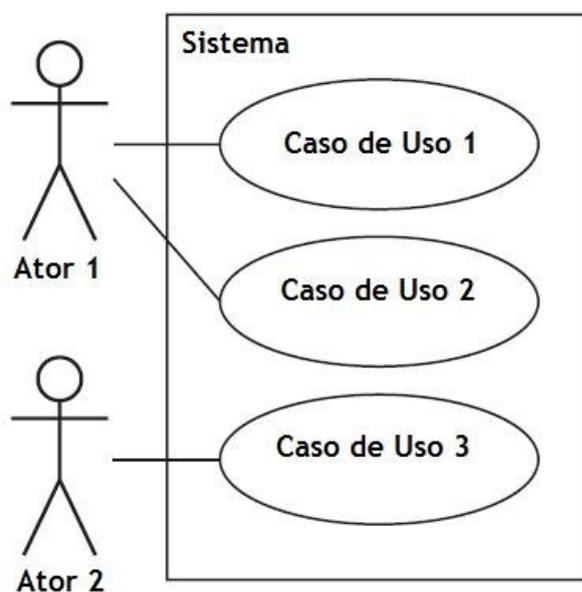


Figura 4 - Exemplo de ator e caso de uso

Assim, após a identificação dos requisitos, é necessário detalhar quais casos de uso serão necessários para atender aos desejos identificados pelos clientes. Nesse momento é preciso ajudar aos clientes a pensar como estruturar o software, em termos de funções, para atender aos anseios identificados.

Como exemplo, temos a análise dos requisitos anteriormente apresentados na Tabela 1. Lembrando que os requisitos são textos que descrevem os desejos dos clientes e os casos de uso são as traduções dos requisitos em funções do produto, a partir de um entendimento do que foi solicitado e o que pode ter em um produto.

Inicialmente, utilizaremos o RF1: “O sistema deve permitir cadastrar e controlar todos os aspectos relacionados aos requisitos de um projeto, permitindo visualizar isso e acompanhar sua evolução, incluindo as pessoas que trabalharam no projeto, os analistas e o gerente do projeto.”

De acordo com o texto do requisito, o sistema deve possibilitar o cadastro e controle dos requisitos. Com isso, podemos identificar uma função do produto: Cadastro de requisitos. Esse então já é um dos casos de uso do sistema. Como existem requisitos funcionais e não funcionais é preciso decidir se iremos realizar esse cadastro em um único caso de uso ou se serão necessários dois casos de uso: um para o cadastro de requisitos funcionais e outro para requisitos não funcionais. Nesse caso em específico, vamos utilizar um único caso de uso para registro dos requisitos. Vamos nomear esse caso de uso como *Gestão de requisitos*.

Em RF1 existe a menção ao registro das pessoas que trabalharam no projeto. Isso significa que teremos que ter um cadastro dos envolvidos no projeto. Teremos que cadastrar o gerente, os analistas e os clientes envolvidos. Com isso, é possível identificar que teremos que cadastrar os membros do projeto. Isso dá origem a mais um caso de uso: *Gestão de membros*.

Continuando a análise, para atender RF2 é necessário termos um caso de uso para a Gestão de Casos de Uso. No entanto, teremos também que registrar os atores associados ao caso de uso. Isso nos remete a um novo caso de uso: *Gestão de atores*.

O RF3 leva a um novo caso de uso: Revisão. No entanto, foi descrito que as revisões serão guiadas por critérios independentes para cada projeto. Isso nos leva a identificar um novo caso de uso: *Gestão de critérios*.

O RF4 solicita a existência de uma forma de acompanhar o projeto. Isso pode ser resumido a partir de uma função do produto que gerasse a especificação em um formato contendo todos os dados do projeto, incluindo requisitos, casos de uso, atores, etc. Por conta disso, o RF4 nos leva a identificar mais um caso de uso: *Geração da especificação*.

O RF5 refere-se ao modelo de negócio do sistema. Para que alguma pessoa possa utilizá-lo, é necessário que seja realizado o cadastro de um projeto. Esse projeto terá um gerente associado que a partir de então poderá cadastrar seus analistas, clientes, os requisitos, casos de uso etc. Dessa forma, é necessário que o sistema tenha um caso de uso Cadastro de Projeto, que deve ser feito por um administrador, responsável por liberar acesso ao software. No entanto, um projeto possui diversos dados adicionais, como seu escopo, datas de execução, limites do produto, que também precisam ser cadastrados. Isso nos leva a ter um outro caso de uso, para que seja possível registrar tais dados, mas utilizado pelo gerente do projeto e não pelo administrador do sistema que será chamado de *Controle do projeto*.

Analisando o RF6, que trata da geração da documentação, encontramos a necessidade relatada já está contemplada com a proposição do caso de uso Geração da documentação, identificado quando analisamos o RF4. Dessa forma, ao propormos o caso de uso Geração da documentação, não só atendemos ao requisito RF4, como também ao RF6.

A Figura 5 apresenta um diagrama de casos de uso com todos os casos de uso identificados, além dos atores associados. O diagrama de casos de uso oferece uma visão ampla do sistema, exibindo as funções existentes e os papéis associados. Isso não representa ainda o detalhamento dos requisitos, pois é necessário descrever de forma muito mais detalhada os requisitos. Isso inclui a determinação das regras de negócio associadas a cada um dos casos de uso.

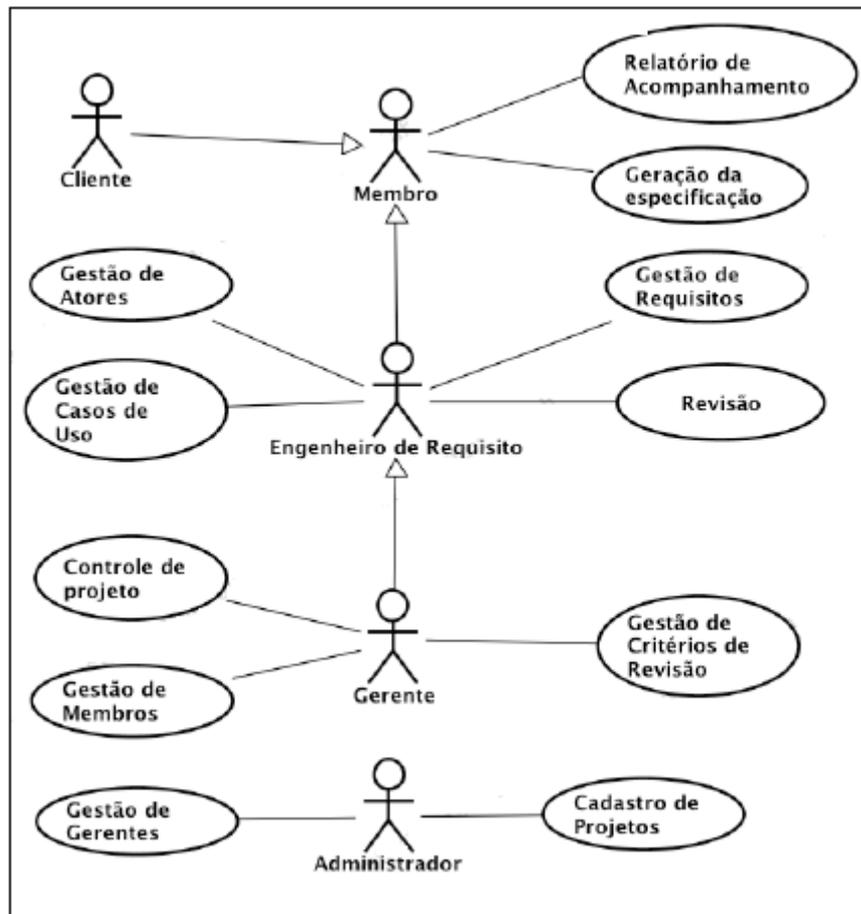


Figura 5 - Diagrama de casos de uso para o projeto de software

A Tabela 3 mostra a lista de casos de uso identificados no sistema, com a identificação dos requisitos associados. Essa ligação entre caso de uso e requisito é conhecida como rastreabilidade. Isso nos permite saber quem deu origem a quê. Assim, é possível saber que um requisito deu origem a um determinado caso de uso.

Tabela 3 - Lista de casos de uso identificados

ID	Caso de uso	Requisito associado	Descrição
UC1	Gestão de Requisitos	RF1	Cadastro de requisitos funcionais e não funcionais associados ao projeto.
UC2	Gestão de Membros	RF1	Cadastro de todos os envolvidos no projeto.
UC3	Gestão de Casos de Uso	RF2	Definição dos casos de uso do projeto.
UC4	Gestão de Atores	RF2	Definição dos atores que irão interagir no projeto.
UC5	Revisão	RF3	Revisão de um requisito ou de um caso de uso, observando os critérios pré-estabelecidos no projeto para a revisão.
UC6	Gestão de Critérios de Revisão	RF3	Cadastro de critérios a serem utilizados em uma revisão.
UC7	Cadastro de Projeto	RF5	Cadastro de um projeto, com definição do seu gerente, feito pelo administrador do sistema.
UC8	Gestão de Gerentes	RF5	Cadastro de um gerente de projeto, feito pelo administrador do sistema.
UC9	Controle de Projetos	RF5	Controle do projeto, com detalhamento de informações sobre o mesmo, feito pelo gerente do projeto.
UC10	Geração da especificação	RF6, RF4	Geração da especificação de requisitos, utilizando um formato pré-definido, contendo todos os dados registrados no projeto.
UC11	Relatório de Acompanhamento	RF6, RF4	Emissão de um relatório contendo uma indicação do estado do projeto, a partir do estado de cada caso de uso que o compõe.

A Tabela 4 exibe a lista de atores identificados.

Tabela 4 - Lista de atores do projeto

Nº	Ator	Descrição
1.	<u>Cliente</u>	Clientes de um projeto, normalmente responsável pelo fornecimento de informações para moldagem do produto.
2.	<u>Administrador</u>	Responsável pelo controle do uso do sistema, liberando acesso aos Gerentes a partir do cadastramento de um projeto.
3.	<u>Gerente</u>	Responsável pelo controle de um projeto, definindo a equipe e suas tarefas.
4.	<u>Membro</u>	Pessoa que faz parte da equipe que trabalha no projeto.

2.3.3.1. Detalhamento dos Casos de uso

A atividade de Detalhamento dos Casos de Uso é uma subatividade do Detalhamento dos Requisitos. Existem muitas formas de ser descrever um caso de uso. Uma forma é utilizar o conceito de fluxos dos casos de uso. Cada fluxo detalha passo a passo o que deve acontecer em determinada parte do produto. Essa descrição é geralmente feita por meio de textos seguindo formatos pré-estabelecidos. Notações muito informais são chamadas de histórias de usuário” (user stories) e são comumente adotadas por metodologias ágeis.

Cada caso de uso deve possuir ao menos a descrição de suas pré-condições, assim como um fluxo principal, que representa um caminho de execução que normalmente é o mais utilizado para o caso de uso.

No exemplo de fluxo principal a seguir, é possível observar que existem passos relacionados com ações do sistema (ReqG) e passos relacionados a ações do ator (Administrador).

- **Fluxo Principal**

1. O **ReqG** exibe a Tela de Gestão de Gerentes.
2. O Administrador informa os dados para pesquisa por Gerentes.
3. O Administrador aciona o comando Pesquisar.
4. O **ReqG** recupera e exibe na lista Gerentes recuperados os Gerentes que atendem aos parâmetros de pesquisa informados, ordenados pelo Nome em ordem crescente.

Os fluxos são descritos em linguagem natural, na forma de uma sequência de passos. Cada passo corresponde a uma ação de um ator ou do produto e que devem aparecer explicitamente como sujeitos da frase. Outros atores podem aparecer como objetos verbais de uma ação. Nesse caso, provavelmente tais atores estejam ligados ao caso de uso utilizando-se setas que indicam a direção da comunicação no diagrama de casos de uso. Condições e iterações podem aparecer nas descrições dos casos de uso (se alguma coisa, para cada coisa faça isso).

Além do fluxo principal, que descreve uma parte do caso de uso que provavelmente seja a mais utilizada, existem fluxos alternativos e subfluxos. Os fluxos alternativos

(também chamados de fluxos excepcionais) são descrições de alternativas de execução que podem ser iniciadas sempre que suas pré-condições forem atendidas (Tabela 5). Em geral, os casos de uso não possuem pré-condições e alguns tem apenas o fluxo principal.

Tabela 5 - Fluxo alternativo - inclusão de um novo gerente

Precondições	1. O <u>Administrador</u> acionou o comando <i>Novo</i> .
Passos	<ol style="list-style-type: none"> 1. O <u>Administrador</u> preenche os <i>Dados do Gerente</i>. 2. O <u>Administrador</u> aciona o comando <i>Salvar</i>. 3. O ReqG verifica que não existe um Gerente com email e login informados. 4. O ReqG salva os <i>Dados do Gerente</i>.

Na descrição do fluxo alternativo acima existe a especificação de precondições. Essas precondições detalham o que deve acontecer para que o fluxo entre em execução. No caso, para que a inclusão de um novo gerente aconteça, é necessário que o ator associado ao caso de uso acione o comando novo, pois essa é a indicação que se deseja que o fluxo seja executado.

No fluxo apresentado, há especificação de restrições no seu funcionamento. No passo 3 do fluxo é especificado que o sistema (ReqG) verificará se uma determinada condição é atendida. Isso significa que o fluxo só continuará se ela for verdade. Caso ela não seja verdade, o fluxo não continuará sua execução. Nesse caso, foi especificada uma condição simples, relacionada à verificação da existência de um gerente com determinadas informações. No entanto, poderíamos ter especificado condições bem mais complexas. Essas condições serão traduzidas nas diversas regras de negócio de um produto durante sua implementação.

Não precisamos definir mensagens ao usuário neste momento. A maioria dos iniciantes na descrição de casos de uso é tentado a descrever a verificação contida no passo 3 do fluxo alternativo exibido anteriormente contendo uma mensagem ao usuário, normalmente da seguinte forma: O ReqG verifica se não existe um gerente com e-mail informado. Se existir o ReqG emite a mensagem “Gerente já cadastrado!”. É importante frisar que mensagens para o usuário fazem parte do projeto (design) do sistema, uma etapa posterior aos requisitos e análise e devem seguir convenções e padrões de usabilidade.

Os subfluxos são utilizados para descrever conjuntos de passos que foram extraídos de algum fluxo por serem grandes, complexos ou com potencial de serem reutilizados em outros fluxos. Para acionar a execução de um subfluxo é necessário especificar isso de forma direta: O ReqG executa o Subfluxo X.

2.3.4. Definição dos Protótipos de Interface

A Definição dos Protótipos de Interface especifica, de forma detalhada, os requisitos relacionados as fontes de entrada e saída de dados no produto. Nas interfaces gráficas de usuário, existem questões que claramente representam requisitos dos produtos, tais como formatos de dados e comandos. Outros detalhes, como formatos de telas e janelas, são aspectos de desenho da interface de usuário e não devem ser tratados durante a fase de Requisitos.

Os protótipos de interfaces, durante o levantamento de requisitos, devem ser focados em descobrir as informações e restrições importantes ao requisito. Nenhum aspecto de execução ou usabilidade deve ser tratado nesse momento.

É importante definir que tecnologia utilizar para geração desses esboços, uma vez que eles não devem demandar muito esforço e nem devem focar em tecnologia específica, visto que isso pode limitar o espaço da solução sem que haja essa necessidade. Os esboços devem ser descritos de forma que o usuário consiga entender seu objetivo e entenda seu funcionamento, independente da tecnologia a ser utilizada.

2.3.5. Revisão dos Requisitos

A revisão dos requisitos é a atividade realizada para garantir que o padrão prescrito pela organização foi realmente seguido e que os requisitos identificados atendam aos critérios de qualidade solicitados, permitindo o seu correto entendimento e, por conseguinte, a realização do projeto adequado bem como da implementação apropriada derivada da evolução dos requisitos (Fig.6).

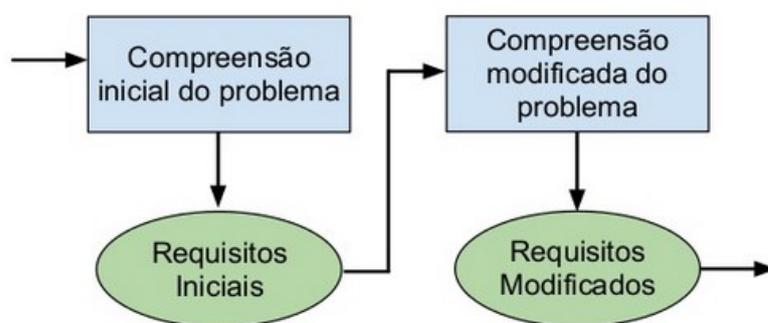


Figura 6 - Evolução dos requisitos

O projeto pode determinar que critérios devem ser utilizados para a realização das revisões, para então analisar cada requisito à luz dos critérios selecionados. Dessa forma, podemos visualizar que uma revisão nada mais é que uma leitura e posterior análise dos requisitos, tendo em mente aspectos pontuais a serem avaliados.

Um exemplo de revisão para parte dos requisitos é apresentado na Tabela 6. Nela podemos notar a análise de alguns requisitos e casos de uso, com base em critérios predefinidos. A partir dessa análise serão registrados os eventuais conflitos e acompanhado os passos para sua resolução.

Tabela 6 - Trecho da revisão de uma especificação de requisitos

Nr.	Requisito	Caso de Uso	Ambigüidade	Clareza	Completude	Conflitos
1	RF1	UC2	Aprovado	Aprovado	Não aprovado	Não foi especificado a ordem em que os resultados devem ser exibidos.
2	RF2	UC3	Aprovado	Aprovado	Aprovado	-
3	RF5	UC8	Não aprovado	Aprovado	Aprovado	O caso de uso parece que poderia ser agrupado com o caso de uso Gestão de Membros, não havendo necessidade de criação de um caso de uso adicional.

3. Gerenciamento de Requisitos

O gerenciamento de requisitos abrange as atividades que tem por objetivo designar os atributos para os requisitos do software, definir suas visualizações dos requisitos de forma que as prioridades e o rastreamentos dos requisitos possam ser realizados.

Por outro lado, mudanças nos requisitos ocorrem durante todo o processo de desenvolvimento do software, desde o levantamento de requisitos até durante a operação do sistema em produção. Isso ocorre devido à descoberta de erros, omissões, conflitos, inconsistência nos requisitos, melhor entendimento dos usuários sobre as suas necessidades, problemas técnicos, mudanças de prioridades do cliente, mudanças no negócio, concorrentes, mudanças econômicas, mudanças no ambiente de software, mudanças organizacionais, etc. (Fig.7).

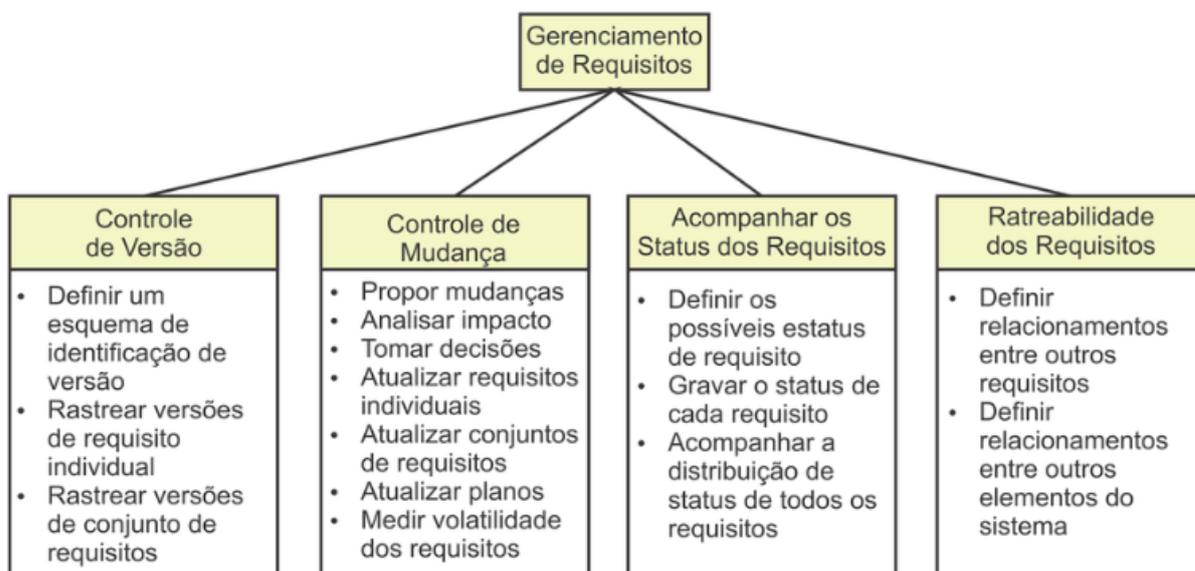


Figura 7 - Gerenciamento de requisitos

Para minimizar os problemas causados por essas mudanças é necessário gerenciar requisitos. O Processo de Gerencia de Requisitos também envolve atividades que ajudam a equipe a identificar, controlar e rastrear requisitos e gerenciar mudanças de requisitos em qualquer momento ao longo do ciclo de vida do software.

Portanto, os objetivos do processo são gerenciar alterações nos requisitos acordados, gerenciar relacionamentos entre requisitos, gerenciar dependências entre requisitos e outros documentos produzidos durante o processo de software. Dessa forma, a gerência de requisitos possui as seguintes atividades:

- Controle de mudanças;
- Controle de versão;
- Acompanhamento do estado dos requisitos;
- Rastreamento de requisitos.

A definição de um processo apropriado para uma organização é muito importante e traz diversos benefícios, pois uma boa descrição de um processo fornece orientações e reduz a probabilidade de erros ou esquecimentos. O mais importante é saber que não existe um processo ideal, portanto adaptar um processo para as necessidades internas é sempre a melhor escolha ao invés de impor um processo à organização.

4. Questões

1. Qual a definição de requisito?
2. Qual o objetivo de uma Especificação de Requisitos?
3. O que é a Engenharia de Requisitos?
4. O que é o fluxo de requisitos?
5. Quem participa do levantamento de requisitos?
6. O que é o escopo de um projeto?
7. O que é o gerenciamento de requisitos?