

Engenharia de Software

Engenharia de Requisitos

Prof. Luís Fernando Garcia
LUIS@GARCIA.PRO.BR



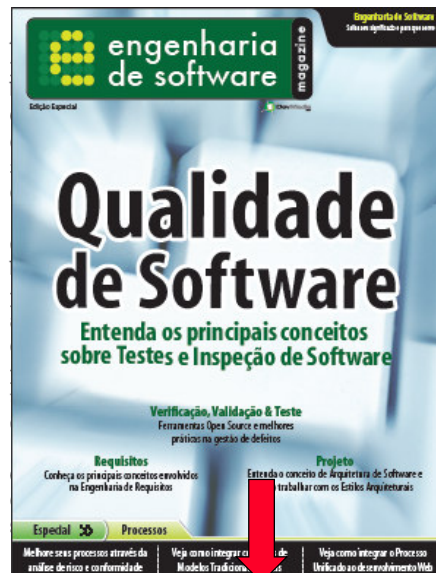
Ana Luiza Ávila
analuizavila@yahoo.com.br

É bacharel em Ciências da Computação pela Universidade Salvador (UNIFACS) e Mestre em Ciências da Computação pela PUC-Rio na linha de Engenharia de Software.

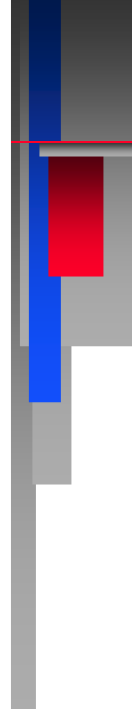


Rodrigo Oliveira Spínola
rodrigo@sfmagazine.com.br

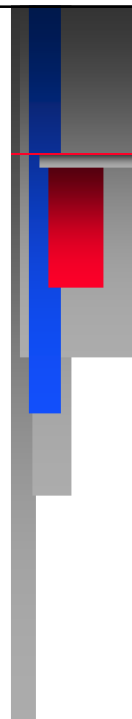
Doutorando em Engenharia de Sistemas e Computação (COPPE/UFRJ). Mestre em Engenharia de Software (COPPE/UFRJ, 2004). Bacharel em Ciências da Computação (UNIFACS, 2001). Colaborador da Kali Software (www.kalisoftware.com), tendo ministrado cursos na área de Qualidade de Produtos e Processos de Software, Requisitos e Desenvolvimento Orientado a Objetos. Consultor para implementação do MPS.BR. Atua como Gerente de Projeto em projetos de consultoria na COPPE/UFRJ. É colaborador da Engenharia de Software Magazine.



46 Engenharia de Software Magazine – Introdução à Engenharia de Requisitos



Desenvolver software é uma atividade complexa por natureza. Uma das razões para esta afirmação é que não existe uma única solução para cada cenário de desenvolvimento. Além disso, lidamos o tempo todo com pessoas, o que torna o sucesso do projeto bastante relacionado à competência da equipe e à forma como trabalham, e, para dificultar ainda mais, muitas vezes não fazemos uso de um processo bem definido para apoiar as atividades do projeto.



desenvolvimento. Uma recente matéria publicada na revista Exame relata o crescimento do número de empresas que atingiram níveis de maturidade considerando modelos como MPS.BR e CMMI. Este resultado é um forte indicador de que as empresas nacionais estão se preocupando com a qualidade dos serviços que oferecem, conseguindo, dessa forma, uma inserção maior no mercado internacional de desenvolvimento de software.

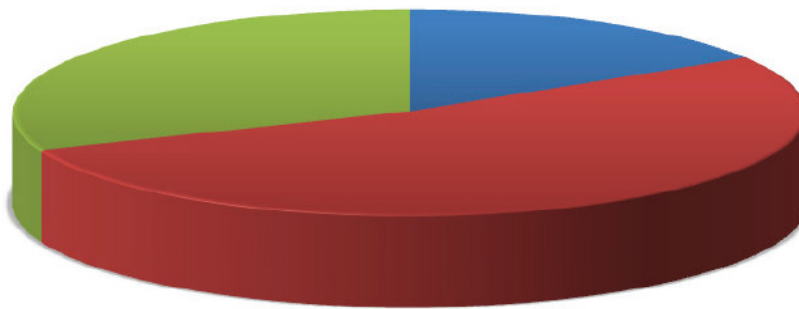
	% do Custo de Desenvolvimento	% dos erros introduzidos	% dos erros encontrados	Custo relativo de correção
Análise de Requisitos	5	55	18	1
Projeto	25	30	10	1 - 1.5
Códificação e teste de unidade	50			
Teste	10	10	50	1 - 5
Validação e Documentação	10			
Manutenção		5	22	10 - 100

Tabela 1. Cenário atual de desenvolvimento.

Neste ponto, é importante analisamos os dados da **Tabela 1**. Perceba que o custo das atividades relacionadas à análise de requisitos é baixo. Por outro lado, é nesta fase que grande parte dos defeitos são inseridos. Podemos perceber ainda analisando a primeira linha que o custo de correção destes problemas nesta fase é baixo. Continuando a análise, percebemos que estes defeitos não são tratados no momento devido, o que pode aumentar bastante o custo com o projeto.

Reforçando a importância que as atividades relacionadas a requisitos devem possuir na indústria de software, estudo realizado pelo Standish Group, considerando 350 companhias e 8.000 projetos de software, em 1995 revelou que (ver **Figura 1**):

- 16,2% dos projetos são finalizados com sucesso, ou seja, cobre todas as funcionalidades em tempo e dentro do custo previsto;
- 52,7% dos projetos são considerados problemáticos, ou seja, não cobre todas as funcionalidades exigidas, custo aumentado e está atrasado.
- 31,1% dos projetos fracassam, ou seja, o projeto é cancelado durante o desenvolvimento.




■ Sucesso ■ Problemático ■ Fracasso

Figura 1. Distribuição da conclusão dos projetos de software.



Fatores Críticos	%
1. Requisitos Incompletos	13,1%
2. Falta de Envolvimento do Usuário	12,4%
3. Falta de Recursos	10,6%
4. Expectativas Irreais	9,9%
5. Falta de Apoio Executivo	9,3%
6. Mudança de Requisitos e Especificações	8,7%
7. Falta de Planejamento	8,1%
8. Sistema não mais necessário	7,5%

Tabela 2. Fatores críticos do sucesso.



dos principais fatores estão relacionados às atividades de requisitos: (1) Requisitos Incompletos; (2) Falta de Envolvimento do Usuário; (6) Mudança de Requisitos e Especificações.

Neste ponto, sabemos que um trabalho mais criterioso na área de requisitos é fundamental para o sucesso de projetos de software. A partir da próxima seção

Requisitos

Existem diferentes definições encontradas na literatura técnica para requisitos:

- Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos;
- As descrições das funções e restrições são os requisitos do sistema;
- Um requisito é uma propriedade que o software deve exibir para resolver algum problema no mundo real;
- Uma condição ou uma capacidade que deve ser alcançada ou estar presente em um sistema para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto...

no qual o cliente faz parte. É importante estar atento para esta definição: embora o requisito seja definido pelo cliente, nem sempre o que o cliente quer é o que o negócio precisa. Cabe à equipe de consultores identificar a real necessidade do negócio.

Neste contexto, requisitos são importantes para:

- Estabelecer uma base de concordância entre o cliente e o fornecedor sobre o que o software fará;
- Fornecer uma referência para a validação do produto final;
- Reduzir o custo de desenvolvimento (como vimos anteriormente, requisitos mal definidos causam retrabalho).

Entendida a definição de requisitos, é preciso conhecer seus tipos.

Requisitos funcionais

São requisitos diretamente ligados a funcionalidade do software, descrevem as funções que o software deve executar. Alguns exemplos são:

- O software deve permitir o cadastro de clientes;
- O software deve permitir a geração de relatórios sobre o desempenho de vendas no semestre;
- O software deve permitir o pagamento das compras através de cartão de crédito.

Requisitos não funcionais

São requisitos que expressam condições que o software deve atender ou qualidades específicas que o software deve ter. Em vez de informar o que o sistema fará, os requisitos não-funcionais colocam restrições no sistema. Alguns exemplos são:

- O software deve ser compatível com os browsers IE (versão 5.0 ou superior) e Firefox (1.0 ou superior);
- O software deve garantir que o tempo de retorno das consultas não seja maior do que 5 segundos.

Requisitos de domínio

São requisitos derivados do domínio da aplicação e descrevem características do sistema e qualidades que refletem o domínio. Podem ser requisitos funcionais novos, restrições sobre requisitos existentes ou computações específicas. Dois exemplos de requisitos do domínio são:

- O cálculo da média final de cada aluno é dado pela fórmula: $(Nota1 * 2 + Nota2 * 3)/5$;
- Um aluno pode se matricular em uma disciplina desde que ele tenha sido aprovado nas disciplinas consideradas pré-requisitos.

Engenharia de Requisitos

Existem diferentes definições encontradas na literatura técnica para engenharia de requisitos:

- Termo usado para descrever as atividades relacionadas à investigação e definição de escopo de um sistema de software;
- Processo sistemático de desenvolvimento de requisitos através de um processo cooperativo de análise onde os resultados das observações são codificados em uma variedade de formatos e a acurácia das observações é constante-

Embora coerentes, estas definições podem ser melhoradas. Perceba que elas referem-se apenas às atividades relacionadas à produção de requisitos. Entretanto, nada é dito a respeito da gerência destas atividades, também conhecida como gerência de requisitos. Com isto em mente, podemos evoluir a definição de engenharia de requisitos para: termo usado para descrever as atividades relacionadas à produção (levantamento, registro, validação e verificação) e gerência (controle de mudanças, gerência de configuração, rastreabilidade, gerência de qualidade dos requisitos) de requisitos. A Figura 2 representa essa definição.

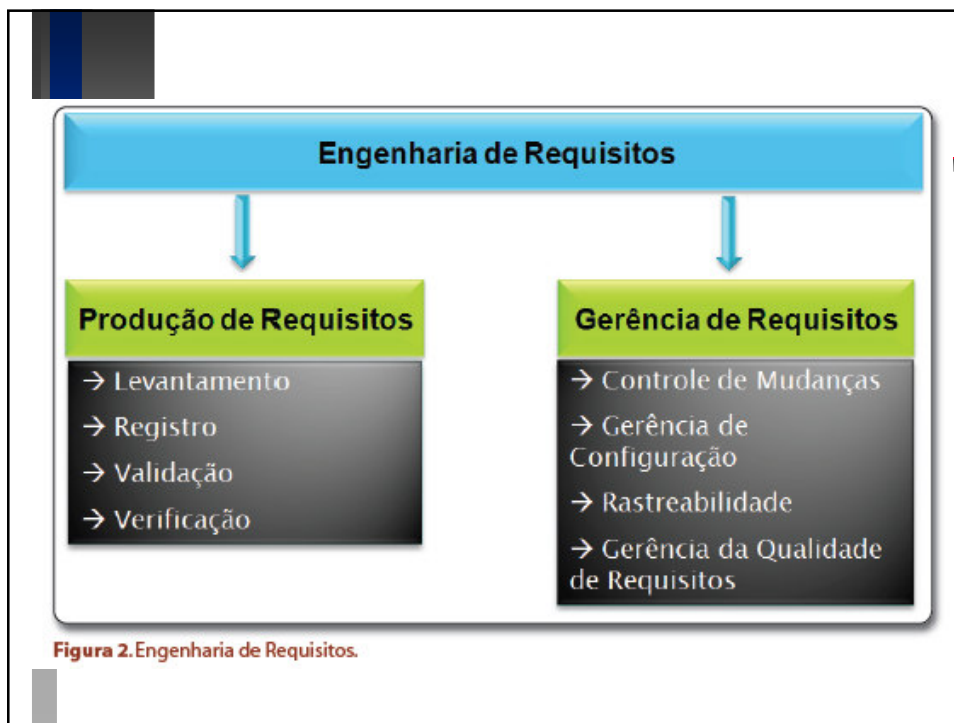


Figura 2. Engenharia de Requisitos.

Desta forma, os dois conceitos base (produção e gerência) devem ser considerados em conjunto ao se definir estratégias de trabalho com requisitos nas organizações (ver Figura 3).

Neste ponto podemos citar alguns dos principais objetivos da engenharia de requisitos:

- estabelecer uma visão comum entre o cliente e a equipe de projeto em relação aos requisitos que serão atendidos pelo projeto de software;
- registrar e acompanhar requisitos ao longo de todo o processo de desenvolvimento;
- documentar e controlar os requisitos alocados para estabelecer uma baseline para uso gerencial e da engenharia de software;
- manter planos, artefatos e atividades de software consistentes com os requisitos alocados.

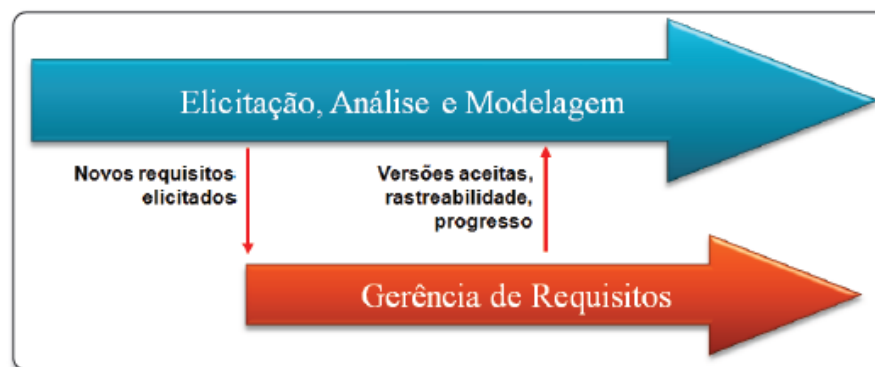


Figura 3. Produção e Gerência de Requisitos.

Para apoiar o alcance destes objetivos, é importante que se tenha um processo de engenharia de requisitos bem definido. Um processo de engenharia de requisitos é um conjunto estruturado de atividades a serem seguidas para criar, validar e manter um documento de requisitos. Poucas organizações têm um processo de ER explicitamente definido e padronizado. Entretanto, sugere-se que cada organização deva desenvolver um processo adequado à sua realidade. A Figura 4 apresenta um modelo genérico de atividades que pode descrever a maioria

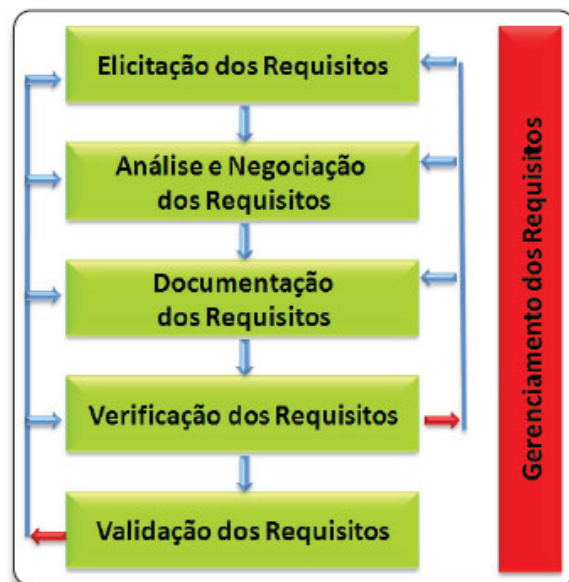
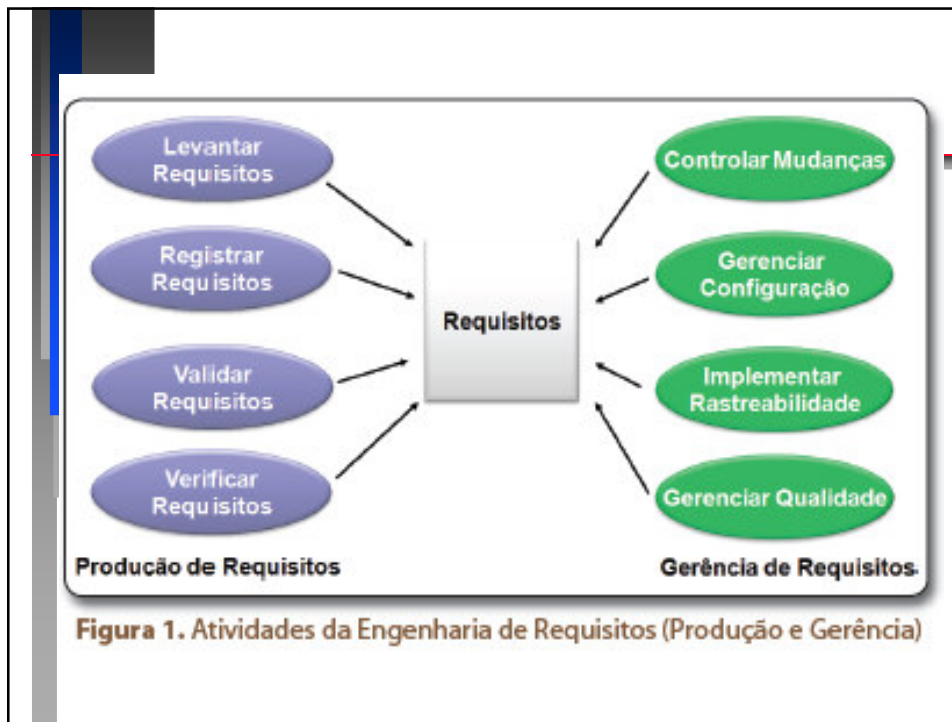


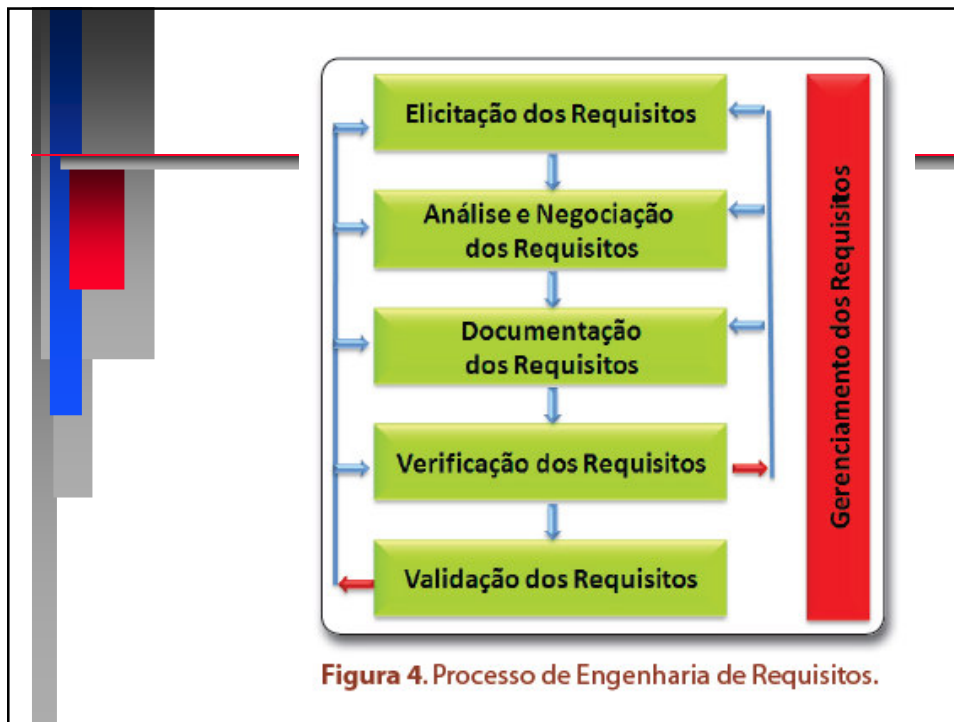
Figura 4. Processo de Engenharia de Requisitos.



Apesar do aparente fluxo entre as atividades, não existe uma fronteira explícita entre elas. Na prática existe muita sobreposição e interação entre uma atividade e outra.

Como entradas para o processo são consideradas:

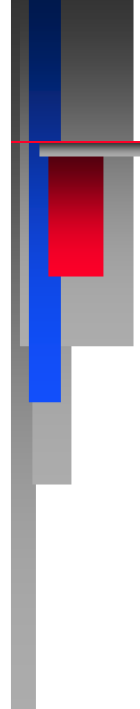
- Descrições do que os stakeholders necessitam para suportar suas atividades;
- Informações a respeito do sistema que será substituído ou de qualquer sistema com o qual o sistema sendo definido terá que interagir;
- Padrões vigentes na organização a respeito de práticas de desenvolvimento de sistemas, gerência de qualidade, etc.;
- Regulamentos externos, tais como leis, regulamentos de segurança ou saúde;
- Informações gerais sobre o domínio de aplicação.



Levantamento

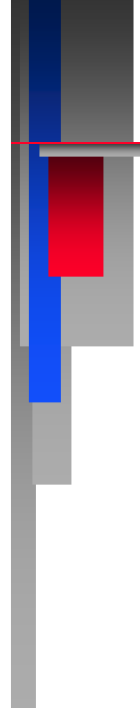
Esta atividade relaciona-se à obtenção dos requisitos do software. Para isto, analistas e engenheiros de software trabalham com clientes e usuários finais para descobrir o problema a ser resolvido, os serviços do sistema, o desempenho necessário, restrições de hardware e outras informações.

Existem algumas técnicas que apóiam as atividades de levantamento de requisitos. Uma breve descrição de algumas delas é:



- **Entrevista:** esta técnica resume-se em “conversas” realizadas com o usuário (entrevistado) para levantar os requisitos do sistema a ser desenvolvido. Podemos decompor esta técnica nas seguintes atividades:

- o Ler material de suporte;
- o Estabelecer os objetivos da entrevista;
- o Decidir quem entrevistar;
- o Preparar o entrevistado;
- o Decidir os tipos de questões e a sua estrutura.



- **Prototipação:** é uma versão inicial de um sistema para experimentação. Permite aos utilizadores identificar os pontos fortes e fracos do sistema por ser algo concreto que pode ser criticado. Temos dois tipos de protótipos:

- o Protótipos “Throw-away”: ajudam o levantamento e desenvolvimento dos requisitos e suportam os requisitos mais difíceis de perceber;
- o Protótipos Evolutivos: ajudam o desenvolvimento rápido de uma versão inicial do sistema e suportam os requisitos bem definidos e conhecidos.

• JAD (Joint Application Development):
é uma técnica que permite a interação
entre pessoas que necessitam tomar
decisões que afetem múltiplas áreas
de uma organização. Esta técnica en-

volve atividades de preparação para as
reuniões, sessões de workshop com os
participantes, agenda para as reuniões,
participantes assumindo papéis de faci-
litador / condutor e documentador além
de facilidades visuais, como a utilização
de flipchart, quadro negro.

Esta técnica deve ser utilizada nos casos
onde existe a necessidade de consenso
entre diversos usuários, pois possibilita
a todos os envolvidos ter uma visão glo-
bal do sistema, ajudando a consolidar
interesses de diversos usuários quanto
ao sistema a ser desenvolvido.

O objetivo desta técnica é aumentar
o comprometimento e participação do
usuário e obter subsídios para elaborar o
documento de Especificação de Requisi-
tos para o sistema com consenso de todos
de forma a ser uma validação formal dos
requisitos do sistema.

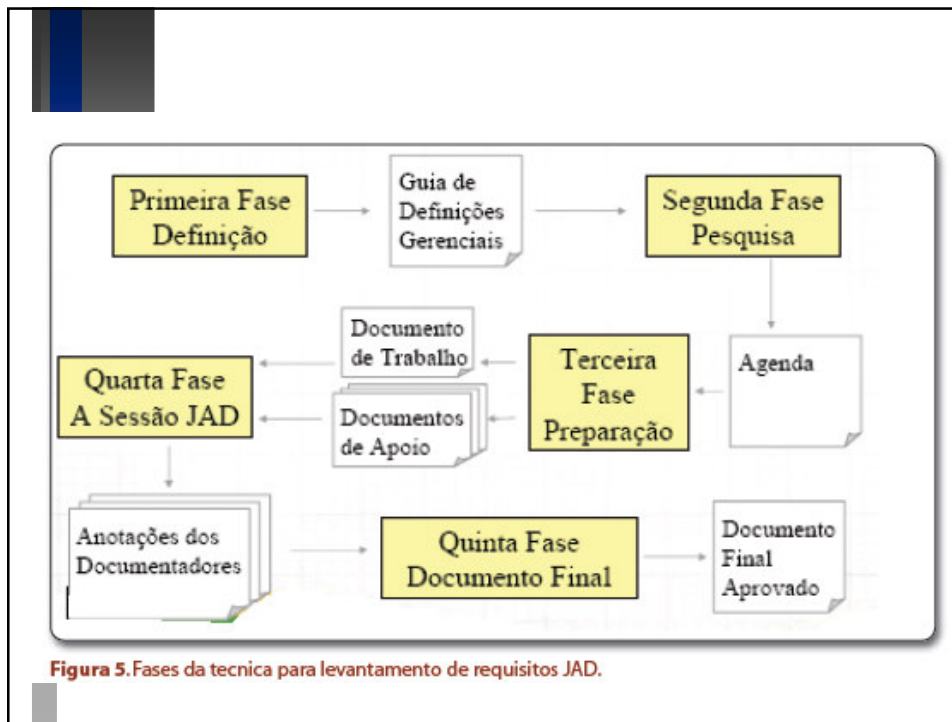


Figura 5. Fases da técnica para levantamento de requisitos JAD.

A atividade de levantamento de requisitos não é trivial. Existe um conjunto grande e variado de fatores que a tornam complexa, por exemplo:

- Falta de conhecimento do usuário das suas reais necessidades
 - Usuário com vaga noção do que precisa e do que um produto de software pode oferecer.
- Falta de conhecimento do desenvolvedor do domínio do problema
 - Desenvolvedor sem conhecimento adequado do domínio, o que leva a decisões erradas.
- Domínio do processo de levantamento

- de requisitos pelos desenvolvedores
- o Desenvolvedor não ouve o que os usuários têm a dizer e força suas próprias visões e interpretações.
- Comunicação inadequada entre os desenvolvedores e usuários
 - o Usuários incapazes de expressar suas necessidades apropriadamente;
 - o Significados diferentes a termos comuns.
 - Dificuldade do usuário tomar decisões
 - o Falta de entendimento sobre as conseqüências das decisões ou as alternativas possíveis.
 - Problemas de comportamento
 - o O levantamento de requisitos é um processo social;
 - o Conflitos e ambigüidades nos papéis que os usuários e desenvolvedores desempenham.
 - Questões técnicas
 - o Complexidade crescente dos sistemas atuais.

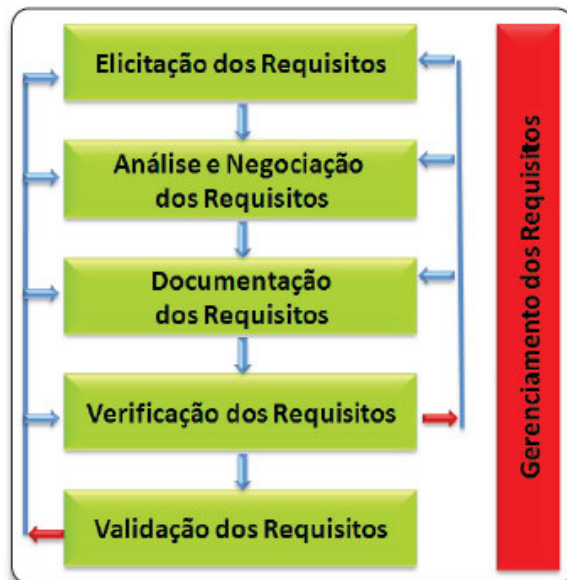


Figura 4. Processo de Engenharia de Requisitos.

Registro

Uma vez identificados e negociados, os requisitos devem ser documentados para que possam servir de base para o restante do processo de desenvolvimento.

Entre os muitos problemas que enfrentamos na documentação de requisitos, certamente, administrar o grande volume de informações gerado pelo processo de requisitos é um dos principais.

Os requisitos são documentados em um nível apropriado de detalhe. Em geral é produzido um documento de especificação de requisitos, de forma que todos os stakeholders possam entendê-lo.

O registro dos requisitos num documento próprio facilita o controle de alterações de todos os envolvidos na manutenção dos requisitos, bem como a geração de versões do documento e a facilidade de acesso por todos os envolvidos.

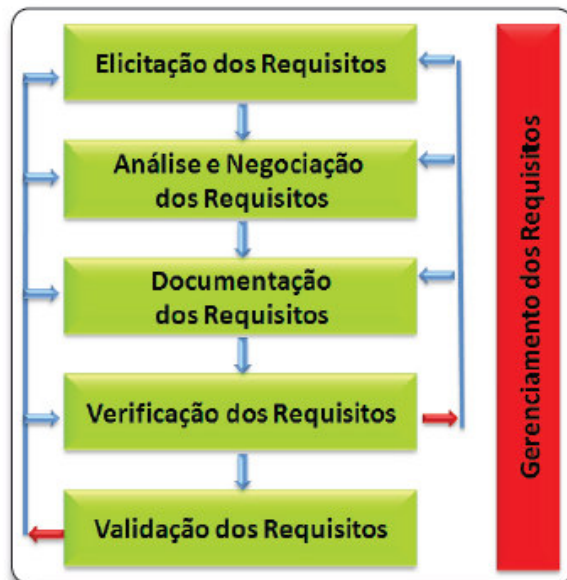


Figura 4. Processo de Engenharia de Requisitos.

Verificação

Esta atividade examina a especificação do software, de forma a assegurar que todos os requisitos foram definidos sem ambigüidades, inconsistências ou omissões, detectando e corrigindo possíveis problemas ainda durante a fase de definição dos requisitos.

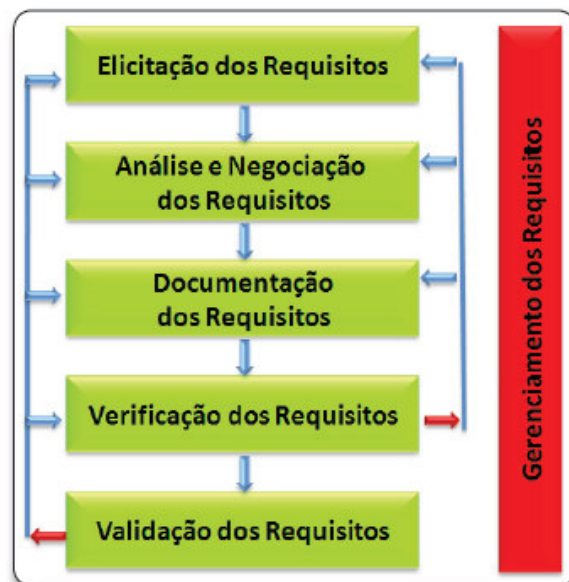


Figura 4. Processo de Engenharia de Requisitos.

Validação

A validação representa a atividade em que obtemos o aceite do cliente sob determinado artefato. No cenário de engenharia de requisitos, esta atividade significa aprovar junto ao cliente os requisitos que foram especificados. Embora

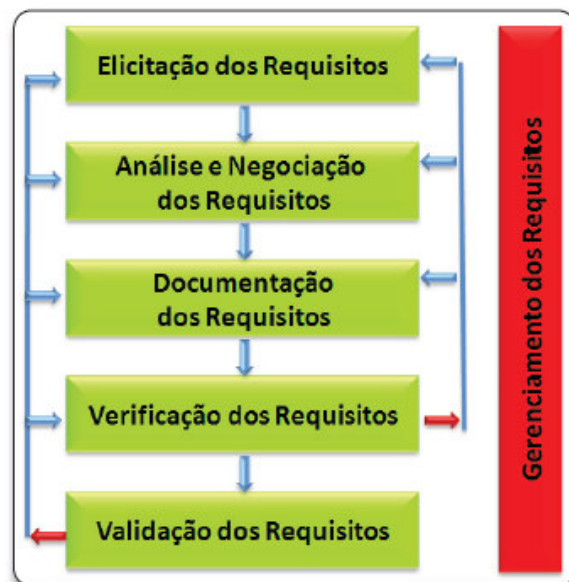


Figura 4. Processo de Engenharia de Requisitos.



Gerência de Requisitos

Requisitos são por natureza voláteis. Diversos fatores contribuem para sua instabilidade ao longo do tempo. Mudanças externas no ambiente (mudanças de legislação, mudanças no mercado, mudança no posicionamento estratégico da empresa), erros incorridos no processo de requisitos, entre outros. Todos esses fatores fazem com que seja necessário alterar os requisitos. Tais alterações precisam ser conduzidas de forma ordenada para que não se perca controle sobre o prazo e o custo do desenvolvimento.

Denominamos a atividade de administrar os requisitos ao longo do tempo de gerenciamento de requisitos. Os benefícios desta atividade são percebidos no médio prazo, sendo que são necessários investimentos no curto prazo. Assim, a atividade é, muitas vezes, tida como um fardo desnecessário à condução do processo. Contudo, sua não implementação

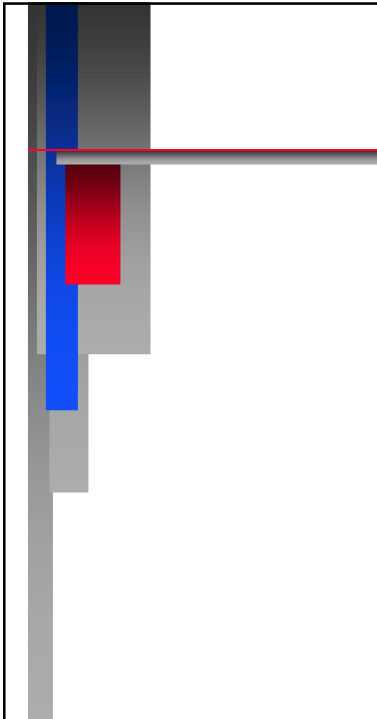


Controle de Mudanças

Conforme foi citado anteriormente, os requisitos são voláteis e, portanto sofrem mudanças ao longo do tempo, para conduzir estas mudanças recomenda-se preparo e planejamento. Uma das maneiras bastante utilizadas para organizar estas mudanças é a "baseline" de requisitos que nos permite diferenciar o que era o requisito original, o que foi introduzido e o que foi descartado. Além disto, é interessante estabelecer um único canal para controle de mudanças, bem como utilizar um sistema para este controle.

Apresentamos a seguir uma sugestão de passos que devem ser seguidos para um processo de controle de mudança:

- Checar validade da solicitação de mudança;
- Identificar os requisitos diretamente



afetados com a mudança;

- Identificar dependências entre requisitos para buscar os requisitos afetados indiretamente;
- Assegurar com solicitante a mudança a ser realizada;
- Estimar custos da mudança;
- Obter acordo com usuário sobre o custo da mudança.

Após a realização desta análise, podemos aceitar ou rejeitar a mudança, conforme os impactos e custos que ela pode ter no sistema.



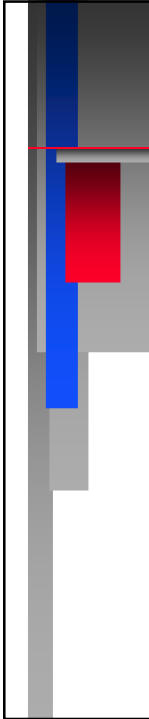
Gerência de Configuração

Durante o ciclo de vida do desenvolvimento, o software passa por uma série de modificações, desde a especificação dos requisitos até a implantação do sistema. A gerência de configuração de software existe no intuito de definir critérios que permitam realizar tais modificações mantendo-se a consistência e a integridade do software com as especificações, minimizando problemas decorrentes ao processo de desenvolvimento, através de um controle sistemático sobre as modificações.

A criação de um plano de gerência de configuração consiste em estabelecer normas, ferramentas e templates que permitam gerenciar de maneira satisfatória os itens de configuração de um sistema, que são definidos por Pressman como: “cada um dos elementos de informação que são criados durante o desenvolvimento de um produto de software, ou que para este desenvolvimento sejam necessários, que são identificados de maneira única e cuja evolução é passível de rastreamento”.

Rastreabilidade

No centro da atividade de gerenciamento de requisitos está a rastreabilidade. Esta é definida como a habilidade de se acompanhar a vida de um requisito em ambas as direções (por exemplo: partindo de requisitos e chegando a projeto ou, partindo de projeto e chegando a requisitos) do processo de software e durante todo o seu ciclo de vida.



A dificuldade envolvida com a rastreabilidade está no grande volume de informações gerado. As informações do processo de requisitos devem ser catalogadas e associadas aos outros elementos de forma que possam ser referenciadas através dos diversos itens de informação registrados. É um trabalho extenso que, sem o auxílio de ferramentas adequadas, muito provavelmente não poderá ser feito

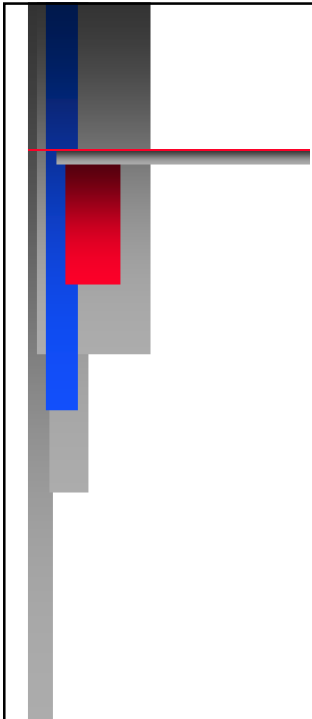
Para implementar a rastreabilidade, usualmente é confeccionado em conjunto com a especificação de requisitos um artefato chamado matriz de rastreabilidade, que tem como objetivo mapear os rastros dos requisitos descritos na especificação.



Gerência de Qualidade de Requisitos

Segundo o padrão IEEE 830, devemos considerar alguns critérios de qualidade ao trabalharmos com requisitos:

- **Correção:** um documento de requisitos é considerado correto se todos os requisitos representam algo que deve estar presente no sistema que está sendo desenvolvido, ou seja, os requisitos



reais do usuário devem coincidir com os requisitos identificados. Esta não é uma tarefa trivial e parte de seu sucesso está associada a uma boa atividade de validação dos requisitos.

- Não ambigüidade: um conjunto de requisitos é não ambíguo quando somente pode ser interpretado por todos os envolvidos em um projeto de uma única maneira.

- Completude: um conjunto de requisitos é dito completo quando descreve todas as demandas de interesse dos usuários. Estas demandas incluem requisitos funcionais, de desempenho, restrições, atributos e interfaces externas.

- Consistência: um conjunto de requisitos é dito consistente se nenhum subconjunto destes requisitos entra em conflito com os demais requisitos do sistema.

- Verificabilidade: um requisito é verificável se existe uma forma efetiva, em termos de tempo e custo, para que pessoas ou ferramentas indiquem se um sistema cumpre o requisito (IEEE). Em quase todas as situações, é difícil provar de forma conclusiva que um requisito é cumprido por um software. Entretanto, escrever bem o requisito pode ajudar a aumentar a confiança na avaliação.

- Modificabilidade: um conjunto de requisitos é modificável quando seu estilo e estrutura é tal que as alterações podem ser realizadas de forma simples e consistente com os demais requisitos.

Soluções Concretas para Problemas Práticos da Engenharia de Requisitos

Marcelo Nascimento Costa
marcel@kallsoft.com.br
 Diretor de Tecnologia da Kall Software. Mestre em Engenharia de Sistemas e Computação pela COPPE/UFPE. Doutor em Ciência da Computação pela UFPA. Especialista em CMMI, tendo participado de diversos implementações e análises deste modelo. Professor do curso de Ciência da Computação do Centro Universitário Metodista Bennett.

Marcos Kallneveski
mkall@kallsoft.com.br
 Diretor Executivo da Kall Software. Doutorando e mestre em Engenharia de Software da COPPE/UFPE. Bacharel em Ciência da Computação pela UFPA. Consultor de implementação, instrutor, avaliador e membro da equipe técnica do MPS.BR, tendo participado de diversas análises deste modelo. Professor do curso de Ciência da Computação do Centro Universitário Metodista Bennett. Professor de pós-graduação e IT Expert da UFPA.

Rodrigo Oliveira Spínola
rod@kallsoft.com.br, rod@spinoia.com.br
 Doutorando em Engenharia de Sistemas e Computação (COPPE/UFPE), Mestre em Engenharia de Software (COPPE/UFPE, 2004), Bacharel em Ciência da Computação (UNIFACS, 2001). Colaborador da Kall Software (www.kallsoft.com.br), tendo ministrado cursos na área de Qualidade de Produto e Processos de Software, Requisitos e Desenvolvimento Orientado a Objetos. Consultor para implementação do MPS.BR. Atua como Gerente de Projeto e Análise de Requisitos em projetos de consultoria na COPPE/UFPE. É Colaborador da Engenharia de Software Magazine.

De que se trata...
 Consideram um processo...
 seja capaz...
 relacionado...
 tante o des...
 lma inicia...
 dificuldade...
 tutar um pr...
 encontra-s...

Para que...
 Existem di...
 processo co...
 ta de requ...
 de softwa...
 tudade, c...
 arcabouço...
 exigências...

A tua...
 tã o...
 de...
 processos p...
 desenvolver...
 adoção de m...
 MPS (SOF...
 2006) é umi...
 nacionais e...
 qualidade...

boas práticas da engenharia de software. Uma das atividades fundamentais da engenharia de software exigidas desde os níveis iniciais dos modelos é a engenharia de requisitos (Campos et al., 2008 – artigo publicado na edição 7 da Engenharia de Software Magazine). A importância dos requisitos e alguns conceitos iniciais relacionados à engenharia de requisitos do software são destacados em (Ávila e Spínola, 2008 – artigo publicado na edição 1 da Engenharia de Software Magazine).

Existem diferentes maneiras de estruturar um processo com atividades relacionadas à engenharia de requisitos em empresas desenvolvedoras de software. Acreditamos que modelos de maturidade, como o MPS, possam servir como um arcabouço para a definição deste processo. As exigências relacionadas à engenharia de requisitos no modelo MPS podem ser encontradas em (SOFTEX, 2007a) e maneiras de implementar estas exigências podem ser encontradas em (SOFTEX, 2007b). Além

Tratando Problemas Práticos da Engenharia de Requisitos

As subseções seguintes apresentam dificuldades e problemas agrupados por atividade da engenharia de requisitos, tendo em vista as atividades do modelo genérico de atividades descrito na **Figura 3**. Foram identificados problemas também na fase que é executada durante a execução das demais atividades da engenharia de requisitos, a gerência de requisitos. Esta lista de dificuldades e problemas foi obtida com base na experiência prática de anos de consultoria de apoio para a implantação do processo de engenharia de requisitos em empresas de

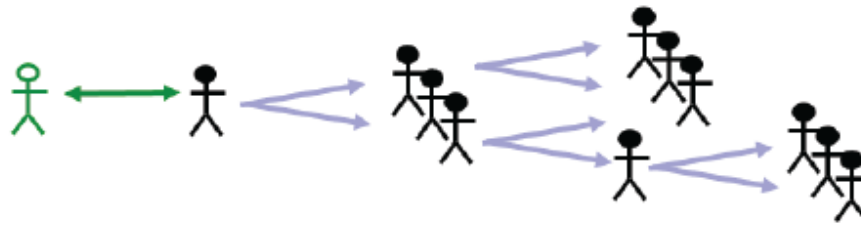
Tabela 1. Problemas e soluções associados à elicitação de requisitos

Problema	Elicitação de
Envolver interessados inapropriados. Os envolvidos no suporte ao levantamento dos requisitos não trabalham ou não conhecem totalmente a operação do sistema. Por exemplo, a organização necessita melhorar a usabilidade (interface) do software para melhorar o tempo de atendimento dos clientes da organização. Nesta situação, o analista de requisitos precisa entrevistar os envolvidos especificamente com o operacional da aplicação e, em algumas situações, a gerência não permite que a equipe operacional se envolva na definição da solução. O problema fundamental consiste que o gerente possui uma visão apenas estratégica/tática do sistema, porém não "sofre" com os problemas específicos da usabilidade dos sistemas. Provavelmente, ele fornecerá uma solução que lhe parece mais adequada, mas que pode não resolver efetivamente o problema para qual o projeto foi definido.	

Tabela 1. Problemas e soluções associados à elicitação de requisitos

Discussão da Solução

Uma das maneiras de endereçar este problema é estruturar o processo de elicitação de modo a envolver uma entrevista inicial com o patrocinador (diretor, gerência) para identificar os interessados, identificando seus papéis e interesses. Adicionalmente, durante as entrevistas, pode ser interessante



Uma vez identificados, os mesmos devem ser envolvidos no processo, participando de entrevistas ou de atividades como workshops de requisitos ou sessões JAD (Wieggers, 2003).

Tabela 1. Problemas e soluções associados à elicitação de requisitos

Problema

Problemas políticos na organização. A organização passa por problemas políticos, incluindo a reestruturação de departamentos e papéis/responsabilidades. Nesses casos, não existe uma definição de quem vai realmente ser responsável pelo sistema. O atual responsável pela área, que utilizará o sistema, participa do processo de elicitação de requisitos, modelando o escopo do sistema de acordo com as necessidades atuais da organização. Após a aprovação desse escopo, a organização sofre a reestruturação e o novo responsável, designado para esta área, possui outra visão do negócio e solicita a redefinição completa do escopo.

Tabela 1. Problemas e soluções associados à elicitação de requisitos

Discussão da Solução

Este risco pode ser mitigado com a utilização de um ciclo de vida iterativo incremental, onde nem toda a funcionalidade é definida no início do processo.

Adicionalmente, para cada um dos incrementos funcionais produzidos, o processo de engenharia de requisitos deve prever a obtenção explícita do comprometimento com os requisitos definidos.

A partir da obtenção deste comprometimento, os requisitos devem poder ser alterados somente através do processo de gerência de requisitos, que envolve a análise de impacto das modificações solicitadas. Dependendo do impacto, alterações contratuais podem ser necessárias.

Análise e Negociação de Requisitos

Problema

A linguagem natural e a abstração os requisitos de alto nível dificultam o mapeamento das capacidades macro em requisitos funcionais e não funcionais. Constantemente, os requisitos de usuário são descritos de uma forma muito abstrata, o que dificulta a tradução para requisitos funcionais e não-funcionais do sistema.

Essa dificuldade se origina também de casos em que os usuários não têm a visão correta das necessidades do negócio que serão atendidas e qual a situação futura do sistema após a entrada em produção. Essa falta de visão pode tornar os requisitos de alto nível muito abstratos, deixando a definição de funcionalidades concretas sob responsabilidade do engenheiro de requisitos.

Análise e Negociação de Requisitos

Discussão da Solução

Para tratar esta dificuldade é necessário aumentar a interação entre o engenheiro de requisitos e o usuário. Isto pode ser atingido através de workshops de requisitos ou sessões JAD envolvendo os interessados. Outra prática que ajuda a entender bem os objetivos de negócio do cliente é iniciar as entrevistas da elicitação com perguntas abertas, do tipo:

Quais são os problemas?

Por que eles precisam ser resolvidos?

Existem outras razões para eles serem resolvidos?

Quais os benefícios esperados de uma solução bem sucedida?

Como esses problemas são resolvidos hoje? Qual é a situação atual?

Qual seria a situação desejada, ou seja, como você gostaria de resolver os problemas?

Pode ser interessante ainda verificar a possibilidade de empregar técnicas de elicitação complementares como etnografia (observação do ambiente operacional do cliente). As definições funcionais realizadas a partir dos macro requisitos podem ser validadas junto ao usuário utilizando prototipação e revisões.

Análise e Negociação de Requisitos

Problema

Separar premissas relacionadas ao desenvolvimento do sistema dos seus requisitos. Comumente ocorre confusão entre requisitos não funcionais e premissas de desenvolvimento. Podemos definir premissas como o conjunto de condições para que o sistema possa ser desenvolvido ou possa entrar em produção. Nestes casos, é comum, por exemplo, o analista de requisitos definir as fases de desenvolvimento do sistema, o nome do usuário que deverá validar o sistema, o número de horas por cada fase, entre outros. Estes exemplos, são claramente identificados como premissas para o desenvolvimento do sistema e não podem ser classificados como requisito não-funcional de nenhum tipo.

Análise e Negociação de Requisitos

Discussão da Solução

Este problema pode ser endereçado estabelecendo templates e diretrizes que orientem a separação adequada do conteúdo do documento de requisitos e fazer uso de revisões técnicas para assegurar a qualidade dos documentos produzidos.

Documentação de Requisitos

Problema

Dificuldade na descrição de requisitos funcionais e não-funcionais. Os requisitos funcionais e não-funcionais (conforme definição da seção 2) devem ser claros para o entendimento do usuário e também para permitir a correta especificação dos casos de uso durante a fase de especificação de requisitos. Alguns problemas que encontramos na descrição dos requisitos funcionais:

Omissão - Informação necessária não incluída;

Ambigüidade – Informação passível de ter múltiplas interpretações;

Inconsistência - Informações conflitantes;

Fato Incorreto - Informação que não é verdadeira para as condições especificadas.

Informação Estranha - Informação desnecessária.

Documentação de Requisitos

Discussão da Solução

Um instrumento que pode ser utilizado para encontrar problemas na descrição dos requisitos funcionais é a utilização de revisões técnicas formais.

Entre as revisões técnicas e formais destacamos as inspeções e os walkthroughs. Um dos benefícios das inspeções é a disseminação das boas práticas na organização. Adicionalmente, treinamento pode ser conduzido focando nos problemas encontrados nas inspeções. Mais informações sobre inspeções de requisitos podem ser encontradas em (Kalinowski et al., 2007).

Documentação de Requisitos

Problema

Complexidade no detalhamento dos casos de uso para definição da solução. Constantemente, os casos de uso ficam extremamente complexos, com, por exemplo, 30 passos no fluxo principal, 5 fluxos alternativos, 5 fluxos de exceção e dezenas de regras de negócios. Nestes casos, a leitura do caso de uso se torna bastante cansativa e difícil para os usuários e, até mesmo, para os envolvidos na área de TI da organização.

Documentação de Requisitos

Discussão da Solução

Este problema pode ser de difícil solução, uma vez que a complexidade pode ser inerente à solução sendo descrita. Uma recomendação para facilitar a compreensão do caso de uso é manter os passos simplificados, contudo assegurando que todas as informações necessárias (informações recebidas, opções disponibilizadas, informações fornecidas e ações realizadas) estejam descritas. A separação do caso de uso em diversos casos de uso relacionados através de inclusões (includes) e extensões (extends) pode ser pensada, desde que os casos de uso resultantes atendam a um objetivo facilmente compreendido (Cockburn, 2007).

Documentação de Requisitos

Problema

Falta de padronização no detalhamento de casos de uso. A falta de padronização dentro de uma mesma organização leva a dificuldades no entendimento das soluções e na garantia da qualidade da especificação de requisitos produzida.

Discussão da Solução

Novamente a solução envolve estabelecer um padrão dentro da organização, definindo explicitamente um template e diretrizes de preenchimento. Adicionalmente é interessante estabelecer um processo de garantia da qualidade para assegurar que os artefatos produzidos na engenharia de requisitos sigam corretamente os padrões estabelecidos.

Problema

Alteração constante dos requisitos, causando retrabalho. Modificações em alguns requisitos podem causar grande impacto sobre o esforço de desenvolvimento do software. Por exemplo, a alteração de um requisito não-funcional relacionado ao desempenho do produto (tempo de resposta), pode tornar toda uma arquitetura de solução (por exemplo, em camadas) inadequada.

Discussão da Solução

Em um cenário de constantes mudanças a utilização de um ciclo de vida iterativo incremental, onde nem toda a funcionalidade é definida no início do processo, pode se mostrar mais adequada. Adicionalmente, para cada um dos incrementos funcionais produzidos, o processo de engenharia de requisitos deve prever a obtenção explícita do comprometimento com os requisitos definidos. A partir da obtenção deste comprometimento, os requisitos devem poder ser alterados somente através do processo de gestão de requisitos, que envolve a análise de impacto das modificações solicitadas. A análise de impacto pode ser facilitada caso a rastreabilidade entre requisitos e os demais produtos de trabalho possa ser identificada. Dependendo do impacto, alterações contratuais podem ser necessárias.

Documentação de Requisitos

Problema	Discussão da Solução
Complexidade na definição da solução funcional. Comumente documentos de requisitos possuem diversos casos de uso complexos. Alguns documentos atingem o tamanho de centenas de páginas, dificultando o detalhamento e entendimento de todos os requisitos necessários para o projeto.	Quebrar o sistema em módulos ou conjuntos de caso de uso que possam ser produzidos em iterações separadas, seguindo um ciclo de vida iterativo incremental. As especificações destes módulos devem ser tratadas em documentos separados e consequentemente menores.
Problema	Discussão da Solução
Detalhamento técnico desnecessário durante a especificação funcional do sistema. Uma parte considerável dos analistas de requisitos veio de equipes de desenvolvimento e, invariavelmente, utilizam termos técnicos (exemplos: formatos de arquivos XML, conversões de dados, comunicação pela rede, etc.) para a especificação dos requisitos. A especificação funcional deve se limitar a descrever o que o sistema realiza e não como (Wieggers, 2003).	Um instrumento que pode ser utilizado para encontrar este tipo de problema na especificação funcional é a utilização de revisões técnicas formais. Adicionalmente, treinamento pode ser fornecido para esclarecer o conteúdo apropriado da especificação funcional.

Verificação de Requisitos

Verificação de Requisitos	
Problema	Discussão da Solução
Requisito não funcionais não verificáveis. Alguns requisitos não funcionais (desempenho, confiabilidade, portabilidade, usabilidade, etc.) são frequentemente descritos de forma que não possam ser verificados durante os testes. Por exemplo, como testar que um software deve ser rápido, que uma interface deve ser amigável, que um software é confiável, ou que um software deve possuir portabilidade.	<p>Devido à sua própria definição, requisitos não-funcionais devem ser mensuráveis. Assim, deve-se associar forma de medida/referência a cada requisito não-funcional elicitado.</p> <p>A organização pode disponibilizar exemplos de unidades de medida para cada tipo de requisito não funcional. Alguns destes exemplos seguem:</p> <p>Desempenho: tempo de resposta em segundos para uma dada configuração de ambiente, número de transações processadas por segundo, etc.</p> <p>Usabilidade: tempo de treinamento necessário, número de quadros de ajuda, etc.</p> <p>Confiabilidade: tempo médio entre falhas, taxa de ocorrência de falhas, etc.</p> <p>Portabilidade: sistemas operacionais destino, browsers para visualização, etc.</p> <p>As revisões técnicas formais devem assegurar que os requisitos não funcionais estejam descritos de forma verificável.</p>

Validação de Requisitos	
Problema	Discussão da Solução
<p>Dificuldades para validação de casos de uso por parte do usuário. O usuário comum (não envolvido com a área de tecnologia da informação) pode possuir dificuldade no entendimento da semântica da notação de herança, pois não consegue abstrair o comportamento comum e as diferenças entre os casos de uso pai e filho. As duas outras notações semânticas includes e extends complicam a validação pela necessidade de navegação entre os casos de uso e depois o retorno para o caso de uso que chamou o caso estendido ou incluído. Algumas situações existem mais de três níveis de chamada de casos de uso, ou seja, o caso de uso X inclui o Y, que inclui o Z e que, finalmente, chama o W. Essa estratégia dificulta a compreensão do detalhamento do caso de uso.</p>	<p>Esta dificuldade é inerente da descrição funcional através de casos de uso.</p> <p>O que pode amenizar este problema é evitar o uso excessivo de heranças entre casos de uso e treinar o usuário na leitura de documentos que envolvam descrições de casos de uso. A visualização da solução descrita nos casos de uso pode ainda ser facilitada através do uso de protótipos de telas.</p> <p>Outra prática interessante para a validação, quando o usuário tem dificuldade na leitura dos documentos, é a utilização de walkthroughs (Melo et al., 2007) em que uma apresentação a respeito do conteúdo do documento é realizada (muitas vezes pelo próprio autor do documento), explicando os diferentes casos de uso.</p> <p>Outra prática interessante neste caso seria o uso de protótipos para apoiar o entendimento dos requisitos por parte do usuário.</p>
Problema	Discussão da Solução
<p>Documentos funcionais grandes dificultam a validação. Como foi citado acima, um documento funcional muito extenso e complexo dificulta a validação da definição funcional, principalmente por parte do usuário, dificultando o entendimento da solução funcional completa. Neste cenário, muitas vezes é possível perceber que a validação por parte do usuário ocorreu de forma mais intensa nas primeiras páginas do documento.</p>	<p>O problema pode ser endereçado realizando mais de uma revisão de validação do documento selecionando amostras do documento envolvendo apenas alguns dos casos de uso. Neste caso é importante identificar os casos de uso que possuem relacionamentos entre si e que devem ser validados em conjunto.</p> <p>Pode ser interessante ainda acrescentar protótipos no documento de especificação funcional. Protótipos tendem a fornecer uma percepção mais clara da funcionalidade a ser desenvolvida e normalmente recebem mais atenção na validação por parte do usuário do que descrições de casos de uso.</p>

Gerência de Requisitos	
Problema	Discussão da Solução
<p>Dificuldades de estabelecer uma estratégia para a atualização e reutilização de casos de uso. Ter uma estratégia para reutilização de casos de uso pode ser importante, por exemplo, para a fase de manutenção em que novos casos de uso podem ser definidos reutilizando descrições anteriores. Não é comum a organização se preparar dessa maneira para a implantação do processo de gerência de requisitos, o que dificulta o armazenamento e atualização/reutilização dos casos de uso.</p>	<p>Para tratar este problema seria necessário criar uma biblioteca de casos de uso estruturada de acordo a divisão funcional (módulos) dos sistemas da organização e, preferencialmente, considerar cada caso de uso como um documento separado. Este documento poderia ser tratado como um item de configuração para armazenamento em um sistema de controle de versões.</p>
Problema	Discussão da Solução
<p>Como representar a atualização de casos de uso? Qual estratégia escolher para representar qual (is) passo(s) e regra(s) de negócio(s) foram alterados e qual tipo de solicitação de mudança que causou essa alteração no caso de uso. Desta forma, mantém-se um histórico da evolução do caso de uso.</p>	<p>Uma maneira de destacar as atualizações realizadas nos casos de uso é o preenchimento de um histórico de versões no início da descrição de cada caso de uso, informando a data, as alterações realizadas e o responsável pelas alterações. Adicionalmente, se o caso de uso é tratado como um item de configuração separado em um sistema de controle de versões, as alterações podem ser informadas no momento da atualização do repositório deste sistema.</p>

Problema	Discussão da Solução
Dificuldade de integração das práticas de gerência de requisitos com gerência de configuração. Um problema comum é a falta de disciplina dos analistas de requisitos para trabalhar integrado com as práticas da gerência de configuração.	Para tratar este problema é necessário definir um processo de gerência de configuração. A gerência de configuração envolve o controle de versões e a gerência de solicitações de mudança. Mais informações sobre gerência de configuração podem ser encontradas em (Dantas, 2008 – artigo publicado na edição 2 da Engenharia de Software Magazine). Neste processo a evolução dos requisitos deve estar alinhada com a evolução das baselines de requisitos, ou seja, deve-se conseguir identificar quais solicitações de mudança obrigaram a alterar o documento funcional já aprovado e validado pelos stakeholders. Os requisitos acordados em uma baseline somente podem ser alterados através de solicitações de mudança.
Problema	Discussão da Solução
Trabalhar com o backlog de casos de uso inexistentes para sistemas em manutenção. Muitos sistemas começam a ser definidos funcionalmente através de casos de uso depois de um certo tempo de entrada em produção. Nesta situação, não existem casos de uso descrevendo o sistema, então, quando surgem demandas evolutivas (pequenas alterações no sistema), acaba sendo necessária a descrição inteira dos casos de uso para se conseguir fornecer um contexto para os stakeholders que validarão os casos de uso.	Neste caso, esforço deve ser investido na engenharia reversa dos casos de uso (documentação dos casos de uso a partir dos sistemas existentes). Para sistemas muito grandes pode ser interessante considerar uma abordagem evolutiva para a criação dos casos de uso, onde inicialmente a documentação consiste de um resumo do caso de uso (contendo, por exemplo, apenas o fluxo principal). Casos de uso específicos podem ser refinados no momento em que uma manutenção na funcionalidade que eles descrevem for realizada.

Problema	Discussão da Solução
Dificuldades de implantação e manutenção da rastreabilidade dos requisitos. Muitas vezes a implantação na prática de uma estratégia para estabelecer a rastreabilidade não é trivial. Muitas ferramentas são pouco flexíveis e adaptáveis e planilhas acabam sendo usadas frequentemente. Assim, o esforço para criação da rastreabilidade pode ser alto.	Uma maneira de reduzir o esforço é estabelecer um processo de desenvolvimento que integre as atividades relacionadas ao estabelecimento da rastreabilidade com as próprias atividades de engenharia do software. Planilhas podem ser usadas, mas dificultam essa integração. Algumas ferramentas CASE permitem o estabelecimento dos links de rastreabilidade no momento da criação e atualização dos artefatos.
Problema	Discussão da Solução
Dificuldades de estabelecimento retroativo de rastreabilidade dos requisitos. Como criar uma matriz de rastreabilidade para sistemas pré-existentes em que a rastreabilidade não foi considerada durante o desenvolvimento.	O estabelecimento retroativo de rastreabilidade pode ser extremamente custoso e uma análise de custo/benefício deve ser considerada antes de realizar esta tarefa. Esta análise deve levar em consideração o volume de requisição de mudança no software e o impacto das modificações. Caso este volume não seja grande pode não ser interessante investir este esforço. Na medida em que manutenções forem realizadas a rastreabilidade referente a estas manutenções pode ser estabelecida.

Ferramentas	
Problema	Discussão da Solução
<p>Custo das ferramentas para gerência de requisitos. As ferramentas para suportar o processo de engenharia de requisitos possuem um custo considerável.</p>	<p>Para compensar o custo da ferramenta, o processo de engenharia de requisitos deve ser muito bem definido e compatível com a ferramenta. Adicionalmente, os recursos devem ser bem treinados e devem possuir um suporte adequado.</p> <p>É importante ressaltar que um processo de engenharia de requisitos pode ser estabelecido utilizando somente ferramentas gratuitas (editores, planilhas, sistemas de controle de versão, sistemas de registro de solicitações de mudança, etc). Tendo isto em vista, é necessário conhecer explicitamente as vantagens obtidas com a ferramenta a ser adquirida e realizar uma análise de custo/benefício.</p>

Recursos Humanos	
Problema	Discussão da Solução
<p>Falta de conhecimento dos analistas de requisitos em um domínio específico do problema. O domínio de aplicação pode se mostrar complexo, como por exemplo, financeiro, seguros e telecomunicações. Isto causa impacto na área de tecnologia de informação, pois idealmente o analista de requisitos deveria possuir um conhecimento de todas as áreas de domínio para poder detalhar corretamente as regras de negócio pertinentes ao domínio da aplicação. Caso não tenha o conhecimento desse domínio, o analista de requisitos poderá gerar um documento funcional com regras de negócios incorretas ou com um nível de abstração funcional muito alto.</p>	<p>O processo de elicitação deve prever esforço dedicado à compreensão do domínio. Este esforço deve ser explicitado tanto para o cliente quanto para a equipe do projeto.</p> <p>No caso de projetos em domínios muito complexos e específicos o envolvimento com os clientes deve ser intensificado e pode ser interessante considerar a contratação de suporte constante de consultores externos, especialistas no domínio.</p>