Artigo Engenharia de Software - Introdução à Engenharia de Requisitos

Neste artigo, faremos uma introdução à Engenharia de Requisitos, atividade base para as demais tarefas associadas ao desenvolvimento de software.

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.

Entende-se por processo, neste contexto, como sendo um conjunto de atividades bem definidas com os respectivos responsáveis por execução, ferramentas de apoio e artefatos produzidos. Ou seja, define-se como a equipe deverá trabalhar para alcançar o objetivo: desenvolver software com qualidade dentro de prazos, custos e requisitos definidos.

A boa notícia é que muitas empresas estão se movimentando no sentido de definirem detalhadamente seus processos para apoiarem suas atividades de 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.

Neste artigo, faremos uma introdução à Engenharia de Requisitos, atividade base para as demais tarefas associadas ao desenvolvimento de software.

> Relacionado: Curso de Engenharia de Software

Engenharia de Software e Requisitos

Vimos na introdução que se busca cada vez mais o apoio dos fundamentos da engenharia de software no desenvolvimento de sistemas. Entendemos engenharia de software como sendo, de acordo com o IEEE, a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, operação e manutenção de software. Sistemática por que parte do princípio de que existe um processo de desenvolvimento definindo as atividades que deverão ser executadas. Disciplinada por que parte do princípio de que os processos definidos serão seguidos. Quantificável por que se deve definir um conjunto de medidas a serem extraídas do processo durante o desenvolvimento de forma que as tomadas de decisão relacionadas ao desenvolvimento do software (por exemplo, melhoria de processo) sejam embasadas em dados reais, e não em “achismos”. Alguns de seus principais objetivos são:

- Qualidade de software;
- Produtividade no desenvolvimento, operação e manutenção de software;
- Permitir que profissionais tenham controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de qualidade desejados.

Entretanto, o cenário de desenvolvimento de software atual e o cenário idealizado junto à engenharia de software ainda estão distantes. Vários fatores contribuem para isso, podemos citar dois:

- O não uso dos fundamentos da engenharia de software para apoiar as atividades do desenvolvimento;
- O mau uso dos fundamentos da engenharia de software para apoiar as atividades do desenvolvimento.

Isso tem diversas conseqüências. Gostaríamos de destacar neste artigo o crescente custo com manutenção dos sistemas. Consideramos como manutenção neste artigo como sendo qualquer retrabalho (em nível de requisitos, projeto, codificação, teste) causado por uma definição do domínio do problema mal elaborada nas fases iniciais do 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.

Embora simples, esta análise nos permite concluir que é importante que seja dada maior importância às atividades relacionadas à especificação dos requisitos do 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.

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.

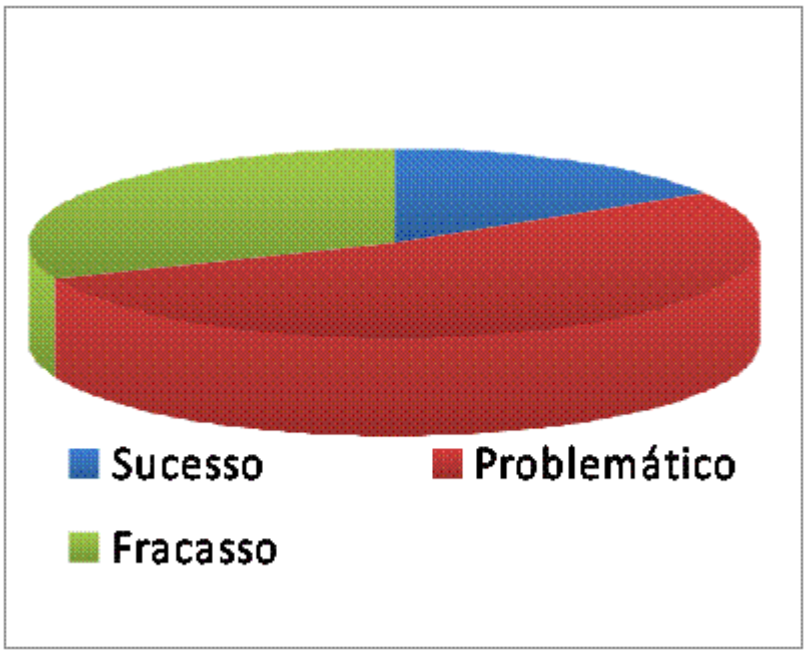

Tabela 2. Distribuição da conclusão dos projetos de software.

O Standish Group ainda fez uma análise sobre os fatores críticos para sucesso dos projetos de software. O resultado dos dez mais lembrados pode ser visto na **Tabela 2**. Podemos perceber que

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

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.

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 conheceremos a definição de requisitos e algumas de suas definições relacionadas antes de discutirmos mais profundamente a engenharia de requisitos.

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...

Percebe-se que as citações encontradas definem o mesmo conceito sob diferentes perspectivas. Podemos entender requisitos como sendo o conjunto de necessidades explicitadas pelo cliente que deverão ser atendidas para solucionar um determinado problema do negócio 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 calculo 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.

A partir da próxima seção apresentaremos os conceitos envolvidos na engenharia de 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 constantemente verificada;
- Processo de descobrir, analisar, documentar e verificar as funções e restrições do sistema.

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 3** representa essa definição.

Figura 3. 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 4**).

Figura 4. Produção e Gerência de Requisitos.

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.

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 5** apresenta um modelo genérico de atividades que pode descrever a maioria dos processos de engenharia de requisitos. Perceba que ele está concentrado nas atividades de produção dos requisitos.

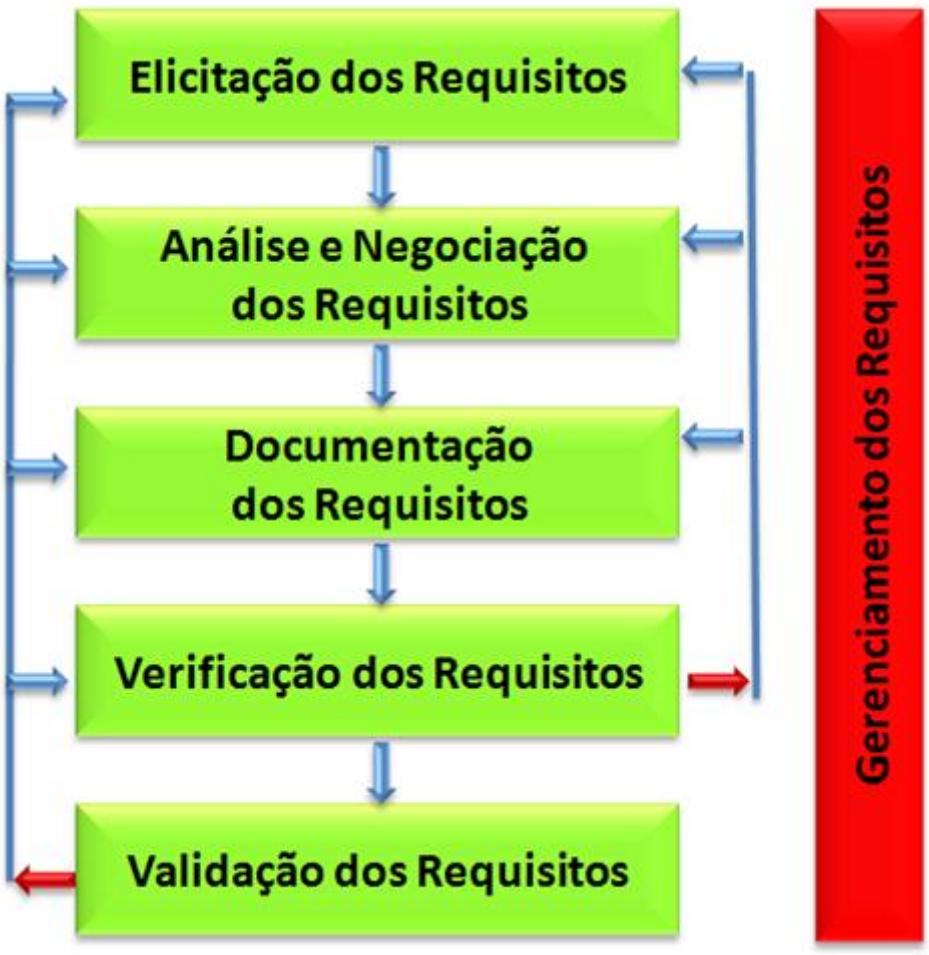

Figura 5. Processo de Engenharia de Requisitos.

Apesar do aparente fluxo entre as atividades, não existe uma fronteira explícita 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.

Em paralelo à execução das atividades definidas no processo da **Figura 5**, está o processo de gerência dos requisitos, voltado ao endereçamento de modificações nos requisitos. Os aspectos relacionados às atividades de gerência podem ser vistos na **Figura 3**.

A partir de agora conheceremos cada um dos conceitos que, em conjunto, definem a engenharia de requisitos.

Produção de Requisitos

A cada fase do ciclo de vida do software produzimos um documento contendo uma representação distinta do software a ser construído. Cada um desses documentos representa o software em um determinado nível de abstração. A tendência é diminuirmos o nível de abstração através da inclusão de mais e mais detalhes, até que, sua última representação seja o código fonte na linguagem escolhida.

Um dos artefatos produzidos no início do processo de desenvolvimento de software é a sua especificação de requisitos. Ele é base para as demais atividades de desenvolvimento e sua qualidade é fundamental para o sucesso do projeto. Uma especificação de requisitos bem elaborada é pré-requisito para um software de qualidade, embora não seja garantia disso. Desta forma, durante a produção de requisitos devemos possuir, além das atividades essenciais de levantamento e especificação, atividades relacionadas à garantia da qualidade. Conheceremos nesta seção as quatro atividades base relacionadas com a produção de requisitos.

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:
    - Ler material de suporte;
    - Estabelecer os objetivos da entrevista;
    - Decidir quem entrevistar;
    - Preparar o entrevistado;
    - Decidir os tipos de questões e a sua estrutura.
- Uma entrevista pode ser estruturada de três diferentes formas:
    - Estrutura em pirâmide: iniciamos as entrevistas com perguntas mais especificas sobre o sistema e fechamos com perguntas mais genéricas. Geralmente utilizadas com usuários mais relutantes;

- Estrutura em funil: iniciamos as entrevistas com perguntas mais genéricas sobre o sistema e fechamos com perguntas mais especificas. Geralmente utilizadas com usuários que tem uma relação mais afetiva com o assunto;
- Estrutura em diamante: esta estrutura combina as duas estruturas anteriores e é utilizadas para manter a usuário entrevistado interessado no assunto e para isto se utiliza de perguntas variadas.

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

Algumas das desvantagens da prototipação são os custos de aprendizagem e os custos de desenvolvimento.

- 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 envolve atividades de preparação para as reuniões, sessões de workshop com os participantes, agenda para as reuniões, participantes assumindo papeis de facilitador / 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 global 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 Requisitos para o sistema com consenso de todos de forma a ser uma validação formal dos requisitos do sistema.

O JAD é divido quem quatro fases. A **Figura 6** apresenta estas fases e os artefatos produzidos por cada uma delas.

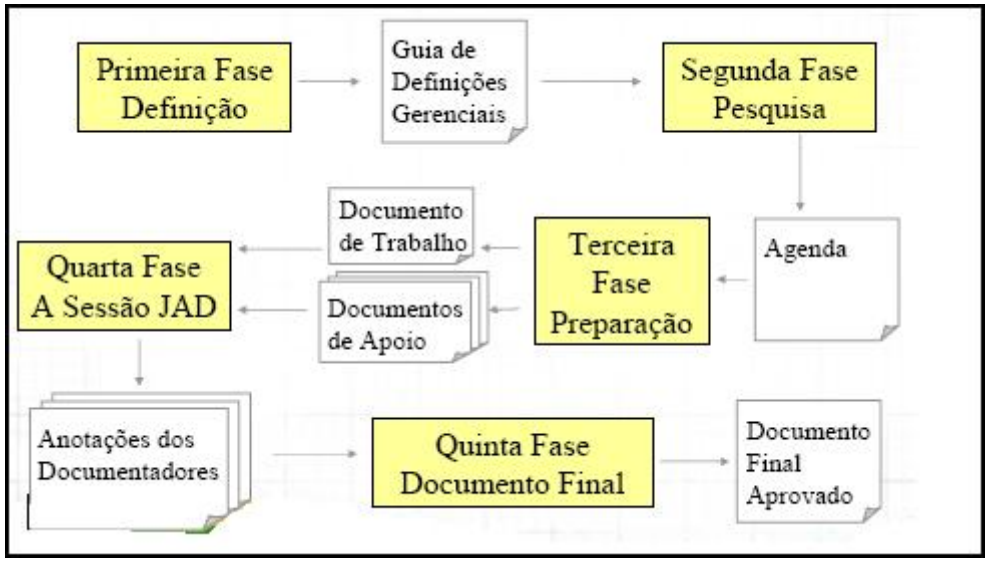

Figura 6. Fases da tecnica para levantamento de requisitos JAD.

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

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.

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.

Neste contexto, sabe-se que o custo da correção de defeitos aumenta na medida em que o processo de desenvolvimento progride. Revisões de artefatos de software têm se mostrado uma abordagem eficiente e de baixo custo para encontrar defeitos logo após terem sido introduzidos, reduzindo o retrabalho e melhorando a qualidade dos produtos. Não é em vão que modelos de maturidade de processo de software, como o CMMI e o MPS BR exigem a condução de revisões.

Um tipo particular de revisão de software são as inspeções. Inspeções possuem um processo de detecção de defeitos rigoroso e bem definido. Os benefícios de se aplicar inspeções são maiores para artefatos desenvolvidos no início do processo, como o documento de requisitos. Conheceremos um pouco mais sobre inspeção de software em um segundo artigo a ser publicado na segunda edição da Engenharia de Software Magazine.

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 aparentemente simples, esta atividade pode ser bastante dificultada pelo cliente ou mesmo por um processo de validação inadequado utilizado pela empresa.

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 faz com que as economias de curto prazo sejam logo suplantadas pelas despesas no longo prazo, verificadas com superação de custo e prazo nos projetos conduzidos.

Veremos a partir de agora algumas das atividades que devem ser consideradas durante a gerência dos requisitos.

Controle de Mudanças

Conforme foi citado anteriormente, os requisitos são voláteis e, portanto sofrem mudanças ao logo 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”.

Em cada fase do processo de desenvolvimento, um conjunto bem definido de itens de configuração deve ser estabelecido. A este conjunto é dado o nome de baselines. Estas baselines servem como marco no processo de desenvolvimento, pois ao final de cada fase é estabelecida uma nova baseline e, portanto, todos os itens de configuração estão sob total controle de seus desenvolvedores. Desta forma, nesta baseline é guardada “uma foto” dos artefatos criados até aquele momento:

- tornando mais fácil realizar modificações nos artefatos de maneira controlada;
- possibilitando ter um controle sistemático sobre todos os itens de configuração abordados em cada fase do ciclo de vida do software, e;
- tornando possível que sejam realizadas, mais facilmente, avaliações sobre seu grau de evolução.

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.

Os rastros dos requisitos podem ser de dois tipos:

- Pré-rastreabilidade: os rastros (artefatos: plano de negocio, atas de reunião com o usuário) que fundamentaram a criação do requisito.
- Pós-rastreabilidade: os rastros (artefatos: código, documentos) que se formaram a partir do requisito criado.

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.

A gerência, neste cenário, é responsável por manter uma infra-estrutura necessária para atividades de verificação que tornem possível investigarmos a qualidade dos requisitos que estamos definindo.

Conclusão

A **engenharia de requisitos define, sem dúvida, um dos mais importantes conjuntos de atividades a serem realizadas em projetos de desenvolvimento de software**. Embora não garanta a qualidade dos produtos gerados, é um pré-requisitos básico para que obtenhamos sucesso no desenvolvimento do projeto. Este artigo é apenas uma breve introduç ão ao tema, que deverá ser bastante explorado em edições futuras da Engenharia de Software Magazine.

Links Úteis

- Wikipedia - Engenharia de software:
  Engenharia de software é uma área da computação voltada à especificação, desenvolvimento, manutenção e criação de software,
- Guia de profissoes - Engenharia de Software:
  O engenheiro com esta formação dedica-se ao desenvolvimento de softwares e programas computacionais.
- Base de dados de estados e cidades:
  Baixe agora uma base de dados de estados e cidades brasileiros no formato SQL para incorporar ao seu projeto.

Saiba mais sobre Engenharua de Software ;)

- Introdução a Engenharia de Requisitos:
  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.
- Artigo Engenharia de Software - Alguns Fundamentos da Engenharia de Software:
  Engenharia de Software é o mesmo que Ciência da Computação? Ou é uma entre muitas especialidades da Ciência da Computação? Ou dos Sistemas de Informação, ou do Processamento de Dados, ou da Informática, ou da Tecnologia da Informação?
- Breve estudo sobre engenharia de componentes:
  Este artigo descreve assuntos relacionados à componentização, que devem ser analisados antes de um estudo mais aprofundado neste universo da componentização de software. Os conceitos tratados serão a Engenharia de software, engenharia web e enge