

APLICAÇÃO WEB – DESENVOLVENDO COM QUALIDADE

Asciole Andrade Neto¹

Resumo

Este artigo descreve a importância da utilização de alguns processos que devem ser utilizados na engenharia de software para a obtenção de bons resultados no desenvolvimento de aplicações Web. Tais resultados se referem à eficiente coleta dos requisitos e administração destes durante a execução do projeto, a aplicação do processo unificado e a aplicação de testes de software. Para tanto, foi descrito neste trabalho a importância e os conceitos das principais atividades relacionadas à construção de sistemas Web. Neste Contexto, destaca-se a Análise e Produção de Requisitos, alguns processos específicos para o desenvolvimento de sistema de informações os quais fazem parte do processo de desenvolvimento unificado para web e as atividades de teste de software.

Palavras-chave: Análise de Requisitos. FDD(Wireframes). Planilha de Requisitos. Projeto Linear. Protótipo de Interfaces. Testes de Software.

Abstract

This article describes the importance of using some processes to be used in software engineering to achieve successful development of Web applications such results relate to the efficient collection and administration of these requirements during project execution, application the unified process and implementation of software testing. To that end, we describe here the importance of the key concepts and activities related to building web systems this context, we highlight the analysis and production requirements, some specific procedures for developing information systems which are part of unified development process for web and software testing activities.

Keywords: Requirements Analysis. FDD (Wireframes). Sheet requirements. Linear Design. Prototype Interfaces. Software Testing.

¹- Pós-graduando em Sistemas de Informação para Web pela Faculdade de Administração e Negócios de Sergipe - FANESE

INTRODUÇÃO

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, e ainda o mau uso dos fundamentos da engenharia de software para apoiar as atividades do desenvolvimento.

No entanto, neste cenário, diversas práticas de qualidade de software são necessárias para o desenvolvimento de aplicações que funcionem perfeitamente de acordo com os objetivos e o ambiente para o qual foi projetado. Sendo assim, neste artigo descreverei os conceitos e a importância dos principais processos para o desenvolvimento de aplicações Web como a análise e produção de requisitos, o processo unificado voltado ao desenvolvimento web e os testes de software.

No processo de concepção de uma aplicação ou até mesmo de um pequeno módulo desta, é indispensável e imprescindível o processo de levantamento de requisitos. Com este processo, os analistas irão entender quais as reais necessidades dos usuários e com isso poderão especificar uma solução que melhor irá atender ao negócio do cliente. Portanto, a não utilização da engenharia de requisitos no processo de desenvolvimento de um software pode resultar em um produto sem qualidade, mal-especificado, e o pior de tudo, que não atende as necessidades do cliente.

No mesmo sentido, o processo unificado pode ser utilizado para melhor organizar o desenvolvimento da aplicação. Esse processo é composto por várias ferramentas, entre elas, planilha de requisitos, projeto linear, web content, wireframe e protótipo de interface. Cada uma dessas ferramentas possui objetivos específicos a serem alcançados no desenvolvimento de aplicações web.

Um outro processo importante na construção de aplicações web, como também em outras arquiteturas, é o teste de software. Através deste processo podemos verificar se o produto atingiu suas especificações e funcionou corretamente no local para o qual foi construído.

O presente artigo tem como objetivos principais: (i) descrever a importância e a metodologia da análise e produção de requisitos; (ii) descrever os processos unificados voltado para o desenvolvimento de aplicações web; (iii)

descrever a importância do processo de teste de software para o desenvolvimento de aplicações web de qualidade.

O artigo está estruturado em outras quatro seções. A seção 2 descreve a importância da engenharia de requisitos e os processos que a compõe. A seção 3 descreve o processo unificado voltado ao desenvolvimento web. A seção 4 descreve a importância do teste de software no desenvolvimento de aplicações web com o objetivo da obtenção de excelência na qualidade. E finalmente na seção 5 a conclusão deste trabalho.

1. ANÁLISES E PRODUÇÃO DE REQUISITOS

Requisito é uma funcionalidade do sistema para que este possa alcançar os resultados para os quais foi desenvolvido. É um conjunto de necessidades que deverão ser atendidas para que o cliente venha a solucionar o problema de seu negócio. Porém, nem sempre o que o cliente anseia é o que realmente deve ser feito, contudo, cabe aos analistas identificar a real necessidade do negócio e assim, atingir os objetivos do cliente de forma coerente.

Segundo Spínola e Ávila (2008), um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos.

Existem três tipos de requisitos, segue a definição de cada um deles:

- **Requisitos Funcionais:** São requisitos ligados às funções operacionais do sistema e descrevem funções que o sistema deve executar. Como por exemplo: o software deve permitir o cadastro de funcionários, e deve permitir o cadastro de seus dependentes caso este o tenham;
- **Requisitos não Funcionais:** São os requisitos de usabilidade, disponibilidade e eficiência que o sistema deve atender ou qualidades específicas que o software deve ter. Os requisitos não funcionais descrevem as características internas de qualidade do software: o software deve garantir que o tempo de retorno das consultas não seja maior que 5 segundos;
- **Requisitos de Domínio:** São requisitos que refletem qualidade do sistema. Podemos dizer que são validações internas que qualificam o sistema no princípio da integridade de dados. Por Exemplo: Um dependente pode ser cadastrado desde que seu titular esteja em dias com suas mensalidades.

1.1 Produção de Requisitos

A produção de requisitos é uma das primeiras e, em minha opinião, a mais importante etapa no processo de desenvolvimento de software.

Segundo Spínola e Ávila (2008), uma especificação de requisitos bem elaborada é pré-requisito para um software de qualidade, embora não seja garantia disso. Desta forma, além de uma boa especificação dos requisitos a serem implementados, devemos também nos preocupar com as formas de medidas da qualidade do software.

São quatro as atividades base relacionadas com a produção de requisitos: levantamento, registro, verificação, validação, todas bastante eficientes conforme descritas nos tópicos a seguir.

1.1.1 Levantamento

Esta atividade consiste na obtenção dos requisitos, juntamente com os clientes e usuários finais do sistema, objetivando, dessa forma, o descobrimento do problema a ser resolvido, os serviços do sistema, restrições de hardware e de desempenho entre outras informações. Existem algumas técnicas que apoiam as atividades de levantamento de requisitos, dentre essas técnicas estão:

Entrevista: esta técnica resume-se em “conversas” realizadas com o usuário (entrevistado) para levantar os requisitos do sistema a ser desenvolvido;

Prototipação: é uma versão pobre do sistema a ser criado, porem permite aos utilizadores identificar os pontos fortes e fracos dando subsídios para a equipe de analistas elaborarem uma especificação de requisitos de melhor qualidade. A maior desvantagem desta atividade é o custo de desenvolvimentos dos protótipos.

1.1.2 Registro

Após a coleta dos requisitos, estes devem ser documentados para que possam servir de base para o restante do processo de desenvolvimento.

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 em um documento próprio facilita o controle de alterações de todos os envolvidos, bem como a geração de versões do documento.

O controle de versão dos documentos de requisitos deve ser feito por algum sistema de controle de versão, o qual facilita o acesso por todos os envolvidos e em caso de problemas temos um histórico guardado de todos os arquivos gerados.

1.1.3 Verificação

Este processo se resume em examinar a especificação do software, de forma a dar segurança de que todos os requisitos foram definidos sem ambigüidades, inconsistências ou omissões, detectando e corrigindo possíveis problemas ainda durante a fase de definição dos requisitos. Essa fase é muito importante, pois sabemos que o custo para correção de defeitos no sistema aumenta à medida em que o processo de desenvolvimento progride.

Segundo Spínola e Ávila (2008), 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.

1.1.4 Validação

Nesta fase, obtemos a confirmação dos clientes para os requisitos obtidos no processo de levantamento. Na visão da engenharia de requisitos, esta atividade significa aprovar junto ao cliente os requisitos que foram especificados.

Apesar desta atividade aparentemente ser simples, ela pode ser bastante dificultada pelo cliente ou mesmo pelo processo de validação utilizado. No entanto, após a fase da validação dos requisitos e tendo todos estes sido aprovados pelos stakeholders do projeto, podemos passar para a próxima fase do desenvolvimento do software. Pode-se retornar ou não para esta fase caso haja algum problema nas etapas posteriores que tenha que ser corrigido em algum requisito ora levantado.

2 - PROCESSOS UNIFICADOS UTILIZADOS NO DESENVOLVIMENTO DE APLICAÇÕES WEB

No processo de desenvolvimento de aplicações web, pode-se utilizar inúmeros tipos de artefatos. Porém, alguns desses processos são específicos para o desenvolvimento de aplicações para Web, são estes: Planilha de Requisitos, Projeto Linear, Web Content, FDD e Protótipo de Interfaces. A seguir, o detalhamento dos processos citados.

2.1 Planilha de Requisitos

Para o desenvolvimento de um sistema Web, é necessário um levantamento dos requisitos, conforme vimos no tópico anterior. Portanto, para que possamos organizar e enumerar todos os requisitos levantados, faz-se necessário a utilização de um artefato denominado planilha de requisitos que consiste em um documento que contém informações da situação e de controle, bem como a descrição e comentários sobre a implementação dos requisitos levantados.

Nesta planilha, podemos ter as seguintes informações a respeito dos requisitos, e estes devem ser atualizados conforme sua implementação:

- Código: Identifica unicamente um requisito a fim de que se possa controlá-lo através do projeto.
- Descrição: Nesta coluna descreve-se o requisito.
- Categoria: Indica qual é o tipo do requisito.
- Prioridade: Indica o nível de importância que o requisito possui para o sistema em geral, podendo ser baixa, média ou alta.
- Dificuldade: Indica o nível de dificuldade para implementar este requisito, podendo ser baixa, média ou alta.
- Atendido: Representa o status do requisito, indicando se o mesmo foi ou não implementado no sistema.
- Comentários: Fornece informações sobre o requisito, dizendo, por exemplo, o motivo que um determinado requisito ainda não foi implementado (indicando mais especificamente, quais são as dificuldades).

A planilha de requisitos é um artefato mutável, ou seja, que está apta a sofrer alterações no decorrer do desenvolvimento do projeto, dependendo do negócio onde o sistema irá atuar.

As alterações nos requisitos também são registradas na planilha de requisitos, tornando-se um meio indispensável para a consulta das mudanças feitas no decorrer do projeto.

A extração dos requisitos, conforme visto no tópico anterior deve ser feita à medida que os casos de uso, ou seja, Análise do Negócio do sistema são realizados e validados. Portanto, a montagem da planilha de requisitos aumenta à medida que os casos de uso são aprovados pelo cliente.

2.2 Projeto Linear

O Projeto Linear é um dos processos mais importantes para o desenvolvimento de um sistema Web, assim como a planilha de requisitos. Esse processo consiste no mapeamento dos requisitos listados na planilha de requisitos e nas áreas ou páginas de uma aplicação web. Cada área ou página será identificada com um código, que por sua vez será relacionado com nenhum, um ou mais códigos dos requisitos mapeados através da Planilha de Requisitos. É necessária a utilização do projeto linear em conjunto com a planilha de requisitos, pois este completa a atuação do outro.

Segundo Aquino (2008), através deste documento busca-se um maior controle do sistema, pois se houver quaisquer modificações nos requisitos, o fornecedor saberá quais áreas devem sofrer mudança.

Este é também um artefato mutável, ou seja, que está sujeito a sofrer alterações no decorrer do projeto. A Figura abaixo apresenta um exemplo de como seria uma simples representação de um Projeto Linear, mostrando algumas áreas do site com seus respectivos requisitos relacionados. Note que na coluna requisitos estão listados apenas os códigos dos respectivos requisitos que estão enumerados na planilha de requisitos.

Projeto Linear		
Código	Áreas do site	Requisitos
1	Área de pré cadastro	Req0001, Req0002, Req0007
1.1	Formulário	Req0031
1.1.1	Mensagem de erro	Req0011, Req0012, Req0013
1.1.2	Mensagem de sucesso	Req0051, Req0032
2	Home page personalizada	Req0011, Req0042, Req0047, Req0048, Req0057, Req0017, Req0027
2.1	Login	Req0011, Req0042, Req0047
2.1.1	Mensagem de erro	Req0023, Req0028, Req0029
3	Esqueci minha senha	Req0068
3.1	Formulário	Req0011, Req0055, Req0098, Req0048, Req0057, Req0017, Req0027
3.1.1	Mensagem de erro	Req0033, Req0039, Req0050
3.1.2	Mensagem de sucesso	Req0051, Req0032

Figura 1 – Projeto Linear – Fonte: Revista Engenharia de Software

2.3 Web Content

O Web Content é um artefato de software responsável pelo armazenamento de todo o conteúdo textual utilizado em uma aplicação web, guardando informações de metadados sobre as páginas da aplicação. Não existe um documento padrão de Web Content, informações como título da página, descrição, palavras-chave, conteúdo da página, figuras, gráficos e diagramas são informações importantes que podem compor um documento de Web Content. Normalmente cada empresa que desenvolve aplicações web possui o seu. O Web Content é formado de acordo com os requisitos do sistema. É muito importante lembrar que esse artefato é formado não só de uma, mas várias seções, onde cada uma indica o conteúdo de cada página do site. Com o Web Content, o fornecedor consegue agrupar e gerenciar melhor o conteúdo de um site.



The image shows a web form for content management. It has a light gray background and a red border. The fields are as follows:

- Código:** A small text input field.
- Título da página:** A medium-sized text input field.
- Meta título:** A medium-sized text input field.
- Meta descrição:** A medium-sized text input field.
- Meta palavras chave:** A medium-sized text input field.
- Conteúdo:** A large, empty rectangular box for the main content.
- Fotos, diagramas, gráficos:** A small text input field.

Figura 2: Web Content – Fonte: Revista Engenharia de Software

2.4 FDD (Wireframes)

Segundo Aquino (2008), FDD (Functional Design Document) é um conjunto de Wireframes onde cada um representa uma página da aplicação e um Wireframe é uma maquete da página Web que se dirige somente à disposição de elementos, não à estética.

O FDD é o esboço de como será a página quando ficar pronta desprezando qualquer tipo de design, cores e imagens. A vantagem de utilizar Wireframes para o desenvolvimento de páginas web é que ele mostra como ficará disposto o fluxo da informação estabelecido anteriormente no Projeto Linear. Ele pode ser desenvolvido pelo arquiteto de informação e é um trabalho fundamental porque auxilia a equipe de desenvolvimento na tarefa de produzir o site de maneira mais rápida, padronizada e eficiente. O uso de um FDD estabelece uma forte ligação da arquitetura da informação com a estrutura do site, colocando a informação no seu respectivo local.

Na elaboração de um wireframe, o arquiteto busca representar esqueleticamente os objetos que compõe a página web como textos, imagens, mecanismos de busca entre outros, com quadrados e círculos, traços contínuos e pontilhados, palavras em negrito e sublinhadas, largura e altura das páginas, padronizados para todos os wireframes.

Além disso, um FDD bem organizado pode oferecer fortes soluções para os problemas de usabilidade. Outra característica importante deste artefato é que ele pode informar onde encontrar o conteúdo para aquela respectiva página dentro do Web Content. O FDD possui um papel muito importante no desenvolvimento de páginas web, pois conduz a proposta de layout a ser construída pelo designer, mas em contrapartida possui a desvantagem de não apresentar uma solução gráfica para o projeto. Em relação ao desenvolvimento de um Web Site, o FDD torna-se um dos artefatos mais completos, que auxiliam muito os programadores, pois eles criam uma relação entre a página a ser implementada e o conteúdo a ser aplicado.

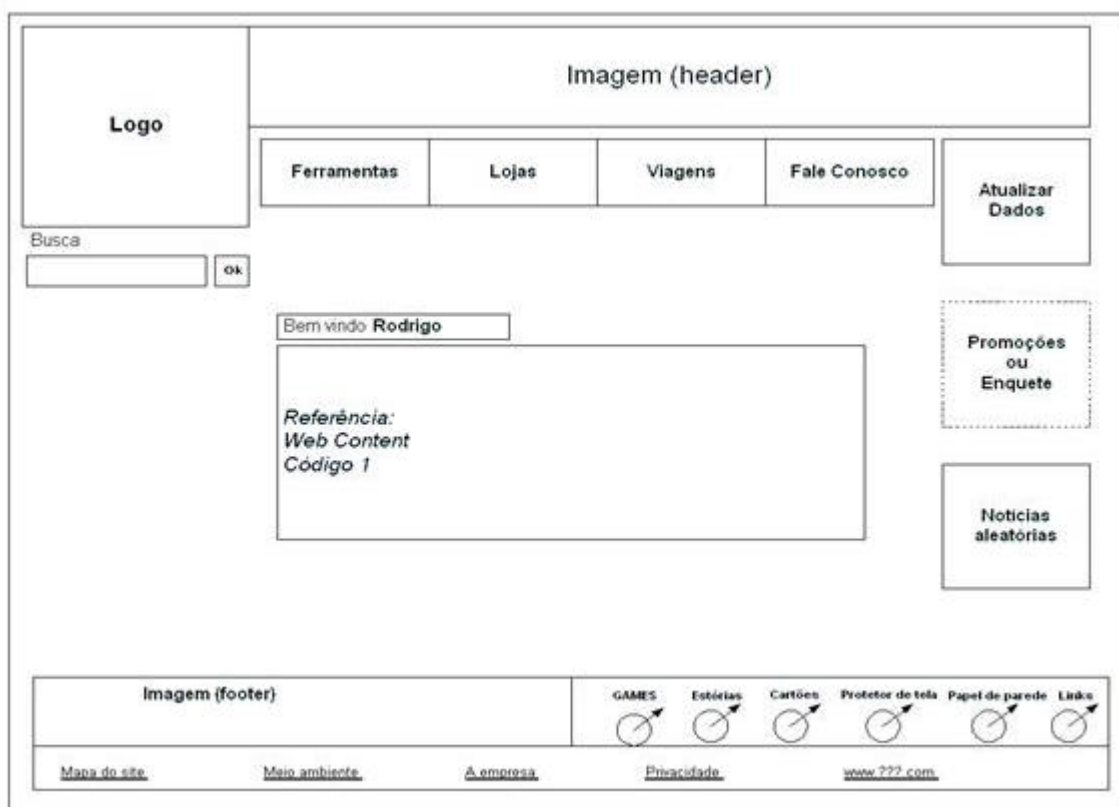


Figura 3 – FDD – Fonte: Revista Engenharia de Software

Segue abaixo alguns exemplos de softwares que auxiliam na criação de Wareframes.

- Visio;
- Freehand;
- QuarkXPress;
- InDesign;

2.5 Protótipo de Interfaces

O protótipo de interface é uma parte da aplicação idealizada e implementada pelo designer e que é submetida à aprovação pelo cliente, ou seja, é a representação gráfica, não necessariamente funcional, de um produto em concepção.

A utilização de prototipagem permite a consolidação de conhecimento a cerca do produto que deverá ser implementado, pois com uma “maquete” do sistema, a equipe poderá discutir as possíveis soluções para eventuais empecilhos que venha a surgir. Para chegar até o protótipo, o designer precisa utilizar o FDD ou pelo menos uma parte dele para ter noções de como será a divisão do site. A principal função deste artefato é fornecer ao cliente quais serão as cores básicas da aplicação, uma parte da arquitetura de informação e como ficarão disponibilizadas as informações os usuários dentro do site. Uma das vantagens na utilização deste artefato é direcionar totalmente a equipe de análise e projeto, bem como a equipe de implementação.

Quanto à classificação os protótipos podem ser:

- Baixa fidelidade: são representações mais rudimentares onde são investidos pouco tempo e poucos recursos sem exigir uma capacidade técnica elevada. São mais utilizados na fase de especificação de requisitos;
- Média fidelidade: essa é uma nova modalidade de protótipo que agrega características da baixa-fidelidade e da alta-fidelidade. Consiste na implementação computadorizada da aplicação com funcionalidade limitadas;

- Alta fidelidade: são protótipos que se aproximam o máximo do produto idealizado. São representações executáveis, possuindo código fonte e criadas com base na linguagem de programação escolhida para o projeto;

Exemplo de software que auxiliam na criação de protótipos de interface de média fidelidade:

- Visio;
- Axure;
- Absolute Layout(NetBeans);
- SILK;
- DENIM;
- DEMAIS;

3. TESTES DE SOFTWARE

O teste de software é um dos processos primordiais para a obtenção da qualidade do sistema desenvolvido, através dele podemos verificar se o produto atingiu suas especificações e funcionou corretamente no local para o qual foi projetado. O seu objetivo é identificar falhas, também chamadas de “*bugs*”, em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final.

Em linhas gerais, testar um sistema que acaba de ser desenvolvido, consiste em verificar de forma controlada se suas funcionalidades estão sendo executadas de acordo com o especificado. O objetivo principal dessa tarefa é mostrar o número máximo de falhas com o mínimo de esforço, ou seja, mostrar aos desenvolvedores se os resultados estão de acordo com os padrões estabelecidos.

Conhecer a diferença entre Defeito, Erros, e Falhas é de suma importância para entendermos os objetivos dos testes de software. O defeito é um ato inconsistente no desenvolvimento de um software que gerar resultado inesperado, por exemplo, uma instrução ou comando incorreto. Já o erro, é o resultado concreto de um defeito num artefato de software que pode ser a diferença entre o valor obtido e o valor esperado de uma operação, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa

constitui um erro. Entretanto a falha é o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.



Figura 4 – Defeito, Erro, Falha – Fonte: Revista Engenharia de Software

No processo de desenvolvimento de software, a maioria dos defeitos, senão todos são humanos. Apesar dos melhores métodos de desenvolvimento, os melhores profissionais e ferramentas excelentes, os defeitos permanecem, tornando a atividade de teste de software fundamental durante o projeto. Esta atividade corresponde ao último processo de avaliação do produto antes da sua entrega final

O tamanho do sistema a ser desenvolvido e a quantidade de pessoas trabalhando na equipe do projeto são dois fatores diretamente proporcionais à complexidade do teste de software, pois esses dois fatores têm como consequência o aumento da ocorrência de defeitos. Assim, a falha ocorre quando o sistema não se comporta de acordo com o esperado pelo usuário. Um bom exemplo disso é a utilização pelo funcionário da contabilidade de funcionalidade do sistema que somente a diretoria poderia ter acesso. Esse tipo de falha pode ocorrer por diversos motivos, dentre eles:

- Falha na especificação do requisito;
- A especificação pode conter requisitos impossíveis de serem implementados devido a limitações de hardware ou software;
- A base de dados pode está mal modelada, de forma que não seja permitido distinguir os tipos de usuário;

- Pode ser que haja um defeito no algoritmo de controle dos usuários.

Sempre devemos executar testes de software continuamente, pois segundo as leis de Murphy, “se uma coisa pode sair errada ela sairá”, “se tudo parece estar indo bem é porque não foi observado direito” e, por último, “a natureza sempre está a favor das falhas”.

O teste de software é uma das atividades mais custosas do processo de desenvolvimento de software, pois pode envolver uma quantidade significativa dos recursos de um projeto. O rigor e o custo associado a esta atividade dependem principalmente da complexidade da aplicação a ser desenvolvida. Diferentes categorias de aplicações requerem uma preocupação diferenciada com as atividades de teste.

4. CONSIDERAÇÕES FINAIS

Neste artigo foi possível ver os conceitos e a importância de alguns processos de desenvolvimento de aplicações Web. Apesar da importância da utilização dos processos apresentados, muitas empresas ainda não fazem uso desses recursos. Entretanto, esse cenário começa a mudar dada a crescente necessidade de integração de sistemas. Neste contexto, 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-requisito básico para que obtenhamos sucesso no desenvolvimento do projeto. No mesmo sentido, os testes de software são muito importantes para o alcance dos objetivos do negócio e satisfação do cliente.

REFERÊNCIAS:

AQUINO, Rodrigo S Prudente de. O processo Unificado Integrado ao Desenvolvimento Web. **Revista Engenharia de Software**. 1. ed. Pág 29 a 37, 2008.

ÁVILA, Ana Luiza. SPÍNOLA, Rodrigo Oliveira. Introdução a Engenharia de Requisitos. **Revista Engenharia de Software**. 1. ed. Pág 46 a 52, 2008.

BARTIE, Alexandre. **Processo de Teste de Software**. Disponível em:

<http://imasters.uol.com.br/artigo/6102/des_de_software/processo_de_teste_de_software_-_parte_01>. Acesso em 28 jun. 2010, 15:22:09

NETO, Arilo Cláudio Dias. Introdução a Teste de Software. **Revista Engenharia de Software**. 1. ed. Pág 54 a 59, 2008.

MOREIRA, Fábio Lúcio. Estimando seus Requisitos. **Revista. NET Magazine**. 76. ed. Pág 52 a 58, 2010.

VIEIRA, Guilherme Alves. Corrêa, Fernando Antônio Diniz. Uma visão prática e realista do desenvolvimento de sistemas orientado a objetos. **Revista. NET Magazine**. 62. ed. Pág 57 a 64, 2009.

_____. Uma visão prática e realista do desenvolvimento de sistemas orientado a objetos. **Revista. NET Magazine**. 61. ed. Pág 63 a 70, 2009.