



UMA ABORDAGEM BASEADA NOS PRINCÍPIOS ÁGEIS PARA O DESENVOLVIMENTO DE SISTEMAS WEB

RATUCHENEI, Victor Avila¹; DINIZ, Eliane da Silva Alcoforado¹, LOPES, João Ladislau Barbara²

¹ Dept^o de Ciência da Computação – IFM/UFPeI

² Centro de Informática - UFPeI

Campus Universitário – Caixa Postal 354 – CEP 96010-900.
vratuchenei.ifm@ufpel.edu.br

1. INTRODUÇÃO

Atualmente, a internet direciona uma grande parte da produção de software do mercado atraído pelas especificidades que esta plataforma propicia, incentivando também a migração de muitas outras aplicações para este ambiente, que buscam tirar proveito de características como: baixo custo de utilização, livre oferta de produtos e serviços, controle de processos sobre produtos e serviços, geograficamente distribuídos, entre outras tantas vantagens (DINIZ, 2000 apud MATTAR, 2007), elevando-se a um estágio de integração horizontal e vertical como ferramenta de negócio, ou seja, a automatização dos processos internos da organização, incluindo a utilização dos recursos relevantes, que se encontram dentro ou fora das fronteiras da organização, visando à criação de valor para seus clientes/consumidores (DIAS, 2002).

Decorrente da demanda, os sites Web cresceram de uma porção de *scripts* para milhares de linhas de código. Os projetos desse tamanho requerem planejamento e gerenciamento similares aos realizados nos ambientes tradicionais de desenvolvimento de software (WELLING; THOMSON, 2003). Porém, o imediatismo é uma das razões pelas quais as metodologias usadas na Engenharia de Software tradicional tornam-se inadequadas ao processo de desenvolvimento para a Web, uma vez que, nesse ambiente os prazos de planejamento são muito curtos, fazendo com que as equipes de desenvolvimento sofram pressões constantes para ter o software em funcionamento o mais rápido possível. Outra razão que torna os processos tradicionais de software impróprios aos projetos de aplicativos Web é a evolução contínua do ambiente, isto significa que, a exemplo das estratégias de negócio em constante mudança, o apoio que o projeto Web produzirá como ferramenta de negócio deve acompanhar as estratégias, que em sua

maioria são altamente concentradas, impulsionadas pelo conteúdo e em contínua evolução (PRESSMAN, 2005), mostrando-se inconveniente, por exemplo, o congelamento dos requisitos durante a concepção do projeto (PENTEADO, 2006).

Analisando a Web como um ambiente de negócios dinâmico e mutável é difícil ou impossível prever como um sistema baseado nesta plataforma evoluirá com o passar do tempo. Neste contexto, mostram-se viáveis metodologias que seguem os princípios ágeis de desenvolvimento. Dentre as várias metodologias e técnicas que foram desenvolvidas observando esses princípios pode-se destacar: *Extreme Program*, *Scrum* e *Test Driven Development*. Na seção 2 serão apresentadas as principais características das metodologias escolhidas. Na seção 3 será apresentada a metodologia de desenvolvimento desse trabalho. Na seção 4 apresenta a Abordagem Proposta. Na seção 5 será apresentada a Conclusão. Finalizando, serão apresentadas, na seção 6 as referências utilizadas na elaboração desse texto.

2. METODOLOGIAS ÁGEIS

- *Extreme Program* - XP é metodologia ágil para equipes pequenas e médias e que irão desenvolver software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software. Os quatro valores fundamentais da metodologia XP são: comunicação, simplicidade, feedback e coragem. A partir desses valores, possui como princípios básicos: feedback rápido, presumir simplicidade, mudanças incrementais, abraçar mudanças e trabalho de qualidade (WIKIPEDIA). Dentre as variáveis de controle em projetos (custo, tempo, qualidade e escopo), há um foco explícito em escopo. Para isso, recomenda-se a priorização de funcionalidades que representem maior valor possível para o negócio. Desta forma, caso seja necessário a diminuição de escopo, as funcionalidades menos valiosas serão adiadas ou canceladas. A XP incentiva o controle da qualidade como variável do projeto, pois o pequeno ganho de curto prazo na produtividade, ao diminuir qualidade, não é compensado por perdas (ou até impedimentos) a médio e longo prazo (TELLES, 2005).

- *Scrum* é uma metodologia ágil para gestão e planejamento de projetos de software. No *Scrum*, os projetos são divididos em ciclos (tipicamente mensais) chamados de *Sprints* (SCRUM; 1998). O *Sprint* representa um *Time Box* dentro do qual um conjunto de atividades deve ser executado. As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como *Product Backlog*. No início de cada *Sprint*, faz-se um *Sprint Planning Meeting*, ou seja, uma reunião de planejamento na qual o *Product Owner* prioriza os itens do *Product Backlog* e a equipe seleciona as atividades que ela será capaz de implementar durante o *Sprint* que se inicia. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog* (IMPROVEIT; 2008). A cada dia de uma *Sprint*, a equipe faz uma breve reunião, chamada *Daily Scrum*. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia. Ao final de um *Sprint*, a equipe apresenta as funcionalidades implementadas em uma *Sprint Review Meeting*. Finalmente, faz-se uma *Sprint Retrospective* e a equipe parte para o planejamento do próximo *Sprint*, assim reinicia-se o ciclo (WIKIPEDIA; 2008).

- *Test Driven Development*-TDD é uma técnica que consiste em iterações curtas do processo de software, onde os novos casos de teste abrangendo a melhoria desejada ou novas funcionalidades são escritos em primeiro lugar, então, a

produção código necessário para passar os testes é implementada e, finalmente, o software é refatorado acomodando as mudanças (WIKIPEDIA). A disponibilidade de testes antes de reais implementações garante ao desenvolvimento um rápido retorno depois de qualquer mudança. TDD é um método de concepção de software, e não simplesmente um método de ensaio de práticas sobre processos de software. Está relacionado com as práticas do Extreme Programming. Além de antecipar a identificação e correção de falhas durante o desenvolvimento outra vantagem é a automatização da documentação de código, pois testes de unidade são feitos para cada trecho de código implementado (IMPROVETI).

3. METODOLOGIA

Inicialmente, foram estudadas a filosofia e as diretrizes de desenvolvimento dos Princípios Ágeis. Esses Princípios são preceitos os quais se fundamentam um conjunto de processos de desenvolvimento conhecido como metodologias ágeis.

Esses Princípios ditam que valorizar mais indivíduos e suas interações, software em funcionamento, colaboração com o cliente e resposta às mudanças em lugar de processos e ferramentas, documentação abrangente, negociação de contratos e planejamento do trabalho tornam-se mais dinâmicas as etapas do processo de desenvolvimento de todas as áreas envolvidas no projeto, mantendo uma melhor qualidade e eficiência no seu desenvolvimento para atingir uma resposta rápida à mudança e maior controle sobre os prazos e orçamentos do projeto.

As Metodologias Ágeis, em sua maioria tentam minimizar o risco pelo desenvolvimento do software em curtos períodos, chamados de iteração, os quais duram até quatro semanas. Cada iteração, composta pelas etapas de análise de requisitos, projeto, codificação, teste e documentação, geram um protótipo do projeto de software em sua íntegra, possibilitando, ao fim de cada iteração, a implantação de uma nova versão do software capacitando a reavaliação das prioridades do projeto. Para a elaboração da abordagem metodológica voltada ao processo de desenvolvimento Web, objetivo desse trabalho, foram escolhidas as Metodologias Ágeis: *-XP*, *Scrum* e *-TDD* apresentadas na seção anterior. Essa escolha decorreu da observação das características relevantes das mesmas.

4. ABORDAGEM PROPOSTA

A abordagem proposta combina as metodologias ágeis apoiando-se nas práticas de gestão e organização do Scrum juntamente com as práticas reais de programação do Extreme Program (Kiniberg, 2007). Mesmo apresentando práticas e valores muito semelhantes, juntas estas metodologias complementam-se, abordando diferentes áreas. TDD é acrescentado às práticas da XP automatizando os testes de unidade, racionalizando a escrita de código e também se mostrando como um método formal de documentação e casos de uso. Aliado ao Scrum o TDD auxilia na estimativa de prazos e custos do projeto. A afinidade com que as práticas e valores que estas metodologias demonstram quando combinadas contemplam as especificidades da plataforma Web. A Fig.01 apresenta um esquemático da abordagem proposta.



Fig.01 – Planejamento, teste e desenvolvimento.

5. CONCLUSÃO

Para este trabalho foi realizado um estudo de caso para o desenvolvimento de uma aplicação para Web onde, utilizando-se a abordagem metodológica proposta, foram implementados a estrutura do sistema, seu projeto de interface e seu modelo de navegação. O resultado obtido a partir da abordagem proposta, foi considerado satisfatória para os objetivos inicialmente estabelecidos.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- MATTAR, Alexandre. **Critérios da avaliação da qualidade da informação em sistemas de internet banking**. Dissertação apresentada ao Departamento de Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo como requisito para a obtenção do título de Mestre em Administração. São Paulo, 2007.
- DIAS, Isabel de Meiroz. **Metodologia para a formulação de estratégias de presença na internet: um estudo de caso**. Dissertação apresentada ao Departamento de Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo como requisito para a obtenção do título de Mestre em Administração. São Paulo, 2002.
- WELLING, Luke; THOMSON, Laura. *PHP and MySQL Web Development*, 2 ed. – Indiana: Sams Publishing, 2003.
- TELLES, Vinícius Manhães. **Um estudo de caso da adoção das práticas e valores do extreme programming**. Dissertação apresentada ao corpo docente da Coordenação do Instituto de Matemática e Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção do grau de Mestre em Informática. Rio de Janeiro, 2005.
- PRESSMAN, Roger S. *Software engineering: a practitioner's approach*, 5th ed. McGraw-Hill series in Computer Science, 2005, 29, p. 769-798.
- PENTEADO, Rosângela Aparecida Dellosa. **Agile: Métodos Ágeis**. Departamento de Computação da Universidade Federal de São Carlos, 2006.
- AGILE MANIFESTO SIGNATORIES. Disponível em: <<http://agilemanifesto.org/sign/display.cgi>>. Acesso em: 08 de jun. 2008.
- IMPROVEIT. *Extreme Programming: Metodologia De Desenvolvimento Ágil*. Disponível em: <<http://www.improveit.com.br/xp>>. Acesso em: 21 mai. 2008.

WIKIPÉDIA. Programação Extrema. Disponível em:

<http://pt.wikipedia.org/wiki/Programa_extrema>. Acesso em 08 ago. 2008.

IMPROVEIT. SCRUM: Metodologia Ágil para gestão e planejamento de projetos.

Disponível em: <<http://www.improveit.com.br/scrum>>. Acesso em: 21 de mai. 2008.

WIKIPÉDIA. Scrum. Disponível em: <<http://pt.wikipedia.org/wiki/Scrum>>. Acesso em: 08 ago. 2008.

IMPROVEIT. Desenvolvimento Orientado a Testes. Disponível em:

<<http://www.improveit.com.br/xp/praticas/tdd>>. Acesso em: 08 ago. 2008.

WIKIPÉDIA. Test-Driven Development. Disponível em:

<http://en.wikipedia.org/wiki/Test_driven_development>. Acesso em 08 ago 2008.