



## ENGENHARIA WEB: METODOLOGIAS DE ENGENHARIA DE SOFTWARE ADEQUADAS AO DESENVOLVIMENTO DE WEBAPPS

Thiago Aparecido Aoki<sup>1</sup>, Aline Maria Malachini Miotto<sup>2</sup>

**RESUMO:** Este trabalho tem como objetivo a realização de uma análise sobre as principais diferenças entre as abordagens tradicionais de engenharia de software e as abordagens focadas no suporte ao desenvolvimento de aplicações web. Um estudo detalhado destas abordagens de desenvolvimento foi realizado e uma análise comparativa, baseada em nos seguintes critérios: suporte ao desenvolvimento de requisitos que evoluem com o tempo; suporte, durante o desenvolvimento, a mudanças; suporte a tempo de desenvolvimento curto; suporte a atividades de engenharia de software como comunicação, planejamento, construção e implantação; foi desenvolvida. Observou-se com o desenvolvimento deste trabalho que as metodologias que melhor suportam esses critérios, e assim melhor se adéquam ao desenvolvimento de aplicações WEB são XP e SCRUM.

**PALAVRAS-CHAVE:** Engenharia de Software, engenharia Web, metodologias ágeis.

### 1 INTRODUÇÃO

Em anos anteriores (aproximadamente na década de 50) um dos maiores problemas na criação de softwares era a falta de qualidade, pois o seu desenvolvimento acontecia de forma caótica, ou seja, sem nenhuma forma de estruturação e planejamento. Com o passar do tempo notou-se que quanto maior o sistema, mais complicada sua implementação e manutenção. Nesse contexto, vários problemas no desenvolvimento de software surgiram, tais como: atraso na entrega do produto final, insatisfação do cliente, falta de funcionalidades do produto final, dificuldade na manutenção do produto, entre outros problemas que aumentavam consideravelmente o custo do software.

Esta fase da história (por volta da década de 60) ficou conhecida como “*The Software Crisis* (Crise do Software)”. Para conter esta crise em 1968 na NATO - *Conference on Software Engineering* (Conferência sobre Engenharia de Software da OTAN) foram discutidas soluções para os problemas observados, e então surgiu o termo Engenharia de Software<sup>3</sup>, proposto por Fritz Bauer

Pressman (2006) descreve que hoje com o desenvolvimento tecnológico e com a Internet é possível criar desde sites simples até um sistema completo de e-commerce. Porém, atualmente o desenvolvimento para web acontece de forma desenfreada, sem

<sup>1</sup> Discente do curso de Sistemas de Informação do Centro Universitário de Maringá, [lincolnfpsantos@hotmail.com](mailto:lincolnfpsantos@hotmail.com).

<sup>2</sup> Docente dos cursos de Análise e Desenvolvimento de Sistemas, Sistemas de Informação e Sistemas para Internet do Centro Universitário de Maringá, [amiotto@cesumar.br](mailto:amiotto@cesumar.br).

<sup>3</sup> Segundo Pressman (1995) Engenharia de Software é um rebento da engenharia de sistemas e de hardware. Ela abrange um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos – que possibilita ao gerente o controle do processo de desenvolvimento do software e oferece ao profissional uma base para a construção de software de alta qualidade produtivamente.

nenhuma estruturação ou planejamento. Para evitar que o desenvolvimento na web se torne caótico, é necessário que exista uma estrutura e planejamento, que proporcione sustentabilidade ao desenvolvimento de aplicativos web.

Segundo Pressman (2006) a engenharia web aplica “princípios científicos sólidos, de engenharia e de gestão, e abordagens disciplinadas e sistemáticas para o bem-sucedido desenvolvimento, implantação e manutenção de sistemas e aplicações de alta qualidade baseados na Web”.

De acordo com Sommerville (2007), com a implementação de metodologias de engenharia focadas nas características das aplicações web é possível obter uma melhor estrutura no desenvolvimento de WebApps, e assim permitindo a redução de custo de manutenção destas.

Este trabalho está organizado da seguinte forma: na seção 02 são apresentados os critérios utilizados para a realização da análise comparativa desenvolvida neste trabalho. Na seção 03 são apresentados os principais resultados obtidos da análise realizada. Finalmente na seção 04 são apresentadas as conclusões desta pesquisa.

## 2 MATERIAL E MÉTODOS

### 2.1. Critérios definidos para análise comparativa

Após realizar uma análise comparativa entre as metodologias de engenharia de software tradicionais e engenharia de software atuais, com foco principalmente no desenvolvimento ágil (BASSI FILHO, 2008), notou-se que as metodologias da engenharia de software ágeis têm: maior flexibilidade com relação a mudanças que ocorrem durante o projeto, o tempo do projeto é mais curto e dividido em partes (No Scrum chamado de Sprints (PAULA, 2010)) e o principal foco é o recurso humano e não o processo. Já as metodologias de software tradicionais têm: maior dificuldade em suportar mudanças no decorrer do projeto, ressaltando que essas mudanças ocorrem quando o projeto já foi iniciado e papéis de indivíduos muito bem definidos.

Comparando as metodologias estudadas com os critérios para construção de WebApps, sugeridos por Pressman (2009), listados na tabela 1, pode-se observar que as metodologias ágeis de engenharia de software adaptam-se melhor, pois possuem maior ênfase em ciclos de vida curtos e confortam mudanças no software de forma natural. Já nas metodologias da engenharia de software tradicionais tem maior dificuldade em enquadrar-se, pois o custo de mudanças no software é relativamente mais alto comparado às metodologias ágeis.

**Tabela 1** - Critérios para criação de WebApps

<b>Critério</b>	<b>Critérios relevantes para construir WebApps</b>
1	<b>Requisitos evoluem com o tempo:</b> Quando você inicia um projeto de WebApp, pode haver incerteza sobre alguns elementos da estratégias de negócios, do conteúdo e da funcionalidade a ser entregue, questões de interoperabilidade e muitas outras facetas do problema
2	<b>As mudanças ocorrerão com frequência:</b> Como a incerteza é uma parte inerente da maioria dos projetos da WebApp, as mudanças nos requisitos são comuns. Além disso, o feedback do usuário (baseado em uma avaliação dos incrementos entregues) e alterações nas condições de negócios podem ocasionar mudanças.
3	<b>Linhas de tempo são curtas:</b> Isso alivia a criação de documentação de engenharia volumosa, mas não impede a realidade simples de que a análise do problema, o projeto e o teste devam ser documentados de alguma maneira.
4	<b>Suporte a atividades de Engenharia Web:</b> comunicação, planejamento, construção e implantação.

Fonte: Elaborado pelo autor do trabalho

### 3 RESULTADOS ALCANÇADOS

Com base nos critérios apresentados na tabela 1, foi realizada uma análise de um conjunto de metodologias estudadas neste trabalho, buscando identificar quais metodologias melhor se adaptam a esses critérios. O resultado desta análise é apresentado na tabela 2.

**Tabela 2 - Análise comparativa**

<b>Modelo Cascata</b>			
<b>Critério 1</b>	<b>Critério 2</b>	<b>Critério 3</b>	<b>Critério 4</b>
Uma vez que os requisitos são definidos no início do projeto, eles não podem ser alterados.	Não trata as mudanças que ocorrem no decorrer do projeto.	Melhor implementado em projetos de grande porte.	Atividades: Engenharia de sistemas; Análise de requisitos; Projeto, codificação; Teste e Manutenção.
<b>Modelo XP (Xtreme Programming)</b>			
<b>Critério 1</b>	<b>Critério 2</b>	<b>Critério 3</b>	<b>Critério 4</b>
Os requisitos são recolhidos de forma incremental, de acordo com a necessidade do cliente.	Para cada interação que ocorre, adaptações aos requisitos já implantados são permitidos, entrando como atividade da próxima interação.	Cada interação tem o padrão de 2 semanas.	Atividades: comunicação, planejamento, construção e implantação.
<b>Modelo Scrum</b>			
<b>Critério 1</b>	<b>Critério 2</b>	<b>Critério 3</b>	<b>Critério 4</b>
Assim como o XP os requisitos são recolhidos de forma incremental.	Cada alteração que é detectada nos requisitos é colocada na próxima interação.	Interação padrão de 2 semanas	Atividades: comunicação planejamento, construção e implantação.
<b>Modelo Incremental</b>			
<b>Critério 1</b>	<b>Critério 2</b>	<b>Critério 3</b>	<b>Critério 4</b>
Os requisitos são recolhidos de forma incremental.	Antes do incremento os requisitos podem ser alterados, porém após a implementação não é possível.	As interações são realizadas em tempos curtos (Cada interação no pode ter mais que 20 mil linhas de código), porém o tempo de projeto pode variar.	Atividades: Definir requisitos iniciais, atribuir requisitos aos incrementos, Projetar arquitetura de sistema, desenvolver incremento de sistema, validar incremento, integrar incremento, validar sistema.
<b>Modelo Prototipação</b>			
<b>Critério 1</b>	<b>Critério 2</b>	<b>Critério 3</b>	<b>Critério 4</b>
É criado um protótipo para evitar as incertezas do cliente.	Para cada interação é necessário a aprovação do cliente.	Não apresenta especificações tempo para projetos.	Atividades: coleta e refinamento dos requisitos, projeto rápido, construção do protótipo, avaliação do protótipo pelo cliente, Refinamento do protótipo, engenharia do produto.

## 4 CONCLUSÃO

Neste trabalho foram estudadas diferentes metodologias de engenharia de software tradicionais, tais como: Cascata, Prototipação, Incremental, e também foram discutidas algumas das principais metodologias ágeis, tais como: XP e SCRUM.

Observou-se que as metodologias tradicionais têm maior foco no processo de documentação do software, pois despendem de maior tempo no planejamento. Além disso, as metodologias tradicionais são mais restritas às alterações no decorrer do projeto de software, pois se acredita que gastando esforços excessivos no planejamento essas alterações serão mitigadas ou até mesmo extintas.

Nas metodologias ágeis o principal foco é a entrega do software de forma incremental, onde para cada incremento é realizado um conjunto de metodologias estruturadas. Assim vimos que as metodologias ágeis, devem considerar um conjunto de critérios, sendo eles: requisitos evoluem com o tempo, as mudanças ocorrerão com frequência, linhas de tempo são curtas e um conjunto de atividades como comunicação, planejamento, construção e implantação. Visualizamos também que as atividades de documentação (diagrama sequencia, diagrama de interação, entre outros) das metodologias ágeis são executadas somente quando é necessário obter um melhor entendimento dos requisitos.

Com base na análise realizada neste trabalho, observamos que as metodologias como Xp e Scrum atendem melhor ao critérios definidos para a criação de WebApp de qualidade. Já as metodologias tradicionais como Cascata, Prototipação e Incremental não suportam de forma adequada os critérios definidos para criação de WebApps.

## REFERENCIAS

BASSI FILHO, Dairton Luiz. **Experiências com desenvolvimento ágil**, 2008. 170 p. Dissertação (Mestre em Ciência) - Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo.

PAULA, Renata de Souza Alves. **Implantação do Scrum para melhoria no gerenciamento de projetos de software**. Engenharia de Software Magazine, p. 8-17, 2010.

PRESSMAN, Roger S. **Engenharia de software**. São Paulo: Pearson Makron Books, 1995. 1056 p.

PRESSMAN, Roger S. **Engenharia de software**. 6. Ed. São Paulo: McGraw-Hill, 2006. 720 p.

SOMMERVILLE, Ivan. **Engenharia de software**. 8. Ed. São Paulo : Pearson Addison-Wesley, 2007.