



Método ágil na construção de interfaces web baseadas em padrões

Giovane Montine Moreira Gurgel¹, Laís Stephane Noronha Tôrres Moura², Juliana Amorim Dias de Oliveira³

¹Professor do curso técnico em informática do IFRN – Câmpus Mossoró. Bolsista pesquisador. e-mail: giovane.gurgel@ifrn.edu.br

²Aluna do curso técnico em informática do IFRN – Câmpus Mossoró. Bolsista de iniciação científica. e-mail: laisastephane@hotmail.com

³Aluna do curso técnico em informática do IFRN – Câmpus Mossoró. Bolsista de iniciação científica. e-mail: julianadias100@hotmail.com

Resumo: Este trabalho busca de um processo de software que trate das diferentes questões envolvidas na etapa de criação de interfaces para aplicações web. A falta de atenção com o uso dos padrões web é algo comum a muitas equipes de desenvolvimento. De modo similar, equipes também buscam por metodologias de desenvolvimento que não sejam muito burocráticas. No entanto, as metodologias tradicionais também possuem muitos pontos positivos. A metodologia proposta nesse trabalho tenta unir as duas visões (ágil e formal) a fim de obter um processo que seja mais interativo com o cliente, mais eficiente e eficaz seja atendendo aos requisitos do projeto quanto ao gerar interfaces baseadas em padrões. O processo possui quatro fases (concepção, elaboração, construção e transição) e envolve várias equipes (Gerente de Projeto, Designer Gráfico, Arquiteto de Informações, Equipe de User Experience (UX), Programadores, Equipe de Teste) que desenvolvem suas atividades tendo em vista a metodologia ágil Scrum. Ao combinar o ágil com o formal, o processo ficou mais dinâmico ao mesmo tempo em que se torna mais controlado. No entanto, é preciso observar como equipes se comportam usando a metodologia. Trabalhos futuros podem desenvolver um software que acompanhe a aplicação desse método ou adaptá-lo a outras metodologias ágeis.

Palavras-chave: interfaces web, padrões web, métodos ágeis

1. INTRODUÇÃO

Observando os avanços no processo de desenvolvimento de software, uma das medidas tomadas para aprimorá-lo seria cada vez mais burocratizá-lo, com muitos documentos e muitas regras. As metodologias ágeis oferecem uma alternativa, garantindo o controle necessário ao mesmo tempo em que são mais “rápidas” de serem executadas. Devido ao maior contato com o cliente, consegue-se ainda reduzir o risco de desconformidade de produto e insatisfação do cliente.

É possível destacar que as equipes de desenvolvimento de software cada vez mais trabalham de forma ágil. Isso devido os projetos estarem mais dinâmicos, com requisitos passíveis de alterações e datas de entrega do software curtas. O foco das metodologias ágeis baseia-se em pessoas e não em processos e planejamento. Essas metodologias têm sido mais adotadas por organizações pequenas e médias que precisam de um processo adequado a sua realidade e dentro dos seus recursos.

É nesse contexto que esse trabalho está inserido, em busca de um processo de software que trate das diferentes questões envolvidas na etapa de criação de interfaces para aplicações web. Apresenta uma nova técnica para realizar algo (WAZLAWICK, 2011). Atualmente já existem iniciativas de tornar o desenvolvimento das interfaces mais estruturado, como a formatação das interfaces através de regras Cascading Style Sheets (CSS) e dos trabalhos realizados pelo World Wide Web Consortium (W3C) sobre acessibilidade na web. No entanto, se não houver uma metodologia de desenvolvimento que aborde todos esses temas, as interfaces acabam sem o devido tratamento necessário (GURGEL; PINTO; ARAUJO, 2011). Por exemplo, muitas interfaces precisam depois de finalizado seu desenvolvimento, resolver problemas relacionados à acessibilidade e até mesmo de usabilidade.

A seção dois apresenta a metodologia Scrum que será base para a proposta. A seção três descreve os padrões web e ressalta sua importância atualmente. A seção quatro apresenta os resultados da pesquisa, a metodologia envolvendo aspectos ágeis e tradicionais no desenvolvimento de software. As conclusões são apresentadas na seção cinco onde também são relacionados trabalhos futuros.

2. SCRUM

Assim como outras metodologias ágeis, o processo de desenvolvimento de software baseado em Scrum possui características descritas no manifesto ágil (BECK et al., 2001), tais como interações entre indivíduos, colaborações com o cliente, valorizando o funcionamento do software e se esse atende às necessidades do cliente. Schwaber (2004) não considera Scrum como um processo onde todos os passos são bem definidos. O Scrum proporciona um conjunto de práticas que permitem acompanhar – e não prever – tudo que está acontecendo. Tem foco na gerência de projetos. Nesse sentido, Aguanno (2005) constata o uso do Scrum em outros tipos de projetos.

O Scrum é baseado em três princípios: a transparência garante a visibilidade de informações necessárias a quem está gerenciando os resultados, o segundo é o ato de inspecionar o processo com frequência necessária para identificar variações inaceitáveis. O processo pode ser modificado a cada nova inspeção. O terceiro princípio é adaptação. Detectado o problema no processo, o ajuste deve ser feito o mais rápido possível para minimizar desvios posteriores (SCHWABER; SUTHERLAND, 2010).

A Figura 1 mostra a estrutura básica do Scrum. Ao fim de cada iteração, o produto ganha novas funcionalidades. As adaptações são feitas ao longo da inspeção diária, cada componente do time de desenvolvimento podem inspecionar o outro e ajudar a fazer a adaptação necessária. O framework Scrum consiste em times que possuem vários papéis, as iterações são eventos com duração fixa – time-boxes, artefatos também são gerados de acordo com algumas regras.



Figura 1 – Estrutura básica do Scrum.

Fonte: Schwaber (2004), adaptado.

O Scrum é associado a projetos com poucas pessoas, uma vez que são poucos papéis na equipe e as fases são rápidas e dinâmicas. As equipes Scrum possuem três papéis (SCHWABER; SUTHERLAND, 2010):

- Mestre do Scrum (ScrumMaster): garante que o processo seja compreendido e seguido. Auxilia os clientes na definição de quem seria o Product Owner. Na verdade, o ScrumMaster ensina o Product Owner como trabalhar;
- Dono do produto (Product Owner): é uma pessoa e não um comitê. Mantém os requisitos do produto (Backlog), garantindo o valor do trabalho realizado pela equipe. Assim, todos sabem o que tem maior prioridade do que se deve trabalhar;
- Time: executa o trabalho propriamente dito. O ideal é um time com sete pessoas, mais ou menos duas pessoas. Quanto menos membros na equipe, maior interação. Cada time é auto-organizável, nem mesmo o ScrumMaster diz ao time como transformar o Backlog do produto em incrementos de funcionalidades. A composição de cada time pode ser mudada, mas a produtividade pode ser reduzida.



3. PADRÕES WEB

Os padrões do W3C, através de suas recomendações, abrangem todo o processo de construção de uma página, desde a sua programação até a sua adaptação para as mais diversas mídias e público. O uso dessas recomendações se torna fundamental, pois permite que uma página esteja inserida em um contexto de padronização internacional, garantindo a qualidade técnica além de possibilitar um maior consenso e acordo entre os mais diversos desenvolvedores.

A elaboração da interface é uma parte crucial do desenvolvimento de qualquer projeto, pois é ela que estabelecerá a conexão entre o usuário e a ferramenta criada, seja de forma física, perceptiva ou conceitual (BENYON, 2011). O mesmo acontece com o desenvolvimento de interfaces voltadas para a Web, onde é importante considerar diversas variáveis como usabilidade, acessibilidade, velocidade de acesso, etc.

Desse modo, construir uma interface com boa usabilidade, acessibilidade, entre outros fatores, é um grande desafio para a equipe de desenvolvedores. Ao atender esses pontos, a interface estará ao alcance de um grande número de pessoas, vencendo as barreiras ao acesso (BENYON, 2011). Esses fatores são considerados determinantes para a qualidade de uma aplicação web.

Do design gráfico até o teste, é importante considerar o perfil do usuário e a forma como ele irá utilizar a aplicação web. A participação do usuário é considerada crítica para o sucesso do desenvolvimento de uma interface (MATTIA e WEISTROFFER, 2008), por isso, é importante focar durante a construção em formas que permitirão às pessoas que utilizarão o site o melhor acesso possível.

O desenvolvimento de interfaces de grandes sites é sempre multidisciplinar, envolvendo uma equipe de pessoas para concretizá-lo. O tamanho da equipe e a quantidade de pessoas envolvidas variam de acordo com o tamanho do site e os seus requisitos. Cada membro da equipe concentra-se em apenas uma parte do processo (ROBBINS, 2010) na qual ele tem especialidade ou maior conhecimento. Outro papel fundamental para o desenvolvimento de um projeto é a figura de um gerente de projeto (SOMMERVILLE, 2007), para organizar e planejar as atividades além de supervisionar o trabalho dos outros membros da equipe.

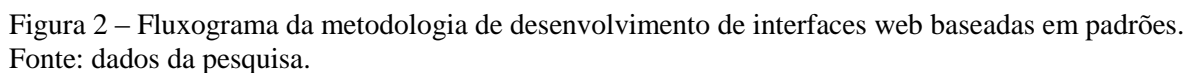
Nesse sentido, outro conceito de padrões - WebApp patterns - define um conjunto de soluções já conhecidas por todos que desenvolvem aplicações web. Esses padrões reúnem o conhecimento e experiência adquirida ao longo de projetos. Os padrões podem ser armazenados em repositórios para que a equipe de desenvolvimento busque por soluções de problemas recorrentes ao processo de criação de aplicações web (PRESSMAN, 2009). Os padrões podem envolver todas as questões relativas ao desenvolvimento de interfaces web. Assim, também contribuem para a construção de aplicações web de qualidade.

4. RESULTADOS

Uma das características do Scrum é tentar utilizar boas práticas desde o começo. A vasta comunicação e aprendizagem ao longo do desenvolvimento de projetos permitem que sempre seja observado aquilo que deu certo e o que deu errado e com isso tomar melhores decisões nas fases seguintes. Dessa forma, não foi difícil adaptar essa metodologia ao desenvolvimento baseado em padrões.

Apesar de parecer que vai de encontro com a filosofia do Scrum, outros autores (SILVA e CUNHA, 2006; COHEN e MONEY, 2008) já tentaram conciliar o Scrum aos modelos tradicionais de desenvolvimento de software. Foi constatado que é possível realizar essa união. As metodologias tradicionais oferecem mais controle a grandes equipes e as ágeis trazem maior agilidade em algumas etapas do processo de desenvolvimento.

Apesar da característica de interação com o usuário que as metodologias ágeis proporcionam, o foco excessivo do Scrum em equipe e prazos pode causar prejuízos, por exemplo, na usabilidade de interfaces (SINGH, 2008). Nesse sentido, a metodologia desenvolvida nesse trabalho tenta aproximar



- Gerente de Projeto;
- Designer Gráfico;
- Arquiteto de Informações;
- Equipe de User Experience (UX);
- Programadores;
- Equipe de Teste.

- **Concepção:** Nessa etapa serão definidos os requisitos da interface, determinados pelo Product Owner. Além de uma pesquisa de usuário realizada pelo Gerente de Projeto será criado nessa fase o Backlog do Produto, uma lista detalhada com tudo que deve haver na interface;

- **Elaboração:** Na etapa de elaboração, um modelo de como a interface deve ser construído. O web designer bem como o arquiteto de informações terão que trabalhar para criar um protótipo executável da interface e apresentar ao Product Owner, de forma a atender as exigências contidas do Backlog da Sprint. É importante também que desde essa fase seja adotado o conceito de padrões. O designer e o arquiteto de informações devem elaborar o seu protótipo de acordo com os padrões disponíveis, de forma a otimizar o desenvolvimento;
- **Construção:** Na construção, ocorrerá o desenvolvimento do codificação em si. Utilizando-se dos conhecimentos de HTML e CSS, sem esquecer da utilização dos padrões escolhidos na etapa anterior. Durante essa etapa, o código estará submetido a constantes testes em busca de erros e a uma avaliação feita pela equipe de UX, que definirá os padrões de usabilidade e acessibilidade a serem alcançados por aquele código, buscando sempre a qualidade do produto. Quando concluída, essa etapa deverá gerar um produto semipronto;
- **Transição:** A essa etapa final, caberá um esforço intenso do Product Owner e do Gerente de Projeto no sentido de avaliar o produto entregue na fase de construção. A equipe de testes realizará os mais variados testes, incluindo e priorizando aqueles em que o produto será diretamente testado com o usuário final. Esse tipo de teste gerará um relatório, que vai servir de base para a análise final, que determinará se o produto vai precisar de reparos.

Uma importante parte da metodologia Scrum diz respeito ao conceito de “done”, ou seja, quando determinada etapa do processo estará pronta, concluída. É importante que o product owner e todas equipes concordem com as definições estabelecidas para o conceito de finalização (KNIBERG, Henrik). Dessa forma, definimos um conceito de done para cada fase, de forma que o produto possa ser avaliado da melhor forma possível e sair com o melhor resultado:

- **Elaboração:** Nessa fase, o conceito de done é definido de acordo com o consentimento do Product Owner. Ou seja, o protótipo só será considerado pronto quando estiver de acordo com o pedido pelo cliente e isso será definido pelo mesmo;
- **Construção:** O conceito de done dessa etapa será um conjunto entre as avaliações feitas pelas equipes de teste e de UX, dependendo de um retorno positivo de ambos os níveis de qualidade esperados;
- **Transição:** O conceito de “done” dessa fase será dado tanto pelo o Product Owner, que dará analisará se todos os critérios expostos nos backlogs do produto foram cumpridos tanto como pela equipe de teste, que verá se a interface está realmente pronta para ser utilizada pelo usuário final.

A figura 3 exibe a participação das equipes ao longo das fases. A participação de cada equipe nessa metodologia é algo que precisa ser cada vez mais analisado, pois pode cada integrante de equipe precisa agir de modo ágil. Como constatado por Overhage, Schlauderer e Birkmeier (2011), é preciso haver um equilíbrio entre autonomia e responsabilidade dos participantes. Assim como um alto nível de habilidades interpessoais.

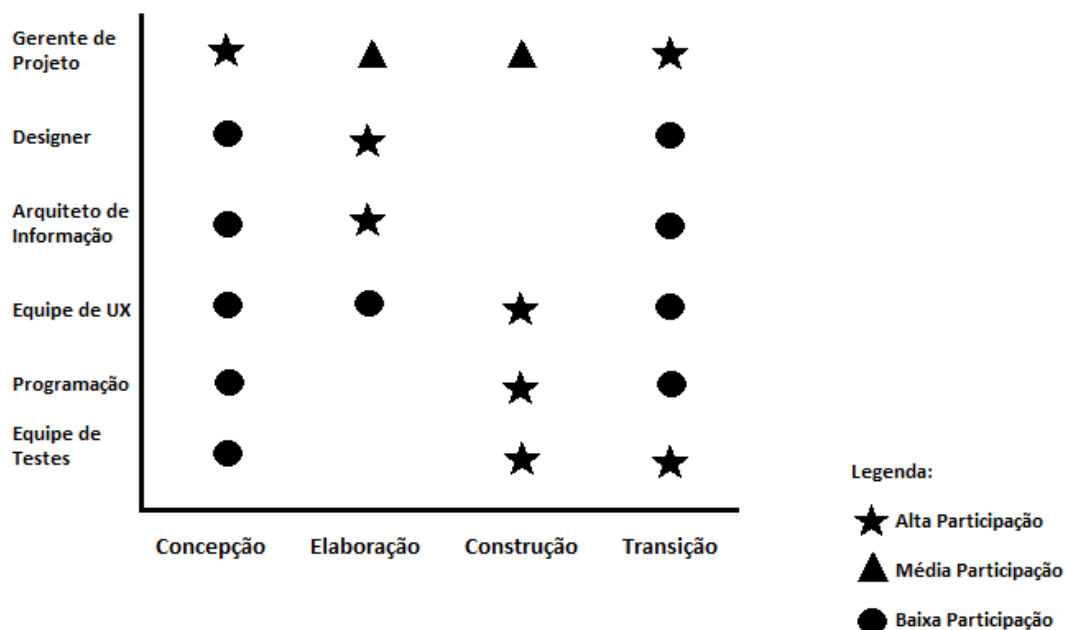


Figura 3 – Participação das equipes durante as fases de desenvolvimento da interface.
 Fonte: dados da pesquisa.

O quadro 1 resume as fases, subetapas, equipes envolvidas e artefatos gerados.

Quadro 1 – Descrição das fases, subetapas, equipes e artefatos gerados.

Etapas	Subetapas	Equipe	Artefato gerado
Concepção	Análise de atividades do usuário (a)	Gerente de Projeto; Designer; Arquiteto de Informação; Equipe de UX; Time de Programadores; Equipe de Teste.	Nesta etapa, é gerado o Relatório de Usuário contendo todas as especificações do projeto.
Elaboração	Produção de Protótipo (b)	Gerente de Projeto; Designer; Arquiteto de Informação; Equipe de UX.	Após a Elaboração, será gerado um protótipo executável do produto.
	Modelo de Arquiteto de Informação (c)		
	Avaliação dos Projetos (d)		
Construção	Avaliação de UX (e)	Gerente de Projeto; Equipe de UX; Time de Programadores; Equipe de Teste.	No fim da etapa de Construção, ao terem sido concluídas as etapas da Avaliação de UX e Programação, o produto já estará como semi-pronto.
	Implementação (f)		
	Teste (g)		
Transição	Avaliação com o Product Owner (h)	Gerente de Projeto; Designer; Arquiteto de Informação; Equipe de UX; Time de Programadores; Equipe de Teste.	Na etapa de transição, após os testes com o Product Owner e com o usuário, é gerado o Relatório de Usuário que Analisa o
	Teste (i)		

			resultado final a partir da visão do cliente, que indicará, depois, se o produto está pronto ou se há alguma falha para ser corrigida.
--	--	--	--

- Gerente de Projeto reúne-se com o Product Owner para receber todas as recomendações sobre o produto e realizará testes para conhecimento do público-alvo do software;
- Web Designer desenvolve um protótipo do produto, baseado nas especificações do Relatório de Usuário e, para tal, busca e consulta uma biblioteca de padrões onde pode selecionar padrões já conhecidos e indicados, de uma biblioteca conhecida ou de alguma que venha a ser desenvolvida em conjunto com os programadores de acordo com o foco do projeto. Pode-se utilizar também uma ferramenta CASE que auxilie a criação da interface com o uso de padrões;
- É estruturado um plano de arquitetura de informação adequado aos objetivos do software, focando a melhor assimilação do conteúdo da página pelos futuros usuários;
- É realizada uma reunião de avaliação dos trabalhos do Designer e do Arquiteto de Informação, junto com o Product Owner, para ser gerado um protótipo executável;
- A Equipe de UX realiza aqui as avaliações necessárias para testar a usabilidade, acessibilidade, etc. do software. A equipe também deverá avaliar esses aspectos nos padrões escolhidos, sendo possível consertá-los para o futuro uso. No caso de já encontrada alguma dificuldade ou necessidade de melhoramento, o projeto é reenviado à etapa de Elaboração para ser corrigido;
- O Time de Programadores implementa o código para o funcionamento do site a partir dos padrões escolhidos pelo web designer. Os programadores poderão também desenvolver novos padrões bem como alterar os já existentes na biblioteca escolhida de forma a melhor adaptar o código;
- A Equipe de Teste realiza as avaliações necessárias sobre o código e a parte visual do site, e retornará o projeto ao Time ou à Equipe de UX caso algum erro seja encontrado;
- O Gerente de Projeto reúne-se com o Product Owner para análise do produto gerado que, se aprovado pelo mesmo, seguirá às demais subetapas;
- Após a avaliação com o Product Owner, são realizadas avaliações da utilização do software com o usuário, determinadas pelo Gerente de Projeto, que podem durar uma semana ou um mês, por exemplo.

5. CONCLUSÕES

A metodologia apresentada nesse trabalho está em sintonia com outros trabalhos que procuram aplicar os pontos fortes das abordagens ágeis e tradicionais no desenvolvimento de software. O Scrum é uma metodologia muito adaptável, mas, ao focar no desenvolvimento de software, foi percebido que ele precisa de muitas alterações para atender as atividades relacionadas à produção de software.

Foi percebido que a rapidez que o Scrum permite a falta de atenção a aspectos importantes no desenvolvimento de interfaces web. Ao combinar o ágil com o formal, o processo ficou mais dinâmico ao mesmo tempo em que se torna mais controlado. O aspecto formal auxilia a superar a falta de atenção que ainda é dada às interfaces web. É preciso ficar sempre atento aos padrões web e isso geralmente não recebe a devida atenção.

Espera-se que essa metodologia possa ser avaliada em situações reais para verificar sua eficácia na produção de interfaces web. A ênfase em padrões por si só já deve proporcionar um aumento de eficiência. Mesmo assim, é preciso observar como equipes se comportam usando a metodologia. Trabalhos futuros podem desenvolver um software que acompanhe a aplicação desse método ou adaptá-lo a outras metodologias ágeis.



REFERÊNCIAS

AGUANNO, Kevin. **Managing agile projects**. Lakefield: Multi-Media Publications Inc., 2005. 422p.

BECK, Kent; BEEDLE, Mike; BENNEKUM, Arie van; COCKBURN, Alistair; CUNNINGHAM, Ward; FOWLER, Martin; GRENNING, James; HIGHSMITH, Jim; HUNT, Andrew; JEFFRIES, Ron; KERN, Jon; MARICK, Brian; MARTIN, Robert C.; MELLOR, Steve; SCHWABER, Ken; SUTHERLAND, Jeff; THOMAS, Dave. Agile Manifesto. 2001. Disponível em: <<http://agilemanifesto.org/>> Acesso em: 08/07/2012. BLOTNER, Joseph A. Agile Techniques to Avoid Firefighting at a Start-Up, 2002.

BENYON, David. **Interação Humano-Computador**. 2 ed. Rio de Janeiro: Pearson Education – BR, 2011. 464 p.

COHEN, Stephen J.; MONEY, William H. Bridge Methods: Complementary Steps Integrating Agile Development Tools & Methods with Formal Process Methodologies. **Proceedings of the 41st Hawaii International Conference on System Sciences**, 2008.

GURGEL, Giovane Montine Moreira ; PINTO, K. L. N. ; ARAUJO, M. A. . Ferramenta CASE para o gerenciamento de componentes de interface web. In: **VI Congresso Norte Nordeste de Pesquisa e Inovação (CONNEPI)**, 2011, Natal. VI CONNEPI, 2011.

MATTIA, Angela; WEISTROFFER, Heinz Roland: Information System Development: A Categorical Analysis of User Participation Approaches. In: **41st Hawaii International Conference on System Sciences**, 2008.

OVERHAGE, Sven; SCHLAUDERER, Sebastian; BIRKMEIER, Dominik. What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance. **Proceedings of the 44th Hawaii International Conference on System Sciences**, 2011.

PRESSMAN, Roger S. **Web engineering : a practitioner's approach**. New York: McGraw-Hill, 2009. 474p.

ROBBINS, Jennifer Niederst. **Aprendendo Web Design**. 3 ed. Porto Alegre: Bookman, 2010. 478 p.

SCHWABER, Ken. **Agile Project Management with Scrum**. Redmond: Microsoft Press, 2004.

SCHWABER, Ken; SUTHERLAND, Jeff. Scrum Guide. 2010. Disponível em <<http://www.scrum.org/scrumguides/>> Acesso em: 17/07/2012.

SILVA, José Gonçalo A. Oliveira Basto da; CUNHA, Paulo Rupino da. Reconciling the Irreconcilable? A Software Development Approach That Combines Agile and Formal. In **39th Hawaii International Conference on System Sciences**, 2006.

SINGH, Mona. U-SCRUM: An Agile Methodology for Promoting Usability. **AGILE '08 Conference**. , n. 4-8, p. 555-560, 2008.

SOMMERVILLE, Ian. **Engenharia de Software**. 8ª edição. Rio de Janeiro: Pearson, 2007. 568 p.



WAZLAWICK, Raul Sidnei. **Metodologia de pesquisa para ciência da computação**. Rio de Janeiro: Elsevier, 2008. 159 p.