

# XP e Scrum sob uma Abordagem Comparativa

Cleyton Vanut Cordeiro de Magalhães<sup>1</sup>, Ronnie Edson de Souza Santos<sup>2</sup> e Ítalo César de Souza Belo<sup>3</sup>

## Introdução

Processo de Desenvolvimento de *Software* é um conjunto de atividades executadas com a finalidade de obter um produto de *software* através da definição de sub-processos, responsáveis, artefatos e um fluxo de realização de atividade [1]. À medida que as organizações tornam-se cada vez mais dependentes da indústria do *software*, ficam mais evidentes os problemas relacionados ao processo de desenvolvimento desses sistemas, tais como: alto custo, alta complexidade, dificuldade de manutenção, e frequentes disparidade entre as necessidades dos usuários e o produto desenvolvido.

A necessidade da construção de sistemas de maneira rápida, ocasionada pelo ritmo acelerado de mudanças na tecnologia da informação e pelo grande dinamismo no ambiente de negócios, fez surgir, a partir de 2001, uma nova abordagem para o processo de desenvolvimento. Nessa nova metodologia, conhecida como Desenvolvimento Ágil, o projeto é visto como um conjunto de pequenas tarefas, ao invés de um processo completo, onde se procura aumentar a leveza e a rapidez no desenvolvimento do *software* [2].

As atividades relacionadas com o desenvolvimento ágil são estabelecidas tomando-se o foco na priorização de indivíduos e interações, no software funcional, na colaboração com o cliente e na resposta a mudanças, ao invés de enfatizar processos e ferramentas, documentação extensiva, negociação de contratos e a atitude de seguir rigidamente um plano. Nos últimos tempos, dois destes métodos ágeis se tornaram bastante populares em ambientes de desenvolvimento, a saber, o XP (Extreme Programming) e o Scrum.

O XP é uma metodologia ágil de desenvolvimento de software através da qual busca-se criar sistemas de melhor qualidade através de um conjunto de valores, princípios e práticas que diferem bastante das tradicionais. Ele é indicado para ser usado com equipes pequenas e médias desenvolvendo software com requisitos vagos ou que mudam frequentemente. Seu foco é a codificação e os valores em que se sustenta são comunicação simples, porém eficiente; simplicidade no design, algoritmo e tecnologias utilizadas; *feedback* em relação à qualidade do código e ao andamento do projeto e coragem para aplicar mudanças que venham a surgir durante o desenvolvimento [3].

O *Scrum* é comumente aplicado a pequenas equipes e em ambientes de desenvolvimento que sofrem constantes mudanças, sendo o seu foco a maximização da habilidade da equipe em responder de forma ágil aos desafios emergentes. Desta forma, pode-se resolver rapidamente qualquer tipo de problema que apareça durante o processo de desenvolvimento, uma vez que todos os membros sabem exatamente o que acontece ao longo do projeto. Suas principais características são: divisão do processo em pequenos ciclos de desenvolvimento chamados *Sprint* (no qual um conjunto de funcionalidades pré-definidas e concluídas são entregues aos clientes em intervalos regulares), o monitoramento do progresso do processo através de reuniões diárias com toda a equipe e, além disso, reuniões com os *stakeholders* no fim de cada ciclo de desenvolvimento (o que transforma o cliente em um integrante da equipe) [4].

Apesar de o XP e o Scrum serem fundamentados em princípios comuns às metodologias ágeis de desenvolvimento de software, eles possuem, nas atividades que os compõem, diferenças bastante visíveis. Levando-se isso em consideração, foi realizada uma comparação entre esses dois métodos, apontando os pontos coincidentes deles, suas divergências e, além disso, suas respectivas utilizações em ambientes de desenvolvimento no Brasil.

## Material e Métodos

O presente trabalho tem um enfoque exploratório e descritivo. Ele foi realizado através de revisão bibliográfica e entrevistas com pessoas que têm experiência com o tema.

## Resultados e Discussão

Observando-se os resultados da pesquisa realizada pode-se verificar que os dois métodos têm como principal semelhança a velocidade de desenvolvimento, que é associada à diminuição dos *bugs* de implementação e a maior qualidade do produto final. Isso se torna possível por haver maior interação da equipe durante o processo de construção do produto e pelo fato de ambos os métodos priorizarem o *software* funcionando ao invés de uma documentação extensa, bastante formal e detalhada. Outro ponto comum entre

<sup>1</sup> Graduando do Curso de Sistemas de Informação da Universidade Federal Rural de Pernambuco - UFRPE, Unidade Acadêmica de Serra Talhada – UAST. E-mail: ronnie.gd@gmail.com

<sup>2</sup> Graduando do Curso de Sistemas de Informação da Universidade Federal Rural de Pernambuco - UFRPE, Unidade Acadêmica de Serra Talhada – UAST. E-mail: cleyton.vanut@gmail.com

<sup>3</sup> Professor Assistente da Universidade Federal Rural de Pernambuco – UFRPE, Unidade Acadêmica de Serra Talhada – UAST. E-mail: italobelo@yahoo.com.br

os dois é a adaptação a mudanças e a maior colaboração do cliente, que recebe com frequência versões executáveis do sistema. Além disso, no fim de cada ciclo, são definidas novas prioridades que devem ser desenvolvidas em um prazo determinado, que é conhecido como iteração.

Em relação às diferenças existentes entre os dois métodos, elas estão presentes em quase todo o processo, desde a implementação de requisitos até a validação das iterações e do produto final (Tabela 1).

O XP, por ser um método orientado a testes, estes são escritos antes da atividade de desenvolvimento e todas as funcionalidades só possuem valor se forem testadas e obtiverem 100% (cem por cento) de aprovação [5]. No *Scrum*, as funcionalidades priorizadas no *Product Backlog*, as quais recebem a denominação de *Sprint Backlog*, são implementadas e testadas paralelamente. Isso ocorre no *Sprint* (período no qual ocorre o desenvolvimento) [6].

No XP é sugerido que, à medida que o código vai sendo gerado, ele deve ser integrado, evitando mais cedo problemas de incompatibilidade. No fim do projeto, a entrega do sistema ocorre quando o cliente estiver completamente satisfeito e não tiver mais nada a acrescentar em relação às funcionalidades [5]. No *Scrum* não há sugestão sobre como a integração deve ser realizada, mas ela ocorre ao final de cada *Sprint*. Além disso, a entrega do sistema acontece quando não existirem mais itens no *Product Backlog* a serem desenvolvidos [6].

Outras particularidades do XP são a programação em duplas, para obtenção de *feedback* do código e encorajamento da equipe, para a realização de mudanças quando necessárias, e a inserção de comentários no próprio código, com o intuito de diminuir a quantidade de documentos. Além disso, a comunicação é feita de forma livre, desde que seja eficaz. No XP, existe um foco maior nas atividades de planejamento, para avaliar as dificuldades e a viabilidade do projeto, e a validação das integrações ocorrem paralelamente à implementação.

No *Scrum*, não há indicação de que é necessário criar documentação no código (como no XP), porém, os documentos tradicionais devem possuir o que for mais relevante e, além disso, necessitam ser reduzidos, com o intuito de no final do processo existir apenas artefatos de grande importância para o projeto. Outra prática exclusiva é a realização de reuniões diárias, a fim de refletir sobre o trabalho do dia anterior, do dia corrente e do seguinte. A validação da integração dos incrementos, diferentemente do XP, ocorre no último dia de cada *Sprint* através de uma reunião chamada de Revisão da *Sprint*.

Quanto à utilização desses métodos ágeis de desenvolvimento de software por empresas brasileiras,

podemos citar a Improve it, sediada no Rio de Janeiro, e a Objective Solutions, em São Paulo, as quais utilizam XP, sendo esta última corporação pioneira na utilização desse método no Brasil. Já o *Scrum*, é utilizado pelas empresas Knowtec, a H2J e a OnCast Technologies, em Santa Catarina, a InterBusiness Technologies, no Paraná, a Ci&T, em São Paulo, a Fundação Paulo Feitosa (FPF), no Amazonas, a FemaHosp, no Maranhão, a Powerlogic S.A., em Minas Gerais e a Living Consultoria, no Rio de Janeiro.

Existem ainda aquelas que optaram por usar ambos os métodos, como é o caso da Locaweb, empresa de hospedagens de sites que vem utilizando-os desde 2007.

## Conclusão

O *Scrum* e o XP compartilham as práticas do manifesto ágil, como equipes pequenas trabalhando em requisitos instáveis ou desconhecidos em iterações curtas para fornecer maior visibilidade no desenvolvimento. Entretanto, eles apresentam também algumas diferenças, como por exemplo, a definição dos objetivos no escopo do projeto e a maneira como realizar o planejamento no decorrer das iterações.

Muitos afirmam que o *Scrum* promove práticas de gerenciamento e engenharia de software enquanto o XP tem seu foco maior nas práticas de engenharia. Diante disso, acredita-se que a melhor maneira de utilizar esses dois métodos é de forma complementar, ou seja, deve-se unir os pontos que os dois possuem em comum com as melhores práticas de cada um em busca da melhor qualidade do processo de desenvolvimento de software.

## Agradecimentos

Agradecemos a Jeff Sutherland, Igor Macaúbas e Marcos Silva Pereira pelas respostas referentes ao contato por *e-mail*.

## Referências

- [1] SANTOS, R.E.S.; MAGALHÃES, C.V.C; CORREIA NETO, J.S. 2009. *Scrum – Principais Vantagens do Método no Processo de Desenvolvimento de Software*. In: 1º Encontro Regional de Tecnologia e Negócios, UFRPE/UAST.
- [2] FREITAS, H. R.; MIRANDA, J. M. 2008. *Metodologias Ágeis para o Desenvolvimento de Software, Scrum e Extreme Programming X Metodologias Tradicionais RUP*. Universidade Cândido Mendes – Campos, RJ, Brasil.
- [3] Improve it. *Extreme Programming*. Disponível em <http://improveit.com.br/xp> Acesso em set. 2009.
- [4] SUTHERLAND, J.; JAKOBSEN, C. R.; JOHNSON, K. 2007. *Scrum and CMMI Level 5: The Magic Potion for Code Warriors*. IEEE Computer Society, USA.

- [5] BECK K. *Extreme Programming Explained*. 2000. Massachusetts: Addison-Wesley.
- [6] SCHWABER, K. 1995. *Scrum Development Process*. In: *OOPSLA'95 Workshop on Business Object Design and Implementation*. Springer-Verlag. Atlanta, USA.

**Tabela 1:** Principais diferenças entre Scrum e XP.

Atividade	Scrum	XP
<b>Implementação</b>	<p>As funcionalidades priorizadas no <i>Product backlog</i> formam o <i>Sprint backlog</i> e são implementadas na <i>Sprint</i>.</p> <p>É necessário criar a documentação do sistema, mas não há indicação de que ela deve ser feita no código-fonte.</p> <p>A implementação é feita em duplas.</p>	<p>Estórias do usuário são transformadas em funcionalidades do sistema em cada iteração por duplas de programadores.</p> <p>Deve-se criar uma documentação do sistema no código-fonte.</p> <p>Não há uma indicação de que a implementação deve ser realizada em duplas. Embora não existam restrições sobre a adoção dessa prática.</p>
<b>Testes</b>	Testes acontecem paralelamente à implementação.	O desenvolvimento é orientado a testes, ou seja, estes devem ser escritos antes das funcionalidades serem implementadas.
<b>Integração do sistema</b>	A integração das funcionalidades ocorre no fim de cada Sprint.	A integração acontece à medida que o código é gerado.
<b>Entrega</b>	A entrega ocorre quando não existirem mais itens no <i>Product Backlog</i> .	A entrega ocorre quando não tiver mais nada a acrescentar em relação às funcionalidades.