

Experimentos da programação em pares baseada em um ambiente de desenvolvimento de software

João de Jesus P. Sousa¹, Christopher P. Nascimento¹, William R. Malvezzi¹

¹Pós graduação em residência em Engenharia de Software – Centro de pós graduação e extensão (CPGE). FUCAPI - Av. Gov. Danilo de Mattos Areosa, 381 - Distrito Industrial - CEP: 69075-351 CEP: 69075-351 – Manaus – AM – Brasil

{joaodejesusps,workcpn}@gmail.com, wmalvezzi@hotmail.com

Abstract. *This article presents experiences occurred in an environment of software development, through of pair programming of a specialization course in software engineering.*

Resumo. *Este artigo apresenta experiências ocorridas de um ambiente de desenvolvimento de software, por meio da programação em pares de um curso de especialização em engenharia de software.*

1. Introdução

Desde o ano de 1996, quando Kent Beck criou a metodologia XP (*Extreme Programming*), na qual constatou que a prática de se programar em equipe e em dupla, mostrou-se um dos melhores métodos a serem executados no desenvolvimento de *software*. Para Souza (2007), essa prática é viável devido a quatro dimensões a partir das quais os projetos podem ser melhorados: Comunicação, Simplicidade, *FeedBack* e Coragem.

De acordo com, Soares (2004), uma das vantagens da programação em dupla é a possibilidade dos desenvolvedores estarem continuamente aprendendo um com o outro. Com isso, sabe-se que para toda forma de trabalho, existe vantagens e desvantagens. A intenção da programação em dupla é fazer com que as pessoas se tornem mais comunicativas, tornando-as flexíveis, trazendo para o ambiente de trabalho o diálogo mútuo, aonde os desenvolvedores não venham a transformar o recinto de trabalho como se fosse uma disputa de conhecimento individual e esquecendo que, o mais interessante às empresas produtoras de *software* é a satisfação do cliente pela qualidade do produto adquirido.

Este artigo está organizado em 5 seções. A seção 2 expõe conceitos relevantes referentes à proposta por meio das citações, a seção 3 apresenta trabalhos relacionados, a seção 4 discorre tanto sobre o experimento quanto aos resultados adquiridos e por fim a seção 5 descreve a conclusão deste trabalho.

2. Apresentação Teórica

Na programação em pares é importante levar em consideração dois papéis fundamentais: o piloto e o copiloto. A dupla deve entrar em consenso para que não venha ocorrer à concentração de um único papel ficar sobrecarregando um membro da dupla (Melo et al., 2010). Com base nessas informações pode-se afirmar que, o aprendizado só é enriquecido à parte mais fraca, quando esta manifestar sua vontade em adquirir o conhecimento repassado pela parte mais forte, caso contrário, não tem ajuda que resolva o problema de quem não está interessado em aprender. Para (Borges et al., 2007), é necessário, ainda, que os pares sejam permutados a cada nova atividade, de forma a disseminar o conhecimento adquirido entre os aprendizes.

Segundo Berni (2010), “a programação em pares pode ter benefícios como melhorar a qualidade do código, evitar o retrabalho, melhorar a comunicação da equipe, ou simplesmente aumentar a produtividade por ter uma pessoa sempre ao lado”. Já para (Silva et al., 2004), “todo o código produzido é escrito com dois programadores em cada máquina. Em cada ciclo trocam-se os pares de programadores”.

3. Trabalhos Relacionados

(Rocha et al., 2011) relata a prática de programação em pares, vivenciada por residentes ingressos da FabSoft e a relevância dessa técnica.

Williams (2008) mostra uma visão geral da programação em pares tanto no mercado de trabalho quanto no ambiente educacional e a constatação da relevância dessa técnica em relação ao desempenho dos estudantes para a formação profissional como na aderência de uma nova forma de interação do conhecimento.

4. Apresentação do estudo do ambiente

Para sustentar esses argumentos foi realizado um estudo de caso, baseado em um ambiente sólido de programação de *software*.

A pressuposição desvendada por este estudo é ocorrente de uma observação direta evidenciado em um ambiente de desenvolvimento de *software*. Alguns desenvolvedores opinam que o "método em pares" não traz nenhum ganho na produtividade do processo de desenvolvimento

A intenção deste estudo de caso é realizar uma analogia entre os dois métodos de se programar “em pares” e “unitária”, qual dos dois métodos observados é mais relevante para se aplicar em um ambiente de desenvolvimento de *software*.

4.1. Estudantes envolvidos

Dezesseis estudantes de um curso de especialização em engenharia de *software*, sendo seis do sexo feminino e dez do sexo masculino. Estes foram divididos em duas equipes, cada uma com oito integrantes, sendo que as mesmas receberam projetos distintos. Os estudantes eram habituados na programação unitária e nunca tinham exercido o papel da programação em pares.

4.2. Procedimentos utilizados

Para facilitar o entendimento dos *frameworks* que seriam utilizados no processo de desenvolvimento para ambas as equipes, foram realizados treinamentos da infra a ser utilizada para os dois projetos tanto à equipe 1 de programação em pares como para a equipe 2 de desenvolvimento unitário.

No projeto 1 com a equipe 1 – O sistema a ser desenvolvido foi estipulado em cinco interações (*sprint*), sendo que, esta era composta de quatro pares. Cada dupla desenvolvia uma funcionalidade do sistema por interação/*sprint*, totalizando 20 funcionalidades.

No projeto 2 com a equipe 2 – o sistema a ser desenvolvido foi estipulado em quatro interações (*sprint*), sendo que, esta era composta de oito desenvolvedores, onde cada um desenvolvia parcialmente uma ou duas funcionalidades do sistema por interação/*sprint*, totalizando 16 funcionalidades.

As equipes iniciaram as atividades no mesmo período, sendo que a equipe dois utilizou o método unitário até o fim. A Equipe um iniciou e concluiu seu projeto com a programação em pares.

4.3. Resultados

Apesar do projeto 1 ter uma quantidade de funcionalidades maior em relação ao projeto 2, esta conseguiu realizar suas entregas em tempo hábil como mostra a Tabela 1.

A equipe 2, mesmo tendo um número menor de funcionalidades esta só conseguiu entregar todas as funcionalidades do projeto com 20% de atraso em relação ao tempo estipulado como se pode observar na Tabela 1.

Tabela 1. Número de Requisitos e Tempo

	Equipe 1	Equipe 2	Entrega Equipe 1	Entrega Equipe 2	(%)Atraso Equipe 1	(%) Atraso Equipe 2
Total de Funcionalidades	20	16	20	16	-	-
Sprint	5	4	-	-	-	20
Total de Desenvolvedores	8	8	-	-	-	-
Funcionalidades Concluídas	20	16	20	16	-	-

A equipe 2 não obteve sucesso em suas entregas, devido ao método utilizado promover esforços concentrados em apenas uma pessoa, originando erros devido a ausência de comunicação os quais deveriam ser reconhecidos acarretando impactos indesejáveis como: falta de qualidade do código, atraso na entrega, excesso de erros no teste, retrabalho e aumento do custo. Ver figura 1:

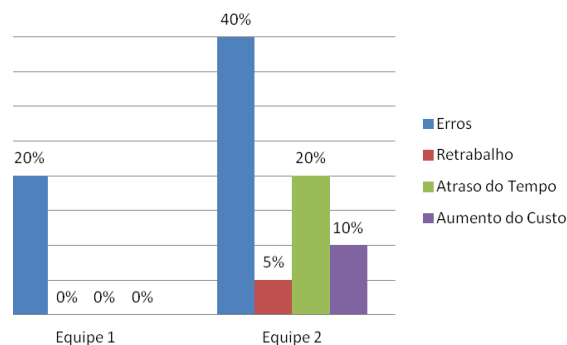


Figura 1. Demonstrativo dos resultados obtidos entre as duas equipes

4. Conclusão

A percepção que se tem com a aplicação dessa metodologia de programação em pares demonstrada por meio de pesquisas e análise de um ambiente de desenvolvimento de *software*, onde ela foi utilizada é que, através dos rendimentos adquiridos, obtém-se a relevância dessa metodologia em relação aos benefícios alcançados como a qualidade do *software*, minimização de erros, eliminação de retrabalho, cumprimento do prazo estipulado e redução de custos. Para ressaltar esta demonstração comparativa entre essas duas políticas de implementação, fez-se a evidência de dois projetos advindos dos resultados que destacam a proeminência do modelo em pares.

Referências

Berni, J. C. A. (2010). "Gestão para o processo de desenvolvimento de software científico, utilizando uma abordagem ágil e adaptativa na microempresa". Disponível em: <http://cascavel.cpd.ufsm.br/tede/tde_arquivos/12/TDE-2010-05-07T112147Z-2605/Publico/BERNI,%20JEAN%20CARLO%20ALBIERO.pdf>. Acessado em: 23 de Abril de 2012.

- Borges, R. M., Pinto, S. C. C. S., Barbosa, J. L. V. Barbosa, D. N. F. (2007). "Usando o modelo 3C de colaboração e Vygotsky no ensino de programação distribuída em pares". Disponível em: <<http://ceie-sbc.tempsite.ws/pub/index.php/sbie/article/view/555/541>> Acessado em: 02 de Maio de 2012.
- Melo, C. O., Santos Junior, C. D., Ferreira, G. R. M., Kon, F. (2010). "Um estudo exploratório dos fatores associados ao estímulo do aprendizado em times ágeis na indústria", In: 7th Experimental Software Engineering Latin American Workshop, 2010, Goiania.
- Rocha, C. L. A., Botelho, E. A. S., Araújo, M. V. C. (2011). "Programação em Pares no Desenvolvimento dos Residentes da Fabsoft", XIV Semana de Informática - EIN. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/ein/2011/Artigo_12.pdf>. Acessado em: 26 de Abril de 2012.
- Silva, L. F., Leite, J. C. S. P., Breitman, K. K. (2004). "Ensino de Engenharia de Software: Relato de Experiências", XII WEI - Workshop de Educação em Informática - SBC. Disponível em: <http://www.inf.puc-rio.br/~lyrene/silva_WEI2004.pdf>. Acessado em: 22 de Maio de 2012.
- Soares, M. S. (2004) "Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software". Disponível em: <http://wiki.dcc.ufba.br/pub/Aside/ProjetoBitecIsaac/Met._Ageis.pdf>. Acessado em: 30 de Abril de 2012.
- Souza, L. M. (2007). "Método Ágil XP (EXTREME PROGRAMMING)". Disponível em: <http://intranet.fia.edu.br/aceso_site/fia/academos/revista3/6.pdf>. Acessado em: 02 de Maio de 2012.
- Williams, L. (2008). "Pair Programming". Disponível em: <http://collaboration.csc.ncsu.edu/laurie/Papers/ESE%20WilliamsPairProgramming_V2.pdf>. Acessado em: 28 de Junho de 2012.