

Desenvolvendo Aplicações Moveis com Programação Pareada: um estudo de caso

Hugo M. T. de Azevedo¹, Lucyan do N. Peixoto¹, Adilson J. dos Santos¹

¹Centro Universitário Cesmac – Maceió – AL – Brasil

{hugo_targino,lucyan}@outlook.com, adilson.al@gmail.com

Resumo. *Entre as práticas da Extreme Programming (XP), a programação pareada é uma das mais aplicadas, sobretudo pela capacidade em reduzir o número de erros no código, melhorando a sua qualidade. Assim, o presente trabalho aborda, a partir do desenvolvimento de um app para a plataforma Android, os benefícios de se trabalhar com programação em par.*

Abstract. *Among the practices of Extreme Programming (XP) the pair programming is one of the most applied, above all capacity to reduce the number of errors in the code, improving its quality. So, this work approach, from the development of an app for the Android platform, the benefits of working with pair programming.*

1. Introdução

A forma utilizada pelos métodos tradicionais para o desenvolvimento de software se depara com algumas dificuldades em seu processo. O cliente não tem participação ativa no andamento do projeto, tendo assim muitas vezes resultado de funcionalidades que não agrega valor, acarretando prejuízo financeiro e desperdício de mão de obra. Muitas vezes o processo toma mais tempo do que o previsto pelo cronograma, gerando gastos inesperados no orçamento ou cortando-se etapas que levam ao desenvolvimento software de baixa qualidade. Dados apresentados pelo Standish Group (2009 apud STEFFEN, 2012) mostram que 24% do software falharam antes da sua entrega, 32% tiveram sucesso e 44% foram entregues, mas não cumpriram o que foi prometido, seja por atraso na entrega, ou por não atender as necessidades do cliente, ou por apresentarem defeitos.

Neste contexto, as Metodologias Ágeis surgiram como resposta para os problemas encontrados no processo de desenvolvimento de software, tendo como intuito o aumento da satisfação do cliente, focando em funcionalidades que mais agregam valor ao produto final, acelerando o processo de desenvolvimento e produzindo software de alta qualidade (LIBARDI; BARBOSA, 2010, p.1).

Jeffries (2013), diz que *Extreme Programming (XP)* é uma metodologia ágil que se constitui em valores e práticas que devem ser estudadas e aplicadas ao desenvolvimento de software, sempre visando a satisfação do cliente. Diante dessas práticas existe a da programação pareada, que possibilita que o desenvolvimento se torne mais eficaz, pois como se trabalha em dupla, ambos pensam na solução de um

mesmo problema e assim é possível produzir código de melhor qualidade, praticamente eliminando os erros.

Este trabalho tem como objetivo analisar a programação pareada a partir do desenvolvimento de um protótipo para dispositivos móveis, com foco na aprendizagem da equipe (par).

Inicialmente, foi realizada uma pesquisa bibliográfica sobre o tema da pesquisa. Após esta etapa, foi realizada a Programação em Par de um protótipo de aplicativo para *Android*. O protótipo foi desenvolvido nas salas da biblioteca x, localizada no x, sendo utilizado o *Android SDK*, desenvolvido em linguagem nativa *Java* na plataforma *Java Development Kit (JDK)*, *Java SE 7u13* e utilizando o ambiente de desenvolvimento *Eclipse*, versão 4.2.

Durante a prática da Programação em Par, foram observadas todas as dificuldades e também as facilidades encontradas, sendo anotadas para posterior análise. O código desenvolvido foi analisado para verificar a incidência de erros e não atendimento a padronização adotada.

2 Fundamentação Teórica

De acordo com Costa Filho et al. (2005), os métodos ágeis adotam a forma de adaptação para o desenvolvimento de software, onde o planejamento da construção do software é de forma contínua e o levantamento de requisitos é identificado aos poucos, tornando o desenvolvimento do projeto mais flexível para a realização de mudanças, ou seja, para o desenvolvimento de projetos onde os requisitos do sistema sejam instáveis e o planejamento imprevisível, é ideal a utilização de metodologias ágeis, onde “para ser realmente considerada ágil a metodologia deve aceitar a mudança ao invés de tentar prever o futuro.” (SOARES, 2004, p.5)

A *Extreme Programming (XP)* é a Metodologia Ágil mais conhecida para o desenvolvimento de software, seguindo os princípios propostos pelo Manifesto Ágil e tendo como objetivo criar sistemas de melhor qualidade, enfatizando o desenvolvimento do projeto de maneira rápida, focando na satisfação do cliente e gerando uma maior economia (CASTRO, 2006, p.28). Essa metodologia foi criada por Kent Beck e Ward Cunningham a partir de ideias e experiências no desenvolvimento de software. (SANTANA JUNIOR, 2008, p.8)

Conforme Wells (2009), *Extreme Programming (XP)* apresenta valores e não um conjunto de regras, onde valores são maneiras de trabalhar com harmonia. A XP tem como base cinco valores fundamentais para o andamento do projeto: *Feedback*, Comunicação, Simplicidade, Coragem e Respeito. “*Feedback* é o mecanismo fundamental que permite que o cliente conduza o desenvolvimento diariamente e garanta que a equipe direcione as suas atenções para aquilo que irá gerar mais valor.”(TELES, 2006a, p. 22)

Para a utilização dos valores, a XP propõe práticas a serem estudadas e praticadas. “As práticas são vetores de onde você está para onde você pode chegar com XP. Em XP, você progride em direção a esse estado ideal de desenvolvimento eficaz.” (TELESa, 2006, p. 1)

Borborema (2007) afirma que as práticas que o XP propõe são coleções de atividades que devem ser realizadas pela equipe de desenvolvimento para a produção de software. Com a aplicação dessas práticas em conjunto é apresentada uma maior

eficácia na construção de um projeto de software, pois ocorre um complemento de uma prática para outra, assim demonstrando um excelente resultado do produto final.

As práticas são divididas em duas partes: primárias e corolárias. As práticas primárias podem ser adotadas de início e de forma segura para melhorar o esforço de desenvolvimento de software. Cada uma delas pode gerar um benefício imediato. As práticas corolárias são de domínio mais difícil e devem ser adotadas depois de se colocar em uso as práticas primárias. (TELES, 2006b).

A programação em par faz parte das práticas primárias, sendo uma das práticas mais utilizadas na metodologia *Extreme Programming* (XP). Consiste em formar duplas de programadores que trabalham no mesmo problema, ao mesmo tempo e utilizando apenas um computador. Belem (2009) afirma que na programação em par existem dois papéis: o Condutor que digita o código, e o Navegador que revisa cada linha de código que for digitada, procurando por erros e pensando na aplicação como um todo, visando sempre a melhoria no código.

Segundo Kniberg (2006), algumas vantagens para a utilização da prática da programação em par são: o aumento da qualidade do código, disseminação e nivelamento do conhecimento da equipe, erros são praticamente eliminados, expansão de soluções para o problema, aumento do foco da equipe, pressão do par, entre outros. Para Teles (2006c), “um dos principais benefícios da programação em par é a permanente inspeção de código que ocorre durante seu uso”.

Com a utilização da programação em par ocorre a revisão contínua do código, pois enquanto o condutor está digitando, o navegador está revisando o código para correção imediata de eventuais erros que possam aparecer. “Enquanto um desenvolvedor está focado no código e às vezes não percebe quando insere erros no software, seu par tem uma visão melhor de um todo e ajuda-o a enxergar os erros facilmente.” (MAIA, 2012, p.1). Assim ocorrendo uma melhora na qualidade do software. Além disso, com a utilização da programação em par o conhecimento é disseminado, ou seja, a equipe sabe como funciona o sistema, dando uma maior flexibilidade e retirando a dependência de um único desenvolvedor.

Teles (2006c) chama atenção para o fato da programação em par ser bastante eficaz para a disseminação do conhecimento da equipe, pois a prática elabora um ambiente de aprendizado contínuo por se trabalhar em par, onde o revezamento dos pares proporciona um maior compartilhamento de informações com toda a equipe de desenvolvimento e proporcionando também o conhecimento amplo do projeto que a equipe se encontra desenvolvendo. O que é corroborado por Gomes (2011, p. 1), ao afirmar que “existem diversas opiniões sobre o mesmo assunto, cada um pensa de uma maneira diferente, tiveram experiências de vida diferentes. Programar em par faz essa diversidade no modo de pensar ser uma maneira de aprender mais, enxergar o problema de diversos pontos de vista”.

Em relação ao erro no código, Teles (2006c) diz que erros encontrados imediatamente no desenvolvimento são fáceis de ser corrigidos e que para a correção posterior irão consumir um maior tempo para correção.

Com relação ao nivelamento de conhecimento Kuhn e Pamplona (2009) afirmam que um dos grandes benefícios da programação em par é a troca de conhecimento que ela proporciona, pois a troca de experiências torna o

desenvolvimento do software mais simples. “A Programação em Par também permite a troca de conhecimentos entre desenvolvedores menos experientes e desenvolvedores mais experientes, tendo como benefício uma equipe nivelada sempre por cima” (ABDO, 2012, p.1) e o conhecimento de toda a equipe de desenvolvimento é ampliado e compartilhado entre a equipe. (SILVA, 2012, p.25). Essa ideia é reforçada por Soares (2004, p.4), quando diz que “uma grande vantagem da programação em dupla é a possibilidade dos desenvolvedores estarem continuamente aprendendo um com o outro.” (SOARES, 2004, p.4)

Manter o foco na implementação do código é bastante complicado para o desenvolvedor devido a gama de distrações que ocorrem no dia-a-dia, como exemplo: redes sociais, bate-papo, e-mail, problemas pessoais entre outros, ocasionando diminuição no andamento do projeto, ainda mais que a ação de codificar necessita de bastante concentração (TELES, 2004, p. 92). Sendo assim “a programação em par faz a concentração aumentar, pois sempre terá alguém do seu lado esperando a resolução do problema, interagindo com você” (GOMES, 2011, p. 1). Segundo Teles (2004), a programação em par auxilia no foco por meio da pressão do par, onde a responsabilidade e o compromisso aumentam com o projeto, devido ao envolvimento de outra pessoa, reduzindo drasticamente as distrações e assim proporcionar uma melhor produtividade do par.

Fica evidente que a utilização da programação em par dará um grande auxílio para um melhor desenvolvimento de software, evitando algumas das dificuldades encontradas na construção de um projeto.

3 Implantação e Análise da Programação em Par

3.1 Escopo do protótipo

O protótipo consiste em uma aplicação móvel para ajudar os usuários a cumprir metas pessoais, como o desenvolvimento de um projeto, a preparação para um concurso, a realização de determinado trabalho etc. A proposta é que o usuário insira as metas que deseja alcançar em determinado período de tempo e associe a cada uma delas uma determinada penalidade (prenda) já cadastrada, que ele mesmo deve se impor caso não cumpra a meta no prazo estipulado.

As penalidades têm como objetivo incentivar o usuário a cumprir suas metas, e consiste em compromisso dele consigo mesmo, embora faça parte do escopo futuro o desenvolvimento de funcionalidade para tornar este compromisso público através das redes sociais, se o usuário assim desejar. Cada prenda pode ser a realização de um trabalho social ou comunitário (assim conscientizando os usuários da importância do seu apoio para tornar a sociedade mais justa), ou outros pequenos gestos menos nobres, sempre com foco no incentivo ao cumprimento da meta pelo usuário.

3.2 Ambiente do protótipo

Para realizar o cadastro de nova meta, o usuário deverá acionar o botão "Adicionar Meta" (Figura 1). Ao fazer isso, ele será transferido para uma nova tela de cadastro (Figura 2). Nesta, o usuário terá disponível um formulário onde serão inseridos os dados da meta desejada: título; data inicial; data final (data de cumprimento da meta); prenda a cumprir caso a meta não seja alcançada.

Figura 1 - Tela inicial do protótipo

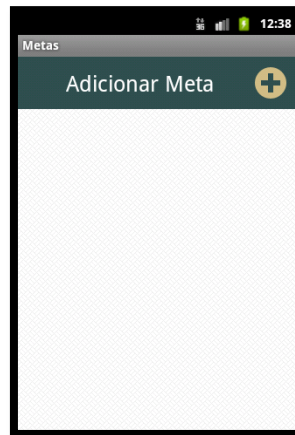


Figura 2 - Formulário de cadastro



Após inserir o título, ao clicar nos botões das datas, será mostrado um alerta para que o usuário informe (Figura 3) as mesmas, de acordo com o período desejado para a realização da meta.

Figura 3 - Alerta com formulário de data



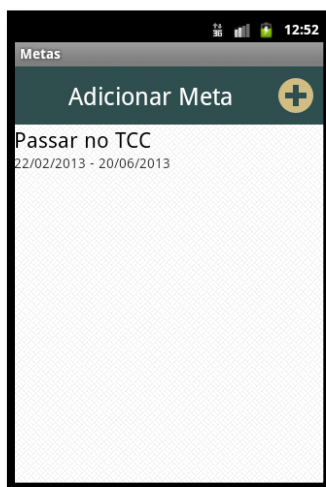
Clicando no botão das prendas, será mostrada uma lista com as prendas (Figura 4) que o usuário deve escolher e se comprometer em pagar caso não alcance a meta desejada.

Figura 4 - Alerta com lista de prendas cadastradas



Com todos os campos preenchidos, o usuário clica no botão "Criar" (Figura 2), para que a meta seja cadastrada e assim o aplicativo é redirecionado para a tela inicial sendo possível verificar a meta que foi cadastrada (Figura 5).

Figura 5 - Tela inicial com metas cadastradas



3.3 Resultados obtidos com a utilização da programação em par

No começo a aprendizagem foi consideravelmente lenta, já que a plataforma na qual o protótipo foi desenvolvido nunca tinha sido utilizada por ambos os programadores. Provavelmente, se um dos componentes da dupla tivesse o conhecimento da plataforma poderia ter acelerado o processo de aprendizagem através da colaboração, onde o conhecimento adquirido seria maior e assim teria uma maior facilidade no processo de desenvolvimento do protótipo. A maioria das dúvidas encontradas pela dupla no decorrer do desenvolvimento do protótipo foram semelhantes, sendo a busca pela possível solução realizada de forma separada, e depois analisada de forma conjunta, para um melhor entendimento, assim possibilitando o conhecimento da plataforma e melhorando o processo de aprendizagem. Com o passar do tempo a dupla começou a se familiarizar com a plataforma e, conseqüentemente, as dúvidas foram diminuindo.

O uso da programação em par possibilitou a disseminação do conhecimento, haja vista que algumas das experiências que foram obtidas de trabalhos individuais foram compartilhadas, auxiliando assim no nivelamento (para cima) do conhecimento dos desenvolvedores.

Outra vantagem que foi identificada com a utilização da prática está relacionada com o maior conhecimento e melhor entendimento do código, pelo fato de se ter trabalhado em par, durante todo o período do desenvolvimento do protótipo. Isso possibilitou que ambos trabalhassem no mesmo código e conseqüentemente proporcionou o conhecimento do código como um todo, algo que não seria tão fácil de ser realizado trabalhando separadamente, pois neste caso seria necessário, durante um período de tempo, realizar explicação do código para que ocorresse o entendimento do mesmo pelo outro desenvolvedor, incorrendo em tempo adicional no desenvolvimento do protótipo.

Utilizando a programação em par se obteve a construção de um código de alta qualidade, já que um dos desenvolvedores (o navegador) por determinado momento ficava responsável por analisar o código e encontrar possíveis maneiras de melhorá-lo, ou seja, deixá-lo mais simples (legível). Antes de realizar as mudanças no código era realizado um pequeno debate sobre a maneira de melhorá-lo e assim possibilitar que o condutor ficasse a par da ideia proposta pelo navegador.

O único dos benefícios relatados na literatura consultada sobre programação pareada que não foi observado durante o desenvolvimento do protótipo foi a pressão do par. De fato, ocorreram distrações de ambas as partes por se trabalhar em ambiente não favorável e aberto ao público, além das distrações do dia-a-dia (e-mail, telefone, redes sociais etc.), de onde se conclui que a pressão do par não funcionou como se esperava. Na interpretação desse resultado, também deve ser considerado que o trabalho se deu fora de ambiente profissional (empresa), onde os desenvolvedores provavelmente possuíam um nível menor de compromisso com metas e prazos rígidos.

Enfim, a prática da programação em par mostrou ser bastante eficiente no desenvolvimento do protótipo, contribuindo com a aprendizagem de uma nova plataforma (Android), no que possibilitou soluções simples e eficazes por meio da comunicação verbal contínua. Dessa forma, trouxe uma maior confiança na elaboração do código.

4 Considerações Finais

No decorrer do presente trabalho, discutiu-se sobre os problemas que afetam o desenvolvimento de software, e como a adoção de metodologias ágeis é uma opção viável para amenizar ou até mesmo evitar a ocorrência desses problemas, melhorando o processo como um todo.

Como parte da pesquisa, foi desenvolvido um protótipo para dispositivos móveis *Android*, utilizando a prática da programação pareada da *Extreme Programming* em um contexto de aprendizagem da linguagem, buscando mostrar as vantagens do uso da referida prática no desenvolvimento de software.

Foi possível confirmar que com a utilização da programação em par ocorreram alguns benefícios, em relação ao desenvolvimento tradicional (individual). Erros foram facilmente encontrados e corrigidos no momento da codificação, houve aprendizagem colaborativa sobre o novo ambiente, e a criação do protótipo transcorreu de forma tranquila. No decorrer do processo, aconteceram conversações para melhorar a qualidade do código e o trabalho de desenvolvimento em par possibilitou a disseminação de conhecimento.

Referências

- ABDO, S. (2012) "Programação em Par". Disponível em: <<http://tiemprosa.com.br/programacao-em-par/>>. Acesso em: 24 abr. 2013.
- BELEM, T. (2009) Programação em par. Disponível em: <<http://imasters.com.br/artigo/13992/outros/programacao-em-par/>>. Acesso em: 11 mar. 2013.
- BORBOREMA, T. (2007) "Impacto da aplicação da metodologia XP nas organizações de desenvolvimento de software". Disponível em: <http://artigocientifico.uol.com.br/uploads/artc_1196095637_11.pdf>. Acesso em: 01 Mai. 2013
- CASTRO, A. V. (2008) "Desenvolvimento ágil com programação extrema eficácia e disciplina extrema no desenvolvimento orientado a objetos de software". Disponível em: <http://devagil.files.wordpress.com/2008/01/desenvolvimentoagil_com_xp_beta10_vinicius_ac0.pdf>. Acesso em: 21 abr. 2013.

- COSTA FILHO, E. (2005) et al. Padrões e Métodos Ágeis: agilidade no processo de desenvolvimento de software. Disponível em:
<<http://sugarloafplop2005.icmc.usp.br/papers/9673.pdf>>. Acesso em: 15 abr. 2013.
- GOMES, D. (2011) "Programação em par". Disponível em:
<<http://blog.danielfariati.com/2011/08/22/programacao-em-par/>>. Acesso em: 15 mar. 2013.
- JEFFRIES, E. R. (2013) "What is Extreme Programming?". Disponível em:
<<http://xprogramming.com/what-is-extreme-programming/>>. Acesso em: 23 abr. 2013.
- KNIBERG, H. (2006) "Scrum e XP direto das Trincheiras". Disponível em:
<<http://www.cti.ufu.br/sites/cti.ufu.br/files/scrum-e-xp-direto-das-trincheiras.pdf>>. Acesso em 11 mar. 2013.
- KUHN, G; PAMPLONA, V. (2009) "Apresentando XP: encante seus clientes com Extreme Programming". Disponível em: <<http://javafree.uol.com.br/artigo/871447/>>. Acesso em: 16 mar. 2013.
- LIBARDI, P; BARBOSA, V. (2010) "Métodos Ágeis". Disponível em:
<http://www.ft.unicamp.br/liag/Gerenciamento/monografias/monografia_metodos_ageis.pdf>. Acesso em: 16 mar. 2013.
- MAIA, F. (2012) "Grandes vantagens da programação em par". Disponível em:
<<http://fredericoarantes.wordpress.com/2012/07/31/grandes-vantagens-da-programacao-em-par/>>. Acesso em 16 mar. 2013.
- SANTANA JÚNIOR, A. C. (2008) "Avaliação da utilização de metodologias ágeis no contexto dos modelos de qualidade de software". Disponível em:
<<http://www.dominiopublico.gov.br/download/texto/cp099169.pdf>>. Acesso em: 21 abr. 2013.
- SILVA, E. (2012) "Desenvolvimento Guiado por Testes". Disponível em:
<<http://www.fatecsp.br/dti/tcc/tcc00061.pdf>>. Acesso em: 16 mar. 2013.
- SOARES, M. (2004) "Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software". Disponível em:
<<http://www.dcc.ufla.br/infocomp/artigos/v3.2/art02.pdf>>. Acesso em: 15 abr. 2013.
- STEFFEN, J. (2012) "O que são essas tais de metodologias Ágeis?". Disponível em:
<https://www.ibm.com/developerworks/mydeveloperworks/blogs/rationalbrasil/entry/mas_o_que_s_c3_a3o_essas_tais_de_metodologias__c3_a1geis?lang=en>. Acesso em: 11 mar. 2013.
- TELES, V. M. (2004) "Extreme Programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade", São Paulo: Novatec.
- _____. (2006a) "Práticas do XP". Disponível em:
<http://improveit.com.br/xp/praticas/programacao_par>. Acesso em: 11 mar. 2013.
- _____. (2006b) "Extreme Programming". Disponível em:
<<http://desenvolvimentoagil.com.br/xp/praticas/>>. Acesso em: 05 mar. 2014.
- _____. (2006c) "Programação em Par". Disponível em:
<http://improveit.com.br/xp/praticas/programacao_par>. Acesso em: 11 mar. 2013.

WELLS, D. (2009) "The Values of Extreme Programming". Disponível em:
<<http://www.extremeprogramming.org/values.html>>. Acesso em: 21 abr.2013.