



## 5º Congresso de Pós-Graduação

### PRODUTIVIDADE DE DESENVOLVEDORES SOFTWARE PROGRAMANDO EM PAR

#### Autor(es)

GERSON NUNHO CARRIEL

#### Orientador(es)

PLÍNIO ROBERTO SOUZA VILELA

#### 1. Introdução

Na engenharia de software existe a busca por melhora nos processo para desenvolvimento de Software. Segundo PFLEEGER(1998), um melhor entendimento sobre quais características levam para um código rapidamente bem desenvolvido, irá nos ajudar a treinar desenvolvedores a serem mais eficazes e eficientes. A programação em pares tem sido apontada como uma boa alternativa para a melhoria da produtividade em desenvolvimento de software. BECK(2000) considera que a produtividade de dois programadores trabalhando em par é mais alta que dois programadores trabalhando em separado. Em resposta a essa afirmação foi realizado um trabalho que visa verificar essa afirmativa sobre produtividade em software, isso foi feito através de um experimento onde foi realizada análise sobre defeitos e o tempo de desenvolvedores trabalhando em pares e em separado. Este documento visa apresentar os resultados desse experimento realizado na Universidade Metodista de Piracicaba, com a função de confirmar ou não a afirmação, que os programadores em pares produzem software de melhor qualidade, com menos defeitos e com produtividade superior segundo BECK(2000) e JEFFRIES(2001). Para realizar a validação ou refutação da afirmativa, o experimento foi realizado com duas equipes de voluntários, sendo uma para programação em pares e outra para programação tradicional. Entende-se como programação em par, a atividade de dois programadores trabalhando ativamente no mesmo problema, na mesma máquina no mesmo momento, conforme AUER(2002) e a programação tradicional é aquela onde o indivíduo trabalha só em seu computador, que é o padrão no mercado de trabalho. A principio podemos ser levados a acreditar que a produtividade será dividida em dois, que dois programadores em par realizarão a mesma tarefa demorando o dobro do tempo que dois programadores realizando essa atividade em separado, porém como pode ser visto em trabalho realizado com o mesmo enfoque, o resultado apresentado não é esse, WILLIANS(2000) apresenta resultado com uma diferença de 15% do tempo a mais para o desenvolvimento em par da mesma tarefa, porém essa forma de trabalhar, que demora um pouco a mais, resulta em 15% menos defeitos. Através dessas informações SUCCI(2001) afirma que esses 15% a mais no tempo é recuperado na correção de defeitos e mostra que os defeitos a menos tornam o custo da programação em par inferior, considerando assim a programação em par mais produtiva.

## 2. Objetivos

---

O objetivo do trabalho é descobrir se a programação em pares resulta em melhor produtividade no desenvolvimento de software.

## 3. Desenvolvimento

---

Nas sessões seguintes teremos a apresentação do que foi o experimento, como foi a formação da equipe e o planejamento para realização do experimento.

3.1 - O Experimento A importância do tema é verificar se a programação em pares pode ser utilizada na engenharia de software como uma ferramenta que aumente a produtividade da equipe de desenvolvimento, pois gera um código mais livre de defeitos e portanto um código de melhor qualidade. Para realização foi organizado um experimento onde existiriam dois grupos trabalhando a solução de um mesmo problema, esse grupos trabalhariam de forma diferente, um grupo em programação em pares e o outro na forma tradicional. Ao final do experimento era esperado que o número de defeitos entre as duas equipes fosse analisada e que o tempo de desenvolvimento da atividade também fosse considerada para comprovar ou não a afirmação da produtividade na modalidade de programação em par.

3.2 - Formação da equipe Para formar a equipe convidamos alguns alunos da Universidade de Piracicaba, dos convidados existia um que já havia terminado o último ano, não era mais aluno da Universidade. O grupo de voluntários ficou em cinco pessoas. No dia do experimento, um dos cinco não compareceu, e foram formados dois grupos com dois desenvolvedores.

3.3 - Planejamento do Experimento O experimento ficou limitado em pessoas e no tempo disponível, foram quatro voluntários e o tempo para o desenvolvimento do experimento ficou definido em quatro horas. Para que existisse a possibilidade de realização do experimento, os voluntários foram entrevistados previamente em reunião realizada uma semana antes do experimento ser realizado, para avaliação do conhecimento que eles já possuíam em desenvolvimento de software. Dessa reunião, ficou definido que o projeto utilizaria o Banco de Dados MySQL e o software desenvolvido seria em PHP e JavaScript, as linguagens que o grupo mais dominava. Após definição tecnológica, ficou claro que existia a necessidade de um detalhamento grande da atividade que seria desenvolvida e deixando para os voluntário apenas a preocupação em desenvolver em PHP, para garantir que no período a atividade seria finalizada dentro da janela de tempo disponível dos voluntários, ficou planejado que os programadores receberiam: Uma especificação de requisitos, com detalhamento técnico do banco de dados e lay-out de tela; As telas deveriam ser entregues em HTML, para que os desenvolvedores trabalhassem apenas o desenvolvimento em PHP e o JavaScript nas validações da aplicação; Foi entregue, juntamente com a especificação, um script para criação do Banco de Dados e das tabelas necessárias para o desenvolvimento do projeto; Outra situação definida anteriormente era que a aplicação a ser desenvolvida deveria ter utilidade prática para motivar o grupo, nesse sentido foi projetada uma aplicação para realização de enquetes na internet com a restrição de um usuário poder responder apenas uma única vez a enquete.

3.4 - O dia do Experimento No dia do experimento foi apresentado para o grupo o objetivo da aplicação, isso às 8h30m da manhã, deixado claro que não existia nenhuma disputa entre os grupos e que eles deveriam no tempo estimado criar uma aplicação simples de enquete, as validações necessárias de navegação feitas através de JavaScript e que os dados básico necessários estariam no banco de dados, os dois grupos levaram em média uma hora para organizar o ambiente, criar a estrutura de banco de dados e deixar o ambiente disponível para realização do experimento, o desenvolvimento do software, que iniciou às 9h30m.

## 4. Resultados

---

As duas duplas conseguiram finalizar as tarefas, e ambos os aplicativos analisados estavam 100% funcionais e, segundo avaliação, livres de defeitos. A dupla de programadores, que trabalhava cada uma em seu computador, finalizou a tarefa às 11h:30m, 45 minutos antes dos programadores em par, que finalizaram as atividade às 12h15m. O resultado disse experimento deveria apresentar dois tipos de informações: 1a. Quantos erros cada um dos grupos realizou. 2a. Quanto tempo demorariam cada uma das equipes. Para a primeira questão, não tivemos resposta alguma, ao final os dois grupos realizaram a tarefa e entregaram o software sem defeito algum. Isso pode ser entendido devido a natureza da aplicação que não possuía muita

iteração com o usuário e não apresentava muita complexidade. A segunda questão nos forneceu a informação que os programadores trabalhando em par, na mesma máquina, demoraram 37,5% a mais do tempo que os programadores que trabalharam em máquinas diferentes. Além desses valores em relação ao tempo, foi verificado também que na dupla que programava em máquinas separadas a existência de uma diferença na experiência dos programadores com as tecnologias utilizadas, sendo que um deles desenvolveu 2/3 da carga dos trabalho enquanto o outro desenvolveu o 1/3 restante.

## 5. Considerações Finais

---

Ao final, com a inexistência de defeitos nas aplicações geradas pelas duas equipes, não foi possível verificar a validação ou não da afirmativa de que a programação em par apresenta menos defeitos que a programação tradicional. Em relação ao tempo para o desenvolvimento tivemos um resultado maior do que o esperado para os programadores em par. O resultado de 37,5% a mais no tempo necessário para a realização da tarefa pela dupla programando em par está acima de outros experimentos, que atingem 15% a mais do tempo de desenvolvimento em comparação com o tempo de dois programadores em equipamentos separados, como pode ser visto em WILLIAMS(2000). Uma dificuldade encontrada está no número de participantes do grupo, que era pequeno e não dava margem para cálculo de desvios, como pode ser visto na atividade da dupla de programadores em máquinas separadas que teve a distribuição de atividades dividida em 2/3 da carga de trabalho para um com mais experiência e o outro 1/3 da carga de trabalho realizada pelo outro integrante da dupla. Deste experimento fica evidenciada a necessidade de uma amostra maior, para que os valores dos dados possam ser validados através de técnicas estatísticas e que os desvios não gerem influência direta nos resultados. Outro fato percebido foi o de não existir experiência por parte dos programadores em par em trabalhar juntos em um único computador, isso mostrou a necessidade de existir um tempo maior para que aprendam a trabalhar em pares, habituando os participantes com o processo e garantindo assim que não exista desvios nos resultados por falta de experiência nessa modalidade de programação, já a programação individual é a naturalmente exercida por todos e não necessita de treinamento específico. Nesse experimento ficaram algumas lições para trabalhos futuros: A primeira delas é o grau de complexidade da aplicação, esse deverá ser maior, e ser uma aplicação onde o usuário tenha mais interação. A segunda reside no fato de que os programadores em par devem possuir maior experiência com essa nova forma de trabalhar, portanto é necessário que os programadores em par façam um período de adaptação ao método.

## Referências Bibliográficas

---

- BECK, Kent. **Extreme programming explained: embrace the change**. Addison Wesley, 2000.
- AUER, Ken; MILLER, Roy. **Extreme Programming Applied**. Addison Wesley. 2002.
- JEFFRIES, Ron; ANDERSON, Ann; HENDRICKSON, Chet. **Extreme Programming Installed**. Addison Wesley.2001.
- PFLIEGER, Shari Lawrence. **Software Engineering theory and practice**. Prentice Hall, 1998.
- SUCCI, Giancarlo; Marchesi, Micheli. **Extreme Programming Examined**. Addison Wesley. 2001.
- WILLIAMS, Laurie; KESSLER, Robert; CUNNINGHAM, Ward, et al. **Strengthening the case for pair programming**. IEEE Software, v. 17, n. 4, 2000.