

Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise

Erik Arisholm
Hans Gallis
Tore Dyba
Dag I.K. Sjøberg

IEEE Transactions on Software Engineering
2007

Pedro Araujo Pires

Introdução

O que é *pair programming*?

- 2 programadores na mesma tarefa
- 1 computador e 1 teclado
- 2 papéis:
 - Motorista (escreve o código e foca nos detalhes de codificação)
 - Navegador (verifica o trabalho do motorista, busca alternativas, pesquisa referências, etc.)

Introdução

- Vários estudos sobre *pair programming* foram feitos, e chegaram à conclusão de que PP tem vários benefícios sobre a programação individual:
 - Aumento da corretude e de outras medidas de qualidade de software.
 - Diminuição do tempo de desenvolvimento (*time to market*), com *overhead* ínfimo de programadores-hora.

Introdução

- No entanto, esses estudos não podem ser comparados:
 - Amostra da população (estudantes x profissionais).
 - Quantidade de treino em PP.
 - Falta de poder estatístico (poucos *subjects*).
 - Diferentes tratamentos às variáveis dependentes (como a corretude foi medida? Tempo de desenvolvimento inclui tempo de retrabalho?)

Introdução

- Há um ponto em comum:
 - Nenhum deles leva em conta a complexidade das tarefas de programação.
 - Pode depender da complexidade do sistema, e da experiência dos programadores.
- É esperado que a complexidade dos sistemas e a experiência dos programadores tenham um grande impacto em **quando** e **como** PP é benéfico.

Introdução

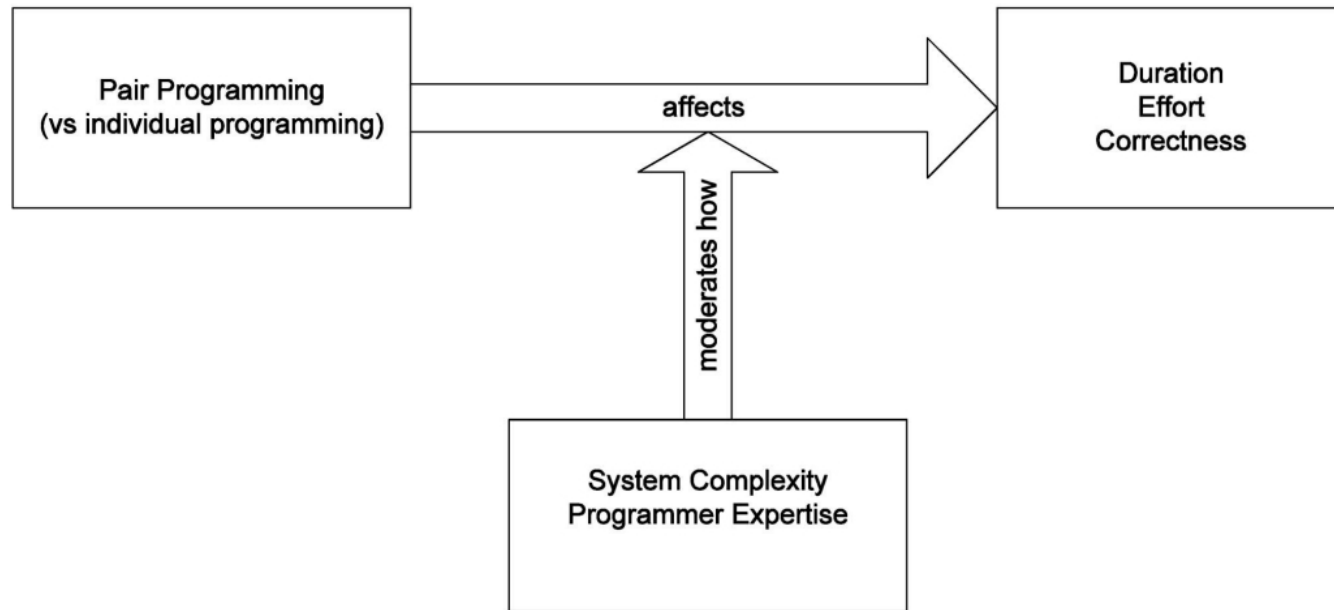
- Para investigar essas questões, foi realizado um estudo empírico para responder à seguinte questão:

What is the effect regarding duration, effort and correctness of pair programming for various levels of system complexity and programmer expertise when performing change tasks?

Experimento

Modelo Conceitual

- Duração → *time to market*
- Esforço → custos
- Corretude → qualidade



Experimento

- Complexidade do sistema
 - Estilo de controle centralizado x delegado
- Experiência dos programadores
 - Categoria (júnior, intermediário, sênior)
 - Pré-teste.

Experimento

- Duração
 - Tempo para executar um conjunto de tarefas.
 - Somente considerado para tarefas executadas corretamente.
- Esforço
 - Total de programadores-hora necessários para realizar as tarefas.
- Corretude
 - Programa final possui a funcionalidade requerida?

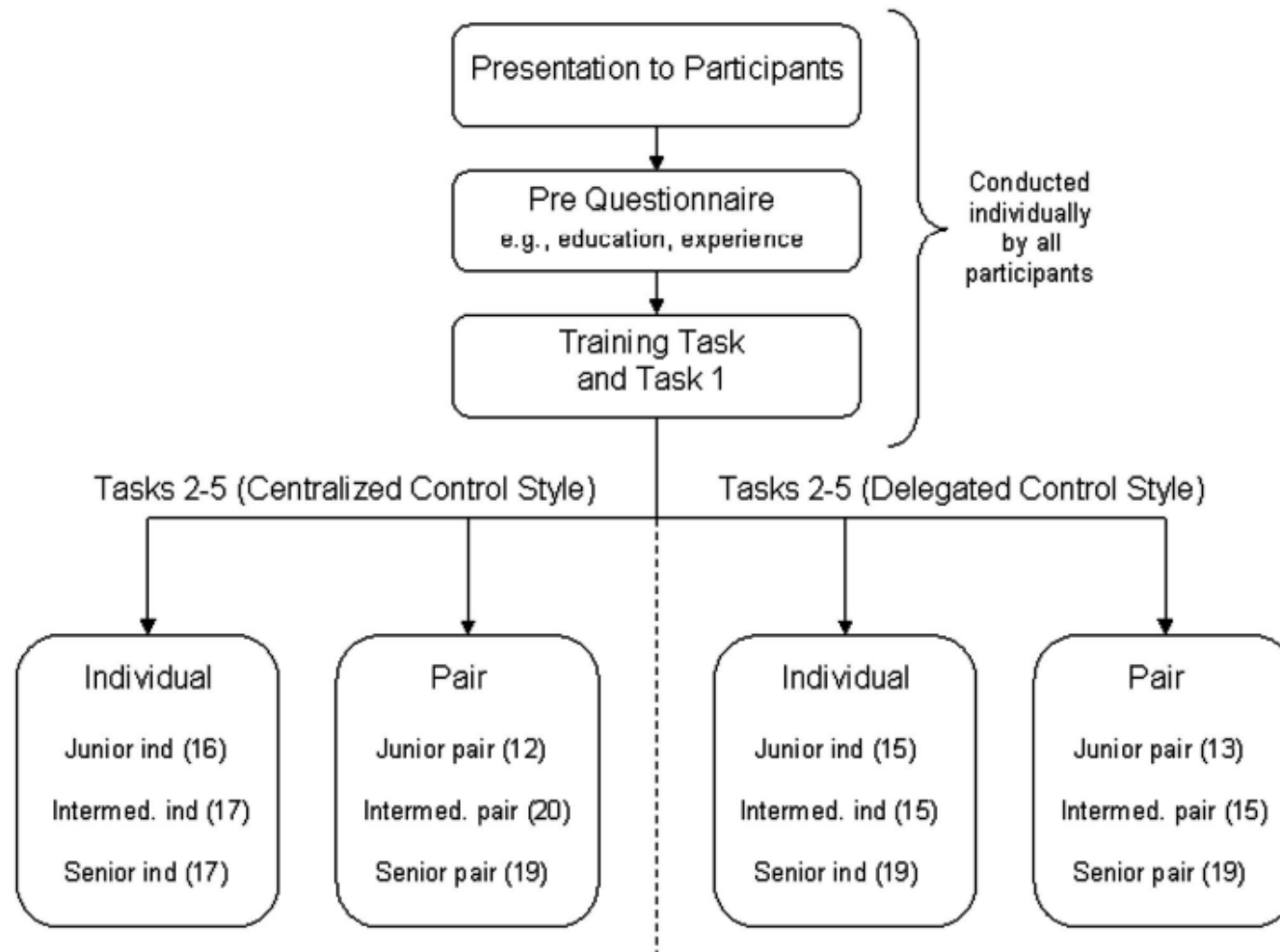
Experimento

H0 ₁	Effect of Pair Programming on Duration: The time taken to perform change tasks is equal for individuals and pairs.
H0 ₂	Moderating Effect of System Complexity on Duration: The difference in the time taken to perform change tasks for pairs versus individuals does not depend on system complexity.
H0 ₃	Moderating Effect of Programmer Expertise on Duration: The difference in the time taken to perform change tasks for pairs versus individuals does not depend on programmer expertise.
H0 ₄	Effect of Pair Programming on Effort: The effort expended on performing change tasks is equal for individuals and pairs.
H0 ₅	Moderating Effect of System Complexity on Effort: The difference in the effort expended on performing change tasks for individuals and pairs does not depend on system complexity.
H0 ₆	Moderating Effect of Programmer Expertise on Effort: The difference in the effort expended on performing change tasks for pairs versus individuals does not depend on programmer expertise.
H0 ₇	Effect of Pair Programming on Correctness: The correctness of the maintained programs is equal for individuals and pairs.
H0 ₈	Moderating Effect of System Complexity on Correctness: The difference in the correctness of the maintained programs for pairs versus individuals does not depend on system complexity.
H0 ₉	Moderating Effect of Programmer Expertise on Correctness: The difference in the correctness of the maintained programs for pairs versus individuals does not depend on programmer expertise.

Experimento

- 2 fases (2001 e 2004/05)
- Amostra de 295 consultores Java
 - 99 Individuais
 - 98 Pares
- Pré-teste para avaliar diferenças de habilidade entre os dois grupos.

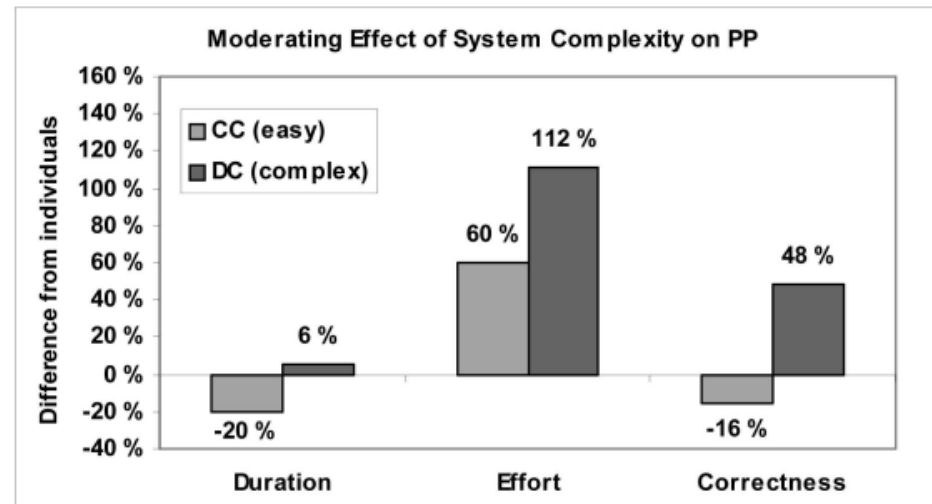
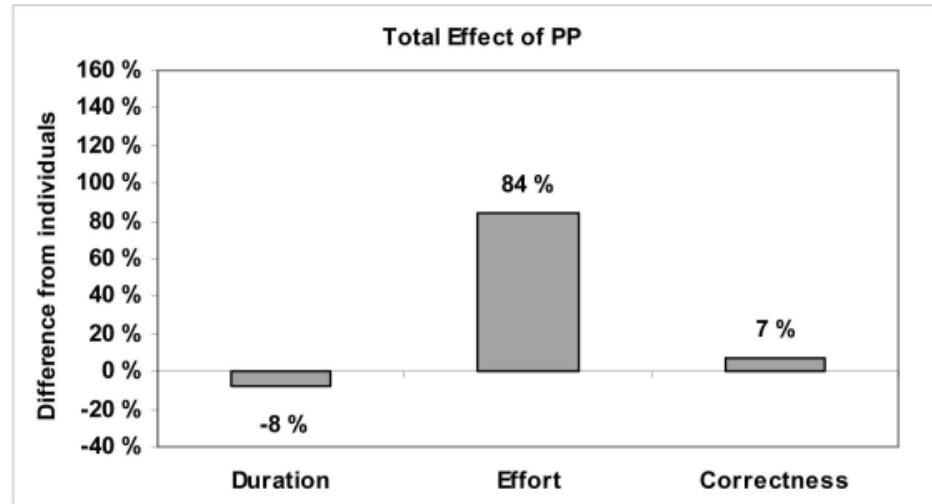
Experimento



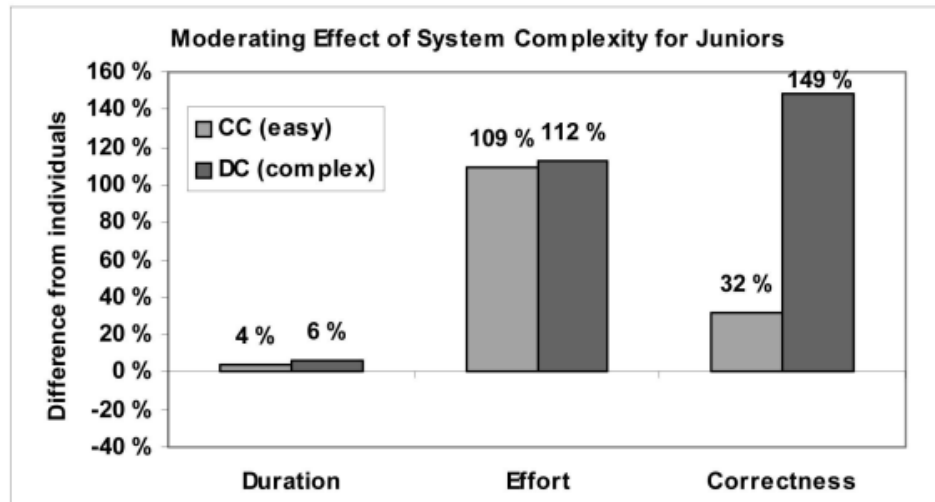
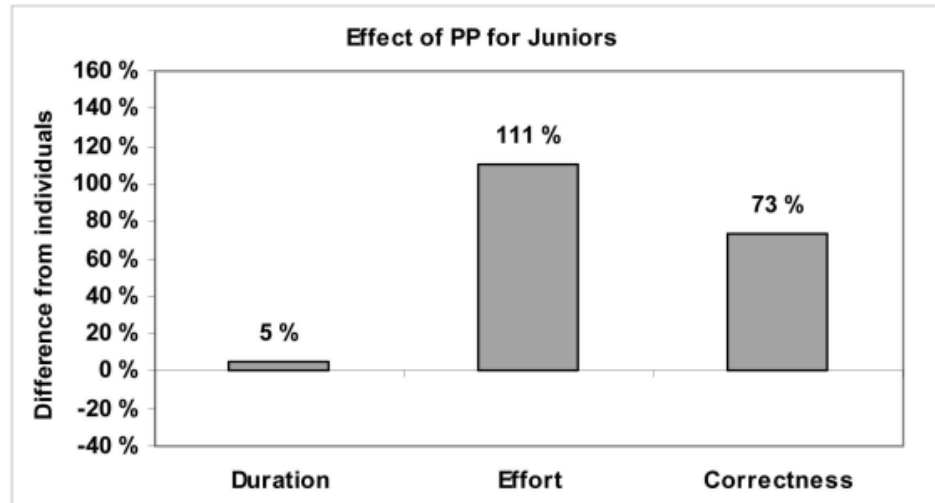
Experimento

- Tarefas: *Coffee Machine*
 - Implementar botão de “*coin return*”.
 - Implementar “caldo” como nova bebida.
 - Checar se todos os ingredientes estão disponíveis para a bebida escolhida.
 - Fazer uma bebida a partir dos ingredientes disponíveis.

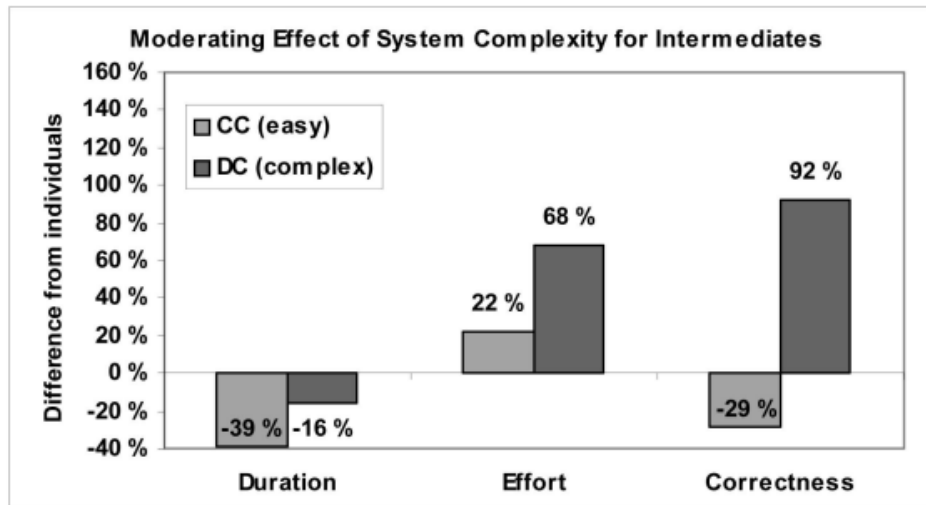
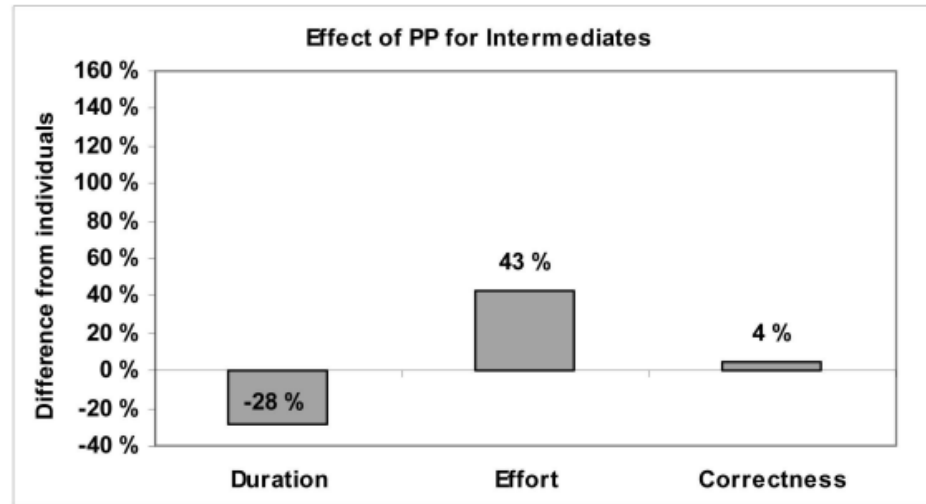
Resultados



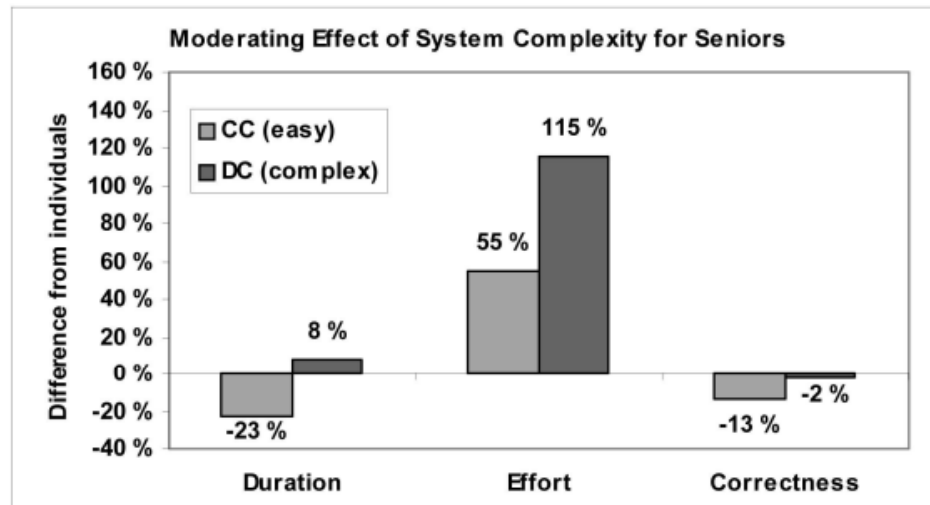
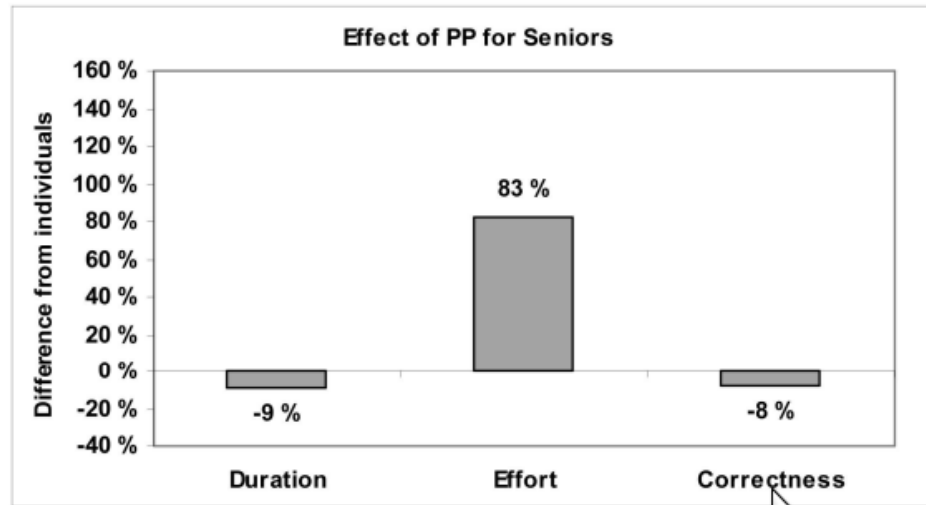
Resultados



Resultados



Resultados



Resultados

H0	Reject?	Effect size estimates
H0 ₁	no	The pairs worked 8% faster than did individuals.
H0 ₂	yes	For the <i>CC</i> control style, the pairs worked 20% faster than did individuals. For the <i>DC</i> control style, the pairs worked 6% slower than did individuals.
H0 ₃	no	Junior pairs worked 5% slower than did junior individuals. Intermediate pairs worked 28% faster than did intermediate individuals. Senior pairs worked 9% faster than did senior individuals.
H0 ₄	yes	The pairs required 84% more effort than did individuals.
H0 ₅	yes	For the <i>CC</i> control style, the pairs required 60% more effort than did individuals. For the <i>DC</i> control style, the pairs required 112% more effort than did individuals.
H0 ₆	no	Junior pairs required 111% more effort than did junior individuals. Intermediate pairs required 43% more effort than did intermediate individuals. Senior pairs required 83% more effort than did senior individuals.
H0 ₇	no	The pairs had a 7% increase in the proportion of correct solutions (odds ratio = 1.28).
H0 ₈	yes	For the <i>CC</i> control style, the pairs had a 16% decrease in the proportion of correct solutions compared with the individuals (odds ratio = 0.46). For the <i>DC</i> control style, the pairs had a 48% increase in the proportion of correct solutions compared with the individuals (odds ratio = 3.56).
H0 ₉	no	The junior pairs had a 73% increase in the proportion of correct solutions compared with the individuals (odds ratio = 5.48). The intermediate pairs had a 4% increase in the proportion of correct solutions compared with the individuals (odds ratio = 1.15). The senior pairs had an 8% decrease in the proportion of correct solutions compared with the individuals (odds ratio = 0.65).

Conclusões

- Programadores junior individuais não têm habilidade necessária para realizar tarefas com qualidade aceitável (especialmente as mais complexas).
- Programadores junior em pares obtiveram um aumento significativo em corretude (comparados com os junior individuais), e aproximadamente a mesma que programadores senior individuais.
- Se programadores junior forem designados para tarefas complexas de manutenção, eles devem trabalhar em pares.

Conclusões

- *Pair programming* é útil para programadores júnior resolvendo tarefas complexas de manutenção.
 - Aproximadamente, gastam o mesmo tempo e obtém a mesma corretude que programadores sênior individuais.

Trabalhos Futuros

- Experimentos foram feitos com sistemas relativamente pequenos.
 - Experimentos futuros serão feitos com sistemas maiores e com tarefas mais complexas.
- Duplas nunca haviam trabalhado juntos antes.
 - Experimentos futuros serão feitos com diferentes níveis de coesão da dupla.