Diretrizes RUP®/XP: Programação aos Pares

Robert C. Martin
Object Mentor, Inc.

Rational Software White Paper

TP 158, 3/01

Rational®
the software development company

Índice Analítico

Visão Geral 1
Emparelhamento, uma breve descrição 1
O caso para emparelhamento 1
A Prática 1
Emparelhamento 1
Alterando os pares 2
Propriedade coletiva 2
Andamento e colaboração 2
O que o desenvolvedor solitário pode fazer? 2
Algumas pessoas não gostam do emparelhamento 3
Mobília, Dependências e Logísticas 3
Local do monitor e do teclado 3
Cercado 3
Nos cantos 3
Problemas e Preocupações 3
O emparelhamento divide a produtividade em dois 3
Disputas entre parceiros 3
Especialistas 3
Barulho 4
Caubóis 4
Impedimentos físicos ou de estilo 4
Como a equipe pode planejar o emparelhamento? 4
Conclusão 4
Referências 4

Visão Geral

Emparelhamento, uma breve descrição

A Programação aos Pares é uma técnica em que o software em um projeto é gravado por pares de programadores. Cada par funciona em uma única estação de trabalho. Um membro do par dirige a estação de trabalho enquanto o outro observa, cuidadosamente o código sendo produzido. O condutor está pensando táticamente, preocupado com a linha de código que está escrevendo atualmente. O observador está validando a sintaxe e está pensando estrategicamente em todo o programa. Eles trocam essas funções freqüentemente e o código resultante é gravado mais rápido do que se por uma única pessoa e tem menos defeitos. Além disso, o código é intimamente conhecido por pelo menos dois desenvolvedores.

O caso para emparelhamento

Considere uma sessão de revisão de código típica. Um módulo que exigiu oito horas para que uma pessoa o desenvolvesse é revisado por uma hora por oito pessoas. O resultado da rede é que 16 horas por pessoa são gastas no módulo. No entanto, os revisores não podem gastar o tempo necessário para se familiarizarem com o código e sua revisão é bastante superficial. Um único desenvolvedor está intimamente familiarizado, mas talvez familiarizado demais para localizar grande parte dos defeitos.

Contraste isso com a prática da programação aos pares. Se o módulo requer oito horas para ser desenvolvido, um total de 16 horas por pessoa será expandido. No entanto, neste caso, *dois* desenvolvedores terão conhecimento íntimo do código. Os defeitos ocultos de um desenvolvedor estarão visíveis para o outro.

O caso para programação aos pares é simples, mas as repercussões são sutis e difíceis de alcançar. A programação aos pares é simplesmente uma maneira muito mais efetiva de gravar e revisar o código. Com duas pessoas intimamente familiarizadas com um módulo, muito menos defeitos serão gravados no código. O código terá uma estrutura melhor e um conhecimento íntimo estará em duas vezes mais cérebros. Se esses fossem os únicos benefícios, eles seriam suficientes, mas o ato de emparelhamento fornece ainda mais benefícios.

Os pares são mais corajosos: o que um único programador pode ter medo de tentar, um par terá a coragem de tentar e a habilidade de avaliar.

O emparelhamento estimula o trabalho em equipe: já que os módulos não são gravados por uma única pessoa, o código se torna propriedade da equipe, em vez de um desenvolvedor específico.

O emparelhamento estimula a difusão do conhecimento: quanto mais desenvolvedores se emparelharem uns com os outros, mais conhecimento do sistema é difundido por toda a equipe. O resultado é uma equipe cujo os membros estão familiarizados com todo o sistema em vez de cada membro saber somente sua parte específica.

O emparelhamento promove a produtividade: uma pessoa programando sozinha passa por intermitências de energia seguidas por períodos de inatividade relativa. Os pares medem os passos um do outro. Quando um fica cansado, eles trocam as funções. Eles tentam manter a intensidade ligada por muito mais tempo do que uma única pessoa pode normalmente tolerar.

O emparelhamento é divertido: trabalhar com outro desenvolvedor é educacional, estimulante e muito divertido. O emparelhamento aumenta a satisfação profissional e a ética geral.

A Prática

Emparelhamento

O emparelhamento começa quando o desenvolvedor responsável por uma tarefa pede ajuda a alguém. A regra é: quando solicitado, você deve dizer sim. Isso não significa que você tem que parar o que está fazendo imediatamente. Em vez disso, significa que você deve negociar um tempo em que você possa oferecer essa ajuda e outro tempo em que você possa obter ajuda.

O parceiro não assume a responsabilidade pela tarefa. Essa responsabilidade permanece com o proprietário da tarefa. Nem o parceiro se compromete a ficar com o proprietário até que a tarefa esteja concluída. O parceiro se compromete apenas a ajudar.

Um membro do par se torna o condutor enquanto o outro observa. O condutor digita o código, executa o compilador, os testes de unidade e assim por diante. O observador examina cada teclada, cada comando, cada resultado de teste e oferece ajuda e sugestões. Ambas as partes estão comprometidas todas as vezes.

Às vezes, o condutor saberá melhor o que fazer e o observador estará simplesmente acompanhando. Em outras vezes, o observador ditará o que fazer para o condutor. Às vezes, o condutor ficará frustrado e entregará o teclado para o observador, trocando as funções. Outras vezes, o observador pedirá o teclado e trocará as funções. Isso acontecerá muitas vezes em uma sessão de emparelhamento.

Alterando os pares

Os parceiros não são em longo prazo. Uma sessão de emparelhamento típica dura em média meio dia. O parceiro pode desistir da dupla por qualquer motivo. Quando isso ocorre, o proprietário da tarefa deve encontrar outro parceiro. Isso pode significar que é a vez do proprietário da tarefa retornar um favor para alguém que emparelhou com ele ou ela na semana passada. Por outro lado, talvez ele ou ela deva pedir a alguém com a experiência certa para ajudar com um problema particularmente difícil.

Essa alteração de pares faz com que o conhecimento do sistema seja difundido por toda a equipe de desenvolvimento. Dentro de pouco tempo, cada membro da equipe terá gasto tempo trabalhando em quase todas as partes do sistema. Isso reduz drasticamente a sensibilidade de modificação do projeto e torna cada programador mais confiante em lidar com todo o sistema.

Propriedade coletiva

Já que todos trabalham em todos os módulos diferentes no sistema, ninguém tem um módulo específico. Isso significa que a responsabilidade para o sistema não não está dividida em uma base de módulo por módulo. Em vez disso, toda a equipe é coletivamente responsável por todo o sistema. Qualquer membro da equipe pode registrar a saída e alterar qualquer módulo no sistema por qualquer motivo. Quando um par faz uma alteração para o módulo X e que a alteração faz com que os testes de unidade do módulo Y falhem, o par repara o módulo Y.

Andamento e colaboração

A programação aos pares é uma forma muito intensa de comunicação. O diálogo verbal é freqüentemente discutido e um observador externo pode ter problemas em entender o sentido. Como um observador, você pode ouvir o par expressando palavras singulares como: “ponto e vírgula” ou “parênteses”. Ou você pode simplesmente ouvir gemidos menos articulados enquanto os programadores concordam ou discordam do que está aparecendo na tela. Os dois estão tão intimamente comprometidos no código que está aparecendo que a maioria da comunicação é não verbal. A linguagem corporal tem uma função importante. Um parceiro pode dizer quando o seu parceiro está desconfortável com o código, embora nenhuma palavra seja dita. Um trejeito, um suspiro, um tique nervoso—tudo conspira para aumentar a largura da banda de comunicação entre os parceiros.

Às vezes um parceiro pegará o mouse enquanto o outro opera o teclado. O controlador do mouse controla o local dentro do módulo em que o trabalho será realizado. O controlador do teclado controla o conteúdo que é alterado ou incluído nesse local. Em outras vezes, um parceiro estará digitando e o outro irá prever uma chamada de função a caminho e abrirá os documentos API para a página certa enquanto o codificador precisar da especificação.

Quando um parceiro fica cansado, o outro pode tomar a frente, premitindo que o seu parceiro ou parceira descanse exercendo a função do observador. Em outras vezes, ambos os parceiros terão muita energia e trocarão o teclado e o mouse freqüentemente.

Em resumo, há poucas regras e poucos procedimentos. A única restrição real é que ambas as partes têm que permanecer comprometidas e a comunicação entre elas deve ser intensa. Um par em que um parceiro está digitando e o outro está olhando pela janela não é um emparelhamento verdadeiro.

O que o desenvolvedor solitário pode fazer?

Você não pode estar aos pares todo o tempo. Alguns projetos—ou seja, os que adotam a *Programação eXtrema* (XP) (consulte o processo [1]) de referência—siga a regra que os pares devem produzir todo o código de produção. Nesse caso, quando você não está emparelhando você pode verificar seu e-mail, ler uma nova técnica ou API, ler o código com o qual você não está familiarizado ou conversar com os investidores sobre a iteração atual ou os planos futuros. Sem dúvida, há sempre algo lucrativo que um desenvolvedor pode encontrar para fazer nas horas em que um parceiro não pode ser encontrado.

Alguns projetos são menos rígidos em relação ao emparelhamento. Alguns permitirão que os desenvolvedores únicos gravem os testes. Outros permitem que os desenvolvedores únicos gravem classes abstratas ou interfaces. Outros ainda simplesmente

permitem que os desenvolvedores decidam quando é o melhor momento de trabalhar em pares. No entanto, uma coisa é clara—estudos mostraram que as taxas de defeitos caem dramaticamente quando o emparelhamento é praticado.

Algumas pessoas não gostam do emparelhamento

Algumas pessoas se sentem desconfortáveis com o conceito de emparelhamento. Em nossa experiência, aqueles que realmente tentarem por uma semana perceberão que o desconforto evapora e que o emparelhamento pode ser aproveitado e útil. Muito poucos continuam a não gostar da prática. Então, para a maioria das pessoas, é simplesmente uma questão de tentar e se acostumar. Para aqueles que fazem uma tentativa honesta de ainda percebem que não gostam da prática, a equipe terá que encontrar algo apropriado para eles fazerem.

Mobília, Dependências e Logísticas

Local do monitor e do teclado

O local da mobília é criticamente importante para o emparelhamento bem-sucedido. Uma pessoa não pode se emparelhar bem se os parceiros não conseguirem sentar próximos um do outro e trocar o teclado rapidamente. A regra é: você tem que conseguir manipular o teclado e o mouse para frente e para trás sem mudar de lugar.

A melhor organização é normalmente uma mesa longa e lisa. Coloque o monitor no meio e coloque duas cadeiras de frente para o monitor. Sentem-se com o monitor entre vocês. Certifiquem-se de que é fácil deslizar o teclado e o mouse para frente e para trás entre vocês. Certifique-se também de que enquanto você tem o teclado, você está confortável e em posição ereta. Certifique-se de que o monitor está visível para ambos os parceiros sem ter que rodá-lo.

Cercado

Para facilitar a alteração de parceiros, é freqüentemente inteligente trabalhar em uma organização cercada. Coloque várias estações de pares em uma sala. Utilize cadeiras com rodinhas e linóleo ou assoalho. Organize as estações de trabalho para que os pares fiquem de frente um para o outro. A meta aqui é aumentar a comunicação. Às vezes, as comunicações mais importantes são as que são descobertas por acaso. Desejamos aumentar as chances para que essas comunicações aconteçam.

Nos cantos

Atualmente muitos cubículos têm estações de trabalho colocadas nos cantos. O desenvolvedor senta de frente para um canto do cubículo com um monitor em frente. Enquanto isso é conveniente para o trabalho individual, é quase impossível emparelhar bem nesse ambiente. Se você tem cubículos com estações de trabalho nos cantos, coloque as estações em pares em algum outro lugar—talvez em uma sala de conferência. O) emparelhamento em uma mesa de sala de conferência utilizando um laptop normalmente é muito eficiente.

Problemas e Preocupações

O emparelhamento divide a produtividade em dois

Parece que duas pessoas trabalhando juntas em uma tarefa consumirão duas vezes mais horas por pessoa do que uma pessoa trabalhando na mesma tarefa. Embora isso seja razoável, não parece ser esse o caso. Estudos independentes (consulte a referência [2]) mostraram que pouca ou nenhuma produtividade é perdida no trabalho em pares. Esses mesmos estudos mostram que o emparelhamento diminui substancialmente a taxa de defeitos *e o tamanho do código*, enquanto aumenta muito prazer no trabalho.

Disputas entre parceiros

O proprietário da tarefa tem a palavra final em todas as disputas de design, mas a melhor maneira de definir u8ma disputa pe tentar ambas as idéias e escolher a que funciona melhor.

Especialistas

A sabedoria convencional sugere que os desenvolvedores que se especializam em uma área específica, como bancos de dados ou GUIs, devem aplicar seus esforços somente nessas áreas. Em um ambiente de programação aos pares, entretanto, esses especialistas se tornam mentores para outros que não compartilham sua especialidade. Os desenvolvedores podem receber

tarefas de fora de sua especialidade e, em seguida, recrutar ajuda dos especialistas. Desse modo, o conhecimento tende a ser difundido pela equipe do projeto, reduzindo bastante a sensibilidade de modificação do projeto.

Barulho

Um par fará barulho ao programar. Um cercado cheio de pares manterá um barulho contínuo em volume baixo. Alguns acham que esse barulho será peturbador e causará distração. Isso não tem provado ser um problema significante. Se você achar que o barulho o está distraindo, vá para uma sala de conferência por um tempo.

Caubóis

Muitas equipes acham que são proprietários orgulhosos de um ou mais codificadores de caubóis. Esses são os programadores que pegam o trabalho mais rapidamente que qualquer outra pessoa, não conseguem trabalhar com os outros e ninguém mais tem permissão (ou conseguiria, se permitido) ler o código. A melhor coisa a fazer com esses desenvolvedores é movê-los para outro projeto ou para uma função em que não estejam escrevendo códigos de produção. Talvez eles possam escrever ferramentas de vida curta ou fazer algum teste de tortura maníaca.

Impedimentos físicos ou de estilo

Algumas pessoas utilizam o teclado QWERTY. Outras preferem DVORAK. Algumas precisam de teclados, mouses, vídeos, pedais especiais entre outras coisas. Algumas não conseguem programar sem fones de ouvido e música alta. Outros têm que se cercar de pacotes de salgadinhos vazios. Alguns gostam de emacs. Outros gostam de VI. Outros querem trabalhar no WordPad.

Sem dúvida, os impedimentos que podem ser nomeados são inúmeros. Mas cada um pode ser resolvido com um pouco de reflexão e a habilidade de comprometimento. Uma equipe que permite que essas coisas os impeça é uma equipe que simplesmente não será bem-sucedida em qualquer coisa que tentar.

Como a equipe pode planejar o emparelhamento?

Não acreditamos que as tarefas devam ser designadas para os pares. Em vez disso, cada desenvolvedor deve ser responsável por um conjunto de tarefas. Gostamos que os pares sejam formados informalmente. Cada desenvolvedor, preseguindo suas próprias responsabilidades, pede a outro desenvolvedor para ajudar rapidamente. O proprietário da tarefa mantém essa propriedade e contabilidade. Os parceiros são apenas ajudantes.

Cada desenvolvedor deve levar em consideração a quantidade de tempo em que ele ou ela estará em pares ao aferecer as estimativas para as tarefas.

Conclusão

A programação aos pares é uma alternativa bem testada, bem aceita para as revisões de código. Mais do que isso, é uma maneira fundamentalmente diferente de escrever o software. Os benefícios vão além da produtividade e qualidade e afetam a eficiência e a moral da equipe.

Referências

[1] *eXtreme Programming eXplained*, Kent Beck, Addison Wesley, 2000.

[2] *Strengthening the Case for Pair Programming*, Laurie Williams, Universidade de Utah, Julho/Agosto 2000 IEEE Software.

Rational®
the software development company

Duas Sedes:

Rational Software
18880 Homestead Road
Cupertino, CA 95014
Tel: (408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
Tel: (781) 676-2400

Sem custo: (800) 728-1212
E-mail: info@rational.com
Web: www.rational.com
Localização Internacional:www.rational.com/worldwide