Principes et conseils RUP/XP : Programmation par paire

Robert C. Martin
Object Mentor, Inc.

Livre blanc de Rational Software

TP 158, 3/01

Rational®
the software development company

Table des matières

Présentation

Travail par paire, brève description

La programmation par paire est une technique dans laquelle le logiciel d'un projet est écrit par des paires de programmeurs. Chaque paire travaille ensemble sur le même poste de travail. L'un des deux commande le poste de travail, c'est le pilote, pendant que l'autre regarde, observant attentivement le code produit. Le pilote pense de manière tactique, préoccupé par la ligne de code qu'il est en train de saisir. L'observateur valide la syntaxe et pense de manière stratégique à la totalité du programme. Ils échangent fréquemment leurs rôles et le code résultant est écrit plus vite que par une seule personne et avec moins d'erreurs. De plus, le code est étroitement connu par au moins deux développeurs.

Cas de travail par paire

Examinons une session de révision de code standard. Un module requérant huit heures de développement pour une personne est révisé en une heure par huit personnes. Le résultat net est que 16 heures pour une personne sont passées sur le module. Cependant, les réviseurs ne peuvent pas consacrer le temps nécessaire pour se familiariser avec le code et leur révision est relativement superficielle. Un seul développeur est infiniment familier du code, mais peut-être trop pour trouver la masse d'erreurs.

Comparons ceci avec la pratique de programmation par paire. Si le module requiert huit heures pour être développé par la paire, un total de 16 heures pour une personne est dépensé. Cependant, dans ce cas, *deux* développeurs ont une connaissance étroite du code. Les erreurs oubliées par un développeur, sont flagrantes pour l'autre.

Le cas de la programmation par paire est simple, mais les répercussions sont subtiles et d'une portée considérable. La programmation par paire est simplement un moyen bien plus efficace pour écrire et réviser le code. Avec deux personnes connaissant parfaitement un module, beaucoup moins d'erreurs sont introduites dans le code. Ce dernier est mieux structuré et deux cerveaux en ont une parfaite connaissance. Si cela n'en était que les seuls avantages, ce serait suffisant, mais le fait de travailler à deux fournit bien d'autres avantages.

Les paires sont plus courageuses. Ce qu'un seul programmeur peut avoir peur de tenter, une paire aura le courage de le faire et les capacités de l'évaluer.

Il favorise le travail d'équipe. Comme les modules ne sont pas écrits par une seule personne, le code devient la propriété de l'équipe, et non d'un développeur en particulier.

Le travail par paire augmente l'étendue des connaissances. Plus le nombre de développeurs travaillant par paire est important, plus la connaissance du système se diffuse à toute l'équipe. Il en résulte une équipe dont les membres sont familiarisés avec tout le système, plutôt que chaque membre connaissant uniquement sa propre partie.

Il favorise la productivité. Une personne programmant seule passe par des phases d'explosion d'énergie, suivies de périodes d'inactivité relative. Les paires se stimulent. Lorsque l'un des deux est fatigué, ils échangent leur rôle. Ils parviennent ainsi à maintenir l'intensité activée bien plus longtemps qu'une seule personne ne peut généralement supporter.

Le travail par paire est divertissant. Travailler avec un autre développeur revêt un caractère éducatif, stimulant et amusant. Il augmente la satisfaction du travail et le moral général.

En pratique

Travail par paire

Le travail par paire commence lorsque le développeur responsable d'une tâche demande à une autre personne de l'aider. La règle est la suivante : si l'on vous demande de l'aide, vous ne pouvez pas refuser. Cela ne signifie pas que vous devez immédiatement arrêter ce que vous êtes en train de faire. Au contraire, cela signifie que vous devez négocier le moment durant lequel vous pouvez offrir cette aide et un autre où vous pourrez obtenir de l'aide en retour.

Le partenaire de la paire n'assume pas la responsabilité de la tâche. Cette responsabilité incombe au propriétaire de la tâche. Le partenaire ne s'engage pas à demeurer avec le propriétaire jusqu'à ce que la tâche soit terminée. Il ne s'engage qu'à aider.

Un membre de la paire devient le pilote, celui qui contrôle le poste de travail, tandis que l'autre regarde. Le pilote saisit le code, exécute le programme de compilation, exécute les tests, etc. L'observateur examine chaque séquence de touches, chaque commande, chaque résultat de test et offre son aide et ses suggestions. Les deux parties sont engagées à tout instant.

Parfois, le pilote sait mieux ce qu'il faut faire, et l'observateur ne fait que suivre. En revanche, il arrive également que l'observateur dicte au pilote ce qu'il faut faire. Le pilote se sent alors parfois frustré, passant le clavier à l'observateur, ils échangent ainsi leurs rôles. Il arrive aussi que l'observateur demande le clavier et qu'ils échangent leurs rôles. Ceci se produit de nombreuses fois au cours d'une session de travail par paire.

Changement de paires

Les partenaires de paires ne sont pas définis à long terme. Une session typique de travail par paire dure environ une demi-journée. Pour une raison ou une autre, un partenaire peut décider de quitter la paire. Dans ce cas, le propriétaire de la tâche doit trouver un autre partenaire. Ceci peut signifier qu'il est temps pour le propriétaire de la tâche de rendre la pareille à quelqu'un qui s'est associé à lui la semaine précédente. D'un autre côté, peut-être doit-il demander à quelqu'un ayant la bonne expérience de l'aider pour un problème particulièrement tenace.

Ce changement de paires entraîne la diffusion des connaissances du système dans toute l'équipe de développement. En un temps record, chaque membre de l'équipe aura passé du temps à travailler dans presque tous les domaines du système. Ceci réduit considérablement la sensibilité du projet à la rotation et rend chaque programmeur plus confiant avec tout le système.

Propriété collective

Comme chacun travaille sur les différents modules du système, personne n'en détient un en particulier. Ceci signifie que la responsabilité du système n'est pas divisée sur une base module par module. En revanche, l'équipe toute entière est collectivement responsable du système dans sa totalité. Tout membre de l'équipe peut extraire et modifier un module du système pour quelque raison que ce soit. Si une paire modifie le module X et que cette modification entraîne l'échec des tests de l'unité Y, la paire doit réparer le module Y.

Stimulation et collaboration

La programmation par paire est une forme de communication très intense. Le dialogue échangé est souvent bref et un observateur extérieur risque d'avoir des difficultés à en saisir le sens. En tant qu'observateur, il se peut que vous entendiez la paire prononcer des mots singuliers, tels que “point-virgule” ou “parenthèse fermante”. Ou bien il se peut que vous entendiez simplement des grognements moins articulés signifiant que les programmeurs sont ou ne sont pas d'accord sur ce qui s'affiche à l'écran. Tous deux sont si étroitement liés au code qui s'affiche qu'une bonne partie de la communication n'est pas verbale. Le langage du corps jour un rôle important. L'un des deux partenaires peut faire comprendre que son homologue ne maîtrise pas le code, sans même dire un mot. Une grimace, un soupir, un geste nerveux, tout concourt à augmenter la bande passante de communication entre les partenaires.

Parfois, il arrive que l'un des partenaires saisisse la souris, pendant que l'autre est au clavier. Celui qui tient la souris contrôle l'emplacement où aura lieu le travail dans le module. Celui qui est au clavier contrôle le contenu qui est modifié ou ajouté à cet endroit. Parfois, l'un des partenaires est en train de saisir au clavier et l'autre prévoit un appel de fonction et ouvre les documents d'API à la bonne page, juste au moment où le codeur a besoin de la spécification.

Si l'un des partenaires se sent fatigué, l'autre prend le relais, lui permettant ainsi de se reposer en jouant le rôle de l'observateur. Il arrive aussi que les deux partenaires soient remplis d'énergie et échangent le clavier et la souris fréquemment.

En résumé, peu de règles et encore moins de procédures. La seule contrainte réelle est que les deux parties doivent rester liées et que la communication entre les deux doit être intense. Une paire dans laquelle l'un des partenaires saisit au clavier et l'autre regarde par la fenêtre n'est pas vraiment une paire.

Que peut faire un développeur seul ?

Vous ne pouvez pas travailler à deux tout le temps. Certains projets, comme ceux qui adoptent le processus de *programmation eXtreme* (XP) (voir référence [1]) suivent la règle qui dicte que des paires doivent produire tout le code de production. Dans ce cas, si vous ne travaillez pas par paire, vous pouvez consulter votre messagerie, vous informer sur une nouvelle technique ou API, prendre connaissance d'un code qui ne vous est pas familier ou discuter avec les décisionnaires de l'itération en cours ou de plans futurs. Bien sûr, il y a toujours quelque chose de bénéfique à faire pour un développeur qui ne trouve pas de partenaire pendant quelques heures.

Certains projets sont moins stricts sur le travail par paire. Il arrive que des développeurs soient autorisés à écrire des tests seuls. D'autres permettent à des développeurs d'écrire seuls des classes abstraites ou des interfaces. D'autres encore permettent tout simplement aux développeurs de décider quand il est préférable de travailler par paire. Une chose est claire, cependant, des études ont montré que le taux d'erreurs chute considérablement lors du développement par paire.

Certaines personnes n'aiment pas travailler à deux

Certaines personnes n'apprécient pas le concept du travail par paire. L'expérience a montré que ceux qui le tente pendant une semaine ou plus, voient leur malaise disparaître et qu'ils apprécient le travail par paire et le trouve utile. Très peu d'entre eux continuent à ne pas apprécier cette pratique. Ainsi, pour la plupart des gens, c'est simplement une question d'essayer et de s'y habituer. Pour ceux qui font une tentative honnête et persistent à ne pas apprécier la pratique, l'équipe devra leur trouver quelque chose d'approprié à faire.

Meubles, locaux et logistique

Emplacement de l'écran et du clavier

L'emplacement des meubles est considérablement important pour un travail par paire efficace. Une paire ne peut pas fonctionner correctement si les partenaires ne peuvent pas s'asseoir l'un à côté de l'autre et échanger rapidement la position du clavier. La règle est la suivante : vous devez pouvoir échanger le clavier et la souris sans changer de sièges.

Le meilleur agencement est généralement une simple table longue. Placez l'écran au milieu et deux chaises en face. Installez l'écran entre vous. Assurez-vous qu'il est possible de faire glisser le clavier et la souris d'une place à l'autre. Vérifiez également que lorsque vous avez le clavier, vous êtes à l'aise et bien assis. Assurez-vous que l'écran est visible par les deux partenaires sans qu'ils doivent le faire pivoter.

Bullpen

Pour faciliter le changement des partenaires dans les paires, il est souvent utile de travailler selon un arrangement "bullpen". Installez plusieurs postes de travail paires dans une seule pièce. Utilisez des chaises à roulettes et des sols carrelés ou revêtus de linoléum. Disposez les postes de travail de telle sorte que les paires se fassent face. L'objectif est d'augmenter la communication. Parfois, la communication la plus importante est celle qui est lancée au hasard. Nous voulons augmenter les chances qu'une telle communication ait lieu.

Angles

Aujourd'hui, de nombreux locaux cubiques présentent des postes de travail dans les angles de pièces. Le développeur est assis face à un angle de la pièce avec un écran en face de lui. Ceci est parfaitement adapté au travail individuel, mais il est quasiment impossible d'effectuer un bon travail par paire dans cet environnement. Si vos locaux sont cubiques avec des postes de travail dans les angles, configurez alors ailleurs les postes de travail pour paires, dans une salle de conférence par exemple. Le travail par paire sur une table de salle de conférence avec un ordinateur portable est souvent très efficace.

Questions et problèmes

Le travail par paire réduit la productivité de moitié

Il va sans dire que deux personnes travaillant ensemble sur une tâche passent le double de temps par rapport à une seule personne travaillant sur la même tâche. Aussi raisonnable que cela puisse être, il ne semble pas que ce soit le cas. Des études (voir référence [2]) ont montré que très peu de productivité, voire aucune, est perdue lors du travail par paire. Ces mêmes études montrent que le travail par paire réduit le taux d'erreurs *et la taille du code*, tout en augmentant considérablement la satisfaction au travail.

Conflits entre partenaires d'une paire

Le propriétaire de la tâche a le dernier mot dans toutes les discussions relatives à la conception, mais le meilleur moyen de régler un conflit est d'essayer les deux solutions et de choisir la meilleure.

Spécialistes

La sagesse suggère que les développeurs qui se spécialisent dans un domaine particulier, comme les bases de données ou les interfaces utilisateur, appliquent leurs efforts individuellement dans ces domaines. Dans un environnement de programmation par paire cependant, ces spécialistes deviennent des mentors pour ceux qui ne partagent pas leur spécialité. Les développeurs peuvent s'engager dans des tâches ne faisant pas partie de leur spécialité, puis chercher de l'aide auprès de

spécialistes. De cette façon, la connaissance des spécialités tend à se diffuser à toute l'équipe projet, réduisant les failles du projet dues à la rotation du personnel.

Bruit

Des développeurs travaillant par paire font du bruit lors de la programmation. Dans un local configuré en "bullpen" rempli de paires de programmeurs règne un constant bourdonnement de faible niveau. Certains pensent que ce bruit est gênant. Mais, ceci ne s'est pas révélé être un problème important. Si vous trouvez le bruit gênant, vous pouvez aller vous installer dans une salle de conférence pour quelque temps.

Cowboys

De nombreuses équipes pensent qu'elles sont les dignes détentrices d'un ou deux codeurs "cowboy". Ce sont des programmeurs qui réalisent le travail plus rapidement que quiconque, mais ne savent pas travailler avec les autres. Personne n'est autorisé à lire leur code (ou serait dans l'incapacité de le faire, s'il y était autorisé). La meilleure chose à faire avec ce type de développeurs est de les exclure du projet ou de leur donner un rôle dans lequel il n'écrive pas le code de production. Ils peuvent par exemple, écrire des outils temporaires ou pratiquer des tests intensifs.

Obstacles physiques et défauts de style

Certains utilisent des claviers QWERTY, d'autres préfèrent DVORAK. D'autres encore requièrent des claviers, des souris, des écrans, des pédaliers spéciaux, etc. Certains ne peuvent pas programmer sans casque, ni musique forte. D'autres doivent s'entourer d'emballages de Twinkie vides. Certains préfèrent les emacs. D'autres aiment l'éditeur VI. Et d'autres encore veulent travailler dans WordPad.

Les obstacles pouvant être cités sont innombrables. Mais chacun d'eux peut être résolu avec un peu de bonne volonté et de compromis. Une équipe qui laisse de telles choses l'entraver, est une équipe qui échouera simplement dans toutes ses entreprises.

Comment l'équipe peut-elle planifier le travail par paire ?

Nous ne pensons pas que des tâches doivent être affectées à des paires. En revanche, chaque développeur doit être responsable d'un ensemble de tâches. Nous préférons que les paires se forment de manière informelle. Chaque développeur, endossant ses propres responsabilités, demande une aide rapide aux autres développeurs. Le propriétaire de la tâche conserve la propriété et la responsabilité. Les partenaires des paires ne font qu'aider.

Chaque développeur doit tenir compte de la durée du travail par paire lors des estimations de tâches.

Conclusion

La programmation par paire est une solution largement testée et bien acceptée pour les révisions de code. Plus encore, il s'agit d'un moyen fondamentalement différent d'écriture de logiciel. Les avantages vont bien au-delà de la productivité et de la qualité et ont un effet sur la solidité et le moral de l'équipe.

Références

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

[2] *Strengthening the Case for Pair Programming*, Laurie Williams, Université de l'Utah, juillet/août 2000 IEEE Software.

Rational
the software development company

Sièges :

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

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

Appel gratuit : (800) 728-1212
Adresse électronique : info@rational.com
Site Web : www.rational.com
Sites internationaux : www.rational.com/worldwide