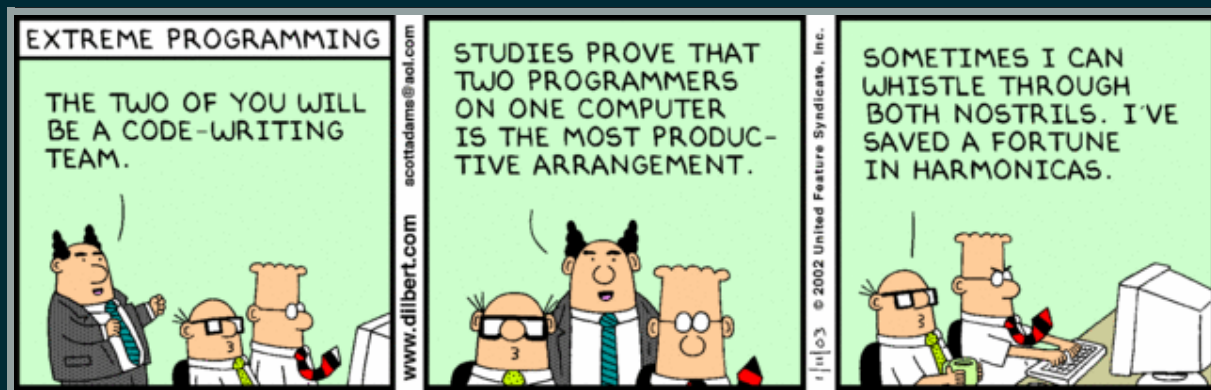


'IL Y AVAIT VRAIMENT DU GÂTEAU':

PAIR-PROGRAMMING UN RETOUR D'EXPÉRIENCE



PLAN

- qu'est-ce que le pair programming?
- notre expérience
- la littérature



Andreas Kundig @andreaskundig
David Hodgetts @David_Hodgetts

**QU'EST-CE QUE LE PAIR
PROGRAMMING?**

DÉFINITION (WIKIPEDIA)

La programmation en binôme (ou pair programming en anglais) est une méthode de travail dans laquelle deux développeurs travaillent ensemble sur la même partie de code, en binôme sur un même poste de travail

- le premier, appelé pilote (driver), a le clavier. C'est lui qui va travailler sur la portion de code à écrire.
- le second, appelé copilote (partner), est là pour l'aider, en suggérant de nouvelles possibilités ou en décelant d'éventuels problèmes.

Les rôles s'échangent régulièrement pendant la séance de programmation.

2. L'HISTOIRE

Kent Beck popularise la pratique avec les méthodologies de l'Extreme Programming...

...la pratique est néanmoins plus ancienne:

Dans les années 50, **Fred Brooks**, l'auteur de "Mythical man month" évoque un projet qu'il a réalisé en pair à l'époque de ses études: "we produced 1500 lines of defect free code; it ran correctly the first time"

Dans les années 70, **Dick Gabriel**, le concepteur de Common Lisp et "l'inventeur" des patrons de conception évoque des sessions de binomage.

Dans les années 80, **Larry Constantine** parle de "dynamic duos".

Dans les années 90 **James Coplien**, publie dans un papier de recherche de Bell Labs "Developing in pairs": "Some problems are bigger than any one individual"

NOTRE EXPÉRIENCE

GONGFU.IO

Gong-fu I/O

Pour participer Projets

Contact Accueil

Pratiquer ensemble pour apprendre

Un projet d'étude — un ou plusieurs sujets d'intérêt commun — des rencontres régulières — pour perfectionner ensemble, en petit groupe, en apprenant l'un de l'autre et par l'étude, notre pratique du développement logiciel. Partage et plaisir, de débiter à nouveau.

Suivre @gongfuio



Rencontres régulières

Actuellement: 1x par semaine
avec un ordi portable
le jeudi de 17h30 à 19h30
à Unimail Genève

Entre pairs

Olivier Lange · Jacques Couvreur
Yannick Grillet · Nicolas Peru
Ben Lucido · Patrick Lange
Yannis Jaquet · Rudi Farkas
... et vous?

Réalisations en cours

Softshake App

Le programme de la conférence de Softshake 2012 pour iPhone et Android. Etude et exploration de PhoneGap, jQueryMobile, HTML5 et MongoDB.

SweetTweets

Nouveau binôme formé le 12.04.2012. Etude et exploration de l'API de Twitter, de PhoneGap, jQueryMobile, HTML5 et MongoDB, appliquée à la réalisation d'une application de filtrage et de classification des tweets de JSGenève.

Tous les projets »

Montrer la progression

Le code source des projets d'étude est déposé en l'état sur github.com/gongfuio

Nous y avons créé une organisation, plutôt qu'un compte utilisateur, qui permet de former des équipes et chaque participant peut contribuer publiquement aux projets.

Code source »

EXQUIS

"Cadavre exquis" pour des animations javascript.

Vanilla js + canvas + un peu de node.js

Fonctions et structures de données plutôt que de l'orienté objet.

CADRE DE L'EXPÉRIENCE

- Projet personnel: pas de délai, pas d'obligation de résultat.
- Deux heures par semaine.
- Dans un bar le plus souvent....
- ...avec un petit laptop.
- vim, puis sublime, finalement emacs -> apprendre à faire des compromis
- En remote pour quelques mois.

BILAN

cerveau + cerveau = ?



ASPECTS POSITIFS

- Plus de patience, discipline, concentration
- Live code review
- Bugs plus faciles à trouver
- Conception (algorithmes, architecture)
- Transmission de savoir et exercice de communication
- Apprendre à deux
- Motivation++

ASPECTS NÉGATIFS



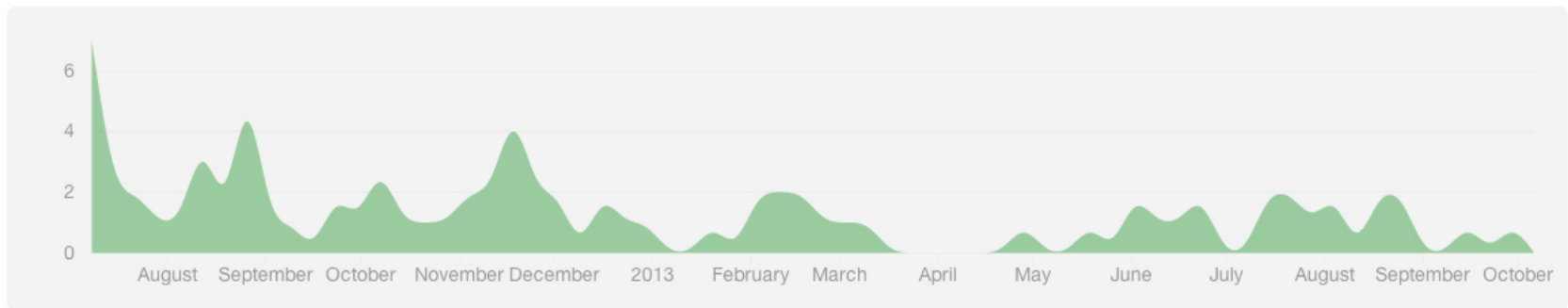
ASPECTS NÉGATIFS

- Les bonnes pratiques ne sont pas automatiquement au rendez-vous
- Toutes les tâches ne se prêtent pas au pair programming
- Lâcher son partenaire en allant trop vite (in the zone alone)
- Demande plus d'énergie
- Triple programming?
- Un seul committeur

July 8th 2012 - October 6th 2013

Commits to master, excluding merge commits

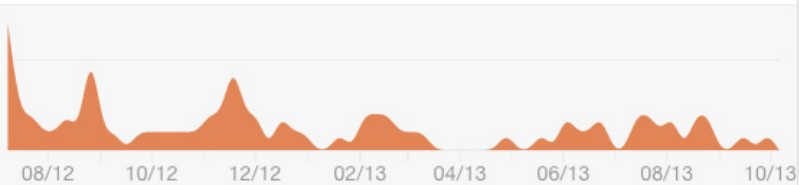
Contribution Type: **Commits** ▾



david-demainlalune

#1

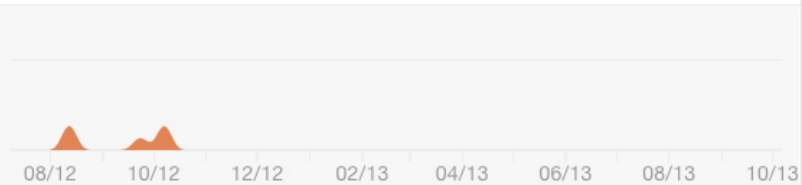
78 commits / 63,467 ++ / 60,717 --



andreaskundig

#2

5 commits / 25 ++ / 26 --



L'EXPÉRIENCE DU REMOTE PAIR PROGRAMMING

- Skype (voix uniquement)
- Teamviewer
- Et le bilan est très positif



OPTIONS TECHNIQUES (CODING)

Screen sharing (teamviewer, skype, etc.)

Terminal sharing (screen, tmux)

Funky cloud stuff (floobits, screenhero.com, cloud 9)

LITTÉRATURE

'PAIR PROGRAMMING: WHAT'S IN IT FOR ME?' (MICROSOFT RESEARCH 2008)

487 sondés

106 ont déjà pratiqué le pair programming (21.7%)

64.4% pense que la pratique fonctionne pour eux

62.8% pense que la pratique fonctionne pour leur partenaire

48.2% estime que la pratique est bénéfique pour l'équipe

39.2% estime que la pratique est bénéfique pour l'organisation.

RÉACTIONS POSITIVES

“greater understanding of a larger codebase across the team.”

“higher quality code in terms of consistency with guidelines.”

“quickly ramp-up new members,”

enables “users to learn new techniques faster.”

“every-one learns constantly from each other.”

LE TOP TEN DES RÉACTIONS POSITIVES

1. Fewer Bugs
2. Spreads Code Understanding
3. Higher Quality Code
4. Can Learn from Partner
5. Better Design
6. Constant Code Reviews
7. Two Heads are Better than One
8. Creativity and Brainstorming
9. Better Testing and Debugging
10. Improved Morale

RÉACTIONS NÉGATIVES

“if I have a choice, I can employ one star programmer instead of two programmers who need to code in a pair.”

Pairing “reduces the freedom of work hours of individual contributors.” (scheduling)

“Many partnerships fail due to personality conflicts,” (personality clash) - “pairs get sick of each other.”

“Sometimes we waste time on discussion,”

“tends to drag the faster/smarter/better person down.”

“if the partners” abilities are imbalanced, it could be that one partner become obsolete in the process.”

LE TOP TEN DES RÉACTIONS NÉGATIVES

1. Cost efficiency
2. Scheduling
3. Personality clash
4. Disagreements
5. Skill differences
6. Programming style differences
7. Hard to find a partner
8. Personal style differences
9. Distractions
10. Misanthropy
10. Bad Communication
10. Metrics/Hard to Reward Talent

LES QUALITÉS DU BON PARTENAIRE

“usually looks at things from a different angle.”

“able to think ... even [from] a different role (e.g. test and dev together).”

“blocks on different things than I do,”

“some-one who complements my thinking and skills in terms of technical and design skills.”

“willing to cooperate and step away from the PC with me to work on design or other details.”

“able to adapt to different working styles.”

“good listener,”

“articulate,”

“enjoy debating and discussing code,”

“sense of humor,”

“comfortable with people around them.”

LE TOP TEN DES QUALITÉS DU BON PARTENAIRE

1. Complementary Skills
2. Flexibility
3. Good Communications
4. Smart
4. Personable
6. At Least the Same Skills as Me
7. Strong programmer
7. Better Skills than Me
7. Able to Focus
10. Knowledgeable

LES QUALITÉS D'UN TEAM QUI PRATIQUE LE PP

“Communication, communication, communication.” (this is MS after all ;))

An ideal team “consists of easy-going people who want to listen and share ideas with each other.”

“A little diversity here is good.”

“cooperative personalities, they work well together, rather than trying to compete with one another.”

“permissive to mistakes.”

LE TOP TEN DES QUALITÉS D'UN TEAM QUI PRATIQUE LE PP

1. Good Communications
2. Complementary Skills
3. Compatible Personalities
4. Team Works Effectively
5. No Ego
6. Fast and Efficient
7. Flexibility
8. Common Goals
8. Good Quality
10. Same programming skills
10. Collaborative
10. Work Well Together

'PAIR PROGRAMMING ILLUMINATED', LAURIE WILLIAMS, 2002

ON PEUT AUSSI FAIRE UN EFFORT

Take breaks

Practice humility

Be confident / be receptive

Communicate

Listen

Be a team player

Hone the balance between compromise and standing firm

'MAKING SOFTWARE: WHAT REALLY WORKS, AND WHY WE BELIEVE IT', ANDY ORAM, GREG WILSON, 2010

A RETENIR

"suitable length for pairing session 1.5h to 4h (pairing can be mentally exhausting)"

"frequent pair rotation is beneficial for knowledge spread"

"rotating roles (driver, navigator) is important to keep both partners engaged."

"possession of keyboard as a subtle effect on decision making, with the driver often being the final decision maker"

"frequent switching of the keyboard helps keep both partners engaged."

"PP must be structured and organised (proclaimed hours for PP)

(space layout, large desk, large screen, wireless mouse and kb, whiteboards, and isolation, pairs make noise)"

tête à tête (Pivotal labs)





OUI MAIS LES GAINS DU BINÔMAGE, ÇA SE MESURE?

- Une étude examine deux projets dans l'industrie. Dans le premier, le pair programming n'a pas d'effet sur le nombre de bugs. Dans le second, le pair programming produit six fois moins de bugs.
- Une autre étude mesure une réduction de 39% du nombre de bugs.
- D'autres encore montrent des gains certains dans les phases de Design, et le travail sur les parties complexes, mais pas pour le travail plus simple.
- ???

LAURIE WILLIAMS 1999

- 39 étudiants avec une certaine expérience de la programmation, répartis en 14 binômes et 13 solos.
- 4 travaux pratiques à réaliser.
- Les binômes passent 15% de tests en plus (90% au lieu de 75%),
donc $0.1/0.25 = 40\%$ de bugs en moins.
- Les binômes rendent toujours à temps, les solos sont parfois en retard.
- Les binômes écrivent 20% de code en moins (+ de cohésion et des classes aux responsabilités mieux définies).
- Les binômes prennent 15% plus de temps (quand on multiplie par deux). Mais ce n'est pas statistiquement significatif.

ESTIMATION DU GAIN DE PRODUCTIVITÉ

Williams combine ses résultats avec des mesures de moyennes tirées d'autres études:

- 22 lignes de code écrites par heures
- 6 bugs pour 1000 lignes
- 1 semaine pour corriger un bug trouvé chez le client

Elle calcule la productivité en lignes de code correctes par heure-personne et obtient:

- 4.3 en solo
- 7.4 en binôme

EN RÉSUMÉ

Gain de qualité du code

Meilleure répartition du savoir

Cycle plus rapide

Plus de motivation et meilleur esprit d'équipe

MALGRÉ CES AVANTAGE, L'ADOPTION RESTE BASSE

Les habitudes du solo programming

Les raisons économiques ("ça coûte le double")

Les difficultés de coordination

LES SOLUTIONS DE WILLIAMS

Les habitudes du solo programming

coding dojo, code retreat, petit projet pilote

LES SOLUTIONS DE WILLIAMS

Les raisons économiques

Commencer petit, gagner une meilleure compréhension de la pratique, projet pilote

LES SOLUTIONS DE WILLIAMS

Les difficultés de coordination

Meeting quotidien pour dynamiquement créer les paires en fonction des tâches du jour

CONCLUSION

RÉFÉRENCES:

- "Pair Programming Illuminated", Laurie Williams, Robert Kessler, 2002
- "Making Software: What Really Works, and Why We Believe It", Andy Oram, Greg Wilson, 2010
- "Pair Programming: What's in it for Me?", Andrew Begel, Nachiappan Nagappan, 2008
- <http://www.pairprogramwith.me/>
- <http://rubyrogues.com/126-rr-remote-pair-programming-with-sam-livingston-gray/>