

# Le Binôme, une pratique Agile

*Que multiplie-t-on vraiment par deux ?....*

N. Charpentier & J. Couvreur

*XP Day – Paris – 2007*



# Scénario

- Qui sommes nous... pour en parler ?
- La théorie ( 2/3 )
  - Des règles pour les développeurs
  - ... Outillage ...
  - Des gains pour les managers
- La pratique ( 1/3 )
- Synthèse et débat ( 1/4 ;o)



# Jacques COUVREUR

[www.2ia.net](http://www.2ia.net)

- **Technologies** (*langages, architectures*)
- **Projets** (*durée, taille, contrat, domaine métier, client*)
- **Agilité**
  - *Test Driven Development (2000...)*
  - *eXtreme Programming (2001...)*
  - *Scrum Master, [Agile-Swiss.org](http://Agile-Swiss.org) (2006)*

- *A votre place l'année dernière...*      [jacques.couvreur@hortis.ch](mailto:jacques.couvreur@hortis.ch)

*Chez*



*cette année !*



# Nicolas CHARPENTIER

[nicolas.charpentier@cellicium.com](mailto:nicolas.charpentier@cellicium.com)

- Langages
  - Script: TCL
  - Objet: C++, Python, Java
  - Fonctionnel: **Erlang**
- Domaines
  - Télécommunication, Ferroviaire, Finance
  - produits et forfaits
- Agilisateur/Extremiste depuis 2001



# QUI...



# QUI...



- **N'a jamais binômé ?**
- **A binomé à l'occasion ?**
- **A fait du binômage intensif ?**

# QUI...



- **Ne veut jamais binômer ?**
- **A envie de binômer un jour ?**
- **Fait tout pour binômer lundi ?**

# QUI...

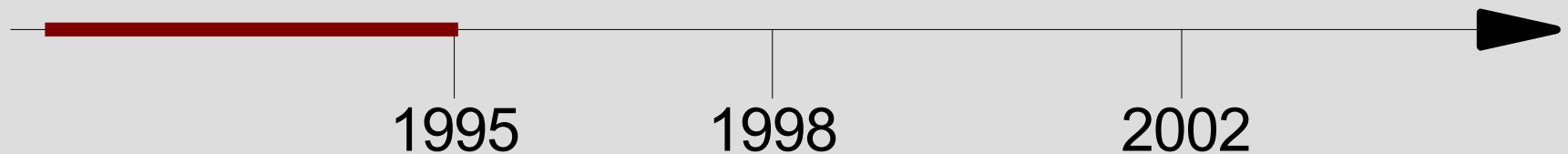


- Ceux qui en ont déjà fait :
  - Quel est votre ressenti ?
  - Pour vous ? Pour le projet ?
- Ceux qui n'en n'ont jamais fait :
  - Comment voyez-vous le binôme ?



# Il était une fois...

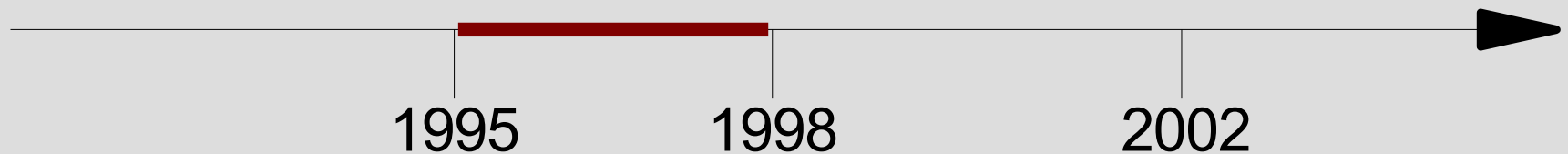
*Laurie Williams*



Avant 1995:

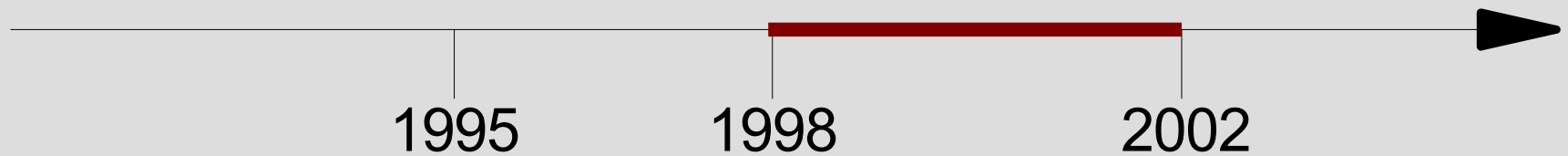
- On en parle,
- On en fait (Fred Brooks: The\_Mythical\_Man-Month)

# Il était une fois...



- 1995: Premières parutions
  - Larry Constantine in “Constantine on Peopleware”
  - Jim Coplien décrit un modèle d'organisation de développement en binôme
  -
- 1998: XP et ses 12 pratiques

# Il était une fois...



- Dès 1998: Maturation du principe
- 
- 2002: “Pair programming illuminated” (L. Williams)

# Pour le développeur



# Qu'est ce qu'un binôme ?



# Qu'est ce qu'un binôme ?

- 2 Développeurs
- 1 bureau
- 1 machine
- 1 portion de code
- Durable...

# Qu'est ce qu'un binôme ?

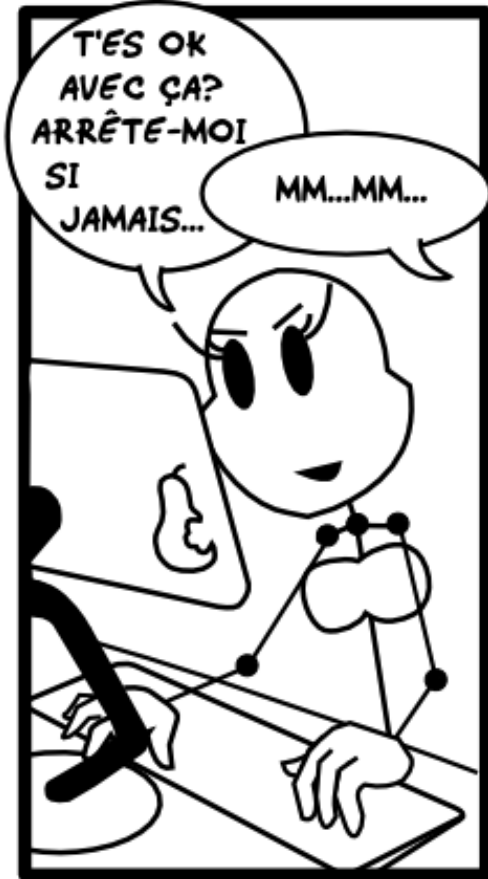
- 2 Développeurs
- 1 bureau
- 1 machine
- 1 portion de code
- Durable...

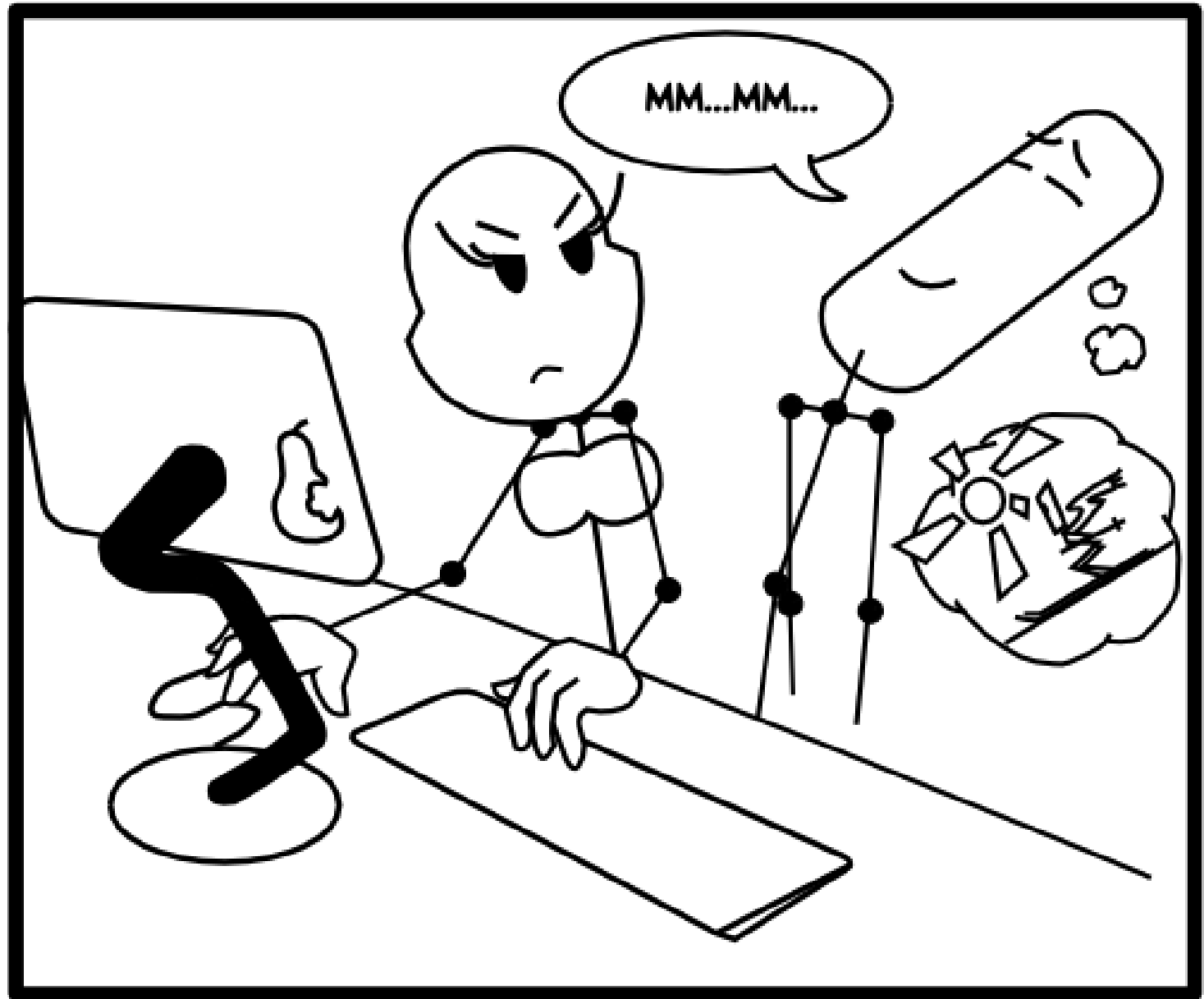
**Des dérives...**

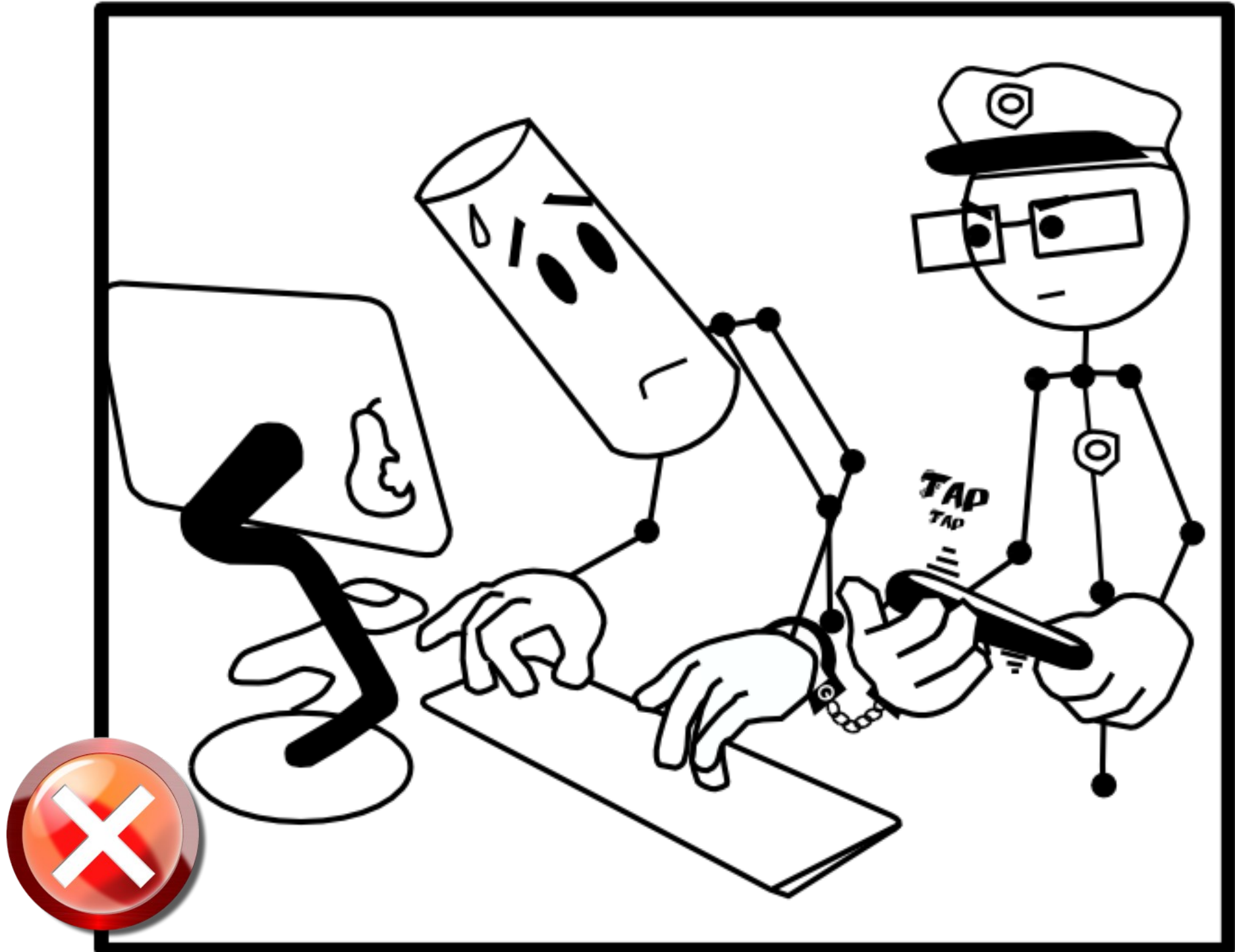
# Le binôme ce n'est pas...







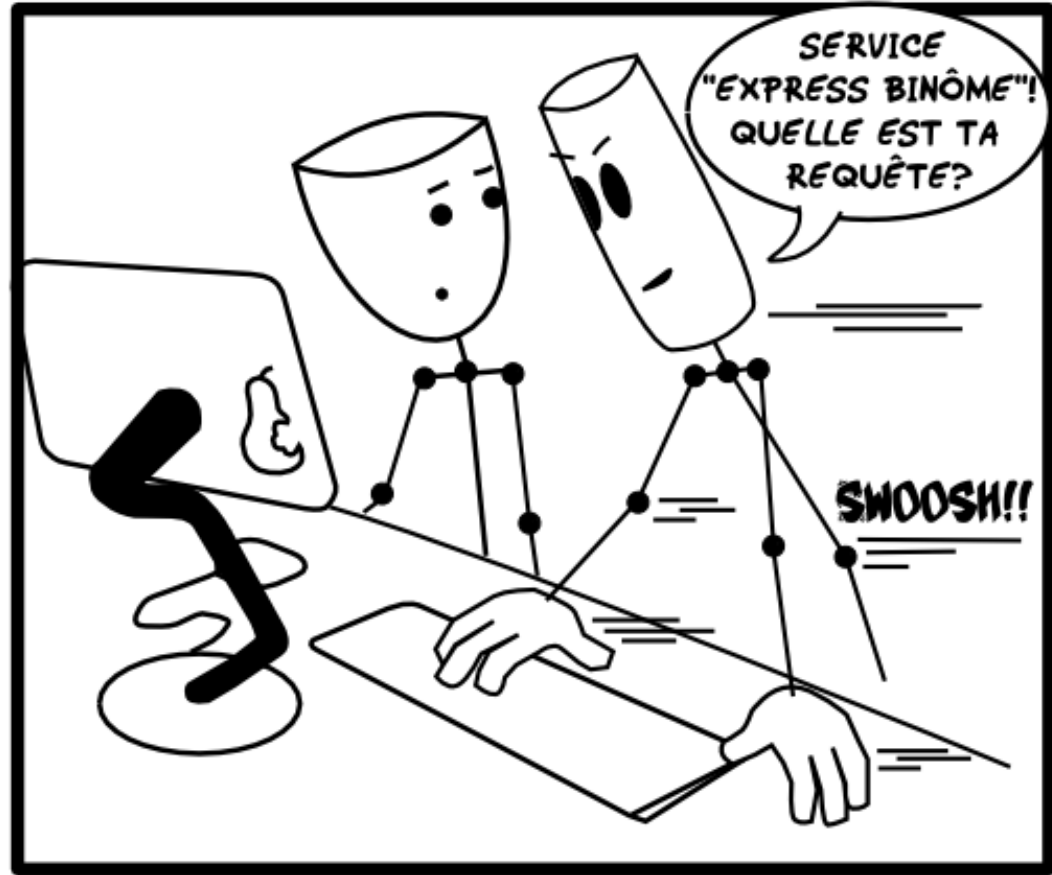


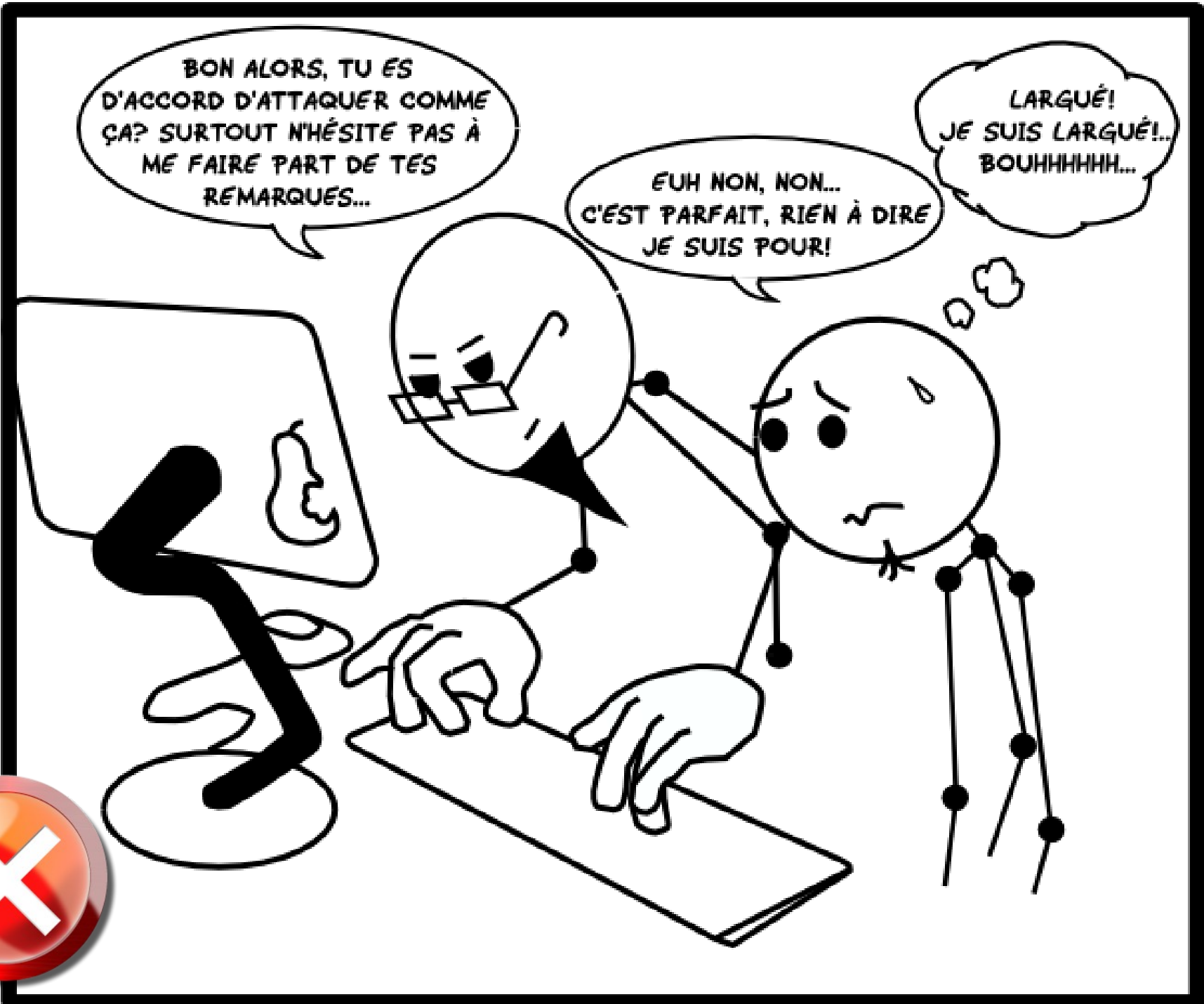


Xcoders J.LAYAT JERRY@AGILE-SWISS.ORG 15/04/07

VOILÀ J'AI FINI  
DE CODER! JE TE LAISSE  
FAIRE LE COMMIT PENDANT  
QUE JE VAIS BOIRE UN CAFÉ!  
JE REPENDRAIS LE CLAVIER  
DANS 5 MINUTES...



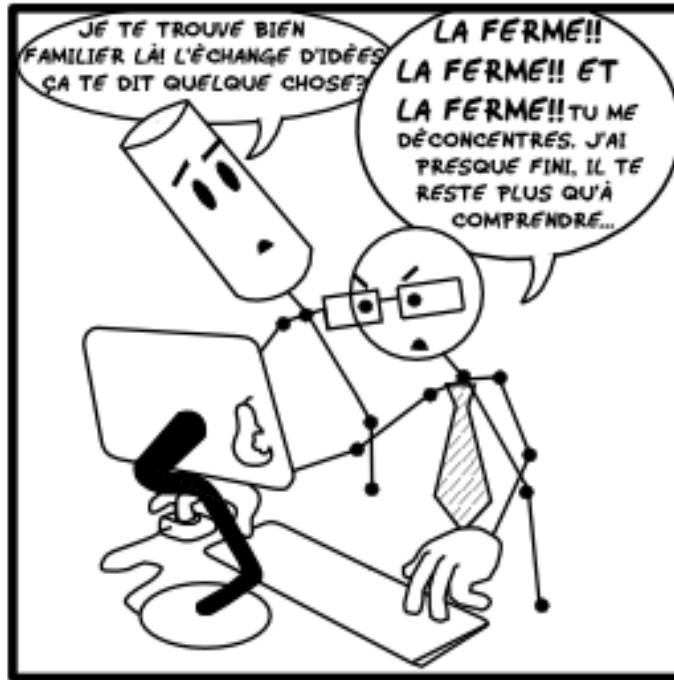




BON ALORS, TU ES D'ACCORD D'ATTAQUER COMME ÇA? SURTOUT NHÉSITE PAS À ME FAIRE PART DE TES REMARQUES...

EUH NON, NON... C'EST PARFAIT, RIEN À DIRE JE SUIS POUR!

LARGUÉ!  
JE SUIS LARGUÉ!..  
BOUHHHHH...



# Les moments de la vie





# Les moments de la vie

- Le développement des tâches
- Et pour cela...
  - La définition des tâches
  - La formation des binômes
  - Le rapport à l'équipe



# Le développement

- Qualité technique
  - Technique de binômage

*Le TDD  
Test  
Driven  
Development*

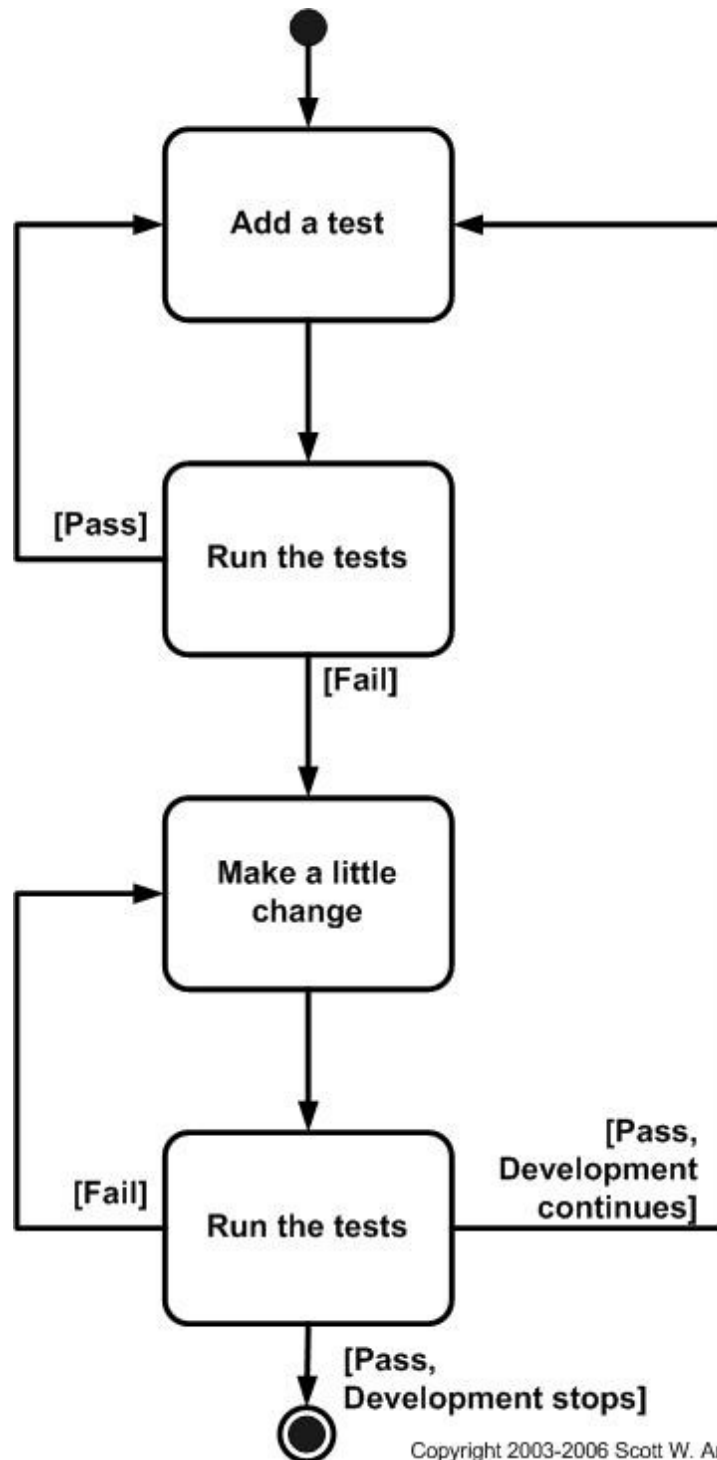
*Ping Pong  
Development*

# Le TDD

*Test*

*Driven*

*Development*

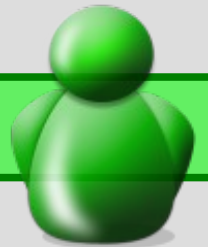


Copyright 2003-2006 Scott W. Ambler



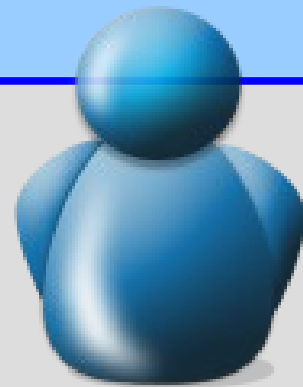
# Développement Ping...

## Le développeur 1 au clavier



1. **Écrire** un test de programmeur
2. **Compiler** le test
3. Vérifier qu'il **échoue** (pour la bonne raison) : **RED**

## Le développeur 2 au clavier

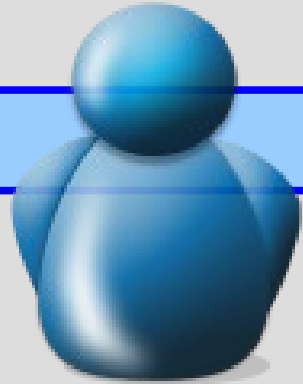


4. Écrire le **code minimum** pour le faire passer
5. Vérifier que le test **passe** : **Green**
6. **Remanier** le « code qui sent » : **Refactor**

# Développement ...Pong

## Le développeur 2 au clavier

1. **Écrire** un test de programmeur
2. **Compiler** le test
3. Vérifier qu'il **échoue** (pour la bonne raison) : **RED**



## Le développeur 1 au clavier

4. Écrire le **code minimum** pour le faire passer
5. Vérifier que le test **passe** : **Green**
6. **Remanier** le « code qui sent » : **Refactor**



# Développement Ping Pong

1. **Écrire** un test de programmeur
2. **Compiler** le test
3. Vérifier qu'il **échoue** (pour la bonne raison) : **RED**

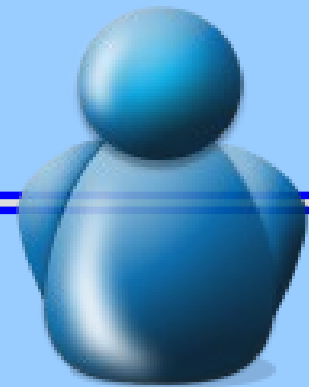
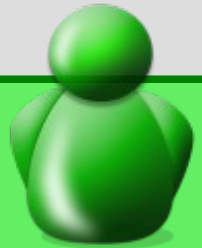
4. Écrire le **code minimum** pour le faire passer
5. Vérifier que le test **passe** : **Green**
6. **Remanier** le « code qui sent » : **Refactor**

7. **Écrire** un test de programmeur
8. **Compiler** le test
9. Vérifier qu'il **échoue** (pour la bonne raison) : **RED**

10. Écrire le **code minimum** pour le faire passer
11. Vérifier que le test **passe** : **Green**
12. **Remanier** le « code qui sent » : **Refactor**

Micro tâche 1

Micro tâche 2



# Le développement

- Qualité technique
  - Technique de binôme
  - Environnement de travail

*Environnement  
efficace*

*Adapté*

*Homogène*

# Le développement

- Qualité technique
  - Technique de binômage
  - Environnement de travail
- Qualité humaine

*7h de travail  
intensif*

*Côte à côte*

*A épier*

*Etre épier*



# Au *Planning Game*

- Les membres :
  - **Découpent** les scénarios,
  - En tâches qu'ils **estiment**
- Comme si ils travaillaient dessus demain
  - Pouvoir intervenir sur n'importe laquelle
- Les tâches ont une granularité fine

# Au *Planning Game*

- Les membres :
  - Découpent les scénarios,
  - En tâches qu'ils **estiment**
- Comme si ils travaillaient dessus demain
  - Pouvoir intervenir sur n'importe laquelle
- Les tâches ont une granularité fine

**Les binômes détiennent  
Les clefs de leur destinée**



# La (dé)formation du binôme

- Quand
  - Au stand-up meeting
  - En cours de journée
- Par Qui
  - L'équipe (*voire le Coach au début*)
- Comment
  - En fonction des besoins
  - Pour favoriser le transfert de connaissance (fonctionnelle ou technique)

# Le binôme dans l'équipe

- Pollenisation
  - Gestionnaire de version (update puis commit)
  - Gestionnaire de tâches
  - Le *stand-up meeting*...
- Amélioration
  - Le *stand-up meeting*...
  - Planning Game (rétrospective de fin d'itération)
- Rotation
  - Le *stand-up meeting*...

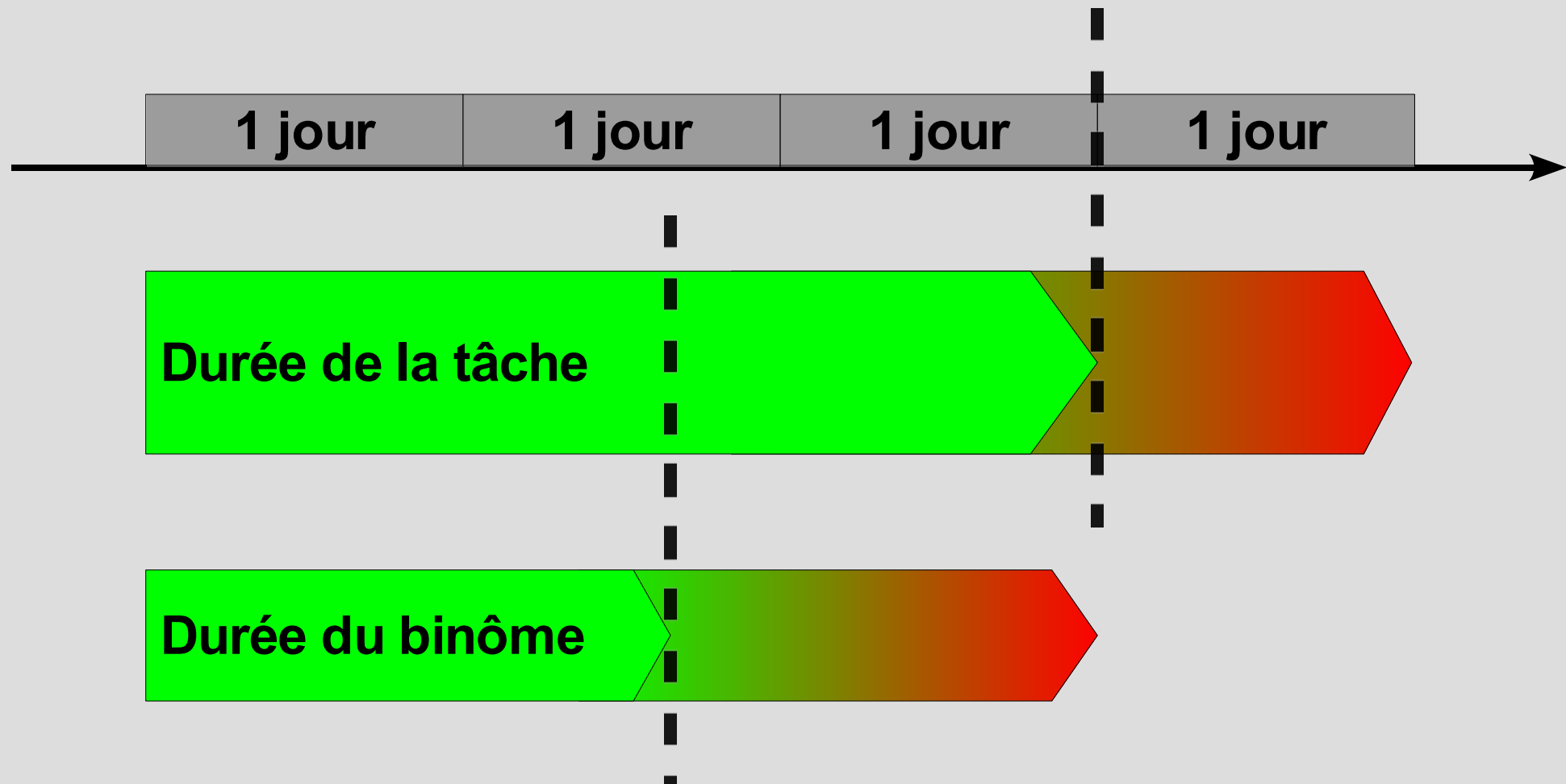
# Le *Stand-up Meeting*



- **Objectifs par binôme**
  - Résumer la journée passée
  - Planifier la journée à venir
  - Identifier ce qui empêche d'avancer
- **Objectif final**
  - Permuter / former les binômes

**Compromis entre  
Identifier les bugs  
& Efficacité**

# « Les meilleures sont les plus courtes »

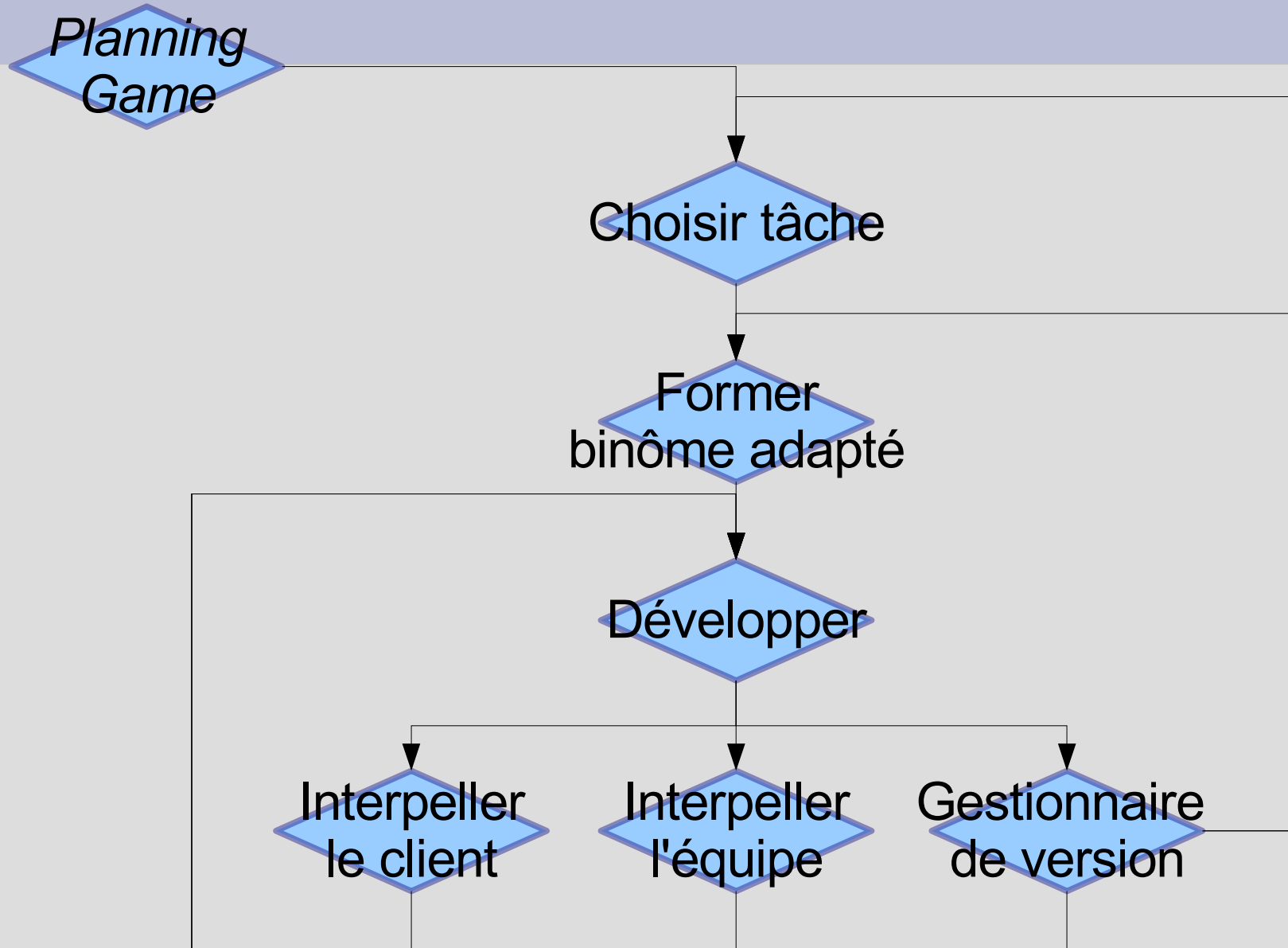


# Le *Stand-up Meeting*

- **Trait d'union entre deux journées**  
Le matin, soir ou midi...
- **La communication,**  
Lieu et instant d'échange
- **Prendre de la hauteur,**  
Pas mécanique
- **Ne pas prendre de décision,**  
Il faut être efficace ( $\leq 20$  minutes) !



# Les moments de la vie





# Questions

*Difficile ?*



*Mon binôme  
est  
caractériel !*

*Sur la  
durée ?*

# L'Outillage



# Outillage

- Le binôme
  - Favorise la **COMMUNICATION**
  - Par rapport aux processus et aux outils

# Minimum mais optimal !

- Donc **PEU** d'outil, mais de **BONS** outils
  - Peu de documentation écrite
  - Une intégration continue
  - Une « boîte à tâche » où l'on peut :
    - Définir des tâches
    - Y associer une estimation du temps
    - Y saisir le temps effectif de développement
  - Un « bon » IDE...

# Un « bon » IDE

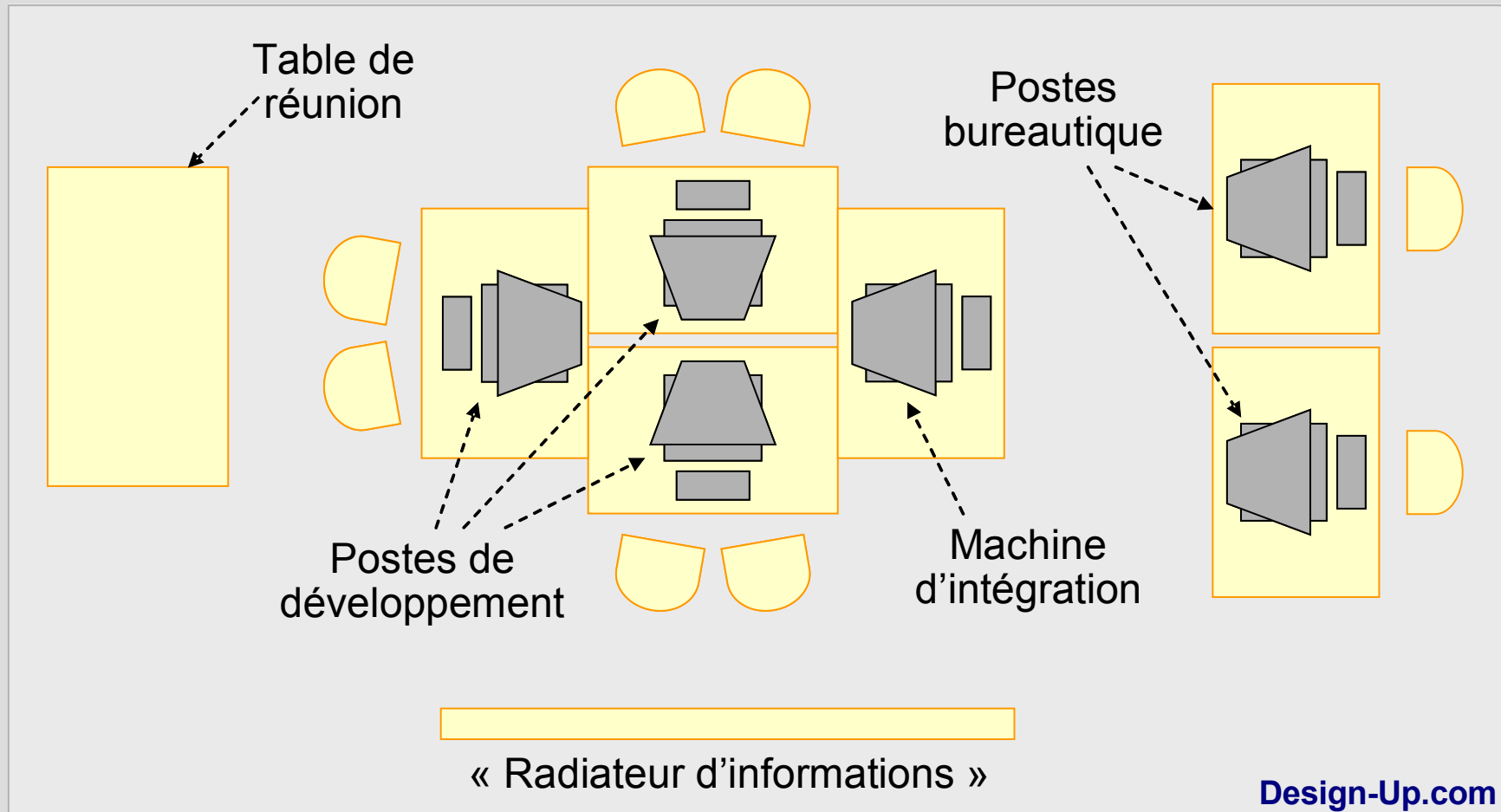
- IDE maîtrisé
- IDE optimisé
- Pour tout membre de binôme !

# Un « bon » IDE

- IDE maîtrisé
- IDE optimisé
- Pour tout membre de binôme !

**Idem pour l'environnement**

# La War Room



# Pourquoi...

- Réduire le nombre d'outils
- Optimiser ces outils

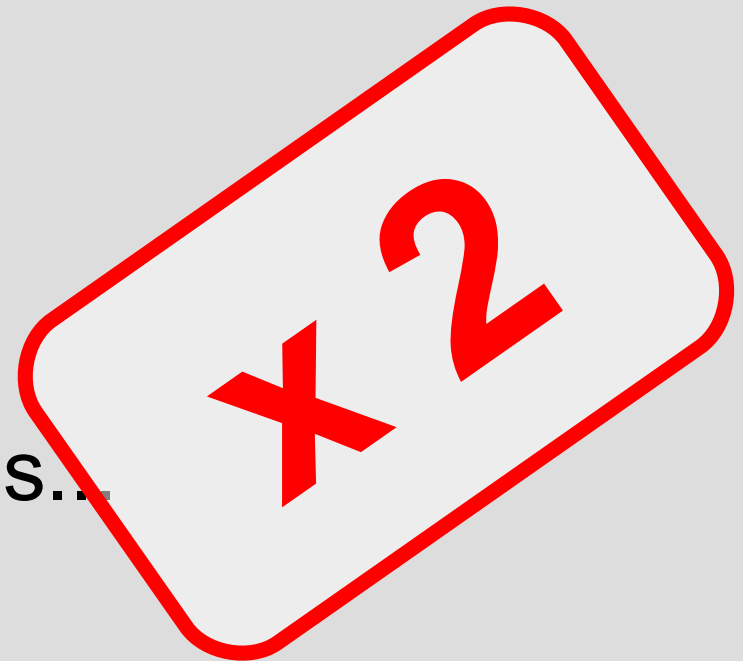
Sinon tout gâché...



# Pourquoi...

- Réduire le nombre d'outils
- Optimiser ces outils

Sinon tout gâchis...



# Questions

*Le « meilleur »  
IDE ?*



*C'est quoi  
l'Intégration  
Continue ?*

*Pour toute  
techno ?*

# Pour le manager



# Partie d'un tout

**Binôme de personnes**

- Association de membres
- Combinant qualités et défauts
- Pour résoudre les problèmes

# Partie d'un tout

**Binôme de personnes**

- Association de membres
- Combinant qualités et défauts
- Pour résoudre les problèmes

**Equipe de Binômes**

# A court terme

- C'est plus cher
- Si on ne compte pas les **bienfaits**
- Problème
  - Difficile de quantifier les dépenses relatives aux bienfaits

# A long terme

- **Cohésion de groupe**
- Diffusion de la **connaissance**
- Production de **Conception**
- **Amélioration** : application et processus

# Cohésion de groupe

- On apprend à se connaître
  - Plus efficaces ensemble avec le temps
- Une ambiance qui s'entend...
- Emulation positive

*« Tu m'as aidé hier, je t'aide aujourd'hui »*

*« Ensemble dans la même galère »*



# Diffusion de la connaissance

« Le binôme favorise

la **COMMUNICATION**

par rapport aux processus et outils... »

# Diffusion de la connaissance

- Une documentation écrite
  - Coûte à être maintenue
  - Seulement avec le bon outil
  - Seulement par les « élus »
  - Ne s'adapte pas à l'interlocuteur

*« Ecrite hier,  
Périmée aujourd'hui,  
Délaissée demain »*

# Diffusion de la connaissance

- Une information verbale
  - S'améliore plus facilement dans le temps
    - Plus concise, réduite au nécessaire
    - A jour
    - S'adapte aux interlocuteurs
    - Dynamique, vivante
  - Les 2 interlocuteurs participent
    - On pose plus facilement les questions (d'un côté)
    - On hésite toujours à dire des bêtises (de l'autre)

# Refonte Properties



General

Security

Customize



Refonte

Type: File Folder  
Location: G:\Projets  
Size: 105 GB (113'135'142'840 bytes)  
Size on disk: 111 GB (119'681'345'788 bytes)  
Contains: 1'782'115 Files, 44'391 Folders  
Created: mercredi, 5. janvier 2005, 09:59:32

Attributes:  Read-only  
 Hidden

Advanced...

OK

Cancel

Apply

# Diffusion de la connaissance

Il faut des moyens



# Diffusion de la connaissance

## Il faut des moyens

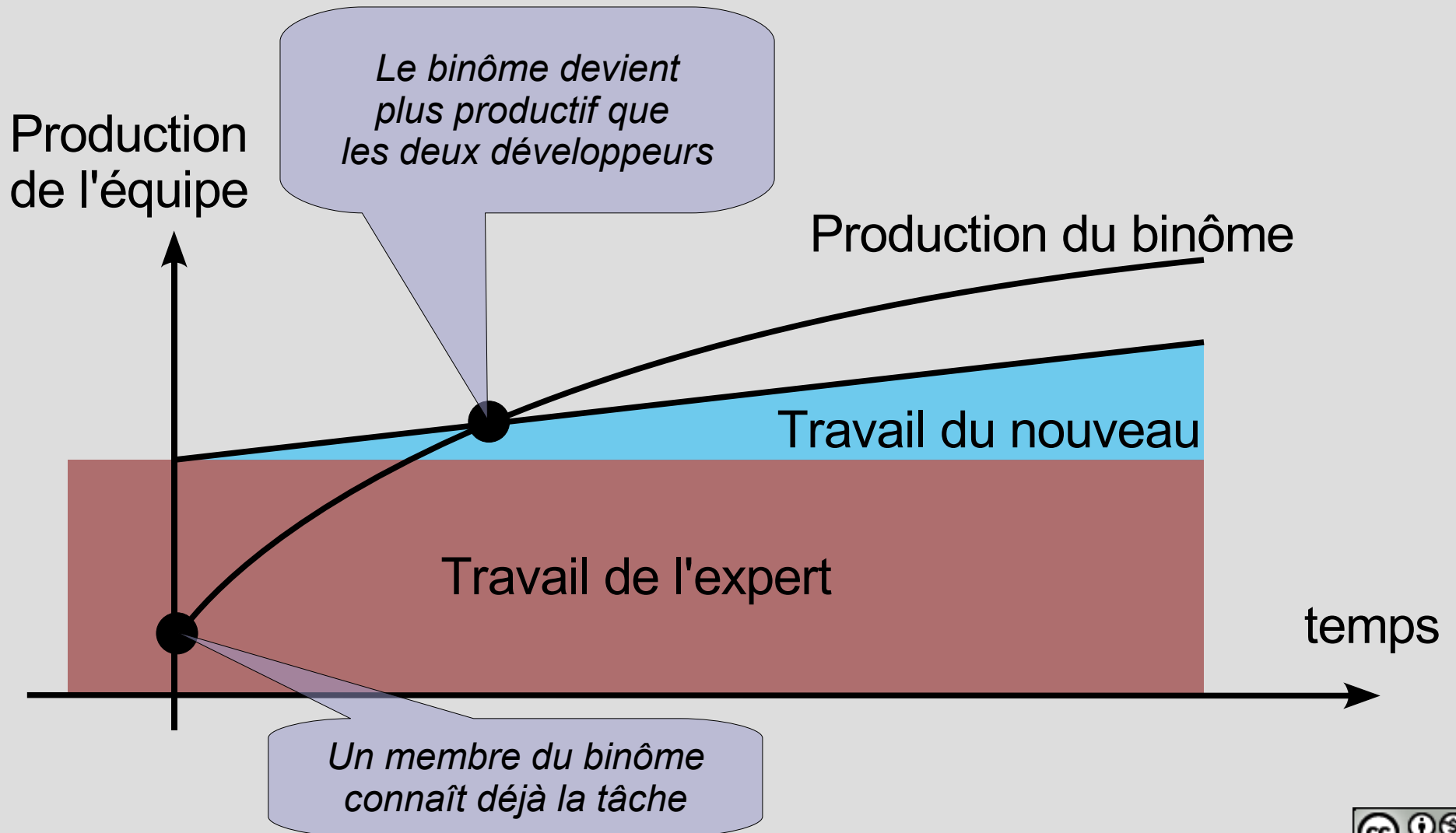
- Savoir « Qui sait quoi » grâce à
  - La **rotation** des binômes
  - Le ***Stand-up meeting***
- Un environnement ouvert pour faciliter
  - La **demande** à un autre
  - L'**intervention** d'un autre
- Des personnes curieuses et courageuses
  - Pour **poser** des questions
  - Pour **chercher** la réponse



# Diffusion de la connaissance

- Transfert technique & fonctionnel
- Plus de « boîte noire » dans le code
- Renforcé par
  - Propriété collective du code
  - Conventions de code
  - Tests (unitaires et fonctionnels)
  - Intégration Continue

# Diffusion de la connaissance





# Resistance au *turn-over*

- Ma dernière mission XP, en l'espace d'un an :
  - J'étais le plus ancien de l'équipe
  - Donc renouvellement complet de 8 développeurs
- Cela s'est fait avec :
  - Aucune perte de qualité  
(confère résultats des tests et satisfaction client)
  - Aucune "perte de connaissance"
- Qui peut en dire autant ?...

# Meilleure réutilisation

- Deux fois plus de chances de connaître
  - L'existence d'une librairie
  - La façon d'utiliser un *framework*
  - La résolution d'un problème

# Profils adaptés

Diffusion de l'information,  
Résistance au turn-over,  
Nécessite aussi des profils adaptés :

- Pas de référent sur l'application
  - Tant technique
  - Que fonctionnel
- Pas de rétention d'information
- Personne n'est indispensable

# Production de conception

- Principale force du binôme : La réflexion
- Réflexion partagée  $\Rightarrow$  Conception plus
  - Rapide
  - Approfondie
  - Robuste
  - Agréable

# Production de conception

- Principale force du binôme : La réflexion
- Réflexion partagée ⇒ Conception plus
  - Rapide
  - Approfondie
  - Robuste
  - Agréable



**Synergie**

# Meilleure maîtrise des délais

- L'appropriation collective de l'application
- Associée à un planning game
- Où les développeurs :
  - Définissent les tâches
  - Les estiment
- Induisent des estimations plus justes

# Productivité Constante

- Minimum d'interruptions
  - Sur la minute
    - Un des deux membres à une baisse de régime
  - Sur l'heure
    - Un « visiteur » hésite à déranger un binôme
  - Sur la journée
    - Un membre doit s'absenter
  - Sur l'itération
    - Un membre part en vacances
  - Sur le projet
    - Un membre quitte le projet



# Productivité Constante

- Important de maintenir la dynamique
  - Au sein du binôme
    - *Ping Pong Development...*
  - Entre binômes
    - Rotations...
  - Limiter les heures supplémentaires
    - Rythme durable

**Alterner pour entretenir**



# Meilleure qualité

- Avec le développement ping pong
  - Synergie positive
  - Le copilote est plus pointilleux sur les tests
- Amélioration continue
  - Harmonisation
  - Conventions de code
  - Propriété collective du code
  - Remaniement
- Moins de retours de bug
  - Moins de dépenses
  - Satisfaction client

# Meilleure qualité

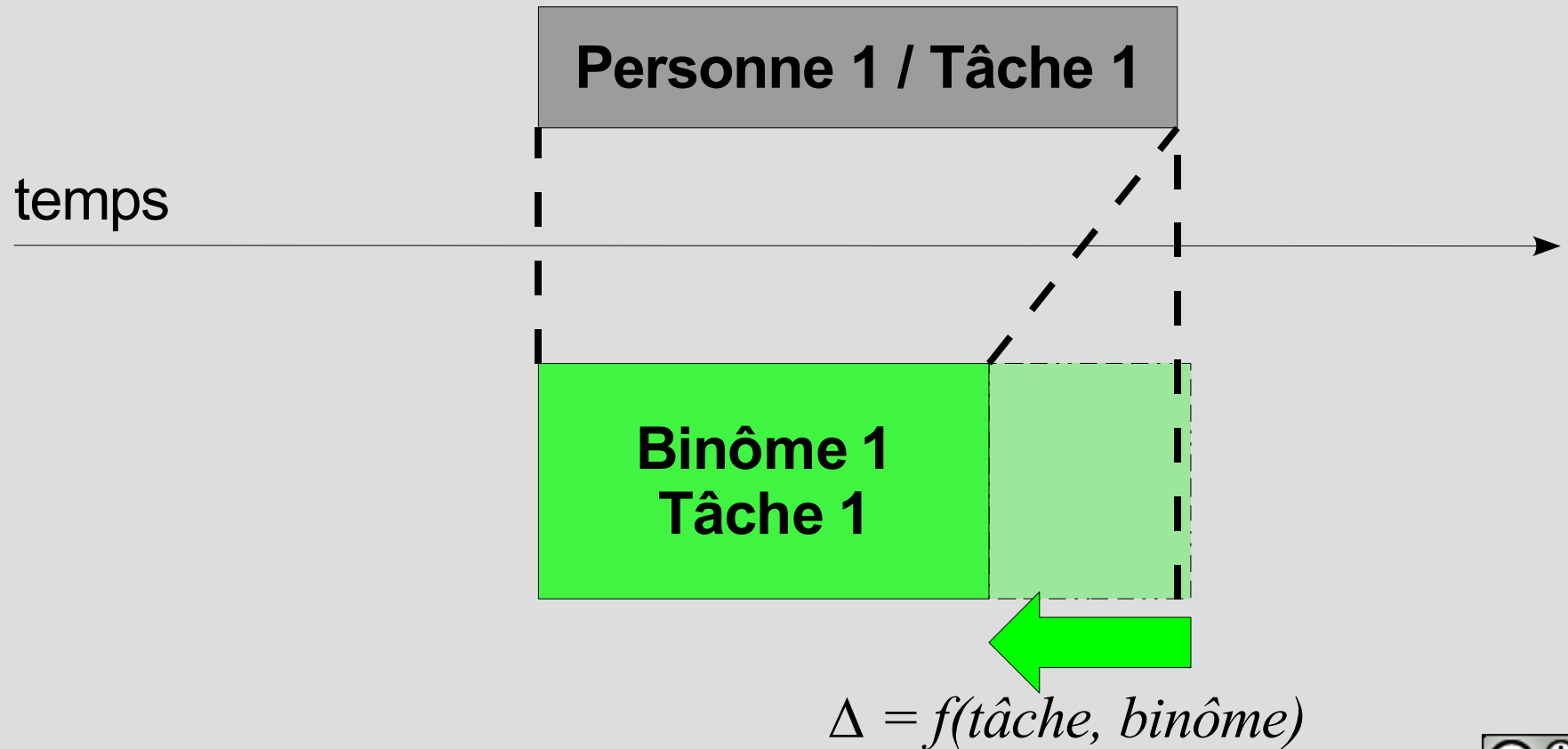
- Compétition saine
  - Copilote plus facilement rigoureux
  - Pilote incité prendre exemple sur lui...
  - Voir à l'être encore plus
- Encouragement mutuel
  - « L'union fait la force »
- Permet de surmonter les défis
  - On ne laisse pas de défaut
  - On fait le remaniement nécessaire,
  - En lieu et place d'une verrue

Amélioration  
Continue

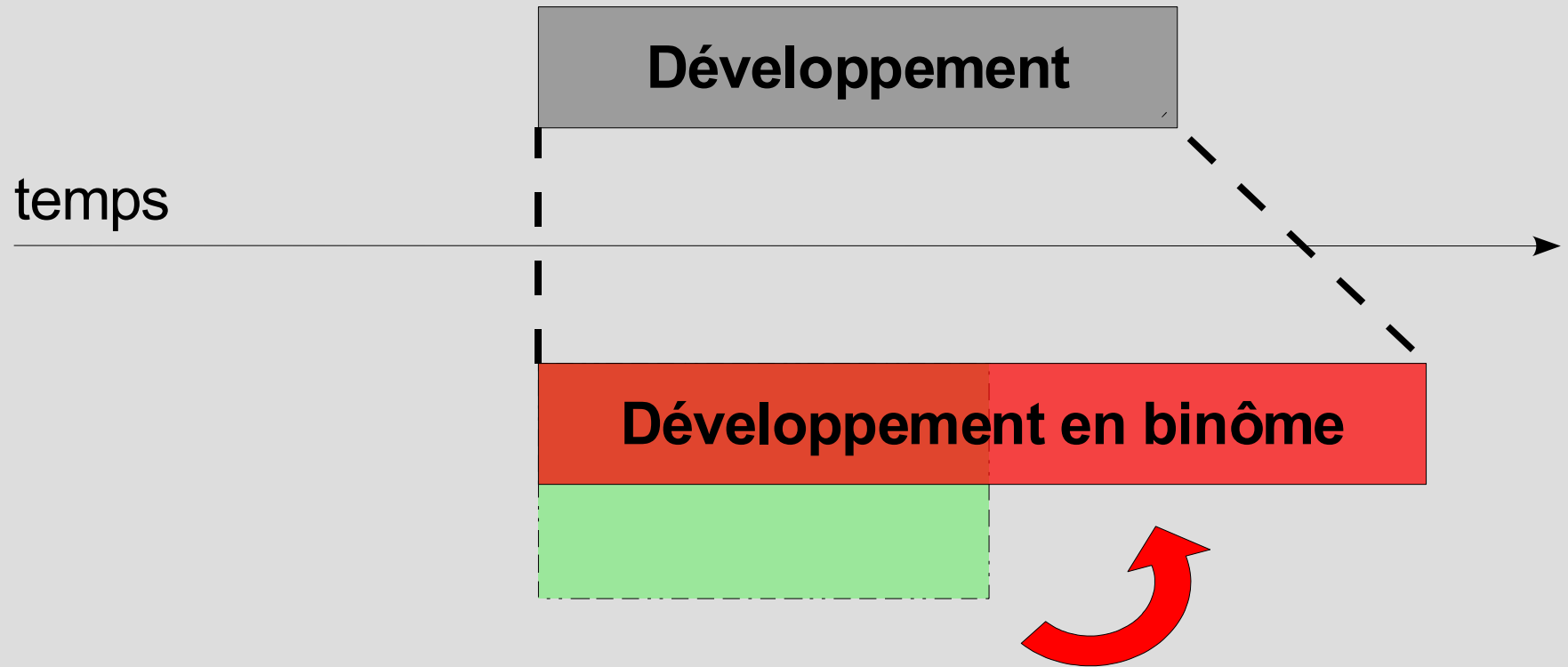
# Coût du Binôme ?



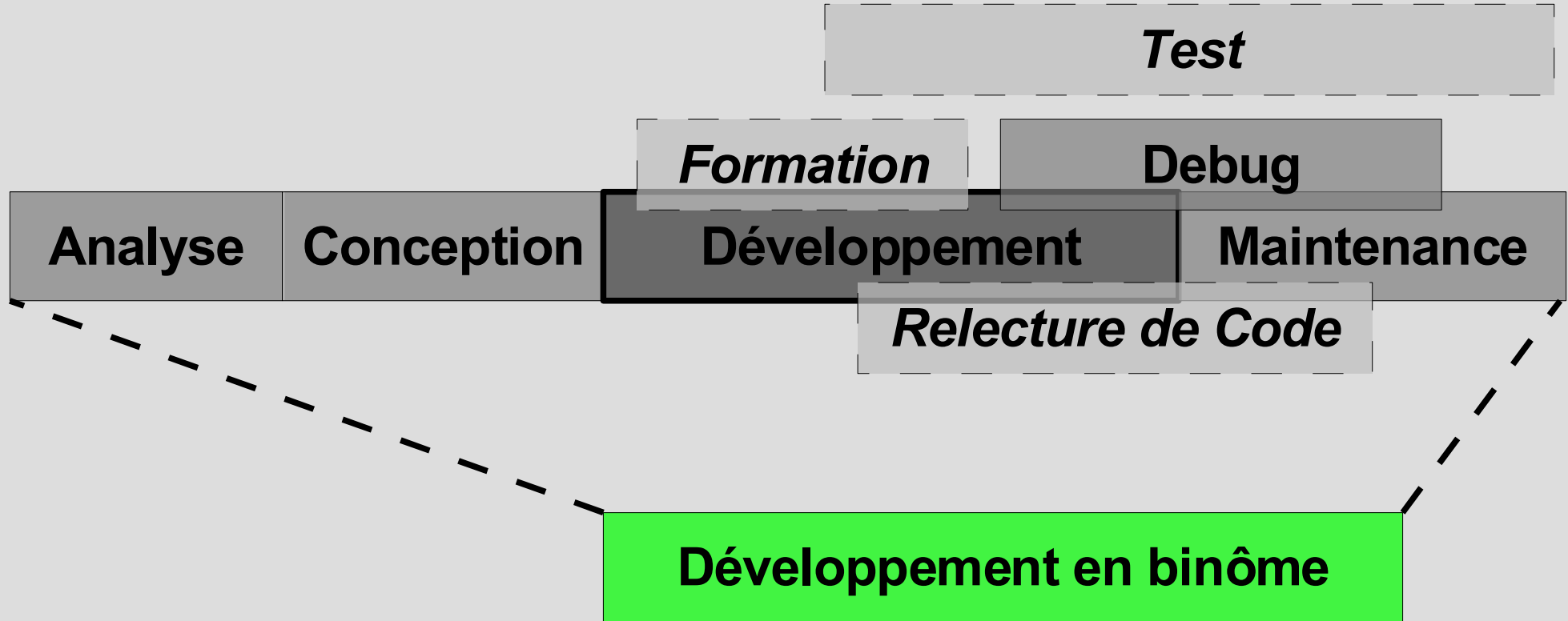
# Coût du Binôme ?



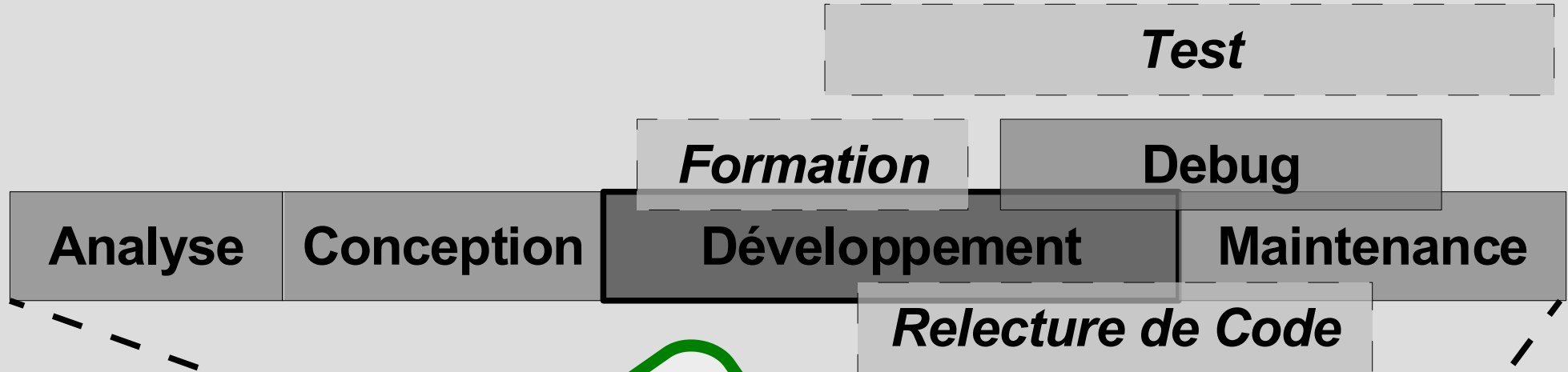
# Coût du Binôme ?



# Coût du Binôme ?



# Coût du Binômage !!!



Turn Over --  
Réutilisation++

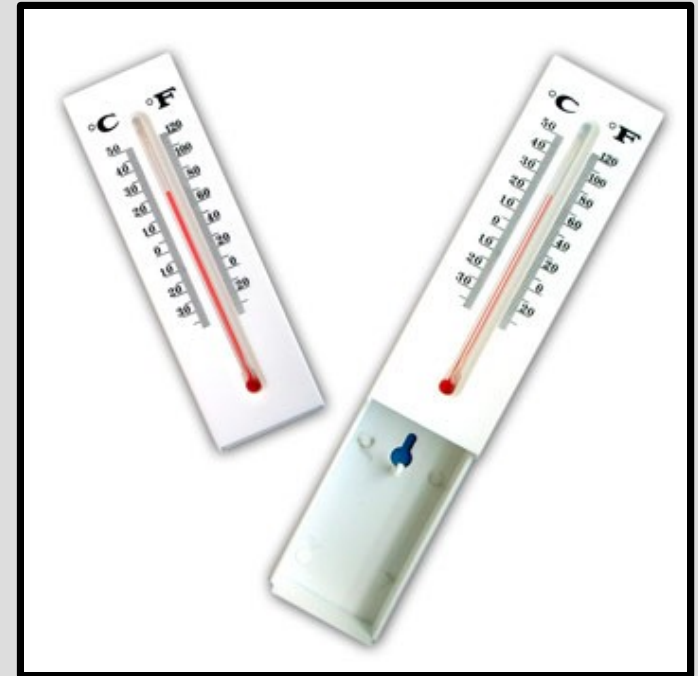
Développement en binôme

Rythme soutenu  
Qualité Croissante



# Un révélateur

- Développeur « différent »
- Conception biaisée
- Développements non maîtrisés





# Mise en place

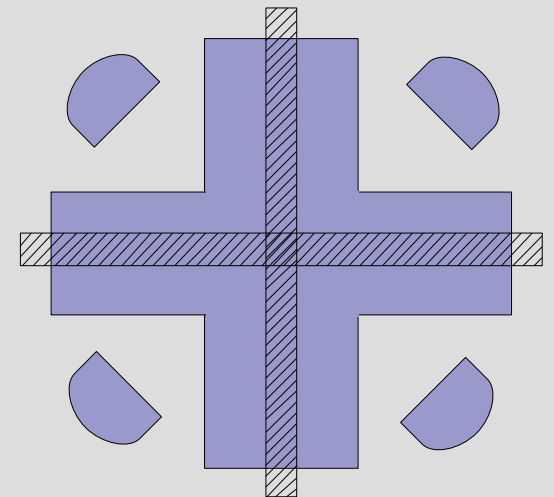


- Quand
  - 1h par jour, 1/2 journée par semaine
  - A heure fixe, périodiquement, systématique
  - Pour casser habitudes et pressions du “Monomage”
- Quoi
  - **Surtout pas sur une tâche difficile**
  - **Pour se concentrer sur le processus**
- Comment
  - Retrospective sur la tâche
  - Retrospective sur le processus

# Mise en place



- Qui
  - Les réfractaires
    - Les identifier
    - Les sensibiliser (ou appeler Léon...)
    - Les rassurer
  - Mixer volontaires / réfractaires
- Où
  - Espace de travail adapté
- Voir l'article de Dominic Williams  
[http://www.dominicwilliams.net/en/pair\\_diff.html](http://www.dominicwilliams.net/en/pair_diff.html)



# Recruter un bon « 1/2 binôme »



- **Binômer (très) souvent nécessite :**
  - Ouverture d'esprit & Remise en question
  - Courage (de se mettre à nu)
  - Communication
  - Respect
  - Capacité à tourner sur les tâches
  - A ne pas être « responsable produit »
  - Aptitude à travailler à deux
    - Accepte le « viol de l'intimité »
    - Partager la gloire...
    - Et les erreurs (Binôme est une mini-équipe)

# Recruter un bon « 1/2 binôme »



- Relationnel > Technique
- Développeur moins (pas ;o) technique...
  - Moins cher
  - Et plus de choix (Anti “Experts J2EE” ...)
- Car
  - La technique s'apprend
  - Former un débutant ou déformer un gourou
  - Difficile d'agiliser quand (dé)formé autrement

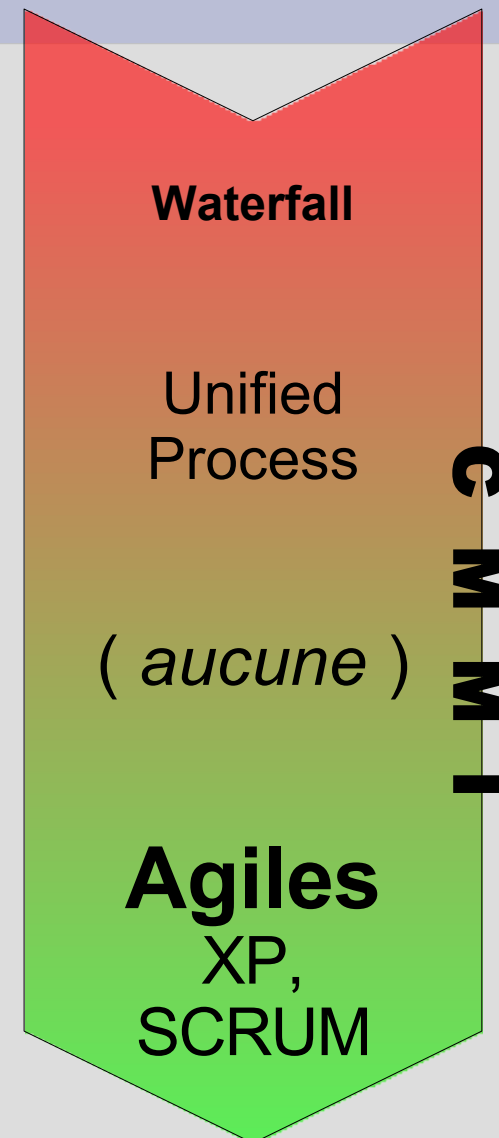
# Recruter un bon « 1/2 binôme »



- Valeurs *eXtreme Programming*
- En entretien, en demander la définition
- Essayer les mises en situation
- Faire participer n membre(s) de l'équipe
- Recruter des « 1/2 binôme » complémentaires

# Binômage et « méthodologies »

- Non cloisonnement
  - Des tâches
  - Des modules
  - Des personnes
- Itérative et incrémentale
  - Pas de conception initiale et complète
- Valorisant les personnes
- Donc la communication



# Questions

*Comment  
évaluer les  
gains ?*

*Et quand on  
m'impose une  
équipe ?*



*Binômer sur  
toutes les  
tâches ?*

# En pratique...





# Atelier Binômage

- 10' : Binomage
- *Stand-Up*
- 10' : Binomage
- *Stand up et Retrospective*

# Le *Stand-up Meeting*

- **Trait d'union entre deux journées**  
Le matin, soir ou midi...
- **La communication,**  
Lieu et instant d'échange
- **Prendre de la hauteur,**  
Pas mécanique
- **Ne pas prendre de décision,**  
Il faut être efficace ( $\leq 20$  minutes) !



# Le *Stand-up Meeting*



- **Objectifs par binôme**
  - Résumer la journée passée
  - Plannifier la journée à venir
  - Identifier ce qui empêche d'avancer
- **Objectif final**
  - Permuter / former les binômes

**Compromis entre  
Identifier les bugs  
& Efficacité**

# Conclusion



# Conclusion (1/4)



- Binôme plus coûteux...

Mais que mesure-t-on ?

- Seulement les lignes de code par mois
- Ou des choses inutiles insignifiantes comme :
  - Le nombre de bugs
  - La “véracité” des estimations
  - ....
- Dit autrement : **LA SATISFACTION**  
(client ET chef de projet)

# Conclusion (1/4)

Dépense Immédiate

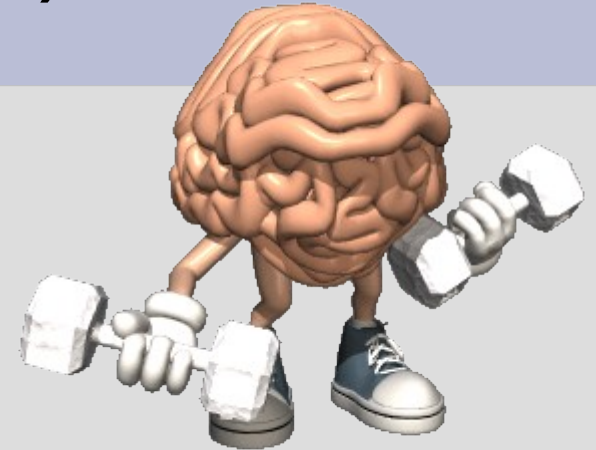


VS.

Retour sur Investissement

# Conclusion (2/4)

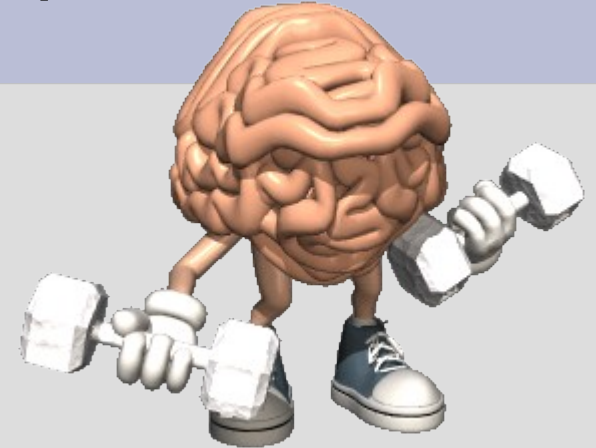
- 1 Binôme =
  - 2 Bras (comme un monôme)
  - **2 Cerveaux** (valeur ajoutée)
  
- **Binômage valorisé**  
**Quand cerveau valorisé**



# Conclusion (2/4)

## Cerveau valorisé quand

- Réflexion valorisée
  - Valorisation des personnes
  - Processus Itératifs & Incrémentaux
  - Tâches et projets au moins sur moyen terme
- Confrontation valorisée
  - Binômes pertinents, complémentaire
  - Echange entre binômes





# Conclusion (3/4)

Le binômage est :

- **Contraignant**
- **Intensif**



# Conclusion (3/4)

Le binômage est :

- **Contraignant**  
    ➔ **Profil adapté**
- **Intensif**  
    ➔ **Rythme durable**



# Conclusion (4/4)

- Binômage facilité et amplifié par :
  - Rythme soutenable
  - Propriété collective du code
  - Les conventions de code
  - Les Tests (unitaires et fonctionnels)
  - Le TDD et remaniement
  - Stand-Up meeting
  - Le client sur site
  - Planning Game

**Pratiques Agiles**

# Le déficit du binôme

Un « grand penseur » contemporain a dit :

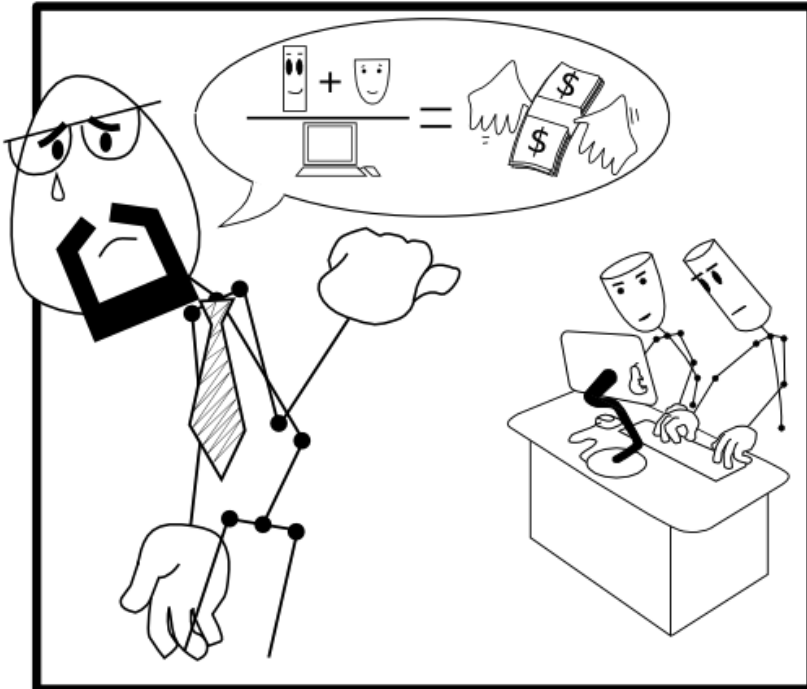
$$\ll 1 + 1 = 1$$

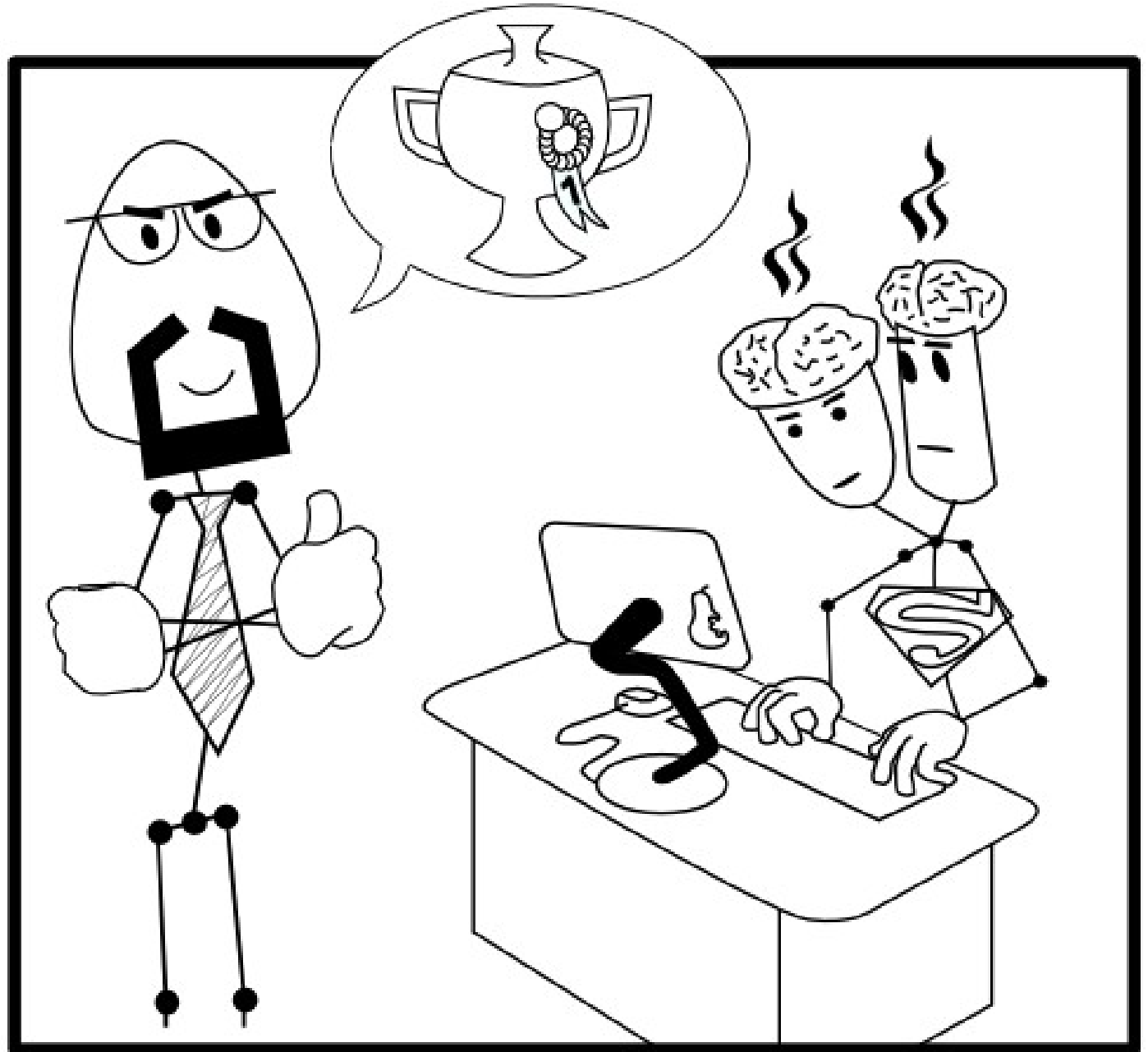
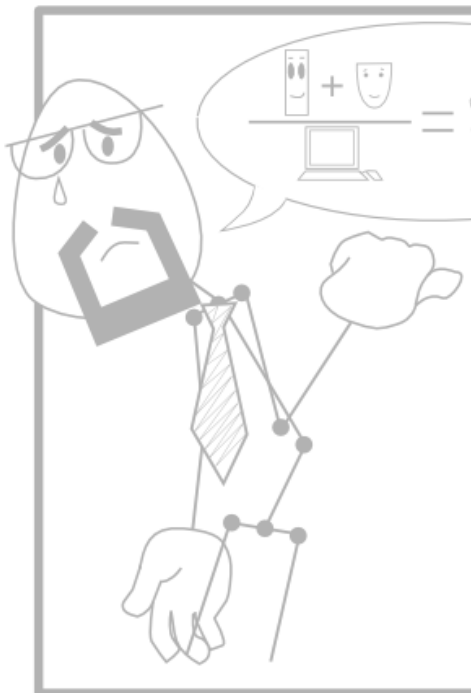
[...]

$$1 + 1 = 11 \gg$$

... *Jean-Claude Van Dame*

# Conclusion





Xcoders

J.LAYAT

JERRY@AGILE-SWISS.ORG 15/04/07

Xcoders

J.LAYAT JERRY@AGILE-SWISS.ORG 15/04/07

# Pour aller plus loin...

- Quelques ressources sur le Binôme :
  - Sites
  - Parutions
  - Livres

# Sites



- Laurie WILLIAMS : [Selling Pair Programming](#)
- [A Pair Programming Experience](#)
- Dominic Williams
  - [Don't restrict pair programming to difficult tasks](#)
- JournalDuNet : [La programmation en binôme](#)
- [Agile-Swiss.org](#)
  - Merci Jérôme pour les XCodeurs





# Publications

- Les publications de Laurie WILLIAMS
  - The Costs and Benefits of Pair Programming
  - Strengthening the Case for Pair-Programming
  - Pair Programming: Experience the Difference
  - ...

# Livres

## Pair Programming Illuminated



Laurie Williams and Robert Kessler

- **Pair Programming Illuminated**  
*De Laurie Williams (encore elle ;o), 2002*
  - Ch 3 : The Seven Synergistic Behaviors of Pair Programming
  - Ch 4 : Overcoming Management Resistance to Pair Programming
  - Pair Programming: Experience the Difference

# QUI...



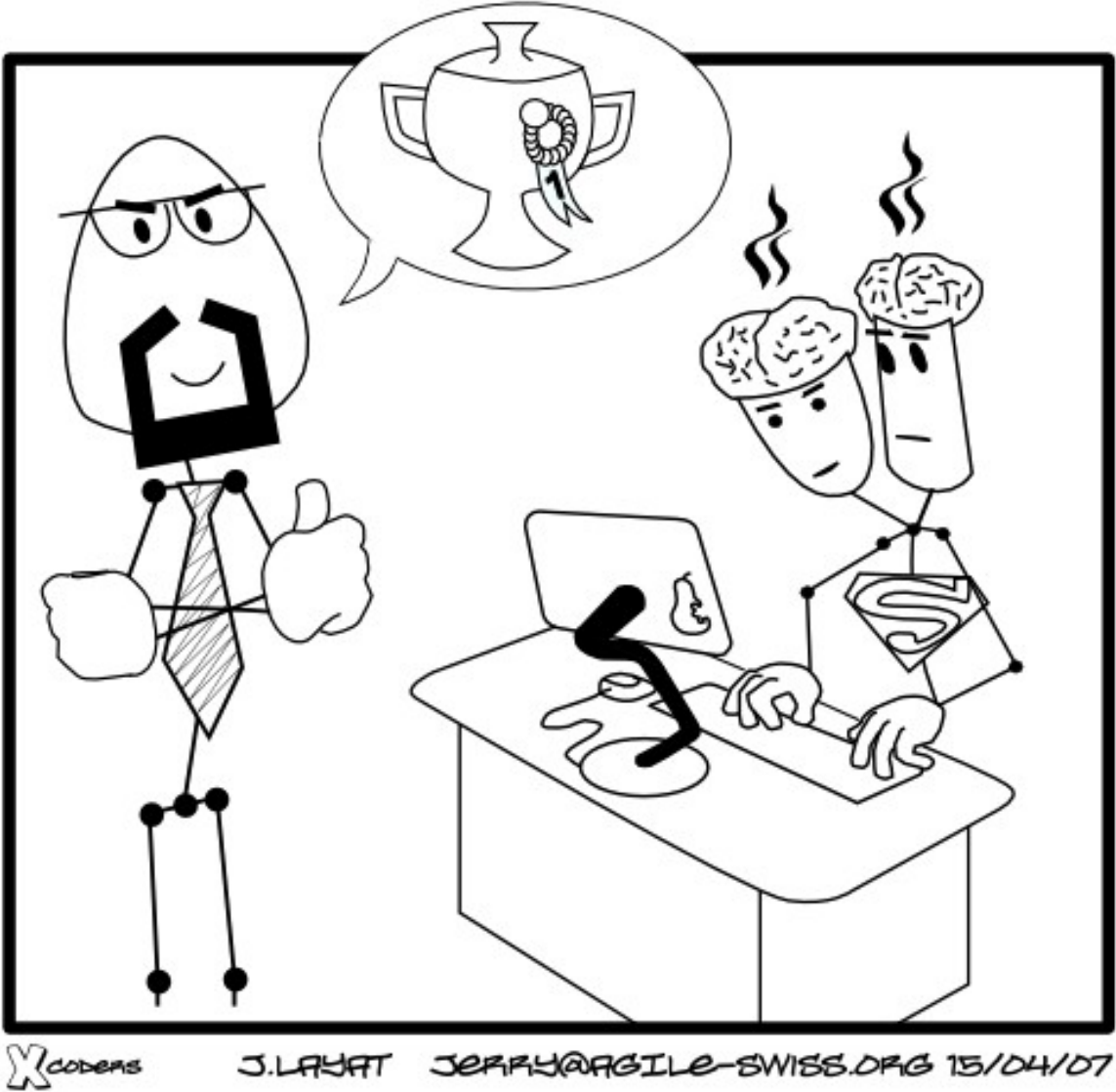
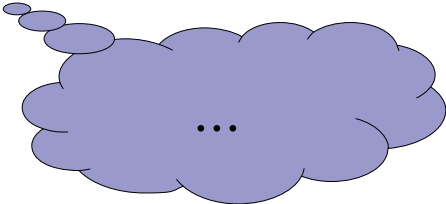
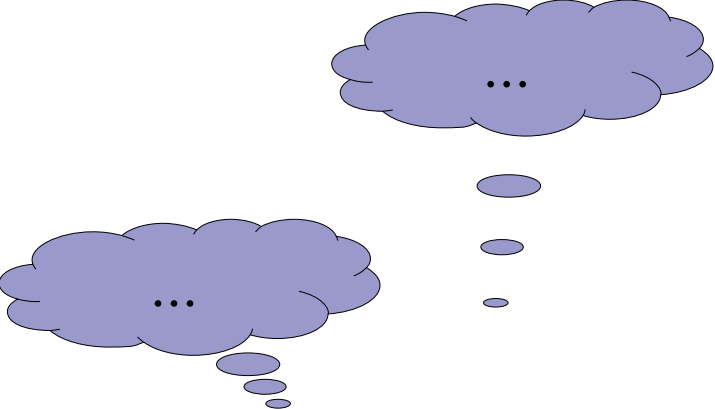
- **Ne veut pas binômer ?**
- **A envie de binômer un jour ?**
- **Fait tout pour binômer lundi ?**

# QUI...



- Ceux qui en ont déjà fait :
  - Quel est votre ressenti ?
  - Pour vous ? Pour le projet ?
- Ceux qui n'en n'ont jamais fait :
  - *Comment voyez-vous le binôme ?*

# Questions



# Débat



# Un référent



- Toujours garder un referent sur la tache / story ... ??

# Chacun le sien...



- Chacun son clavier ?
- Chacun son écran ?
- Chacun sa machine ?



# Ensemble... A distance !



- Est ce possible ?
  - Techniquement parlant : OUI
- Est souhaitable ?
  - La polémique est là ....

# Sur toutes les tâches ?



- Binômer sur toutes les tâches ?
  - Fastidieuses
  - “Simplisimes”
  - Non risquées
  - Exploratoire
  - Peu importantes
  - Peu difficile
  - Rébarbatives, mécaniques, répétitives, pénibles
  - Y a pas à réfléchir
  - Y a pas de risques...

# Sur toutes les tâches ?



- Binômage sur les tâches rébarbatives
  - Profiter pour faire des choses plus funs, plus intelligentes
  - Le “pénible” devient “récréation”

# Sur toutes les tâches ?



- “Exploration binômiale”
  - Quand on explore à deux
    - Deux vitesses d'exploration différentes
    - Deux axes d'exploration différents
  - On continue à “binômer sur l'exploration”
    - Chacun sur un PC
    - Deux PC côte à côte
    - “Poste d'exploration”
    - War Room adaptée

# Les horaires



- Le binôme doit synchroniser ces heures de travail. Par extension, l'équipe aussi
- Trop d'écart nuis au transfert de connaissance.

# Typologie de projet



- Projet type pour ou contre le binôme
  - Technologies
  - Domaine métier
  - Périmètre fonctionnel
  - Durée
  - Nombre de ressources
  - Compétence des ressources (+/- imposées)
  - Méthodologie
  - Client

# Dimension sociologique



- Bulle d'intimité
  - Les photos sur le poste de travail
  - Le lecteur mp3 sur les oreilles
  - ...

# Comportement de l'expert



- L'expert :
  - Prendre le temps nécessaire
  - Ecouter
  - Valoriser le novice
  - Ne pas toucher au clavier !
- Le novice :
  - Courage, Curiosité, Confiance en soi,
  - Epanouissement



# Toujours accompagné ?



- Favoriser le transfert de connaissance
- Si besoin organiser le travail pour favoriser des moments de libertés

# Resistance au *turn-over*

- Valoriser les experts en les incitant à :
  - Communiquer,
  - Former,
  - Ecouter,
  - Synthétiser leur pensée
- Devenir des référents sur le processus du binôme, sur la méthodologie
- Les amener doucement mais sûrement à des postes de plus haute responsabilité

# Le développeur qui pose problème



- Pas forcément “très problématique”
  - Plus facile de former un débutant  
Que de déformer un gourou..
- Détecter plus rapidement
- D'avantage néfaste
  - Peut infecter plus largement l'application
  - Peut incommoder les autres membres
    - Plus vite
    - Plus largement
- Ne pas laisser gangrener...  
Car cela peut être rapide !



# Production de conception

- Tâches privilégiées : avec de la conception
- Fonctionnement privilégié : le TDD
- Méthodes privilégiées : itératives