



Atelier des Prénoms

Atelier Agile

V. Deslandres, IUT Université Lyon1
(d'après Henrik Kniberg 2011)

Durée

- 20' pour la simulation
- 10 à 20' pour le debriefing



Introduction

- Combien pensez-vous mettre de temps pour écrire le prénom AUGUSTINE ?
 - (en lettres majuscules)
- Combien de temps faudrait-il pour écrire 6 fois ce prénom ?

	Estimé	Tour#1	Tour#2
1 prénom			
6 prénoms			

Facteurs influents ?

■ Quels sont les **facteurs** qui vont influencer ce temps ?



- Longueur du prénom
- Complexité du prénom
- Stylo choisi
- La qualité attendue
- La facilité d'élocution
- La capacité à écrire
- Etc.

**Allons-y, tâchons
de trouver la
vérité !**

Déroulement de l'atelier



- Constituer 2 équipes de 7/8, en parallèle
- Séparation des rôles :
 - 6/7 **clients** (qui ne savent pas écrire)
 - 1 **développeur** (qui sait écrire, et qui a le bon outil)
- Un chronomètre

1 Développeur



4 – 6 Clients



Dave



Lisa



Bob



Eric



Maria

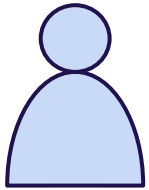


Consigne :

- Le client donne son ticket au dév et énonce un prénom (de son choix)
- Le dév écrit sur le ticket
- Le client valide, note le temps

2 post-it par client → un par tour

client



Le client note les temps

- de début :
- de fin :
- et calcule la durée :

Le fournisseur écrit le prénom du client

tour
#1

Le client note les temps

- de début :
- de fin :
- et calcule la durée :

Le fournisseur écrit le prénom du client

tour
#2

Précisions Clients



- En cas de bug (faute, illisible, mal écrit), **le client refuse la livraison**
 - Le DEV repart pour une nouvelle version
- Vous pouvez échanger avec le développeur
 - Mais ne pas montrer ou expliquer comment écrire le prénom demandé
(le client n'a pas connaissance de « l'écriture »)

Précisions Développeurs

- Suivre la politique de l'entreprise



Politique de l'entreprise

Ecrire en **majuscules**

Livrer vite le Client

- « Ne jamais faire attendre le client »
- « Client impatient, client pas content ! »
- **« Plus tôt on commence une tâche, plus tôt on l'aura fini »**
- et plus vite on passera à un autre Client...

Précisions Développeurs (suite)



- Pour ne faire attendre aucun Client, approche méthodique : « **Avancer sur chaque projet** » :
 - Écrire la 1^{ère} lettre de tous les prénoms
 - Puis la 2^{ème},
 - etc.
- Tous les clients seront **satisfaits, aucun n'attendra.**
- Quand une fiche est terminée : le client **écrit la durée** sur la fiche

ATTENTION ! Le client peut **répéter** le prénom à chaque fois, mais PAS les lettres (nécessaires au développeur)

- (Il donne l'objectif final, ne dit pas comment y arriver)

Prêt ? 1^{er} round

- Lancer le chrono
- GO !



Evaluation des temps



- Temps **médian** équipe 1 :
 - (classer les tickets durée croissante : prendre celle du milieu)
- Temps **médian** équipe 2 :
- Par rapport au temps estimé ?

	estimé	réel
1 prénom		
6 prénoms		

Rappel : facteurs influents



- Longueur du prénom
- Complexité du prénom
- Outil choisi
- La qualité attendue
- La capacité à écrire
- Etc. mais **n'a-t-on pas oublié quelque chose ?**

Le multi-tâche !

2ème Round



- Mêmes équipes, mêmes noms : échanger les développeurs !
- Nouvelle politique :
 - **Limiter le WIP à 1**, pour toutes les tâches
- Le développeur ne doit travailler qu'avec UN client à la fois
- C'est lui qui **tire le travail** à lui
- La tâche est **chronométrée**



Développeur

Client Jeff

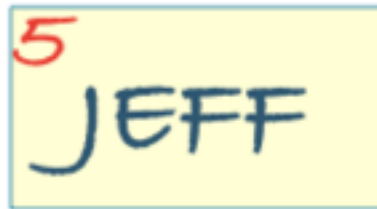
0
Demande
le travail



1

Note le temps de départ
Soumet le projet

2
Livre



3

Note le temps de fin
Calcule la durée du projet

Go !

Estimation des temps ?

- Cette fois « en mode **focalisé** »



	Estimé	Réel (multitâche)	Réel (focalisé)
1 prénom			
6 prénoms			

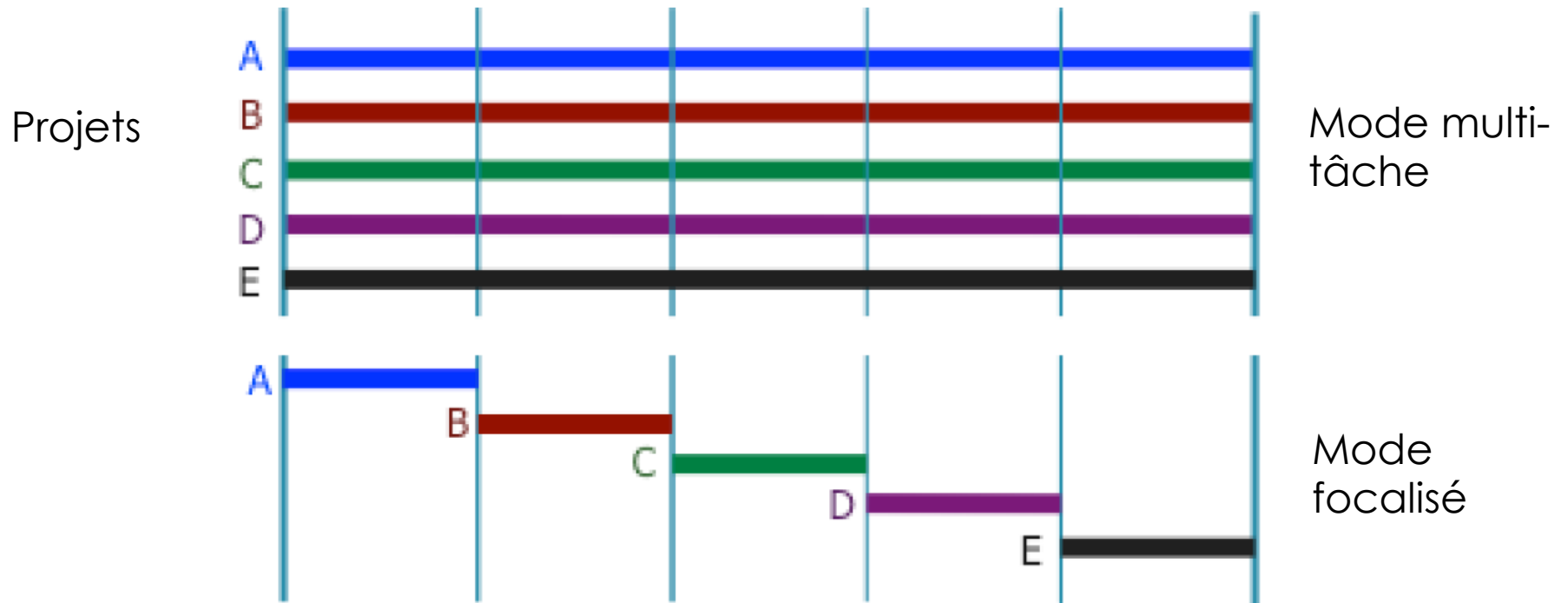
Conclusion



- Le **multitâche** est mauvais pour la rentabilité
 - Délais
 - Mauvaise qualité
 - Stress
- Quantité de **gaspillages inutiles**
- C'est pourtant si **facile** de s'y laisser entraîner...
- Ici, tâche simple; alors pour du développement logiciel, où chaque changement de contexte est bien plus difficile et long...
- Travailler à *flux tiré* plutôt que *poussé* est toujours meilleur (rentable)

Loi de Little

- Quand on fait n choses en même temps, chaque chose prend n fois plus de temps



Impact sur le planning



- Influence sur le **planning de livraison** : que savait-on après 10 sec. dans chaque round ?
 - 1^{er} round : rien, les projets ont démarré mais on ne sait pas quand ils finiront
 - 2^{ème} round : après 10 sec., on a 2 projets finis, donc on sait combien de temps environ prendront ceux qui restent
- **Valeur de l'estimation ?**
 - Au 1^{er} tour : estimation **très aléatoire** ! Même si on connaît tous les facteurs à l'avance (complexité du prénom, longueur etc.), le multi tâche fait que si un projet est ajouté, la durée de réalisation va augmenter de toute façon.
 - Au 2^{ème} tour : estimations **plus justes**, d'après l'expérience, de la durée de chaque projet (mené de façon isolée)

Impact sur la qualité du produit



- Si le **Développeur** n'a pas bien compris la demande du **Client**, il faut attendre *la fin* de tous les projets en multi-tâche (ex. 60 sec.), alors qu'en mode focalisé, c'est immédiatement après la livraison (5 sec.)
- En focalisé on est plus large : on a le temps de parfaire le travail

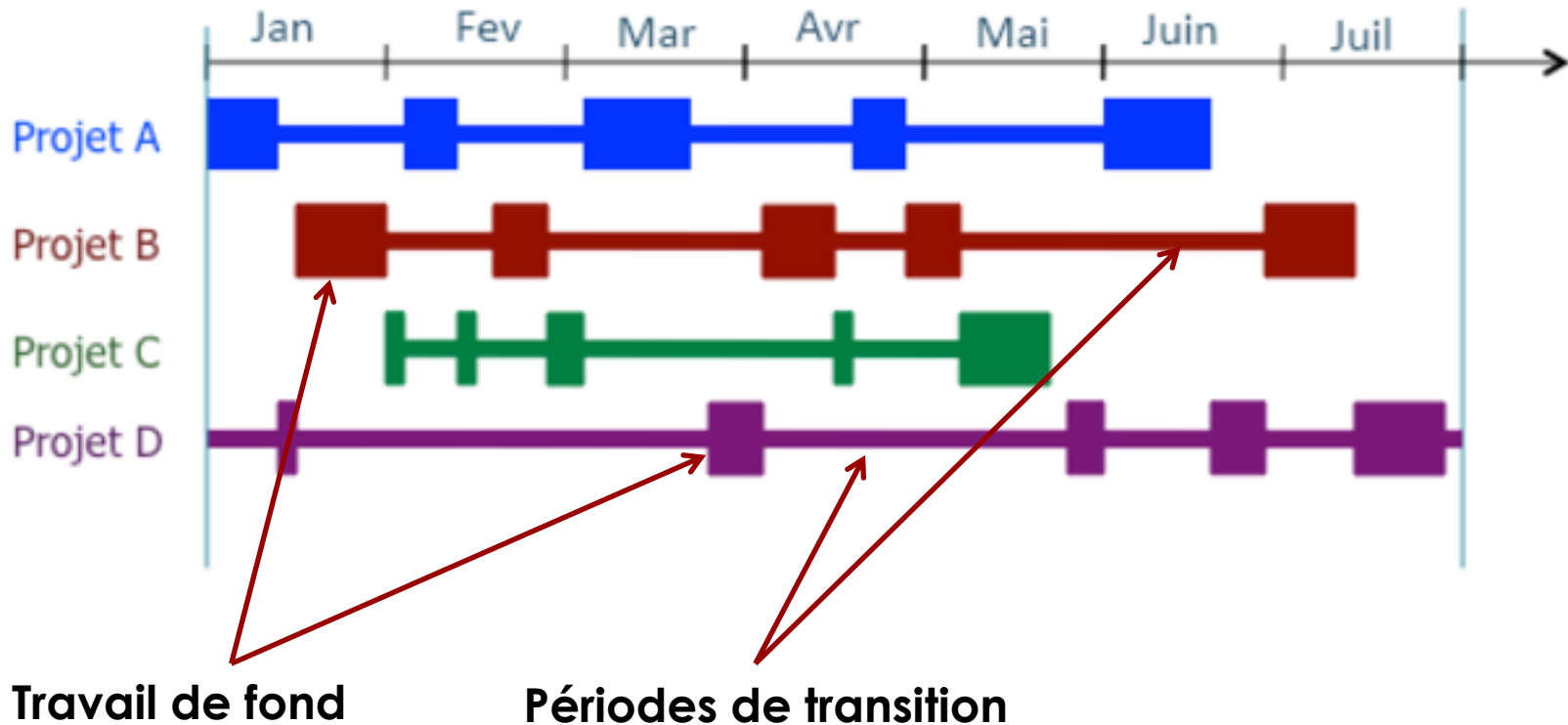
Est-ce parfois une bonne chose ?



- Le multitâche est rarement une bonne chose
- Peut-être sur 2 projets, quand l'un prend est bloqué (en attente d'une ressource externe), alors on avance sur l'autre
 - C'est du faux multi-tâche
- **IMPORTANT** : au moins limiter le WIP à une valeur
 - Ça peut être une autre valeur que 1

Exemple

- Une équipe, 4 projets en parallèle

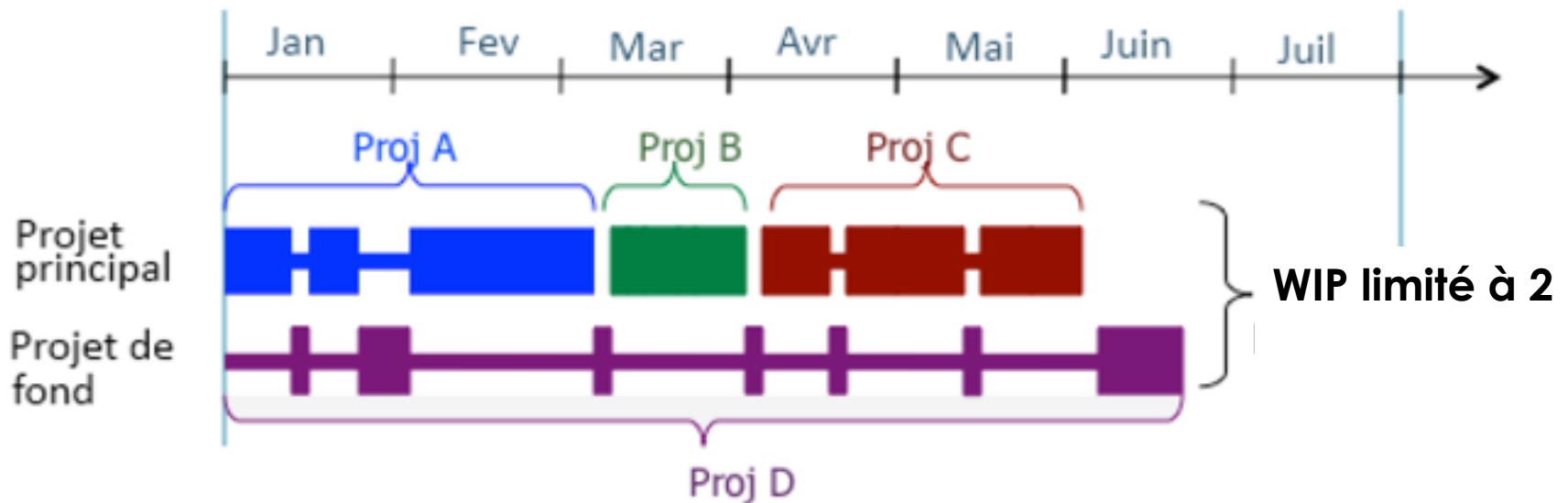


Ici en multi-tâche, et en focalisé ?

En mode focalisé



- On place en « projet de fond » un projet sans plan de travail et sans engagement (ex. projet interne)



Référence



Le jeu original se trouve ici :

- <http://www.crisp.se/henrik.kniberg/multitasking-name-game>