

Search

Menu

- [À propos](#)
- [Votre carrière](#)
- [Orga. personnelle](#)
- [Orga. d'équipe](#)
- [Outils](#)
- [Création d'entreprise](#)
- [Événements](#)
- [Technique](#)

Blogroll

- [43 Folders](#)
- [À propos de la création d'entreprise](#)
- [A smart bear](#)
- [Ask the wizard](#)
- [Créer une micro-startup](#)
- [Des livres à lire](#)
- [Éclaireur.net](#)
- [Entreprise 2.0](#)
- [FineInfo](#)
- [Gestion et versions des sources](#)
- [Jester Management](#)
- [Joel on software](#)
- [L'exemple de Gordon Ramsay](#)
- [Le triangle Qualité, Coût, Délai](#)
- [Les clés de la réussite](#)
- [Les connaissances de base](#)
- [Les erreurs à ne pas faire en entretien](#)
- [Les listes](#)
- [Les méthodes agiles concrètement](#)
- [Les réunions](#)
- [Les spécifications à problème](#)
- [onPK](#)
- [Rands in repose](#)
- [Réduire les goulots d'étranglement](#)
- [Signal vs. Noise](#)

- [Synergeek](#)
- [Web Worker Daily](#)

Blogroll

- [43 Folders](#)
- [À propos de la création d'entreprise](#)
- [A smart bear](#)
- [Ask the wizard](#)
- [Créer une micro-startup](#)
- [Des livres à lire](#)
- [Éclaireur.net](#)
- [Entreprise 2.0](#)
- [FineInfo](#)
- [Gestion et versions des sources](#)
- [Jester Management](#)
- [Joel on software](#)
- [L'exemple de Gordon Ramsay](#)
- [Le triangle Qualité, Coût, Délai](#)
- [Les clés de la réussite](#)
- [Les connaissances de base](#)
- [Les erreurs à ne pas faire en entretien](#)
- [Les listes](#)
- [Les méthodes agiles concrètement](#)
- [Les réunions](#)
- [Les spécifications à problème](#)
- [onPK](#)
- [Rands in repose](#)
- [Réduire les goulots d'étranglement](#)
- [Signal vs. Noise](#)
- [Synergeek](#)
- [Web Worker Daily](#)

Blogroll

- [43 Folders](#)
- [À propos de la création d'entreprise](#)
- [A smart bear](#)
- [Ask the wizard](#)
- [Créer une micro-startup](#)
- [Des livres à lire](#)
- [Éclaireur.net](#)
- [Entreprise 2.0](#)
- [FineInfo](#)

- [Gestion et versions des sources](#)
- [Jester Management](#)
- [Joel on software](#)
- [L'exemple de Gordon Ramsay](#)
- [Le triangle Qualité, Coût, Délai](#)
- [Les clés de la réussite](#)
- [Les connaissances de base](#)
- [Les erreurs à ne pas faire en entretien](#)
- [Les listes](#)
- [Les méthodes agiles concrètement](#)
- [Les réunions](#)
- [Les spécifications à problème](#)
- [onPK](#)
- [Rands in repose](#)
- [Réduire les goulots d'étranglement](#)
- [Signal vs. Noise](#)
- [Synergeek](#)
- [Web Worker Daily](#)

Blogroll

- [43 Folders](#)
- [À propos de la création d'entreprise](#)
- [A smart bear](#)
- [Ask the wizard](#)
- [Créer une micro-startup](#)
- [Des livres à lire](#)
- [Éclaireur.net](#)
- [Entreprise 2.0](#)
- [FineInfo](#)
- [Gestion et versions des sources](#)
- [Jester Management](#)
- [Joel on software](#)
- [L'exemple de Gordon Ramsay](#)
- [Le triangle Qualité, Coût, Délai](#)
- [Les clés de la réussite](#)
- [Les connaissances de base](#)
- [Les erreurs à ne pas faire en entretien](#)
- [Les listes](#)
- [Les méthodes agiles concrètement](#)
- [Les réunions](#)
- [Les spécifications à problème](#)
- [onPK](#)
- [Rands in repose](#)

- [Réduire les goulots d'étranglement](#)
- [Signal vs. Noise](#)
- [Synergeek](#)
- [Web Worker Daily](#)

Méta

- [Connexion](#)
- [Flux RSS des articles](#)
- [RSS des commentaires](#)
- [WordPress.org](#)



[Application concrète des méthodes agiles](#)

mai 12, 2011 at 8:23 by Amaury

Tags: [équipe](#), [méthodes agiles](#), [organisation](#), [réunion](#), [scum](#), [spécifications](#)

J'ai déjà parlé de plusieurs méthodes de travail sur ce blog, depuis le [cycle en V](#) jusqu'aux [méthodes agiles](#) en passant par les [cycles itératifs](#). Pour illustrer tout ça, je vais vous parler de la manière dont nous gérons les projets dans mon entreprise.

Le contexte

Pour situer un peu les choses, il est nécessaire que je vous parle un peu de l'entreprise elle-même.

Fine Media est éditeur de sites Web ; ce n'est pas une «web agency», nous sommes notre propre client. L'entreprise a officiellement commencé son activité en septembre 2007, elle existe donc depuis plus de 3 ans et demi. Nous sommes actuellement une trentaine de personnes.

Les deux premières années se sont déroulées dans un climat classique de start-up à la française, dans un joyeux mélange d'élaboration de process, de développements itératifs rapides, de recrutements stratégiques et de bonne humeur. Le cap des deux ans d'existence et de la vingtaine de personne s'est passé naturellement, l'entreprise quittant d'ailleurs à ce moment ses premier locaux (sous les toits et un peu serrés) pour un immeuble plus confortable.

L'équipe technique a évolué lentement mais sûrement. J'ai immédiatement embauché un administrateur système et une infographiste. Au départ, je réalisais l'intégralité des développements. Avec l'arrivée successive de nos développeurs, j'ai pu me recentrer sur les projets cruciaux. Nous avons désormais une équipe constituée d'un administrateur, de trois développeurs, de deux infographistes/intégrateurs et de moi-même (directeur technique). Cela reste une équipe réduite, mais qui représente quand même un quart des effectifs. Le reste de l'entreprise est constitué de

référéncéurs, de rédacteurs, de community managers, et de commerciaux. Les référenceurs ne sont pas considérés comme faisant partie de l'équipe technique, car ils ont un rôle très transverse, à la fois actif au niveau de la technique et de la rédaction.

Comme je le dis plus haut, nous avons longtemps abordé les développements de la même manière que dans la plupart des start-ups. En privilégiant la réactivité, nous avons réussi à mettre en place des choses relativement complexes dans un laps de temps assez court. Quand il le fallait, j'avais l'autorité nécessaire pour faire des choix structurels, soit pour réclamer plus de stabilité dans les développements, soit pour lancer les [refactoring](#) qui s'imposaient.

Ceux qui suivent le blog savent que la fin de l'année dernière a été une période un peu mouvementée, avec un [énorme refactoring](#). En 4 mois, on a remis à plat et recodé l'ensemble de tout ce qu'on avait développé en 4 ans sur notre plate-forme principale. Le problème avec ce genre de situation, c'est qu'il est très difficile d'établir un plan de marche : On a beau [tronçonner les projets en tous petits bouts](#), il reste toujours des zones d'ombre. Les projets développés durant la «période start-up» manquaient évidemment de spécifications, et cela s'est fait sentir le jour où on a voulu lister toutes les fonctionnalités qu'il fallait remettre au propre.

Ces 4 mois de travail ont donc été réalisés sur un mode «best-effort» qui ne peut pas durer très longtemps. Cela consiste à faire de son mieux, donc à se mettre une pression relativement forte en continu, avec les horaires et le stress qui vont de paire. Au final, cela a généré beaucoup de frustrations dans l'entreprise ; chez les informaticiens qui en faisaient beaucoup sans jamais avoir l'impression que c'était suffisant ; chez les «fonctionnels» qui avaient peur que la refonte prenne un retard incommensurable.

Refonte organisationnelle

Une fois cette période passée, le risque était grand de «rester» sur le même rythme de travail... c'est-à-dire sans revenir à de vraies méthodes de travail. Il était donc important de mettre en place une organisation stable.

J'ai commencé par faire un constat sur les enjeux :

- Choisir la bonne direction => Mieux spécifier
- Maîtriser les coûts => Mieux planifier
- Construire du solide => Vérifier la qualité
- Pérenniser l'équipe => Penser à long terme

D'accord, mais concrètement, ça veut dire quoi ? Plus spécialement, la planification est un miroir aux alouettes : Il ne sert pas à grand-chose de passer 2 heures à établir un planning précis, puisque dès le lendemain il risque d'être chamboulé ; et comme on n'aura pas envie d'y passer 2 heures tous les jours, on finira par laisser le planning à l'abandon. Le problème, c'est que souvent le planning sert de fondement à toute

l'organisation ; un planning qui n'est pas à jour et c'est toute la gestion de projet qui s'écroule.

J'ai donc imposé une organisation. Comme je l'ai déjà exprimé sur ce blog, je pense que [les méthodes ne doivent pas être appliquées à la lettre](#). Il faut plutôt s'inspirer des différents enseignements, utiliser le «jus de cerveau» de ceux qui sont passés par là avant nous, et composer la meilleure recette en fonction des personnes, des habitudes, des projets, ...

Je me suis donc basé sur les quelques bonnes pratiques de [méthodes agiles](#) que nous suivions déjà dans l'entreprise, que j'ai enrichi en utilisant la méthode [Scrum](#).

Le rythme des projets est donné par la durée des [cycles itératifs](#). Cela peut varier suivant les entreprises et les projets, mais habituellement cela tourne entre 3 semaines et un mois et demi. J'ai fait le choix de la simplicité, avec des cycles qui durent un mois, découpé ainsi :

1. **Définition des objectifs.** La première journée de chaque cycle est dédiée à l'analyse technique des projets, à l'évaluation des durées des tâches. On prend la liste des projets en attente, et on tente de définir ceux qui vont être développés pendant ce cycle.
2. **Sprints de développement.** Les deux premières semaines (les trois premières pour les mois un peu longs) sont consacrées aux développements. On garde la notion de sprints hebdomadaire, car il est plus motivant d'avoir des objectifs qui se comptent en jours plutôt qu'en semaines.
3. **Sprint de stabilisation.** Une semaine entière est utilisée à stabiliser les développements. Cela comprend les tests fonctionnels, l'écriture de documentation, les petits refactoring, ou encore la résolution de bugs non critiques qu'on n'a jamais le temps de traiter.
4. **Sprint de déploiement.** La dernière semaine du cycle commence par l'installation des projets en pré-production suivi du déroulement des tests complets et d'éventuels débuggages. Elle se poursuit par la mise en production et la recette, suivie là encore de débuggages s'il le faut.
5. **Évaluation.** La dernière journée du cycle sert à reprendre la liste des tâches qui ont été laissées de côté pendant le mois, pour les intégrer correctement à la liste de priorités du cycle suivant.

Le fait de caler les itérations sur les mois calendaires offre certains avantages. Très simplement, personne ne se demande quand sera la prochaine mise en production (ce qui arriverait avec des cycles de 3 ou 5 semaines, ou avec des cycles de durée variable). De la même manière, le processus d'écriture des spécifications fonctionnelles est simplifié : Si un projet n'est pas spécifié – ou que je n'ai pas validé cette spécification – avant la fin du mois, le projet en question ne sera pas développé le mois suivant.

Supports de suivi

Pour accompagner ces cycles, nous utilisons deux documents de suivi :

- **Une liste des projets.** C'est la vision «macro», qui contient les projets et les “grosses tâches”, triés par ordre de priorité. Ce document a une révision mensuelle et un suivi hebdomadaire. C'est le support de communication transverse entre les différentes équipes (fonctionnelle, technique, design, référencement). Concrètement, il s'agit d'une feuille de calcul qui est maintenue par une seule personne.
- **Une liste des tâches.** C'est la vision «micro», qui contient le découpage fin des tâches à réaliser. Cette liste a une révision hebdomadaire et un suivi quotidien. Nous utilisons pour cela l'outil de gestion de tickets [Flyspray](#) ; les tâches y sont répertoriées, et les détails de spécification sont tenus à jour dans les commentaires de tâches. C'est l'outil principal de l'équipe technique.

Réunions

Sur le plan des réunions, nous avons déjà un rythme efficace, juste équilibre entre la réunionite inutile qui fait perdre du temps et le manque de réunion qui empêche de prendre des décisions. Nous avons juste affiné le process.

Le fonctionnement de base est constitué de 2 types de réunions :

- **Les réunions de suivi** sont faites chaque lundi matin. Y assistent le PDG, le directeur technique (moi) et le chef de projets fonctionnel. Suivant les besoins, nous sommes susceptibles de faire appel à d'autres intervenants. Ces réunions servent à vérifier la bonne avancée des cycles de développement, à apporter des précisions sur des spécifications fonctionnelles, à réfléchir aux futurs développements et à préparer les cycles suivants. Elles durent entre une demi-heure et une heure.
- **Les réunions techniques** sont faites tous les matins. Y assistent les développeurs et intégrateurs. On y discute des spécifications techniques, du détail des implémentations en cours, on “challenge” les idées des uns et des autres. Ces réunions servent aussi à détecter au plus tôt les risques de dérive et de retard, pour prendre rapidement les décisions qui pourraient s'imposer. Elles durent entre 5 et 30 minutes, en fonction des travaux en cours et des besoins de chacun. Si un point doit être discuté plus en profondeur, on le fait par la suite en plus petit comité.

Rôles

Il y a plusieurs rôles-clés dans l'entreprise :

- Le PDG donne les grandes directions et tranche lorsqu'on se trouve dans une situation de blocage.
- Le chef de projets fonctionnel centralise et synthétise les idées d'évolutions,

lance les réunions de travail nécessaires, et est le garant des spécifications fonctionnelles.

- Le directeur technique a un rôle consultatif durant l'écriture des spécifications fonctionnelles (pour faire des arbitrages en fonction de leur complexité technique) et «protège» l'équipe technique en filtrant les demandes impromptues. Il est responsable de la stabilité et de l'évolutivité de la plateforme technique.
- Les développeurs écrivent les spécifications techniques des projets sur lesquels ils vont devoir travailler. Ils révisent les spécifications écrites par les autres membres de l'équipe. Ils doivent [évaluer la durée](#) de leurs développements, comprendre l'équilibre [qualité-coût-délai](#) de chaque projet, mener leurs développements en respectant les normes de l'entreprise, écrire les [tests unitaires](#) et s'assurer de la qualité de leur travail.

J'ai choisi de ne pas employer les termes utilisés par la méthode Scrum, comme “*product owner*” ou “*Scrum master*”, car il aurait fallu prendre le temps de les inculquer à tout le monde. Là, les termes restent génériques, mais les rôles sont clairs pour tous.

Bilan

Nous avons commencé les livraisons mensuelles au début de cette année, nous préparons la cinquième en ce moment même. Pour le moment, le bilan est très positif.

La qualité des projets a été largement améliorée, ce qui n'est pas étonnant dans la mesure où on consacre une semaine entière aux tests et une semaine entière à la validation en pré-production et à la mise en prod.

Les projets sortent avec moins de retard. C'était le plus gros reproche qui m'était opposé quand j'ai présenté la nouvelle organisation ; la moitié du mois étant consacrée à autre chose que du développement, cela voulait forcément dire qu'on allait développer les choses deux fois plus lentement. Et pourtant, ce n'est pas le cas. Cela s'explique par deux choses :

1. Le temps nécessaire à une mise en production est plus ou moins incompressible. Il faut de toute manière mettre les projets en pré-prod, faire des tests fonctionnels, monter en prod, re-valider. Quand on s'autorisait plusieurs MEP par mois, ce temps était multiplié (même si peu de gens s'en rendaient compte). En ne faisant plus qu'une seule MEP par mois, on perd moins de temps à ce niveau-là.
2. Quand il n'y avait pas de deadline fixe sur les projets, il y avait des ajouts incessants qui étaient demandés *en cours de développement*, ce qui est l'une des pires choses qui puisse arriver. Les projets prenaient donc naturellement du retard ; au moment où les demandes d'ajout arrivaient, tout le monde était d'accord pour repousser la date de sortie du projet ; mais au final, il restait une

perception négative du genre «*On n'avait pas dit que ce projet devait sortir il y a 2 semaines ? Si, mais tu as ajouté 3 semaines de boulot entre-temps. Ah oui, c'est vrai, mais c'est quand même chiant d'être en retard...*». Désormais, on s'arrange pour que les projets soient segmentés de manière à sortir des évolutions successives, mois après mois ; tout le monde sait à quoi s'attendre, personne n'est pris au dépourvu.

Tout n'est pas encore parfait. Les spécifications fonctionnelles ont encore du mal à être prêtes à 100% au bon moment. Encore trop souvent, je ne reçois en début de mois qu'un simple *brief* sur lequel il faut passer encore du temps et de l'énergie pour aboutir à une spécification complète. Mais les choses s'améliorent ; nous n'avons plus le cas extrême des spécifications qui nous sont fournies après le développement, ce qui était arrivé quelques fois auparavant.

La pression sur l'équipe technique est différente. Si nous n'avons plus le problème du best-effort (avec la perception que s'il y a du retard c'est que les développeurs ne vont pas assez vite, même s'ils sont les seuls à bosser 10 heures par jour), les mois passent à toute vitesse. Chaque sprint est quand même bien rempli – normal, on n'est pas là pour se tourner les pouces – et on aurait bien besoin d'une journée supplémentaire dans la semaine.

Leave a Comment

Laisser un commentaire

Votre adresse de messagerie ne sera pas publiée. Les champs obligatoires sont indiqués avec *

Commentaire

Nom *

Adresse de contact *

Site web

☒ Notifiez-moi des commentaires à venir via email. Vous pouvez aussi [vous abonner](#) sans commenter.

1 Comment to “Application concrète des méthodes agiles”



[Thomas Turlourat](#)

Structuration tres interessante.

Comme tu le souligne en début d'article, vous êtes votre propre client, du coups vous pouvez vous imposer certaines règles.

Je suppose que même avec des clients extérieurs on doit pouvoir réaliser ce type d'approche, cela necessite peut-être un poste supplémentaire pour le relationnel client ou alors la modification du rôle CDP.

En tout cas, cette organisation est vraiment interessante, cela ne doit pas être là seule non plus, mais elle a le mérite d'être posée et définie. J'adere totalement ! En plus de simplifier la vie de tout le monde, cela donne une valeur qualité supplémentaire à l'entreprise.

Armetiz.info

[View Full Site](#)