

## 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](#)



[Les spécifications à problème \(3\) Spécifications qui arrivent après le développement.](#)

juin 9, 2011 at 9:51 by Amaury

Tags: [développement](#), [spécification](#)

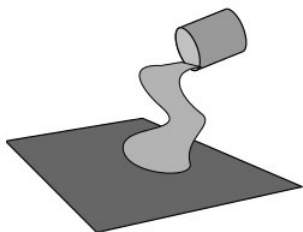
Ce billet est le dernier d'une série de trois articles consacrés aux spécifications et à leurs problèmes. Les deux précédents concernaient :

- [l'absence de spécification](#)
- [les spécifications qui changent en cours de développement](#)

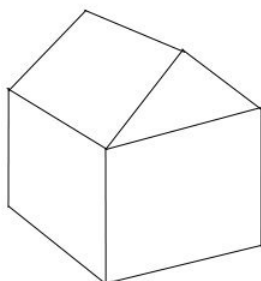
## En image

### Spécifications qui arrivent après le développement

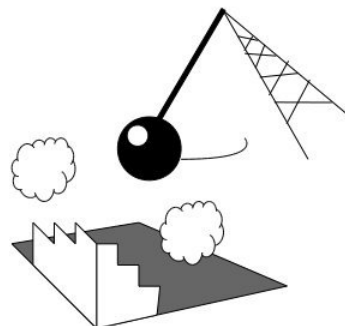
Jour 1 : on coule le béton



Jour 2 : on construit les murs et le toit



Jour 3 : on détruit tout et on recommence



Explication du client : « Je ne vous avais pas dit que je la voulais plus petite ? Pourtant c'est logique, c'est moins cher ! »

## Version texte

- Jour 1 : on coule le béton

- Jour 2 : on construit les murs et le toit
- Jour 3 : on détruit tout et on recommence

Explication du client : « *Je ne vous avais pas dit que je la voulais plus petite ? Pourtant c'est logique, c'est moins cher !* »

## Mon avis

Ce cas est mon préféré (si je puis dire). Parce qu'il est tellement évident que tout le monde pense en être prémuni. Et pourtant, même les meilleures organisations tombent encore dedans de temps en temps.

La principale cause qui amène à cette situation, c'est lorsqu'un planning de développement a été décidé, mais que la spécification n'est pas prête à temps. Par exemple, en basant vos développements sur un [cycle itératif](#), vous connaissez les dates de début des prochaines itérations. Théoriquement, les spécifications fonctionnelles devraient être écrites au préalable, de manière à pouvoir commencer chaque cycle en connaissant précisément le travail qui devra y être accompli.

Malheureusement, personne n'est à l'abri d'un raté dans l'écriture des spécifications. La réflexion peut être plus compliquée que prévu. Il peut y avoir des querelles de clochers, qui font que plusieurs personnes s'affrontent sur des points-clés, ce qui bloque l'avancement des choses. Même en ayant les idées claires, une spécification peut comporter de nombreux éléments (explication textuelle, wireframe, maquettes, storyboard, cahier de test, ...) qui peuvent prendre plus de temps que prévu à délivrer.

Il n'y a pas de solution miracle contre ce problème. C'est dommage, car j'ai vu plusieurs fois des développements être entamés en se basant sur des spécifications incomplètes, sur la seule bonne foi que « on sait où on va, les maquettes ne feront que confirmer tout ça ». Et lorsque les maquettes arrivent, on se rend compte qu'il faut casser une partie importante du travail, pour le refaire autrement. Avec au passage tous les problèmes déjà entrevus dans l'article sur les spécifications qui changent en cours de développement.

Alors que faire ? Deux choses :

1. **Anticiper.** Prévoir les choses plus tôt. Fixer des dates de livraison des spécifications qui ne soit pas la veille du cycle de développement.
2. **Découper.** Comme dit plus haut, une spécification est composée de plusieurs éléments. Il ne faut pas attendre que tout soit terminé pour commencer à passer en revue les éléments terminés. Si une expression de besoin est incorrecte, vous pouvez le détecter rapidement, n'attendez pas que la spec textuelle et les maquettes soient terminés avant d'y mettre votre grain de sel.

# 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.

[View Full Site](#)