

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 \(1\) Pas de spécification](#)

juin 7, 2011 at 1:04 by Amaury

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

Il existe 3 grands types de problèmes avec les spécifications. Il y a quelques mois, j'ai préparé une présentation pour expliquer [notre nouvelle organisation](#) dans mon entreprise. Dans cette présentation, j'abordais les spécifications, en tentant d'expliquer les conséquences négatives qu'il peut y avoir lorsqu'elles ne sont pas préparées correctement.

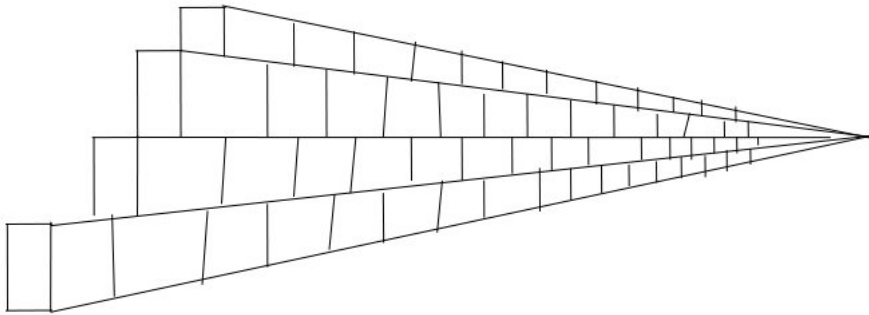
Pour rendre mon propos immédiatement compréhensible, j'ai fait un parallèle avec l'architecture «classique», qui est souvent utilisée pour imager l'architecture informatique. Soyez indulgents quant à la qualité des dessins, ça a été fait avec OpenOffice Impress, qui n'est pas vraiment idéal pour ça (ni aidé par mes talents).

Je vous invite aussi à lire les deux articles suivants de cette série :

- [les spécifications qui changent en cours de développement](#)
- [les spécifications qui arrivent après le développement](#)

En image

Pas de spécification



Explication du maçon : « *Moi, tant qu'on me dit rien, je fais du mur !* »

Version texte

On peut voir un mur de briques qui s'éloigne à l'infini.

Explication du maçon : « *Moi, tant qu'on me dit rien, je fais du mur !* »

Mon avis

Trop souvent – et quel que soit le type d'entreprise dans laquelle j'étais – je me suis retrouvé dans des situations problématiques, simplement parce que les personnes censées réfléchir aux spécifications fonctionnelles ne prenaient pas le temps de penser à tous les cas particuliers de ce qu'ils voulaient mettre en place, ou parce qu'ils estimaient que certaines choses étaient tellement évidentes qu'il n'y avait pas besoin de les exprimer clairement.

Dans le meilleur des cas, l'équipe technique est obligée d'éclaircir ces zones d'ombre, car il est impossible d'écrire du code « flou ». Cela fait alors perdre beaucoup de temps, car il faut faire des allers-retours avec le client, pour déterminer précisément le périmètre de son besoin. Et si le besoin n'a pas été suffisamment bien pensé (s'il l'avait été, la spécification aurait été faite), on perd encore plus de temps.

Dans le pire des cas, ce sont carrément des fonctionnalités entières qui ne sont pas développées. Vous connaissez l'exemple le plus débile que j'ai vu dans le genre ? Un site qui a été développé sans fonctionnalité « Mot de passe oublié ». Les fonctionnels n'en ont pas parlé parce que ça leur semblait aller de soi. Les développeurs avaient le nez dans le guidon, submergés de boulot qu'ils étaient, et se sont contentés de développer ce qui a été spécifié. Évidemment, ce genre d'erreur n'est découverte qu'après la mise en production finale...

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.

7 Comments to “Les spécifications à problème (1) Pas de spécification”



[Eric](#)

Pour l'exemple du mot de passe, c'est plutôt un problème lié au contrat plutôt qu'à la spécification.

S'il n'y a pas de « contrat » à discuter, alors :

- en toute bonne foi, le fonctionnel n'a pas forcément à penser à ce genre de détail
- en toute bonne foi, le développeur peut également l'oublier (c'est quand même une belle erreur)
- le développeur doit pouvoir intégrer facilement et rapidement la fonctionnalité (sinon, ce n'est pas un bon développeur)

Bon, c'est bien sûr dans le monde idéal des bisounours qui travaille sans contrat. Mais

je ne pense pas que cet oubli dans les spécifications soit un problème d'un point de vue uniquement technique.

Je pense que le code EST la spécification, et donc, que le code aide à trouver ce type de détail qu'il n'est pas possible de prévoir, car un humain, aussi intelligent soit-il n'est pas un compilateur et ne sais donc pas réfléchir et parler de manière formelle. Par contre, il sait coder de manière formelle.



Amaury

Mmh... oui et non.

Pour commencer, le but est de satisfaire le client, plutôt que de simplement remplir le contrat. Le propos des [méthodes agiles](#) n'est pas ailleurs.

Dans l'exemple que je donnais, l'entreprise étant son propre client, le chef de projet fonctionnel joue le rôle du client ; il doit donc être capable de lister les fonctionnalités qu'il attend. Et c'est vrai, tout le monde est de bonne foi. Mais cela reste un vrai problème, parce qu'au final le résultat qui est mis en production n'est pas satisfaisant. Et même avec des cycles itératifs, on se retrouve vite à devoir attendre au moins 2 semaines avant la mise en prod suivante...

D'ailleurs, si je disais qu'il s'agit d'un exemple débile, c'est bien parce qu'il s'agit d'un petit truc, tellement évident pour tout le monde que personne n'y pense. C'est trop con, mais ce n'est pas une excuse.

Les vrais problèmes se posent lorsqu'il y a des développements importants, représentant plusieurs jours/homme ou plusieurs semaines/homme, qui sont laissés sans spec. Est-ce alors aux développeurs de remplir les blancs ? Évidemment que non, sinon ce serait eux qui décideraient du résultat à la place du client (qu'il soit interne ou externe). Difficile à justifier.

C'est la raison pour laquelle je ne pense pas que le code == la spec. Si personne n'est capable d'exprimer clairement ce qui doit être fait, autant ne rien faire car le résultat sera forcément insatisfaisant.

L'analogie avec le BTP reste valable. Si je veux me faire construire une maison, je vais passer par un architecte, à qui je vais expliquer mes attentes ; il les transmettra sous forme d'instructions aux différents corps de métier qui interviendront sur le chantier. Parce que si je me contente d'embaucher un maçon, un plombier et un électricien, et que je leur dis simplement «*Construisez-moi la maison de mes rêves*», il ne faudra pas que je vienne pleurer...



Grunk

Le gros avantage de la non spécification (et je le vie à peut près tous les jours à mon grand désespoir) c'est que le client à la fâcheuse manie de venir et dire « Ha non non , j'ai jamais demander ça moi » ... A quoi on à bien envie de lui répondre qu'il n'a jamais rien demander ...



Amaury

@Grunk : C'est tellement vrai ! 😊



Pierre-Yves

Il y a tant de développeurs qui trouvent que les spécifications ne servent à rien . Et bien souvent je n 'arrive pas à leur faire comprendre que c'est indispensable . Sans les spécifications, cela donne des applications bancales . Evidement ce n'est que trop tard qu'il s'en aperçoivent mais à ce moment , je ne suis déjà plus dans l'entreprise ou j'ai averti depuis longtemps que j'arrêtais le projet.C'est navrant !



Nico

Personnellement, si le client ne rédige pas de spécifications, je le fais à sa place puis je le fais valider AVANT de commencer estimation/planification/développement.
Mon supérieur hiérarchique direct est d'accord avec ce principe et cela nous sauve bien souvent 😊