

Alors que je lisais [cet article](#) sur les méthodes agiles, j'en viens à me poser quelques questions sur 2 des termes généraux employés :

- Communication (les individus et leurs interactions)
- Des fonctionnalités opérationnelles (plutôt que de la documentation exhaustive)

Mes questions par rapport à ces affirmations :

- Qu'est-ce que la communication ?
- Qu'est-ce que de la documentation exhaustive ?

Dans ce que je comprends, communication veut dire communication verbale exclusivement. Pourtant communiquer n'est un acte de parole mais aussi, et entre autres, d'écrit. Pour moi communiquer efficacement, signifie faire passer un message compréhensible et s'assurer que le message est compris pour tous les participants et surtout compris de la même manière.

Prenons un exemple :

Jean : "Tu mets du bleu sur ce mur-là"

Pierre : "Ok je le mets en bleu"

Jean revient, et le mur est en bleu roi.

Jean : "Mais non ! Pas le bleu roi mais le bleu pastel"

Voilà une situation qui arrive régulièrement lorsqu'on utilise le langage verbal et non écrit.

L'écrit prend plus de temps, mais ce temps là est mis à profit dans la réflexion de l'expression de nos pensées et il est probable que si Jean avait écrit le message, il l'aurait relu et la précision serait venue d'elle-même.

Le résultat aurait été que Pierre n'aurait dû repeindre en bleu pastel et l'inévitable "Je te l'avais dit", "Mais non", "Mais si" n'aurait pas eu lieu.

La plupart du temps les choses simples ne requièrent que quelques mots à écrire et le gain de temps est immédiat. L'autre gain à plus long terme est lorsqu'on doit revenir sur un détail ou une fonctionnalité quelques semaines, mois, années plus tard. On sait qui a participé à la discussion et l'on peut demander des précisions ou simplement se souvenir.

Et maintenant que veut mesurer le mot "exhaustive" dans "documentation exhaustive" ? C'est étrange ce mot dans un projet. Je pense que exhaustif devrait être remplacé soit par inutile ou soit par "auto-parlante".

Ces mots sont encore plus étranges, me direz-vous ?

Par inutile, j'entends par exemple la création de javadoc sur les getter et les setter ou encore sur des méthodes qui portent un nom explicites, du genre "findUserById". Dans ce cas la documentation est inutile car "auto-parlante".

Il en va de même pour les use cases. La plupart du temps ils peuvent être décrit textuellement et la création de diagramme de séquence, d'état ou autre est totalement inutile. Il est inutile de décrire par un diagramme 3 interactions entre 2 objets. Par contre, si la fonctionnalité est complexe, certains diagrammes sont très utile pour aider à sa compréhension. Dans ce cas la communication écrite est plus importante que la communication verbale.

Pour revenir une dernière fois sur la partie "les individus et leurs interactions", je comprends que les outils sont à éviter. Pourtant pour communiquer sur le long terme, des outils de communications sont indispensables car ils sont le garant de la mémoire du projet et l'on ne dépend plus de la mémoire des individus qui, comme on le sait, vont de projet en projet.

Dans un projet, ce qui est réellement important c'est le projet lui-même. Il a son propre écosystème. Sans lui, rien n'existe.

Le projet devient alors le produit d'individus, de la communication passée, présente et future, des outils et des technologies employés et de tout ce que j'oublie.

Bref il a sa propre existence et doit être considéré comme un tout. Il utilisera sa propre méthodologie, composées de plusieurs éléments de méthodologie qui semble adapté au besoin.

Je pense que les personnes clefs sont les architectes de l'écosystème du projet.

Pour moi, présenter une méthode Agile à une personne qui n'en a pas la connaissance mais veut l'implémenter sur ses projets est extrêmement dangereux s'il n'en a pas l'expérience et peut être tout aussi coûteux, voire plus, qu'un projet sur base de RUP par exemple, qui lui veut garder chaque minute écrite.

Une méthode agile marchera sur des projets à courts termes mais pas sur des projets qui demandent un investissement sur le long terme.

J'ai ouvert le débat sur le Google Groups des CastCodeurs, [et c'est par là](#) .

[Joomla SEF URLs by Artio](#)