

## Agilité et critiques courantes

Emmanuel CHENU  
[emmanuel.chenu@gmail.com](mailto:emmanuel.chenu@gmail.com)  
<http://emmanuelchenu.blogspot.com/>

3ème édition, 17/01/09

Concernant les démarches agiles de développement logiciel, j'entends souvent les critiques suivantes:

1. *Ces démarches manquent de rigueur et de discipline.*
2. *Ces démarches abolissent la conception.*
3. *Ces démarches n'offrent pas de vision à long terme du déroulement du projet.*
4. *Ces démarches engendrent une reprise perpétuelle du travail fini.*
5. *Ces démarches ne permettent pas de s'engager sur un contour fonctionnel fixe.*
6. *Ces démarches ne sont pas appropriées pour les produits critiques.*
7. *Ces démarches échouent comme les autres.*
8. *Ces démarches sont inappropriées pour le développement géographiquement distribué.*
9. *Ces démarches sont inappropriées aux grosses équipes.*
10. *Ces démarches sont appropriées pour des développeurs compétents et motivés.*
11. *Ces démarches risquent d'élaborer des architectures non-évolutives.*
12. *Ces démarches vivent sur l'utopie du consensus dans l'équipe.*

A l'occasion de cours, de rencontres, de conférences, de présentations et du travail au quotidien, je suis souvent confronté à ces critiques. Je suis alors amené à argumenter à leur encontre ou à en atténuer la force.

Voici une tentative de mettre par écrit ces argumentaires afin de les réutiliser plus aisément. Les publier permettra peut-être de les enrichir à partir de l'expérience d'autres praticiens.

### **1. Ces démarches manquent de rigueur et de discipline.**

Parce que les agilistes<sup>1</sup> accordent plus de valeur à l'interaction avec les personnes qu'aux processus et outils, on peut penser qu'ils négligent les processus et les outils.

Parce que les agilistes accordent plus de valeur à un produit opérationnel qu'à une documentation exhaustive, on peut penser qu'ils négligent la documentation.

Parce que les agilistes accordent plus de valeur à la collaboration avec le client qu'à la négociation de contrat, on peut penser qu'ils négligent les engagements.

Parce que les agilistes accordent plus de valeur à la réactivité face au changement qu'au suivi d'un plan, on peut penser qu'ils négligent la planification.

Certes, des développeurs qui négligeraient les processus, les outils, la documentation, les engagements et la planification manqueraient de rigueur et de discipline dans leur démarche.

Par contre, accorder plus de valeur à une seconde activité plutôt qu'à une première ne veut en rien dire que la première est négligée. La valeur de la première est admise. Simplement, davantage de valeur est accordée à la seconde.

---

1 Praticien du développement agile.

De nombreuses pratiques agiles imposent une très grande rigueur et beaucoup de discipline à l'équipe. Voici une liste de pratiques disciplinées et rigoureuses:

- Chaque incrément doit être potentiellement mis en production.
- Chaque incrément de logiciel est démontré au client.
- Une version opérationnelle du logiciel est disponible à tout moment. Elle incorpore les derniers développements.
- Toute ligne de code est écrite à deux devant un même ordinateur.
- Le restant à faire est identifié, priorisé par le client et estimé par l'équipe.
- Des tests d'acceptation systématiques mesurent l'avancement du développement.
- L'équipe se réunit brièvement tous les jours pour statuer sur l'avancement, le restant à faire et les problèmes.
- Un test qui échoue doit être écrit avant d'écrire ou modifier toute ligne de code.

**Ainsi, je pense qu'il est plus exact d'affirmer qu'une démarche agile est particulièrement rigoureuse et disciplinée.**

## **2. Ces démarches abolissent la conception.**

Parce que l'architecture n'est pas exhaustivement pensée préalablement à l'implémentation, parce que celle-ci est émergente et parce qu'elle est continuellement améliorée, on peut penser que l'activité d'architecture est éliminée des développements agiles. De plus le rôle d'architecte n'est pas reconnu dans la communauté agile. Tout développeur est architecte.

Certes, la phase de conception est bien éliminée. Néanmoins, l'activité de conception est omniprésente puisque menée en continue et conjointement au test.

Concernant la conception, les agilistes font les postulats suivants:

- L'architecture évolue au cours d'un développement. Il est humainement impossible de concevoir exhaustivement et correctement préalablement à l'implémentation et au test.
- Une architecture simple est évolutive.
- Une architecture est adéquate si elle est simple, testable et testée.

**Ainsi, je pense qu'il est plus exact d'affirmer que l'architecture est omniprésente dans un développement logiciel agile.**

## **3. Ces démarches n'offrent pas de vision à long terme du déroulement du projet.**

Parce que les agilistes accordent plus de valeur à la réactivité face au changement qu'au suivi d'un plan, on peut penser qu'ils négligent la planification. Parce que les fonctionnalités détaillées du logiciel ne sont pas exhaustivement spécifiées préalablement à l'implémentation, on peut penser que le développement avance à l'aveugle.

Concernant la planification, les agilistes font les postulats suivants:

- La planification évolue au cours du temps. Il est humainement impossible de planifier exhaustivement et de manière pertinente préalablement au développement.
- La planification est grossière au départ et s'affine au cours du développement en se basant sur des mesures.

Voici quelques pratiques agiles de planification qui découlent de ces postulats:

- Dès le début du projet, le restant à faire est identifié, priorisé et estimé. Seul le restant à faire prioritaire est précisément décrit et précisément estimé.
- Le restant à faire est régulièrement actualisé en faisant intervenir les besoins actuels du

- client.
- Le développement est itératif et incrémental pour collecter des mesures d'avancement qui servent à planifier.
- La planification est continuellement affinée sur la base de mesures collectées.
- L'avancement et le restant à faire sont actualisés quotidiennement lors de courtes réunions d'équipe.

**Ainsi, je pense qu'il est plus correct d'affirmer que les démarches agiles se basent sur une planification tôt et en continue qui gagne en pertinence alors que le développement avance.**

#### **4. Ces démarches engendrent une reprise perpétuelle du travail fini.**

Parce que le développement est itératif et que le code est remanié pour être amélioré on peut penser que rien n'est jamais fini.

Il ne faut pas oublier que la démarche est itérative **et incrémentale**. Certes, les développeurs vont réitérer les activités de développement, mais à chaque fois pour un nouvel incrément de fonctionnalité. Puisque toutes les activités de développement sont menées par incrément de fonctionnalité, il s'agit bien de maximiser la quantité de travail réellement terminé. L'objectif est bien que chaque incrément de fonctionnalité soit potentiellement mis en production et donc terminé.

Par contre, une éventuelle reprise de fonctionnalité déjà existante et le remaniement de l'architecture sont inévitables si on est convaincu que les besoins d'un client évoluent au cours du développement et qu'on ne peut exhaustivement concevoir de manière pertinente préalablement à l'implémentation et au test.

**Ainsi, je pense qu'il est plus correct d'affirmer que les démarches agiles maximisent la quantité de travail terminé.**

#### **5. Ces démarches ne permettent pas de s'engager sur un contour fonctionnel fixe.**

Parce que parmi les variables de pilotage projet (coût, date, qualité et contour) c'est le contour qui est utilisé comme variable d'ajustement, on peut penser qu'une démarche agile ne garantit pas un contour au client. Pour obtenir une garantie de contour, le client doit s'attendre à ce que la durée du développement s'allonge.

Les agilistes font le postulat que la qualité n'est pas une variable d'ajustement pour le pilotage d'un projet. En d'autres termes, réduire la qualité en dessous du niveau nécessaire attendu par le client et estimé par l'équipe met systématiquement en péril la réussite du projet. Si le contour attendu est fixe, alors la date de livraison finale est la variable d'ajustement du projet. Par contre, si la date reste fixe, alors le contour fonctionnel devient la variable d'ajustement du projet. Il s'agit alors de s'assurer que le contour prioritaire est garanti. Justement, la démarche de construction itérative et incrémentale pilotée par les priorités du client assure que le logiciel ne contient que ce qui lui apporte le plus de valeur.

**Ainsi, je pense qu'il est plus correcte d'affirmer qu'à coût, date et qualité fixes, une démarche agile garanti un contour prioritaire.**

## **6. Ces démarches ont inappropriées pour produits critiques.**

Parce que les agilistes accordent plus de valeur à l'interaction avec les personnes qu'aux processus et outils, on peut penser qu'ils négligent les processus et les outils.

Parce que les agilistes accordent plus de valeur à un produit opérationnel qu'à une documentation exhaustive, on peut penser qu'ils négligent la documentation.

Parce que les processus, les outils et la documentation vont de pair avec les produits certifiés et parce que les démarches agiles passent parfois pour peu rigoureuses et peu disciplinées, on peut penser qu'elles ne sont pas appropriées pour le développement de produits critiques.

Pour ce qui est de la rigueur et de la discipline, je vous propose de vous référer à l'argumentaire du premier point.

Pour le reste, les certifications des produits critiques assurent, entre autres, que le produit contient exclusivement du code implémentant des besoins opérationnels. Il se trouve que dans une démarche par incrément pilotée par des scénarios opérationnels, seul le code répondant à ces exigences est développé.

Les audits vérifient que tous les besoins opérationnels sont implémentés et vérifiés par test. Il se trouve que le pilotage par les tests impose des tests d'acceptation pour tous les scénarios opérationnels.

Les certificateurs s'assurent que tout le code est testé. La pratique du « test first », où le code n'est écrit que pour faire passer un test écrit antérieurement, garantit que l'intégralité du code est systématiquement couvert par test.

Les auditeurs vérifient que tous les produits du développement ont été revus. La pratique du travail en binôme impose une revue en continue et systématique de tout le code et des tests.

Les certifications s'assurent que les tests vérifiant les besoins opérationnels n'ont pas été écrits en regard du code. La pratique du « test first » assure cette indépendance puisque le test est rédigé avant le code qu'il couvre.

Alors, si les démarches agiles sont adaptées pour fournir les preuves et la documentation requises pour les audits de certification, alors les démarches agiles sont appropriées au développement de produits certifiés.

**Ainsi, je pense qu'il est plus correcte d'affirmer qu'à condition d'être adaptées, les démarches agiles sont une aide au développement de produits critiques.**

## **7. Ces démarches échouent comme les autres.**

Parce que de nombreux projets de développement logiciel qui ont adopté une démarche agile en général et Scrum en particulier échouent, on peut penser que ces méthodes sont aussi inefficaces que les autres.

Il faut savoir que Scrum est une démarche généraliste de développement de produit non spécifiquement à base de logiciels. Cette démarche ne comprend aucune pratique technique.

Or, il est impossible d'élaborer avec succès un produit technique comme un logiciel sans l'application de pratiques techniques. La pratique de Scrum pour le développement de logiciels doit alors impérativement être accompagnée de pratiques techniques de développement logiciel tel qu'on en trouve dans l'Extreme-Programming par exemple. D'ailleurs, on oublie souvent et vite que Scrum impose que tout incrément de produit soit potentiellement livrable, c'est à dire apte à être mis en production. Or, il est déraisonnable de mettre en production un produit qui n'a pas été exhaustivement testé. Ainsi, Scrum sous-entend l'adoption de pratiques techniques tels que le test, pourvu qu'on veuille bien entendre.

Plus généralement, les projets de développement de logiciels échouent à cause d'une mauvaise

transmission de l'information, à cause d'une peur de l'échec qui pousse à vouloir dompter la complexité du problème en son entier et de manière prédictive, à cause d'un manque de rigueur et de discipline, et à cause d'un travail d'équipe détérioré par l'individualisme. Ces causes d'échec sont toutes liées à la nature humaine et ses faiblesses.

Les démarches agiles prennent compte de ces problèmes humains. Elles poussent très haut la communication et accordent une très grande importance à l'esprit d'équipe grâce à l'espace commun de travail, au travail en binômes qui alternent, aux équipes pluridisciplinaires, aux courtes réunions quotidiennes d'avancement et aux réunions de planification et d'amélioration. Les démarches agiles incitent au morcellement de la complexité du problème par une succession de constructions partielles mais concrètes et par la recherche constante de la simplicité. La rigueur et la discipline dans la démarche de travail sont incitées par le test et la relecture systématiques et par l'intransigeance sur le niveau de qualité.

**Ainsi, je pense qu'il est plus correct d'affirmer que comme toute entreprise humaine exigeant un travail d'équipe, les démarches agiles sont faillibles. Par contre, elles elles présentent l'avantage de tenir compte de la nature humaine et ses faiblesses.**

#### **8. Ces démarches sont inappropriées pour le développement géographiquement distribué.**

Parce que les démarches agiles promeuvent la co-localisation de l'équipe, le travail en binômes qui alternent et la communication orale en face à face, on peut penser qu'elles sont peu adaptées pour le développement géographiquement distribué.

Soyons honnêtes, toute démarche de travail en équipe perd en efficacité à partir du moment que l'équipe est géographiquement éclatée. Les démarches agiles n'échappent bien sûr pas à cette difficulté, comme tout autre méthode de développement. Dans ces cadres de développements logiciels géographiquement distribués, la perte d'efficacité est due à la perte de communication, à la perte d'esprit d'équipe et aux désynchronisations.

Les démarches agiles accordent une attention toute particulière à la communication et à l'esprit d'équipe. Il s'agit alors de trouver des solutions applicables dans le cadre d'une dispersion géographique. Les désynchronisations sont limitées par la pratique de l'intégration continue et de la courte réunion quotidienne d'avancement.

**Ainsi, je pense qu'il est plus correct d'affirmer que comme toute démarche de développement, les méthodes agiles souffrent de la dispersion géographique. Par contre, elles sont plus robustes aux inconvénients engendrés.**

#### **9. Ces démarches sont inappropriées pour les grosses équipes.**

Parce que les démarches agiles promeuvent la co-localisation de l'équipe et la communication orale en face à face, on peut penser qu'elles sont peu adaptées dans le cadre de grosses équipes.

Soyons honnêtes, toute démarche de travail en équipe perd en efficacité alors que l'équipe dépasse en nombre un certain seuil critique. Les démarches agiles n'échappent bien sûr pas à cette difficulté, comme n'importe quelle autre méthode de développement. Pour les équipes nombreuses, la perte d'efficacité est due à la perte de communication, à la perte d'homogénéité et aux désynchronisations.

Les démarches agiles accordent une attention toute particulière à la communication. Une haute bande passante de communication pourra être maintenue par une organisation fractale en petites équipes. L'homogénéité peut être assurée par le travail en binômes qui permutent. Les désynchronisations sont limitées par la pratique de l'intégration continue et des courtes réunions quotidiennes d'avancement.

**Ainsi, je pense qu'il est plus correct d'affirmer que comme toute démarche de développement, les méthodes agiles perdent en efficacité pour de grandes équipes. Par contre, elles sont plus robustes aux inconvénients engendrés.**

#### **10. Ces démarches sont appropriées pour des développeurs compétents et motivés.**

Les démarches agiles recommandent de bâtir les projets autour de personnes motivées et d'accorder une attention continue à l'excellence technique et à la qualité. Pour les agilistes, les meilleures architectures, spécifications et conceptions sont issues d'équipes qui s'auto-organisent. Enfin, à intervalle régulier, les équipes réfléchissent aux moyens de devenir plus efficaces, puis accordent et ajustent leurs comportements dans ce sens.

C'est pourquoi on peut penser que les démarches agiles sont surtout adaptées pour des développeurs motivés et compétents.

Soyons honnêtes, toute démarche de développement est plus efficace avec des développeurs motivés et compétents. Mais, il est exact qu'il faut du courage pour travailler avec le client, pour ne jamais transiger avec le niveau de qualité et pour communiquer en toute transparence.

A leur avantage, les démarches agiles promeuvent la formation accélérée des nouveaux équipiers par le travail en binômes qui alternent, par l'espace partagé et par l'équipe pluridisciplinaire. Les démarches agiles accordent le droit à l'erreur en limitant l'impact des erreurs par le morcellement du développement en une succession de constructions partielles et concrètes, par le test systématique, par la relecture systématique dans le cadre du travail en binômes et par la pratique de l'intégration continue. L'entraide est incitée par le travail en binômes qui alternent. La motivation est animée par la satisfaction du travail accompli et de qualité.

L'excellence technique n'est pas pré requise, mais construite par une attention continue comme dans la pratique du remaniement.

**Ainsi, je pense qu'il est plus correct d'affirmer que les démarches agiles aident à élever la compétence et la motivation des développeurs.**

#### **11. Ces démarches risquent d'élaborer des architectures non-évolutives.**

Parce que le développement itératif et incrémental ne tient compte que des besoins prioritaires et actuels, les architectures retenues risquent de ne pas pouvoir accepter la prise en compte des besoins de demain.

Les agilistes partent du postulat qu'il est humainement impossible de concevoir exhaustivement et correctement préalablement à l'implémentation et au test. D'ailleurs, tous ceux qui ont essayé ont malgré tout modifié leur architecture en cours de développement. Ainsi, une architecture conçue en avance de phase ne permet pas la prise en compte de tous les besoins.

Plutôt que dépenser de l'énergie à mal anticiper le futur, les démarches agiles misent sur une construction incrémentale qui garantit l'implémentation des besoins prioritaires. Les priorités peuvent être attribuées à partir de critères de retour sur investissement, de risques techniques ou d'impact sur l'architecture. Pour les incréments suivants, les démarches agiles assurent une évolution souple de l'architecture en retenant toujours la solution la plus simple qui soit viable et en systématisant le test. Ainsi, l'architecture pourra évoluer sans craindre les régressions.

**Ainsi, je pense qu'il est plus correct d'affirmer que les démarches agiles conduisent à des architectures évolutives qui acceptent les besoins futurs.**

### ***12. Ces démarches vivent sur l'utopie du consensus dans l'équipe.***

Pour les agilistes, les meilleures architectures, spécifications et conceptions sont issues d'équipes qui s'auto-organisent. À intervalle régulier, les équipes réfléchissent aux moyens de devenir plus efficaces, puis accordent et ajustent leurs comportements dans ce sens. Ainsi, l'efficacité d'une équipe réside en grande partie en sa capacité à établir et maintenir le consensus. Pour certains, le consensus au sein d'une équipe travaillant pour une organisation dont le but est de générer des profits est une utopie et il est dangereux d'y accorder une si grande importance.

Je n'ai pas de réponse objective à cette critique. Je pense qu'il en va de sa foi en l'homme. Pense t-on que les agissements de l'homme convergent vers son bien ou sa perte? En tout cas, la contrainte et la peur n'ont jamais convergé vers des victoires belles et durables.

Cette liste de critiques n'est pas exhaustive. Il ne s'agit que de celles face auxquelles j'ai été confronté à plusieurs reprises. Les argumentaires de défense ne sont que des points de vues purement personnels issus de mon expérience, mais enrichis par le parcours d'autres praticiens. Les publier permettra peut-être de les compléter à partir l'expérience d'autres agilistes.