



Approche Agile, pour et contre

Ici, pas de vision Scrummaster ou Dev, mais un regard externe, manager ou auditeur de projet IT

Sept Arguments Pour l'approche Agile

1. Une application opérationnelle vite disponible et offrant tôt de la valeur Client. En premier lieu, la priorisation du backlog suit l'ordre décroissant de la valeur Client. De plus, l'application, passant en continu des tests, est sans cesse corrigée et améliorée.
2. Une application avec une vision Métier et non IT : facilité d'utilisation (définition par le métier) et richesses des cas prévus (user stories). Plus qu'une approche IT, trop souvent exclusivement en Top-Down global amenant à une présentation sur le même plan des cas courants et des cas rares (modèle fonctionnel SA). Les Dev garantissent « naturellement » l'homogénéité de l'ergonomie et les règles sécurité des profils, d'autant plus que le représentant du client (product owner) peut clairement expliciter ces règles dans les critères d'acceptation des livraisons.
3. Meilleure définition des fonctionnalités. L'approche Agile permet et dépasse l'IKIWISI (I'll know it when i see it – Attention : pas au sens United States Supreme Court) mais avec un risque mineur et gérable, trop de détails sur les fonctions prioritaires et pas assez de travail (temps) sur les autres.
4. Efficacité des développements : une équipe autoorganisée se basant sur les propositions de développeurs (solutions, délais) et donc leur engagement à faire.
5. Coût et délai maîtrisés. Toujours respecter « si j'ajoute, j'enlève » et pas d'« avenant ESN sur change management ». Les fonctionnalités non implantées sont normalement de plus faible valeur-client. Il est donc possible d'arrêter, avec un logiciel incomplet, mais qui plante les fonctionnalités les plus importantes pour le client. *Reste l'estimation initiale de la productivité (~vélocité) de l'équipe comme facteur de sur-estimation du faisable. Et seconde embuche à cette productivité, l'accord au sein de l'équipe, des cultures différentes à imbriquer.*
6. Des tests continus et rejouables. Le rejeu de tous les tests sur chaque sprint assure naturellement la non-régression. *Un plus, des Tests Unitaires (sur un Framework de type Junit) et pas seulement des tests fonctionnels*
7. Le projet peut être arrêté à tout moment avec un résultat connu, opérationnel et stable. Le coût de démobilisation de l'équipe subsiste

Sept Arguments contre l'approche Agile

1. On ne peut pas être agile sur tous les projets d'une DSI : Projets de mise en place ou refonte d'infrastructure, migration dans le cloud, montées de version éditeurs, développement de softs de base (OS, drivers, middleware,...), applications industrielles temps réel, évolutions réglementaires sur softs anciens, Convergence de SI suite à fusion, mise en place d'ERP (méthode éditeur ou intégrateur), mise en place de SaaS, Projets Sécurité, finalisation RGPD,



projets Qualité (déploiement de Process DSI), projets atypiques : décommissionnement, migration des données, conversion/ portage de systèmes, [Note : cela représente environ 40 à 70% de l'effort Projet d'une DSI]. Pourquoi ? Projets purement IT (sans le Métier), projets exclusivement techniques (basés sur l'architecture de la solution, la mise en service de technologies, les modifications des composants du SI, ...) et non sur la progressivité de la réalisation des fonctionnalités. L'Agile : Surtout les projets de développement d'applications Métier, Mise en place de progiciels, développement d'applications BtoC.

2. Disponibilité et aptitude du métier : La disponibilité est d'abord le fait d'un management Métier qui n'investit pas suffisamment sur la conception de ses outils de travail et donc d'outils assurant une meilleure performance. De plus la disponibilité des Utilisateurs Métier est difficile à assurer sur des projets longs (plus de 9 mois). Ils doivent aussi apprendre à travailler dans le cadre d'une culture et d'une organisation différente, des règles et des relations Projet Agile souvent très différentes des organisations classique Métier.
3. Difficultés pour obtenir un système de bonne qualité : Le design du code et de la base de données est fait par ajouts/modifications successifs et non par une conception d'ensemble. Un Code continuellement augmenté, sans reconception d'architecture, donc une dette technique possiblement importante. *La pratique d'itérations (sprints) sans avancement, uniquement centré sur le refactoring, les approches software craftsmanship sont parfois utilisées. Des outils classiques peuvent aussi être employés : framework de test unitaire, Couverture des tests, mesures qualité (identification des codes dupliqués, complexité, ...).*
4. La documentation d'exploitation et de maintenance peut être absente ou très faible : ce n'est pas la priorité des Dev. Leur mise à jour sur chaque sprint est un effort souvent « non prévu »
5. Les très grands projets, qui demandent des équipes de développement importantes, sont difficiles en Agile ; ils devraient être restructurés en programme, plusieurs projets coordonnés, dont certains en mode Agile (Attention un programme ne se mène pas comme un projet). Des approches Agile (Safe, Spotify) peuvent répondre à cette problématique, mais leur mise en place est une démarche d'entreprise et non de projet.
6. Un projet agile exige une adaptation de la DSI : Production ne sachant pas faire des mises en prod rapide. Direction des études sans outils, mesures et process de management de l'agilité.
7. Attention un projet peut demander un travail exploratoire en phase d'avant-projet avec prototypage et allers-retours. Pourquoi ne pas commencer par un design thinking pour mieux cerner le sujet dès le début et démarrer sur un backlog de qualité. L'agile « classique/simple » s'applique mieux aux applications maîtrisables par le Métier, voir connues ou « existantes ailleurs » qu'aux applications demandant une inventivité importante, c'est-à-dire une exploration/conception initiale forte des fonctionnalités pratiques et de leur mise en œuvre.

Des critiques autres sur l'Agile : Tyrannie de l'urgence, « changement permanent, [source de mal-être au travail](#) », les profits importants des sociétés de consulting pour la mise en place de l'Agile (cf. Biblio plus bas)



Sept points que ne résolvent ni l'Agile ni les autres approches prédictives ou hybrides

1. Réponse aux contraintes incontournables de délai de mise en service. L'éternel fléau des projets, largement accentué (causé ?) par le lancement tardif des projets et les difficultés de mise à disposition de ressources internes ou externes
2. Budget insuffisant, en raison de contraintes externes face à l'imprécision des estimations (estimation initiale sur le backlog et sur les releases planning), d'où un budget mal taillé. Il s'agit souvent ici d'un problème de processus budgétaire, ignorant trop les imprécisions normales des estimations et leur gestion dans le temps ; mais aussi d'une maîtrise des techniques d'estimation : peu de base de données des coûts réels des projets et de retour d'expérience sur les estimations.
3. Maîtrise des environnements et des flux. Trop peu de capitalisation, diffusion et partage de connaissances au sein des DSI et des procédures IT parfois lourdes.
4. Maîtrise des technologies neuves ou innovantes. La course continue pour des connaissances complexes et perpétuellement à remettre à jour. La veille technologique ne doit pas être un process réservé à des spécialistes mais une pratique quotidienne de tous les ingénieurs et techniciens IT.
5. Maîtrise d'architectures Système complexes. Le poids des couches empilées dans le temps et de l'assemblage de briques hétérogènes, liés à la qualité de la documentation.
6. Niveau de maturité des DSI : les contraintes interdisent les bonnes règles de travail, la primauté va au résultat immédiat, la conscience de la dette technique des projets impliquée par ces pratiques émerge lentement.
7. Instabilité ou dissension dans l'organisation. Vivent les réorganisations, les changements de management et d'objectifs stratégiques, les chapelles Métier et les luttes de pouvoir ! Cela a peu d'impact sur les projets IT.

Biblio complémentaire : les arguments contre largement diffusés

<https://lesmondesdutravail.net/agile/>

https://www.lemonde.fr/emploi/article/2021/02/18/la-methode-agile-tant-vantee-en-management-est-la-risee-des-jeunes-sur-facebook_6070366_1698637.html