

ETAT DE L'ART : METHODES DE GESTION DES PROJETS INFORMATIQUES

Les méthodes itératives

Hugues MEUNIER

RAPPELS

- **Toutes les méthodes ont le même but : la maîtrise du budget, du planning et de la qualité des projets de développement informatique**
- **Plusieurs approches cohabitent :**
 - ▶ L'approche chaotique : pas de méthode de gestion de projet
 - ▶ Les approches « classiques » : les méthodes séquentielles ou en cascade (méthodes en V ou en Y) et les méthodes en spirale
 - ▶ Les approches « modernes » : les méthodes itératives et/ou agiles
- **La problématique est complexe car le système comprend certaines composantes prédictibles et d'autres imprédictibles : il faut donc prendre en compte des variables d'adaptabilité et de gestion des risques.**
- **Il n'y a pas de méthode miracle ni de méthode universelle ! Chaque méthode doit être customisée pour s'adapter aux réalités de chaque entreprise**

RAPPELS

- **CMMI n'est pas une méthode mais un modèle d'amélioration des pratiques existantes dans l'entreprise.**
- **Les méthodes agiles sont itératives et incrémentales : proposition non réciproque (les méthodes itératives ne sont pas toutes agiles : ex. RUP EUP UP)**
- **Un incrément est le résultat d'une itération**
- **Agile n'est pas XP (d'autres méthodes existent SCRUM, DSDM, MSF ...)**
- **Des discussions sont en cours pour savoir si RAD est agile**

METHODES SEQUENTIELLES

- **Toute méthode impliquant des cycles linéaires (en cascade, V, Y) est une méthode séquentielle**
- **Le principe est une succession d'étapes**
- **Des exemples sont : Merise, cycles en V, Y et cascade**
- **Principaux avantages : la documentation et les acteurs sont bien définis**
- **Principaux inconvénients : pas ou peu d'adaptabilité à des variations des conditions (nouvelles exigences), peu reproductible d'un projet à l'autre, quelques petits problèmes en début de projet peuvent engendrer de gros décalage dans le tryptique coûts, délais, qualité (sensibilité forte aux événements)**

DESCRIPTION DE MERISE

- **Merise (Fin des années 70) met en jeu 4 étapes fondamentales**
 - ▶ Expression des besoins (cahier des charges et exigences)
 - ▶ Modèle conceptuel (MCC, MCD et MOT)
 - ▶ Modèle logique (le logiciel)
 - ▶ Modèle physique (l'architecture physique)

- **Merise vise à évoluer à partir d'un SI manuel vers un SI informatisé.**

- **Merise était et reste adapté pour les applications sur gros systèmes mais est inadapté pour les applications C/S et WEB**

- **Pas de gestion du risque**

- **Merise a été largement abandonné dans les entreprises même si certaines méthodes y sont apparentées**

DESCRIPTION DES METHODES EN V et Y

- **Le processus de développement est linéaire (V) ou mixte (Y)**

- **Les projets suivent un ensemble de grandes étapes ponctuées par des livrables :**
 - ▶ Etude d'opportunité : note d'opportunité
 - ▶ Etude de cadrage : note de cadrage
 - ▶ Etude des exigences : cahier des charges
 - ▶ Conception générale : spécifications générales
 - ▶ Conception détaillée : spécifications détaillées
 - ▶ Développement : le logiciel
 - ▶ Recette : PV de recette fonctionnelle
 - ▶ Intégration : PV de recette technique
 - ▶ Déploiement (y compris la mise en production)
 - ▶ Bilan de projet

- **L'étape n+1 commence après l'étape n sur la branche considérée**

DESCRIPTION DES METHODES EN V et Y (suite)

- **Les responsabilités et les rôles sont en général clairement définis (Directeur de projet, CPI, CPU, développeur ...);**
- **Les instances de pilotages également (comité de pilotage, comité de suivi ...)**
- **Néanmoins, les étapes préliminaires (avant le développement) prennent du temps ... parfois 6 mois. Le développement commence mais les besoins ont changé : que fait-on ?**
- **Le poids de la documentation à fournir pèse sur les équipes**
- **Ces méthodes privilégient la dualité ME – MO par le partage trop tranché des responsabilités de chacun**
- **Difficile d’avoir une visibilité intermédiaire sur ce qui a été fait et ce qu’il reste à faire ...**

METHODES ITERATIVES NON AGILES

- **Principalement deux groupes de méthodes :**
 - ▶ xUP (RUP, EUP ...)
 - ▶ RAD

- **Ces méthodes sont basées sur des itérations (une itération comprend un ensemble de tâches que l'on répète sur une petite partie du projet)**

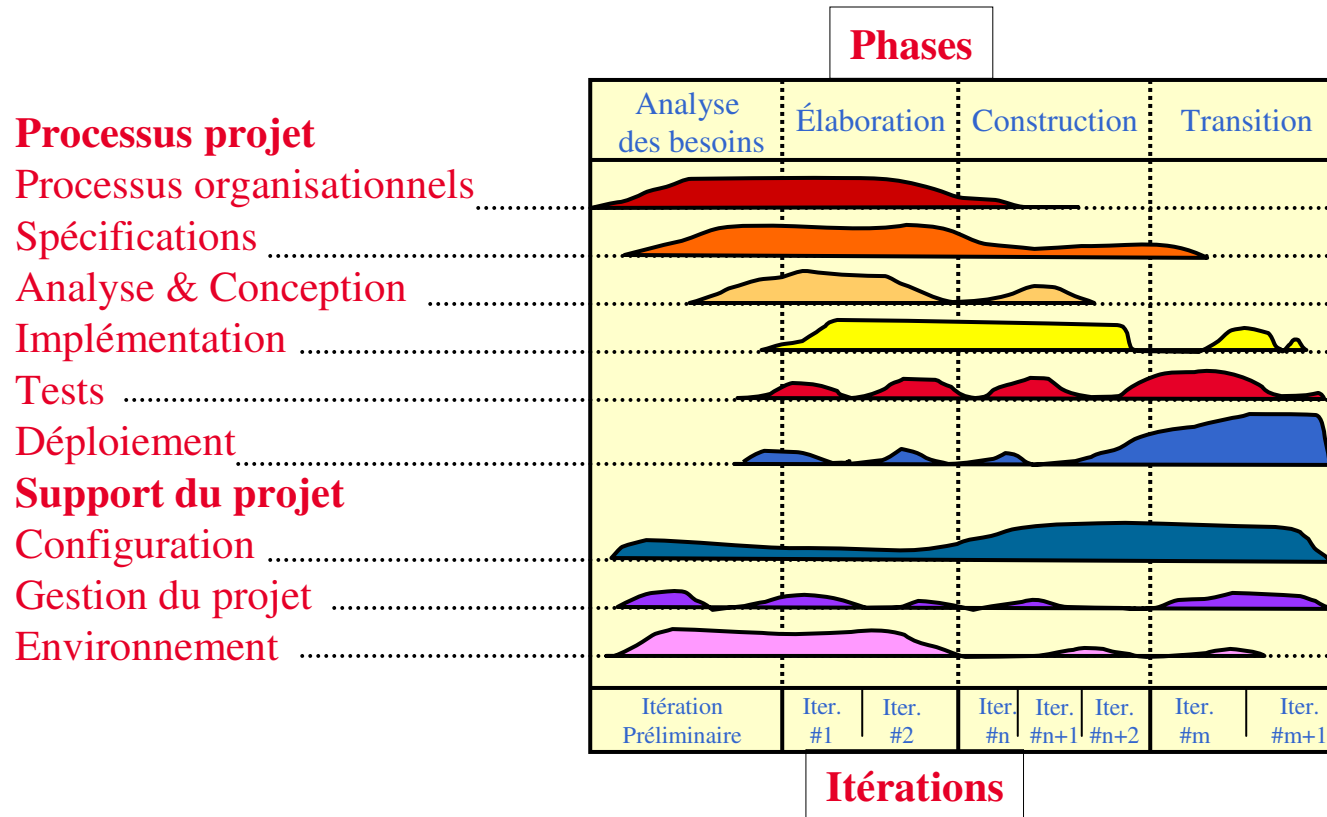
- **L'intérêt principal est le découpage fin du projet**

- **Les méthodes xUP sont fortement couplées avec le langage de modélisation UML**

Présentation de RUP

- RUP est une plate-forme de gestion des développements orientés objet
- RUP a été conçu par Rational racheté par IBM en 2005. RUP est totalement intégré dans la suite IBM RAD
- RUP propose un ensemble de pratiques et d'outils collaboratifs dans un référentiel commun
- RUP est itératif et incrémentale
- RUP est centré sur l'architecture et les composants et s'appuie sur UML

Présentation de RUP (suite)



- Les processus du projet sont traités en parallèle. Chaque processus suit les phases d'analyse des besoin, d'élaboration, de construction et de transition. Chaque phase est découpée en itérations qui définissent les incréments.

Avantages et inconvénients de RUP

- **Méthode bien formalisée et complète dans la partie développement et gestion des changements;**
- **Les documents standards sont mis à disposition. Les acteurs et instance de pilotage sont formalisés.**
- **Gestion du risque intégrée à la méthode**
- **La méthode est fastidieuse à mettre en place (nécessite des experts pour la mise en place et la customisation)**
- **La méthode est jugée lourde par ceux qui l'ont pratiquée**
- **Propriétaire IBM via leur suite RAD**
- **Ne traite pas les parties techniques (intégration, mise en production) à l'inverse d'EUP**

Présentation de RAD

- **RAD (Rapid Application Development) est une méthode apparue dans les années 80 pour le développement client/serveur.**
- **Adaptée depuis pour le développement Web**
- **La méthode n'est pas à proprement parlé itérative mais la plupart des implémentations définissent des itérations dans la phase de construction**
- **La méthode définit le cycle de développement en 5 phases :**
 - ▶ L'initialisation
 - ▶ Le cadrage
 - ▶ Le design
 - ▶ La construction
 - ▶ La finalisation
- **Un nouvel acteur apparaît : le GAR (groupe d'animation et de rapport)**

Avantages et inconvénients de RAD

- **Méthode bien documentée avec des modèles réutilisables**
- **Les acteurs et les instances de pilotage sont identifiés**
- **Méthode basée sur une gestion collaborative**
- **Rien sur les tests unitaires**
- **Rien sur les parties techniques (intégration et mise en production)**
- **Pas de gestion des changements**
- **Pas d'outils intégrés gérant la méthode**

METHODES AGILES

- **Les méthodes agiles sont des méthodes itératives qui s'adaptent aux changements de contexte et de spécifications du projet.**

- **Le manifeste :**
 - ▶ Priorité aux personnes et aux interactions par rapport aux procédures et outils
 - ▶ Priorité aux applications fonctionnelles par rapport à une documentation pléthorique
 - ▶ Priorité de la collaboration avec le client par rapport à la négociation du contrat
 - ▶ Priorité de l'acceptation du changement par rapport à la planification

- **Les méthodes agiles permettent de créer un espace de confiance entre les participants où chacun œuvre au but final : livraison d'une application conforme aux attentes du client (les utilisateurs) sans sacrifier les coûts et le planning.**

- **Agile ne signifie pas chaos ni bidouillage ! Les différentes méthodes précisent la documentation, les phases, les acteurs et les best practices.**

LES 12 PRINCIPES DES METHODES AGILES

- **Notre plus grande priorité est de satisfaire le client en lui livrant très tôt et régulièrement des versions fonctionnelles de l'application sources de valeur.**
- **Accueillir les demandes de changement à bras ouverts, même tard dans le processus de changement (production de systèmes flexibles).**
- **Livrer le plus souvent des versions opérationnelles de l'application.**
- **Clients et développeurs doivent coopérer quotidiennement tout au long du projet**
- **Construire des projets autour d'individus motivés. Leur donner l'environnement et le support dont ils ont besoin et leur faire confiance pour remplir leur mission.**
- **La méthode la plus efficace pour communiquer des informations à l'équipe reste la conversation en face-à-face.**
- **Le fonctionnement de l'application est le premier indicateur de l'avancement du projet.**
- **Le projet doit avancer à un rythme soutenable (privilégier la qualité à la durée du projet)**

LES 12 PRINCIPES DES METHODES AGILES (suite)

- Porter une attention continue à l'excellence technique et à la conception
- La simplicité, art de maximiser la quantité de travail à ne pas faire, est essentielle
- Les meilleures architectures, spécifications et conceptions sont le fruit d'équipes qui s'auto-organisent.
- A intervalles de temps réguliers, l'équipe s'interroge sur la manière de devenir plus efficace, puis ajuste son comportement en conséquence.

LES DIFFERENTES METHODES AGILES

- **Il existe actuellement de nombreuses méthodes agiles :**
 - ▶ XP (Extreme Programming)
 - ▶ SCRUM
 - ▶ MSF For agile Development
 - ▶ ASD (Adaptive Software Programming)
 - ▶ DSDM (Dynamic Software Development Method)
 - ▶ Crystal Clear
 - ▶ FDD (Feature Driven Development)

- **Chacune possède ses propres spécificités et correspond à des tailles de projet définies (il est difficile de concevoir le suivi de XP pour un projet de 100 personnes !)**

PRESENTATION DE LA METHODE XP

- **XP est une méthode d'Extreme Programming créée en 1996 par Kent Beck et Ward Cunningham**

- **Les points suivants sont favorisés dans XP :**
 - ▶ La communication entre les différents intervenants
 - ▶ La simplicité (il coûte moins cher de réaliser un système simple en lui rajoutant des fonctionnalités plutôt que de concevoir un système complexe peu utile)
 - ▶ Le feedback (indispensable pour accueillir favorablement le changement)
 - ▶ Le courage (pour les développeurs et le client)

LES 12 PRINCIPES DE LA METHODE XP

- Feedback rapide
- Assumer la simplicité
- Changements incrémentaux
- Accueillir le changement à bras ouverts
- Un travail de qualité
- Apprendre à apprendre
- Faible investissement au départ
- Jouer pour gagner
- Des expériences concrètes
- Communication ouverte et honnête
- Responsabilités acceptées
- Adaptation aux conditions locales
- Voyager léger
- Mesures honnêtes

LES PRATIQUES DE LA METHODE XP

- **Planning game** : un brainstorming qui réunit les développeurs, chef de projet et utilisateurs qui permet de définir et prioriser les besoins (sous la forme de user stories) et le planning.
- **Petites releases** : les itérations doivent être courtes (2 semaines conseillées)
- **Utilisation de métaphores** pour décrire l'architecture (pour que tout le monde comprenne)
- **Conception simple** : toujours développer la solution la plus simple
- **Stratégie orientée tests** (unitaires et fonctionnels)
- **Refactoring du code** en permanence pour le rendre plus simple et plus lisible

LES PRATIQUES DE LA METHODE XP (suite)

- **Programmation en binôme : 1 poste de travail pour 2 développeurs**
- **Propriété collective du code (mise à disposition dans un espace collaboratif)**
- **Intégration continue (au moins plusieurs fois par jour)**
- **Un rythme de travail soutenable (pas plus de 40h/semaine)**
- **Client sur site**
- **Standards de code respectés par tous (tout le monde développe de la même manière)**
- **Une équipe de développement regroupée au même endroit pour favoriser la collaboration**

LES ACTEURS D'UN PROJET XP

- **Le développeur :**
 - ▶ Travaille en binôme
 - ▶ A une double compétence : programmation et conception
 - ▶ Est autonome

- **Le client :**
 - ▶ Apprend à exprimer ses besoins sous la forme de user stories
 - ▶ Comprend les besoins utilisateurs mais également le problème et l'environnement business
 - ▶ Doit écrire les cas de tests fonctionnels

- **Le testeur (optionnel) : assiste le client pour décrire ses tests fonctionnels**

- **Le tracker : aide l'équipe à estimer les temps de développement de chaque user stories et contrôle l'avancement vis-à-vis du planning**

- **Le consultant : apporte son expertise pour aider l'équipe à résoudre ses problèmes**

- **Le Big Boss (chef de projet) : apporte à l'équipe courage et confiance**

DEROULEMENT D'UN PROJET XP

- **Un projet démarre toujours par la phase d'exploration (1 mois maximum)**

- **Cette phase d'exploration permet de :**
 - ▶ De définir la liste des fonctionnalités à inclure sous la forme de user stories
 - ▶ De planifier le projet :
 - Chaque user story fait l'objet d'une cotation (en points virtuels par exemple 5 points pour développement la consultation d'une fiche cliente)
 - L'équipe donne alors une estimation de sa vélocité c'est-à-dire le nombre de points que peuvent prendre en charge les développeurs pour une itération
 - Cette vélocité sera constamment réajustée de telle manière que la vélocité de l'itération n soit égale à la vélocité de l'itération n-1
 - Le client établit lui-même le plan de développement en affectant les user stories aux itérations (en veillant à ne pas dépasser la vélocité estimée)
 - ▶ Côté outillage, la simplicité est de mise : chaque user story fait l'objet d'une fiche cartonnée épinglée sur un tableau (l'équipe garde en permanence une vue sur l'avancement et le contenu du projet)

- **La fin de la phase d'exploration donne lieu à la première livraison (version 0) qui est en général une maquette mais comprenant des fonctionnalités utiles pour le business**

DEROULEMENT D'UN PROJET XP (suite)

- **Les livraisons suivantes sont réalisées lorsque l'incrément est réalisé (à la fin de l'itération)**
- **Chaque itération est composée des phases habituelles :**
 - ▶ Design
 - ▶ Codage
 - ▶ Tests
- **Le design est réalisé via des métaphores (tout le monde comprend)**
- **Le design doit être vérifié par des spike solutions (petits projets de test)**
- **Le codage doit respecter les normes définies**
- **Il faut coder le test avant la fonction**
- **Le code doit passer les tests avant d'être ajouté à la release**

DEROULEMENT D'UN PROJET XP (suite)

- **Les changements repassent par la phase de planification et suivent les même cycles que le développement initial**
- **La mise en production fait l'objet d'itérations**
- **Des réunions d'équipe sont organisées quotidiennement ou bi-quotidiennement**

AVANTAGES INCONVENIENTS DE XP

- La méthode permet, s'il y a adhérence, de développer très vite des applications
- Les utilisateurs sont impliqués dans le développement donc l'application répond aux besoins
- Les acteurs sont bien identifiés mais les responsabilités sont floues quelquefois
- Pas de documentation à l'origine
- Pas de gestion des risques dans la méthode
- Peu adaptée à des gros projets (> 20 personnes)
- Peu adaptée à une relation contractuelle entre un client et un fournisseur
- En conclusion : la méthode demande une adhérence complète à tous ses principes mais nécessite forcément des adaptations à la culture d'entreprise (compromis difficile à trouver)

PRESENTATION DE LA METHODE SCRUM

- **La méthode SCRUM est une méthode créée en 1990 par Ken Schwaber et Jeff Sutherland**
- **SCRUM est un terme anglais signifiant « mêlée ».**
- **Le principe de SCRUM est de focaliser l'équipe sur des itérations de 30 jours appelées sprint pour développer un ensemble de fonctionnalités**
- **SCRUM nécessite une implication forte des utilisateurs (via le directeur de produit)**
- **Les différentes phases du développement sont :**
 - ▶ Phase initiale de planning et architecture permet de produire le backlog produit (document listant les fonctionnalités attendues)
 - ▶ Ensemble des sprints (un sprint fait l'objet d'un backlog sprint qui regroupe l'ensemble des items à développer)
 - ▶ Phase de clôture
- **Un sprint est réalisé en isolant l'équipe de toute perturbation extérieure**

ACTEURS DE LA METHODE SCRUM

- **Le SCRUM master est responsable de la gestion des problèmes non techniques, d'organiser et animer la mêlée quotidienne et du backlog du sprint**
- **Le directeur de produit est le représentant métier. Il est responsable du backlog produit et de planifier les release**
- **L'équipe qui a un rôle collectif. Il n'y a pas de hiérarchie et tous participent aux tâches de l'équipe. L'équipe planifie et choisit les items à développer dans le backlog du sprint**
- **Les intervenants sont toutes les autres personnes ne participant pas activement au projet (directeur, expert)**

AVANTAGES INCONVENIENTS DE SCRUM

- La méthode gère parfaitement l'organisation de l'équipe et les différents acteurs
- SCRUM tend à produire un résultat conforme aux attentes
- Les mêlées quotidiennes et le découpage en sprint favorise l'adaptation et le feedback
- Peu de documentation fournie par la méthodes (2 documents : le backlog produit et les backlogs sprint)
- Pas d'information sur le contenu et la gestion des sprints
- Ne fonctionne pas s'il n'y a pas auto-organisation de l'équipe
- Pas de gestion des risques et des tests

PRESENTATION DE LA METHODE MSF

- **MSF (Microsoft Software Framework) for agile development regroupe un ensemble de best practices de l'éditeur sur le développement des logiciels**
- **MSF v4 est complètement intégré à la gamme des outils de développement Microsoft (Visual Studio Team System)**
- **MSF est une méthode agile qui permet de guider le chef de projet tout au long du cycle de développement**
- **MSF est centrée sur des scénarii et sur les tests**

PRINCIPES DE LA METHODE MSF

- **Partenariat avec le client**
- **Communication rapide dans l'équipe**
- **Travailler ensemble vers le même but**
- **L'excellence de la qualité**
- **Rester agile, s'adapter au changement**
- **Faire du déploiement une habitude**
- **Feedback**

LES ACTEURS DE LA METHODE MSF

- **Analyste métier**
- **Chef de projet**
- **Développeur**
- **Testeur**
- **Architecte**
- **« Release manager »**

DEROULEMENT D'UN PROJET MSF

- Le « CPU » remplit un document qui expose la vision métier du projet contenant les descriptifs, les attentes et les contraintes
- Le « CPU » crée et décrit les personnages du projet (personnages virtuels représentatifs des utilisateurs) et leurs actions
- Le CPI définit la longueur des itérations
- Le CPU définit les contraintes de tests (unitaires, performance, qualité du code, nb d'anomalies acceptables)
- Organisation d'un brainstorming pour lister les scénarii utilisateurs : une liste Excel est produite
- Les analystes métier décrivent chaque scénario dans un document synthétique (1 à 2 pages) contenant, le cas échéant, des organigrammes pour les traitements complexes

DEROULEMENT D'UN PROJET MSF (suite)

- Le « CPI » crée des tâches rattachées aux scénarii (design, développement, tests, recette, intégration)
- Les itérations sont définies dans un planning. Chaque fin d'itération donne lieu à un incrément qui doit contenir une version fonctionnelle du produit
- Les itérations sont remplies de tâches à effectuer et le tout est planifié; le développement peut commencer
- Chaque journée commence par une réunion d'équipe qui fait le point sur le reste à faire et les objectifs de la journée
- Chaque fin d'itération permet d'ajuster les paramètres projet (longueur de la prochaine itération, contenu des itérations, attribution des rôles et tâches)

DEROULEMENT D'UN PROJET MSF (suite)

- **L'itération finale est technique et consiste à produire et mettre en production la version release du produit**
- **L'intégration est soit continue soit journalière (builds nocturnes)**
- **L'automatisation du processus de développement est complet :**
 - ▶ Le code est produit par le développeur
 - ▶ Les tests unitaires doivent être produits en même temps
 - ▶ Lors du checkin (enregistrement des sources dans le référentiel source), des vérifications sont faites sur les résultats des tests unitaires, de la qualité du code ou des performances
 - ▶ Le code est compilé à la volée ou la nuit et déployer sur un serveur de recette
 - ▶ La recette donne éventuellement lieu à une liste de bugs. Des tâches sont réattribuées aux développeurs pour correction
 - ▶ Etc etc etc

AVANTAGES INCONVENIENTS DE MSF

- **La méthode est bien formalisée et documentée surtout si les outils Microsoft sont utilisés (TFS, Vstudio, Office, project);**
- **Les acteurs sont bien identifiés**
- **Gestion des risques prise en compte**
- **Très bonne intégration aux outils de Microsoft**
- **Reporting intégré**
- **Propriétaire Microsoft**
- **Nécessite comme toute méthode une appropriation et une customisation à ne pas négliger**

CONCLUSION

- Il n'y a pas de méthode miracle qui fonctionne pour tous les types et toutes les tailles de projets
- Pour les petits projets, XP est la référence et commence à pénétrer le marché des grandes entreprises
- Pour le développement .NET, MSF est la référence et les outils fournis par Microsoft permettent une bonne productivité
- Il vaut mieux une méthode en cascade bien suivie par tout le monde qu'une méthode agile mal ou peu comprise
- Les méthodes agiles sont assez peu adaptées dans le cadre de relations contractuelles avec un sous-traitant (forfait, régie, régie forfaitisée?) sauf s'il existe une relation de confiance entre les interlocuteurs
- Les méthodes agiles (surtout XP) ne sont pas non plus adaptées à des marchés de type appel d'offre public

CONCLUSION (suite)

- **Néanmoins certains principes sont évidents :**
 - ▶ Implication forte des utilisateurs
 - ▶ Adaptabilité plutôt que prédictibilité
 - ▶ Orientation forte autour du test et du risque
 - ▶ Mise en place d'outils de collaboration
 - ▶ Relation de confiance dans l'équipe (ME, MO)
 - ▶ Tout le monde au même niveau d'informations

- **Et les méthodes agiles ont des références sérieuses dans l'entreprise**

- **De plus rien n'empêche la customisation d'une méthode agile pour l'adapter au contexte d'une entreprise**