



MÉTHODOLOGIE DE PROJETS

CYCLE EN V

&

METHODES AGILES

Par: Vanessa GIACOMONI

VERSION	OBJET	DATE
0	Création	30 Juillet 2017

Table des matières

I. RAPPELS METHODOLOGIQUES:.....	3
II. LE CYCLE EN V:.....	4
III. LES METHODES AGILES:.....	5
III.1. LES VALEURS DU «MANIFESTE AGILE»:.....	5
III.2. PREMIER PRINCIPE: ITÉRATION DES PHASES:.....	5
III.3. DEUXIEME PRINCIPE: LIVRER AU PLUS TÔT CE QUI PEUT L'ÊTRE:.....	6
III.4. METHODES AGILES- DÉROULEMENT GÉNÉRAL:.....	8
III.5. METHODES AGILESET MARCHES PUBLICS:.....	8
III.6. DIFFÉRENTES MÉTHODES AGILES:.....	8

I. RAPPELS METHODOLOGIQUES:

Quelles que soient les méthodologies de conduite de projets logiciels suivies, celles-ci distinguent toujours les étapes (ou PHASES) de réalisation suivantes:

ETAPE (PHASE)	TRAVAUX ASSOCIES
SPECIFICATION DU BESOIN	<ul style="list-style-type: none"> Recueil et analyse du besoin des utilisateurs, définition des exigences (fonctionnelles, opérationnelles, techniques, etc.) que le produit doit satisfaire; Définition des interfaces externes du produit (en particulier: IHM).
CONCEPTION PRELIMINAIRE	<ul style="list-style-type: none"> Détermination de l'architecture globale la plus adaptée aux exigences; Définition des différents composants et des relations (interfaces techniques) entre ces composants. Définition pour chacun de ces composants des spécifications qu'ils doivent respecter pour s'intégrer au produit final (interfaces, traitements, performances, normes techniques, etc.).
CONCEPTION DETAILLEE	<ul style="list-style-type: none"> Développement des différents composants (codage); Tests unitaires de ces composants; Itération de ces deux étapes jusqu'à ce que les spécifications relatives à chacun de ces composants soient respectées.
INTEGRATION	Intégration (agregation) de chacun des composants au logiciel global. Après intégration de chaque composant, les tests unitaires de tous les composants déjà intégrés doivent être appliqués, pour éviter les EFFETS DE BORD.
VALIDATION & QUALIFICATION	<ul style="list-style-type: none"> Tests de validation des exigences globales exprimées par les documents de SPECIFICATION. Qualification: tests d'utilisation en environnement opérationnel;

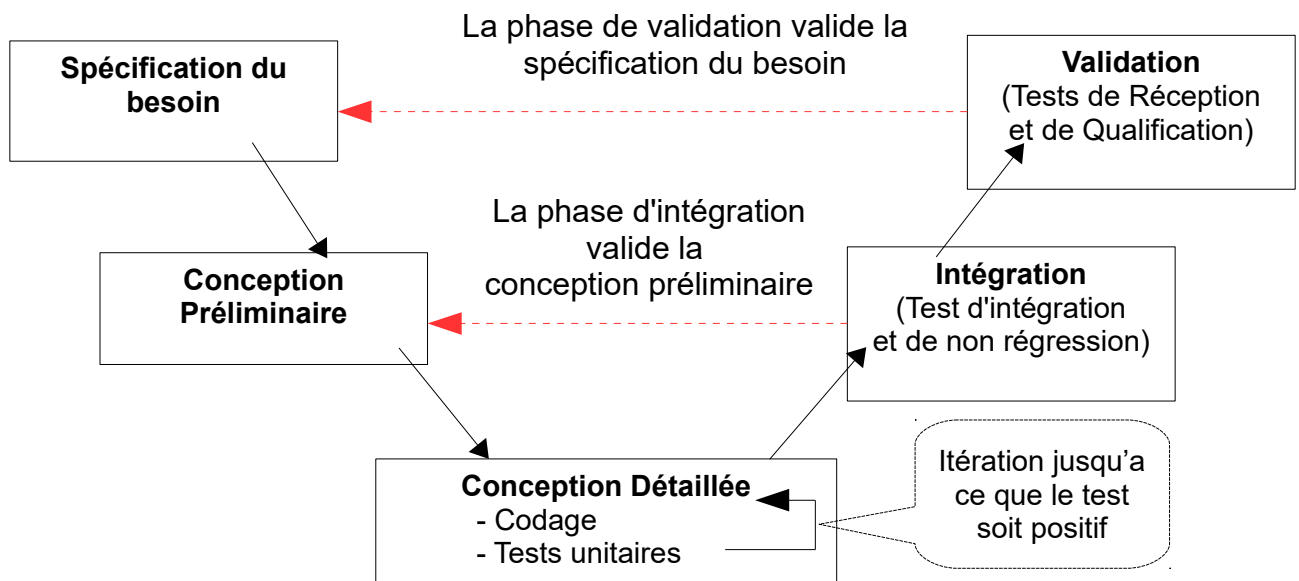
Comme il a été dit plus haut, ces 5 phases se retrouveront dans toutes les méthodes de conduite de projets logiciels. En revanche, la manière d'enchaîner ces phases diffère.

De ce point de vue, on peut classer ces méthodes en deux grandes catégories:

- Les méthodes «classiques», basées sur le concept de «CYCLE EN V»;
- Les méthodes dites AGILES, basées sur des cycles «en spirales»;

II. LE CYCLE EN V:

Le cycle en V enchaîne les cinq phases l'une après l'autre, sans retour en arrière ni itération. Cependant, à l'intérieur de chaque phase, la validation des résultats peut être obtenue après plusieurs itérations: ainsi, lors de la phase de conception détaillée, la réalisation de chaque composant est effectuée par l'itération de la séquence CODAGE-TESTS UNITAIRES jusqu'à ce que le résultat des tests soit positif (composant conforme à ses spécifications).



Les méthodes basées sur le cycle en V sont souvent qualifiées de PREDICTIVES. En effet, une fois la phase de spécification du besoin bouclée, le planning du projet est assez facile à évaluer, puisque ces spécifications deviennent contractuelles et le client ne peut plus les modifier. Cependant, cet avantage a des revers:

- Le client n'intervient qu'en phase de spécification et en phase de validation. d'autre part, la livraison est effectuée en un seul bloc: de ce fait, le client perd contact avec l'équipe de réalisation pendant les trois autres phases, ce qui diminue son implication dans le projet;
- Pour la même raison, si le recueil du besoin a été imparfaitement réalisé ou mal analysé, les défauts n'apparaîtront qu'en fin de projet (phase de validation), bien trop tard pour espérer les corriger sans trop de dégâts (dépassement des délais ou des coûts, abandon d'exigences, etc.).

Ce sont ces défauts que les méthodes agiles sont sensé atténuer.

III.LES METHODES AGILES:

III.1.LES VALEURS DU «MANIFESTE AGILE»:

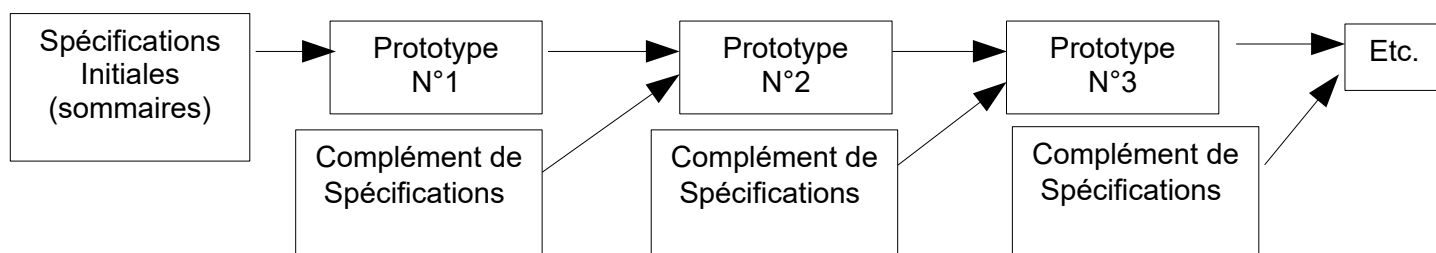
Les partisans des méthodes agiles ont élaboré une sorte de «MANIFESTE AGILE» qui met en avant les quatre valeurs fondamentales suivantes:

- Les individus et leurs interactions sont plus déterminants pour la gestion d'un projet que les processus et les outils utilisés;
- Il est plus important de créer des logiciels opérationnels que d'élaborer une documentation exhaustive;
- Une collaboration étroite avec les clients est plus efficace qu'une négociation contractuelle;
- Il est plus important de savoir s'adapter au changement que de suivre rigoureusement un plan préétabli.

De ce fait, contrairement au cycle en V où la création du logiciel se fait par une succession de phases sans itération de celles-ci ni retour en arrière, les méthodes agiles reposent sur un constant questionnement des résultats de chaque étape, à la fois par les clients et les développeurs, avec de fréquentes itérations. On peut dégager deux principes essentiels:

III.2.PREMIER PRINCIPE: ITÉRATION DES PHASES:

1. Des spécifications SOMMAIRES sont initialement élaborées;
2. Ces spécifications conduisent à un premier PROTOTYPE (après une phase de conception de ce prototype);
3. Ce prototype fait l'objet d'une évaluation par le client. Cette évaluation peut conduire celui-ci à modifier ou compléter les spécifications initiales (dans certaines mesures et après négociation avec le développeur);
4. A partir de ces nouvelles spécifications, un deuxième prototype est élaboré;
5. Ce deuxième prototype fait également l'objet d'une évaluation par le client. Cette évaluation peut conduire celui-ci à modifier ou compléter les nouvelles spécifications (également dans certaines mesures et après négociation avec le développeur);
6. A partir de ces nouvelles spécifications, un troisième prototype est élaboré;
7. Et ainsi de suite jusqu'à ce que le produit soit jugé satisfaisant...



Cette procédure a l'avantage de maintenir le client au plus près de la réalisation et de lui permettre d'affiner progressivement ses exigences en collaboration avec le développeur. Elle minimise donc les risques d'incompréhension ou d'imprécision dans l'expression des exigences.

Les inconvénients sont de deux ordres:

- La procédure exige beaucoup d'engagement du client tout au long du projet en termes de mise à disposition de ses personnels;
- La procédure permet difficilement de prévoir la durée du projet: la plus ou moins grande disponibilité des personnels du client, mais aussi leur connaissance plus ou moins grande de l'environnement d'utilisation du produit interviendront beaucoup dans cette durée.

III.3.DEUXIEME PRINCIPE: LIVRER AU PLUS TÔT CE QUI PEUT L'ÊTRE:

Contrairement au cycle en V où l'ensemble du logiciel est livré en une seule fois, les méthodes agiles conseillent de livrer en plusieurs lots, chaque lot étant FONCTIONNEL, c'est à dire implémentant complètement une ou plusieurs des FONCTIONNALITES du produit.

Exemple: *Supposons que le produit final soit un site marchand supportant les fonctionnalités suivantes:*

1. *Présentation du catalogue des produits;*
2. *Gestion du panier d'achat;*
3. *Gestion des comptes clients;*
4. *Gestion facturation et expédition;*
5. *Etc.*

Il peut être prévu de livrer un premier lot supportant la fonctionnalité n°1 dès que le logiciel correspondant sera entièrement réalisé et validé, alors que les autres lots seront toujours en développement.

Ces livraisons «dès que possible» permettent d'entretenir la relation client-fournisseur, de repérer le plus tôt possible les difficultés éventuelles et de former «en avance» les équipes d'utilisateurs.

Cependant, développer séparément les différentes fonctionnalités d'une application pose fatalement de nombreux problèmes, dont les plus courants sont:

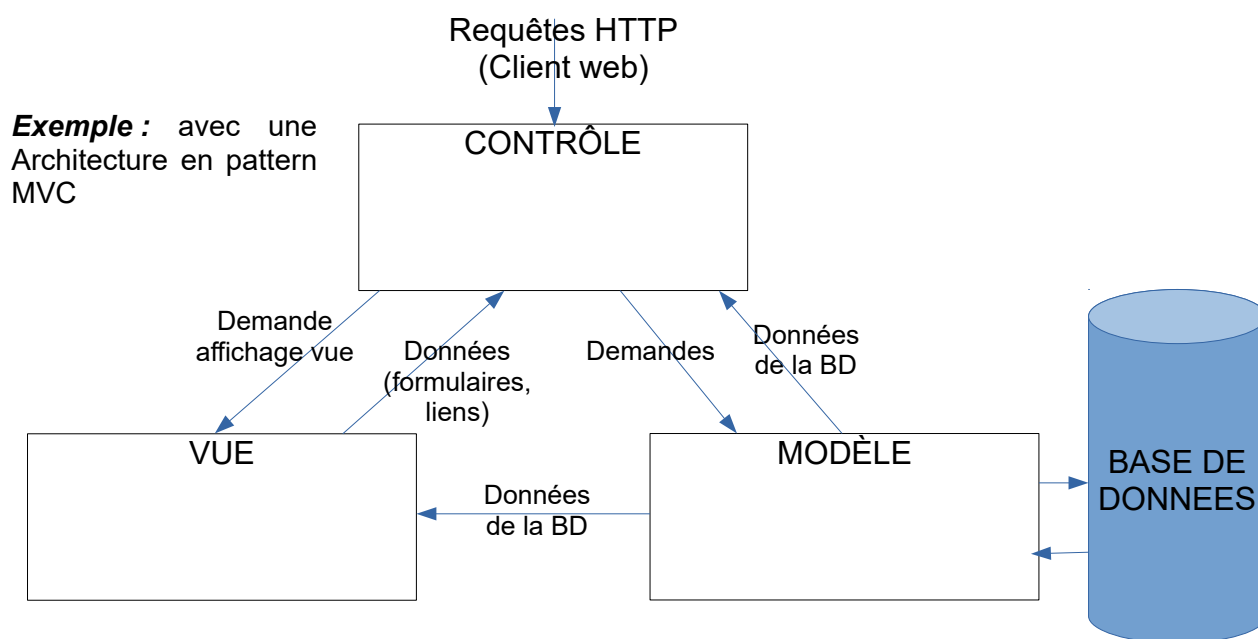
- Les incompatibilités et effets de bord (conflits de ressources);
- Les redondance: certains mécanismes communs risquent d'être développés plusieurs fois ou bien refondus et complétés à chaque introduction d'une nouvelle fonctionnalités;
- Etc.

Il est donc recommandé de réaliser en premier un NOYAU D'APPLICATION rassemblant les

mécanismes et ressources communs à toutes les fonctionnalités. La durée de réalisation de ce noyau peut représenter 30 à 40% de la durée totale, ce qui dégrade sensiblement le concept de «livraison au plus tôt».

EXEMPLE: Prenons le cas du site marchand évoqué plus haut. Si notre intention est de livrer au plus vite la fonctionnalité «**présentation du catalogue des produits**» il nous faudra pour cela disposer d'un NOYAU D'APPLICATION comprenant au minimum:

- La VUES du catalogue permettant d'afficher la liste des articles disponibles ainsi que celle qui permet d'afficher le détail d'un article sélectionné;
- Les fonctions (ou méthodes) du MODÈLE qui permettent de récupérer les informations d'un article dans la TABLE DES ARTICLES de la base de données;
- Les fonctions (ou méthodes) du MODÈLE qui permettent de créer, modifier et supprimer un article dans la TABLE DES ARTICLES de la base de données;
- Les méthodes du CONTRÔLEUR qui permettent d'activer ces différentes fonctions ou méthodes et d'afficher ces vues.



Nous pouvons noter que ce premier noyau sera assez élaboré et que sa réalisation consommera sûrement une bonne partie des ressources du réalisateur. D'autre part, ce noyau doit être conçu pour être évolutif, afin de permettre de lui adjoindre les éléments nécessaires à l'intégration des autres fonctionnalités. Il devra donc exiger une étude architecturale poussée.

REMARQUE: Notons qu'une variante du cycle en V se rapproche beaucoup du fonctionnement décrit. Il s'agit du **DÉVELOPPEMENT INCREMENTAL**, dans lequel on réalise d'abord le NOYAU de l'application, muni de ses fonctions essentielles, puis on livre l'application en plusieurs LOTS, chaque lot supportant une ou plusieurs fonctionnalités des spécifications initiales. Cette démarche de réalisation est compatible avec le type de marché public appelé **MARCHÉ A TRANCHES CONDITIONNELLES**.

III.4.METHODES AGILES- DÉROULEMENT GÉNÉRAL:

En fait, les méthodes existantes appliquent une démarche que l'on pourrait qualifier de SEMI-ITÉRATIVE.

En effet, la phase de production proprement dite (conception détaillée, intégration) est précédée de plusieurs étapes «en séquence»:

- L'étude du besoin initiale;
- La conception préliminaire initiale;
- Une planification initiale.

Les travaux correspondant à ces phases correspondent sensiblement à ceux des phases correspondantes d'un cycle en V. La différence est qu'ils sont moins poussés dans le détail. Ils permettent de définir un «NOYAU» d'application qui va supporter les ajouts exigés par la démarche incrémentale.

Ce n'est qu'après ces premières activités que les démarches itératives et incrémentales sont utilisées.

III.5.METHODES AGILESET MARCHES PUBLICS:

L'utilisation de méthodes agiles dans le cadre de marchés publics de développement de logiciels se heurte à deux difficultés:

- Le fait que les spécifications de besoin initiales puissent évoluer en cours de réalisation peut poser le problème de l'égalité de traitement des candidats au marché (possibilité de recours des candidats non choisis pour défaut d'information);
- La difficulté d'établir un planning de la réalisation fiable.

III.6.DIFFÉRENTES MÉTHODES AGILES:

Les méthodes les plus connues actuellement sont XP (eXtreme Programming) et SCRUM. Ces deux méthodes ont des caractéristiques assez semblables:

- Les équipes de développement travaille directement avec le client;
- Les cycles de développement sont très courts (d'une à deux semaines maximum);
- L'équipe de développement travaille en collaboration totale sur la base de binômes;
- Le code est testé et nettoyé tout au long du processus de développement;
- Des indicateurs permettent de mesure l'avancement du projet afin de mettre à jour le plan de développement.

Les autres méthodes disponibles sont:

- RAD (Rapid Application Development - Développement rapide d'applications).
- DSDM (Dynamic Software Development Method)
- UP (Unified Process)
- RUP (Rational Unified Process).