

Gestion de projet informatique

Stéphane Grare



Gestion de projet **informatique**

Stéphane Grare





Le code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée notamment dans les établissements d'enseignement, provoquant une baisse brutale des achats de livres, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans autorisation de l'éditeur ou du Centre Français d'Exploitation du Droit de Copie, 20, rue des Grands-Augustins, 75006 Paris.

PREFACE

On parle tous de projets dans la vie de tous les jours : nos projets de vacances, projets de carrière, projets d'avoir des enfants... Le terme projet est donc un terme du vocabulaire courant, auquel on associe une signification relativement claire et précise : c'est un ensemble d'actions que nous souhaitons entreprendre, pour atteindre un but (devenir parent, embrasser une nouvelle carrière...). En ce sens, le projet est bien « le brouillon de l'avenir » : une ébauche, mais pas encore une réalisation.

Cette notion de projet nous vient du latin « projectum de projicere », qui signifie littéralement « jeter quelque chose vers l'avant ».

Au premier abord, un projet est une chose ou un ensemble de choses que l'on se propose de faire, une intention, une ébauche.

Latins et Anglo-saxons accordent un sens assez différent à la notion de projet. Si pour nous le projet n'est qu'une action ou un ensemble d'actions que l'on projette de réaliser, dans la culture anglo-saxonne le projet désigne une notion concrète, incluant la planification, l'anticipation des risques, les acteurs impliqués... Bref, cette notion recouvre un concept plus précis, concret et pragmatique, qui appelle l'action. Nous parlerons, par la suite, de projet en ce sens.

On dénote, de manière assez intuitive, une notion forte de temporalité dans la notion de projet : un projet est généralement une aventure temporaire (ayant à ce titre un début et une fin). Il ne s'agit donc pas d'un processus répétitif : un projet est unique.

Outre les projets personnels, la majorité des projets impliquent plusieurs personnes (une compagne ou un compagnon pour devenir parent, éventuellement une famille pour partir en vacances...). On parle alors d'acteurs du projet. Ces acteurs constituent autant de ressources du projet.

En plus de ces ressources « humaines », un projet peut nécessiter, dans sa réalisation, des ressources matérielles : une voiture pour partir en vacances, une robe de mariée, des bouteilles de champagne...

L'ensemble de ces ressources représente un coût : Salaires et rémunérations pour les ressources humaines, prix d'achat ou de location pour les ressources matérielles. Un projet fait donc généralement l'objet d'une budgétisation.

Enfin, le projet aboutit, normalement, à la production de résultats matériels et immatériels. On appelle ces résultats des livrables, qui représentent les résultats attendus du projet.

Un projet est une chose ou un ensemble de choses que l'on se propose de faire en un temps donné, mettant en œuvre des ressources humaines et matérielles faisant l'objet d'une budgétisation, et aboutissant à un ensemble de livrables.

TABLE DES MATIÈRES

Préface.....	4
Le cycle de vie d'un projet.....	8
Modèles de développement.....	9
Modèle en cascade.....	12
Modèle en V.....	15
Modèle en W.....	15
Modèle incrémental.....	16
Modèle évolutif.....	17
Modèle de cycle RAD.....	17
Modèle en spirale.....	19
UP.....	21
RUP.....	25
Le cycle de l'Extreme Programming.....	26
Répondre à un appel d'offre décisionnel.....	30
Appel d'offre décisionnel.....	30
La réponse.....	30
Facteurs clés du succès.....	31
Facteurs clés en mode liste.....	33
Recueil de besoins.....	34
Objectifs.....	34
Méthodologie.....	34
Démarche de la conception cible.....	43
Guide d'entretiens.....	45
CMMI (Capability Maturity Model Integration).....	48
Le modèle CMMI.....	48
Maturité.....	48
Historique.....	49

Descriptif du modèle	49
Le niveau 1	50
Le niveau 2 (managed / discipliné).....	50
Le niveau 3 (Defined / ajusté).....	51
Le niveau 4 (Quantitatively managed / Géré quantitativement).....	52
Le niveau 5 (optimizing / en optimisation)	52
Les autres composants.....	53
La planification	54
Solutions décisionnelles.....	62
Technique et maîtrise d'œuvre.....	62
Projet décisionnel	62
Specifications	64
Définition des indicateurs.....	65
Philosophie générale des restitutions	66
Modes de restitution proposés	67
Facteurs de succès.....	70
Factures de succès en terme décisionnel.....	70
Les caractéristiques d'un projet décisionnel	70
Facteurs de succès et risques du projet.....	72
Les facteurs clés de succès.....	72
La maîtrise des risques intrinsèques à la mise en place d'un système d'information décisionnel.....	72
Logiciel décisionnel	74
Business Objects.....	74
Installation Business Objects sur Linux.....	74
Installation BusinessObjects	78
Documentation du projet	90
Spécifications fonctionnelles	90
Architecture applicative.....	90

Architecture technique	90
Justification.....	91
Le dossier de réalisation	91
Le référentiel fonctionnel.....	91
Le référentiel technique	92
Plan des tests unitaires	92
Le dossier de recette.....	92
Le Dossier technique.....	93
Manuel d'installation et de désinstallation.....	93
Manuel d'exploitation.....	93
Guides d'exploitation.....	94
Documents d'utilisation.....	94
Le dossier de maintenance	95
Rapport des actions correctives.....	95
Rapport des actions évolutives et adaptatives.....	95
Le dossier de pilotage.....	95
Compte-rendu de réunion	95
Rapport d'avancement (ou synthèse de projet).....	95
Bilan de projet	96

LE CYCLE DE VIE D'UN PROJET

Un projet est une opération dans laquelle des ressources humaines, financières et matérielles sont organisées d'une façon originale, pour réaliser un ensemble de fournitures, selon des spécifications définies, avec des contraintes de coûts et délais, de façon à obtenir un changement bénéfique défini par des objectifs quantitatifs et qualitatifs.

Un projet est un ensemble unique d'actions coordonnées, avec des dates définies de début et de fin, entreprises par un individu ou une entité pour atteindre des objectifs spécifiés en respectant des paramètres de coûts, délais et performances.

Définition de l'AFNOR : "Un projet est un ensemble d'actions à réaliser pour satisfaire un objectif défini, dans le cadre d'une mission précise, et pour la réalisation desquelles on a identifié non seulement un début, mais aussi une fin".

Définition de l'AFNOR : "La gestion de projet est l'ensemble des méthodes, outils d'évaluation, de planification et d'organisation permettant d'atteindre ses objectifs en respectant les contraintes de performance, de délais, et de coûts."

Le management de projet consiste à planifier, organiser, piloter et maîtriser tous les aspects d'un projet, ainsi que la motivation de tous ceux qui sont impliqués dans le projet et la maîtrise de la relation client, de façon à atteindre les objectifs de façon sûre et dans les critères définis de coûts, délais et performances. Cela inclut les tâches de direction nécessaires aux performances du projet.

Un objectif est une contrainte qui va être « imposée » au système projet afin qu'il se réalise dans un cadre. Ce cadre est imposé par le commanditaire.

Un projet comporte 3 niveaux d'objectifs :

- > Les objectifs de qualité.
- > Les objectifs de temps.
- > Les objectifs de coût.

Objectifs de qualité : Ce sont tous les éléments qui vont « qualifier » le produit qui va sortir du projet. Ces éléments vont constituer les performances du produit ; ce sont ces performances qui vont satisfaire le besoin.

Objectifs de temps : C'est le calendrier dans lequel le projet doit se réaliser. Ce calendrier comporte une date de début du projet, une date de fin du projet, des échéances intermédiaires éventuelles.

Objectifs de coût : C'est la somme du coût des ressources nécessaires à la mise en œuvre du projet.

Un objectif doit répondre à un certain nombre de critères :

- > Il doit être mesurable : car il faut pouvoir le visualiser et le comprendre, il doit donc être quantifié. Cela permettra par ailleurs de savoir si l'objectif a été atteint par la mesure des résultats à la fin du projet.

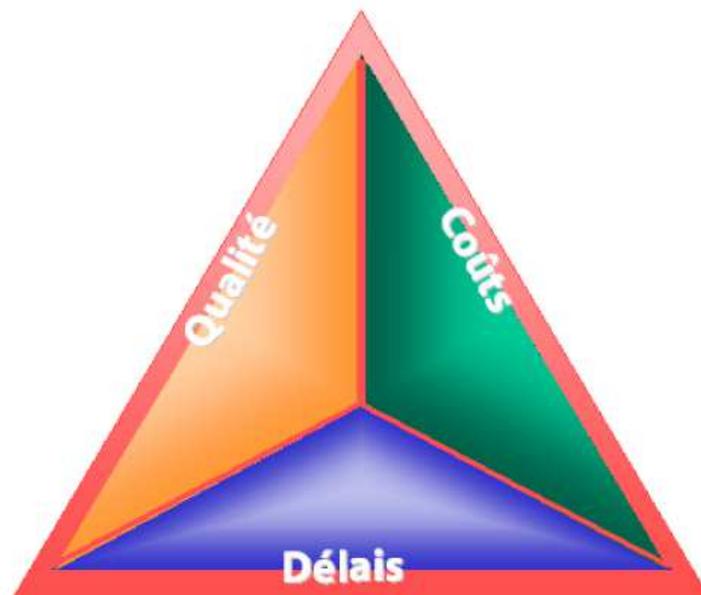
-> Il doit être réalisable : car il faut pouvoir atteindre l'objectif. L'objectif impliquera un engagement du chef de projet ; et pour que celui-ci s'engage sur les objectifs, il faut bien sûr que l'objectif soit atteignable.

-> Il doit pouvoir être négocié : afin d'obtenir un engagement mutuel entre celui qui fixe l'objectif et celui qui se propose de l'atteindre, la négociation s'engage afin d'avoir un accord mutuel qui donne toutes les chances au projet d'aboutir.

-> Il doit être partagé : si l'objectif va être réalisé par un groupe de personnes il faut que cet objectif soit compris par toutes afin qu'il n'y ait aucune ambiguïté entre elles et sur le but à atteindre.

-> Il doit être individualisé : on ne fixe pas un objectif directement à un groupe de personnes. On s'assure que l'objectif a été réparti entre ces personnes et que chacune connaisse sa part d'objectif à atteindre.

Les 3 objectifs Coût/Qualité/Temps sont interdépendants entre eux et interagissent pendant le projet. Si on modifie un seul des objectifs en cours de projet, les deux autres objectifs vont être modifiés.



L'assemblage des 3 objectifs s'appelle le triangle de la performance.

Le cycle de vie d'un projet informatique est composé des étapes et des enchaînements nécessaires pour réaliser le produit ou le service qui font l'objet du projet. Le cycle de vie doit être adapté, en fonction de la complexité du produit à fabriquer. Généralement, il a pour but de participer à la maîtrise des risques de fabrication, ainsi que d'assurer la qualité du produit fini.

MODELES DE DÉVELOPPEMENT

Il existe des "modèles de développement" pour le découpage des projets SI. Un modèle est une abstraction de quelque chose de réel qui permet de comprendre avant de construire ou

de retrouver les informations nécessaires pour effectuer des modifications et extensions. Le modèle simplifie la gestion de la complexité en offrant des points de vue et niveaux d'abstraction plus ou moins détaillés selon les besoins. Il facilite la communication entre les différents intervenants sur le projet. Il supporte la conduite et la gestion des processus de développement et de maintenance.



Les modèles de cycle de vie du logiciel décrivent à un niveau très abstrait et idéalisé les différentes manières d'organiser la production. Les étapes, leur ordonnancement, et parfois les critères pour passer d'une étape à une autre sont explicités (critères de terminaison d'une étape - revue de documents -, critères de choix de l'étape suivante, critères de démarrage d'une étape).

Le cycle de vie d'un projet informatique est composé des étapes et des enchaînements nécessaires pour réaliser le produit ou le service qui font l'objet du projet. Le cycle de vie doit être adapté, en fonction de la complexité du produit à fabriquer. Généralement, il a pour but de participer à la maîtrise des risques de fabrication, ainsi que d'assurer la qualité du produit fini.

On appelle "cycle de vie canonique" le cycle de vie de base d'un projet informatique. En fonction de l'enchaînement des phases de projet, on débouche sur un cycle de vie en cascade, en V, ou évolutif. D'autre part, en fonction de la complexité du résultat à produire, s'il s'agit d'un projet ou d'un programme, le cycle de vie sera une combinaison adaptée des phases canoniques. Le cycle de vie canonique ne décrit pas les processus de support de projet tels que le processus de management, l'assurance qualité, la logistique.

L'étude d'opportunité permet de définir et d'évaluer les objectifs tangibles et intangibles du projet :

- > Définir les objectifs, le périmètre et les grandes lignes de la solution
- > Vérifier qu'ils sont alignés avec la stratégie de l'entreprise
- > Identifier les risques et les moyens de les contenir
- > Définir les gains attendus
- > Définir les coûts maximaux

L'étude de faisabilité permet d'établir la faisabilité du projet, c'est-à-dire d'établir s'il existe une solution technique optimale répondant aux exigences techno-économiques identifiées lors de la phase précédente :

- > Établir l'expression des besoins
- > Identifier les solutions possibles
- > Identifier les risques techniques, financiers et projet
- > Préconiser la solution optimale
- > Faire une estimation détaillée du projet
- > Faire un planning
- > Rédiger un cahier des charges

La phase de conception définit le processus de réalisation. Elle comprend :

- > La conception de l'architecture générale du système
- > La conception de l'architecture détaillée
- > La conception détaillée de chaque composant (interfaces, services, base de données)
- > La conception des plans de tests

La phase de réalisation est la phase où est produit l'objet du projet. Elle comprend :

- > La programmation des composants
- > Les tests unitaires
- > La réalisation des jeux d'essai
- > Les tests d'interface
- > Les tests d'intégration (reprises, performance...)

La phase de recette est la phase de contrôle qualité, où l'on s'assure que le produit fini répond bien à la demande initiale, à l'intérieur des contraintes posées :

- > Les tests de validation
- > Les tests en parallèle
- > Les tests de non régression
- > Les tests de réception ou recette

La phase de mise en production s'achève par la mise en production.

- > Les tests de mise en production

La phase de maintenance est la phase qui permet à l'objet du projet de se maintenir au cours du temps.

- > La gestion du changement et des évolutions

Il existe plusieurs types de modèles :

- > Les modèles de cycle de vie linéaire
 - Modèle en cascade
 - Modèle en V
 - Modèle en W

Chaque phase du cycle de vie doit être réalisée avec tous les détails requis avant de passer à la phase suivante. Il y a en particulier un effort important à fournir pour la documentation.

Il n'existe pas de version du logiciel exécutable avant la fin du développement. Toute reprise, en cas d'écart entre la compréhension des besoins et le besoin réel, est coûteuse.

-> Les modèles de cycle de vie non linéaire

Les logiciels réalisés suivant ces modèles sont complétés par itérations ou incréments successifs. Les détails de réalisation peuvent être reportés afin de produire une version opérationnelle du logiciel au plus tôt. Ce type de modèles semble plus approprié pour prendre en compte le caractère évolutif des besoins des utilisateurs.

- Modèle incrémental
- Modèle évolutif
- Modèle en spirale
- Modèle de cycle RAD
- ...

Il faut souligner la différence entre étapes (découpage temporel) et activités (découpage selon la nature du travail). Il y a des activités qui se déroulent dans plusieurs étapes (ex : La spécification, la validation et la vérification), voire dans toutes les étapes (ex : La documentation).

MODELE EN CASCADE

Dans ce modèle, les étapes sont réalisées de façon séquentielle. Chaque étape donne lieu à l'établissement d'un document.

Analyse des besoins ou analyse préalable :

Document final : Cahier des charges + plan qualité

- > Qualités fonctionnelles attendues en termes des services offerts,
- > Qualités non fonctionnelles attendues : Efficacité, sûreté, sécurité, facilité d'utilisation, portabilité...
- > Qualités attendues du procédé de développement (ex : Procédures de contrôle qualité).

Le cahier des charges peut inclure une partie destinée aux clients (définition de ce que peuvent attendre les clients) et une partie destinée aux concepteurs (spécification des besoins).

Étude préliminaire ou étude de faisabilité ou planification :

Document final : Rapport d'analyse préliminaire ou schéma directeur

- > Définition globale du problème,
- > Différentes stratégies possibles avec avantages/inconvénients,
- > Ressources, coûts, délais.

Analyse du système :

Analyse détaillée de toutes les fonctions et autres caractéristiques que le logiciel devra réaliser pour l'utilisateur, tel que vu par l'utilisateur.

Résultat : Document de spécifications qui définit le « quoi » du logiciel. Ce document doit être rédigé dans un langage aussi formel que possible.

- > Modélisation du domaine,
- > Modélisation de l'existant (éventuellement),
- > Définition d'un modèle conceptuel (ou spécification conceptuelle),
- > Plan de validation.

Conception générale :

Définition de l'architecture générale du logiciel. Pas de détails sur la manière dont les éléments composant le système seront implantés.

- > Résultat : Document de conception générale

Conception détaillée :

Spécification de la manière dont chacun des composants du logiciel sera réalisé et dont ils interagiront.

- > Résultat : Documents de conception détaillée.

Codage et tests unitaires :

Document final : Dossiers de programmation et codes sources

- > Traduction dans un langage de programmation,
- > Tests avec les jeux d'essais par module selon le plan de test.

Intégration et tests de qualification :

- > Composition progressive des modules,
- > Tests des regroupements de modules,
- > Test en vraie grandeur du système complet selon le plan de test global (« alpha testing »).

Installation :

Mise en fonctionnement opérationnel chez les utilisateurs. Parfois restreint dans un premier temps à des utilisateurs sélectionnés (« beta testing »).

Maintenance :

- > Maintenance corrective (ou curative),
- > Maintenance adaptative,
- > Maintenance perfective.

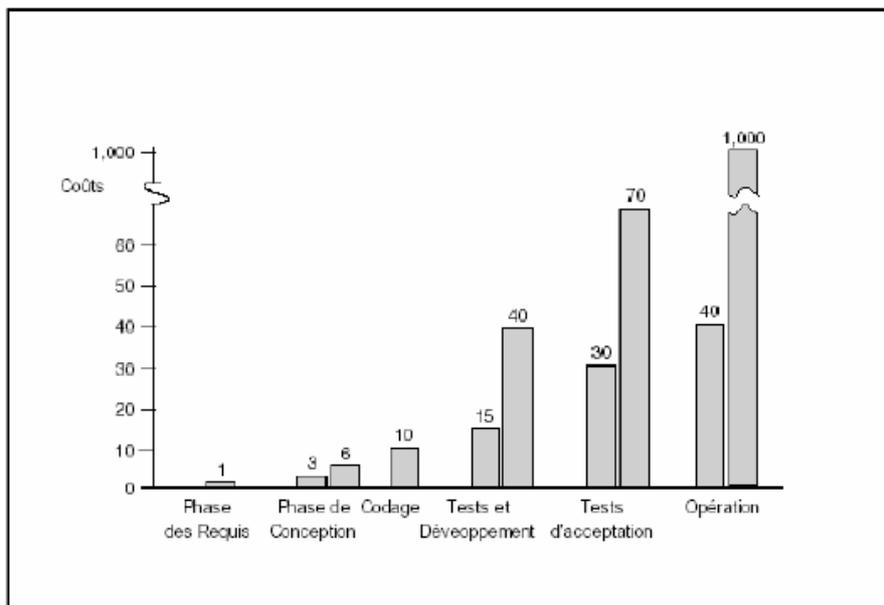
Activités transversales à tout le cycle de vie :

-> Spécification, documentation, validation et vérification, management.

Des études ont été menées pour évaluer le coût des différentes étapes du développement:

Type de système	Conception	Implantation	Test
Gestion	44 %	28 %	28%
Scientifique	44 %	26 %	30%
Industriel	46 %	20 %	34%

Mais c'est la maintenance qui coûte le plus cher.



Coûts relatifs de correction des exigences, lorsque découvertes à différents stades

Les avantages du modèle en cascade

- > On a une idée claire de ce qu'il y a à faire.
- > Il est simple à comprendre.
- > Il permet une normalisation des cadres conceptuels et terminologiques des différentes activités.

Inconvénients :

- > Il ne reflète pas la façon dont le code est réellement développé.
- > Il manque de flexibilité en cas d'imprévu.
- > Il n'y a pas de retour avant la livraison chez le client.

MODELE INCREMENTAL

- > Division du logiciel en sous-systèmes (un par fonctionnalité)
- > Première version : Logiciel partiel
- > Chaque nouvelle version ajoute un nouveau sous-système

Ce modèle consiste à développer le futur logiciel par lots et à établir un planning de réalisation de ces lots avant le démarrage du processus de développement. Chaque lot doit fournir un produit qui sera immédiatement mis en exploitation dans un environnement opérationnel.

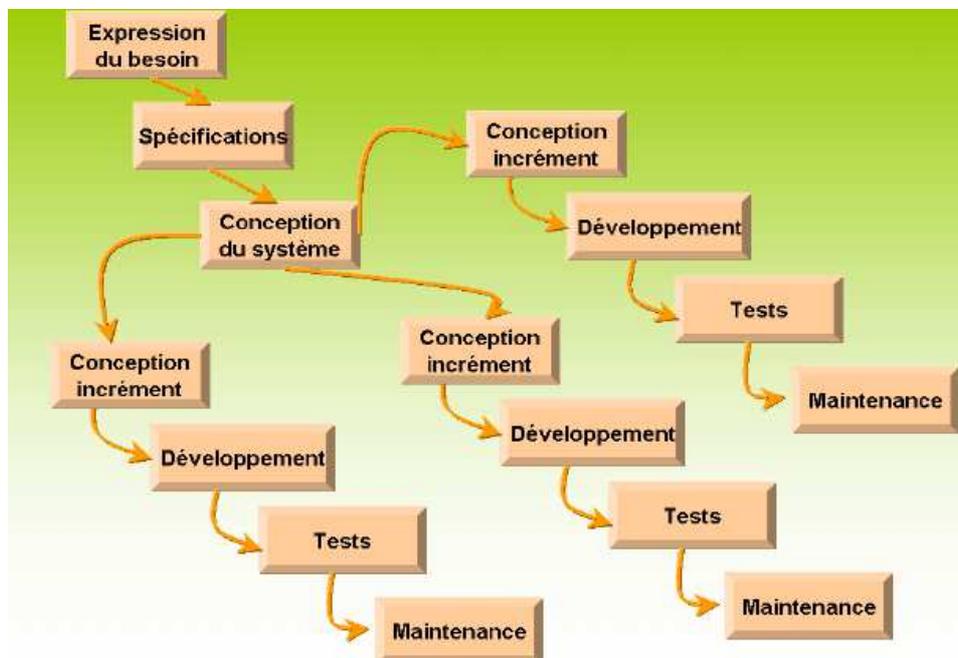
L'ensemble des produits en exploitation à un moment donné constitue une version provisoire du logiciel dont la durée de vie s'étend jusqu'à la livraison du prochain lot.

Les avantages d'un tel modèle :

- > Les efforts de maintenance d'une version provisoire sont faibles.
- > Chaque nouvelle version améliorant la précédente, il y a une meilleure adéquation entre les besoins des utilisateurs et les besoins couverts par le logiciel.

Les inconvénients :

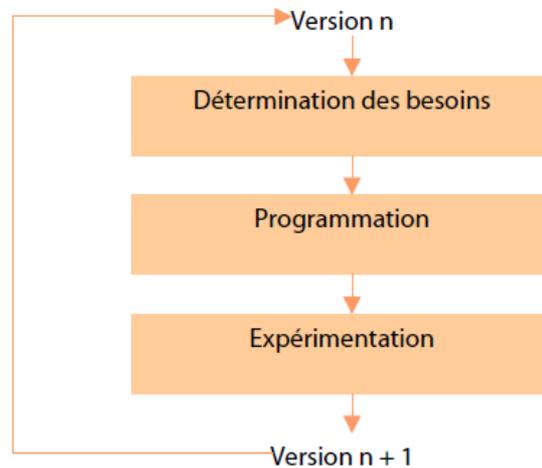
- > Il faut spécifier dès le début du développement l'architecture globale du logiciel et définir les lots qui seront développés : le noyau, les incréments qui doivent être fonctionnellement indépendants, leur intégration.
- > L'intégration de nouveaux composants est de plus en plus complexe.



C'est un modèle adapté aux grands projets. Néanmoins, l'architecture du système doit permettre de définir des domaines suffisamment découplés. Dans le cas contraire, certains incréments doivent attendre que les incréments avec lesquels ils sont liés soient suffisamment développés. Lorsqu'on leur propose un développement par lot, les Maîtrises d'Ouvrage doivent vérifier le couplage des domaines.

MODELE EVOLUTIF

- > Approche itérative
- > Division du logiciel en sous-systèmes (un par fonctionnalité)
- > Première version : Coquille complète du logiciel
- > Chaque nouvelle version apporte une modification / amélioration à un sous-système



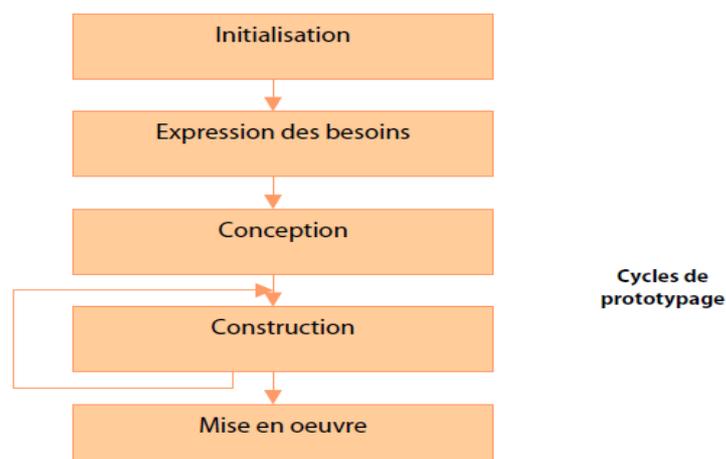
Avantages

- > Formation précoce des utilisateurs, retours rapides.
- > Création précoce de nouveaux marchés pour les nouvelles fonctionnalités.
- > Focalisation sur le nouveau domaine d'expertise à chaque étape (version).
- > Détection précoce des problèmes imprévus (correction immédiate du système en développement).

Inconvénients

- > Risque de la remise en cause du noyau (fonctionnalités de base) au cours du développement.

MODÈLE DE CYCLE RAD



Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

