|  |
| --- |
|  |
| Application Type |
| Méthodologie |
|  |
| **Thomas Auguey** |
| **20/08/2015** |

|  |
| --- |
| Méthode de développement suivit par les participants à l’élaboration du projet. |

SOMMAIRE

[Introduction 3](#_Toc434333199)

[Lexique 4](#_Toc434333200)

[Recueil du besoin 5](#_Toc434333201)

[Expression du besoin 5](#_Toc434333202)

[Modélisation du besoin 5](#_Toc434333203)

[Implémentation du besoin 5](#_Toc434333204)

[Tests et validation 5](#_Toc434333205)

[Incrémentation de la version 5](#_Toc434333206)

# Introduction

Cette méthode de développement s’appuie sur une méthode descendante du haut (le besoin) vers le bas (l’implémentation). On crée une première implémentation simple, on la valide puis on incorpore petit à petit les nouveautés en réitérant le cycle à chaque développement, chaque cycle impose une incrémentation de la version mineur jusqu'à atteindre la version majeur validée par le client.

Ce cycle de développement est répété par phase à chaque modification du projet:

1. Recueil du besoin (cahier des charges)
2. Expression du besoin (cas d’utilisations)
3. Modélisation du besoin (MCD/MPD/MOO)
4. Implémentation du besoin (Codage)
5. Tests, validation
6. Incrémentation de la version

Cette méthode est une base sur laquelle tout développeur peut s’appuyer pour communiquer et participer autour d’un projet. Elle n’a pas pour prétention d’être parfaite ni définitive mais le plus simple et le plus efficace possible dans la majorité des cas.

# Lexique

**MCD** : Model conceptuel de données. Diagramme représentant les données relationnelles sous forme d’entités et de relations.

**MPD** : Model physique de données. Diagramme représentant les données relationnelles sous forme de tables et de clés.

**MOO** : Model orienté objet. Diagramme représentant les données relationnelles sous forme de classes et de pointeurs.

**Langage orienté objet :** Langage de programmation intégrant les concepts de classes de données et de polymorphisme.

# Recueil du besoin

Dans cette phase le développeur récupère le besoin du client pour le retranscrit le plus lisiblement dans la section « Cahier des charges » du projet.

# Expression du besoin

L’expression du besoin consiste à reprendre le besoin du cahier des charges pour en extraire les éléments et leurs relations, les actions et les acteurs du système. Cette phase est une phase d’étude du cahier des charges en prévision de son intégration dans le système, c’est à ce moment que l’on peut reformuler le besoin du client et proposer des choix technologiques.

* Définir les éléments
* Définir les relations
* Définir les acteurs

# Modélisation du besoin

Suivant les technologies choisies la modélisation est représentée de différentes manières :

* Reprend les éléments du cahier des charges pour les représenter sous forme d’entités (MCD), tables (MPD) et classes (MOO)
* Reprend les cas d’utilisations du cahier des charges pour identifier et documenter les fonctions de la logique métier

# Implémentation du besoin

L’implémentation reprend ou définit l’architecture de l’application (généralement dérivées du patron MVC pour les langages orienté-objet et d’un patron de fonctions pour les langages procéduraux) pour implémenter la logique métier et l’accès aux données modélisées dans la phase précédente.

Dans tous les cas, l’implémentation tire partie des avantages du langage de programmation et des technologies pour arriver au but final qui est une implémentation simple, robuste et cohérente avec le système.

* Reprendre le MPD mon écrire le code persistant (SQL)
* Reprendre le MOO pour écrire le code applicatif (C#, C++, Java, …)
* Reprendre les cas d’utilisations pour écrire les procédures stockées (SQL) et les méthodes applicatives (langages déclaratifs)

# Tests et validation

* Tester le système avec les tests unitaires
* Présentation et validation par le chef de projet et le client

# Incrémentation de la version

* Incrémenter la version mineure (indique le nombre de modifications du projet)
* Incrémenter la version majeure (indique le nombre de distribution au client)
* Documenter les modifications dans le suivit de version
* Documenter les modifications dans les révisions