|  |
| --- |
| Les patriotes |
| Méthodologies Agiles et Ingénierie Logicielle |
| Classe fétiches : Baby-sitter et Enfant  BIT – Magali  GUILLAUME – Josselin  MARY – Laurent  QUARCIA - Anthony |

|  |
| --- |
| 03/03/2015 |

# Introduction

# Synthèse

## Rôles et contributions

Notre groupe est composé en deux équipes : Magali BIT et Laurent MARY sont en charge de la maîtrise d’œuvre tandis qu’Anthony QUARCIA et Josselin GUILLAUME représentent la maîtrise d’ouvrage.

La MOA a donc imaginé un scénario mélangeant les deux classes fétiches utilisées, il en a résulté la gestion des baby-sitters. Ils ont donc ensuite formulé des tests fonctionnels et les ont proposés comme GuideLine à la MOE afin de respecter le « Test Driven Development ». L’écriture des tests et du code a été fait en respectant la politique du « Pair Programming». Il y a eu une évolution des besoins tout au long du projet grâce à cette méthode AGILE.

De plus, la MOA a dépeint la solution à travers des diagrammes présents dans la suite de ce rapport afin de faciliter la compréhension des attentes à chaque du projet.

Suite à l’écriture des tests fonctionnels par la MOA, la MOE a mis en place l’environnement de développement, a développé le squelette du code ainsi que le corps de toutes les méthodes. Ensuite, elle a mis en place les design patterns requis et a testé toutes les méthodes unitairement. Cela a été fait en « Pair Programming » ce qui a permis d’aller plus vite dans la compréhension des besoins pour la MOE et la compréhension de la solution pour la MOA.

## Hypothèses et contexte de travail

Concernant le contexte de travail, nous avons décidé de diviser le travail en plusieurs sprints. Tout d’abord le l’ « Acte 1 » et l’ « Acte 2 » dont les périodes temporelles communes. L’acte 1 a pour objectif de définir les « User Stories », le délai 3 jours a été respecté. L’acte 2 quant à lui, est réservé à la mise en place technique de la solution. Son délai de 4 jours a également été respecté même si certains outils ont été ajoutés par la suite. L’acte 3 concerne le premier jet d’écriture des tests fonctionnels. Il a donc concerné une partie de la MOA et une partie de la MOE (respect du « Pair Programming »). Cet acte 3 représente, comme prévue, le mieux l’aspect Agile de notre travail car il contient l’écriture des tests fonctionnels et le code de la solution respectant les normes Agiles, mais également l’expression d’erreur fonctionnelle par la MOE puis corrigée par la MOA au sein d’un fichier partagée remplaçant Google Code. L’acte 4, la suite et fin des tests fonctionnels, a donc été prévue à la suite.

## Originalité de la solution, valeur ajoutée perçue

## Joies et peines

N’ayant jamais pratiqué les méthodes Agiles, ce projet nous a permis de réaliser à quel point le « Pair Programming » est bénéfique et permet à la solution d’être efficace et rapide. Nous avons donc pris du plaisir à l’appliquer. Il fut cependant compliqué, parfois impossible, de mettre en accord tous les outils. C’est pourquoi nous avons utilisé « » à la place de Cobertura et un simple fichier partagé afin de relever les erreurs fonctionnelles et non Google Code.

# Les outils utilisés

## Github

GitHub est un service web d'hébergement et de gestion de développement de logiciels, nous l’avons donc utilisé afin de partager le code de notre projet mais également quelques annexes (diagrammes, impressions d’écrans, …).

Table des matières

[Introduction 1](#_Toc413170166)

[Synthèse 2](#_Toc413170167)

[1) Rôles et contributions 2](#_Toc413170168)

[2) Hypothèses et contexte de travail 2](#_Toc413170169)

[3) Originalité de la solution, valeur ajoutée perçue 2](#_Toc413170170)

[4) Joies et peines 2](#_Toc413170171)

[Les outils utilisés 4](#_Toc413170172)

[Github 4](#_Toc413170173)

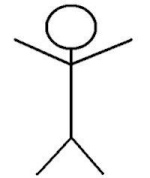
[Sommaire 5](#_Toc413170174)

[Diagramme de cas d’utilisation 6](#_Toc413170175)

[Diagramme d’état-transitions : 7](#_Toc413170176)

[Scénario d’exécution :   8](#_Toc413170177)

# Diagramme de cas d’utilisation



Utilisateur

Ici il faut commenter le diagramme de cas d’utilisation. Ça parait évident.

# Diagramme d’état-transitions :

EnfantChezParents

EnfantAuLit

EnfantAMangé

EnfantChezBS

Ici il faut commenter le diagramme de cas d’utilisation. Ça parait évident.

# Scénario d’exécution :

Ecran d'accueil

Ecran d'accueil

Gestion parent

Gestion baby-sitter

Créer parent

Choisir parent existant

Choix de gestion

Choix de gestion

Choix de gestion

Créer enfant

Choisir enfant existant

Choisir baby-sitter

Emmener Enfant

Récupérer Enfant

Ici il faut commenter le scénario. Ça parait évident.