06/04/2020

Gaël Mariot

Centre de formation Professionnel Technique

Documentation Scrum’o’Wall

Travail de Diplôme ES 2020

# Table des matières

[1 Table des matières 1](#_Toc37055021)

[2 Introduction 2](#_Toc37055022)

[3 Cahier des charges 2](#_Toc37055023)

[3.1 Durée du diplôme 2](#_Toc37055024)

[3.2 Titre du projet 2](#_Toc37055025)

[3.3 Objectifs du projet 2](#_Toc37055026)

[3.4 Description détaillée 2](#_Toc37055027)

[3.4.1 Développement agile 2](#_Toc37055028)

[3.4.2 Stockage multi-projet 2](#_Toc37055029)

[3.4.3 Historique des sprints 2](#_Toc37055030)

[3.5 Maquettes 3](#_Toc37055031)

[3.5.1 Fenêtre des projets 3](#_Toc37055032)

# Introduction

La validation de la formation de technicien ES en informatique du Centre de Formation Professionnel Technique nécessite la réalisation d’un projet sur une durée de neuf semaines afin de démontrer les compétences acquises durant ce cursus.

# Cahier des charges

## Durée du diplôme

Ce travail est réalisé du 06 avril au 8 juin 2020.

## Titre du projet

Scrum’o’Wall

## Objectifs du projet

Le but de ce projet est de créer un logiciel interagissant avec le mur tactile NCI Lab situé dans la salle de Technicien ES du CFPT Informatique. Ce logiciel permettra la gestion de projets en méthode agile et utilisera la fonctionnalité multipoint de l’écran permettant ainsi l’utilisation du logiciel par plusieurs utilisateurs simultanément.

## Description détaillée

### Développement agile

Le développement agile est une méthode de travail très répandue dans le pôle informatique car il permet de pouvoir maintenir un produit plus facilement et se concentre sur des petites parties d’un projet effectuées lors de sprint plutôt que sur le projet entier d’un coup.

### Stockage multi-projet

Le logiciel devra pouvoir contenir plusieurs projets et les garder en mémoire afin de permettre à plusieurs équipes de travailler sur le même mur. Cela permet d’apporter un gain de place aussi bien sur le stockage de la machine que physiquement.

### Historique des sprints

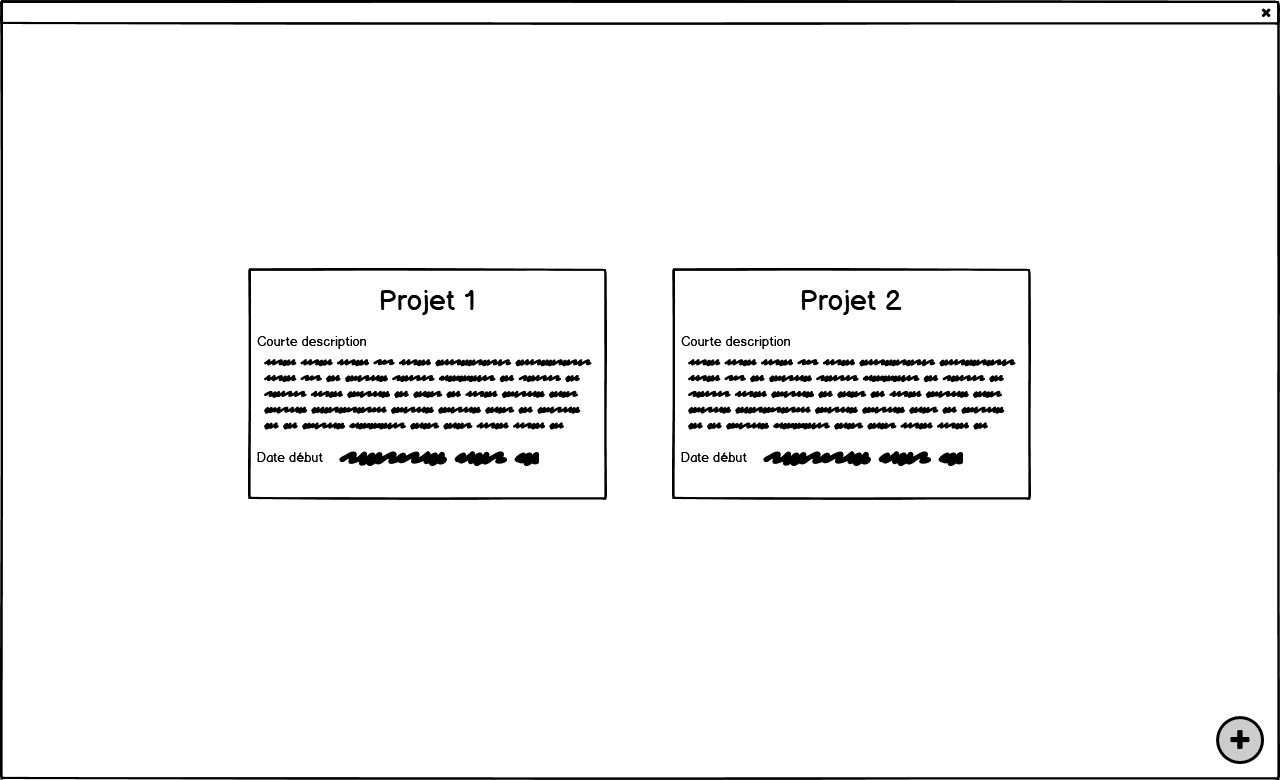
Le logiciel permettra de voir les sprints selon leur ordre chronologique pour permettre aux utilisateurs de voir le chemin qu’ils auront parcouru et ils pourront planifier en avances les sprints suivants.

### Détection multipoint

Le logiciel pourra gérer une interaction multipoint permettant aux utilisateurs de travailler sur tout l’écran simultanément.

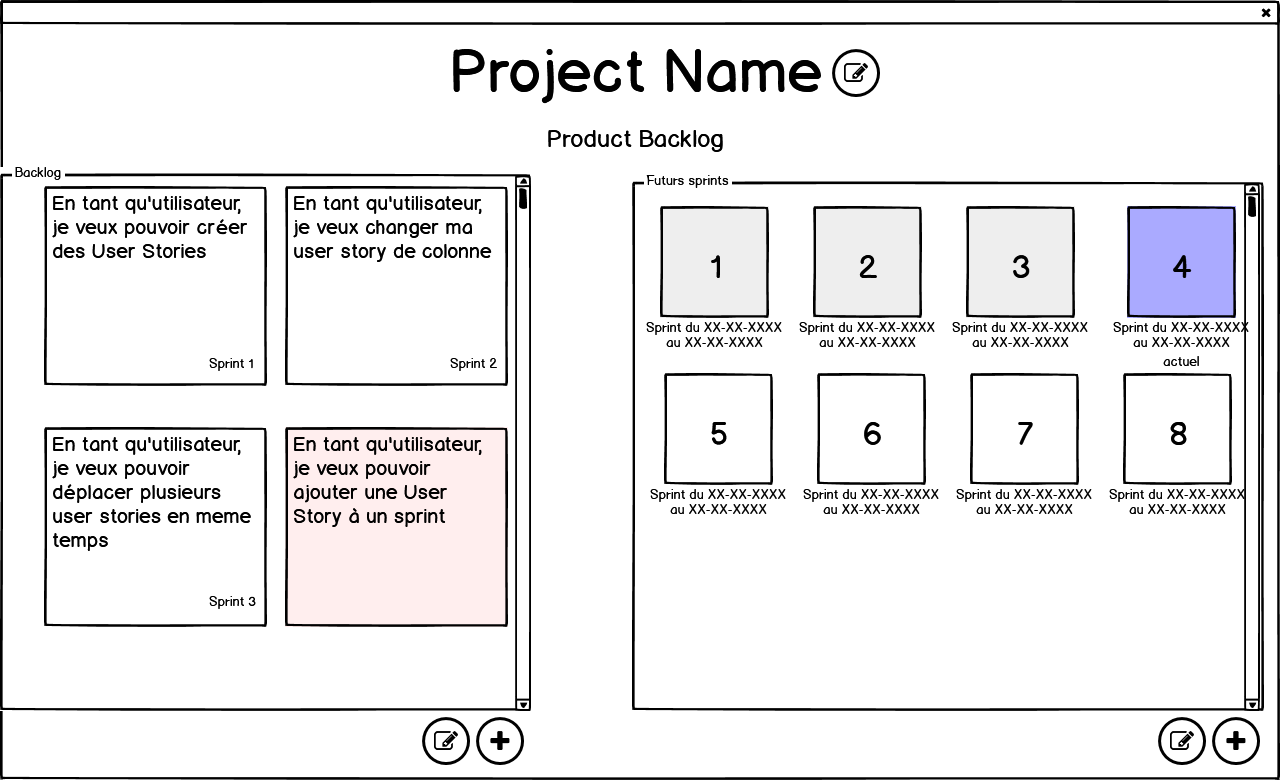
## Maquettes

### Fenêtre principale



Sur cette fenêtre, une mosaïque avec les différents projets déjà créé sera afficher. Un bouton permettra de rajouter un projet. Un maintient prolongé permettra également de rajouter un projet par un menu contextuel.

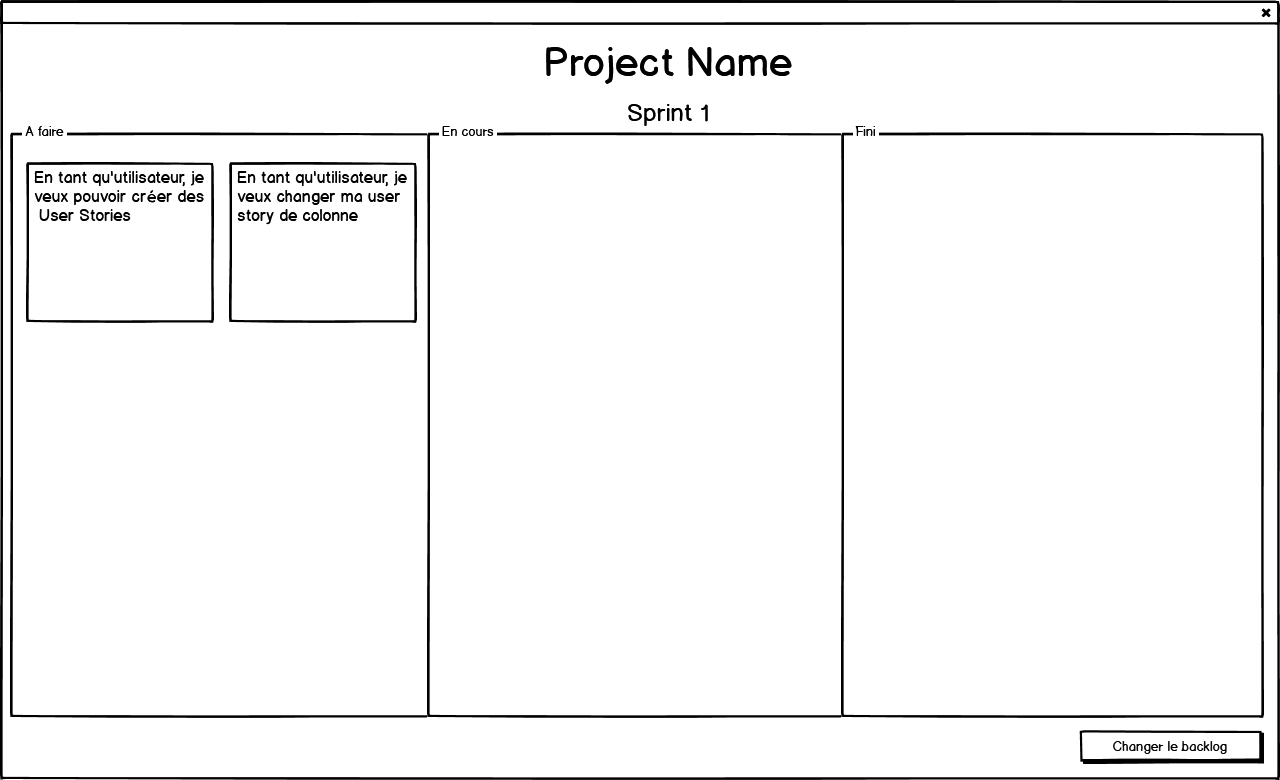
### Fenêtre de projet



Cette fenêtre s’affichera dès l’ouverture d’un projet. On peut changer le nom du projet ainsi que sa description en cliquant sur le bouton à côté du nom ou en déclenchant le menu contextuel via un appui prolongé.

La partie gauche permettra de créer des fonctionnalités à rajouter dans le projet à travers des Use Case ou les modifier alors que la partie de droite permettra de rajouter et d’éditer des Sprints et leur rajouter des Use Case via un glisser-déposer sur le sprint voulu. Les sprints déjà passé dans la date seront légèrement grisé et ne pourront pas être modifié alors que l’actuel sera mis en évidence.

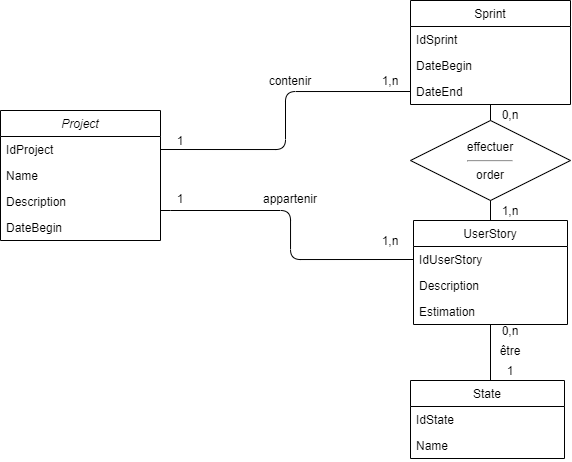
### Fenêtre de sprint



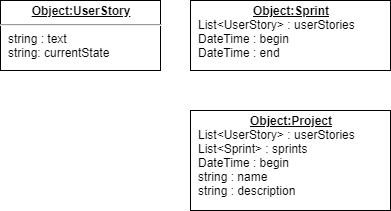
Cette fenêtre affiche le sprint sur lequel on aura appuyé. Un bouton permet de revenir au backlog. Un appui prolongé ouvrira le menu contextuel qui permettra de rajouter des Use Case ou modifier les Use Case existant.

Le menu contextuel permettra de rajouter des colonnes pour permettre aux gens de gérer au mieux leur projet de la façon qui leur convient le mieux.

## Modèle de données



## Modèle de classe



## Planning

## SWOT

### Forces

Ce projet permettra d’apporter une gestion plus facile et centralisée des projets avec les méthodes agiles. De plus, le fait de pouvoir interagir avec les tableaux rien qu’en les touchant rendra l’application encore plus intuitive.

### Faiblesses

Ce projet est cependant un peu faible techniquement. C’est pourquoi la détection multipoint est indispensable et représentera le défi de l’application afin d’être fonctionnel et intéressant pour l’utilisateur.

### Opportunités

C’est la première fois que je crée une application prévue pour un appareil tactile avec C# et j’ai donc eu l’opportunité de pouvoir m’atteler à un projet qui pourra être utiliser tout en découvrant le multi-touch avec C#.

### Menaces

Malheureusement, le mur tactile n’est plus mis à jour et la dernière version des pilotes du mur tactile est prévue pour Windows 8. De plus, le mur tactile est difficile à calibrer afin de ne pas avoir d’erreur de réception.

Enfin, le Covid-19 étant survenu juste avant le début du projet, je ne peux pas commencer mon projet sur le matériel ciblé. Même si l’école d’informatique m’a fourni un matériel permettant d’accomplir mon projet tout en restant confiné, la différence entre le mur et l’écran pourraient me nuire.

## Outils

### Interface graphique

Afin de réaliser l’interface graphique, j’ai décidé d’utiliser WPF. En effet, le multi touch est déjà géré sur WPF et non sur Windows Form. De plus, WPF permet de créer des applications multi-plateformes.

### Base de données

Afin d’intégrer une base de données, j’ai choisi d’utiliser Access car c’est une technologie avec laquelle je suis déjà familier. En effet, n’étant déjà pas familier avec WPF et la détection multipoint, j’ai pensé qu’il serait mieux de ne pas miser sur trop d’inconnus et utiliser un système de base données que je connais bien et qui est très bien intégré avec C#.

## Environnement

Afin de réaliser ce projet, j’ai à ma disposition un ordinateur avec Windows 10 ainsi que l’édition Community de Visual Studio 2019. On m’a également fourni un écran tactile Iiyama ProLite T2735MSC afin de palier aux problèmes du Covid-19