

### Méthodologie séquentielle :

#### Les principes :

Il faut définir une maîtrise d'ouvrage (le client ou prestataire du client) et une maîtrise d'œuvre (l'équipe chargée de la réalisation).

1. La maîtrise d'ouvrage doit écrire les spécifications fonctionnelles avec les comportements de l'application (la bible du projet).
2. La maîtrise d'œuvre écrit les spécifications techniques et conçoit l'architecture globale de l'application (cadre technique).
3. L'écriture du code est réalisée par les développeurs en se basant sur les spécifications et le diagramme.
4. La vérification de l'application et de ses fonctionnalités est réalisée à l'aide de tests unitaires pour tester le code.
5. Enfin la maîtrise d'ouvrage réalise des tests de recette en testant manuellement l'application et sa bonne conformité.

#### Les avantages :

Elle permet :

- de réaliser les phases les unes après les autres dans un ordre chronologique afin de ne rien oublier,
- d'estimer le temps et le budget pour la réalisation du projet,
- de simplifier la relation avec le client en évitant le contact systématique.

#### Les inconvénients :

- Le logiciel étant testé uniquement à la fin du projet, le client ne verra le résultat que plusieurs semaines/mois après le démarrage de celui-ci.
- Chaque étape conditionnera la suivante. Il faudra donc réaliser :
  - o 100% des spécifications fonctionnelles avant de commencer la conception,
  - o 100% des spécifications techniques avant de commencer le code,
  - o 100% du code avant de commencer les tests.

### Méthodologie agile :

#### Les principes :

C'est une approche itérative qui améliore un résultat et incrémentale qui permet la construction en découpant les tâches.

La réalisation se fait de manière collaborative dans un délai contraint. Elle permettra la création d'un logiciel ou d'une application optimale qui répond aux besoins changeants des utilisateurs.

On ne développera pas nécessairement toutes les implémentations de fonctionnalités qui ne sont pas nécessaires. Par contre, on pourra facilement intégrer des changements et modifier les priorités en ayant une approche séquentielle.

#### Les avantages :

La méthode est :

1. Itérative : permet une amélioration continue d'une fonctionnalité à travers les erreurs rencontrées lors de la réalisation.
2. Incrémentale : le découpage des fonctionnalités permet de se concentrer sur celles qui sont essentielles, de mettre régulièrement en production des fonctionnalités afin de les présenter aux clients et de les modifier si nécessaire.
3. Adaptative : facilite la communication entre les utilisateurs pour la réalisation du projet. Celui-ci évolue rapidement grâce aux échanges réguliers.
4. Collaborative : chaque membre peut choisir sa tâche à réaliser. L'accent est mis sur le produit qui vise à être le plus proche des attentes des utilisateurs. La collaboration entre développeurs est grandement préconisée.

#### Les inconvénients :

- Il est impératif de bien choisir son équipe. Les interactions entre les membres eux-mêmes ainsi que le client sont primordiaux afin d'échanger sur l'avancement du projet. Cela nécessite une forte implication de chacun pour éviter tout retard et répondre aux besoins du client.

- Il sera compliqué de modifier des éléments en cours de route car en cas de modification, l'ensemble des étapes devra être corrigé.

**Les types de projets à proscrire avec cette méthode :**

- Les projets ayant une certaine incertitude et une évolution rapide dû aux exigences des utilisateurs.
- Ceux avec des budgets et un temps de développement sans réelles contraintes et avec des méthodes mettant l'accent sur la flexibilité et l'adaptation.

- Il est également difficile de visualiser le projet dans son intégralité car certaines fonctionnalités sont développées au fur et à mesure.
- Les délais et le budget ne peuvent pas être précis sauf en délimitant des ordres de priorité.

**Les types de projets à proscrire dans cette méthode :**

- Les projets qui ont des exigences claires et définies dès le départ.
- Ceux qui impliquent des infrastructures de grande envergure et des exigences de sécurité strictes qui requièrent une planification et une conception approfondies.
- Ceux qui ont des temps définis, des coûts et des demandes spécifiques
- Les organisations qui n'ont pas de capacité d'adaptation.

## **Le choix de la méthodologie**

Pour le développement de cette application Android, il est recommandé d'utiliser une méthodologie agile, telle que Scrum (mêlée). En plus d'avoir une plus grande flexibilité, elle permet de répondre rapidement aux exigences changeantes du client.

Dans un projet Scrum, le travail est découpé en sprints (*cycles*) de deux à quatre semaines, au cours desquelles les membres de l'équipe travaillent sur les éléments prioritaires du backlog (*un grand tableau*).

Un sprint est découpé en story :

- Manuel utilisateur (spécifications fonctionnelles),
- Documentation technique (spécifications techniques),
- Code,
- Tests.

À la fin de chaque sprint, l'équipe présente une version fonctionnelle de l'application qui peut être testée par les utilisateurs et les parties prenantes.

Cette approche offre plusieurs avantages pour un projet de développement d'application Android :

- Elle permet de livrer rapidement des fonctionnalités, ce qui peut aider à obtenir des retours rapides sur le produit tel que le déploiement des fonctionnalités liés au contrôle du stock.
- Elle permet une collaboration étroite avec le client ou les parties prenantes, pour garantir que le développement de l'application répond à leurs besoins.
- Elle offre une flexibilité pour faire des ajustements au produit en cours de route, en fonction des retours d'expérience et des besoins évolutifs.
- Elle favorise la communication au sein de l'équipe de développement, ce qui peut améliorer la qualité du produit final.

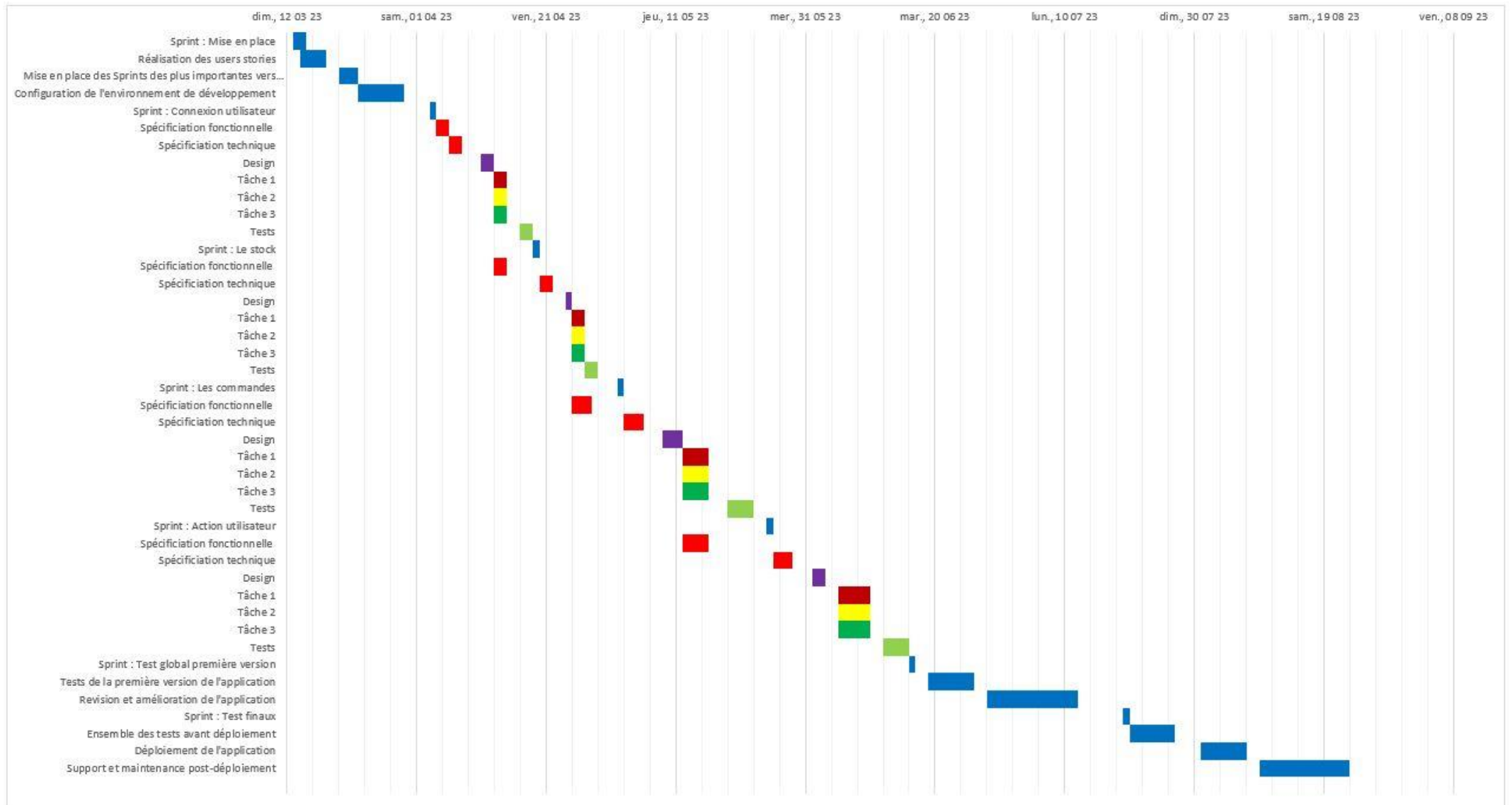
La contrainte de temps (dans notre cas un délai de 6 mois) :

La méthode Scrum peut être adaptée en fonction des besoins du projet pour s'assurer que les objectifs sont atteints dans les délais impartis. Elle nécessite une bonne planification et une forte collaboration entre les membres de l'équipe pour la réussite du projet.

## Macroplanning OC Pizza dans le cadre du développement de l'application android

Tâche	Qui ?	Date de début	Jours prévus	Date de fin	Les acteurs
<b><u>Sprint : Mise en place</u></b>	Tous	lundi 13 mars 2023	1 jour(s)	lundi 13 mars 2023	Tous (OCPIZZA, Julien, Emilie, Nathalie, Lucas, Corentin, Grégoire, Eric, Futur Chef de projet)
Réalisation des users stories	Tous	mardi 14 mars 2023	4 jour(s)	vendredi 17 mars 2023	Leader aspect technique (Julien)
Mise en place des Sprints des plus importantes vers les moins	Tous	lundi 20 mars 2023	3 jour(s)	mercredi 22 mars 2023	Leader aspect technique (Julien)
Configuration de l'environnement de développement	Tous	jeudi 23 mars 2023	7 jour(s)	vendredi 31 mars 2023	Equipe studio designer (Emilie)
<b><u>Sprint : Connexion utilisateur</u></b>	Tous	lundi 3 avril 2023	1 jour(s)	lundi 3 avril 2023	Développeuse (Nathalie)
Spécification fonctionnelle	Leader (Julien)	mardi 4 avril 2023	2 jour(s)	mercredi 5 avril 2023	Développeur (Lucas)
Spécification technique	Leader (Julien)	jeudi 6 avril 2023	2 jour(s)	vendredi 7 avril 2023	Développeur (Corentin)
Design	Designer (Emilie)	mardi 11 avril 2023	2 jour(s)	mercredi 12 avril 2023	Assurance Qualité (Grégoire)
Tâche 1	Dev (Nathalie)	jeudi 13 avril 2023	2 jour(s)	vendredi 14 avril 2023	
Tâche 2	Dev (Lucas)	jeudi 13 avril 2023	2 jour(s)	vendredi 14 avril 2023	
Tâche 3	Dev (Corentin)	jeudi 13 avril 2023	2 jour(s)	vendredi 14 avril 2023	
Tests	Assurance Qualité (Grégoire)	lundi 17 avril 2023	2 jour(s)	mardi 18 avril 2023	
<b><u>Sprint : Le stock</u></b>	Tous	mercredi 19 avril 2023	1 jour(s)	mercredi 19 avril 2023	
Spécification fonctionnelle	Leader (Julien)	jeudi 13 avril 2023	2 jour(s)	vendredi 14 avril 2023	
Spécification technique	Leader (Julien)	jeudi 20 avril 2023	2 jour(s)	vendredi 21 avril 2023	
Design	Designer (Emilie)	lundi 24 avril 2023	1 jour(s)	lundi 24 avril 2023	
Tâche 1	Dev (Nathalie)	mardi 25 avril 2023	2 jour(s)	mercredi 26 avril 2023	
Tâche 2	Dev (Lucas)	mardi 25 avril 2023	2 jour(s)	mercredi 26 avril 2023	
Tâche 3	Dev (Corentin)	mardi 25 avril 2023	2 jour(s)	mercredi 26 avril 2023	
Tests	Assurance Qualité (Grégoire)	jeudi 27 avril 2023	2 jour(s)	vendredi 28 avril 2023	
<b><u>Sprint : Les commandes</u></b>	Tous	mardi 2 mai 2023	1 jour(s)	mardi 2 mai 2023	
Spécification fonctionnelle	Leader (Julien)	mardi 25 avril 2023	3 jour(s)	jeudi 27 avril 2023	
Spécification technique	Leader (Julien)	mercredi 3 mai 2023	3 jour(s)	vendredi 5 mai 2023	
Design	Designer (Emilie)	mardi 9 mai 2023	3 jour(s)	jeudi 11 mai 2023	
Tâche 1	Dev (Nathalie)	vendredi 12 mai 2023	4 jour(s)	mercredi 17 mai 2023	
Tâche 2	Dev (Lucas)	vendredi 12 mai 2023	4 jour(s)	mercredi 17 mai 2023	
Tâche 3	Dev (Corentin)	vendredi 12 mai 2023	4 jour(s)	mercredi 17 mai 2023	
Tests	Assurance Qualité (Grégoire)	vendredi 19 mai 2023	4 jour(s)	mercredi 24 mai 2023	
<b><u>Sprint : Action utilisateur</u></b>	Tous	jeudi 25 mai 2023	1 jour(s)	jeudi 25 mai 2023	
Spécification fonctionnelle	Leader (Julien)	vendredi 12 mai 2023	4 jour(s)	vendredi 19 mai 2023	
Spécification technique	Leader (Julien)	vendredi 26 mai 2023	3 jour(s)	mercredi 31 mai 2023	
Design	Designer (Emilie)	jeudi 1 juin 2023	2 jour(s)	vendredi 2 juin 2023	
Tâche 1	Dev (Nathalie)	lundi 5 juin 2023	5 jour(s)	vendredi 9 juin 2023	
Tâche 2	Dev (Lucas)	lundi 5 juin 2023	5 jour(s)	vendredi 9 juin 2023	
Tâche 3	Dev (Corentin)	lundi 5 juin 2023	5 jour(s)	vendredi 9 juin 2023	
Tests	Assurance Qualité (Grégoire)	lundi 12 juin 2023	4 jour(s)	jeudi 15 juin 2023	
<b><u>Sprint : Test global première version</u></b>	Tous	vendredi 16 juin 2023	1 jour(s)	vendredi 16 juin 2023	
Tests de la première version de l'application	Tous	lundi 19 juin 2023	7 jour(s)	mardi 27 juin 2023	
Revision et amélioration de l'application	Tous	mercredi 28 juin 2023	14 jour(s)	mardi 18 juillet 2023	
<b><u>Sprint : Test finaux</u></b>	Tous	mercredi 19 juillet 2023	1 jour(s)	mercredi 19 juillet 2023	
Ensemble des tests avant déploiement	Tous	jeudi 20 juillet 2023	7 jour(s)	vendredi 28 juillet 2023	
Déploiement de l'application	Tous	lundi 31 juillet 2023	7 jour(s)	mardi 8 août 2023	
Support et maintenance post-déploiement	Tous	mercredi 9 août 2023	14 jour(s)	mardi 29 août 2023	

## Diagramme de GANTT OC Pizza dans le cadre du développement de l'application android



## Matrice RACI OC Pizza dans le cadre du développement de l'application android

Tâche	Lola et Franck (OC_PIZZA)	Eric (Chief Technical Officer (CTO))	Alexandra (Responsable)	Futur chef de projet	Karim (collègue expérimenté)	Julien (LEADER DEV)	Emilie (DESIGNER)	Nathalie (Dev)	Lucas (Dev)	Corentin (Dev)	Grégoire (QA)
Spécifications fonctionnelles	C		A		I	R					
Spécifications techniques	C		A		I	R					
Solution fonctionnelle	C		A		I	R					
Diagramme de classe	C	A				R	I	I	I	I	I
Modèle physique de données	C	A				R	I	I	I	I	I
Réalisation des scripts SQL	C	A				R	I	I	I	I	I
Choix de la méthodologie de projet	C	A				R	I	I	I	I	I
Macroplanning et diagramme de GANTT	C	A				R	I	I	I	I	I
Réalisation d'un RACI	C	A				R	I	I	I	I	I
Création des designs	C	I		A		C	R	I	I	I	I
Conception de l'architecture	C	I		A		R	R	R	R	R	I
Développement des fonctionnalités	I	C		A		R	C	R	R	R	I
Tests unitaires	I	C		A		C	I	I	I	I	R
Tests d'intégration	I	C		A		C	I	I	I	I	R
Tests de validation	I	C		A		C	I	I	I	I	R
Documentation technique	C	C		A		R	R	R	R	R	R
Validation client	C	A	R	C		I	I	I	I	I	I
Déploiement	C	A	R	C		I	I	I	I	I	I
Formation des employés d'OC Pizza	C	A	R	C		I	I	I	I	I	I

### Légende

R	<u>Réalisé</u>
A	<u>Approbateur</u>
C	<u>Consulté</u>
I	<u>Informé</u>