Spécifications techniques

OC Pizza

Open Classrooms

Auteur : Mickaël HORN – Développeur Junior



Table des matières

**Aucune entrée de table des matières n'a été trouvée.**

# Contexte du projet

OC Pizza est un groupe récent de pizzeria qui connaît un franc succès et se développe rapidement.

Fondé par Franck et Lola, OC Pizza est spécialisé dans les livraisons et dans les pizzas à emporter.

Aujourd’hui, le groupe compte 5 points de ventes et a pour projet d’en ouvrir 3 de plus d’ici 6 mois.

OC Pizza cherche à changer de système informatique car il ne correspond plus aux besoins du groupe.  
En effet, le système actuel ne permet pas de gérer de manière centralisée et individuelle les différentes pizzerias.

Les responsables ne peuvent pas suivre correctement ce qui se passe dans les points de ventes et les livreurs n’ont pas la possibilité d’indiquer en temps réel que la livraison a bien été effectuée.

Voici une liste exhaustive des besoins du groupe :

* Améliorer la gestion des commandes en passant par le cycle de vie complet, c’est-à-dire la réception, la préparation et pour finir, la livraison
* Avoir un suivi en temps réel des commandes en préparation et en livraison
* Disposer du stock en direct pour savoir quelles pizzas peuvent être réalisées ou non
* Mise en place d’un site Internet
  + Passer des commandes
  + Payer en ligne
  + Modifier ou annuler la commande si elle n’a pas encore été préparée
* Aider les pizzaiolos en leur proposant un aide-mémoire qui contiendra la recette de chaque pizza
* Recevoir les notifications quand la commande est prête à être livrée ou réceptionnée.

# Une image contenant texte Description générée automatiquementComparatif des méthodes

# Choix de la méthode

Après avoir pris en compte les différences parmi les méthodologies de projet citées plus haut, nous devons prendre une décision sur quelle méthode appliquer.

Le cycle en V semble adapté à notre cas, et je vais détailler pourquoi.

Le client a clairement exprimé ses besoins, sous forme de liste, que nous avons préalablement étayés dans les spécifications fonctionnelles, ce qui est déjà très rassurant pour la réalisation de ce projet.

Il semble savoir où il va et on peut donc en déduire qu’il ne devrait pas y avoir de modifications importantes qui pourraient compromettre notre méthodologie séquentielle et linéaire.

Nous pouvons nous passer de la méthodologie agile, car nous n’avons pas besoin d’avancer « à petit pas » étant donné qu’on a affaire à un projet bien cadré.

La méthodologie en cascade est plus simple, mais le cycle en V, du fait de ses tests complets, nous assureras une livraison de qualité.

Pour terminer, je m’appuie sur un élément concret qui est survenu lors de la construction des spécifications fonctionnelles qui est la date de livraison.

En effet, le client souhaite que le projet soit livré pour l’ouverture des 3 nouvelles pizzérias, c’est-à-dire dans 6 mois.

Le client a déjà son idée sur la date de livraison et nous devons lui fournir une vraie date.

C’est un des plus gros défauts de la méthodologie agile, ce manque de visibilité sur le temps, on n’a pas d’idée précise du temps et du coût final.

# Macroplanning

RACI (Responsibility Assignment matrix)

# Comparatif des différentes méthodologies

|  |  |  |  |
| --- | --- | --- | --- |
|  | Agile | Cascade | V |
| Principes clés | L'équipe, soit des individus et des interactions, plutôt que des processus et des outils  L’application, c'est-à-dire des fonctionnalités opérationnelles plutôt que de la documentation exhaustive  La collaboration avec le client, plutôt que la contractualisation des relations  L’acceptation du changement, plutôt que le suivi d'un plan | La production des livrables définis au tout début du projet  La livraison de ces livrables à une date précise et définie lors du cadrage du projet  La phase ne se termine que lorsque cette dernière a été vérifiée puis validée  Si un client n’est pas satisfait d’un livrable, l’équipe devra le retravailler jusqu’à ce qu’il soit parfait | La conception (analyse des besoins, spécifications, conception générale et détaillée) |
| Avantages | Flexibilité  Collaboration et communication fréquente avec le client  Meilleure visibilité du client sur le projet  Le client peut changer de direction s’il le souhaite  Maîtrise des coûts | Permet de tenir les délais et les coûts, ce qui n’étire pas le budget du client  Bonne visibilité du projet, car chaque phase est l’une à la suite de l’autre  Méthode appréciée grâce à sa grande simplicité  Bonne documentation | Planification et conception des tests avant le codage (pas certains)  Suivi proactif des défauts (ils sont détectés à un stade précoce)  Risques minimisés et meilleure qualité du produit grâce aux tests |
| Inconvénients | Documentation maigre car le dialogue est privilégié  Les clients doivent rester disponibles  Pas adapté pour les entreprises aux structures hiérarchiques très fortes, à cause du fonctionnement collaboratif  Malgré un bon contrôle des coûts, la vision du budget pour la totalité du projet est difficile | Impossible de revenir sur une phase déjà terminée et validée par le client  Processus de planification très long (codage retardé)  Si les exigences de départs sont défectueuses, le projet peut échouer  Intervention tardive de l’assurance qualité  Péremption du produit (projets assez longs, dont les besoins du client peuvent avoir évolués) | Comme le modèle en cascade, méthode cloisonnée (si une phase comporte des erreurs ou des oublis, la suivante s’en retrouve affectée)  Manque de communication (les acteurs ne jouent pas au même moment et communiquent surtout via les documents)  Péremption du produit (projets assez longs, dont les besoins du client peuvent avoir évolués) |
| Projets à proscrire | Les projets parfaitement bien cadrés dont l’échéance est bien définie, qui ne nécessite pas de personnalisations | Les projets dont les clients apportent des modifications régulières | |

Nouveau projet

* Modéliser les objets du domaine fonctionnel (Diagramme de classe UML MCD, avec multiplicité et types de relations) en PDF ou JPEG
* Initier la structure du domaine fonctionnel dans un SGBD (création de la BDD + modèle physique MPD + les scripts SQL pour créer la BDD vierge + jeu de données de test)
* Élaborer le schéma de la ou des bases de données que vous comptez créer (MLD)

Pour la présentation

* Choisir une méthodologie de projet adaptée au contexte (fait un tableau comparatif entre la méthodologie choisie et les autres méthodologies non choisies)
* RACI (Responsibility Assignment matrix), avec identification des différents acteurs, qui gère quoi dans le projet
* Macroplaning définissant les jalons du projet + estimations de livraison de ces derniers

Questions :

1. Au niveau du livrable 2, on demande soit un MPD ou un MLD en format PDF + une BDD au format SQL (livrable 3). Cependant, ce dernier point pour moi représente le MPD et le MLD est également demandé dans le mail plus haut par la schématisation de la BDD.
   1. Réponse : Tkt faut faire les 3 représentations : MCD / MLD / MPD
2. On constitue un document de spécification techniques ? Car dans les livrables, on doit tous les séparer.
   1. Réponse : Oui ! Tu fais le document de spécifications techniques qui regroupe tout et également les livrables en séparé
3. Support de présentation au format PDF ? On ne peut pas utiliser de PP ?
   1. Réponse : Fait le PP que tu présenteras le jour de la soutenance, traduit le en PDF, il le faut
4. Plus besoin des anciens diagrammes de l’ancien projet (organisation physique des composants + les différents composants du système avec leurs interactions)
   1. Réponse : Non plus besoin ! C’est peut-être d’ailleurs une des raisons pour laquelle le projet a été remodulé