

M3301 : Méthodologie de la production d'application

*Rédaction du document de gestion des
risques du groupe n°7 :*

“Mijotons”



Poursuite du projet du semestre 2 de l'année 2020 - 2021 (Groupe 25)



ABDELHAK Yanis

MARIS Xan

OSWALD Bastien

ROYET Jules

DUT INFORMATIQUE

Semestre 3 | 2021 -2022

TD1 - TP2

Commanditaire & tuteur : DOURISBOURE Yon

Présentation générale du projet	2
Identification des niveaux de risques	3
Niveau 1	3
Niveau 2	3
Niveau 3	3
Niveau 4	3
Risques non traités et justification	4
Erreurs stratégique	4
Difficultés de partenariat	4
Identification des risques et proposition de résolution	5
Difficultés de management	5
Cas du commanditaire indisponible : niveau 2	5
Cas du collègue indisponible : niveau 3	5
Cas d'une mauvaise communication : niveau 3	6
Cas d'une mauvaise gestion du planning : niveau 3	6
Difficultés financières	6
Cas de la commercialisation de l'un des logiciels utilisés : niveau 4	6
Difficultés de marché	7
Cas d'une accusation de plagiat : niveau 2	7
Défaillances techniques	7
Cas d'impossibilité d'utilisation d'un outils : niveau 3	7
Cas de pertes de données : niveau 4	7
Cas d'un outil externe inutilisable : niveau 3	8
Obstacles légaux	8
Cas du non respect des conditions de distribution d'Open Food Facts : niveau 3	8
Problèmes de sécurité	9
Cas du vol/piratage de données utilisateur	9
Problème d'inadaptabilité utilisateur	9
Cas d'une interface pas assez intuitive	9
Conclusion	10

Présentation générale du projet

Dans le cadre de la seconde année du DUT Informatique, nous devons réaliser une application. C'est dans ce cadre que notre groupe, composé de Abdelhak Yanis, Maris Xan, Oswald Bastien et Royet Jules, a décidé, sous l'impulsion de notre enseignant tuteur et commanditaire Dourisboure Yon, de réaliser Mijotons dont nous avons commencé la conception au semestre 2 de l'année 2020 - 2021 (anciennement, groupe n°25). Notre projet concerne le développement d'une application native multi-fonctionnelle, mono-plateforme que nous avons baptisée : "Mijotons". C'est une application basée sur le thème de la cuisine, plus particulièrement : l'élaboration de recettes. Cette idée d'application nous est venue l'an dernier alors que nous venions de rentrer en DUT. En effet, nous avons dû être confrontés à de nouvelles problématiques personnelles telles que : la gestion des courses et celle des repas. De ce fait, nous avons cherché des repères afin d'optimiser ces petites tâches du quotidien, mais nous n'avons rien trouvé à la hauteur de nos exigences. Alors nous est venu l'idée de développer l'application qui résoudrait les problèmes que nous avons eus l'an dernier et qui, par conséquent, aiderait les futurs étudiants du BUT de l'IUT, et autres formations naturellement. Bien que notre application soit disponible et utilisable par le plus grand nombre, l'objectif de notre application reste d'améliorer principalement la qualité de vie de nos confrères étudiants en optimisant le temps des tâches quotidiennes et le coût de celles-ci. De plus, l'un des effets indirects mais non-négligeables de notre application est qu'elle vise à réduire le gaspillage alimentaire en proposant des recettes avec nos restes de nourriture.

Mijotons sera une application native, nous allons la développer en C#(CSharp) via Xamarin et la mettre en ligne sur le Play Store une fois terminée.

D'autre part, nous aurons un seul et unique acteur pour cette application : l'utilisateur.

Cet acteur aura donc accès aux différentes fonctionnalités suivantes :

- l'utilisateur pourra accéder à la fonctionnalité de recherche de recette en fonction des ingrédients sélectionnés mais aussi à la fonctionnalité de recherche de restaurant.
- d'autre part, l'acteur pourra utiliser les fonctions de gestion de liste de courses mais aussi de gestion de favoris et d'historique.

Identification des niveaux de risques

La hiérarchisation des risques est une étape importante dans la gestion des risques et donc par conséquent dans le développement de notre projet car celle-ci permet de classer le niveau de gravité de la réalisation ou non-réalisation d'une tâche.

Nous avons donc opté pour une hiérarchisation en 4 niveaux de risque :

Niveau 1

Le risque peut être plus ou moins laissé de côté, car il ne met pas en péril le reste du développement.

Niveau 2

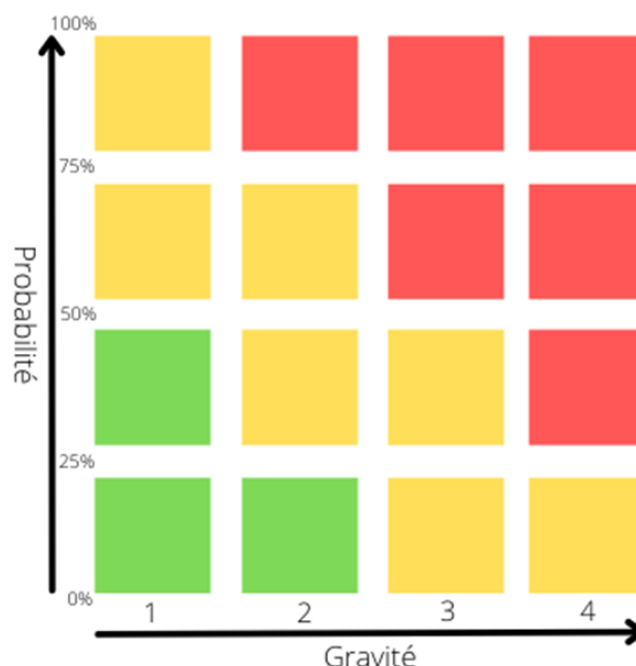
Le risque commence à être encombrant pour le déroulement du projet, il peut être mis de côté au maximum une semaine car sinon les conséquences sont trop lourdes à assumer.

Niveau 3

Le risque est conséquent, il peut gravement mettre en péril notre projet, il doit donc être traité dans les 3 jours qui suivent.

Niveau 4

Le risque est inacceptable, le développement est mis en attente et le risque doit être corrigé immédiatement !



Risques non traités et justification

Erreurs stratégique

Une Erreur stratégique pourrait se traduire par la création et la réalisation de projet non-conforme avec la finalité de l'organisation, ou bien la réalisation d'un projet non adapté à l'environnement de l'organisation.

Nous n'allons pas traiter le risque de produire des erreurs stratégiques car celles-ci opèrent au sein d'organisations. Or, nous ne sommes pas une organisation, par conséquent, cela ne nous concerne donc pas.

Difficultés de partenariat

Un partenariat est une association d'entreprises, d'institutions en vue de mener une action commune.

Une difficulté de partenariat apparaît dans le cas d'un partenaire frauduleux ou bien non-fiable. Dans notre cas, le taux d'apparition de ce risque est nul car nous ne ferons aucun partenariat. Il est donc évident que nous ne traiterons pas d'élément de résolution pour ce risque.

Identification des risques et proposition de résolution

Nous avons choisi de présenter notre gestion des risques projet par type de risque car cette organisation nous permet de parler de tous les risques possibles pour notre application, même ceux qui n'auront sûrement pas d'impacts.

Difficultés de management

Les difficultés de management sont des obstacles liés à une mauvaise communication entre une ou plusieurs équipes de projet mais aussi à une mauvaise conception du projet. En outre, ce sont des tâches plutôt délicates à traiter dans notre cas car, notre groupe de projet est à la fois la MOE et la MOA. Nous sommes donc à la fois clients et concepteurs/développeurs, il n'y a donc pas de grosses équipes annexes à gérer.

Cas du commanditaire indisponible : niveau 2

Nous allons traiter le cas du commanditaire indisponible. Ce risque a une probabilité de 40 % de faire surface. Nous avons calculé ce taux à partir du délai entre la réception du mail de demande de rendez-vous par notre tuteur et sa réponse au cours de nos dernières entrevues. Son niveau 2 est justifié par le fait que ce risque aurait pour conséquence de ralentir l'avancement de notre projet voire même de geler certains éléments.

Notre solution, pour combler ce risque est de relancer notre professeur tuteur par mail, mais aussi d'essayer de le croiser dans son bureau ou lors de cours d'enseignements au sein de l'IUT. En outre, nous continuerons tout de même à avancer sur le projet en parallèle en procédures dégradées.

Cas du collègue indisponible : niveau 3

Ce risque a une probabilité de 30 % de survenir. Celui-ci est classé niveau 3 car il aurait pour conséquence de nous obliger à revoir notre planning ainsi que les tâches affectées à chacun des acteurs (membre du groupe).

Dans un cas pareil, le travail assigné au collègue en incapacité de travailler sera réparti de manière égale au sein des autres membres du groupe. Par ailleurs, nous travaillerons plus afin de livrer le projet dans les délais impartis.

Cas d'une mauvaise communication : niveau 3

Avec une probabilité d'apparition de 60%, la mauvaise entente au sein du groupe, provoquant notamment la rupture de la communication et par conséquent le ralentissement voir la mise en arrêt total du projet est classée niveau 3 car il est évident qu'elle serait une impasse conséquente au rendu du projet.

Si cette situation venait à faire surface, nous serions dans l'obligation de faire une réunion en groupe de projet (avec si besoin l'aide du professeur tuteur) afin de solutionner le problème en prenant une décision convenant à tout le monde.

Cas d'une mauvaise gestion du planning : niveau 3

Nous avons estimé la probabilité de ce risque à 30 %, car nous pensons avoir passé assez de temps sur la production du planning pour l'application. De ce fait, nous avons pris des amplitudes assez conséquentes pour chaque tâche, nous permettant d'avoir une certaine aisance.

Néanmoins, ce risque est tout de même classé niveau 3 car dans le cas où il arriverait, cela provoquerait une réédition totale de notre travail en amont, ce qui serait donc une tâche conséquente.

Par ailleurs, pour se prémunir de ce genre de risque, nous nous devons de réétudier au fur et à mesure de la réalisation de notre projet, les fonctionnalités et délais de chaque tâche afin de s'assurer que le tout est réalisable dans les temps que nous avons convenus. Nous devons également effectuer un versionning de l'application via la méthode MoSCoW, permettant de passer d'une version à l'autre en cas de manque de temps.

Difficultés financières

Les difficultés financières correspondent à une mauvaise gestion du budget pouvant notamment entraîner de véritables gouffres financiers.

Cas de la commercialisation de l'un des logiciels utilisés : niveau 4

Ce cas de n'est pas vraiment très fréquent, d'où sa faible probabilité d'apparition (10%). En effet, les sociétés ne peuvent changer de méthode de commercialisation pour un produit d'un jour à l'autre. En outre, ce genre de décision stratégique est médiatisé, ce qui nous laisserais le temps de rebondir. Néanmoins, dans le cas où l'on se trouverait dans cette situation, il nous faudrait reprendre le code de 0 avec un nouveau logiciel de programmation, ce qui est donc une perte de temps monumentale quant à l'avancement du projet. Son niveau est donc placé à 4 en raison des conséquences qu'il entraîne. Nous avons donc prévu certains recours en cas de problèmes comme coder en Java en utilisant React Native ou Flutter.

Difficultés de marché

La difficulté de marché correspond à une clientèle mal ciblée par la MOE ou bien une présomption de copie à l'égard de notre application.

Cas d'une accusation de plagiat : niveau 2

Cette situation a peu de chance de se produire car nous avons rigoureusement observé les entreprises concurrentes de notre application lors de la phase d'étude de l'existant. Sa probabilité est donc placée à 25%. D'autre part, nous voulons limiter la participation à de telles procédures judiciaires afin d'éviter la perte de temps et donc par conséquent le ralentissement du projet.

Dans le cas de l'apparition d'une telle situation, nous entrerons également en procédure dégradée. L'élément lié à la mise en abîme du projet sera immédiatement modifié et/ou supprimé selon les convenances de l'intéressé car nous ne voulons pas entamer de procédure judiciaire à l'égard de ce projet.

Défaillances techniques

Cas d'impossibilité d'utilisation d'un outils : niveau 3

Les défaillances techniques correspondent à des soucis de technologies non maîtrisées, des risques difficiles à prévoir en amont et identifiables seulement au moment de la réalisation de celui-ci.

Ce problème a une faible probabilité de faire surface (ici 20%) car l'IDE que nous utilisons fonctionne avec CSharp, un dérivé de C que l'on étudie en ce moment en cours de système. Dans le cas où l'on serait entièrement perdu, on pourrait basculer sur un IDE fonctionnant avec un langage que l'on connaît mieux comme C++ par exemple ou bien Java. En outre, nous nous devons de commencer à chercher des tutos manuscrits et vidéos afin de se renseigner.

Cas de pertes de données : niveau 4

La perte de données de l'application telle que la suppression ou modification totale du code de l'application peut avoir de lourdes conséquences sur l'avancée du code. En outre, ce type de situation peut arriver régulièrement surtout lors d'utilisation de framework de versionning d'application tel que Git par exemple. De ce fait, nous avons placé sa probabilité à 40% et son risque niveau 4. Il est évident que ce genre de situation entraînerait un ralentissement du projet dans le cas où nous n'aurions aucune sauvegarde de celui-ci.

Pour pallier cela, nous devons régulièrement sauvegarder notre projet au fur et à mesure que nous codons et implémentons des outils externes.

Cas d'un outil externe inutilisable : niveau 3

Nous utilisons dans notre cas, des API externes tels que l'API d'Open Food Facts qui synthétise un large nombre d'ingrédients de cuisine mais aussi de recettes. Il est donc nécessaire de traiter les situations où cet API est indisponible ou inexploitable. Cette base de données est déjà en ligne depuis une dizaine d'années, il est donc peu probable que celle-ci disparaisse. En outre, si elle venait à disparaître, cela handicaperait grandement notre projet car toutes nos fonctionnalités cesseraient de fonctionner, d'où son niveau 3.

Afin de résoudre ce problème, nous pensons déjà à développer nos propres bases de données sur les serveurs d'Amazon DB, chose que nous maîtrisons d'or et déjà grâce à nos cours.

Obstacles légaux

Les obstacles légaux correspondent à des contrats de travail mal rédigés avec son client ou encore le projet enfreint certaines lois informatiques décrétées par l'Etat.

Cas du non respect des conditions de distribution d'Open Food Facts : niveau 3

Ce problème a très peu de chances d'arriver car nous allons lire soigneusement en amont toutes les conditions nécessaires à l'utilisation et l'exploitation de l'API et des données d'Open Food Facts, c'est pour cette raison que sa probabilité d'apparition est placée à 5%. Néanmoins, cela aurait pour conséquence de potentielles poursuites en justice qui mettrait la production de l'application à l'arrêt total, d'où son niveau 3 de gravité.

Nous n'avons donc pas de réels recours à cette situation si ce n'est que de lire attentivement les conditions déjà existantes et éventuellement se tenir informer de potentiels changements de celles-ci.

Problèmes de sécurité

Les problèmes de sécurité d'une application correspondent à des ouvertures non maîtrisées du SI, avec notamment accès à des sessions utilisateurs et donc vol de données.

Cas du vol/piratage de données utilisateur

Nous allons donc traiter le cas de l'intrusion qui a une probabilité d'apparition de 10% car il est évident qu'avant de mettre l'application sur le marché, nous allons réaliser une batterie de tests en tout genre notamment au niveau de la sécurité pour nous assurer de l'étanchéité totale de notre application.

Néanmoins, les données à voler sur nos utilisateurs sont minimales et inexploitable car il s'agit des recettes favorites et des recettes consultées (historique).

Dans le pire des scénarios, un utilisateur pourrait nous amener en justice pour cela, mais ce cas reste rare. En outre, il est évident que si nous nous rendons compte de la faille, nous la re-boucherons au plus vite en ayant évidemment retiré au préalable et temporairement notre application du marché.

Problème d'inadaptabilité utilisateur

L'inadaptabilité utilisateur fait référence à l'incapacité pour un utilisateur d'utiliser l'application correctement. Le projet n'est donc pas accepté par les utilisateurs.

Cas d'une interface pas assez intuitive

Ce problème a pour conséquence le fait que l'application soit inutilisée et rejetée par les utilisateurs. Sa probabilité d'apparition est d'environ 30% car ce genre de problème est rare mais il ne faut tout de même pas le négliger.

En outre, son niveau de gravité est placé à 4 car celui-ci entraînerait une refonte totale du CDCF et CDCT ainsi que de la programmation de l'application afin de solutionner le problème dans sa globalité.

Conclusion

En conclusion, nous dirons donc que la gestion des risques de notre projet nous a permis d'identifier plus amplement les différentes contraintes auxquelles nous pourrions être exposés mais aussi les conséquences de celles-ci. Par ailleurs, cela nous a d'autre part permis de nous positionner sur des scénarios de crise que nous avons élaborés pour chaque type de risque et que nous pourrions donc à l'avenir, mettre en pratique si ceux-là surviennent.

Finalement, bien que la plupart des risques entraînent des conséquences plutôt négatives ou neutres pour notre projet, certains de ces risques ont des conséquences plutôt positives comme par exemple la re-conception et le re-développement de certaines parties de l'application afin que celle-ci soit bonifiée.