

Le document que je vais vous présenter aujourd’hui est une synthèse du document de reporting complet que j’ai effectué pour l’entreprise.

----- **DIAPO** -----

Alors, au sommaire de ce document, nous effectuons une brève présentation des évènements important de notre startup.

Nous verrons ensuite, en quoi consiste notre activité.

Je vous présenterais également l’ensemble des processus que je planifie de mettre en place au sein de notre équipe technique.

Nous ferrons, ensuite, le point sur la planification des fonctionnalités à venir ainsi que nos objectifs pour l’année qui vient.

Enfin nous terminerons, sur le budget annuel pour la partie technique de l’activité.

----- **DIAPO** -----

Tout d’abord la startup.

----- **DIAPO** -----

Notre startup est une entreprise Française développant une application de réveil intelligent pour smartphone.

----- **DIAPO** -----

Je l’ai fondée en 2017 avec Alexandre Meunier.

----- **DIAPO** -----

Nous venons de lever 200 000 € auprès de business angels.

----- **DIAPO** -----

L’activité technique ensuite.

----- **DIAPO** -----

Notre application est donc une application pour smartphone qui écoute la respiration de l’utilisateur durant la nuit pour déterminer les phases les plus propices au réveil.

L’activité technique se résume pour le moment au seul développement de cette application sur la plateforme iOS.

L’application est encore au stade de prototype et contient de nombreux bugs mais possède néanmoins déjà des utilisateurs actifs.

----- **DIAPO** -----

À ce jour, notre application ne dispose que de deux fonctionnalités distinctes :

----- DIAPO -----

La possibilité d'ajouter un ou plusieurs réveils qui utiliseront notre algorithme pour déterminer la meilleure phase pour s'activer.

----- DIAPO -----

Une analyse statistique simplifiée de la qualité du sommeil des dernières nuits. Permettant à nos utilisateurs de déterminer le nombre d'heures moyen passé dans une phase de sommeil profond.

----- DIAPO -----

Afin de détecter les phases les plus propices au réveil notre application utilise un algorithme qui se base sur l'utilisation de réseaux neuronaux artificiels et l'apprentissage machine.

L'entraînement de notre réseau neuronal a nécessité des capacités informatiques très importantes compte tenu du nombre de données devant être traité.

À ce titre, nous avons décidé d'externaliser l'hébergement de l'architecture nécessaire sur le service de cloud d'Amazon.

Un logiciel adapté à l'apprentissage machine supervisé que nous utilisons dans notre algorithme a également été utilisé, il s'agit du logiciel TensorFlow.

----- DIAPO -----

Concernant les bugs qui impactent notre application maintenant.

Nous les avons listés ici selon leur sévérité, c'est-à-dire selon le pourcentage d'utilisateur touché et selon l'importance du crash (crash total de l'application ou simple bug d'affichage...).

Le niveau de sévérité a été défini ainsi :

1. **Sévérité haute** : crash ou bug bloquant et pourcentage d'utilisateur touché $\geq 20\%$.
2. **Sévérité moyenne** : crash ou bug bloquant et pourcentage d'utilisateur touché $\geq 5\% \text{ et } < 20\%$
3. **Sévérité faible** : crash ou bug bloquant et pourcentage d'utilisateur touché $< 5\%$. Tous les bugs non bloquants (affichage...).

----- DIAPO -----

Au total nous avons 4 bugs de sévérité haute

----- DIAPO -----

1 bug de sévérité moyenne

----- DIAPO -----

3 bugs de sévérité faible

----- DIAPO -----

----- DIAPO -----

Nous le savions avant de démarrer, le marché que nous tentons de conquérir est un marché déjà saturé. Le nombre d'applications tentant d'améliorer le réveil de ses utilisateurs est énorme.

C'est pourquoi, antérieurement au développement de notre application, nous avons effectué une analyse approfondie des forces en présence qui nous a permis d'en savoir plus sur les forces et les faiblesses de nos concurrents.

Cette analyse s'est attachée à conserver la plus grande impartialité quant aux avantages et inconvénients recensés.

Notre application n'étant pas encore disponible sur le système Android, et ne comptant pas l'être dans un avenir proche, nous nous sommes concentrés sur l'ensemble des applications disponibles sur iOS.

----- DIAPO -----

Premier concurrent le mode sommeil du système iOS.

Depuis la version 10.3.2 d'iOS, Apple propose à l'ensemble de ses utilisateurs une fonctionnalité censée améliorer la qualité du sommeil dénommée « Sommeil ».

Cette fonctionnalité permet à l'utilisateur de dormir le même nombre d'heures chaque nuit en fonction de son heure de réveil.

L'application se charge d'afficher une notification au moment du couché et de régler le réveil conformément au souhait de l'utilisateur.

Même si cette application n'effectue qu'un « simple » calcul d'heures, elle pourrait représenter un danger pour notre entreprise étant donné son taux de pénétration sur le marché (la totalité des smartphones iPhone exécutant une version égale ou supérieure à la version 10.3.2).

Les utilisateurs pourraient ne pas être incités à rechercher une autre application sur l'App store si celle-ci leur permettait réellement de faciliter leur réveil.

Avantages

1. Installée sur la totalité des iPhones exécutant une version égale ou supérieure à la version 10.3.2.
2. Entièrement gratuite.
3. Parfaitement intégré au téléphone sans nécessité d'installation.
4. Simple d'utilisation.
5. Ne nécessite pas de conserver son téléphone près de soi la nuit.

Inconvénients

1. Efficacité faible. Simple calcul d'heures ne se basant sur aucun paramètre scientifique (phases de sommeil profond, léger...).
2. Pas de fonctionnalité supplémentaire (une notification au coucher, un réveil le matin).

Différenciation

Notre meilleur avantage reste le fait de pouvoir réveiller l'utilisateur au moment le plus propice (i.e. : phases de sommeil léger) et donc de pouvoir garantir un réveil facilité à l'utilisateur.

Des fonctionnalités comme l'analyse statistique de la qualité du sommeil sur les derniers jours ou la comparaison pays par pays des heures habituelles de lever et de coucher constitueront également un avantage non négligeable.

----- DIAPO -----

----- DIAPO -----

L'équipe technique ensuite.

----- DIAPO -----

D'un point de vue organisationnel, je suis parti du principe que nous ne savions pas ce qui allait ou non fonctionner et donc que chaque fonctionnalité et chaque processus que nous souhaiterions mettre en place dans notre application ou dans l'équipe devrait prouver son efficacité avant d'être définitivement mis en place ou écarté.

Ce qu'Eric Ries, auteur du livre « Lean Startup » appelle « l'innovation continue ». Mettre en place, mesurer, apprendre, recommencer.

L'application de ce principe implique, entre autres, que la méthodologie de gestion de projet choisie permette la livraison d'un projet à intervalles réguliers et non d'un seul bloc.

Il implique également que des « indicateurs clés » de réussite ou d'échec soient mis en place pour chaque fonctionnalité et processus mis en place.

----- DIAPO -----

L'ensemble des contraintes inhérente à l'innovation continue exposées ci-dessus m'ont convaincu d'utiliser SCRUM qui se présente comme la méthodologie la plus populaire parmi les méthodes de gestion de projet « agiles » existantes.

Au contraire des méthodes traditionnelles qui se basent sur la livraison d'un projet en une fois d'un seul bloc, les méthodes dites « agiles » encouragent la découpe d'un projet en fonctionnalités et la livraison régulière d'un produit partiel mais utilisable.

Le côté très expérimental de cette méthode nous sera d'un grand secours dans notre quête d'apprentissage de ce qui fonctionne ou ne fonctionne pas.

----- DIAPO -----

Afin que la distribution des différentes tâches se fasse de la meilleure des façons possible, plusieurs compositions d'équipe étaient possibles.

Chaque composition comportait des avantages et des inconvénients qu'il m'a fallu prendre en compte pour faire le choix le plus adapté à notre situation.

Lors de l'utilisation d'« équipes métiers », chaque équipe est uniquement responsable de la mise en oeuvre d'un seul composant du projet. Les développeurs front-end s'occupent de l'interface utilisateur, les développeurs back-end de la logique métier et les designers du design.

Lors de l'utilisation d'« équipes de fonctionnalités », chaque équipe est responsable de la mise en oeuvre d'une fonctionnalité complète du projet.

L'utilisation simultanée d'équipes métiers et de fonctionnalités quant à elle est la jonction des deux.

En ce qui nous concerne

Compte tenu des ressources humaines réduites dont nous disposons à ce jour, une composition en tant qu'équipe métier semble impossible à réaliser.

À ce titre, j'ai choisi d'établir pour notre entreprise une composition des équipes en tant qu'équipes de fonctionnalités.

Chaque équipe travaillera donc de manière totalement autonome à la mise en place de fonctionnalités complètes sans dépendre des résultats de la mise en oeuvre dans une autre équipe.

----- DIAPO -----

Concernant les embauches envisagées maintenant, À ce jour, notre application Bower n'a toujours pas atteint ce que Marc Andreessen appelle le « product/market fit »¹, ce moment particulier où un produit répond à la demande non résolue d'un marché de taille importante et connaît une forte croissance.

Sans ce « product/market fit » nous ne pouvons faire de projections sur une éventuelle rentabilité future et sommes donc contraints, pour garantir notre survie le plus longtemps possible, de restreindre l'utilisation de l'argent déjà levé.

À ce titre, j'ai décidé de limiter les embauches à une seule et unique personne.

L'interface et l'expérience utilisateur de notre application sera quant à elle confiée à une personne en freelance.

Cette personne nous permettra de bénéficier d'un travail de type professionnel tout en bénéficiant d'une très grande flexibilité quant aux dépenses engendrées et au travail pouvant être demandé.

¹ http://pmarchive.com/guide_to_startups_part4.html

Et le développement de l'application pour Android ?

Plusieurs contraintes m'ont poussées à poursuivre le développement sur iOS et mettre de côté le développement pour Android.

Tout d'abord, l'état de notre application, encore au stade de prototype, et le peu d'utilisateurs que nous possédons, me font penser qu'il est capital pour nous de valider notre concept avant de penser se lancer sur un autre système.

Ensuite, mon absence de connaissances dans le développement Android, entraînerait l'impossibilité de juger de la qualité du code qui sera écrit. Si ce risque peut être pris lorsque le concept est validé et les premiers revenus générés il pourrait être fatal dans les premiers mois d'une startup.

De plus, la duplication massive, au niveau du design, du code source, des bugs et de la planification des fonctionnalités de l'application que cela entraînerait ralentira au maximum notre processus d'apprentissage.

Enfin, la planification séparée du projet, en plus des difficultés de gestion que cela entraînerait, provoquerait l'isolement total du développeur au sein de l'équipe et serait néfaste à l'esprit d'équipe nécessaire au début de toute startup.

----- DIAPO -----

Une fois, l'embauche effectuée, les équipes seront repartis de cette façon. L'équipe 1 composé du CTO. La deuxième équipe composée du développeur iOS que nous allons recruter.

----- DIAPO -----

Dans une contexte « agile » la gestion des bugs doit être prise en considération et ajustée en fonction de la planification du projet en sprints.

La première étape dans la mise en place d'une gestion cohérente est de disposer d'outils efficaces pour détecter et suivre l'état de chaque bug.

Détection et suivi des bugs



Deux canaux de détection des bugs sont utilisés à ce jour dans l'équipe :

1. Lors des « crashes » de notre application, c'est-à-dire lorsque celle-ci se ferme subitement un outil automatisé a été utilisé. Il s'agit de l'outil Crashlytics (<https://try.crashlytics.com/>).
2. Pour les « bugs », c'est-à-dire lorsque le comportement de l'application n'est pas celui souhaité, la détection sera effectuée de manière indirecte par les utilisateurs de notre application.

Les « bugs » et « crashes » de notre application seront ajoutés dans **GitHub** (<https://www.github.com>), un outil de versionnement de code source que nous utilisons dans l'équipe, et qui dispose d'une puissante interface de gestion des bugs.

De façon manuelle pour les « bugs » remontés par nos utilisateurs et de façon automatique pour les « crashes » grâce à l'intégration de *Github* avec *Crashlytics*.

Méthode de traitement employée

Nous avons décidé de suivre la méthode préconisée par Mitch Lacey auteur du livre « The Scrum Field Guide »².

Cette méthode opte pour un mode de traitement différent selon la sévérité du bug rencontré :

1. Lorsque le bug est d'une sévérité moyenne ou haute, celui-ci est traité en temps réel par la personne ayant écrit le code incriminé. Cette personne devra donc arrêter tout ce qu'elle faisait jusque-là et se consacrer de façon unilatérale au traitement du bug.
2. Lorsque le bug est d'une sévérité faible, celui-ci sera ajouté à la liste des tâches du prochain sprint.

----- DIAPO -----

En plus de la planification de l'ensemble des fonctionnalités à venir, des objectifs en matière de performances de l'équipe ont été définis à long terme.

Ces objectifs à long terme correspondent aux valeurs souhaitées pour l'année à venir pour les indicateurs de performance définis précédemment.

Le succès ou non de ces objectifs, déterminera les performances de l'ensemble des processus mis en place au sein de l'équipe technique.

En cas d'échec, des mesures supplémentaires devront être mises en place.

----- DIAPO -----

² [https://www.amazon.com/gp/product/0133853624/ref=as_li_tl?](https://www.amazon.com/gp/product/0133853624/ref=as_li_tl?ie=UTF8&camp=1789&creative=390957&creativeASIN=0133853624&linkCode=as2&tag=mitlacass-20&linkId=TOXA6N3B2XXL3N7G)