

Comment avons-nous construit notre app ?

Vous voulez lancer une app ? Le retour d'expérience de Sunday pourra peut-être vous donner quelques pistes à explorer.

Temps de lecture : minute

29 octobre 2019

"Pourquoi ne peut-on pas interagir sur votre App?" "Pourquoi on ne peut pas télécharger les photos ?" "Pourquoi ne peut-on pas envoyer depuis notre Samsung de 2009 ?" ... autant de questions que les utilisateurs se posent. Une seule réponse : cela coûte cher surtout quand on propose un service gratuit...

Donc on fait des choix, dans le jargon, on "itère", c'est-à-dire que l'on priorise ce qui est nécessaire versus ce qui peut attendre. Cela engendrant une certaine frustration de l'équipe car on aimerait tout faire MAIS le mieux est l'ennemi du bien, il est parfois plus raisonnable de développer au fur et à mesure car il vaut mieux 1 skate qui roule qu'une Ferrari sans moteur... Et comment fait-on ces choix ? Voici quelques tips made in Sunday.

Développement en interne ou en externe ?

Les deux mon capitaine. Chez Sunday la compétence Produit est en interne, le développement en externe au départ pour plusieurs raisons :

- Qui mieux que nous pouvons penser le produit ?
- Internaliser les compétences en interne dès le début nous aurait demandé de lever 10 millions d'euros dès le premier jour... car hardware et software, cela demande une expertise à tous les niveaux.

- Hardware is Hard, ingénieurs, développeurs confirmés, product owners, SCRUM Master... autant de professionnels qui sont nécessaires à faire tourner la machine. Donc nous, nous appuyons sur des compétences externes régionales pour la plupart pour fédérer l'éco-système.

Aujourd'hui, nous commençons à internaliser la R&D progressivement dans les équipes.

Sortez viiiiiite un MVP

À part si la compétence technique est dans vos équipes, rassemblez les fonctionnalités de base nécessaires à être lancées le plus rapidement possible. Minimum Valuable Product. Tout est dans le nom. Le produit minimum assurant la viabilité de votre idée. La fonctionnalité démontrant la nécessité d'aller plus loin.



À lire aussi

100 outils pour lancer sa startup sans développeur ni graphiste

(et presque sans argent)

Vous allez être certainement frustrés car vous ne pourrez pas sortir toutes les fonctionnalités mais c'est le jeu, cela permet de tester rapidement l'usage de votre app et son utilité.

Autant vite se rendre compte de l'expérience vécue par votre communauté avant d'aller plus loin et d'itérer, de pivoter, d'ajuster ou d'arrêter tout simplement.

Choisissez votre méthode de développement

Nous avons choisi une méthode agile fonctionnant en Sprint de 15 jours.

Première étape : Impact Mapping

Le board se réunit avec les équipes business, produit et marketing pour décider de la stratégie et de la vision du produit. Le but est de répondre aux questions suivantes pour donner du sens à ce que l'on fait !

Objectif / Pourquoi :

Quel objectif voulons-nous atteindre ? Pourquoi faisons-nous un nouveau produit ou une nouvelle phase de Développement ?

Qui :

- Quelles sont les parties prenantes ?
- Quelles sont les personnes concernées par l'objectif que nous avons défini ?
- Qui sont les clients ou utilisateurs de notre produit ?
- Qui peut nous aider à atteindre notre objectif ou au contraire nous en empêcher ?

Comment :

Nous allons définir pour chaque typologie d'utilisateurs les méthodes pour arriver à l'objectif.

Quoi :

Qu'allons-nous délivrer ? Fonctionnalité, user story, autre ?

Deuxième étape : Story Mapping

Lors de cette étape, il s'agit de raconter l'histoire de l'utilisateur produit afin de :

- Lister les user stories
- Identifier les résultats associés
- Prioriser selon les résultats et la valeur ajoutée

Le but est d'organiser le backlog produit afin d'identifier les stories les plus prioritaires.

Une fois les user stories les plus prioritaires identifiées, elles sont organisées en Sprint. Durant chaque Sprint (cela peut être des Sprints de 2 semaines par exemple) les stories sélectionnées vont être réalisées en faisant appel aux développeurs, product owners, mais également en incluant les équipes business et marketing. Une fois le sprint "Done", nous nous réunissons pour une réunion Démo/Rétro/Planning.

Cette revue prend place à chaque fin de Sprint et se divise en 3 parties :

- Démo : présentation et démonstration des fonctionnalités développées durant le Sprint écoulé
- Retro : bilan du Sprint écoulé, il s'agit de faire un point sur ce qui s'est bien passé, mal passé, les interrogations ou incompréhensions

suscitées par le Sprint.

- Planning : planification du prochain sprint

Écoutez vos early adopters

Chez Sunday, nous avons mis en place les Sunday Together, une soirée à la fin de chaque grosse session de sprints, avec la venue de nos investisseurs, de nos équipes et de nos utilisateurs. Nous y faisons la démonstration des dernières fonctionnalités, un retour à chaud des utilisateurs et nous récoltons leurs retours, remarques, idées...

Egalement, nous avons créé des groupes privés nommés Sunday Early Birds FR en France et Sunday Early Birds WWW pour le monde. Dans ces groupes, les premiers utilisateurs de Sunday sont au courant en avant-première de nos nouveautés car ils ont été les premiers à nous rejoindre et participent à l'amélioration permanente de Sunday.

Ne lâchez pas tout d'un coup

On a tendance à vouloir tout mettre d'un coup mais il faut maîtriser le niveau de maturité de son App pour toujours l'améliorer et l'adapter au besoin de ses utilisateurs. Rien ne sert de courir il faut partir à point. Chez Sunday, ce sont les pôles Communication et Business qui décident ensemble avec le Product Owner de la mise en production des fonctionnalités pour qu'elles soient en phase avec le calendrier global du produit.

La tech cela peut paraître cher !

Alors, c'est quoi une app à 1 million d'euros ? En tout cas, c'est ce que nous avons largement dépensé pour créer Sunday ! La Recherche & Développement, l'ingénierie, la sécurité, les infras, le design... cela coûte cher quand on veut le meilleur. Et vous, comment ça se passe chez vous ?

Article écrit par Anais Richardin