

Veille Technologique

Sommaire :

1^{ère} partie explication

2^{ème} partie : méthodologie

3^{ème} partie : cycle de veille

I – Qu’est ce que le développement mobile ?

- 1) Introduction (mode cible explication)
 - a. Définition
 - b. Choix de veille
 - c. Méthode push et pull
 - d. Outils pour effectuer les recherches
 - e. Problématique
- 2) Brevets, Applications développées
 - a. Site officiel
 - b. Exemples d’applications
- 3) Veille commerciale
 - a. Définition
 - b. Chiffre d’affaires / niveau commercial

II –

- 1) Hybride
 - a. Qu’est-ce que c’est ?

- b. Avantages
 - c. Inconvénients
- 2) Natif
 - a. Qu'est-ce que c'est ?
 - b. Avantages
 - c. Inconvénients
- 3) Web
 - a. Qu'est-ce que c'est ?
 - b. Avantages
 - c. Inconvénients

III –

- 1) React Native
 - a. Qu'est-ce que c'est ?
 - b. Avantages
 - c. Inconvénients
- 2) Flutter 3
 - a. Qu'est-ce que c'est ?
 - b. Avantages
 - c. Inconvénients

Conclusion

- 1) Mon avis

I – Qu'est-ce que le développement mobile ?

a) J'ai décidé de choisir la veille technologique sur le développement mobile multi-plateforme puisque l'avenir mobile est l'avenir de la technologie avec la démocratisation des smartphones. Le futur du développement mobile est brillant et stable pour l'avenir.

b) Nous passons 86 heures par mois sur nos téléphones, pratiquement chaque personne en possède un ce qui montre une possibilité d'amélioration en développant des applications. Le développement mobile multi-plateforme permet d'utiliser des applications sur différents types de plateforme, que ce soit mobile, pc, minitel...

c) Pour me concentrer sur mes recherches, j'ai utilisé deux méthodes :

La méthode push : C'est ce qui nécessite d'effectuer les recherches soi-même, de collecter des informations en se rendant sur différents sites ou ressources précises, parfois par le biais de signets et pages favorites (moteurs de recherche, les conférences, vidéos, les réseaux sociaux, revues spécialisées, articles scientifiques, etc.). Cela permet d'obtenir un résultat précis.

La méthode pull : C'est l'information qui vient à nous, il faut paramétrer un système des ensembles des outils (abonnement aux newsletters ou listes de messageries spécialisées, Alertes Google, suivi des hashtags, souscription à des fils RSS, réception des notifications, etc.) pour recevoir automatiquement des informations. Cela permet de gagner du temps mais présente des risques.

d) Pour m'aider dans mes recherches

<https://smart-origin.com/2018/01/02/codenatif2/>

https://fr.wikipedia.org/wiki/Application_hybride

https://fr.wikipedia.org/wiki/Code_natif

<https://medium.com/@vitaliypodoba/top-15-mobile-app-development-frameworks-2022-cb6bab975b61>

Images utilisées :

<https://fr.yeeply.com/blog/application-native-hybride-ou-web/>

<https://www.journaldunet.com/ebusiness/internet-mobile/1084127-part-de-marche-des-os-mobiles-en-france/>

Frameworks :

<https://aventique.paris/developpement-mobile/>

React Native :

<https://reactnative.dev/>

[Prendre documentation avec exemple d'actualités](#)

<https://www.bam.tech/article/pourquoi-utiliser-react-native>

<https://www.bam.tech/article/qu-est-ce-que-react-native>

<https://medium.com/friflex/cross-platform-and-native-development-of-mobile-apps-575467b61546>

Flutter :

<https://flutter.dev/>

https://www.frandroid.com/android/535194_quest-ce-que-flutter-loutil-permettant-de-creer-des-applications-android-et-ios

<https://medium.com/flutter/introducing-flutter-3-5eb69151622f>

e) Quel est le Framework le plus performant pour le développement natif entre React Native et Flutter ?

II –

1) Brevets, Applications développées

a)

Afin de développer mes recherches, j'ai utilisé les sites officiels des Framework de React Native et de Flutter afin d'appuyer mes recherches avec des informations officielles.

b)

Applications développées en React Native : Facebook, Instagram, Airbnb, Tesla, Discord

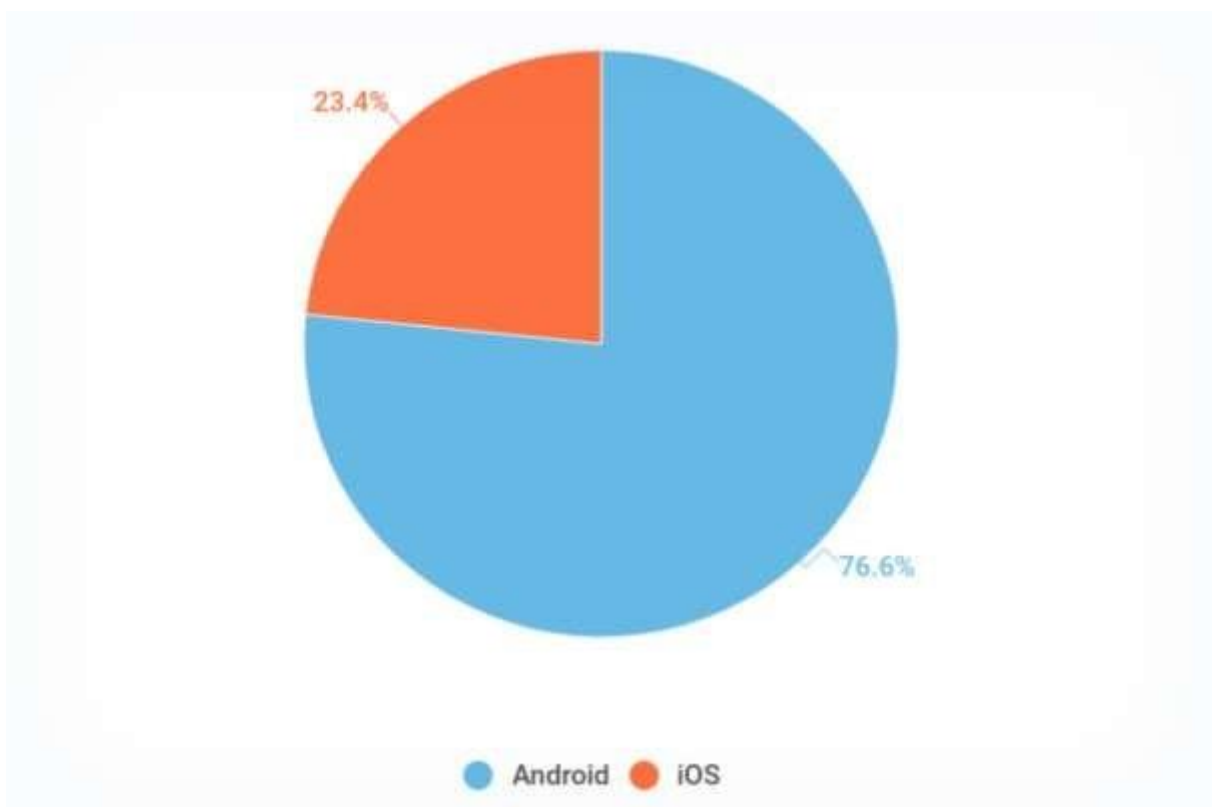


Applications développées en Flutter : BMW, Google Pay, Nubank, eBay, Toyota

3) Veille commerciale

a) La veille commerciale est d'anticiper les évolutions du marché pour se constituer un avantage concurrentiel.

b) Il existe deux systèmes d'exploitation qui dominent sur le marché : Android et IOS. Le premier est développé par Google et représente environ 85% du marché en France. Tandis que son principal concurrent, IOS qui est développé par Apple, représente un peu moins de 15% du marché. Ces deux systèmes d'exploitation s'adressent tous deux à des marchés distincts. Par exemple, les utilisateurs IOS sont en général plus aptes à dépenser de l'argent dans les applications et sont également plus exigeants. Pour cette raison, il est commun de trouver une application payante IOS qui sera gratuite sur Android. Les business Model ne sont donc pas les mêmes pour ces deux plateformes.



On peut également comparer les deux systèmes d'exploitation mobiles avec des ordinateurs Windows et Mac. Une application Mac ne fonctionnera pas sur Windows et vice-versa. Les langages d'écritures sont différents, les règles sont différentes, les possibilités sont différentes.

Aujourd'hui plus que jamais auparavant, les smartphones sont au cœur de nos vies et nous accompagnent au quotidien. Les applications que nous téléchargeons pour combler nos besoins confirment bien cette tendance. Concernant le marché des apps mobile, la domination de Google et Apple ne devrait pas s'arrêter de sitôt tant leur avance sur la concurrence est importante.

III –



1) Développement Natif

a)

Bien que le terme d'app native n'est pas très connu du grand public, cela **représente la grande majorité des applications que nous téléchargeons chaque jour**. Une application native est une application mobile qui est développée spécifiquement pour un des systèmes d'exploitation utilisé par les smartphones et tablettes (iOS, Android, etc.). Les applications natives pour

iPhones sont par exemple développées avec le langage de développement Objective-C. Parce qu'elles sont développées pour le système opérant, ils fournissent de plus grand engagement de la part des utilisateurs que les applications hybrides. Elles ont accès aux appareils des disque durs et de leurs capacités, tel que les détections de voix et les caméras qui ne sont pas disponibles via un moteur de recherche mobile.

b)

- **Meilleure rapidité**, fiabilité et dotée d'une meilleure réactivité ainsi qu'une résolution supérieure ce qui assure une meilleure expérience utilisateur.

- **Elle permet un accès plus facile** à toutes les fonctionnalités du téléphone, de l'accéléromètre en passant par la caméra et même le micro.

- **Les notifications push**, uniquement disponibles sur les apps native. Ces notifications vous permettent d'alerter vos utilisateurs et d'attirer leur attention chaque fois que vous le souhaitez, que ce soit pour du nouveau contenu ou une offre promotionnelle.

- **Ne requiert pas forcément internet** pour fonctionner, ce qui est un réel avantage. Même en 2017, il existe encore des zones très peu couvertes par le réseau internet, et permettre à ses utilisateurs d'accéder à l'app sans connexion web est un très gros point fort à ne pas négliger.

c)

- Le **coût lié au développement** d'une application native est un frein car généralement plus élevé si elle est portée sur plusieurs plateformes (afin d'être disponible pour un maximum de mobinautes) que le coût d'un site mobile ou d'une Web App. Il faudrait potentiellement prévoir un développement sur chaque technologie, et donc un coût supplémentaire si l'on souhaite se positionner sur tous les modèles.

- Elle doit respecter les règles définies par les différentes sociétés des plateformes mobiles. Que ce soit l'approbation nécessaire des Apps Store pour diffuser l'application ou ses mises à jour, les conditions tarifaires imposées ou la non-compatibilité avec les autres systèmes d'exploitation mobiles.

- Pour que l'utilisateur ait accès à la dernière version, il faut qu'il la mette à jour depuis le store contrairement aux sites mobiles et WebApp qui se mettent à jour directement.
- Il faut développer deux fois si l'on veut que l'application soit sur IOS et Android.

2) Développement web

a)

On a tous un jour fait face à une application web, mais l'on ne le savait peut-être pas. Par exemple, on a tous déjà fait face à une « version mobile » d'un site web. La **version mobile d'un site web**, c'est une application web. À l'opposé de l'app native se trouve l'application web. Pourquoi à l'opposé ? Car c'est une application qui fonctionne comme un Template de site web, sans aucune différence de plateforme, de système d'exploitation, de code... Cela veut dire que **vos utilisateurs n'auront pas à installer l'application** sur leur smartphone, le stockage reste donc intact.

b)

- Les utilisateurs n'ont pas à installer l'application sur leur smartphone, le stockage reste intact. Cela est un point fort lorsque l'on voit à quel point la mémoire de nos téléphones se remplit vite. Entre les applications, les musiques et les photos, les gigas s'accumulent vite et on se retrouve rapidement à gérer son stockage malgré nous.

- En général, le but d'une application web est de **rendre du contenu disponible**, ou du moins fonctionnel sur mobile. Elle se comporte comme une application normale, ressemble à une application normale, mais son accessibilité et son prix n'ont rien à voir avec les applications mobiles. En effet, une application web coûte jusqu'à trois fois moins cher qu'une application native.

c)

- Elle ne permet pas d'utiliser les fonctionnalités au maximum de leur potentiel, et surtout, elle **ne fonctionne généralement pas sans internet**.

- Les applications web sont également plus lentes et moins adaptées à chaque plateforme.
- Elles ne sont pas développées spécifiquement pour une plateforme, ce qui limite leur adaptabilité.
- Le fait qu'une application web soit créée en une seule version pour toutes les plateformes et systèmes d'exploitation ne lui permet pas d'avoir des performances équivalentes à tout autre type d'application.
- Le fait que l'on ne puisse pas télécharger d'application web sur les stores rend le processus de fidélisation extrêmement compliqué. Ce qui influe grandement sur le trafic de l'application et donc son nombre de téléchargements.
- L'envoi de notifications push est également impossible avec une application web.

Bien sûr, tout n'est pas négatif avec les applications web, et les possibilités restent tout de même très larges. Il faut juste savoir que les cibles et objectifs de ce genre d'application sont différents des autres apps.

3) Développement Hybride

a)

Une application hybride est une application pour mobiles qui combine des éléments HTML5 sous forme de web application mobile et des éléments d'une application native permettant d'utiliser les fonctionnalités natives des smartphones et d'être distribuée en tant qu'application sur les plateformes d'applications (App Store, Android Market, etc.).

b)

- Plus rapides et moins chères à développer qu'une application native.
- Elles offrent plus de possibilités qu'une application web.
- La maintenance de l'application est plus facile puisqu'il n'y a qu'une seule version à revoir sur plusieurs plateformes. Si l'on souhaite ajouter des nouvelles fonctionnalités ou faire des mises à jour, on n'aura besoin de le faire qu'une seule fois.

c)

- Les performances de l'application sont moins bonnes et moins stables puisque le système est moins adapté à chaque plateforme.

- Nous sommes moins libres de faire ce que l'on souhaite. L'application hybride devra rester simple.
- Communauté de développeur plus faible.

III –

1) React Native

a)

React Native est un **Framework** développé par les équipes de **Facebook** en 2015 dans le but d'accélérer le développement et la maintenance de ses **applications mobiles**. Cette application permet de développer des applications mobiles IOS et Android native.

b)

- Possède une base de code unique Javascript pour toutes les plateformes.
- Une recompilation du code plus efficace qui permet un gain de temps de développement de 30%.
- L'homogénéité des applications IOS et Android car une unique base de code permet d'avoir deux applications mobiles homogènes.
- Une communauté de développeur plus importante puisque selon Stackoverflow, près de 70% des développeurs connaissent Javascript. Ainsi qu'il est possible de maîtriser le Framework en quelques mois contrairement aux autres qui prennent plusieurs années.
- Une expérience utilisateur Native puisque React Native utilise les composants natifs et qu'il est possible d'écrire des applications mobiles aussi performantes qu'une application native sans avoir à maîtriser les langages IOS et Android.
- Une technologie pérenne grâce à sa communauté puisque depuis 4 ans, il est l'un des projets les plus populaires sur Github avec plus de 2000 contributeurs et une communauté très active.

c)

- Les défis du débogage et de la comptabilité, React Native est toujours dans la version Beta. C'est pourquoi elle présente encore quelques problèmes apparents, tels que la complexité des applications de débogage, entre autres limitations, y compris des problèmes de comptabilité. React Native peut devenir compliqué lorsque le besoin de débogage se fait sentir.
- Il y a toujours un besoin de développeurs natifs afin de mettre en œuvre certaines fonctionnalités uniques, le développeur Javascript moyen ne sera pas en mesure de faire. Si les développeurs peuvent tirer parti des bibliothèques pour effectuer de nombreuses fonctions, elles ne seront pas d'un grand secours face aux problèmes natifs. Il faudra toujours des développeurs natifs dans certains projets d'applications.
- Dépend de Facebook car il a été développé par lui. Cela constitue une lacune puisque si Facebook cesse de fournir une sauvegarde de la plateforme, tout s'effondrera.
- La gestion de la mémoire n'est pas exceptionnelle car si nous avons besoin d'une application pour créer des calculs massifs, il faut choisir une autre plateforme de développement.
- Problème concernant les licences et les brevets car en 2017, Facebook a mis à jour ses politiques de licences concernant les projets open source. Cela signifie qu'un utilisateur peut être bloqué sur Facebook si un projet crée avec l'open source de projets à un problème avec les licences.
- Adoption lente des dernières fonctions, elles arrivent en retard malgré l'innovation dans la technologie mobile, une assistance native est nécessaire pour utiliser ces fonctionnalités.
- Défis en matière de sécurité avec Javascript car il est bien connu pour son retard dans ce problème.

2) Flutter 3

a)

Flutter est un framework développé par Google, le plus récent de tous. De ce fait, les ingénieurs ont pu observer les points forts et les faiblesses de chaque outil existant. Cela permet de créer de belles applications multiplateformes compilées nativement à partir d'une seule base de code. Il utilise un langage de programmation créé par Google : le Dart, c'est un langage de programmation orientée-objet et un langage avec une grande productivité et un langage portable pour le développement multiplateforme.

b)

- Réduction des coûts puisqu'au lieu d'effectuer deux développements natifs en parallèle, développer une application mobile en Flutter permet de réduire le temps de développement.
- Les performances car Google a choisi de s'éloigner des paradigmes habituels en ayant une stratégie proche des technologies du jeu vidéo. Comme avec React Native, qui lui utilise le langage Javascript. Pour Flutter nous retrouvons des widgets, il utilise un moteur graphique, comme dans un jeu vidéo, qui va créer un « Canvas » et va « dessiner » ses widgets sur cette toile. Seule cette toile sera traduite en langage natif, et non pas les widgets qui la compose. Cela permet d'assurer un comportement et des performances similaires sur Android et iOS. Le moteur de rendu de Flutter est développé en C++, qui est un langage offrant de meilleures performances que celui de React Native développé en Javascript.
- Sponsorisé pas seulement par Google mais aussi par toute la communauté, c'est-à-dire que les fichiers open source permet à beaucoup de personnes de participer.

c)

- Les performances puisqu'aujourd'hui Flutter est le meilleur framework de développement d'applications multiplateformes en termes de performances. Cependant, ses performances ne dépassent pas celles d'une application développée directement en langage natif. C'est un stade technique qui sera, de toute façon, quasiment impossible d'atteindre. Flutter est limité niveau animation.
- Flutter est jeune, la première version alpha sortie date de 2017. Pour un framework de développement, cette information est capitale puisqu'un outil jeune veut dire moins de

développeurs capables de l'utiliser, ce qui implique une communauté faible. La communauté autour d'un langage ou un framework est un élément essentiel lors de la phase de développement. En cas de problème, les chances de trouver une solution sont réduites.

Conclusion :

a)

Ce que je pense c'est que le développement mobile multiplateforme sera de plus en plus intense les années à venir puisque la majorité des personnes dans le monde possède un téléphone mobile et qu'elles utilisent des applications mobiles à chaque instant. Concernant le développement de ces applications, le développement natif serait vraiment utile si l'on souhaite faire des applications complexes avec beaucoup de fonctionnalité mais plus au long au développement. Tandis que le développement en hybride permettrait de développer plus rapidement mais en ayant moins de fonctionnalités pour l'application, cela pourrait se résoudre dans les prochaines années à venir.