

LE GLOSSAIRE

DE VOTRE RÉUSSITE MOBILE

par XInflexsys



POUR QUOI FAIRE ?

Une application mobile, tout le monde sait de quoi il s'agit. Nous en téléchargeons régulièrement et en utilisons (une poignée) quotidiennement.

Pourtant, lorsqu'on se lance dans un projet de conception d'applications mobiles, on découvre un monde nouveau peuplé de mots dont on ne connaît pas toujours le sens.

Étant donné le budget que peuvent représenter de tels projets, il est préférable de comprendre de quoi on parle.

Ce glossaire a donc été élaboré à l'attention des personnes qui souhaitent décrypter le « jargon » de la mobilité et mettre ainsi toutes les chances de leur côté pour faire **de leur projet mobile une réussite exemplaire**



UN
GLOSSAIRE !



AGILE

La méthode Agile est une méthode de gestion et de développement de projets informatiques.

Il en existe de nombreuses déclinaisons, comme Scrum et Extreme programming.

Toutes ces déclinaisons de la méthode Agile ont pour particularités d'impliquer très tôt les utilisateurs et de permettre une plus grande réactivité que les méthodes de développement traditionnelles.

C'est la raison pour laquelle la majorité des équipes de développements mobiles les utilisent aujourd'hui. Elles s'appuient sur un cycle de développement itératif et respectent 4 valeurs fondamentales déclinées dans le Manifeste Agile.

Concrètement, développer Agile, c'est co-crée avec les utilisateurs.

Ces méthodes ont l'avantage de permettre aux utilisateurs de clarifier leurs exigences au fur et à mesure de l'avancée du projet, et donc de favoriser une plus grande interaction entre des bêta-testeurs et les développeurs.

- + La liste de toutes les fonctionnalités à développer est appelée « **backlog** ».
Le backlog est découpé en « **sprints** » d'une durée de 2 à 3 semaines.

ANALYTICS

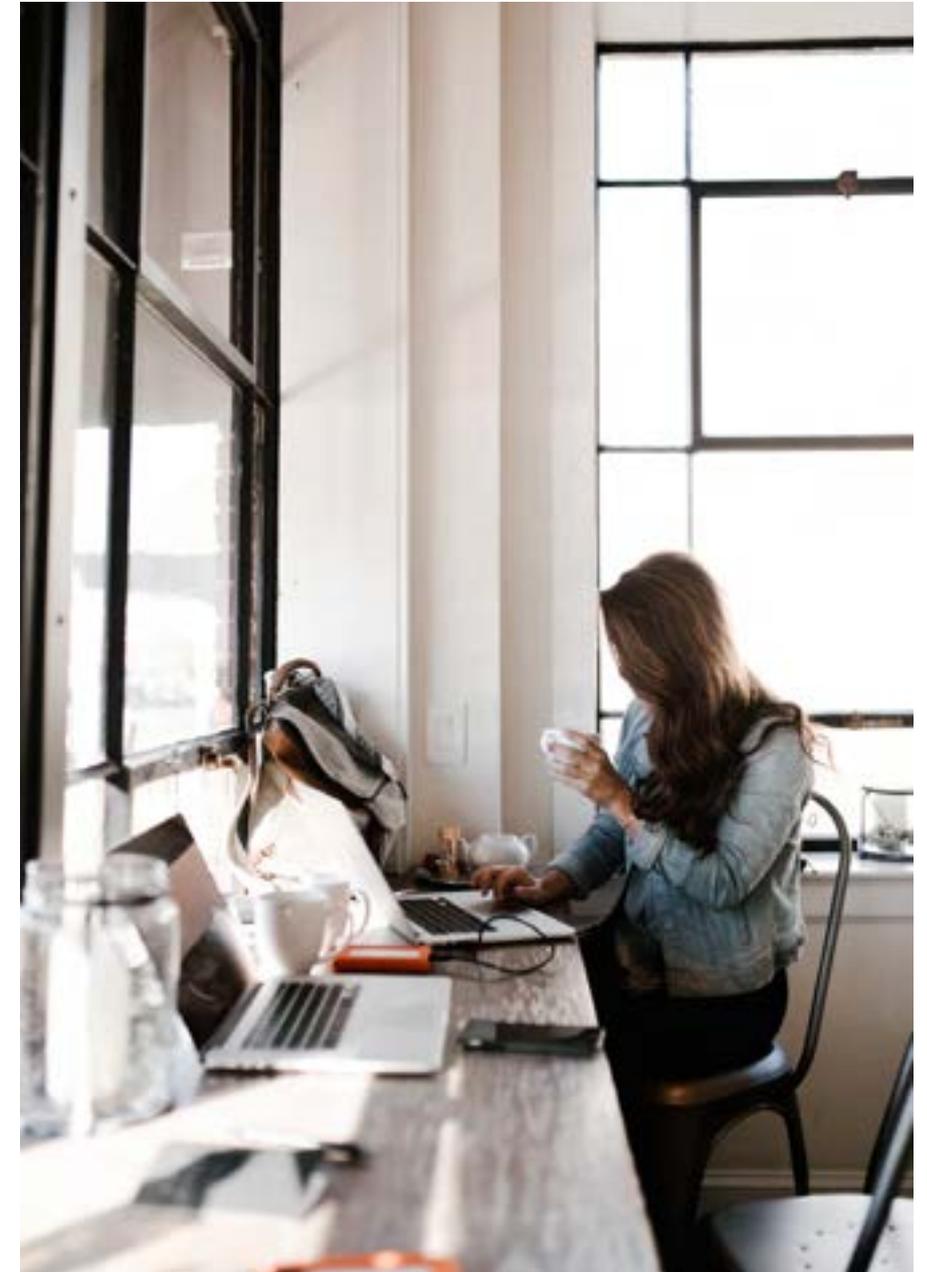
Les solutions Analytics sont là pour vous permettre de mesurer la performance de votre application mobile.

Un projet de conception d'application mobile ne s'arrête pas lorsqu'elle est mise en production !

À ce stade, il est capital de comprendre comment elle est utilisée et d'apporter si besoin, les ajustements nécessaires, afin de conserver intact dans le temps **l'engagement des utilisateurs**.

Les solutions Analytics permettent de garder un œil sur la performance de votre application mobile, tout au long de son **cycle de vie** (Phases d'acquisition, d'engagement, de rétention), au travers d'indicateurs clés de performance (Key Performance Indicators ou KPIs) telles que :

- Nombre de personnes l'ayant téléchargée
- Profil de son audience (démographiques, techniques...)
- Nombre d'utilisateurs actifs
- Fréquence d'utilisation
- Comportement dans l'application (chemins suivis, pages vues, actions menées...)
- Temps passé dans l'application (durée de session)
- Taux d'abandon





API

Une API (Acronyme de « Application Programming Interface ») est une Interface Applicative de Programmation.

Il s'agit d'un ensemble d'éléments permettant à un logiciel d'offrir des services à d'autres logiciels. Concrètement, l'API propose une bibliothèque (on parle de librairie) de briques technologiques prêtes à être utilisées et dont le programmeur va se servir pour permettre à son application d'interagir avec d'autres logiciels. Les API sont documentées afin d'expliquer comment les utiliser, mais leur intérêt réside dans le fait que le développeur n'a pas besoin de comprendre leur logique interne.

Les développeurs d'applications mobiles s'appuient couramment sur des API pour permettre :

- La connexion et la synchronisation des données avec un serveur, en mode sécurisé
- Le **développement en mode natif** (iOS, Android, Windows), **hybride** (HTML5, CSS Phonegap, Sencha Ext JS) ou Javascript sur client Mobile (WebApp)

AUTHENTIFICATION

L'authentification désigne la procédure qui permet de vérifier l'identité d'une personne ou d'une entité.

En matière d'application mobile, les technologies d'authentification ont pour objectif de vérifier l'identité d'une personne qui souhaite se connecter à un service ou effectuer une transaction.

Au cours des dernières années, ces technologies ont considérablement évolué : elles sont devenues plus difficiles à compromettre que les méthodes traditionnelles s'appuyant sur des mots de passe, tout en simplifiant l'expérience utilisateurs des mobinautes.

Elles sont désormais moins contraignantes, intégrées à leur équipement (tablette, smartphone, smartwatches...) et/ou à leurs applis.

⊕ **L'authentification forte (appelée aussi « authentification à deux facteurs ») ou multiple (multifacteurs)** désigne une procédure qui combine plusieurs méthodes d'authentification pour vérifier l'identité de l'utilisateur : par exemple, mot de passe et empreinte digitale. Cette double (ou multiple) vérification, particulièrement appréciée dans les secteurs critiques tels que la banque, complique la tâche d'un hacker qui, même s'il parvient à franchir le premier obstacle (le mot de passe), ne sera pas arrivé au bout de ses peines.

↘ [Enrôlement d'un terminal mobile](#)





BACKLOG

Le backlog désigne la liste de toutes les fonctionnalités à développer dans le cadre d'un projet mobile (terminologie **Agile**).

Dans le cadre de projets mobiles, on peut difficilement s'appuyer sur les méthodes de développement classiques car les tests interviennent trop tard dans le processus.

La majorité des équipes de développements mobiles utilisent donc des méthodes Agile s'appuyant sur **un principe de développement itératif** : le projet donne lieu à un « **backlog** » comprenant toutes les « user stories » de l'application à développer (**une « user story » étant une description simple d'une fonctionnalité de l'application**).

Chaque « user story » est affectée d'une priorité définie par l'équipe Métier, et étudiée par l'équipe Projet qui identifie l'ensemble des tâches de développement et de validation nécessaires à sa mise en œuvre.

Lorsque ce backlog est finalisé, il est validé par les deux parties (Prestataire et Client) et un planning de production est établi.

BEACON

Les beacons sont de petits capteurs Bluetooth qui repèrent la présence de smartphones à proximité (dans un rayon de quelques mètres) et permettent de déclencher une action telle que l'envoi d'une **notification**.

Différents cas d'usage peuvent être envisagés grâce aux beacons, notamment :

- Application Mobile-to-Store (par exemple, une promotion appropriée est adressée au client dès qu'il franchit la porte du magasin ; ou bien un jeu sur mobile avec des lots à gagner proposé uniquement en étant physiquement présent dans la boutique)
- Application de **géolocalisation indoor** et de **geofencing** dans le domaine du Smart Building (gestion des salles de réunion ou réservation d'une salle à la volée ; orientation dans un bâtiment ; guidage pour rejoindre une personne ou retrouver son véhicule dans un parking...)





CAHIER DES CHARGES

Le cahier des charges est le document dans lequel vous détaillez ce que vous attendez de votre application mobile et sur lequel le prestataire va se baser pour élaborer sa proposition commerciale.

Concevoir une application mobile nécessite de rédiger un cahier des charges détaillé car cela vous permettra de fixer un cadre budgétaire, puis de comparer les offres entre elles.

L'objectif est d'avoir une idée précise du niveau de complexité de l'application à développer. Gardez en tête que plus l'appli est basique, moins elle est coûteuse et plus elle est rapide à développer. Donc, tout en élaborant le cahier des charges, **privilégiez la simplicité et l'utilité des fonctionnalités** pour les usagers. Ne proposez une fonction que si vous êtes complètement sûr qu'elle représente un véritable intérêt.

Vous pouvez également prévoir une première version proposant les fonctionnalités incontournables et planifier des évolutions dans un second temps.



Téléchargez la Check-list [« Concevoir le Cahier des Charges de votre application mobile »](#)

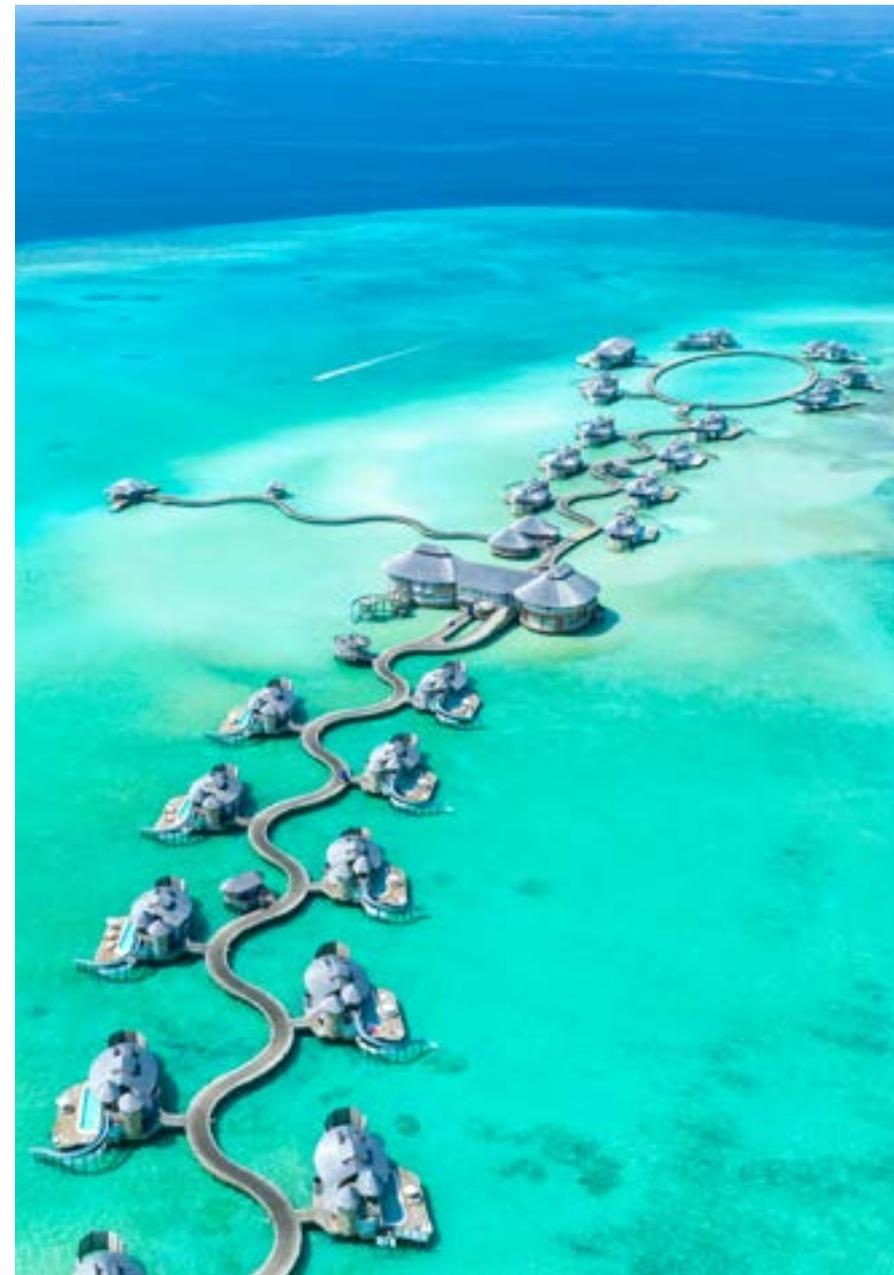
CONNECTEUR

Les connecteurs permettent à votre application mobile d'accéder aux données ou aux sources d'information de l'entreprise.

On peut trouver des **connecteurs génériques** vers des applications standards (CRM, ERP, GED, Bases de données...) ou des **connecteurs spécialisés** (vers des applications Métier ou des Web Services, les réseaux sociaux, des tiers de confiance pour des applications où la signature électronique entre en jeu par exemple, des solutions de **localisation indoor/outdoor**, etc.)

Ces connecteurs permettent de **diminuer les temps et coûts de développement**, et de dissocier l'application mobile du système de back-office auquel elle accède ; ainsi, dans le cas d'un changement de solution de back-office (CRM par exemple), l'application mobile n'est pas impactée : seul le connecteur doit être remplacé.

- **Le CRM** (acronyme de Customer Relationship Management) est un logiciel permettant de gérer les données relatives aux clients et aux prospects.
- **L'ERP** (acronyme de Enterprise Resource Planning), également appelé PGI (Progiciel de Gestion Intégrée), est un logiciel qui permet de gérer l'ensemble des activités de l'entreprise : gestion comptable, gestion commerciale, gestion des stocks...
- **La GED** (acronyme de Gestion Électronique de Documents) est un système permettant la numérisation (donc l'archivage) et l'indexation de documents.





CYCLE DE VIE

Le cycle de vie d'une application mobile couvre son développement, sa mise en production et ses mises à jour successives.

Comme tous les produits mis sur le marché, une application mobile a un cycle de vie... et parfois cette vie peut être très courte ! Selon Google, **26% des applis mobiles sont abandonnées après leur première utilisation.**

Un projet mobile ne se conclut donc pas, comme on pourrait le croire, par la mise en production de l'application. Car si l'application ne rencontre pas son public, autant dire que l'investissement consenti pour la développer est perdu.

Pour que la vie de l'application mobile soit longue, il est important de **garantir sa bonne adoption par les utilisateurs** (on parle alors d'engagement) et de **les fidéliser** (en leur proposant des mises à jour répondant à leurs demandes d'amélioration).

- + Conserver intact **l'engagement des utilisateurs** après la mise en production de l'application mobile, est l'objectif principal que vous devez poursuivre. Pour cela, gardez en tête de **rester proche des utilisateurs** (qu'ils soient internes ou externes) en mesurant régulièrement leur satisfaction ou leur engagement au travers des KPIs (Key Performance Indicators ou Indicateurs Clés de Performance) que vous aurez définis (Cf **Analytics**).

DESIGN THINKING

Véritable outil au service de l'innovation, le Design Thinking est une méthodologie qui permet de transformer les idées et les projets en actions réelles et en prototypes tangibles.

Le Design Thinking s'inspire de la méthode du « brainstorming » et la complète en allant plus loin pour lancer des projets concrets. Cette approche, développée à Stanford dans les années 80, a pour but d'appliquer la démarche d'un designer pour répondre à un problème ou à un projet d'innovation, lors d'**ateliers de co-conception**. Cette manière de réfléchir et d'innover s'appuie principalement sur des retours d'utilisateurs.

La démarche se découpe en 5 étapes : Immersion, Idéation, Décision, Prototype, Test.

Elle a pour objectif de :

- **débloquer l'innovation** d'une organisation en faisant travailler ensemble des collaborateurs exerçant des métiers très différents. On parle de co-conception et d'intelligence collective.
- **donner vie rapidement à des projets répondant à des besoins**, et ainsi pouvoir tester et lancer des produits et services en un temps record. On se retrouve donc avec des idées résolvant de vrais problèmes que l'on conceptualise ensuite en prototypes en 1 ou 2 journées de travail maximum.
- **développer la cohésion** en entreprise en faisant travailler les collaborateurs ensemble.





DÉVELOPPEMENT HYBRIDE

Les applications hybrides sont des appli développées en HTML5 (comme les **WebApp**), puis générées en tant qu'**appli native**.

Il existe en effet 3 types d'applications mobiles : celles développées en mode **natif (c'est-à-dire dans le langage de programmation propre à l'OS sur lequel elles vont fonctionner)**, celles développées en mode hybride et les **WebApp (un site internet conçu pour être consulté depuis un navigateur)**.

Les appli hybrides ont donc pour ambition de combiner les avantages des WebApp avec ceux des appli natives.

AVANTAGES

- **Mutualisation de l'effort de développement.**
Les applis fonctionnent sur Android et iOS en utilisant le même code
- **Homogénéité des applis**
- Optimisation du coût de **maintenance** évolutive
- Accès aux fonctionnalités intrinsèques du terminal mobile. **Notification, géolocalisation...**
- **Disponibilité des compétences de développement Web.**
Plus largement répandues que les compétences iOS et Android

INCONVÉNIENTS

- **Techno unique pour tous les OS.** Cela peut gêner les utilisateurs adeptes de l'un ou l'autre OS en terme d'ergonomie
- **Moindre réactivité qu'une appli native.** Le code HTML doit être réinterprété par le système
- **Process de publication nécessaire pour diffuser l'appli sur les stores.** Cela constitue un inconvénient pour les développeurs mais un avantage pour les utilisateurs notamment lors des mises à jour

DÉVELOPPEMENT NATIF

Une application native est une application développée en utilisant les outils de développement spécifiques à la plateforme (Android, iOS, Windows) choisie.

AVANTAGES

- **Performance optimale.**
Pas de temps de latence puisque le langage est compatible avec le système et n'a pas besoin d'être réinterprété
- Permet une **utilisation offline de l'appli**, c'est-à-dire sans couverture réseau ni wifi
- Accès aux fonctionnalités intrinsèques du terminal mobile. **Notification, géolocalisation...**
- Utilisation compatible avec des **objets connectés** développés pour la même plateforme
- Fluidité de navigation

INCONVÉNIENTS

- **Un développement est nécessaire** pour chaque plateforme (**Android, iOS, Windows**)
Cela augmente les coûts et temps de développement
- **Rareté** des compétences en développement
- **Process de publication** nécessaire pour diffuser l'appli sur les **stores**. Cela constitue un inconvénient pour les développeurs mais un avantage pour les utilisateurs notamment lors des mises à jour
- Évolution et **maintenance** coûteuses

↗ Voir aussi [Développement hybride](#) et [WebApp](#)





ENGAGEMENT UTILISATEUR

L'engagement d'un utilisateur en faveur de votre appli mobile désigne son niveau d'intérêt (téléchargement, ouverture, fréquence d'usage...)

Les raisons sont nombreuses qui font que l'on se désintéresse des applis mobiles installées sur notre smartphone : lenteur, faiblesses ergonomiques, caractère intrusif des notifications, manque d'intérêt...
Un récent Baromètre des Usages Mobiles publié par EBG est riche d'enseignements à ce propos.

Alors, comment être sûr de conserver intact l'engagement des utilisateurs ?
Tout simplement en intégrant les utilisateurs au projet, avant et pendant le développement, puis durant tout le cycle de vie de l'application :

- **Avant même que le cahier des charges ne soit établi**, en mettant en place des ateliers de cocréation ou de **Design Thinking**
- **Pendant le développement**, en s'appuyant sur la **méthode Agile** dont l'avantage est de permettre aux utilisateurs de clarifier leurs exigences au fur et à mesure de l'avancée du projet
- **En accompagnant les utilisateurs lors du déploiement** et tout au long du **cycle de vie** de l'application pour garantir une bonne appropriation et tenir compte de leurs remontées pour faire évoluer l'application
- **En analysant la façon dont l'application est utilisée** au travers d'indicateurs clés (**Analytics**) pour mettre en œuvre les évolutions pertinentes

ENRÔLEMENT D'UN TERMINAL MOBILE

L'enrôlement permet de valider l'inscription d'un terminal mobile dans la liste des utilisateurs approuvés d'une application.

Il s'agit d'un cas particulier d'**authentification** forte.

Lorsqu'un utilisateur inscrit son terminal (lors de la procédure d'enrôlement), il est authentifié et les contenus/applications sont mis à sa disposition selon les restrictions associées à son profil.

L'enrôlement s'effectue une seule fois et se déroule en plusieurs étapes. La fiabilité de cette approche réside dans le fait **qu'il n'est plus besoin, une fois l'enrôlement réalisé, de faire transiter des mots de passe sur Internet.**





FLAT DESIGN

Le flat design est un style graphique qui privilégie la simplicité et la lisibilité et qui se caractérise par des formes simples et des aplats de couleur vive.

Le flat design est très utilisé dans les applications mobiles car il permet **une plus grande ergonomie tout en optimisant le temps de téléchargement** des pages.

Il peut, cependant, être difficile à adapter à certaines identités visuelles ou être perçu comme trop simpliste ou « flashy ».

↗ Voir aussi [UI Design](#), [UX Design](#)

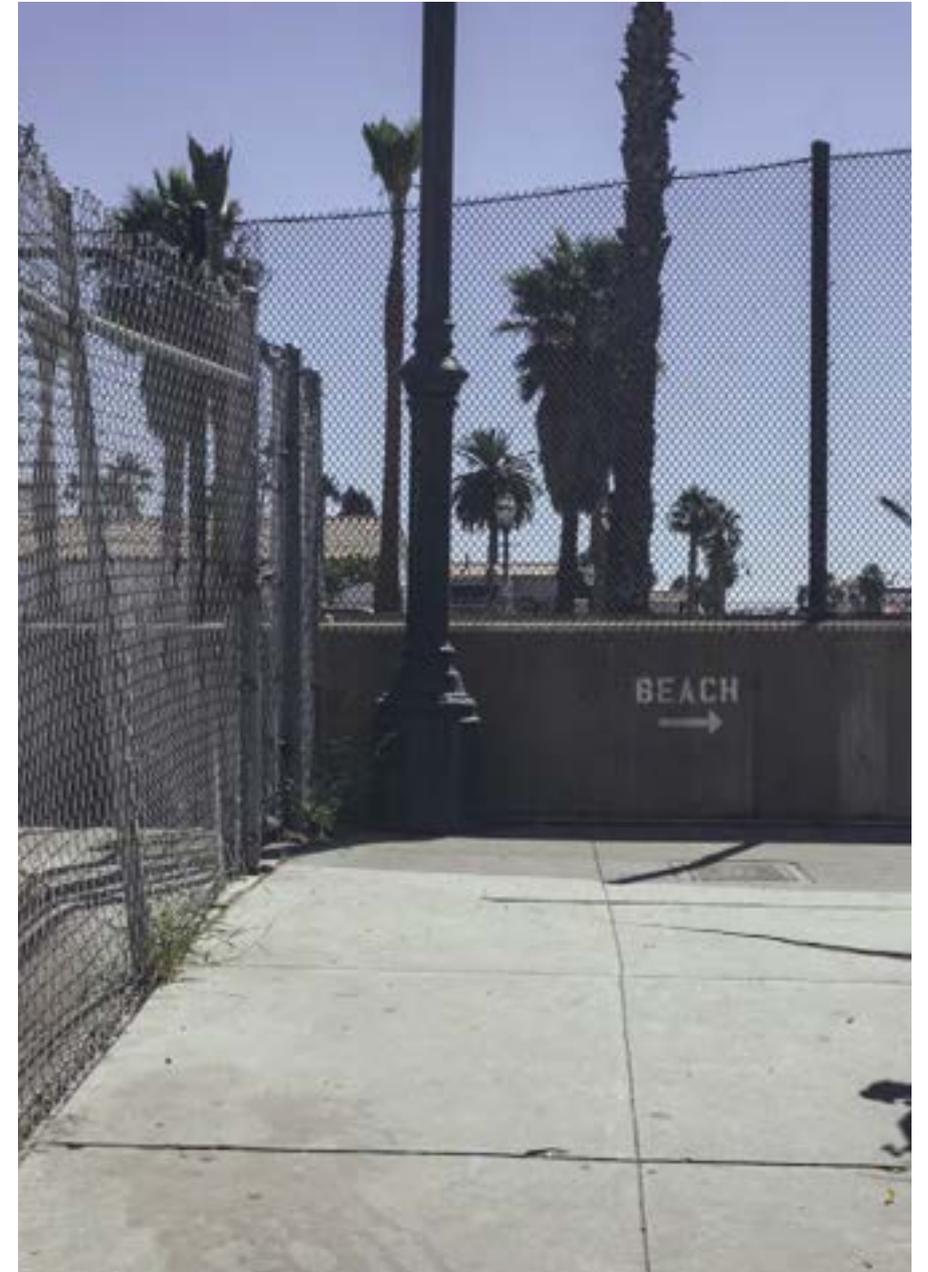
GEOFENCING

Le Geofencing (ou géo-repérage) est un dispositif permettant de définir une zone entourée d'une clôture virtuelle, afin de pouvoir déclencher une alerte lorsqu'un appareil ou une personne entre ou sort de cette zone.

Le Geofencing permet par exemple:

- d'envoyer une **push notification**, un SMS ou un email, lorsqu'une personne pénètre dans une boutique (usage marketing) ;
- d'envoyer une alerte au personnel soignant lorsqu'une personne ayant des difficultés d'orientation sort d'un périmètre défini (usage en EHPAD) ;
- de réserver une salle de réunion à la volée en entrant simplement dans la pièce (usage Smart building).

 Le Geofencing s'appuie sur une fonction intrinsèque des terminaux mobiles, la **géolocalisation**. Selon la précision dont on a besoin pour géolocaliser la personne, différentes méthodes sont utilisées dont le GPS du smartphone ou des **beacons**.





GÉOLOCALISATION (INDOOR/OUTDOOR)

La géolocalisation est une fonctionnalité intrinsèque des terminaux mobiles dont la finalité est de localiser un objet ou une personne grâce à ses coordonnées géographiques.

Cette fonctionnalité s'appuie sur différentes technologies (Satellite avec le GPS, Wi-Fi, etc.) selon la précision dont on a besoin.

Si cette fonctionnalité apparaît évidente (« il suffit d'activer un GPS ! »), elle peut constituer cependant un réel défi technique selon la précision de géolocalisation dont on a besoin ou selon qu'il s'agit de géolocalisation indoor ou outdoor.

On parle de géolocalisation indoor pour des usages à l'intérieur de bâtiments (Applications de Smart Building, de mobile-to-store, de suivi médical...) et de géolocalisation outdoor en extérieur (Applications de guidage, gestion de flottes de véhicule, etc.)

ⓘ **Si le GPS suffit généralement pour de la géolocalisation outdoor, il peut s'avérer décevant pour de la géolocalisation indoor.** Pourtant, vous l'avez constaté, votre téléphone arrive à vous localiser même à l'intérieur de bâtiments. En réalité, dans ce cas, cette localisation n'est pas effectuée grâce à la technologie GPS, mais grâce à un autre service fourni par Apple ou Google qui **scanne les réseaux WIFI et cellulaires** qui se trouvent à proximité et les comparent à une grande base de données pour déterminer votre localisation.

↗ Voir aussi : [geofencing](#), [beacon](#)

MAINTENANCE

Dès que votre application mobile est déployée, la garantie démarre. Cette garantie peut, selon les cas, couvrir les maintenances corrective, préventive et évolutive.

- La **maintenance corrective** comprend la correction d'éventuels bugs survenant pendant la période de garantie : cette durée est prédéfinie dans le contrat qui vous lie avec votre prestataire.
- La **maintenance préventive** permet de garantir la stabilité de l'application en cas d'évolution d'un des éléments sur lesquels elle s'appuie (par exemple, une API).
- Enfin, la **maintenance évolutive** concerne le développement de nouvelles fonctionnalités (nécessitant une ou plusieurs « user stories » recensées dans un nouveau **Backlog**).

⊕ Lorsque la maintenance d'une application est déléguée à un prestataire externe, on parle de **Tierce Maintenance Applicative (TMA)**. Cette prestation est détaillée dans un document contractuel précisant le périmètre de ce qui est couvert (maintenance corrective, préventive et/ou évolutive), les délais d'intervention, etc.





MENU BURGER

Le menu communément appelé « menu burger » d'une application mobile est représenté par 3 traits horizontaux, placés l'un au-dessous de l'autre.



Il ne s'agit que d'un surnom (peut-être lié au régime alimentaire des développeurs !) car en réalité, son vrai nom en langage Windows est « **Split view** » : en effet, en cliquant dessus, **l'écran en cours se scinde en 2 parties** dont l'une présente les différentes rubriques de l'application.

L'avantage du menu burger est qu'il prend peu de place à l'écran... tellement peu qu'on peut carrément l'oublier.

Une alternative au menu burger est le **Tap bar**. Il se présente, comme son nom l'indique, comme une barre, souvent située en bas d'écran (pour être plus facile à atteindre avec les pouces), comprenant des icônes incitant à l'interaction. Ce type de menu est idéal pour mettre en avant les fonctions principales de l'application.

De nombreuses applications mobiles proposent les deux types de menus, afin de satisfaire les préférences du plus grand nombre de mobinautes, mais aussi (et surtout) pour des raisons d'ergonomie.

MOCKUP

Un mockup (maquette en anglais) est un prototype plus ou moins abouti, permettant aux futurs utilisateurs de visualiser l'application mobile, avant même que les développeurs n'aient commencé à travailler.

Il permet donc de **tester l'expérience utilisateur (UX/User eXperience)** et de vous assurer ainsi que l'appli répond bien aux attentes exprimées.

Dans un premier mockup (qui peut être fait à la main sur papier dans un premier temps), on matérialise l'**ergonomie logicielle**, c'est-à-dire la cinématique (navigation, enchaînement des écrans), les boutons, etc.

Après les adaptations nécessaires, on aboutit à un mockup final qui peut être réalisé dans un logiciel spécifique afin de tester directement votre prototype sur un smartphone ou une tablette. Ce mockup comprend des éléments cliquables permettant de **simuler le fonctionnement de l'application**.

Lorsque le design de l'application est validé, il est temps de mettre le projet entre les mains des développeurs pour qu'ils codent l'application.

↗ Voir aussi : [UX Design](#), [UI Design](#), [Storyboard](#), [Wireframe](#)





MVP (MINIMUM VIABLE PRODUCT)

Bien loin des parquets de la NBA, dans le domaine des applications mobiles, un MVP est une application mobile pour laquelle on a fait le choix de « sacrifier » des fonctionnalités jugées comme non prioritaires afin d'obtenir un lancement plus rapide et moins coûteux.



Les avantages d'un MVP sont multiples :

- se focaliser sur les fonctions essentielles, qui ont été préalablement identifiées au cours d'un atelier de cadrage
- tester le marché et les utilisateurs
- se familiariser avec les contraintes techniques et le contexte d'utilisation
- présenter un prototype de la future solution à des investisseurs potentiels

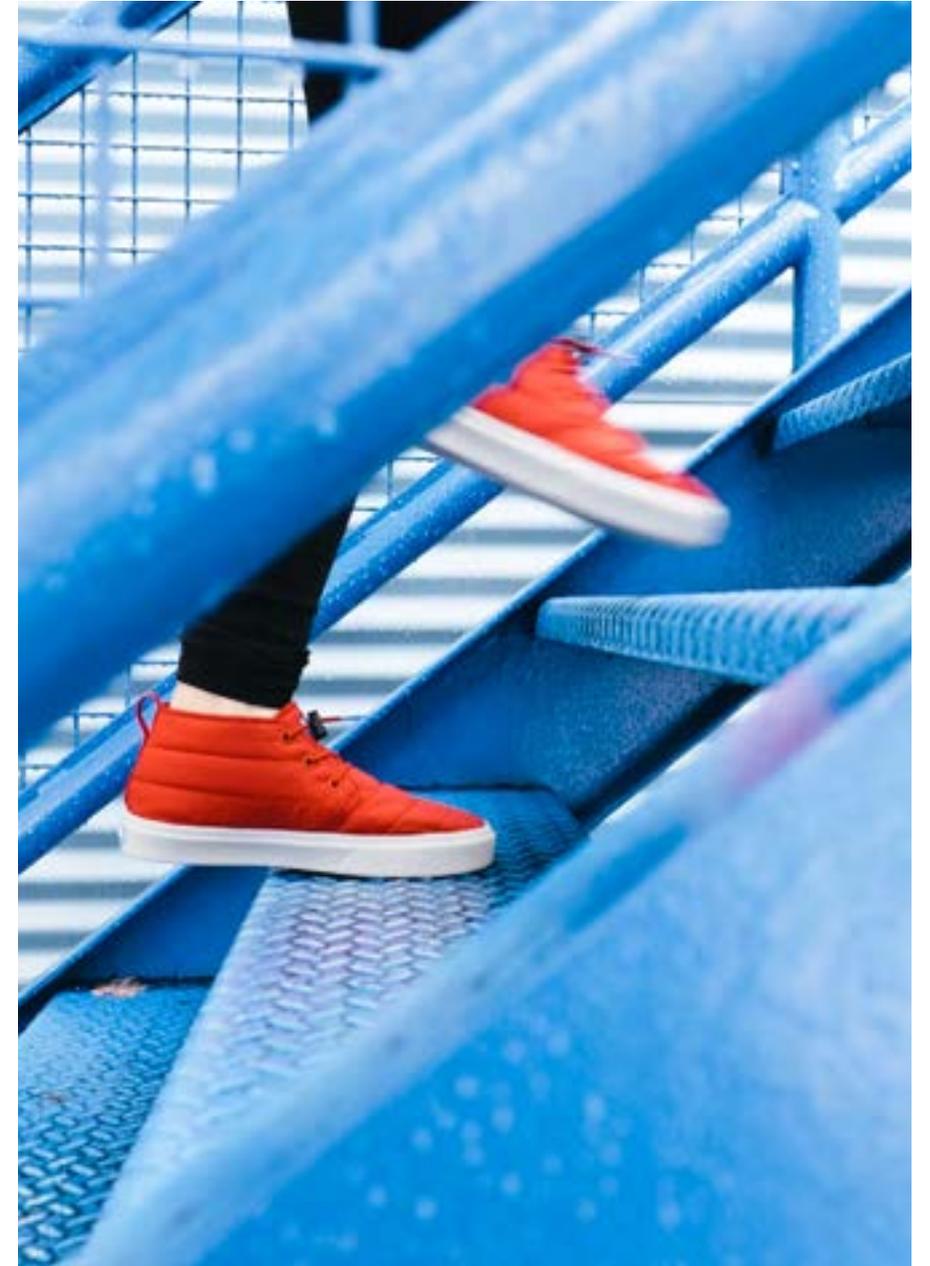
PROGRESSIVE WEBAPP

Une Progressive WebApp (PWA) est en fait un site internet conçu pour un équipement mobile, mais qui propose certaines des fonctionnalités intrinsèques d'une appli native ou hybride.

La PWA offre par ailleurs une expérience utilisateur plus proche de celle offerte par une **appli native** que de celle offerte par une **WebApp**.

Comme la WebApp, elle est accessible depuis un browser (et non depuis un **store**), **et peut être accessible directement par un raccourci comme une application native**. Elle peut donc être lancée ainsi hors du navigateur, sans la barre d'adresse propre au navigateur, donc comme une application native.

Sans remplacer les **applis natives/hybrides** et même les WebApp, les PWA peuvent constituer **une étape intermédiaire dans la construction de votre stratégie mobile**.





PUSH NOTIFICATION

La notification push est un message (généralement court) qui est envoyé depuis un serveur vers le terminal mobile (smartphone, tablette) des utilisateurs de l'application, qu'ils soient en train de l'utiliser ou pas.

Elle a pour objectif de « **déclencher une session utilisateur ou d'apporter une information instantanée*** ».

Les notifications mobiles créent **l'engagement** même si elles sont **(parfois) perçues comme intrusives** par beaucoup d'utilisateurs. Selon une étude menée en 2015 par Localytics, les utilisateurs qui ont autorisé les notifications ouvrent une application en moyenne 14,7 fois par mois, contre 5,4 fois pour une application sans notification push.

*Source : « Les push notifications, mode d'emploi 2015 », Mobile Marketing Association France



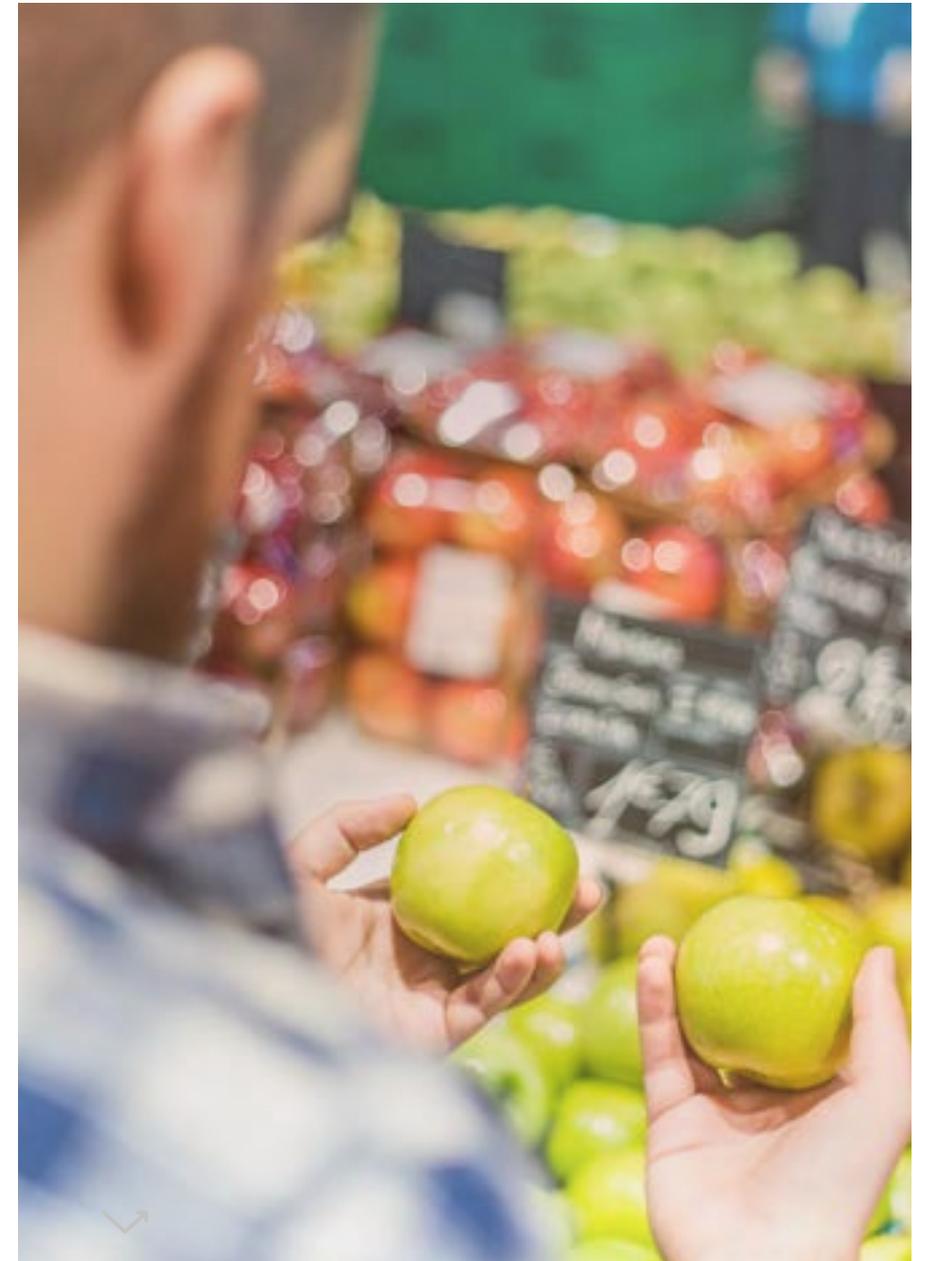
Pour qu'elles soient perçues comme utiles, il est important de montrer à l'utilisateur quel bénéfice il peut retirer des notifications qu'il se verra adresser. C'est le cas, par exemple, lorsqu'elles sont **personnalisées ou contextualisées** (déclenchement de l'alerte push suite à un événement précis ou à un moment précis de l'usage de l'appli, déclenchement d'un push **géolocalisé** quand le mobinaute entre dans une zone géographique précise, etc.)

QUALITÉ D'UNE APPLI MOBILE

Le rôle d'un service Qualité en matière de développement d'application mobile est de vérifier que les développements réalisés sont conformes aux périmètres spécifiés.

En dehors des aspects fonctionnels de l'application, les tests d'applications mobiles sont tout à fait spécifiques aux équipements sur lesquels ils fonctionnent et concernent :

- l'installation,
- la performance (une appli trop lente sera abandonnée ou désinstallée),
- la connexion (fonctionnement hors connexion par exemple),
- la montée en charge du serveur (on « stresse » le serveur pour voir comment il réagit) ou du réseau,
- l'ergonomie (cohésion de l'interface utilisateur, utilisabilité),
- la mémoire,
- la sécurité (par exemple, une application bancaire ne devra pas être hackable),
- le bon fonctionnement sur les différentes tailles de matériels (smartphone, tablette, phablette...),
- etc.





RECETTE

Contractuellement importante, la recette est l'étape d'acceptation de l'application par le client ; elle clôture le contrat de prestation.

La recette est l'une des dernières étapes d'un projet de développement d'application mobile, et intervient juste avant sa mise en production. Il s'agit donc d'une **sorte de répétition générale**.

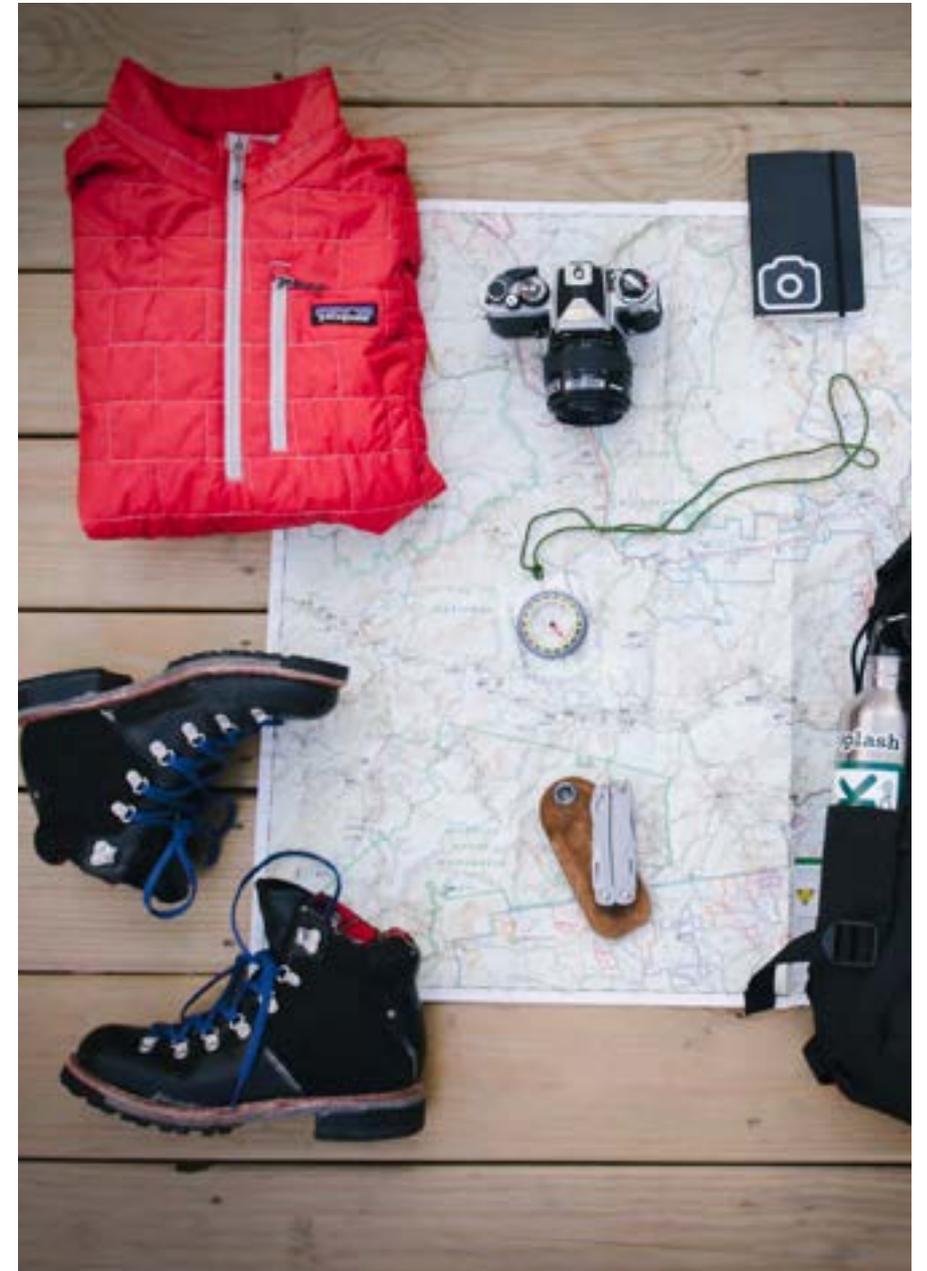
La recette s'appuie sur un cahier de recettes, dans lequel sont référencés tous les tests à mettre en place pour s'assurer du bon fonctionnement de l'application. L'application est testée dans un environnement **comparable à l'environnement de production**.

La recette donne lieu à un compte-rendu dans lequel les éventuels derniers bugs ou dysfonctionnements sont indiqués, accompagnés de leur degré de criticité.

SOFTWARE DEVELOPMENT KIT (SDK)

Un SDK est un ensemble de briques de code préprogrammées (qu'on appelle souvent des « bibliothèques ») qui sont utilisables telles qu'elles et que l'on peut intégrer facilement dans un programme.

Les SDK facilitent donc le travail des développeurs en leur faisant gagner du temps.
Par exemple, chacun des systèmes d'exploitation iOS et Android fournit un SDK spécifique **permettant la programmation d'applications mobiles**, d'utiliser les fonctions « natives » du terminal et proposant des éléments graphiques qu'il n'est pas besoin de redévelopper.





STORE

Les stores sont des plateformes de téléchargement d'applications mobiles.

Il en existe pour chaque système d'exploitation (**App Store** pour les appli Apple sur iPhone ou iPad, **Play Store** pour les appli Android, **Windows Store** pour les appli fonctionnant sur Windows Phone).

Une fois que votre application mobile est développée, vous pouvez donc la mettre à disposition de ses utilisateurs sur le store approprié (sauf s'il s'agit d'une **WebApp** car dans ce cas, il n'est pas nécessaire de la publier sur un store, elle s'ouvre simplement dans un navigateur web).

Nous venons de parler de **stores publics** mais il existe aussi des **stores privés** : si votre entreprise souhaite mettre à disposition d'un groupe privé d'utilisateurs (par exemple ses employés nomades : commerciaux, techniciens de maintenance...) des applications Métier sur mobile ou tablette, elle peut utiliser un store privé pour **déployer et gérer ces applications de manière privée et sécurisée**. Ces stores sont similaires aux stores publics mais ils sont accessibles uniquement par des utilisateurs identifiés.



Publier votre appli sur un store **ne suffit pas pour qu'elle soit adoptée** par ses utilisateurs potentiels : un plan de communication pertinent vous permettra de ressortir sur les stores, de même qu'un contenu adéquat fidélisera vos utilisateurs et maintiendra leur **engagement** sur la durée.

STORYBOARD

Représentation graphique simple de la manière dont l'appli va fonctionner, c'est-à-dire sa « cinématique », et décrivant donc le parcours utilisateur.

À l'origine, ce terme est utilisé dans le cinéma pour désigner l'ensemble des plans qui vont constituer le film.

Le storyboard est **l'un des livrables indispensables** dans un projet de conception d'application mobile ; il précède le **wireframe**, puis le **mockup** qui permet de mettre en œuvre des **tests utilisateurs**.





SWIPE

Le Swipe (qui signifie « Balayer » en anglais) est l'une des interactions tactiles que peut effectuer un mobinaute à l'aide de son doigt sur l'écran de son smartphone ou de sa tablette.

Le Swipe (balayage latéral) permet de faire défiler des pages de contenus, de décrocher le téléphone ou encore de fermer une application.

Voir aussi [Tap](#)

SYNCHRONISATION

La fonction de synchronisation permet à votre application de fonctionner en mode déconnecté.

Une grande partie de l'intérêt des applications mobiles réside dans la possibilité de garder l'application et ses données avec soi, tout le temps et en tout lieu. Mais le fait est que la mobilité impose, de temps à autre, **la perte de l'accès à une connexion Internet.**

C'est là qu'intervient la synchronisation (on parle aussi de fonctionnement en **mode connecté / déconnecté**). Lorsque vous perdez la connexion internet, l'application fonctionne en mode déconnecté et stocke temporairement les données dans le terminal mobile.

L'application embarque **un moteur de synchronisation** qui gère les éventuels conflits de données, les droits d'authentification, la sécurisation de la base de données implémentée sur le mobile, etc.

Lorsque la connexion internet revient, la synchronisation se met automatiquement en route.

Cette fonctionnalité est particulièrement intéressante par exemple pour les **applications mobiles à destination de commerciaux nomades.**





TAP

Le tap sur l'écran tactile de votre smartphone ou de votre tablette est l'équivalent du clic de la souris.

Le Tap permet d'ouvrir une application, de sélectionner un contenu, de saisir du texte dans une zone, etc.

↘ Voir aussi : [Swipe](#)

TEST

Les tests ont pour objectif de vérifier la **qualité** du code écrit par les développeurs.

En mode **Agile**, les tests interviennent au cours de chaque Sprint et donnent lieu à une **recette**, sur la base des exigences validées par l'entreprise (ou par le service émetteur de la demande).

Il existe différents types de tests :

- **Tests unitaires** qui concernent une seule action à l'intérieur d'une fonctionnalité,
- **Tests d'intégration** qui concernent l'ensemble d'une fonctionnalité comportant plusieurs actions, ou bien plusieurs fonctionnalités interagissant entre elles,
- **Tests de non régression** pour s'assurer que de Sprint en Sprint, des actions ou fonctionnalités précédemment testées ne sont pas dégradées.

⊕ À chaque fonctionnalité correspondent des scénarii détaillant les comportements des futurs utilisateurs et décrivant chaque action. On parle ici de « **campagne de tests** ». Le testeur reproduit donc ces scénarii et s'assure que tout est bien conforme à ce qui a été spécifié d'un point de vue **fonctionnel, technique, performance ou ergonomie**.





TEST UTILISATEUR

Un test utilisateur est une session au cours de laquelle on observe une personne, représentative de la cible visée par la future application, utiliser un prototype (**mockup**).

Il est utile, avant de démarrer le développement, de s'assurer que l'application envisagée correspond bien aux besoins des utilisateurs concernés.

Les tests utilisateurs se déroulent donc **en fin de phase de conception**. À ce stade, le projet graphique est finalisé et donne lieu à la livraison d'un « prototype » (aussi appelé « mockup »). **Ce mockup qui présente tous les aspects visuels** de l'application et **simule son fonctionnement**, est soumis à un groupe d'utilisateurs représentatif de la cible.

Cette mise en situation permet de recueillir les commentaires en termes d'**UX/UI Design** et d'effectuer les réglages nécessaires avant même que le développement n'ait démarré.

TOAST

Aussi indispensable que lors d'un brunch dominical, un toast est une petite fenêtre d'information qui s'affiche discrètement en bas de page pour donner une précision sur l'action réalisée.

Moins intrusifs que les pop-ups, les toasts informent les utilisateurs de différents événements comme la confirmation d'envoi d'un message, la rencontre d'un bug, ou un changement de connectivité du réseau.

Les toasts sont des outils d'information préférables aux boîtes de dialogue parce qu'ils n'interrompent pas le travail de l'utilisateur et lui permettent de lire l'information affichée au moment qui lui convient.

Ils aident à **améliorer l'expérience** globale de l'utilisateur et à le guider dans son usage. L'oublier, c'est prendre le risque de perdre l'utilisateur.





UX DESIGN

L'UX* Design regroupe l'ensemble des moyens et pratiques permettant de mettre en œuvre l'expérience utilisateur la mieux adaptée à la cible concernée (en terme d'utilisabilité, d'ergonomie et de ressenti).

On appelle « **Expérience utilisateur** », l'**expérience vécue par une personne** lors de son interaction avec un équipement ou une interface. La mission d'un UX designer est donc de faire en sorte que cette expérience soit la plus agréable possible, et de manière durable (quelles que soient les conditions d'utilisation ou l'humeur de l'utilisateur !)

De la qualité de l'expérience utilisateur dépend **l'adoption de l'application et l'engagement des utilisateurs dans la durée**. D'où l'intérêt de mettre en place des ateliers de **Design Thinking** ou des **tests utilisateurs**.

Dans le cadre d'un projet de développement d'application mobile, ce qui relève de l'UX Design, c'est par exemple :

- **La simplicité de prise en main** de l'appli (on n'a pas besoin d'expliquer au mobinaute comment faire)
- **L'efficacité des actions réalisées** (pas plus d'un ou deux clics pour arriver à ses fins)
- **La navigabilité de l'application** (les mobinautes sont, par nature, peu patients !)
- **La présentation des informations** (ajustée aux profils des mobinautes)

*UX = User eXperience / Expérience Utilisateur

↗ Voir aussi : [UI Design](#)

*« La plupart des gens font l'erreur de penser que le design, c'est l'apparence (...). Ce n'est pas notre avis. Le design, ce n'est pas seulement l'apparence et le style. Le design, c'est comment ça marche. »**

Steve Jobs

*« Most people make the mistake of thinking that design is what it looks like[...] That's not what we think design is. It's not just what it looks like and feels like. Design is how it works. »

UI DESIGN

L'UI* Design désigne l'ensemble des moyens permettant de créer une interface agréable et facile à prendre en main, afin d'offrir une expérience la plus agréable possible aux utilisateurs.

Pour cela l'UI designer s'attache particulièrement aux **éléments concrets de la mise en forme des pages**, c'est-à-dire : la charte ou identité visuelle de l'application (design du logo, couleurs, typographie), les boutons, les éléments graphiques (illustrations, visuels...), la navigation...

Si l'UI designer s'attache particulièrement à l'**esthétique** de l'application, **il se doit de prendre en compte l'aspect pratique également** : Imaginez que l'on vous offre une voiture au design incroyablement futuriste... mais qu'il vous soit impossible de comprendre comment la démarrer.

Parfois, il peut même faire des préconisations qui vont à l'encontre de principes d'ergonomie établis, de manière à **privilégier l'expérience utilisateur** !

*UI = User Interface (Interface Utilisateur)

↘ Voir aussi : [UX Design](#)





WEBAPP

Une WebApp, aussi appelée site mobile, est un site internet développé de manière ergonomique dans le but d'être consultable depuis un navigateur mobile.

Quels sont ses avantages ou inconvénients par rapport à un **développement hybride** ou **natif** ?

AVANTAGES

- **Facilité d'installation**
Aucune installation sur le terminal mobile n'est nécessaire
- **Accessibilité** depuis tout type de mobiles
- **Facilité d'accès**
- **Coûts de développement** peu élevés
- **Nouvelles versions** de l'application immédiatement disponibles, ne nécessite pas de publication sur les **stores**

INCONVÉNIENTS

- **Moins ergonomique**
Utilisée dans un navigateur web
- **Pas d'accès à l'environnement de l'OS**
Pas d'utilisation possible des fonctionnalités intrinsèques du terminal mobile
- **Connexion Internet obligatoire**
- Pas de figuration sur les **stores**
- **Pas de notification** possible

WIREFRAME

Un wireframe (qui signifie « représentation fil de fer » en anglais) est une représentation simple et visuelle (filaire) d'une interface.

Un wireframe permet de présenter pour un écran de votre application mobile, les différentes zones et types de contenus qu'elles contiennent, les fonctionnalités offertes et l'ergonomie.

Il s'agit donc **d'un des livrables intervenant au tout début de la conception** de votre application mobile.

Les wireframes sont précédés par les **storyboards** et donnent lieu à des **mockups**.

Selon l'objectif de votre wireframe, vous pouvez aller dans un niveau de détail plus ou moins grand.

Les wireframes peuvent s'esquisser à la main ou à l'aide de logiciels spécialisés (Balsamiq, InVision...)



POUR ALLER PLUS LOIN...

BIBLIOGRAPHIE

- « Sprint : Résoudre les problèmes et trouver de nouvelles idées en cinq jours », de Braden Kowitz, John Zeratsky, Jake Knapp – Eyrolles
- « Design d'expérience utilisateur : Principe et méthodes UX », de Sylvie Daumal – Eyrolles

RESSOURCES INFLEXSYS

- [Documentations](#)
- [Blog](#)
- [Webinars \[Replays\]](#)

Suivez-nous :



www.inflexsys.com

05 56 69 39 55

Inflexsys est une Entreprise de Services du Numérique spécialisée dans le développement d'applications mobiles professionnelles et d'objets connectés.

S'appuyant sur une solide expertise mobile, Inflexsys conseille et accompagne ses clients, PME ou Grands-Comptes, dans leur transformation digitale en concevant **des services mobiles novateurs sur-mesure et des solutions personnalisables pour la relation-client digitale.**

www.inflexsys.com

