

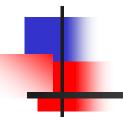




NET C# XX Xamarin

## Chapitre I

#### Introduction générale















#### Plan

- Présentation
- Conception d'applications mobiles
- Plateformes de développement
- 4 Développement natif, hybride et multiplateforme











Dév. d'applications mobiles

### 1. Présentation (1/3)

C'est quoi le développement d'applications mobiles?

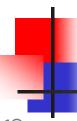












### 1. Présentation (2/3)

#### C'est le processus de fabrication de logiciels

- Pour machines telles que
  - Smartphone
  - Tablette/liseuse numérique
  - Assistant digital
  - Autre (objet intelligent etc.)
- Pour SE tels que
  - Android
  - iOS/iPadOS
  - Autre (Windows Mobile/CE, SymbianOS etc.)
- Avec des langages tels que
  - Java
  - C#
  - JavaScript, HTML5











### 1. Présentation (3/3)

- Applications mobiles => pleine croissance!
  - Accès à l'information
    - Immédiat/temps réel
    - Pratique
  - Moyen le plus populaire pour accéder à internet (particuliers/entreprises)
- Avantage pour les entreprises
  - Rester à la page
  - Rester attentif à la clientèle
  - Répondre efficacement aux besoins des
    - Clients
    - Partenaires
    - Employés











- Applications mobiles ≠ autres type d'applications (desktop, web etc.)
- => Contraintes de développement différentes!
- => Mais possèdent aussi des avantages
- Contraintes relatives à
  - Matériel
  - Logiciel
  - Plateforme utilisée
  - **Environnement**
  - Utilisateur et/ou type d'utilisation
  - Tâches implémentées
  - Processus de développement d'applications









- Ces contraintes et avantages influencent la conception d'applications mobiles
- => Exemples de contraintes et de leur d'influence sur le développement d'application









## 2. Conception d'applications mobiles



- Taille d'écran limitée
  - => Interface graphique utilisateur (GUI) plus simple/moins encombrée
  - => Composants graphiques de plus grande taille
- Ecrans tactiles prédominent
  - => GUI optimisée pour l'utilisation tactile et gestuelle
- Batterie à autonomie limitée
  - => Utilisation intelligente de ressources/économie d'énergie
- Processeur moins puissant/taille mémoire moins élevée
  - => Utilisation/optimisation intelligente des ressources
  - => Distribution de charge avec des ressources à distance (cloud etc.)
- Ressources matérielles plus limitées











## 2. Conception d'applications mobiles



- GUI : seul écran affichable à la fois
  - => Organiser l'application sous forme d'écrans
  - => Hiérarchiser intelligemment la navigation entre écrans
- Absence du multitâches applicatif
  - => Utiliser un modèle multitâches coopératif à un seul thread
- Ressources du SE plus limitées

#### Contraintes environnementales

- Connectivité : grande dépendance des réseaux GSM/3G/4G (facturés au Ko/Mo/Go)
  - => Utilisation économique des données mobiles à leur détection
- Latence des réseaux mobiles











- Appareils mobiles différents, avec des capabilités différentes relatives à
  - Capteurs
  - Connectivité
  - Périphériques
  - Etc.
- => Il convient à l'application d'adresser ces différences















- Obligation/interdiction d'utiliser un/des langages de programmation spécifiques
- Obligation d'utiliser une marque précise pour le hardware de développement
- Obligation de passage par une plateforme de publication
  - Obligation de se conformer à des normes de qualité
  - Obligation de respecter une charte de publication
- Payement de droit au développement d'applications
- Prélèvement de pourcentage sur les bénéfices











- Utilisateur de l'application
- Processus de développement
- Pas que de contraintes, mais avantages aussi!
  - Appareil constamment allumée
  - Possibilité d'une connexion réseau constante
  - Utilisation nomade/mobile
  - Présence d'une panoplie de capteurs/périphériques (Appareil photo, GPS, boussole, Gyroscope etc.)
  - Accès immédiat et constant pour l'utilisateur

=> Apprendre en considération dans la conception d'applications mobiles







## 3. Plateformes de développement (1/2)

- Quelle plateforme de développement d'applications mobiles choisir?
- Ça dépend!
  - Cas d'utilisation de l'application
  - Plateforme de l'audience ciblée
- Exemples
  - Utilisateurs majoritairement équipés d'iPhones
    - Application iOS
  - Utilisateur équipés d'iPhones + Android phones
    - Application multiplateforme iOS/Android







# 3. Plateformes de développement (2/2)



- Beaucoup ciblent Android d'abord
  - 70% de smartphones sous Android
  - Google play store moins restrictif que Apple app store
- D'autre choisissent iOS
  - Meilleure rétention utilisateur
  - Optimisation plus simple
- Autre considérations dans le choix de plateforme
  - Stratégies de monétisation
  - Comportement utilisateur anticipé
  - Facteur géographique
  - Facteurs culturels











## 4. Développement natif, hybride et multiplateformes (1/4)



- Confusion?
- Meilleure approche de développement à choisir?
- Eléments de décision
  - Budget
  - Fonctionnalités de l'application
  - Expertise du développeur
  - Audience ciblée
- Différentes approches de développement
  - Natif
  - Hybride
  - Multiplateforme









## 4. Développement natif, hybride et multiplateformes (2/4)

#### Approche native

- Application et code source séparée pour chaque plateforme
- Code compilé en code machine (natif)
- Différents kits de développement pour différents SE
  - iOS
    - IDE : AppCode, Atom, Xcode
    - LP: C++, Swift, C-Sharp, Objective-C
  - Android
    - LP: Kotlin, Java
- Avantages
  - Meilleur expérience GUI, pas de problèmes de comptabilité, performances élevées, toutes les fonctionnalités du SE sont disponibles au développeur
- Inconvénient
  - Application différente pour chaque SE: plus de coût (temps, effort, connaissances)







**X** Xamarin

### 4. Développement natif, hybride et multiplateformes (3/4)

#### Approche hybride

- Code (unique pour toutes les plateformes) exécuté sur un composant WebView (ce sont des pages web)
- Le composant est mis dans un emballage natif « thin wrapper »
- **Technologies** 
  - Outils de création
    - Visual Studio, Ionic, Apache Cordova
  - - Javascript, CSS, HTML
- Avantages
  - Développement rapide, plusieurs plateformes supportées par le même code source
- Inconvénient
  - Expérience utilisateur (UX) limitée, moindre réactivité, support d'applications complexes plus limité, accès plus restreint au fonctionnalités du SE (à travers des plugins)



## NET C# XX Xamarin





- Code source unique
- Code compilé en code machine natif pour chaque plateforme
- Technologies
  - React native (Facebook), Xamarin.Forms et MAUI .NET (Microsoft), Flutter (Google)
- L'approche multiplateforme reprend les avantages des approches natives et hybrides, tout en minimisant leurs inconvénients
- => Notre choix de développement (avec MAUI .NET en C#)

