

Vývoj mobilných aplikácií*

Lujza Šufliarska

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xsufliarska@stuba.sk

30. september 2021

Abstrakt

V súčasnej dobe má skoro každý človek smartfón, v ktorom má množstvo aplikácií. Tieto aplikácie používa dennodenne a sú dôležitou súčasťou nie len týchto zariadení, ale aj každodenného života. Vieme, že smartfóny majú rôzne operačné systémy a medzi ne patrí aj iOS, pre smartfóny od spoločnosti Apple, a android, pre zariadenia od iných výrobcov. V mojom článku sa zameriam na vývoj mobilných aplikácií pre zariadenia so systémom iOS a android. Pozrieme sa na to, ako sa aplikácie vyvíjajú, etapy tvorby aplikácií, vývojové prostredia, front-end a back-end vývoj, UX a UI, dizajn aplikácií, wireframy a ďalšie. ...

1 Úvod

Motivujte čitateľa a vysvetlite, o čom píšete. Úvod sa väčšinou nedelí na časti.

Uveďte explicitne štruktúru článku. Tu je nejaký príklad. Základný problém, ktorý bol naznačený v úvode, je podrobnejšie vysvetlený v časti ?? . Dôležité súvislosti sú uvedené v častiach ?? a ?? . Záverečné poznámky prináša časť 13.

2 Mobilná aplikácia

Mobilná aplikácia je softwarová aplikácia vytvorená napríklad pre smartphony a tablety, ktorú je potrebné nainštalovať. Zariadenia sú už vopred vybavené niektorými (základnými) aplikáciami, ale oveľa viac aplikácií (vrátane tých predinštalovaných) je k dispozícii prostredníctvom aplikácie určenej pre ich šírenie – pre systém Android je to aplikácia Obchod Play a pre systém iOS App Store.

Mobilné aplikácie sú určené napr. na produktivitu, navigáciu, komunikáciu, zábavu, šport, fitness a mnoho ďalšieho. V súčasnej dobe máme aplikácie skoro na všetko. Jedným z najpopulárnejších druhov aplikácií sú sociálne siete. Týchto aplikácií je nesmierne veľa a sú tiež najpopulárnejšou oblasťou vývoja.

Načo sú nám mobilné aplikácie, keď existujú webové stránky s rovnakým obsahom? Mobilné aplikácie nám na rozdiel od web stránok ponúkajú väčšiu

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Vladimír Mlynarovič

interaktivitu, konkrétnejšie informácie v jednoduchej a intuitívnej podobe a okrem toho, aplikácia je menšia, prehľadnejšia, pohodlnejšia, ľahšia na prácu, ... ako webová stránka otvorená na mobilnom zariadení.

3 Klasifikácia aplikácií

- Webové aplikácie

Webové aplikácie sú prístupné cez internet. Tieto aplikácie nepotrebujeme stiahnuť, otvárame ich v prehliadači (Safari, Chrome, Firefox...). Fungujú však len vtedy, keď sme pripojení na internet, a preto sú najmenej praktické.

- Hybridné aplikácie

Hybridné aplikácie sú kombináciou webovej a natívnej aplikácie. Je možné ich programovať naraz pre obidve platformy. Pri zložitejších funkciách je nutné doprogramovať pre jednotlivé platformy jednotlivé súčasti samostatne. Tento druh aplikácií môže, ale nemusí vyžadovať internetové pripojenie.

- Natívne aplikácie

Natívne aplikácie sú aplikácie napísané v programovacích jazykoch určených pre iOS alebo pre Android a bežia priamo na operačnom systéme zariadenia. Väčšina mobilných aplikácií je natívnych. Pretože je určená primárne pre jednu platformu, tak sa tento druh aplikácie napr. rýchlejšie inštaluje, ľahšie obsluhuje, nie je závislá od internetového pripojenia, ľahko spolupracuje s funkciami zariadenia... Tento vývoj si však vyžaduje vysoké náklady.

- Cross-platformové aplikácie

Cross-platformové aplikácie sú vytvorené pre viacero platforiem súčasne. Kód je pre každú platformu skompilovaný do natívnej aplikácie. Tento vývoj môže byť efektívny a menej náročný na zdroje.

- Progresívne aplikácie

Tento typ aplikácií sa stáva čoraz populárnejším. Aplikácia sa nainštaluje do mobilného zariadenia prostredníctvom prehliadača a nie pomocou obchodu s aplikáciami.

Aplikácie môžeme ešte rozdeliť do skupín podľa ich účelu:

- Zábava - hry
- Pomocníci –počasie, hľadanie informácií...
- Sociálne siete
- I-komercia – nákup cez aplikácie
- ...

4 Proces vývoja aplikácie

Vývoj mobilných aplikácií je zložitý proces vytvárania aplikácií určených pre mobilné zariadenia. Toto odvetvie je stabilné a výnosné a zaisťuje množstvo pracovných pozícií.

Ako každý vývoj, tak aj vývoj mobilných aplikácií pozostáva z niekoľkých krokov, ktoré tiež závisia od nejakých činiteľov. Nikde nie je presne napísaný postup pre vývoj aplikácií. Ale takou základnou kostrou práce môžu byť tieto body, respektíve fázy:

1. Plán
2. Grafický návrh aplikácie a prototypovanie
3. Programovanie aplikácie
4. Testovanie
5. Umiestnenie aplikácie na trh
6. Údržba aplikácie

4.1 Plán

Ešte predtým ako začneme vyvíjať samotnú aplikáciu, treba poznať niekoľko odpovedí na otázky charakterizujúce predstavovanú aplikáciu. Základom k úspešnému projektu je poznať cieľ. Je dobré sa zamyslieť nad niekoľkými otázkami, ako sú napríklad:

- Čo je zmyslom našej aplikácie?
- Je takáto aplikácia potrebná?
- Aké problémy má aplikácia riešiť?
- Pre akú cieľovú skupinu má byť aplikácia určená?
- Aký máme budget?

4.2 Grafický návrh aplikácie

Teraz prichádza na rad dizajn, kreslenie, storyboarding a wireframing. Najprv si vytvoríme základnú predstavu aplikácie – UI (user interface – užívateľské rozhranie) a UX (user experience) dizajn (čo je UI a UX je popísané v nasledujúcich podkapitolách).

Na vytvorenie mockupov a screenshotov môžete použiť napr. Adobe XD, Adobe Photoshop, Adobe Illustrator alebo aj Sketch. Storyboard alebo wireframing je podstatnou súčasťou vývoja. Wireframe predstavuje analytickú fázu do grafickej podoby. V priemere sa nakreslí 120 – 150 obrazoviek. Všetky tieto obrazovky a stavy prvkov pomáhajú vo vizualizácii všetkých scenárov.

Po navrhnutí wireframov a dizajnu môžeme začať oživovať aplikáciu. Vyrobíme klikateľný prototyp. Opisujeme všetky prechody medzi obrazovkami a stavmi obrazovky, všetky scenáre, animácie... Na to môžeme použiť nástroje Marvel, Flinto, InVision alebo Adobe XD, pomocou ktorých vytvoríme prototyp a uvidíme prvé reakcie a funkcie aplikácie.

4.2.1 UI

User interface zahŕňa jednotlivé stránky, podstránky, tlačidlá, vizuálne prvky, ktoré pomáhajú pri komunikácii s aplikáciou alebo zariadením. Ide o navrhnutie vizuálne atraktívneho softvéru a ľahko navigovateľného softvéru pre všetky typy zariadení. Všetky prvky musia byť navrhnuté tak, aby ich úlohy a funkcie boli na prvý pohľad jasné. Tiež musia byť umiestnené na správnych miestach.

4.2.2 UX

User experience je zážitok, skúsenosť užívateľa po používaní aplikácie alebo webu. Chceme urobiť prepínania medzi jednotlivými podstránkami, stránkami, funkciami urobiť čo najjednoduchšie a intuitívne, aby užívateľ použil náš produkt opakovane a odporučil ho svojim známym. UX zahŕňa aj faktory ako sú dizajn, dostupnosť, marketing, použiteľnosť, výkon systému, užitočnosť...

4.2.3 Wireframe

Wireframe („drôtený model“) pomáha pri návrhu dizajnu, obsahu a funkcií. Väčšinou je to prvá vec, ktorá sa ukazuje klientovi. Prezentuje rozmiestnenie a formu projektu. Definuje textový aj grafický obsah, rozmiestnenie funkčných prvkov, navigáciu, nadpisy, kľúčový text a tlačidlá. Nie je to ale grafický návrh, neobsahuje obrázky, nie sú dôležité farby... Pomocou neho sa vytvorí vzor pre grafikov a vývojárov a použije sa aj na diskusiu s klientom. Robiť zmeny a úpravy je oveľa jednoduchšie a rýchlejšie vo wireframe, ako v hotovom grafickom návrhu alebo naprogramovanej verzii. Wireframe je možné robiť na papier alebo aj v softvéri.

4.3 Programovanie aplikácie

Teraz si treba vybrať operačný systém, na ktorom bude aplikácia bežať. Môžeme si vybrať len jeden alebo môžeme vyvinúť cross-platformovú aplikáciu, ktorá najlepšou voľbou. Takto zaistíme, že aplikácia bude fungovať na každom zariadení.

Väčšinou sa to dá rozdeliť na 3 časti:

- Back-end vývoj
- APIs – aplikačné programové rozhranie
- Front-end vývoj

4.3.1 Back-end vývoj

Back-end je vývoj potrebný zo strany servera. Používa sa na uloženie, zaistenie a spracovanie dát. Je to to, čo sa deje, keď užívateľ interaguje s aplikáciou. Dejú sa tu registrácie, prihlásenia, posielanie správ, ukladanie údajov na cloud a mnoho ďalšieho. Back-endový vývoj sa zameriava na ukladanie informácií do vzdialenej databázy, spracovanie vstupu užívateľských dát, skriptovanie na pridanie logiky do interaktivity a vytváranie architektúry, ktorá umožňuje rýchle a jednoduché triedenie informácií. Niektoré aplikácie ako kalkulačka, fotoaparát,

kompas nepotrebujú back-endový vývoj, pretože na to, aby fungovali nepotrebujú internetové pripojenie ani ukladanie a obnovovanie údajov a dát zo vzdialeného servera. Ale aplikácie ako Amazon, Netflix, Uber nemôžu fungovať bez back-endu.

4.3.2 APIs – aplikačné programové rozhranie

Application programming interface – API je rozhranie, ktoré definuje spôsoby, ktorými môže aplikácia požadovať služby od knižníc alebo operačných systémov. Aplikácie môžu medzi sebou komunikovať, spoločne pracovať a odosielať si údaje bez toho, aby sa muselo pracovať osobitne s každým programom. Teda umožňuje dvom aplikáciám nájsť sa a vzájomne spolupracovať. Vďaka nemu je rozšírenie alebo napojenie na ďalší systém jednoduché. API pomáha užívateľom rýchlo integrovať a pripojiť sa k vzdialenému cloudovému dátovému serveru.

4.3.3 Front-end vývoj

Front-end je to čo vidí používateľ. Stará sa o zobrazenie konkrétnych dát, ktoré sú aplikáciou spracované.

4.4 Testovanie

Pri testovaní si všímame či sú texty na správnych miestach, či sú funkčné všetky odkazy, tlačidlá a funkcie... Dobrými nástrojmi na testovanie sú napríklad Solidify alebo Framer. Netreba si myslieť, že keď niečo fungovalo v predchádzajúcich verziách, že to funguje aj vo finálnej verzii. Preto sa treba vrátiť k návrhom a prejsť si všetky funkcie krok za krokom.

Funkčnosť aplikácie je nesmierne dôležitá, preto testujeme na skoro všetkých dostupných zariadeniach. Je dobré testovať aj na modeloch, ktoré dokážu odhaliť množstvo chýb. Väčšinou sú to teda problémové zariadenia, ako napríklad Samsung s5 mini.

4.5 Umiestnenie aplikácie na trh

Pred publikovaním aplikácie si treba založiť účet na Google Play alebo App Store. Vytvorenie účtu môže trvať aj niekoľko dní, a preto na to netreba zabudnúť.

Pri publikácii je dôležité myslieť aj na pravidlá obchodov App Store a Google Play, ktoré sú rozdielne. Napríklad. Google hneď nekontroluje aplikáciu, ale môže to niekedy urobiť. Na druhej strane, Apple ju kontroluje hneď a môže to trvať aj 2 týždne. Existuje aplikácia PreApps, ktorá vám poskytne spätnú väzbu ešte predtým, ako svoju aplikáciu spustíte do obehu.

4.6 Údržba aplikácie

Máte hotovú aplikáciu, ktorú používa niekoľko užívateľov. Ale tu práca nekončí. Aplikáciu treba ďalej udržiavať, optimalizovať, vylepšovať, aby bola stále aktuálna a používaná. Keby sme sa o aplikáciu nestarali, raz by sa zrútila, nefungovala a nakoniec by bola odstránená z obchodu.

5 Propagácia aplikácie – pred spustením a po

Propagácia aplikácie sa dá rozdeliť na propagáciu pred spustením a po. Pred spustením aplikácie sa treba zamerať na vytvorenie záujmu o aplikáciu. Treba poskytnúť všetky dôležité informácie originálnym spracovaním. Dobrým nápadom je spolupracovať s osobami, ktoré majú široký dosah prostredníctvom sociálnych sietí. Môžu si vyskúšať aplikáciu ešte predtým, ako je sprístupnená verejnosti a spropagujú ju cez svoje sociálne siete najlepšie pozitívnu recenziou. Aplikáciu môžete propagovať aj prostredníctvom médií alebo platených článkov. Netreba zabudnúť na atraktívny popis, informatívne obrázky v obchodoch s aplikáciami a prípadne aj propagačné video.

Po spustení aplikácie sa zamerajte na sociálne siete. Sociálne siete sú najpoužívanejším druhom aplikácií, a preto je najlepším miestom na propagáciu. Pripravte atraktívne vizuály, videá a pripojte aj link na stiahnutie aplikácie. Ľudia si skôr stiahnu aplikáciu odporúčenú známymi alebo priateľmi. Toto je dobré využiť a poskytnúť odmeny za každého pozvaného užívateľa. Udržujte kontakt. Odpovedajte na otázky a dotazy v obchodoch s aplikáciami. Nezabudnite aj na negatívne komentáre, ktoré vám tiež poskytujú spätnú väzbu. Ako vidíte, je množstvo možností, ako propagovať aplikáciu pred aj po spustení. Stačí si len vybrať, ísť s dobou, aktualizovať propagáciu a teda venovať sa tomu.

6 Platformy na vývoj aplikácií

Platformy na vývoj aplikácií sa skladajú z rôznych nástrojov a komponentov, ktoré umožňujú vývojárom písať, testovať a vydávať aplikácie. Medzi platformy na vývoj aplikácií patria napríklad:

- **Kotlin** – Tento programovací jazyk sa stal oficiálnym jazykom pre Android. Je možné s ním pracovať napr. vo vývojárskom prostredí Android Studio.
- **Swift** – Swift je oficiálnym programovacím jazykom pre iOS aplikácie. Bol vyvinutý spoločnosťou Apple, a aby uľahčil prácu programátorom. Často je používaný XCode ako vývojárske prostredie, ktorý je tiež produktom Apple.
- **Flutter** – Umožňuje multiplatformový vývoj. Výslednú aplikáciu je možné zkompilovať do natívneho kódu pre Android aj iOS. Je to mladá technológia, vyvinutá spoločnosťou Google, ktorá používa programovací jazyk Dart 2.
- **React Native** – Hybridný framework od Facebooku, ktorý umožňuje písanie aplikácii pre Android aj iOS.
- **Apache Cordova** – Jedná sa o ďalší hybridný framework, ktorý je od spoločnosti Apache.

7 Monetizácia aplikácie

Na to, aby ste mali zisk z aplikácie, nemusí byť aplikácia hneď spoplatnená. Najrozšírenejším spôsobom zárobku sú reklamy. V aplikácii môžete poskytnúť

miesta pre reklamu. Tá môže byť spoplatnená napr. na základe počtu kliknutí na ňu. Alebo môžete poskytnúť užívateľom spoplatnenú verziu bez reklám. Ďalším spôsobom je FREEMIUM model. Aplikácia je voľne dostupná, ale má obmedzené funkcie, ktoré majú používateľa tak zaujať, aby si zaplatil plnú verziu PREMIUM so všetkými funkciami. V aplikácii môžete vytvoriť špeciálne predmety, herné vychytávky, doplnkové funkcie... , ktoré si budú môcť užívatelia kúpiť za reálne peniaze. Tento spôsob tvorí až 50% obratu celého trhu. Takže aj spôsobov, ktorými môžete mať zisk z aplikácie je množstvo. O mnoho viac ako bolo spomenutých.

8 Cena aplikácie

Ak sme si dali aplikáciu vyrobiť firme, tak nás určite zaujíma aj cena. Cena aplikácie je ovplyvňovaná viacerými faktormi. Medzi ne patrí napríklad dostupnosť aplikácie. To znamená, že na akých zariadeniach bude aplikácia dostupná. Na Android, iOS alebo oboch? Tak isto aj na akých verziách operačného systému bude dostupná. Tu treba dať pozor hlavne na iOS.

Ďalším faktorom je napr. zložitosť a rozsah funkcií. Ideálne je najprv vytvoriť aplikáciu so základnými funkciami, ku ktorým sa potom pridávajú zložitejšie funkcie, ktoré môžu vývoj predĺžiť a zdražiť.

Dizajn pri mobilných aplikáciách nie je len vzhľad aplikácie, ale aj užívateľské rozhranie (user interface – UI), ale aj funkčnosť a navigácia aplikácie, tzv. user experience – UX.

Veľkosť vývojárskeho tímu tiež ovplyvňuje cenu. Počet členov závisí od zložitosti aplikácie, od požadovaných funkcií. Väčšinou základ tvorí projektový manažér, UX dizajnér, minimálne 2 vývojári a tester. Vo väčšom tíme už nájdeme viacej vývojárov, ktorí sú špecializovaní na určité oblasti – Android, iOS, backend...

Na čom by sa nemalo šetriť je testovanie výslednej verzie aplikácie. Pozorné testovanie môže predísť ďalším nákladom a urýchliť projekt. Ak máme projekt rozdelený na viacej fáz, je dobré po každej fáze všetko otestovať. Nemôžeme zabudnúť na dlhodobé výdavky na údržbu, úpravu aplikácie, správu dát, vytváranie updateov, poskytovanie podpory užívateľom, výdavky na propagáciu aplikácie...

9 iOS vs. Android vývoj

Počet užívateľov smartphomov rastie každým dňom a tak isto narastá aj počet vývojárov aplikácií. V súčasnosti najpoužívanějšími mobilnými operačnými systémami sú iOS, od firmy Apple a Android, od spoločnosti Google. Vývoj aplikácií pre tieto systémy je odlišný a tieto rozdiely si povieme v tejto kapitole.

Programovací jazyk

Na vývoj aplikácií pre systém Android sa používa hlavne programovací jazyk Java, vďaka ktorému je vývoj pre tento systém jednoduchší, pretože je používaný väčšinou vývojárov. Ďalšie programovacie jazyky, ktoré sa môžu použiť na vývoj Android aplikácie sú napríklad Kotlin, Flutter a React Native.

Na aplikácie pre iOS sa používa objective-C. Je to exkluzívnejšie, čo môže byť prekážkou pre vývojárov menej kvalifikovaných v iných programovacích ja-

zykoch. Okrem objective-C je možné použiť napríklad programovací jazyk Swift, Flutter alebo React Native.

ESTE DOPLNIM

10 Vývoj aplikácie pre Android

11 Vývoj aplikácie pre iOS

12 Iná časť

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť 12.1), a potom na ešte nejaké (časť 12.1).¹

Môže sa zdať, že problém vlastne nejestvuje [1], ale bolo dokázané, že to tak nie je [2,3]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [4]. Dôležité veci možno *zdôrazniť kurzívou*.

12.1 Nejaké vysvetlenie

Niekedy treba uviesť zoznam:

- jedna vec
- druhá vec
 - x
 - y

Ten istý zoznam, len číslovaný:

¹Niekedy môžete potrebovať aj poznámku pod čiarou.

1. jedna vec
2. druhá vec
 - (a) x
 - (b) y

13 Záver

14 Zdroje

ESTE VYGOOGLIT

Okej tu sú zatiaľ zdroje, inak mi ich zatiaľ nechce brať

<https://sk.eyewated.com/co-je-mobilna-aplikacia/>

<https://winpcguide.ru/sk/the-security-of-your-computer/procedura-razrabotki-mobilnogo-prilozheniya/>

<https://www.eliteml.sk/blog/mobilne-aplikacie-v-skratke/>

<https://aws.amazon.com/mobile/mobile-application-development/>

<https://sk.eyewated.com/android-os-vs-apple-ios-co-je-pre-vyvojarov-lepsie/>

https://cs.wikipedia.org/wiki/V%C3%BDvoj_mobiln%C3%ADch_aplikac%C3%AD

C3%AD

<https://www.emm.sk/blog/our-customers-have-traditionally-valued-special-attention/>

<https://synetech.cz/cs/blog/vyvoj-mobilnich-aplikaci-krok-po-kroku>

<https://touch4it.sk/blog/ux-a-ui-dizajn>

<https://www.openxcell.com/mobile-app-development/>

<https://buildfire.com/understanding-mobile-app-development-lifecycle/>

<https://sk.wikipedia.org/wiki/Wireframe>

<https://pixelfield.cz/vyvoj-aplikaci/#C1>

<https://www.vivantina.com/marketingovy-slovník/api/>

http://wiki.gis.com/wiki/index.php/Application_programming_interface

<https://www.emm.sk/blog/proces-vyvoja-mobilnej-aplikacie-cast-2/>

<https://www.eliteml.sk/blog/ako-postupovat-pri-vyvoji-mobilnej-aplikacie-podme-krok-za-krok>

<https://themanifest.com/app-development/blog/long-term-app-maintenance>

<https://sk.eyewated.com/android-os-vs-apple-ios-co-je-pre-vyvojarov-lepsie/>

<https://karaokeonlain.ru/sk/kak-razrabatyvat-bolshie-prilozheniya-etapy-process-sozdaniya/>

<https://ieeexplore.ieee.org/document/7972725>

Literatúra

- [1] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.
- [2] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. *Software Process: Improvement and Practice*, 10:143–169, April/June 2005.
- [3] Krzysztof Czarnecki and Chang Hwan Peter Kim. Cardinality-based feature modeling and constraints: A progress report. In *International Workshop on Software Factories, OOPSLA 2005*, San Diego, USA, October 2005.

- [4] Carnegie Mellon University Software Engineering Institute. A framework for software product line practice—version 5.0. http://www.sei.cmu.edu/productlines/frame_report/.