

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	Letní semestr 2016
Autoři – jméno, příjmení, xname	Bc. Anton Danilov, xdana09 Bc. Vladimír Danilov, xdanv09 Bc. Marek Chadim, xcham30
Téma	Metodiky pro vývoj mobilních aplikací
Datum odevzdání	15. 5. 2016

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky

Studijní program: Aplikovaná informatika
Obor: Informační systémy a technologie

Zlepšování procesů budování IS - Metodiky
pro vývoj mobilních aplikací

2016

Abstrakt

Hlavním cílem této práce je popsat metodiky, uplatňované při vývoji mobilních aplikací. Práce se zaměřuje nejprve na specifika mobilních aplikací. Následuje popis a srovnání agilních a rigorózních metodik. V závěrečné části jsou pak popsány dvě často používané konkrétní metodiky.

Klíčová slova

Metodika, vývoj, aplikace, mobilní, agilní

Abstract

The main objective of this paper is to describe the methodology applied in the development of mobile applications. The work focuses first on the specifics of mobile applications. Following is a description and comparison of agile and rigorous methodologies. The final section then describes two frequently used a particular methodology.

Keywords

Methodology, development, application, mobile, agile

Obsah

Úvod	6
1 Specifika mobilních aplikací	8
1 Rigorózní metodiky	10
2 Agilní metodiky	13
3 Porovnání rigorózních a agilních metodik.....	15
4 Nativní aplikace	19
5 Webové aplikace.....	20
6 Často používané metodiky.....	20
6.1 Metodika pro vývoj UX mobilní aplikace.....	20
6.2 Mobile-D	22
7 Závěr	24
8 Citovaná literatura.....	25

Úvod

Trh s mobilními telefony prodělal za posledních 15 let neuvěřitelný rychlý vývoj. Od zařízení, které uměla pouze volat na určité číslo, po chytré telefony, jež umí sami najít nejbližší obchod, objednat zboží v určitý čas apod. S trochou nadsázky lze říct, že jejich primárním účelem již není telefonické spojení a v jistém ohledu na ně může být pohlíženo jako na malé kapesní počítače. Tyto změny mají dopad na celou moderní společnost a na všechny její ekonomické odvětví. S příchodem chytrých mobilních telefonů, které zpravidla obsahují vyspělý operační systém, velký dotykový displej a uživatelské rozhraní uzpůsobené pro dotykové ovládání, se z mobilních telefonů stal univerzální komunikační nástroj.

Chytré mobilní telefony se stávají v dnešní době nedílnou součástí každého člověka. Lidé používají mobilní telefony například pro přístup k internetu, video-konference, přístup na sociální sítě a mnoho dalších činností, usnadňující každodenní pracovní i soukromý život.

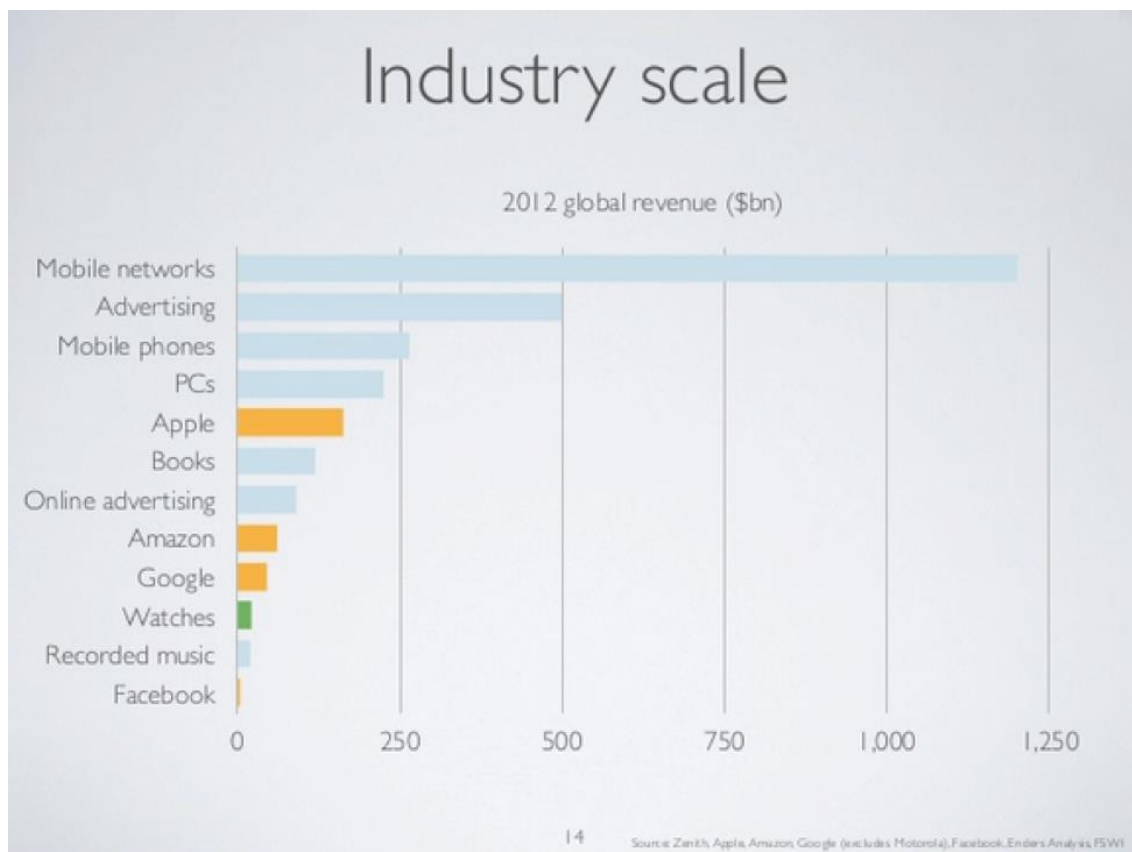
Důkazem o obrovském rozmachu chytrých mobilních telefonů může být fakt, že v roce 2013 tato zařízení poprvé předčila v prodeji běžné telefony. V roce 2014 bylo prodáno celkem 1,88 miliard telefonů, z nichž 1,245 miliard tvořily právě smartphony¹.

¹ Smartphone (z angl.) čili chytrý telefon je mobilní telefon, který využívá pokročilý mobilní operační systém a aplikační rozhraní, jež umožní instalaci nebo úpravy programů (Oxford Dictionaries, 2016).



Obrázek 1 - Vývoj počtu stažených aplikací v roce 2013 (Evans, 2013)

Jak si lze na obrázku 1 všimnout, počet stažených aplikací neustále roste, což je také důsledek již zmíněného růstu prodeje chytrých mobilních telefonů. Očekává se, že v roce 2017 budou příjmy za prodané mobilní aplikace zhruba na úrovni 63,5 miliardy korun, přičemž by měly každým rokem nadále růst a to v rozmezí 15 – 20% za rok.



Obrázek 2 - Objem příjmů v jednotlivých oblastech ve světě za rok 2012 (Evans, 2013)

Na obrázku 2, můžeme vidět výrazný růst ve všech kategoriích aplikací. Pro výzkum trhu mobilních komunikací je jasné, že mobilní aplikace nástrojů samotných marketingových komunikací jsou transformovány do mediálních distribučních kanálů. Ukazatelé, jako je rozsah, četnost kontaktů a zachování sazby v mobilních aplikacích je oslňující. Ve Spojených státech je použití mobilních aplikací je již větší než všechny ostatní mediální kanály - to je 82% času.

Velký vliv na poptávku chytrých telefonů mají sociální sítě jako například Facebook², největší sociální síť na světě. Facebook pomocí mobilních aplikací zprostředkovává okamžitou sociální provázanost a nepřetržitou sociální interakci.

1 Specifika mobilních aplikací

Mobilní aplikace se v dnešní době těší velké a neustále stoupající popularitě. Stále více firem, ale dokonce i jejich jednotlivých produktů, má svou vlastní mobilní aplikaci.

² Facebook je rozsáhlý společenský webový systém sloužící hlavně k tvorbě sociálních sítí, komunikaci mezi uživateli, sdílení multimediálních dat, udržování vztahů a zábavě.

Mobilní zařízení mají několik specifík, která ovlivňují vývoj aplikací, které jsou jim určeny a která je třeba zohledňovat ve všech jeho fázích.

Mezi hlavní a největší vlastnosti mobilních systémů patří jejich rychlý vývoj. S příchodem prvních mobilních zařízení se vývojáři museli vypořádat hlavně s hardwarovým omezením, spočívající především ve velmi omezeném množství paměti a nevykonného CPU³. Výpočetní výkon dnešních smartphonů je však již více než dostatečný a navíc se neustále zvyšuje. Kromě obvyklého Moorova zákona⁴, tedy exponenciálního růstu výpočetního výkonu, zde hraje roli i poměrně časté vydávání nových verzí operačních systémů (příkladem může být operační systém Android, jehož nové verze vychází každý rok, přičemž i v průběhu roku je vydáno několik větších či menších upgradů verzí stávajících). Často jsou vydávány také verze systému pro vývojáře, například i půl roku před samotným komerčním nasazením této verze systému (Google, 2016).

Další vlastnost je roztržštěnost hardwarových zařízení (Sawers, 2015). Hlavně se jedná o rozdílné velikosti obrazovky (mobilní telefon, tablet a chytré hodinky) a hustotu obrazových bodů na ni. Toto se týká hlavně platformy Android, avšak částečně i ostatních platforem. Tato vlastnost často prodlužuje vývoj, protože pokud má vypadat aplikace dobře na všech zařízeních, musí se tento fakt při jejím vývoji zohlednit.

Co se týče způsobů vývoje, tak je třeba brát v potaz několik faktorů. Kromě nativních nástrojů lze aplikace vyvíjet i multiplatformně (knihovny libGDX⁵, Unity3D⁶ aj.) popřípadě jako webovou aplikaci (libGDX, 2013) (Unity3D, 2016) (Google, 2016).

Jako poslední specifikum bych uvedl rozdíly mezi vývojem aplikací na telefon či tablet a klasickou desktopovou verzí. Vývojáři by měli například při návrhu GUI⁷ brát ohled na velikost obrazovky uživatele a sníženou přesnost při používání dotykové

³ Centrální procesorová jednotka (zkratka CPU, anglicky central processing unit) (Kasík, 2015) je v informatice označení základní elektronické součásti v počítači, která umí vykonávat strojové instrukce, ze kterých je tvořen počítačový program a obsluhovat jeho vstupy a výstupy.

⁴ Mooreův zákon je empirické pravidlo, které roku 1965 vyslovil chemik a spoluzakladatel firmy Intel Gordon Moore. Jeho původní znění bylo: „počet tranzistorů, které mohou být umístěny na integrovaný obvod, se při zachování stejné ceny zhruba každých 18 měsíců zdvojnásobí“ (Intel, 2015)

⁵ libGDX je zdarma a open-source herní framework pro vývoj aplikací, napsaný v programovacím jazyce Java

⁶ Unity 3D je multiplatformní herní engine vyvinutý společností Unity Technologies. Byl použit pro vývoj her pro PC, konzole, mobily a web.

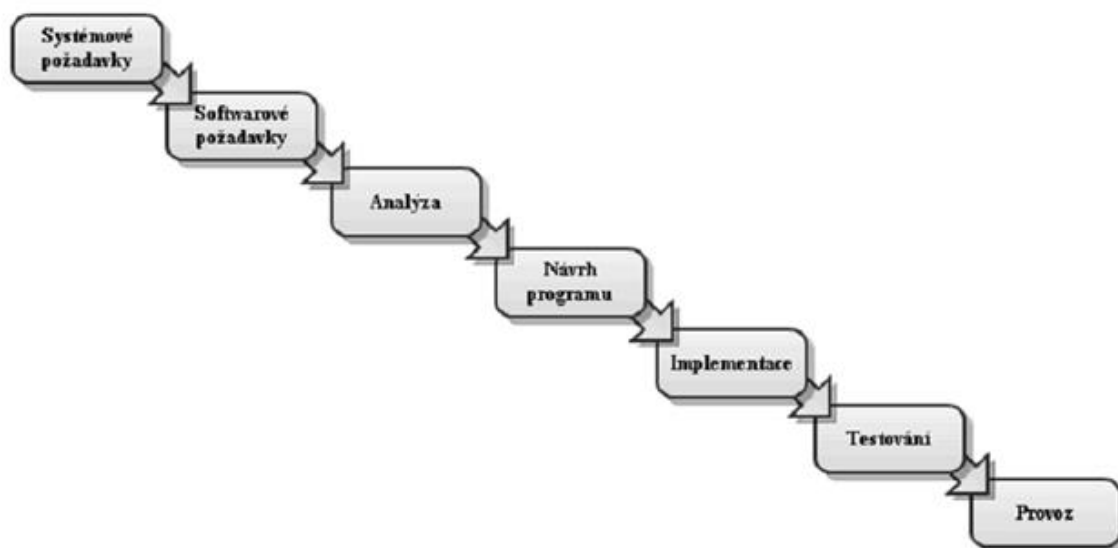
⁷ Grafické uživatelské rozhraní (anglicky Graphical User Interface, známe pod zkratkou GUI) je uživatelské rozhraní, které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků.

obrazovky. Velká tlačítka by mohla zakrývat obsah aplikace nebo by v opačném případě byla obtížně dostupná. Dále například efektivita psaní na softwarové klávesnici je mnohem nižší než na té počítačové. Dobré aplikace se proto vždy snaží, aby se uživatel dostal k důležitému obsahu v rámci několika kliknutí.

Je třeba zvážit všechny výše uvedené faktory před výběrem té nejvhodnější metodiky pro vývoj mobilní aplikace.

1 Rigorózní metodiky

Rigorózní metodiky představují největší skupinu metodik, které se zabývají podrobným popisem procesu vývoje software. Kromě samotného vývoje rovněž kladou velký důraz na podrobnou dokumentaci. Celý proces je charakterizován tzv. vodopádovým modelem, který znázorňuje posloupnost jednotlivých činností v průběhu celého vývojového cyklu. Podle této skupiny metodik je vývoj software jasně definovaný proces, od kterého se nelze odchýlit (Evans, 2013).



Obrázek 3 - Vodopádový model

Označení vodopádový model vychází z přirovnání posloupnosti jednotlivých fází k protékání vody vodopádem, jak je znázorněno na obrázku 3. Winston W. Royce⁸ jej poprvé definoval ve svém článku „Managing the Development of Large Software System“ v roce 1970. Model byl vyvinut s cílem lépe se vyrovnat s rostoucí složitostí produktů leteckého průmyslu. Royce se ve svém článku zmiňuje o sedmi základních fázích:

- Systémové požadavky (System requirements)
- Softwarové požadavky (Software requirements)
- Analýza (Analysis)
- Návrh programu (Program design)
- Implementace (Coding)
- Testování (Testing)
- Provoz (Operations)

Základní myšlenka vodopádového modelu vychází ze sekvenčního přístupu k jednotlivým fázím. Model je charakteristický tím, že vstoupit do další fáze lze pouze tehdy, je-li ta předchozí kompletně dokončena a uzavřena (Royce, 1970).

V každém projektu je nutné uvažovat nad třemi základními atributy - časem, náklady a rozsahem projektu (zadání). Některé z nich jsou fixní (neměnné), jiné se naopak v průběhu projektu mohou měnit. V případě rigorózních metodik je neměnnou veličinou rozsah projektu - resp. to, co je zadáním od zákazníka.

Tento přístup umožňuje velmi dobré řízení projektu, jelikož role a jednotlivé fáze během vývojového cyklu jsou předem jasně definované. Bohužel z výše popsaného vyplývají i zřejmé nevýhody (Jonák, 2015).

Hlavní nevýhodou vodopádového modelu je fakt, že v praxi, zejména u rozsáhlejších projektů, nelze prakticky dokončit jednu fázi a zahájit další, bez toho aniž by se k ní v budoucnu bylo nutné opět vracet. Požadavky klientů se mohou v průběhu realizace programu měnit. Ti mohou vznášet další požadavky například po zhlédnutí prototypu, jelikož si uvědomí potřeby, které je zprvu nenapadly. V takovýchto

⁸ Winston Walker Royce (15 srpna 1929 - 7. 6.1995) byl americký počítačový vědec, ředitel Lockheed Software Technology Center v Austinu, Texas.

případech je nutné zapracovat veškeré změny do specifikace a další dokumentace, která je součástí projektu (Royce, 1970).

Rigorózní metodiky jsou považovány za tzv. "těžké" - jsou velmi podrobné, obsahují hodně formalit a jsou specifické direktivním řízením. Procesy jsou opakovatelné a předpokládají definování všech požadavků na řešení předem (Jonák, 2015).

Mezi další nevýhody patří prakticky nulová možnost reagovat v průběhu vývojového cyklu na změnové požadavky zákazníka. Finální verze aplikace je zákazníkovi předána až v době, kdy se již nelze vrátit do fází oprav a úprav. Navíc, jak vyplývá z předchozího odstavce, rigorózní metodiky se změnovými požadavky nepočítají, z čehož plyne, že produkt je dodán v takové podobě, jaká odpovídá dříve definovaným požadavkům. Tato situace může vést k neúspěchu celého projektu. Z pohledu testování je pak velmi nešťastné, když fáze testování nastává až po samotné implementaci, tedy ve chvíli kdy je aplikace téměř připravena na předání zákazníkovi. Opravy chyb v této fázi jsou časově mnohem náročnější, než např. ve fázi návrhu. Pokud se v tomto kroku objeví chyba v analýze, může vyústit až v kompletní přepracování projektu (Royce, 1970).

Do rigorózních metodik můžeme zařadit kromě vodopádového ještě spirální model, Unified Process, Rational Unified Process a Enterprise Unified Process (TechTerms, 2016).

Spirálový model vychází z modelu vodopádového, ale přináší dvě zcela nové a zásadní vlastnosti. Iterativní přístup a podrobnou analýzu rizik. Proto je také někdy řazen do skupiny přístupů řízených riziky (risk-driven approach). Iterativní přístup byl důsledkem toho, že u spousty projektů se nepodařilo stanovit přesnou a úplnou specifikaci požadavků, což činilo vodopádový model takřka nepoužitelným. Ukázalo se, že bude mnohem lepší stanovit na počátku jen rámec architektury a funkčnosti celého systému a ten postupně rozpracovávat do detailů. Cyklické opakování jednotlivých kroků vývoje, tzv. iterace, znamenalo ve své době přelom v chápání životního cyklu.

Rational Unified Process (dále jen RUP), jenž je komerčním produktem firmy Rational, která byla později koupena firmou IBM, je rozsáhlá a detailně propracovaná metodika vývoje software. Celá metodika je objektově orientovaná a patří do skupiny

přístupů řízených případy užití (use-case-driven approach). Vývoj podle RUP probíhá v iterativních cyklech. První cyklus se nazývá úvodní vývojový cyklus a jeho výsledkem je funkční softwarový produkt implementující nejpodstatnější část funkcionality. Zde ovšem vývoj nekončí, ale pokračuje v různém množství rozvíjejících cyklů. Jejich přesný počet závisí na rozsahu projektu.

Každý cyklus se skládá ze čtyř fází (v závorce je uvedeno typické rozdělení doby pro jednotlivé fáze úvodního vývojového cyklu) (TechTerms, 2016):

- Zahájení (10%)
- Projektování (30%)
- Realizace (50%)
- Předání (10%)

Enterprise Unified Process (EUP) je rozšířením Rational Unified Process a byl vyvinut Scott W. Ambler a Larry Constantine v roce 2000 a do současné podoby přepracován v roce 2005 Scottem Amblerem, Johnem Nalbonem a Michaellem Vizdosem. EUP byl původně představen za účelem odstranění některých nedostatků RUP (Michl, 2012).

V důsledku zmíněných nedostatků rigorózních metodik se hledaly nové možnosti a přístupy, jak vyvíjet software. Zákazníci jsou stále častěji tlačeni k neustálým změnám v jejich obchodních modelech. Globalizace znamená více příležitostí, ale také větší nároky na obě strany obchodního styku. Z těchto důvodů se na přelomu 20. a 21. století začalo mluvit o agilních metodikách (Jonák, 2015).

2 Agilní metodiky

Agilní metodiky kladou (na rozdíl od rigorózních) velký důraz na zákazníka. Software musí dodávat hodnotu pro zákazníka, a nikdo jiný, než zákazník sám nemůže tuto hodnotu určit. Agilní metodiky představují velkou změnu paradigmatu - nejedná se o proces, ale spíše o přístup k vývoji software (Jonák, 2015). Agilní metodiky jsou postaveny na těchto dvanácti principech (Beck, a další, 2001):

- Hlavní prioritou je vyhovět zákazníkovi za pomoci včasných a průběžných dodávek softwaru.
- Jsou vítány změny v požadavcích a to i v pozdějších fázích vývoje. Agilní procesy podporují změny, které by mohly vést ke zvýšení konkurenceschopnosti zákazníka.
- Fungující software je dodáván v pravidelných, krátkých časových úsecích několika týdnů, měsíců, přičemž preference na kratší časový úsek.
- Lidé z byznysu a vývojáři musí pracovat spolu každý den v průběhu celého projektu.
- Projekty jsou budovány motivovanými jedinci. Je jim poskytnuto takové pracovní prostředí a podpora, jakou potřebují. Zároveň je do nich kladena důvěra, za odvedení kvalitní práce.
- Nejúčinnější a nejefektivnější sdělování informací při vývoji je formou osobní komunikace.
- Fungující software je hlavním měřítkem pokroku.
- Agilní procesy podporují udržitelný vývoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržovat spolu krok po neomezenou dobu.
- Trvalá pozornost k technické dokonalosti a dobrý design zlepšuje agilitu.
- Jednoduchost – umění maximalizovat množství práce, která nemusí být provedena – je základem.
- Nejlepší architektury, požadavky a návrhy vznikají v samostatně se organizujících týmech.
- Tým pravidelně uvažuje nad tím, jak být efektivnějším, na základě toho vylepší své postupy a přizpůsobí i své chování.

Rigorózní metodiky totiž předpokládají, že zákazník vlastně neví, co chce. Proto je na dodavateli, aby vytvořil takový produkt, který by zákazník mohl v budoucnu potřebovat a který by mu byl užitečný. Hlavní myšlenkou tohoto přístupu je vytvoření systému tolerantního ke změnám a případné změny akceptovat v budoucnu.

Agilní metodiky také upřednostňují neustálou interakci a kooperaci v rámci týmu, před "standardizací" lidí v organizaci a jejich neustálou kontrolou. Jsou otevřenější změnám, více flexibilní a nesledují tolik plán (Jonák, 2015).

3 Porovnání rigorózních a agilních metodik

Předchozí kapitoly charakterizovali dvě rozdílné skupiny metodik pro softwarový vývoj. Jednalo se především o teorii. Následující část bude zaměřena na praktické uplatnění těchto přístupů a na to, jaké konkrétní metodiky jsou v současnosti používány v IT firmách (Jonák, 2015).

Tabulka 1 – srovnání rigorózních a agilních metodik (Popelka, 2009)

	Rigorózní metodiky	Agilní metodiky
Proces vývoje software	Přesně a podrobně definovaný proces, lze ho opakovat.	Empirický proces, není možné ho opakovat, ale vyžaduje adaptaci.
Změny a požadavky	Minimalizace změn (sběr požadavků předem a plánování předem), změny jsou součástí řízení změn.	Akceptace změn (přírůstkové shromažďování požadavků, plánování pro iteraci) přehodnocení požadavků, snaha o umožnění změn (v závislosti na nových znalostech).
Zapojení zákazníka nařízení projektu	Bez zásahu. Nedůvěra v zákazníka - po podpisu smlouvy a vymezení dokumentu specifikace požadavků v počáteční fázi. Zákazníka zajímá až konečné řešení	Participace a spolupráce - zákazník se aktivně zapojuje do celého projektu, může měnit priority funkcí při každé iteraci.
Kvalita	Více zaměřeny na kvalitu vlastních procesů, než na výsledek pro zákazníka.	Zaměření na priority zákazníka, na užitnou hodnotu pro zákazníka, tedy na výslednou kvalitu produktu.

Způsob a rozsah řešení	Podrobný popis procesů a činností, složité řešení. Při vývoji obsaženy všechny funkce, snaha o zabudování budoucích požadavků	Eliminace nepotřebných činností, jednoduché řešení. Při vývoji zahrnuty jen potřebné funkce, snaha o minimalizaci, žádné začlenění budoucích požadavků
Rozložení teamového know-how	Každý člen týmu má přiřazeny činnosti, v rámci role, kterou zastává. Požadavek na specializaci zaměstnanců. Převládá písemná forma komunikace (dokumentace).	Sdílení znalostí v týmu, spolupráce (kooperace) a aktivní komunikace v rámci týmu, společné řešení problémů, řízení a integrace znalostních toků
Lidský faktor	Pohled na lidi jako na sekundární faktor, procesy doplňovány rozsáhlou dokumentací. Jedinci jsou nahraditelní.	Lidé jako primární a klíčový faktor úspěchu projektu. Využití schopností a dovedností jedinců (individualit), důraz na znalosti, kvalifikaci.
Způsob vývoje	Vodopádový životní cyklus, iterativní s dlouhými iteracemi.	Přírůstkový (inkrementální) vývoj s velmi krátkými iteracemi.

V následující tabulce, rovněž porovnávající rigorózní a agilní metodiky, je vybráno sedm základních rozdílů, které mezi těmito dvěma skupinami jsou.

Tabulka 2 – Sedm základních rozdílů mezi rigorózními a agilními metodikami (Buchalceová, 2009)

Posuzované hledisko	Rigorózní metodiky	Agilní metodiky
Změny	Snaha minimalizace	Jsou vítány, možnost přehodnocování

		požadavků
Rozsah řešení	Snaha zabudování budoucích požadavků	Pouze požadované funkce, požadavek minimalizace
Způsob řízení	Na základě nedůvěry, direktivní řízení a kontroly	Vůdcovství a spolupráce, je formováno na důvěře a respektu
Forma komunikace	Převážně písemná	Důraz na komunikaci tváří v tvář
Dokumentace	Rozsáhlá dokumentace	Podstatná není dokumentace, ale pochopení
Způsob vývoje	Píše vodopádový, případně iterativní, přírůstkový s dlouhými iteracemi	Přírůstkový vývoj s velmi krátkými iteracemi
Zákazník	Podílí se na projektu jeho v začátcích a na konci; zaměření vývoje je orientováno více na jednotlivé procesy, než na výsledky pro zákazníka	Přírůstkový (inkrementální) vývoj s velmi krátkými iteracemi.

Poslední State of Agile je z roku 2013. Průzkumu se zúčastnilo přes 3 500 respondentů z řad projektových manažerů, scrum masterů, a dalších pozic, především z velkých společností (od 100 do 4000 zaměstnanců). Většinu účastníků tvořili firmy z USA (66%) a z Evropy (20%).

Ačkoliv téměř 90% všech dotázaných má určitou zkušenost a agilním přístupem, jen polovina používá k řízení svých projektů některé ze skupiny agilních metodik. Zajímavé je ale zjištění, že v případě dobré zkušenosti s tímto přístupem, je agilní vývoj prosazován na velkém množství projektů.

Z konkrétních agilních metodik žebříčku jednoznačně dominuje Scrum (55%), následovaný kombinací Scrumu a XP (extrémní programování, 11%). První příčky uzavírá Kanban (5%). Na opačném konci je DSDM (Dynamic systems development method), Agile Modeling či AgileUp (1%).

Díky velkému počtu zastánců agilního přístupu, by se mohlo zdát, že tyto metodiky budou v dnešní době oblasti softwarového vývoje, respektive řízení projektů obecně, dominovat. Jak je ale patrné z výsledků "State of Agile", jejich zastoupení je porovnatelné s ostatními přístupy. Ačkoliv to nevypadá na velký úspěch, opak je pravdou. Toto poloviční zastoupení si agilní metodiky vybudovaly jen za poslední necelá dvě desetiletí. V roce 2001 spatřil světlo světa „Manifest agilního programování“, který znamenal de facto změnu paradigmatu vývoje software.

Agilní přístup je v dnešní době na vzestupu. Svědčí o tom například prohlášení vysokých činitelů v rámci NATO na konferenci ITEC 2014, kteří zde zmínili nutnost prosazení agilní cesty v rámci vojenských IT projektů.

Není těžké také nalézt spoustu publikací o prosazujících agilní metodiky v rámci jednotlivých armád - např. "Adapting SCRUM to the Italian Army: Methods and (Open) Tools" od Angelo Messina, či prohlášení ministerstva obrany USA o přechodu agilní vývoj.

Je ovšem nutné si uvědomit, že kdo prosazuje jednu jedinou metodiku ve své základní podobě a nepřipouští jakoukoliv změnu, tak nepochopil smysl metodického přístupu jako celku. Jak už bylo ostatně zmíněno výše v tomto článku, žádný přístup nevznikl sám o sobě ve „vakuu“. Vždy proto obsahuje i některé prvky z jiných oblastí, které pokud možno vhodně kombinuje a doplňuje. Proto je nutné uvažovat nad adaptací určité metodiky jako nad neustálým vývojovým procesem a přizpůsobit tomu své očekávání.

Agilní vývoj (nebo jakýkoliv jiný) nikdy nebude považován za jeden jediný, či univerzální přístup. Každý projekt je závislý na mnoha vstupních faktorech a je z tohoto důvodu specifický. Těmito faktory může být velikost firmy, týmu, typ projektu, ale také kulturní prostředí, mezilidské vztahy, typologie osobností, apod. Vždy je nutné počítat s tím, že vedle sebe bude existovat celá řada přístupů, které budou vhodné vždy v určitém prostředí a pro konkrétní typ projektu (Jonák, 2015).

Následující kapitoly popisují jednotlivé typy mobilních aplikací, se kterými se lze v současnosti setkat.

4 Nativní aplikace

Nativní aplikace jsou aplikace vytvářené pro konkrétní platformu (např. Android⁹, iOS¹⁰) a hardwarovou konfiguraci. Jsou rychlé, spolehlivé a umí lépe využívat hardwarových schopností telefonu (např. fotoaparát, kalendář, GPS¹¹). Velkým omezením dneška je malá výdrž baterie, nativní aplikace se přitom dají naprogramovat tak, aby ji více šetřily. Menší energetická náročnost může být dána například tím, pokud může aplikace fungovat off-line (Michl, 2012).

Nativní mobilní aplikace mají své výhody i nevýhody. Určitě je velmi nepraktické, že naprogramovaná aplikace funguje pouze na jedné platformě. Pokud tedy chceme, aby byla kompatibilní se všemi nejpožívanějšími mobilními platformami, musíme napsat pro každou zvlášť kód. Výhodou je, že dané aplikace je odladěná pro konkrétní platformu, především co se týče plynulejšího a stabilnějšího chodu a plného přístupu k nativním vlastnostem mobilního zařízení (PixelField, 2012).

Další výhodou nativních mobilních aplikací je jejich propagace. Pokud je vaše aplikace dobrá, v App Store¹² se neztratí, Apple¹³ na ni upozorní. Podobně může pomoci i Google Play¹⁴ aplikacím pro Android. Navíc jsou lidé (více tedy ti s iPhone) srozumění s tím, že za aplikace se někdy musí zaplatit. Pokud se tak má stát, budou mít lidé rozhodně menší zábrany tak učinit na místě, které považují za bezpečné a kde to udělali již v minulosti (Michl, 2012).

⁹ Android je mobilní operační systém založený na jádře Linuxu, který je dostupný jako otevřený software (open source).

¹⁰ iOS je mobilní operační systém vytvořený společností Apple Inc.

¹¹ Global Positioning System, česky Globální polohovací systém, zkráceně GPS, je vojenský globální družicový polohový systém provozovaný Ministerstvem obrany Spojených států amerických

¹² App Store je obchod s aplikacemi pro zařízení s operačním systémem iOS a Mac OS X. Je provozován firmou Apple, která ho zprovoznila 10. července 2008.

¹³ Apple Inc., dříve Apple Computer Inc., je americká firma se sídlem ve městě Cupertino v Kalifornii.

¹⁴ Google Play je online distribuční služba, která vznikla 6. března 2012 (Kasík, 2015) spojením služeb Google Music a Android Market.

5 Webové aplikace

Mobilní web je nejuniverzálnější formou, která je funkční napříč platformami. Jedná se o podobu stránek, jež by měla být přizpůsobena omezením mobilního prohlížeče i malých obrazovek telefonů. Uživatelská přívětivost je pochopitelně závislá na rychlosti připojení, která se liší dle lokality, poskytovatele, druhu datového tarifu atd.

U mobilního webu a dalších webových variant aplikací se musíte o marketing postarat sami. Pokud lidé chtějí vaše služby či internetovou stránku používat i z obrazovek mobilních telefonů sami od sebe, vyťukat vaše jméno do vyhledávače či URL políčka je napadne jako první. Další důležitou informací je, jak často vaši službu využijí. Například úvěry obecně lidé neuzavírají každý den a aplikaci pro jejich zařízení si nainstalovali jen chroničtí dlužníci. Podobně jsou potřeba ty nejnižší bariéry vstupu u mobilních verzí e-shopů. Nativní aplikaci si pak mohou dovolit velké e-shopy samy promovat jako doplněk pro zákazníky, jež jsou součástí věrnostního programu (Michl, 2012).

6 Často používané metodiky

V rámci této práce jsme provedli i průzkum dostupných metodik pro vývoj mobilních aplikací. Nejzajímavější z nich byla diplomová práce Bc. Petra Šolína na téma „Návrh metodiky UX designu pro mobilní aplikace“. Další často používanou metodikou je Mobile-D. Obě zmíněné metodiky budou popsány v následující části.

6.1 Metodika pro vývoj UX mobilní aplikace

Zmíněná diplomová práce si kladla za cíl navrhnout metodiku pro návrh UX designu, tedy tzv. uživatelského prožitku z daného produktu. Autor rozdělil celý proces návrh do 6 fází – výzkum trhu, koncept, design, vývoj aplikace, implementace a

uvolnění aplikace. Každé fáze se potom dělí na dimenze – aktivita, vstup, činnost, výstup a role.

Výzkum trhu

Samotný výzkum trhu se sestává ze dvou dílčích fází – definice obchodních požadavků a uživatelský průzkum.

Hlavními aktivitami při definici obchodních požadavků jsou rozhovory se zainteresovanými stranami a analýza konkurence. Při těchto činnostech se zjistí klientské cíle a požadavky, konkurenceschopnost aplikace či silné a slabé stránky konkurenčních aplikací. Hlavními vstupy jsou požadavky klienta a data a informace konkurenčních projektů, výstupy potom seznam cílů a termínů projektu a silné a slabé stránky konkurentů.

Uživatelský průzkum potom probíhá pomocí rozhovorů s uživateli a skupinovou moderovanou diskuzí. S pomocí odpovědí jednotlivých uživatelů se zformují cíle, zkušenosti a myšlenky jednotlivých uživatelů vztahující se k danému projektu.

Koncept

V další fázi se připravuje koncept dané aplikace. Ten se skládá ze tří základních částí: Persony, uživatelský postup a třídění karet.

V dílčí fázi persony se vytvářejí fiktivní profily uživatele představující cílovou skupinu uživatelů produktu. Uživatelský postup představuje scénáře toho, jak bude uživatel procházet a ovládat danou aplikaci. Tato část také obsahuje grafické znázornění ovládání aplikace. V poslední fázi se potom vytváří informační architektura aplikace s pomocí třídění jednotlivých funkcí do kategorií.

Design

Ve třetí fázi dochází k vlastnímu designu. Ten probíhá v pěti dílčích procesech – drátěný model (wireframe), prototyp, test prototypu, mockup a test mockup.

V designové fázi probíhá postupná tvorba modelu aplikace. Nejdříve se vytvoří drátěný model, tedy pouze rozložené prvky budoucí aplikace. V další fázi se vytvoří první dynamická verze aplikace – prototyp – a nechá se otestovat uživateli. Z jejich

připomínek se následně upraví model a vytvoří se tzv. mockup prototyp. V tomto modelu je již nasazený budoucí grafický design aplikace. Ten se poté testuje tak dlouho, dokud nesplňuje požadované podmínky.

Vývoj aplikace

Po navržení aplikace přichází na řadu její vývoj. Ten se skládá ze samotného vývoje aplikace a jejího otestování dle předpřipravených scénářů. Jako podklady slouží modely připravené v minulé fázi. Na této fázi se podílí různé role včetně informačního architekta, klienta, vývojářského týmu, projektového manažera a další.

Implementace

Ve fázi implementace dochází k zavádění aplikace do mobilního zařízení a integraci ostrých dat. Funkcionalita aplikace se podrobí závěrečnému testování.

Uvolnění aplikace

V poslední fázi probíhá samotné uvolnění aplikace k jejímu použití. V této fázi také probíhá podpora a případné školení uživatelů pro danou aplikaci.

V diplomové práci autor následně ukazuje velice praktickou ukázkou fiktivního projektu spolu s užitím dané metodiky. Metodika se zdá být velice praktická a jistě by našla své uplatnění nejen při vývoji mobilních aplikací (Šolín, 2012).

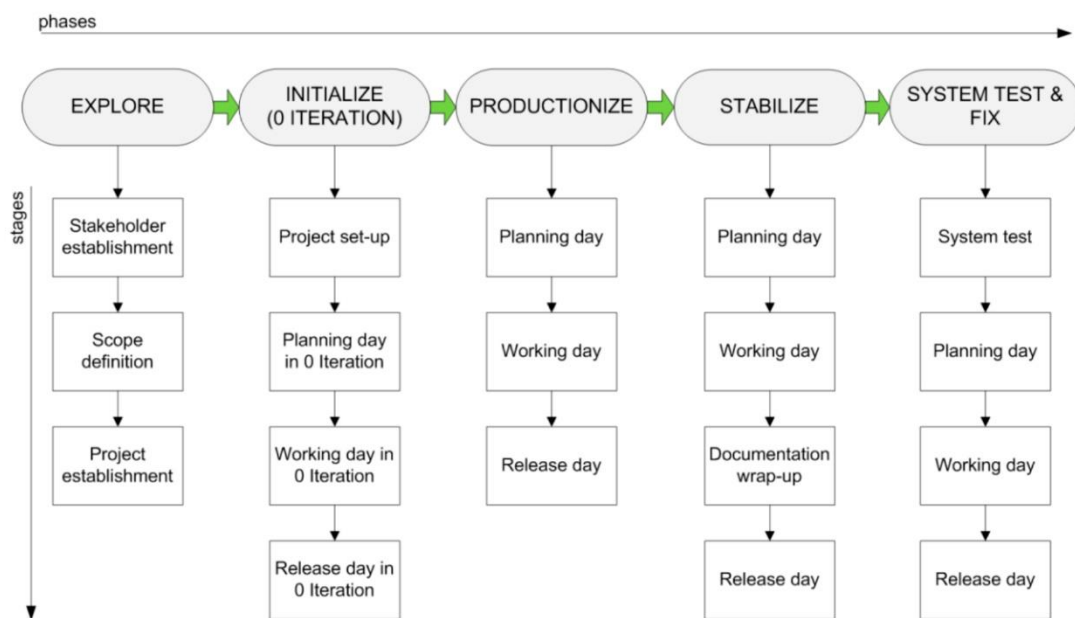
6.2 Mobile-D

Metodika Mobile-D je určena především pro malé týmy do deseti lidí. Celý vývoj by pak měl trvat deset týdnů. Metodika je založena především na aktivní spolupráci a komunikaci jak mezi členy týmu, tak se zákazníkem a skládá se z devíti hlavních prvků, uplatňujících se v průběhu vývojového cyklu:

1. Phasing and Placing
2. Architecture Line
3. Mobile Test-Driven Development
4. Continuous Integration
5. Pair Programming
6. Metrics

7. Agile Software Process Improvement
8. Off-Site Customer
9. User-Centered Focus

Metodika se dále skládá z pěti fází: Explore, Initialize, Productionize, Stabilize, System Test & Fix. V první fázi je sestaven vývojový tým a stanoveny všechny charakteristiky projektu (projektový plán, sběr požadavků). V další fázi dochází k ujasnění si všech požadavků a zajištění zdrojů, nutných pro úspěšné ukončení projektu. Následuje fáze implementace. V této části dochází zároveň k tvorbě akceptačních testů. Samotný vývoj pak probíhá podle metodiky Test-Driven Development. Poslední dvě části (Stabilize a System Test & Fix) jsou určeny k dokončení produktu a jeho testování. Detailní rozbor jednotlivých částí je zobrazen na obrázku 4 (Spataru, 2010):



Obrázek 4 – fáze metodiky Mobile-D

7 Závěr

Ve své první části se tato práce věnuje rychlému nárůstu chytrých mobilních zařízení a jeho důsledkům. Tím nejdůležitějším je především adekvátní růst množství nových mobilních aplikací, určených pro tato zařízení. Jsou zde popsána jejich specifika a především to, čím se liší od běžných desktopových aplikací. Z těchto specifík vyplývají i odlišnosti v jejich vývoji. Tím největším rozdílem je především obrovská dynamika trhu, odlišné požadavky na ergonomii, velmi rychlý vývoj mobilních operačních systémů a značná diverzifikace mobilních zařízení.

Poté se práce dostává k další části, ve které jsou popsány dvě hlavní skupiny metodik, rigorózní a agilní. Následuje jejich srovnání, z čehož práce vyvozuje, že pro vývoj mobilních aplikací jsou vhodné metodiky agilní. To je dáno především již zmíněnou značnou dynamikou tohoto trhu. Naopak rigorózní metodiky nejsou použitelné v podstatě vůbec, neboť jsou příliš rigidní a tudíž je nelze aplikovat na takto flexibilní oblast. Zároveň však práce zdůrazňuje, že každou metodiku je třeba přizpůsobit konkrétním požadavkům, neboť vývoj jakékoliv mobilní aplikace (a aplikace obecně) je specifický proces, ovlivněn vždy mnoha faktory. V poslední části jsou popsány dvě konkrétní používané metodiky.

8 Citovaná literatura

Beck, Kent a kol. 2001. Principles behind the Agile Manifesto. [Online] 2001. [Citace: 14. 05 2016.] <http://agilemanifesto.org/principles.html>.

Buchalceková, Alena. 2009. *Metodiky budování informačních systémů. 1. vydání.* Praha : Oeconomica, 2009. str. 206. ISBN 978-80-245-1540-3.

Evans, Benedict. 2013. Mobile is eating the world, autumn 2013 edition. [Online] 2013. [Citace: 14. 05 2016.] <http://ben-evans.com/benedictevans/2013/11/5/mobile-is-eating-the-world-autumn-2013-edition>.

Google. 2016. Android N Developer Preview. [Online] 2016. [Citace: 15. 05 2016.] <http://developer.android.com/preview/index.html>.

—. 2016. Web Apps. [Online] 2016. [Citace: 14. 05 2016.] <http://developer.android.com/guide/webapps/index.html>.

Hajdin, Tomáš. 2005. *Agilní metodiky vývoje software.* Brno : Masarykova univerzita v Brně, Fakulta informatiky, 2005.

Intel. 2015. 50 Years of Moore's Law. [Online] 2015. [Citace: 13. 05 2016.] <http://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>.

Jonák, Stanislav. 2015. Vývojový cyklus software: Dominují agilní metodiky trhu? [Online] 2015. [Citace: 15. 05 2016.] <http://www.middleware.cz/projektovizeni0/24-pristupy-k-vyvoji-software-dominuje-agilni-vyvoj-trhu>.

Kasík, Pavel. 2015. „Výkon počítačů roste už 50 let podle mého zákona!“ diví se Gordon Moore. [Online] 2015. [Citace: 14. 05 2016.] http://technet.idnes.cz/mooruv-zakon-gordon-moore-intel-rozhovor-fgh-/tec_technika.aspx?c=A150416_232158_tec_technika_pka.

libGDX. 2013. libGDX - Desktop/Android/BlackBerry/iOS/HTML5 Java game development framework. [Online] 2013. [Citace: 13. 05 2016.] <https://libgdx.badlogicgames.com/>.

Michl, Petr. 2012. Je lepší nativní aplikace nebo mobilní web? [Online] 2012. [Citace: 15. 05 2016.] http://www.m-journal.cz/cs/internet/je-lepsi-nativni-aplikace-nebo-mobilni-web-__s281x9241.html.

Oxford Dictionaries. 2016. Smartphone. [Online] 2016. [Citace: 14. 05 2016.] <http://www.oxforddictionaries.com/definition/english/smartphone>.

PixelField. 2012. Hybridní aplikace. [Online] 2012. [Citace: 13. 05 2016.] <http://www.vyvojmobilniaplikace.cz/hybridni-aplikace/>.

Popelka, Vladimír. 2009. *Srovnávací analýza metodik vývoje software*. Praha : Vysoká škola ekonomická v Praze, 2009.

Royce, Winston. 1970. Managing the development of large software systems. [Online] 1970. [Citace: 15. 05 2016.] <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>.

Sawers, Paul. 2015. There are 18,796 distinct Android devices, according to OpenSignal's latest fragmentation report. [Online] 2015. [Citace: 13. 05 2016.] <http://thenextweb.com/mobile/2014/08/21/18796-different-android-devices-according-opensignals-latest-fragmentation-report/#gref>.

Šolín, Petr. 2012. *Návrh metodiky UX Designu pro mobilní aplikace*. Praha : Vysoká škola ekonomická v Praze, 2012.

Spataru, Andrei Cristian. 2010. *Agile Development Methods for Mobile Applications*. Edinburgh : University of Edinburgh, 2010.

TechTerms. 2016. EUP. [Online] 2016. [Citace: 15. 05 2016.] <http://techterms.com/definition/eup>.

Unity3D. 2016. Unity - Game Engine. [Online] 2016. [Citace: 15. 05 2016.] <https://unity3d.com/>.