

DIPLOMOVÁ PRÁCE

Použití agilních metodik SCRUM a Kanban při vývoji mobilních aplikací

The Use of SCRUM and Kanban Agile Methodologies in Developing Mobile Application

STUDIJNÍ PROGRAM

Projektové řízení inovací

VEDOUCÍ PRÁCE

doc. Ing. Lenka Švecová, Ph.D.

VOJTOVÁ

KLÁRA

2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	Klára	Jméno:	Vořtová	Osobní číslo:	461044
Fakulta/ústav:	Masarykův ústav vyšších studií (MÚVS)				
Zadávací katedra/ústav:	Oddělení manažerských studií				
Studijní program:	Projektové řízení inovací				
Studijní obor:	-				

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:
Použití agilních metodik SCRUM a Kanban při vývoji mobilní aplikace

Název diplomové práce anglicky:
The Use of SCRUM and Kanban Agile Methodologies in Developing Mobile Application

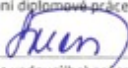
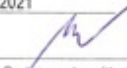

Pokyny pro vypracování:
CÍL PRÁCE: Cílem DP je propojení dvou agilních metod při vývoji mobilní aplikace.
PRÍNOS PRÁCE: Přínosem DP je zefektivnění vývoje aplikace propojením dvou agilních metod.
OSNOVA: (1) Úvod; (2) Projekt a projektové řízení; (3) Agilní přístupy k projektovému řízení; (4) Aplikace agilních metodik při vývoji; (5) Závěr

Seznam doporučené literatury:
(1) WYSOCKI, Robert. Effective project management: traditional, agile,extreme. 8th edition. John Wiley, 2019.
(2) DOLEŽAL, Jan. Projektový management: komplexně, prakticky a podle světových standardů. Praha: Grada Publishing, 2016
(3) PICHLER, Roman. Agile product management with Scrum: creating products that customers love. Addison-Wesley, 2010.
(4) KNIBERG, Henrik a Mattias SKARIN. Kanban and SCRUM: making the most from both. C4Media, 2010.

Jméno a pracoviště vedoucí(ho) diplomové práce:
doc. Ing. Lenka Švecová, Ph.D., ČVUT v Praze, Masarykův ústav vyšších studií


Jméno a pracoviště konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: 12. 12. 2019 Termín odevzdání diplomové práce: 30. 4. 2020 / 20. 8. 2020
Platnost zadání diplomové práce: 30. 9. 2021

 Podpis vedoucí(ho) práce
 Podpis vedoucí(ho) ústavu/katedry
 Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

24. 4. 2020
Datum převzetí zadání


Podpis studenta(ky)

VOJTOVÁ, Klára. Použití agilních metodik SCRUM a Kanban při vývoji mobilních aplikací.
Praha: ČVUT 2020. Diplomová práce. České vysoké učení technické v Praze, Masarykův
ústav vyšších studií



**MASARYKŮV ÚSTAV
VYŠŠÍCH STUDIÍ
ČVUT V PRAZE**

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracovala samostatně. Dále prohlašuji, že jsem všechny použité zdroje správně a úplně citovala a uvádím je v příloženém seznamu použité literatury.

Nemám závažný důvod proti zpřístupňování této závěrečné práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Praze dne: 14. 05. 2020

Podpis:

Poděkování

Chtěla bych poděkovat své vedoucí diplomové práce doc. Ing. Lence Švecové, Ph.D. za cenné rady, věcné připomínky a vstřícnost při zpracovávání této práce.

Abstrakt

Tato diplomová práce se zabývá zefektivněním vývoje mobilní aplikace za použití kombinace agilních metodik SCRUM a Kanban. Teoretická část obsahuje dvě hlavní kapitoly, které se zabývají pojmy projekt a projektové řízení a přístupy k projektovému řízení. V praktické části je představen vývoj konkrétní mobilní Aplikace pomocí kombinace agilních metodik SCRUM a Kanban, doplněný o přínosy nové metodiky a návrhy na zlepšení.

Klíčová slova

Projekt, projektové řízení, agilní metody, SCRUM, Kanban, mobilní aplikace, IT vývoj

Abstract

This diploma thesis is focused on streamlining the development of mobile applications using a combination of agile SCRUM and Kanban methodologies. The theoretical part contains two main chapters that deal with the concepts of project and project management and approaches to project management. The practical part presents the development of a specific mobile application using a combination of agile methodologies SCRUM and Kanban, supplemented by the benefits of the new methodology and suggestions for improvement.

Key words

Project, project management, agile methods, SCRUM, Kanban, mobile applications, IT development

Obsah

Úvod	5
1 Projekt a projektové řízení	7
1.1 Projektové řízení	7
1.1.1 Historie projektového řízení	7
1.2 Projekt.....	9
1.2.1 Definice projektu.....	9
1.2.2 Životní cyklus projektu	11
2 Přístupy k projektovému řízení	13
2.1 Agilní přístupy k projektovému řízení	14
2.1.1 SCRUM	15
2.1.2 Kanban.....	23
3 Aplikace agilních metodik při vývoji	27
3.1 Společnost XY	27
3.1.1 Vývojové oddělení.....	27
3.1.2 Běžný průběh vývoje mobilní aplikace pomocí SCRUM metodiky.....	28
3.1.3 Nástroje používané během vývoje mobilní aplikace pomocí SCRUM metodiky ve Společnosti XY	32
3.2 Aplikace XY.....	34
3.2.1 Funkcionality Aplikace XY.....	34
3.3 Kombinace SCRUMU a Kanbanu při vývoji mobilní Aplikace XY.....	35
3.3.1 Začátek vývoje pomocí metodiky SCRUM.....	36
3.3.2 Přejít na řízení vývoje Aplikace XY pomocí kombinace metodik SCRUM a Kanban	38
3.4 Přínosy metodologie Scrumban a návrhy na zlepšení	42
3.4.1 Zefektivnění vývoje	42
3.4.2 Tým.....	45
Závěr	47
Seznam použité literatury	49
Seznam obrázků	52

Úvod

Projekt a projektové řízení jsou sice poměrně mladými pojmy, ve většině druhů podniků se však používají velmi frekventovaně. Neodborná veřejnost jistě zná tradiční metody k projektovému řízení, jakými jdou vodopádové metody. Tyto metody byly velmi populární v 90. letech 20. století, kdy se každá větší společnost pyšnila pozicí Projektového manažera. Od přelomu tisíciletí se však obzvláště IT společnosti začaly zajímat o nový styl řízení, kterým je takzvaný agilní přístup. Ten se od vodopádových metod liší hlavně možností rychlejší reakce na změnu a upozadováním složitých technických specifikací. Stejně jako vodopádový přístup, i agilní přístup je nadřazeným pojmem pro skupinu metodik, které spojují agilní principy. Tyto metodiky se od sebe jinak často liší a záleží na každé společnosti, případně týmu, kterou z metodik pro svůj projekt či produkt zvolí.

Teoretická část této diplomové práce se nejprve zaměřuje na definici a historii pojmů projekt a projektové řízení obecně. V druhé kapitole teoretické části jsou pak představeny dva hlavní přístupy k projektovému řízení – vodopádový a agilní. Tato kapitola se zaměřuje na rozdíly mezi přístupy a prostřednictvím srovnání poukazuje na to, k jakému typu projektu je jaký přístup vhodnější. Jelikož je praktická část zaměřena na agilní metody projektového řízení, je jim ve druhé kapitole praktické části věnováno více pozornosti. Po obecné charakteristice agilního přístupu jsou dvě agilní metodiky představeny podrobněji. Jedná se o SCRUM a Kanban.

V praktické části této práce je pak představen a popsán průběh vývoje mobilní aplikace ve Společnosti XY při použití SCRUMU. V druhé části je pak popsán průběh vývoje konkrétní mobilní Aplikace XY, jejíž vývoj začal pomocí metodiky řízení SCRUM, nicméně po dvou měsících se tým rozhodl pro novou metodiku řízení, a to Scrumban, která je kombinací právě SCRUMU a Kanbanu. Na tomto projektu jsem se osobně podílela na pozici juniorního Product Ownera. Poslední část praktické části této práce obsahuje přínosy kombinace metodik SCRUM a Kanban a také návrhy na zefektivnění výsledné metody. Přínosy a návrhy na zlepšení jsou rozděleny do dvou kategorií – zefektivnění vývoje a tým.

TEORETICKÁ ČÁST

1 Projekt a projektové řízení

První kapitola této diplomové práce bude zaměřena na projekt a projektové řízení z různých pohledů. Bude vysvětleno, jak jsou tyto pojmy vnímány v dnešní době a jejich definice. Dále bude uvedena stručná historie vývoje projektového řízení do té podoby, ve kterém ho známe dnes. Definice projektového řízení bez vymezení pojmu projekt by nebylo úplné, proto mu bude také věnována kapitola. Na závěr budou představeny dva hlavní přístupy k projektovému řízení, kterými jsou tradiční a agilní a zároveň bude každý z těchto směrů více přiblížen. Jelikož praktická část této práce bude zaměřena na agilní metody projektového řízení, je těmto metodám věnováno více pozornosti.

1.1 Projektové řízení

Projektové řízení je obor, který je v porovnání s ostatními poměrně mladý. Pojmy projekt a projektové řízení jsou v dnešním slova smyslu vázány až na období po 2. světové válce. To neznamená, že do té doby lidstvo neznalo žádné postupy a úkoly, pouze to nenazývali projektem a projektovým řízením. (Doležal, Máchal, Lacko a kolektiv, 2012, s. 22)

Projektové řízení vzniklo jako odpověď na problém řízení převážně na základě zkušeností a intuice manažerů. Tento způsob řízení nebyl dostatečně efektivní při řízení vysoké míry změn, a proto bylo potřeba řízení pojmout z projektového pohledu. (Hrazdilová Bočková, 2016, s. 16) Dle Locka je úkolem projektové řízení předpovědět co nejvíce hrozeb a překážek a zároveň naplánovat a kontrolovat kroky, které tyto hrozby a překážky co nejvíce eliminují. (Lock, 2007, s. 1) Toto tvrzení mnoho autorů doplňuje o další definice. Dle Posnera je projektové řízení řízením procesu změny, tedy řízení cesty od jednoho stavu ke druhému. (Hrazdilová Bočková, 2016, s. 24). Jednu z nejvíce známých definic uvádí PMBOK Guide, což je mezinárodně uznávaná příručka projektového řízení. Tato příručka definuje projektové řízení jako: „...aplikaci vědomostí, dovedností, nástrojů a technik na aktivity projektu tak, aby byly naplněny požadavky projektu“. (2017, s. 11) Tato příručka dále definici rozšiřuje o myšlenku, že projektové řízení je prostředkem, který umožňuje organizacím řídit projekty efektivně a účinně.

1.1.1 Historie projektového řízení

Od počátku lidské existence lidé spolupracovali, aby vytvořili a řídili konstrukční projekty, jakými jsou stavby. Tyto dávné stavby dokládají důkazy o dřívějších formách jakéhosi projektového řízení. Žádná z historických staveb, jakými jsou například egyptské pyramidy nebo Colosseum v Římě, by nemohla vzniknout bez projektového řízení a projektových manažerů. (Chiu, 2010, s. 1-2) Rozdíl mezi projektovým řízením, které známe teď a tím, které probíhalo ve starověku, je určitě v rychlosti výměny informací. Ve starověku předání zprávy trvalo dny, týdny až měsíce v závislosti na

vzdálenosti, kterou musela překonat. Co naopak ve starověku nebyl problém, byly zdroje a čas. Pyramidy se stavěly desítky let, některé evropské katedrály dokonce stovky let a v průběhu jejich stavby se vystřídal několik generací stavitelů. V dnešní době jsou projekty z pohledu zdrojů i času silně omezeny. (Doležal a kolektiv, 2016, s. 14)

Projektové řízení tak, jak ho chápeme v dnešní době, bylo v 60. letech 20. století doménou pouze komplexních a velmi nákladných projektů. V 70. letech 20. století se projektové řízení začalo pomalu přesouvat i směrem k jednoduššímu podnikatelskému prostředí ale dominantou byly stále velké a komplikované projekty. Jednalo se převážně o inženýrské projekty, které svou komplexitou dodnes vynikají. (Hrazdilová Bočková, 2016, s. 21)

Svůj rozvoj zažívá projektové řízení od 80. let 20. století, kde se ujímá hlavní role ve všech podnikatelských aktivitách. Tento proces s sebou přináší potřebu přechodu od formálního projektového řízení do jeho komplexní podoby. V tomto období bylo nahlížení managementu společností na projektové řízení poměrně negativní, jelikož postupy byly zastaralé (již 50 let se používaly stejné postupy). Oproti tomu, jak bylo vnímáno projektové řízení v obchodních společnostech, stojí fakt, že v tomto období docházelo k uskutečnění projektů, které byly gigantické povahy – například let člověka do vesmíru. (Hrazdilová Bočková, 2016, s. 21 – 22)

Termín projektové řízení byl poprvé použit v roce 1953 v letectví. V tomto roce rostly obavy USA z ruských balistických mezikontinentálních raket, a proto byl velký tlak na US Air Force, aby brala vývoj raket velmi vážně. První použití pojmu projektové řízení je připisováno společnosti The Martin právě roku 1953. (Morris, 2012, s. 3)

Rozmachu projektového řízení napomohl vznik světových organizací sdružujících projektové manažery. Asi nejznámější organizace projektových manažerů vznikla v roce 1969 v USA pod názvem Project Management Institute, která položila základy novodobému projektovému řízení prostřednictvím vytvoření Project Management Body of Knowledge, což je dílo známé hlavně pod názvem PMBOK. Tento standard, který vznikl v roce 1996, je neustále rozvíjen a přizpůsobován novým trendům, proto je stále hojně využíván hlavně ve velkých mezinárodních firmách. (Doležal a kolektiv, 2016, s. 28)

Další neméně důležitou organizací, která sdružuje projektové manažery, je IPMA (*International Project Management Association*). Tato organizace vznikla v roce 1965 jako mezinárodní síť pro výměnu zkušeností s projektovým řízením. IPMA se zaměřuje na kompetence jednotlivých projektových manažerů a členů jejich týmů, díky čemuž je ponechán určitý prostor pro kreativitu jednotlivců. IPMA využívá svůj kompetenční model ICB, který rozděluje kompetence na tři oblasti: technické, behaviorální

a kontextové. Tyto oblasti jsou pak dále členěny na elementy kompetencí, které dále tyto oblasti rozvíjejí. (Doležal a kolektiv, 2012, s. 26 – 27)

V roce 1989 poté vznikla metodologie PRINCE2 jako standardní metoda pro efektivní projektové řízení pro Britské informační a komunikační technologické projekty. Nejen kvůli svému původu je tato metodologie velmi využívána ve veřejném sektoru. Tato metodologie, na rozdíl od většiny ostatních metodik projektového řízení, netvrdí, že projektový management je kompetence projektového manažera. Dle této metodologie je projektový management kompetence samotné organizace, což znamená, že za úspěch projektu jsou zodpovědní všichni stakeholdeři, nejen projektový manažer. Výhodou PRINCE2 je jistě to, že pravidla nejsou považovány za pevně daná, tím pádem můžou být upravena dle potřeb projektu. Naopak nevýhoda PRINCE2 může nastat v případě, kdy organizace nedisponuje dostatečně seniorními pracovníky, kteří jsou schopni zvládnout komplexnost této metodiky. (Molen, 2015, s. 214 – 217)

1.2 Projekt

Jako každý pojem, i pojem projekt má mnoho definicí. Níže budou představeny takové, které jsou buď dostatečně komplexní nebo nám na problematiku poskytují zajímavý náhled.

1.2.1 Definice projektu

Němec (2002, s. 11) charakterizuje projekt jako cílevědomý návrh na uskutečnění určité inovace v přesně daných termínech zahájení a ukončení. Tato definice jistě není nesprávná ale pro naše potřeby příliš obecná. Tvrdit, že každý projekt je inovací, je velmi odvážné a nemusí tomu tak vždy být. Dle Locka (2007, s. 1) je charakteristickým znakem projektu jeho "novost". Tento autor tvrdí, že každý projekt je krokem do neznáma. Tak je tomu i v případě, kdy je nějaký projekt opakován – zkrátka žádné dva projekty nejsou totožné. Tato definice nám také k úplnému pochopení nepostačí ale doplňuje definici předchozí.

Podle PMBOK (2017, s. 5) může být za projekt považováno úsilí vytvořit jakýkoliv produkt, službu nebo prostě výsledek: *„Projekty jsou zavázány splnit cíle prostřednictvím doručení výstupů. Cíl je definován jako výsledek, ke kterému má směřovat práce, strategická pozice, které má být docíleno, účel, kterého má být dosaženo, výsledek, který má být obdrženo, produkt, který má být vyroben nebo služba, která má být vykonána. Výstup je definován jako jakákoliv unikátní a prokazatelný produkt, výsledek nebo schopnost vykonávat službu, která je požadována pro dokončení procesu, fáze nebo projektu. Výstup může být jak hmatatelný, tak nehmatatelný“*. Tato definice je dle mého názoru nejvýstižnější, jelikož obsahuje

veškeré proměnné. Definice dokonce sama vysvětluje v ní použité termíny, což je pro pochopení důležité.

Doležal, Krátký a Cingl (2013, s. 9) nabízí pět znaků, dle kterých lze akci považovat za projekt. Je to taková akce, která je jedinečná, vymezená, různorodá, komplexní a riziková. Jedinečná v ohledu novosti (stejně jako uvádí Lock, 2007, s. 1), vymezená ve smyslu konkrétního ohraničení z pohledu času, financí a dalších zdrojů, různorodá z pohledu dovedností různých lidí. Komplexnost dle autorů v tomto případě znamená fakt, že řešení problému není jednoduché a nelze ho snadno vyřešit. Poslední ze znaků je rizikovost, která vyplývá z předchozích čtyř znaků – při jejich kombinaci vzniká mnoho neznámých, které mohou akci uškodit. (2013, s. 9)

A co je tedy cílem projektu? Na tuto otázku odpovídá takzvaný **trojimperativ** (= trojí podmínka), který má za dodržení a vyvážení těchto tří podmínek maximalizovat užitek projektu. Základním principem trojimperativu je provázání zmíněných tří podmínek neboli veličin, kdy pokud dojde u první ke změně a druhá zůstane stejná, musí se třetí veličina změnit odpovídajícím způsobem. (Doležal, Máchal, Lacko, 2012, s. 66) **První** podmínkou je kvalita, která se bude dle druhu projektu lišit. U projektu, jehož výstupem je fyzický výrobek, půjde o specifikaci rozměrů, množství, způsobu dodání, apod. U projektu, jehož výstupem je služba, se může jednat například o vymezení, pro koho služba je a jakým způsobem bude vymezena. **Druhou** podmínkou je čas. Důležité je určení data zahájení a dokončení projektu a potvrzení důležitých milníků. **Třetí** podmínkou jsou náklady, mezi které patří primárně dvě hlavní kategorie – finanční prostředky a lidská síla. Do první kategorie spadají hlavně položky jako například pořízení materiálu nebo nákup služeb od externích dodavatelů. Do druhé kategorie spadá hlavně interní kapacita pracovníků. (Dvořák, Mareček, 2017, s. 16)

Dalším důležitým prvkem projektu jsou jeho **zainterесované strany**, takzvaní *stakeholdéři*. Mezi stakeholdery patří kdokoliv, kdo projekt nějakým způsobem ovlivňuje nebo je projektem ovlivněn. Nejběžnějšími stakeholdery jsou zákazník nebo klient, dodavatel, který bude provádět jednotlivé úkoly (ať už vlastní silou nebo prostřednictvím subdodavatelů) a investor projektu. (Lock, 2007, s. 9 – 10) V závislosti na typu projektu se mění i stakeholdéři. K výše zmíněným může přibýt například u stavebního projektu zástupci archeologů nebo ekologů. Ve veřejném sektoru se můžeme setkat například s vládou v pozici stakeholdera. (Morris, Pinto, 2012, s. 273) Na začátku projektu je důležité tyto stakeholdery identifikovat a zároveň zmapovat jejich zájmy na projektu. Důležitým úkolem projektového manažera je dosáhnout toho, aby projekt uspokojil potřeby a představy všech stakeholderů. Pokud některá skupina stakeholderů nesouhlasí s rozhodnutími v rámci projektu, je úkolem projektového manažera buď nespokojeného stakeholdera přesvědčit o správnosti rozhodnutí nebo najít takový kompromis, aby byli všichni stakeholdéři spokojeni. Tento proces se nazývá řízení zainterесovaných stran. (Morris, Pinto, 2012, s. 278) Neděje se pouze na

počátku projektu, ale stakeholderům by měla být věnována pozornost v podobě tvorby a implementace komunikační strategie po celou dobu trvání projektu.

Z pohledu komplexity a zasazení projektu do řídicí struktury, je třeba rozlišovat samostatně stojící projekt, projekt jako součást programu a projekt jako součást portfolia. **Program** v tomto případě znamená skupinu více projektů, které mají společně dosáhnout jednoho cíle prostřednictvím svých výstupů. Jedná se o situace, kdy je výhodnější tyto projekty spojit, než je řídit odděleně. **Portfolio** je skupina programů, či programů a dalších projektů. Jedná se tedy o nadřazenou skupinu a nejkomplexnější skupinu. Tyto tři typy se tedy liší projektovým řízením, kdy ke každému typu je potřeba přistupovat rozdílně. (A guide to the project management body of knowledge, 2017, s. 13 – 14)

1.2.2 Životní cyklus projektu

Projekt lze rozčlenit do pěti fází z pohledu jeho životního cyklu (počet fází se někdy u různých autorů liší). Tyto fáze můžeme považovat za procesy, kde výstup jednoho procesu slouží jako vstup do dalšího procesu. Pojmenování a posloupnost jednotlivých procesů se může mírně lišit v závislosti na autorovi. V této práci je použito rozdělení dle Hrazdilové Bočkové, dle které jsou životní fáze projektu tyto: zahájení, plánování, realizace, monitorování a kontrola, ukončování. (2016, s. 32) doplněné o jejich specifikaci od Dvořáka a Marečka (2017 s. 18).

Fáze **zahájení** (*Initiating*) má za cíl zhodnotit, zda je projekt proveditelný a realizovatelný. V této fázi se zkoumá, zda je možné dodat to, co zadavatel projektu požaduje. Zkoumá se zároveň, zda je realizace proveditelná za daných podmínek – tedy zda rozsahu projektu odpovídají finanční a časové mantinely.

Pokud se v první fázi vyhodnotí, že projekt splňuje kritéria popsané výše, přistoupí se k druhé fázi – **plánování** (*Planning*). Tato fáze je asi nejnáročnější pro projektového manažera, který má za úkol projekt naplánovat. Při plánování se může projektový manažer řídit dle zásad trojimperativu.

Po plánování projektu přichází na řadu samotná **realizace** (*Implementing*). V této fázi přichází na řadu hlavně realizační tým projektu, který přivádí projekt k životu. Realizační fáze vyplývá z fáze plánovací, nicméně vzhledem k charakteru projektu a projektového řízení dochází často ke změnám, jež tento plán modifikují.

Monitorování a kontrola (*Controlling*) je proces, který probíhá současně s realizací. V této fázi je úkol projektového manažera jasný – sledovat naplňování plánu a sledování odchylek. Tyto odchylky můžou být použity jako podklad pro rozhodování o dalším průběhu realizace projektu. Toto rozhodování by mělo opět odpovídat plánu dle trojimperativu.

Poslední fází životního cyklu projektu je **ukončování** (*Closing*). Tato fáze může nastat jak na konci projektu, tak v jeho průběhu, kdy se jedná o předčasné ukončení. K tomu je přistoupeno většinou v momentu, kdy se projekt výrazně odchýlí od svého plánu a jeho pokračování buď není možné, nebo je o jeho konci rozhodnuto příslušným orgánem. V případě, kdy je projekt řádně dokončen, dochází v této fázi k předání projektu zadavateli projektu.

2 Přístupy k projektovému řízení

V současné době jsou známy dva hlavní přístupy k projektovému řízení – tradiční (*waterfall*) a agilní. Jelikož je praktická část této práce zaměřena na agilní metody, bude jim věnována celá samostatná kapitola. I tak je ale nutné krátce přiblížit i tradiční přístupy k projektovému řízení. Největší rozdíl mezi tradičním a agilním přístupem je přístup k případným změnám v průběhu životního cyklu projektu. Tradiční přístupy k projektovému řízení jsou vůči změnám méně tolerantní. Cílem u tradičních přístupů je projekt na začátku naplánovat a poté se při realizaci od plánu co nejméně odchylovat. V takovém případě může dojít k tomu, že se tým soustředí více na plán než na přidanou obchodní hodnotu. (Křivánek, 2019, s 111)

Dalo by se říci, že tradiční přístupy jsou vhodné u projektů, které jsou podobné těm, které byly už v minulosti uskutečněny a předpokládá se, že při jejich realizaci nedojde k žádným překvapením. Ve fázi zahájení klient dostatečně specifikuje požadovaný výstup a tým spolu s projektovým manažerem stanoví, jakým způsobem ho bude dosaženo. Projekty, které jsou vhodné pro tradiční způsob projektového řízení mají několik charakteristických znaků – pokud je projekt splňuje, je vhodné použít tradiční přístup. (Wysocki, 2019, s. 44)

Prvním znakem je nízká složitost. Nemusí to vždy znamenat, že je projekt jednoduchý, spíše se může podobat projektu, který už byl někdy týmem dodán. Druhým znakem je předpoklad, že nebude potřeba projekt po naplánování příliš měnit. Každý požadavek na změnu je totiž časově náročný a negativně ovlivňuje časový plán. Další důležitý předpoklad se týká hlavně IT projektů – jedná se o jasnou technologickou infrastrukturu. Pokud není na začátku projektu jasné, jaká technologie bude použita, nebo tým zvolenou technologii předem nezná, není vhodné použít tradiční přístup. Důležitým předpokladem pro použití tradičního přístupu je známé a předvídatelné prostředí. Toho je docíleno pomocí minulých zkušeností, zevrubnou přípravou projektového manažera a jistotou klienta ohledně zadání projektu. Důležitým předpokladem je samozřejmě i zkušený tým v oblastech, které jsou pro daný projekt stěžejní. Tento předpoklad opět snižuje riziko změn v průběhu projektu. Tradiční přístup k projektovému řízení je obecně řečeno vhodný u takových projektů, u kterých je možno naplánovat každý dílčí krok a úkol. K tomu často slouží Work Breakdown Structure (WBS), což je nástroj sloužící právě pro detailní plánování projektu. WBS obsahuje rozpad projektu na potřebné zdroje – časový plán, požadavky na zdroje a žádoucí funkcionality výsledného produktu nebo služby. (Wysocki, 2019, s. 43 – 45)

V rámci tradičního přístupu je tedy k projektu přistupováno jako k celku. V projektových kruzích se o něm hovoří jako o vodopádovém přístupu (*waterfall*). Toto pojmenování si vysloužil díky tomu, že při pohledu z profilu vypadá projektový plán jako vodopád – projekt probíhá postupně a často dochází k tomu, že nějaká část musí být dokončena, aby se mohlo začít s další částí. Vodopádový přístup má jako kterýkoliv

jiný své výhody a nevýhody. Mezi výhody patří vysoká transparentnost, jelikož každý stakeholder má možnost nahlédnout do plánu a rychle získat představu o průběhu, současném stavu projektu a hlavně kdy bude za stávajících podmínek dodán. Do výhod je nutné zařadit i určitou stabilitu, kdy je možné vytvořit jednotnou rutinu pro všechny podobné projekty. Tím je možné docílit snadného přechodu pracovníků mezi projekty, případně zjednodušit souběh několika projektů. Hlavní nevýhoda má vazbu na cíl projektu. Málokdy se v praxi podaří na začátku projektu (což může být měsíce i roky před dodáním konečného výstupu) konkrétně a jasně specifikovat požadavky na výstup. Pokud v průběhu projektu dojde ke značným změnám požadavků (v reakci na konkurenční prostředí, legislativu, nové technologie, apod.), může původní specifikace přestat dávat smysl a může se stát buď neuskutečnitelnou nebo nepotřebnou. To může v krajních případech vést i k předčasnému ukončení projektu bez dodání konečných výstupů. K tomu může dojít z důvodu již výše zmíněného – každá změna je zásah a ohrožuje průběh projektu. (Dvořák, Mareček, 2017, s. 21 – 23)

2.1 Agilní přístupy k projektovému řízení

Nejprve je nutné si přiblížit, co vlastně slovo agilní znamená, jelikož se s ním v běžné mluvě příliš nesetkáváme. Autoři Šachová a Kunce (2019, s. 15) nabízejí poměrně srozumitelnou definici: „Agilní je dynamický, rychlý, interaktivní, přizpůsobivý, iterativní, zábavný, hravý, rychle reagující na změnu, ...“ (2019, s. 15) Agilnost nemá jasná pravidla a neexistuje jeden správný postup, proto agilní přístupy chápeme spíše jako způsob myšlení. Na rozdíl od tradičních principů, agilní přístup je zaměřen na spolupráci komunikaci a připravenost na změnu. Oproti tradičním přístupům je v agilním světě změna v podstatě vítaná, jelikož může přinést zadavateli větší přidanou hodnotu, což je u agilních principů společný jmenovatel úspěchu. I když není agile žádný striktní proces, má svá vlastní pravidla. Základním stavebním kamenem agilu je agilní manifest, který shrnuje čtyři základní pilíře a tím přibližuje, co znamená být agilní. (Šachová, Kunce, 2019, s. 15)

První hodnota manifesta zní: „**Jednotlivci a interakce před procesy a nástroji**“. (Agilemanifesto.org, 2001) Tato věta neříká, že by nebyly procesy a nástroje důležité, ale týmová práce a dobře sehraný tým jsou pro úspěch dle manifesta důležitější. Šochová a Kunce (2019, s. 17) tvrdí, že „*je známým faktem, že spolupracující týmy mají lepší výsledky než skupiny individuálně pracujících jednotlivců. Procesy a nástroje jim pomáhají dosáhnout výsledků, ale nejsou pro jejich úspěch nijak klíčové.*“

Druhá hodnota dle manifesta zní „**Fungující software před vyčerpávající dokumentací**“. (Agilemanifesto.org, 2001) Toto tvrzení jde proti vodopádovým přístupům, které na podrobné dokumentaci staví úspěch projektu. U agilních přístupů, které jsou používány hodně v IT prostředí, je nejdůležitějším cílem doručit fungující produkt. To neznamená, že by dokumentace nebyla důležitá, ale neměla by převážet nad produktem. Případy, kdy je dobré konkrétnější dokumentaci vytvářet, je v případě

interní dokumentace, která může sloužit pro budoucí týmy jako know-how. Při samotném vývoji je možné dokumentaci funkcionalit nahradit dobrou komunikací v týmu. Snaha u agilních přístupů je omezit na minimum, aby čas strávený nad dokumentací nepřevýšil její přidanou hodnotu. (Šochová, Kunce, 2019, s. 19)

Třetí hodnota agilního manifesta zní „**Spolupráce se zákazníkem před vyjednáváním o smlouvě**“. (Agilemanifesto.org, 2001) Stěžejním principem u agilních přístupů je zaměřenost na přidanou hodnotu pro zákazníka. Ten nemusí mít na začátku projektu jasnou představu o tom, co vlastně potřebuje a pokud je specifikace ve smlouvě podrobně vymezena, může to být pro další spolupráci problém. Manifesto proto poukazuje na fakt, že by mezi týmem a zákazníkem měla probíhat pravidelná a otevřená komunikace ohledně zákaznickových potřeb místo toho, aby se vztah řídil čistě na základě striktní smlouvy. To určitě neznamená, že smlouvy nejsou důležité, ale neměly by stát v cestě lepšímu produktu. (Šochová, Kunce, 2019, s. 19)

Poslední hodnotou manifesta je „**Reagování na změnu před dodržováním plánu**“. (Agilemanifesto.org, 2001) Tímto bodem se opět agilní přístupy vymezují od těch tradičních. Zatímco v tradičních přístupech projektového řízení nejsou změny v průběhu realizace projektu příliš vítané, agilní přístupy to přímo podporují. Technologie se totiž neustále mění a to, co se mohlo zdát jako nejlepším řešením na začátku projektu, nemusí být nejlepším řešením na jeho konci. Plány by tedy měly sloužit převážně jako vodítko, ale tým s klientem by se jich neměly držet za každou cenu. (Šochová, Kunce, 2019, s. 19)

Zatímco tradiční metody projektového řízení jsou vhodné pro široké portfolio projektů, agilní přístupy jsou používány hlavně při vývoji softwaru, nicméně již dnes jsou rozšířeny i řady dalších oblastí. Dále se tedy v této práci bude místo slovního spojení „projektové řízení“ vyskytovat „řízení při vývoji softwaru“, aby bylo zachováno názvosloví většiny agilních metodik. Mezi nejznámější a nejpoužívanější agilní přístupy při vývoji systému patří SCRUM, Kanban, XP (Extreme Programming), Lean Software Development and Feature Driven Development¹. V této práci budou blíže představeny pouze metodiky SCRUM a Kanban, jelikož na ty je zaměřena praktická část této práce.

2.1.1 SCRUM

Agilní metodika SCRUM je *framework* (rámec) pro potřeby produktového a softwarového vývoje, který je postavený na zpětné vazbě. Tento rámec zahrnuje interaktivní design, samo-organizující tým a neustálé vylepšování. (Morris, 2017, s. 21)

¹ Více o těchto přístupech například v této publikaci: HAUNTS, Stephen. *A Gentle Introduction to Agile and Lean Software Development*. Amazon Digital Services, 2018. ISBN 1980274088.

Podle McKenny (2016, rok, s. 26) je SCRUM nejvíce oblíbeným frameworkem mezi agilními metodami projektového řízení. Podle něj SCRUM podporuje získání zpětné vazby v krátkém čase, snižuje riziko a umožňuje návratnost investice v kratším čase oproti vodopádovým metodám projektového řízení.

Pojmenování SCRUM pro způsob řízení softwarového vývoje bylo poprvé použito Kenem Schwaberem a Jeffem Sutherlandem na počátku 90. let 20. století. Termín „scrum“ původem pochází z prostředí rugby, ve kterém pojem scrum znamená v překladu mlýn. Jedná se o taktiku, kdy se hráči semknou těsně k sobě a vytvoří tak tým, který je schopen rychlé reakce na změny, což je klíčovým prvkem právě SCRUM frameworku. (Robinson, 2016, s. 70 – 71)

SCRUM stojí na třech pilířích, kterými jsou transparentnost, kontrola a přizpůsobení. **Transparentností** je myšleno to, že všechny aspekty budou definovány za použití společného standardu. Toho je možné docílit tím, že všichni účastníci procesu budou používat stejnou terminologii týkající se daného procesu. Dalším takovým nástrojem je zajištění definice „Done“, která musí být stejná jak pro ty, kteří práci vykonávají, tak pro ty, kteří výslednou práci kontrolují. Pilíř **kontrola** má dvě hlavní oblasti. První je kontrola toho, aby probíhaly všechny Scrum ceremonie tak, aby byl proces vývoje co nejefektivnější. Druhou oblastí je kontrola reálného pokroku prací oproti plánu. Poslední pilíř – **přizpůsobení** – navazuje na pilíř druhý. Pokud je zjištěno, že se jeden nebo více aspektů odchyluje za hranici přijatelných hodnot a tím výsledný produkt nebude přijatelný, musí být proces nebo materiál upraven. K této úpravě musí dojít v nejkratším možném čase. (Schwaber, Sutherland, 2017, s. 9 – 10)

Metodika SCRUM namísto pojmu projekt pracuje s pojmem produkt. Produkt je jakýkoliv výrobek, služba nebo nápad, který přináší svým uživatelům nějakou hodnotu – jedná se tedy o výsledek práce. Zatímco pojem projekt může být chápán spíše jako jasně vydefinované a ohraničené úsilí, které umožní vznik produktu. Produkt je vyvíjen v rámci projektu, což znamená, že za dobu životního cyklu produktu se může vyskytnout více projektů. Dalším rozdílem je fakt, že projekt má jasně definováno, co musí být dodáno, zatímco produkt nikoliv. Produkt je vyvíjen v závislosti na změně potřeb uživatelů v čase. (Gomez, 2019)

SCRUM role

V metodice SCRUM rozeznáváme tři role: **Scrum Master**, **Product Owner** a **vývojový tým**. Tyto tři role dohromady formují **Scrum tým**. Níže budou všechny role blíže představeny, ale je důležité zmínit, že úspěch produktu se odvíjí přednostně od schopnosti lidí, kteří v rolích vystupují. Pojem **zákazník** bude sice často zmiňován ale v rámci SCRUMu není definován. Další postavou, která se bude v dalším textu vyskytovat ale není definována SCRUMem, je **uživatel**. Zákazník a uživatel většinou není

stejná osoba. Zákazník je často osoba nebo společnost, která vývoj produktu iniciuje, financuje a poskytuje ho uživatelům za poplatek nebo zdarma, zatímco uživatel je koncovým uživatelem produktu. Tento rozdíl je nutný vnímat, jelikož má každá skupina na SCRUM tým jiné požadavky. (Maximini, 2015, s. 289)

Product Owner

Ken Schwaber a Jeff Sutherland (2017, s. 14) popisují **Product Ownera** jako osobu, která je zodpovědná za maximalizaci hodnotu produktu, na němž pracuje vývojový tým. Product Owner je tedy zodpovědný za to, jak bude produkt úspěšný a jakou hodnotu uživatelům přinese. Dle autorů mezi další zodpovědnosti Product Ownera patří starost o Product Backlog, jehož účel bude vysvětlen později. Aby produkt přinášel hodnotu, je nutné, aby Product Owner udržoval Product Backlog viditelný, transparentní a aby všichni členové týmu měli přehled o tom, na čem se bude dále pracovat. (2017, s. 14) Role Product Ownera obnáší zodpovědnost nejen za Product Backlog ale zároveň by tato osoba měla mít v ideálním případě možnost posledního slova. Každý nápad nebo změna oproti Backlogu by měla být schválena právě Product Ownerem. Ten však nemá působit jako samostatná jednotka ale úzce spolupracovat s vývojovým týmem a Scrum Masterem. Jedině díky této kolaboraci lze přinést skutečnou hodnotu produktu. (Pichler, 2010, s. 2)

Do role Product Ownera je možné dosadit různé osoby na základě toho, co produkt potřebuje – nelze je však v průběhu vývoje měnit. Před začátkem vývoje je potřeba určit, jaké dovednosti jsou pro tento konkrétní produkt od role Product Ownera vyžadovány a určit takovou osobu, která jim odpovídá. Při vývoji aplikace či webu se bude jednat o osobu, která má technické znalosti, při vývoji fyzického produktu může roli Product Ownera zastat například obchodní zástupce, brand manager nebo přímo zákazník. Při výběru Product Ownera je opravdu důležité, aby vybraná osoba produktu rozuměla a znala ho. (Pichler, 2010, s. 3)

Scrum Master

Zatímco hlavní starostí Product Ownera je produkt, **Scrum Master** je zodpovědný za používání metodiky SCRUM, čili za proces jako takový. Scrum Master má zajistit, aby měl vývojový tým vše, co potřebuje pro svou práci a zároveň aby vývojový tým rozuměl a dodržoval předepsané postupy. Pokud je potřeba, funguje Scrum Master jako prostředník mezi Product Ownerem a vývojovým týmem, pokud je to potřeba. Pokud tedy dojde mezi Product Ownerem a vývojovým týmem k neshodě, je na Scrum Masterovi, aby obě strany navedl ke společnému řešení. K těmto neshodám většinou dochází, jelikož mají Product Owner a vývojový tým jinou představu například z pohledu technického řešení. Na Scrum Masterovi poté není to, aby rozhodl „co je

správně“, ale aby oběma stranám pomohl s komunikací a nalezení společného řešení, které bude pro produkt nejvíce přínosné. (Schwaber a Sutherland, 2017, s. 18)

Scrum Master plní řadu funkcí v závislosti na tom, kdo je potřebuje. Product Ownerovi pomáhá Scrum Master se zajištěním toho, aby vývojový tým dobře rozuměl cíli, rozsahu práce a produktu samotnému. Dále Scrum Master pomáhá Product Ownerovi se samotným Product Backlogem z pohledu zvýšení efektivity a maximalizace hodnoty produktu. Ve vztahu k vývojovému týmu plní role Scrum Mastera pozice kouče, psychologa a učitel. Scrum Master pracuje se členy vývojového týmu na samoorganizaci a pokud je třeba, na odstraňování překážek. Scrum Master by měl dělat vše pro to, aby měl tým nejlepší možné předpoklady pro splnění úkolu. Ve vztahu k okolnímu světu může Scrum Master pomoci s pochopením SCRUM principů tým, kteří metodiku SCRUM nebo vývoj softwaru pomocí agilního přístupu neznají. (Schwaber a Sutherland, 2017, s. 19 – 21)

Vývojový tým

The Development Team, jehož název lze v češtině přeložit jako **vývojový tým**, je jedno z mála označení, které lze ve SCRUMu snadno přeložit z angličtiny do jiného jazyka, proto bude používán v tomto případě český ekvivalent. Vývojový tým je složen z pracovníků různého zaměření dle potřeby, kteří pracují na dodávkách. V oblasti IT se jedná nejčastěji o programátory, testery a analytiky. Vývojový tým v metodice SCRUM by měl být z pohledu potřebných technologií soběstačný a jeho členové by se měli vzájemně znalostně a dovednostně doplňovat. Další z požadavků na vývojový tým je takový, aby byli členové schopni samoorganizace, jelikož je na členech, jak svojí práci vykonávají. Úkolem Product Ownera a Scrum Mastera je totiž vést vývojový tým k dokončení produktu, ale způsob, jakým každý jednotlivý člen týmu vykonává svou odbornou činnost, je zcela v kompetenci jednotlivých členů. (Morris, 2017, s. 34 - 35) Vývojový tým by měl být tak malý, aby si zachoval svojí schopnost rychle reagovat a zároveň tak velký, aby byl schopen doručovat dostatečnou hodnotu. Nejčastěji se tedy jako optimální velikost vývojového týmu uvádí počet 3 – 9 pracovníků. Tento počet nezahrnuje Product Ownera a Scrum Mastera, kteří nejsou členy vývojového týmu. (Schwaber a Sutherland, 2017, s. 17 – 18)

SCRUM události

SCRUM framework je postaven na pěti ceremoniích (událostech). Jedná se o různé typy schůzek nebo časových intervalů. Dodržování této kombinace událostí má za úkol zkoumat a přizpůsobovat SCRUM artefakty, které budou představeny později. Mezi těchto pět událostí patří **Sprint**, **Sprint planning**, **Daily Scrum**, **Sprint Review** a **Sprint Retrospective** (Miller a Ripley, 2020, s. 4). Dále v této práci budou některé z těchto

pojmu přeloženy pro lepší pochopení, nicméně vzhledem ke zvyklostem bude používáno jejich anglické pojmenování.

Sprint

Sprint, neboli v češtině běh, je označení pro časové období, kdy má být doručený buď celý, nebo část produktu. Jeho délka se odvíjí od zvyklostí dané společnosti, nicméně nejčastější délka sprintu jsou dva týdny nebo měsíc. Nový Sprint začíná okamžitě po ukončení předchozího Sprintu a k veškerým ostatní SCRUM událostem dochází právě v rámci Sprintu. V rámci sprintu lze mluvit o dílčím projektu, který má být na konci Sprintu dokončen. V průběhu Sprintu by nemělo docházet ke změnám cíle Sprintu, nicméně je možné po shodě mezi Product Ownerem a vývojovým týmem upravit objem práce. Pokud v průběhu Sprintu dojde k závažným důvodům, mezi které může patřit například zásadní směna požadavků na výsledný produkt, může Product Owner rozhodnout o jeho ukončení. V takové chvíli je nutné začít plánovat nový Sprint. (Schwaber a Sutherland, 2017, s. 23 – 25)

Sprint Planning

Agilní metody, mezi které SCRUM patří, se sice oproti tradičním přístupům liší tím, že na začátku projektu nevzniká rozsáhlá dokumentace, součástí které je plán, nicméně to neznamená, že v rámci SCRUMu k plánování vůbec nedochází. Pro plánování slouží **Sprint Planning**, který je startem nového Sprintu. Délka jednoho Sprintu se odvíjí hlavně od délky Sprintu. Při výpočtu se vychází z doporučení, kdy by Sprint Planning pro měsíční Sprint měl trvat přibližně 8 hodin. Pokud tedy je Sprint dvoutýdenní, měl by Sprint Planning trvat přibližně 4 hodiny. Této ceremonie se účastní celý SCRUM tým, tedy Product Owner, Scrum Master a vývojový tým. Mezi hlavní otázky, na které je potřeba v rámci Sprintu odpovědět, jsou, na čem a jakým způsobem se bude ve Sprintu pracovat. Na odpovědích na tyto otázky se musí všichni členové shodnout, což je jedním ze základních principů SCRUMu. Product Owner většinou navrhuje funkcionality, které by měly být v rámci jednoho Sprintu hotovy, ale finální slovo má vývojový tým. Může totiž docházet k tomu, že nějaká funkcionality dává smysl z produktového hlediska, ale není zatím možná z důvodu technologie. Cílem Sprintu je mít hotový seznam funkcionalit, na kterých se bude v průběhu Sprintu pracovat a které budou na konci Sprintu dodány. Tento seznam by neměl být v průběhu Sprintu měněn, pokud nedojde k objevení závažné překážky. V takovém případě může Product Owner rozhodnout o odložení dané funkcionality na příští Sprint. (Roudias, 2015, s. 143)

Daily Scrum

Účelem **Daily Scrumu**, neboli každodenní schůzky celého vývojového týmu, je posouzení pokroku u naplánované práce a případně upozornění na problémy. Přítomnost Scrum Mastera není nutná, jelikož tým by měl být schopen malé překážky sám odstranit. Pokud vzejde z Daily Scrumu vážný problém, až poté je Product Owner upozorněn.. Starostí Scrum Mastera je při Daily Scrumu pouze to, zda Daily Scrum proběhl. Daily Scrum by měla být krátká událost, ideální doba trvání se udává patnáct minut. Pokud se v průběhu Daily Scrumu ukáže, že práce neběží podle plánu, problém se řeší až po této události a řeší ho pouze osoby, kterých se týká. (Fowler, 2018, s. 90)

Sprint Review

Sprint Review je událost na konci každého Sprintu, na které je představeno, co je v rámci Sprintu dodáno. Této schůzky se má zúčastnit jak celý Scrum tým, tak i klient. Product Owner nejdříve vysvětlí, jaká práce byla odvedena, vývojový tým má poté prostor pro diskusi ohledně toho, co se podařilo, co se nepodařilo a jak se případné problémy řešily. Vývojový tým také poté představí reálný výsledek své práce a poskytne klientovi prostor pro dotazy. Na závěr Product Owner zhodnotí stav Backlogu, situaci na trhu a požadavky klienta, aby podle těchto informací případně upravil směřování produktu. (Schwaber a Sutherland, 2017, s. 34 – 35)

Sprint Retrospective

Sprint Retrospective je poslední událostí Sprintu. Účelem Sprint retrospektiv je zhodnotit, jak Sprint probíhal z pohledu lidí, vztahů, procesů a nástrojů, identifikovat, co se povedlo a co ne, a hlavně přijít s plánem na zlepšení pro další Sprint. Tuto schůzi vede Scrum Master, který má za úkol zajistit, aby schůze probíhala v pozitivním a produktivním duchu. Výstupem by měl být konkrétní seznam kroků ke zlepšení, které budou implementovány v dalším Sprintu. (Fowler, 2018, s. 97 – 99)

Scrum artefakty

Scrum artefakty jsou od toho, aby poskytly transparentnost klíčových informací, které jsou potřebné pro úspěšné doručování práce Scrum týmem. Artefakty představují objem práce nebo hodnotu, kterým by měl každý člen Scrum týmu rozumět a jednat se v podstatě o seznamy toho, co je potřeba udělat. Mezi Scrum artefakty patří **Product Backlog**, **Sprint Backlog** a **Increment**. (Roudias, 2015, s. 146 – 147)

Product Backlog

Pojem **Product Backlog** je velmi těžko přeložitelný do češtiny, nicméně se jedná o soupis všech funkcionalit, o kterých je známo, že musí být součástí hotového produktu. Za obsah a dostupnost Product Backlogu je zodpovědný Product Owner. Product Backlog se od svých obdobných seznamů funkcionalit u tradičního projektového řízení liší tím, že Product Backlog není nikdy považován za hotový. U tradičních přístupů je žádané, aby seznam funkcionalit byl ideálně hotový ještě před začátkem prací. V tom se SCRUM odlišuje, jelikož věří, že v průběhu vývoje může a mělo by docházet ke změnám v závislosti na potřebách uživatelů. (Schwaber a Sutherland, 2017, s. 38 – 39)

Jednotlivé požadavky nebo funkcionality jsou psány prostřednictvím **user stories** (uživatelských příběhů), které mají jasně definovanou formu. Každá user story obsahuje krátký příběh psaný v 1. osobě čísla jednotného (v rozsahu jedné věty), který odpovídá na otázku, co uživatel potřebuje a proč to potřebuje. Tato forma pomáhá vývojovému týmu pochopit, jakou roli má daná funkcionalita plnit. User story dále obsahuje seznam akceptačních kritérií, která mají být splněna, aby mohla být funkcionalita nakonec považována za hotovou a další parametry jako prioritu, velikost, závislost atd. User story tedy neodpovídá na otázku, jakým způsobem má být funkcionalita dosaženo. Toto rozhodnutí závisí plně na vývojovém týmu. (Moreira, 2017, s. 201 – 203)

Sprint Backlog

Sprint Backlog obsahuje soupis funkcionalit obsažených v Product Backlogu, které mají být hotové v rámci konkrétního Sprintu. Objem práce na určitý Sprint je předem definován pomocí definování velikosti user stories. K tomu dochází při Sprint Planningu a postupů pro odhad velikosti každé user story je několik. Společným cílem ale je hrubý odhad toho, jak dlouhou dobu práce na konkrétní user story zabere a podle těchto odhadů poté dochází k sestavení Sprint Backlogu. Sprint Backlog může být v průběhu Sprintu měněn tak, aby bylo možné dosáhnout cíle Sprintu. Tuto pravomoc má pouze vývojový tým. (Schwaber a Sutherland, 2017, s. 42 – 44)

Increment

Za Increment může být považována všechna práce z Product Backlogu, která má definici "Done", což znamená, že výsledek práce je použitelný a odpovídá týmové definici "Done", neboli hotovo. Increment je tedy přírůstek odvedené práce, která byla naplánována a dokončena v rámci jednoho Sprintu.

2.1.2 Kanban

Termín **Kanban** znamená v japonštině cedule nebo obrázková karta. Tento název pochází z prostředí výroby aut pod značkou Toyota. V této společnosti používali tyto karty v případě, kdy potřebovali dát vědět dodavateli, že je potřeba dodat určitá součástka. Stejná karta byla přiložena k součástce, když byla doručena a po montáži byla karta odejmuta a znova použita v případě, kdy byla potřeba další součástka stejného typu. Důležité v tomto procesu byl fakt, že počet karet zůstával stále stejný. Počet karet nemohl být změněn bez formálního rozhodnutí. Toto pravidlo mělo za účel, jak udržet rychlost výroby, tak zbytečně nevyrábět velké množství součástek, které nebylo možno využít. Stejný princip je používán v Imperial Garden v Tokiu. Při vstupu obdrží návštěvník vstupenku, kterou u východu musí vrátit. Počet vstupenek je omezen, takže tímto jednoduchým systémem je regulován počet návštěvníků. Návštěva této zahrady inspirovala Davida J. Andersona, který v té době pracoval v Microsoftu a potýkal se s problémem, kdy si pracovníci „nabírali“ mnoho úkolů, což pramenilo ve špatnou kvalitu výstupu a zpoždění v dodávkách. Spolu s dalšími vytvořil Kanban metodu, která je hojně využívána při softwarovém vývoji. (Björkholm a Björkholm, 2015, s. 7)

Mezi hlavní přednosti Kanbanu jako jednoduchého nástroje projektového řízení je to, že schůzky probíhají pouze, pokud jsou potřeba a současně nejsou vyžadovány žádné schůzky ohledně procesu, což je velký rozdíl oproti SCRUMu. Další předností je dobrá vizualizace projektového workflow (pracovního postupu). Právě na vizualizaci je Kanban postaven, jelikož celý tým vidí, kde vzniká a je problém a může se na něj zaměřit. Zároveň Kanban díky zobrazení práce v jednotlivých stádiích neumožňuje přeskakovat důležité kroky a tím minimalizuje možnost označit práci jako dokončenou, pokud tomu tak vážně není. Další předností Kanbanu je minimalizace práce, která umožňuje týmu přizpůsobovat priority na denní bázi, což umožňuje dodat práci včas. Kanban, jako žádný přístup k projektovému řízení, není dokonalý. Co ale umožňuje, je zjednodušit projektový management, snížit čas strávený schůzkami a možnost rychlé reakce na změnu, což mají agilní přístupy společné. (Brechner, 2015, s. 2 – 3)

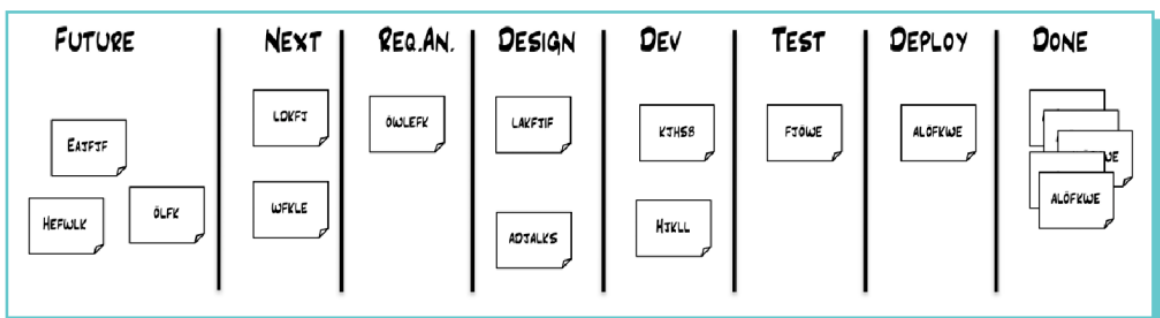
Dle jednoho z hlavních průkopníků Kanbanu, Davida J. Andersona, stojí Kanban na čtyřech základních principech. První princip říká: „Kanban začíná tam, kde již systém existuje“. Tento princip v podstatě znamená, že zavést Kanban není nijak časově ani znalostně obtížné, pokud už je nastavený nějaký základní způsob práce. Druhý princip je založen na respektování současného stavu. V Kanbanu nejsou zpochybňovány již zavedené procesy nebo již hotové funkcionality. Třetí princip souvisí s objemem práce. Kanban je založen na malých postupných krocích, ne na velkém objemu práce, která se dělá najednou. Zároveň je důležité, aby všichni zúčastnění došli ke shodě týkající se jakékoliv změny. Aby byl Kanban prospěšný, musí být zavedený management ve všech úrovních společnosti, což je čtvrtým principem. Aby docházelo k neustálému zlepšování, musí mít každý zúčastněný možnost přispět svým nápadem. V praxi jsou

to totiž právě pracovníci na úrovni operativy, kteří mají nejlepší vhled do každodenních procesů. (Klaus a Kaltenecker, 2015, s. 5)

Kanban je oproti SCRUMu více variabilnější ve smyslu, že si ho týmy mohou přizpůsobit dle potřeby. Kanban totiž neurčuje, jak dlouho má trvat dodání jedné funkcionality. Na rozdíl od SCRUMu, který je takzvaně timeboxed, což znamená, že je se jedná o proces, který je časově ohraničený, Kanban může být pokládán spíše za event-driven proces. Není tedy řízený časem, ale událostmi, kdy každá událost může trvat různě dlouho dle potřeby té konkrétní události. Pro plánování se v Kanbanu nepoužívá velikost úkolů jako u SCRUMu, nýbrž odhad času dodání. Kanban také nenařizuje, jak velký má být tým nebo jaké role v něm mají být zastoupeny. Zde je opět ponechán prostor pro nastavení jednotlivých týmů. (Kniberg a Skarin, 2010, s. 50)

Kanban praktiky

Tomas a Jannika Björkholmovi (2015, s. 9) uvádí šest praktických kroků při používání Kanbanu. První je založena na vizualizaci práce, kdy je důležité zobrazit jak jednotlivé kroky v procesu, tak práci, která se v jednotlivých krocích vyskytuje. Zobrazení je možné pomocí několika způsobů. Může se jednat o softwary, které toto zobrazení umožňují, jako je například velmi populární JIRA. Velmi často se ale v Kanbanu využívá obyčejná bílá tabule, kterou mají na očích všichni členové týmu. Další výhodou klasické tabule je možnost rychlého přizpůsobení potřebě, například přidáním nového kroku či popisu. Na Obrázku 1 je vyobrazeno, jak taková Kanbanová tabule může vypadat.



Obrázek 1 - Příklad Kanban tabule

Zdroj: Björkholm a Björkholm, 2015, s. 9

Druhým důležitým krokem je limitovat **work in progress** (rozpracovanou práci). Na Obrázku 1 se jedná o všechny kroky kromě Future a Done. Ostatní sloupce ukazují stádia rozpracovanosti. Nastavení limitu by mělo zabránit pomalejším členům týmu s pocitem zahlcení. Pokud se v nějakém kroku nakupí více úkolů, nemohou být do něj přesunuty úkoly z předchozího kroku. Mít v jednom kroku mnoho rozpracovaných úkolů není v Kanbanu žádoucí. Pokud se stane, že mají pracovníci úkoly v nějakém

z dřívějších kroků hotové a není je možné posunout dál z důvodu překročení kapacity následujících kroků, mohou buď přestat pracovat nebo lépe zkusit pomoci svým kolegům. Třetí krok je o zlepšování toku úkolů. Zde je důležité určení doby dodání úkolu, který by měl být určen s ohledem na složitost a schopnosti pracovníků. Cílem je samozřejmě snížení na minimum se zachováním kvality. Čtvrtý krok říká, že je důležité zajistit, aby všichni zúčastnění procesu a principům rozuměli a zároveň měli možnost ho zlepšovat. U tohoto kroku nejsou konkrétní nástroje, jak tohoto kroku docílit – záleží na společnosti, jak se k tomuto kroku postaví. Pátý krok souvisí se třetím krokem, který si zakládá na co nejrychlejším dodání. Pátý krok se ale zaměřuje právě na kvalitu, a to pomocí zpětné vazby od uživatelů softwaru. Tuto zpětnou vazbu je možné získávat jak přímo od uživatelů při rozhovorech nebo pomocí takzvaných automatizovaných testů, které jsou součástí kódu a které na případné chyby v softwaru upozorní. Šestý krok nabádá k použití všech teorií ohledně toku práce, které mohou pomoci se zkrácením doby dodání. Mezi takové teorie patří například Lean, teorie chaosu, teorie her nebo teorie omezení. (Björkholm a Björkholm, 2015, s. 8 – 11)

PRAKTICKÁ ČÁST

3 Aplikace agilních metodik při vývoji

Pro praktickou část mé diplomové práce jsem zvolila popis použití agilních metodik SCRUM a Kanban při vývoji mobilní aplikace XY, dále jenom Aplikace XY, ve Společnosti XY, pro kterou v současné době pracuji. Ve Společnosti XY se při vývoji běžně postupuje dle metodiky SCRUM, nicméně vývoj Aplikace XY si žádal zapojení principu Kanban z důvodu nedostatků metodiky SCRUM. V praktické části této diplomové práce nejprve představím Společnost XY včetně běžného procesu vývoje a poté ukážu na konkrétním případu implementaci kombinace agilních metodik SCRUMU a Kanbanu.

3.1 Společnost XY

Společnost XY je součástí velké IT společnosti, která působí na českém i slovenském trhu. Společnost XY je v rámci skupiny profilovaná jako agentura zaměřující se na vývoj frontendu mobilních a webových aplikací. Frontend si jednoduše můžeme představit jako to, co při používání aplikace vidíme. Funkcionality, které nevidíme a běží na pozadí aplikace, nazýváme backend.

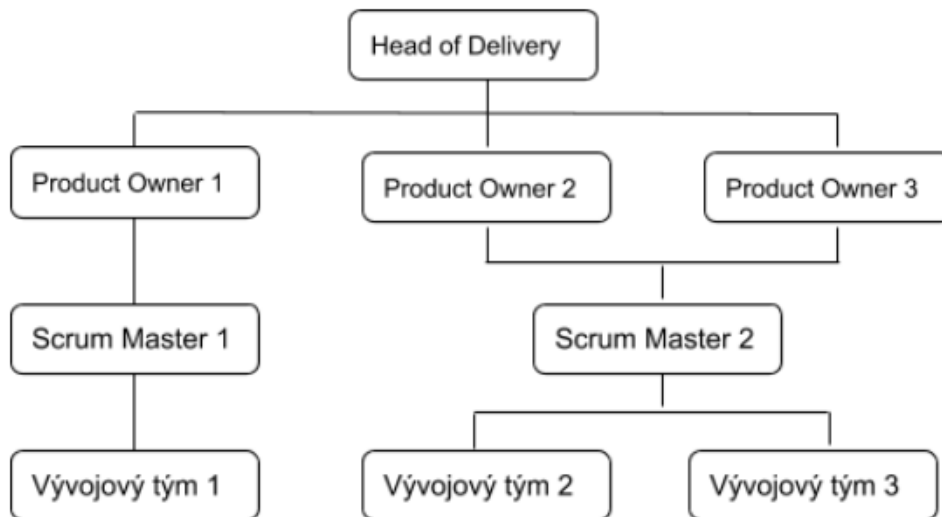
Společnost XY sídlí na pražském Smíchově a v současné době má cca padesát pracovníků. Z těchto padesáti pracovníků cca čtyřicet tvoří vývojové oddělení. Zbýlých deset pracovníků tvoří vedení společnosti, backoffice, designové oddělení, obchodní oddělení a HR oddělení. Backoffice, designové, obchodní a HR oddělení slouží jako podpůrná oddělení pro zajištění zakázek, lidských zdrojů a chodu Společnosti XY jako takové.

Společnost XY tak, jak je nyní známá, vznikla teprve před rokem rozdělením původní společnosti. Původní společnost vznikla před deseti lety jako startup, který se zabýval jak vývojem mobilních a webových aplikací, tak online marketingem. V létě 2019 se původní společnost rozdělila na marketingovou část, která vystupuje pod původním názvem a na Společnost XY, která se stala součástí větší IT skupiny. Mezi klienty Společnosti XY patří větší i menší převážně tuzemské společnosti. Klienty můžeme rozdělit na dvě skupiny – společnosti, které se sami IT vývojem zabývají a na Společnost XY se obracejí kvůli know-how a na společnosti, které podnikají v jiných oblastech a služby Společnosti XY potřebují jako podpůrné.

3.1.1 Vývojové oddělení

Organizační struktura vývojového oddělení Společnosti XY je zobrazena na Obrázku 2. Head of Delivery, který je zodpovědný za celé vývojové oddělení, má pod sebou jednotlivé Product Ownery. V ideální situaci je každý Product Owner "zodpovědný" za svůj tým. SCRUM sice říká, že ve SCRUM týmu nefunguje podřízenost a nadřízenost,

nicméně ve Společnosti XY je Product Owner, jako osoba zodpovědná za produkt, v organizační struktuře vývojovému týmu a Scrum Masterovi nadřazený.



Obrázek 2 - OS vývojového oddělení

Zdroj: Klára Vojtová

Pokud se jedná o komplexní produkt, je týmu vyhrazen samostatný Scrum Master. Pokud se jedná o produkt menšího rozsahu nebo pokud je SCRUM tým sehraný, může mít jeden Scrum Master na starost týmů více. Tato situace může nastat i v případě, kdy na produktu pracuje více týmů. Poslední složkou organizační struktury jsou vývojové týmy. Ve Společnosti XY je zvykem tvořit vývojové týmy o velikosti cca 5 pracovníků. Jejich specializace se odvíjí od potřeb aplikace, nicméně většinou vývojový tým tvoří tři frontend vývojáři, jeden Quality Assurance Engineer, který je zodpovědný za testování a jeden pracovník, který je zodpovědný za infrastrukturu aplikace.

3.1.2 Běžný průběh vývoje mobilní aplikace pomocí SCRUM metodiky

Jelikož se praktická část této práce zabývá uskutečněným vývojem konkrétní mobilní aplikace, v této podkapitole přiblížím, jak takový vývoj pomocí metodiky SCRUM ve Společnosti XY běžně probíhá. I když se zaměřím na vývoj mobilní aplikace, vývoj webové aplikace se příliš neliší.

Jak jsem již popsala výše, SCRUM tým se vždy skládá z Product Ownera, Scrum Mastera a vývojového týmu. Tento SCRUM tým pracuje ve čtrnácti denních Sprintech tak, aby bylo možné na konci každého Sprintu ukázat klientovi nové nebo upravené

funkcionality. To, jakým způsobem vypadá práce SCRUM týmu den po dni, představím níže.

Na začátku každého nového produktu je potřeba zvolit správně SCRUM tým. Jeho složení se odvíjí od druhu produktu. Jediná role, u které se dívá na zkušenosti s danou problematikou, je Product Owner, jelikož on za produkt odpovídá a zkušenosti s podobnými produkty jsou vítány. Vývojový tým se naopak skládá dle technologických znalostí – zkušenosti s podobným produktem nejsou potřeba. Scrum Master se poté dosazuje na základě seniority dle velikosti týmu a komplexnosti produktu.

Ceremonie

Sprint Planning

Na začátku každého Sprintu probíhá cca čtyřhodinový Sprint Planning, který se většinou koná v pondělí dopoledne. Této schůzky se účastní celý SCRUM tým, tedy Product Owner, Scrum Master a vývojový tým. Účelem této schůzky je rozhodnout, které funkcionality musí být ve Sprintu dodány. Product Owner, jako člověk zodpovědný za produkt, navrhuje funkcionality, které dle něj přinesou největší business hodnotu. Je poté na vývojovém týmu, jestli s návrhem souhlasí, či nikoliv. Pokud není možné funkcionality do Sprintu dle vývojového týmu zařadit z technického hlediska, hledá se jiná funkcionality. Je opravdu důležité, aby se všichni účastníci Sprintu dohodli na objemu práce, který musí být v rámci Sprintu dodán a zavázali se k jeho plnění. Na konci schůzky tedy musí být všem jasné, jaké user stories budou na konci Sprintu dodány a předvedeny klientovi.

Daily Scrum

SCRUM tým se každé pracovní ráno schází na krátkou schůzku, jejímž účelem je odhalit případný problém s určitou funkcionalitou. Jelikož se jedná o IT vývoj, problémy se objevují často, nicméně je potřeba, aby o nich tým a Product Owner věděli včas. Tato schůzka většinou trvá 20 minut a ve Společnosti XY jsou velmi oblíbené tyto schůzky provozovat ve stoje. To zaručuje rychlý průběh a zamezuje zbytečným průtahům. Úkolem Scrum Mastera je zajistit, aby byl celý vývojový tým přítomný a pokud tomu tak není, zajistit, aby se to už neopakovalo. Přítomnost Product Ownera není dle SCRUM metodiky nutná, nicméně ve Společnosti XY se jedná o běžnou praxi.

Grooming

Grooming je schůzka, která probíhá většinou v druhé půlce Sprintu a účastní se jí celý SCRUM tým. Jejím účelem je posoudit, které user stories by měly být zařazeny do dalšího Sprintu na základě potřeb produktu. Tato schůzka může připomínat Sprint Planning, nicméně Grooming se může považovat za přípravu na Sprint Planning. SCRUM tým na něm získá lepší představu o tom, jakým směrem se bude vývoj produktu posouvat. Účelem Groomingu tak není připravit seznam user stories, které budou součástí dalšího Sprintu, jedná se spíše o prioritizaci a podklad právě pro Sprint Planning. Dalším důvodem pro Grooming je rozpracování user stories, které byly do té doby pouze předpřipravené. Na Groomingu je prostor takové user stories dodefinovat s pomocí celého SCRUM týmu.

Sprint Review

Tento typ schůzek je ve Společnosti XY v podstatě jediný, kdy přijde vývojový tým do kontaktu s klientem. Pokud je to jenom trochu možné, je klient pozvaný na každou Sprint Review schůzku, která se odehrává na konci každého Sprintu. Cílem je představit klientovi buď nové, nebo upravené funkcionality, které si může klient zkusit a případně poskytnout zpětnou vazbu. Videálním případem by mělo proběhnout testování s budoucími uživateli, což v praxi nebývá příliš běžné. K uživatelskému testování většinou dochází při přípravě designu, k čemuž dochází na začátku projektu před samotným vývojem a poté v poslední fázi vývoje, kdy už je většina produktu hotova a ladí se detaily.

Sprint Retrospective

Této schůzky se musí účastnit celý SCRUM tým a jedná se o poslední schůzku každého Sprintu. Hlavní postavou této schůze je Scrum Master, jehož úkolem je zajistit, aby na konci této schůze vznikly akční kroky pro další zdokonalení práce SCRUM týmu. Aby toho Scrum Master dosáhl, je potřeba přesvědčit zbytek SCRUM týmu o přínosnosti této schůze, což nebývá lehký úkol. Tyto schůzky jsou obzvláště důležité u SCRUM týmů, jejichž členové spolu nejsou zvyklí pracovat. V takových případech bývá velký prostor pro zlepšení převážně v oblasti komunikace. Pokud na sebe nejsou členové SCRUM týmu zvyklí, může docházet k různým druhům nedorozuměním, ať už pracovním, či osobním. Na Sprint Retrospective schůzi se ale nemusí řešit pouze komunikace. Mezi další častá témata patří užívané nástroje, SCRUM proces jako takový, věci, které se v rámci Sprintu povedly a naopak problémy, které se naskytly.

Sprint Retrospective se odehrává nejčastěji v pátek, aby bylo v následující pondělí možné začít s novým Sprintem. Samozřejmě ne vždy je možné s ceremoniemi možné

čekat na určitý den, například kvůli státním svátkům, dovoleným atd, nicméně běžná praxe ve Společnosti XY je začínat Sprint Planningem v pondělí a končit Sprint Retrospektivou v pátek.

Činnosti

Vývoj aplikace je velice komplexní projekt. Aby byla aplikace – tedy produkt – úspěšná a funkční, je třeba při vývoji projít několika kroky. Každý z těchto kroků může být považován za činnost. SCRUM tabule, která je ve Společnosti využívána, kopíruje právě tyto činnosti.

Programování

První sloupec SCRUM tabule (To Do) zastává roli Sprint Backlogu. V tomto sloupci se na začátku Sprintu nachází všechny user stories, které mají být na konci Sprintu dokončeny. Jak již bylo zmíněno dříve, tyto user stories připravuje a zadává do Product Backlogu Product Owner, odkud je poté za souhlasu vývojového týmu přidává do Sprint Backlogu. Frontend vývojáři si poté na začátku a v průběhu celého Sprintu ze sloupce To Do vybírají user stories, na kterých začínají pracovat. V ten moment přesouvají user story do sloupce In Progress, aby bylo všem jasné, že práce na dané user story začala.

Testování

Pokud je práce frontend vývojáře hotová, posune user story do sloupce Quality Assurance, kde si jí přebírá QA Engineer. Ten má za úkol práci vývojáře otestovat a odhalit případné vady či nepřesnosti. Testování může probíhat dvěma způsoby – automatizovaně nebo manuálně. Automatizované testy jsou v podstatě malé části kódu, které QA Engineer, neboli tester, do kódu naprogramuje. Tyto testy poté po spuštění automaticky danou funkcionální testují a pokud funkcionální nereaguje tak, jak má, test na to testera sám upozorní. Tyto automatizované testy mohou běžet na pozadí aplikace, aniž by je tester musel stále dokola pouštět. Manuální testování naopak vyžaduje od testera plnou pozornost, jelikož při tomto typu testování tester doslova prochází danou funkcionální částí aplikace na několika typech zařízení a kontroluje, jestli se daná funkcionální chová a reaguje tak, jak je zamýšleno.

Code Review

Code Review je činnost, při které vývojář prochází kód jiného vývojáře, aby odhalil případné nedokonalosti. Jeho účelem je najít chyby v kódu, které první vývojář

přehlédl, čímž se zvyšuje kvalita kódu. Chyby v kódu nejsou vzácné, dochází k nim poměrně často, zvláště u funkcionalit, se kterými vývojář nemá zkušenosti. V takovém případě je žádoucí, aby test prošel kontrolou. Tato činnost je často spojována s testováním.

PO Acceptance

Když je funkcionalita zkontrolována po technické stránce, vrací se zpátky k Product Ownerovi. Ten musí posoudit, zda funkcionalita odpovídá požadavkům v user story. Nejprve posuzuje, zda funkcionalita plní svůj účel, splňuje akceptační kritéria a poté zda odpovídá designu. Pokud není s výsledkem spokojen, vrátí user story buď vývojáři nebo testerovi. Pokud je vše v pořádku, může jít user story do produkce

Deployment

Deployment je poslední částí vývoje mobilní aplikace. Tato činnost je velmi komplikovaná, ale nicméně se jedná o proces, kdy se aplikace dostane z počítačů programátorů k uživatelům. Nejprve je potřeba dát veškerý kód dohromady. Poté je nutné aplikaci nainstalovat na server. V ten moment je možné mobilní aplikaci nahrát do Google Play (Android) nebo App Store (iOS). Po schválení aplikací těmito službami si můžou uživatelé stáhnout aplikaci do svých mobilních zařízení a začít ji používat.

3.1.3 Nástroje používané během vývoje mobilní aplikace pomocí SCRUM metodiky ve Společnosti XY

Společnost XY, jako každá podobná společnost zabývající se IT vývojem, používá pro organizaci práce mnoho online nástrojů. Tyto nástroje byly zapojovány ve Společnosti XY postupně dle potřeby. Mezi nejpoužívanější patří JIRA, GitHub, InVision a Slack. Všechny tyto nástroje jsou poskytovány třetími stranami. Společnost XY využívá u některých z nich placené licence, u některých používá verzi, která je zadarmo. Toto rozhodnutí vždy závisí na tom, zda verze, která je zadarmo, stačí pro potřeby týmu.

JIRA

Product Backlog, potažmo Sprint Backlog, je ve Společnosti XY vytvořen v systému JIRA. Tento velmi populární nástroj umožňuje vytvářet, třídit a přiřazovat user stories daným pracovníkům. Systém JIRA také umožňuje vytvářet SCRUM tabule, které umožňují všem členům SCRUM týmu přehled ohledně práce. Standardní SCRUM tabule Společnosti XY má sedm sloupců: To Do, On Hold, In Progress, Quality Assurance, Code

Review, PO Acceptance, Ready For Deploy. Činnosti, které se odehrávají v jednotlivých sloupcích, popíšu dále.

JIRA je tedy zásadním nástrojem pro práci SCRUM týmu. Product Owner má stále přehled, na jakých user stories se zrovna pracuje a jaké user story jsou již hotové. Vývojovému týmu poté JIRA slouží jako prostředek, kterým si předávají jednotlivé úkoly. Nemusí se sejít osobně, aby si práci předali, nemusí si složitě dopisovat ohledně práce, stačí v user story změnit osobu, která je za ní zodpovědná a JIRA automaticky o této změně informuje buď pouze pracovníka, kterého se to týká, nebo i Product Ownera. Díky nástroji JIRA tak každý pracovník ví, na čem má pracovat a Product Owner má nad celým Sprintem přehled.

GitHub

GitHub je nástroj, který slouží hlavně pro sdílení kódu mezi vývojáři. Tento nástroj je ve Společnosti XY využíván především pro sdílení kódu a revizi kódu, pokud je potřeba. Díky tomu může několik vývojářů pracovat na jedné funkcionalitě v reálném čase. Pokud potřebuje některý z vývojářů vidět část kódu, která ho zajímá, jednoduše se do nástroje přihlásí a vidí nejnovější verzi. Takto nedochází ke ztrátě přehledu nad verzí kódu, která by mohla nastat, pokud by si vývojáři kód posílali například pomocí emailu.

InVision

InVision je nástroj, který neslouží přímo pro účely vývoje, ale pro návrh designu. Ten navrhují designéři většinou ve spolupráci s Product Ownerem před začátkem vývoje produktu. Na hotové designy se většinou Product Owner odkazuje v user stories, jelikož designy jasně udávají například pozici a rozměry tlačítek, textových polí atd. Cílem vývojářů je tedy nejen dosáhnout funkčnosti jednotlivých funkcionalit, stejně tak je důležitý vzhled aplikace a přehlednost – k těmto aspektům se přihlíží právě při zhotovování designů. Při vývoji zvláště mobilních aplikací může v průběhu vývoje vyjít najevo, že není možné nějakou část aplikace naprogramovat tak, jak bylo namýšleno v designu. Nejčastějším důvodem může být fakt, že funkcionalita není podporována mobilními softwary nebo by nebyla uživatelsky příjemná například z důvodu finálního umístění na obrazovce mobilního zařízení.

Slack

Slack je nástroj sloužící hlavně pro komunikaci. Slack je zároveň jediný z výše zmíněných nástrojů, které používají všichni pracovníci Společnosti XY. Nejedná se o nic víc než firemní chat, který slouží pro běžnou komunikaci. Ve Společnosti XY je poměrně

běžná praxe práce z domova, neméně běžná i práce z druhé strany světa. V takových případech je jednoduchý chat pro celý SCRUM tým nezbytný. Slack umožňuje vytvářet kanály zaměřené na určitá témata, společné i osobní konverzace. Nejposlednější předností Slacku je možnost integrace s nástrojem GitHub, což ještě více ulehčuje vývojářům práci.

3.2 Aplikace XY

Mobilní Aplikace XY byla vyvíjena pro účely pomoci sluchově postiženým při každodenních aktivitách. Hlavním principem je možnost spojení sluchově hendikepovaným s tlumočnickem do znakové řeči pomocí videohovoru za malý poplatek. Tlumočnick může sluchově hendikepovanému pomoci v nejrůznějších situacích, například při návštěvě lékaře, úřadu či obchodu. Pokud sluchově hendikepovaný potřebuje pomoci, stačí v Aplikaci XY stisknout tlačítko, které ho spojí s jedním z tlumočnicků. Objednatelem této aplikace je spolek, který sdružuje sluchově hendikepované.

3.2.1 Funkcionality Aplikace XY

Nejdůležitější funkcionalitou je tedy jistě videohovor. Důležité jsou ale i další funkcionality, které má většina mobilních aplikací společné, například registrace, přihlášení, zadání a autorizace platební karty apod. je také nutné myslet na to, že Aplikace XY má dva typy uživatelů. Prvním typem jsou samozřejmě sluchově hendikepovaní. Pro tyto uživatele je důležitá jednoduchost aplikace, aby jí mohli používat v momentech, kdy se nutně potřebují s někým domluvit. Druhou skupinou uživatelů jsou tlumočníci do znakové řeči. Potřeby této skupiny se liší od potřeb skupiny první. Tlumočníci potřebují dobrou kvalitu videa, by byli schopní odezírat znakovou řeč. Dále je pro ně důležité mít přehled nad uskutečněnými hovory a obdržet peníze za svou službu.

Registrace a přihlášení

Aby mohl uživatel začít používat Aplikaci XY, musí se nejprve registrovat pomocí jednoduchého formuláře. Prvním krokem je výběr, zda uživatel působí v roli zákazníka, v případě sluchově hendikepovaného, případně o tlumočnicka. registrace je zadání emailové adresy a hesla. Po tomto kroku je nutné vyplnit jméno, příjmení, emailová adresa, telefonní číslo a heslo. Po odsouhlasení Zásady ochrany osobních údajů a Obchodních podmínek je možné pokračovat v registraci. V případě, kdy je emailová adresa již zaregistrována, aplikace registraci nepovolí a vyzve uživatele k zadání jiné emailové adresy, případně ho přesměruje na formulář Přihlášení. Po splnění přechodných podmínek je tlumočnick registrován, sluchově hendikepovaný je ještě v posledním kroku registrace vyzván k zadání informací své platební karty. Po tomto kroku je registrace dokončena.

Pokud je uživatel již v aplikaci registrován, musí se vždy při otevření aplikace přihlásit. Přihlašovací formulář je stejný jak pro sluchově hendikepovaného, tak pro tlumočníka, je potřeba vyplnit pouze email a heslo. Pokud není jeden z údajů správný, aplikace vyzve uživatele k nápravě. V případě, kdy uživatel zapomněl své heslo, přihlašovací formulář obsahuje i tlačítko „Zapomenuté heslo“. Aplikace po kliknutí na toto tlačítko odešle na zadanou emailovou adresu email s instrukcemi pro obnovu hesla.

Videohovor

Uskutečnění videohovoru je klíčové pro obě skupiny uživatelů. Pokud potřebuje sluchově hendikepovaný pomoci, stačí otevřít Aplikaci XY, stisknou tlačítko „Spojit s tlumočnickem“ a vyčkat, na koho ho aplikace přepojí. Stisknutím tlačítka se spustí algoritmus, který vybere z online tlumočnicků jednoho. V ten moment se sluchově hendikepovanému zobrazí video s tlumočnickem, který je schopen překládat znakovou řeč do řeči mluvené a naopak. Po uskutečnění hovoru se oběma uživatelům zobrazí informace, jak dlouho hovor trval a kolik stál Kč. Sluchově hendikepovaný má navíc možnost tlumočnicka ohodnotit.

Můj účet

Aplikace XY umožňuje spravovat uživatelský účet. Uživatel může v svém účtu nastavit profilovou fotografii a spravovat své údaje, například heslo a jméno, případně svůj účet smazat. Dále uživatelský účet umožňuje zobrazit přehled uskutečněných hovorů včetně délky trvání a ceny. Uživatelská účet tlumočnicka navíc umožňuje zvolit, zda je tlumočnick v dané chvíli k dispozici. Rovněž ve svém účtu vidí tlumočnick průměrné hodnocení od zákazníků. Uživatelský účet také poskytuje možnost zjistit si o aplikaci více nebo jí doporučit známému.

3.3 Kombinace SCRUMU a Kanbanu při vývoji mobilní Aplikace XY

Projekt, jejímž výsledným produktem je Aplikace XY, začal v červnu 2020, samotný vývoj poté o měsíc později, tedy v červenci 2020. Na začátku projektu bylo nutné sladit očekávání klienta, Společnosti XY a budoucích uživatelů. Doba, ve kterém měla být Aplikace XY vyvinuta a připravena ke zveřejnění, byla na začátku projektu odhadnuta na maximálně 5 měsíců. Tomuto odhadu odpovídal i finanční odhad, který se pohyboval ve výši 1.500.000 Kč bez DPH. Časový a finanční odhad byl proveden na základě požadavků klienta na aplikaci na začátku projektu. V IT vývoji je velice složité

odhadovat cenu díla a časový harmonogram na začátku projektu, zvláště pokud se jedná o aplikaci, která obsahuje speciální požadavky.

V červnu 2020 tedy proběhl úvodní workshop, jehož výstupem byl přehled situací a problémů, které měla Aplikace XY řešit. Product Owner poté tento přehled rozpadl na menší části, které odpovídaly jednotlivým funkcionalitám a s designérem navrhl prototyp aplikace. Tento prototyp prošel procesem schvalování s klientem a současně s potenciálními uživateli, jelikož několik zástupců klienta jsou lidé se sluchovým hendikepem. Po lehké úpravě byl design připravený k vývoji.

3.3.1 Začátek vývoje pomocí metodiky SCRUM

Aby mohl vývoj Aplikace XY začít, bylo nutné nejprve sestavit SCRUM tým. Jelikož se jednalo o vývoj aplikace, která musí řešit specifické požadavky uživatelů, bylo jasné, že je potřeba zkušený tým. SCRUM tým byl tedy sestaven z nejzkušenějšího Product Ownera, který sice nemá mnoho zkušeností s mobilními aplikacemi, ale zato je velmi zkušený ve stavění webových aplikací od začátku do konce. K němu byl přiřazen seniorní Scrum Master a vývojový tým, který je zvyklý spolupracovat. Součástí vývojového týmu byly tři frontend vývojáři, dva testeři a jeden pracovník zodpovědný za infrastrukturu, který se měl na produktu podílet až v poslední fázi. Já osobně jsem se na projektu podílela na pozici juniorního Product Ownera. Mým úkolem bylo pomoci seniornímu Product Ownerovi s přípravou user stories a rovněž jsem se účastnila všech týmových schůzek. I když není ve Společnosti XY běžné, aby se na takovém projektu podílelo více Product Ownerů, jednalo se o zajímavý projekt, a proto mi to bylo vedením společnosti umožněno.

Na začátku projektu také byla zvolena technologie, pomocí které byla Aplikace XY vyvíjena. Jedná se o React Native, což je frontendový jazyk, ve kterém je možné vyvíjet pro obě platformy (Android a iOS) většinu kódu společně. Tím se liší od takzvaného nativního vývoje, kdy probíhá vývoj zvláště pro Android a zvláště pro iOS. Výhodou React Native je tedy ušetření nákladů a času. Dále bylo na začátku projektu rozhodnuto, že se bude aplikace vyvíjet takzvaně serverless, což znamená, že aplikace nepoběží na fyzickém serveru, nýbrž v cloudu. Toto řešení je poměrně nové ale představuje budoucnost. Přestože jsme s tím neměli ve Společnosti XY příliš zkušeností, rozhodli jsme se tuto technologii použít. Byla to příležitost zkusit si něco nového, což je v IT vývoji nezbytné.

Jako každý podobný produkt, který Společnost XY vyvíjí, i vývoj Aplikace XY byl od počátku řízen metodikou SCRUM. Délka Sprintů byla nastavena na čtrnáct dní jako obvykle. Prvních čtrnáct dní v červenci 2020 připravoval seniorní Product Owner s pomocí juniorního Product Ownera user stories dle požadavků klienta a připraveného designu, které byly přidány do Product Backlogu. Začínalo se od user

stories, které bylo plánováno zařadit do prvních Sprintů. Tyto user stories byly připraveny do detailu, zbytek user stories, které odpovídaly funkcionalitám, na kterých se mělo pracovat v pozdější fázi vývoje, byly pouze předdefinovány.

Sprint 1 a 2

První Sprint začal Spint Planningem v pondělí 15. června 2020. Na této schůzce bylo do Sprint Backlogu z Product backlogu několik user stories, které měly být na konci Sprintu hotovy. Nebudu zde uvádět konkrétní výčet user stories, ale jednalo se o funkcionality spojené s registračním a přihlašovacím formulářem, což jsou funkcionality, se kterými se většinou u mobilních aplikací začíná.

První Sprint probíhal poměrně bez problémů, vývojový tým pracoval na user stories, které nebyly příliš komplikované, proto Sprint Review, které se konalo 26.června 2020 za účasti klienta, proběhlo úspěšně. V průběhu prvního Sprintu samozřejmě proběhl první Grooming, na kterém se probíralo, co za user stories budou následovat. Výsledkem bylo zaměřit se na dokončení přihlašovacích a registračních formulářů, například dokončit různé typy hlášek, kterými Aplikace XY upozorní uživatele, že zadal nesprávné údaje. Vývojový tým dále upozornil na funkcionality spojené s videohovorem a platbami, u kterých očekával komplexnost, proto bylo navrženo zařadit user stories týkajících se těchto problémů co nejdříve.

Druhý Sprint odstartoval 29. července 2020. Jeho hlavním cílem bylo dokončit přihlašovací a registrační formuláře. Vývojový tým pracoval, schůzky probíhaly, vše šlo podle plánu. Klient byl zatím s naší prací spokojen. Product Owneři v průběhu druhého Sprintu začali zjišťovat více informací o možnostech videohovoru a platebních metod. V podstatě nám hned bylo jasné, že potřebujeme, aby možnosti zjistili sami vývojáři, jelikož všechna řešení vyžadovala hluboké technické znalosti. Na Groomingu, který proběhl 6. srpna 2020, vyjádřili Product Onweři obavy ohledně videohovoru a platební brány a celý SCRUM tým rozhodl, že tímto směrem se musí ubírat další Sprints. Na Sprint Review 9. srpna 2020 byly klientovi představeny téměř kompletní registrační a přihlašovací formuláře, které se mohl klient vyzkoušet.

Sprint 3 a 4

Na Sprint Planningu 12. srpna byly vybrány user stories, které se týkaly videohovoru. Product Owneři se dokonce rozhodli připravit user story, jejímž účelem bylo nastudování informací týkající se videohovorů. Zbytek vývojového týmu měl pracovat na user stories, které měly připravit kód na integraci videohovorů.

Již první týden třetího Sprintu na Daily Scrumu zazněla obava vývojáře, který studoval napojení videohovoru. Došel totiž k závěru, že se jedná o problematiku, se kterou nemá

nikdo zkušenosti. Hlavní Product Owner proto rozhodl o pozastavení přípravných prací týkající se kódu a pověřil nejzkušenějšího vývojáře dalším výzkumem. Zbytek vývojového týmu měl mezitím pracovat na drobných nedodělcích z minulého Sprintu. Druhý týden třetího Sprintu byly vývojovému týmu představeny nastudované poznatky. Na základě nich museli Product Owneři okamžitě začít pracovat na nových user stories, které sloužily pro potřeby napojení videohovorů. Na konci třetího Sprintu proběhlo Sprint Review s výsledkem, který nebyl pozitivní. Klient si stěžoval na dodanou práci, jelikož v porovnání s předchozími Sprinty se jednalo o drobnosti. Hlavní Product Owner se snažil situaci vysvětlit a slíbil zlepšení v dalším Sprintu.

Čtvrtý Sprint začal opět Sprint Planningem, na kterém bylo rozhodnuto, že celý vývojový tým bude pracovat na videohovoru. Hlavním problémem nebyl videohovor sám o sobě, vývojový tým musel přijít s algoritmem, podle kterého bude aplikace vybírat z tlumočnicků, kteří jsou k dispozici. I přes veškerou snahu vývojového týmu se ani na dalším Sprint Review nepodařilo klientovi představit novou funkcionalitu. Jednalo se tedy již o druhý Sprint, který skončil neuspokojivým výsledkem. Vývojový tým nebyl schopen dodávat naplánované funkcionality, ke kterým se na začátku Sprintu zavázal, jelikož se požadavky často ukázali složitější, než bylo na první pohled patrné. SCRUM tým byl tedy nucen přijít s řešením, které by nabídlo reagovat na změny v kratším časovém úseku a zároveň zbavil vývojový tým takového tlaku. Členové vývojového týmu si totiž začali stěžovat Scrum Masterovi na přílišný tlak od Product Ownerů, na které zase tlačil klient a vedení Společnosti XY.

Na Sprint Retrospectivě (Sprint 4) jsme proto spolu se Scrum Masterem navrhli odlišný přístup řešení, jelikož současný nevyhovoval ani SCRUM týmu, ani klientovi. Nová metodologie měla zachovat prvky SCRUMU, jelikož podle této něj probíhá vývoj v celé Společnosti XY. Hledali jsme tedy jiný agilní přístup, který v podstatě pomohl SCRUM zjednodušit. Nakonec bylo rozhodnuto o Kanbanu, se kterým měl Scrum Master již zkušenosti.

3.3.2 Přejít na řízení vývoje Aplikace XY pomocí kombinace metodik SCRUM a Kanban

Po čtvrtém Sprintu tedy bylo odsouhlaseno, že se nově bude postupovat podle takzvaného Scrumbanu, což je kombinace metodik SCRUM a Kanban. Tento systém nemá přesně určená pravidla, záleží pouze na celém týmu, jaká pravidla si od každé metodologie zvolí. Použití Scrumbanu bylo pro Společnost XY poměrně přelomové, jelikož to byla první zkušenost svého druhu.

Důvody pro použití kombinace SCRUMU a Kanbanu

Důvodů pro použití SCRUMU a Kanbanu bylo několik. Prvním problémem při vývoji Aplikace XY bylo příliš mnoho komplikací, které vyvstaly z důvodu nedostatečných zkušeností vývojového týmu s danou problematikou. S probíhajícím vývojem totiž bylo jasné, že videohovor nebude jedinou komplikací. Vývojový tým dále upozornil na potenciální problém s platební metodou, kdy platba musela probíhat přes systém třetí strany, což je vždy komplikované, jelikož nikdo nemůže dopředu odhadnout, jak přesně systém třetí strany funguje a jak se bude chovat. Další potenciální problém jsme očekávali při nasazení do cloudu, jelikož jsme s tím neměli dřívější zkušenosti ale pro toto opatření jsme se rozhodli, jelikož jsme si mysleli, že je Aplikace XY pro toto řešení perfektním kandidátem.

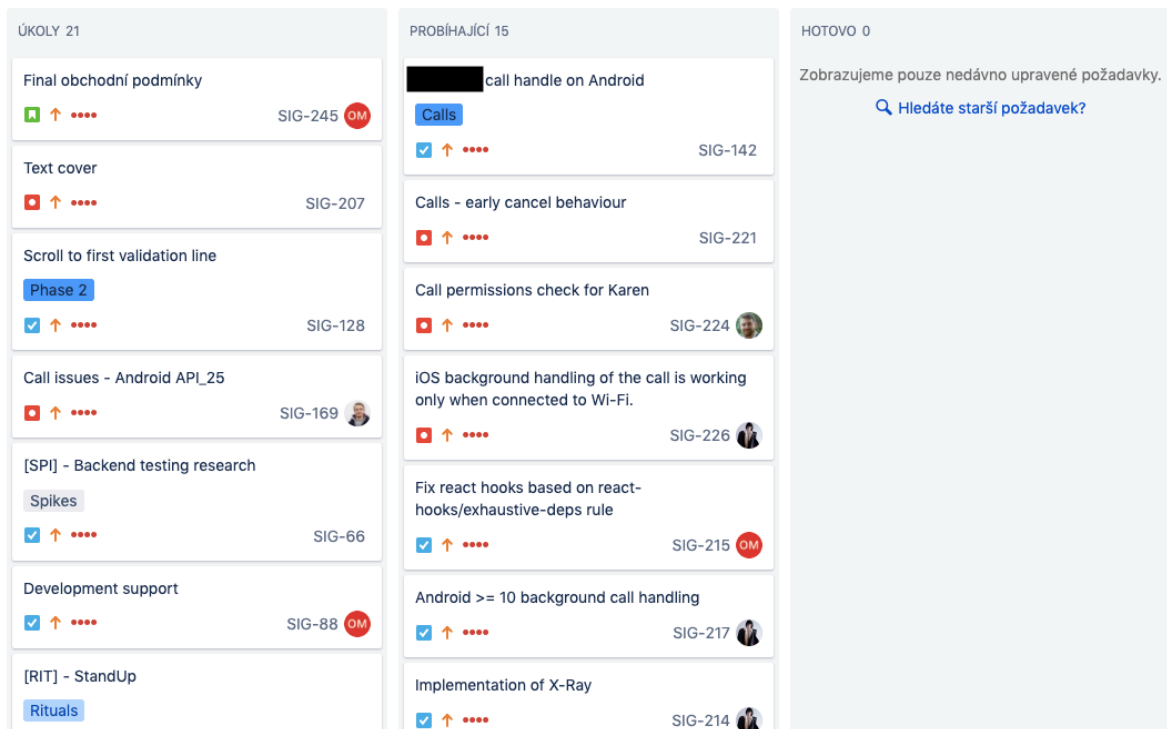
Dalším důvodem byla větší míra flexibility, kterou Kanban oproti SCRUMU nabízí. Díky zavedení Kanbanu bylo možné pružněji reagovat na nastalé situace, aniž by tým musel dodržovat čtrnáctidenní Sprints. Čtrnáct dní byla totiž v případě vývoje mobilní Aplikace XY příliš dlouhá doba na plánování objemu práce. Dá se říci, že se objem potřebné práce měnil každých pár dní, s čímž SCRUM příliš nepočítá. Aplikace XY zkrátka obsahovala příliš mnoho proměnných, se kterými se na začátku nepočítalo a v průběhu vývoje se objevovaly další.

Třetím důvodem, který nás vedl k zavedení Scrumbanu, byla psychická pohoda týmu. S přibývajícím problémy totiž bylo hlavně na vývojovém týmu vidět frustrace kvůli malým pokrokům v rámci Sprintů. Z tohoto důvodu bylo pro Scrum Mastera motivovat vývojový tým do práce, která byla potřeba. V podstatě se vývojáři dostali do stádia, kdy s každým vyřešeným problémem přibývalo několik nových.

Vývoj prostřednictvím metodologie

Scrumban

Scrum Master tedy přišel s následujícím řešením. Ze SCRUMU zůstaly SCRUM role, jelikož na ty je tým zvyklý a Kanban žádné speciální role nepředepisuje. Ze SCRUMU dále zůstaly některé ze schůzek. Mezi schůzky, které byly zachovány, patřil Daily Scrum, Grooming a Sprint Review, které bylo přejmenováno na Demo, jelikož se už nestahovalo ke konkrétnímu Sprintu. Naopak schůzky Sprint Planning a Sprint Retrospective byly zrušeny. Z Kanbanu byla přijata Kanban tabule v systému JIRA, která byla oproti SCRUM tabuli velmi zjednodušená. Nová Scrumban tabule obsahovala pouze tři sloupce a to ÚKOLY, PROBÍHAJÍCÍ A HOTOVO tak, jak je ukázáno na Obrázku 3.



Obrázek 3 - Ukázka Scrum tabule

Zdroj: Interní JIRA Společnosti XY

V pondělí 9. září byl vývojovému týmu představen nový způsob práce. První reakce vývojového týmu byly smíšené, jelikož se jednalo o novou věc. Ještě ten den se začalo s vývojem podle nového systému. Cílem bylo sledovat, kolik user stories se nachází ve sloupci PROBÍHAJÍCÍ, aby nebyl vývojový tým pod přílišným tlakem.

Hlavní Produkt Owner se však po zavedení nového systému obával toho, že vývojový tým nebude pocítovat tlak žádný, což také není žádoucí a tím pádem se vývoj výrazně zpomalí. Přišel proto s návrhem zavést pravidelné schůzky, na kterých by se aktualizovala roadmapa projektu. K těmto schůzkám docházelo nepravidelně, vždy dle potřeby Product Ownera. Ten po změně metodologie představil týmu zcela jednoduchou roadmapu, jejíž součástí byly pouze velké části aplikace s daty dodání. Nebyly tedy naplánovány jednotlivé user stories, pouze větší celky. Pokud se práce zpozdila oproti roadmapě, nejdříve byl zanalyzován důvod zpoždění a až poté byla roadmapa aktualizována.

Důležitým krokem také bylo informovat o změně metodologie klienta, jelikož ten byl na začátku projektu se SCRUMEM a obeznámen a tato metodologie mu přišla vhodná. Klient na začátku vyjádřil obavu se zpožděním dokončením produktu, proto trval na pravidelných Demo schůzkách, aby neztratil nad produktem přehled. S tímto požadavkem neměl ani Product Owner, ani vývojový tým problém, proto došlo ke shodě. Jediným rozdílem oproti Sprint Review bylo to, že Demo schůzky nebyly

striktně nastavené, většinou se uskutečňovaly, pokud tým dokončil nějaký celek nebo na přímou žádost klienta.

Již po prvních týdnech od změny metodologie bylo poznat, že z pohledu fungování týmu se jednalo o krok dobrým směrem. Oproti tomu problémy s vývojem přetrvávaly a veškeré obavy, které vývojový tým v prvních Sprintech předpokládal, nastaly. Největším problémem bylo použití cloudu na rozdíl od fyzického serveru. Při vývoji aplikace, která má na konci běžet v cloudu, je na tento fakt nutné myslet již od začátku a je nutné v průběhu celého vývoje spolupracovat s pracovníky, kteří se zabývají infrastrukturou aplikace, takzvanými architekty. I přesto, že s architektem bylo už od začátku počítáno, nikdo nedokázal odhadnout, kolik práce ho díky cloudu čeká. Tato komplikace stála vývojový tým mnoho práce, jelikož když si uvědomili svojí chybu, byl už vývoj aplikace v pokročilém stádiu a s touto informací se musela velká část kódu změnit.

Přes všechny tyto problémy se dařilo vývojovému týmu pokračovat v práci, nicméně v říjnu 2020 bylo jasné, že k dokončení aplikace v listopadu, jak bylo slíbeno, nedojde. V tu dobu byl dokončen jiný velký projekt, proto bylo možné do vývojového týmu přidat dalšího frontend vývojáře, který pracoval na dálku ze Srbské pobočky. Tato skutečnost přinutila celý tým přesunout fyzické schůzky do online podoby. Stejně tak musela komunikace na schůzkách probíhat v angličtině, což je v IT vývoji poměrně běžné, a proto to nebyl žádný významný problém.

Každý den probíhala ráno krátká schůzka, původně pojmenovaná Daily Scrum, na které se řešilo, na čem bude který pracovník vývojového týmu pracovat. Jelikož došlo ke změně ze SCRUM tabule, ve které bylo jasné, v jakém stádiu se jaká user story nachází, bylo po přechodu na Scrumbanovou tabuli poměrně složité zvláště pro Product Ownery držet krok s průběhem vývoje. Proto se tyto každodenní schůzky staly nejdůležitější částí procesu. Od změny metodologie bylo ještě důležitější sledovat, který pracovník pracuje na které user story.

Grooming schůzky se od původního nastavení příliš nelišily, jediný rozdíl spočíval v tom, že nedocházelo k tvoření Sprint Backlogů, tím pádem bylo nutné jiné odlišení user stories, na kterých se má začít pracovat. Toho bylo docíleno pomocí prioritizace, kdy na každé Groom schůzce byly vybrány user stories, které měli největší prioritu a dostaly tak označení vysoká priorita. Další dva stupně priorit byly střední a nízká. Jelikož nedocházelo k tvoření Sprint Backlogů, který byl původně zastoupen ve SCRUM tabuli, byly všechny user stories s původního Product Backlogu přesunuty do sloupce ÚKOLY ve Scrumbanové tabuli. Pokud se ve sloupci PROBÍHAJÍCÍ nacházelo málo user stories a současně k málo user stories ve sloupci PROBÍHAJÍCÍ byla přiřazena vysoká priorita, v ten moment byla svolána Grooming schůzka.

V listopadu 2019, což měl být termín dokončení Aplikace XY dle původního odhadu, byl stanoven nový termín dokončení. S přihlédnutím k množství komplikací, které v průběhu vývoje již nastaly a jaké byly ještě očekávány, se termín dokončení posunul na únor 2020. Klient s tímto termínem souhlasil, jelikož mu bylo jasné, že aplikace, kterou vyvíjíme, je mnohem složitější, než kdokoliv na začátku projektu mohl očekávat. I přes veškerou snahu vývojářů se však ani v únoru 2020 nepodařilo Aplikaci XY nasadit do produkce, jelikož finální testování videohovorů proběhlo s neuspokojivým výsledkem. Po dohodě s klientem bylo odsouhlaseno, že nedojde ke zveřejnění aplikace, která není zcela funkční. Aplikaci XY se v čase dokončení této diplomové práce stále nepodařilo nahrát ani do App Store, ani do Google Play a celý tým pracuje na co nejrychlejším dokončení.

3.4 Přínosy metodologie Scrumban a návrhy na zlepšení

Přínosy, které měla kombinace SCRUMU a Kanbanu přinést jsem se rozhodla rozdělit na dvě roviny. První rovinou je zefektivnění vývoje, druhou pak vliv na tým jako takový. Po uvedení přínosů, kterých jsem byla při vývoje Aplikace XY svědkem, uvedu několik návrhů na zlepšení, které by dle mého názoru vývoj ještě zefektivnil.

3.4.1 Zefektivnění vývoje

Přínosy

Zda změna metodologie pomohla při vývoji Aplikace XY je poměrně těžké posoudit, jelikož do problému vstupovalo více proměnných. Hlavním problémem byla komplexnost aplikace, kterou na začátku projektu podcenili všichni stakeholderi. Vývoj Aplikace XY měl původně trvat pět měsíců a cena za Aplikaci XY se měla pohybovat kolem částky 1.500.000 Kč bez DPH. Ani jeden z těchto cílů nebyl splněn, naopak byly oba tyto cíle převýšeny o více jak dvojnásobek. Důvodem je fakt, že v době naceňování a odhadu hrubého harmonogramu nebyly známy veškeré funkcionality a příslušná technická řešení. K tomuto problému dochází k IT vývoji poměrně často, pokud se nejedná o aplikaci, která má na trhu již mnoho konkurenčních aplikací podobného druhu. Není však tak časté, že by původní odhady byly takto nepřesné.

Jelikož není možné zjistit, jak dlouho by trval vývoj, kdyby se pokračovalo ve SCRUM metodice, jelikož žádné dva IT projekty nejsou stejné, je velmi těžké určit, co je přínosem a nikoliv. Dle mého názoru však bylo velkým přínosem Scrumbanu možnost ještě pohotovější reakce na potřebné změny, než je tak u SCRUMU. Tým se mohl rozhodovat, na jaké user story se má pracovat, velmi rychle, a to v řádu hodin. Jistě, ve SCRUMU funguje volnost vybírat si user stories dle preferencí pracovníka, vždy si však může vybírat pouze z user stories, které se nachází ve Sprint Backlogu. Při použití Scrumban metody mohlo být rozhodnuto o přiřazení vysoké priority téměř okamžitě, stačilo se dohodnout s Product Ownerem, případně svolat Grooming.

Věřím, že tento přínos proces vývoje urychlil, jelikož si tým mohl přidávat práci v podstatě kdykoliv, aniž by musel čekat na Sprint Planning. Přínosem je jistě také zrychlení dodání hotových user stories a zkvalitnění kódu díky větší autonomii jednotlivých pracovníků, kterou Scrum nabízí.

Návrhy na zlepšení

Dle mého názoru byly největšími problémy komplexnost aplikace, nové technologie a změna metodologie v průběhu vývoje. Prvním dvěma bodům se obecně nedá nijak vyhnout, naopak s novými technologiemi jako jsou rozšířená či virtuální realita bude k těmto problémům docházet více často. Určitě je ale možné se z předešlých chyb poučit. Při dalším podobném projektu navrhuji výše zmíněné problémy zohlednit již při úvodních schůzkách a také při vytváření cenových a časových odhadů. Agilní vývoj je často spojován s nastavením Time and Material, kdy je na začátku klientovi oznámen pouze hrubý finanční odhad. Zákazník v takovém případě platí většinou na měsíční bázi čas, který byl vývojem stráven. Celková částka tedy není na začátku projektu známa. Je jasné, že takové nastavení spolupráce může být pro některé klienty nepřijatelné, jelikož mají na produkt jasně stanovený rozpočet. Další podmínkou pro toto nastavení je také důvěra mezi zhotovitelem a klientem, aby si mohl být klient jistý, že zhotovitel nebude práce na produktu zbytečně prodlužovat. Netvrdím, že by nastavení typu Time and Material na začátku projektu výrazně ovlivnilo výsledek vývoje, nicméně rozhodně by předešlo několika problémům. Když bylo klientovi oznámeno, že odhadovaný rozpočet nebude stačit, postavili jsme tak klienta do nelehké pozice. Finanční prostředky, které měl na vývoj Aplikace XY na začátku projektu alokované a přesahovaly odhad, již totiž investoval do marketingu. I když byla podepsána smlouva na částku 1.500.000 Kč, popis funkcionalit v příloze této smlouvy neobsahoval ty, které se ukázaly během vývoje jako žádoucí a přinášející přidanou hodnotu uživatelům. V ten moment stál klient před rozhodnutím, zda ponechat všechny funkcionality nebo některé vynechat. V případě, kdy by od začátku byla nastavena spolupráce na systému Time and Material, klient by si jistě v záloze nechal více prostředků a rozhodování by poté nebylo tak obtížné. Proto bych při dalším podobném projektu navrhovala klientovi blíže představit nastavení Time and Material.

Změna metodologie v průběhu vývoje nepřinesla takový výsledek, jaký byl očekáván. Rozhodně bych pro příští projekt doporučila metodologii zvážit ještě před startem vývoje. Je sice pravda, že většina vývojových společností používá pouze jednu metodologii řízení projektů, případně jednu vodopádovou a jednu agilní. Dle mého názoru by to tak být ale nemělo. Zvláště pokud se vývojová společnost řídí projekty dle některé z agilních metodik, měla by sama ke každému novému projektu přistupovat agilně, což znamená, že by měla zvolit metodologii, která bude nejvíce odpovídat potřebám projektu. Pokud by ke Scrumu, který je poměrně jednoduchý na zavedení pro SCRUM tým, přistoupila Společnost XY hned na začátku, věřím, že by projekt dopadl pozitivněji. V době, kdy byl ještě projekt řízen pomocí metodiky SCRUM

totiž došlo při dvěma Sprintům k momentu, kdy na konci Sprintu nedodal tým téměř nic, protože byla práce na začátku Sprintu špatně rozvržena. Kdyby již tehdy postupoval tým podle Scrumbanu, mohli by rychleji na vzniklou situaci reagovat a výsledek by nemusel být tak negativní.

Další mé doporučení se týká plánování. Při vývoji Aplikace XY byla práce při Scrumbanu rozdělena na několik větších částí. Zde bych se ale přiklonila k rozdělení aplikace na menší celky, které je snazší plánovat. Ve SCRUMU se nejprve zvolí délka trvání Sprintu a při plánování se snaží tým do Sprint Backlogu přidat takový počet user stories, který se dá jejich názoru stihnout. Já navrhuji opačný postup. Nejprve odhadnout čas u samostatných user stories, poté si zvolit ty s nejvyšší prioritou a shodnout se na době, za kterou budou hotové tyto konkrétní user stories.

Další mé doporučení navrhuji použít v případě, kdy se blíží Demo. V ten moment by se měl celý tým zaměřit na user stories, které již má rozpracované a nezačínat pracovat na dalších, pokud není sloupec PROBÍHAJÍCÍ prázdný. Výhodou Scrumbanu sice je, že si pracovníci můžou brát user stories z prvního sloupce dle uvážení, pokud není překročen maximální počet položek ve druhém sloupci, vždy by tak ale mělo docházet s ohledem na co možná největší efektivitu. Toto by mělo platit dvojnásob, pokud se blíží Demo s klientem. V tu chvíli by měli všichni členové týmu pracovat na rozpracovaných částech, pokud jim to jejich znalosti jenom trochu umožní.

Posledním mým doporučením, týkající se zvýšení efektivity vývoje pomocí metodiky Scrumban, je více se inspirovat Kanbanem a Scrumban tabulí převést do fyzické podoby. Při vývoji Aplikace XY k tomuto kroku nedošlo především z důvodu, že celý tým je zvyklý pracovat v systému JIRA. Věřím, že přenesení tabule do fyzické podoby by měla lepší efekt na tým z psychologického hlediska. Přetáhnout user stories pomocí myše z jednoho sloupce do druhého není to stejné jako vlastní rukou posunout fyzický lístek po tabuli. Nepřikláním se k myšlence celý Product Backlog přepsat na papírové lístečky, zvolila bych metodu, kdy bude Product Backlog veden v online podobě. Na fyzické tabuli by se zapisovaly pouze názvy user stories, které mají vysokou prioritu. V momentě, kdy začne vývojář na user story pracovat, přesune user story jak na fyzické, tak v online tabuli. Proti tomuto nastavení může být tvrzení, že se jedná o práci navíc. Pokud se ale bude jednat o tým, který sedí u jednoho stolu, fyzickou tabuli můžou mít přímo u sebe a tím pádem se nejedná o téměř žádné zdržení či práci navíc. Při každodenních schůzkách se tým může shromáždit kolem fyzické tabule, která je přehlednější. Zároveň se jedná o „fyzický důkaz“, pokud se některá user story dlouho nachází v prostředním sloupci, což znamená větší motivaci pro tým user story dokončit.

3.4.2 Tým

Přínosy

První přínos pro tým, který byl znát již první týden po přistoupení k Scrumbanu, bylo zmírnění tlaku. Změna metodiky znamenala pro tým v podstatě nový začátek. Tým měl čas se na chvíli zastavit, zhodnotit dosavadní průběh vývoje a zkusit přijít s novým řešením. Fakt, že k názoru změně metodiky došli společně, tým více stmelil a připomněl, že v týmu neexistuje „my“ a „oni“. Bez týmové práce nemůže fungovat ani jedna ze zmíněných metodik.

Druhým přínosem pro tým bylo jistě vyzkoušení si nové metodiky. Pro většinu vývojářů je potřeba se ve své práci stále posouvat a učit se nové věci. Tyto potřeby odlišují výborné vývojáře od těch průměrných. Pokud tedy máme tým, který stále hledá nové výzvy a touha po dalších znalostech je to, co je žene dopředu, je představení nové metodiky to pravé. V týmu tedy díky nové výzvě vzroste motivace, jelikož je nová metodika probudí ze zajetého stereotypu. Přínosem pro organizaci je pak možnost získat nové projekty na základě nového přístupu.

Třetím a posledním přínosem pro tým je ušetření času díky menšímu počtu schůzek. Při přechodu na Scrumban došlo ke zrušení Sprint Planningu a Sprint Retrospektivy, což v případě čtrnáctidenního Sprintu znamená ušetření přibližně šest hodin v rámci Jednoho Sprintu. Tento čas může tým využít pro práci na user stories, což je tím nejdůležitějším pro doručení produktu. Konkrétně při vývoji Aplikace XY bylo ušetřeno zhruba 100 hodin, které netrávil tým na schůzce, což je vysoké číslo. Samozřejmě nějakou část tohoto času tým trávil plánováním, rozhodně se ale nejednalo o tak velký počet hodin.

Návrhy na zlepšení

Jelikož přínosy pro tým byly více zřetelné, nebylo možné přijít s mnoha návrhy na zlepšení. Jeden podstatný návrh bych však ráda zmínila. Tým k závěru, že je potřeba změnit metodiku, došel společně ale jak přesně bude Scrumban probíhat, to vymyslel a zavedl SCRUM Master. Ten byl úkolem pověřen, jelikož zodpovídá za dodržení postupů při použití SCRUMU. Navrhuji, aby příště došlo nejdříve k představení Kanbanu (Pokud tým přestupuje na Scrumban a do té doby používal Kanban, tak k představení SCRUMU). Poté bych vyzval jednotlivé členy týmu, aby rozdělili prvky z obou metodik na dva skupiny – na ty, které chtějí zanechat a které naopak nikoliv. Na závěr by došlo k diskusi celého týmu, jejímž výsledkem by měl být seznam prvků, které si tým přeje zanechat a práce Scrum Mastera by poté bylo přijít s nápadem, jak veškeré návrhy skloubit. Na principu shody celého týmu SCRUM stojí, proto by stejným postupem měl vzniknout i Scrumban. SCRUM i Kanban totiž pracují s myšlenkou, že jednotlivý

pracovníci mají větší pravděpodobnost, že například práci pravděpodobně stihnou v domluveném termínu, pokud si ho určí sami. I v případě, kdy by se termín, který jim byl nakázán, shodoval s tím, který si sami zvolili. Záleží na tom, že pracovníci se k termínu sami zavázali.

Závěr

Cílem této práce bylo propojení dvou agilních metodik projektového řízení, konkrétně SCRUMU a Kanbanu. Přínosem práce pak mělo být prostřednictvím kombinace těchto metodik zefektivnit vývoj.

Prostřednictvím literární rešerše byl nejprve vypracován teoretický základ práce, který posloužil pro vypracování části praktické. V praktické části této diplomové práce pak byl nejdříve nastíněn vývoj mobilní aplikace pomocí metodiky SCRUM, který je běžně používán Společností XY. Poté byla představena konkrétní Aplikace XY, jejíž vývoj započal v polovině roku 2019. Jelikož u Aplikace XY došlo v průběhu vývoje ke změně metodiky ze SCRUMU na kombinaci SCRUMU a Kanbanu, tedy Scrumban, bylo možné poukázat na přínosy kombinace těchto metodik.

Mezi hlavní přínosy, které se po přechodu na Scrumban do projektu promítly, patřila rozhodně schopnost rychleji reagovat na změnu požadavků aplikace a možnost rychleji postupovat v případě výskytu problému. Tato skutečnost byla velmi očividná při přechodu ze SCRUMU na Scrumban. Díky snížení počtu pravidelných schůzek, které SCRUM metodika vyžaduje, měl tým více času pracovat na potřebných funkcionalitách.

Jako další přínos Scrumban metodiky vidím zlepšení psychického rozpoložení celého týmu a zvýšení motivace. K těmto zlepšením došlo díky větší autonomii, kterou Scrumban členům týmu poskytuje. Kvůli absenci pravidelných schůzek, které slouží pro plánování práce, poskytuje Scrumban větší volnost při výběrů user stories, kterým se budou konkrétní pracovníci věnovat. To s sebou přináší větší zodpovědnost členů týmu, která zpravidla zvyšuje motivaci práci včas a kvalitně dokončit. Tato skutečnost pomohla vývoj Aplikace XY zefektivnit prostřednictvím zrychlení dodání jednotlivých user stories a zároveň větší kvalitou kódu.

Jelikož zavedení nové metodiky nepřineslo takové výsledky, jaké byly očekávány, v poslední části práce jsem se zaměřila na několik návrhů na zlepšení, které by měly vývoj dalších mobilních aplikací pomocí metodiky Scrumban v budoucnu ještě více zefektivnit. K důvodům, které vedly k neuspokojivému výsledku, patřila komplexita Aplikace XY a použití nových technologií, se kterými neměli členové týmu zkušenosti. I přes tyto důvody, které je těžké ovlivnit, doporučuji v budoucnu návrhy popsány níže.

Prvním doporučením je plánování dokončení user stories po menších částech. To znamená, že místo stanovení datumů dokončení větších částí aplikace, doporučuji plánování omezit na menší celky, u kterých je jednodušší odhadnout datum dokončení. Tímto způsobem může být docíleno nižší míry pocitu přetížení členů týmu. Pokud jsou totiž členové týmu donuceni k nesplnitelným termínům, vede to k více chybám a demotivaci, která určitě práci neurychlí.

Další návrhem je před schůzkou s klientem, na kterém jsou představeny nové funkcionality, zaměřit veškerou pozornost týmu na user stories, které jsou již rozpracované a nezačínat práci na nových user stories, pokud není jisté, že práce na rozpracovaných stories bude před schůzkou dokončena. Takto můžou být klientovi představeny funkcionality, které jsou stoprocentně funkční a můžou být otestovány uživateli, což přinese užitečnou zpětnou vazbu.

Dále je doporučena větší inspirace metodikou Kanban, konkrétně v případě fyzické tabule. Fyzická tabule, na rozdíl od tabule vedené v některých z vývojových systémů, umožňuje přímou iteraci mezi členy týmů a user stories, a tím výrazněji upozorňuje na user stories, které se dlouhou dobu nachází v prostředním sloupci, což poukazuje na fakt, že práce na nich stále není hotová. Toto řešení však navrhuji uskutečnit pouze v případě, kdy je celý tým alokovan ve společné kanceláři.

Posledním návrhem, který vyplynul z průběhu vývoje Aplikace XY, je zohlednit při zavádění nové metodiky názor všech členů týmu. Diskuze nad procesy a nástroji, které v nové metodice zanechat a které nikoliv, pomůže členům týmům s lepším přechodem na metodiku, která je pro ně nová a zjednoduší tak týmovou adaptaci.

Seznam použité literatury

1. BJÖRKHOLM, Tomas a Jannika BJÖRKHOLM, 2015. Kanban in 30 days. Impactt Publishing. ISBN 1783000902.
2. BRECHNER, Eric, 2015. Agile Project Management with Kanban. Microsoft Press. ISBN 0735698953.
3. DOLEŽAL, Jan, 2016. Projektový management: komplexně, prakticky a podle světových standardů. Praha: Grada Publishing. Expert (Grada). ISBN 978-80-247-5620-2.
4. DOLEŽAL, Jan, Jiří KRÁTKÝ a Ondřej CINGL, 2013. 5 kroků k úspěšnému projektu: 22 šablon klíčových dokumentů a 3 kompletní reálné projekty. Praha: Grada. Management (Grada). ISBN 978-80-247-4631-9.
Upravit citaci
5. DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO, 2012. Projektový management podle IPMA. 2., aktualiz. a dopl. vyd. Praha: Grada. Expert (Grada). ISBN 978-80-247-4275-5.
6. DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO, 2012. Projektový management podle IPMA: 2., aktualizované a doplněné vydání. 2., aktualiz. a dopl. vyd. Praha: Grada. Expert (Grada). ISBN 978-80-247-4275-5.
7. DVOŘÁK, Drahošlav a Martin MAREČEK, 2017. Project Portfolio Management. Brno: Computer Press. ISBN 978-80-251-4893-8.
8. FOWLER, Frederik M., 2018. Navigating Hybrid Scrum Environments: Understanding the Essentials, Avoiding the Pitfalls. Apress. ISBN 1484241630.
9. GOMEZ, Jose, The Difference Between Product and Project Management. Koombea.com [online]. 2019 [cit. 2020-04-10]. Dostupné z: <https://www.koombea.com/blog/the-difference-between-product-and-project-management/>
10. HRAZDILOVÁ BOČKOVÁ, Kateřina, 2016. Projektové řízení: Učebnice [online]. Martin Kolářček - E-knihy jedou [cit. 2020-03-03]. ISBN 978-80-7512-431-9. Dostupné z: https://gumroad.com/l/projektove_rizeni
11. CHIU, Y. C., 2010. An Introduction to the History of Project Management: From the Earliest Times to A.D. 1900. Delft: Eburon Academic Publishers. ISBN 978-90-5972-437-2.
12. KNIBERG, Henrik a Mattias SKARIN, 2010. Kanban and Scrum: making the most of both. C4Media. ISBN 9780557138326.

13. KŘIVÁNEK, Mirko, 2019. Dynamické vedení a řízení projektů: Systémovým myšlením k úspěšným projektům. Praha: Grada Publishing. ISBN 978-80-271-0480-6.
14. LEOPOLD, Klaus a Siegfried KALTENECKER, 2015. Kanban Change Leadership: Creating a Culture of Continuous Improvement. Wiley. ISBN 1119019702.
15. LOCK, Dennis, 2007. The Essentials of Project Management. 3. Aldershot: Gower Publishing Limited. ISBN 978-0-566-08805-6.
16. MAXIMINI, Dominik, 2015. The scrum culture: introducing agile methods in organizations. Cham, Switzerland: Springer. Management for professionals (Springer). ISBN 9783319118260.
17. MCKENNA, Dave, 2016. The Art of Scrum: How Scrum Masters Bind Dev Teams and Unleash Agility. Pennsylvania: Apress. ISBN 9781484222768.
18. MILLER, Tod a Ryan RIPLEY, 2020. Fixing Your Scrum: Practical Solutions to Common Scrum Problems. Pragmatic Bookshelf. ISBN 1680506978.
19. MOLEN, Michiel van der, 2015. Successful Project Sponsorship: A Time-Saver for the Busy Executive. Kogan Page. ISBN 9780749474249.
20. MOREIRA, Mario E, 2017. The Agile Enterprise: Building and Running Agile Organizations. Berkeley, CA: Apress. ISBN 978-148-4223-901.
21. MORRIS, David, 2017. Scrum in easy steps: An ideal framework for agile projects. In Easy Steps Limited. ISBN 9781840787825.
22. MORRIS, Peter W. G., Jeffrey K. PINTO a Jonas SÖDERLUND, 2012. THE OXFORD HANDBOOK OF PROJECT MANAGEMENT. Oxford: OXFORD UNIVERSITY PRESS. ISBN 01-996-5582-0.
23. MORRIS, Peter W. G. a Jeffrey K. PINTO, ed., 2007. The Wiley Guide to Project, Program and Portfolio Management. Hoboken: John Wiley. The Wiley guides to the management of projects. ISBN 978-0-470-22685-8.
24. NĚMEC, Vladimír, 2002. Projektový management. Praha: Grada. Poradce. ISBN 80-247-0392-0.
25. PICHLER, Roman, c2010. Agile product management with Scrum: creating products that customers love. Upper Saddle River, NJ: Addison-Wesley. Addison-Wesley signature series. ISBN 03-216-0578-0.
26. ROBINSON, Nick, 2016. Agile Software Development Essentials [online]. BookRix [cit. 2020-04-10]. ISBN 9783739651743. Dostupné z: <https://www.amazon.co.uk/Agile-Software-Development-Essentials-Robinson-ebook/dp/B01F536YKQ>
27. ROUDIAS, Jihane, 2015. Mastering Principles and Practices in PMBOK, Prince 2, and Scrum: Using Essential Project Management Methods to Deliver Effective and Efficient Projects (FT Press Project Management). Pearson FT Press. ISBN 0134060814.

28. SCHWABER, Ken a Jeff SUTHERLAND, 2017. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game [online]. [cit. 2020-04-10]. Dostupné z: <https://www.scrumguides.org/download.html>
29. ŠOCHOVÁ, Zuzana a Eduard KUNCE, 2019. Agilní metody řízení projektů. 2. vydání. Brno: Computer Press. ISBN 978-80-251-4961-4.
30. WYSOCKI, Robert K., c2009. Effective project management: traditional, agile, extreme. 5th ed. Indianapolis, IN: Wiley Publishing. ISBN 978-0-470-42367-7.
31. WYSOCKI, Robert K., 2019. Effective project management: traditional, agile, extreme, hybrid. 8th ed. Indianapolis, IN: Wiley Publishing. ISBN 978-1-119-56280-1.
32. A guide to the project management body of knowledge, 2017. Sixth Edition. Newtown Square: Project Management Institute. ISBN 978-1628251845.
33. Project Management Guidebook [online], [cit. 2020-04-23]. ISBN 0-473-10445-8. Dostupné z: <https://www.ngoconnect.net/sites/default/files/resources/PROJECT%20MANAGEMENT%20GUIDEBOOK.pdf>
34. Manifesto for Agile Software Development [online], 2001. [cit. 2020-03-05]. Dostupné z: <http://agilemanifesto.org/>

Seznam obrázků

Obrázek 1 - Příklad Kanban tabule	24
Obrázek 2 - OS vývojového oddělení.....	28
Obrázek 3 - Ukázka Scrumban tabule	40

