

<b>Semestrálna práca ku kurzu 4IT421 Zlepšovanie procesov budovania IS</b>	
<b>Semester</b>	LS 2017
<b>Autori</b> <b>-meno, priezvisko, xname</b>	Marián Bača, xbacm08 Jozef Flóro, xfloj07
<b>Téma</b>	Agile developement at enterprise level – Misconceptions That Jeopardize Success
<b>Dátum odovzdania</b>	14.05.2017

## Abstrakt

Cieľom semestrálnej práce je predstaviť, vysvetliť a vyvrátiť mylné predstavy, ktoré panujú pri agilnom vývoji v podnikovom prostredí. Tieto mylné predstavy môžu totiž ohroziť úspech celého procesu vývoja, takže je nevyhnutné ich uviesť na pravú mieru. Práca tiež pojednáva o tom, ktoré prvky sú nevyhnutné k tomu, aby bol agilný vývoj vo veľkom podniku úspešný.

## Kľúčové slová

enterprise, agile, developement, vývoj, podnik, misconceptions, mylné predstavy, biznis analýza

# Obsah

Obsah.....	2
Úvod .....	3
1. Agilný vývoj .....	4
2. Agilný prístup v podnikovom prostredí .....	5
2.1. Fáze projektu v agilných metodikách: .....	5
3. Mylné predstavy o agilnom vývoji .....	6
3.1. Dôležitosť biznis požiadavkou .....	6
3.2. Biznis potreby a užívateľské príbehy .....	7
3.3. Compliance a požiadavky na audit : Užívateľské príbehy stačia .....	8
3.4. Obchodná reorganizácia a agilný prístup: Agilný model drasticky zmení prístup k biznisu.....	9
3.5. Potreba biznis analýz: biznis analýzy nie sú potrebné, zdržujú.....	10
3.6. Požiadavky na softvér: Softvér na riadenie požiadaviek nie je potrebný .....	11
4. Nevyhnutné prvky úspechu agilného vývoje .....	12
Záver .....	14
Zdroje .....	15

# Úvod

Naša práca sa venuje téme agilného vývoja v podniku respektíve mylným predstavám o agilnom vývoji, ktoré ohrozujú podniky a ich projekty. Tieto mylné predstavy môžu totiž ohroziť úspech celého procesu vývoja, takže je potrebné ich uviesť na pravú mieru. Spôsob vývoja softwaru prechádza za posledné roky pomerne významnými zmenami, ktoré sa postupne objavujú s väčším či menším úspechom.

V niekoľkých kapitolách najskôr definujeme, čo je to agilný vývoj a čím je špecifický. Taktiež si viac rozoberieme dôležitosť agilného vývoja v biznise, ťažkosti s nasadením agilnej metodiky na podnikovej úrovni a časté vynechávanie alebo prehliadnutie požiadavkou pri agilnom vývoji. Definujeme a porovnáme si hlavné rozdiely, ktoré spočívajú v tradičných metodikách tzv. vodopádového vývoja oproti agilným metodikám.

V ďalšej kapitole sa práca zaoberá agilným prístupom v podnikovom prostredí. Uvedieme si ako v praxi vypadá agilné riadenie projektov, kedy sa používa a prečo je dôležité aby bol agilný prístup interaktívny a flexibilný v podnikom prostredí.

V hlavnej časti práce si predstavíme mylné predstavy o agilnom vývoji. Budeme sa zaoberať dôležitosťou biznis požiadavkou, potrebami biznisu a užívateľskými príbehmi, obchodnou reorganizáciou a agilným prístupom, požiadavkami o dodržaní predpisov a auditu, potrebou biznis analýz a neposlednom rade požiadavkami na softvér. Na základe analýzy týchto faktorov, ktoré ovplyvňujú úspech celého procesu vývoja sa budeme vedieť vyvarovať chybám pri vývoji a riadení projektov.

V ďalšej časti práce sa budeme venovať prvkom, ktoré sú neoddeliteľnou súčasťou úspechu agilného vývoja. Zavedenie agilných techník je výraznou mentálnou zmenou fungovania prístupu k tvorbe softwaru pre všetky úrovne riadenia projektu, či už pre všetkých členov tímu na strane dodávateľa alebo zákazníka. Vzhľadom k náročnosti a úspechu zavedenia agilných techník do projektu sa po zvládnutí mylných predstáv, ktoré ohrozujú úspech a následná vhodná aplikácia agilných metód, ktorá prináša trvalé odstránenie bariér efektívneho vývoja, povedie k dlhodobo funkčnému softvéru a trvale rastúcej hodnote.

# 1. Agilný vývoj

Softvérové inžinierstvo je disciplína, ktorá sa zaoberá problémami pri vývoji komplexných softvérových systémov. Rozlišujeme tradičné (rigorózne) a agilné (ľahké) metodiky. Do každej z týchto dvoch skupín môžeme zaradiť niekoľko rôznych modelov.

Pred vznikom agilných metodík, ku ktorému oficiálne došlo na prelome 20. a 21. storočia, boli k vývoju softvéru využívané tradičné metodiky projektového riadenia. Pre lepšie pochopenie vzniku agilných metodík, si najskôr definujeme tradičné metodiky vývoja a ich charakteristiky. (Ward & Legorreta, 2010)

Tradičné metodiky vývoja kladú dôraz na presnú špecifikáciu všetkých požiadaviek na začiatku projektu a na ich podrobnú dokumentáciu a zmluvné odsúhlasenie. Koncový zákazník nie je v styku s vývojovým tímom, takže nemôže do procesu vývoja nijako zasahovať. V prípade zmien požiadaviek v priebehu začína spravidla celý proces od začiatku. Vývojový tím sa riadi iba dokumentáciou. Zákazník dostáva objednaný softvér po častiach, alebo celý až na konci.

Moderné trendy ukazujú potrebu rýchlejšieho a flexibilnejšieho vývoja softwaru, s vynaložením čo najnižších nákladov. Klasické metódy si však zakladajú na podrobnom plánovaní, dokumentácií a presnému vymedzeniu požiadaviek pred začiatkom vývoja. Táto nezhoda viedla k potrebe nových metodík, pružnejších a rýchlejších, ktoré by boli schopné reagovať na meniace sa potreby aj v priebehu projektu.

Agilný vývoj je definovaný štyrmi hodnotami a obsahuje dvanásť princípov, ktoré sú v Agilnom manifeste. (Cunningham, Manifest Agilného vývoje software. Dostupné na, 2001) Týmto manifestom oficiálne vznikla zjednotená kategória Agilných metodík vývoja softwaru. Zároveň bola tiež vytvorená Aliancia pre agilný vývoj. Vo vyššie spomenutom manifeste sa zakladatelia zhodli na štyroch hlavných hodnotách agilného vývoja:

- 1. Jednotlivci a interakcie** pred procesmi a nástrojmi
- 2. Fungujúci softvér** pred vyčerpávajúcou dokumentáciou
- 3. Spolupráca so zákazníkom** pred vyjednávaním o zmluve
- 4. Reagovanie na zmeny** pred dodržiavaním plánu

## 2. Agilný prístup v podnikovom prostredí

Agilné projektové riadenie je interaktívny spôsob riadenia projektov. Charakteristické je vytváranie priebežných prototypov a ich upresňovanie a schvaľovanie zákazníkom.

Agilný prístup k riadeniu projektov sa uplatňuje v projektoch, pri ktorých je jasný rámcový cieľ, ale z najrôznejších dôvodov nemožno presne definovať všetky dlhodobé požiadavky bez priebežných prototypov. Používa sa teda, keď nemožno určiť detailný plán projektu vrátane detailných požiadaviek (čo je postup typický pre tradičný, vodopádový prístup). Agilný prístup k riadeniu projektov je interaktívny, pružný a prírastkový. V praxi to znamená tesnú a neustálu (inkrementálnu) spoluprácu medzi projektovým tímom, ktorý vytvára priebežné prototypy a medzi zákazníkom, ktorý dáva spätnú väzbu na základe ktorej sa upresňuje zadanie. Agilné riadenie projektov sa preto uplatňuje u veľmi komplexných systémov u ktorých sa detailné požiadavky tvoria alebo upresňujú priebežne na základe skúseností s prototypmi z jednotlivých iterácií. Pri agilných metódach práce sa realizujú malé porcie výsledkov (prototypy) v každom vývojovom cykle v úzkej spolupráci so zákazníkom.

### 2.1. Fáze projektu v agilných metodikách:

1. Nultá iterácia – prvá krátka analýza a naprogramovanie základných činností.
2. Analýza zmeny – výber toho, čo sa bude implementovať, analýza, dizajn zmien
3. Implementácia požadovanej vlastnosti či zmeny
4. Predvedenie klientovi – skutočne dokončená práca
5. Pokiaľ nie je produkt hotový, späť na bod 2
6. Pokiaľ je produkt hotový, nasleduje údržba a rozvoj produktu

Agilný prístup k riadeniu projektov vyžaduje schopných jednotlivcov, ktorí sú schopní tento spôsob riadenia zvládnuť. Nie je možné ho univerzálne uplatniť vždy, vo všetkých typoch projektov a vo všetkých tímoch. Uplatňuje sa vo vývoji softvéru, ale rovnako tak dobre v ostatných oblastiach, kde je projekt silne inovačný, vyžaduje priebežné korekcie a nápady, kde je možné všetko priebežne komunikovať so zákazníkom. Agilný prístup k riadeniu projektov má blízko k lean technikám a prístupom ako je Kaizen alebo Six Sigma, pretože má silne prozákaznícky charakter, všetky aktivity v priebehu vývoja sú zamerané účelovo na dosiahnutie požadovaného výsledku a tým je minimalizované plytvanie výraznejšie než u tradičných postupov. (Managementmania, 2016)

### 3. Mylné predstavy o agilnom vývoji

Táto kapitola pojednáva o šiestich najbežnejších mylných predstavách, ktoré panujú pri agilnom vývoji v podnikovom prostredí. Každéj mylnej predstave je venovaná jedna kapitola, v ktorej je dôkladne vysvetlená a vyvrátená.

#### 3.1. Dôležitosť biznis požiadaviek

Častou mylnou predstavou je, že biznis požiadavky nie sú potrebné, ale podnikový agilný vývoj neznamená úplný odklon od požiadavkou. Činnosti, ako zisťovanie kritických požiadavkou a rozsahu sú stále potrebné. (Zive, Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success, 2016) Medzi kritické elementy, ktoré potrebné definovať napred, patria:

- **Identifikácia biznis rolí** – pod biznis rolou sa chápe buď stroj alebo osoba. Role sú kolekciami aktivít ktoré umožňujú používateľovi použiť jednu alebo viacero biznis scenárov v organizácii.
- **Vykonávanie analýzy stakeholderov** – v tomto kroku sa vlastne zisťuje, ktoré zainteresované strany v rámci projektu existujú a aké sú ich požiadavky.
- **Určenie potrieb biznisu** – potreby biznisu sú dôležité najmä preto, že určujú, ktoré veci sú dôležité preto, aby podnik bol fungoval úspešne, efektívne a ziskovo.
- **Zabezpečenie financovania projektu** – tento krok je nevyhnutný pre úspešné dokončenie projektu
- **Zachytenie a analýza biznis procesov** – V tomto kroku sa určuje ktorých procesov sa bude zmena týkať a ktoré procesy sa pridajú alebo sa zmenia
- **Definovanie rozsahu riešenia** – rozsah riešenia definuje množinu schopností, ktoré riešenie poskytuje, tak aby pokrývalo určitú potrebu biznisu
- **Vytvorenie a obhájenie obchodného scenára** – dobrý a obhájiteľný obchodný scenár popisuje dôvody a prínosy projektu a je dôležitý pre ďalšie smerovanie projektu

Určitá analýza požiadavkou je potrebná pred začatím projektu a mala by sa začať definíciou požiadavkou na vysokej úrovni. Tieto požiadavky sú potom ďalej dekomponované do väčších detailov a usporiadané (Higgins, 2009) Hore vymenované elementy poskytujú solídny základ projektu a poskytujú práve toľko detailov aby sa určil rozsah riešenia a biznis potreby.

(Leffingwell & Stewart, 2016). Pri tom je potrebné dbať na jednoduchosť, tj. neodkloniť sa od princípov agility. Zostávajúce detaily je potrebné odložiť do ďalších iterácií. V každej iterácii sa potom požiadavky rozpracujú do väčších detailov vo forme scenárov príbehov vizualizácii a modelov. Pri tom je potrebné nezabúdať na princíp „just enough“ – ak je požiadavka dostatočne vysvetlená a spracovaná, nemá zmysel ju rozoberať do väčších detailov. (Higgins, 2009)

### 3.2. Biznis potreby a užívateľské príbehy

Užívateľské príbehy v agilnom vývoji ponúkajú definíciu požiadavkou na vysokej úrovni, ktorá obsahuje práve toľko informácií, koľko je potreba, aby vývojári určili odhady na jej implementáciu. Dôležitými vlastnosťami užívateľských príbehov sú:

- Zákazníci píšu užívateľské príbehy, nie vývojári.
- Užívateľské príbehy popisujú širokú škálu typov požiadavkou.
- Každý užívateľský príbeh by mal obsahovať relatívny indikátor veľkosti, tj. koľko by mala trvať jeho implementácia
- Užívateľský príbeh by mal obsahovať indikátor, ktorý informuje o jeho prioritě
- Voliteľne môže obsahovať jedinečný identifikátor, kebyže sa chce udržať nejaká forma dohľadateľnosti (Ambler, 2003-2014 )

Problém ale spočíva v tom, že užívateľské pohľady ponúkajú pohľad len z jednej perspektívy. V podniku sa nachádza mnoho stakeholderov, napríklad produktoví manažéri, vrcholový management, finančné oddelenie a operatívne oddelenie, a tí majú odlišné potreby a požiadavky. Mnoho z týchto potrieb sú nefunkčné požiadavky a obmedzenia. (Zive, 2016)

Projekty vo veľkých podnikoch sú často veľké a komplikované takže sa v nich pri implementácii užívateľských príbehov naráža na rôzne problémy. BRD dokumenty v podnikoch majú svoju tradíciu a ľudia sú na to zvyknutí, takže nechápu užívateľské príbehy a ich zmysel. Z toho vyplývajú ďalšie problémy ako ich nedostatočná kvalita a dlhý čas dodania. (Matsugu, 2016)

Ďalším problémom užívateľských príbehov je, že im chýba schopnosť abstrakcie – umožniť rozličné pohľady a to buď viacnásobnou abstrakciou získať celkový pohľad, alebo schopnosť

zameriť sa na detail. (Zive, Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success, 2016) Užívateľské príbehy sú totiž textového charakteru a potrebujú byť doplnené vizuálnymi artefaktmi ako modely, mockupy alebo simulácie, aby sa dokázala získať celková predstava o požiadavkách. (Matsugu, 2016)

### 3.3. Compliance a požiadavky na audit : Užívateľské príbehy stačia

Užívateľské príbehy majú len malú pridanú hodnotu, čo sa týka požiadaviek na dodržiavanie smerníc a predpisov (compliance) a požiadaviek na audit a nie sú pre nich úplne dostačujúce. Táto skutočnosť pramení z viacerých príčin.

Prvou príčinou je trvácnosť užívateľských príbehov. Ich životnosť je obmedzená v tom zmysle, že keď sa raz implementujú, tak sa zahodia. Je síce možnosť tieto užívateľské príbehy uchovávať a manažovať ale stále nie je úplne ľahké alebo možné spojiť užívateľský príbeh s určitou vlastnosťou aplikácie. (Zive, Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success, 2016)

Perzistentné a sledovateľné požiadavky sa stali bežnou súčasťou v podnikoch a to hlavne kvôli auditu a compliance. (Zive, Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success, 2016) Sledovateľné požiadavky znamenajú, že je možné spojiť každú požiadavku s jej zdrojom a zároveň s dizajnovým prvkom, zdrojovým kódom a testovacími prípadmi, ktoré majú overiť to, že požiadavka bola správne implementovaná. Tieto požiadavky sú označené jedinečnými identifikátormi a napísané v štruktúrovanej forme a s nízkou granularitou. (Wiegers, 1999)

Častým prístupom podnikov je používanie hybridného prístupu k požiadavkám, ktorý sa pohybuje medzi tradičným prístupom, tj písaním obsiahlych dokumentov BRD a čistým agilným prístupom, ktorý však nestačí, kvôli tomu, že užívateľské príbehy nezachytia nefunkčné požiadavky typu výkon, dostupnosť a bezpečnosť. Podniky často experimentujú medzi týmito dvoma prístupmi a to tak, aby im zvolený prístup čo najviac vyhovoval. (Zive, Why Agile Requirements Tools are the Future, According to Forrester, 2016)

Compliance dokumenty môžu byť vytvárané inkrementálne. Tento prístup vyžaduje jednotný repozitár alebo jednotný zdroj, kde sú dokumenty uložené. Následne je možné tieto



dokumenty regulárne upravovať a reprodukovat'. Na konci iterácii je tak možné si potvrdiť, že výsledná práca pokryla všetky požiadavky a je v súlade so všetkými požiadavkami. (Zive, 2016)

### 3.4. Obchodná reorganizácia a agilný prístup: Agilný model drasticky zmení prístup k biznisu

Mnoho veľkých spoločností dnes úspešne implementovalo agilný prístup vo vývoji a mnoho z nich úspešne implementuje agilné princípy v biznise. Aj keď agilný prístup neznamena drastickú zmenu v prístupe k biznisu, nie je úplne možné sa vyhnúť zmenám. Medzi tieto zmeny patrí napríklad:

- **Zmena organizačnej štruktúry tak, aby bola viac orientovaná na produkt** - so stabilnými tímami a dedikovanými prostriedkami
- **Zlepšenie interakcie medzi biznisom a IT** – proces vývoja je manažovaný silným vlastníkom produktu, ktorý úzko spolupracuje s IT vo všetkých fázach vývoja
- **Predefinovanie rolí v rámci organizácie** – rola projektového manažéra je minimalizovaná a línioví manažéri sa zameriavajú na budovanie schopností.
- **Zmena modelov rozpočtovania a plánovania** – Rozpočtovanie v štýle venture capital, kde sa vpusť do produkcie MVP a ďalšie zdroje závisia od výkonu produktu. (Comella-Dorda, Lohiya, & Speksnijder, 2016)

Benefity škálovania agilného prístupu do biznisu aj na základe týchto štyroch princíпов sú jasné. Pri agilnom vývoji IT a biznis vytvárajú produkty a služby spolu namiesto prostého zbierania požiadaviek a vlastností. Tímy môžu pracovať s MVP a vypúšťať nové vydania nie po mesiacoch alebo rokoch, ale po dňoch a týždňoch. (Comella-Dorda, Lohiya, & Speksnijder, 2016)

Napriek týmto zmenám sa nejedná o drastickú zmenu v prístupe k biznisu. Rozhodovací proces na každom stupni manažmentu zostáva rovnaký, jedinou zmenou je zdroj dát. Napríklad, projektový manažér sa vždy bude snažiť merať postup projektu, porovnávať ho z očakávanými výstupmi a znižovať riziko. Jedinou zmenou pri rozhodovaní bude to, že bude

používať dáta z backlog burn-up namiesto toho, aby použil tradičný WBS. (Zive, Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success, 2016)

### 3.5. Potreba biznis analýz: biznis analýzy nie sú potrebné, zdržujú

V tradičnom vodopádovom modeli majú biznis analytici viacero rolí, medzi ktoré patrí napríklad zisťovanie rozsahu riešenia, preklad biznisových potrieb do jazyka, ktorému rozumejú vývojári, modelovanie a dokumentovanie, byť prostredníkom v komunikácii alebo reprezentácia stakeholderov. (Ambler, Rethinking the Role of Business Analysts: Towards Agile Business Analysts?, 2013-2014) Z tohto vyplýva skutočnosť že biznis analýza nie je len prosté zbieranie požiadaviek ale zahrňuje kritické a strategické myslenie, ktoré smeruje k pochopeniu potrieb biznisu jeho stratégií a jeho cieľom. Je teda nesprávne si myslieť, že biznis analýza pri agilnom vývoji nie je potrebná. (Zive, Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success, 2016)

V kontexte agilného vývoja je rola biznis analytika extrémne dynamická, dokáže sa rýchlo adaptovať a zamerať sa na flexibilitu. Medzi ich činnosti patrí uskutočnenie zlepšenia biznisového procesu, asistovanie pri zbieraní požiadaviek od zákazníka, zabezpečenie toho, že pri testovaní sa použijú najlepšie dostupné postupy, asistovaní pri dizajne projektu a z času na čas zastupovať vlastníka produktu. K tomu patrí definovanie priorit pri implementácii požiadaviek a oprave defektov a vypracovanie konceptov užívateľských príbehov. Biznis analytik musí čo najviac zlepšiť spoluprácu a vybudovať porozumenie medzi vývojovými tímami a klientmi. (Appirio, 2015)

Pri agilnom vývoji v podniku musí biznis analytik stále vedieť zodpovedať na tieto kritické otázky:

- Ktorý pracovný postup sa dá zautomatizovať a ktorý sa má nechať tak ako je?
- Aká je pridaná hodnota toho, keď zmeníme určitú vlastnosť danej aplikácie?
- Aké vzniknú biznisu náklady ak sa nedodá dodávka podľa plánu?
- Ako sa dá zlepšiť spôsob akým biznis funguje automatizáciu niektorých úloh a za akú cenu?
- Ako bude aplikácia vydaná a podporovaná? (Leffingwell & Stewart, 2016)

Aj keď rola biznis analytika je trochu odlišná a môže sa líšiť od tradičných metódik, stále zostáva kľúčovým členom tímu a hrá významnú rolu pri dodávke produktu zákazníkom.

### 3.6. Požiadavky na softvér: Softvér na riadenie požiadaviek nie je potrebný

Najlepšia šanca na úspech je vtedy, keď procesy podnikového agilného vývoja sú podporené aplikáciou na správu požiadaviek, ktorá integruje agilný vývojový tím so zvyškom biznisu.

(Zive, Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success, 2016)

Každý podnik má unikátne požiadavky čo sa týka softvéru na riadenie požiadaviek. Tieto požiadavky závisia na použitej metodike, vyvíjaného softvéru, procesoch v podniku a na firemnej kultúre. Avšak dajú sa nájsť niektoré kľúčové vlastnosti, ktoré sú snád' spoločné pre každý podnik:

- Schopnosť pridávať a upravovať požiadavky, užívateľské príbehy a prípady použitia
- Odvodzovanie detailných technických úloh z požiadaviek pre účely vybudovania časového plánu projektu
- Spojenie akceptačných testov s jednotlivými požiadavkami
- Grafické nástroje na riadenie požiadaviek
- Podpora spolupráce v reálnom čase
- Dashboardy a reporty (Inflectra, dátum neznámy)

Najlepšie je použiť aplikáciu, ktorá je stavaná na riadenie požiadaviek v prostredí podnikového agilného vývoja. To znamená, že bude kompatibilná s podnikovým agilným vývojom a nebude obmedzovať alebo znižovať efektívnosť agilného vývoja, ale naopak ho bude podporovať. (Leffingwell & Stewart, 2016)

## 4. Nevyhnutné prvky úspechu agilného vývoje

Jedným z hlavných kľúčových prvkov agilného vývoja je schopnosť dodávok funkčných verzií systému v krátkych časových intervaloch. Vďaka tomu je možné priebežne meniť požiadavky a zaistiť požadovanú kvalitu výsledku. Vzhľadom ku krátkodobým cyklom je tiež vysokou mierou pravdepodobnosti, že sa dodrží časový plán a náklady. Oproti tomu u tradičných metód je pre dlhú dobu trvania projektu veľká neistota dodržania časového harmonogramu a nákladov. Problematické riešenie zmien v priebehu projektu ďalej znamená neistotu funkčnosti softwaru podľa aktuálnych potrieb klienta, na konci projektu.

Pre agilný vývoj je tiež typický a neoddeliteľný prvok iteratívny a inkrementálny vývoj s krátkymi iteráciami, kedy vývoj prebieha v krátkych cykloch a nové funkcionality sú dodávané postupne. Zákazník tak priebežne získava pridanú hodnotu a môže obratom poskytovať spätnú väzbu a zmeny týkajúce sa ohľadne ďalšieho vývoja a potrebných funkcionalít. Tým sa eliminuje neistota, že na konci dostane niečo, čo vlastne nechcel.

Komunikácia medzi zákazníkom a vývojovým tímom je preto nutná k vývoju softwaru presne podľa potrieb zákazníka a k odstráneniu nedostatkov a prípadných chýb. Zákazník by sa mal tiež podieľať na plánovaní jednotlivých iterácií.

Ďalším faktorom je priebežné testovanie. To je vzhľadom k dynamicky sa meniacemu systému a požiadavkou potrebné vykonávať priebežne a automatizovane. Testovací plán by mal byť pripravený už pred implementáciou novej časti, k eliminácii časového nesúladu. Vždy je potrebné otestovať celý systém, nie len novú implementovanú časť. Pre lepšie pochopenie prvkov agilného vývoja vznikli na základe skúseností princípy agilného vývoja, ktoré sú neoddeliteľnou súčasťou úspešnosti projektu:

1. Najvyššou prioritou je uspokojiť zákazníka včasným a priebežným dodávaním funkčného softwaru. Pre zákazníka má vyššiu hodnotu funkčná časť kódu, než grafický (UML) návrh architektúry a modulov.
2. Sú vítané zmeny požiadavkou a to aj v neskorších fázach vývoje. Agilné procesy podporujú zmeny vedúce k zvýšeniu konkurencieschopnosti zákazníka. Agilná metodika zmeny priamo očakáva, preto sa nepracuje na tom, čo práve nie je potrebné.

3. Dodávať funkčný software v intervaloch týždňov až mesiacov, s preferenciou v kratších periódach.
4. Ľudia z biznisu a vývoja musia spolupracovať denne po celú dobu projektu. Vďaka tomuto je možné v priebehu vývoja postupne upresňovať požiadavky a podľa toho priamo ovplyvňovať vývoj.
5. Vytvoriť projekty pre motivovaných jednotlivcov. Vytvoriť prostredie a podporovať ich potreby na základe dôvery za odvedenú prácu. Ľudský faktor je kľúčom k úspechu.
6. Najefektívnejším a najúčinnejším prvkom úspechu je poskytovanie informácií vývojovému tímu pomocou osobnej komunikácie. Agilný vývoj uprednostňuje osobný kontakt pred písanou komunikáciou.
7. Funkčný software je primárny ukazovateľ úspechu. Nutnosť hodnotenia výsledkov.
8. Agilné procesy propagujú trvalý rozvoj. Sponzori projektu, vývojári a užívatelia by mali byť schopní trvale udržiavať tempo rastu.
9. Agilitu zvyšuje neustála pozornosť venovaná technickej výnimočnosti a dobrému dizajnu. Dizajnu je nutné venovať pozornosť na denne pretože taktiež zvyšuje agilitu.
10. Jednoduchosť – umenie maximalizovať množstvo nevykonanej práce, kľúčový prvok k úspešnému zvládnutiu neustálych zmien je nutné, aby procesy a postupy boli čo najjednoduchšie.
11. Najlepšie architektúry, požiadavky a návrhy vznikajú pomocou samo organizovaného tímu. Dôvera a komunikácia vedie ku kreativite, ktorá je základom dobrých návrhov.
12. Tým v pravidelných intervaloch prehodnocuje ako byť efektívnejší a následne upravuje svoje fungovanie. Je nutné hľadať neustála možnosti na zlepšenie a zefektívnenie. (Cunningham, Principy stojící za Agilním Manifestem., 2001)

Z vyššie uvedených poznatkov vyplýva, že hlavným prvkom úspechu agilného vývoja je čo najrýchlejší vývoj systému podľa predbežných požiadavkou a na základe spätnej väzby od zákazníka systém upraviť do finálnej podoby. Umožniť zmeny je totiž efektívnejšie, než sa im snažiť zabrániť byrokraciou. Čím viac zapojíme zákazníka do procesu plánovania, tým väčšiu pridanú hodnotu pre neho bude mať výsledok na konci každého šprintu. Dokumentácia požiadavkou nehrá zásadnú rolu, oproti tomu schopnosť dynamicky software upravovať podľa potrieb klienta je dôležitou zásadou.

## Záver

V našej práci sme sa snažili zhodnotiť agilný vývoj v podnikovom prostredí a skúmali sme mylné predstavy o agilnom vývoji, ktoré môžu ohroziť úspech podniku. Je dôležité si uvedomiť, že všetky tieto mylné predstavy sú súčasťou takmer všetkých projektov a je potrebné sa im vyvarovať. Na základe analýzy zahrňujúcej štúdium agilného vývoja a mylných predstáv v agilnom vývoji sme sa zhodli na tom, že spôsob vývoja softwaru prechádza za posledné roky pomerne významnými zmenami, no v podnikovom prostredí panujú buď nedostatočné informácie alebo vedomosti k tomu, aby podniky dokázali pružne reagovať na meniace sa podmienky pri vývoji softwaru.

V práci sme definovali rozdiely a špecifiká medzi tradičným a agilným vývojom. Zaoberali sme sa ťažkosťami, ktoré môžu nastať pri zavedení agilnej metodiky na podnikovej úrovni. V neposlednom rade sme sa dozvedeli o problémoch akými sú vynechávanie alebo prehliadnutie požiadavkou pri agilnom vývoji. Uvedomili sme si dôležitosť včasnej identifikácie problémov a nástrahy agilného vývoja v biznise. V ďalšej časti sme definovali najväčšie rozdiely agilného vývoja v porovnaní s vývojom tradičným. Zistili sme, že agilný vývoj je oveľa efektívnejší v konečnom dôsledku pretože je viac interaktívny a flexibilnejší v podnikovom prostredí. Zhodnotili sme ako v praxi vypadá agilný vývoj a jeho potreba pri zavedení do projektov, pri ktorých nejde presne definovať všetky dlhodobé požiadavky a kedy nemôžeme presne určiť detailný plán projektu.

V hlavnej časti sme vymedzili mylné predstavy, ktorými boli dôležitosť biznis požiadavkou, potreby biznisu a užívateľské príbehy, obchodná reorganizácia a agilný prístup, požiadavky o dodržaní predpisov a auditu, potreba biznis analýz a v neposlednom rade požiadavky na softvér. Potvrdilo sa, že je potreba ich brať na vedomie pri úspešnom agilnom vývoji v podnikovom prostredí. Dospeli sme k niekoľkým záverom a to, že agilný vývoj (enterprise agile) sa líši od agilného vývoja napríklad v startupu. V podnikovom prostredí sa musí brať do úvahy „Corporate compliance“. Agilný vývoj na podnikovej úrovni vyžaduje viac ako len user stories a biznis analýza hrá stále kľúčovú rolu. Pre úspešné nasadenie agilného vývoja je potreba zmena myslenia firmy „agile mindset“ pretože agilný prístup je v prvom rade o ľuďoch. Je potreba úprava firemnej kultúry alebo nástrojov podporujúcich vývoj. No v konečnom dôsledku je však nasadenie agilnej metodiky pre podniky veľkým prínosom.

## Zdroje

- Ambler, S. W. (2003-2014 ). *User Stories: An Agile Introduction*. Dostupné na Internete: Agile modeling: <http://www.agilemodeling.com/artifacts/userStory.htm#Introduction>
- Ambler, S. W. (2013-2014). *Rethinking the Role of Business Analysts: Towards Agile Business Analysts?* Dostupné na Internete: Agile Modeling: <http://agilemodeling.com/essays/businessAnalysts.htm>
- Appirio. (2015). *The Busine Analyst's Role in Agile*. Dostupné na Internete: Appirio: <https://appirio.com/tech-blog/the-business-analysts-role-in-agile>
- Comella-Dorda, S., Lohiya, S., & Speksnijder, G. (2016). *An operating model for company-wide agile development*. Dostupné na Internete: McKinsey & Company: <http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/an-operating-model-for-company-wide-agile-development>
- Cunningham, W. (2001). *Manifest Agilního vývoje software*. Dostupné na Internete: Agile Manifesto: <http://agilemanifesto.org/iso/cs/manifesto.html>
- Cunningham, W. (2001). *Principy stojící za Agilním Manifestem*. Dostupné na Internete: Agile manifesto: <http://agilemanifesto.org/iso/cs/principles.html>
- Higgins, T. (2009). *Business Analyst | Authoring Requirements in an Agile World*. Dostupné na Internete: BAtimes: <https://www.batimes.com/articles/authoring-requirements-in-an-agile-world.html>
- Inflectra. (dátum neznámy). *What is Requirements Management?* Dostupné na Internete: Inflectra: <https://www.inflectra.com/spirateam/highlights/understanding-requirements-management-tools.aspx>
- Leffingwell, D., & Stewart, D. (2016). *Misconceptions of Enterprise Agile*. Dostupné na Internete: Modern Analyst: <http://media.modernanalyst.com/7%20Misconceptions%20of%20Enterprise%20Agile.pdf>

- Managementmania. (2016). *Agilní projektové řízení*. Dostupné na Internetu:  
Managementmania: <https://managementmania.com/cs/agilni-projektove-rizeni>
- Matsugu, B. (2016). *Six Reasons User Stories Don't Work in Real-World Enterprise Agile*.  
Dostupné na Internetu: Blueprint Software Systems:  
<http://www.blueprintsys.com/blog/six-reasons-user-stories-dont-work-real-world-enterprise-agile/>
- Ward, C., & Legorreta, L. (2010). *Beyond Waterfall and Agile Methods: Towards a New Contingency Model for IT Project Management*. Dostupné na Internetu:  
[http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1400254](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1400254)
- Wiegers, K. E. (1999). *Writing Quality Requirements*. Dostupné na Internetu: Karl E. Wiegers:  
<http://www.processimpact.com/articles/qualreqs.html>
- Zive, R. (2016). *Agile Development at the Enterprise Level: Misconceptions That Jeopardize Success*. Dostupné na Internetu: InfoQ: [https://www.infoq.com/articles/agile-enterprise-misconceptions?utm\\_source=infoqWeeklyNewsletter&utm\\_medium=WeeklyNL\\_EditorialContent\\_culture-methods&utm\\_campaign=10182016news](https://www.infoq.com/articles/agile-enterprise-misconceptions?utm_source=infoqWeeklyNewsletter&utm_medium=WeeklyNL_EditorialContent_culture-methods&utm_campaign=10182016news)
- Zive, R. (2016). *Why Agile Requirements Tools are the Future, According to Forrester*.  
Dostupné na Internetu: Blueprint: <http://www.blueprintsys.com/blog/why-agile-requirements-tools-are-the-future/>