

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství



Diplomová práce

Projektové řízení ve společnosti RIA.com

Bc. Mariia Ploshchenko

© 2020 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Mariia Ploshchenko

Systémové inženýrství a informatika

Projektové řízení

Název práce

Projektové řízení ve společnosti RIA.com

Název anglicky

Project Management in the RIA.com company

Cíle práce

Cílem diplomové práce je vypracování návrhu zavedení agilního řízení projektů ve společnosti RIA.com.

Metodika

Po nastudování odborné literatury bude proveden sběr dat a poznatků o projektovém řízení ve společnosti RIA.com. Na základě zjištěných dat bude provedena analýza současného stavu a identifikace slabých míst. Na případové studii projektu vývoje portálu Auto.Ria.com budou zjištěny a ověřeny konkrétní typy problémů, které vznikají při realizaci SW projektů a které by bylo možné a vhodné ošetřit právě zavedením agilního řízení. Na základě rozboru budou vypracovány vlastní návrhy na zlepšení stavu se zavedením agilní metodiky řízení (např. SCRUM). Součástí doporučení budou také návrhy na řízení rizik při agilním řízení projektů. Výsledky práce budou průběžně diskutovány v praxi. Literární rešerše, která vznikne souběžně s částí praktickou, bude zaměřena na rozbor teorie projektového řízení v tradičním pojetí a pojetí agilním.

Postup zpracování:

1. Nastudování odborné literatury a sběr dat (březen-květen 2018)
2. Analýza současného stavu ve společnosti RIA.com (září-říjen 2018)
3. Vypracování případové studie (říjen-prosinec 2018)
4. Tvorba vlastních návrhů (únor-květen 2019)
5. Diskuze a ověřování vlastních návrhů v praxi (říjen-prosinec 2019)
6. Sepsání a kompletace literární a praktické části práce (leden-březen 2020)

Doporučený rozsah práce

60 – 70 stran

Klíčová slova

Projektové řízení; Softwarové projekty; Agilní řízení projektů; Agilní metodiky; SCRUM; Projektový tým.

Doporučené zdroje informací

- AXELOS GLOBAL BEST PRACTICE. *PRINCE2 Agile*[®]. Norwich: Axelos, 2015. ISBN 978-0-11-331467-6.
- CINGL, O. – DOLEŽAL, J. – KRÁTKÝ, J. *5 kroků k úspěšnému projektu : 22 šablon klíčových dokumentů a 3 kompletní reálné projekty*. Praha: Grada, 2013. ISBN 978-80-247-4631-9.
- DOLEŽAL, J. – MÁCHAL, P. – LACKO, B. *Projektový management podle IPMA*. Praha: Grada, 2009. ISBN 978-80-247-2848-3.
- KERZNER, H. *Project management : a systems approach to planning, scheduling, and controlling*. Hoboken: John Wiley & Sons, 2013. ISBN 978-1-118-02227-6.
- KNIBERG, H. a M. SKARIN. *Kanban and Scrum making the most of both*. 1.vyd. USA: C4Media, 2010. 120s. ISBN 978-0-5571-3832-6.
- POPPENDIECK, M. – POPPENDIECK, T D. – SCHWABER, K. *Lean software development : an agile toolkit*. Boston: Addison-Wesley, 2003. ISBN 0-321-15078-3.
- PROJECT MANAGEMENT INSTITUTE. *A guide to the project management body of knowledge (PMBOK[®] guide)*. Newtown Square, Pennsylvania: Project Management Institute, 2013. ISBN 978-1-935589-67-9.
- SCHWABER, K. – BEEDLE, M. *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall, 2002. ISBN 0-13-067634-9.
- STELLMAN, A. *Learning Agile*. Oreilly & Associates Inc, 2014. ISBN: 9781449363819.
- SVOZILOVÁ, A. *Projektový management*. Praha: Grada, 2011. ISBN 978-80-247-3611-2.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

doc. Ing. Jan Bartoška, Ph.D.

Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 15. 11. 2019

doc. Ing. Tomáš Šubrt, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 11. 2019

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 08. 03. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Projektové řízení ve společnosti RIA.com" jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne

Poděkování

Ráda bych touto cestou poděkovala doc. Ing. Janu Bartoškovi, Ph.D. za odborné vedení mé diplomové práce, cenné rady a věnovaný čas.

Chci také poděkovat společnosti RIA.com a vývojovému týmu AUTO.RIA za poskytnutí možnosti spolupráce a potřebné podklady.

Projektové řízení ve společnosti RIA.com

Abstrakt

Diplomová práce se věnuje popisu metodik řízení projektů, používaných pro vývoj softwaru. Je zde vysvětlen Vodopádový model vývoje softwaru, principy Agile a vybrané agilní metodiky Kanban, Lean Software Development, Extrémní programování a Scrum.

Cílem je vypracování návrhu zavádění agilního řízení ve společnosti RIA.com.

Na základě studia odborné literatury tým AUTO.RIA zvolil Scrum jako metodiku, vhodnou pro vývoj. Z tohoto důvodu byla v teoretické části práce Scrumu věnována větší pozornost. Podrobně jsou popsány jednotlivé praktiky, schůzky a role, které by měl tým dodržovat pro úspěšnou aplikaci Scrumu.

V praktické části práce je charakterizována vybraná společnost RIA.com, je popsán současný stav a nástroje řízení projektů, jsou zjištěny problémy projektového týmu AUTO.RIA při vývoji softwaru.

Součástí práce je vypracování konkrétních návrhů na zavádění Scrumu do vývojového týmu AUTO.RIA. Dále je provedena analýza software pro podporu agilního řízení, identifikace a řízení rizik. K získání potřebných dat jsou použity vlastní pozorování, rozhovory s manažery a zaměstnanci, dotazníkové šetření.

Vlastní návrhy a doporučení jsou prezentovány vedení společnosti RIA.com.

Klíčová slova: Projektové řízení; Softwarové projekty; Vodopádový model; Agilní řízení projektů; Agilní metodiky; SCRUM; Scrum master; Projektový tým; Scrum board; Trello.

Project Management in the RIA.com company

Abstract

The thesis is devoted to the description of project management methodologies used for software development. It describes the Waterfall model of software development, principles of Agile and selected agile methodologies Kanban, Lean Software Development, Extreme Programming and Scrum.

The aim is to develop a proposal for the introduction of agile management in RIA.com.

Based on the study of literature, the AUTO.RIA team chose Scrum as a methodology suitable for development. For this reason, Scrum was given more attention in the theoretical part of the thesis. Individual practices, meetings and roles that a team should follow for a successful Scrum application are described in detail.

In the practical part of the thesis, the selected company RIA.com is characterized, the current status and tools of the project management are described, problems of the AUTO.RIA project team during software development are identified.

Part of the work is to elaborate specific proposals for introducing Scrum into the AUTO.RIA development team. Furthermore, analysis of software to support agile management, identification and risk management is performed. To obtain necessary data are used own observations, interviews with managers and employees, questionnaire survey.

Own suggestions and recommendations are presented to RIA.com management.

Keywords: Project management; Software projects; Waterfall model; Agile project management; Agile methodologies; SCRUM; Scrum master; Project team; Scrum board; Trello.

Obsah

1 Úvod.....	11
2 Cíl práce a metodika.....	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Základy projektového řízení	13
3.2 Projektové řízení vývoje softwaru	17
3.3 Vodopádový model vývoje softwaru	20
3.4 Agilní metodiky vývoje softwaru.....	24
3.4.1 Lean	28
3.4.2 Extrémní programování	29
3.4.3 Kanban	31
3.4.4 Scrum	33
3.5 Detailní popis vybrané metodiky Scrum.....	33
3.5.1 Scrum Team	34
3.5.2 Scrum artefakty	37
3.5.3 Scrum schůzky	40
3.6 Srovnání Agilního a Vodopádového přístupu.....	46
4 Popis a rozbor vybrané organizace	49
4.1 Představení společnosti RIA.com	49
4.2 Popis internetového portálu AUTO.RIA.com.....	50
4.3 Projektové řízení ve společnosti RIA.com.....	51
4.3.1 Organizační struktura společnosti.....	51
4.3.2 Řídící struktura společnosti	52
4.3.3 Metodika a nástroje řízení projektů	56
4.4 Zjištění problémů projektového týmu při vývoje softwaru.....	59
5 Vlastní výzkum a dosažené výsledky	68
5.1 Zavádění Scrumu do vývojového týmu AUTO.RIA	68
5.2 Zhodnocení přechodu na Scrum.....	79
5.3 Podpora agilního řízení týmu	85
5.4 Řízení rizik při zavádění Scrumu	90
6 Závěr	95
7 Seznam použitých zdrojů	98
8 Přílohy.....	101

Seznam obrázků

Obrázek 1. Projektový trojimperativ	14
Obrázek 2. Projektová fáze životního cyklu projektu.....	15
Obrázek 3. Fáze Vodopádového modelu.....	20
Obrázek 4. Kanban tabule.....	32
Obrázek 5. Scrum Team	34
Obrázek 6. Burndown graf.....	38
Obrázek 7. Scrum proces vývoje softwaru	45
Obrázek 8. Projektový trojimperativ	47
Obrázek 9. Organizační struktura společnosti RIA.com	52
Obrázek 10. Řídící struktura společnosti RIA.com	53
Obrázek 11. Životní cyklus projektu ve společnosti RIA.com.....	56
Obrázek 12. Dotazník sebehodnocení	60
Obrázek 13. Počet let ve společnosti RIA.com (členové týmu).....	61
Obrázek 14. Hodnocení spokojenosti členů týmu AUTO.RIA	62
Obrázek 15. Dotazník hodnocení projektového týmu AUTO.RIA	63
Obrázek 16. Počet let ve společnosti RIA.com (kolegové)	64
Obrázek 17. Hodnocení spokojenosti kolegů	65
Obrázek 18. Scrum board v systému JIRA.....	87
Obrázek 19. Scrum board v Trello.....	88
Obrázek 20. Myšlenková mapa	90
Obrázek 21. Matice rizik	92

Seznam tabulek

Tabulka 1. Cíle a výstupy jednotlivých fází Vodopádového modelu.....	21
Tabulka 2. Výhody a nevýhody Vodopádového modelu	22
Tabulka 3. Výhody a nevýhody agilních metodik vývoje softwaru.....	27
Tabulka 4. Srovnání Agilního a Vodopádového přístupu	47
Tabulka 5. Dodržování agilních principů při řízení projektu AUTO.RIA.	84
Tabulka 6. Požadavky na software od zaměstnanců.....	85
Tabulka 7. Hodnocení rizik	91
Tabulka 8. Monitorování rizik	93

Seznam příloh

Příloha A. Registr rizik	101
Příloha B. Obrana vůči rizikům	104

1 Úvod

Diplomová práce je zaměřena na prostředí softwarového inženýrství, které je kvůli rychlému vývoji technologii velice turbulentní. Softwarové projekty jsou dynamické, s nejasnými požadavky, mají specifické podmínky, vyžadují včasnou reakci na změny a neustálou inovaci. Pro řízení těchto specifických projektů se vytvářejí speciální metody a přístupy. Diplomová práce popisuje Vodopádový model vývoje softwaru, principy Agile a vybrané agilní metodiky Kanban, Lean Software Development, Extrémní programování a Scrum.

Cílem diplomové práce je vypracování návrhu zavádění agilního řízení do vývojového týmu AUTO.RIA. V společnosti RIA.com se jedná o první projekt, který bude takto řízen.

Na základě studia odborné literatury tým AUTO.RIA zvolil Scrum jako metodiku, vhodnou pro vývoj. Důvodem bylo to, že Scrum umožňuje lépe organizovat práce v týmu, rychle reagovat na změny, dodávat zákazníkovi hodnotu v pravidelných iteracích, sledovat chod práce na projektu do detailů. Z tohoto důvodu v teoretické části práce byla Scrumu věnována větší pozornost. Podrobně jsou popsány jednotlivé praktiky, schůzky a role, které by měl tým dodržovat pro úspěšnou aplikaci Scrumu. Poté se práce zaměřuje na srovnání Agilního a Vodopádového přístupu k řízení projektů, určení jejich výhod a nevýhod. Proto jsou popsány základní charakteristiky projektového řízení, životní cyklus, mezinárodní standardy a vodopádový model projektového řízení.

V praktické části práce je charakterizována vybraná společnost RIA.com, její organizační struktura a portfolio projektů, je popsán současný stav a nástroje projektového řízení, jsou zjištěny problémy týmu AUTO.RIA při vývoje softwaru.

Součástí práce je vypracování konkrétních návrhů na zavádění Scrumu do vývojového týmu AUTO.RIA. K získání potřebných dat jsou použity vlastní pozorování, rozhovory s manažery a zaměstnanci, dotazníkové šetření. Dále je provedena analýza softwaru pro podporu agilního řízení, identifikace a řízení rizik.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem diplomové práce je vypracování návrhu zavádění agilního řízení projektů ve společnosti RIA.com.

Teoretická část obsahuje popis a rozbor vybraných agilních metodik se zaměřením na metodiku Scrum. V praktické části jsou využity získané znalosti z teoretické části a znalosti aktuálního stavu projektového řízení ve společnosti RIA.com. Na základě těchto informací jsou vypracovány vlastní návrhy na implementaci Scrumu do týmu AUTO.RIA.

2.2 Metodika

Literární rešerše bude zaměřena na rozbor teorie projektového řízení v tradičním pojetí a pojetí agilním. Po nastudování odborné literatury bude proveden sběr dat a poznatků o projektovém řízení ve společnosti RIA.com. Na základě zjištěných dat bude provedena analýza současného stavu a identifikace slabých míst.

V praktické části budou zjištěny a ověřeny konkrétní typy problémů, které vznikají při vývoji portálu auto.ria.com a které by bylo možné vyřešit právě zavedením agilního řízení. K získání potřebných dat budou použity vlastní pozorování, rozhovory s manažery a zaměstnanci, dotazníkové šetření. Na základě rozboru budou vypracovány vlastní návrhy na zlepšení stavu se zavedením Scrumu. Součástí doporučení budou také návrhy na řízení rizik při agilním řízení projektů. Výsledky práce budou průběžně diskutovány a prezentovány vedení společnosti RIA.com.

Harmonogram zpracování:

1. Nastudování odborné literatury a sběr dat (březen-květen 2018)
2. Analýza současného stavu ve společnosti RIA.com (září-říjen 2018)
3. Vypracování případové studie (říjen-prosinec 2018)
4. Tvorba vlastních návrhů (únor-květen 2019)
5. Diskuse a ověřování vlastních návrhů v praxi (říjen-prosinec 2019)
6. Sepsání a kompletace literární a praktické části práce (leden-březen 2020)

3 Teoretická východiska

3.1 Základy projektového řízení

Podle profesora H.Kerznera, **projektové řízení** je souhrn aktivit, spočívající v plánování, organizování, řízení a kontrole zdrojů společnosti s relativně krátkodobým cílem, který byl stanoven pro realizaci specifických cílů a záměrů. [11]

Podle vyjádření J. Doležala, projektové řízení dnes nepředstavuje soupis norem, nařízení či směrnic, ale spíše všeobecná doporučení a nejvhodnější postupy. Moderní principy projektového řízení jsou systémový přístup, strukturování problému, přiměřené prostředky, týmová práce, aplikace zásad trvalého zlepšování, integrace. [6]

Projekt je předmětem projektového řízení. Podle H.Kerznera může být za projekt považován jakýkoliv soubor aktivit či úkolů, který:

- má konkrétní cíl, jehož naplnění by mělo být dosaženo v rámci určitých specifikací;
- má vymezená data zahájení a ukončení;
- má své limity financování;
- čerpá lidské a nelidské zdroje (např. lidé, vybavení);
- a prochází přes řadu funkčních linií. [11]

PMI klade důraz především na dočasnost a unikátnost a definuje takto: „*Projekt je dočasná snaha o vytvoření unikátního produktu, služby nebo jiného výsledku. Dočasná povaha je vyjádřena definovaným začátkem a koncem. Koncem se rozumí dosažení projektových cílů nebo skutečnost, kdy dojde k ukončení projektu, protože cíle nebudou či nemohou být naplněny, anebo také když zanikne účel projektu.*“ [18]

Podle definice standardu IPMA® ICB v3.1: „*Projekt je jedinečný časově, nákladově a zdrojově omezený proces realizovaný za účelem vytvoření definovaných výstupů (rozsah naplnění projektových cílů) v požadované kvalitě a v souladu s platnými standardy a odsouhlasenými požadavky.*“ [6]

Správně definovaný cíl projektu je jedním z klíčových faktorů jeho úspěchu. Metoda SMART je jedním z nejpoužívanějších a nejefektivnějších nástrojů pro tvorbu realistických a dosažitelných cílů. Podle SMART cíl by měl splňovat následující charakteristiky:

- S (Specific) - cíl musí být definován přesně;
- M (Measurable) - splnění cíle musí být možné změřit;

A (Accepted) - cíl musí být akceptovaný odpovědnou osobou;

R (Realistic) - cíl musí být reálný;

T (Time Specific) - cíl musí mít daný termín.

Z definice projektu vyplývá, že jeho úspěch je měřen třemi parametry, tedy čas (schedule – dodržení harmonogramu), náklady/zdroje (cost – dodržení rozpočtu), rozsah (scope – dodržení kvality výstupů). **Projektový trojimperativ** graficky znázorňuje tato omezení. Na vrcholech trojimperativu jsou parametry projektu, a uvnitř leží jeho specifický cíl.

Obrázek 1. Projektový trojimperativ



Zdroj: vlastní zpracování dle [3]

Pro úspěšné dokončení projektu platí, že dynamický systém “čas-náklady-rozsah” musí být udržován v rovnováze. Lze projekt považovat za úspěšný, pokud:

- je projekt funkční;
- je výstupní produkt projektu na trhu včas;
- je výstupní produkt v plánované jakosti a ceně;
- jsou splněny požadavky a uspokojena očekávání všech zainteresovaných stran;
- je dosahována předpokládaná návratnost vložených prostředků. [6]

Udržení projektového trojimperativu v rovnováze je největším uměním projektového manažera. V praxi se jedná o řešení optimalizačního problému s ohledem na význam jednotlivých parametrů projektu pro zákazníka. Nicméně, zákazník vždycky chce získat požadovaný výsledek v nejlepší možné kvalitě, v co nejkratším čase a za nejnižší náklady. Proto projektový manažer, jako osoba zodpovědná za dosažení projektových cílů, musí

dohodnout se zákazníkem nejdůležitější parametr v tomto trojúhelníku. Pokud je nejdůležitějším parametrem čas, musí se vhodně upravit původní rozsah a/nebo náklady tak, aby byl dodržen harmonogram a zároveň naplněny požadavky zákazníka.

Každý projekt má konkrétní datum zahájení a datum ukončení. Mezi těmito dvěma daty projekt prochází životním cyklem. Podle PMBOK je **životní cyklus projektu** souborem obecně následných fází projektu, jejichž názvy a počet jsou určeny potřebami kontroly organizace, která je v projektu angažována. [18]

Podle definice J. Doležala životní cyklus projektu má tři základní fáze.

Předprojektová fáze – má za účel prozkoumat příležitost pro projekt a posoudit proveditelnost daného záměru. Tato fáze má obecně dát odpověď na strategické otázky projektu – odkud projekt směřuje, kam chce dojít, jakou cestu je vhodné zvolit a zda má smysl projekt vůbec realizovat. Vstupy jsou situace v podniku, předpokládaný vývoj trhu apod. Výstupem je finální rozhodnutí o zahájení projektu. [6]

Projektová fáze – je nejobsáhlejší a zaměřuje se na vlastní řízení projektu. Obvykle je rozdělena na etapy Zahájení, Realizace a Ukončení projektu.

Obrázek 2. Projektová fáze životního cyklu projektu



Zdroj: vlastní zpracování dle [29]

1. zahájení projektu – v souladu s předchozími událostmi dochází k definování a ověření cíle projektu, jmenování projektového týmu, upřesnění požadovaných výstupů;
2. realizace projektu – projektový tým definuje rozsah projektu (např. formou WBS), vytváří projektový plán, identifikuje činnosti k realizaci a vytvoří harmonogram, monitoruje průběh projektu a porovnává s plánem, provádí korekční opatření v případě odchylek;

3. ukončení projektu – dochází k předání výstupů, podpisu akceptačních protokolů, fakturaci. Zpracovává se závěrečná zpráva o projektu, která obsahuje souhrn zkušeností z realizace, doporučení, rady a tipy do dalších projektů. [6]

Po akceptaci se výstup projektu dostává do provozu, který již obvykle není součástí projektu.

Poprojektová fáze – v této fázi dochází k analyzování průběhu projektu, určených dobrých i špatných zkušeností, vyhodnocení naplnění cílů projektu. [6]

Fáze životního cyklu projektu nejsou fixní a závisí na typu konkrétního projektu a jeho složitosti. Každopádně tyto fáze se oddělují určitými cíli, průběžnými výstupy a jejich význam umožňuje lépe kontrolovat průběh, sledovat hlavní ukazatele a finanční rizika projektu.

Mezi hlavní světové standardy projektového řízení patří PMI, IPMA, PRINCE2.

Standard PMI a jeho základní dokument PMBOK Guide je považován za nejstarší a nejobecnější standard projektového řízení. PMBOK popisuje 47 logicky seskupených procesů projektového řízení, který jsou zařazeny do 5 procesních skupin: přípravné procesy, procesy pro plánování, proces realizace, dozorčí a kontrolní proces, ukončovací procesy. Popis každé skupiny je velice podrobný, včetně doporučených nástrojů a technik, vstupů a výstupů každého procesu (zejména dokumentace), role projektového manažera a organizační struktury. Celkový PM je velice formalizován, důraz je kladen na kvalitu plánování a vedení podrobné projektové dokumentace. Mezi základní dokumenty projektu PMI řadí *Základní listinu projektu* (Project Charter – formální vymezení projektu: vize, cíle, příležitosti, rizika, zdroje), *Rozsah a oblast projektu* (Project Scope Statement (Logický rámec, WBS) – vymezuje stav, kterého má být dosaženo), *Plán řízení projektu* (Project Management Plan (Harmonogram) – uvádí, jak bude práce prováděna). [18]

IPMA (International Project Management Association) je celosvětovým standardem, který definuje kompetence (schopnosti, znalosti, dovednosti) projektového manažera, požadované pro úspěšné řízení projektů, programů a projektového portfolia. [8] Kompetencí se rozumí demonstrováná znalost, schopnost aplikovat a také demonstrovat osobní vlastnosti. [6] Významy různých kompetencí se liší podle typu projektů (výstavba, obchodní služby nebo veřejný sektor apod.). Kompetence se dělí na kontextové kompetence (zabývají se veškerými souvislostmi projektu), behaviorální kompetence (chování a jednání projektového manažera v rámci týmu a se stakeholdery), technické kompetence (metody a postupy, které jsou potřebné pro úspěšné řízení projektů). Zároveň

IPMA skrze své národní asociace tvoří komunitu projektových manažerů napříč všemi obory, která propaguje a využívá nové přístupy v projektovém řízení. [7]

PRINCE2 (zkratka vzniklá z PProjects IN Controlled Environments 2nd Version) je metodika pro projektové řízení pocházející původně z Anglie. Stojí na 7 principech propojených 7 procesy a popisuje 7 témat. Jde o procesně a projektově orientovanou metodiku řízení projektů, složenou z nejlepších postupů ověřených v praxi, založenou na definici procesů a odpovědností. Pro každý projekt je třeba definovat rozsah projektu, jeho dodávky a produkty v požadované kvalitě, na tyto výstupy stanovit rozpočet a časový harmonogram. Základní charakteristikou projektu je jeho jedinečnost, nejistota a změna, proto každý projekt provází různá rizika, která je třeba definovat, počítat s nimi, plánovat protipatření a v průběhu projektu je aktualizovat.

Základním kamenem metodiky PRINCE2 je možnost přizpůsobit metodiku konkrétnímu projektu v rámci specifického prostředí. Dnes PRINCE2 je jednou z nejrozšířenějších metodik projektového řízení na světě. Představuje oficiální metodiku projektového řízení ve veřejné správě ve Velké Británii, Francii a dalších zemích. [14]

Standardy mají podobnou základní filozofii, většinou jde jen o jiný úhel pohledu na tutéž oblast. Liší se místem vzniku, podkladem, ze kterého byly vytvořeny, i způsobem zpracování. Standardy mají obrovský přínos v tom, že si pracovníci na projektech dokážou vzájemně porozumět, pochopit se a efektivně spolupracovat. [6]

3.2 Projektové řízení vývoje softwaru

V dnešní době projektové řízení zahrnuje spousty různých nástrojů, praktik a metod, které umožňují naplánovat a realizovat kterýkoliv projekt.

Software je dnes všude okolo nás a svěřujeme mu důležité informace, své peníze a někdy dokonce i své zdraví či životy. Proto jsou kladeny stále větší požadavky na jeho správnost, spolehlivost a bezpečnost. V procesu vývoje softwaru se potřeby uživatele transformují na požadavky na software, na jejichž základě je vytvořen návrh systému, podle něhož je softwarový systém implementován, otestován, a nakonec je předán uživateli. Řízení vývoje softwaru je specifické kvůli pravděpodobnosti významných změn požadavků a očekávání. Před projektovým manažerem stojí vždy nelehký úkol, protože musí najít rovnováhu mezi mnoha často vzájemně se vylučujícími cíli, riziky a zároveň uspokojit v maximální možné míře požadavky zákazníka a koncových uživatelů. [30]

Proces vývoje softwaru definuje kdo, kdy a co má dělat, aby bylo dosaženo požadovaného cíle. První stranou je zákazník, který sponzoruje vývoj softwaru (má na zavedení softwaru zájem) a specifikuje požadavky. Druhou stranou je dodavatel, který systém vyvíjí. Posledním aktérem jsou uživatelé, kteří upřesňují požadavky na systém a kteří systém budou používat. Uživatelé nemají vždy zájem na zavedení nového softwarového systému (na rozdíl od zákazníka). Typickým důvodem je, že se uživatelé nového systému bojí – budou se muset naučit něco nového a nejsou si jisti, že to zvládnou. Když uživatelé nemají zájem na úspěchu projektu, budou se snažit jeho vývoj všemožně brzdit, například vymyšlením nesmyslných požadavků (v úvodních fázích) či neobjektivní kritikou softwaru (při přebírání). V takovém případě je dodavatel v nezáviděníhodné situaci a úspěch projektu je značně nejistý. [30]

Životní cyklus vývoje softwaru se obvykle sestává z následujících etap.

1. etapa: Analýza a specifikace požadavků. Je úvodní etapou při vývoji, ve které jsou definovány, analyzovány a specifikovány požadavky zákazníka na software. Výstupy této etapy jsou analýza možných rizik a plán akceptačního testování (testy, na jejichž základě zákazník software akceptuje). [30]

2. etapa: Architektonický a Podrobný návrh. Architektonický návrh – slouží k ujasnění koncepce systému a k jeho dekompozici, tzn. vymezení funkcionality jednotlivých subsystémů a definování vztahů mezi nimi. Výstupem je plán testování celého systému (tedy správnost integrace jednotlivých subsystémů do jednoho celku). Je vhodné také naplánovat postup nasazení systému do provozu, a to včetně zaškolení uživatelů. [30]

Podrobný návrh se soustřeďuje na podrobnou specifikaci softwarových součástí, na výběr algoritmů realizujících požadované funkce, na stanovení logické a fyzické struktury dat a na způsoby ošetřování chybových a neočekávaných stavů. Při podrobném návrhu se také plánují práce na implementaci součástí, s čímž souvisí vytvoření požadavků na lidské zdroje a odhad doby trvání a nákladů na projekt. Výstupem detailního návrhu by měl být také návrh testů součástí, a to včetně testovacích dat. [30]

3. etapa: Implementace. Zahrnuje programovou realizaci softwarových součástí, vypracování dokumentace a testování implementovaných součástí. [30]

4. etapa: Integrace a testování. Po implementaci a testování jednotlivých součástí nastupuje testování celého systému, při kterém se objeví chyby, které nebylo možné odhalit izolovaným testováním samostatných součástí systému. Objevené chyby jsou

samozejmě opravovány, protože nelze vynechat s odkazem na to, že už součásti otestovány byly, protože opravou nalezených chyb se do programu mohou zanést chyby nové. [30]

Akceptační testování a instalace spočívá v testování systému uživatelem. Na základě akceptačního testování se pak zákazník rozhodne buď systém převzít nebo při objevení závažnějších nedostatků softwaru je převzetí odloženo do doby, než dodavatel nedostatky opraví. Po akceptování systému uživatelem (zákazníkem) začíná instalace softwarového systému u zákazníka a školení uživatelů. [30]

5. etapa: Provoz a údržba. Provoz a údržba vyžaduje průběžné řešení problémů, které s nasazením používáním softwaru souvisejí. Tato etapa také zahrnuje opravy nalezených chyb, rozšiřování softwaru o nové funkce či přizpůsobování softwaru měnícím se požadavkům okolí. [30]

Jak rostla složitost vyvíjených softwarových systémů, objevovalo se stále více problémů s vývojem softwaru – softwarové projekty se neúnosně prodražovaly, zpožďovaly a výsledný produkt nesplňoval očekávání zadavatelů. Kvalita vyvinutého softwaru je určena třemi základními faktory – cenou, časem a splněním požadavků. Tyto faktory jdou však v principu proti sobě, a proto se při tvorbě softwaru vyskytují následující problémy:

1. neúplnost nebo nejasnost požadavků (13,1 %)
2. nedostatek zájmu a podpory ze strany uživatele (12,4 %)
3. nedostatek zdrojů, tj. podhodnocený rozpočet a krátké termíny (10,6 %)
4. nerealistické očekávání (9,9 %)
5. malá podpora od vedení dodavatele nebo odběratele (9,3 %)
6. změna požadavků a specifikace (8,7 %)
7. nedostatečné plánování (8,1 %)
8. vyvíjený systém už není potřeba (7,5 %). [30]

K úspěšnosti softwarových projektů přispívají zájem uživatelů, podpora vedení zákazníka, jasně definované požadavky, dobré plánování, realistické očekávání, správná dekompozice úlohy, kompetentnost zúčastněných. [30]

3.3 Vodopádový model vývoje softwaru

V rámci vývoje softwaru je možné v současnosti sledovat využívání dvou zcela odlišných přístupů k řízení projektů. Na jedné straně je to klasický přístup zastoupený tradičními projektovými metodikami, které jsou procesně orientované a z hlediska jednotlivých procesů detailně popisují vstupy, výstupy, nástroje a metody. Na druhé straně je to přístup agilní, který před procesy a nástroji upřednostňuje projektové výsledky, práci týmu a plánování.

Základním a nejstarším modelem životního cyklu software je Vodopádový model. Životní cyklus tohoto modelu připomíná vodopád (jednotlivé etapy jsou seřazeny za sebou), z čehož vzešel i název – Waterfall nebo Vodopádový model. [30] První zmínka o tomto procesním modelu je spjata s Winstonem W. Roycem, který tento proces kritizoval a označil jej za nefunkční. Přesto Vodopádový model v době svého vzniku představoval obrovský pokrok, hlavně díky rozdělení procesu do jednotlivých fází. Vodopádový model je postaven na myšlence, že „vývoj software se skládá z několika základních fází, které musí následovat jedna po druhé“. [6]

Obrázek 3. Fáze Vodopádového modelu



Zdroj: vlastní zpracování dle [6]

Vodopádový model se skládá z 7 fází, přičemž každá fáze se soustředí na konkrétní soubor aktivit a projektových procesů.

Tabulka 1. Cíle a výstupy jednotlivých fází Vodopádového modelu

	Fáze	Cíle	Výstupy
1.	Specifikace požadavků	<ul style="list-style-type: none"> – definovat cíl projektu a způsoby jeho dosažení – pochopit záměr zákazníka: k čemu bude daný systém používán, komu je určen, v čem má usnadnit práci nebo co má nahradit – popsat do nejmenších detailů potřeby a požadavky zákazníka – definovat termíny dokončení prací – určit rozpočet 	<ul style="list-style-type: none"> – Úvodní studie, která shrnuje informace o zákazníkovi, důvody dodání a hlavní potřeby a požadavky
2.	Analýza a návrh	<ul style="list-style-type: none"> – definovat celkový rozsah systému – určit technické detaily a navrhnout architekturu – odsouhlasit přesný popis systému a jeho funkcionalit 	<ul style="list-style-type: none"> – Návrh, který obsahuje přesný popis systému a jeho funkcionalit – Náčrtky budoucího uživatelského rozhraní
3.	Implementace	<ul style="list-style-type: none"> – Vývoj systému v podobě, která byla odsouhlasena 	<ul style="list-style-type: none"> – naprogramovaný systém
4.	Integrace	<ul style="list-style-type: none"> – integrační testy, které mají za úkol odhalit a posléze odstranit vzniklé chyby 	<ul style="list-style-type: none"> – testování proběhlo komplexně – systém se choval podle stanoveného popisu
5.	Testování	<ul style="list-style-type: none"> – uživatelské testování, které by mělo odhalit chyby či odchylky chování 	<ul style="list-style-type: none"> – ověřený systém je připraven k spuštění
6.	Instalace	<ul style="list-style-type: none"> – úspěšné spuštění systému pro všechny uživatele – podpora provozuschopnosti, odstranění vad a chyb – získání zpětné vazby od uživatelů 	<ul style="list-style-type: none"> – funkční systém – uživatele by neměl překvapit žádný defekt
7.	Údržba	<ul style="list-style-type: none"> – udržovat produkt ve funkčním stavu a případně jej s dalším vývojem vylepšovat 	<ul style="list-style-type: none"> – plánování budoucích vylepšení

Zdroj: vlastní zpracování dle [6]

V průběhu jednotlivých fází postupuje projektový tým pod vedením projektového manažera, dle projektového plánu, tak, aby projekt dosáhl svého předem stanoveného cíle

ve stanovený čas, v předem stanoveném rozsahu a nákladech. Další krok nesmí být zahájen, dokud není dokončen předchozí. Libovolné přechody vpřed nebo vzad není povoleny. Tato model předpokládá, že vývoj software lze přesně definovat, popsat a opakovaně realizovat. Vodopádový model má tři základní charakteristické rysy:

- úplné zadání – zadání projektu je přesně stanoveno již od počátku projektu,
- lineární průběh – každá fáze následuje po předchozí, nikdy se nevrací zpět,
- jednoznačnost – vždy jasně v jaké fázi se nachází projekt. [17]

Pomocí Vodopádového přístupu snadno řídit projekt, protože od počátku definovaný cíl a jasně určené role jednotlivých účastníků, alokované zdroje a nastaven časový rámeček. Úkolem manažera projektu je pak dodržet stanovený postup, čas a rozpočet. Ale tento přístup bude dávat vynikající výsledky pouze v projektech s jasně a předem definovanými požadavky a způsoby jejich realizace.

V době svého vzniku Vodopádový model umožnil rozdělení celého procesu vývoje do logicky oddělených částí, což pomohlo opakování postupu vývoje. Ale v posledních letech Vodopádový model ztrácí svou vedoucí pozici.

Tabulka 2. Výhody a nevýhody Vodopádového modelu

Výhody	Nevýhody
1. přehlednost, jednoduchost a snadná pochopitelnost	1. nízká flexibilita
2. jednoznačná struktura	2. není možné měnit projekt v jeho průběhu
3. flexibilita v počátečních fázích	3. problém plánování zdrojů, které je potřeba plánovat dlouho dopředu
4. jasný přehled požadavků	4. mezi začátkem projektu a spuštěním produktu je náročné udržet motivaci a soustředění týmu
5. orientace na čas a rozpočet	5. vysoké riziko zpoždění jednotlivých fází, které jsou na sobě závislé
6. odolnost vůči změnám personálu	6. nedostatečná spolupráce se zákazníkem
7. transparentnost řízení	7. chybějící zpětná vazba od produktových manažerů, kteří hodnotí výsledek až po dokončené implementaci
8. snadná kontrolovatelnost	8. prognóza časových a finančních nákladů může během implementačního procesu vzrůst
9. klade důraz na dokumentaci	9. neinformovanost vývojářů o průběhu vývoje produktu jako celku
	10. ve fázi testování jsou nejčastěji zjištěny kritické chyby, jejichž odstranění je nesmírně nákladné
	11. náročně se identifikují rizika

Zdroj: vlastní zpracování dle [17]

Hlavní nevýhodou Vodopádového modelu je provádění změn. Software je testován na konci životního cyklu, což může být příliš pozdě, aby něco změnit. Náklady na zavedení změn je důvodem přechodu na jiný model vývoje softwaru.

Důležitým důvodem pro kritiku této metodiky je minimální zapojení zákazníka a nedostatek zpětné vazby. Často se stává, že během procesu vývoje se může něco změnit, a proto produkt po dokončení již nesplňuje potřeby klienta. Kdyby byl zákazník zapojen do procesu vývoje, nestalo by se to.

Vodopádový model klade poměrně značný důraz na to, aby bylo vše precizně zdokumentováno. Většinou platí, že vývojáři začínají s implementací až v okamžiku, kdy mají k dispozici kompletní analýzu a návrh toho, jak by měl výsledný systém vypadat. Pro zákazníka je to často mírně frustrující v tom smyslu, že velmi dlouho od zadání nevidí žádné výsledky. To rovněž nevyhovuje i vývojářům, neboť jsou nuceni k častému jednání se zákazníkem, k vyplňování a čtení rozsáhlé dokumentace. [17]

V oblasti vývoje softwaru je vodopádový model již považován za překonaný, protože postrádá flexibilitu a možnost reagovat na vzniklé události. Model také neumožňuje návraty k předchozím fázím projektu, a proto pokud dojde k chybě na začátku projektu, například v analýze, je oprava chyby v pozdější fázi velmi nákladná. Na rozdíl od stavebnictví, kde je vodopádový model stále použitelný, vývoj softwaru nevyžaduje mnoho dokumentace a plánování v rané fázi projektu.

Vodopádový model se uplatňuje v projektech většího rozsahu, na kterých spolupracuje více týmů nebo se integruje více různých technologií a systémů. Za vhodné se také považují "jednorázové" projekty (nevyžadující pozdější vývoji), s fixními náklady a požadavky. Nebo například projekty, kde je díky předchozím zkušenostem možné přesně odhadnout průběh procesů.

Vodopádový model je rovněž vhodný v situaci, kdy je nutné přesně dodržet funkcionalitu, termíny, a rozpočet. Určitá byrokracie je vyvážena tím, že model jasně určuje role, kompetence a do značné míry i časový harmonogram. [17]

3.4 Agilní metodiky vývoje softwaru

Agilní metodiky vznikly jako alternativa k tradičním metodikám a jsou ideální pro projekty s nejasným nebo často se měnícím zadáním. Umožňují rychlý vývoj softwaru a zároveň dokážou reagovat na změnu požadavků v průběhu vývojového cyklu. Podle těchto metodik se správnost systému ověří jedině pomocí rychlého vývoje, předložení zákazníkovi a následných úprav dle zpětné vazby. [2]

Slovo “agilní“ pochází z anglického slova “Agile“, které znamená čilý, aktivní, horlivý, dynamický. Agile je o spolupráci a komunikaci, připravenosti na změnu, má své jasná pravidla. Jeho zásady jsou dělat to, co má smysl a dělat to tak, jak to nejlíp člověk umí a tím zajistili business value, tedy hodnotu pro zákazníka. [25]

Agilní metodiky pomáhají týmu myslet efektivněji, pracovat účinně a dělat lepší rozhodnutí. Tyto metody a metodologie zahrnují všechny oblasti tradičního vývoje softwaru jako projektové řízení, návrhu software a jeho architektury a postupné zlepšování procesů. Každá z těchto metod a metodologií se skládá z nejlepších praxí, které jsou zjednodušené a optimalizované, aby byly co nejjednodušší k zavedení. [27]

Podstatou Agilního přístupu je princip *"přizpůsob a kontroluj"*, což znamená intenzivní zapojení zákazníka, aby konečné řešení co nejvíce odpovídalo jeho potřebám. Jedním z nejdůležitějších aspektů této filozofie je výhoda nebo možnost rychle se adaptovat na změny v kterémkoliv okamžiku celého projektu v závislosti na zpětné vazbě od zákazníka, vývoji trhu a dodat na trh finální produkt nebo službu přesně podle aktuálních požadavků. Agilní přístup optimalizuje procesy a projektovou dokumentaci tak, aby přinášela zákazníkovi co nejvyšší hodnotu v porovnání s vynaloženým úsilím.

Klíčový princip — vývoj přes krátké iterace. Každá iterace představuje samostatný projekt s plánováním, návrhem, programováním, testováním a zhodnocením (retrospektivou). Na konci zákazník dostane pracovní kód, nebo produkt. Agilní přístup umožňuje rychle měnit priority v užití zdrojů, technologií a znalostí jako odezvu na náhlé změny situace na trhu a případné hrozby či příležitosti.

Agilní přístup se tedy snaží o eliminaci zbytečné byrokracie, zbytečného dokumentování každé sebemeně významné události či aktivity, o zjednodušení procesů změny. Ale nikoliv o eliminaci živelnou, eliminaci za každou cenu. Agilní přístup v žádném případě není návrat k anarchii, neznamená nepořádek, znamená stejně pořádek jako přístup tradiční. Jen znamená jiný přístup k pořádku, jiný přístup k rozumně chápaným procesům, jiný přístup k efektivitě. [17]

V poslední době implementace Agilního přístupu stala průmyslem: na toto téma se vyrábí speciální literatura, shromažďují tematické konference, na trhu práce se objevily zcela nové profese, Scrum master, Product Owner, Agile coach.

Manifest agilního vývoje softwaru — základní dokument, který obsahuje popis hodnot a principů agilního vývoje softwaru, vyvinutý v únoru 2001 na zasedání 17 předních světových vývojářů a expertů. Manifest shrnuje ve čtyřech bodech to, co znamená být agilní:

- ✓ *“Jednotlivci a interakce mají přednosti před procesy a nástroji”* - spolupracující tým má větší šanci na úspěch, protože vzájemná spolupráce a komunikace dokáže převážit hodnoty, přinášející z přesně definovaných procesů a nakoupeného softwaru. Manifest nic neříká o tom, že by neměly existovat procesy nebo dohody. Neříká nic ani o tom, že by týmy neměly využívat žádné nástroje, ale říká, že týmy by měly mít možnost si nástroje vybrat a využívat pouze ty, které jim pomáhají k dosažení kvalitního výsledku. [25]
- ✓ *“Fungující software mají přednosti před vyčerpávající dokumentací”* - doporučení Manifestu je, aby interní dokumentace byla co nejstručnější a obsahovala pouze klíčové informace. Ostatní informace je efektivněji dokumentovat přímo v kódu. [25]
- ✓ *“Spolupráce se zákazníkem mají přednosti před vyjednáváním o smlouvě”* - když je produkt tvořen pro zákazníka, je vhodné se zákazníkem na produktu spolupracovat. Smlouvy jsou důležité, ale neměly by být prostředkem k nahrazení spolupráce a komunikace se zákazníkem. Zákazník možná ne vždy přesně ví, co přesně chce při podpisu smlouvy, a to bývá často příčinou, kdy se mění rozsah smlouvy. Je důležité si uvědomit, že spokojený zákazník je ten, který dostal, co potřebuje a nemusí to být ten, který dostal výstup, který podepsal ve smlouvě. [25]
- ✓ *“Reagování na změny mají přednosti před dodržováním plánu”* - tržní prostředí i technologie se neustále mění a spolu s tím se mění také požadavky a problémy zákazníků. Ti se také musí přizpůsobovat trendům a konkurenci. Dodavatel se musí přizpůsobit a v jakékoliv fázi projektu musí být možné provádět změny. Tyto změny mohou být způsobeny také zjištěním, že vyvíjený projekt vlastně neřeší jeho problémy, a že by vlastně potřeboval něco úplně jiného. Projektové

plány jsou důležité, ale neměly by řídit vývoj a spolupráci. Každé plány se mění a jejich dogmatické dodržování může přinést více škody než užítku. [25]

V tomto Manifestu jsou popsány 12 principů, na kterých jsou založeny agilní metody řízení projektů:

1. Naší nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodáváním hodnotného software.
2. Víťáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.
3. Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.
4. Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.
5. Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.
6. Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.
7. Hlavním měřítkem pokroku je fungující software.
8. Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.
9. Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.
10. Jednoduchost a umění minimalizovat množství nevykonané práce jsou klíčové.
11. Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.
12. Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti. [20]

Podnikání je postaveno na vytváření zisku prostřednictvím uspokojování potřeb svých zákazníků. Za tímto účelem je nutné co nejčastěji analyzovat výsledky práce na projektu a upravovat plány, kontrolovat zpětnou vazbu od uživatelů a trhu jako celku.

Podnikové procesy se zrychlují díky multifunkčním a samo organizujícím se týmům, které vyřeší většinu úkolů s minimální potřebou koordinace. Agilní metodiky přispívají k rozvoji pracovníků a zvyšují jejich motivaci podporou nezávislosti při rozhodování. Product Owner neurčuje podmínky pro provádění úkolů, ale odhaluje problémy uživatelů. Vývojáři zase vnímají potřeby vzniku úkolů, vytváří hodnotu výsledku pro uživatele a

samostatně rozhodují. Tímto způsobem získávají vývojáři vysoký stupeň zodpovědnosti a vedení podporuje jejich dobré rozhodnutí prostřednictvím zpětné vazby. Pokud se něco pokazilo, Agilní přístup doporučuje manažerům vysvětlovat příčiny a důsledky této chyby. [27]

Tabulka 3. Výhody a nevýhody agilních metodik vývoje softwaru

Výhody	Nevýhody
1. spokojenost zákazníka	1. potřeba specifické firemní kultury a podpory top-managementu
2. šetření peněz a času	2. nutnost neustálé komunikace se zákazníkem
3. jasná viditelnost projektu	3. management si myslí, že může obsah projektu neustále zvětšovat
4. vysoké požadavky na vlastní organizaci a disciplínu v týmu	4. trvalá zpětná vazba může vést k přenosu spuštění projektu
5. rychlé dodání funkčního produktu	5. nové požadavky mohou být v rozporu se stávající architekturou
6. přizpůsobení ke změnám ve kterékoliv fázi projektu (díky krátkým iteracím)	6. potřeba přizpůsobit projektové dokumentace v rámci změny projektu
7. zvýšená konkurenceschopnost (díky rychlým změnám)	7. změny způsobují zvýšení náročnosti
8. neustálá komunikace s reálným světem (díky rychlé adaptace ke změnám)	8. zkreslení odhadu časové náročnosti úkolů
9. dobře kontrolování procesu (díky časté informovanosti o průběhu práce)	9. potřeba častých schůzek
10. minimalizace počtu chyb v konečném produktu (díky pečlivé kontrole kvality, která se provádí na konci každého sprintu)	10. neschopnost vytvářet dlouhodobé plány
11. minimalizace rizik (díky flexibilnímu systému, provádění změn)	11. potřeba pro motivované a vysoce kvalifikované specialisty
	12. velké komplexní projekty nelze snadné otestovat v krátkém čase

Zdroj: vlastní zpracování dle [27]

Díky Agilnímu přístupu se produktivita týmu výrazně zvyšuje a čas vynaložený k dosažení cíle se snižuje. Celý proces je maximálně flexibilní, po každé iteraci tým získává fungující produkt, který analyzuje a může se zlepšit. Takový styl vedení je nejúčinnější při práci s vysoce kvalifikovanými pracovníky. Vzhledem k tomu, že pracovní systém je založen na iteračním principu a každá iterace má svůj vlastní cíl, získáváme pracovní verze

produktu na konci každého sprintu. Použití Agilního přístupu je vhodné v následujících případech:

- v projektu je zapojen zkušený tým;
- dá se využít u projektu, jejichž obsah se stále mění a není ze začátku zcela jasný;
- změny je třeba provést maximálně rychle;
- je třeba rychle vyvinout funkční produkt.

Agilní přístup zlepšuje výsledky, pomáhá přizpůsobit projekt změnám, poskytuje přesnější hodnocení s menším úsilím a umožňuje efektivněji řídit projekt.

Vývoj softwaru je práce, která vyžaduje včasné přijetí správných rozhodnutí. Ředitele, architekti, vedoucí týmů a sami vývojáři pravidelně rozhodují ve prospěch těch nebo jiných nástrojů, platforem a metod. Pak se objevují metodiky, jejichž společným znakem je iterativní, minimalistický přístup a důraz na rychlé tempo vývoje po malých inkrementech. Těmito metodikami jsou zejména Lean, Extrémní programování, Kanban, Scrum, které mají agilní hodnoty a sdílejí některé své hodnoty, nápady a praktiky. [27]

3.4.1 Lean

Termín Lean znamená “štíhlá výroba”, který je založen na snaze celé organizace trvale se zlepšovat, setrvat v nastavených změnách, zamezit zbytečnému plýtvání, zvýšit efektivitu a tím získat co nejlepší uspokojení potřeb zákazníka.

Lean má kořeny v Japonsku, zejména ve firmě Toyota a v její přístupech k řešení problémů. Společnosti prospívají pouze ty změny, které přinášejí prospěch a vyžadují minimální náklady. Mary a Tom Poppendieck použili tuto metodiku a napsali o ní knihu "Lean Software Development". Podle průzkumu PMI, 8 % společnosti neustále používají principy Lean a 26 % často se na ně obracejí. [16]

Lean Software Development navrhuje k aplikování v projektech následujících sedm principů:

1. Zbavit se nadbytečného – odstranit veškeré úkony, které nepřinášejí firmě žádnou hodnotu.
2. Zlepšovat se a učit se již v průběhu – je třeba se poučit z chyb, aby nedošlo k jejich opakování v budoucnu. Zároveň je nutné pochopit potřeby zákazníka a soustředit se na jeho zpětnou vazbu. Ta nám definuje oblasti, které jsou aktuálně důležité.

3. Rozhodnout se co nejpozději – před finálním rozhodnutím je dobré mít co nejvíce informací. Čím delší dobu máme na rozhodování, tím více skutečných faktů máme k dispozici.
4. Dodávat svojí práci, co nejrychleji – dodání produktu zákazníkovi v nejkratším čase nám umožní promptní vazbu od něj. Případné chyby vyřešíme ihned v následující iteraci.
5. Posílit důvěru a pravomoci v týmu – zvýšení pravomocí a lepší naslouchání v týmu podporují motivovanost členů týmu.
6. Zaměřit se na kompletní dojem – snaha ukázat zákazníkovi kompletní strukturu vlastností softwaru – kvalitu, technické funkce, celkovou únosnost systému atd.
7. Soustředit se na celkový výsledek - *“Think big, act small, fail fast; learn rapidly”* - Uvažujte ve velkém, začnete u drobných věcí, konejte rychle, analyzujte své chyby a poučte se z nich. [25]

Výhodou metodiky je rychlé spuštění produktu s nejmenším počtem chyb. Ve své studii společnost BBC zjistila, že Lean zvyšuje rychlost vývoje softwaru na 37 %, a také snižuje počet chyb na 24 %. Také podle studie Lean Business Report je výhodou snížení nákladů na projekty - 27 % IT společností snížilo náklady díky implementaci principů Lean. Nicméně Lean není vhodný pro všechny. Štíhlý vývoj stojí používat, pouze pokud jsou k projektu připojeni zkušení vývojáři, učení během procesu se ukázalo jako nemožné a může produktu ohrozit. [27]

3.4.2 Extrémní programování

Extrémní programování je další známou agilní metodikou softwarového vývoje. Extrémní programování vzniklo v automobilovém průmyslu za účelem optimalizace výrobního procesu v 90. letech 20. století. Tato metodika je postavena na využití osvědčení a známých principů softwarového vývoje v extrémní formě. Tzn., že pokud se osvědčí nějaká metoda – častější testování, zkrácení iterací, nebo například simplifikace programu, využívají se tyto metody vždy, v každé iteraci a v maximální možné podobě. [4]

Extrémní programování je přístup k vývoji softwaru, který využívá dodávky softwaru v krátkých vývojových cyklech. Nesnaží se definovat všechny požadavky na začátku projektu, jako je tomu u projektů řízených Vodopádovým přístupem. Místo toho spoléhá na blízkou spolupráci mezi zákazníkem a vývojovým týmem, kteří společně definují a upřesňují požadavky v průběhu vývoje.

Extrémní programování uznává tyto základní hodnoty:

–Komunikace – všichni angažováni do projektu manažeři, architekti, vývojáři, testeři a zákazníci by měli mezi sebou každý den komunikovat. Dodržování tohoto pilíře pomůže se vyhnout zdlouhavému vyřešení problematické části projektu. Členové týmu nečekají na oficiální schůzku, kdy se tento problém otevře, ale ihned to komunikují se správnou osobou;

–Jednoduchost – tato hodnota je založena na vykonávání činností, které jsou nutné pro danou etapu práce na produktu. Tým si neklade za úkol předvídat potenciální potřeby a požadavky zákazníka, nečiní nic nad rámec potřebného. Kupříkladu člen týmu se nebude věnovat psaní kódu, který může být využit někdy v budoucnu. Takového stavu se tým snaží docílit pomocí neustálé komunikace se zákazníkem;

–Zpětná vazba – v XP se nepřetržitě testuje a teprve na základě výsledků testování se doladují další kroky. Tým by se měl snažit o zkrácení doby čekání a v případě neúspěšného testování ihned zahájit opravu. Neméně důležitou roli hraje i aktivita zákazník, který tomu může napomoci, a to včasnou kontrolou funkcionality produktu a poskytnutím zpětné vazby;

–Odvaha – k rozhodnutí se pustit do řízení projektů metodou Extrémního programování je potřeba odvaha. Zároveň je nutné mít odhodlání svižně přijímat změny, odložit dosavadní práci a začít vše znovu. Odvahu také vyžaduje akceptace zodpovědnosti za chyby a ochota konat věci jinými způsoby, než bylo zvykem. [4]

Principy extrémního programování:

1. plánovací hra – tvorba plánu, na které se podílí všichni členové týmu;
2. malé verze – co nejčastější uvolňování nových verzí;
3. metafora – je náhradou za termín „architektura“, vývoj je popisován příběhem o tom, jak má konečný systém fungovat;
4. jednoduchý návrh – dodržování maximální jednoduchosti systému;
5. testování – u metodiky XP se testování provádí velmi často. Testy jsou tvořeny ještě před vlastní implementací funkcí a pro pokračování je nutné, aby testování proběhlo s pozitivním výsledkem;
6. refaktorizace – při tomto postupu je zdrojový kód maximálně optimalizován;
7. párové programování – veškeré programování je prováděno dvěma programátory na jednom počítači;

8. společné vlastnictví – tato metodika se od většiny ostatních liší tím, že změny zdrojového kódu mohou provádět všichni programátoři;
9. nepřetržitá integrace – celý systém se v průběhu dne vícekrát sestavuje a testuje;
10. 40hodinový pracovní týden – zamezuje pracovnímu vyčerpání a přepínání programátorů. Metodika propaguje filozofii, že odpočatý a spokojený programátor podává kvalitnější pracovní výkony a jeho práce je efektivnější;
11. zákazník na pracovišti – podobně jako je tomu u jiných agilních metodik, i zde je zákazník součástí týmu. Vzhledem k extrémnímu přístupu metodiky XP se přísně dbá na jeho faktickou přítomnost. [16]

Silné stránky extrémního programování spočívají v tom, že díky blízké spolupráci mezi členy týmu a neustálému zapojení zákazníka výsledný produkt bude naplňovat potřeby zákazníka a přinášet mu požadovanou hodnotu. Změnové požadavky jsou vítány, protože umožňují co nejvíce přizpůsobit výsledný produkt potřebám zákazníka. XP se zabývá pouze probíhající iterací tak, aby byl dodán funkční celek co možná nejdříve a aby co nejvíce odpovídal potřebám zákazníka. [16]

Slabé stránky jsou v tom, že bez podrobných a celkových požadavků na projekt je velice obtížné stanovit přesný odhad doby trvání projektu a celkové potřebné náklady. Také bez jasného pochopení všech požadavků před začátkem projektu je často obtížné dosáhnout optimálního designu celkové architektury řešení. XP není vhodný pro velké a komplexní projekty, kde chybí aktivní zákazník a reálná business potřeba. Pokud není tým umístěn společně, nejlépe v jedné místnosti, přináší to do projektu značné riziko. [16]

3.4.3 Kanban

Kanban není systém pro řízení projektů, ale je to metoda pro zlepšení procesů tzn. kroky pro tým k vytvoření a dodání software. Vychází z Lean metodologie. Vylepšení software v Kanbanu je ponecháno v rukou týmu, kdy členové týmu přicházejí s vlastními návrhy na úpravu software nebo nacházejí vlastní řešení pro problémy. [27]

Kanban vznikl v roce 1970, původně v automobilovém průmyslu. Toyota jej začala uplatňovat ve své sériové výrobě a urychlila tím svůj výrobní proces. Kanban znamená v japonštině „Vizuální karta“ a tomu slovnímu spojení odpovídají i samotné principy:

- Vizualizace Workflow;
- Omezení rozdělané práce (WIP);
- Dodržení dodací lhůty. [12]

Kanban lze popsat jednoduchou frází "Snížení práce, která se právě provádí (*work in progress*)". Jeho úkolem je vyrovnání různých odborníků uvnitř týmu a vyhnout se situaci, kdy designéři pracují dny, a vývojáři si stěžují na nedostatek nových úkolů.

Metoda Kanban je známá užitím tabule, která slouží k vizualizaci vývoje. Požadavky jsou rozděleny do jednotlivých úkolů, které jsou dále seřazeny dle priorit. Tým ihned vidí na tabuli pokroky a splněné úkoly a může se efektivně věnovat dalším požadavkům. Metoda je velice jednoduchá a účinná. Velice často je využívána v kombinaci s metodikou Scrum či Extrémního Programování. [12]

Obrázek 4. Kanban tabule



Zdroj: vlastní zpracování dle [12]

Sloupce jsou fáze procesu vývoje, mají různé názvy a použití. 1. sloupec *Udělat "To Do"* - Backlog, kam dávány úkoly, které jsou připraveny k zahájení. 2. sloupec *Vývoj "In Progress"* - zde úkol je až do dokončení. 3. sloupec *Testování* – zde úkol je během testování. Pokud jsou nalezeny chyby — návrat karty do Vývoje, pokud ne — přesouvá se dál. 4. sloupec *Spuštění* – úkol je testován a připraven ke spuštění. 5. sloupec *Hotovo "Done"* – sem se kartička dostane pouze tehdy, kdy práce na úkolu je zcela dokončena.

Kartičky jsou pracovní úkoly. Jednotlivé kartičky ve sloupcích mají různou barvu podle specifického určení, zpravidla podle priorit nebo podle zařazení činnosti k určitému oboru (design, objekty apod.). Tato barva je v rámci kartičky neměnná zleva doprava, protože jakmile se této kartičce vlastnost určí, již se nemění.

Důležitým parametrem je počet současně „*In Progress*“ – tedy právě zpracovávaných činností. Toto číslo je pevně dané, většinou určené manažerem projektu a nesmí být překročeno. Omezení pomocí WIP zajistí, že v případě maximálního stavu v oblasti vývoje bude následující volný vývojář využit pro jinou důležitou činnost. Zajistí se tím plynulý tok práce směrem ke kolonce „*Hotovo*“ namísto velkého množství nedodělaných, ale zpracovávaných úkolů. Hlavním ukazatelem účinnosti je celková doba

„Lead time“, která určuje, jak dlouho trvá kartičce (tedy pracovnímu úkolu) od prvního sloupečku v Backlogu do jeho přesunutí v Hotovo. Úkol proběhl rychle – tým pracoval produktivně a soudržně. Úkol byl odložen – je třeba přemýšlet, v jaké fázi a proč vznikly zpoždění a jejichž práci je třeba optimalizovat.

3.4.4 Scrum

Scrum je nejpopulárnější agilní metodika, která byla poprvé představena v roce 1995 Kenem Schwabem a Jeffem Sutherlandem a její název vychází z tradičního rozmístění tzv. "mlýna" v rugby. Název metody odkazuje na její rychlost a adaptivní schopnosti a zároveň na nutnost spolupráce v rámci týmu. Podle Scrum odborníka Jeffa Sutherlanda je to „*rámec, který umožňuje řešit komplexní problémy adaptivní metodou a dodávat tak produkty nejvyšší možné hodnoty*“. [22]

3.5 Detailní popis vybrané metodiky Scrum

Na základě studia odborné literatury tým AUTO.RIA zvolil Scrum jako metodiku vhodnou pro vývoj. Základem Scrumu je optimalizace procesu a zajištění ideálního výsledku díky rozdělení času, produktu a organizace na dílčí části. Podobně jako u dalších agilních metodik i Scrum využívá výhod iterativního a inkrementálního přístupu. Iterace je základem Scrumu. Metoda je založena na principu vlastní organizace týmu, komunikaci a sdílení informací, které podporují spolupráci v týmu. Pro samotný vývoj je charakteristická vysoká flexibilita a definice požadavků se v průběhu vývoje mohou upravovat. Nepracuje s projektovým plánem jako celkem, ale soustředí se na řízení iterací. Ačkoliv používá některé základní projektové přístupy, jako je plánování backlogu, tak tradiční projektové problémy, jako je řízení rizik a rozsáhlé detailní plánování, metodika Scrum neřeší.

Scrum využívá iterační přístup a stojí na třech pilířích:

1. *Transparentnost* – důležité aspekty procesu musí být viditelné těm, kteří mají vliv na výsledek. Transparentnost vyžaduje, aby tyto aspekty používaly společný standard, takže pozorovatelé budou rozumět tomu, co vidí.
2. *Kontrola* – uživatelé Scrumu musí kontrolovat artefakty a postup směrem k cíli tak často, aby se daly odhalit nepřijatelné odchylky v procesu. Frekvence kontroly nicméně nesmí být tak vysoká, aby stála v cestě skutečné práci. Kontrola je nejužitečnější, když ji provádějí kvalifikovaní lidé přímo na místě, kde se pracují.

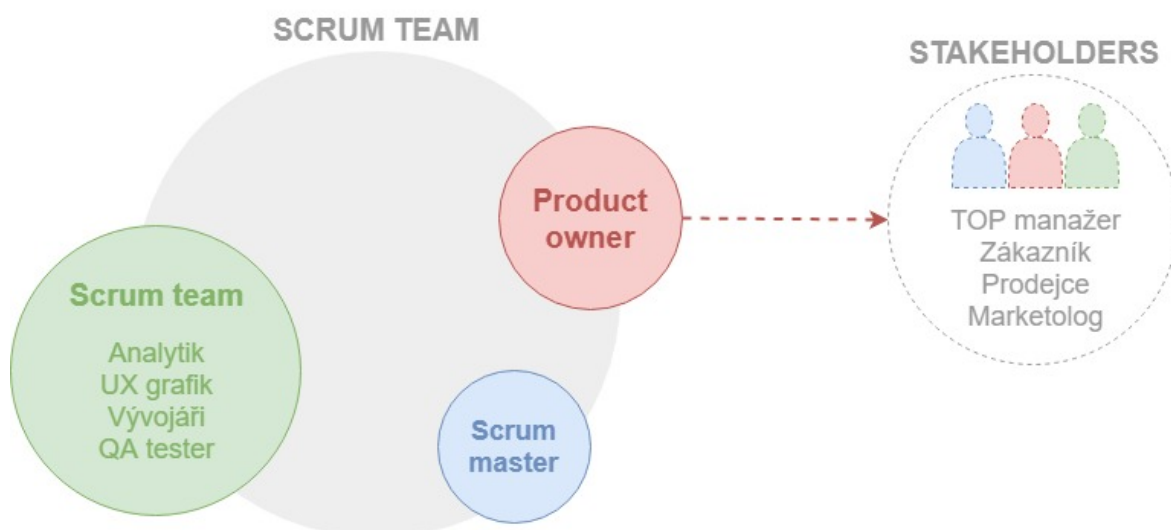
3. *Adaptace* – jestliže revizor na základě kontroly rozhodne, že jeden nebo více aspektů procesu jsou mimo přijatelné hranice a že výsledný produkt bude nepřijatelný (neakceptovatelný), pak je nutné tento proces adaptovat. Změna musí být provedena co nejdříve, aby byla minimalizována budoucí odchylka. [22]

Metodika Scrum obsahuje celou řadu odborných názvů jednotlivých aktivit a Scrum ceremonií, které byly do češtiny také přejaty ve svém původním anglickém originále.

3.5.1 Scrum Team

Členové Scrum týmu jsou programátoři, testeré, analytici, architekti, designéři, ale i uživatelé, jednoduše řečeno každý, kdo se podílí na práci a vytváří ten správný produkt pro jeho majitele (vlastník produktu) a to vše při zachování efektivity. [9]

Obrázek 5. Scrum Team



Zdroj: vlastní zpracování dle [24]

Scrum tým je složen z profesionálů a má následující charakteristiky:

- je sebeorganizující* – sami si volí, jak provedou práci, nejsou tedy přímo vedeny nikým zvenčí;
- je multifunkční* – mají všechny schopnosti potřebné k tomu, aby dokončili svou práci bez toho, že by museli čekat na někoho, kdo není součástí týmu;
- členové vývojového týmu všichni mají titul "vývojář" bez ohledu na to, jakou práci vykonávají;
- mají společnou odpovědnost za výsledek. [22]

Mezi základní dva úkoly člena týmu patří implementace jednotlivých uživatelských příběhů a připomínkování projektu. Samozřejmostí je vývoj a implementace jednotlivých

úkolů a tvorba výsledného softwaru. Druhým úkolem, který mají členové týmu, je připomínkování v podstatě všeho, co se projektu týká. Připomínkování musí být přednášeno věcně, ve vhodnou dobu, a především ve vhodné podobě.

Stakeholder je nepostradatelnou součástí projektu. Stakeholdeři jsou:

1. *Zákazníci*, tedy ti, kteří projekt financují (známí též jako sponzoři), komunikují převážně s vlastníkem produktu;
2. *Uživatelé*, ti, kteří budou využívat výstupy Scrum týmu, komunikují hlavně s vývojovým týmem, poskytují mu přímou zpětnou vazbu;
3. *Manažeři*, kteří poskytují organizační rámec, ve kterém Scrum tým působí, je tedy úzká komunikace se Scrum Masterem, který jim pomáhá v otázkách, jak nejlépe podpořit Scrum tým v jeho efektivitě. [9]

Agilní proces se snaží Stakeholdera začlenit do projektu, v tomto se liší agilní metody od jiných projektových metod. Stakeholder určuje, jaké jsou jeho priority a podílí se na případných změnách v průběhu vývoje a tím může ovlivnit v průběhu funkcionalitu produktu, čímž se stává součástí týmu. Stakeholder může být jak zevnitř firmy, tak i zvenku. [25]

Aby mohl být Stakeholder součástí týmu, je zapotřebí rozumět jeho potřebám a byznysu a mít vzájemný respekt a důvěru. V agilním světě vztah je popsán v Agilním manifestu „*Upřednostňujeme funkční software před jednáním o smlouvě*“, což znamená, že se nesnažíme Stakeholderovi slepě dodat to co si objednal, ale udělat z něj partnera a během domluvené času doladovat funkcionalitu produktu. [25]

Aby mohly fungovat základní principy Scrumu, je zapotřebí zavést specifické role, které tradiční metody managementu neměly, jedná se o Product Owner a Scrum Master. Není vhodné kombinovat role Scrum mastera s Product Ownerem. Obvykle nefunguje ani kombinace role s vývojářem či testerem, kde vývojáři upřednostňují svou technickou práci před prací Scrum mastera. Následně dochází k tomu, že tým je bez Scrum mastera. [25]

Product Owner ověřuje, zda tým pracuje efektivně z obchodní stránky, což znamená, že definuje priority, rozhoduje, která funkcionalita bude mít největší prioritu, na které se bude pracovat později a na které vůbec ne. Definuje vize projektu a její transparentní komunikaci mezi týmem, zákazníkem a firmou. [25]

Primární úlohou Product Ownera je zaměřit se na efektivitu, se kterou je produkt vytvářen, a to, že jde o ten správný produkt pro jeho zákazníky. Dále je odpovědný za s tím spojenou optimalizaci ROI (ziskovost investicí), má na starosti zajištění existence společné

vize produktu, správu produktového Backlogu a v neposlední řadě má na starosti schvalování dodaných inkrementů produktu na konci každé iterace. Odpovídá také na otázky členů týmu týkajících se toho, co se bude vytvářet a proč. [9]

Product Owner je jediný člověk, který je zodpovědný za správu produktového Backlogu, která zahrnuje tyto činnosti:

- jasná formulace položek produktového backlogu;
- uspořádání položek produktového backlogu tak, aby co nejlépe odrážel produktovou vizi a cíle;
- zajišťování toho, aby vývojový tým dodával vždy nejlepší hodnotu;
- zajištění transparentnosti a dostupnosti backlogu tak, aby bylo každému srozumitelné, co obsahuje a na čem bude Scrum tým v nejbližší době pracovat;
- zajištění toho, že vývojový tým dostatečně rozumí položkám v backlogu. [22]

Scrum master zaujímá roli vedoucího týmu, ovšem je to vedoucí, který týmu hlavně slouží. Je zodpovědný za osvojování a dodržování pravidel Scrumu. Scrum master pomáhá lidem v okolí vývojového týmu rozpoznat, které jejich interakce s týmem jsou prospěšné a které ne. Pomáhá každému změnit své chování tak, aby vývojový tým mohl vytvářet maximální hodnotu. [22] Vede Scrum tým k samo-organizování, řeší možné spory a překážky, které brání vývojovému týmu pracovat na produktu, dále pomáhá produktovému vlastníkovi pochopit a plnit jeho roli správně, a v neposlední řadě napomáhá začlenit Scrum metodiku vně organizace, ve které je metodika používána. [9]

Scrum Master musí být komunikativní, vnímavý a měl by umět tlumit případné konflikty v týmu a podporovat schopnost týmu, aby dokázali přijít na řešení vlastními silami i za cenu lokální neefektivnosti. Jeho hlavním cílem a povinností je především: odstraňovat problémy, motivovat tým k lepším výsledkům, pomáhat týmu dosáhnout jeho cílů, tým chránit před vnějšími vlivy, které by mohly odvádět od práce. Komunikuje s okolím a vysvětluje jim, které interakce jsou užitečné a které ne. Tímto Scrum master pomáhá maximalizovat hodnotu, kterou vytváří Scrum tým. [25]

Služby Scrum mastera vlastníkovi produktu:

- hledá techniky pro efektivní správu produktového backlogu;
- rozvíjí u vývojového týmu potřebu jasných a stručných položek produktového backlogu;
- pomáhá z plánování produktu v empirickém prostředí;

- sleduje, že vlastník produktu spravuje produktový Backlog v duchu dosažení maximální hodnoty;
- vysvětluje a aplikuje principy agilního vývoje;
- moderuje podle potřeby všechny schůzky ve Scrumu. [22]

Služby Scrum mastera vývojovému týmu:

- vede vývojový tým směrem k sebeorganizovanosti a multifunkčnosti;
- učí a vede vývojový tým k tomu, aby vytvořil produkty s vysokou přidanou hodnotou;
- odstraňuje překážky, které brání vývojovému týmu v postupu;
- moderuje podle potřeby všechny schůzky ve Scrumu;
- koučuje vývojový tým v prostředí organizací, ve kterých ještě Scrum není správně pochopen a přijat. [22]

Služby Scrum mastera organizaci:

- vede a školí organizaci v osvojování Scrumu;
- plánuje implementaci Scrumu v organizaci;
- pomáhá zaměstnancům a ostatním zúčastněným stranám pochopit a osvojit si Scrum a empirický vývoj produktu;
- iniciuje změny, které vedou k vyšší produktivitě produktového týmu;
- spolupracuje s ostatními Scrum mastery na efektivním zavádění Scrum v organizaci. [22]

3.5.2 Scrum artefakty

Artefakty Scrumu reprezentují práci a její hodnoty různými způsoby, které jsou užitečné pro poskytování transparentnosti a dále jsou příležitostí pro kontrolu a adaptaci. Artefakty definované Scrumem jsou navrženy pro maximální zvýšení transparentnosti klíčových informací tak, aby porozumění danému artefaktu bylo jednoznačné. [22] Scrum artefakty jsou: Sprint, Produkt Backlog, Sprint Backlog.

Sprint je základní vývojovou iterací. Každý sprint můžeme považovat za projekt v (maximálně) měsíčním rozsahu. Součástí každého sprintu je popis toho, co má být vytvořeno (tj. návrh cílového produktu a flexibilní plán, který bude sloužit jako vodítko pro provádění prací), dále samotná práce a výsledný. [22]

Cíle sprintu (Sprint Goal) má být dosaženo implementací produktového backlogu. Definice cíle říká vývojovému týmu, proč vlastně vytváří přírůstek. Cíl je definován na

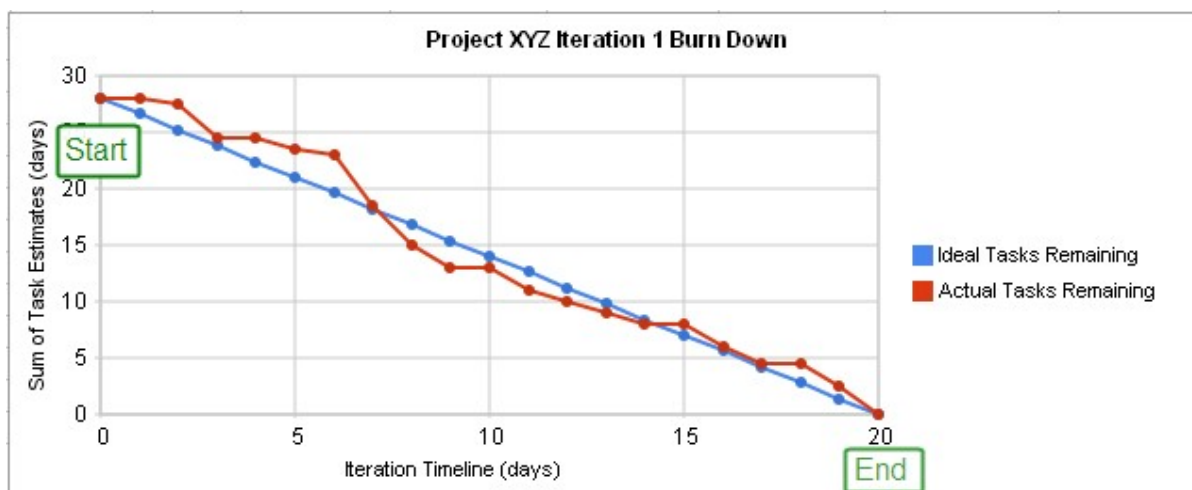
plánovací schůzce. [22] V případě, že Scrum tým zjistí kdykoliv během Sprintu, že stanovený cíl již není dosažitelný nebo už nebude sloužit potřebám organizace, produktový vlastník je oprávněn okamžitě ukončit Sprint. [9]

Sprint je rozdílný oproti běžným iteracím tím, že není vnitřně strukturován. Na začátku sprintu se naplánuje práce a poté se analyzuje, navrhuje, vyvíjí, testuje a nasazuje bez striktního členění fází. [17] Během sprintu:

- je vytvořen (v kvalitě „hotovo“) potenciálně nasaditelný přírůstek produktu;
- se neprovádí žádné změny ohrožující cíl sprintu (Sprint Goal);
- neklesá kvalita dosažení cíle sprintu;
- může být mezi vlastníkem produktu a vývojovým týmem znovu projednán a upřesněn rozsah. [22]

Ve Scrumu Ganttův diagram k dispozici není, ale máme takzvaný Burndown graf, který zcela jinak zobrazuje časovou náročnost sprintu. Burndown graf ukazuje množství práce v rámci jednoho sprintu a vizualizuje předpokládaný konec projektu. Slouží k provedení analýzy tempa práce pro včasné dokončení sprintu a podporuje tým v udržení rychlosti. Burndown graf má na ose Y množství práce naplánované pro celý sprint (story points). Na ose X má celkový počet dnů zbývajících do konce sprintu. Jak práce postupuje, vkládá se do grafu, kolik práce zbývá udělat.

Obrázek 6. Burndown graf



Zdroj [5]

Ideální průběh je stav, jemuž se chceme přiblížit a vůči kterému aktuální stav porovnáváme. Z toho důvodu ideální stav v grafu ponecháváme (modrá řada v grafech), aby byly lépe vidět odchylky. Cílem naší práce je v nepravidelném tempu s minimálními odchylkami vyprázdnit celý Backlog. [17]

U zpožděného projektu je zcela zřejmé, že tempo dokončování činností je výrazně nižší, než je ideální průběh. Příčinami mohou být špatně naplánovaný sprint, podcenění naplánovaných úkolů, nízké tempo vývojového týmu nebo chybějící člen týmu z důvodu nemocenské. Příčinou však může být i skutečnost, že vývojový tým ve svém aktuálním složení nemá kvality na vypořádání všech uložených úkolů. Z těchto důvodů je nutné měření provádět průběžně během sprintu, aby bylo možné identifikovat problém, najít a následně odstranit příčinu. [17]

Zrychlený průběh vypadá na první pohled jako naprosto bezproblémový a většina manažerů bude mít radost. Ovšem tento stav může skrývat několik problémů, které v budoucnu vyplynou na povrch. Samozřejmě se nemusí dít nic závažného a tým například pouze nasadil vyšší tempo. To samo osobě v krátkém čase nevádí, ale z dlouhodobého hlediska jsou naším cílem stabilní výkony. Příčinou také může být fakt, že sprint byl nevhodně naplánován a hodnocení úkolů bylo přeceněno. Jako velmi nevhodná se také nabízí možnost, že práce byla nekvalitně odvedena, díky čemu jsou úkoly plněny rychleji. Tuto příčinu lze zjistit testováním, které chyby odhalí a následně může přinést časovou náročnost pro opravy. [17]

Na základě grafu má Product Owner možnost analyzovat nejkritičtější úkoly, také může udělat revizi původních odhadů. Měřit a na základě měření se rozhodovat, zda je nutné něco upravit či nikoliv, to je základní vlastností procesu řízení projektů.

Produktový Backlog je seznam všech funkcí, požadavků, rozšíření a oprav chyb, které představují všechny změny, které budou provedeny v produktu v příštích vydáních. Produktový Backlog je jediným zdrojem požadavků na jakoukoliv změnu v produktu. Vlastník produktu je zodpovědný za obsah, dostupnost a prioritizaci backlogu. [22]

Produktový Backlog je seznam položek představujících práci, kterou je třeba provést v určitém čase. Jednotlivé položky jsou popsány na funkční úrovni a může je do produktového backlogu přidávat kdokoliv, ale jedině vlastník produktu určuje, v jakém pořadí budou vývojovým týmem vykonávány. [9]

Základní vlastností produktového backlogu je, že jde o žijící dokument. Jednotlivé požadavky v podobě položek produktového backlogu časem (s každým Sprintem) vznikají, upravuje se jejich úroveň detailu, zpřesňují se, jiné naopak zanikají. Jiný typ změn se může týkat úpravy vztahu mezi hodnotou, časem a náklady. [9] Produktový Backlog je dynamický, stále se mění tak, aby obsahoval to, co produkt potřebuje, aby byl vhodný,

konkurenceschopný a užitečný. Změny mohou být způsobeny změnami požadavků byznysu, podmínek na trhu nebo technologií. [9]

Správcem Product backlogu je pouze Product Owner, který odpovídá za jeho obsah, přiděluje priority a třídí požadavky podle nich. Požadavky jsou popsány dle jednotného vzoru ve stylu "Já jako zákazník chci mít přehled mých služeb online", který se nazývá User Story. Požadavky jsou sestaveny tak, že je zřejmé, jakou hodnotu představují pro uživatele. Vývojový tým je odpovědný za všechny odhady. Vlastník produktu může ovlivnit vývojový tým tím, že mu pomůže porozumět a vybrat kompromisy, ale finální odhad je vytvořen vývojovým týmem, který bude provádět odhadovanou práci. [22]

Sprint Backlog je množina úkolů vybraných z produktového backlogu včetně plánu dodání produktového přírůstku a splnění cíle sprintu. Backlog sprintu je prognóza vývojového týmu jaká funkcionality bude dodána v následujícím sprintu a kolik práce bude potřebné k dodání této funkcionality. Backlog sprintu ukazuje všechnu práci, kterou si vývojový tým vytyčil jako nezbytnou ke splnění cíle daného sprintu. [22]

Sprint Backlog se během sprintu mění, vývojový tým ho upravuje po celou dobu sprintu. Tyto úpravy jsou potřebné proto, že vývojový tým během sprintu získává více poznatků o tom, co je potřeba ke splnění cíle sprintu. Pokud je požadována další práce, vývojový tým ji přidá do sprint backlogu. Podle toho, kolik je na úkolech odpracováno, nebo jestliže jsou dokončeny, provádí se aktualizace odhadu. Pokud se některé úkoly ukážou jako zbytečné, jsou z backlogu odstraněny. Jedině vývojový tým může změnit sprint Backlog v průběhu sprintu. Sprint Backlog tak představuje jasný, transparentní a aktuální obraz práce, kterou vývojový tým plánuje v průběhu sprintu dokončit. Sprint Backlog je výlučně záležitostí vývojovému týmu. [22]

Většina vývojových týmů volí způsob vizualizace pomocí tabule, známé z metodiky Kanban. Je zde jasně vidět, jaká práce je naplánována na Sprint a její aktuální stav, ve kterém se daná úloha právě nachází. [9]

3.5.3 Scrum schůzky

Nedílnou součástí Scrumu jsou pracovní schůzky. Scrum zajišťuje pravidelnost a minimalizují potřebu dalších, Scrumem nedefinovaných schůzek. Všechny činnosti jsou časově ohraničené (time-fixed), tzn. že každá činnost má určenou svou maximální délku trvání. Činnosti mohou skončit vždy, když je dosaženo jejich cíle; tím je zajištěno, že činnosti zaberou přiměřenou dobu a nedochází k plýtvání časem. [22] Před každou

schůzkou si musí být moderátor jistý, že ostatní účastníci schůzky znají přesný účel, proč je schůze svolána. Moderátor také musí myslet na všechno, co bude ke schůzce potřeba. Je potřeba svolat osoby potřebné k projednávanému účelu, naplánovat kdy a kde se schůzka uskuteční a jak bude schůzku moderovat.

Priority and pre-planning Meeting. Product Owner definuje funkcionalitu, ale hned dalším jeho úkolem je dát funkcionalitě prioritu. Musí vybrat User Story, které se budou řešit v dalším sprintu. Product Owner standardně získává priority úkolů od zákazníků, kteří si říkají, co je pro ně klíčové.

Většinou Product Owner seřazuje priority úkolů v Product Backlogu, kde čím výš se úkol nachází, tím větší má prioritu. Cílem tedy pre-planningu je vytáhnout z Product Backlogu prioritní User Story, aby tým měl potom z čeho si vybírat na další sprint. [25] Tuto informaci o prioritách by tým měl dostat jeden až dva dny před planningem, aby se na plánování sprintu mohl připravit. Existují dvě základní možnosti, jak s prioritami pracovat. První varianta je pravidelné plánování preplanningové schůzky pro určování priorit. Druhou variantou je mít celý Backlog po celou dobu projektu seřazený a členové týmu si postupně odebírají stories do sprint backlogu. Nic však nebrání tomu, aby se priority v průběhu projektu mohou měnit. [25]

Sprint Planning Meeting. Práce, která má být vykonána během sprintu, je plánována na plánovací schůzce (Sprint Planning). Tento plán je vytvářen v součinnosti celého Scrum týmu. [22]

Vstupními informacemi této schůzky jsou: produktový Backlog, poslední přírůstek produktu, plánovaná kapacita vývojového týmu pro příští sprint a výkon vývojového týmu v předchozím sprintu. [22]

První část má odpovědět na otázku "*Co můžeme dodat na konci tohoto Sprintu?*". Jde o skutečně detailní "workshop požadavků". Je zde přítomen vlastník produktu, a to hlavně proto, aby navedl vývojový tým na správný směr a i proto, aby odpovídal na otázky jeho členů. Tým může taktéž v této části využít znalosti z jejich rychlosti z minulosti jako prediktor (Yesterday's weather).

Druhá část je označovaná jako workshop "design" a odpovídá na otázku „Jakým způsobem budeme pracovat na projektu“. [9]

Průběh plánovací schůzky:

- vývojový tým odhaduje, které všechny funkčnosti budou během sprintu vyvinuty;

- Product Owner probírá záměr sprintu a položky produktového backlogu, které (pokud budou dokončeny) by měly zajistit splnění cíle sprintu. Počet položek produktového backlogu zařazených do sprintu závisí výhradně na rozhodnutí vývojového týmu;
- celý Scrum tým se snaží společně porozumět obsahu sprintu;
- poté, co vývojový tým odhadne, které položky produktového backlogu během sprintu dodá, Scrum tým definuje cíl sprintu (Sprint Goal). Cíl sprintu má být dosažen v průběhu sprintu;
- stanovil-li vývojový tým cíl sprintu a vybral položky produktového backlogu pro nadcházející sprint, může začít uvažovat o tom, jak během sprintu vytvoří plánovaný přírůstek (jak jej převede do stavu "hotovo");
- položky produktového backlogu zařazené do tohoto sprintu spolu s plánem na doručování položek tvoří Backlog sprintu. [22]

Na konci plánovací schůzky jsou práce naplánované vývojovým týmem na první dny sprintu rozloženy – často až na části o jednodenní (nebo menší) pracovních. Vývojový tým by měl být schopen vlastníkovému produktu a Scrum masterovi vysvětlit, jakým způsobem hodlá jako sebeorganizující tým dosáhnout cíle sprintu a vytvořit očekávaný přírůstek. [22]

Výsledkem plánování Sprintu je Sprint Backlog, jehož obsah je seznam úloh a plán, který musí být týmem plněn. Sprint Backlog má nejčastěji podobu fyzické tabule, respektive může jít o jednoduchý softwarový nástroj zobrazující jednotlivé úkoly vzhledem ke stanovenému plánu. [9]

Daily Scrum. Denní schůzka je 15minutová schůzka vývojového týmu určená pro synchronizaci aktivit a vytvoření plánu na dalších 24 hodin. [22] Role Scrum Mastera zde vystupuje v pozici moderátora diskuse. Výhoda je být u tabule, kde jsou napsány/vylepeny všechny úkoly, které se chtějí stihnout aktuální sprint, a to z důvodu ať je vidět, kolik toho ještě zbývá. [22]

Vývojovému týmu by mělo být každý den jasné, jakým způsobem se chce společně sebeorganizovat pro dosažení cíle sprintu a pro vytvoření očekávaného přírůstku na konci sprintu. Proto na schůzce členové vývojového týmu odpovídají na otázky:

"Co jsem včera udělal proto, abych pomohl vývojovému týmu splnit cíl sprintu?"

"Co budu dělat dnes proto, abych pomohl vývojovému týmu splnit cíl sprintu?"

"Vidím nějaké překážky, které brání mně nebo vývojovému týmu ve splnění cíle sprintu?"

Vývojový tým pomocí denní schůzky kontroluje svůj postup směrem k cíli sprintu. Kontroluje, zda dosavadní postup směřuje k dokončení katalogu sprintu. [22] Denní schůzka zlepšuje komunikaci, eliminuje potřebu dalších schůzek, identifikuje překážky vývoje kvůli jejich odstranění, vyzdvihuje a podporuje rychlá rozhodnutí a zlepšuje úroveň znalostí každého člena týmu o projektu. Je klíčovou schůzkou zajišťující kontrolu a adaptaci. [22]

Sprint Review. Zhodnocení sprintu se koná na konci každého sprintu a je zaměřené na funkční část produktu, která byla v tomto sprintu vytvořena. Během vyhodnocení sprintu Scrum tým a ostatní zúčastněné strany (stakeholders) probírají výsledky sprintu. Předvedení přírůstku má za cíl získat zpětnou vazbu a podpořit spolupráci. Scrum master se stará o to, aby vyhodnocení proběhlo a aby účastníci rozuměli jeho účelu; také všem pomáhá udržet se v time-boxu (4 hod.). [22]

Vyhodnocení sprintu má následující části:

- účastníky jsou členové Scrum týmu a klíčoví zástupci zúčastněných stran, které pozve vlastník produktu;
- vlastník produktu objasňuje, které položky produktového backlogu byly dokončeny („hotovo“) a které nikoliv;
- vývojový tým diskutuje o tom, co se během sprintu dařilo, k jakým problémům docházelo a jak tyto problémy byly řešeny;
- vývojový tým demonstruje „hotové“ výsledky a odpovídá na dotazy týkající se přírůstku;
- vlastník produktu diskutuje o aktuálním stavu produktového backlogu. Na základě dosaženého pokroku odhaduje pravděpodobné datum dokončení;
- celá skupina společně navrhne, co bude řešeno dále, takže schůzka pro vyhodnocení sprintu poskytne důležitý vstup pro následující plánovací schůzku. [22]

Výsledkem vyhodnocení sprintu je revidovaný produktový Backlog, a dále jeho vytipované položky jako kandidáti na zařazení do dalšího sprintu. Produktový Backlog může být také celkově přizpůsobený novým příležitostem. [22]

Sprint Demo. Těžištěm je produkt, respektive jeho nasaditelná část, která byla vývojovým týmem vytvořena. Jde tedy o ukázkou nových dokončených částí během Sprintu. Primárním účelem této schůzky je kontrola, co vývojový tým dodal a získání zpětné vazby od zákazníka. Tím dochází k přizpůsobení plánu (produktový Backlog) pro

následující Sprint. Sprint Dema se účastní vývojový tým, produktový vlastník, Scrum Master a zákazník (uživatel) a dochází k němu na konci každého Sprintu. V ideálním případě dochází ke stanovení cíle pro následující Sprint. [9]

Na konci každého sprintu je důležité získat zpětnou vazbu od zákazníka. Zákazníkovi se během schůzky předvede, co bylo během sprintu vyvinuto a tým získá zpětnou vazbu na dokončené user stories. Prezентují se pouze dokončené user stories, tj. ty, které mají splněná akceptační kritéria, fungují, jsou otestovány, zrevidovány a jsou popsány v dokumentaci. Schůzku obvykle uvádí Product Owner a výsledky sprintu prezentují členové týmu, kteří obvykle nejlépe vědí, jak systém funguje a co zákazníkovi ukázat. [25]

Retrospective. Využití předchozího sprintu je zaměřena přímo na práci samotného Scrum týmu. Retrospektiva následuje po vyhodnocení sprintu, ale ještě před plánovací schůzkou. Při jednoměsíčních sprintech by měla trvat tři hodiny. Scrum tým má během retrospektivy sprintu příležitost ke kontrole sama sebe a k naplánování kroků, pomocí kterých dojde během příštího sprintu ke zlepšení. [22] Na rozdíl od Sprint Dema, který je zaměřen na produkt, retrospektiva je zaměřena na proces. Jde o způsob, jakým členové Scrum týmu spolupracují, včetně jejich zručností, použitých praktik a nástrojů. Účelem retrospektivy je neustálé zlepšování způsobu práce. [9]

Cílem retrospektivy sprintu je:

- kontrola toho, jak probíhal poslední sprint s ohledem na lidi, vztahy, procesy a použité nástroje;
- identifikace a seřazení hlavních aspektů, které fungovaly dobře a které je možné zlepšit;
- vytvoření plánu na zavedení jednotlivých vylepšení; tento plán je vytvářen stejným způsobem, kterým Scrum tým provádí svou běžnou práci při vývoji produktu. [22]

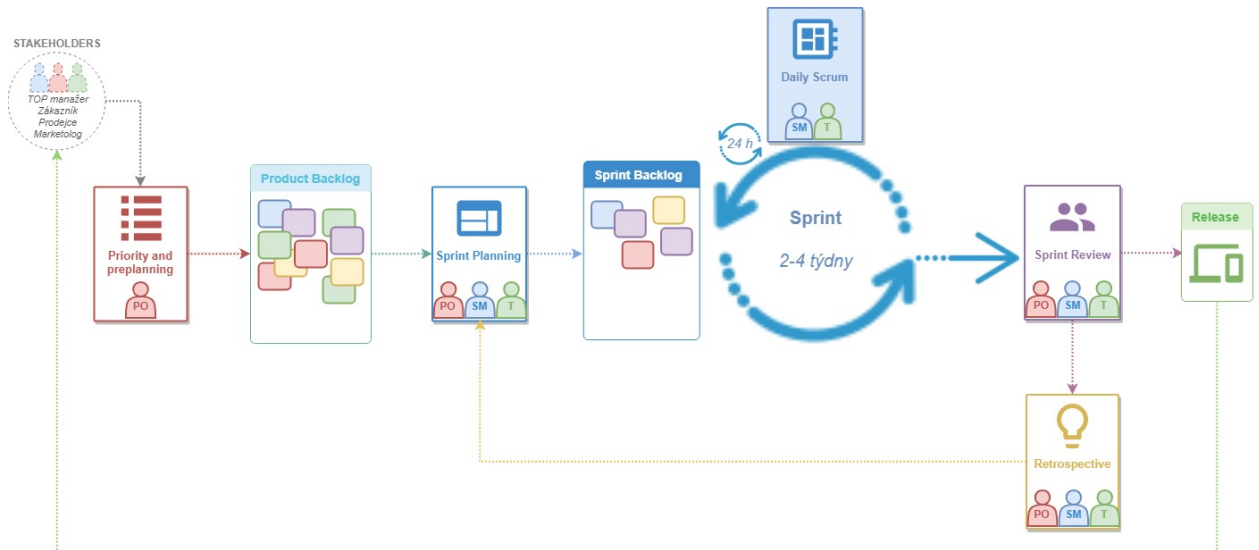
Scrum tým by měl mít na konci retrospektivy hotový seznam zlepšení, která v dalším Sprintu zavede. Zavádění těchto zlepšení v dalším Sprintu je vlastně adaptací plynoucí z kontroly, kterou Scrum tým provedl sám na sobě. I když zlepšení mohou být zaváděna kdykoliv, retrospektiva sprintu přináší týmu metodickou a formalizovanou příležitost k zaměření se na kontrolu a adaptaci. [22]

Konečný meeting. Je skutečně posledním meetingem, který je na projektu svolán. Na tomto meetingu probíhá předání veškeré dokumentace, odsouhlasení akceptačních

protokolů a dohody o spravování systému v tzv. maintenance fázi. Tu je nutné definovat a odsouhlasit smluvně. Výsledkem meetingu je formální a oficiální ukončení projektu.

Scrum se zaměřuje na kvalitativní kontrolu pracovního procesu. Proces vývoje je rozdělen na sprinty, které přísně stanoveny podle času, a mohou trvat od 2 do 4 týdnů.

Obrázek 7. Scrum proces vývoje softwaru



Zdroj: vlastní zpracování dle [12]

Obrázek 7 ukazuje postupy během Sprintu, od zadání ve formě backlogu, přes plánování sprintu, jednotlivé denní cykly až po výsledné demonstrace. Workflow v jednom sprintu zahrnuje několik fází:

- na začátku probíhá plánování a je definován Sprint Backlog,
- v průběhu vývoje tým provádí analýzu požadavků, detailní design a samotný vývoj,
- 15minutové stand-upy se konají každý den, aby mohli členové týmu upravit svou práci a shrnout průběžné výsledky,
- vyvinuté části se hned testují a případné chyby jsou okamžitě opraveny,
- sprint je zakončen schůzkou k vyhodnocení sprintu a získávání zpětné vazby od zákazníka,
- na retrospektivě, která poté následuje, tým probírá, co se podařilo a co je naopak nutné změnit. Pro měření úspěšnosti práce jsou používán Burndown chart.

Tento koloběh se opakuje až do předem stanoveného data ukončení projektu, nebo dosažení požadované podoby produktu.

3.6 Srovnání Agilního a Vodopádového přístupu

Agilní a Vodopádový přístupy k řízení projektu — dvě naprosto odlišné metodiky.

Vznik Agilního přístupu byl podnícen problémy, které se objevovaly při řízení projektů vývoje softwaru, kde šlo zejména o otázky příliš dlouhého uvolnění, příliš dlouhé doby stabilizace, složitého vykonávání změn v průběhu projektu, klesající kvality výstupů, a šlo také o skutečnost, že neúspěchy kazí pracovní morálku. [9]

Hlavním problémem bylo, že používané metody kladly vysoký důraz na detailně definovanou specifikaci vstupů, a metody přeměňující tyto vstupy na výstupy byly předem známy, jinými slovy, bylo je možné předpovědět, což pro vývoj softwaru není ideální a dalo by se uvažovat až o jeho nepoužitelnosti. [9]

Iterativní metodiky reagují na kritiku Vodopádového modelu projektového řízení. Konkrétně na nutnost dokončení jednotlivých fází, které na sebe postupně navazují. Iterativní metody řízení jsou založené na jednotlivých iteracích – fází projektu, stejně jako je tomu u Vodopádového řízení. Avšak tyto fáze mohou probíhat současně v cyklech. Jednotlivé fáze tak mohou opakovat jednu či více aktivit projektu dle potřeby. [14]

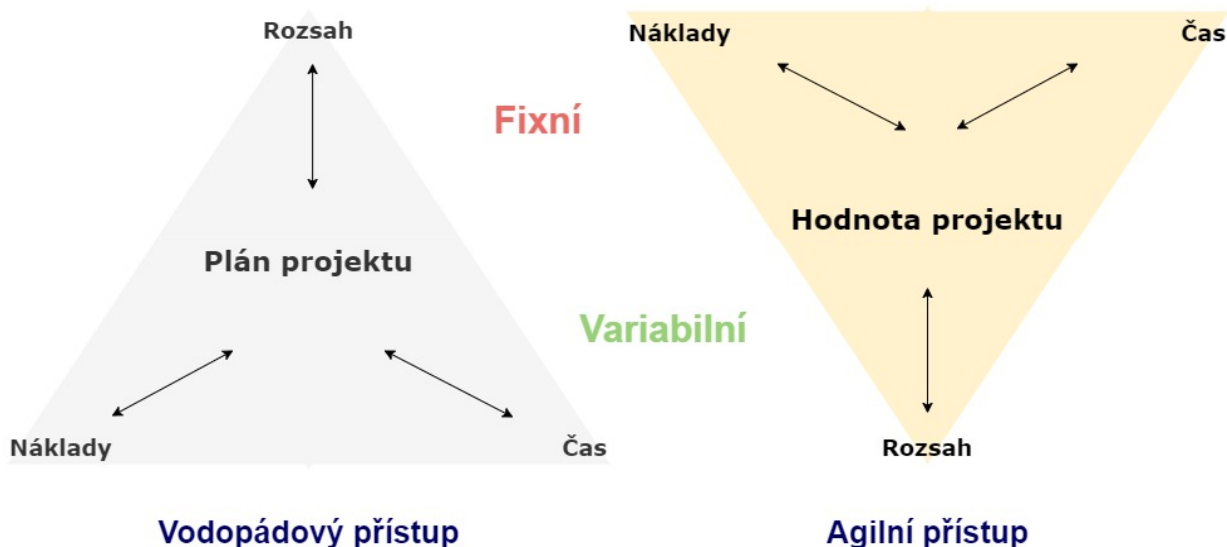
Pro Agilní přístup je charakteristické:

- týmu je zadán jasný cíl;
- tým je samo-organizující;
- tým dostává zpětnou vazbu z okolí;
- týmová práce je zaměřena na neustálé zlepšování;
- organizace má přehled o progresivitě práce v týmu;
- tým a management komunikují o pokrocích práce a rizicích. [9]

Konfrontace Agilního a Vodopádového přístupu není tak teoretická, jako spíše praktická. Metodika je navržena na základě konkrétních úkolů, množství práce, cíle zákazníka a rozpočtu. Pokud Vodopádový přístup poukazuje na konkrétní kroky v práci, pak Agilní přístup snadno přizpůsobí spotřebitele konečného výrobku a požadavky zákazníka.

Agilní přístup je ideální pro IT společnosti a start-up projekty. Vodopádový přístup neztrácí pozice ve stavebních projektech nebo projektech, kde klíčovou omezující podmínkou je doba trvání projektu. Agilní a Vodopádový přístupy se liší v proměnných projektového trojimperativu.

Obrázek 8. Projektový trojimperativ



Zdroj: vlastní zpracování dle [9]

Z Obrázku 8 se dá odvodit, že Agilní přístup považuje za neměnné čas a zdroje. Proměnnou veličinou je rozsah, který je přizpůsobován prioritám zákazníka. Naopak při Vodopádovém přístupu stanovena množina požadavků (rozsah) je považována za neměnnou. Čas a náklady, potřebné na realizaci, jsou proměnnými veličinami.

Nejdůležitější rozdíly shrnuje Tabulka 4. Srovnání Agilního a Vodopádového přístupu. Z tabulky je patrné, že Agilní přístup přináší mnohem vyšší flexibilitu a tím zvyšují přínos pro zákazníka při stejné investici v rámci minimálního časového nasazení.

Tabulka 4. Srovnání Agilního a Vodopádového přístupu

	Agilní přístup	Vodopádový přístup
Základ	–předem jasný rámcový cíl, ale není definovaný všechny požadavky	–předem jasně daný plán projektu
Princip	–těsná spolupráce mezi projektovým týmem a zákazníkem	–detailní plánování jednotlivých kroků a následné dodržování postupu
Zákazník	–může zasáhnout do procesu, podle jeho aktuálních potřeb produkt se upravuje	–definuje produkt jenom na počátku projektu

Tým	<ul style="list-style-type: none"> –malý cross-funkční tým, –vysoká míra komunikace, –celková odpovědnost za výsledek, –manažer nemá roli vedoucího týmu 	<ul style="list-style-type: none"> –velký tým specialistů, –komunikace pomocí reportů a zpráv, –za výsledek každé fáze jsou zodpovědní různí specialisté, –manažer plní roli šéfa a je plně zodpovědný za práci podřízených
Vývoj	<ul style="list-style-type: none"> –rozpočet není přesně definován a závisí na množství vykonané práce –dokumentace se tvoří ve chvíli, kdy je zapotřebí a obsahuje pouze nejdůležitější údaje –změny jsou možné mezi jednotlivými iteracemi na základě zpětné vazby –Value Driven Development: produkt se dodává po částech a parametry se mění na základě zpětné vazby od zákazníka 	<ul style="list-style-type: none"> –rozpočet je definovaný podle plánovaného předpokladu vykonané práce –dokumentace je připravena v maximální míře předem –změny jsou nemožné nebo možné jen s velkými obtížemi, protože ovlivní termín i rozpočet –Plan Driven Development: dodaný produkt odpovídá většinově parametrům dle zadání projektu, je možné jeho využívat až v okamžiku plného předání
Vhodné projekty	<ul style="list-style-type: none"> –je vhodný pro malé týmy a méně vyspělé týmy, které potřebují pevnější řízení a nepřiliš rozsáhlé projekty 	<ul style="list-style-type: none"> –je vhodný pro specifické, rozsahově malé a řešením jednoduché projekty, které jsou vyvíjeny několika málo spolupracujícími lidmi

Zdroj: vlastní zpracování dle [14], [27]

Nejběžnějšími důvody, proč společnosti přecházejí na agilní metodiky řízení projektů, jsou vyšší míra úspěšnosti projektu, dodržování harmonogramu, lepší týmová spolupráce a motivace, efektivita vynaloženého úsilí, menší odmítavý postoj koncového uživatele. Přejít na agilní metodiky neznamená jen změnu procesu nebo vyplňování jiných formulářů, ale je to celá filozofie, kterou firma musí pochopit a přijmout.

4 Popis a rozbor vybrané organizace

4.1 Představení společnosti RIA.com

Na počátku 21. století internet dodal impuls pro změnu v oblasti obchodování. Zákazníci si oblíbili nakupování přes internet, a proto se museli přizpůsobit i prodejci. Vznikla potřeba vytvoření web-stránek, které pomohou setkat se kupujícím a prodávajícím. Toto jsou stránky, které společnost RIA.com vytváří.

Společnost RIA.com je mladá a moderní softwarová firma, která se zabývá především vývojem a rozvojem vlastních internetových portálů a poskytováním poradenských služeb svým zákazníkům. Společnost byla založena v roce 2004 a v současnosti působí na Ukrajině. Hlavní sídlo společnosti se nachází ve městě Vinnycja.

Ve společnosti RIA.com pracuje v současné době 200 zaměstnanců (průměrný věk je 25 let). V důsledku dynamického rozvoje společnosti a časové náročnosti řešených projektů lze v blízké budoucnosti očekávat nárůst především programátorů a zaměstnanců v oblasti vývoje.

Společnost má velkou řadu zákazníků ve všech segmentech trhu. Mezi zákazníky portálu AUTO.RIA.com naleznete oficiální dovozce a prodejce automobilových značek na Ukrajině, středně velká a velká komerční autocentra, lokální autobazary, soukromé prodejce.

RIA.com se dlouhodobě zaměřuje na kvalitu svých služeb a liší se od konkurentů tím, že vytváří "ověřený internet" a kontroluje obsah, který publikuje na svých stránkách. Na úkolech pracují celá oddělení, týmy inspektorů a moderátorů. Pro ověření se používají algoritmy strojového učení, databáze, partnerství se státními organizacemi (např. státní kontrola auta).

RIA.com má největší zákaznickou základnu na Ukrajině a je zde považována za lídra internetového obchodu. Projekty společnosti:

1. AUTO.RIA je automobilový portál č.1 na Ukrajině pro ty, kteří chtějí koupit nebo prodat auto. Na webu je denně přidáno zhruba 12 000 nových inzerátů. Web obsahuje veškeré služby pro motoristy: nákup a prodej automobilů, zprávy, katalog prodejců automobilů po celé Ukrajině, služby aut a mnoho dalšího;
2. DOM.RIA je lídrem na Ukrajiněském realitním internetovém trhu. Web poskytuje služby pro majitele domů, realitní kanceláře, soukromé realitky, banky, developery;

3. RIA.com - portál zboží a služeb, který zahrnuje 12 kategorií výrobků: vše pro automobily, počítače a telefony, domácí spotřebiče, služby, domácí zvířata a další.

Přibližně 12 milionů web-uživatelů přistupuje měsíčně k prostředkům AUTO.RIA, DOM.RIA a RIA.com. To znamená, že každý třetí Ukrajinský uživatel internetu je obeznámen s produkty RIA.com.

Společnost RIA.com se snaží klást velký důraz na sledování aktuálních trendů v oblasti své činnosti, pracovat na neustálém zlepšování svých produktů a vycházet jak ze zkušeností vlastních, tak svých klientů.

4.2 Popis internetového portálu AUTO.RIA.com

Jedním z neustále se rozvíjejících portálů je vlastní start-up s názvem AUTO.RIA.com (dále jen AUTO.RIA.). Název samotný evokuje přímo webovou stránku, protože ta je z hlediska uživatele klíčová. Jedná se o úplně první portál na Ukrajině tohoto druhu.

V roce 2004 se ve společnosti RIA.com zrodil první nápad na nový start-up, který spočíval v kopírování inzerátů na prodej automobilů z regionálních novin na webovou stránku. Podstatou nápadu na vznik tohoto webu byl především fakt, že ve městě Vinnyca existují dva velké autobazary. Na těchto trzích se zúčastnili prodejci a kupci ojetých vozů z celého regionu. AUTO.RIA. měl za cíl přinést konkrétnímu kupujícímu nejlepší možnou nabídku ojetých vozů na jednom místě. Uživatel však na základě nejnižší ceny měl neustále také vidět alternativní nabídku. To pomáhalo zejména v případě, kde není preferovaná konkrétní značka, ale ve výsledku může mít na rozhodnutí zákazníka vliv.

První verzi aplikace se podařilo poměrně bez problémů spustit v srpnu roku 2004. Protože počet uživatelů byl od počátku velmi malý, dochází paralelně se spuštěnou verzí k dalšímu vývoji a rozvoji webu. Z technického hlediska a z hlediska pohodlí byl tento web pro zákazníka nedostatečný, výsledky vyhledávání nebyly přehledné a zákazníka spíše omezovaly, než aby mu nabízely lepší a širší možnosti.

Vzhledem k průběžné práci na nové verzi programu bylo AUTO.RIA v roce 2005 aktualizováno na verzi 2.0. Ve druhé fázi byl celkově přepracován a zjednodušen vzhled stránek, které se tak staly mnohem přehlednější a srozumitelnější novým uživatelům. Došlo zejména k technologickému vylepšení algoritmů pro vyhledávání inzerátů. Všechny

změny, které měly přímý vliv na uživatele, byly zpracovány v souladu se zpětnou vazbou od zákazníků.

V několika následujících iteracích se ovšem přepracování portálu ukázalo jako nezbytné pro budoucí rozvoj. Z tohoto důvodu byla zvolena náročnější, avšak mnohem snáze rozšiřitelná technologická platforma, která tvoří základ další generace AUTO.RIA. Byl sestaven nový, zeštíhlený tým lidí, který zahrnoval pouze grafika, projektového manažera, obchodníka a 4 developery, z nichž každý se orientoval na jednu hlavní část webu. Z pohledu zákazníka se však příliš nezměnilo a celý projekt tak získal zhruba stejný ohlas jako předchozí verze. Na rozdíl od hladkého průběhu vývoje verze 1.0 se objevily první náznaky rizika ve formě nedostatečného počtu vývojářů. Souběžná práce ukázala nutnost navýšení počtu vývojářů a urychlení dostupnosti verze.

Dnes AUTO.RIA je automobilový portál č.1 na Ukrajině a díky tomu prodejci mohou prodávat online své auta a portál se stává místem, kde lze koupit auto, auto příslušenství, pojištění a vše potřebné pro motoristy pod jedním zákaznickým účtem. Projekt nemá zadavatele, v roli sponzorů vystupují pouze ředitel a ostatní jednatelé společnosti, kteří projekt podpořili.

4.3 Projektové řízení ve společnosti RIA.com

4.3.1 Organizační struktura společnosti

Společnost RIA.com je projektově orientovaná organizace, ve které se značná část procesů a činností odehrává formou projektů. Její organizační struktura má prvky maticové organizační struktury. To je klasická vertikální liniová struktura, která je kombinována s horizontálně fungujícími vytvářenými týmy, které se věnují projektům. Princip spočívá v uvolňování lidí pro projekt pouze na omezenou dobu nebo na částečný úvazek a jinak se ponechávají jako součást liniového oddělení. Jednotliví členové projektových týmů tedy zůstávají na svých stálých funkčních pozicích v rámci stávající organizační struktury, na kterých plní běžné i projektové úkoly, a to z několika různých projektů.

Při maticovém uspořádání vznikají dvě skupiny útvarů:

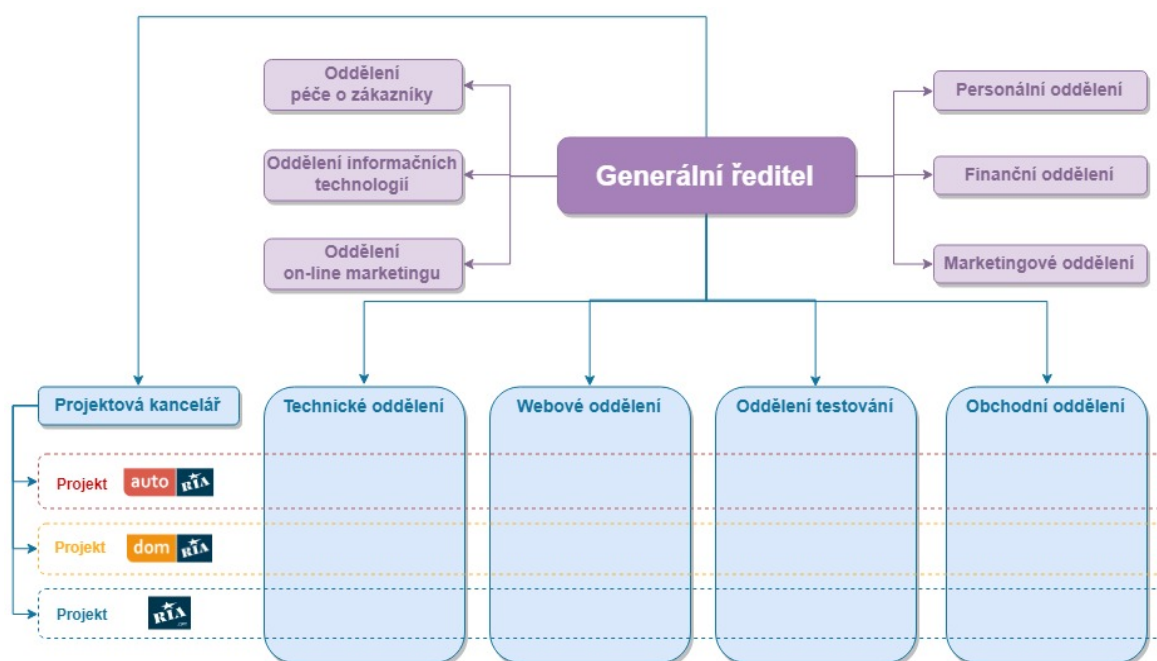
1. skupina projektových útvarů (AUTO.RIA.com, DOM.RIA.com a RIA.com)
2. skupina funkčních útvarů (marketing, péče o zákazníky, finance, personalistika).

Každý z projektových týmů má svého projektového manažera, který má plnou pravomoc stanovit priority a řídit práci lidí přiřazených na projektu.

Jednotlivé pracovníci také spadají do řídicích kompetencí ředitelů svých oddělení. Linioví manažeři zde mají roli správců zdrojů. Starají se svým lidem o potřebné vybavení, o vhodná školení a vůbec o co nejlepší připravenost svých lidí k vykonávání prací na projektech. Maticová organizační struktura tak rozděluje pravomoc vydávat příkazy a provádět rozhodnutí mezi liniové a projektové manažery.

Další část procesů společnosti má prvky klasické liniové organizační struktury. Pozice a vztahy nadřízenosti a podřízenosti jsou uspořádány a orientovány vertikálně. Každý nadřízený má jasně přidělené podřízené a každý podřízený má jasně přiděleného nadřízeného. Organizační struktura společnosti je zobrazena na Obrázku 9.

Obrázek 9. Organizační struktura společnosti RIA.com



Zdroj: vlastní zpracování

Přínosy maticové organizační struktury jsou zvýšená motivace a pocit odpovědnosti zaměstnanců, rozvoj jejich zkušeností a kvalifikace a reakce na požadavky zákazníků.

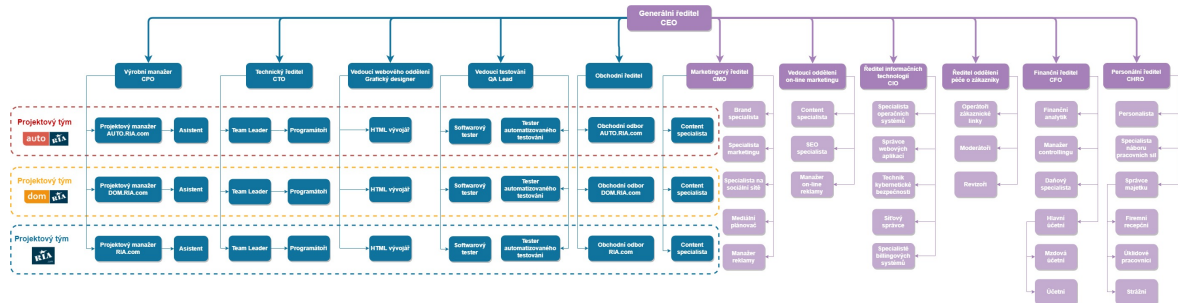
Problémy maticové organizační struktury jsou vztahy dvojí podřízenosti, které často vedou k nedorozuměním a konfliktům; časově náročnější rozhodování, které se uskutečňuje většinou formou porad; nezbytné školení v mezilidských vztazích, neboť práce v týmech vyžaduje neustálou spolupráci.

4.3.2 Řídící struktura společnosti

Určitým nebezpečím v maticové struktuře je přílišné, neefektivní využívání zdrojů projektovými manažery, což snižuje efektivitu celé organizace. Aby k tomu nedocházelo,

je nutné použít dobře fungující kontrolní mechanismus. Takový mechanismus je Projektová kancelář, jejíž odpovědností je koordinace všech projektů a projektových manažerů. Řídící struktura společnosti je zobrazena na Obrázku 10.

Obrázek 10. Řídící struktura společnosti RIA.com



Zdroj: vlastní zpracování

Projektová kancelář je podpůrný administrativní orgán řízení projektu, je tvořena manažery a asistenty projektů a podílí se na každodenním operativním řízení projektů. [28]

Projektová kancelář se skládá z projektových manažerů a jejich asistentů, vedoucím je Výrobní manažer (CPO). CPO je zodpovědný za celé produktové portfolio společnosti. Jeho hlavní úlohou je identifikovat vhodné příležitosti na trhu a přizpůsobovat nabídky společnosti tak, aby co nejlépe do poptávky trhu zapadla.

Hlavní úkoly projektové kanceláře jsou:

- iniciace a řízení projektů, programů a portfolií;
- sestavování projektových týmů;
- koordinace procesů v projektech;
- obsluhu všech administrativních a dokumentačních potřeb projektů;
- komunikace a zajištění hladkého chodu všech informačních toků;
- reporting o průběhu a o výsledku projektů a programů. [18]

Projektová kancelář dále zajišťuje podporu a školení pro manažery a členy projektových týmů.

Klíčovou osobou je projektový manažer, pod jehož přímým vlivem je veškeré projektové dění od tvorby projektového plánu přes obsazení jednotlivých odborných pozic projektu, koordinaci úkolů, finalizaci a předání výstupů projektu zákazníkovi až po administrativní uzavření projektu. Projektový manažer je osoba, odpovědná za splnění cílů projektu při dodržení všech stanovených charakteristik projektu. [29]

Projektový manažer má pravomoci a odpovědnosti vztažené k projektu, z hlediska maticové organizační struktury horizontálně orientované, a je odpovědný za projekt ve všech jeho fázích.

Úkoly projektového manažera jsou:

- příprava pro každou etapu projektu zadání pro realizační tým (včetně stanovení termínu splnění zadání);
- plánování časování a zdroje projektu a odpovědnost za dodávku projektu ve stanoveném čase a rozsahu;
- zajištění a kontrola plnění úkolů a kvality výstupů a odpovědnost za celé řešení;
- kontrola průběhu projektů podle Harmonogramu (porovnání aktuálního stavu projektu vzhledem k časovému plánu);
- kontrola úkolů a postupu (porovnání aktuálního stavu projektu se staven projektu při poslední kontrole a kontrola splněných úkolů realizačním týmem);
- kontrola nákladů vůči rozpočtu (porovnání pracností stráveného času na projektu s celkovým počtem vzhledem k aktuálnímu datu);
- aktualizace reportu (dvakrát měsíčně vyplňovat report o aktuálním stavu projektu);
- příprava závěru projektu a hodnocení celého projektu (dosažení cíle projektu v daném čase, finanční stránky projektu);
- sbírání zkušeností z projektu (co se povedlo, co se nepovedlo, jaké byly příčiny problémů a jejich řešení);
- archivace veškeré projektové dokumentace (projektové, technické a obchodní dokumenty).

Projektový manažer nedisponuje žádným zdrojem přímo, zdroje jsou spravovány příslušnými liniiovými manažery a jsou přiděleny projektu podle parametrů schválených v plánu projektu (rozpočet, čas, rozpis práce, technologické potřeby). Projektový manažer pak řídí a koordinuje využívání těchto zdrojů uvnitř jim řízeného projektu. Pro úspěšné řízení projektu je důležité, aby měl projektový manažer dobré pracovní vztahy s příslušnými liniiovými manažery a aby průběžně informoval o stavu projektu a o případných změnách jak ve vertikálním směru (v hierarchii projektových řídicích složek), tak ve směru horizontálním. [29]

Projektový tým se společně podílí na vývoji jednoho web portálu. Projektový tým se skládá z projektového manažera a asistenta, team leadera a týmu programátorů. Tým nemá stálého testera, HTML vývojáře nebo kontent specialisty. V případě potřeby projektový

manažer odesílá žádost řediteli oddělení, který pak jmenuje odpovědného specialista. Výsledkem je, že na projektu pracují pokaždé noví specialisté, kteří jsou pouze povrchně obeznámeni s projektem a nemohou tak vzít v úvahu všechny nuance. Obchodní oddělení se vývoje nezúčastňuje, ale prodává již implementované portálové služby.

Požadavky na funkcionalitu, úpravy nebo změny přicházejí od projektového manažera, který je v tomto případě vedoucí týmu, nebo přímo od zákazníka přes Informační systém Kayako. Členové týmu si práci vybírají sami dle důležitosti jednotlivých úkolů a dle svých nejlepších znalostí jednotlivých modulů web portálu. Celý proces plánování a kontroly plnění jednotlivých úkolů řídí projektový manažer.

Cílem Oddělení péče o zákazníky je maximalizace dlouhodobé spokojenosti zákazníků. Nejlepším oceněním dobré péče o zákazníka je vysoká míra reputace značky a vysoká míra loajality a retence zákazníků. Vztah se zákazníkem se buduje s každou interakcí. Není podstatné, jestli je to prodej, dodávka, telefonický kontakt, nebo také přijetí jeho stížnosti či reklamace. Každý jednotlivý kontakt zanechává nějaký dojem a vytváří celkový vztah zákazníka ke společnosti.

Funkce Oddělení péče o zákazníky jsou technická podpora zákazníků (Informační systém Kayako), řízení a vyřizování reklamací, Back Office (např. zpracování dat klientů, analytické výstupy), moderování inzerátů a dalších údajů před zveřejněním na webu.

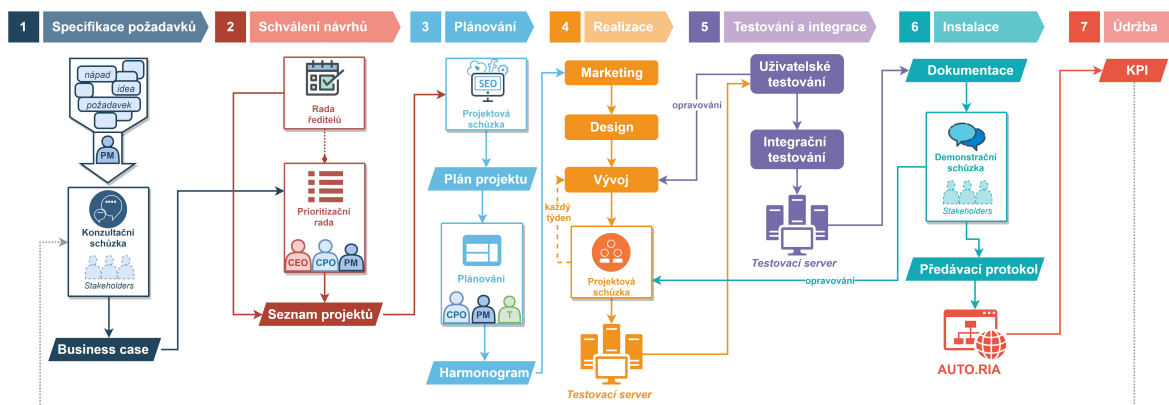
Péče o zákazníky využívá Informační systém Kayako, různé aplikace pro rychlé řešení zákaznických problémů, sociální a komunitní sítě pro komunikaci s nimi a získávání zpětné vazby.

Zákazníci své požadavky zadávají přes webové rozhraní. Díky tomu se požadavky dostanou do Informačního systému Kayako, kde je přijmou operátoři. Ti ověří, zda má zákazník na požadavek nárok a určí jeho prioritu. Následně je požadavek předán na zodpovědného projektového manažera, který požadavek přijme, vyřeší a předá zpět na support, který informuje zákazníka o způsobu řešení. Na konci každého měsíce vypracovává supportní tým pro každého projektového manažera report, ve kterém je uveden přehled realizovaných požadavků, rozdělení požadavků na záruční nebo mimozáruční a počet odpracovaných hodin na řešení požadavků.

4.3.3 Metodika a nástroje řízení projektů

Projekty ve společnosti RIA.com jsou řízeny pomocí Vodopádového přístupu. Jeho hlavní předností je jednoduchost, jak v jeho pochopení, tak i v řízení. Obrázek číslo 11 podpisuje jednotlivé fáze řízení projektů ve společnosti RIA.com.

Obrázek 11. Životní cyklus projektu ve společnosti RIA.com



Zdroj: vlastní zpracování

Tyto fáze jsou pevně nastaveny a není možné mezi nimi přeskakovat, vracet se zpět nebo některé fáze opakovat. Každá jednotlivá fáze je zakončena zpracováním předem daných dokumentů. Mezi jednotlivými fázemi je schvalovací proces, přes který musí všechny dokumenty projít, aby vývoj mohl pokročit do další fáze. Máme díky tomu dobrý přehled o stavu projektu, a jaká část je již splněna. Charakteristickým rysem Vodopádového modelu jsou sekvenčně seřazené fáze bez iterací.

1. Specifikace požadavků. Nápad pro budoucí projekty zadávají nejčastěji Oddělení péče o zákazníky a Obchodní oddělení na základě požadavků uživatelů. Také každý zaměstnanec společnosti může podat vlastní návrh na realizaci na webu. Návrh musí být založen v Informačním systému Kayako a musí obsahovat obecné zadání projektu s jeho obecně definovanými přínosy.

Projektový manažer sestavuje list obecně zadaných požadavků. Na základě sestavených požadavků jsou přizváni jednotliví zástupci relevantních oddělení a koná se tak zvaná Konzultační schůzka. Výstupem je konsolidace požadavků, identifikace všech dotčených systémů a návrh obecného řešení. Všechny požadavky jsou zafixovány a žádný další požadavek není bez změnového řízení do projektu přijat. Výstupem této fáze je Business case projektů, který projektový manažer bude prezentovat na Prioritizační radě.

2. Schválení návrhů. Schválení návrhů má na starosti CPO. Výstupem této fáze je schválený předběžný Rozpočet a Seznam projektů.

Veškeré požadavky, které mají dopad na web portály zpracovává projektový manažer a schvaluje generální ředitel. Pak jejich prioritita a období provádění prochází schvalováním na Prioritizační radě. Prioritizační rada je orgán podřízený Radě ředitelů a skládá se z generálního ředitele, CPO, projektových manažerů a vedoucích oddělení, kteří se budou podílet na realizaci budoucích projektů.

Na Prioritizační radě projektové manažeři:

1. informují o výsledcích realizovaných projektů a o stavu probíhajících projektů;
2. prezentují Business case projektů, které budou realizovány příští měsíc, aby stanovit priority realizace. Hlavní roli pro prioritu hraje především finanční přínos společnosti, předpokládané náklady, nebo reakce na konkurenci.

Prioritizace strategických projektů a rozdělování investičního rozpočtu probíhá na čtvrtletní Radě ředitelů. Rada ředitelů je nejvyšším prioritizačním orgánem a zároveň i nejvyšším eskalačním bodem. Projekty, schválené Radou ředitelů, mají nejvyšší prioritu, protože jsou přímo spojeny s implementací strategie společnosti.

Projektová kancelář spravuje a hlídá všechny projekty napříč firmou a řídí projektovou Road mapu.

3. Plánování. Po schválení návrhů sestavuje projektový manažer plány projektů, které budou realizované příští měsíc. Na tvorbě toho plánu se podílí celý projektový tým. Jedná se především o rozfázování harmonogramu a rozpad jednotlivých činností na kratší časové úseky, které jsou lépe kontrolovány a řízeny. Dále následuje odhad pracnosti pro fázi realizace. Jednotliví účastníci budoucího projektu poskytují projektovému manažerovi odhad pracnosti za jejich konkrétní oblast. Navíc projektový manažer a klíčoví stakeholderi se domluví na procesu řešení změn. Výstupem této fáze je Plán projektu a odhad pracnosti, odsouhlasen všemi stranami.

4. Realizace. Dále sleduje realizace projektu. Fáze realizace začíná detailním designem požadovaného řešení. Dle detailního řešení je prováděna samotná realizační práce. V průběhu realizace projektu projektový manažer svolává týdenní projektovou schůzku. Schůzky se účastní celý projektový tým. Schůzka slouží k revidování stavu na projektu, k řešení aktuálních problémů projektu a k diskusi o nápadech pro budoucí projekty.

V případě, že dojde v průběhu realizace ke změně, projektový manažer svolává neplánovanou Projektovou schůzku, na které bude rozhodnuto o změně v projektu. Úkolem projektového manažera je představit situaci/problém a také několik návrhů řešení

tak, aby bylo rozhodnuto o nejlepším řešení. V případě změny, která má vliv na rozsah projektu je nutné svolat Prioritizační radu, která rozhodne o řešení situace. Změny v projektu, které schválí Prioritizační rada, musí projektový manažer zapracovat do plánu projektu. Obecně platí, že přidání téhož požadavku do projektu na začátku je mnohem snazší, kratší a levnější než přidávání požadavku v rámci změnového řízení.

5. Testování a integrace. Po dokončení realizační práce na projektu je nutné celý funkcionál portálu otestovat. Po úspěšném dokončení testů včetně integračního a uživatelského testování lze aktualizovat web portál pro všechny uživatele. Softwarový tester provádí samotné vykonávání testů, analyzuje a zpracovává výsledky testů, připravuje a udržuje vstupní a výstupní dokumenty a data potřebná pro testování. Tester automatizovaného testování aktualizuje a doplňuje běžné testy.

6. Instalace. Po úspěšném dokončení testů včetně integračního a uživatelského testování je aktualizován testovací server. Projektový manažer připravuje předávací projektovou dokumentaci a svolává Demonstrační schůzku, kde za účasti všech stakeholderů ukazuje implementovanou funkcionalitu na testovacím serveru. Součástí schůze je také přenos funkčnosti projektu na podporu Oddělení péče o zákazníky. Po podepsání předávacího protokolu Vedoucím Oddělení péče o zákazníky lze aktualizovat web portál pro všechny uživatele.

7. Údržba. Po spuštění aktualizací je společně s týmem provedeno zhodnocení projektu podle klíčových metrik (KPI). Probíhá měření dosažených hodnot vybraných metrik a srovnání s plánovanými hodnoty. Výsledky KPI se přímo promítají do hodnocení projektového manažera. KPI přímo ovlivňují úspěšnost vize organizace, vedou k dosažení cílů organizace a vylepšení hospodářských výsledků.

Pokud se projekt ukázal jako neúspěšný, je to neúspěchem projektového manažera. Projektový manažer naléhavě svolává Konzultační schůzku pro generování nápadů a začíná se nový životní cyklus projektu.

Hlavním nástrojem pro sběr požadavků je systém Kayako. Kayako umožňuje poskytovat výjimečné služby zákazníkům ve více jazycích v živém chatu, e-mailu, Facebooku a Twitteru. Kayako pomáhá řídit požadavky a konverzace na jakémkoli kanálu a lépe podporovat zákazníky. Pomocí Kayako Oddělení péče o zákazníky a Obchodní oddělení komunikují se zákazníky. Ze strany projektového týmu uživatelem je projektový manažer, který je zodpovědný za sběr a analýzu požadavků na nové verze produktu.

Klasickým nástrojem pro plánování je Ganttův diagram s plánovanými zdroji pro všechny projekty měsíc dopředu. Ganttův diagram se sestavuje v Excelu. Výhodou Ganttova diagramu je, že se dá poměrně jednoduše zjistit, zda se projekt stihne dodat včas se všemi požadavky, které obsahuje.

4.4 Zjištění problémů projektového týmu při vývoje softwaru

Koncem roku 2017 se společnost RIA.com začala potýkat s určitými problémy v oblasti vývoje softwaru. Problémy vznikly především na základě dynamického rozvoje společnosti a rostoucího počtu nových požadavků k web portálům a pracovníkům.

K identifikaci problémů bylo navrženo provést průzkum všech zaměstnanců, zapojených do vývoje a rozvoje webových portálů. Pro provedení průzkumu byla použita modifikovaná metoda “360° zpětná vazba”. V klasické verzi 360stupňová zpětná vazba nabízí lidem jedinečnou možnost zjistit, jak jejich pracovní chování vnímají jejich nadřízení, kolegové, podřízení, interní a externí zákazníci a dodavatelé. Důležitou součástí je také sebehodnocení. Zpětná vazba je zpravidla anonymní, což znamená, že hodnocený nemá informaci o tom, jaké hodnocení od koho obdržel.


Modifikace metodiky spočívala v tom, že bude ohodnocen celý projektový tým. Pro hodnocení byl vybrán projektový tým AUTO.RIA, protože je to nejdůležitější portál v portfoliu společnosti a jeho problémy je nutné řešit především. Projektový tým byl hodnocen „ze všech možných stran“: od nadřízených (CEO, CPO); od kolegů a vedoucích pracovníků z jiných oddělení (Marketingové, Personální, Finanční oddělení); od ostatních vývojových týmů a jejich vedoucích pracovníků (Týmy DOM.RIA a RIA.com, Technické, Webové a QA oddělení); od zákazníků (Obchodní oddělení, Oddělení péče o zákazníky); sebehodnocení vlastními členy týmu (projektový manažer, asistenti PM, team leader, vývojáři).

S pomocí projektového manažera AUTO.RIA byly vytvořeny dva dotazníky. První dotazník sebehodnocení je určen pro členy týmu. Obsahuje 10 tvrzení, které musí být v závislosti na spokojenosti hodnoceny na stupnici od 1 do 5. Následují 4 otevřené otázky, kde každý zaměstnanec může napsat svůj názor na problémy, které vznikají při zadávání úkolů, plánování a při vývoji softwaru. Na konci dotazníku je možnost doplnit návrhy pro maximalizaci výkonu týmu. Dotazníkové šetření bylo uskutečněno rozesláním emailů s odkazem na online Google forms. Tento servis byl zvolen proto, že je bezplatný a pohodlný pro vyplnění.

Obrázek 12. Dotazník sebehodnocení

Sebehodnocení

Ráda bych Vás požádala o vyplnění dotazníku, který poslouží jako podklad pro mou diplomovou práci "Projektové řízení ve společnosti RIA.com"



Počet let ve společnosti *

0 1 2 3 4 5

do 1 roku 5 a více let

Ohodnoťte Vaši celkovou spokojenost (1 - nedostatečné, 5 - výborné) *

	1	2	3	4	5
Informování o strategii a cílech oddělení a společnosti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Motivace k splnění firemních cílů	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Včasná informace, potřebná k provedení práce	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Uznání pracovního úsilí	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Míra kontroly Vaší práce	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pracovní komunikace a spolupráce uvnitř týmu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Komunikace při pracovních schůzkách	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pracovní iniciativa a produktivita týmu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Řešení pracovních problémů	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Zpětná vazba od nadřízeného	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Jaké problémy existují při zadávání úkolů?

Odpověď

Jaké problémy existují při plánování?

Odpověď

Jaké problémy existují při vývoji software?

Odpověď

Co byste vylepšili pro maximální efektivitu týmu?

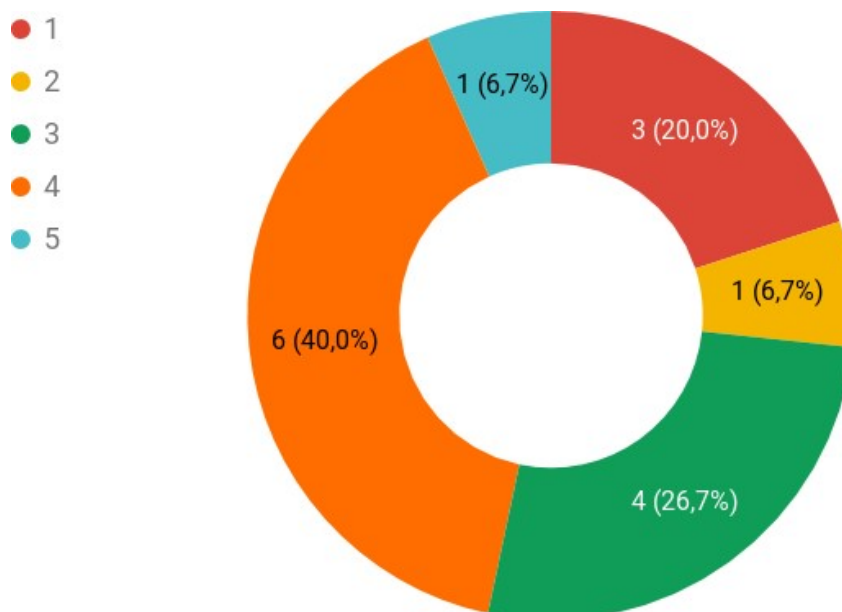
Odpověď

[ODESLAT](#)

Zdroj: vlastní zpracování

Dotazník sebehodnocení vyplnilo 15 členů týmu AUTO.RIA. Diagram č. 13 zobrazuje počet let práce ve společnosti.

Obrázek 13. Počet let ve společnosti RIA.com (členové týmu)



Zdroj: vlastní zpracování

Členové týmu ohodnotili spokojenost s pracovními aspekty (na stupnici od 1 do 5). Výsledky průzkumu jsou uvedeny na obrázku č. 14. Hodnoty 5 (zcela spokojen), 2 (spíš nespokojen) a 1 (zcela nespokojen) v žádném tvrzení nepřevládají.

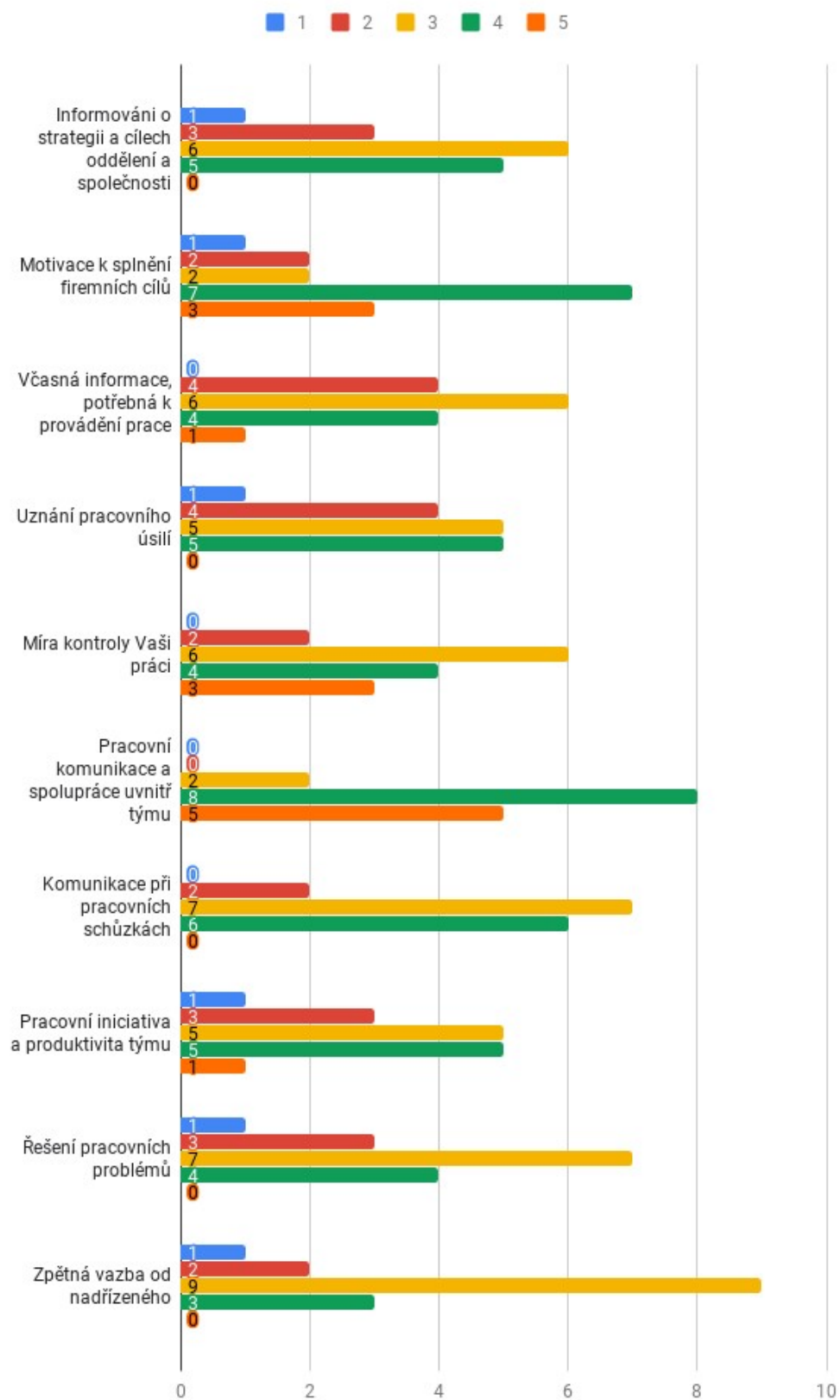
Hodnota 4 (spokojen) převládá v následujících tvrzeních:

- *Pracovní komunikace a spolupráce uvnitř týmu (53,33 %)*
- *Motivace k splnění firemních cílů (46,67 %)*
- *Uznání pracovního úsilí (33,33 %)*
- *Pracovní iniciativa a produktivita týmu (33,33 %)*

Hodnota 3 (spíš spokojen) převládá v následujících tvrzeních:

- *Zpětná vazba od nadřízeného (60 %)*
- *Řešení pracovních problémů (46,67 %)*
- *Komunikace při pracovních schůzkách (46,67 %)*
- *Informování o strategii a cílech oddělení a společnosti (40 %)*
- *Včasná informace, potřebná k provádění práce (40 %)*
- *Míra kontroly Vaší práci (40 %)*
- *Uznání pracovního úsilí (33,33 %)*
- *Pracovní iniciativa a produktivita týmu (33,33 %).*

Obrázek 14. Hodnocení spokojenosti členů týmu AUTO.RIA




Zdroj: vlastní zpracování

Obrázek 15. Dotazník hodnocení projektového týmu AUTO.RIA

Hodnocení týmu AUTO.RIA

Ráda bych Vás požádala o vyplnění dotazníku, který poslouží jako podklad pro mou diplomovou práci "Projektové řízení ve společnosti RIA.com"



Počet let ve společnosti

0 1 2 3 4 5

do 1 roku 5 a více let

Ohodnoťte Vaši celkovou spokojenost

	Spíše ano	Ano	Spíše ne	Ne
Jste spokojen s informování o strategii a cílech projektu AUTO.RIA?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Jste spokojen s rychlostí a kvalitou dodávek?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Jste spokojen s kvalitou vedení projektu?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Plní členové týmu své funkce?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Komunikace při pracovních schůzkách funguje správně?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Neformální komunikace funguje správně?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projektový manažer je otevřený novým návrhům?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Spory a konflikty jsou řešeny správnými komunikačními kanály?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Jaké problémy existují při zadávání úkolů?

Odpověď _____

Jaké problémy existují při plánování?

Odpověď _____

Jaké problémy existují při vývoji software?

Odpověď _____

Co byste vylepšili pro maximalizaci efektivní spolupráce?

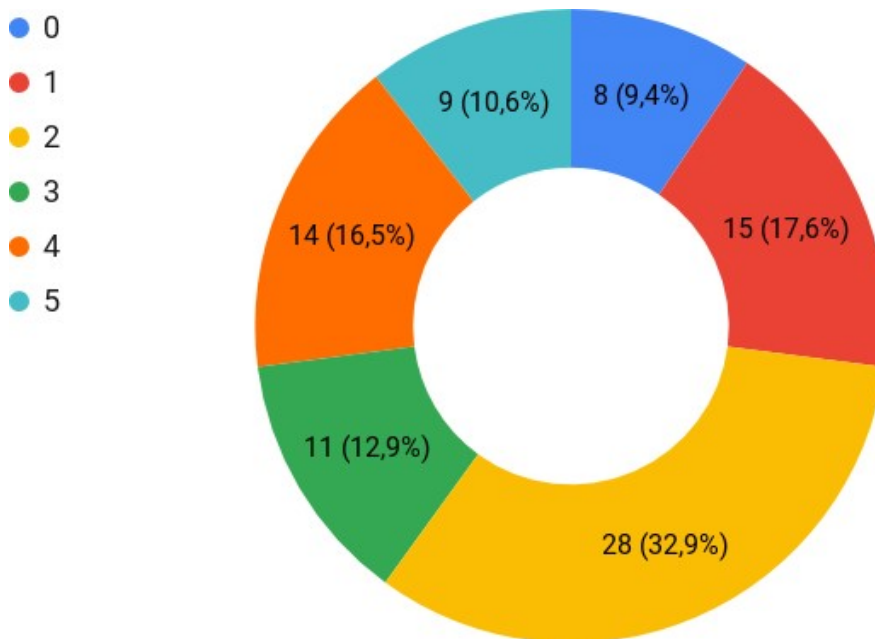
Odpověď _____

ODESLAT

Zdroj: vlastní zpracování

Zpětnou vazbu poskytli jen ti lidé, kteří často komunikují s projektovým týmem AUTO.RIA. Každému hodnotiteli byla zaslána e-mailem instrukce spolu s unikátním odkazem na webovou stránku, kde byl připraven dotazník. V rámci dotazníkového šetření bylo osloveno 85 kolegů. Termín jejich práce ve společnosti je znázorněn na obrázku 16.

Obrázek 16. Počet let ve společnosti RIA.com (kolegové)



Zdroj: vlastní zpracování

Kolegové jsou spokojeni (Ano a Spíše ano) s následujícími pracovními aspekty:

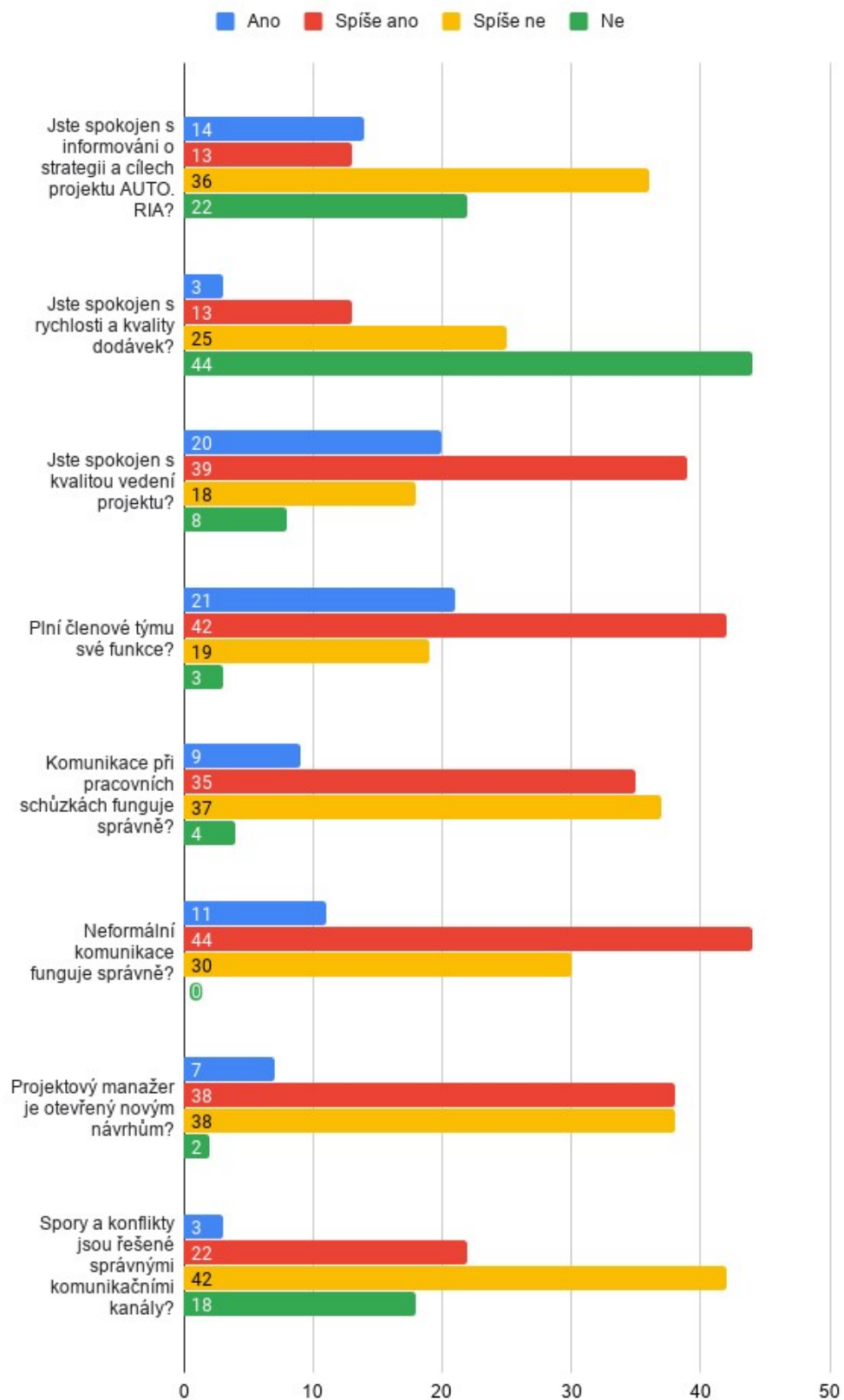
- Členové týmu AUTO.RIA plní své funkce (74,12 %)
- Spokojenost s kvalitou vedení projektu (69,14 %)
- Správné fungování neformální komunikace (64,71 %).

Názory kolegů jsou rozděleny (50 % / 50 %) ohledně *Komunikace a otevřenosti projektového manažera vůči novým návrhům*. Vzhledem k velkému počtu uživatelů webu auto.ria.com dostává projektový manažer obrovské množství komentářů a návrhů na vylepšení. Tyto návrhy nejsou vždy relevantní a v souladu s cíli projektu, a proto nejsou zapracovány. Projektový manažer by měl věnovat pozornost kvalitě odpovědí a komunikaci s uživateli a zaměstnanci, jejichž návrhy budou zamítnuty.

Kolegové nejsou spokojeni (Ne a Spíše ne) s následujícími pracovními aspekty:

- Rychlost a kvalita dodávek (81,18 %)
- Způsoby řešení sporů a konfliktů (70,59 %)
- Informování o strategii a cílech projektu AUTO.RIA? (68,24 %).

Obrázek 17. Hodnocení spokojenosti kolegů



Zdroj: vlastní zpracování

360° zpětná vazba přináší vyšší objektivitu, směřuje pozornost na vnitřní zákazníky, ovlivňuje firemní kulturu, přináší podněty pro rozvoj. Generální ředitel se rozhodl právě pro tuto formu vyhodnocení, aby se dozvěděl, jak je projektový tým vnímán z různých perspektiv. Výsledky průzkumů byly prezentovány na schůzi vedoucích všech oddělení. V průběhu schůze byla diskuse o důvodech, proč průzkumy dopadly tímto způsobem. V důsledku toho byly formulovány základní problémy projektového týmu AUTO.RIA.

1. Problémy při zadávání úkolů. Manažeři a zaměstnanci ignorují systém vytváření požadavků Kayako a zadávají úkoly přímo vývojářům v libovolné formě (ústně, telefonicky, v chatu, poštou). V důsledku požadavky nejsou pevně stanoveny, vývojáři jsou nuceni pod tlakem dávat příliš optimistické odhady časů, úpravy jsou prováděny okamžitě a bez testování.

2. Neprůhledná práce týmu AUTO.RIA. Lidské zdroje jsou plánovány na měsíc dopředu, ale ve většině případů však není možné na základě původního plánu určit, kdo na jakém úkolu pracuje. Zákazník, ředitel a další zúčastněné strany nevědí, v jaké fázi je vývoj úkolu. Složitě řídicí procesy těžko kontrolovat, a daleko ne všichni pracovníci chápou své povinnosti a někteří vůbec nevědí, jaké mají pravomoci a jaké činnosti jsou jejich odpovědnosti.

3. Nedostatečná produktivita týmu AUTO.RIA. Neustálé boje za lidské zdroje mezi vedoucími oddělení a projektovými týmy vedou k poklesu produktivity a efektivit v týmech. V týmech neexistuje téměř žádná zastupitelnost, vývojáři vzájemně do své práce nevidí a nerozumí jí i přes to, že pracují často na stejném požadavku. Spuštění mnoha procesů najednou a neustále přepínání z úkolu na úkol také má za následek zcela neflexibilní projektový tým. Existuje konflikt zájmů kvůli různým kritériím pro vyhodnocení efektivit (vývojáři se snaží co nejrychleji psát kód, testéři dlouho kontrolují všechny možné chyby, Oddělení péče o zákazníky neschvaluje spuštění atd.).

4. "Technický dluh" (úkoly, spojené s podporou stávajícího softwaru). Požadavky na podporu stávajících funkcí webu často mají poměrně velké změny (například, žádost o změnu metody výměny dat). Kritické problémy na webu vyžadují rychlé řešení a ovlivňují harmonogram probíhajících projektů.

5. Problémy s aktualizací webu auto.ria.com. Ředitel stanovuje termín ukončení projektu, což ve finále vyústí v problémy, které prodlouží včasné spuštění. Nedostatek realistického projektového plánu, protože plán nepočítá s požadavky s vysokou prioritou, které přichází obvykle před spuštěním finálního produktu. Projekty jsou ovlivňovány

jinými projekty, které se vyvíjí ve stejném čase. Nulová zpětná vazba od zákazníka a nedostatečné testování během vývoje vede ke spuštění nových funkcí se spoustou chyb.

Zmíněné problémy vedou k neočekávanému růstu rozsahu práce a přetěžování zaměstnanců, ke snížení kvality a zvýšení chybovosti implementovaných komponent. Jasná definice a pochopení problémů se stala základem pro další strategická rozhodnutí. Bylo zřejmé, že RIA.com musí projít zásadní změnou a začít se orientovat na zákazníky, na jejich potřeby, očekávání a aktuální znalosti i na to, zda se s konečným produktem dobře pracuje každému jednotlivému člověku.

5 Vlastní výzkum a dosažené výsledky

Praktická část této diplomové práce je věnována optimalizaci vývojového procesu v týmu AUTO.RIA. Vývojový tým potýkal s celou řadou problémů v oblasti zadávání úkolů, plánování a řízení, které mohou být opraveny správným nastavením metodiky vývoje. Proto hlavním úkolem je rozšířit povědomí o agilních procesech a zavést metodiku Scrum pro vývoj softwaru.

5.1 Zavádění Scrumu do vývojového týmu AUTO.RIA

Scrum pomáhá řešit úkoly, které se mění během vývoje, aby účinně a kreativně dodávat produkty zákazníkům s nejvyšší možnou hodnotou. Zavádění Scrumu se zaměří na zvýšení produktivity vývojového týmu, menší chybovost, umožnění současné práce několika vývojářů, zrychlení testování, lepší dorozumívání mezi zákazníkem a vývojářem, větší transparentnost práce týmu pro vedení projektu.

Prvním krokem bylo seznámení s existujícím stavem vývoje, zjišťování vztahu týmu AUTO.RIA k agilním metodám a k potřebě změn. Tým uznal, že existuje prostor pro zlepšení aktuálního stavu a byl ochoten zkusit některé součásti Scrumu a Kanbanu. Byly provedeny hodinové pohovory s projektovou manažerkou, aby doplnila svůj náhled zevnitř týmu a ze řídicí pozice. Dále byly sbírány informace o všech technických, organizačních a finančních okolnostech změny. Získané údaje byly použity k vytvoření návrhu změny, který byl prokonzultován s vedením týmu a prezentován internímu týmu a generálnímu řediteli. Při prezentaci byl prostor pro vytvoření nápadu na lepší postup zavedení agilních procesu a vytvoření plánu akcí.

Sprint č.1 – Plánování.

Bylo rozhodnuto provést první sprint s délkou trvání 1 měsíc, zaměřený na organizační a technické úpravy. Cílem je přesvědčit členy vývojového týmu o výhodách agilního přístupu a naučit se základy Scrumu bez většího riziku. Tým zůstává ve stejném složení a pořád pracuje ve vodopádovém modelu. První sprint přinese jasný obraz o tom, co je potřeba vylepšit, než se tým na Scrum opravdu přejde.

Tým se sešel na plánovacím meetingu, kde byl vytvořen podrobnější plán uskutečnění změn a prioritizace úkolů pro sprint. Byly naplánovány následující úkoly:

1. Seznámení se Scrumem

1.1. Organizace školení „*Agile Essentials with Scrum and Kanban*“;

- 1.2. Organizace školení „*Agile Project Management*“;
- 1.3. Organizace školení „*Agile Team Facilitation*“.
- 2. Transformace vývojového týmu
 - 2.1. Definování nových rolí;
 - 2.2. Stanovení účastníků do nového Scrum týmu;
 - 2.3. Organizace školení pro Scrum Mastera a Product Ownera.
- 3. Příprava pracovního prostoru pro tým
 - 3.1. Organizace pracovních míst pro všechny členy týmu v jedné kanceláři;
 - 3.2. Zakoupení a nastavení Scrum boardu;
 - 3.3. Rozvrh pro budoucí schůzky.
- 4. Příprava na plánování sprintu
 - 4.1. Formát zápisu User stories a tvorba Product Backlog;
 - 4.2. Prioritizace User stories a umístění na Scrum board;
 - 4.3. Odhad náročnosti User stories pomocí Story points.

Každý úkol má komentář s uvedením popisu plnění a nečekaných ztěžujících okolností. Největší ztěžující okolnosti však bylo to, že změna probíhala během normálního fungování webu, to znamená, že tým musel řešit běžné pracovní úkoly. Následně úkoly byly rozděleny mezi všemi účastníky týmu. Tým získal všechno potřebné pro začátek realizace změny.

Sprint č.1 – Realizace.

1.1. Organizace školení „*Agile Essentials with Scrum and Kanban*“.

Před změnou metodiky projektového řízení bylo uspořádáno společné školení se společností Scrum Ukraine. Jeho hlavní cíle: seznámit se s metodologií, porozumět terminologii, vypracovat všechny techniky ve formě hry, simulovat hlavní činnosti, rozdělit role a registrovat odpovědnosti. Jedná se o základní školení, které je určeno pro začátečníky v agilním vývoji a zaměřuje se na základní pojmy Agile. V programu: Agilní příběh; Agilní myšlení; základy agilního řízení procesů ve vývoji produktu; Scrum a Kanban jako nejoblíbenější týmové implementace Agile přístupu; role, principy, nástroje a artefakty v Scrumu a Kanbanu; Scrum a Kanban v praxi (hry a simulace); přizpůsobení a provádění agilního procesu; Agilní plánování; procesy neustálého zlepšování.

Díky účasti na školení projektový tým AUTO.RIA se seznámil s agilním přístupem jako způsobem myšlení prostřednictvím příkladů ze skutečného života, naučil se používat

nejoblíbenější metody Kanban a Scrum, získal znalosti a dovednosti, které umožní vědomě řídit vývojový proces a přizpůsobit jej konkrétnímu projektu.

1.2. Organizace školení „Agile Project Management“.

Školení je určeno pro pokročilé účastníky se zkušenostmi z týmové práce a relevantními teoretickými a praktickými znalostmi. V programu: organizace pohodlného pracovního postupu, který maximalizuje potenciál všech účastníků vývoje softwaru; řízení vývoje softwaru z hlediska agilního přístupu, plánování iterace, požadavků, rizik; organizace interakce mezi dodavatelem, zákazníkem a konečným uživatelem, z čehož mají prospěch všechny strany. Školení se zúčastnili projektový manažer a jeho asistenti, team leader. Účastníci školení se naučili používat Agile Management pro konzistentní a strukturovaný přístup k vývoji softwaru.

1.3. Organizace školení „Agile Team Facilitation“.

Školení se zaměřuje na organizaci pracovního procesu, který umožňuje maximalizovat potenciál všech účastníků vývoje. V programu: vytvoření atmosféry spolupráce v týmu, asistence při samoorganizaci; základní postupy a techniky facilitace; nástroje pro pracovní setkání k vyřešení problémů v těsném časovém rámci; odhalování společných překážek v týmové práci. Školení se zúčastnili projektový manažer a jeho asistenti, Tim-leader, potenciální Scrum master. V důsledku školení se účastníci dozvěděli, jak identifikovat překážky v týmové práci a vyhýbat se jim; jak provádět zajímavé a produktivní týmové akce; jak zlepšit týmový výkon zvýšením příspěvku každého účastníka.

2.1. Definování nových rolí.

Týmový model Scrum je navržen tak, aby optimalizoval flexibilitu, kreativitu a produktivitu. Scrum tým je sebeorganizovaný (self-organizing) a multifunkční (cross-functional). Samoorganizující se tým si určují, jak nejlépe vykonávat svou práci, a multifunkční tým má všechny kompetence potřebné pro dosažení práce bez závislosti na ostatních, kteří nejsou součástí týmu.

Scrum tým se skládá z role Product Owner (vlastník produktu), vývojového týmu (development team) a z role Scrum Master (obvykle se nepřekládá).

Product Owner je zodpovědný za dlouhodobou vizi produktu, za kvalitu produktu jako celku, Product Backlog, prioritizaci úkolů a kontrolu toho, jak dodány kus práce odpovídá plánované představě. Také je zodpovědný za reportování stavu projektu k řediteli.

Projektový manažer sbírá požadavky na produkt. Je zodpovědný za monitorování průběhu realizace projektu a reportuje aktuální stav projektu Product Ownerovi.

Scrum master má na starost veškeré Scrum procesy a schůzky. Je zodpovědný za dodržení všech rituálu, sledování produktivity týmu a odstranění překážek v jeho práci.

Největší změnou je transformace vývojového týmu z komponentního na funkční. To znamená přesun HTML vývojáře, testera a content specialisty přímo do týmu AUTO.RIA. Funkční tým má obrovskou výhodu, protože je schopen doručit plně otestovaný a rozšířený produkt, který se dá vydat.

2.2. Stanovení účastníků do nového Scrum týmu.

Scrum tým AUTO.RIA je složen z product ownera, projektového manažera, scrum mastera, týmového vedoucího, 5 vývojářů, HTML vývojáře, testera a content specialista.

Product Owner je rolí, kterou na sebe vezme produktová manažerka Marina K. Roli projektového manažera převezme asistent produktové manažerky Ondřej K. Roli Scrum mastera na sebe vezme manažerka Irina S. Nynější vývojáři budou plnit své dosavadní role. HTML vývojář, tester a content specialista se stali členy vývojového týmu, ale jsou stále podřízeni vedoucím svých oddělení.

Jedním z nejdůležitějších kroků je zavedení testování jako součást vývoje. Původní stav byl takový, že naprosto vše od drobných defektů až po testování produktu jako celku se provádělo na oddělení testování. Tím pádem dochází k nahromadění opravených defektů a k testování před samotným vydáním nové verze produktu.

Přidělení testera přímo do vývojového týmu přináší mnoho výhod: rychlá odezva na chyby v kódu; menší režie na plánování testování; lepší komunikace mezi testerem a developerem. Po přiřazení testeru do Scrum týmu je nutné rozdělit testování, které se bude provádět v rámci týmu a v rámci externího oddělení. Vzniká tak rozdělení na interní a externí QA.

Interní QA (testeři ve vývojových týmech) budou zodpovědní za: testování všech nových požadavků; testování všech částí vytvořených v rámci týmu; testování všech oprav defektů; finální testování požadavků jako celku.

Externí QA budou zodpovědní za: finální ověření před samotným vydáním produktu; ověření defektů v týmech, kde není tester; výpomoc na verifikaci defektů týmům, které nestíhají testovat; výpomoc s akceptačním testováním požadavků.

Testování kódu přímo v týmu eliminuje čekání na testování kódu a snižuje čas reakce opravy chyby na minimum, což zlepšuje i kvalitu produktu.

2.3. Organizace školení pro Scrum Mastera a Product Ownera.

Projektový manažer se svými asistenty se zúčastní školení "*Scrum Master + Product Owner*". Tento dvoudenní kurz je školením pokročilé úrovně a je tvořen vyváženou směsí výkladu a praktických cvičení. Soustřeďuje se na aplikování Scrumu i dalších agilních přístupů a technik v práci v rolích Scrum Mastera a Product Ownera. Kurz naučí, jak rozdělit týmové kompetence podle rolí, jak využít potenciál "vyššího výkonu" se stávajícími zdroji a jak efektivně řešit problémy se všemi zúčastněnými stranami.

Obsah kurzu z velké části odpovídá předmětu certifikačních zkoušek pro Scrum Mastera a Product Ownera. Pro přípravu na zkoušku je třeba po tomto kurzu prostudovat doporučenou literaturu a získat praktickou zkušenost s aplikováním Scrumu v příslušné roli.

3.1. Organizace pracovních míst pro všechny členy týmu v jedné kanceláři.

Koncept Agilní kanceláře je založen na flexibilním přístupu k organizaci práce zaměstnanců, který zahrnuje vytvoření multifunkčních oblastí pro týmovou práci a tvůrčí činnosti, jednání a neformální komunikaci. Tento přístup vyžaduje určitou mobilitu a sjednocení z kancelářské infrastruktury. V rámci přípravy Agilní kanceláře byla vytvořena pracovní skupina zaměstnanců různých úrovní.

Agilní kancelář je otevřený prostor na samostatném patře, kde nejsou žádné kanceláře vedoucích. Celý prostor je vytvořen tak, aby zaměstnanci mohli v případě potřeby shromáždit se v malých skupinách nebo zůstat úplně sami. V tomto prostoru jsou:

- stálá pracovní místa, přidělená konkrétním zaměstnancům;
- samostatná zóna pro ranní stand-up schůzky. Hlavní složkou této zóny je Scrum-board, na kterou je možné psát a nalepovat nálepky. V případě potřeby se Scrum-board může pohybovat po otevřeném prostoru;
- dva velké sály pro týmovou práci uprostřed kanceláře vybavené projektor, velkými obrazovkami, konferenčním příslušenstvím a kulatým stolem;
- samostatná kávová zóna pro neformální komunikaci, která je oddělena od hlavního pracovního prostoru zelenou zdí živých rostlin;
- místa pro práci v soukromém prostředí.

Nyní zde bude umístěn jeden tým AUTO.RIA, ale v budoucnu se sem přestěhují i další vývojové týmy.

Flexibilita Agilní kanceláře umožnila zajistit pohodlné pracovní podmínky, stala se nástrojem pro optimalizaci nákladů a způsobem, jak zvýšit efektivitu a loajalitu zaměstnanců.

3.2. Zakoupení a nastavení Scrum boardu.

Na začátku se využívá pouze klasický Scrum board. Tým AUTO.RIA používá Scrum board, složený ze 7 sloupců:

Inbox – slouží jako seznam podnětů. Jsou to nápady, které se ve výsledku nemusí stát reálnými úkoly, ale je dobré je zmínit. Příkladem mohou být nápady pro použití technologií v rámci projektu nebo potenciální problémy, které by mohly nastat během řešení určitých úkolů. Výhodou pak je, že veškeré podněty jsou na jednom místě a odpadá dohledávání v e-mailech a jiných komunikačních nástrojích;

Backlog – obsahuje konkrétní úkoly nebo větší celky, které bude třeba dále rozpadnout. Jedná se vlastně o to-do list, který je seřazen dle priority od nejdůležitějšího až po méně důležitý úkol;

Sprint – obsahuje úkoly pro aktuální týden, vychází ze sprint planningu. Každá kartička má pouze jednoho řešitele a celý sloupec je seřazen dle priority;

In Progress – sloupec, kam patří úkoly, na kterých kolegové aktuálně pracují. Každý zde může mít jenom jednu kartičku, protože dokáže v jednu chvíli pracovat na jednom úkolu. Úkol má z In Progress tři možné cesty. Může se stát, že je potřeba jej na určitou dobu odložit kvůli jinému úkolu a v tu chvíli putuje zpět do Sprintu. Jindy se může stát, že na úkolu nelze pracovat, protože ho něco blokuje – tehdy patří do sloupečku Blocked. V ideálním případě je úkol ze strany řešitele dokončen a pak cestuje kartička do sloupce Akceptace.

Blocked – blokové nebo jinak čekající úkoly, přičemž v komentáři je zapotřebí vysvětlit, čím je úkol blokován;

Akceptace – sloupec s úkoly čekajícími na kontrolu testerem a schválení.

Done – vyřešené a schválené úkoly. Kartička se v tomto sloupci může ocitnout pouze tehdy, je-li úkol hotový a otestovaný/zkontrolovaný.

První realizovaný návrh tabule úplně odpovídal popisu. Následně product owner doplnila úkoly do tabule, a tak ji vývojáři mohli začít používat.

3.3. Rozvrh pro budoucí schůzky.

Ruší se projektové schůzky a řídicí výbory. Celý realizační proces se nově rozděluje na sprinty, ve kterých se opakují pravidelné schůzky.

Plánování bude probíhat na začátku každého sprintu. Během schůzky si členové týmu zvolí (podle stanovených priorit), jaké činnosti budou v následující iteraci vykonávat. Teto rozhodnutí je neměnné po dobu celého sprintu.

Stand-up se bude konat každé ráno a slouží k rychlé kontrole splněných úkolů. Během schůzky se bude aktualizovávat Scrum board.

Review meeting bude probíhat na konci sprintu, kde zadavatel úkolu zkontroluje výsledek.

Retrospektiva je jako přímá možnost přednést návrhy na zlepšení a bude se konat na konci každého sprintu.

Sprint by měl mít prvních pár týdnů po zavedení délku ideálně jeden týden, aby si členové týmu mohli zvyknout na nový pracovní plán. Je to ze začátku dobrá zpětná vazba a skvělá příležitost zjišťovat tak chyby na týdenní bázi.

4.1. Formát zápisu User stories a tvorba Product Backlog.

Pro lepší spolupráci týmu a zákazníka a snazší prioritizace Product Backlogu jsou použity User Stories. User Story je nejčastější forma zápisu položek Backlogu. Je to popis funkcionality, scénář, který je představen ve formě krátkých jednoznačných vět, kde uživatel uvádí, co chce získat a za pomoci jaké činnosti.

User Story musí být nezávislá, popsatelná, přinášet hodnotu, ohodnotitelná, malá a testovatelná – tedy v originále: Independent, Negotiable, Valuable, Estimatable, Small, Testable requirement (zkráceně “INVEST”).

Je-li User Story příliš velká je většinou těžké říct co je jejím obsahem a co už ne, a je tak pro tým neuchopitelná. Proto musí User Story dodržovat akceptovaný formát „*Jako Uživatel, chci Funkcionalitu abych dostal Hodnotu*“ User Story vývojářům říká nejen, co mají dělat, ale i pro koho a hlavně proč.

Navrhnout User Story může kdokoliv. Nicméně Scrum zavádí roli Product Ownera, který je za celý Backlog a User Story v něm zodpovědný. A proto je Product Owner ten, kdo User Story nakonec do backlogu akceptuje, a přiřadí jí v závislosti na business value prioritu.

User Story přicházejí k Product Ownerovi přes Informační systém Kayako. Vyplňovací formulář v Kayako obsahuje pole Název, Popis, Tagy, Způsob testování, Časový odhad. Název se zobrazuje ve všech výpisech a musí po dohodě obsahovat „*KDO chce CO a PROČ*“. Popis je volitelný, může obsahovat obrázek nebo URL. Tagy obvykle

vyplňuje Product Owner. Způsob testování obsahuje kroky k reprodukcí problému. Časový odhad si doplňuje Scrum Master, k tomu používá tzv. Story Points.

4.2. Prioritizace User Stories a umístění na Scrum board.

Product Owner akceptuje User Story do Product Backlogu a přiřadí jí prioritu v závislosti na business value. Poté celý seznam upřednostňuje podle těchto priorit.

Na Scrum Boardu jsou User Stories umístěny na kartičkách různých barev, v závislosti na navrhované klasifikaci. User Story obsahuje: Název, Stručný popis úkolu, Jméno autora, Datum zadání úkolu, Časový odhad, Deadline, Případné komplikace.

Blokující chyby jsou umístěny na červených kartičkách, Kritické chyby – na oranžových, Kosmetické chyby – na žlutých, Nové požadavky – na zelených, Zbývající úkoly jsou na bílých kartičkách. Pravidlem, které tým stanovil, bylo, že se kapacita sprintu musí rozdělit mezi nové požadavky, opravu existujících chyb, dokumentaci a technický dluh. Poměr je dynamický a dá se podle potřeb měnit v každém sprintu (20 % času na nové požadavky, 45 % času na opravu chyb, 35 % času na dokumentaci a technický dluh).

4.3. Odhad náročnosti User stories pomocí Story points.

Pro odhad náročnosti úkolů tým AUTO.RIA používá metodu *Planning Poker*. Planning pokeru se účastní všichni členové týmu (vývojáři, HTML vývojář, tester a content specialista). Product Owner se účastní Planning pokeru, ale neměl by se zapojovat do samotného odhadování. Scrum master se účastní jako moderátor hry, také nehodnotí úkoly. Cílem Planning pokeru je vytvořit co možná nejpřesnější odhad co možná nejlevněji, to znamená za krátký čas a bez spotřebování mnoha zdrojů.

Plánování probíhá takto:

- na začátku Product Owner čte User stories z Product Backlogu (podle priorit) a odpovídá na jakékoliv otázky;
- poté si každý odhadce dostává balíček karet. Na každé kartě je napsána hodnota odhadu ve Fibonacciho sekvenci (0, 1, 2, 3, 5, 8, 13 atd.), karta Otazník značí totální nejistotu, karta Coffee break slouží pro případy, kdy je potřeba si dát přestávku. Scrum nedoporučuje používat pro odhad klasickou jednotku času, velice rozšířeným jsou takzvané Story points. Story points je naprosto abstraktní jednotka, kde jeden Story point představuje čas na dokončení nejmenšího úkolu. Odhadce soukromě vybírá kartu, která reprezentuje jeho odhad;
- ve stejný okamžik všichni otočí svoji kartu. Pokud se odhady liší, pak vysoký nebo naopak nízký odhad musí expert před ostatními obhájit;

- po skončení diskuse každý odhadce provede nový odhad vybráním karty. Cílem je, aby se odhadci shodli na jednom odhadu, který lze použít.

Plánovací poker kombinuje znalecké posudky, analogii a zábavný přístup, který vede k rychlým, ale velice spolehlivým odhadům.

Sprint č.1 – Demo.

Výsledky prvního sprintu jsou prezentovány vedení společnosti RIA.com. Prezentace se zúčastnili ostatní projektové týmy, vedoucí všech oddělení a další zaměstnanci, kteří požádali o účast. Hlavní část prezentace byla představena Product Ownerem a Scrum Masterem. Programátoři týmu AUTO.RIA vyjádřili své dojmy a budoucí očekávání. Na konci setkání ředitel společnosti poděkoval týmu za dobrou práci a vyjádřil souhlas s pokračováním v implementaci Scrumu.

Sprint č.1 – Retrospektiva.

Většina týmu se seznámila se Scrumem a agilním způsobem vývoje. Dobrovolně se několik lidí podstoupilo školení na Scrum mastera. Vývojáři získají větší volnost, ale také zodpovědnost ve vývoji, protože si sami určují, jak bude výsledný produkt vypadat a jaké technologie budou využity. Získají možnost být více kreativní, protože jednotlivé dodávky softwaru sami prezentují a obhajují zákazníkovi.

Na Demo zaměstnanci vyjádřili přání blíže se seznámit se Scrumem. Product Owner provede školení pro zúčastněné strany, aby se seznámili se specifiky Scrumu a s tím, jak bude práce na projektu probíhat. Je nutné vysvětlit, kdy je nezbytné jejich zapojení a zpětná vazba, jak vyplnit nový formulář zadávání úkolů, jak bude probíhat akceptace dokončených úkolů.

Sprint č.2 – Plánování.

Na začátku plánování tým AUTO.RIA učinil následující rozhodnutí:

- byl stanoven cíl a podle toho název plánovaného sprintu;
- délka trvání sprintu byla nastavena na 2 týdny;
- Demonstrace, Retrospektiva a Plánování dalšího sprintu se budou konat v úterý (demo v 10.00, retrospektiva v 11.00, plánování po obědě);
- Daily Scrum bude se konat každý den v 9.15.

V předchozím sprintu byl sestaven prioritizovaný Product Backlog a schválená metoda Planning Poker pro odhad náročnosti úkolů. Během plánování sprintu č.2 tým si vybíral z Product Backlogu úkoly s nejvyšší prioritou a odpovídající cíle sprintu, odhadoval jejich náročnost a udělal z toho Sprint Backlog. Tým vybíral úkoly podle

stanoveného pravidla: 20 % nových funkcí, 45 % chyb, 20 % technický dluh, 15 % dokumentování. Vybrané úkoly pak byly zapsány na příslušných kartičkách a umístěny do sloupce Sprint na Scrum Board. Plánování sprintu č.2 trvalo 6 hodin, nejvíce času bylo věnováno Planning Pokeru.

Sprint č.2 – Realizace.

Každý den v 9.15 probíhala 15minutová Daily Scrum schůzka vývojového týmu. Každý člen týmu odpovídal na tři otázky: "Co jsem včera udělal?" "Co budu dělat dnes?" "Jaké překážky vidím?" Hlavním cílem schůzky bylo zjistit, zda někdo z týmu měl problémy s plněním úkolů. V důsledku byl aktualizován Scrum Board, každý člen týmu věděl, co dnes budou dělat ostatní a v jaké fázi bylo dosažení cíle sprintu.

V průběhu sprintu vývojáři těsně komunikovali a spolupracovali, docházelo k technickým konzultacím ze strany Team Leadera. Vzniklé otázky probírali s Product Ownerem nebo zákazníkem, a tak od nich dostávali zpětnou vazbu v procesu vývoje.

Na konci sprintu bylo vyřešeno 93 % úkolů ze Sprint backlogu. Product Owner definoval, že je s výsledkem spokojen. Před spuštěním však bylo nutné vyvíjené webové služby napojit k jiným již existujícím systémům na webu. Aby nedošlo k chybám, tým se rozhodl rychle napsat automatické testy.

Sprint č.2 – Demo.

Vzhledem k tomu, že zákazníci jsou zaměstnanci společnosti, Sprint Review a Sprint Demo probíhaly na stejné schůzce. Každý vývojář prezentoval své úkoly, odpovídal na otázky, dostával zpětnou vazbu od zákazníků a náměty do budoucna, které byly následně zapracované do Product Backlogu.

Sprint č.2 – Retrospektiva.

Na Retrospektivě byl zhodnocen celkový průběh vývoje, co se povedlo a nepovedlo během sprintu. Zúčastnili se projektový tým AUTO.RIA, Scrum Master a Product Owner. Cílem bylo neustálé zlepšování týmu. Po skončení prvního plnohodnotného sprintu tým AUTO.RIA stanovil, že nedošlo k žádným velkým překvapením a jeho cíl byl naplněn.

Scrum Masterem byl zpracován Sprint Report, který uvádí seznam dokončených a nedokončených úkolů. V první řadě byly analyzovány nedokončené úkoly, příčiny a možné způsoby řešení. Byly identifikovány následující problémy:

1. Spolupráce s marketingovým oddělením. I když je Content specialista v týmu a plní své úkoly včas, musí být všechny texty schváleny Marketingovým ředitelem. Vzhledem k tomu, že marketingové oddělení pracovalo ve svém režimu a do jeho plánů

nebyly přidány termíny sprintů, potřebné texty nebyly schváleny včas a podle toho nemohly být úkoly dokončeny.

Řešení problému: všechny marketingové aktivity plánovat minimálně o měsíc dříve než vývoj, a ve sprint plánovat úkoly se všemi schválenými komponenty (design, text, seo atd.). Dojde tak ke zpřesnění odhadu, ne odložení termínu.

2.Plánování týmu AUTO.RIA a ostatních oddělení společnosti. Tým AUTO.RIA bude i nadále plánovat svou práci jednou za 2 týdny. Plánování celé společnosti stále probíhá na úrovni dlouhodobého plánování na několik měsíců dopředu. Proto je potřeba zavést synchronizační prvek, který by dokázal agilně plánovat tým, ale zároveň dodal plán pro zbytek neagilní společnosti.

Řešení problému: naplánovat setkání manažerů, na kterém budou stanoveny priority všech projektů na 1 měsíc dopředu. Výstupem bude seznam požadavků, pro které především je potřeba podle určených priorit dodat externí vstupy. Na jeho základě bude plánováno marketingové a designové oddělení. Plánování vývojových týmů bude stále probíhat jednou měsíčně. Výstupem pro vodopádové týmy bude jasně definovaný plán s konkrétními termíny dokončení. Výstupem pro agilní tým AUTO.RIA bude Product Backlog s pořadím, v jakém se budou jednotlivé požadavky řešit, ale bez koncových termínů dokončení.

3.Odhad náročnosti úkolů ve Story Pointech. Tým AUTO.RIA se rozhodl zkusit estimaci požadavků ve Story Pointech. V praxi se ukázalo nepochopení podstaty pojmu "Story Point" a následně špatný odhad. Vývojáři zapomínali, že Story pointy neodpovídají klasické jednotce času, a proto některé úkoly trvaly déle, než bylo zaplánováno. Kromě toho, Scrum Master nemohl definovat, jak daleko vývojář s tímto úkolem pokročil.

Řešení problému: tým se rozhodl vrátit k estimaci v hodinách. Zároveň zatím ponechal metodiku hodnocení Planning Poker.

4. Scrum Board. Tým AUTO.RIA používal tabuli v kanceláři. Aby se zaměstnanec mohl seznámit s průběhem plnění úkolů, musí být přítomen v kanceláři u tabule. Vzhledem k tomu, že tým AUTO.RIA se přestěhoval do samostatného patra, zákazníci nebyli spokojeni s nutností neustále chodit k tabuli nebo zjišťovat stav svého úkolu telefonicky.

Řešení problému: vybrat vhodný software pro podporu Scrumu a synchronizovat Scrum Board. Product Backlog by také měl být snadno dostupný celému Scrum týmu.

Byla oceněna výborná práce Scrum Mastera, který vysvětloval podle potřeby aspekty Scrumu, chránil tým před vnějšími vlivy a motivoval k dosažení cíle sprintu.

5.2 Zhodnocení přechodu na Scrum

V této kapitole je popsáno souhrnné zhodnocení přechodu týmu AUTO.RIA na Scrum, dosažení stanovených cílů a doporučení do budoucna.

Problém č. 1 "Zadávání úkolů" je vyřešen automaticky, protože jedním z hlavních artefaktů Scrumu je Backlog. Jedná se o jediný seznam všech úkolů, požadavků na jakékoli změny, které je třeba v produktu provést. Za obsah, dostupnost a prioritizaci backlogu je plně zodpovědný Product Owner. Úkol tedy má jedinou cestu od zákazníka k vývojáři.

Doporučením je automatizace správy Backlogu pomocí různých nástrojů pro podporu agilního řízení týmu.

Řešením problému č. 2 „Neprůhledná práce týmu AUTO.RIA.“ je zvýšení transparentnosti jejich práci. Tento problém je považován za částečně vyřešený díky používání Scrum boardu. Scrum board umožňuje každému zaměstnanci vidět, jaká práce aktuálně probíhá, co bylo splněno a co se teprve chystá. Zároveň, Scrum board zobrazuje chyby, které se během sprintu vyskytly. Tato změna zjednodušuje evidenci splněných úkolů, protože na konci každého sprintu Product Owner vidí, jaký kus práce byl splněn, co se nepovedlo nebo povedlo a proč. Při takovém sledování sprintu nemusí vedení čekat dlouhou dobu v nevědomosti.

Tým AUTO.RIA pro svůj nový styl práce zvolil metodu Scrum, která usiluje o nepřetržité zlepšení a vývoj. Proto doporučením je zachovat vývoj a vítat nápady ke zlepšení, tedy dodržovat filozofii Scrum. Potvrzování pozitivního vlivu změn a oslavování malých úspěchů týmu také má velice motivační charakter. Během lessons learnt má smysl nezapomínat bavit se nejen o tom, co se v dané iteraci nepovedlo. Důležité je zmínit povedené úkoly a změny, takovým způsobem je upevnit.

Dalším doporučením je vybrat vhodný projekt management systém, digitalizovat a synchronizovat Scrum board na tabuli a v systému, zaškolení tým a otevřít jim zabezpečený přístup. Systém musí obsahovat všechny údaje o aktuálních vývojových projektech, marketingových kampaních a finanční údaje. Scrum board v digitální podobě lze kontrolovat odkudkoliv i když není Project manager přímo v kontaktu s týmem.

Problém č. 3 "Nedostatečná produktivita týmu AUTO.RIA." nebyl dostatečně vyřešen, při implementaci jakékoliv změny je nutné počítat s poklesem produktivity a růstem nákladů. Bohužel, kvůli tomu, že produktivita projektu AUTO.RIA nebyla evidována, nelze přesně určit úroveň poklesu produktivity, a proto nelze vypočítat přesný

náklad na změnu. Zároveň, kvůli stejným důvodům nelze spočítat, jak produktivita vyroste.

Doporučením bude použití Burndown Chartu ke sledování výkonu týmu během sprintu. Pouze pokud bude výkon sledován, bude možné jednoznačně říci, zda změny přinesly požadované efekty.

Dalším doporučením je stanovení motivačních cílů rychlosti práce. Jakmile tým bude schopen dobře odhadovat časovou náročnost, lze mluvit o opatřeních ohledně zvýšení produktivity, tím je myšleno zavedení další automatizace a vyhýbání se zbytečné práci.

Problém č. 4 "Technický dluh" je považován za vyřešený díky přijatému pravidlu. Toto pravidlo stanoví, že kapacita sprintu bude rozdělena následovně:

- 20 % úkolů – vývoj nových funkcí;
- 45 % úkolů – oprava chyb na webu;
- 20 % úkolů – technický dluh;
- 15 % úkolů – dokumentování.

Díky tomuto rozdělení dostanou úkoly, spojené s podporou stávajícího softwaru, nejvyšší prioritu, a budou naplánovány a opraveny v nejbližších sprintech.

V budoucnu je třeba věnovat zvláštní pozornost odstranění technických problémů. Aktuálně tým AUTO.RIA má pouze jeden neveřejný server pro testování a vývoj, což znamená, že aplikaci není možné testovat a vyvíjet najednou. Požadovanou změnou je zakoupení dalšího serveru. Nový výkonnější server umožní pracovat na vývoji aplikace několika členům týmu současně, zkrátí čekací dobu a zrychlí tok práce.

K řešení **problému č. 5 "Aktualizace webu auto.ria.com"** byly zavedeny dvoutýdenní sprinty. Takové rozvržení práce znamená, že již od prvních týdnů tým AUTO.RIA teoreticky má produkt k uvolnění. Zároveň je zákazník vždy informován, zda se vývoj ubírá adekvátním směrem, což šetří čas a peníze.

Důležitým krokem bylo zavedení Scrum Mastera, osoby, která se stará o odstranění překážek. Takovým způsobem vývojáři nemusí ztrácet svůj čas na řešení organizačních a technických věcí.

Díky pravidelné zpětné vazbě od zákazníků tým zlepšil kvalitu kódu a přispěl k vytvoření relevantnějších funkčností produktu. A protože tým začal dodávat produkty po menších částech, měli méně práce s jejich doladěním než za vodopádové metody.

Díky aplikování častějšího a okamžitého testování po dokončení každé části softwaru tým měl méně práce s finálním testováním produktu, a zákazníci museli čekat kratší dobu na doručení funkcí či řešení.

Je zřejmé, že chybovost je velkým důvodem zpomalení pracovního procesu. Automatizace buildu a testování jsou prevencí proti chybám během manuální práce. Zavedení automatického testování a automatického buildu ušetří finanční a lidské zdroje. Manuální testování jednoho úkolu trvá přibližně hodinu, zatímco vytvoření automatického testu trvá přibližně 3-4 hodiny. Zároveň automatické testy budou napsány pouze pro základní funkcionality, které se při testování každé nové verze opakují a automaticky zabraňují vysoké chybovosti.

Dále je nutné se zamyslet nad sledováním chybovosti. Sledování chybovosti a ověření efektivnosti automatického buildu může napomoci odhalit zdroj vzniku chyb. Pokud zdroj chyb je odstraněn, zákazník bude dostávat kvalitnější produkt rychleji.

Dalším doporučením je zavedení automatického deploymentu, což zahrnuje automatické testování, automatický build a automatické umístění na server pro zákazníky. Taková změna uvolní čas vývojářů a zmenší chybovost.

Nehledě na všechny tyto změny nelze s jistotou říct, zda cíle byly dosaženy, protože plynulo málo času na to, aby se to dalo s jistotou tvrdit. Doporučená doba sledování práci týmu po změnách je minimálně půl roku.

Členové týmu AUTO.RIA jsou obeznámeni se Scrumem velmi okrajově. Proto doporučením je absolvování kurzu pro pokročilé Agilní řízení v IT, který je tvořen vyváženou směsí teoretických a praktických cvičení. Samozřejmě, takové školení znamená velkou investici pro společnost RIA.com, proto doporučení bude mít větší smysl, pokud implementované změny přinesou pozitivní výsledky.

Dotazníkové šetření. Aby byl proces zavedení agilního řízení zhodnocen nejen podle dosažení stanovených cílů, ale také na základě subjektivního vnímání, zpracovala jsem dotazníkové šetření. Hodnocení má za cíl zjistit:

- jaké změny jsou nejvíce a nejméně efektivní a jaké jsou další podněty ke změnám;
- jaké překážky kolegové vidí na cestě ke změnám;
- zda provedené změny řešily problémy projektu.

Dotazníkové šetření bylo uskutečněno rozesláním emailů s odkazem na online Google forms. Tento servis byl zvolen proto, že je bezplatný a pohodlný pro vyplnění.

Dotazník se skládal z devíti otázek. Většina z nich nenabízí varianty odpovědí a předpokládá psaní odpovědí samostatně. Na dotazník odpovědělo celkem 45 zaměstnanců – členy vývojového týmu AUTO.RIA a kolegové, kteří pracují na oddělení Marketingu a Péče o zákazníka a jsou nejčastějšími zadavateli požadavků.

Otázka č.1 Jaká z metodik je vhodnější pro řízení IT projektů, Scrum nebo Waterfall (argumentujte svou odpověď)?

Většina dotazovaných souhlasí s tvrzením, že je Scrum viditelnější, flexibilnější, méně rizikový a vhodnější pro řízení IT projektů než Waterfall. Zobecněné argumenty kolegů:

“Scrum je flexibilní přístup, který má viditelnější pokroky ve vývoji, umožňuje přidat požadavky v průběhu vývoje a podporuje průběžné testování”.

“Scrum je postaven na průběžném vývoji, který nejprve dodá funkční základní funkcionality a poté každým dalším sprintem dodává vylepšení”.

Otázka č.2 Vnímáte přechod do Scrumu pro tým AUTO.RIA jako vhodný? Ohodnoťte svoji spokojenost se zavedenými změnami (na stupnici od 1 (nejméně) do 10 (nejvíce))

Průměrná hodnota ukazatele je 8,7. Všechny respondenti vnímají přechod do Scrumu jako dobrý směr, způsob zpřehlednit práci a nastavit lepší komunikaci během vývoje. Názory některých zaměstnanců:

Vývojář – *„Je dobré, že přesně vím, co budu další dva týdny dělat“.*

Vývojář – *„Je super, že nemám paralelně rozpracovaných několik úkolů“.*

Otázka č.3 Jaké ze zavedených změn v rámci přechodu do Scrumu jsou nejúčinnější?

Nejvýznamnější změny dle výsledků jsou rekonstrukce kancelářských prostor, Scrum board, testování přímo v týmu a dvoutýdenní sprinty.

Team leader – *„Testování přímo v týmu je pro mě největším přínosem Scrumu. Nemusím dlouho čekat, dokud někdo defekt ověří. V případě, že je něco špatně, můžu se k opravě vrátit hned další den s čerstvou znalostí kódu, ušetří to spoustu hodin analýzy“.*

Tester – *„Je pro mě lepší být součástí vývojového týmu, protože znám jejich aktuální problémy, můžu testovat největší problémy prioritně, mít dotazy přímo na vývojáře a ústní komunikace je mnohem rychlejší než přes email nebo chat“.*

Otázka č.4 Jaké překážky zpomalují zavedení Scrumu?

Dva respondenti zdůrazňují, že projektu chybí přehlednost. Jeden respondent zmínil obecnou nedůvěřivost lidí ke změnám. Jeden respondent uvedl jako problém chaos v procesu vývoje, myslí si, že naplánované sprinty by měly přispět ke striktnějšímu dodržování termínů. Několik respondentů vnímá, že přechod na Scrum je velmi nákladný kvůli potřebným certifikacím a školením pro zaměstnance.

Otázka č.5 Je pro Vás pohodlnější spolupráce s týmem z agilního prostředí? Co jste ocenil(a) a co naopak chybí?

Většina je spokojenější s prací agilního týmu. Kolegové oceňují flexibilní spolupráci, rychlost a flexibilitu dodávek, možnost průběžně prioritizovat jednotlivé User Stories.

Scrum master – *„Máme více času na technický dluh a na psaní automatických testů“*.

Otázka č.6 Jak přímá komunikace s vývojáři ovlivňuje srozumitelnost úkolů?

Podle názoru většiny kolegů (71 %), zpětná vazba od zákazníků je velmi nápomocná pro vytvoření relevantních funkcí a dosažení správného řešení. Takže, pravidelná komunikace se zákazníky má pozitivní dopad na srozumitelnost úkolů a je přínosem aplikované metodiky Scrum.

Otázka č.7 Zkrátil se čas mezi zaznamenáním problému a jeho vyřešením?

Další otázka se týkala zkrácení času mezi zaznamenáním problému a jeho vyřešením. Více než polovina kolegů (63 %) zpozorovala klesající tendenci délky času na vyřešení nahlášeného defektu. Jedním z důvodů tohoto zkrácení je zadávání defektů do Backlogu a stanovení priorit, a tím pádem lepší transparentnost defektů pro všechny účastníky procesu.

Ředitel oddělení péče o zákazníky – *„Mít na konci každé iterace produkt, který je připraven a funkční, je perfektní. Díky tomu jsme schopni získat validní zpětnou vazbu od uživatelů“*.

Otázka č.8 Změnila se kvalita vyřízených úkolů po aplikování Scrumu?

Mezi kolegy převládá názor, že kvalita se zlepšila (53,4 %), pouze o několik procent méně kolegů (41,5 %) si myslí, že kvalita zůstala na stejné úrovni, a pouze 5,1 % zaznamenalo zhoršení kvality vyřízených úkolů. Důvodem takových výsledků může být to, že aplikování průběžného testování zajišťuje, aby vývojáři byli včas informováni o případných defektech a aby kvalita kódu byla na dobré úrovni.

Otázka č.9 Jaké další změny byste navrhl(a) pro tým AUTO.RIA?

Kolegové mají plnou řadu doporučení, která by jim spolupráci s týmem zjednodušila a zpříjemnila. Nejčastěji zmíněné doporučení jsou: lepší testovací prostředí, prioritizace napříč agilních dodávek, lepší kontrola nákladů.

Posledním doporučením je v případě spokojenosti s výsledky použít Scrum přístup ve všech vývojových týmech. Pokud ředitel RIA.com uzná Scrum jako oficiální přístup pro společnost, zjednoduší to práci pro týmy a nebude třeba hledat kompromis.

Na závěr je provedena analýza dodržování agilních principů při řízení projektu.

Tabulka 5. Dodržování agilních principů při řízení projektu AUTO.RIA.

№	Princip	Plnění
1	Včasné a průběžné dodávání hodnotného software	Částečně Ano
2	Možné změny v požadavcích v průběhu vývoje	Ano (po dohodě s PO)
3	Dodávání fungujícího software v pravidelných intervalech	Ano (jednou za 2 týdny)
4	Denně spolupráce vývoje a Product Ownera	Ano
5	Zaměstnanci jsou motivováni a snaží se dělat svou práci dobře	Ano
6	Osobní konverzace jako nejefektivnější způsob sdělování informací	Ano
7	Hlavním měřítkem pokroku je fungující software	Částečně Ano
8	Procesy podporují udržitelný rozvoj	Částečně Ano
9	Neustálá pozornost technické výjimečnosti a dobrému designu	Ano
10	Jednoduchost	Částečně Ano
11	Samoorganizující se tým	Částečně Ano
12	Zamyšlení o zvýšení efektivity týmu	Ano

Zdroj: vlastní zpracování

Z tabulky je zřejmé, že řízení projektů v týmu AUTO.RIA splňuje všechny agilní principy, sedm úplně a pět částečně.

Princip č.1. Včasné a průběžné dodávání hodnotného software je splněno částečně, protože tým AUTO.RIA nemůže vždy poskytnout funkční část softwaru na konci sprintu. Aktuálně tým má pouze jeden neveřejný server pro testování a vývoj, proto s ohledem na časový limit (2 týdny) nejsou všechny úkoly testovány a dokončeny včas.

Princip č.7. Tým ještě nedefinoval kritéria kvality pro fungující software, a ještě nejsou automatické testy pro kontrolu chyb při napojení k jiným již existujícím systémům.

Princip č.10. Tým zatím nepoužívá žádný software na podporu Scrumu, proto nemůže zajistit, aby všechny informace, týkající se projektu, byly dostupné pro celý tým a snadno se změnilo během realizace projektu. Doporučením je výběr vhodného softwaru, který bude intuitivní a poskytne všechny nástroje pro agilní řízení projektů.

Princip č.11. Tým AUTO.RIA jako experiment mění své principy řízení projektů. Ředitel v této fázi neposkytl úplnou svobodu, takže tým nelze hodnotit jako zcela samoorganizující. V případě pozitivních výsledků se změny dotknou i ostatních týmů.

V důsledku lze konstatovat, že projekt AUTO.RIA má zavedenou většinu principů Agile a Scrum.

5.3 Podpora agilního řízení týmu

Během procesu vývoje již tým AUTO.RIA využíval různé nástroje, ale v průběhu času se pořád vyskytovaly problémy a neshody s jejich používáním. Proto proces výběru nového software pro podporu Scrumu začíná určováním, kdo se systémem bude pracovat. Prvním krokem je sesbírání všech požadavků pro všechny druhy uživatelů.

Tabulka 6. Požadavky na software od zaměstnanců

Základní požadavky	Podpora pro Scrum	Požadavky Project Ownera
<ul style="list-style-type: none"> – nastavitelný proces; – hromadná modifikace záznamů; – filtrování; – full-text vyhledávání; – možnost připojit soubor; – nastavitelné role a uživatelský přístup; – integrace test systému; – různé druhy položek (defekt, theme, epic, story, task); – musí mít dobře zdokumentované API; – musí umět zachovat vazbu mezi požadavky; – reportování a grafy; – nastavitelné varování a odesílání e-mail zpráv; 	<ul style="list-style-type: none"> – prioritizovaný backlog; – možnost přiřadit story a tasky k projektům; – závislosti mezi požadavky; – sprint Planning; – scrum board; – burndown graf; – průběh na každém úkolu; – hromadná aktualizace položek; – oznamování změn; – rozšířené filtrování na bázi autora, role; 	<ul style="list-style-type: none"> – možnost vidět aktuální stav projektů; – suma odpracovaných hodin, zbývajících hodin; – požadavky, které jsou ve zpoždění; – možnost vidět stav celé iterace; – požadavky ve stavech done/progress/not started v iteraci; – estimace projektu a celého plánu; – plánování zdrojů; – příprava a spuštění testů; – monitorování průběhu sprintu;

Zdroj: vlastní zpracování

Pro řízení procesu vývoje webu AUTO.RIA bylo posuzováno několik programových řešení z oblasti agilního řízení.

JIRA od společnosti Atlassian je nejznámější a nejrozšířenější software v oblasti agilního vývoje. JIRA podporuje a usnadňuje proces řízení projektů a požadavků, nabízí flexibilní a uživatelské nástroje pro řízení a sledování pracovníků při výkonu plnění úkolů. Uživatelé si mohou upravit workflow i vizualizaci dle vlastních preferencí a potřeb.

Základní funkce:

- podpora projektového řízení (interní a externí řízení požadavků a úkolů);
- management procesů;
- neustále dostupné informace pro tým přes webové rozhraní;
- sledování a vyhodnocování kapacit;
- historie projektové komunikace;
- podpora pro klientský servis a helpdesk;
- sdílení komunikace, informací a dokumentů v týmu;
- sledování stavu projektu a řešení požadavků zákazníkem;
- úkoly podle priorit, termínů dokončení;
- reporty, projektové statistiky, historie evidence.

Ovládání JIRA je velice intuitivní, pokud jde o základní každodenní úkoly. Systém se dá poměrně snadno na základní úrovni upravovat přímo v administrační části. Pokud jsou potřeba větší změny, je potřeba stáhnout po úpravě zdrojového kódu JIRA, který Atlassian dodává společně s JIRA. Kód je otevřen pro úpravy v konkrétní firmě a upravované řešení se nesmí nikde dále distribuovat. Další možností je programování vlastních modulů, které se do JIRA dají přidat a rozšířit tak funkcionalitu. JIRA má rovněž API, které umožňuje přístup k databázi, v případě, že existuje zájem dělat statistiky nebo zobrazovat data tak, jak to JIRA neumožňuje.

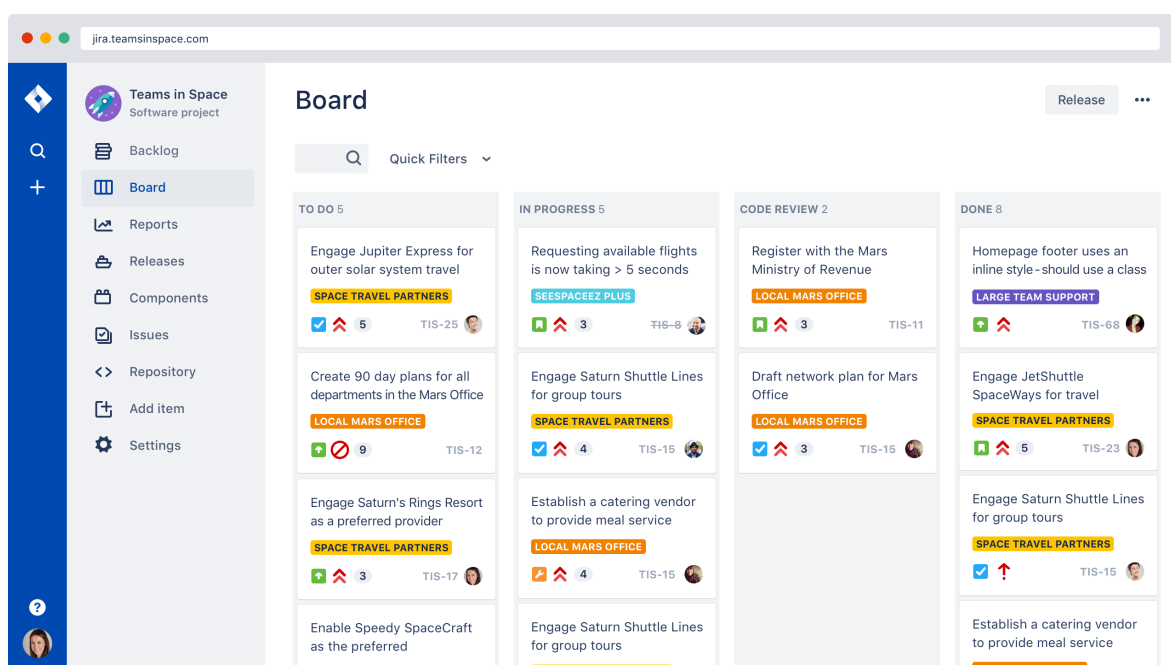
JIRA podporuje základní věci pro použití ve Scrumu. První z nich je Scrum board, bez kterého se žádný tým ve Scrumu neobejde. Každý člen týmu má možnost vidět jak svoje naplánované tasky, tak tasky na celý jeho tým. Může tasky vizuálně kontrolovat a táhnutím myši měnit jejich stav přesunutím do další části Scrum boardu.

Dalším prvkem je Product a Sprint Backlog. V Product backlogu jsou umístěny všechny Epic, na kterých tým bude v budoucnu pracovat, zároveň je to kompletní rozsah produktu. Každý Sprint má svůj vlastní Sprint Backlog, na který je navázaný Sprint report.

Backlog se dá jednoduše prioritizovat a jednotlivé tasky táhnutím myši přidat z Product backlogu do Sprint backlogu.

JIRA umožňuje ze všech dat udělat dashboard, který obsahuje všechny aktuální informace a dá se tak jednoduše aktuálně sledovat stav projektů. Z pohledu Project managementu je velice zajímavý graf poměru vytvořených nových tasků a počtu uzavřených tasků. Z tohoto grafu se dá zjistit, zda zátěž v podobě nových úkolů na tým je z dlouhodobého hlediska zvládatelná a zda výkonnost týmu stačí na splnění všech úkolů. Kromě samotných nástěnek umí JIRA také předkládat agilní reporting v reálném čase, kdykoliv si uživatel přeje vidět výstup.

Obrázek 18. Scrum board v systému JIRA



Zdroj [31]

Výhody: mobilní aplikace, silná podpora řízení backlogu, spousta aplikací pro rozšíření základního software a možnost integrace na jiné informační systémy, úprava software dle vlastní potřeby. Výhodou pro práci developerů je integrace a spojení s klasickými vývojářskými nástroji jako je Bitbucket nebo GitHub.

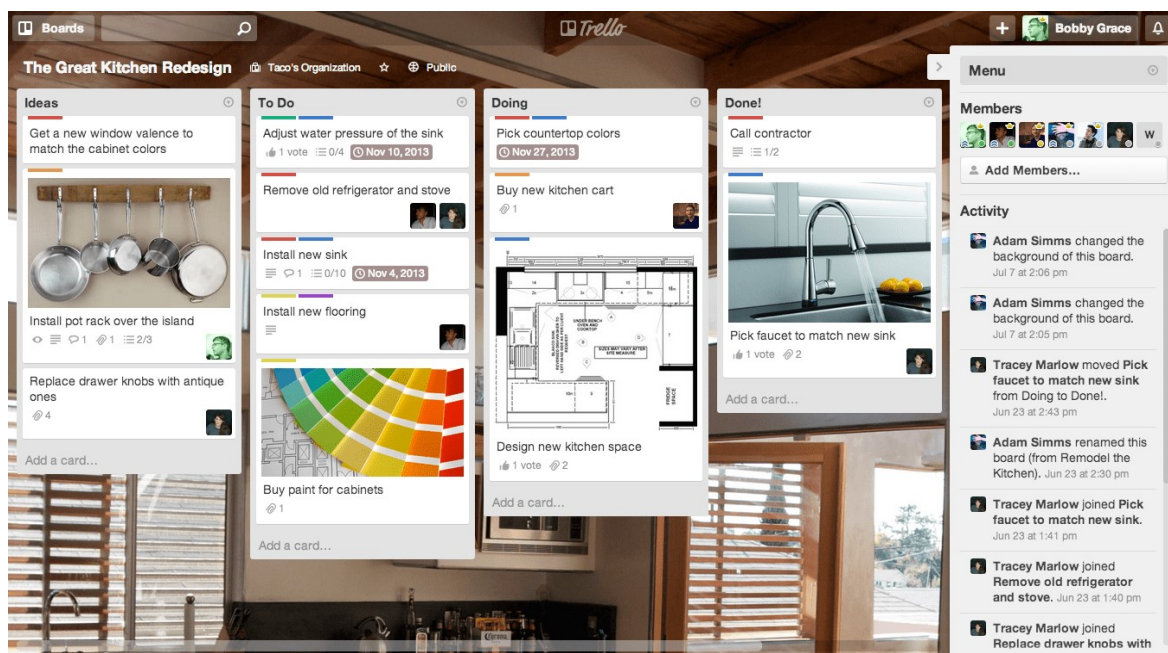
Nevýhody: jedná se o poměrně složitou aplikaci, pro kterou je nutné zaškolení jednotlivých uživatelů, vyšší cena (450\$ měsíčně za verzi v cloudu pro 100 uživatelů nebo 6000\$ jednorázově za serverovou verzi pro 100 uživatelů).

Trello je nejznámější a nejrozšířenější aplikací, která nahrazuje klasickou nástěnku ve Scrumu. Vzniklo už v roce 2011 a tvůrcem originálního programu byla firma Fog Creek

Software, nicméně divize Trello byla v roce 2017 prodána firmě Atlassian. Dle průzkumu tato firma opanuje 53 % programových řešení.

Trello je cloudová služba, která umožňuje přístup k informacím z jakéhokoliv připojeného zařízení, protože data jsou uložena na dostupném zdroji. Základním prvkem v Trello je Karta – tu mohou vytvářet všichni členové týmu a mohou ji přiřadit do správného sloupce ve Scrumu. Karta v podstatě definuje určitý úkol nebo cíl, ale může v sobě mít další podseznamy úkolů, které jsou k jejímu splnění zapotřebí. Trello pak graficky znázorňuje i aktuální stav v procentech dokončení. Informace se zásadně vyměňují uvnitř dané karty, ke které se vztahují, a to včetně příkládaných odkazů nebo souborů.

Obrázek 19. Scrum board v Trello



Zdroj [32]

V Trello je možné nastavit práva k nástěnce a je na výběr ze tří možností:

1. administrátor – může měnit kartičky, mazat uživatele a měnit nastavení nástěnky;
2. normální – může měnit kartičky, nemůže měnit nastavení nástěnky;
3. pozorovatel – může si kartičky zobrazovat, komentovat a hlasovat o nich, ale nemůže nic měnit.

Trello má své nativní aplikace pro všechny mobilní systémy. Každá pracovní deska i jednotlivé karty mají navíc unikátní URL adresu, která umožňuje snadné sdílení nejen v rámci týmu, ale v případě potřeby také se zákazníkem.

Pro projektový tým AUTO.RIA navrhuji software Trello, protože je flexibilní a je možné ho použít i v rámci bezplatné verze. Řeší se v něm klasické scénáře agilního vývoje včetně plánování sprintů apod.

Při používání Trelly projektovým týmem AUTO.RIA navrhuji uplatnit několik klíčových principů, které zvýší výslednou efektivitu:

1. nejdůležitější úkol je nahoře (díky tomu řešitel okamžitě ví, který úkol má dělat);
2. jeden úkol v jeden moment (nikdo nedokáže řešit v jeden moment více úkolů najednou; pokud řešitel pracuje na jiném úkolu, vrací původní úkol zpět);
3. kartička začíná slovesem (sloveso charakterizuje konkrétní činnost, upřesňuje se tím zadání a volba odpovědného řešitele);
4. jeden člověk na kartičku (cílem tohoto principu je zajistit jasnou odpovědnost za konkrétní úkol; pokud je na kartičce více lidí, snižuje se pravděpodobnost, že se úkol vyřeší);

Trello umožňuje několik způsobů, jak zadat dobu trvání úkolu. Správné měření a odhadování doby, která je potřebná pro jednotlivé úkoly, je velmi důležitá. Oblíbeným doplňkem pro Trello je tzv. Scrum for Trello, které už v názvu kartičky obsahuje informaci o odhadovaném potřebném času a také o času, který byl na daném úkolu nakonec skutečně potřeba.

Tým AUTO.RIA se rozhodl použít Slack jako nástroj, zajišťující textovou komunikaci a sdílení souborů. Aplikace Slack nabízí obecnou chatovací místnost pro komunikaci mezi dvěma uživateli, a také malé či zvláštní chatovací místnosti, které je možné používat pro různá témata. Obchod s rozšířením nabízí velmi zajímavé integrace, díky kterým bude vaše práce se Slackem a celým týmem zase o kus efektivnější.

Jedná se o velmi jednoduchý a přehledný komunikační nástroj. Aplikace je vytvořená přímo pro firemní užití a nabízí vysoké standardy v zabezpečení. Primárně je k aplikaci Slack přistupováno skrze prohlížeč na počítači. Slack ovšem nabízí také aplikaci pro mobilní telefony s operačním systémem iOS a Android. Výhodou je nejen jeho jednoduché a velmi přehledné prostředí, ale také to, že je v základu zdarma a umožňuje propojení mezi Slackem a dalšími aplikacemi. Díky tomu je možné pracovat i mimo kancelář.

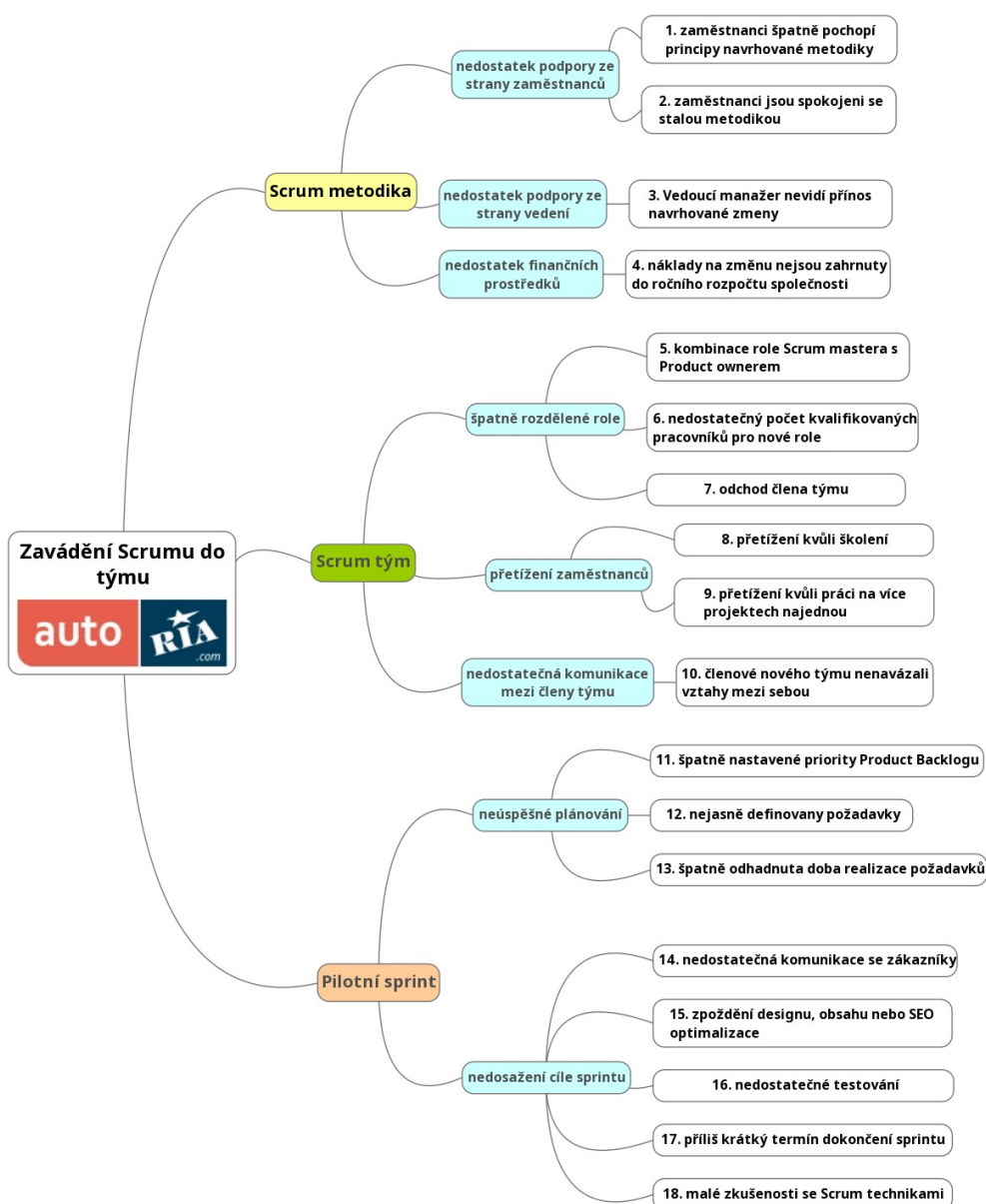
Navržené řešení díky častějšímu opakování jednotlivých činností přinese společnosti lepší dodržování procesních kroků, protože doba mezi jednotlivými kroky bude kratší a šance na osvojení si špatných návyků tak bude nižší.

5.4 Řízení rizik při zavádění Scrumu

Společnost RIA.com nemá v současné době jednotný systém analýzy a řízení rizik. Řízení rizik je vnímáno, ale je to jen formální a nemá to praktické dopady do řízení.

Identifikace rizik je jednou z nejvýznamnějších fází řízení rizik, a tedy také klíčovým prvkem pro tvorbu doporučení při zavedení Agilního přístupu do projektového řízení společnosti RIA.com. Cílem identifikace rizik je sestavit seznam formulovaných hrozeb a dopadů. Rizika jsem identifikovala na základě vlastního pozorování a snažila jsem se nalézt, která nebezpečí mohou ohrozit zavedení Scrumu do řízení projektu AUTO.RIA.

Obrázek 20. Myšlenková mapa



Zdroj: vlastní zpracování

Výstupem je Registr rizik (viz Příloha A.), jehož cílem je vyjmenování a identifikace všech potenciálních rizik. Registr rizik zahrnuje aktuální popis 18 rizik, které mohou nastat při implementaci Scrumu ve společnosti RIA.com. Výsledky identifikace rizik se používá pro analýzu a hodnocení rizik. Během analýzy se do Registru rizik zaznamenává:

1. Pravděpodobnost výskytu pomocí číselného vyjádření na stupnici 1-10 (1 – téměř určitě riziko nenastane, 10 – velmi pravděpodobně riziko nastane),
2. Ohodnocení dopadu – dopad rizika v kontextu projektových cílů, hodnota 1 – minimální dopad (neovlivní projekt), 10 katastrofický dopad (projekt končí),
3. Očekávaná hodnota rizika (OHR) – určí se vynásobením pravděpodobnosti rizika a kvantitativní hodnoty jeho negativního dopadu.

Každému riziku byla přiřazena výše pravděpodobnosti, že nastane a dopad jaký by tato situace měla na projekt. Celkový význam rizika vyjadřuje hodnota rizika, která je součinem dopadu a pravděpodobnosti výskytu.

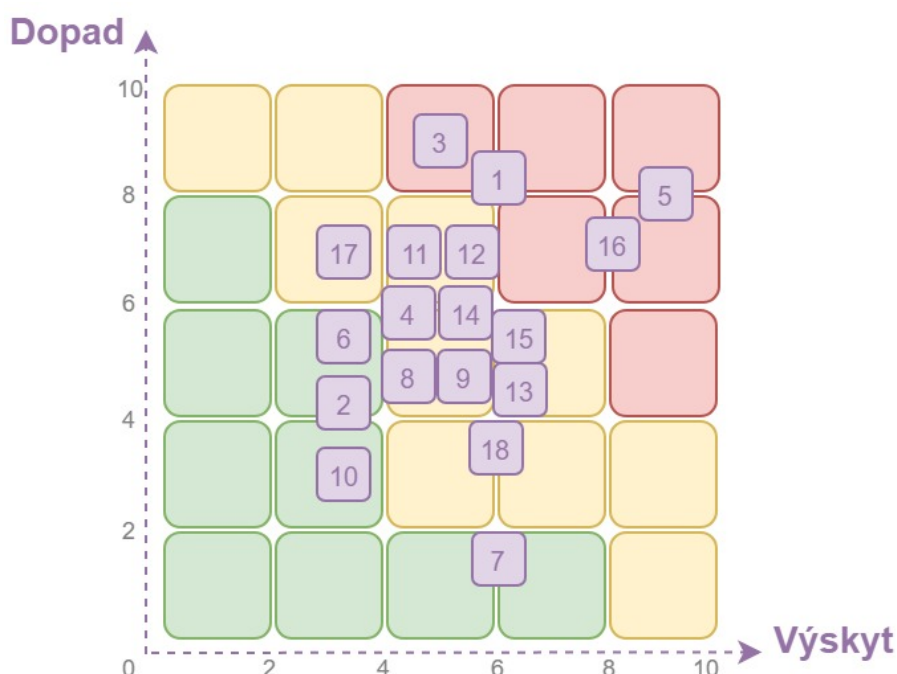
Tabulka 7. Hodnocení rizik

ID	Název	Výskyt	Dopad	Hodnota
5	kombinace role Scrum mastera s Product Ownerem	9	8	72
16	nedostatečné testování	8	7	56
3	Vedoucí manažer nevidí přínos navrhované změny	5	9	45
1	zaměstnanci špatně pochopí principy navrhované metodiky	6	8	48
12	nejasně definované požadavky	6	7	42
15	zpoždění designu, obsahu nebo SEO optimalizace	6	7	42
11	špatně nastavené priority Product Backlogu	5	7	35
14	nedostatečná komunikace se zákazníky	5	7	35
13	špatně odhadnuta doba realizace požadavků	5	6	30
9	přetížení kvůli práci na více projektech najednou	5	6	30
17	příliš krátký termín dokončení sprintu	4	7	28
4	náklady na změnu nejsou zahrnuty do ročního rozpočtu společnosti	4	6	24
8	přetížení kvůli školení	4	6	24
18	malé zkušenosti se Scrum technikami	6	4	24
6	nedostatečný počet kvalifikovaných pracovníků pro nové role	3	6	18
2	zaměstnanci jsou spokojeni se stálou metodikou	3	5	15
10	členové nového týmu nenavázali vztahy mezi sebou	3	5	15
7	odchod pracovníka týmu	6	2	12

Zdroj: vlastní zpracování

Risk Appetite nastavuje pravidla projektovému manažerovi pro řízení rizik na projektu. To, jak velkou hodnotu rizika si můžeme dovolit přijmout, by mělo vyplynout z firemní strategie řízení rizik. Jelikož společnost RIA.com takovou strategii nemá, musí si hodnotu akceptovatelného rizika určit Vedoucí manažer ve spolupráci s projektovým týmem. Vedoucí manažer se rozhodl, že pokud je hodnota nižší než 20, může být takové riziko akceptováno. Akceptovaná rizika nejsou dále řešena, nicméně je třeba je dále v průběhu realizace monitorovat a prověřovat, zda se jejich parametry nemění. I tato rizika by tedy měla mít své vlastníky – osoby odpovědné za řešení daného rizika. Výstupem je Matice rizik.

Obrázek 21. Matice rizik



Zdroj: vlastní zpracování

Podle výsledků uvedených na obrázku 21 je zřejmé, že:

1. mezi kritická patří rizika: 1, 3, 5, 16
2. mezi středně významná rizika patří: 4, 8, 9, 11, 12, 13, 14, 15, 17, 18
3. mezi méně významná rizika pak patří: 2, 6, 7, 10

Z analýzy rizik lze vidět, že Projekt je velmi rizikový. Z celkových 18 rizik má 10 rizik střední hodnotu a 4 rizika dokonce vysokou hodnotu. Tato situace je velmi nežádoucí.

Nejkritičtější jsou rizika, ohrožující přijetí metodiky Scrum ve společnosti RIA.com. Je zřejmé, že špatné vysvětlení navrhované metodiky způsobí odmítavý postoj ze strany zaměstnanců a Vedoucího manažera. Proto je velmi důležité provést úvodní prezentaci a vysvětlit všechny výhody Scrumu, kvalitativně zpracovat zpětnou vazbu a určit

ekonomický prospěch společnosti. To je nepochybně jedna z prvních oblastí, na kterou by bylo vhodné se zaměřit.

Obrana vůči rizikům (viz Příloha B.) má za cíl snížit celkovou hodnotu všech rizik na takovou úroveň, aby projekt byl s vysokou pravděpodobností úspěšně realizovatelný. Je třeba se zamyslet, jaká konkrétní opatření budou přijata a kdo za ně bude zodpovídat. Pro rizika, která byla ohodnocena jako kritická či středně významná, budou navržena opatření pro snížení jejich hodnoty. Méně významná rizika budou akceptovaná.

Tabulka 8. Monitorování rizik

ID	Název	Metoda	Vlastník rizika
1	zaměstnanci špatně pochopí principy navrhované metodiky	Akceptační protokol (hlasování)	Senior Scrum master
2	zaměstnanci jsou spokojeni se stálou metodikou	Checklist (školení, hlasování, schůzky, osobní setkání)	Senior Scrum master
3	Vedoucí manažer nevidí přínos navrhované změny	Akceptační protokol	Senior Scrum master
4	náklady na změnu nejsou zahrnuty do ročního rozpočtu společnosti	Pracovní výkaz Rozpočet společnosti	Vedoucí manažer
5	kombinace role Scrum mastera s PO	Akceptační protokol	Personalista
6	nedostatečný počet kvalifikovaných pracovníků pro nové role	Checklist (školení, výběrové řízení, hodnocení znalosti a dovednosti zaměstnanců)	Personalista
7	odchod pracovníka týmu	Checklist (vnitřní tréninky, přenos záležitostí a úkolů)	Personalista
8	přetížení kvůli školení	Checklist (plán školení, adaptační plán)	Personalista
9	přetížení kvůli práci na více projektech najednou	Akceptační protokol	Personalista
10	členové nového týmu nenavázali vztahy mezi sebou	Checklist (teambuilding, prezentace projektu, komunikační matice)	Scrum master AUTO.RIA
11	špatně nastavené priority Product Backlogu	Checklist (Pre-planning, hodnocení) Akceptační protokol (výsledek Pre-planningu)	Product Owner
12	nejasně definované požadavky	Akceptační protokol	Product Owner
13	špatně odhadnuta doba realizace požadavků	Pracovní výkaz	Team Leader
14	nedostatečná komunikace se zákazníky	Checklist (pracovní schůzky)	Product Owner
15	zpoždění designu, obsahu nebo SEO optimalizace	Akceptační protokol	Product Owner
16	nedostatečné testování	Pracovní výkaz	Team Leader
17	příliš krátký termín dokončení sprintu	Pracovní výkaz	Scrum master AUTO.RIA
18	malé zkušenosti se Scrum technikami	Checklist (školení, ověření skutečnosti)	Senior Scrum master

Zdroj: vlastní zpracování

Součástí monitoringu rizik je různé audity a kontroly, které mají za účel ověřit efektivitu zvolených opatření proti rizikům. Do procesu kontroly a sledování rizik vstupují: Plán řízení rizik, Registr rizik, Schválené požadavky na změny. Monitorování je nutné provádět průběžně a stále až do uzavření projektu, protože může dojít k řadě možných událostí. Některá rizika mohou nastat kdykoliv. Mezi taková například patří: odchod pracovníka týmu, špatně stanovené požadavky zákazníkem. Proti těmto rizikům je třeba mít se na pozoru po celou dobu trvání projektu. V případě zaznamenané změny rizika nebo nové vážné hrozby dochází k eskalaci podle odpovědností a podle Plánu řízení rizik.

Ve společnostech RIA.com není definován eskalační mechanismus, který by podle předem známých odpovědností rozhodoval ve věci řízení rizik. Za řízení rizik na projektu odpovídá projektový manažer, který nemá možnosti vyvinout reakce na rizika (zastavit projekt, přidělit více zdrojů atd). Projektový manažer eskaluje rizika s významným dopadem na cíle projektu okamžitě Vedoucímu manažerovi, který má pravomoc, odpovědnost, a hlavně schopnost k reakci na riziko a měl by tedy vyvíjet adekvátní opatření proti rizikům.

6 Závěr

Tato diplomová práce je věnována zavádění agilního přístupu Scrum pro řízení softwarových projektů ve vývojovém týmu AUTO.RIA. Cílem je vypracování vlastních návrhů a doporučení pro zavádění agilních metodik do projektu, který byl doposud řízen pomocí vodopádového přístupu.

V teoretické části práce jsou popsány základní charakteristiky projektového řízení, životní cyklus, mezinárodní standardy. Je zde vysvětlen Vodopádový model vývoje softwaru, jeho jednotlivé fáze, výhody, nevýhody a vhodné projekty. V oblasti vývoje softwaru je vodopádový model již považován za překonaný, protože postrádá flexibilitu a možnost reagovat na vzniklé události. Software je testován na konci životního cyklu, což může být příliš pozdě, aby něco změnit. Model také neumožňuje návraty k předchozím fázím projektu, a proto pokud dojde k chybě na začátku projektu, například v analýze, je oprava chyby v pozdější fázi velmi nákladná. Náklady na zavedení změn je důvodem přechodu na jiný přístup k vývoji softwaru.

Agilní metodiky vznikly pro řešení problémů při řízení softwarových projektů, kde šlo zejména o otázky příliš dlouhého uvolnění, příliš dlouhé doby stabilizace, složitého vykonávání změn v průběhu projektu a klesající kvality výstupů. Tyto metodiky jsou ideální pro projekty s nejasným nebo často se měnícím zadáním, umožňují rychlý vývoj softwaru a zároveň dokážou reagovat na změnu požadavků v průběhu vývojového cyklu. V diplomové práci jsou popsány metodiky Kanban, Lean, XP a Scrum. Je detailně rozebrán agilní přístup Scrum, jsou rozepsány role ve Scrum týmu, artefakty a schůzky.

Dále se práce zaměřuje na vzájemné porovnání Agilního a Vodopádového přístupu k řízení projektů, určení jejich výhod a nevýhod. V důsledku toho je zřejmé, že Agilní přístup přináší mnohem vyšší flexibilitu, dodržování harmonogramu, lepší týmovou spolupráci a motivaci, efektivitu vynaloženého úsilí a úspěšné dokončení projektu. Z tohoto důvodu tým AUTO.RIA zvolil Scrum jako metodiku, vhodnou pro vývoj.

V praktické části práce je charakterizována vybraná společnost RIA.com, její organizační struktura a portfolio projektů. Následně je popsán současný stav a způsob řízení projektů. Společnost se specializuje především na vývoj vlastních softwarových produktů a už dlouho se rozhoduje pro změnu metodiky projektového řízení. Vedení společnosti si nemůže dovolit riskovat a změnit tuto metodiku najednou na všech projektech. Proto byl vybrán experimentální projekt – internetový portál AUTO.RIA.com.

K identifikaci problémů bylo provedeno průzkum zaměstnanců, zapojených do vývoje webu auto.ria.com. Pro provedení průzkumu byla použita modifikovaná metoda "360° zpětná vazba". Projektový tým AUTO.RIA byl hodnocen nadřízenými, kolegy z jiných oddělení, zákazníky a členy vlastního týmu. V důsledku analýzy dotazníkového šetření byly formulovány základní problémy projektového týmu AUTO.RIA při vývoje softwaru:

1. Problémy při zadávání úkolů;
2. Neprůhledná práce týmu AUTO.RIA;
3. Nedostatečná produktivita;
4. "Technický dluh";
5. Problémy s aktualizací webu auto.ria.com.

Zmíněné problémy vedly k přetěžování zaměstnanců, růstu rozsahu práce a zvýšení chybovosti implementovaných komponent. Jasná definice problémů se stala základem pro další strategická rozhodnutí. Bylo zřejmé, že RIA.com musí projít zásadní změnou a začít se orientovat na zákazníky, na jejich potřeby a očekávání a na to, zda je vyvíjený web vhodný pro každého uživatele.

Diplomová práce se dále věnuje vypracování konkrétních návrhů na zavádění Scrumu, výběru vhodného software a řízení rizik. K získání potřebných dat jsou použity vlastní pozorování, rozhovory s manažery a zaměstnanci, dotazníkové šetření. Výsledný Návrh se skládá z jednotlivých po sobě navazujících kroků včetně vhodných aktivit:

1. Seznámení se Scrumem;
2. Transformace vývojového týmu;
3. Příprava pracovního prostoru;
4. Příprava na plánování sprintu;
5. První plnohodnotný Sprint.

Navrhované změny se soustředí na zlepšení komunikace, zefektivnění vývojového procesu a zvýšení objemu dokončené práce. Na konci kapitoly byla provedena analýza dodržování agilních principů při řízení projektu navrženou metodou. Z tabulky č.5 je zřejmé, že řízení projektů v týmu AUTO.RIA splňuje všechny agilní principy, sedm úplně a pět částečně.

Dále byla provedena analýza software pro podporu agilního řízení. Pro projektový tým AUTO.RIA navrhuji software Trello, protože je flexibilní a je možné jej použít i v rámci bezplatné verze. Řeší se v něm klasické scénáře agilního vývoje včetně plánování

sprintů, řízení Scrum board apod. Oblíbeným doplňkem pro Trello je tzv. Scrum for Trello. Jako nástroj, zajišťující textovou komunikaci a sdílení souborů, navrhuji Slack. Navržené řešení přinese lepší dodržování procesních kroků a efektivnější komunikace.

Aby zhodnotit proces zavádění Scrumu na základě subjektivního vnímání, proběhlo dotazníkové šetření, které mělo za cíl zjistit, jaké změny jsou nejvíce a nejméně efektivní, jaké překážky kolegové vidí na cestě ke změnám, zda provedené změny řešily problémy projektu. Bylo vybráno 45 respondentů z divizí marketingu a péče o zákazníka, kteří hodnotili vzájemnou spolupráci s vývojáři, jejich flexibilitu, kvalitu a rychlost dodávek. Výsledky tohoto dotazníkového šetření ukazují, že jsou kolegové s agilním řízením spokojeni. Líbí se jim především vysoká flexibilita a rychlost dodání, zlepšily se komunikace a spolupráce v rámci vývoje. Spokojenost vývojářů s jejich prací se zvýšila a získalo se tak více času na kontrolu prováděné práce, testování a dokumentování kódu. Rovněž se zlepšilo obecné povědomí o práci kolegů. Zaměstnanci přijali Scrum ve větší míře kladně a společně se snaží tyto metodiky dodržovat a neustále si je osvojovat.

V celém procesu je stále spousta chyb a pořád je potřeba proces zlepšovat. Ale i přes veškeré problémy se zavádění Scrumu prokazatelně vyplatí. Za nejpodstatnější výhody se považují získání pravidelné zpětné vazby od zákazníků, kratší doba doručení produktu, méně času stráveného plánováním, flexibilita procesů, inovace a srozumitelnější úkoly.

Proto tento návrh může sloužit jako praktická ukázka pro zavádění Scrumu dalším vývojovým týmům ve společnosti RIA.com. Cíl diplomové práce byl tak splněn a její přínos byl potvrzen projektovým týmem AUTO.RIA.

7 Seznam použitých zdrojů

1. *A guide to the project management body of knowledge (PMBOK guide)*. Fifth edition. Newtown Square, Pennsylvania: Project Management Institute, [2013]. ISBN 978-1-935589-67-9 [online]. Dostupný z: https://dinus.ac.id/repository/docs/ajar/PMBOKGuide_5th_Ed.pdf
2. *Agilní metodiky* [online]. Dostupný z: https://cs.wikipedia.org/wiki/Agilní_metodiky
3. BARTOŠKA, Jan. *Přednáškové materiály*. Praha: ČZU v Praze, 2020
4. BUCHALCEVOVÁ, Alena. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1075-7.
5. *Burndown chart* [online]. Dostupný z: https://cs.wikipedia.org/Burndown_chart
6. DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO. *Projektový management podle IPMA*. 2., aktualiz. a dopl. vyd. Praha: Grada, 2012. Expert (Grada). ISBN 978-80-247-4275-5.
7. *Internationall Project Management Association (IPMA)* [online]. Dostupný z: https://cs.wikipedia.org/wiki/International_Project_Management_Association
8. *IPMA® Česká republika* [online]. Dostupný z: <https://www.ipma.cz/>
9. HUNDERMARK, Peter. *Do Better Scrum*. InfoQ, 2015. ISBN 978-1-4799-2504-9.
10. KADLEC, Václav. *Agilní programování: metodiky efektivního vývoje softwaru*. Brno: Computer Press, 2004. ISBN 80-251-0342-0.
11. KERZNER, Harold. *Project management: a systems approach to planning, scheduling, and controlling*. Twelfth edition. Hoboken, New Jersey: Wiley, 2017. ISBN 978-1-119-16535-4.
12. KNIBERG, Henrik, SKARIN, Mattias. *Kanban and Scrum - Making the Most of Both*. C4 Media Inc, 2010. ISBN 978-0-557-13832-6.
13. KNIBERG, Henrik. *Scrum and XP from the Trenches*, C4 Media Inc, 2007, ISBN 978-1-329-22427-8
14. MÁCHAL, Pavel, Martina ONDROUCHOVÁ a Radmila PRESOVÁ. *Světové standardy projektového řízení: pro malé a střední firmy: IPMA, PMI, PRINCE2*. Praha: Grada, 2015. Manažer. ISBN 978-80-247-5321-8.

15. *Manifest agilního vývoje software* [online]. Ward Cunningham © 2001. Dostupný z: <http://agilemanifesto.org/iso/cs/manifesto.html>
16. *Metodiky vývoje software. Studijní opora* [online]. 2018. Dostupný z: <https://mvso.cz/Metodiky-vyvoje-software-studijni-text.pdf>
17. MYSLÍN, Josef. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016. ISBN 978-80-251-4650-7.
18. *PMI - Project Management Institut (studijní materiál)* [online]. 2011. Dostupný z: https://moodle.czu.cz/uvod_do_PMI_studijni_text.pdf
19. *PRINCE2 - PRojects IN Controlled Environments no. 2 (studijní materiál)* [online]. 2011. Dostupný z: https://moodle.czu.cz/uvod_do_PRINCE2_studijni_text.pdf
20. *Principy stojící za Agilním Manifestem* [online]. Ward Cunningham © 2001 [vid. 2017-10-01]. Dostupný z: <http://agilemanifesto.org/iso/cs/principles.html>
21. ŘEPA, Václav. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999. ISBN 80-86119-13-0.
22. SCHWABER, Ken, SUTHERLAND, Jeff. *Průvodce Scrumem. Pravidla hry*. [online]. ScrumGuides.org © 2016. Dostupný z: <http://www.scrumguides.org/docs/scrumguide.pdf>
23. SCHWALBE, Kathy. *Řízení projektů v IT: kompletní průvodce*. Brno: Computer Press, 2011. ISBN 978-80-251-2882-4.
24. *Scrum role: Scrum tým* [online]. 2014. Dostupný z: <https://manifesto.co.uk/scrum-roles-scrum-team/>
25. ŠOCHOVÁ, Zuzana a Eduard KUNCE. *Agilní metody řízení projektů*. Brno: Computer Press, 2014. ISBN 978-80-251-4194-6.
26. SOMMERVILLE, Ian. *Softwarové inženýrství*. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.
27. STELLMAN, Andrew a Jennifer GREENE. *Learning Agile*. Beijing: O'Reilly, [2014]. ISBN 978-144-9363-826.
28. SVOZILOVÁ, Alena. *Projektový management: systémový přístup k řízení projektů*. 3., aktualizované a rozšířené vydání. Praha: Grada Publishing, 2016. Expert (Grada). ISBN 978-80-271-0075-0.
29. SVOZILOVÁ, Alena. *Projektový management*. 2., aktualiz. a dopl. vyd. Praha: Grada, 2011. Expert (Grada). ISBN 978-80-247-3611-2.

30. *Úvod do softwarového inženýrství IUS (studijní opora)* [online]. 2006. Dostupný z: http://suave_skola.varak.net/lectures_IUS/opora_IUS.pdf
31. *Jira Software* [online]. Dostupný z: <https://www.atlassian.com/cs/software/jira>
32. *Trello* [online]. Dostupný z: <https://trello.com/>

8 Přílohy

Příloha A. Registr rizik

ID	Název	Scénář	Aktivum	Návrh prevence	Odpovědnost
1	zaměstnanci špatně pochopí principy navrhované metodiky	špatné vysvětlení metodiky způsobí odmítavý postoj ze strany zaměstnanců	Scrum metodika Zaměstnanci	– prezentace výhod pro zaměstnance – průzkum zaměstnanců	Senior Scrum master
2	zaměstnanci jsou spokojeni se stálou metodikou	zaměstnanci nejsou motivováni a nechtějí se účastnit implementace Scrumu	Scrum metodika Zaměstnanci	– provedení workshopu, teambuildingu nebo školení pro nasazení Scrumu – pozitivní motivace zaměstnanců – rozdělování projektů dle preferencí	Senior Scrum master
3	Vedoucí manažer nevidí přínos navrhované změny	Vedoucí manažer nevidí potřebu ve změně metodiky a ekonomický přínos této změny	Scrum metodika	– prezentace výhod pro Vedoucího manažera	Senior Scrum master
4	náklady na změnu nejsou zahrnuty do ročního rozpočtu společnosti	společnost nepřiděluje finanční prostředky na zavedení nové metodiky	Scrum metodika	– předběžné plánování rozpočtu na zavedení – vytvoření finančního plánu projektu a zajištění externích zdrojů a dotací (úvěr z jiného projektu v rámci společnosti) – do plánu projektu přidat čas na pečlivou kontrolu všech nákladů	Vedoucí manažer
5	kombinace role Scrum mastera s Product Ownerem	bývalý projektový manažer zastoupí obě role v počátečních fázích implementace Scrumu	Scrum tým	– vysvětlení obrovského rozdílu mezi rolími – provádění individuálních školení pro každou roli – výběr vhodných pracovníků s ohledem na jejich kvalifikaci	Vedoucí manažer
6	nedostatečný počet kvalifikovaných pracovníků pro nové role	nové role zastoupí stávající pracovníci bez potřebné praxe a znalostí	Scrum tým	– provádění individuálních školení pro každou roli – provedení workshopu nebo teambuildingu pro nasazení Scrumu – výběrové řízení mezi zaměstnanci společnosti – hodnocení znalosti a dovednosti zaměstnanců	Personalista

7	odchod pracovníka týmu	změna pracovní pozice v rámci firmy nebo úplná výpověď	Scrum tým	<ul style="list-style-type: none"> – vnitřní tréninky pro sdílení zkušeností – rozdělování projektů dle preferencí 	Scrum master AUTO.RIA
8	přetížení kvůli školení	provádění dalších školení a vzdělávacích programů mimo pracovní dobu	Scrum tým	<ul style="list-style-type: none"> – koordinace plánu školení s manažery – příprava adaptačního plánu pro každého zaměstnance 	Senior Scrum master
9	přetížení kvůli práci na více projektech najednou	designér a marketingové specialisti pracují na více projektech najednou	Scrum tým	<ul style="list-style-type: none"> – spolupráce s externisty – zahrnutí copywritera do týmu – plánování ve sprint pouze těch úkolů, pro které je schválen design nebo optimalizace 	Scrum master AUTO.RIA
10	členové nového týmu nenavázali vztahy mezi sebou	do stávajícího projektového týmu byli přidáni noví zaměstnanci, kteří nahradí nové role ve Scrum týmu	Scrum tým	<ul style="list-style-type: none"> – teambuilding pro nový Scrum tým – prezentace projektu pro nové členy týmu – příprava komunikační matice 	Scrum master AUTO.RIA
11	špatně nastavené priority Product Backlogu	Product Backlog v sobě soustředí veškeré požadavky na funkcionalitu produktu, stále se mění podle aktuálních potřeb a nikdy není úplný. Product Owner pomocí určení priority má přímý vliv na položky budoucího Sprintu, jelikož Tým vždy řeší položky nejvyšší priority. Kvůli špatnému určení priority se nejdůležitější úkoly nemusí do sprintu	Pilotní sprint	<ul style="list-style-type: none"> – schválení priorit s Vedoucím manažerem – povinný Preplanning, aby byly vybrány úkoly pro následující sprinty – sdílet Product Backlog zainteresovaným skupinám 	Product Owner
12	nejasně definované požadavky	vývojáři nechápou problém a nemohou nabídnout vhodné řešení	Pilotní sprint	<ul style="list-style-type: none"> – příprava šablony pro User stories – provedení diskusí i v průběhu projektu, aby se obě strany ujistily, že projekt je vypracovaný dle požadavků zákazníka, a nedocházelo k nedorozuměním 	Product Owner

13	špatně odhadnuta doba realizace požadavků	malé zkušenosti se Scrum technikami hodnocení termínů úkolů	Pilotní sprint	<ul style="list-style-type: none"> – použití pro plánování skutečné doby trvání jednotlivých činností, které jsou známy z předchozích projektů – založení časové rezervy, která může být vypočtena jednotlivě na činnosti 	Scrum tým
14	nedostatečná komunikace se zákazníky	není stanovené způsoby a techniky externí komunikace, a proto uživatel často dostává produkt, který nesplňuje jeho očekávání a požadavky nebo který vůbec nechtěl	Pilotní sprint Zákazníci Uživatelé	<ul style="list-style-type: none"> – stanovení odpovědné osoby pro komunikace mezi zákazníkem a projektovým týmem – spolupráce s Oddělením péče o zákazníky 	Product Owner
15	zpoždění designu, obsahu nebo SEO optimalizace	možnost nekvalitně odvedené práce, přetížení zaměstnanců a časové zpoždění sprintu z důvodu zpoždění designu, obsahu nebo SEO optimalizace	Pilotní sprint	<ul style="list-style-type: none"> – spolupráce s externisty – plánování ve sprint pouze těch úkolů, pro které je schválen design nebo optimalizace 	Product Owner
16	nedostatečné testování	vzhledem k tomu, že testování je poslední činnost prováděná před spuštěním, tato fáze se zkracuje kvůli nedostatku času a může se stát, že produkt nebude dostatečně testován	Pilotní sprint	<ul style="list-style-type: none"> – vytvoření časové rezervy pro to, aby bylo na testování výrobku věnováno tolik času, kolik je ho skutečně potřeba – příprava přesného checklistu pro testování – zajištění dostatečného počtu pracovníků 	Scrum master Scrum tým
17	příliš krátký termín dokončení sprintu	tým nestihne se přeorientovat na dvoutýdenní iterace	Pilotní sprint	<ul style="list-style-type: none"> – dostatečné plánování – časová rezerva – zajištění dostatečného počtu pracovníků – použití Scrum tabuli 	Scrum master Scrum tým
18	malé zkušenosti se Scrum technikami	malé zkušenosti se Scrum technikami, s používáním softwaru	Pilotní sprint	<ul style="list-style-type: none"> – školení potřebným Scrum technikám 	Scrum master

Zdroj: vlastní zpracování

Příloha B. Obrana vůči rizikům

ID	Název	OHR	Postoj k riziku	Opatření	Zodpovědnost pro zajištění
1	zaměstnanci špatně pochopí principy navrhované metodiky	48	odmítnutí	provést průzkum a tajné hlasování "pro nebo proti Scrumu" a začít plánovat implementaci až po schválení	Senior Scrum master
2	zaměstnanci jsou spokojeni se stálou metodikou	15	akceptace	předchozí změny ve vedení nebyly významné a nevyřešily problémy zaměstnanců, a tak oni ztratili svou motivaci a již nechtějí nic měnit. Provedení teambuildingu pro nasazení Scrumu a pozitivní motivace zaměstnanců snižují toto riziko	Personalista
3	Vedoucí manažer nevidí přínos navrhované změny	45	odmítnutí	začít plánovat implementaci Scrumu až po schválení tohoto nápadu u Vedoucího manažera	Senior Scrum master
4	náklady na změnu nejsou zahrnuty do ročního rozpočtu společnosti	24	přenesení	po schválení rozpočtu Vedoucím manažerem je riziko převedeno na finanční oddělení	Finanční ředitel
5	kombinace role Scrum mastera s Product ownerem	72	odmítnutí	nastavit povinnou podmínku: rozdělení těchto rolí v každém týmu, nebo alespoň jeden Scrum master pro několik týmů v počáteční fázi implementace (mají stejné problémy a potíže)	Personalista
6	nedostatečný počet kvalifikovaných pracovníků pro nové role	18	oslabení	provést školení zaměstnanců a následné hodnocení znalosti a dovednosti pro nové role	Personalista
7	odchod pracovníka týmu	12	akceptace	riziko má nízkou hodnotu, protože společnost RIA.com své zaměstnance pozitivně motivuje benefity a příjemným pracovním prostředím. Riziko se dá snížit tím, že pracovníci dostávají projekty dle vlastních preferencí a účastní se sdílení zkušeností	Scrum master AUTO.RIA
8	přetížení kvůli školení	24	oslabení	nastavení správného time managementu zaměstnanců a odhalení slabých míst jejich plánování, odhalení přetíženosti, soustředění na důležitých úkolech	Personalista
9	přetížení kvůli práci na více projektech najednou	30	přenesení	správná volba a složení týmu jsou důležité pro minimalizaci důsledků rizika, proto možná opatření jsou spolupráce s externisty a zahrnutí copywritera do týmu	Manažeři příslušných oddělení
10	členové nového týmu nenavázali vztahy mezi sebou	15	akceptace	riziko má nízkou hodnotu, protože společnost RIA.com nepracuje na dálku, všichni zaměstnanci jsou ve stejné kanceláři a dobře se znají. Proto stačí uspořádat teambuilding a prezentaci projektu pro nový Scrum tým	Scrum master AUTO.RIA
11	špatně nastavené priority Product Backlogu	35	oslabení	pravidelná údržba Backlogu, která zjednodušuje plánování a pomáhá dobře chápat, v jakém stadiu se nachází projekt a kam směřuje jeho vývoj. Kvalitní popis User Story na 2-3 sprinty dopředu a následný odhad přidané hodnoty a nákladu umožňují správně seřadit jednotlivé úkoly podle priorit	Product Owner

12	nejasně definované požadavky	42	přenesení	snížení významu tohoto rizika lze dosáhnout především vhodnější komunikací se zákazníky. Nejasnosti v zadávání úkolů lze snížit vhodnými opatřeními, jak přesně definovat úkol (šablona User Story) a důsledným dodržováním pravidel	Asistent Product Ownera
13	špatně odhadnuta doba realizace požadavků	30	oslabení	s postupem času jednotliví členové týmu dokáží přesněji odhadovat čas potřebný pro realizace požadavků, čímž riziko postupně snižuje svůj význam. Dopad tohoto rizika lze snížit použitím vhodnější metody pro odhadování doby v Agilním řízení nebo použitím skutečné doby trvání jednotlivých činností, které jsou známy z předchozích projektů. Také je vhodné založení časové rezervy, která může být vypočtena jednotlivě na činnosti, anebo přidána jedna na celý sprint	Team Leader
14	nedostatečná komunikace se zákazníky	35	přenesení	pro komunikaci mezi zákazníkem a projektovým týmem bude stanovena odpovědná osoba, která bude věnovat jí větší pozornost	Asistent Product Ownera
15	zpoždění designu, obsahu nebo SEO optimalizace	42	přenesení	plánování ve sprint pouze těch úkolů, pro které je schválen design nebo optimalizace	Manažeri příslušných oddělení
16	nedostatečné testování	56	oslabení	vytvoření časové rezervy pro testování, spuštění automatických testů ke kontrole kompatibility nových a aktuálních funkcí, příprava přesného checklistu pro testování, v kritických situacích spolupráce s testery z jiných vývojových týmů	Team Leader
17	příliš krátký termín dokončení sprintu	28	oslabení	zařazení do sprintu časové rezervy a rezervy lidských zdrojů (zajištění dostatečného počtu pracovníků)	Scrum master AUTO.RIA
18	malé zkušenosti se Scrum technikami	24	oslabení	důsledné proškolení a ověření skutečnosti, že celý tým pochopil Agilní techniky. Je také vhodné upravit konkrétní techniky, nikdy ovšem nesmí ztratit svou základní vlastnost – flexibilitu	Senior Scrum master

Zdroj: vlastní zpracování