



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

ÚSTAV SOUDNÍHO INŽENÝRSTVÍ

INSTITUTE OF FORENSIC ENGINEERING

ODBOR INŽENÝRSTVÍ RIZIK

DEPARTMENT OF RISK ENGINEERING

PODPORA MANAGEMENTU RIZIK V AGILNÍM VÝVOJI PRODUKTU

SUPPORT OF RISK MANAGEMENT IN AGILE PRODUCT DEVELOPMENT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Zdeněk Vafek

VEDOUCÍ PRÁCE

SUPERVISOR

doc. RNDr. Jitka Kreslíková, CSc.

BRNO 2022

Zadání diplomové práce

Student: **Bc. Zdeněk Vafek**
Studijní program: Řízení rizik technických a ekonomických systémů
Studijní obor: Řízení rizik technických systémů
Vedoucí práce: **doc. RNDr. Jitka Kreslíková, CSc.**
Akademický rok: 2021/22
Ústav/odbor: Odbor inženýrství rizik

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Podpora managementu rizik v agilním vývoji produktu

Stručná charakteristika problematiky úkolu:

Seznámit se se znalostními oblastmi managementu projektů dle aktuálního standardu PMI. Zaměřit se na procesy znalostní oblasti managementu rizik. Seznámit se s principy a metodami agilního řízení projektů. V jednotlivých znalostních oblastech identifikovat vhodné formy uplatnění agilního přístupu. Zaměřit se na jejich možné kombinace pro řízení rizik v projektech vývoje softwarových produktů. Seznámit se s oblastmi testování softwaru. Specifikovat požadavky a po konzultaci s vedoucí navrhnout, implementovat a otestovat programovou aplikaci pro podporu použití agilního přístupu k managementu rizik v projektech. Zhodnotit dosažené výsledky a diskutovat možnosti dalšího rozšíření vytvořené programové aplikace.

Cíle diplomové práce:

Vytvořit programovou aplikaci pro podporu použití agilního přístupu k managementu rizik v projektech vývoje softwarových produktů.

Seznam literatury:

A Guide To The Project Management Body Of Knowledge: Sixth Edition, Project Management Institute, 2017. 756 s., ISBN 978-1-628251-84-5.

Agile Practice Guide: global standard Project Management Institute, 2017. 167 s., ISBN 978-1-62825-199-9.

VINAY, P., F. Manage Software Testing, Auerbach Publications, 2008. 573 s., ISBN 978-0-849-39383-9.

SCHWALBE Kathy. Řízení projektů v IT: Computer Press, a.s., 2007, 720 s., ISBN 978-80-251-1526-8.

SMEJKAL, Vladimír a Karel RAIS. Řízení rizik ve firmách a jiných organizacích. 4., aktualiz. a rozš. vyd. Praha: Grada, Expert (Grada). 2013. 488 s., ISBN 978-80-247-4644-9.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

prof. Ing. Karel Pospíšil, Ph.D., LL.M.
ředitel

Abstrakt

Diplomová práce se zabývá oblastí projektového managementu v agilním vývoji produktu dle standardu PMI (Project Management Institute) na základě metodických příruček. V rámci agilního vývoje je zde brán důraz rovněž na metody testování produktu a jejich spojení s agilním prostředím. Na základě sesbíraných poznatků je zde navržena programová aplikace pro podporu managementu rizik v agilním vývoji produktu.

Abstract

The diploma thesis deals with the area of project management in agile product development according to the PMI (Project Management Institute) standard on the basis of methodological manuals. As part of agile development, emphasis is also placed on product testing software methods and their connection with the agile environment. Based on the collected knowledge, a software application is designed to support risk management in agile product development.

Klíčová slova

Agilní vývoj, testování, aplikace, management, web, Python, Django

Keywords (example)

Agile development, testing, application, management, web, Python, Django

Bibliografická citace

VAFEK, Zdeněk. *Podpora managementu rizik v agilním vývoji produktu*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/135673>. Diplomová práce. Vysoké učení technické v Brně, Ústav soudního inženýrství, Odbor inženýrství rizik. Vedoucí práce Jitka Kreslíková.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Podpora managementu rizik v agilním vývoji produktu jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom/a následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně

.....

Podpis autora

Poděkování

Na tomto místě práce bych rád poděkoval vedoucí mé závěrečné práce doc. RNDr. Jitce Kreslíkové CSc. za poskytování cenných rad, čas strávený konzultacemi a kontrolami. Poděkování rovněž patří mé rodině za podporu během mých studií a přítelkyni za její nekonečnou trpělivost a stylistickou kontrolu práce.

OBSAH

OBSAH	13
1 ÚVOD	17
2 ÚVOD DO PROJEKTOVÉHO MANAGEMENTU	18
2.1 Projekt	18
2.2 Projektové řízení.....	18
2.3 RIPRAN	19
3 ŘÍZENÍ PROJEKTU DLE PROJECT MANAGEMENT INSTITUTE (PMI)	20
3.1 Řízení integrace projektu	21
3.1.1 Vytvoření zadávací listiny projektu (charta projektu).....	21
3.1.2 Vytvoření plánu řízení projektu.....	21
3.1.3 Realizace plánu projektu	22
3.1.4 Řízení projektových znalostí.....	22
3.1.5 Monitorování a kontrola projektových prací.....	23
3.1.6 Celková koordinace změn.....	23
3.1.7 Uzavření projektu nebo fáze	24
3.2 Řízení rozsahu prací projektu	24
3.2.1 Plánování řízení rozsahu	25
3.2.2 Sběr požadavků	25
3.2.3 Definování rozsahu	25
3.2.4 Vytváření Work Breakdown Structure (WBS).....	26
3.2.5 Ověřování rozsahu	26
3.2.6 Kontrola rozsahu	27
3.3 Řízení časového plánu v rámci projektu	27
3.3.1 Plánování řízení časového rozvrhu	27
3.3.2 Definování činností.....	28
3.3.3 Řazení činností	28
3.3.4 Odhadování doby trvání činností.....	29
3.3.5 Tvorba časového rozvrhu	29
3.3.6 Kontrola časového rozvrhu	30
3.4 Řízení nákladů projektu	30
3.4.1 Plánování řízení nákladů.....	30
3.4.2 Odhadování nákladů	31
3.4.3 Sestavení rozpočtu.....	31

3.4.4	<i>Kontrola nákladů</i>	32
3.5	Řízení kvality v rámci projektu	32
3.5.1	<i>Plánování řízení kvality</i>	32
3.5.2	<i>Zabezpečování kvality</i>	33
3.5.3	<i>Kontrola kvality</i>	33
3.6	Řízení zdrojů v rámci projektu	34
3.6.1	<i>Plánování řízení zdrojů</i>	34
3.6.2	<i>Odhad množství zdrojů</i>	35
3.6.3	<i>Získávání zdrojů</i>	35
3.6.4	<i>Vytvoření projektového týmu</i>	36
3.6.5	<i>Vedení týmu</i>	37
3.6.6	<i>Kontrola zdrojů</i>	37
3.7	Řízení komunikace v rámci projektu	38
3.7.1	<i>Plánování řízení komunikace</i>	38
3.7.2	<i>Řízení komunikace</i>	38
3.7.3	<i>Kontrola komunikace</i>	39
3.8	Řízení rizik projektu	39
3.8.1	<i>Plánování řízení rizik</i>	40
3.8.2	<i>Identifikace rizik</i>	41
3.8.3	<i>Kvalitativní analýzy rizik</i>	41
3.8.4	<i>Kvantitativní analýza rizik</i>	42
3.8.5	<i>Plánování reakcí na rizika</i>	42
3.8.6	<i>Reakce na rizika</i>	43
3.8.7	<i>Sledování rizik</i>	43
3.9	Řízení obstarávání v rámci projektu	44
3.9.1	<i>Plánování obstarávání</i>	44
3.9.2	<i>Poptávání</i>	45
3.9.3	<i>Kontrola obstarávání</i>	45
3.10	Řízení zainteresovaných stran	46
3.10.1	<i>Identifikace zainteresovaných stran</i>	46
3.10.2	<i>Plánování zapojení zainteresovaných stran</i>	47
3.10.3	<i>Řízení zapojení zainteresovaných stran</i>	47
3.10.4	<i>Kontrola zapojení zainteresovaných stran</i>	48
4	AGILNÍ VÝVOJ	49
4.1	Principy agilních metod	49

4.2	Cyklus agilního vývoje produktu.....	50
4.2.1	<i>Koncept</i>	50
4.2.2	<i>Zahájení</i>	50
4.2.3	<i>Iterace</i>	51
4.2.4	<i>Uvolnění</i>	51
4.2.5	<i>Produkce</i>	51
4.2.6	<i>Stažení/nahrazení produktu</i>	52
4.3	Metody agilního řízení.....	52
4.3.1	<i>Scrum</i>	52
4.3.2	<i>Lean</i>	53
4.3.3	<i>Kanban</i>	54
4.3.4	<i>Extrémní programování (XP)</i>	55
4.3.5	<i>Crystal</i>	55
4.3.6	<i>Scrumban</i>	56
5	TESTOVÁNÍ SOFTWARE	57
5.1	Životní cyklus testování softwaru	58
5.2	Testovací metody.....	59
5.2.1	<i>Metoda White Box</i>	59
5.2.2	<i>Metoda Black Box</i>	59
5.2.3	<i>Metoda Grey Box</i>	59
5.3	Zlepšování testovacích procesů	59
5.3.1	<i>Automatizace testů</i>	60
6	NÁVRH APLIKACE	61
6.1	Uživatelské role.....	61
6.2	Požadavky na uživatelské rozhraní	62
7	IMPLEMENTACE APLIKACE	64
7.1	Programová část.....	64
7.2	Databáze	65
7.3	Uživatelské rozhraní	65
7.3.1	<i>Projects (Projekty)</i>	66
7.3.2	<i>Project details (Detaily projektu)</i>	67
7.3.3	<i>LFA (Logická rámcová matice)</i>	68
7.3.4	<i>Time sheet (Časový rozvrh projektu)</i>	68
7.3.5	<i>Tasks (Úkoly)</i>	69
7.3.6	<i>Issues (Problémy)</i>	69

7.3.7	<i>Risk register (Registr rizik)</i>	70
7.3.8	<i>Company members (Zaměstnanci společnosti)</i>	71
7.3.9	<i>Statistiky</i>	71
7.3.10	<i>Admin page (Admínská stránka)</i>	72
8	VYHODNOCENÍ APLIKACE	73
8.1	Test výkonu aplikace	73
8.2	Automatizované testy	74
8.3	Uživatelské testování	74
8.4	Rozšíření	75
8.4.1	<i>Připojení na verzovací systém</i>	75
8.4.2	<i>Přidání cloudového úložiště</i>	75
8.4.3	<i>Přidání metod pro analýzu rizik</i>	75
8.4.4	<i>Zapojení umělé inteligence</i>	75
9	ZÁVĚR	76
	SEZNAM POUŽITÝCH ZDROJŮ	77
	SEZNAM TABULEK	79
	SEZNAM GRAFŮ	79
	SEZNAM OBRÁZKŮ	79
	SEZNAM ZKRATEK	80
	SEZNAM PŘÍLOH	80
	PŘÍLOHA Č. 1: MODEL DATABÁZE	81

1 ÚVOD

Práce se zabývá znalostní oblastí managementu projektů dle standartu PMI (Project Management Institute), konkrétně na procesy v oblasti managementu rizik a následným seznámením s agilními metodami vývoje softwaru. Zmíněné oblasti jsou nutné k definování podmínek pro vytvoření programové aplikace pro podporu použití agilního přístupu k managementu rizik v projektech vývoje softwarových produktů.

V první kapitole je popsán úvod do projektového managementu, kde je vysvětlen pojem projekt dle standardu ISO 10006 [1] včetně definice a jeho omezujících kritérií. Druhá část se zaměřuje na vysvětlení pojmu projektové řízení. Závěr kapitoly se věnuje analytické metodě RIPRAN (RIsk PRoject ANalysis) použité pro vyhodnocování rizik v rámci webové aplikace.

Následující kapitola je rešerše standartu PMI dle příručky A Guide To The Project Management Body Of Knowledge, definicí jejich jednotlivých znalostních oblastí a definování jejich procesů.

Agilní metody, jakožto metody vývoje softwaru, které rychle reagují na změnu požadavků a jsou založené na iterativním, nebo inkrementálním vývoji, jsou popsány v kapitole 4 Agilní vývoj. Přítomnost těchto metod se postupně rozrůstá i mimo oblasti softwarového vývoje.

Kapitola 5 Testování softwaru se zabývá rešerší různých technik testování v programů, které napomáhají udržení kvality produktu. Kapitola se nejvíce zaměřuje na automatizované testování, které začíná být v momentální době stále více využíváno kvůli úspoře financí.

Následující kapitoly obsahují provedení návrhu a realizace webové aplikace pro podporu managementu rizik v agilním vývoji. Ve zmíněných kapitolách se rovněž nachází postup, jak s aplikací pracovat a také návrhy na její vylepšení.

2 ÚVOD DO PROJEKTOVÉHO MANAGEMENTU

Projektový management označuje způsob řízení zdrojů za účelem realizace daného cíle – projektu. Projekt je definován jako časově ohraničené úsilí vynaložené k vytvoření jedinečného produktu nebo služby. Pro řízení projektů se uplatňují standardy jako je Project Management Institute (PMI), International Project Management Association (IPMA) a PRojects IN Controlled Environments (Prince2).

2.1 PROJEKT

Projekt je dle standardu ISO 10006 [1] chápán jako sít činností, které mají definovaný formální začátek a konec. Dále obsahuje předem přidělené zdroje směřující k vytvoření určitého produktu. Projekt má předem stanovený rozpočet, v rámci něj má být dosaženo určitých výsledků. S realizací projektu je vždy spojeno určité riziko, které je v projektovém managementu definováno jako nejistá událost, nebo podmínka, která, pokud nastane, má negativní, nebo pozitivní vliv na dosažení cíle projektu. Každý projekt je omezen následujícími kritérii:

- projekt je unikátní dílo, odlišné v nějakém ohledu od podobných,
- aktivity spojené s projektem se týkají více než dvou osob,
- aktivity vyžadují více než dva týdny úsilí osob,
- budou vyžadovány nové postupy, nebo nové technologie,
- rozpočet je těsný,
- některé úkoly jsou vázány na dokončení jiných úkolů,
- vývoj produktu se skládá z několika fází, které musí být koordinovány,
- je vyžadováno měření kvality.

2.2 PROJEKTOVÉ ŘÍZENÍ

Jedná se o využití dovedností, nástrojů a technik v projektových činnostech se snahou splnit, případně překročit očekávání zainteresovaných stran od projektu. Projektové řízení musí vždy vyrovnávat proti sobě požadavky mezi:

- rozsahem prací, časem, náklady a kvalitou,
- zájmovými skupinami s různými potřebami a očekáváním,
- naplánovanými a neplánovanými potřebami.

Takto řízený projekt však musí vždy obsahovat plánování, organizování, sledování a řízení částí projektu po celou dobu jeho existence. V případě nevhodného, či žádného řízení, je projekt vystaven rizikům:

- není přesně definované, co je výsledkem projektu,
- zapomenutí na část projektu,
- nedodržení termínu odevzdání,
- překročení rozpočtu,
- není možné včas identifikovat ohrožení termínu, rozpočtu, kvality, nebo rozsahu výsledného produktu.

2.3 RIPRAN

Autorem metody RIPRAN [2] je doc. Ing. Branislav Lacko CSc. z Fakulty strojního inženýrství VUT. První verze metody byla využívána pro analýzu rizik při vývoji automatizačních systémů. V dnešní době se využívá třetí verze metody, která splňuje požadavky normy ČSN EN 62 198 Management rizik projektu. Návrh metody byl rovněž proveden tak, aby splňoval standardy IPMA a PMI. Metoda slouží ke zpracování analýzy rizik projektu, kde je prováděna před samotnou realizací projektu a v průběhu jeho zpracování. Metoda je rozdělena do šesti následujících fází:

- příprava analýzy rizik projektu,
- identifikace rizik projektu,
- kvantifikace rizik projektu,
- návrh opatření snižující, nebo eliminující vliv rizik na projekt,
- celkové zhodnocení rizikovosti projektu,
- sledování a vyhodnocování rizik projektu.

Metoda využívá pro zápis hodnot pravděpodobnosti a důsledků rizika tabulkovou metodu, která je velmi přehledná a informativní i bez základní znalosti analýzy rizik.

Informace o metodě RIPRAN™

Metoda RIPRAN™ (Risk PROject ANalysis) je určena zejména pro analýzu projektových rizik. Autorem metody je Branislav Lacko. Metoda vznikla původně pro analýzu rizik automatizačních projektů. Praxe ukázala, že metodu je možno aplikovat pro analýzu rizik širokého spektra různých projektů a v určitých případech i pro analýzu jiných druhů rizik, než jsou projektová rizika. RIPRAN™ je ochranná známka, registrovaná Úřadem průmyslového vlastnictví Praha pod registračním číslem 283536. [2]

3 ŘÍZENÍ PROJEKTU DLE PROJECT MANAGEMENT INSTITUTE (PMI)

Project Management Institute je nezisková organizace z United States of America (USA) zabývající se zkoumáním metod pro projektový management s cílem dosažení co nejefektivnějších výsledků. Výsledkem výzkumu společnosti je metodologická příručka A Guide To The Project Management Body Of Knowledge [3], o kterou se opírá následující kapitola. Příručka obsahuje souhrn postupů, používaných procesů a metod z praxe světových odborníků z projektového prostředí. Zmíněná kniha je aktualizována přibližně každé tři roky tak, aby odpovídala nejnovějším nárokům na potřeby projektových manažerů působících především ve společnostech zabývajících se informačními technologiemi (IT). Vzhledem k původu PMI je především využívána ve společnostech sídlících převážně v Severní Americe.

Základní přístup PMI spočívá v procesním pojetí problematiky projektového řízení, kde je definováno pět hlavních procesů a deset oblastí znalostí.

Znalostní oblasti dle PMI:

- řízení integrace projektu,
- řízení rozsahu projektu,
- řízení plánování projektu,
- řízení ceny projektu,
- řízení jakosti projektu,
- řízení zdrojů projektu,
- řízení komunikace v rámci projektu,
- řízení rizik projektu,
- řízení obstarávání v rámci projektu,
- řízení zainteresovaných stran projektu.

Zmíněné znalostní oblasti se mohou skládat ze skupin procesů, obsahující předem definované vstupy, výstupy a metody (techniky).

Definované skupiny procesů dle PMI:

- iniciační,
- plánovací,
- realizační,
- monitorovací a ovládací,

- ukončovací.

PMI klade důraz především na plánovací skupinu procesů, kde je definováno celkem 24 procesů z celkem 47, které jsou rozděleny mezi jednotlivé skupiny.

3.1 ŘÍZENÍ INTEGRACE PROJEKTU

Řízení integrace projektu se zabývá popisem procesů a aktivit pro správnou koordinaci jednotlivých částí projektu. Tímto způsobem zasahuje sedmi procesy do všech skupin procesů projektu.

3.1.1 Vytvoření zadávací listiny projektu (charta projektu)

Zakládací listina slouží k formálnímu vzniku projektu. Jejím obsahem jsou počáteční požadavky na produkt dle zainteresovaných stran. Dále je zde určen projektový manažer, který po podpisu dokumentu získává právo nakládat se zdroji na aktivity spojené s projektem.

Vstupy:

- smlouvy,
- obchodní dokumenty,
- Organizační procesní aktiva (OPA)
- faktory podnikového prostředí.

Výstupy:

- zakládací listina projektu,
- protokol strategických cílů a omezení.

3.1.2 Vytvoření plánu řízení projektu

Plánování jednotlivých částí projektu ze všech procesů a jejich následné spojení. Úkolem je vypracování souboru obsahujícího podrobný seznam činností v časovém sledu celého životního cyklu projektu.

Vstupy:

- zakládající listina projektu,
- výstupy plánování procesů z jiných znalostních oblastí,
- OPA,
- obchodní dokumenty.

Výstupy:

- plán řízení rizik.

3.1.3 Realizace plánu projektu

Proces obsahující realizaci jednotlivých částí projektu dle plánu řízení projektu. Jedná se o realizaci jednotlivých činností a implementaci schválených změn k dosažení cílů projektu. Zmíněný proces je vykonáván po celý životní cyklus projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- seznam schválených požadavků na změny,
- obchodní dokumenty,
- OPA.

Výstupy:

- projektové dokumenty (ukončení fáze, či etapy projektu, ukončení projektu),
- dokumentace metrik spojených s projektem,
- seznam požadavků na změny,
- seznam problémů a nedostatků,
- úprava plánu řízení projektu,
- úprava projektových dokumentů,
- úprava OPA.

3.1.4 Řízení projektových znalostí

Proces sloužící ke sbírání znalostí z předchozích projektů tak, aby mohly být využity ke zefektivnění dosažení cílů aktuálního projektu. Využívají se zde jednotlivé know-how členů týmu. Zmíněný proces je využíván po celou dobu životního cyklu projektu.

Vstupy:

- plán řízení projektu,
- cíle projektu,
- znalosti členů týmu,
- obchodní dokumenty,
- OPA.

Výstupy:

- úprava plánu řízení projektu,
- úprava OPA.

3.1.5 Monitorování a kontrola projektových prací

Jedná se o proces, který je prováděn po celou dobu životního cyklu projektu. Jeho úkolem je sledování, posuzování a kontrola postupu tak, aby odpovídal plánu řízení projektu. Daný proces má za úkol projektový manažer.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- metriky,
- seznam požadavků na změny,
- obchodní dokumenty,
- OPA.

Výstupy:

- změna plánu řízení projektu,
- změna projektových dokumentů,
- seznam přijatých změn projektu.

3.1.6 Celková koordinace změn

Proces posouzení požadavků na změny v rámci projektu. Jedná se o proces schvalování změn a jejich řízení ve vztahu k jednotlivým částem projektu. Celkovou koordinaci změn schvaluje projektový manažer na základě dokumentů vyplývajících z předchozí projektové dokumentace.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- seznam požadavků na změnu,
- obchodní dokumenty,
- OPA.

Výstupy:

- seznam přijatých požadavků na změnu,
- úprava plánu řízení projektu,
- úprava projektových dokumentů.

3.1.7 Uzavření projektu nebo fáze

Formální ukončení všech aktivit spojených s daným projektem, nebo jeho fází. Vyhodnocení úspěšnosti dokončení na základě měřených metrik. V případě neúspěšného ukončení jsou dokumenty přezkoumány a je nutné vyhodnotit důvody, které vedly k neúspěšnému konci.

Vstupy:

- zadávající listina,
- plán řízení projektu,
- projektové dokumenty,
- obchodní dokumenty,
- OPA.

Výstupy:

- finální produkt,
- zpráva zhodnocení projektu,
- úprava projektové dokumentace,
- úprava OPA.

3.2 ŘÍZENÍ ROZSAHU PRACÍ PROJEKTU

Řízení rozsahu projektu slouží k jasnému definování a kontrole činností zahrnutých do projektu definovaných pomocí šesti hlavních činností. Zmíněná oblast zajišťuje, že všechny zainteresované strany budou jasně chápat jednotlivé produkty vytvořené v rámci projektu. Zde je rozsah myšlen jako rozsah produktu, nebo rozsah prací projektu.

Rozsah produktu definuje vlastnosti a funkce zahrnuté do produktu. Jeho dokončení je následně kontrolováno dle specifických požadavků na výsledný produkt.

Rozsah prací projektu obsahuje jednotlivé činnosti, které je nutné vykonat při realizaci projektu. Jejich splnění je kontrolováno dle plánu projektu.

3.2.1 Plánování řízení rozsahu

Proces určený pro tvorbu plánu řízení projektu. Slouží k definování, potvrzování a kontrolování rozsahu projektu.

Vstupy:

- plán řízení projektu,
- zakládající listina projektu,
- obchodní dokumenty,
- OPA.

Výstupy:

- plán řízení rozsahu projektu,
- plán řízení požadavků.

3.2.2 Sběr požadavků

Slouží k jasné definici rozsahu produktu, ke kterému dochází pomocí nástrojů jako jsou rozhovory se zákazníkem a dotazníky. Jedná se o dokumentaci požadavků zainteresovaných stran potřebných k dosažení cílů projektu.

Vstupy:

- zakládací listina,
- projektové dokumenty,
- obchodní dokumenty,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- dokumentace požadavků,
- matice stopovatelnosti požadavků.

3.2.3 Definování rozsahu

Definování rozsahu projektu vychází ze sběru požadavků, na jejichž základě se vytváří detailnější popis produktu a projektu.

Vstupy:

- zakládací listina projektu,
- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- stanovení rozsahu projektu,
- úpravy dokumentů.

3.2.4 Vytváření Work Breakdown Structure (WBS)

Jedná se o hierarchickou strukturu prací na projektu. Pro tvorbu WBS je nutné přesně znát jednotlivé úkony nutné k dosažení jasného cíle projektu a ovládat problematiku s ním spojenou. Projekt je zde rozdělen na menší, lépe zvládnutelné cíle tak, aby bylo možné lépe řídit projekt.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- shrnutí rozsahu projektu,
- úpravy dokumentů.

3.2.5 Ověřování rozsahu

Jedná se o formální schválení rozsahu projektu od všech zainteresovaných stran. Ověřování je vhodné vykonávat po celou dobu životního cyklu projektu tak, aby výsledný produkt odpovídal požadavkům zákazníka.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- ověřené cíle a výstupy,

- údaje o výkonu práce.

Výstupy:

- přijaté cíle,
- informace o výkonu práce,
- seznam požadavků na změnu,
- úprava dokumentů.

3.2.6 Kontrola rozsahu

Proces průběžně kontrolující rozsah produktu a projektu sloužící k tomu, aby nedošlo k narušení integrace projektu. Má za úkol koordinovat změny tak, aby nedošlo k překročení výchozí hodnoty rozsahu projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- informace o výkonu práce,
- seznam požadavků na změnu,
- upravený plán řízení projektu,
- upravené příslušné dokumenty.

3.3 ŘÍZENÍ ČASOVÉHO PLÁNU V RÁMCI PROJEKTU

Znalostní oblast sloužící k sestavení časového plánu projektu a jeho řízení z časového pohledu. Hlavní důvod využívání časových plánů je získání přehledu o času dokončení projektu pro všechny zainteresované strany. Časový plán lze také využít pro řízení projektu za využití kontroly časových milníků jednotlivých částí projektu a případně odhalit časové zpoždění dříve, než bude mít negativní vliv na dokončení projektu.

3.3.1 Plánování řízení časového rozvrhu

Slouží k určení časového plánu projektu. Proces definuje konkrétní postupy a dokumentaci pro plánování a řízení časového rozvrhu projektu a jeho kontrolu.

Vstupy:

- zakládací listina projektu (včetně časových milníků),
- plán řízení projektu,
- OPA.

Výstup:

- plán řízení časového rozvrhu projektu.

3.3.2 Definování činností

Proces pro určení a zdokumentování jednotlivých aktivit na projektu pro vytvoření různých výstupů projektu. Pro definování těchto aktivit se využívá WBS.

Vstupy:

- plán řízení projektu (včetně WBS),
- OPA.

Výstupy:

- seznam projektových činností,
- časové milníky projektu,
- atributy činností rozšiřující předcházející výstupy,
- požadavky na změny z analýzy projektu,
- úprava plánu řízení projektu.

3.3.3 Řazení činností

Proces sloužící k identifikaci a dokumentaci jednotlivých vzájemných závislostí mezi jednotlivými činnostmi a tím jednoznačnému určení časové souslednosti jednotlivých úkonů. Proces je vykonáván po celou dobu životního cyklu projektu.

Vstupy:

- plán řízení projektu,
- plán řízení časového rozvrhu,
- projektové dokumenty (popis jednotlivých aktivit a časové milníky),
- OPA.

Výstupy:

- síťový graf projektu,

- úprava projektových dokumentů týkající se jednotlivých aktivit a časových milníků.

3.3.4 Odhadování doby trvání činností

Proces k určení celkové doby trvání jednotlivých činností. K získání daných informací je nutné vyhodnotit počet dostupných zdrojů, míru zkušenosti členů týmu s danou problematikou a týmovou motivací. K získané odhadované hodnotě času bývá přidávána časová rezerva.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- OPA.

Výstupy:

- odhad doby trvání jednotlivých činností,
- podklady k časovým odhadům,
- úprava projektových dokumentů.

3.3.5 Tvorba časového rozvrhu

Proces pro analýzu pořadí jednotlivých činností projektu, dostupných zdrojů a časových omezení. Činnosti nesou informace o odhadovaném datu začátku a startu činnosti. Pro vytvoření časového rozvrhu se využívá metoda kritické cesty (CPM), která nám dává informaci o kritické (časově nejdelší) cestě v časovém rozvrhu s nulovou časovou rezervou. Dle časového rozvrhu je možné monitorovat práce na projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- OPA.

Výstupy:

- směrný plán časového rozvrhu,
- harmonogram projektu,
- informace k rozvrhu,
- projektové kalendáře identifikující skutečnou dobu práce,
- požadavky na změny v řízení projektu,

- úprava plánu řízení projektu,
- úprava projektových dokumentů zabývajících se aktivitami a riziky projektu.

3.3.6 Kontrola časového rozvrhu

Proces pro monitorování stavu projektu a jeho jednotlivých činností vzhledem k časovým milníkům. V případě nesrovnalosti je nutné upravit časový rozvrh.

Vstupy:

- plán řízení projektu (část týkající se rozsahu a řízení času),
- projektové dokumenty (časový rozvrh a registr znalostí),
- údaje o výkonu práce s informacemi o započatých a ukončených aktivitách,
- OPA.

Výstupy:

- informace o stavu projektu vzhledem ke směrnému plánu projektu,
- odhad časového rozvrhu v závislosti na aktuální situaci,
- požadavky na změny časového plánu,
- úprava plánu řízení projektu,
- úprava projektových dokumentů.

3.4 ŘÍZENÍ NÁKLADŮ PROJEKTU

Řízení nákladů projektu se skládá z procesů, které zajišťují kontrolu nad finální cenou projektu tak, aby nepřekročil schválený rozpočet.

3.4.1 Plánování řízení nákladů

Proces slouží pro určení způsobu odhadování, řízení a kontrolování jednotlivých nákladů spojených s projektem.

Vstupy:

- zakládací listina,
- plán řízení projektu,
- OPA.

Výstupy:

- plán řízení nákladů.

3.4.2 Odhadování nákladů

Proces sloužící ke tvorbě hrubého odhadu ceny výsledného produktu na základě aktuálních informací o nutných zdrojích projektu. Je nutné zvážit také, zda je nutné zdroje pořizovat, nebo je možné je pronajmout. Zhodnocení nákladů spjatých s projektem je vhodné provádět opakovaně, protože čím více se blíží konec životního cyklu projektu, tím přesnější můžeme cenu odhadnout.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- odhad celkové ceny včetně finanční rezervy,
- seznam jednotlivých naceněných činností,
- úprava projektových dokumentů.

3.4.3 Sestavení rozpočtu

V procesu dochází k sečtení nákladů projektových činností, čímž dojde k vytvoření směrného plánu rozpočtu. Dle rozpočtu je následně kontrolována cena projektu v jednotlivých fázích a na případnou změnu je nutné včas reagovat.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- obchodní dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- směrný plán rozpočtu,
- požadavky na financování,
- úpravy projektových dokumentů.

3.4.4 Kontrola nákladů

Proces sloužící ke kontrole, zda náklady nepřevyšují rozpočet projektu. Tuto činnost je nutné vykonávat neustále během projektu, aby se zabránilo možným finančním ztrátám. V případě překročení rozpočtu je nutná změna rozpočtu, která však musí být schválena i zákazníkem.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- požadavky na financování projektu,
- data o výkonnosti,
- OPA.

Výstupy:

- informace o výkonu práce,
- odhad nákladů,
- seznam požadavků na změny,
- úprava plánu řízení projektu,
- úprava projektových dokumentů.

3.5 ŘÍZENÍ KVALITY V RÁMCI PROJEKTU

Znalostní oblast sloužící k plánování, řízení a kontrolování kvality projektu. Jakost projektu je dána dostupnými finančními prostředky, časem, dostupnými zdroji a měla by se řídit především požadavky všech zainteresovaných stran na produkt. Zda je výsledná kvalita projektu odpovídající požadavkům však určuje až zákazník. Pokud není spokojený s produktem, mohou být negativně ovlivněny náklady na projekt.

3.5.1 Plánování řízení kvality

Proces sloužící k identifikaci jednotlivých požadavků na projekt a případných standardů s ním spojených. Dále je definována dokumentace postupů sloužící ke kontrole kvality projektu a dodržení všech požadavků v průběhu projektu.

Vstupy:

- základní listina,
- plán řízení projektu,

- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení kvality,
- metriky kvality,
- úprava projektových dokumentů,
- úprava plánu řízení projektu.

3.5.2 Zabezpečování kvality

Jedná se o proces, který je aplikován po celou dobu životního cyklu projektu. Vychází z procesu plánování řízení kvality a má za úkol dohlížet nad dodržováním všech standardů a tím se přiblížit ke splnění kvalitativní požadavků na výsledný produkt.

Vstupy:

- plán řízení projektu,
- plán řízení kvality,
- projektové dokumenty,
- OPA.

Výstupy:

- zpráva o kvalitě,
- dokumentace testů a jejich vyhodnocení,
- požadavky na změnu plánu,
- úprava řízení projektu,
- úprava registru rizik a nabytých znalostí.

3.5.3 Kontrola kvality

Proces sledování a zaznamenávání jednotlivých výstupů z aktivit souvisejících s projektem. Na základě kontroly kvality je zabezpečena celková kvalita projektu tak, aby odpovídala požadavkům zákazníka. V případě nesouladu je možné dle plánu určovat postupy odstraňování příčin nevyhovujících úkonů.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- schválené požadavky na změnu,
- seznam cílů,
- data o výkonu práce,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- dokumenty výsledků měření kvality,
- ověřené cíle,
- informace o výkonu práce na projektu,
- seznam požadavků na změny,
- úprava plánu řízení projektu,
- úprava projektových dokumentů.

3.6 ŘÍZENÍ ZDROJŮ V RÁMCI PROJEKTU

Řízení zdrojů obsahuje celkem šest procesů popisujících identifikaci, rezervaci a řízení lidských zdrojů pro úspěšné dokončení projektu. Jedná se především o zajištění, že konkrétní zdroje budou k dispozici na správném místě v momentě, kdy jsou potřeba. Zdroje v projektovém smýšlení lze dělit na fyzické a lidské. Lidské zdroje reprezentují členové týmu, které musí manažer správně řídit tak, aby neztráceli motivaci k plnění projektových činností a jejich činnost byla efektivně využita. Fyzické zdroje jsou reprezentovány potřebným vybavením, jako jsou například programy a přístroje.

3.6.1 Plánování řízení zdrojů

Jedná se o proces plánování s úkolem nalezení způsobů, jak odhadnout, získat, řídit a používat jednotlivé zdroje. Při plánování je nutné uvažovat o možnosti, že všechny dané zdroje mohou být v daném čase k dispozici. Pro jejich řízení je vhodné využívat matici přiřazení zodpovědnosti RACI (Responsible, Accountable, Consulted, Informed). Jedná se o seznam osob zapojených do projektu a u každé činnosti je jim přiřazena role.

Vstupy:

- zakládací listina projektu,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení zdrojů,
- úpravy projektových dokumentů,
- zakládací listina týmu.

3.6.2 Odhad množství zdrojů

Proces určující počet jednotlivých zdrojů, které jsou nutné pro úspěšné dokončení projektu. Je nutné jej provádět paralelně s ostatními procesy plánování.

Vstupy:

- projektové dokumenty,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- seznam požadavků na zdroje,
- seznam informací o odhadech zdrojů,
- hierarchická struktura zdrojů (WBS resource breakdown structure – hierarchické rozdělení prací),
- úprava projektových dokumentů.

3.6.3 Získávání zdrojů

Proces určený pro alokaci zdrojů potřebných pro projekt. Zdroje mohou být získány interně, nebo externě (pro ty je nutné využít obstarávací procesy). Zdroje jsou následně přiřazeny k daným projektovým činnostem. Při neúspěšné, nebo nevhodné alokaci zdrojů je ohroženo úspěšné dokončení projektu.

Vstupy:

- plán řízení zdrojů,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- přidělení lidských zdrojů,
- přidělení fyzických zdrojů,
- kalendář zdrojů,
- požadavky na změny projektových dokumentů,
- úprava OPA.

3.6.4 Vytvoření projektového týmu

Proces pro zdokonalování projektového týmu, který probíhá po celou dobu životního cyklu projektu. Nejdůležitější roli v tomto procesu hraje projektový manažer, který má za úkol vytvořit, motivovat, vést a inspirovat členy týmu. Kvalita výsledného produktu je pak značně ovlivněna měkkými dovednostmi manažera.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- hodnocení výkonnosti týmu,
- požadavky na změny,
- úprava plánu řízení projektu,
- úprava projektových dokumentů,
- úprava faktorů projektového prostředí,
- úprava OPA.

3.6.5 Vedení týmu

Jedná se o proces, který opět probíhá po celou dobu trvání projektu a má jej na starost především projektový manažer. Je zde sledována výkonnost jednotlivých členů týmu, řeší se případné konflikty členů týmu a výsledkem je optimalizace prací na projektu. Zároveň jsou řešeny personální potřeby zdrojů.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- zpráva o pracovním výkonu,
- hodnocení týmu,
- faktory podnikové prostředí,
- OPA.

Výstupy:

- seznam požadavků na změny,
- úprava plánu řízení projektu,
- úprava projektových dokumentů,
- úprava faktorů podnikového prostředí.

3.6.6 Kontrola zdrojů

Proces sloužící k ověřování správné alokace zdrojů a případné zajišťování změn v požadavcích na zdroje. Případně je možné v procesu uvolňovat již nepotřebné fyzické zdroje.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- údaje o výkonnosti,
- smlouvy s externími dodavateli zdrojů,
- OPA.

Výstupy:

- informace o výkonnosti,
- seznam požadavků na změny,
- úprava plánu řízení projektu,

- úprava projektových dokumentů.

3.7 ŘÍZENÍ KOMUNIKACE V RÁMCI PROJEKTU

Řízení komunikace v rámci projektu je znalostní oblast popisující procesy potřebné pro zajištění komunikace nad projektem se všemi zainteresovanými stranami tak, aby informace byly dostupné v čase, kdy jsou potřeba a zároveň byly kompletní. Z toho důvodu je nutné vybudovat komunikační protokol a definovat, jakým způsobem budou informace předávány.

3.7.1 Plánování řízení komunikace

Proces pro vypracování vhodného přístupu a plánu komunikace v závislosti na komunikačních a informačních potřebách zainteresovaných stran projektu. Klade se zde důraz rovněž na to, jakým způsobem budou informace ukládány.

Vstupy:

- zakládací listina projektu,
- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení komunikace,
- úpravy plánu řízení projektu.

3.7.2 Řízení komunikace

Proces slouží k zajištění informačního toku na projektu, tak aby byl plynulý a efektivní. Úkolem je včasné získávání, tvorba, ukládání, sledování a uspořádání informací. Proces je vykonáván po celou dobu životního cyklu projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- zpráva o pracovním výkonu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- popis projektové komunikace,
- dokumenty řízení komunikace,
- úprava plánu řízení projektu,
- úprava projektových dokumentů,
- úprava OPA.

3.7.3 Kontrola komunikace

Proces je vykonáván po celou dobu trvání projektu a zaručuje dostupnost aktuálních informací zainteresovaným stranám. Dále ověřuje, zda se daný plán řízení komunikace dodržuje.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- údaje o výkonu práce,
- faktory podnikové prostředí,
- OPA.

Výstupy:

- informace o výkonu práce,
- požadavky na změny v komunikaci,
- úprava plánu řízení projektu,
- úprava projektových dokumentů.

3.8 ŘÍZENÍ RIZIK PROJEKTU

Samotné riziko na sebe v nějaké formě musela převzít každá společnost, jinak by nebyla schopna vyhledávat nové příležitosti na trhu, případně ani začít s podnikáním, nebo novým projektem. Z projektového hlediska není riziko chápáno pouze jako negativní působení proti projektovým aktivitám, ale jako jakákoliv událost působící na projekt v pozitivním i negativním směru. Z toho důvodu je žádoucí vyhledávat možná rizika před tím, než jakýmkoliv způsobem začnou ovlivňovat samotný projekt. Úkolem řízení rizik projektu je snížit pravděpodobnost a dopad negativních rizik projektu na hodnotu blížící se nule, zatímco pozitivním rizikům zvýšit hodnotu pravděpodobnosti a dopadu na co nejvyšší úroveň.

Řízení rizik v projektu není pouze záležitostí vypořádání se s nově vzniklým rizikem, ale do této oblasti spadá rovněž predikování rizik a následný proces, jak se s danými riziky vypořádat. Predikce rizika se opírá o možnost identifikace nových (v ideálním případě budoucích) rizik a jejich následné analýze, dle které je možné rozhodnout nad jeho akceptací. V závislosti na daném riziku je prováděno rovněž plánování a využívání metod pro řízení samotného rizika a jeho následného monitorování.

Rizika v projektovém managementu je možné rozdělit do dvou skupin dle místa jeho vzniku na vnitřní a vnější riziko. Vnitřní rizika vznikají v souvislosti s činnostmi, na které má projektový tým vliv. Nejčastějšími riziky v dané oblasti je zařazení pracovníků na dané pracovní pozice, případně neshody na pracovišti. Vnější rizika jsou taková, na které projektový tým nemá vliv. Nejčastější příčinou je změna legislativy, nebo nečekaný vývoj trhu.

3.8.1 Plánování řízení rizik

Proces plánování řízení rizik slouží k definování postupu aktivit řízení rizik a jejich plánování. Proces je možné opakovat vícekrát, ale je žádoucí, aby byl ukončen v začátečních fázích projektu. V ostatních fázích projektu by měl být použit pouze v případě větší změny, kdy je možnost, že dosavadní řešení není dostačující.

Vstupy:

- zakládací listina projektu,
- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.
- Při plánování řízení rizik je vhodné aplikovat alespoň jednu z následujících technik:
- expertní posudek,
- datová analýza,
- schůzky týmu se zainteresovanými stranami a zákazníky.

Výstupem procesu je plán řízení rizik projektu, který je částí plánu řízení projektu. Výstup z procesu plánování řízení rizik obsahuje na základě typu projektu následující položky.

Výstupy:

- strategie rizik,
- metody,

- role,
- financování,
- časové rozvržení,
- kategorie rizik,
- tolerance zainteresovaných stran k rizikům,
- stanovení pravděpodobnosti výskytu rizika a jeho důsledků,
- matice pravděpodobnosti a dopadu rizika,
- formát zpráv,
- sledování procesů.

3.8.2 Identifikace rizik

Identifikace rizik je v prostředí dle PMI chápáno jako proces zaměřující se na chápání zdrojů rizik a jejich návaznosti na jednotlivé fáze projektu. Pro identifikaci rizik platí, že je čím dál kvalitnější, čím více odborníků z oboru se na identifikaci podílí. Do skupiny podílejících se na identifikaci rizik obvykle patří zástupci zúčastněných stran, projektový tým a projektový manažer. Zmíněná skupina by měla obsahovat členy z různých oborů tak, aby byla rizika identifikována co nejkomplexněji.

Vstupy:

- plán řízení projektu,
- úprava projektových dokumentů,
- faktory podnikového prostředí,
- OPA,
- dokumentace obstarávání.

Výstupy:

- registr rizik,
- zpráva o rizicích,
- úprava projektových dokumentů.

3.8.3 Kvalitativní analýzy rizik

Proces provedení kvalitativní analýzy slouží v projektovém prostředí k hodnocení dopadu na organizaci, či jedince v případech, kdy není možné stanovit peněžní hodnotu rizika, nebo nejsou známi dostatečné údaje k provedení kvantitativní analýzy. Rovněž je možné využít proces pro identifikaci konkrétních rizik, které potřebují podrobnější přezkoumání, případně v situacích, kdy

je nutné rychlé rozhodnutí. Pro sběr dat je zde využíváno dotazníkových metod, nebo verbálního hodnocení expertů z oboru.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- OPA,
- faktory projektového prostředí.

Výstupy:

- upravené projektové dokumenty.

3.8.4 Kvantitativní analýza rizik

V projektovém prostředí je na proces kvantitativní a semikvantitativní analýzy rizik uplatňován velký důraz, protože je možné převést hodnoty rizik na finance. Na základě údajů z minulosti, nebo modelováním je možné odhadnout pravděpodobnost rizika a jeho dopad na projekt. Na základě zmíněných parametrů je pak možné odhadnout cenu rizika spjatého s projektem.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- OPA,
- faktory podnikového prostředí.

Výstupy:

- upravené projektové dokumenty.

3.8.5 Plánování reakcí na rizika

Proces pro plánování reakcí na rizika se zaměřuje na vymezení konkrétních hodnot pro zařazení jednotlivých rizik na základě jejich pravděpodobnosti a dopadu do skupin, které definují reakci na riziko. Na rizika je možné reagovat jeho snížením (přiblížit se co nejbližší jeho eliminaci), pojištěním (převedení rizika), přijetí rizika s protipatřením a celkové přijetí rizika.

Vstupy:

- plán řízení projektu,

- projektové dokumenty,
- OPA,
- faktory podnikového prostředí.

Výstupy:

- seznam požadavků na změny,
- úprava plánu řízení projektu,
- úprava projektových dokumentů.

	Vysoká pravděpodobnost	Nízká pravděpodobnost
Vysoký dopad	Snížení rizika	Pojištění
Nízký dopad	Přijetí s protipatřením	Přijetí rizika

Tab. č. 1 Obecný přístup k riziku

3.8.6 Reakce na rizika

Proces zajišťující postup reakce na rizika dle procesu plánování reakcí na rizika tak, aby byly zajištěno dodržování dohodnutých postupů.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- OPA,
- faktory podnikového prostředí.

Výstupy:

- seznam požadavků na změny,
- úprava projektových dokumentů.

3.8.7 Sledování rizik

Sledování rizik je kontrolní proces v průběhu celého trvání životního cyklu projektu. Proces má za úkol zajistit monitorování položek z registru rizik a reakcí na jednotlivá rizika. Tímto procesem je možné zvýšit pravděpodobnost úspěšného dokončení samotného projektu. Do procesu sledování rizik rovněž patří identifikace a návrh protipatření na nově vzniklá rizika v průběhu trvání životního cyklu projektu.

Vstupy:

- plán řízení projektu
- projektová dokumentace,
- metriky,
- dokumentace vykonávané práce.

Výstupy:

- seznam požadavků na změnu,
- úprava plánu řízení projektu,
- úprava projektových dokumentů,
- úprava OPA.

3.9 ŘÍZENÍ OBSTARÁVÁNÍ V RÁMCI PROJEKTU

Znalostní oblast sloužící k definování procesů zajišťující obstarávání projektů z hlediska nákupu a nabytí produktů, či služeb. Do zmíněné znalostní oblasti patří především uzavírání dohod a smluv. V oblasti IT sem patří především smlouvy ohledně dohod, objednávek, či zajištění potřebného softwaru. Potřebné smlouvy musí být však schváleny rovněž lidmi s potřebnými zkušenostmi či vzděláním.

3.9.1 Plánování obstarávání

Jedná se o proces, jehož úkolem je správný odhad potřebného vybavení, které je nutné pořídit od externích zdrojů. Součástí je rovněž získání co nejpřesnějšího odhadu způsobu, místa a času využití daného vybavení. Nevhodné naplánování může vést ke zvýšení celkové ceny produktu.

Vstupy:

- zakládací listina projektu,
- obchodní dokumenty,
- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení obstarávání,

- strategie obstarávání,
- cenové nabídky od externích zdrojů,
- výběrová kritéria zdrojů,
- rozhodnutí, zda je potřeba využívat externí zdroj, nebo danou část vyrobit,
- nezávislé odhady cen,
- úprava projektových dokumentů,
- úprava OPA.

3.9.2 Poptávání

Proces sloužící k výběru správného dodavatele na základě důležitých požadavků pro projekt. Úkolem je vybrat správný produkt s ideální kombinací ceny, času a znalosti využívání poptávaného produktu.

Vstupy:

- plán řízení projektu,
- dokumentace požadavků,
- nabídky dodavatelů,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- zvolený dodavatel,
- smlouva s dodavatelem,
- požadavky na změny plánu projektu,
- úpravy plánu řízení projektu,
- úprava projektových dokumentů.

3.9.3 Kontrola obstarávání

Kontrolní proces sloužící k zajištění správného řízení vztahů s dodavateli, sledování vývoje produktu a provádění potřebných změn ve vývoji produktu. Hlavním úkolem procesu je zajištění dodržení smluvních podmínek využívání produktů od externích dodavatelů.

Vstupy:

- plán řízení projektu,
- dokumentace obstarávání,

- smlouvy,
- dokumentace požadavků,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- ukončené smlouvy o obstarávání,
- úprava dokumentace požadavků,
- úprava dokumentace obstarávání,
- požadavky na změny řízení projektu.

3.10 ŘÍZENÍ ZAINTERESOVANÝCH STRAN

Znalostní oblast sloužící k zajištění všech zainteresovaných stran do projektu, čímž jsou myšleny strany mající vliv na produkt, případně ho nějakým způsobem ovlivňující. Následně je zanalyzována míra vlivu jednotlivých stran a navržena strategie, jak se stranami komunikovat a následně jsou vyjasněny požadavky stran na výsledný produkt. Požadavky se však mohou v průběhu vývoje měnit, proto je vhodné využít pro komunikaci agilní metodu, což by mělo snížit riziko neúspěšného ukončení projektu.

3.10.1 Identifikace zainteresovaných stran

Proces probíhající v inicializační fázi projektu, který je následně opakován po každé významné změně. Proces slouží k identifikaci všech zainteresovaných osob, skupin a organizací nějakým způsobem ovlivňující činnosti, nebo výstupy projektu. Dalším krokem je analýza vlivu, zájmu a angažovanosti.

Vstupy:

- zakládací listina,
- plán řízení projektu,
- smlouvy,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- seznam zúčastněných stran,

- seznam požadavků na změny,
- úprava plánu řízení projektu,
- úprava projektových dokumentů.

3.10.2 Plánování zapojení zainteresovaných stran

Proces pro zajištění účasti zainteresovaných stran v projektu. V daném procesu je nutné zohlednit zájmy, očekávání a dopad na projekt jednotlivých stran. Výsledný dokument – plán zapojení zúčastněných stran definuje, jakým způsobem jsou jednotlivé strany zapojeny do projektu a jak je nutné s nimi jednat.

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- projektové dokumenty,
- smlouvy,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán zapojení zúčastněných stran.

3.10.3 Řízení zapojení zainteresovaných stran

Proces řídící se plánem zapojení zúčastněných stran projektu, který je vykonáván po celý životní cyklus projektu. Jedná se o komunikaci se zapojenými stranami ohledně řešení problémů projektu. Tím je zaručena plná informovanost všech zainteresovaných stran o stavu projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- seznam požadavků na změny,
- úprava plánu řízení projektu,

- úprava projektových dokumentů.

3.10.4 Kontrola zapojení zainteresovaných stran

Proces sloužící ke kontrole zainteresovaných stran a jejich vztah k projektu. Pokud se vyskytne problém, je nutné upravit strategii pro podporu zapojení jednotlivých stran. Proces je uskutečňován po celou dobu životního cyklu projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- údaje o vykonané práci,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- informace o výkonu práce,
- seznam požadovaných změn zapojení zainteresovaných stran,
- úprava plánu řízení,
- úprava projektových dokumentů.

4 AGILNÍ VÝVOJ

Agilní vývoj je druh přístupu k vedení projektu zaměřující se na dodávání produktu v závislosti na rychle měnících se podmínkách. V poslední době se jedná o stále více využívanou metodu pro vývoj softwaru. Agilní přístup se rovněž vyskytuje stále častěji i v jiných oborech, než pouze v IT. Velkým přínosem metody je rychlá reakce na změny, vysoká přizpůsobivost a metody jsou snadno aplikovatelné na komplexní problémy. Cílem agilních metod je co nejdříve a co nejlevněji dodat zákazníkovi produkt, který bude splňovat všechny jeho požadavky, bude snadné produkt udržovat a zároveň bude mít plnou podporu. [4, 5]

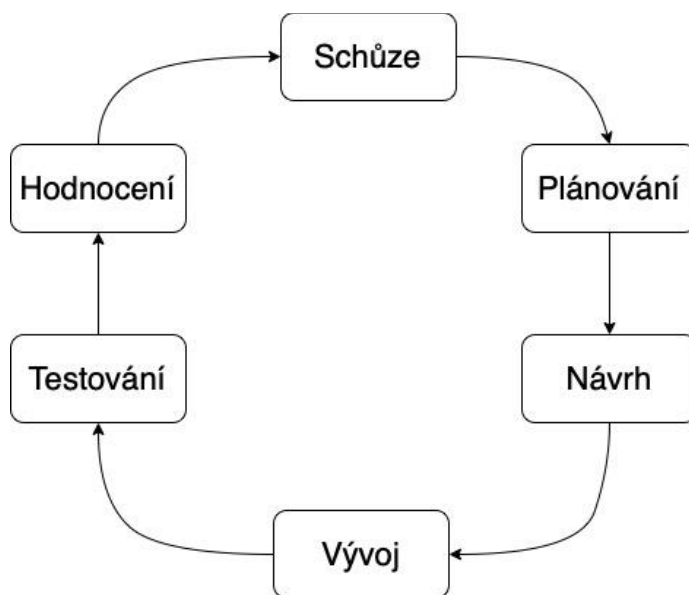
4.1 PRINCIPY AGILNÍCH METOD

Podpisem Manifestu agilního vývoje software vznikla Aliance, která definovala 12 principů pro úspěšné využívání agilních metodik [6]:

- Nejvyšší prioritou je včas a kontinuálně dodávat software, který zákazníkům přináší hodnotu.
- Změnu požadavků je možné provést i v pozdějších fázích vývoje, protože tým může zákazník získat konkurenční výhodu.
- Dodání fungujícího produktu v intervalech týdnů až měsíců, s preferencí kratší periody.
- Uživatelé a vývojáři spolupracují denně na projektu.
- Motivovaní jedinci, kteří mají vytvořené podmínky pro práci a mají podporu vedení, jsou klíčovým faktorem úspěchu projektu.
- Nejefektivnějším způsobem přenosu informací v rámci vývojového týmu je osobní komunikace.
- Primární mírou úspěchu je fungující software.
- Agilní procesy předpokládají „zdravý“ vývoj.
- Perfektní technické řešení i návrh.
- Zásadním požadavkem je jednodušší řešení.
- Nejlepší architektury, požadavky a návrhy vznikají od samo organizujících se týmů.
- Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování i zvyklosti.

4.2 CYKLUS AGILNÍHO VÝVOJE PRODUKTU

Základní verze využívání agilní metodiky pro vývoj produktu je možné rozdělit do šesti základních bodů. Od tradičního vodopádového modelu jsou tyto body opakovány vícekrát v průběhu vývoje, což znamená, že zde dochází k postupnému přehodnocování požadavků na produkt a je pevně daným krokem přehodnocování rizik, což napomáhá snížení rizik projektu na minimální hodnotu. [7, 8]



Obr. č. 1 - Cyklus agilního vývoje produktu

4.2.1 Koncept

Prvním krokem agilního vývoje je koncept, sloužící k identifikaci obchodních příležitostí a k odhadu času nutného k realizaci projektu. Na základě získaných informací je následně možné rozhodovat o realizaci projektu.

4.2.2 Zahájení

Během zahajovací fáze jsou rozvrhnuty pracovní pozice a najmutí příslušní zaměstnanci a je jasně stanoveno financování projektu na základě komunikace se zákazníkem a jeho požadavcích na produkt. Hlavním krokem fáze zahájení je využití zkušeností z již ukončených projektů, dle kterých je možné sestavit časový plán s definicí očekávaných dat ukončení jednotlivých sprintů, na základě náročnosti jednotlivých úkolů.

4.2.3 Iterace

Iterace je krokem k reálnému vytváření produktu na základě požadavků a průběžné zpětné vazby. Název iterace má z důvodu cyklického vytváření, kdy je vývoj produktu rozdělen do jednotlivých fází (sprintů), kdy na konci každého sprintu je cílem mít vylepšený a funkční produkt. [8]

V prostředí informačních technologií (IT) se jedná o spustitelný kód obsahující opravené chyby po předchozím sprintu a přidané nové funkcionality dle časového plánu. Každý vývojový cyklus se zde skládá z následujících čtyř bodů [9]:

- definice požadavků na základě produktových backlogů, sprint backlogů a zpětné vazby od zainteresovaných stran,
- vývoj produktu na základě stanovených požadavků,
- testování zajištění kvality a produktová dokumentace,
- získávání zpětné vazby od zainteresovaných stran za účelem definování požadavků na další sprint.

4.2.4 Uvolnění

V této fázi dochází k finálnímu testování produktu a opravení případných drobných problémů, chyby bývají označeny pomocí ticketu popisující chybu a jejich prioritu. Během uvolnění produktu dochází k finalizaci uživatelské příručky a produktové dokumentace. Po dokončení těchto kroků je produkt uvolněn. [10] V oblasti IT se jedná především o automatizované testování softwaru, kde je brán zřetel na funkčnost samotného programu, jeho výkonnost a stabilitu. Pokud program projde testováním, je nasazen na produkční server, čímž se stane dostupným pro uživatele.

4.2.5 Produkce

Krok produkce je určena k provádění nezbytné podpory pro funkcionalitu a údržbu produktu a zároveň učít uživatele produkt správně využívat. Zmíněná podpora musí být prováděna až do chvíle, kdy je produkt vyřazen. V IT je nezbytné kontrolovat hladký chod produktu z toho důvodu, že často dochází k problémům přetížení systému a následného zpomalení produktu, reagování na možné bezpečnostní hrozby a výpadek funkcionality produktu. Na zmíněné chyby se zde reaguje vygenerováním chybového ticketu s označením hot-fix, což označuje vysokou prioritu a opravu přímo na produkčním serveru.

4.2.6 Stažení/nahrazení produktu

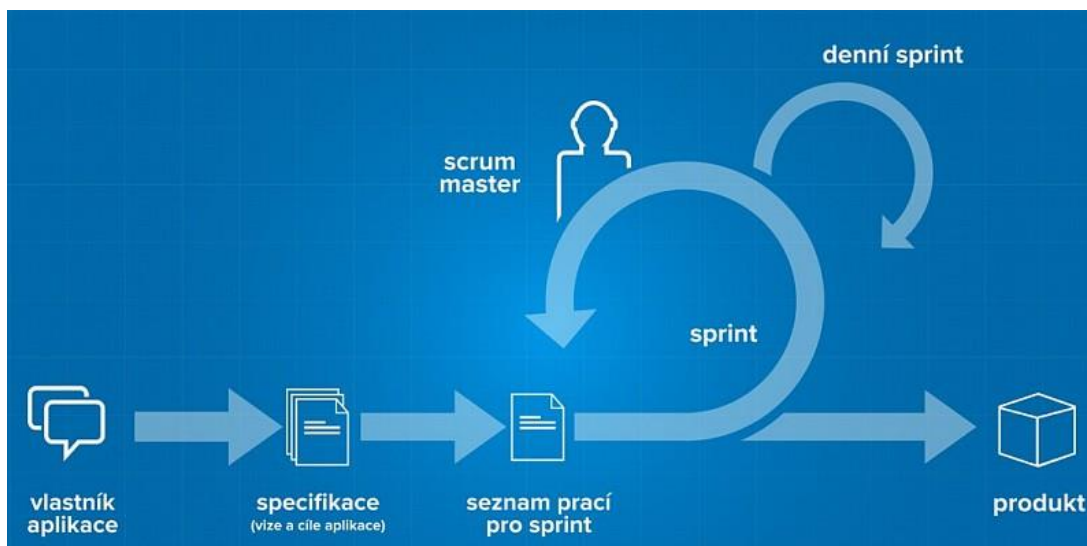
Poslední krok agilního cyklu je vyřazení stávající verze z produkce. V tento okamžik jsou uživatelé produktu informováni od jeho stažení, případně nahrazení. K tomuto kroku dochází v momentě, kdy je produkt zastaralý, nebo je proti obchodnímu modelu společnosti.

4.3 METODY AGILNÍHO ŘÍZENÍ

Veškeré agilní metody vychází z hlavní myšlenky, kterou je přizpůsobit se veškerým přichozím změnám co nejefektivněji, a přitom nadále poskytovat produkt s odpovídající kvalitou a funkcionalitou. Jednotlivé metody mají však mírný rozdíl ve způsobu definice jednotlivých kroků vývoje produktu. [5]

4.3.1 Scrum

Metoda Scrum je využívána pro dodávání komplexních produktů, z toho důvodu je nutné metodu využívat na týmech s různými specializacemi tak, aby byl tým schopný splnit veškeré požadavky zainteresovaných stran. Metoda je nejlépe využitelná u menších týmů z toho důvodu, že funkci kouče týmu přebírá samotný tým. Scrum tým využívá tři role – Scrum Master, Produktový vlastník a vývojový tým. [4, 11]



Obr. č. 2 - Diagram metody Scrum

Úkolem Scrum Mastera je zajištění správného pochopení metodiky scrum – rozvíjet tým do samostatnosti, efektivnosti a spokojenosti. Scrum Master má rovněž za úkol vedení pravidelných schůzek týmu a musí zajistit dodržení jejich maximální doby trvání tak, aby nebyla ohrožena efektivita schůzek a zároveň pozornost členů týmu. Tyto schůze obvykle probíhají každý den a každý člen týmu zde oznámí co dokončil předchozí den, co má v plánu dokončit dnes a zda se

vyskytly nějaké komplikace. Roli Scrum Mastera by neměl zastupovat nikdo z vývojového týmu, nebo samotný Produktový vlastník, protože by to mohlo vést k narušení metodiky uvnitř týmu. [4, 12, 13]

Produktový vlastník je člen týmu, který je přímo zodpovědný za výsledný produkt. Sám určuje hodnoty výsledného produktu a pokouší se maximalizovat jeho hodnotu v závislosti na komunikaci se všemi zainteresovanými stranami. Hlavní pracovní náplní je starání se o produktový Backlog, což je seznam všech úkonů nezbytných k úspěšnému dokončení produktu – jednotlivým položkám v seznamu rovněž přiřkládá jejich priority. [13]

Vývojový tým by se měl dle metody samostatně organizovat, tudíž jednotliví členové si rozdělují práci dle svých schopností. Rovněž by do týmu neměly vstupovat prvky přímého řízení od nadřízeného pracovníka, protože produkt zná nejlépe samotný vývojový tým. [13]

Riziko spjaté s dodržením požadavků zainteresovaných stran je sníženo využíváním tzv. sprintů. Sprint je v agilních metodách chápán jako jeden cyklus a u metody Scrum je jeho délka běžně mezi jedním až čtyřmi týdny. Úkolem vývojového týmu během sprintu je splnit všechny úkoly, které byly na dané časové období naplánovány. V případě prodloužení některého z úkolů je možné daný úkol přenést do dalšího sprintu, nikoliv však prodloužit sprint.

Plánování jednotlivých sprintů probíhá před jeho začátkem a je zde nutné stanovit, co v nadcházejícím sprintu vytvořit a jakým způsobem je možné se k těmto výsledkům dopracovat. Obvykle je na zodpovědnosti produktového vlastníka výběr těchto položek z jeho produktového backlogů na základě jejich priority. Odhadovaný čas pro ukončení dané funkcionality a její náročnost je dána jako odhad zkušenějších členů vývojového týmu. Následně se koná schůze sloužící k seznámení členů s jednotlivými funkcionalitami, po seznámení členů scrum týmu s náročností a nutnou technickou znalostí jednotlivých funkcionalit probíhá rozložení funkcionalit do menších úkolů, které se rozebírají jednotliví členové týmu. [14]

4.3.2 Lean

Lean metoda se postupně vyvíjí od 50. let 20. století ve společnosti Toyota Production System (TPS), kdy bylo nutné začít využívat vysokou úroveň flexibility v prostředí hromadné výroby. Metoda je založena na kultuře neustálého zlepšování, podpoře zaměstnanců a soustředění se na skutečné hodnoty společnosti. Výsledkem má být rychlé, jednoduché a přehledné vytváření produktu a služeb bez zbytečného plýtvání. Tyto požadavky jsou plněny spolu s vyvažováním jednotlivých procesů a následného navázání těchto procesů na zákazníka. [4, 15]

Základní principy metody Lean se dají shrnout do celkem sedmi bodů [16]:

- Eliminace odpadu – nevyužívat nepotřebné služby/produkty.
- Zlepšení učení – na základě vzdělávání pracovníků v rámci činností projektu a na základě toho upravovat prováděné činnosti.
- Rychlé rozhodování – rozhodovat o procesech so nejrychleji, ale za využití co největšího množství informací o dané problematice.
- Rychlé dodávání – dodání výsledného produktu co nejrychleji (snižovat potřebný čas pro vývoj produktu a tím i náklady). Rovněž co nejrychleji reagovat na zpětnou vazbu od zákazníka.
- Posílení týmu – rozdělení odpovědnosti mezi všechny členy týmu a zároveň je začlenit do procesu návrhu produktu.
- Integrita softwaru – nehledět pouze na výsledný produkt, ale brát v úvahu i kvalitní zdokumentování všech částí vývoje produktu.
- Výsledný produkt vidět jako celek – je nutné nad produktem přemýšlet komplexně, ne se pouze specializovat na jediné odvětví.

Pomocí těchto principů je možné výsledný produkt dodávat rychle, a to z důvodu kvalifikovaných členů týmu, kteří mají rozhodovací povinnost a komunikují přímo se zákazníkem. Metoda využívá eliminaci nepotřebných funkcionalit, které by mohly vést ke složitosti a nepřehlednosti výsledného produktu. Metoda se rovněž snaží docílit co nejvyšší kvality výsledného softwaru, a to podporou automatických testů.

4.3.3 Kanban

Metoda Kanban byla vyvinuta zaměstnancem Toyota manufacturing Taiichi Ohno v roce 1953, kde daná metoda byla následně aplikována. Metoda je inspirována plánováním, kontrolou a doplňováním zásob. Princip metody je založen na využívání tabule rozdělené do třech sloupců (úkoly, co se mají vykonat, aktuálně vykonávané úkoly a dokončené úkoly). Metoda se na rozdíl od ostatních metod vyznačuje jednoduchou implementací do projektového prostředí, protože je možné ji začít využívat v jakékoliv části životního cyklu projektu. Další výraznou odlišností od ostatních agilních metod je možnost absence iteračního kroku při běžné aplikaci, čímž jsou vždy plněny požadavky především na základě priority. [4, 17]

Využívání metody Kanban je velmi jednoduché a přehledné, což napomáhá snadné orientaci v postupu prací na projektu pro všechny členy týmu. Díky využívání "pull" systému je možné snadno monitorovat pracovní postup a detekovat slabá místa projektu, případně upozornit na úkony, které blokují celkový postup. Využívání metody Kanban pro tým je nejvhodnější, pokud jsou vyžadovány následující podmínky [4]:

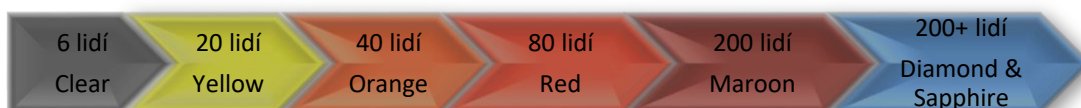
- flexibilita,
- zaměření na neustálé dodávání produktu,
- zvýšená produktivita a kvalita,
- zvýšená efektivita,
- zaměření na členy týmu,
- variabilní pracovní zátěž,
- snížení množství odpadu.

4.3.4 Extrémní programování (XP)

Extrémní programování je agilní metoda, která se zaměřuje na rychlost a nepřetržité dodávání produktu. Základní myšlenkou je reagovat okamžitě na potřeby a požadavky zákazníků. Hlavními body metody jsou zapojení zákazníků, nepřetržité plánování na základě rychlé zpětné vazby, testování a týmová spolupráce. Software vznikající touto metodou je obvykle dodáván každý jeden až tři týdny. V metodě je dána velká zodpovědnost vývojovému týmu, který na základě komunikace rozhoduje do určité míry o prioritách daných funkcionalit, rovněž zodpovídá za funkčnost a její ověření pomocí testování při každém cyklu vývoje produktu. [4]

4.3.5 Crystal

Metoda je nejlehčí a nejvíce přizpůsobivou metodou s klíčovými vlastnostmi jako jsou komunikace, jednoduchost a týmová spolupráce, ale je zde však nutné se spolehnout na dobré vztahy mezi vývojovým týmem a zákazníkem. Metoda klade především důraz na dodávání funkčního softwaru s vysokým zapojením zákazníka, rychlé nasazení nových funkcionalit, a to bez byrokracie, jako je například projektová dokumentace. Metoda chápe projektové prostředí, jako jedinečné a pro každou aplikaci potřebuje využít mírně přizpůsobený soubor postupů a procesů. [4] Pro základní orientaci mezi těmito přístupy je možné využít sbírku agilních procesních modelů, které jsou rozděleny dle počtu členů v týmu (viz Obr. č. 3 - Agilní procesní metody dle počtu členů týmu) [18].



Obr. č. 3 - Agilní procesní metody dle počtu členů týmu

4.3.6 Scrumban

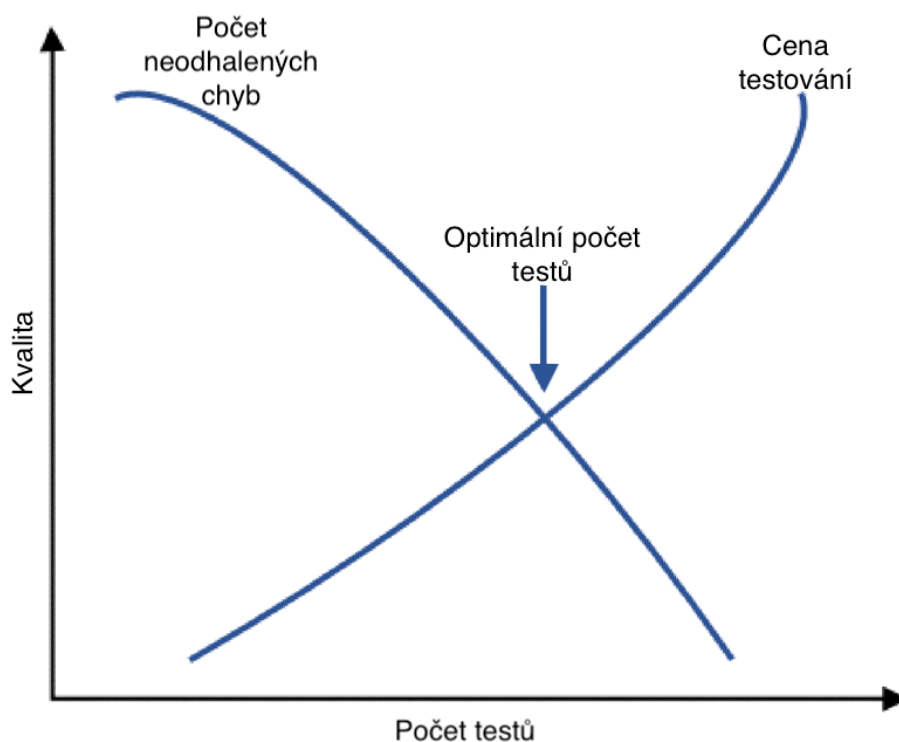
Scrumban byl prvně navržen pro řízení přestupu mezi metodou Scrum na Kanban. Při jeho využívání však došlo k propojení silných stránek obou metod a Scrumban začal být vyvíjen jako nová hybridní agilní metoda. Metoda využívá Scrum k řízení úkonů na projektu a Kanban pro optimalizaci jeho procesů. [4]

Práce je v tomto prostředí organizována do krátkých sprintů, kde jsou jednotlivé úkoly vizualizovány na tabuli Kanban, která je však limitovaná časem sprintu. Během prací na projektu se konají krátké každodenní schůzky, kde členové týmu informují o svém postupu, co mají v plánu na aktuální den a hlásí případné problémy. Plánování nových sprintů/fází probíhá až v momentě, kdy je množství rozpracované práce menší než předem stanovený limit. Scrumban nevyužívá předem stanovené role, zachovává se hierarchie společnosti.[4]

5 TESTOVÁNÍ SOFTWARE

Testování softwaru je proces, který má za úkol vyhodnotit funkčnost vyvíjené aplikace s důrazem na požadavky zainteresovaných stran na výsledný produkt. Během procesu je důležité identifikovat případné vady, o jejichž existenci je nutné informovat vývojový tým. Cílem procesu je kontrolovat a zároveň zvyšovat kvalitu výsledného produktu. Testování je rovněž dle standardu ANSI/IEEE 1008 definováno jako proces analýzy softwarového produktu za účelem detekce rozdílů mezi existujícími a požadovanými podmínkami na produkt a celkového vyhodnocení vlastností produktu. [19, 20]

Proces testování softwaru je možné vnímat rovněž jako činnost zaměřenou na identifikaci možných rizik a následného stanovení optimálního počtu testů. Cílem je, aby testy nebyly nadměrně cenově náročné, což pro IT prostředí znamená časovou a výpočetní náročnost. [21]



Graf č. 1 - Závislost kvality produktu na počtu testů

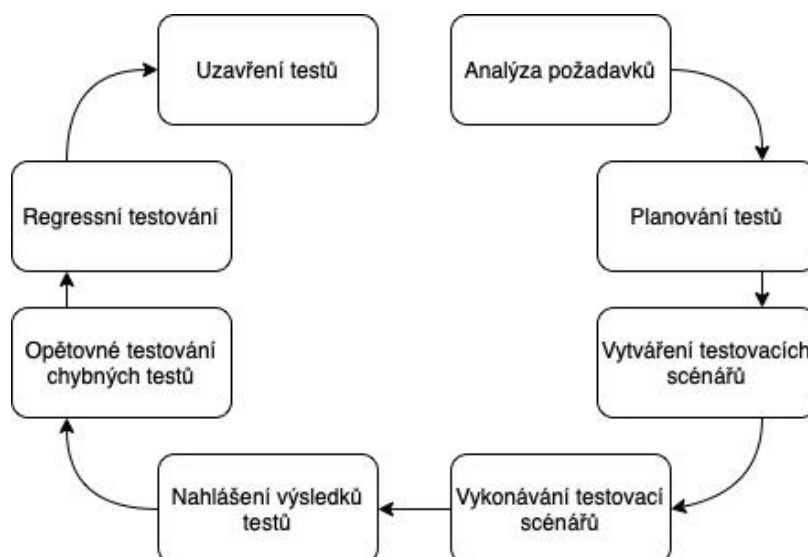
Testování softwaru je rozděleno do třech základních kroků, které jsou nezbytné pro zaručení kvality výsledného produktu a jsou zahrnuty do cyklu vývoje softwaru. Jednotlivé kroky se nazývají testování jednotek, integrační testy a testování systému.

- Testování jednotek – jedná se o testování každé jednotlivé části softwaru, kterou vývojář vytvoří a následně ověřuje samotný vývojář, nebo softwarový tester její funkcionalitu.

- Integrační testy – po vyvinutí a otestování nezávislých částí softwaru přichází jejich integrace, čímž se mohou objevit nové chyby v návaznosti jednotlivých částí. Z toho důvodu se využívají integrační testy, které testují spojení jednotlivých částí.
- Systémové testování je posledním krokem testování, kdy je testován celý výsledný produkt. Zmíněné testování má za úkol odhalit případné chyby při sestavení výsledného produktu.

5.1 ŽIVOTNÍ CYKLUS TESTOVÁNÍ SOFTWARE

Životní cyklus testování softwaru definuje jednotlivé základní kroky pro organizaci celého testovacího procesu od plánování testů až po analýzu jejich výsledků. Prvním krokem je přezkoumání požadavků na produkt dle požadavků všech zainteresovaných stran, dle kterých bude produkt testován. Dalším fází je plánování testů, kde je dohodnuta strategie testování a jejím výstupem je testovací plán, bez kterého nelze testy koordinovat. Třetí fází je návrh testovacích scénářů pro daný produkt obsahující specifikaci vstupních dat, podmínky provádění testů a jejich očekávané výstupy. Zároveň jsou využívána i chybná data tak, aby bylo možné otestovat reakci produktu na špatné vstupy. Fáze samotného testování znamená provádění aktivit z testovacího plánu. V případě, že test projde bez chyby, je test označen za úspěšný, v opačném případě je test spojen s konkrétní nalezenou chybou a je zaveden do výsledků jako neúspěšný. Výsledky jsou následně předány vývojovému týmu, který má za úkol dané chyby opravit a opravené soubory jsou opět přetestovány. [22]



Obr. č. 4 - Životní cyklus testování softwaru

5.2 TESTOVACÍ METODY

Na základě určených testovacích případů je možné zvolit nejvhodnější metodu pro vykonání testů. V oboru testování softwaru jsou využívány především tři metody, a to Black Box, White Box a Grey Box. [22]

5.2.1 Metoda White Box

Testovací metoda White Box je použitelná ve všech testovacích krocích a pro její aplikaci jsou nutné programovací dovednosti. Metoda zajišťuje kontrolu, zda produkt chrání data a zároveň zachovává vyžadovanou funkčnost. Testovací proces zde využívá znalost vnitřního kódu aplikace a tím je schopný otestovat všechny nezávislé části projektu, vnitřní logiku funkcí, kontrolu paměti a funkcionalitu datových struktur. [23]

5.2.2 Metoda Black Box

Jedná se o nejrozšířenější metodu využívanou pro testování softwaru. Testy jsou zde sestavovány na základě kombinace vstupních a výstupních dat, a to bez znalosti vnitřní logiky funkcí a datových struktur. Metoda je určena spíše pro kontrolu funkcionality daných prvků programu. [23]

5.2.3 Metoda Grey Box

Metoda Grey Box je kombinací dvou předchozích metod. Myšlenkou metody je seznámení se testera s vnitřní strukturou kódu a využití těchto znalostí při návrhu testů. Tímto způsobem je metoda schopna zajistit optimalizaci testů a jejich lepší funkcionalitu. [22, 23]

5.3 ZLEPŠOVÁNÍ TESTOVACÍCH PROCESŮ

Se zvyšující se robustností produktů, kdy je stále vysoký požadavek na kvalitu a v agilním prostředí na opětovné dodávání produktu v krátkých časových intervalech, je nutné upřednostňovat určitou skupinu testů. Pro možnost upřednostnění některých testovacích sad jsou využívány algoritmy, které z weblogů vytváří testovací sady relevantní pro relaci uživatele na základě nejvíce využívaných kombinací. Zároveň je nutné provádět empirické studie pro vyhodnocení účinnosti konkrétních testů v závislosti na příslušných testovacích sadách [24]. Nástroje využívané pro sběr těchto dat mají za úkol formátovat protokoly webových aplikací do testovacích sad; získaná data slouží rovněž ke stanovení priority testů. Na základě výstupů probíhá kontrola, zda došlo ke zvýšení poměru detekce chyb a testů, či nikoliv. Pro správnou

funkčnost testů je nutné generovat vstupní hodnoty, které mohou být generovány pomocí genetických algoritmů, čímž je zajištěno dynamické generování dat v závislosti na složitosti programu. [25]

5.3.1 Automatizace testů

V momentální době se jedná o hlavní vylepšení testovacích procesů, díky kterému je možné snížit počet manuálních testů a tím urychlit celý proces. Testovací proces využívá pro automatizované testy software, který má za úkol provádět testy na základě porovnávání skutečných výsledků s očekávanými. [26]

Software pro testování je nejčastěji v podobě aplikačního rámce, který je založen na principu abstrakce. Díky této vlastnosti je možné vytvářet různé skripty pro různé moduly aplikace, jež má být testována, což přispívá komplexnosti celého procesu testování. Každá testovací funkce je po vytvoření zařazena do knihovny, čímž se stane dostupnou v rámci projektu a tím je urychlen vývoj ostatních testů, které mohou využívat jeho funkcionality. Mezi další funkcionality automatizovaných testů patří využívání datově řízeného testovacího frameworku, který má za úkol využívání různých sad testovacích dat uložených v různých souborech. Využitím zmíněného frameworku dochází k optimalizaci počtu testů, a tím i k celkové rychlosti testovacího procesu.

Mezi hlavní výhodu automatizovaných testů při agilním vývoji je jejich zařazení do CI/CD (Continuous Integration / Continuous Delivery) pipeline, čímž je myšlena série kroků nutných k dodání nové verze produktu. Zmíněná metoda je běžně využívána na verzovacích systémech, kdy při nahrání jakékoliv části kódu do repositáře dojde k sestavení aplikace a jejího následného automatického testování. Při ukončení testů jsou uživatelé informováni o jejich výsledku a případné chyby jsou zaznamenány. Některé frameworky pro automatizované testování, jako je například Cypress, nebo Selenium pracují na principu e2e testování, což je metoda ověřující kompletně celý software včetně jeho integrace s externími rozhraními. Zmíněné frameworky podporují využívání záznamů obrazovky z testovacího procesu a tím jsou schopny přehledně ukázat na problematickou komponentu, případně je zde možné využít statistiky úspěšnosti testů a celkové výkonnosti aplikace.

6 NÁVRH APLIKACE

Aplikace byla vyvíjena s důrazem na snadnou rozšiřitelnost, nezávislost na prostředí jejího nasazení, zároveň na zajištění víceuživatelského přístupu a snadného uvolnění nové verze programu. Základní myšlenkou je vytvoření webové aplikace, kde je možné sledovat potřebné informace a kontrolovat postup jednotlivých projektových úkolů a rizik s ním spojených.

Pro přihlášení do aplikace má uživatel přehled o projektech prováděných ve společnosti. Každý projekt obsahuje informace o datu začátku a konce, krátkém popisu a pro lepší orientaci v projektovém procesu využívá logickou rámcovou matici. Projekt podporuje možnost rozdělení do fází skládajících se z jednotlivých úkolů a nezávisle na fázích umožňuje přidávat informace o zjištěných chybách, které je nutné opravit.

Pro zajištění izolovanosti aplikace a snadného nasazení je vhodné využít pro aplikaci Docker, který umožňuje snadnou lokální instalaci díky přenášení všech potřebných závislostí a izoluje aplikaci do kontejnerů. Aplikace se skládá ze dvou kontejnerů. Jeden kontejner zajišťuje chod a izolovanost databáze, druhý kontejner obstarává funkčnost samotné webové aplikace. Pro vývoj byl využit programovací jazyk Python s frameworkem Django pro backendovou část. Framework Bootstrap a jazyk HTML byly použity pro vytvoření frontendové části programu.

6.1 UŽIVATELSKÉ ROLE

Z důvodu různých personálních hierarchií ve společnostech je vhodné rozlišit uživatelské role napříč aplikací tak, aby některé funkcionality, jako je přidání, úprava, nebo odstranění prvků bylo přístupné pouze zaměstnancům s dostatečným oprávněním. V tomto projektu jsou uživatelské role rozděleny do čtyř skupin na základě jejich postavení ve společnosti.

Project manager

Role projektový manažer spravuje účet celé společnosti. Zároveň má přístup ke všem částem projektu. Aplikace mu umožňuje zobrazování, editaci a odstraňování projektů, jeho fází, úloh a nahlášených problémů s aplikací.

Scrum master

Práva na pozici scrum master jsou omezená k úpravám v rámci projektu a jeho editace. Uživatel tedy oproti projektovému manažerovi postrádá práva k úpravám společnosti.

Worker

Jedná se roli přidělenou zaměstnanci společnosti, který by měl mít informace o projektu a jeho fázích. Je jí tedy umožněno přidávání a editace jednotlivých rizik, úkolů a problémů s aplikací.

Customer

Zákaznický účet obsahuje pouze možnost pohledu na postup v projektu, případně přidávání zjištěných problémů s aplikací.

6.2 POŽADAVKY NA UŽIVATELSKÉ ROZHRAŇÍ

Úvodní stránka programu slouží pro přihlášení do aplikace. Po úspěšném přihlášení má uživatel k dispozici seznam projektů vedených pod společností.

Při výběru některého z projektů se zobrazí detail projektu s údaji s ním spjatými. V obrazovce detailu projektu je postranní menu sloužící k pohybu mezi jednotlivými položkami obsaženými v projektu, jako je logická rámcová matice (LRM), úkoly spojené s projektem, časový plán, registr rizik, problémy spjaté s projektem a seznam zaměstnanců společnosti s kontakty.

Záměr	Objektivně ověřitelné ukazatele	Zdroje informací k ověření	Záměr vůči vyššímu záměru organizace
Cíle	Objektivně ověřitelné ukazatele	Zdroje informací k ověření	Předpoklady, za jakých cíl skutečně přispěje a bude v souladu se záměrem
Výstupy	Objektivně ověřitelné ukazatele	Zdroje informací k ověření	Předpoklady, za jakých výstupy skutečně povedou k cíli
Klíčové činnosti	Zdroje	Časový rámec aktivit	Předpoklady, za jakých klíčové činnosti skutečně povedou k výstupním

Tab. č. 2 - Logická rámcová matice

Detail projektu slouží k zobrazení základních údajů o projektu, jako je název, jméno projektového manažera, stav projektu, datum začátku a jeho ukončení, základní popis a jeho přehled fází. Další úzce spjatá položka s detailem projektu je LRM projektu, která zajišťuje přehlednější řízení projektu především v procesu zahájení projektu a určení jeho strategie.

Nejdůležitější částí pro řízení projektu by měla být položka časového plánu projektu, který by se měl skládat z jednotlivých fází a jejich úkolů. V této části aplikace tedy nabízí základní přehled o postupu prací v každé fázi, případně zobrazuje, zda už je daná fáze dokončena. Aplikace rovněž nezávisle na fázi zobrazuje všechny úkoly a případné nahlášené chyby aplikace.

Další položkou je registr rizik, které budou vyhodnocovány na základě analýzy rizik metodou RIPRAN. Pro vyšší přehlednost o úrovni rizika je využito grafického zobrazení rizik v matici. Cílem zkoumání rizik je přesunout zobrazená rizika v matici co nejvíce doprava po hlavní diagonále matice rizik. Dle formuláře RIPRAN 5x5 [2]:

velmi vysoká pravděpodobnost – VVP	nad 0,8
vysoká pravděpodobnost – VP	nad 0,6 do 0,8 včetně
střední pravděpodobnost – SP	nad 0,4 do 0,6 včetně
nízká pravděpodobnost – NP	nad 0,2 do 0,4 včetně
velmi nízká pravděpodobnost – VNP	do 0,02 včetně

Tab. č. 3 RIPRAN: Třídy pravděpodobnosti

velmi velký dopad na projekt – VVD	od 15 % rozpočtu projektu
velký dopad na projekt – VD	nad 10 do 15 % rozpočtu projektu včetně
střední dopad na projekt – SD	nad 5 do 10 % rozpočtu projektu včetně
malý dopad na projekt – MD	nad 1 do 5 % rozpočtu projektu včetně
velmi malý dopad na projekt – VMD	do 1 % rozpočtu projektu včetně

Tab. č. 4 RIPRAN: Třídy dopadu na projekt

	VVD	VD	SD	MD	VMD
VVP	VHR	VHR	VHR	VHR	SHR
VP	VHR	VHR	VHR	SHR	NHR
SP	VHR	VHR	SHR	NHR	NHR
NP	VHR	SHR	NHR	VNHR	VNHR
VNP	SHR	NHR	NHR	VNHR	VNHR

Tab. č. 5 RIPRAN: Přiřazení hodnoty rizika

velmi vysoká hodnota rizika – VVHR
vysoká hodnota rizika – VHR
střední hodnota rizika – SHR
nízká hodnota rizika – NHR
velmi nízká hodnota rizika – VNHR

Tab. č. 6 RIPRAN: Hodnoty rizika

7 IMPLEMENTACE APLIKACE

Na základě návrhu aplikace byla implementována dockerizovaná aplikace za využití jazyka Python a HTML a frameworků Django a Bootstrap.

Docker slouží k zabalení aplikace a jejích závislostí do virtuálního kontejneru, který je následně možné spustit na jakémkoliv systému, což umožňuje běh aplikace i na cloudových systémech.

Python je multiparadigmatický programovací jazyk, což znamená možnost využití jazyka i pro objektově orientovaný přístup. Velkou výhodou jazyka je jeho celková jednoduchost a celková vysoká produktivnost z hlediska psaní programů. Python se rovněž v poslední době stal jedním z nejpobulárnějších programovacích jazyků, díky čemuž je pro něj vyvíjeno množství frameworků usnadňujících celkový vývoj. Díky jeho vysoké popularitě je rovněž velmi aktivní komunita kolem tohoto jazyka, což vede k jeho neustálému vylepšování.

Jedním z frameworků Pythonu je Django, které umožňuje vývoj bezpečných a snadno udržovatelných webových aplikací. Framework obstarává rutinní práci na propojování jednotlivých částí aplikace a tím umožňuje její rychlejší vývoj. Díky jeho využívání je zaručena kompletnost, univerzálnost, zabezpečení, udržovatelnost, škálovatelnost a přenosnost výsledné aplikace.

Pro vytvoření samotného uživatelského rozhraní je využito programovacího jazyka HTML, který je doplněn o sadu nástrojů kaskádových stylů (CSS) zvanou Bootstrap.

7.1 PROGRAMOVÁ ČÁST

V samotném projektu jsou mimo základní struktury kódu uloženy rovněž soubory zajišťující propojení s verzovacím systémem a konfiguračními soubory pro serverovou platformu. Pro sestavení aplikace na základě definice jejího prostředí zde slouží Docker-compose, který má za úkol sestavit dockerfile a tím vytvořit jednotlivé kontejnery pro aplikaci. Docker-compose zde obsahuje podrobnosti o portech, na kterých běží aplikace a na jakém portu běží databáze, včetně údajů k jejímu připojení. Samotná funkcionality projektu je rozdělena do dvou hlavních částí, a to na sekce risk a users, které jsou vytvořeny pomocí Django frameworku. Využívání Django požaduje určitý postup, jak strukturovat jednotlivé soubory.

- Složka migrations obsahuje automaticky generované databázové tabulky,
- složka templates obsahuje prvky uživatelského rozhraní,
- forms.py je spojený s Django formulářem ModelForm, který napomáhá komunikaci mezi formulářem a modelem,

- models.py slouží k zápisu modelů databázových tabulek, Django je následně přeneseno do postgres databáze,
- utils.py slouží k nastavení cest napříč aplikací,
- views.py usnadňují práci s daty mezi uživatelským rozhraním a databází.

7.2 DATABÁZE

Pro ukládání dat byla využita databáze PostgreSQL, což je otevřený objektově-relační databázový systém. Funkce této databáze zahrnují databázové transakce, konzistenci a izolovanost. Z toho důvodu je vhodná pro ukládání uživatelských dat.

Samotná databáze v programu je složena z MODEL DATABAZE

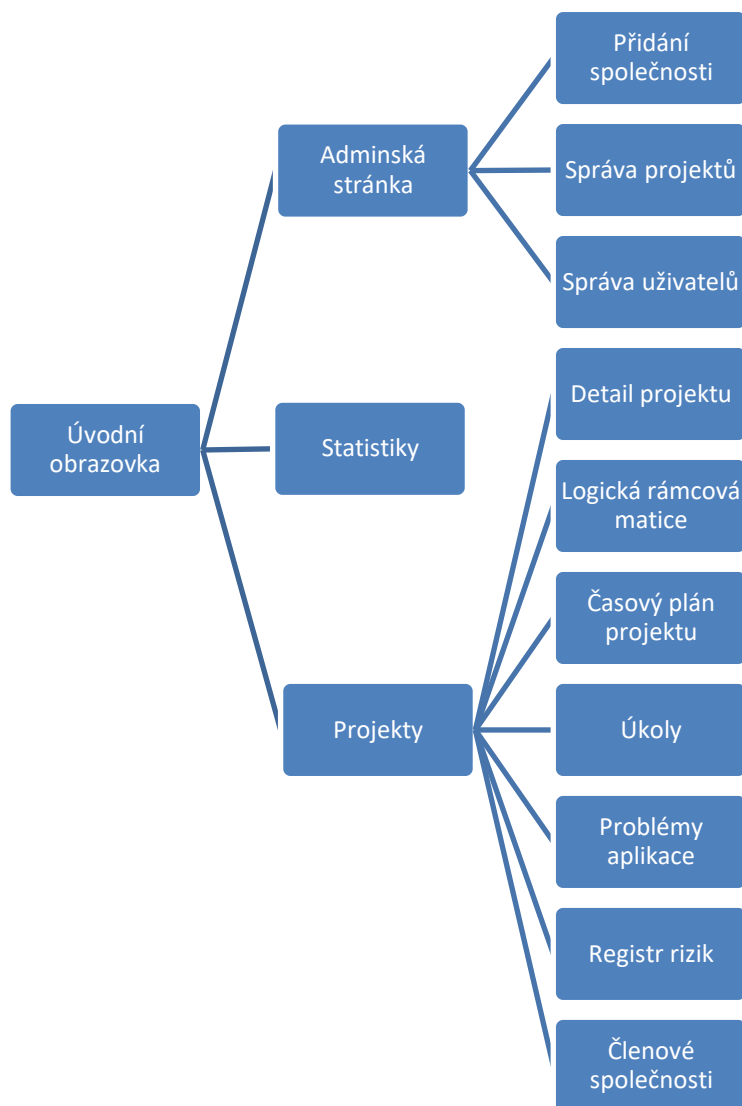
7.3 UŽIVATELSKÉ ROZHRAŇÍ

Aplikace běží na webovém serveru služby Heroku, která disponuje omezenou verzí pro zveřejnění webových aplikací. Přihlášení do aplikace je možné pomocí údajů obsažených v Tab. č. 7 Uživatelské údaje na webové adrese <https://support-risk-management.herokuapp.com>

Uživatel	Heslo
Admin	Superstrongpassword
Project_manager	Superpassword1

Tab. č. 7 Uživatelské údaje

Úvodní obrazovka slouží k přihlášení uživatele do webové aplikace. V případě, že uživatel není dosud registrován, musí požádat admina o zařazení mezi uživatele. Po úspěšném přihlášení jsou dostupné další funkcionality programu, které jsou členěny dle Obr. č. 5 - Struktura uživatelského rozhraní aplikace.



Obr. č. 5 - Struktura uživatelského rozhraní aplikace

7.3.1 Projects (Projekty)

Po úspěšném přihlášení je uživateli zobrazena stránka se seznamem projektů v jeho společnosti a ve vrchní části je zobrazeno hlavní menu aplikace obsahující položky projekty, statistiky, přístup do admínské stránky a možnost odhlášení. V závislosti na právech uživatele je možné jednotlivé projekty editovat, případně je přímo odstranit. Pro lepší orientaci mezi projekty je možné jejich filtrování na základě jejich stavu, nebo jejich vyhledávání přímo podle jména. Nové projekty je nutné vytvářet pomocí stránky Admínské stránky, odkud je možné rovněž spravovat informace týkající se společnosti.

7.3.2 Project details (Detaily projektu)

Pro zobrazení detailů projektu je nutné vytvořit nový projekt, případně kliknout na některý již vytvořený. Po přesměrování se v levé části nachází menu obsahující položky spjaté s projektem. Konkrétně se jedná o detail projektu, logickou rámcovou matici, časový plán projektu, úkoly, problémy s aplikací, registr rizik a seznam zaměstnanců společnosti.

The screenshot displays the 'Project details' page. At the top, there is a navigation bar with 'Project', 'Statistics', and 'Admin page' on the left, and 'Admin' and 'Logout' on the right. A sidebar on the left contains links for 'Project details', 'LFA', 'Time sheet', 'Tasks', 'Issues', 'Risk register', and 'Company members'. The main content area is titled 'Project details' and contains the following information:

Project name	Project_1	Project manager	Project_manager
Stav	active		
Start date	April 1, 2022	End date	July 28, 2022
Description	Project_1 for simulation		

Below this information are 'Edit' and 'Deactivate' buttons. The 'Phases' section features a 'State' dropdown menu and a 'Create Phase' button. A table lists the project phases:

Phase name	Phase manager	Start date	End date	State		
Speculate Planning	Master	April 11, 2022	April 20, 2022	active	Edit	Deactivate
Explore	Master_Project	April 20, 2022	May 11, 2022	active	Edit	Deactivate
Adapt	Master_Project	May 11, 2022	May 18, 2022	active	Edit	Deactivate
Close	Master_Project	May 18, 2022	May 25, 2022	active	Edit	Deactivate

Obr. č. 6 – Aplikace: Detaily projektu

V hlavní části obrazovky jsou k dispozici detailní informace o projektu, jako je jeho název, stav, datum začátku práce na projektu, popisu projektu a výpis jeho jednotlivých fází. Fáze je zde možné na základě oprávnění přidávat, editovat, či odstraňovat. Zobrazování jednotlivých fází je možné filtrovat dle jejich stavu na aktivní a neaktivní. Každou z fází je možné rozkliknout, čímž se zobrazí její detail obsahující podrobnější informace o fázi, jako jsou jednotlivé úkoly v ní obsažené a rovněž rizika spjatá s konkrétní fází. Jednotlivé položky je zde možné opět přidávat na základě oprávnění.

7.3.3 LFA (Logická rámcová matice)

Logická rámcová matice je zde zobrazena pro usnadnění řízení projektu a jeho strategického plánování. Defaultně je matice předvyplněna prvky naznačujícími způsob správného vyplnění matice. Pro její úpravu je nutné stisknout tlačítko *Edit* (upravit), čímž se jednotlivé prvky tabulky změní z popisků na textové vstupy. Úpravu je pak nutné potvrdit tlačítkem *Submit* (potvrdit).

Objectives	Success Measures	Verification	Assumptions
What is the overall broader impact to which the action will contribute?	What are the key indicators related to the overall goal?	What are the sources of information for these indicators?	What are the external factors necessary to sustain objectives in the long term?
What is the immediate development outcome at the end of the project?	Which indicators clearly show that the objective of the action has been achieved?	What are the sources of information that exist or can be collected? What are the methods required to get this information?	Which factors and conditions are necessary to achieve that objective? external conditions
What are the specifically deliverable results envisaged to achieve the specific objectives?	What are the indicators to measure whether and to what extent the action achieves the expected results?	What are the sources of information for these indicators?	What external conditions must be met to obtain the expected results on schedule?
What are the key activities to be carried out and in what sequence in order to produce the expected results?	What are the Measures required to implement these activities.	What are the action costs?	What preconditions are required before the action starts?

[Edit](#)

Obr. č. 7 - Aplikace: Logická rámcová matice

7.3.4 Time sheet (Časový rozvrh projektu)

Položka Time sheet zobrazuje časový plán projektu rozřazený do jednotlivých fází. Každá fáze zde nese informaci o jejím jménu, datu ukončení a stavu. V každé fázi je zobrazený seznam úkolů, které je nutné splnit pro ukončení fáze. Úkoly zobrazují informace o jejich ID, názvu, kdo má daný úkol na starosti, kdo úkol zkontroluje a jeho aktuální stav.

Project Statistics Admin page					Admin Logout
Project details		Envision Until: April 8, 2022			State: inactive
LFA		Speculate Planning Until: April 20, 2022			State: active
Time sheet					
Tasks					
Issues					
Risk register					
Company members					
Task ID	Name	Worker	Check from	State	
5	Activity iden	Project_manager	Project_manager	DONE_AND_CHECKED	
6	Time estimation	Project_manager	Master_Project	DONE_AND_CHECKED	
7	Resources	Project_manager	Project_manager	DONE_AND_CHECKED	
8	Analysis	Project_manager	Master_Project	DONE_AND_CHECKED	
9	Proposal	Project_manager	Master_Project	DONE_AND_CHECKED	
Explore		Until: May 11, 2022			State: active
Adapt		Until: May 18, 2022			State: active
Close		Until: May 25, 2022			State: active

Obr. č. 8 - Aplikace: Časový plán

7.3.5 Tasks (Úkoly)

V položce tasks je seznam všech úkolů spjatých s projektem, které je možné třídit dle jejich stavu: nepřidělené; přidělené; dokončené, nezkontrolované; dokončené, zkontrolované. Na základě oprávnění je v této části aplikace možné přidávat jednotlivé úkoly, které je nutné přiřadit rovnou do fáze projektu. Rovněž lze nahlížet na bližší informace, upravovat, či mazat již vzniklé úkoly.

7.3.6 Issues (Problémy)

Problémy aplikace je zde možné přidávat nezávisle na fázi, či oprávnění. Zde je žádoucí, aby i uživatel s nejnižším oprávněním (zákazník) měl možnost přidat chyby aplikace. Jednotlivé chyby je možné následně třídit dle jejich stavu, případně priority. Položky opět nabízí možnost zobrazení detailu, úpravu a jejich odstranění v závislosti na oprávnění uživatele.

Project Statistics Admin page Admin Logout

- Project details
- LFA
- Time sheet
- Tasks
- Issues
- Risk register
- Company members

Issues

State ▾ Add Issue

Issue ID	Name	Fix from	Check from	State	
1	Unable to save risk	Project_manager	Master	OPEN	Detail Edit Delete
2	Token	Master	Master	ASAP	Detail Edit Delete
3	Edit project	Master	Master	OPEN	Detail Edit Delete
4	Timesheet	Master	Master	ASAP	Detail Edit Delete
5	Database	Master	Master	HIGH-PRIORITY	Detail Edit Delete

Obr. č. 9 - Aplikace: Problémy produktu

7.3.7 Risk register (Registr rizik)

Zobrazení sekce registr rizik je rozděleno do dvou hlavních částí. Ve vrchní části je zobrazena matice rizik, dle metody RIPRAN s formulářem 5x5, kde v jednotlivých buňkách je zobrazen počet rizik v projektu dle míry jejich rizika. Ve spodní části je registr rizik spjatých s projektem, u kterých je možné opět na základě oprávnění zobrazit detail, upravovat riziko, či riziko úplně odstranit. Okno opět nabízí i možnost filtrování prvků registru rizik na základě jejich stavu a přidání nových na základě oprávnění uživatele.

Project Statistics Admin page Admin Logout

- Project details
- LFA
- Time sheet
- Tasks
- Issues
- Risk register
- Company members

Risk register

State ▾ Create Risk

	VHI	HI	MI	SI	VSI
VHP	1	0	0	0	0
HP	0	0	0	0	0
MP	0	0	1	0	0
SP	0	1	1	0	0
VSP	0	0	0	0	0

Risk name	Threat	Starter	Owner	Risk	State	Change date	Phase	
Resources	Insolvency of customer	Customer does not pay for product	Project_manager	SRV	Active	-	Explore	Detail Edit Delete
Implementation	Demand beyond team skills	Difficult system to understand	Master	MRV	Active	-	Speculate Planning	Detail Edit Delete
Planning	Choosing the wrong platform	Misunderstood product requirement or insufficient knowledge of technology	Master	MRV	Active	-	Speculate Planning	Detail Edit Delete

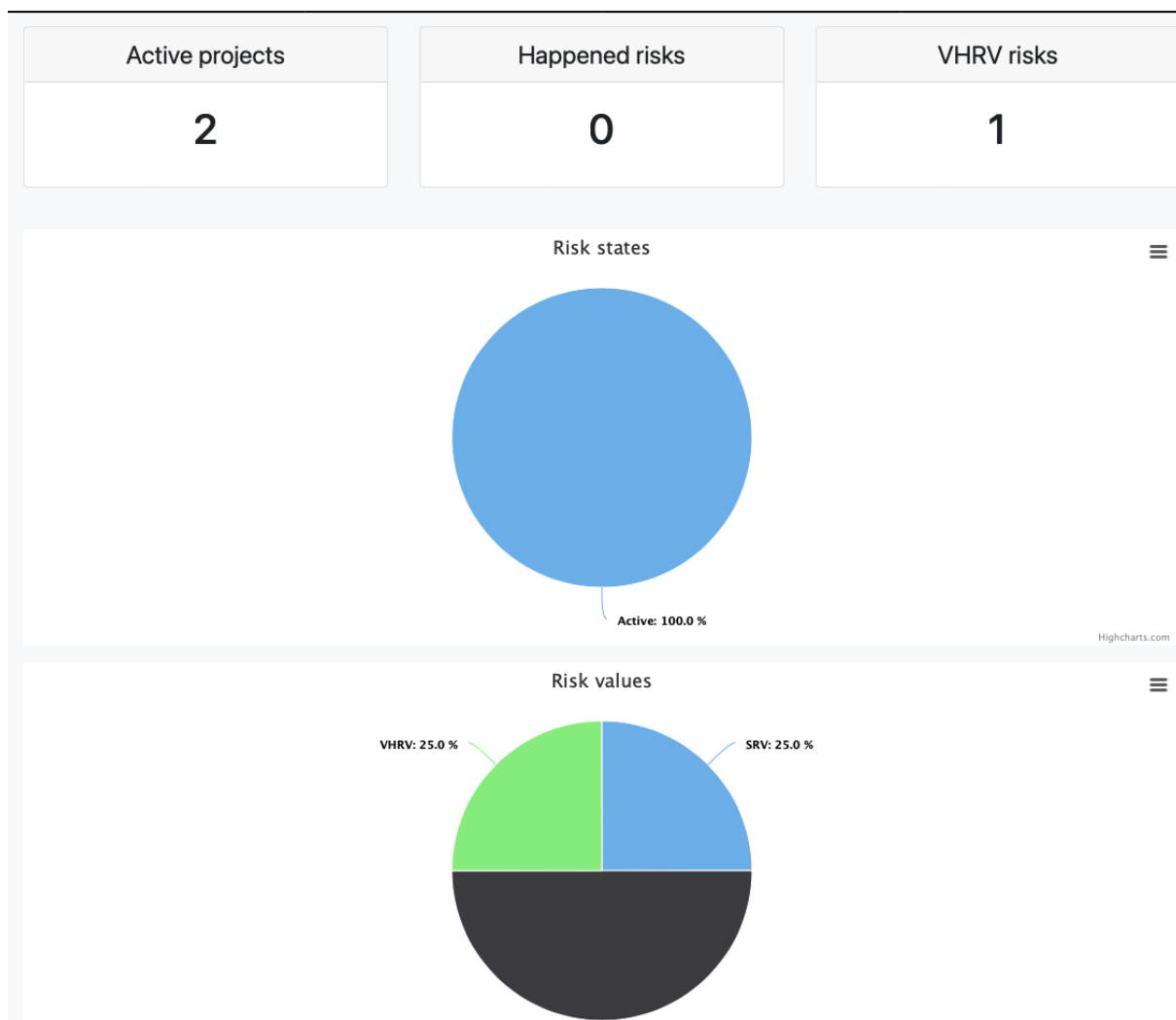
Obr. č. 10 - Aplikace: Registr rizik

7.3.8 Company members (Zaměstnanci společnosti)

V poslední záložce je zobrazen seznam zaměstnanců společnosti spjatých s vykonáváním prací na projektu. Jsou zde zobrazeny základní informace o uživatelích, včetně jejich kontaktů. Uživatele je možné filtrovat na základě jejich role v projektu, případně je vyhledávat přímo podle jména.

7.3.9 Statistika

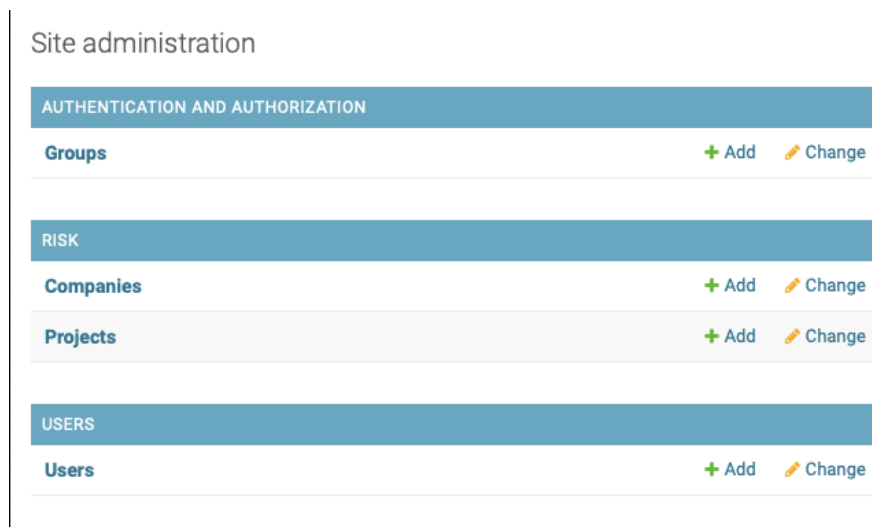
Sekce statistiky je dostupná z hlavního menu aplikace. Jedná se o část programu, která uživatele informuje o počtu aktivních a dokončených projektů, počtu rizik, které se již přihodily a počtu zaznamenaných rizik s nejvyšší mírou rizika.



Obr. č. 11 - Aplikace: Statistika

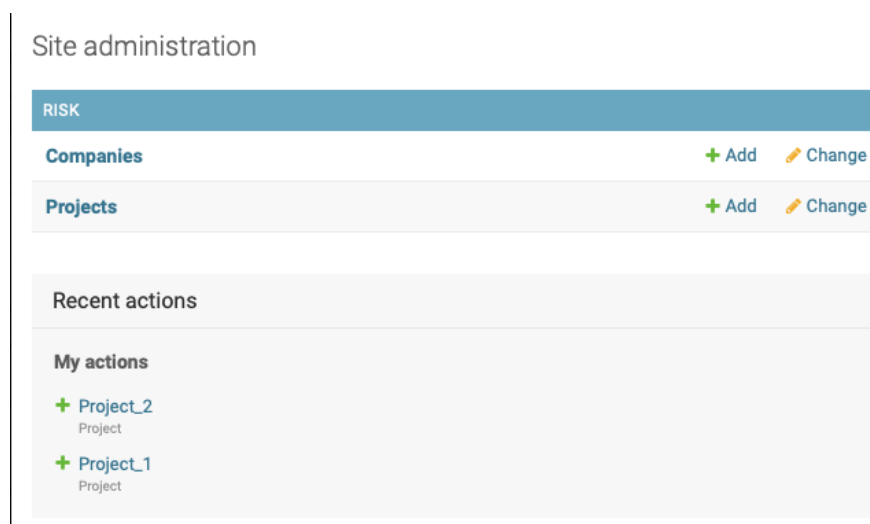
7.3.10 Admin page (Admínská stránka)

Stránka, ze které je přístup ke všem společnostem, projektům a uživatelům, je zobrazena pouze adminovi a z části projektovému manažerovi. V této sekci je možné vytvářet nové uživatelské oprávnění na základě přiřazení jednotlivých aktivit, které může daná skupina uživatelů provádět. Z této strany aplikace je uživatel schopný modifikovat všechny položky zaznamenané v aplikaci napříč různými společnostmi. Admínská stránka je rovněž využívána pro přidání nových uživatelů a společností do aplikace.



Obr. č. 12 - Aplikace: Administrativní přístup pro roli admin

Projektový manažer má v této části programu omezené možnosti, a to přidávat nové projekty a spravovat společnost. V rámci projektů je možné přiřazovat jednotlivé zaměstnance do projektů, určit datum zahájení a konce projektu a přidávat popis.



Obr. č. 13 - Aplikace: Administrativní přístup pro projektového manažera

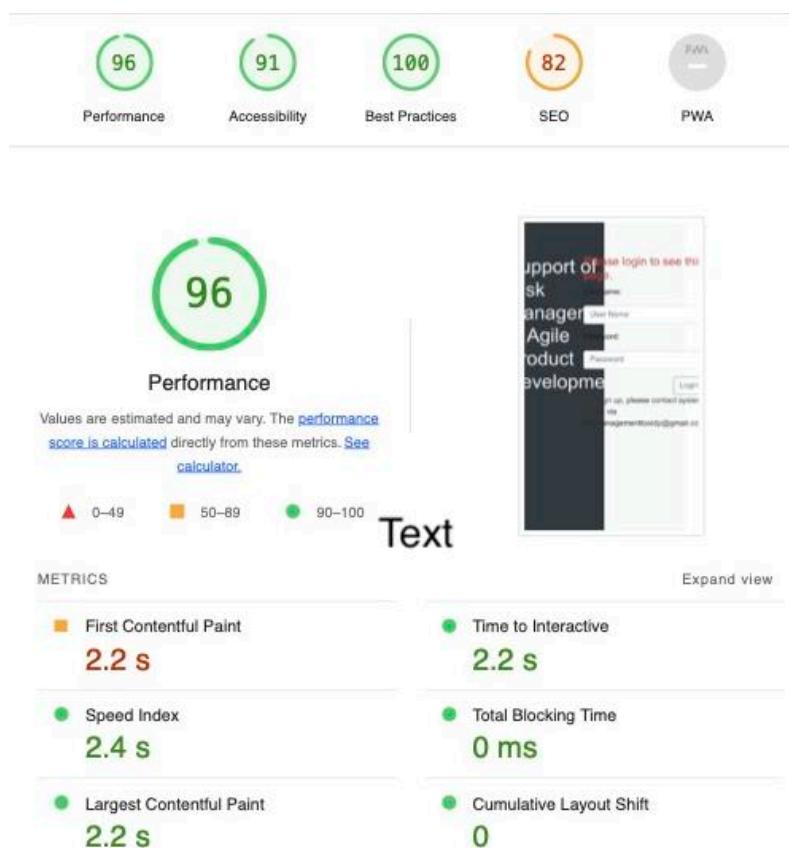
8 VYHODNOCENÍ APLIKACE

Posledním krokem v agilním vývoji aplikace je získání výsledků z testování aplikace a zpětné vazby od zákazníka. Na základě těchto informací je možné rozmýšlet nad novými funkcionalitami, případně zařadit do seznamu problému s aplikací objevené chyby, které je nezbytné vyřešit pro správnou funkcionalitu aplikace.

8.1 TEST VÝKONU APLIKACE

Pro test výkonnosti webové aplikace bylo využito nástroje Google Lighthouse, což je součástí developerského rozšíření pro Google Chrome. Díky zařazení rozšíření do webového prohlížeče je možné testovat libovolnou webovou stránku a zjistit výsledky testů zaměřených na výkonnost, dostupnost a optimalizaci. Výsledky testů jsou rozděleny dle získaných bodů z testů na červené (0-49 bodů), oranžové (50-89 bodů) a zelené (90-100 bodů).

Implementovaná webová aplikace obdržela celkové hodnocení výkonnosti 96 bodů, díky čemuž se dá hodnotit jako stabilní. Při dalším vývoji se však bude nutně zaměřit na zvýšení dostupnosti aplikace pro webové vyhledávače.



Obr. č. 14 - Výsledky testů výkonnosti

8.2 AUTOMATIZOVANÉ TESTY

Automatizované testy byly prováděny pomocí softwaru Cypress, který obstarává testy aplikace a jejich průběh je zaznamenáván. V případě neúspěšného testu je daný moment zaznamenán pomocí snímku obrazovky, který je uložen ve složce projektu. Záznam rovněž případně obsahuje podmínky, za kterých došlo k nežádoucímu chování aplikace.

Aplikace byla pomocí automatizovaných testů otestována na správnou komunikaci a předávání tokenů během přihlašování uživatelů. Zároveň proběhlo testování zasílaných požadavků mezi serverem a uživatelským zařízením, zda http požadavek obsahuje návratovou hodnotu 200. Další testy se zabývají funkcionalitou jednotlivých položek uvnitř aplikace, které byly zaměřeny rovněž na existenci jednotlivých ovládacích prvků aplikace. Veškeré testy program absolvoval bez komplikací.

8.3 UŽIVATELSKÉ TESTOVÁNÍ

Uživatelské testování aplikace proběhlo pomocí simulace řízení projektu a detekce možných vylepšení aplikace. Uživatel zkoušel orientaci v aplikaci a využívání jednotlivých funkcí. Základní orientace probíhala v programu intuitivně. Následně uživatel dostal za úkol vytvořit nový projekt, vyplnit logickou rámcovou matici, navrhnout skladbu projektu z jednotlivých fází a následně určit možná rizika týkající se projektu. Dalším úkolem bylo přiřadit úkoly do fází a jednotlivým úkolům přiřadit také zaměstnance, stejnou činnost měl uživatel vykonat s problémy produktu. Následně byl požádán o přidání nových rizik projektu v závislosti na konkrétní fázi.

Na základě rozhovoru s uživatelem byly získány následující informace o aplikaci. Ovládání aplikace zhodnotil jako intuitivní a jednotlivé prvky byly přehledně zařazeny do aplikace. Nevýhodou zde bylo nutné přidání fáze, která má stejnou dobu trvání jako samotný projekt pro možnost přiřazení rizika do celé doby trvání projektu. Naopak přiřazení rizik do jednotlivých fází hodnotil jako zdůraznění možného problému ve více specifikovaném časovém intervalu. Z pohledu developera však chyběly určité funkcionality, které by znamenaly spojení s verzovacími systémy. Z pohledu osoby řídící projekt by uvítal cloudové uložení souborů a rovněž propojení s verzovacím systémem pro automatickou aktualizaci položek v projektu. Kladně byla hodnocena i metoda pro zobrazení rizik, kde jako uživatel bez větší znalosti analytických metod bez problému vyhodnotil závažnost rizik.

8.4 ROZŠÍŘENÍ

Pro rozšíření aplikace je nutné opět analyzovat požadavky na ni vyvíjené a na základě těchto výsledků následně rozšiřovat funkcionalitu, upravovat uživatelské rozhraní na základě požadavků zákazníka, případně nabídnout možnost propojení s některým dalším programem. Na základě uživatelského testování byla navržena následující vylepšení.

8.4.1 Připojení na verzovací systém

Pro rozšíření aplikace tak, aby ji bylo možné využívat pohodlněji, by bylo potřebné její připojení na verzovací systém, jako je například GitHub a GitLab. Na základě synchronizace s takovým systémem by vznikla automatická kontrola splněných úkolů na projektu. Dále by bylo možné přidávat chyby programu zákazníkem bez možnosti nahlédnutí do kódu aplikace. S připojením na verzovací systém by bylo možné vést konkrétní statistiky ohledně automatizovaných testů a například pomocí modulu Lighthouse v nich rovněž monitorovat výkonnost webové aplikace z pohledu běžného uživatele.

8.4.2 Přidání cloudového úložiště

Nápomocnou funkcionalitou by mohlo být rovněž rozšíření o cloudové úložiště dokumentů, kde by bylo možné uchovávat veškerou projektovou dokumentaci. Přidání těchto dokumentů přímo ke každé části projektu by zajistilo vyšší míru přehlednosti v projektu i případné vracení k některé z částí projektu. To by s sebou neslo úlohu jejich vysokého zabezpečení.

8.4.3 Přidání metod pro analýzu rizik

V současné době je využívána metoda RIPRAN pro analýzu rizik. Do budoucna by bylo vhodné mít možnost volby, jakou metodu pro analýzu využívat, případně je kombinovat. Přidání nových analytických metod by mohlo pomoci k využití aplikace i pro jiné obory, než je IT.

8.4.4 Zapojení umělé inteligence

Připojení umělé inteligence by bylo časově náročné z důvodu sběru potřebného vzorku dat v reálném prostředí. Dalším problémem je zde správná volba algoritmu tak, aby odpovídala charakteru získaných dat. Umělá inteligence by zde měla sloužit pro kontrolu registru rizik a na základě získaných znalostí upozorňovat na možná neodhalená rizika v rámci projektu. V kombinaci s připojením na verzovací systém by mohl být celý projekt tímto způsobem monitorován a upozornit na možné problémy, jako jsou chyby přetížení sítě, přetečení paměti, případně další nedostatky, co se hardwarové oblasti týče.

9 ZÁVĚR

Diplomová práce se v první části zabývala rešerší dle standardu PMI pro řízení projektu v agilním prostředí s hlavním důrazem na management rizik v projektovém prostředí při vývoji produktu.

Následující kapitola se věnovala agilnímu vývoji, zejména popisu jednotlivých kroků procesu a následně navázala výčtem používaných metod. Zmíněné metody vznikly v projektovém prostředí zaměřujícím se na oblast dodávání produktů v IT, což má za cíl usnadnit a urychlit tento proces. Oblast agilních metod se však v posledních letech rozšiřuje i mimo oblast IT, kde je metoda užitečná zvláště díky ušetření financí a rychlé reakci na možné změny.

Na agilní vývoj navazuje kapitola zaměřující se na testování produktu, což je jedním z nejdůležitějších kroků samotného agilního vývoje. Neustálé testování produktu totiž umožňuje držet vždy stabilní verzi produktu a upozorňovat na nedostatky ještě před dodáním produktu zákazníkovi. Výsledky testování výkonnosti aplikace dosahují vysokých hodnot, což znamená, že použité technologie jsou pro aplikaci dostatečné a je možné rozšiřovat funkcionalitu bez velkých zásahů do jádra aplikace.

Následující sekce práce se věnuje požadavkům aplikace pro podporu managementu rizik v agilním vývoji a jejímu návrhu. Na základě získaných znalostí proběhla následně realizace programové aplikace, v níž se odráží prvky agilního prostředí. Aplikace je popsána v kapitole Implementace aplikace a je dostupná na <https://support-risk-management.herokuapp.com>. Pro vývoj aplikace bylo nasimulováno rovněž agilní prostředí, kdy bylo využito verzovacího systému a jeho rozšíření pro kontrolu stability aplikace. Přístup k vytváření aplikace pak probíhal na základě testování funkcionality a vyhodnocení jednotlivých slabých míst, které bylo nutné upravit. Pro testování bylo využíváno především uživatelské testování, které následně bylo rozšířeno o automatizované testy v programové aplikaci Cypress. Dalším ověřením funkcionality bylo rovněž testování uživatelem, který vyvíjený program předem neznal. V posledním kroku jsou diskutovány možné rozšíření, které by měly vést k rozšíření funkcionality celé aplikace. Jako nejpřínosnější rozšíření aplikace bylo vyhodnoceno propojení s verzovacími systémy.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] ČESKÝ NORMALIZAČNÍ INSTITUT. *ČSN ISO 10006 (010333) Management kvality - Směrnice pro management kvality v projektech*. 2019.
- [2] *RIPRAN - Metoda pro analýzu projektových rizik* [online]. [cit. 2022-05-05]. Dostupné z: <https://ripran.cz/>
- [3] PROJECT MANAGEMENT INSTITUTE. *A Guide To The Project Management Body Of Knowledge*. Sixth Edition. 2017.
- [4] PROJECT MANAGEMENT INSTITUTE. *Agile Practice Guide: global standard Project Management Institute*. 2017.
- [5] *What is Agile? | Agile 101 | Agile Alliance* [online]. [cit. 2022-04-05]. Dostupné z: <https://www.agilealliance.org/agile101/>
- [6] *12 Principles Behind the Agile Manifesto | Agile Alliance* [online]. [cit. 2022-01-08]. Dostupné z: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>
- [7] *The Agile Software Development Life Cycle | Object Edge* [online]. [cit. 2022-01-12]. Dostupné z: <https://www.objectedge.com/blog/the-agile-development-life-cycle>
- [8] *What is an Iteration? | Agile Alliance* [online]. [cit. 2022-01-17]. Dostupné z: <https://www.agilealliance.org/glossary/iteration/>
- [9] *Iterative Process in Development & Design | Project-Management.com* [online]. [cit. 2022-02-03]. Dostupné z: <https://project-management.com/iterative-process/>
- [10] *What is a software release?* [online]. [cit. 2022-02-08]. Dostupné z: <https://www.techtarget.com/searchsoftwarequality/definition/release>
- [11] *Je lepší vyvíjet vodopádem nebo scrumem?* [online]. [cit. 2022-02-15]. Dostupné z: <https://www.aitom.cz/co-je-noveho/je-lepsi-vyvijet-vodopadem-nebo-scrumem>
- [12] *What is a Scrum Master? | Atlassian* [online]. [cit. 2022-04-06]. Dostupné z: <https://www.atlassian.com/agile/scrum/scrum-master>
- [13] *Agile Scrum Roles | Atlassian* [online]. [cit. 2022-04-06]. Dostupné z: <https://www.atlassian.com/agile/scrum/roles>
- [14] *Sprint Planning | Atlassian* [online]. [cit. 2022-04-18]. Dostupné z: <https://www.atlassian.com/agile/scrum/sprint-planning>

- [15] *What is Lean Software Development? Definition and Principles* [online]. [cit. 2022-04-24]. Dostupné z: <https://www.techtarget.com/searchsoftwarequality/definition/lean-programming>
- [16] *Guiding Principles of Lean Development | Planview* [online]. [cit. 2022-04-24]. Dostupné z: <https://www.planview.com/resources/articles/lkdc-principles-lean-development/>
- [17] *The Official Guide to The Kanban Method | Kanban University* [online]. [cit. 2022-04-24]. Dostupné z: <https://kanban.university/kanban-guide/>
- [18] *Crystal Agile Framework: Know the Basics | Glossary* [online]. [cit. 2022-04-24]. Dostupné z: <https://chisellabs.com/glossary/what-is-crystal-agile-framework/>
- [19] RICK DAVID CRAIG, Stefan P. Jaskiel. *Systematic Software Testing*. B.m.: Artech House, 2002.
- [20] VINAY, P., F. *Manage Software Testing*. 2008.
- [21] RON PATTON. *Software Testing*. 2nd edition. B.m.: Sams Publishing, 2005.
- [22] DE NICOLA, Giuseppe, Pasquale DI TOMMASO, Esposito ROSARIA, Flammini FRANCESCO, Marmo PIETRO and Orazio ANTONIO. A Grey-Box Approach to the Functional Testing of Complex Automatic Train Protection Systems, 2005, s. 305–317. DOI: 10.1007/11408901_23
- [23] SAWANT, Abhijit A, Pranit H BARI and P M CHAWAN. Software Testing Techniques and Strategies [online]. no date, 2, 980–986 [cit. 2022-05-15]. ISSN: 2248-9622. Dostupné z: www.ijera.com
- [24] SAMPATH, Sreedevi and Renée C BRYCE. Improving the effectiveness of test suite reduction for user-session-based testing of web applications. *Information and Software Technology*. 2012, 54, 724–738. DOI: 10.1016/j.infsof.2012.01.007
- [25] BRYCE, Renée C, Sreedevi SAMPATH, Jan B PEDERSEN and Schuyler MANCHESTER. Test suite prioritization by cost-based combinatorial interaction coverage. DOI: 10.1007/s13198-011-0067-4
- [26] ELFRIEDE DUSTIN, THOMAS GARRETT, BERNIE GAUF, REGINALD H. GARRETT and SAMUEL E. WILSON. *Implementing Automated Software Testing*. 1st edition. 2009.

SEZNAM TABULEK

Tab. č. 1 Obecný přístup k riziku	43
Tab. č. 2 - Logická rámcová matice	62
Tab. č. 3 RIPRAN: Třídy pravděpodobnosti	63
Tab. č. 4 RIPRAN: Třídy dopadu na projekt	63
Tab. č. 5 RIPRAN: Přiřazení hodnoty rizika	63
Tab. č. 6 RIPRAN: Hodnoty rizika.....	63
Tab. č. 7 Uživatelské údaje	65

SEZNAM GRAFŮ

Graf č. 1 - Závislost kvality produktu na počtu testů	57
---	----

SEZNAM OBRÁZKŮ

Obr. č. 1 - Cyklus agilního vývoje produktu.....	50
Obr. č. 2 - Diagram metody Scrum	52
Obr. č. 3 - Agilní procesní metody dle počtu členů týmu	55
Obr. č. 4 - Životní cyklus testování softwaru	58
Obr. č. 5 - Struktura uživatelského rozhraní aplikace.....	66
Obr. č. 6 - Aplikace: Detaily projektu	67
Obr. č. 7 - Aplikace: Logická rámcová matice	68
Obr. č. 8 - Aplikace: Časový plán	69
Obr. č. 9 - Aplikace: Problémy produktu	70
Obr. č. 10 - Aplikace: Registr rizik.....	70
Obr. č. 11 - Aplikace: Statistiky	71
Obr. č. 12 - Aplikace: Administrativní přístup pro roli admin.....	72
Obr. č. 13 - Aplikace: Administrativní přístup pro projektového manažera	72
Obr. č. 14 - Výsledky testů výkonnosti	73

SEZNAM ZKRATEK

PMI Project Management Institute

WBS Work Breakdown Structure

OPA..... Organizační procesní aktiva

SEZNAM PŘÍLOH

Příloha č. 1: Model databáze

Příloha č. 2: Kód programu dostupný rovněž z <https://gitlab.com/Scipy/risk-management-tool>

PŘÍLOHA Č. 1: MODEL DATABÁZE

