

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství



Diplomová práce

Projekt zavedení nového IS ve firmě

Bc. Marek Dolníček

© 2024 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Marek Dolníček

Informatika

Název práce

Projekt zavedení nového IS ve firmě

Název anglicky

Project of Implementation of a New Information System in a Company

Cíle práce

Cílem diplomové práce je analýza problematiky, zhodnocení přínosů a samotné zavedení nového informačního v konkrétním podniku v oblasti projektového řízení.

Díličím cílem je vypracování analýzy, která bude sloužit při procesu zavedení informačního systému v dané firmě se zaměřením na agilní metody řízení projektů.

Metodika

Obsahem teoretické části práce bude charakteristika obecné problematiky projektového řízení, popis informačních systémů a samotná definice projektu.

Větší důraz bude kladen na řízení projektů agilní metodou SCRUM.

Zaměření bude především na projektové řízení v informačních systémech.

Praktická část se bude zabírat zavedením a popsáním informačního systému v dané společnosti.

Výsledkem bude také provedená analýza, která bude vytvořena ještě před zavedením a bude reflektovat potřeby podniku.

V závěru bude zhodnocen výsledek celého projektu a vytvoří se celkový souhrn pozitiv a negativ pro danou firmu.

Doporučený rozsah práce

60-80 stran

Klíčová slova

Informační systém, projekt, projektové řízení, agilní přístup, implementace

Doporučené zdroje informací

DOLEŽAL, J. – MÁCHAL, P. – LACKO, B. – SPOLEČNOST PRO PROJEKTOVÉ ŘÍZENÍ. *Projektový management podle IPMA*. Praha: Grada, 2012. ISBN 978-80-247-4275-5.

SCHWALBE, K. *Řízení projektů v IT : kompletní průvodce*. Brno: Computer Press, 2011. ISBN 978-80-251-2882-4.

SVOZILOVÁ, A. *Projektový management*. Praha: Grada, 2011. ISBN 978-80-247-3611-2.

ŠOCHOVÁ, Z. – KUNCE, E. *Agilní metody řízení projektů*. Brno: Computer Press, 2019. ISBN 978-80-251-4961-4.

TVRDÍKOVÁ, M. – ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI. *Zavádění a inovace informačních systémů ve firmách*. Praha: Grada, 2000. ISBN 80-7169-703-6.

VYMĚTAL, D. *Informační systémy v podnicích : teorie a praxe projektování*. Praha: Grada, 2009. ISBN 978-80-247-3046-2.

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. Martina Houšková Beránková, Ph.D.

Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 16. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 28. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 27. 06. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Projekt zavedení nového IS ve firmě" jsem vypracoval samostatně pod vedením vedoucí diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30. 3. 2024

Poděkování

Rád bych touto cestou poděkoval Ing. Martině Houškové Beránkové, Ph.D. za odborné vedení, poskytnuté konzultace a cenné připomínky při zpracování diplomové práce. Poděkování patří také nejmenované společnosti za poskytnutí důležitých informací a praxe pro zpracování diplomové práce.

Projekt zavedení nového IS ve firmě

Abstrakt

Diplomová práce představuje analýzu procesu implementace systému GitLab ve společnosti zaměřující se na projektové řízení a softwarový vývoj. V teoretické části se práce zabývá definicí projektového řízení, různými moderními metodami řízení projektů, včetně agilních a tradičních přístupů, popisuje informační systémy a poskytuje základ pro praktické nasazení těchto metod. Praktická část se soustřeďuje na analýzu současného stavu projektového řízení, dokumentaci kroků implementace a zkoumá vliv nového systému na firemní procesy. Cílem práce je zdokumentovat celý proces implementace, zhodnotit výhody a případné nevýhody tohoto systému, a to s ohledem na finanční i časové aspekty celého procesu. Dílčím cílem je zanalyzovat jednotlivé fáze implementace a porovnat je podle metodologie. Další důležitou součástí práce je posouzení rizik spojených s implementací a návrhy na jejich prevenci. Výsledkem je sada doporučení pro zlepšení budoucích implementací informačních systémů, stejně jako poskytnutí strategického pohledu na projektové řízení a informační technologie. Práce končí hodnocením celkového úspěchu zavedení systému GitLab a jeho dopadu na cíle organizace.

Klíčová slova: Informační systém, projekt, projektové řízení, agilní přístup, implementace

Project of implementation of a new Information System in a company

Abstract

The thesis presents an analysis of the process of implementing the GitLab system in a company focusing on project management and software development. In the theoretical part, the thesis addresses the definition of project management, various modern project management methods, including agile and traditional approaches, describes modern information systems, and provides a basis for the practical deployment of these methods. The practical part concentrates on the analysis of the current state of project management, documentation of the implementation steps, and examines the impact of the new system on company processes. The goal of the thesis is to document the entire implementation process, evaluate the advantages and potential disadvantages of this system, considering the financial and time aspects of the entire process. A subsidiary goal is to analyze the individual phases of implementation and compare them according to methodology. Another important part of the thesis is the assessment of risks associated with implementation and suggestions for their prevention. The result is a set of recommendations for improving future implementations of information systems, as well as providing a strategic view on project management and information technologies. The thesis concludes with an evaluation of the overall success of the GitLab system implementation and its impact on the organization's goals.

Keywords: Information System, Project, Project Management, Agile Approach, Implementation

Obsah

1 Úvod.....	10
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	13
3 Teoretická východiska	15
3.1 Projektové řízení	15
3.1.1 Historie a vývoj projektového řízení	15
3.1.2 Standardy a metodologie projektového řízení	16
3.2 Definice klíčových pojmů	17
3.2.1 Projekt	17
3.2.2 Trojimperativ projektu	17
3.2.3 Životní cyklus projektu	19
3.2.4 Projektový manažer	21
3.2.5 Metodologie, metodika a metoda	21
3.3 Porovnání metodologií v projektovém řízení	22
3.3.1 Waterfall metodologie	22
3.3.2 Agilní metodologie	24
3.3.3 Klíčové rozdíly	26
3.4 Rozbor standardů projektového řízení	27
3.4.1 PRINCE2	27
3.4.2 PMBOK	30
3.4.3 IPMA	31
3.4.4 Agilní metodika Scrum	33
3.5 Srovnání a kontrast standardů a metodik	35
3.5.1 Porovnání PRINCE2, PMBOK a Scrum	36
3.5.2 Výhody a nevýhody jednotlivých přístupů	36
3.5.3 Porovnání standardů v tabulce	37
3.5.4 Možnosti kombinování různých přístupů	38
3.6 Informační systémy a jejich aplikace v podnikání	38
3.6.1 Definice informačního systému	38
3.6.2 Struktura informačního systému	39
3.6.3 Význam podnikových informačních systémů	40
3.6.4 Klasifikace podle organizační úrovně	40
3.6.5 Typy podnikových informačních systémů	41
4 Vlastní práce	44

4.1	Úvod do problematiky	44
4.2	Charakteristika společnosti	45
4.3	Struktura společnosti	45
4.3.1	Vedení firmy	45
4.3.2	Oddělení programátorů	46
4.3.3	Oddělení designu	46
4.3.4	Obchodní a finanční oddělení	46
4.3.5	Projektové oddělení	46
4.3.6	Marketingové oddělení	46
4.4	Popis zaváděného informačního systému	47
4.5	Důvody výběru zaváděného systému pro daný podnik.....	48
4.6	Fáze zavedení systému v podniku.....	50
4.6.1	Přidělení jednotlivých procesů při zavádění systému.....	51
4.7	Fáze zavádění systému	53
4.7.1	Přípravná fáze	53
4.7.2	Fáze návrhu a vývoje	59
4.7.3	Implementační fáze.....	64
4.7.4	Fáze testování	68
4.7.5	Závěrečná fáze	71
4.8	Finanční a časový harmonogram implementace nového systému	73
4.8.1	Přípravná fáze	73
4.8.2	Fáze návrhu a vývoje	74
4.8.3	Implementační fáze.....	75
4.8.4	Fáze testování	78
4.8.5	Závěrečná fáze	79
4.9	Porovnání metodologie během procesu zavedení systému	83
4.9.1	Waterfall prvky	84
4.9.2	Agilní prvky	84
4.9.3	Doporučení pro zlepšení agilní praxe.....	85
4.10	Rizika při zavádění systému.....	86
5	Závěr	90
6	Seznam použitých zdrojů	92
	Seznam obrázků, tabulek, grafů a zkratk	95
6.1	Seznam obrázků	95
6.2	Seznam tabulek	96
7	Přílohy	97

1 Úvod

V dnešní době je projektové řízení klíčovým prvkem úspěšnosti jakékoliv organizace. Projektové řízení je disciplína, která se zabývá plánováním, organizací, zabezpečením a řízením zdrojů s cílem úspěšně dokončit konkrétní cíle a dosáhnout cílů projektu. Tato práce se zaměřuje na analýzu a implementaci nového informačního systému ve firmě s důrazem na agilní metody řízení projektů a zabývá se nejen teoretickým rozbohem a srovnáváním projektových standardů, ale také praktickou aplikací těchto metodik při implementaci GitLabu do firemního prostředí.

Tato práce bude využívat teorie a postupy v oblasti projektového řízení v praktické aplikaci, která bude spočívat v implementaci softwarového nástroje v komerční organizaci. Může být užitečná pro projektové manažery a další profesionály v oblasti projektového řízení, kteří se snaží vybrat nejvhodnější metodiku pro své specifické potřeby a kontext, a pro ty, kteří se zabývají implementací GitLabu v jejich organizacích.

V současném, rychle se rozvíjejícím obchodním světě, čelí organizace stále komplexnějším projektům a vyšším nárokům na rychlost dodání a flexibilitu. Projektové řízení se stává klíčovým faktorem pro dosažení konkurenční výhody a úspěšnosti na trhu. Tato práce si klade za cíl analyzovat, jak mohou organizace využít agilní metody řízení projektů a standardní metodiky k dosažení optimálního výsledku v podmínkách moderního podnikání.

Implementace informačního systému GitLab do firemního prostředí přináší nové možnosti a výzvy. GitLab je známý svými nástroji pro správu verzí, kontinuální integraci a dodávku softwaru, což může výrazně zvýšit produktivitu vývojových týmů. Tato práce se zaměřuje na využití GitLabu jako konkrétního softwarového nástroje pro demonstraci aplikace projektových metodik.

Výsledky této práce mají praktický přínos pro organizace, které uvažují o implementaci informačního systému podobného GitLabu nebo o vylepšení svého projektového řízení. Analýza a srovnání projektových metodik a jejich aplikace při implementaci GitLabu poskytne cenné poznatky pro projektové manažery a týmy. Zároveň může tato práce sloužit jako základ pro budoucí výzkum a strategická rozhodnutí v oblasti projektového řízení a softwarového vývoje.

Tato práce je rozdělena do několika částí, které postupně prozkoumávají teoretické základy projektového řízení, standardní metodiky jako jsou PRINCE2 či PMBOK, agilní metodiku Scrum, popis informačních systémů a popis implementace informačního systému GitLab do reálného firemního prostředí. Každá část přináší specifický pohled na problematiku projektového řízení a jeho aplikaci v praxi.

Specifickým přínosem této práce je tak podrobný přehled o využití GitLabu jako platformy, která umožňuje integraci různých fází vývoje softwaru a podporuje kolaborativní práci týmů. Zabývá se zde možnostmi, které GitLab nabízí v rámci správy projektů. Od sledování chyb, tvorby dokumentace až po automatizaci testování a nasazení systému na produkční servery a jak tyto funkce mohou pomoci zlepšit efektivitu projektového řízení. Tento aspekt je klíčový pro pochopení, jak moderní nástroje a platformy mohou přinášet značné výhody ve flexibilitě a rychlosti dodávek, což je nezbytné pro úspěch v dynamickém a konkurenčním prostředí.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této diplomové práce je prozkoumat a zhodnotit proces implementace nového informačního systému ve vybrané organizaci, která působí v oblasti projektového řízení a vývoje software. Záměrem je především poskytnout náhled na metodiky, které jsou uplatňovány při řízení projektů. Práce si klade za cíl zanalyzovat a definovat teoretické rámce a principy projektového řízení, včetně standardů a postupů životního cyklu projektu. Současně bude práce směřovat k podrobnému popisu a klasifikaci informačních systémů, jejich typů a charakteristik, aby byl poskytnut přehled o těchto systémech v moderním řízení podniku.

V praktické části se diplomová práce zaměří na analýzu aktuálního stavu projektového řízení ve zvolené společnosti, včetně představení nově implementovaného informačního systému. Hlavním cílem je zdokumentovat a vyhodnotit, jaké výhody či nevýhody tento systém přinesl a jak ovlivnil běžné operace podniku. Důraz bude kladen na proces implementace, jehož jednotlivé fáze budou popsány a zkoumány s ohledem na dosažení stanovených cílů.

Jako jeden z dílčích cílů této práce rovněž nabídne srovnání použitých projektových metodologií, zejména rozdíly mezi agilními přístupy a tradičním Waterfall modelem řízení.

Zároveň si práce klade za další dílčí cíl vytvořit komplexní analýzu rizik spojených s procesem zavedení nového informačního systému, a to včetně návrhů na jejich prevenci a řízení. Výstupem bude doporučení a strategie pro zlepšení budoucích implementací a pro další rozvoj informačních systémů v rámci organizace.

V konečné fázi práce bude provedeno zhodnocení celého procesu zavádění nového systému s cílem určit, zda byly splněny počáteční předpoklady projektu a zdokumentovat celý tento proces implementace.

2.2 Metodika

Diplomová práce bude zahájena literární rešerší, která zahrnuje shromažďování a analýzu relevantních odborných publikací a studií zabývajících se projektovým řízením, a to zejména s ohledem na agilní a tradiční metodiky. Bude provedeno důkladné posouzení současného stavu informačního systému ve firmě, včetně sběru dat a analýzy interních procesů. Toto hodnocení bude zásadní pro identifikaci potenciálních silných a slabých stránek, které by mohly ovlivnit proces zavádění nového systému.

V další fázi bude následovat implementace a hodnocení, při kterém budou všechny kroky nového zavedení pečlivě monitorovány a vyhodnoceny. Podstatná bude také fáze srovnání různých metodik projektového řízení a jejich uplatnění v praxi. Pro identifikaci a řízení potenciálních rizik bude vytvořen registr rizik a závěrečné hodnocení poskytne základ pro zhodnocení celkového přínosu a efektivity nově implementovaných řešení.

1. Literární rešerše:

- Shromáždění relevantních odborných publikací týkajících se projektového řízení.
- Analýza metodologií a jejich uplatnění ve vybraném podniku.

2. Úvod do problematiky:

- Charakteristika a struktura společnosti.
- Popis zaváděného systému.

3. Analýza současného stavu:

- Posouzení stávajícího informačního systému ve firmě.
- Sběr dat o aktuálních pracovních procesech.
- Identifikace klíčových silných a slabých stránek systému.

4. Implementace a hodnocení:

- Vypracování strategie pro zavedení nového informačního systému.
- Finanční a technický návrh implementace.
- Postupná aplikace nového systému v praxi.
- Závěrečná zpráva a hodnocení implementace nového systému.

5. **Srovnání metodik projektového řízení:**

- Vyhodnocení jednotlivých fází podle metodologie projektového řízení.
- Srovnání Waterfall a agilních přístupů v rámci implementace.

6. **Registr rizik:**

- Systematická identifikace potenciálních rizik při zavádění systému.
- Vypracování plánu pro řešení dopadu a korekce rizik.

7. **Závěrečné hodnocení:**

- Zhodnocení dosažení projektových cílů a celkové úspěšnosti implementace.

3 Teoretická východiska

3.1 Projektové řízení

Projektové řízení, jako disciplína zajišťující efektivní dosažení cílů projektu skrze plánování, organizování a řízení různých činností a zdrojů, se stalo nezbytnou součástí moderního managementu. Tento obor, který nabral na významu zejména po druhé světové válce, je dnes považován za klíčový pro úspěšnou implementaci jakéhokoli projektu v různých odvětvích a průmyslech (Doležal a kol., 2012).

Existuje několik různých přístupů k projektovému řízení, které se liší na základě specifik projektu a potřeb organizace. Všechny tyto přístupy mají společný cíl. Zajistit, aby projekt byl dokončen úspěšně a efektivně, s ohledem na kvalitu, rozpočet a časový rámec. V dnešní době je projektové řízení stále častěji považováno za klíčovou součást strategického plánování organizací, které chtějí být konkurenceschopné a adaptabilní v rychle se měnícím obchodním prostředí (Svozilová, 2011).

Význam projektového řízení narůstá s rostoucí složitostí podnikových procesů a globalizací trhu. Moderní projektové řízení se neomezuje pouze na tradiční metody, ale zahrnuje i nové přístupy a technologie, které podporují inovace, spolupráci a efektivní komunikaci mezi týmy a zainteresovanými stranami. Tento vývoj vyžaduje od manažerů projektů nejen technické dovednosti, ale i schopnost adaptace a kreativního řešení problémů v dynamickém a někdy nejistém prostředí (Svozilová, 2011).

3.1.1 Historie a vývoj projektového řízení

Počátky projektového řízení lze vystopovat až do starověku, kdy se začaly využívat první techniky a nástroje pro řízení komplexních stavebních projektů. V moderním pojetí se však projektové řízení začalo vyvíjet v průběhu 20. století, kdy se stalo zásadní součástí řízení a plánování v mnoha průmyslových odvětvích. Během tohoto období vznikly také první metodologie a standardy projektového řízení, které dnes tvoří základ této disciplíny (Kerzner, 2017).

V rámci moderního vývoje projektového řízení vytvořil Henry Gantt Ganttův diagram.

Tento nástroj pro plánování a sledování průběhu projektů se rychle ujal a stal se základem

pro další vývoj v oblasti řízení projektů. Ganttův diagram dokázal vizualizovat časový plán projektu a byl klíčovým nástrojem pro koordinaci práce a zdrojů (Fiala, 2004).

Rozkvět projektového řízení v padesátých a šedesátých letech 20. století byl povzbuzován potřebami kosmického programu a obranného sektoru. V této době byly vyvinuty klíčové metody jako CPM (Critical Path Method) a PERT (Program Evaluation and Review Technique), které umožnily efektivněji plánovat, sledovat a řídit složité projekty. Tyto metody přinesly nové možnosti v grafickém znázorňování projektů a staly se základem pro další inovace v oblasti projektového managementu (Fiala, 2004).

V sedmdesátých letech se projektový management začal více integrovat do akademické sféry a postupně se rozšiřoval do různých průmyslových odvětví. Toto období přineslo další rozvoj metod a nástrojů pro řízení projektů, čímž se rozšířila aplikace projektového managementu mimo tradiční inženýrské a stavební projekty (Hrazdilová, 2016).

V osmdesátých letech dochází k další profesionalizaci a institucionalizaci projektového řízení. Vznik potřeby komplexního projektového řízení vedl k rozvoji softwarových nástrojů a vytváření metodik, které umožnily lépe řídit projekty a dosahovat plánovaných cílů v rámci daného rozpočtu a časového harmonogramu. (Hrazdilová, 2016)

Devadesátá léta přinesla revoluci v oblasti informačních technologií, která měla značný vliv na projektové řízení. V této době se začala využívat metoda TQM (Total Quality Management), která klade důraz na kvalitu a zákaznickou spokojenost. Rozvoj IT a komunikačních technologií umožnil lepší spolupráci týmů a efektivnější řízení projektů (Fiala, 2004).

3.1.2 Standardy a metodologie projektového řízení

V rámci projektového řízení existuje řada standardů a metodologií, které slouží jako rámcové přístupy pro řízení projektů. Patří sem například standardy vydané Project Management Institute (PMI), jako je Project Management Body of Knowledge (PMBOK), nebo metodologie PRINCE2 vyvinutá britskou vládou. Tyto standardy a metodologie poskytují soubor osvědčených postupů, nástrojů a technik, které mohou pomoci projektovým manažerům efektivně řídit projekty a dosáhnout jejich cílů (Kerzner, 2017; Axelos, 2017).

3.2 Definice klíčových pojmů

3.2.1 Projekt

Projekt je základním prvkem projektového řízení, který může být definován jako „časově omezené úsilí vynaložené na vytvoření unikátního produktu, služby nebo výstupu.“ Tato definice zdůrazňuje dočasnou povahu projektu, který má jasně stanovený začátek a konec, a odlišuje se od běžných provozních činností, které jsou zaměřeny na udržení chodu firmy (Schwalbe, 2011).

Jedinečnost projektu spočívá v tom, že i když může být podobný jiným projektům realizovaným v minulosti, situace a specifické okolnosti se liší, což znamená, že každý projekt je svým způsobem originální. Kromě toho projekt charakterizuje nejistota výsledku, což naznačuje, že dosažení cíle projektu může být komplikované a je obklopeno určitými riziky (Schwalbe, 2011).

Další klíčové aspekty projektu zahrnují postupné rozpracovávání, nutnost zdrojů z různých oblastí a význam mít primárního zákazníka nebo sponzora, který projekt podporuje a směřuje k jeho úspěšnému dokončení (Schwalbe, 2011).

Podle PMI (Project Management Institute) je projekt chápán jako „dočasné úsilí vynaložené na vytvoření unikátního produktu, služby nebo určitého výsledku.“ Zde je důraz kladen na dočasnost, která je definována jako časový rámec projektu a na unikátnost, což se týká specifik cílů projektu (Svozilová, 2011).

Cíl projektu je pojímán jako jedinečný výsledek, kterým projekt usiluje o dosažení. Měl by být specificky formulován tak, aby byl snadno měřitelný a kontrolovatelný. Cíl projektu slouží jako základní stavební kámen pro plánování a realizaci projektu, a proto je zásadní jeho důkladná formulace a dohoda mezi všemi zúčastněnými stranami (Doležal a kol., 2012).

3.2.2 Trojimperativ projektu

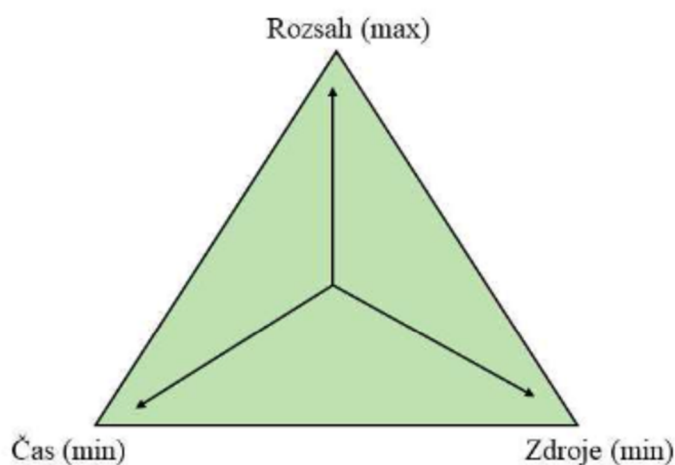
Každý projekt se musí vyrovnat s nevyhnutelnými omezeními, které jsou známy jako trojimperativ projektu. Tento koncept se zaměřuje na tři základní aspekty: rozsah, náklady a čas. Projektový manažer musí najít optimální rovnováhu mezi těmito třemi prvky, aby zajistil úspěšné dokončení projektu. Vzájemná propojenost těchto aspektů znamená, že

jakákoli změna v jednom z nich nevyhnutelně ovlivní dva ostatní. Příkladem je, že pokud se zvýší rozsah projektu bez změny časového plánu nebo rozpočtu, může to vést k překročení nákladů nebo k nedodržení termínů (Schwalbe, 2011).

Tato dynamická rovnováha mezi rozsahem, náklady a časem je často znázorněna ve formě trojúhelníku (viz obr. 1), což usnadňuje vizualizaci vztahu mezi těmito klíčovými faktory projektu. Rozšířená verze tohoto modelu zahrnuje kvalitu jako čtvrtý rozměr, který přidává další úroveň složitosti do rovnice projektového řízení. Kvalita výstupů se stává klíčovým prvkem, který musí být zvážen společně s tradičními omezeními trojimperativu (Doležal a kol., 2016).

1. **Zdroje (náklady)** – představují finanční rozpočet potřebný k dokončení projektu. Řízení nákladů zahrnuje odhad počátečních výdajů a jejich sledování, aby projekt zůstal v rámci rozpočtu.
2. **Čas** – definuje dobu potřebnou k dokončení všech úkolů projektu a dosažení jeho cílů. Efektivní řízení času vyžaduje plánování milníků a dodržení časového harmonogramu.
3. **Rozsah** – určuje, jaké práce budou provedeny a očekává se unikátní výrobek nebo služba. Je to základ pro definici obsahu projektu a jeho očekávaných výstupů (Schwalbe, 2011).

Obrázek 1 – Trojimperativ projektu



Zdroj: Doležal a kol., 2016

3.2.3 Životní cyklus projektu

Životní cyklus projektu je koncipován jako sada definovaných fází, které odpovídají specifickým potřebám kontrolních procesů v organizaci realizující projekt. Tyto fáze jsou navrženy tak, aby reflektovaly charakter, rozsah a cíle projektu, čímž umožňují efektivnější monitorování, řízení a kontrolu nad jeho průběhem. Toto členění na jednotlivé etapy usnadňuje zapojení relevantních zainteresovaných osob a zvyšuje šance na úspěšné dokončení projektu tím, že poskytuje jasný rámec pro sledování pokroku a řešení případných problémů (Svozilová, 2011).

Každá fáze projektu má přesně vymezené cíle, očekávané výstupy, metody jejich ověření, hodnocení a identifikuje klíčové účastníky a jejich role. Tato struktura zajišťuje, že všechny aktivity jsou řádně plánovány a prováděny v souladu s celkovými cíli projektu, což napomáhá efektivnímu využívání zdrojů a optimalizaci pracovních procesů. Tento způsob organizace projektu tak představuje základ pro jeho úspěšné řízení a realizaci (Svozilová, 2011).

Životní cyklus projektu popisuje jednotlivé fáze, kterými projekt prochází od jeho zahájení až po jeho ukončení. Tyto fáze umožňují týmu strukturovat práci, řídit rizika a optimalizovat výsledky. Každá fáze má svá specifika, úkoly a očekávané výstupy (Kerzner 2017; PMI 2017):

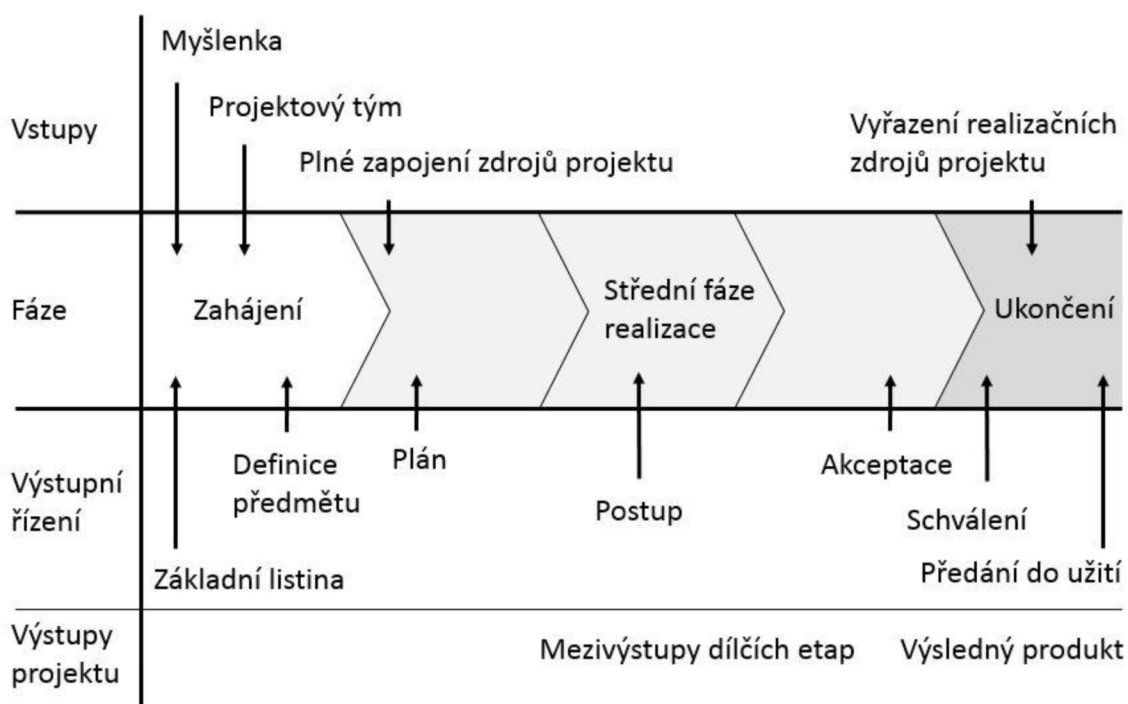
1. **Zahájení:** Tato fáze projektu zahrnuje identifikaci potřeb a stanovení cílů projektu. Je zde také třeba připravit potenciální tým a stanovit strategii realizace. V této fázi se klade důraz na systémový přístup k projektování a jsou zde prováděny počáteční analýzy, jako je hodnocení rizik a odhad požadavků na zdroje (Fiala, 2004).
2. **Plánování:** Během plánovací fáze je vytvořen detailní plán projektu, který zahrnuje definici cíle, rozdělení projektu na jednotlivé činnosti, odhady doby realizace a požadavky na zdroje. Důležitou součástí je také identifikace potenciálních rizik a jejich řízení. Tato fáze často využívá nástroje jako Ganttův diagram pro vizualizaci plánu a jeho časového rozvržení (Fiala, 2004).
3. **Střední fáze realizace:** V realizační fázi je kladen důraz na řízení projektu podle stanoveného plánu a monitorování jakékoliv odchylky od tohoto plánu. Zde dochází k implementaci projektu a je nezbytné provádět kontroly a případné korektivní akce,

aby bylo zajištěno dosažení cílů projektu v rámci daných parametrů kvality, času a rozpočtu (Fiala, 2004).

4. **Ukončení:** Poslední fáze zahrnuje ukončení projektu, testování a ověření, zda byly splněny všechny stanovené cíle. Po úspěšném testování a případných opravách je projekt předán do produktivního užívání. Tato fáze také zahrnuje vyhodnocení projektu a shromažďování zkušeností, které mohou být použity pro zlepšení budoucích projektů (Fiala, 2004).

V každé fázi životního cyklu projektu (viz obr. 2) je důležité zajistit, aby všechny aktivity byly provedeny správně a v souladu s plánem. Při správném řízení projektu a dodržování jeho životního cyklu je mnohem pravděpodobnější, že projekt dosáhne svých cílů a bude považován za úspěšný (Kerzner 2017; PMI 2017).

Obrázek 2 – Životní cyklus projektu



Zdroj: Svozilová, 2011

3.2.4 Projektový manažer

Úspěch v projektovém řízení začíná výběrem správného projektového manažera. Ideální kandidát by měl mít bohaté zkušenosti z předchozích projektů a hluboké znalosti v oblasti řízení projektů. Klíčové aktivity zahrnují odpovědnost za celkový projekt, plánování a koordinaci práce, vedení týmu, řešení problémů a udržování komunikace na všech úrovních. Pro efektivní řízení je důležité mít schopnosti v oblasti řízení lidských zdrojů, zatímco technické detaily mohou být přenechány ostatním členům týmu (Schwalbe, 2011).

Projektový manažer má na starosti nejen vedení projektu od začátku do konce, ale také zajišťuje, že jsou všechny úkoly a aktivity prováděny efektivně a v souladu s plánem. Tato role zahrnuje tvorbu a udržování plánů, obsazení odborných pozic a koordinaci úkolů k dosažení projektových cílů. Projektový manažer tedy musí být schopen pracovat s různorodými týmy a udržovat projekt v souladu s jeho cíli a rozpočtem (Svozilová, 2011).

3.2.5 Metodologie, metodika a metoda

Metodologie výzkumu je základem každého vědeckého bádání, poskytuje systémový rámec pro sběr a analýzu dat. Ve výzkumu je metodologie klíčová pro určení celkového přístupu k výzkumnému projektu, včetně teoretického základu pro výběr metody sběru a analýzy dat. To umožňuje výzkumníkům přistupovat k datům systematicky, což zvyšuje validitu a hodnověrnost jejich zjištění. Metodologie nejenže pomáhá určit, jaké metody budou použity, ale také proč byly vybrány a jak budou interpretovány výsledky (Creswell, 2017).

Metodika je aplikací metodologie ve výzkumu a zahrnuje specifické postupy, které jsou použity k sběru a analýze dat. Detailní vysvětlení metodiky je nezbytné pro pochopení, jak výzkumník přistupuje k datům a jak zajišťuje jejich objektivní a systematické zpracování. Výběr vhodné metodiky je závislý na cílech výzkumu a povaze zkoumaného problému (Creswell, 2017).

Metoda představuje konkrétní techniky sběru a analýzy dat v rámci vybrané metodiky. Je nutné rozlišit mezi různými metodami sběru dat, jako jsou rozhovory, pozorování, dotazníky a analýza dokumentů, a nabízí přehled jejich vhodnosti pro různé typy výzkumných otázek. Každá metoda má svá specifika, která je třeba zvážit při jejím výběru,

včetně toho, jak budou data shromážděna, zaznamenána a analyzována. Důležitá je také reflexe omezení každé metody a jejího vlivu na výsledky výzkumu (Creswell, 2017).

3.3 Porovnání metodologií v projektovém řízení

V softwarovém inženýrství a projektovém řízení bylo za poslední dekády vyvinuto mnoho metodologií. Waterfall, považovaný za tradiční model, a agilní metodologie, které nabízejí nový přístup k projektovému řízení, jsou dva protiklady v této široké škále. Porozumění těmto dvěma přístupům je klíčové pro jakékoli organizace či týmy, které se snaží vybrat nejvhodnější strategii pro své projekty (Kerzner, 2017).

3.3.1 Waterfall metodologie

Definice a struktura:

Na počátku 70. let 20. století Winston Royce představil tradiční procesní přímý model, známý také jako Waterfall model. Tento přístup k řízení projektů se vyznačuje sérií postupně navazujících činností, kde projekt postupuje skrze definované fáze od začátku až do konce. Waterfall model byl značným pokrokem ve své době, jelikož umožnil systematické rozdělení vývojového procesu na jednotlivé, logicky seřazené fáze. Každá fáze je zakončena vyhodnocením, které rozhoduje o přechodu do následující fáze. Pokud výstup z dané fáze nesplňuje stanovené požadavky, není možné projekt posunout dále (Bruckner a kol., 2012; Šilhavý a kol., 2013).

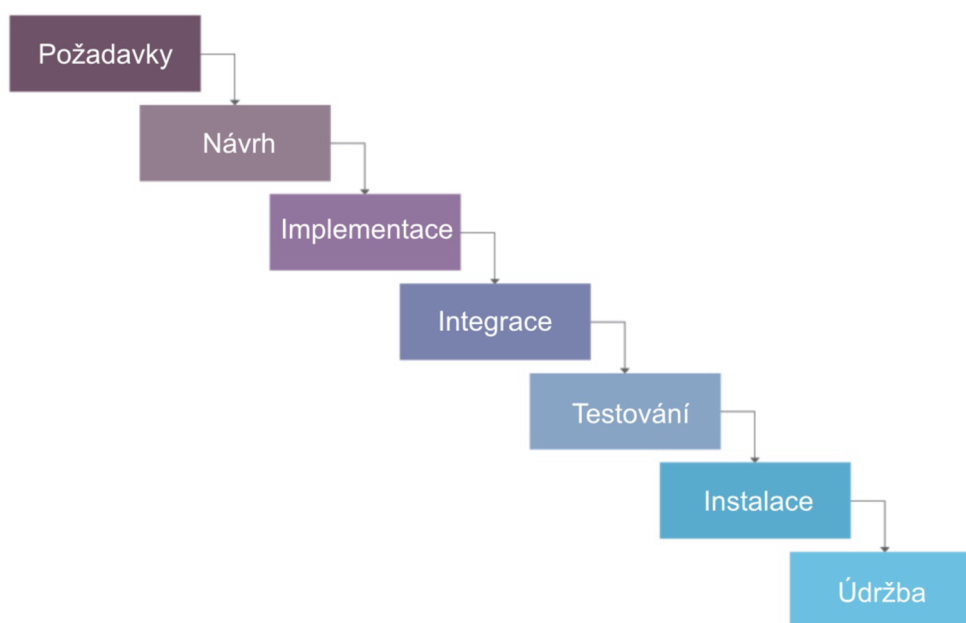
Tradiční přístupy k řízení projektů, jako je Waterfall model, klade velký důraz na pečlivé plánování na začátku projektu a na detailní řízení všech následujících aktivit. Tento model je ideální pro projekty, kde je možné na začátku jasně specifikovat výsledný cíl a kde je zásadní dobře naplánovat všechny kroky a jejich vzájemné závislosti (viz obr. 3). Waterfall model vyžaduje detailně zpracovaný plán a jasně definované cíle projektu (Bruckner a kol., 2012; Šilhavý a kol., 2013).

Projekt v rámci Waterfall modelu prochází sedmi základními fázemi:

1. **Specifikace požadavků** – tato úvodní fáze definuje cíle projektu a určuje, jaké potřeby a požadavky mají být projektovým výstupem naplněny. Výsledkem jsou základní informace o zákazníkovi a jasně formulované požadavky.

2. **Návrh** – ve druhé fázi dojde k analýze požadavků a k posouzení realizovatelnosti projektu. Je vypracován detailní popis systému a jeho funkcionalit, které jsou následně schváleny.
3. **Implementace** – během této fáze probíhá vlastní vývoj produktu podle předchozích specifikací a návrhů.
4. **Integrace** – integrace zahrnuje testování spojení jednotlivých částí produktu s cílem identifikovat a odstranit chyby.
5. **Testování** – v této fázi je produkt testován na straně zákazníka, aby se ověřila jeho funkčnost a splnění specifikovaných požadavků.
6. **Instalace** – po úspěšném testování následuje fáze instalace, kde je produkt implementován do reálného prostředí.
7. **Údržba** – poslední fáze zahrnuje průběžnou údržbu a aktualizace produktu, aby zůstal funkční a odpovídal aktuálním potřebám a technologickému vývoji (Bruckner a kol., 2012; Šilhavý a kol., 2013).

Obrázek 3 – Waterfall metodologie



Zdroj: Vlastní zpracování

Výhody

- **Předvídatelnost** – s jasně definovanými fázemi a kontrolními body mohou manažeři projektu snadno sledovat pokrok a zajišťovat, že všechny požadavky jsou splněny.
- **Dokumentace** – vzhledem k důrazu na počáteční specifikace a plánování, jsou projekty Waterfall často dobře zdokumentovány, což usnadňuje revizi a audit (Kerzner, 2017).

Nevýhody

- **Změny** – Waterfall model může narazit na problémy při nedostatečné specifikaci požadavků na začátku projektu nebo při potřebě častých změn během vývoje, což vede k přepracování, zpoždění a zvýšení nákladů projektu.
- **Riziko** – omezené zapojení zákazníka během vývojového procesu s interakcí pouze na začátku a na konci projektu může vést k nesplnění jeho potřeb a požadavků, což snižuje kontrolu zákazníka nad průběhem a výsledkem projektu (Bruckner a kol., 2012).

3.3.2 Agilní metodologie

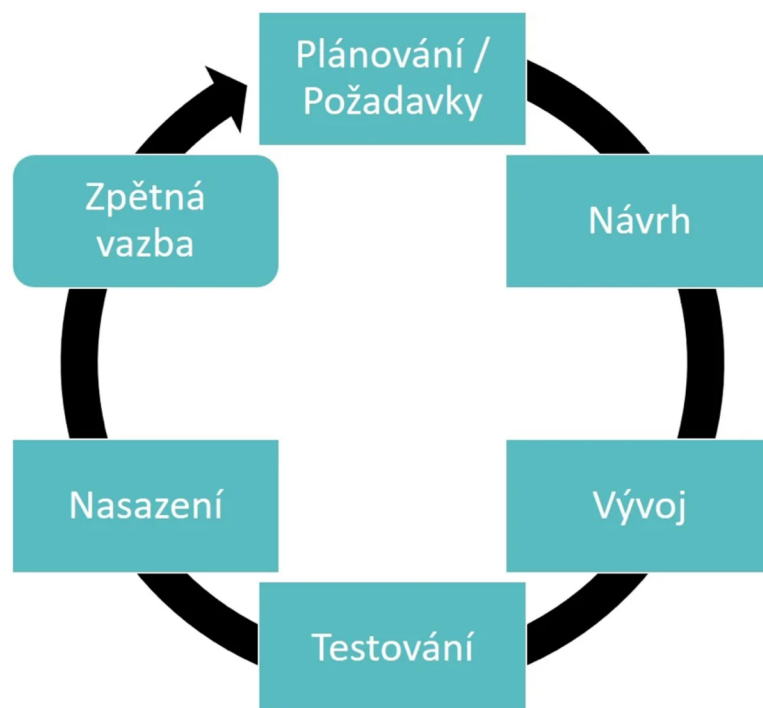
Definice a struktura

Základy agilního přístupu jsou vyjádřeny v Manifestu Agilního vývoje softwaru z roku 2001, který zdůrazňuje hodnoty jako preferenci jednotlivců a interakcí před procesy a nástroji, fungující software před vyčerpávající dokumentací, spolupráci se zákazníkem před vyjednáváním o smlouvách a flexibilitu v reakci na změny před pevným dodržováním plánu. Tyto principy odrážejí důraz na praktické aspekty vývoje a zájem o dosažení nejlepších možných výsledků v reálném čase (Agile Manifesto, 2001).

Agilní metodiky, vycházející z tohoto manifestu, jsou charakterizovány principy, které podporují pružné řízení projektů. Mezi klíčové principy patří průběžné a rychlé dodávání hodnotného softwaru zákazníkovi, otevřenost změnám požadavků i v pokročilých fázích vývoje a důraz na častou spolupráci mezi vývojovým týmem a zákazníkem. Agilní přístup také zdůrazňuje význam motivace týmu, efektivní komunikace a neustálého zlepšování procesů a pracovních metod (Agile Manifesto, 2001).

Agilní přístup k řízení projektů (viz obr. 4), který se vyznačuje dynamikou, interaktivitou a flexibilitou, představuje revoluční změnu ve vývoji softwaru a v projektovém managementu. Zaměřuje se především na spolupráci, komunikaci a adaptabilitu na změny, čímž se odlišuje od tradičních, neohebně strukturovaných metodik. Agilní model poskytuje týmům svobodu v nastavení vlastních pravidel a procesů s cílem zvýšit efektivitu a kvalitu výsledků v co nejkratším čase (Šochová a kol., 2014).

Obrázek 4 – Agilní metodologie



Zdroj: Šimůnek, 2019

Výhody

- **Flexibilita** – agilní metody umožňují pružnou reakci na změny v průběhu projektu a flexibilní nasazování funkcionalit s vysokou hodnotou pro zákazníka, což podporuje úspěšné dokončení projektu bez zbytečného narušení.
- **Efektivita** – díky silné týmové spolupráci jsou agilní metody výrazně efektivnější, což zvyšuje produktivitu a urychluje vývoj.
- **Kvalita** – implementace agilních praktik přispívá k lepší kontrole kvality produktu, minimalizaci technického dluhu a snížení počtu chyb a problémů.

- **Spokojenost** – agilní metody kladou důraz nejen na finanční návratnost, ale i na spokojenost zákazníka a členů týmu. Zákazníci jsou zapojeni do procesu vývoje, což zajišťuje, že obdrží produkt odpovídající jejich potřebám. Současně agilní metody podporují motivaci a zodpovědnost členů týmu, vedoucí k jejich větší spokojenosti (Šochová a kol., 2014).

Nevýhody

- **Menší předvídatelnost** – bez pevného plánu může být těžší předpovědět přesné datum dodání či rozsah produktu.
- **Závislost na týmu** – agilní přístupy vyžadují zkušené týmy, které jsou schopny samostatně se rozhodovat a efektivně komunikovat (Šochová a kol., 2014).

3.3.3 Klíčové rozdíly

Rozdíl v plánování a adaptabilitě:

- **Waterfall** – tento model vyzdvihuje pečlivé plánování na samém začátku projektu. Celý proces je lineární a každá fáze musí být dokončena, než může začít další. Změny jsou možné, ale často komplikované a nákladné, což může vést k zpoždění.
- **Agile** – na rozdíl od Waterfallu se Agile opírá o pružnost a schopnost přizpůsobit se změnám i ve střední fázi vývoje. Týmy mohou rychle reagovat na nové informace nebo požadavky zákazníků a opakovaně vylepšovat produkt (Šimůnek, 2019).

Zpětná vazba:

- **Waterfall** – v tomto přístupu je zpětná vazba omezená především na začátek a konec projektu, což může znamenat, že případné problémy nebo nové požadavky se identifikují a řeší s odstupem.
- **Agile** – agilní přístup zapojuje zpětnou vazbu do každého opakování, což umožňuje týmu a zákazníkům průběžně hodnotit produkt a jeho směřování, což vede ke zvýšené spokojenosti a relevanci výsledku (Šimůnek, 2019).

Rozsah a specifikace:

- **Waterfall** – rozsah projektu je stanoven pevně na začátku a jakékoliv změny mohou být problematické.

- **Agile** – rozsah je flexibilní, mění se a vyvíjí v reakci na zpětnou vazbu a nově zjištěné informace v průběhu projektu (Šimůnek, 2019).

Waterfall metoda může být ideální pro projekty, kde jsou požadavky a specifikace známy dopředu a jsou stabilní, jako například ve vývoji komplexních systémů pro průmyslové aplikace. Naproti tomu Agile, je vhodnější pro projekty v rychle se měnícím prostředí, jako je vývoj mobilních aplikací, kde pružnost a schopnost reagovat na nové trendy a zákaznické preference jsou klíčové (Šimůnek, 2019).

Dnes je běžné, že organizace implementují hybridní přístupy, které kombinují prvky obou metodologií. Tímto způsobem mohou těžit z jasné struktury a plánování Waterfall modelu a současně využívat pružnosti a adaptace na změny, které přináší Agile metody (Šimůnek, 2019).

3.4 Rozbor standardů projektového řízení

V dnešní době, kdy je důraz kladen na efektivitu, inovace a globální spolupráci, je důležité, aby manažeři a týmy měli společný základ, na kterém mohou stavět při plánování, řízení a realizaci projektů. Tuto základnu představují standardy projektového řízení, které slouží jako univerzální jazyk a metodika pro práci na projektech v různých oborech a kulturách (Doležal a kol., 2012).

Mezi hlavní světové standardy patří Project Management Institute (PMI), International Project Management Association (IPMA) a PRINCE2. Tyto standardy se liší místem vzniku, zdroji informací, z nichž byly odvozeny, a způsobem, jakým byly zpracovány. Přesto mají velmi podobnou základní filozofii, která spočívá ve standardizaci pojmů, metod a postupů pro zajištění efektivního a koordinovaného průběhu projektů (Doležal a kol., 2012).

3.4.1 PRINCE2

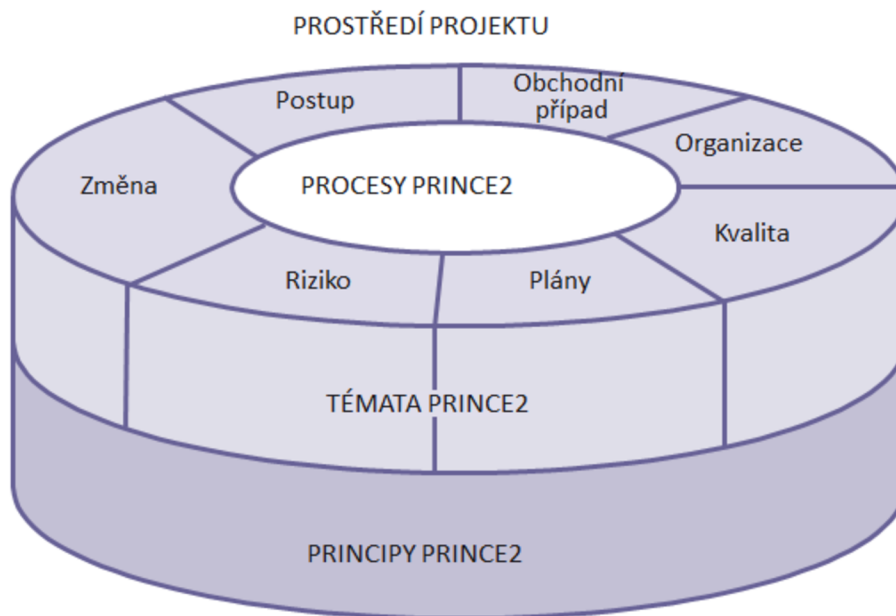
PRINCE2 (PRojects IN Controlled Environments) je jednou z předních metodologií pro řízení projektů. Tato metodologie byla vyvinuta britskou vládní agenturou Office of Government Commerce (OGC) a je v současnosti široce používána v různých odvětvích a geografických oblastech. Je založena na procesně orientovaném přístupu a představuje

ucelený rámec pro efektivní řízení projektů, bez ohledu na jejich velikost, typ nebo průmyslové odvětví (OGC, 2009).

3.4.1.1 Elementy PRINCE2

Struktura PRINCE2 (viz obr. 5) se skládá ze čtyř základních integrovaných prvků, jako jsou principy, témata, procesy a adaptace na konkrétní prostředí projektu. Každý z těchto prvků je dále rozdělen do sedmi principů, sedmi procesů a sedmi témat, čímž vzniká komplexní rámec pro řízení projektů. Tato struktura poskytuje projektovým manažerům flexibilitu a zároveň je vede skrze všechny klíčové aktivity a fáze projektu, což umožňuje účinné řízení a realizaci projektů v různých odvětvích a kontextech (SystemOnLine, 2014).

Obrázek 5 – Struktura PRINCE2



Zdroj: SystemOnLine, 2014

3.4.1.2 Principy PRINCE2

Principy PRINCE2 tvoří základní zásady, na kterých je celá metodologie založena. Mezi principy patří:

- **Podnikatelského zdůvodnění** – neustálé ověřování, že projekt je stále odůvodněn z hlediska podnikatelských cílů.

- **Učení se ze zkušeností** – systematický přístup k identifikaci a využití zkušeností z minulých i současných projektů.
- **Definované role a odpovědnosti** – jasná struktura týmu a jasné rozdělení rolí a odpovědností.
- **Řízení na etapy** – řízení projektu na základě jednotlivých etap s důrazem na řízení rizik a efektivní alokaci zdrojů.
- **Řízení výjimek** – mechanismy pro řešení situací, kdy je překročen plánovaný rozsah nebo jsou ohroženy cíle projektu.
- **Zaměření na produkty** – jasná definice a soustředění se na výstupy projektu.
- **Přizpůsobení prostředí projektu** – metodologie je navržena tak, aby byla flexibilně přizpůsobitelná specifikům jednotlivých projektů (OGC, 2009).

3.4.1.3 Témata PRINCE2

Témata PRINCE2 představují různé oblasti řízení, které musí být v průběhu projektu adekvátně zvládnuty. Tato témata jsou:

- Obchodní případ
- Organizace
- Kvalita
- Plány
- Rizika
- Změny
- Postup

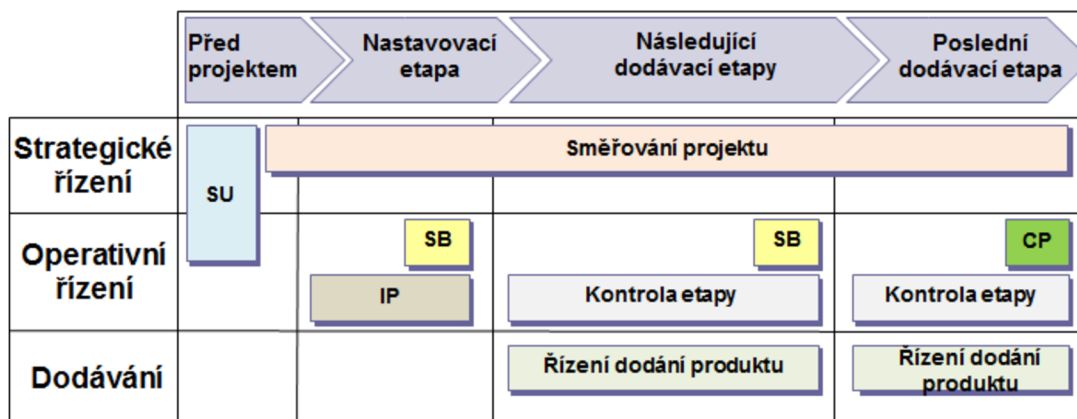
Každé z těchto témat poskytuje specifické postupy a nástroje pro efektivní řízení příslušné oblasti (OGC, 2009).

3.4.1.4 Procesy PRINCE2

Procesy PRINCE2 představují soubor aktivit, které jsou provedeny v rámci projektu (viz obr. 6). Každý proces má jasně definovaný cíl a výstup a je zodpovědný za konkrétní část životního cyklu projektu. Procesy zahrnují přípravu projektu, řízení etap, dodávku

produktů, řízení hranic etap, řízení dodávek, řízení fází a uzavření projektu (OGC, 2009; SystemOnLine, 2014).

Obrázek 6 – Rámcový procesní model PRINCE2



Klíč

SU = Zahájení projektu

IP = Nastavení projektu

SB = Řízení přechodu mezi etapami

CP = Ukončení projektu

Zdroj: SystemOnLine, 2014

3.4.1.5 Přizpůsobení PRINCE2

PRINCE2 je navržen tak, aby byl flexibilně přizpůsobitelný různým projektům. To znamená, že lze upravit rozsah použití jednotlivých procesů, témat a principů v závislosti na velikosti, komplexnosti a specifických požadavcích projektu. Toto přizpůsobení by mělo být vždy provedeno s ohledem na zachování integrity celé metodiky (OGC, 2009).

3.4.2 PMBOK

PMBOK (Project Management Body of Knowledge) je standard vytvořený Project Management Institute (PMI), který představuje soubor nejlepších praktik a standardů v oblasti projektového řízení. PMBOK je strukturován do pěti skupin procesů, jako jsou iniciace, plánování, provádění, monitorování a kontrola a ukončení a deseti znalostních oblastí, což je integrace, čas, rozsah, náklady, kvalita, lidské zdroje, komunikace, rizika, nákupy, zainteresované strany, které pokrývají všechny aspekty řízení projektu (PMI, 2017).

Základní principy a struktura PMBOK, které jsou světově uznávaným standardem, definují ustálené normy, metody a procesy pro efektivní řízení projektů. PMI klade velký

důraz na etiku a profesionální chování, které jsou považovány za klíčové pro úspěch v projektovém managementu. Standard PMI je procesně orientovaný a podporuje aplikaci osvědčených postupů při řízení širokého spektra projektů. PMBOK efektivně rozděluje procesy projektového managementu do pěti hlavních skupin, které jsou propojeny s výsledným produktem projektu. Tento přístup umožňuje projektovým manažerům a týmům lépe pochopit a řídit složitosti související s realizací projektů (Máchal a kol., 2015).

3.4.2.1 Rozbor klíčových prvků PMBOK

PMBOK je založen na přístupu, který integruje různé znalostní oblasti a procesy potřebné pro úspěšné řízení projektu. Každá ze znalostních oblastí se věnuje specifickým aspektům řízení projektu a poskytuje nástroje a techniky, které mohou být použity pro zvládnutí těchto aspektů (PMI, 2017).

3.4.2.2 Jak PMBOK podporuje řízení projektů

PMBOK podporuje řízení projektů tím, že poskytuje rámec, který je flexibilní a lze ho přizpůsobit různým typům projektů. Tento rámec zahrnuje nástroje a techniky, které mohou být aplikovány na různé aspekty řízení projektu, a také definuje základní pojmy a principy, které jsou v projektovém řízení důležité. PMBOK také zdůrazňuje důležitost řízení zainteresovaných stran a komunikace v rámci projektu, což jsou klíčové aspekty pro úspěch projektu (PMI, 2017).

3.4.3 IPMA

IPMA (International Project Management Association) je celosvětově uznávaná organizace, která nastavuje standardy v oblasti projektového řízení a nabízí certifikace pro profesionály v tomto oboru (Máchal a kol., 2015).

Standardy, které vyvíjí a udržuje Mezinárodní asociace pro projektový management (IPMA), se zásadně liší od ostatních metodik řízení projektů tím, že klíčový důraz je kladen na kompetence, což jsou schopnosti a dovednosti projektových a programových manažerů, jakož i členů jejich týmů. Tento přístup, zavedený v 60. letech na základě norem evropských zemí, odmítá striktní procesní modelování ve prospěch flexibilnějšího a osobnějšího přístupu k řízení projektů. Místo pevně daných procesů ICB (International Competence

Baseline), klíčový dokument IPMA, navrhuje určité procesní kroky, jejichž vhodná aplikace závisí na specifikách daného projektu a na jedinečných schopnostech manažerů a týmů (Doležal a kol., 2012).

3.4.3.1 Kompetenční základna

V souladu s touto filozofií je ICB strukturován do tří hlavních kompetenčních oblastí:

- **Technické kompetence** – zahrnují metody, techniky a nástroje.
- **Behaviorální kompetence** – neboli měkké dovednosti.
- **Kontextové kompetence** – schopnost integrace a systémového myšlení.

Každá z těchto oblastí je dále rozdělena na specifické elementy kompetencí, které popisují tematické oblasti, doporučují postupy a kroky a definují požadavky na certifikaci. Tento přístup zajišťuje, že kompetence jsou mezi sebou úzce provázané a vzájemně se podporují, což podtrhuje holistický pohled IPMA na projektové řízení (Doležal a kol., 2012).

IPMA rozpoznává, že každý projekt je unikátní a že neexistuje univerzální přístup k řízení. Proto standard ICB nevyžaduje od projektů, aby byly vedeny striktně procesně, ale umožňuje manažerům používat svůj vlastní úsudek a kreativitu k dosažení cílů projektu. Tento přístup nevylučuje procesní řízení, zejména v organizacích, které drží certifikaci kvality ISO, kde je procesní přístup obvykle požadován. IPMA tedy nabízí flexibilní rámec, který lze adaptovat k různým potřebám a kontextům projektů (Doležal a kol., 2012).

3.4.3.2 Certifikační úrovně

IPMA nabízí čtyři hlavní úrovně certifikace, které reflektují různé úrovně zkušeností a odbornosti v oboru projektového řízení. Tyto certifikace jsou uznané na celém světě a jsou považovány za zlatý standard v profesionálním vzdělávání v oblasti řízení projektů (Máchal a kol., 2015).

3.4.3.3 Význam a přínos

IPMA zdůrazňuje holistický přístup k projektovému řízení, kde je stejný důraz kladen na technické dovednosti, chování a kontextové pochopení. Toto ucelené pojetí je základem úspěšného řízení projektů v různých odvětvích a kulturách (Máchal a kol., 2015).

3.4.4 Agilní metodika Scrum

Scrum je inovativní procesní rámec, který se od devadesátých let využívá k efektivnímu řízení vývoje složitých produktů. Odlišuje se od tradičních metodik tím, že nepředepisuje pevné postupy výroby, ale nabízí flexibilní rámec spojující různé procesy a techniky, které podporují dynamický vývoj projektů. Každá součást tohoto rámce má svůj specifický účel a přispívá k celkovému úspěchu použití Scrumu (Schwaber, Sutherland, 2020).

Základem Scrumu je přizpůsobivost, což znamená, že požadavky zákazníků se mohou během projektu měnit, a proto podporuje flexibilní plánování a rychlou adaptaci na nové situace. Scrum efektivně reaguje na nepředvídatelné změny a umožňuje týmu rychle přizpůsobit práci novým požadavkům (Schwaber, Sutherland, 2020).

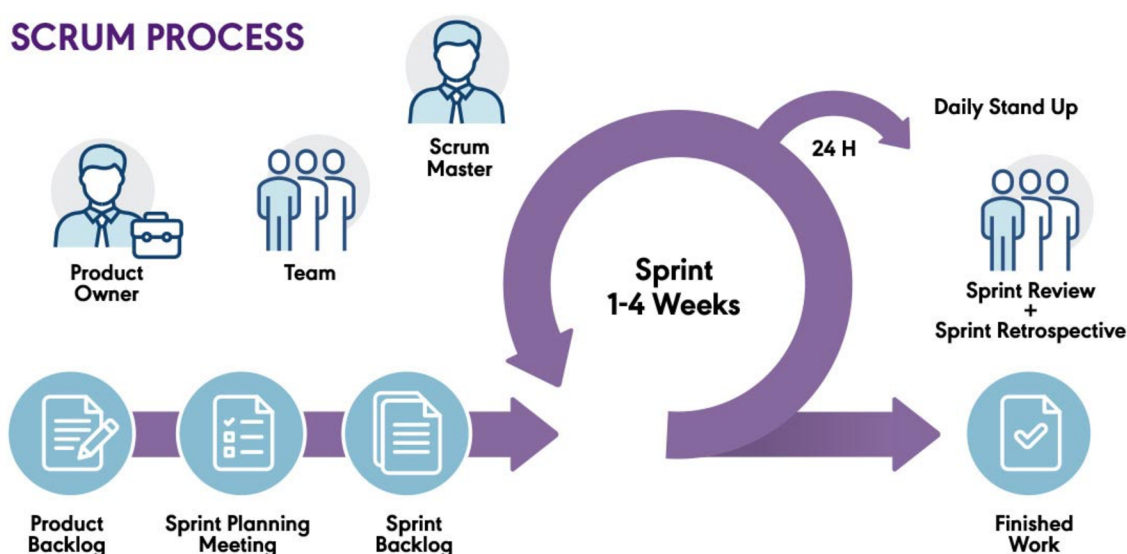
3.4.4.1 Prvky a principy Scrumu

Ve Scrum metodice jsou základními stavebními kameny pravidelně se opakující akce a definované role (viz obr. 7), které společně tvoří strukturu pro efektivní průběh projektů:

- **Sprinty** – tvoří základní rámec pro Scrum, jsou to 30denní cykly práce, během nichž tým pracuje na dosažení konkrétních cílů projektu. Každý sprint obsahuje sérii plánovacích, denních a závěrečných schůzek, stejně jako samotné vývojové práce, a končí výstupem v podobě konečného produktu.
- **Sprint Planning Meeting** – na začátku každého sprintu se tým setkává, aby definoval, co bude v nadcházejícím sprintu realizováno. Cílem je určit úkoly a přiřadit je do sprintu na základě priorit.
- **Daily Scrum Meeting** – každodenní krátká schůzka, během níž členové týmu sdílejí informace o aktuálním stavu své práce, plánovaných aktivitách na následující den a případných překážkách, které jim brání v práci.
- **Sprint Review Meeting** – koná se na konci každého sprintu a slouží k prezentaci dosažených výsledků. Je to příležitost pro tým, zákazníky a zainteresované strany k poskytnutí zpětné vazby a upravení Backlogu podle potřeb projektu.

- **Sprint Retrospective Meeting** – po vyhodnocení sprintu a před dalším plánovacím setkáním tým reflektuje průběh práce ve sprintu s cílem identifikovat příležitosti pro zlepšení a zefektivnění budoucích sprintů (Schwaber, Sutherland, 2020).

Obrázek 7 – Řízení projektů Scrum



Zdroj: PM-Partners Group, 2024

V rámci Scrumu se vyskytují specifické role s definovanými pravomocemi a povinnostmi, které zahrnují:

- **Scrum Master** – zajišťuje spojení mezi týmem a vnějším prostředím, odstraňuje překážky, motivuje a chrání tým před vnějšími rušivými vlivy.
- **Product Owner** – vlastník projektu, který určuje priority, definuje vizi, směr projektu a komunikuje s týmem a zákazníkem.
- **Scrum tým** – multifunkční a vzájemně zastupitelný, pracující na plný úvazek v jedné lokalitě pro maximální efektivitu a kooperaci.
- **Development tým** – skupina vývojářů a specialistů pracujících na technickém řešení projektu, která vytváří produkt podle definovaného Backlogu.
- **Management** – podílí se na definování cílů a požadavků celého projektu a jednotlivých etap.

- **Zákazník** – aktivně zapojen do vývojového procesu, což umožňuje týmu lépe pochopit jeho potřeby a upravit práci podle aktuálních požadavků (Šochová a kol., 2014; Schwaber, Sutherland, 2020).

Důležitým aspektem Scrumu je Backlog, což je seznam všech požadavků projektu, který se rozděluje do Product Backlogu a Sprint Backlogu. Práce na těchto požadavcích probíhá v opakujících se cyklech, tedy sprintech, které vždy směřují k vytvoření funkčního produktu (Schwaber, Sutherland, 2020).

3.4.4.2 Jak Scrum podporuje agilní řízení projektů

Scrum poskytuje silnou podporu agilnímu řízení projektů tím, že poskytuje strukturovaný, ale flexibilní rámec, který umožňuje týmům reagovat na změny a přinášet hodnotu v krátkých, iterativních cyklech nazývaných sprinty. Sprinty jsou obvykle dvoutýdenní nebo měsíční intervaly, během kterých tým pracuje na určité části práce vybrané z produktového Backlogu.

Agilní metodika Scrum také zdůrazňuje roli samosprávných týmů, kde každý člen týmu má jasnou roli a odpovědnost, ale tým jako celek se společně rozhoduje o tom, jak nejlépe dosáhnout svých cílů. To vede k větší motivaci a závazku, protože členové týmu se cítí více zapojeni do procesu a mají větší kontrolu nad svou prací.

Dále Scrum zdůrazňuje hodnotu transparentnosti, kontroly a adaptace. Všechny informace o práci na projektu, včetně stavu úkolů a problémů, jsou viditelné pro všechny členy týmu, což umožňuje rychlé identifikování a řešení problémů. Tým pravidelně provádí retrospektivy na konci každého sprintu, aby se poučil z minulých zkušeností a neustále se zlepšoval (Schwaber, Sutherland, 2020).

3.5 Srovnání a kontrast standardů a metodik

Srovnávání různých standardů a metodik projektového řízení umožňuje organizacím lépe pochopit, jaké přístupy jsou pro ně nejvhodnější v závislosti na charakteristice jejich projektů a prostředí, ve kterém operují.

3.5.1 Porovnání PRINCE2, PMBOK a Scrum

PRINCE2, PMBOK a Scrum představují tři odlišné přístupy k řízení projektů, které mají každý své silné stránky a slabiny. PRINCE2 a PMBOK jsou více procesně orientované a poskytují podrobnou strukturu pro plánování, provádění a kontrolu projektů, zatímco Scrum je více orientovaný na týmovou práci a pružnou adaptaci na změny (PMI, 2017; Axelos, 2017; Schwaber, Sutherland, 2020).

Každá z těchto metodik je vhodná pro určitý typ projektů. Například PRINCE2 a PMBOK mohou být vhodné pro velké, komplexní projekty s pevnými požadavky a s jasně definovanými cíli, zatímco flexibilní Scrum může být lepší volbou pro projekty s vysokým stupněm nejistoty a změny, jako je vývoj software.

3.5.2 Výhody a nevýhody jednotlivých přístupů

Každá metodika projektového řízení má své výhody a nevýhody, které mohou být rozhodující při výběru nejvhodnějšího přístupu k řízení projektů.

PRINCE2 se vyznačuje silnou strukturou, která zahrnuje jasné procesy a role pro účastníky projektu. Tento rámec poskytuje dobrou kontrolu nad projektem a jeho riziky a předpoklady. Na druhou stranu, PRINCE2 může být vnímán jako příliš byrokratický a nepružný pro některé typy projektů, zejména ty, které vyžadují rychlou odezvu na změny (Axelos, 2017).

PMBOK, podobně jako PRINCE2, nabízí podrobný rámec pro řízení projektů, který pokrývá všechny aspekty projektového řízení od zahájení po uzavření projektu. PMBOK je však spíše sbírkou nejlepších postupů než pevným procesem a umožňuje větší flexibilitu při přizpůsobování konkrétní situaci projektu (PMI, 2017).

Scrum, na druhou stranu, se vyznačuje vysokou flexibilitou a schopností rychle reagovat na změny. Jeho silný důraz na komunikaci a spolupráci uvnitř týmu může vést k vyšší produktivitě a motivaci. Nicméně, pro úspěšnou implementaci Scrumu je potřebná transparentnost a sebereflexe, a může vyžadovat hlubší změny v organizační struktuře a praxi (Schwaber, Sutherland, 2020).

IPMA je obzvláště silný v podpoře individuálního rozvoje projektových manažerů. Může být ideální pro organizace, které se snaží rozvíjet své interní schopnosti řízení projektů a posilovat profesionální kapacity svých týmů. Jelikož IPMA zdůrazňuje osobní kompetence,

může někdy chybět pevnější rámec procesů, což by mohlo být nevýhodou pro projekty vyžadující vysokou strukturu a definici (Doležal a kol., 2012).

Při výběru nejvhodnější metodologie projektového řízení je třeba pečlivě zvážit výhody a nevýhody každého přístupu ve vztahu k povaze, požadavkům a kontextu konkrétního projektu. Zatímco PRINCE2 a PMBOK mohou nabídnout silný procesní rámec, Scrum a IPMA nabízí pružnost a důraz na osobní kompetence (viz tab. 1). Ideální volba často spočívá v kombinaci různých metodologií, aby se dosáhlo optimální rovnováhy mezi strukturou a flexibilitou.

3.5.3 Porovnání standardů v tabulce

Tabulka 1 – Porovnání standardů

Parametr/Standard	PRINCE2	PMBOK	Scrum	IPMA
Původ	UK (OGC)	USA (PMI)	Agilní přístup	Globální (založeno v Evropě)
Klíčové komponenty	7 principů, 7 témat, 7 procesů	5 skupin procesů, 10 znalostních oblastí	Sprints, Scrum Master, Product Backlog	ICB (Individual Competence Baseline)
Zaměření	Procesy a role	Procesy a znalostní oblasti	Iterativní vývoj a týmová práce	Kompetence (technické, behaviorální, kontextové)
Flexibilita	Vysoká (přizpůsobitelnost)	Střední (rámcový přístup)	Vysoká (agilní adaptace)	Střední (kompetenční základna)
Certifikace	Ano	Ano	Ano	Ano
Oblast využití	Široký rozsah projektů	Velké projekty a programy	Software a produktový vývoj	Široký rozsah projektů s důrazem na kompetence manažera

Zdroj: Vlastní zpracování

3.5.4 Možnosti kombinování různých přístupů

Existuje množství standardů v oblasti projektového řízení, které byly vytvořeny různými profesními skupinami s nestátním charakterem, ovšem s několika výjimkami. Tyto standardy reflektují myšlenky a zkušenosti svých tvůrců, ovlivněné sociálním a kulturním kontextem původu. Proto by se k těmto standardům mělo přistupovat spíše jako k inspiraci než jako k přísným pravidlům. Důležitým aspektem projektů je jejich jedinečnost, což znamená, že úspěšné metody z jednoho projektu nemusí být stejně účinné v jiném kontextu. Přesto většina standardů projektového řízení sdílí společnou filozofii, využívá podobné metody a terminologii, což výrazně přispívá k lepšímu vzájemnému porozumění a spolupráci mezi pracovníky na projektech. Přesto v základu nesou velmi podobnou filozofii a liší se spíše perspektivou, ze které přistupují k řešení stejných problémů (Doležal a kol., 2012).

Toto se především týká jednotlivých přístupů v izolaci. V praxi však může být pro konkrétní projekt nejlepší kombinace různých prvků z různých přístupů. Tento přístup, známý jako hybridní projektové řízení, kombinuje prvky tradičních a agilních metodologií s cílem dosáhnout optimální rovnováhy mezi kontrolou a flexibilitou (Hass, 2007).

Na druhou stranu, hybridní přístupy také představují určité výzvy. Vyžadují vysokou úroveň dovedností a zkušeností ze strany projektových manažerů, aby bylo možné účinně navigovat mezi různými metodologiemi a najít optimální rovnováhu pro každý konkrétní projekt. Navíc, může být také náročné zvládnout organizační a kulturní změny, které mohou být potřebné pro implementaci hybridních přístupů (Hass, 2007).

3.6 Informační systémy a jejich aplikace v podnikání

3.6.1 Definice informačního systému

Informační systém (IS) může být chápán jako komplexní soubor vzájemně propojených prvků a procesů, které zprostředkovávají shromažďování, zpracování, ukládání a distribuci dat a informací v organizaci. Tento systém je vytvářen a užíván v kontextu reálného světa a je definován svými prvky, vazbami mezi nimi a interakcí s okolním prostředím (Gála a kol., 2015).

Informační systémy jsou tvořeny technickými prostředky, jako jsou počítače a síťové prvky, programové prostředky pro řízení a práci s daty, organizačními pravidly a postupy, lidskou složkou, která zahrnuje uživatele a správce systému, a vnějším okolím, které zahrnuje zákony, normy a další externí data vstupující do systému. (Tvrdíková, 2008).

Efektivita informačního systému závisí na důkladném propojení všech jeho složek a na jeho schopnosti podporovat informační, rozhodovací a řídicí procesy v rámci organizace. Při navrhování informačního systému je nezbytné ujasnění cílů, které má systém řízení podporovat (Tvrdíková, 2008).

3.6.2 Struktura informačního systému

Informační systém je složitá entita, která se skládá z několika klíčových prvků:

1. **Technické prostředky (hardware)** – jedná se o soubor počítačové techniky, která zahrnuje různé typy počítačových systémů a periferních jednotek. Tyto systémy mohou být vzájemně propojeny do počítačové sítě a jsou doplněny o další technické vybavení. Tato komponenta představuje fyzický základ informačního systému.
2. **Technologické prostředky (software)** – tuto kategorii tvoří systémové programy, které zajišťují řízení počítačů, efektivní zpracování dat a komunikaci mezi uživatelem a počítačem. Aplikační programy se věnují specifickým zpracovatelským a komunikačním úlohám, umožňující systému interakci s jeho okolím i jeho vnitřní funkcionalitu.
3. **Organizační prostředky (orgaware)** – jsou to legislativní rámce, sady pravidel a předepsané postupy, které určují způsob provozu a využívání informačního systému. Tyto prostředky mohou zahrnovat také metodické pokyny, normy a další regulační dokumenty.
4. **Lidská složka (peopleware)** – tato složka se týká všech osob zapojených do provozu a využívání informačního systému. Zahrnuje nejen uživatele systému, ale také osoby zodpovědné za správu, údržbu a rozvoj systému. Peopleware je klíčový pro adaptaci a efektivní fungování člověka v prostředí informačních technologií.
5. **Okolí systému** – představuje vnější prostředí, z něhož informační systém čerpá vstupy, tedy data a informace, a kde poskytuje výstupy, což jsou zpracované informace a reporty. Okolí je tvořeno vnějšími informačními zdroji, uživatelskými

nároky, požadavky, technickými a legislativními normami a dalšími faktory, které ovlivňují činnost systému (Vlasák, Bulíčková, 2003; Tvrdíková, 2008).

3.6.3 Význam podnikových informačních systémů

Podnikové informační systémy (PIS) jsou klíčové pro dynamiku a efektivitu podnikových procesů. Jsou chápány jako otevřené systémy, kde primárními vstupy a výstupy jsou informace. Tento systém je sestaven z komponent, které se zabývají shromažďováním, zpracováním, distribucí a přenášením informací v rámci podpory podnikových aktivit. Mezi základní prvky podnikového informačního systému patří lidé, informační technologie, data a procesy transformace a řízení (Gála a kol., 2015).

Podnikový informační systém je základním stavebním kamenem podnikání. Prostřednictvím kombinace technologických a organizačních prostředků zpracovává podniková data a transformuje je do informací a znalostí nezbytných pro správné rozhodování a efektivní řízení podniku (Sodomka, Klčová, 2010).

3.6.4 Klasifikace podle organizační úrovně

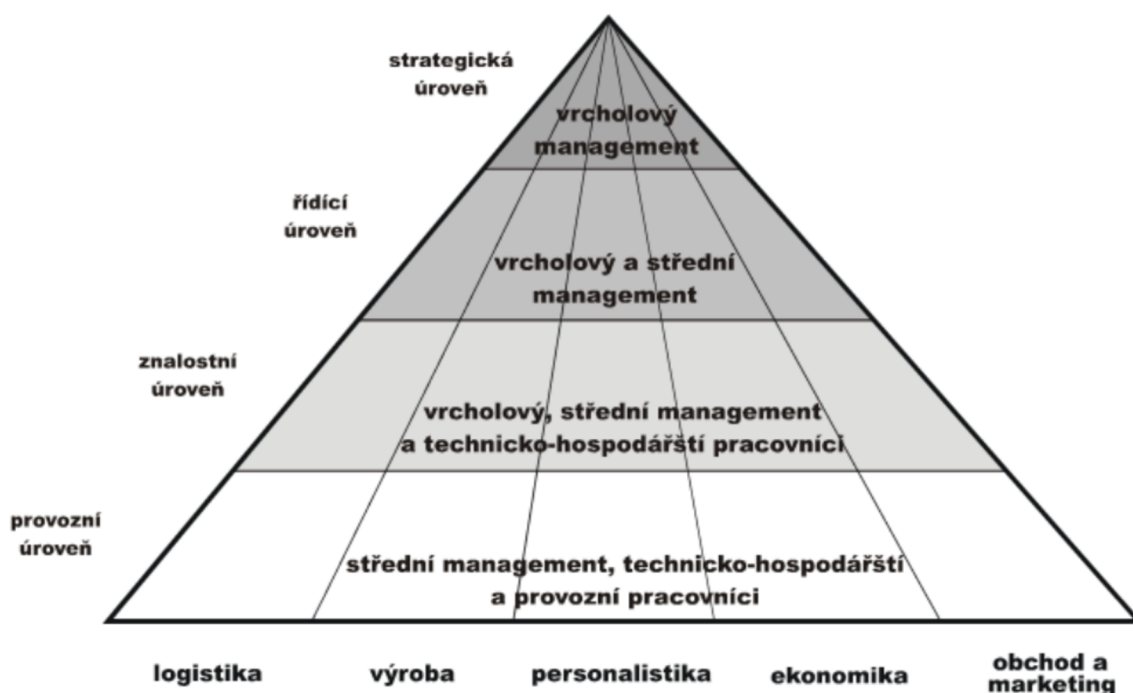
Podniky mají různé organizační úrovně, každá s vlastními informačními potřebami (viz obr. 8). Tyto úrovně lze rozdělit na:

- **Provozní úroveň** – informační systémy zde zajišťují denní operativní aktivity, jako jsou transakce nákupu, prodeje, správa inventáře a finanční transakce. Systémy na této úrovni musí být přesné a aktuální, aby podporovaly každodenní rozhodování a úkoly.
- **Znalostní úroveň** – zahrnuje aplikace, jako jsou ERP a CRM, i nástroje pro osobní informatiku, které podporují manažery a specialisty v jejich denní práci. Umožňují transformaci dat do znalostí a podporují vytváření zkušeností pracovníků.
- **Řídící úroveň** – obsahuje nástroje a systémy pro administrativní úkoly a analýzy dat, které slouží pro rozhodování středního a vrcholového managementu. Zde jsou klíčové reporty, které sumarizují data a poskytují podklady pro rozhodování o řízení podniku.

- **Strategická úroveň** – systémy na této úrovni pomáhají vrcholovému managementu identifikovat dlouhodobé trendy a připravit strategie pro budoucí rozvoj podniku. Informační systémy zde pomáhají předvídat změny v interním i externím prostředí a formují odpovědi na tyto výzvy (Sodomka, Klčová, 2010).

Každá z těchto úrovní poskytuje různé pohledy na podnik a jeho funkce. Přestože se mohou zdát být oddělené, ve skutečnosti jsou vzájemně propojené a vzájemně se ovlivňují. Efektivní informační systém v podniku je proto ten, který dokáže integrovat a synchronizovat všechny tyto úrovně, aby podpořil jednotnou a koordinovanou strategii řízení (Sodomka, Klčová, 2010).

Obrázek 8 – Organizační úrovně podniku



Zdroj: Sodomka, Klčová, 2010

3.6.5 Typy podnikových informačních systémů

V rámci moderního podnikání hrají podnikové informační systémy klíčovou roli v řízení, optimalizaci a integraci různých podnikových procesů. Tyto systémy lze kategorizovat podle jejich funkcí, potřeb podniku a nabídky na trhu. Zásadní pro pochopení a efektivní využití

těchto systémů je holisticko-procesní přístup, který zdůrazňuje význam jejich vzájemného propojení a integrace do jednotného systému (Sodomka, Klčová 2010).

Mezi základní a nejrozšířenější typy podnikových informačních systémů patří:

3.6.5.1 Business Processes

Podnikové procesy představují základní strukturu pro uspořádání činností a toků v organizaci, které jsou nezbytné pro její efektivní fungování. Tyto procesy integrují různé aspekty podnikové činnosti, včetně administrativy, technologie a managementu, aby zajistily hladký průběh operací a dosažení organizací stanovených cílů (Sodomka, Klčová, 2010).

3.6.5.2 ERP (Enterprise Resource Planning) systémy

ERP systémy jsou klíčovým nástrojem pro efektivní plánování a správu podstatných vnitropodnikových procesů, které zahrnují transformaci vstupů na výstupy na všech úrovních řízení organizace. Tyto systémy představují středobod aplikací informačních systémů podniku, pokrývající široké spektrum funkcí a nezbytných procesů. K těm patří například výrobní procesy, řízení lidských zdrojů, logistika a finanční řízení (Tvrđíková, 2008).

Primárním cílem ERP systémů je centralizace firemních funkcí na všech úrovních podniku, což se dosahuje integrací různorodých aplikací sloužících specifickým potřebám jednotlivých oddělení do jediného systému s jednotnou databází. Tato centralizace vede k minimalizaci rizik spojených s neefektivitou zpracování informací, nekonzistencí dat a potenciálními nesrovnalostmi (Tvrđíková, 2008).

3.6.5.3 CRM (Customer Relationship Management) systémy

Customer Relationship Management (CRM) je zásadní pro efektivní správu vztahů se zákazníky, umožňuje sledování aktivit a správu interakcí se zákazníky prostřednictvím datových systémů a aplikací. Integrace komunikačních kanálů jako emailů, call center a maloobchodních bodů posiluje vztahy se zákazníky a podporuje informovanost o trhu.

Při zavádění CRM je klíčové zvolit řešení kompatibilní s existujícími informačními systémy podniku, což zajišťuje podporu všech podnikových aktivit. Účinné využívání CRM vyžaduje správnou správu zákaznických dat, analýzu jejich chování a efektivní komunikaci.

CRM umožňuje firmám lépe reagovat na potřeby zákazníků a přizpůsobit se měnícímu se tržnímu prostředí. Jeho správná implementace vede k zvýšení zákaznické spokojenosti, loajality a celkové ziskovosti firmy (Tvrdíková, 2008).

3.6.5.4 SCM (Supply Chain Management) systémy

Supply Chain Management (SCM) je klíčový pro efektivní řízení dodavatelského řetězce a je úzce propojen s Enterprise Resource Planning (ERP) systémy. SCM využívá soubor nástrojů a procesů, které cílí na optimalizaci a zefektivnění toku zboží, informací a financí od dodavatelů k zákazníkům. Hlavním cílem je zvýšit akceschopnost a spolehlivost celého řetězce díky lepší kooperaci mezi obchodními partnery (Basl, Blažíček, 2008).

3.6.5.5 BI (Business Intelligence) systémy

Business Intelligence (BI) je definován jako soubor přístupů a aplikací informačních a komunikačních technologií, které se zaměřují primárně na podporu analytických a plánovacích aktivit v organizacích. Tento komplex nástrojů a metodologie umožňuje organizacím zpracovávat obrovské objemy dat z různých zdrojů a přeměnit je na cenné informace a znalosti. Princip multidimenzionality, klíčový pro BI, otevírá možnost pohlízet na data z různých perspektiv a získat tak komplexní pohled na realitu (Novotný a kol., 2005).

V dnešním dynamickém a vysoce konkurenčním prostředí představuje schopnost rychle a efektivně rozhodovat na základě kvalitních dat zásadní konkurenční výhodu. Business Intelligence poskytuje nástroje pro efektivní práci s firemními daty, což umožňuje lepší strategické rozhodování a v konečném důsledku vede k obchodnímu úspěchu (Novotný a kol., 2005).

4 Vlastní práce

4.1 Úvod do problematiky

V rámci dynamicky se vyvíjejícího prostředí softwarového průmyslu je pro firmy klíčové, aby neustále inovovaly a přizpůsobovaly se rychlým změnám. Společnost, která se zabývá vývojem webových aplikací, stojí před výzvou, jak udržet krok s neustálými inovacemi a zároveň zůstat konkurenceschopná. Jako odpověď na tuto výzvu bylo rozhodnutí o zavedení nového systému zvaného GitLab, což je platforma pro správu verzí a kooperativní vývoj softwaru. Toto rozhodnutí vykresluje snahy společnosti o modernizaci pracovních procesů a zlepšení efektivity práce.

Vedení firmy, ve snaze zvýšit efektivitu a transparentnost pracovních procesů, identifikovali GitLab jako ideální nástroj pro dosažení těchto cílů. Tento software, který nabízí verzování kódu a podporu pro agilní metodologie, nabízí prostředí, které umožňuje lepší sledování a správu projektů. Zároveň umožňuje snadnější spolupráci mezi různými týmy a odděleními, což je zásadní pro dosažení firemních cílů.

Avšak s rozhodnutím zavést nový systém GitLab přichází i řada výzev. Jednou z hlavních výzev je školení zaměstnanců na nový systém, zejména pro ty, kteří nejsou obeznámeni s Git verzovacím systémem. Další výzvou je integrace GitLabu do stávajících pracovních procesů a systémů. Rovněž je třeba pečlivě zvážit otázky týkající se bezpečnosti a ochrany dat, protože přechod na nový systém může zahrnovat přesun citlivých informací.

V průběhu implementace nového systému autor práce zastával roli zaměřenou nejen na pozorování, ale měl i aktivní účast na projektu, přičemž klíčovým úkolem bylo dokumentování a analýza jednotlivých kroků implementačního procesu. Důležitou součástí práce bylo pečlivé zaznamenávání průběžných aktivit, interakcí mezi členy projektového týmu, stejně jako dokumentace výzev či problémů, které se během procesu objevily. Uspořádání a zpracování informací autorem práce představuje přidanou hodnotu pro tento projekt a poskytuje ucelený pohled na implementaci informačního systému a dopady jeho zavedení na organizaci.

4.2 Charakteristika společnosti

Společnost, která je předmětem vlastní práce, je softwarová firma, založená před více než deseti lety, která se zaměřuje na vývoj webových aplikací pro různorodé klienty, od malých start-upů až po středně velké podniky. S pracovním kapitálem složeným z pěti programátorů, tří designérů, tří marketérů, dvou projektových manažerů, dvou obchodních manažerů a jednoho účetního, vytvářejí produkty odrážející jak technickou záštitu, tak i kreativní přístup k jednotlivým projektům (viz obr. 9). Pro úspěšnou organizaci společnosti se ve vedení společnosti nachází ředitel firmy a technický ředitel.

Popisovaná společnost uplatňuje v rámci svého řízení agilní přístup a flexibilitu, což jí umožňuje rychle reagovat na proměny trhu a požadavky klientů. I přes svou menší velikost se společnost vyznačuje flexibilní strukturou a plochou hierarchií, což přispívá k efektivní spolupráci a zachování osobního přístupu ve své pracovní kultuře.

V následujících kapitolách se podrobněji zaměříme na proces výběru a zavedení GitLabu, včetně analýzy stávajících procesů, plánování implementace, školení týmu a hodnocení dopadu na operace a projekty. Cílem je navrhnout proces zavedení GitLabu do prostředí malé softwarové firmy a tím zvýšit efektivnost komunikace mezi jednotlivými odděleními a zvýšení efektivnosti sledování technologického postupu při tvorbě webových aplikací.

4.3 Struktura společnosti

Efektivní organizační struktura a komunikace jsou základními pilíři úspěšného podniku. Každé oddělení má své specifické role a odpovědnosti, ale jejich vzájemná spolupráce a komunikace jsou nezbytné pro dosažení cílů společnosti.

4.3.1 Vedení firmy

Na vrcholu organizační struktury stojí ředitel firmy, který je klíčovou osobou pro strategické rozhodování a směřování firmy. Společně s technickým ředitelem, který představuje spojnicí mezi vedením a technickými týmy. Zajišťují, že vize firmy je komunikována napříč všemi úrovněmi a že strategie jsou prováděny jednotně.

4.3.2 Oddělení programátorů

Vedoucí programátorů spolu s týmem programátorů stojí za technologickým vývojem a inovacemi produktů a služeb. Spolupráce mezi programátory a projektovým týmem je nezbytná pro zajištění technologické realizace projektových cílů a pro rychlou odezvu na technické požadavky.

4.3.3 Oddělení designu

Oddělení designu, s vedoucím designérů a designéry, hraje zásadní roli ve vizuálním prezentování produktů a značky firmy. Jejich úkoly jsou nezbytně koordinovány s marketingovým týmem, aby byla zajištěna jednotnost všech marketingových materiálů.

4.3.4 Obchodní a finanční oddělení

Vedoucí obchodu a finanční oddělení, spolu s obchodními manažery a účetními, tvoří páteř komerční stránky firmy. Toto oddělení je klíčové pro rozvoj obchodních příležitostí, udržování vztahů se zákazníky a zajištění finančního zdraví společnosti. Pravidelné komunikační toky mezi obchodními a finančními týmy jsou kritické pro efektivní řízení cashflow a rozpočtování.

4.3.5 Projektové oddělení

Projektové oddělení, vedené projektovým vedoucím a manažerem, je zodpovědné za správu a koordinaci projektů ve firmě. Tento tým funguje jako spojka mezi různými odděleními. Zajišťuje, že projekty jsou realizovány v souladu s plány a cíli firmy. Projektové oddělení zajišťuje, že jsou všechna oddělení synchronizována a že je mezi nimi efektivní komunikace.

4.3.6 Marketingové oddělení

Vedoucí marketingu a jeho tým marketérů tvoří kreativní sílu firmy. Jsou zodpovědní za vytváření a realizaci marketingových strategií, propagaci produktů a služeb a komunikaci s trhem. Vzájemná spolupráce mezi marketéry a ostatními odděleními je nezbytná k udržení konzistence značky a marketingových sdělení.

Komunikace mezi jednotlivými odděleními probíhá prostřednictvím pravidelných schůzek, zpráv a interních komunikačních nástrojů. Společné projekty jsou koordinovány tak, aby byla zajištěna vzájemná spolupráce a aby každé oddělení bylo plně informováno o postupu a potřebách ostatních. Zaváděný software GitLab v této struktuře poskytuje podporu pro řízení projektů a usnadňuje průběh práce, což umožňuje lepší spolupráci a efektivnější realizaci firemních cílů.

Obrázek 9 – Hierarchie společnosti



Zdroj: Vlastní zpracování (StarUML)

4.4 Popis zaváděného informačního systému

GitLab je univerzální nástroj pro správu životního cyklu softwarových aplikací, který nabízí komplexní řešení od vývoje přes testování až po nasazení. Systém je postaven na otevřeném zdrojovém kódu a je dostupný v různých verzích, od bezplatného Community Edition až po placené Enterprise verze s pokročilými funkcemi. V rámci verze Enterprise pak nabízí různé předplatné v závislosti na požadovaných funkcích. GitLab také podporuje SCRUM procesy poskytováním nástrojů pro Sprint plánování, Backlog management a retrospektivy. Tímto způsobem se GitLab stává nejen nástrojem pro technickou stránku vývoje, ale také pro organizaci a správu projektů.

Mezi hlavní vlastnosti GitLabu patří:

- **Správa repozitářů** – GitLab poskytuje uživatelům pokročilé možnosti správy Git repozitářů. Umožňuje nejen ukládat a prohlížet kód, ale také spravovat přístupy, větve a značky (tags), což usnadňuje organizaci zabezpečení kódové báze. Git

repozitáře v GitLabu slouží jako centrální bod, kde vývojáři mohou spolupracovat na kódu, sdílet změny a sledovat historii projektu.

- **Automatizace CI/CD** – jádrem GitLabu je jeho CI/CD nástroj, který je plně integrován do systému a automatizuje klíčové části vývojového procesu. Uživatelé mohou definovat Pipeline pro automatizované sestavování, testování a nasazování aplikací, což zvyšuje efektivitu a snižuje potenciál pro lidské chyby. To zahrnuje spouštění testů, sestavování binárních souborů a jejich distribuci do testovacích nebo produkčních prostředí.
- **Správa projektů** – GitLab rozšiřuje své funkce do oblasti správy projektů, poskytující nástroje pro sledování úkolů, chyb, a milníků. Umožňuje týmům vytvářet plány, sledovat postup práce a reagovat na změny projektu agilně. Funkce jako Issue tracking či kanban nástěnky z GitLabu činí užitečnou platformu pro projektové manažery a týmy.
- **Dokumentace** – GitLab poskytuje také možnosti pro tvorbu a sdílení dokumentace. Vestavěná dokumentace a systém pro správu dokumentů umožňuje týmům udržovat veškerou projektovou dokumentaci aktualizovanou a snadno přístupnou. To zahrnuje technické manuály, návrhy, specifikace a jakékoli další dokumenty související s projektem.

4.5 Důvody výběru zaváděného systému pro daný podnik

Při výběru nástroje pro firmu bylo několik klíčových kritérií na základě kterých byl vybrán právě GitLab:

- **Komplexnost řešení** – GitLab nabízí všechny potřebné nástroje pro správu životního cyklu softwarových projektů pod jednou střešou. Jeho univerzální přístup byl ideální pro potřeby firmy, kde je třeba kombinovat vývoj softwaru s účetními a administrativními úkoly.
- **Flexibilita** – díky své modulární struktuře a širokému ekosystému doplňků lze GitLab přizpůsobit konkrétním potřebám firmy, což umožnilo implementovat jen ty funkce, které byly skutečně potřebné.

- **Cena a dostupnost** – s možností využít bezplatnou verzi a později přejít na komerční verzi s dalšími funkcemi nabízí GitLab flexibilní cenovou strukturu, která odpovídá různým rozpočtům.
- **Otevřený zdrojový kód** – jako open-source řešení umožňuje GitLab snadnou modifikaci a rozšiřitelnost, což může být v budoucnosti užitečné pro další specifické potřeby firmy, především jako hostování na vlastních serverech.

Zároveň ale také mohou při zavádění nového systému nastat následující nevýhody nebo výzvy:

- **Zaškolení zaměstnanců** – vzhledem k tomu, že programátoři, designéři a projektový manažeři firmy byli do této doby zvyklí pracovat s GitHubem a software Jira, bude klíčové vytvořit školení na míru pro urychlení adaptace na GitLab. Školení bude muset být zaměřeno na konkrétní workflow dané firmy a předchozí zkušenosti týmu s verzovacími nástroji.
- **Plán migrace dat** – přenos projektů, které byly dosud spravovány v GitHubu a Jira, do GitLabu vyžaduje promyšlený migrační plán. Vzhledem k různorodosti projektů a potřeb klientů firmy bude migrace vyžadovat individualizovaný přístup, aby byla zajištěna kontinuita práce a zachování všech verzí a historie kódu.
- **Závislost na jednom dodavateli** – spoléhání na jediný systém může zvýšit riziko v případě výpadků služby nebo omezení v budoucí podpoře a rozvoji GitLabu.
- **Zabezpečení dat** – bezpečnost je pro klienty a jejich projekty hlavní prioritou. Musí být zajištěno, že GitLab bude správně nakonfigurován a ochráněn proti hrozbám, což může zahrnovat nastavení silných hesel, autentizace a pravidelné bezpečnostní audity.
- **Řízení odporu k novinkám** – ve firmě může být odpor k novým systémům způsoben preferencí stávajících, dobře zaběhnutých procesů. Přejít na GitLab vyžaduje citlivou komunikaci a zapojení všech členů týmu do rozhodovacího procesu, aby byly rozptýleny obavy a byla podpořena pozitivní akceptace nového systému.

Zavedení GitLabu nabízí řadu výhod, včetně zvýšení efektivity pracovních procesů, zlepšení komunikace mezi týmy a zvýšení kvality výsledného softwaru. GitLab také

napomáhá kultuře neustálého zlepšování, podporuje transparentnost a umožňuje firmě lépe reagovat na měnící se požadavky trhu.

Vzhledem k těmto důvodům byl GitLab identifikován jako ideální řešení pro potřeby firmy. Jeho implementace a integrace do firemního prostředí byly klíčové pro zefektivnění interních procesů a dosažení cílů stanovených v této práci. GitLab jako zavedený systém ve firmě představuje více než jen sadu nástrojů pro vývojové týmy. Jeho integrace s agilními metodami, jako je Scrum a jeho CI/CD schopnosti, posouvají firmu do centra inovace a umožňují rychlý a spolehlivý vývoj softwaru, který je nezbytný pro konkurenceschopnost v dnešním rychle se měnícím digitálním světě.

4.6 Fáze zavedení systému v podniku

Zavedení systému GitLab do firemních procesů zaujímá v této práci zásadní místo. Fáze zavedení byly rozčleněny do několika důležitých etap (viz obr. 10), přičemž každá z nich měla přesně definované cíle a odpovědnosti (viz tab. 2). Klíčovým řešitelem a manažerem projektu byl ředitel firmy, který zajišťoval celkové vedení a koordinaci, zatímco detailní řízení jednotlivých etap bylo svěřeno příslušným oddělením. Firma si stanovila rozpočet na zavedení systému **750 000 Kč** a časový horizont **90 dnů** na úspěšné zavedení systému. Implementace byla navržena tak, aby podporovala firemní kulturu neustálého zlepšování a inovací, s důrazem na efektivní komunikaci a transparentnost všech pracovních procesů.

Obrázek 10 – Souhrn všech kroků během zavádění systému

☐ Projekt zavedení systému GitLab
☒ Přípravná fáze
☒ Fáze návrhu a vývoje
☒ Implementační fáze
☒ Fáze testování
☒ Závěrečná fáze

Zdroj: Vlastní zpracování (ProjectLibre)

4.6.1 Přidělení jednotlivých procesů při zavádění systému

Během procesu zavádění systému bylo nutné určit odpovědnost jednotlivých oddělení za dané procesy (viz tab. 2). V textu níže jsou popsány jednotlivé kroky a odpovědnosti oddělení v procesu zavádění nového systému. Odpovědnost je vázaná na celá oddělení, nikoliv na jednotlivce ve firmě.

4.6.1.1 Přípravná fáze

- **Definice cílů projektu** – vedení firmy má na starosti definování strategických cílů pro zavádění systému. Jejich úkolem je vymezit, co si firma od nového systému slibuje a jaké obchodní výhody chce dosáhnout.
- **Analýza současného stavu** – zde se předpokládá analýza stávajících procesů a systémů, aby se pochopilo, kde a jak může být GitLab nejúčinněji implementován. Tento krok může zahrnovat spolupráci různých oddělení.
- **Plánování projektu** – detailní plánování kroků, časových rámců a rozdělení zdrojů potřebných pro implementaci GitLabu. Toto zahrnuje vytvoření projektového plánu.

4.6.1.2 Fáze návrhu a vývoje

- **Technický návrh** – oddělení programátorů je odpovědné za technické aspekty návrhu systému. Musí zajistit, že systém bude splňovat požadované funkčnosti.
- **Rozpočet a finanční plánování** – vedení firmy je odpovědné za finanční stránku projektu, včetně sestavení rozpočtu a finančního plánování pro pořízení předplatného, implementaci a údržby systému.

4.6.1.3 Implementační fáze

- **Nastavení a konfigurace GitLabu** – oddělení programátorů se stará o technické nastavení GitLabu, jeho konfiguraci a přizpůsobení podnikovým potřebám.
- **Migrace dat a integrace systému** – všechna oddělení jsou zapojena do procesu migrace dat a integrace GitLabu se stávajícími systémy.
- **Školení a podpora** – poskytování školení a podpory pro zaměstnance je zásadní pro úspěšné přijetí systému, což zahrnuje všechna oddělení.

- **Průběžné finanční sledování** – vedení firmy je odpovědné za dohled nad finančními aspekty projektu, včetně sledování výdajů a nákladů spojených s implementací.

4.6.1.4 Fáze testování

- **Testování a kontrola v provozu** – všechna oddělení jsou zapojena do testování systému, aby se zajistilo, že GitLab funguje podle očekávání.
- **Identifikace a řešení případných problémů** – vedení firmy, oddělení programátorů a oddělení marketingu jsou zodpovědné za řešení všech problémů, které byly během testování identifikovány a spadají do jejich kompetence.
- **Dokumentace** – oddělení marketingu vytváří dokumentaci, která může zahrnovat uživatelské manuály a vnitrofiremní dokumenty pro používání GitLabu.

4.6.1.5 Závěrečná fáze

- **Zjišťování zpětné vazby** – vedení firmy shromažďuje zpětnou vazbu od zaměstnanců a zákazníků k hodnocení úspěšnosti implementace.
- **Závěrečná zpráva a hodnocení** – zpracování závěrečného zhodnocení projektu, které poskytne přehled o úspěšnosti a případných oblastech pro zlepšení.

Tabulka 2 – Grafické znázornění odpovědnosti za dané procesy

Krok	Popis	Zodpovědné oddělení
1.	Přípravná fáze	
	Definice cílů projektu	Vedení firmy
	Analýza současného stavu	
	Plánování projektu	
2.	Fáze návrhu a vývoje	
	Technický návrh	Oddělení programátorů
	Rozpočet a finanční plánování	Vedení firmy
3.	Implementační fáze	
	Nastavení a konfigurace GitLabu	Oddělení programátorů
	Migrace dat a integrace systému	Všechna oddělení
	Školení a podpora	Všechna oddělení
	Průběžné finanční sledování	Vedení firmy
4.	Fáze testování	
	Testování a kontrola v provozu	Vedení firmy
	Identifikace a řešení případných problémů	Vedení firmy, oddělení programátorů, oddělení marketingu
	Dokumentace	Oddělení marketingu
5.	Závěrečná fáze	
	Zjišťování zpětné vazby	Všechna oddělení
	Závěrečná zpráva a hodnocení	Vedení firmy

Zdroj: Vlastní zpracování

4.7 Fáze zavádění systému

4.7.1 Přípravná fáze

Přípravná fáze projektu zavedení systému GitLab byla v dané menší firmě přizpůsobena rozsahu a specifikům podnikání. Místo rozsáhlého týmu zde definici cílů, analýzu stávajících procesů a plánování projektu zajišťoval malý, ale efektivní tým. Tuto fázi vedl ředitel firmy, technický ředitel ve spolupráci s hlavním vývojářem a projektovým manažerem, kteří společně vytvářeli strategický plán na následné zavedení nového systému.

Tento přístup umožňoval přímou komunikaci a rychlou reakci na vznikající potřeby a problémy, což je v malých firmách nezbytné pro agilní a flexibilní řízení projektů.

4.7.1.1 Definice cílů projektu

Cíle projektu se stanovily na základě strategických potřeb firmy a byly zásadní pro směřování dalšího průběhu implementace. V rámci této aktivity vedení firmy vymezilo specifické a dosažitelné cíle, které by měly být v souladu se standardy projektového řízení s důrazem na agilní metody. Tyto cíle by měly zahrnovat zvýšení efektivity týmů, zlepšení kvality kódu, snížení času na dodání produktů nebo zvýšení spokojenosti zákazníků.

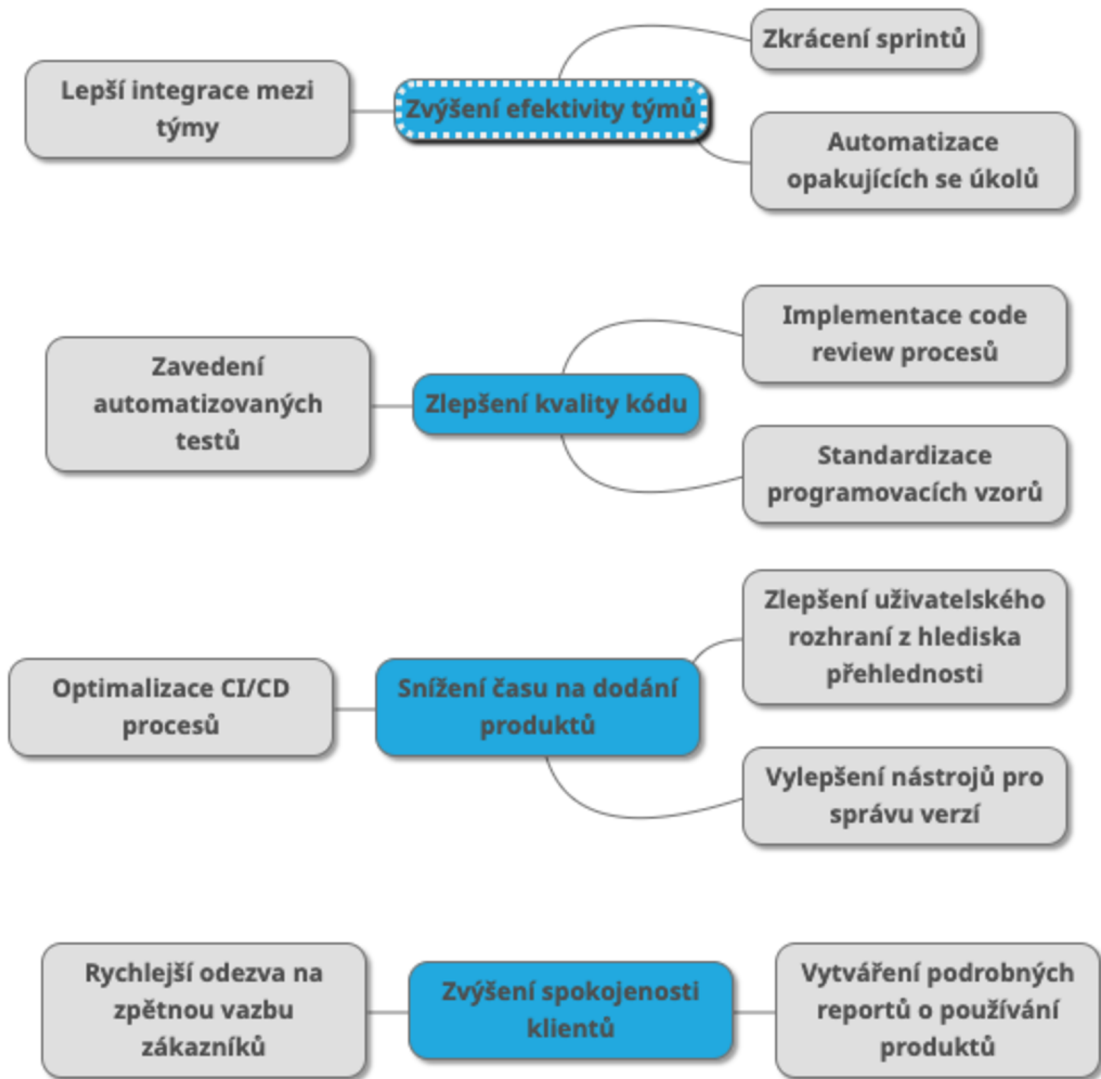
Proces stanovení cílů začínal sérií schůzek mezi ředitelem firmy a technickým ředitelem, během kterých společně identifikovali a definovali klíčové oblasti pro zlepšení a inovaci. Tyto interaktivní diskuse umožnily vytyčit konkrétní cíle, které byly přímo spojeny s firemními cíli a očekáváními.

Cílem bylo porovnat rozdíl mezi současnými procesy a potenciálem, který GitLab nabízí, a to vytvořením jasně strukturovaného seznamu cílů, které byly podrobně rozpracovány a schváleny všemi zúčastněnými stranami. Tento proces dal pevné vodítko pro celý projekt a zároveň sloužil jako měřítko úspěchu.

Jednotlivé postupy zahrnovaly:

- **Sběr nápadů** – každý člen týmu je povzbuzován ke sdílení svých nápadů a poznámek, bez ohledu na jejich postavení ve firmě (viz obr. 11). Veškeré nápady byly přijímány bez předběžného hodnocení, aby se podpořila kreativita.
- **Vizuální techniky** – využívaly se tabule nebo digitální nástěnky pro vizuální organizaci nápadů.
- **Tematické skupiny** – nápady byly kategorizovány do tematických skupin, aby se identifikovaly společné cíle a priority (viz obr. 11).

Obrázek 11 – Nápady na zlepšení projektu



Zdroj: Vlastní zpracování (Mindmup.com)

Po definování cílů následovala fáze revize, kde se vedení soustředilo na zhodnocení a organizaci nápadů z předešlého dne. Tato fáze zahrnovala:

- **Diskuse** – vedení firmy diskutovalo každý nápad a zvažovalo jeho relevanci a přínos pro projekt.
- **Prioritizace** – nápady byly seřazeny na základě jejich očekávaného dopadu na projekt a firmu.

Následovala fáze formulace cílů, kdy se z nápadů stanovily konkrétní cíle projektu. Tento proces zahrnoval:

- **Kritéria** – každý cíl musí být specifický, dosažitelný, relevantní a časově dostupný.
- **Zapojení zúčastněných stran** – k zajištění podpory a porozumění byly cíle prezentovány a diskutovány se všemi klíčovými zainteresovanými stranami.
- **Dokumentace** – cíle byly zaznamenány a zdokumentovány v textové podobě.

Seznam cílů, který byl vytvořen v této fázi, zahrnuje:

1. **Zvýšení efektivity týmů:**

- **Zkrácení sprintů** – sprinty budou plánovány kratší, což povede k rychlejším a častějším iteracím a tím se zvýší adaptabilita a flexibilita týmů.
- **Lepší integrace mezi týmy** – týmy budou pracovat na posílení vzájemné komunikace a spolupráce, aby se zvýšila celková efektivita pracovních procesů.
- **Automatizace opakujících se úkolů** – automatizace bude zaváděna s cílem snížit manuální úsilí a umožnit týmům soustředit se na významnější úkoly.

2. **Zlepšení kvality kódu:**

- **Implementace Code Review procesů** – procesy revize kódu budou vykonávány na jednom místě, aby se zvýšila kvalita a spolehlivost softwaru.
- **Zavedení automatizovaných testů** – automatizované testy budou významnou součástí vývoje, zajišťující konzistenci a snížení chyb v produkci.
- **Standardizace programovacích vzorů** – vzory a standardy kódování budou zavedeny, aby se zajistila jednotnost a čitelnost kódu napříč týmy.

3. **Snížení času na dodání produktů:**

- **Zlepšení uživatelského rozhraní z hlediska přehlednosti** – zdrojové kódy, agenda, přehled o nasazení a komentáře v rámci jednoho systému.
- **Optimalizace CI/CD procesů** – procesy Continuous Integration a Continuous Deployment (CI/CD) budou optimalizovány pro zvýšení efektivity a snížení doby nasazování.
- **Vylepšení nástrojů pro správu verzí** – nástroje pro správu verzí budou vylepšeny, což povede ke snížení chyb při nasazování nových verzí softwaru.

4. Zvýšení spokojenosti klientů:

- **Rychlejší odezva na zpětnou vazbu zákazníků** – procesy zpětné vazby budou zrychleny, aby se zákazníkům dostalo rychlých a relevantních odpovědí.
- **Vytváření podrobných reportů o používání produktů** – reporty a analýzy využití produktů budou pravidelně vytvářeny, což umožní lepší porozumění potřebám a chování zákazníků.

Výsledný seznam cílů představoval jasný plán, který bude sloužit jako základ pro všechny další fáze projektu zavádění GitLabu. Zajištění, že cíle jsou dobře definovány a schváleny všemi zúčastněnými stranami bylo nezbytné pro úspěch projektu a zajistilo, že všechny aktivity budou zaměřeny na dosažení těchto cílů.

4.7.1.2 Analýza současného stavu

Tato část se věnuje analýze systémů a procesů, které byly ve firmě implementovány před zavedením GitLabu. Analýza stávajících pracovních procesů umožnila identifikovat klíčové oblasti pro zlepšení a byla základem pro návrh cílů integrace GitLabu. Zde se nachází praktiky, které se ve firmě používaly v době před zavedením nového softwaru:

- **Software GitHub** – GitHub, jako platforma pro verzování, sloužil jako centrální repozitář pro veškerý zdrojový kód. Jeho silné stránky spočívaly v podpoře distribuovaného verzování a v široké nabídce nástrojů pro kolaboraci, jako jsou Pull Requesty a Code Reviews. GitHub však byl v konfiguraci izolovaným systémem, který nebyl efektivně integrován s dalšími nástroji, což komplikovalo sledování změn a nasazování.
- **Manuální proces nasazování** – bez integrovaného řešení pro CI/CD byli vývojáři nuceni spoléhat na manuální nasazování kódu na produkční servery. Tento postup byl časově náročný a náchylný k chybám, protože vyžadoval pečlivou koordinaci mezi vývojovými a operačními týmy.
- **Software Jira** – pro evidenci a správu projektů, poskytovala robustní řešení pro správu úkolů, sledování postupu a plánování v rámci vývojových týmů. Byla však neefektivně propojena s GitHubem, což vedlo k obtížnému sledování vztahu mezi úkoly a změnami v kódu.

- **Microsoft Excel** – použití Excelu jako nástroje pro sběr zpětné vazby a ukládání poznámek se ukázalo jako neefektivní. Data byla rozptýlena v různých souborech, což ztěžovalo jejich analýzu a sdílení mezi týmy.

Výzvou bylo najít řešení, které by všechny tyto roztržité části sjednotilo pod jednu střechu, aby se zvýšila celková efektivita a zlepšila komunikace. GitLab, s jeho integrovaným přístupem k verzování, CI/CD a správě projektů, se ukázal jako vhodná platforma pro dosažení těchto cílů. Jeho implementací se očekávalo zjednodušení pracovních postupů, zlepšení průhlednosti procesů a zvýšení produktivity práce týmů. Analýza také ukázala, že GitLab nabízí vylepšené možnosti pro zpětnou vazbu a dokumentaci, což by mělo vést k lepšímu porozumění potřebám zákazníků a efektivnějšímu řešení jejich požadavků.

4.7.1.3 Plánování projektu

Plánování projektu zahrnovalo vypracování podrobného plánu implementace, který specifikuje časový harmonogram, alokaci zdrojů, postupy pro správu rizik a strategii komunikace. V rámci agilního prostředí byl tento plán vytvořen s ohledem na flexibilitu i adaptabilitu a umožňoval průběžnou kontrolu. Plán musel být srozumitelný pro všechny členy týmu a měl by být pravidelně aktualizován v průběhu sprintů, což jsou krátké, časově omezené periody, ve kterých týmy pracují na přidělených úkolech.

Zahájení projektu implementace GitLabu bylo charakteristické aktivními kroky směřujícími k integraci a sjednocení procesů verzování a projektového managementu. Nejprve byl vytvořen detailní plán, který definoval kroky k implementaci, zahrnující specifický časový harmonogram a rozdělení zdrojů s důrazem na flexibilitu agilního prostředí.

Konkrétní kroky zahrnovaly:

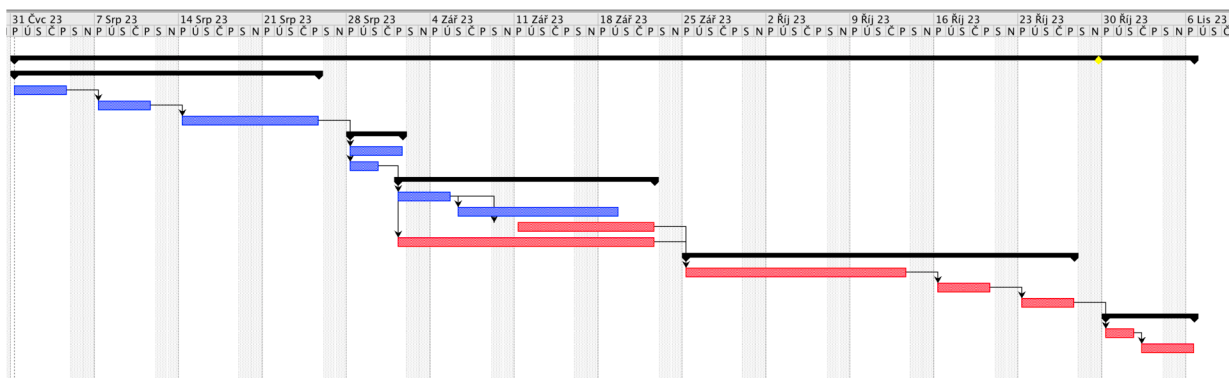
- **Vypracování časového plánu** – byl stanoven harmonogram s jasnými cíly, včetně počáteční analýzy, plánování, konfigurace, testování a nasazení GitLabu. Harmonogram byl designován jako dynamický, umožňující snadné sledování pokroku a přizpůsobení se změnám v průběhu projektu.

- **Alokace zdrojů** – provádělo se hodnocení potřebných lidských, finančních a technologických zdrojů. Rozpočet byl pečlivě rozdělen podle fází. Tento krok bude detailněji popsán v budoucí kapitole.
- **Adaptace na agilní metody** – proběhla příprava na pružné řízení projektu, což znamenalo, že plán byl navržen tak, aby umožňoval průběžné revize a úpravy. To zahrnovalo nastavení pravidelných kontrolních bodů po každém sprintu, během nichž byly úkoly přidělovány a postup byl monitorován.
- **Průběžná inspekce a adaptace** – v souladu s agilními principy byl plán projektu vytvořen s očekáváním, že se bude vyvíjet a měnit. To umožnilo týmu efektivně reagovat na jakékoliv nově vzniklé výzvy či překážky.

Díky těmto krokům bylo zajištěno, že základní plánovací fáze projektu byla postavena na solidním základě, který přihlížel k agilnímu prostředí a potřebám firmy, a současně poskytovala dostatečný prostor pro flexibilitu a inovace (viz obr. 12, podrobnější znázornění je v příloze 1).

4.7.1.4 Ganttův diagram

Obrázek 12 – Znázornění procesů při zavedení systému formou Ganttova diagramu



Zdroj: Vlastní zpracování (ProjectLibre)

4.7.2 Fáze návrhu a vývoje

Fáze návrhu a vývoje projektu implementace GitLabu představovala klíčovou etapu, ve které se formují základy pro úspěšnou integraci a nasazení systému. Tento segment

projektu je rozdělen do dvou hlavních oblastí, tedy technického návrhu a finančního plánování, a každá s vlastními specifickými cíli a výzvami. Zodpovědný tým by měl úzce spolupracovat s ostatními odděleními, aby se ujistil, že technické řešení odpovídá obchodním cílům a uživatelským potřebám.

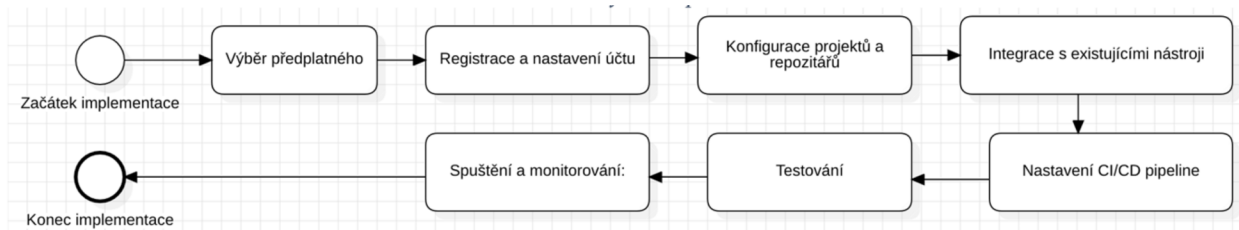
4.7.2.1 Technický návrh

V rámci fáze návrhu a vývoje se oddělení programátorů soustředilo na specifické úkoly spojené s implementací informačního systému GitLab. Aktivity zahrnovaly detailní analýzu dostupných funkcí GitLabu, určení nejlepšího způsobu jejich využití pro zefektivnění vývojových a operačních procesů a integraci GitLabu s ostatními nástroji a systémy používanými ve firmě. Cílem bylo použít robustní, bezpečný a efektivní nástroj, který bude podporovat integraci, dodávání a automatizaci pracovního procesu, přičemž zohlední také budoucí potřeby rozvoje. Zde jsou kroky, které tomu předcházely:

- **Analýza technických požadavků** – tento krok zahrnoval důkladný průzkum, jak GitLab může podpořit a zlepšit stávající pracovní procesy. Programátoři zkoumají, jak mohou využít GitLab pro zlepšení správy kódu, spolupráce na projektu, sledování problémů a automatizace CI/CD. Dále analyzují, jak GitLab zapadá do agilních metodik a jak může podporovat Scrum procesy.
- **Plánování zabezpečení a soukromí** – zabezpečení je kritickým aspektem, zejména pokud jde o správu a ukládání zdrojového kódu. Programátoři zajišťují, že implementace GitLabu splňuje všechny firemní bezpečnostní standardy a ochranu soukromí.

Při implementaci GitLabu jako předplatné služby, tedy bez nutnosti nasazování na vlastním serveru, zahrnoval technický postup několik důležitých kroků, které zajišťují hladký průběh integrace a přizpůsobení systému podle potřeb podniku. Na obrázku č. 13 jsou uvedeny základní technické kroky:

Obrázek 13 – Technický návrh procesů



Zdroj: Vlastní zpracování (StarUML)

Tento postup měl zajišťovat, že implementace GitLabu, jako cloudové služby, proběhne hladce a systém bude efektivně sloužit potřebám podniku, aniž by bylo nutné investovat do vlastní serverové infrastruktury.

4.7.2.2 Rozpočet a finanční plánování

Fáze rozpočtu a finančního plánování byla nezbytná pro zajištění, že zavedení informačního systému GitLab do organizace je finančně udržitelné a že jsou správně alokovány zdroje pro jednotlivé role potřebné k úspěšné implementaci. Nacenení bylo sestaveno s ohledem na všechny fáze implementace GitLabu, včetně nákladů na licence, školení personálu a další přidružené náklady. Důležité bylo zahrnout nejen přímé náklady, ale také nepřímé náklady, jako jsou náklady během přechodu na GitLab a potenciální rizika související s implementací.

Vedení firmy po důkladném zvážení všech dostupných možností (viz tab. 3) a s přihlédnutím k specifickým potřebám organizace se rozhodlo využít placené služby „Premium“ přímo od výrobce softwaru. Toto rozhodnutí bylo motivováno několika klíčovými faktory, které jsou uvedeny níže.

Plán Premium poskytuje rozšířené nástroje pro spolupráci, které jsou zásadní pro podporu agilního vývoje a kolaborativních pracovních procesů. S funkcemi, jako jsou ochrana větví a pravidla pro schvalování Merge Requestů, Premium plán umožňuje lepší kontrolu nad kódem a optimalizuje procesy pro kontrolu výsledného kódu.

Vzhledem k rostoucímu počtu projektů a týmů bylo nezbytné zvolit řešení, které může snadno škálovat. Plán Premium s 50 GiB úložiště a 10 000 CI/CD minutami za měsíc nabízí dostatečné zdroje pro podporu rostoucího objemu práce bez nutnosti častého vylepšování plánu nebo obav z překročení limitů.

Funkce automatizace CI/CD a pokročilé analytické nástroje v plánu Premium umožnily firmě zefektivnit vývojový cyklus a urychlit nasazení nových funkcí a aktualizací. Tato efektivita vede k rychlejšímu uvedení produktů na trh, což představuje konkurenční výhodu.

Tabulka 3 – Možnosti předplatného ve službě GitLab

Plán	Cena	Funkce
Free	\$0 za uživatele/měsíc	Úložiště: 5 GiB Transfer: 10 GiB za měsíc Výpočetní minuty: 400 za měsíc Uživatelů na top-level skupinu: 5 Určeno pro: Osobní projekty a malé týmy začínající s GitLabem Funkce: Základní GitLab funkce
Premium	\$29 za uživatele/měsíc	Vše z Free plánu + Pokročilé funkce: Code Ownership, Protected Branches, Merge Requests s pravidly schválení, pokročilé CI/CD Úložiště: 50 GiB Transfer: 100 GiB za měsíc Výpočetní minuty: 10 000 za měsíc Určeno pro: Rozšiřující se organizace a využití více týmů
Ultimate	Kontaktujte prodejce	Vše z Premium plánu + Vlastnosti: Suggested Reviewers, Dynamic Application Security Testing, Security Dashboards, Vulnerability Management, Dependency Scanning Úložiště: 250 GiB Transfer: 500 GiB za měsíc Výpočetní minuty: 50 000 za měsíc Bezplatné hostitelské účty Určeno pro: Podniky hledající rychlejší dodávání softwaru s důrazem na bezpečnost, dodržování a plánování na úrovni celé organizace

Zdroj: Vlastní zpracování

Dle kurzovního lístku ze dne 28.8.2023 byl směnný kurz amerického dolaru vůči české koruně 22.34 Kč. V rámci firmy bylo potřeba celkem 18 účtů pro 18 zaměstnanců. Po přepočtení na koruny české a sečtení všech účtů byly celkové měsíční náklady na chod systému 11 661 Kč.

Tabulka mzdových nákladů prezentuje hierarchii pozic v různých odděleních společnosti a odpovídající hodinové sazby. Každé oddělení má vedoucí pozici, jejíž hodinová sazba je o 10 % vyšší než u ostatních zaměstnanců v daném oddělení. Dle souhrnu platové sféry ČR za 1. pololetí roku 2023 podle informačního systému o průměrném výdělků (ISPV) bylo ke každé pozici přiřazeno oborově nejbližší kategorie zaměstnání a medián hrubého pracovního platu se vydělil placenou dobou.

Při určení mzdových nákladů společnosti dochází k párování pozic s kategoriemi definovanými v Informačním systému o průměrném výdělků (ISPV), což umožňuje srovnání s průměrnými tržními hodnotami (viz tab. 4). Například, pozice Ředitel firmy je přiřazena kategorii "Personální náměstci (ředitelé)" s kódem ISPV 12121, reflektující jejich roli v personálním řízení a strategickém rozhodování. Podobně, Technický ředitel je přiřazen k "Náměstkům (ředitelům) pro technický rozvoj, výzkum a vývoj" s kódem ISPV 12231, což odráží zaměření na inovace a technologický pokrok.

Programátoři jsou přiřazeni jako "2514, Programátoři počítačových aplikací specialisté", designéři jako "21631, Průmysloví a produktoví designéři", marketéři jako "24311, Specialisté v oblasti marketingu", projektový manažeři jako "12232, Řídící pracovníci v oblasti technického rozvoje", obchodní manažeři jako "33392, Obchodní referenti" a účetní jako "24111, Hlavní účetní".

Tabulka 4 – Mzdové náklady

Oddělení	Pozice	Počet	Hodinová sazba	Hodinová sazba oddělení
Vedení firmy	Ředitel firmy	1	571 Kč	1 113 Kč
	Technický ředitel	1	542 Kč	
Oddělení programátorů	Vedoucí programátorů	1	308 Kč	1 428 Kč
	Programátor	4	280 Kč	
Oddělení designu	Vedoucí designérů	1	254 Kč	716 Kč
	Designér	2	231 Kč	
Oddělení marketingu	Vedoucí marketingu	1	290 Kč	818 Kč
	Marketér	2	264 Kč	
Projektové oddělení	Projektový vedoucí	1	415 Kč	792 Kč
	Projektový manažer	1	377 Kč	
Oddělení financí a obchodu	Finanční a obchodní vedoucí	1	383 Kč	985 Kč
	Obchodní manažer	1	348 Kč	
	Účetní	1	254 Kč	

Zdroj: Vlastní zpracování

Odhad nákladů na školení byl ve výši 32 000 Kč. Tento odhad byl výsledkem důkladného hledání na trhu s cílem najít vhodného konzultanta, který by dokázal poskytnout školení odpovídající potřebám firmy a nově zaváděnému systému.

4.7.3 Implementační fáze

Implementační fáze projektu zavedení informačního systému GitLab zastávalo klíčovou roli v úspěšném nasazení a adopci této platformy v podnikovém prostředí. Tato fáze se obecně zabývá praktickými aspekty přechodu od teoretického návrhu k funkčnímu systému.

4.7.3.1 Nastavení a konfigurace GitLabu

- **Výběr edice GitLabu** – pro potřeby projektu byla zvolena GitLab Enterprise Edition (EE). Tato edice byla preferována pro své rozšířené funkce a možnosti, které překračují to, co nabízí Community Edition. GitLab EE poskytuje pokročilé nástroje pro správu, větší škálovatelnost a lepší integraci s externími nástroji, což jsou základní požadavky pro komplexní projekty organizace.
- **Volba předplatného GitLabu** – v rámci edice GitLab Enterprise Edition bylo zvoleno předplatné Premium. Toto předplatné bylo preferováno pro více pokročilých

funkcí oproti předplatnému Free. Premium účet poskytuje pokročilejší nástroje pro správu a vývoj, větší škálovatelnost, více úložného prostoru, více výpočetních minut na externích serverech a možnosti více týmů.

- **Výběr hostovaného řešení** – pro implementaci bylo rozhodnuto využít GitLab.com, což je hostovaná služba přímo od výrobce software, namísto vlastního hostovaného řešení. Toto rozhodnutí bylo učiněno na základě řady faktorů, včetně snížení nároků na interní IT zdroje, eliminace potřeby správy serverů a infrastruktury, využití vysoké dostupnosti a zabezpečení, které GitLab.com poskytuje. Dále byla výhodou lepší škálovatelnost služby v reakci na růst a rozvoj projektů.
- **Konfigurace GitLabu** – po registraci na GitLab.com byla provedena počáteční konfigurace zahrnující nastavení projektů, uživatelských účtů, skupin a přístupových oprávnění. Důraz byl kladen na vytvoření robustního a bezpečného prostředí, s využitím pokročilých funkcí GitLab EE pro nastavení procesů Continuous Integration/Continuous Deployment (CI/CD), sledování a správu projektů. Nastavení zahrnovalo také definování pravidel pro revizi kódu, správu větví a nastavení integrace s externími systémy.

Výběr GitLab EE a GitLab.com jako hostované platformy poskytovalo organizaci flexibilní a škálovatelné řešení pro správu softwarových projektů.

4.7.3.2 Migrace dat a integrace systému

Migrace dat a jejich integrace do nového systému představovalo důležitou fázi pro softwarových nástrojů. Efektivní přenos dat a jejich bezproblémová integrace byly klíčové pro minimalizaci výpadků a zachování integrity informací. Proces migrace byl prováděn během pracovních dnů a byl dělán současně s vytvářením nových dat. V následujícím textu bude popsán proces migrace dat a integrace systému, který byl realizován v rámci zavádění systému GitLabu v rámci podniku.

- **Migrace ze systému GitHub** – nejprve se provedl export zdrojových kódů a repozitářů, zahrnující všechny větve a štítky, aby se zachovala kompletní historie

verzí. Dále byla data upravena pro kompatibilitu s GitLabem, což zahrnovalo převod formátů a uživatelských oprávnění.

- **Migrace ze systému Jira** – pro přesun z Jira do GitLabu bylo nejprve nutné exportovat všechny úkoly, sprinty a projektové náležitosti. Tato data byla pak transformována tak, aby odpovídala strukturám GitLabu. Všechny úkoly byly řádně přiřazeny s odpovídajícími metadaty, včetně stavů, popisků a komentářů.
- **Migrace z Microsoft Excel** – migrace z Excelu na GitLab vyžadovala konverzi tabulkových dat do formátu, který mohl být importován jako Issues nebo dokumentace v GitLabu. Poznámky a zpětná vazba byly převedeny do příslušných diskusních vláken nebo komentářů u Issues, zatímco kvantitativní data byla importována do GitLabu pro generování reportů a metrik.

Po migraci a integraci následovala fáze testování, během které se ověřovala správnost přenesených dat a funkčnost integrovaných systémů. Testování probíhalo na všech úrovních, od jednotlivých funkcí po celkové procesy.

4.7.3.3 Školení a podpora

Školení a podpora byly zásadní pro zajištění hladkého přechodu na nový systém a jeho úspěšného přijetí uživateli. Efektivní program školení umožňuje zaměstnancům rychle získat potřebné dovednosti a znalosti, zatímco kvalitní podpora zajišťuje rychlé řešení jakýchkoliv vznikajících otázek nebo problémů. Tato kapitola se věnuje přístupům, metodám a strategiím školení a podpory, které byly použity při implementaci GitLabu.

- **Firemní školení** – firma si zaplatila školení externím konzultantem se specializací na GitLab. Školení zaměřené na implementaci GitLabu bylo strukturováno do několika segmentů. Začínalo se úvodem, kde účastníci získali pochopení toho, co GitLab je a proč by měl být používán, spolu s rozdíly mezi GitLabem a GitHubem. Následovala technická část, která pokrývala instalaci GitLabu, základní a pokročilou konfiguraci, administraci uživatelů, skupin a repozitářů a nastavení serveru. Důraz byl kladen na praktické dovednosti, jako je správa repozitářů, práce s větvemi a Merge Requesty. Wiki a Issues byly

představeny s důrazem na efektivní správu a sledování projektových milníků. Docker registry byla probrána z hlediska správy kontejnerů, zatímco GitLab Pages byly prezentovány jako nástroj pro publikaci webových stránek. Kurz dále zahrnoval práci se Snippets, což jsou drobné fragmenty kódu používané pro různé účely v rámci vývoje softwaru.

- **Samostudium** – procházení dokumentace bylo také důležitou součástí školení, kde firemní pracovníci byli povzbuzeni k proaktivnímu procházení oficiální dokumentace GitLabu. Tato fáze umožňovala zaměstnancům setkat se s celou škálou nástrojů a funkcí, které GitLab nabízí, a to přímo v kontextu jejich individuálních pracovních úkolů. Studium dokumentace pomáhalo rozvíjet hlubší porozumění systému a zároveň posilovalo schopnost samostatně vyhledávat informace a řešit problémy.
- **Retrospektivy** – školení vedené vedením firmy bylo zaměřeno na konkrétní problémy a výzvy, které vyplynuly během implementace GitLabu. Během těchto firemních schůzek měli zaměstnanci možnost sdílet zkušenosti, diskutovat o překážkách a společně hledat řešení. Interakce s vedením umožňovala nejen rychlejší adaptaci na nové procesy, ale také podporovala soudržnost týmu a ujistila se, že všechny oddělení mají jasnou představu o tom, jak GitLab efektivně využívat.

Školení a podpora hrály klíčovou roli ve zvyšování efektivity a spokojenosti uživatelů při přechodu na nový systém. V rámci tohoto kroku byly popsány metody a postupy, které zajistily, že zaměstnanci byli dobře připraveni na práci s GitLabem a měli potřebné zdroje pro rychlé řešení jakýchkoliv problémů.

4.7.3.4 Průběžné finanční sledování

Průběžné finanční sledování bylo nezbytné pro zajištění ekonomické efektivity a zodpovědnosti v rámci každého projektu implementace softwaru. Umožňuje manažerům projektů a finančním analytikům monitorovat výdaje, porovnávat je s rozpočtem a včas identifikovat jakékoli odchylky. V krocích, které jsou uvedeny níže, je pohled na postupy, které byly zavedeny pro sledování a řízení finančních aspektů implementace GitLabu.

- **Sledování a analýza nákladů** – v této fázi byly průběžně sledovány všechny náklady a investice související s implementací, včetně nákladů na licence, školení, vývojářské práce a migrace. Byly vytvořeny pravidelné finanční reporty, které poskytovaly přehled o aktuálním stavu výdajů.
- **Finanční reportování a rozhodování** – pravidelné finanční reporty byly nejen nástrojem pro kontrolu, ale také základem pro rozhodování. Zajišťovaly, že jakékoli finanční odchylky byly rychle identifikovány a řešeny, a umožňovaly manažerům projektu přizpůsobit plány a alokace zdrojů podle aktuálních potřeb a situace.

Průběžné finanční sledování bylo zásadní pro zachování kontroly nad rozpočtem projektu a zajištění jeho ekonomické úspěšnosti. V této kapitole byly krátce představeny procesy finančního sledování, které byly implementovány pro zajištění transparentnosti a efektivního využívání finančních zdrojů v průběhu zavádění GitLabu.

4.7.4 Fáze testování

Fáze testování bylo rozhodujícím obdobím v procesu implementace softwaru, kde se ověřuje funkčnost, výkon, a spolehlivost systému před jeho finálním spuštěním do produkčního prostředí. Kvalitní testování zajistí, že systém bude schopen splnit všechny požadavky bez významných chyb nebo problémů, což je zásadní pro bezproblémový běh a přijetí uživateli. Tato kapitola popisuje různé aspekty testovací fáze projektu, včetně nástrojů a technik použitých pro zajištění úspěšného přechodu na GitLab.

4.7.4.1 Testování a kontrola v provozu

Hlavní částí fáze testování byla testovací kontrola v provozu, což je proces zaměřený na ověření, zda systém splňuje stanovené požadavky a zda je připraven pro každodenní operace v rámci organizace. Tato sekce se detailně věnuje následujícím bodům:

- **Testování CI/CD** – testování CI/CD procesů v GitLabu probíhalo pomocí sérií automatizovaných Pipeline, které byly navrženy tak, aby simulovaly reálné vývojové a provozní scénáře. Tyto Pipeline zahrnovaly kroky jako je kompilace kódu, automatizované testy, nasazení na testovací servery a následné produkční nasazení.

- **Bezpečnost** – bezpečnostní testování se soustředilo na identifikaci a minimalizaci bezpečnostních rizik, jako správně nastavená hesla či nastavení pracovních zařízení na straně zaměstnanců. Součástí toho bylo také testování uživatelských oprávnění.
- **Migrace** – testy migrace dat se zaměřily na ověření, že všechna data byla z předchozího systému přenesena do GitLabu bez ztráty nebo poškození. Kontrolovalo se, zda jsou všechny záznamy korektní a zda odpovídají očekávaným hodnotám.
- **Workflow** – testování workflow zahrnovalo praktické testování všech definovaných pracovních postupů v GitLabu, aby se zajistilo, že všechny procesy probíhají hladce a efektivně a že jsou správně nastaveny vazby mezi jednotlivými kroky a úkoly.

4.7.4.2 Identifikace a řešení případných problémů

Po dokončení primární fáze testování a kontroly v provozu následovala klíčová etapa zaměřená na identifikaci a řešení všech vzniklých problémů. Tento krok byl zásadní pro zajištění, že všechny aspekty systému GitLab jsou bez chyb a připraveny pro každodenní operace.

Během testovací fáze implementace cloudového řešení GitLab došlo k problému s GitLab Runners (viz obr. 14), které jsou klíčové pro automatizaci nasazování kódu na produkční servery. Tento problém se projevil tak, že Runners nereagovaly na spouštěcí příkazy nebo selhávaly při pokusu o nasazení kódu. Aby se tato situace vyřešila, bylo nejprve nutné provést diagnostiku prostřednictvím logů Runners a zkontrolovat konfigurace CI/CD Pipeline. Jakmile byly identifikovány konkrétní chyby v konfiguraci nebo v komunikaci s cloudovou infrastrukturou, bylo možné aplikovat opravy a optimalizace, včetně aktualizace Runnerů, přepracování Pipeline skriptů, úpravy oprávnění a síťového nastavení, aby bylo zajištěno plynulé a efektivní nasazování.

Obrázek 14 – Ukázka nahlášené chyby během testování v systému GitLab

CI/CD pipeline fails

Edit

Open Issue created 50 minutes ago by Libor Novák

Při provádění testů CI/CD procesů po migraci dat do GitLabu jsem narazil na problém s GitLab Runners, které nezpracovávaly spouštěcí příkazy a selhávaly při pokusu o nasazení kódu na produkční servery. Tento problém je reprodukovatelný a byl zaznamenán ve více pokusech o nasazení, což naznačuje systémovou chybu. Runners vykazují chybové kódy, které jsou zaznamenány a připraveny k další analýze. Žádám o revizi konfigurace CI/CD pipeline a runnerů, aby byla zajištěna jejich správná funkčnost.

👍 0 🙏 0 😊

Create merge request

Zdroj: Vlastní zpracování (GitLab.com)

V průběhu bezpečnostního testování byl identifikován problém, kdy vývojář neměl přístup k repozitáři jednoho z projektů v GitLabu, což mu znemožňovalo stahování projektu a ukládání nových změn. Tento problém byl způsoben chybným nastavením uživatelských oprávnění. Řešení spočívalo v revizi a úpravě nastavení oprávnění pro daného vývojáře, aby byla zajištěna správná úroveň přístupu k repozitáři. Po úpravě oprávnění bylo potvrzeno, že vývojář již má možnost repozitář stahovat a upravovat obsah. Tato zkušenost vedla k důkladnější kontrole a aktualizaci správy oprávnění, aby se podobné situace v budoucnosti neopakovaly.

Toto byl popis dvou náhodných problémů během testování zavádění nového systému GitLab v podniku. Všechny chyby byly zaznamenány formou Issues v systému GitLab a následně byly řešeny a úspěšně vyřešeny vždy příslušným oddělením. V tabulce č. 5 je graficky zobrazeno výsledné hlášení z celkového testování.

Tabulka 5 – Výsledky testování

Výsledky testování - report chyb	
Kategorie	Počet hlášených chyb
CI/CD	3
Bezpečnost	2
Migrace	1
Workflow	2

Zdroj: Vlastní zpracování

4.7.4.3 Dokumentace

Tvorba dokumentace byla důležitým úkolem, který zajišťuje, že všechny týmy jsou řádně informovány o vlastnostech a výhodách nově implementovaného systému. Vytvoření dokumentace GitLabu bylo provedeno interními zaměstnanci a byla k tomu využita funkcionality systému GitLab zvaná Wiki. Tento proces začal definováním struktury a obsahu dokumentace, kde jednotlivé klíčové oblasti nebo funkce byly diskutovány s konkrétními zaměstnanci nebo týmu s relevantními znalostmi.

Zaměstnanci byli instruováni vedením firmy, jak vytvořit a formátovat stránky Wiki, aby zajistili jednotnost a srozumitelnost celé dokumentace. Důraz byl kladen na určitý styl a terminologii, aby bylo zajištěno, že dokumentace bude konzistentní a snadno pochopitelná pro všechny zaměstnance.

Dokumentaci tvořily návody, postupy, nejlepší praktiky, často kladené otázky a řešení běžných problémů, přičemž každý dokument prošel procesem revize a schválení, než byl publikován. Proces také zahrnoval pravidelné aktualizace a údržbu dokumentace, aby byly informace stále aktuální a odrážely nejnovější změny v systému GitLab.

Během celého procesu tvorby dokumentace byl kladen důraz na detailní a přesné informace, zároveň s udržení vysoké úrovně přístupnosti a uživatelského komfortu. Tímto způsobem se podařilo vytvořit ucelený a komplexní dokumentační zdroj přímo v prostředí GitLabu, který sloužil jako hlavní informační bod pro všechny zaměstnance.

4.7.5 Závěrečná fáze

Závěrečná fáze projektu byla klíčová pro zhodnocení celkového úspěchu a přijetí implementovaného řešení. V této fázi bylo důležité zhodnotit všechny aspekty projektu, od technické implementace až po uživatelskou zkušenost. Hlavním cílem bylo získat objektivní zpětnou vazbu od všech zúčastněných stran a na jejím základě provést závěrečné hodnocení projektu. Během všech předchozích kroků se vždy na konci jednotlivých procesů konaly informativní schůzky, kde docházelo k získávání zpětné vazby a kontroly, zda jde vše podle plánu.

4.7.5.1 Zjišťování zpětné vazby

Zpětná vazba byla shromažďována pomocí rozhovorů mezi vedením a členy jednotlivých oddělení a dotazníků. Tento proces umožňoval týmu získat cenné informace o uživatelské spokojenosti, efektivitě nově zavedených procesů a identifikaci oblastí pro další zlepšení (viz obr. 15).

Data byla sbírána od všech uživatelů systému a byla analyzována s cílem posoudit efektivitu implementovaných řešení. Výsledky dotazníků byly interpretovány tak, aby odhalily klíčové silné stránky a oblasti, kde je třeba zlepšení. Specifická kritika odhalená v dotaznících pak posloužila jako základ pro plánování dalších kroků ve vývoji a optimalizaci procesů v GitLabu.

Obrázek 15 – Zpětná vazba z dotazníků

Zpětná vazba

Edit

Open Issue created 6 hours ago by Roman Černý

Oblast	Komentář
Uživatelské rozhraní a zkušenost	Celkově je rozhraní GitLabu uživatelsky přívětivé, ale našel jsem některé nástroje a nastavení, které nebyly snadno identifikovatelné bez hlubšího ponoření se do dokumentace. Konkrétně, proces nastavení správných přístupových práv k repozitářům pro různé uživatelské role byl komplexnější, než jsem očekával.
Funkčnost a integrace nástrojů	Funkce pro správu kódu jsou robustní a výkonné, avšak při pokusu o integraci s externími nástroji pro sledování chyb jsem narazil na určitá omezení, která bych rád viděl v budoucích aktualizacích.
Výkon a stabilita	Byl jsem překvapen, jak GitLab zvládá současnou správu více projektů bez znatelného poklesu výkonu. Přesto se objevily občasné problémy s prodlevami, když CI/CD pipeline zpracovávala komplexní úlohy, což mělo za následek mírné zpoždění ve vývojovém cyklu.
Podpora a dostupnost zdrojů	Dokumentace je obsáhlá, ale někdy se zdá být příliš teoretická. Ceníl bych si většího množství případových studií a tutoriálů, které bych mohl využít jako příklady pro řešení běžných vývojářských a projektových úkolů.
Celková spokojenost a efektivita práce	Systém GitLab zásadně zlepšil efektivitu našeho vývoje a usnadnil kolaboraci v týmu. Přesto si myslím, že by mohl být vylepšen systém oznámení, aby byl více přizpůsobivý a umožňoval lepší sledování změn a aktualizací v reálném čase.

Zdroj: Vlastní zpracování (GitLab.com)

4.7.5.2 Závěrečná zpráva a hodnocení

Závěrečnou zprávu vytvořilo vedení firmy a ta tvořila finální hodnotící fázi, kde se sumarizují všechny aspekty implementace GitLabu. Informace o finančním hodnocení a o hodnocení zavedení nového systému bude více rozebráno v další hlavní kapitole.

4.8 Finanční a časový harmonogram implementace nového systému

Finanční a časový harmonogram podává přehled o vykonaných finančních a časových úkonech spojených se zaváděním systému GitLab. Obsahuje celkové náklady na projekt, ale i náklady na jednotlivé fáze v průběhu implementace. Dále zahrnuje podrobný časový plán s alokací času podle jednotlivých oddělení firmy. Tento harmonogram byl zásadní pro posouzení úspěšnosti zavedení nového systému a sloužil k potřebám vytvoření závěrečné zprávy.

4.8.1 Přípravná fáze

V první fázi byl kladen důraz na definování cílů a analýzu současného stavu, na což bylo alokováno 50% denní kapacity vedení firmy, aby se minimalizoval dopad na běžný provoz. Celá fáze trvala celkem 20 dnů (viz obr. 16).

Obrázek 16 - Procesy v přípravné fázi

☐ Přípravná fáze	20 dní
Definice cílů projektu	5 dní
Analýza současného stavu	5 dní
Plánování projektu	10 dní

Zdroj: Vlastní zpracování (ProjectLibre)

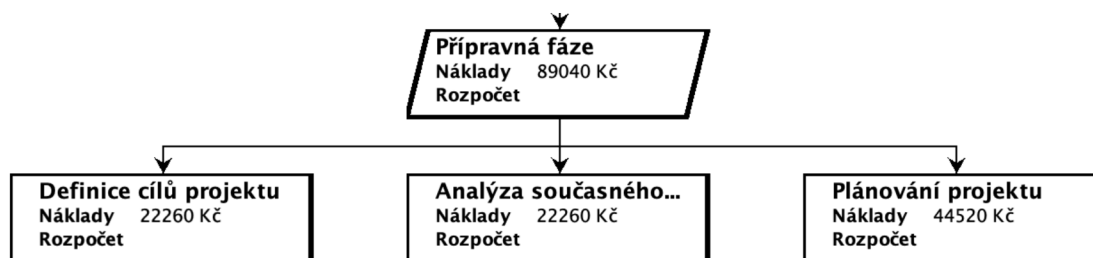
Celkové mzdové náklady na tuto fázi činily 89 040 Kč (viz obr. 17), přičemž část byla přidělena rovnoměrně mezi definici cílů, analýzu současného stavu a plánování projektu. Největší podíl měl proces plánování projektu. Pro výpočet se tedy použila **((hodinová sazba oddělení vedení firmy * 8) * počet dnů) / 2** (viz tab. 6).

Tabulka 6 – Detailní rozpis nákladů přípravné fáze podle dílčích fází

Definice cílů projektu					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	5	50%	22 260 Kč
Celkové náklady					22 260 Kč
Analýza současného stavu					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	5	50%	22 260 Kč
Celkové náklady					22 260 Kč
Plánování projektu					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	10	50%	44 520 Kč
Celkové náklady					44 520 Kč

Zdroj: Vlastní zpracování

Obrázek 17 – Náklady na přípravnou fázi



Zdroj: Vlastní zpracování (ProjectLibre)

4.8.2 Fáze návrhu a vývoje

Ve fázi návrhu a vývoje, která zahrnovala technický návrh a finanční plánování, byl zachován stejný přístup s využitím 50% kapacity oddělení, aby se podpořila detailní příprava na technické aspekty projektu a zároveň efektivně spravovaly finance. Na podprocesu technický návrh pracovalo pouze oddělení programátorů a rozpočet a finanční plánování mělo na starosti vedení společnosti. Celkově tato fáze trvala 5 dnů (viz obr. 18).

Obrázek 18 – Procesy ve fázi návrhu a vývoje

<input checked="" type="checkbox"/> Fáze návrhu a vývoje	5 dní
Technický návrh	5 dní
Rozpočet a finanční plánování	3 dní

Zdroj: Vlastní zpracování (ProjectLibre)

Celkové mzdové náklady na tuto fázi dosáhly 41 916 Kč (viz obr. 19). Náklady za jednotlivé podprocesy byly vypočteny **((hodinová sazba daného oddělení * 8) * počet dnů) / 2** (viz tab. 7).

Tabulka 7 – Detailní rozpis nákladů fáze návrhu a vývoje podle dílčích fází

Technický návrh					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Oddělení programátorů	1 428 Kč	11 424 Kč	5	50%	28 560 Kč
Celkové náklady					28 560 Kč
Rozpočet a finanční plánování					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	3	50%	13 356 Kč
Celkové náklady					13 356 Kč

Zdroj: Vlastní zpracování

Obrázek 19 – Náklady na fázi návrhu a vývoje



Zdroj: Vlastní zpracování (ProjectLibre)

4.8.3 Implementační fáze

Implementační fáze projektu zavedení informačního systému GitLab hrála klíčovou roli v úspěšném nasazení a adopci této platformy v podnikovém prostředí. Tato fáze se obecně zabývala přechodem z teoretického návrhu k funkčnímu systému. Jelikož vedení firmy nechtělo narušit aktuální workflow a chtělo zachovat zdravou činnost firmy, tak vyhradilo na implementaci pouze 25 % časové kapacity zaměstnanců.

Tato fáze celkově trvala 16 dní (viz obr. 20). Na podprocesu nastavení a konfigurace GitLabu se podílel pouze tým programátorů. Na migracích dat a školení se podílely rovnoměrně všechna oddělení firmy. Průběžné finanční sledování vykonávalo pouze vedení firmy.

Obrázek 20 – Procesy v implementační fázi

☐ Implementační fáze	16 dní
Nastavení a konfigurace GitLabu	3 dní
Migrace dat a integrace systému	10 dní
Školení a podpora	10 dní
Průběžné finanční sledování	16 dní

Zdroj: Vlastní zpracování (ProjectLibre)

Implementační fáze celkově firmu stála 278 264 Kč (viz obr. 21). Výpočet pro nastavení a konfigurace GitLabu byl vypočítán jako **((hodinová sazba oddělení programátorů * 8) * počet dnů) / 4**. Procesy migrace a školení byly vykonávány napříč celou firmou, tudíž byla vypočítány pro každé oddělení zvlášť a výsledné náklady činily součet všech nákladů daných oddělení. Výpočet tedy byl **((hodinová sazba daného oddělení * 8) * počet dnů) / 4**. A nakonec průběžné finanční sledování bylo počítáno podobně jako implementační fáze, tedy **((hodinová sazba vedení firmy * 8) * počet dnů) / 4** (viz tab. 8).

Tabulka 8 – Detailní rozpis nákladů implementační fáze podle dílčích fází

Nastavení a konfigurace					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Oddělení programátorů	1 428 Kč	11 424 Kč	3	25%	8 568 Kč
Celkové náklady					8 568 Kč
Migrace dat a integrace systému					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	10	25%	22 260 Kč
Oddělení programátorů	1 428 Kč	11 424 Kč	10	25%	28 560 Kč
Oddělení designérů	716 Kč	5 728 Kč	10	25%	14 320 Kč
Oddělení marketingu	818 Kč	6 544 Kč	10	25%	16 360 Kč
Finanční a obchodní oddělení	792 Kč	6 336 Kč	10	25%	15 840 Kč
Projektové oddělení	985 Kč	7 880 Kč	10	25%	19 700 Kč
Celkové náklady					117 040 Kč
Školení a podpora					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	10	25%	22 260 Kč
Oddělení programátorů	1 428 Kč	11 424 Kč	10	25%	28 560 Kč
Oddělení designérů	716 Kč	5 728 Kč	10	25%	14 320 Kč
Oddělení marketingu	818 Kč	6 544 Kč	10	25%	16 360 Kč
Finanční a obchodní oddělení	792 Kč	6 336 Kč	10	25%	15 840 Kč
Projektové oddělení	985 Kč	7 880 Kč	10	25%	19 700 Kč
Celkové náklady					117 040 Kč
Průběžné finanční sledování					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	16	25%	35 616 Kč
Celkové náklady					35 616 Kč

Zdroj: Vlastní zpracování

Obrázek 21 – Náklady na implementační fázi



Zdroj: Vlastní zpracování (ProjectLibre)

4.8.4 Fáze testování

Během testovací fáze bylo zapotřebí zajistit, že systém funguje bez chyb, což znamenalo, že 25% časové kapacity oddělení bylo dostatečných pro provádění testů a opravy případných chyb. Toto časové rozložení bylo opět zvoleno, aby nenarušovalo chod firmy. Pro proces testování a kontroly v provozu byly zaměstnány všechna oddělení firmy.

Identifikaci a řešení případných problémů řešilo vedení firmy, oddělení programátorů a oddělení marketingu, protože měly největší znalosti v oblasti nového systému. Samotnou dokumentací bylo pověřeno oddělení marketingu, kvůli jeho zkušenostem v oblasti copywritingu a vytváření reklamních materiálů. Celkem tato fáze trvala 27 dní (viz obr. 22).

Obrázek 22 – Procesy ve fázi testování

☐ Fáze testování	27 dní
Testování a kontrola v provozu	15 dní
Identifikace a řešení případných problémů	7 dní
Dokumentace	5 dní

Zdroj: Vlastní zpracování (ProjectLibre)

Celkové mzdové náklady na testovací fázi činily 230 766 Kč (viz obr. 23). Testování a kontrola byla součtem prací všech oddělení, kdy bylo vyčleněno zmíněných 25 % časové kapacity. K těmto nákladům se došlo výpočtem ***((hodinová sazba daného oddělení * 8) * počet dnů) / 4*** a následným sečtením všech oddělení. Podobný výpočet se použil i na identifikaci a řešení případných problémů s tím, že se to týkalo pouze vedení společnosti, oddělení programátorů a oddělení marketingu. Proces dokumentace bylo nákladem pouze oddělení marketingu. Výpočet tedy byl ***((hodinová sazba oddělení marketingu * 8) * počet dnů) / 4*** (viz tab. 9).

Tabulka 9 – Detailní rozpis nákladů fáze testování podle dílčích fází

Testování a kontrola v provozu					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	15	25%	33 390 Kč
Oddělení programátorů	1 428 Kč	11 424 Kč	15	25%	42 840 Kč
Oddělení designérů	716 Kč	5 728 Kč	15	25%	21 480 Kč
Oddělení marketingu	818 Kč	6 544 Kč	15	25%	24 540 Kč
Finanční a obchodní oddělení	792 Kč	6 336 Kč	15	25%	23 760 Kč
Projektové oddělení	985 Kč	7 880 Kč	15	25%	29 550 Kč
Celkové náklady					175 560 Kč
Identifikace a řešení případných problémů					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	7	25%	15 582 Kč
Oddělení programátorů	1 428 Kč	11 424 Kč	7	25%	19 992 Kč
Oddělení marketingu	818 Kč	6 544 Kč	7	25%	11 452 Kč
Celkové náklady					47 026 Kč
Dokumentace					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Oddělení marketingu	818 Kč	6 544 Kč	5	25%	8 180 Kč
Celkové náklady					8 180 Kč

Zdroj: Vlastní zpracování

Obrázek 23 – Náklady na fázi testování



Zdroj: Vlastní zpracování (ProjectLibre)

4.8.5 Závěrečná fáze

V závěrečné fázi projektu byla věnována pozornost hodnocení projektu a shromažďování zpětné vazby, což si vyžádalo dalších 25% časové kapacity vedení firmy. V rámci zjišťování zpětné vazby byly pořádány se zaměstnanci firemní meetingy a dotazníky, které zabraly jednotlivým oddělením 5 % časové kapacity. Závěrečná fáze byla hotova za 6 dní (viz obr. 24).

Obrázek 24 – Procesy závěrečné fáze

☒ Závěrečná fáze	6 dní
Zjišťování zpětné vazby	3 dní
Závěrečná zpráva a hodnocení	3 dní

Zdroj: Vlastní zpracování (ProjectLibre)

Celkové mzdové náklady této fáze byly 19 043 Kč (viz obr. 25). Nejvyšší náklady vykázal proces zjišťování zpětné vazby, kde mělo vysoký podíl nákladů vedení firmy **$((\text{hodinová sazba vedení firmy} * 8) * \text{počet dnů}) / 4$** a dále součet nákladů všech ostatních oddělení **$((\text{hodinová sazba daného oddělení} * 8) * \text{počet dnů}) / 20$** . Závěrečnou zprávu vytvořilo vedení firmy, tudíž výpočet byl **$((\text{hodinová sazba vedení firmy} * 8) * \text{počet dnů}) / 4$** (viz tab. 10).

Tabulka 10 – Detailní rozpis nákladů závěrečné fáze podle dílčích fází

Zjišťování zpětné vazby					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	3	25%	6 678 Kč
Oddělení programátorů	1 428 Kč	11 424 Kč	3	5%	1 714 Kč
Oddělení designérů	716 Kč	5 728 Kč	3	5%	859 Kč
Oddělení marketingu	818 Kč	6 544 Kč	3	5%	982 Kč
Finanční a obchodní oddělení	792 Kč	6 336 Kč	3	5%	950 Kč
Projektové oddělení	985 Kč	7 880 Kč	3	5%	1 182 Kč
Celkové náklady					12 365 Kč
Závěrečná zpráva a hodnocení					
Oddělení	Hodinová sazba oddělení	Sazba za celý den	Počet dnů	Časová alokace/den	Celkové náklady za oddělení
Vedení firmy	1 113 Kč	8 904 Kč	3	25%	6 678 Kč
Celkové náklady					6 678 Kč

Zdroj: Vlastní zpracování

Obrázek 25 – Náklady na závěrečnou fázi



Zdroj: Vlastní zpracování (ProjectLibre)

4.8.5.1 Závěrečná zpráva a hodnocení

V průběhu 73 dnů (viz obr. 26) prošla organizace procesem zavedení systému GitLab, který byl původně naplánován na dobu 90 dnů s rozpočtem 750 000 Kč. Projekt byl dokončen s celkovými náklady 726 012 Kč, což představuje finanční úsporu oproti původnímu rozpočtu, a to i přes potenciální rizika, která byla celkem úspěšně eliminována.

Zavedení systému GitLab tak představovalo důležitý krok k dosažení lepšího projektového řízení a konkurenceschopnosti společnosti. Předpokládá se, že výhody plynoucí z této implementace budou mít pozitivní vliv na kvalitu práce a budou podporovat růst a vývoj společnosti.

Obrázek 26 – Celková časová náročnost procesu zavedení systému

☐ Projekt zavedení systému GitLab	73 dní
☐ Přípravná fáze	20 dní
Definice cílů projektu	5 dní
Analýza současného stavu	5 dní
Plánování projektu	10 dní
☐ Fáze návrhu a vývoje	5 dní
Technický návrh	5 dní
Rozpočet a finanční plánování	3 dní
☐ Implementační fáze	16 dní
Nastavení a konfigurace GitLabu	3 dní
Migrace dat a integrace systému	10 dní
Školení a podpora	10 dní
Průběžné finanční sledování	16 dní
☐ Fáze testování	27 dní
Testování a kontrola v provozu	15 dní
Identifikace a řešení případných problémů	7 dní
Dokumentace	5 dní
☐ Závěrečná fáze	6 dní
Zjišťování zpětné vazby	3 dní
Závěrečná zpráva a hodnocení	3 dní

Zdroj: Vlastní zpracování (*ProjectLibre*)

Náklady na mzdy zaměstnanců byly celkově 659 029 Kč, což tvořilo dominantní část z celkových nákladů na zavedení systému (viz obr. 27).

Obrázek 27 – Grafické znázornění mzdových nákladů za jednotlivá oddělení

Vedení firmy Náklady 244860 Kč Rozpočet 0 Kč	Oddělení programát... Náklady 158793,6 Kč Rozpočet 0 Kč	Oddělení designérů Náklady 50979,2 Kč Rozpočet 0 Kč
Oddělení marketingu Náklady 77873,6 Kč Rozpočet 0 Kč	Finanční a obchodní... Náklady 56390,4 Kč Rozpočet 0 Kč	Projektové oddělení Náklady 70132 Kč Rozpočet 0 Kč

Zdroj: Vlastní zpracování (*ProjectLibre*)

Projekt byl realizován s celkovými náklady ve výši 726 012 Kč (viz tab. 11), přičemž každá fáze byla monitorována z finančního hlediska.

Tabulka 11 – Celkové náklady na zavedení systému

Fáze	Typ nákladů	Částka
Přípravná fáze	Mzdy	89 040 Kč
	Měsíční předplatné	11 661 Kč
Fáze návrhu a vývoje	Mzdy	41 916 Kč
Implementační fáze	Mzdy	278 264 Kč
	Školení	32 000 Kč
	Měsíční předplatné	11 661 Kč
Fáze testování	Mzdy	230 766 Kč
	Měsíční předplatné	11 661 Kč
Závěrečná fáze	Mzdy	19 043 Kč
Náklady celkem		726 012 Kč

Zdroj: Vlastní zpracování

Zpětná vazba od uživatelů, získaná prostřednictvím dotazníků, odhalila silné i slabé stránky, které vyžadují zlepšení. Uživatelské rozhraní bylo oceněno pro jeho přívětivost, avšak byla identifikována potřeba zlepšení v oblasti správy oprávnění a dokumentace. Funkčnost a integrace nástrojů byly hodnoceny jako robustní, s potřebou dalšího rozvoje v integraci externích nástrojů.

Implementace GitLabu byla dokončena s ohledem na předem stanovený finanční a časový rámeček. Projekt byl zahájen s cílem zvýšit efektivitu, zlepšit spolupráci, sjednotit systémy a podpořit růst společnosti skrze inovativní nástroje pro správu a vývoj softwaru, což se nakonec podařilo.

4.9 Porovnání metodologie během procesu zavedení systému

V této kapitole bude poskytnuta analýza procesu zavedení systému GitLab v rámci firmy. Cílem je identifikovat, jaké aspekty implementace odpovídají spíše Waterfall metodologii a které agilnímu přístupu (viz tab. 12) a navrhnout zlepšení v souladu s agilními praktikami, kterými by se ráda ubírala autorem vybraná společnost.

4.9.1 Waterfall prvky

Proces zavedení systému GitLab začal typickými prvky Waterfall metodologie, kde převládá důraz na pevné plánování a detailní předdefinování požadavků. Přestože tento přístup přinesl jasné vymezení projektových cílů a milníků, narazil na výzvy v podobě malé flexibility při reakci na nečekané změny v projektovém prostředí.

4.9.1.1 Přípravná fáze

Začátek projektu zavedení systému GitLab byl charakterizován analýzou stávajícího stavu a identifikací klíčových požadavků. Tento systematický a lineární přístup, typický pro Waterfall metodologii, poskytl pevný základ pro další kroky. Výzvou však byla malá flexibilita při reagování na nečekané změny a nově vzniklé požadavky od zainteresovaných stran, což poukazuje na potřebu integrace agilnějších prvků do tohoto procesu.

4.9.1.2 Fáze návrhu a vývoje

Vytvoření plánu projektu a jeho vizualizace pomocí Ganttova diagramu poskytlo jasné milníky a časové rámce. Toto pevné plánování umožnilo efektivní alokaci zdrojů a koordinaci mezi týmy. Nicméně, pevná struktura plánu se ukázala být méně adaptabilní v reakci na dynamiku projektového prostředí, jak po finanční, tak i po technické stránce, což zdůrazňuje význam pružnějšího a iterativního plánovacího procesu.

4.9.2 Agilní prvky

4.9.2.1 Implementační fáze

Během implementace byly uplatněny agilní postupy, které se projeví ve formě průběžné integrace a nasazování. Tyto postupy umožňovaly týmu rychle reagovat na problémy a provádět nutné změny ve vývoji v reálném čase. Tato agilnost byla klíčová pro úspěšnou implementaci systému GitLab a zdůraznila potřebu pružné reakce na požadavky a technické výzvy, které se objevovaly.

4.9.2.2 Fáze testování

V této fázi bylo testování prováděno kontinuálně a v souladu s agilními principy, které vyžadují průběžné ověřování produktu a jeho funkcionality. Díky tomu bylo možné identifikovat a opravit chyby dříve a sledovat tak jednotlivé změny během migrace a práce uživatelů se systémem.

4.9.2.3 Závěrečná fáze

Projektová závěrečná fáze představovala kombinaci Waterfall a Agile praktik. Přestože bylo hodnocení provedeno po dokončení projektu, což je charakteristické pro Waterfall metodologii a naznačuje lineární přístup s hodnocením na konci, byly během projektu zavedeny určité agilní praktiky, které se projevíly během retrospektivy.

4.9.3 Doporučení pro zlepšení agilní praxe

Pro zvýšení efektivity a flexibility v budoucích projektech bylo doporučeno přijmout pružnější plánovací metody, jako jsou agilní Srinty a Scrum Meetingy. Zdůrazněn byl také význam zapojení koncových uživatelů do celého procesu a potřebu průběžného hodnocení a adaptace na základě získané zpětné vazby.

Během procesu zavedení systému Gitlab v rámci podniku se ukázalo, že kombinace Waterfall a agilních metod může být účinná, ale pro dosažení nejlepších výsledků je klíčová vyšší míra flexibility a adaptace. To potvrzuje, že v digitálním prostředí by měly být agilní přístupy preferovány před tradičními, zejména ve fázích, kde je vyžadována rychlá odezva na proměnlivé podmínky a požadavky.

Tabulka 12 – Vyhodnocení fází projektu podle metodologie

Fáze projektu	Waterfall metoda	Agilní metoda	Použitá metoda
Přípravná fáze	Detailní plánování a specifikace požadavků na začátku projektu	Flexibilní přístup s možností průběžného přizpůsobení	Waterfall
Fáze návrhu a vývoje	Sekvenční návrh a vývoj s důrazem na kompletnost před přechodem k testování	Iterativní návrh a vývoj s průběžnými iteracemi a testováním	Waterfall
Implementační fáze	Kompletní implementace před testováním	Průběžná implementace a testování v každé iteraci	Agile
Fáze testování	Testování po dokončení celého vývoje	Kontinuální testování během celého vývojového cyklu	Agile
Závěrečná fáze	Hodnocení po dokončení projektu	Průběžné hodnocení a retrospektivy po každé iteraci	Waterfall/Agile

Zdroj: Vlastní zpracování

4.10 Rizika při zavádění systému

Tato kapitola se zabývá identifikací, hodnocením a řízením potenciálních rizik, která byla spojena s implementací GitLabu. Registr rizik je klíčovým nástrojem pro zajištění úspěšné realizace projektu a obsahuje seznam možných hrozeb včetně jejich hodnocení, dopadu a strategií prevence.

Každé riziko bylo hodnoceno podle pravděpodobnosti jeho výskytu a potenciálního dopadu na projekt. Pro každé identifikované riziko byly navrženy konkrétní strategie prevence a korekce (viz tab. 13). Rizika byla během celého projektu průběžně monitorována, ale ne všechna opatření byla dodržena.

Tabulka 13 – Grafické znázornění registru rizik

ID	Riziko	Výskyt	Dopad	Návrh prevence	Návrh korekce
1	Technické problémy při migraci dat	Vysoká	Vysoký	Předem testovat migraci na vzorových datech	Připravit plán obnovy pro případ chyb
2	Odpor zaměstnanců vůči novému systému	Střední	Střední	Organizovat školení a workshop	Otevřená komunikace a zmírnění odporu
3	Nedostatečná uživatelská podpora	Nízká	Střední	Zřídit interní helpdesk a FAQ	Vytvoření podpůrného týmu
4	Podcenění časové náročnosti	Střední	Vysoký	Přidělit více zdrojů na projekt	Revize a optimalizace projektového plánu
5	Překročení rozpočtu	Nízká	Střední	Pevně stanovit rozpočet a pravidelně sledovat výdaje	Revize a úspora nákladů v rozpočtu
6	Bezpečnostní rizika související s cloudem	Střední	Vysoký	Zavést silná bezpečnostní opatření a pravidelné audity	Vypracování plánu reakce na události
7	Nedostatečná kompatibilita s existujícími systémy	Vysoká	Střední	Provést důkladnou analýzu kompatibility před migrací	Adaptace systémů pro kompatibilitu
8	Výpadky služby během migrace	Střední	Vysoký	Plánovat migraci mimo pracovní dobu	Procedury pro rychlou obnovu služeb
9	Chyby v nově implementovaných procesech	Vysoká	Střední	Provádět průběžné testování	Průběžné testování a opravy
10	Závislost na externím dodavateli	Nízká	Vysoký	Vybrat spolehlivého dodavatele a mít záložní plán	Instalace systému na vlastním serveru

Zdroj: Vlastní zpracování

Seznam možných rizik:

1. **Technické problémy při migraci dat:**

- **Prevence** – testování migrace na vzorových datech musí být provedeno včas, aby se odhalily a vyřešily jakékoli komplikace před skutečnou migrací dat.
- **Korekce** – v případě výskytu chyb během migrace je třeba mít připravený náhradní plán a postupy pro obnovu dat, aby se minimalizoval dopad na provoz.

2. **Odpor zaměstnanců vůči novému systému:**

- **Prevence** – uspořádání školení a workshopu, které by pomohlo zaměstnancům pochopit výhody systému a zapojit se do procesu změny. Dokumentace by měla

být přizpůsobena potřebám různých uživatelů a měla by zahrnovat praktické aspekty práce s novým systémem.

- **Korekce** – pokud dojde k odporu, je důležité zahájit dialog a zpětnou vazbu, aby se pochopily příčiny odporu a aby bylo možné pracovat na jejich řešení a zlepšení situace.

3. **Nedostatečná uživatelská podpora:**

- **Prevence** – dokumentace a seznam často kladených otázek by měly být zřízeny a pravidelně aktualizovány, aby poskytovaly uživatelům neustálou podporu. Tato opatření zlepšuje rychlost a efektivitu řešení uživatelských dotazů a problémů.
- **Korekce** – při nedostatečné podpoře uživatelů je třeba zvýšit dostupnost a odbornost podpůrného týmu a zlepšit komunikační kanály pro efektivní řešení problémů.

4. **Podcenění časové náročnosti:**

- **Prevence** – přidělení dostatečných zdrojů na projekt by mělo zajistit, že všechny úkoly by měly být dokončeny včas a projekt by měl být v souladu s plánovanými časovými rámci. Tento krok zahrnuje flexibilní řízení týmů pro přizpůsobení se jakýmkoliv neočekávaným změnám.
- **Korekce** – když se projekt protahuje nad rámec plánovaného časového harmonogramu, je potřeba upravit projektový plán a přerozdělit zdroje pro zajištění dodržení termínů.

5. **Překročení rozpočtu:**

- **Prevence** – vytvořením detailního finančního plánu s rezervami pro nepředvídané výdaje jsou nastaveny postupy pro kontrolu a schvalování výdajů. Pravidelné sledování výdajů umožňuje zachovat finanční zdravost celého projektu.
- **Korekce** – pokud dochází k překročení rozpočtu, musí se provést revize výdajů a identifikovat oblasti, kde je možné snížit náklady nebo optimalizovat využití zdrojů.

6. **Bezpečnostní rizika související s cloudovým řešením:**

- **Prevence** – pro zajištění bezpečnosti dat by měly být zavedeny silné bezpečnostní procedury, jako kontrola hesel, uživatelských oprávnění či nastavení repozitářů.
- **Korekce** – v případě vzniku bezpečnostních incidentů je nutné mít přístup k podpoře a dokumentaci v rámci GitLabu.

7. **Nedostatečná kompatibilita s existujícími systémy:**

- **Prevence** – technický tým provádí důkladné analýzy kompatibility, aby zajistil, že nový systém bude fungovat bez problémů s již existujícími nástroji a aplikacemi. Případné problémy jsou řešeny před zahájením migrace, což zajišťuje hladkou integraci.
- **Korekce** – pokud se projeví problémy s kompatibilitou, je potřeba zvážit alternativní řešení nebo upravit nový systém tak, aby byl kompatibilní se stávajícími systémy.

8. **Výpadky služby během migrace:**

- **Prevence** – migrace by měla být naplánována na dobu mimo standardní pracovní hodiny, což minimalizuje přerušení provozu a nenarušuje podnikové operace.
- **Korekce** – je třeba mít připravené procesy pro rychlou obnovu služeb a komunikaci s klienty a uživateli systému GitLab, aby byli informováni o stavu a předpokládané době pro obnovení služeb.

9. **Chyby v nově implementovaných procesech:**

- **Prevence** – nastavení testovacích scénářů, které umožňuje rychlé odhalení a opravy chyb v nových procesech. Toto je klíčové pro udržení kvality a funkčnosti nově zavedených procesů.
- **Korekce** – při identifikaci chyb je třeba provádět okamžité opravy a aktualizace procesů.

10. **Závislost na externím dodavateli:**

- **Prevence** – pro minimalizaci rizik spojených s externími dodavateli by měl být vybrán přímo výrobce software.
- **Korekce** – v případě problémů s cloudovým řešením je možné GitLab hostovat na vlastním serveru.

5 Závěr

Obsah diplomové práce se věnuje zkoumání procesu zavádění softwaru GitLab v rámci organizace. Zpočátku se autor práce soustředil na analýzu stávajících systémů a procesů, což bylo klíčové pro identifikaci a pochopení specifických požadavků společnosti. Důkladná příprava a pochopení potřeb umožnila plynulý přechod a efektivní využití GitLabu pro správu projektů a vývoj softwaru.

Teoretická část práce představuje základní kámen celého projektu. Zabývala se analýzou informačních systémů, procesů a metodologií používaných v projektovém řízení. Cílem bylo zjistit, jak nejlépe integrovat systém tak, aby odpovídal specifickým potřebám a požadavkům společnosti v rámci projektového řízení s důrazem na agilní přístupy, což umožnilo vytvořit pevný teoretický rámec pro praktickou implementaci.

V praktické části práce byla tato teorie aplikována na zavedení systému GitLab. Zabývala se technickými aspekty implementace, včetně výběru a konfigurace softwaru, integrace s existujícími systémy, migrací dat a testováním. Součástí implementace bylo i vytvoření finančního plánu, časového harmonogramu a pravidelné sledování nákladů. Důležitým aspektem bylo také zaškolení zaměstnanců, aby efektivně využívali nový systém pro správu projektů a vývoje softwaru. Byly zdůrazněny výhody GitLabu, jako je lepší sledování změn, ucelenost a lepší možnosti kolaborace mezi týmy.

Z pohledu metodologie došlo při určitých procesech zavádění systému k odchýlení se od agilních metodik a použilo se tradičního vodopádového přístupu. Toto řešení bylo kritizováno pro nižší efektivitu a pro horší reakci na aktuální události, které se staly během procesu zavedení. Vedení firmy tak nebylo schopné zareagovat včas, čemuž by prospěly například pravidelné schůzky po každé fázi projektu či vykonávání fází souběžně.

Dalším dílčím cílem byla analýza rizik, která přispěla k lepšímu pochopení potenciálních překážek a návrhům na jejich prevenci a řízení. Práce přinesla doporučení a strategie pro zlepšení budoucích implementací a může pomoci dalšímu rozvoji informačních systémů v organizaci.

V závěru této diplomové práce je ovšem nutné poukázat na to, že kvůli omezenému rozsahu práce nebylo možné provést podrobné porovnání metodologií projektového řízení při implementaci systému GitLab do takové hloubky, jak by bylo ideální. Toto omezení

neodráží nedostatek důležitosti tématu, ale spíše nutnost respektovat obsahové a časové limity akademické práce. I když byly splněny hlavní cíle a implementace nového systému GitLab byla úspěšně zdokumentována a analyzována, rozsáhlejší srovnání metodologií a analýza rizik by mohla představovat významné rozšíření této práce. Tento aspekt se jeví jako vhodný směr pro budoucí výzkum, který by mohl nabídnout hlubší pochopení optimalizace projektových metodik ve spojení s využíváním moderních nástrojů pro správu projektů a lepší uchopení rizik při implementaci.

Celkové zhodnocení ukázalo, že cíle práce byly úspěšně splněny. Nově zavedený systém má pozitivní dopad na efektivitu projektového řízení a přináší společnosti konkurenční výhodu. Proces implementace byl zdokumentován, což představuje cenný zdroj informací pro další projekty a rozhodnutí v rámci firmy. Tato práce tedy představuje ucelený pohled na zavádění informačního s důrazem na oblasti projektového řízení a vývoje softwaru.

6 Seznam použitých zdrojů

Literatura:

AXELOS. 2017. Managing Successful Projects with PRINCE2. 6th ed. London: TSO (The Stationery Office). ISBN 978-0-1133-1533-8.

BASL, J., BLAŽÍČEK, R. 2012. Podnikové informační systémy: podnik v informační společnosti. 3. aktualizované a doplněné vydání. Praha: Grada Publishing, a.s. ISBN 978-80-247-4307-3.

BRUCKNER, T. a kol. 2012. Tvorba informačních systémů. 1. vydání. Praha: Grada Publishing. ISBN 978-80-247-4153-6.

CRESWELL, J. W. 2017. Research design: Qualitative, quantitative, and mixed methods approaches. 4. vydání. Thousand Oaks: Sage Publications. ISBN 978-1-4522-2610-1.

DOLEŽAL, J. a kolektiv. 2016. Projektový management: Komplexně, prakticky a podle světových standardů. Praha: Grada Publishing, a.s. ISBN 978-80-247-5620-2.

DOLEŽAL, J., MÁCHAL, P., LACKO, B. a kolektiv. 2012. Projektový management podle IPMA. 2., aktualizované a doplněné vydání. Praha: Grada Publishing, a.s. ISBN 978-80-247-4275-5.

FIALA, P. 2004. Projektové řízení – modely, metody, analýzy. 1. vydání. Praha: Professional Publishing. ISBN 80-86419-24-X.

GÁLA, L., POUR, J., ŠEDIVÁ, Z. 2015. Podniková informatika: Počítačové aplikace v podnikové a mezipodnikové praxi. 3. aktualizované vydání. Praha: Grada Publishing, a.s. ISBN 978-80-247-5457-4.

HASS, K. B. 2007. The Business Analyst as Strategist: Translating Business Strategies into Valuable Solutions. Berrett-Koehler Publishers. ISBN 978-1-56726-343-5.

HRAZDILOVÁ BOČKOVÁ, K. 2016. Projektové řízení. Martin Kolářek: E-knihy jedou. Učebnice. ISBN 978-80-7512-431-9.

KERZNER, H. 2017. Project management: a systems approach to planning, scheduling, and controlling. 12th ed. Hoboken: John Wiley & Sons. ISBN 978-1-1191-6535-4.

MÁCHAL, P., KOPEČKOVÁ, M., PRESOVÁ, R. 2015. Světové standardy projektového řízení pro malé a střední firmy. 1. vydání. Praha: Grada Publishing. ISBN 978-80-247-5321-8.

NOVOTNÝ, O., POUR, J., SLÁNSKÝ, D. 2005. Business Intelligence. Praha: Grada Publishing, a.s. ISBN 978-80-247-6685-0.

OFFICE OF GOVERNMENT COMMERCE (OGC). 2009. Managing successful projects with PRINCE2. London: The Stationery Office. ISBN 978-0-1133-1059-3.

PROJECT MANAGEMENT INSTITUTE (PMI). 2017. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 6th ed. Pennsylvania: Project Management Institute. ISBN 978-1-62825-184-5.

SCHWALBE, K. 2011. Řízení projektů v IT. Brno: Computer Press. ISBN 978-80-251-2882-4.

SODOMKA, P., KLČOVÁ, H. 2010. Informační systémy v podnikové praxi. Brno: Computer Press. ISBN 978-80-251-2878-7.

SVOZILOVÁ, A. 2011. Projektový management. 2. vydání. Praha: Grada Publishing, a.s. ISBN 978-80-247-3611-2.

ŠILHAVÝ, R. a kol. 2013. Vybrané aspekty návrhu webových informačních systémů. 1. vydání. Vsetín: Scientific Press. ISBN 978-80-904741-3-0.

ŠOCHOVÁ, Z., KUNCE, E. 2014. Agilní metody řízení projektů. 1. vydání. Brno: Computer Press. ISBN 978-80-251-4194-6.

TVRDÍKOVÁ, M. 2008. Aplikace moderních informačních technologií v řízení firmy: Nástroje ke zvyšování kvality informačních systémů. Praha: Grada Publishing, a.s. ISBN 978-80-247-2728-8.

VLASÁK, R., BULÍČKOVÁ, S. 2003. Základy projektování informačních systémů. Praha: Univerzita Karlova v Praze. Nakladatelství Karolinum. ISBN 80-246-0727-1.

Internetové zdroje:

AGILE MANIFESTO. 2001. Manifest Agilního vývoje software. [online] 1. 1. 2001. [Citace: 10. 1. 2024.] Dostupné z: <http://agilemanifesto.org/iso/cs/manifesto.html>.

PM-PARTNERS GROUP. 2024. What is Scrum? An overview of Scrum and The Agile Journey. [online] 25. 3. 2024. [Citace: 10. 1. 2024.] Dostupné z: <https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview>.

SCHWABER, K., SUTHERLAND, J. 2020. The Scrum Guide. [online] 2020. [Citace: 29. 1. 2024.] Dostupné z: <https://www.scrumguides.org/scrum-guide.html>.

ŠIMŮNEK, D. 2019. Jaký je rozdíl mezi Waterfall a Agile přístupem. [online] 27. 8. 2019. [Citace: 21. 3. 2024.] Dostupné z: <https://www.davidsimunek.com/post/jaky-je-rozdil-mezi-waterfall-a-agile>.

SYSTEMONLINE. 2014. PRINCE2: principy, témata, procesy (2. díl). [online] 2014. [Citace: 20. 3. 2024.] Dostupné z: <https://www.systemonline.cz/rizeni-projektu/prince2-principy-temata-procesy-2.-dil-1.htm>.

Seznam obrázků, tabulek, grafů a zkratk

6.1 Seznam obrázků

Obrázek 1 – Trojimperativ projektu	18
Obrázek 2 – Životní cyklus projektu	20
Obrázek 3 – Waterfall metodologie.....	23
Obrázek 4 – Agilní metodologie.....	25
Obrázek 5 – Struktura PRINCE2.....	28
Obrázek 6 – Rámcový procesní model PRINCE2.....	30
Obrázek 7 – Řízení projektů Scrum.....	34
Obrázek 8 – Organizační úrovně podniku	41
Obrázek 9 – Hierarchie společnosti	47
Obrázek 10 – Souhrn všech kroků během zavádění systému	50
Obrázek 11 – Nápady na zlepšení projektu	55
Obrázek 12 – Znázornění procesů při zavedení systému formou Ganttova diagramu.....	59
Obrázek 13 – Technický návrh procesů	61
Obrázek 14 – Ukázka nahlášené chyby během testování v systému GitLab.....	70
Obrázek 15 – Zpětná vazba z dotazníků	72
Obrázek 16 - Procesy v přípravné fázi.....	73
Obrázek 17 – Náklady na přípravnou fázi	74
Obrázek 18 – Procesy ve fázi návrhu a vývoje.....	74
Obrázek 19 – Náklady na fázi návrhu a vývoje.....	75
Obrázek 20 – Procesy v implementační fázi	76
Obrázek 21 – Náklady na implementační fázi.....	77
Obrázek 22 – Procesy ve fázi testování	78
Obrázek 23 – Náklady na fázi testování	79
Obrázek 24 – Procesy závěrečné fáze.....	80
Obrázek 25 – Náklady na závěrečnou fázi	81
Obrázek 26 – Celková časová náročnost procesu zavedení systému	82
Obrázek 27 – Grafické znázornění mzdových nákladů za jednotlivá oddělení.....	82

6.2 Seznam tabulek

Tabulka 1 – Porovnání standardů	37
Tabulka 2 – Grafické znázornění odpovědnosti za dané procesy	53
Tabulka 3 – Možnosti předplatného ve službě GitLab	62
Tabulka 4 – Mzdové náklady.....	64
Tabulka 5 – Výsledky testování.....	70
Tabulka 6 – Detailní rozpis nákladů přípravné fáze podle dílčích fází	74
Tabulka 7 – Detailní rozpis nákladů fáze návrhu a vývoje podle dílčích fází	75
Tabulka 8 – Detailní rozpis nákladů implementační fáze podle dílčích fází	77
Tabulka 9 – Detailní rozpis nákladů fáze testování podle dílčích fází	79
Tabulka 10 – Detailní rozpis nákladů závěrečné fáze podle dílčích fází.....	80
Tabulka 11 – Celkové náklady na zavedení systému	83
Tabulka 12 – Vyhodnocení fází projektu podle metodologie.....	86
Tabulka 13 – Grafické znázornění registru rizik	87

7 Přílohy

Příloha 1 – Ukázka procesů a Gantt diagramu v programu ProjectLibre.....	98
--	----

	Jméno	Trvání
1	Projekt zavedení systému GitLab	73 dní
2	Přípravná fáze	20 dní
3	Definice cílů projektu	5 dní V
4	Analýza současného stavu	5 dní V
5	Plánování projektu	10 dní V
6	Fáze návrhu a vývoje	5 dní
7	Technický návrh	5 dní C
8	Rozpočet a finanční plánování	3 dní V
9	Implementační fáze	16 dní
10	Nastavení a konfigurace GitLabu	3 dní C
11	Migrace dat a integrace systému	10 dní V
12	Školení a podpora	10 dní V
13	Průběžné finanční sledování	16 dní V
14	Fáze testování	27 dní
15	Testování a kontrola v provozu	15 dní V
16	Identifikace a řešení případných problémů	7 dní V
17	Dokumentace	5 dní C
18	Závěrečná fáze	6 dní
19	Zjišťování zpětné vazby	3 dní V
20	Závěrečná zpráva a hodnocení	3 dní V

