

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DIPLOMOVÁ PRÁCE

2024

Bc. Ondřej Matějka

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Cloudová webová aplikace pro řízení IT projektů
Bc. Ondřej Matějka

Diplomová práce
2024

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Ondřej Matějka**
Osobní číslo: **I22167**
Studijní program: **N0613A140007 Informační technologie**
Téma práce: **Cloudová webová aplikace pro řízení IT projektů**
Zadávací katedra: **Katedra softwarových technologií**

Zásady pro vypracování

Diplomová práce se zabývá tvorbou a implementací cloudové webové aplikace pro řízení IT projektů v různých úrovních dodavatelských řetězců a s odlišnými rolemi. V teoretické části práce budou popsány základní principy a specifika spojená s řízením IT projektů, součástí práce bude i popis obdobných řešení. Práce se bude také zabývat vybranými aspekty frameworku ITIL. Výstupem práce je cloudová webová aplikace, která bude umožňovat plnou správu IT projektů a respektovat jejich specifika.

Rozsah pracovní zprávy: **cca 60 stran**
Rozsah grafických prací: **-**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

BREWER, Jeffrey L. Methods of It Project Management, Fourth Edition. 4th edition. West Lafayette, Indiana: Purdue University Press, 2022. ISBN 9781612497907.
GUCKENHEIMER, Sam a Juan J. PEREZ. Efektivní softwarové projekty. Brno: Zoner Press, 2007. Encyklopedie Zoner Press. ISBN 978-80-86815-62-6.
HUGHES, Bob. Project Management for IT-Related Projects: 3rd edition. 3rd edition. Wiltshire: BCS Learning and Development, 2019. ISBN 9781780174860.

Vedoucí diplomové práce: **Ing. Monika Borkovcová, Ph.D.**
Katedra informačních technologií

Datum zadání diplomové práce: **8. listopadu 2023**
Termín odevzdání diplomové práce: **17. května 2024**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

prof. Ing. Antonín Kavička, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 30. listopadu 2023

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17. 5. 2022

Bc. Ondřej Matějka

PODĚKOVÁNÍ

Prvotní, ale velké poděkování patří vedoucí diplomové práce paní Ing. Monice Borkovcové, Ph.D. pro její velký zápal a odhodlání poskytnout kvalitní vzdělávací prostředí nejen pro mě, ale pro všechny studenty Fakulty elektrotechniky a informatiky Univerzity Pardubice, konkrétně v oboru Informačních technologií. Ať už šlo o pravidelné konzultace a připomínky k práci samotné, nebo o předměty v semestrech, paní M. Borkovcová dbala vždy na úroveň vedení i obsahu hodin, a to i přes dobu postiženou koronavirem SARS-CoV-2. Děkuji, a do budoucna přeji jen samý příznivý vítr do plachet.

Další poděkování patří mým rodičům, kteří mě bezprostředně podporovali materiálně i mentálně během obtížné cesty dospíváním k samostatnosti a moudru. Takovou oporu nelze považovat za samozřejmost a bohužel věřím, že mnozí takovým štěstím nedisponovali a nedisponují v této stránce.

V poslední řadě to jsou lidé, které jsem měl čest poznat během studií. Jsou to kamarádi, bez jichž by tento čas zastal mnohem těžší formy. Také se jedná o učitele a lidé angažující se v okolní akademické sféře, kteří nejen mne provedli a obohatili během společných chvil.

ANOTACE

Diplomová práce se zabývá tvorbou a implementací cloudové webové aplikace pro řízení IT projektů v různých úrovních dodavatelských řetězců a s odlišnými rolemi. V teoretické části práce budou popsány základní principy a specifika spojená s řízením IT projektů, součástí práce bude i popis obdobných řešení. Práce se bude také zabývat vybranými aspekty frameworku ITIL. Výstupem práce je cloudová webová aplikace, která bude umožňovat plnou správu IT projektů a respektovat jejich specifika.

KLÍČOVÁ SLOVA

řízení IT projektů, cyklus softwarového projektu, ITIL, DevOps, Lean, ASP.NET Core

TITLE

Cloud-based web application for IT project management

ANNOTATION

The thesis deals with the creation and implementation of a cloud-based web application for IT project management at different levels of supply chains and with different roles. The theoretical part of the thesis will describe the basic principles and specifics associated with IT project management, and the thesis will also include a description of similar solutions. The thesis will also deal with selected aspects of the ITIL framework. The output of the thesis is a cloud-based web application that will enable full management of IT projects and respect their specifics.

KEYWORDS

IT project management, software project cycle, ITIL, DevOps, Lean, ASP.NET Core

OBSAH

SEZNAM ILUSTRACÍ A TABULEK.....	10
SEZNAM ZKRATEK A ZNAČEK	11
TERMINOLOGIE	12
ÚVOD.....	13
1 Životní cyklus vývoje softwarového projektu a jeho řízení	14
1.1 Etapy	14
1.1.1 Analýza a plánování.....	14
1.1.2 Definování požadavků	14
1.1.3 Návrh	15
1.1.4 Vývoj	15
1.1.5 Testování.....	15
1.1.6 Nasazení.....	15
1.1.7 Údržba.....	16
1.2 Modely	16
1.2.1 Vodopádový model.....	16
1.2.2 Prototypování.....	17
1.2.3 Agilní model	17
1.2.4 Iterativní model.....	18
1.2.5 Spirálový model.....	18
1.2.6 Rational Unified Process	18
1.2.7 Big Bang model	19
1.3 Projektový trojúhelník.....	19
1.3.1 Rozsah.....	20
1.3.2 Zdroje.....	20
1.3.3 Čas	21
2 ITIL 4.....	22
2.1 Seznámení	22
2.2 Historie.....	22
2.3 Pojmy a definice.....	23
2.4 Řízení služeb přes čtyři dimenze.....	29
2.4.1 Organizace a lidé	30

2.4.2	Informace a technologie.....	30
2.4.3	Partneři a dodavatelé.....	32
2.4.4	Hodnotové toky a procesy	32
2.5	Systém hodnot služeb.....	33
2.5.1	Hlavní zásady.....	36
2.5.2	Řízení	44
2.5.3	Hodnotový řetězec služeb	46
2.5.4	Praktiky	51
2.5.5	Neustálé zlepšování	58
3	Analýza konkurenčních řešení webové aplikace	60
3.1	YouTrack.....	60
3.2	Trello	61
3.3	Any.do.....	62
3.4	Nextcloud	63
4	Doporučení pro aplikaci.....	65
5	Aplikace pro projektové řízení	67
5.1	Funkční a nefunkční požadavky.....	67
5.2	Případ užití	71
5.3	Architektura systému	72
5.4	Použité technologie	73
5.4.1	Azure.....	74
5.4.2	ASP.NET	75
5.4.3	Blazor WebAssembly	76
5.4.4	MudBlazor	77
5.4.5	Web API	77
5.4.6	EF Core	78
5.5	Implementace klíčových aplikačních složek.....	78
5.5.1	Logování a notifikace	79
5.5.2	Soft delete	79
5.5.3	Autentizace a autorizace	80
5.6	Ganttův diagram vývoje aplikace.....	81

5.7	Základní průvodce užití aplikace	82
5.8	Budoucí rozšíření a zlepšení aplikace	89
	ZÁVĚR	90
	POUŽITÁ LITERATURA	91
	SEZNAM PŘÍLOH.....	94

SEZNAM ILUSTRACÍ A TABULEK

Tabulka 1: Příklad zainteresovaných stran a hodnot [8 str. 11].....	25
Tabulka 2: Praktiky řízení ITIL [8 str. 76]	51
Obrázek 1: Scrum model [1 str. 45].....	17
Obrázek 2: RUP Model [1 str. 46].....	19
Obrázek 3: Projektový trojimperativ (zdroj vlastní).....	20
Obrázek 4: Čtyři dimenze správy služeb [9]	30
Obrázek 5: Systém hodnot služeb [11]	35
Obrázek 6: Vizualizace Kanban [12].....	41
Obrázek 7: Pět principů metodiky Lean [14].....	43
Obrázek 8: Model neustálého zlepšování [8 str. 66]	59
Obrázek 9: Use Case diagram aplikace (vytvořeno pomocí Visual Paradigm Online, zdroj vlastní)	72
Obrázek 10: Architektura aplikace v jazyce ArchiMate (vytvořeno pomocí Visual Paradigm Online, zdroj vlastní)	73
Obrázek 11: Životní cyklus vývoje webové aplikace v Ganttově diagramu (zdroj vlastní).....	82
Obrázek 12: Ukázka aplikace – vstup do projektu (zdroj vlastní).....	83
Obrázek 13: Ukázka aplikace – hlavní dashboard (zdroj vlastní)	83
Obrázek 14: Ukázka aplikace – správa rozpočtu (zdroj vlastní)	84
Obrázek 15: Ukázka aplikace – specifikace úkolu (zdroj vlastní).....	85
Obrázek 16: Ukázka aplikace – projekce fází projektu (zdroj vlastní).....	85
Obrázek 17: Ukázka aplikace – pozvání uživatele do projektu (zdroj vlastní)	86
Obrázek 18: Ukázka aplikace – skupinové uspořádání (zdroj vlastní).....	87
Obrázek 19: Ukázka aplikace – fáze (nejdůležitější průsečík částí projektu, zdroj vlastní).....	87
Obrázek 20: Ukázka aplikace – správa datových souborů (zdroj vlastní).....	88
Obrázek 21: Ukázka aplikace – vizualizace projektových prostředků (zdroj vlastní)	88
Zdrojový kód 1: Ukázka struktury komponenty Razor v Blazor [26].....	77
Zdrojový kód 2: Ukázka anotace pro řízení přístupu Web API (zdroj vlastní).....	78
Zdrojový kód 3: Přetížení metody SaveChangesAsync pro implementaci Soft Delete (zdroj vlastní)	80
Zdrojový kód 4: Ukázka anotace pro potřebné oprávnění (zdroj vlastní)	80
Zdrojový kód 5: Ověření práv uživatele v middleware (zdroj vlastní).....	81

SEZNAM ZKRATEK A ZNAČEK

SDLC	Software Development Life Cycle
SRLC	Software Release Life Cycle
ITSM	IT Service Management
CSF	Critical Success Factors
SVS	Service Value System
SVC	Service Value Chain
SEO	Search Engine Optimization
DTO	Data Transfer Object
API	Application Programming Interface
DI	Dependency Injection
JS	JavaScript
IoT	Internet of Things

TERMINOLOGIE

Stakeholder	Jedná se o libovolný subjekt, jež je vázán určitým způsobem na organizaci svým zájmem.
Callback	Uložená obslužná rutina, kterou lze později spustit.

ÚVOD

V moderním světě informačních technologií je správa projektů klíčovým faktorem pro úspěch firem a organizací. Používání moderních informačních nástrojů se stává nezbytným pro usnadnění a zefektivnění procesů plánování, organizace a kontroly projektů. Tato diplomová práce se zaměřuje na návrh a vývoj webové aplikace s využitím moderních technologií a metodik.

Hlavním cílem diplomové práce je navrhnout a vyvinout cloudovou webovou aplikaci pro efektivní řízení IT projektů. Aplikace by měla splňovat požadavky na komplexní funkcionalitu, uživatelsky přívětivé rozhraní, využití moderních technologií a dostupnost v cloudovém prostředí. Očekávaným výsledkem práce je funkční a uživatelsky přívětivá aplikace pro efektivní řízení IT projektů, která by mohla přinést řadu benefitů v praxi. Mezi hlavní přínosy patří zefektivnění procesů plánování, organizace a kontroly projektů, zvýšení transparentnosti a informovanosti o stavu projektů, zlepšení spolupráce mezi členy projektových týmů a zúčastněných stran a usnadnění sdílení informací a dokumentů.

Práce uvede v teoretické rovině projektové dimenze času, rozsahu a prostředků. Následuje rozebrání metodik přístupu vývoje softwarových projektů zasazené do životního cyklu, který definuje fáze, jimiž projekty postupně prochází. Hlavní částí je představení moderního rámce ITIL 4 pro řízení a realizaci organizačních plánů za pomoci IT služeb. Práce v kostce rozebere průřez rámcem a zaměří se na hlavní principy a vysvětlení systému hodnot společně s možnými příklady a vybranými inspiracemi pro implementaci do výstupní aplikace.

Diplomová práce se také soustředí na analýzu dostupných a oblíbených řešení, nástrojů pro podporu projektového řízení, a rozebírá jejich důležité aspekty. Další část práce bude představovat technologie pro realizaci výsledné webové aplikace. Autor bude implementovat ucelené řešení poskytované firmou Microsoft pro vývoj softwaru za pomoci balíčku ASP.NET, převážně aplikační programové rozhraní, knihovnu komponent uživatelského rozhraní a rámec Blazor realizující prostředí aplikace. Analýza výstupu bude specifikovat požadavky a další vlastnosti kladené na systém.

Práce představí jeden z komplexních scénářů v rámci fiktivního začínajícího projektu. Vyvinutá aplikace by mohla přispět ke zlepšení efektivity a úspěšnosti IT projektů a stát se zajímavým nástrojem pro firmy a organizace v různých odvětvích.

1 Životní cyklus vývoje softwarového projektu a jeho řízení

SDLC, z anglického výrazu „Software Development Life Cycle“, je označení pro ucelený proces projektu, který ukazuje, jakými fázemi prochází, či měl procházet, dogmaticky vývoj softwaru. Nejedná se o povinnost držet se přesně životního cyklu projektu v oblasti informačních technologií, ale o nástroj poskytující strukturovaný rámec pomáhající procesním týmům dosahovat svých vytyčených cílů.

K výhodám, proč vůbec držet pevně oprátky vývoje projektu s ohledem na tento životní cyklus, patří:

- chápat, co zákazník opravdu chce a potřebuje,
- plánovat projekt společně s náklady na vývoj,
- řídit rizika spojené s projektem a adekvátně na ně reagovat,
- viditelnost a porozumění pro všechny zúčastněné strany procesu a projektu.

1.1 Etapy

Nyní už lze přistoupit k samotným etapám SDLC. Existuje vícero definicí bodů v životním cyklu, nicméně většina z nich lze zobecnit do následujících etap jdoucích po sobě.

1.1.1 Analýza a plánování

Zásadní etapa je Analýza a plánování, kdy se schraňují požadavky všech – jak zákazníků, tak zainteresovaných stran. Analyzují se cíle, finance, prostředky, očekávání, časové milníky, podmínky anebo třeba samotný přínos. Výsledkem může být plán projektu a posouzení proveditelnosti.

Jelikož se jedná o první etapu a zároveň klíčový základní kámen pro další vývoj projektu, podcenění této fáze může mít významné následky na výsledný produkt a celý proces.

1.1.2 Definování požadavků

Ve fázi definování požadavků v rámci životního cyklu vývoje softwaru se systematicky transformují abstraktní požadavky z předchozí fáze do konkrétních specifikací. To zahrnuje přepis a třídění požadavků, identifikaci funkcí a omezení, definici případů užití a scénářů pro popis interakcí mezi uživateli a systémem a vytvoření matice sledovanosti požadavků. Tato matice umožňuje sledovat naplnění požadavků a jejich propojení s funkcemi systému. Dále se aktualizuje plán projektu z hlediska časové a nákladové náročnosti na základě detailní

specifikace požadavků, což umožňuje lepší odhad potřebných zdrojů a času pro dokončení projektu.

1.1.3 Návrh

Návrh představuje etapu, která znova analyzuje předchozí výstupy a transformuje je do dokumentů a modelů, které lze zpětně prezentovat zúčastněným stranám. Ukazuje se, jak bude systém implementován, jak bude dělen, jak se realizují požadavky na systém, jak systém bude reagovat ve specifických situacích i za pomoci diagramů. Specifikuje se prostředí (hardwarové, softwarové, vývojové), vytváří se podrobná dokumentace a pseudokódy. Důležité je pokrýt také okrajové stavy, které mohou nastat. Jelikož systémy vyžadují údržbu, je dobrá praktika myslet i na to, jaká bude budoucí podpora pro opravu chyb, nebo jen reagování na další změny ze strany zákazníka. Na konci se opět aktualizuje celý plán projektu.

1.1.4 Vývoj

Systém je již komplexně popsán v analytických výstupech předchozích etap, a tak jsou na řadě vývojáři pro přetavení návrhové dokumentace do fyzické podoby funkčního prototypu systému za pomoci tvorby kódů a algoritmů. I přes velkou snahu předchozích analýz, dost často nastává stav nepokrytí případů a chybějících specifikací pro implementaci. Dle metodik vývoje lze tak reagovat na tuto skutečnost. V etapě vznikají testovací případy pro další fázi. Plán projektu se průběžně aktualizuje.

1.1.5 Testování

Software se v této fázi podrobuje testům. Cílem je zanalyzovat, jestli systém pokrývá a splňuje veškeré požadavky, jestli je systém stabilní a bezpečný, ošetřený a funkční. K tomu se vytváří protokoly a v případě pochybení se vrací systém zpět do kompetentních rukou pro opravu. Plán projektu se průběžně aktualizuje.

1.1.6 Nasazení

V této fázi nasazení se nejčastěji informační systém předává do rukou zákazníka. Provádí se školení, jak vůbec s novým softwarem zacházet. Stejně tak organizace společnosti si musí na nový systém zvyknout.

Nastává otázka, jak vůbec nový systém do produkce zavést. Pro lepší pochopení vydávání v rámci nového softwaru je příhodné zaměřit se na SRLC (Software Release Life Cycle). Na zavedení se kladou vysoké nároky na efektivitu. Mohou se použít dva přístupy přepínání z původního na nový systém, a to:

- přímé – náhlá výměna starého systému za nový během krátkého časového intervalu, kdy uživatelé nejsou přítomni,
- paralelní – oba systémy běží současně a starý se postupně přesouvá na pozadí a nahrazuje ho nový. Tato situace je příznivější, protože si uživatelé zvykají a samotný systém se re-validuje v ostrém provozu za cenu pomalejšího nasazení.

1.1.7 Údržba

Na údržbu je vynaložena speciální pozornost. Během nasazení se mohou objevit nedostatky, které doposud nebyly odhaleny. Následné chyby je potřeba řešit. Taktéž vzniká důraz na zálohování a pomoc uživatelům.

Aby se zamezilo pravděpodobnosti neúspěchu vývoje softwaru, používají se tyto základní koncepty:

- oblast omezení – stanovení priorit pro požadavky a funkce softwaru tak, aby systém nebyl moc velký,
- progresivní vzestup – vývoj založený na přírůstcích se zpětnou vazbou,
- před-definovaná struktura – vytvoření plánu projektu zahrnující rozpočet, čas a milníky,
- přírůstkové plánování – proces plánování pomocí zvládnutelných, dílčích a realizovatelných kroků.

1.2 Modely

V rámci životního cyklu vývoje se také eviduje několik známých modelů/rámců, jak k implementaci SDLC přistupovat.

1.2.1 Vodopádový model

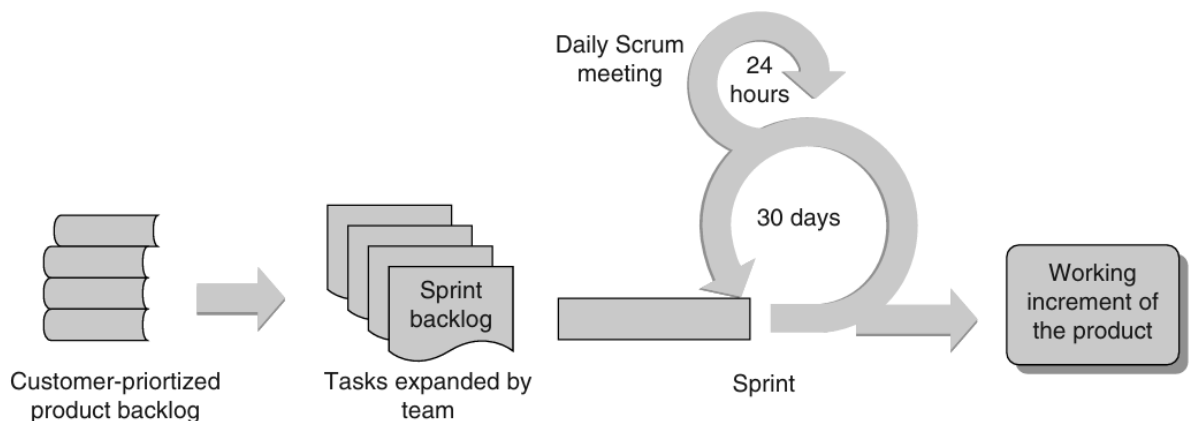
Tento model spoléhá na jasně a plně specifikované zadání, požadavky, jelikož přeskokovat nebo vracet se napříč etapami je náročné, ne-li nemožné. Z toho vychází název samotný, kde reprezentace tekoucí vody z jednoho místa do druhého ilustruje nemožnost vrátit se zpět. Fáze na sebe navazují, mohou se překrývat a prolínat, nicméně jakmile je jedna dokončena, již se k ní nevracíme. Jako metodika pro malý projekt bez potřeby změn s krátkými časovými rámci je pro svoji jednoduchost implementace v takovém případě vhodná. Manažerský dohled v tomto ohledu disponuje nenáročnou kontrolou při plnění projektového plánu. Mezi velké nevýhody patří reakce na problémy a změny, jelikož rámec nenabízí pružnost ve vývoji.

1.2.2 Prototypování

Jedná se o metodu, kdy je cílem vytvořit co nejdřív prototyp požadovaného systému. Ten je následně testován a konzultován přímo se zákazníkem, který vrací hodnotnou zpětnou vazbu. Jde o velice pružný přístup k vývoji softwaru, kde nemusí být přesně dané mantinely na počátcích projektu. Pokud se jedná o komplexnější projekt, na kterém pracuje více skupin, můžou se objevit rychle problémy v jeho vedení. Na místě je také náchylnost na dlouhodobé prodlužování a náklady.

1.2.3 Agilní model

Agilní, od slova čilý, model si zakládá na těsné komunikaci mezi námi a zákazníkem. Představuje takzvaný iterativní vývoj, kde se metodika zaměřuje na krátké a opakované cykly pro realizaci požadavků, na které se získá vzápětí zpětná vazba. I když se vyžadují vyšší nároky na vedení, je tento model velice populární pro pružnost v reakci na změny požadavků a toho, co zákazník skutečně chce. Nejznámějším zástupcem z této kategorie je Scrum.



Obrázek 1: Scrum model [1 str. 45]

Scrum je agilní metodika pro řízení vývoje softwaru, kterou vytvořili Ken Schwaber a Jeff Sutherland. Na rozdíl od tradičních přístupů, které se soustředí na pevně definované procesy a fáze, Scrum vnímá vývoj softwaru jako empirický proces s komplexními vstupy a výstupy, který se neustále transformuje a přizpůsobuje měnícím se okolnostem. Název Scrum je inspirován ragby, kde dva soupeřící týmy střetávají v silovém boji o míč, což slouží jako metafora pro dynamický a nepředvídatelný proces vývoje softwaru v metodice Scrum. Základní principy Scrumu zahrnují hodnotovou orientaci, která se zaměřuje na dodávání hodnoty pro zákazníka v co nejkratším čase, sebeorganizaci týmu, vlastnictví rolí a adaptabilitu, která klade důraz na průběžné zlepšování a adaptaci na změny.

Důležitými artefakty Scrumu jsou:

- **Product Backlog:** Role Product Owner spravuje artefakt, která ho plní prioritizujícími požadavky na systém.
- **Sprint Backlog:** Artefakt zde uchovává již detailně zpracovaná zadání úkolů vycházející z Product Log. Kolekce se zpracovává během následujícího sprintu.
- **Sprint Increment:** Výsledek sprintu a zpracování Sprint Backlogu pro prezentaci a další konzultaci se zákazníkem.

Scrum definuje také tři klíčové role: Product Owner, Development Team a Scrum Master, které mají své specifické úkoly a odpovědnosti při vývoji softwaru.

1.2.4 Iterativní model

Jak bylo zmíněno u agilního modelu, jedná se přístup, pomocí kterého dochází k iterativnímu vylepšování a rozšiřování projektu. Na začátku se lze spokojit s malým množstvím požadavků a postupem času jako strom roste po specificky určených menších pracích výsledný produkt.

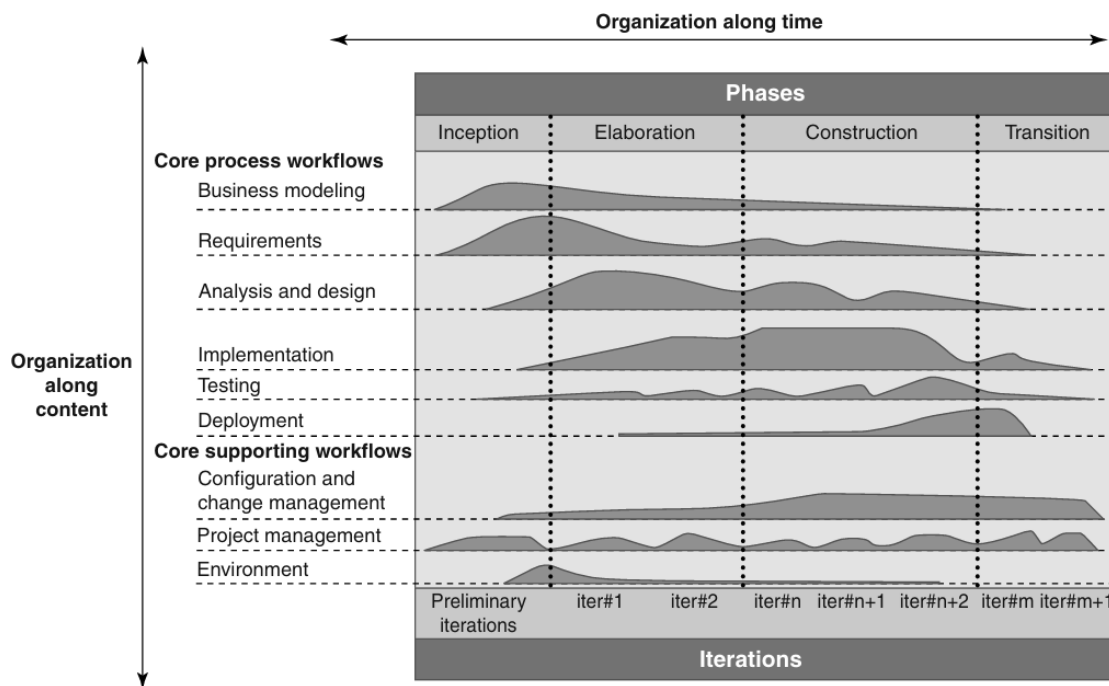
1.2.5 Spirálový model

Spirálový model obohacuje vodopádový model o iterace, a tak zavádí pružnost do zmíněného přístupu. Projekt se dělí na menší části, kde pro každou z nich se dále specifikují, analyzují se rizika, implementují se a naplánuje se další cyklus. V modelu se může aplikovat taktéž samotné prototypování.

1.2.6 Rational Unified Process

RUP, neboli Rational Unified Process, představuje komplexní metodiku pro řízení vývoje softwaru, která se odlišuje od tradičních přístupů tím, že klade důraz na disciplinovaný a iterativní postup vývoje. Jeho základními principy jsou disciplinovaný přístup, iterativní vývoj, zaměření na rizika a uživatele. Rozděluje vývoj do čtyř fází: zahájení, příprava, konstrukce a předávání, přičemž každá fáze obsahuje jednu či více iterací.

V každé iteraci se vytváří projekce softwaru, a to s ohledem na požadavky, které jsou definovány a specifikovány v rámci práce s požadavky. Jednotlivé iterace jsou podpořeny různými pracovními postupy, známými jako disciplíny, které zahrnují modelování, sběr a analýzu požadavků, návrh, implementaci, testování, nasazení a řízení projektu.



Obrázek 2: RUP Model [1 str. 46]

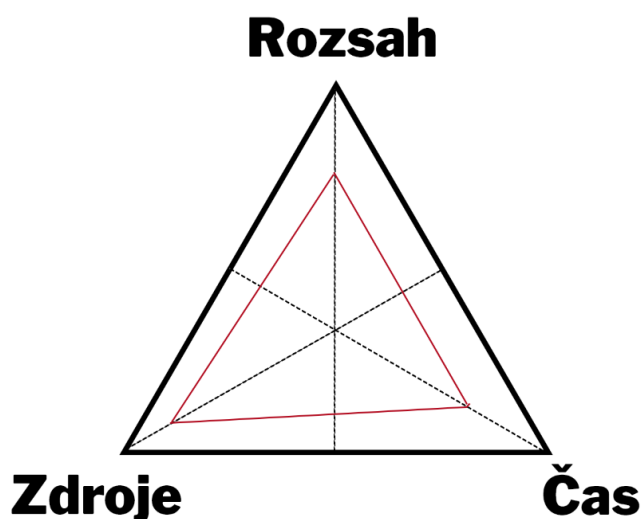
1.2.7 Big Bang model

Model vhodný pro akademické, či malé projekty, kde figuruje pár klíčových osob, nebo běžně jen jedna. Software se tvoří za pochodu bez velkých analýz a plánování. Jaké požadavky přijdou, takové se implementují rovnou do projektu. Takový přístup může být v rámci jednoduchosti velice efektivní pod podmínkou dobré komunikace mezi oběma stranami, nicméně v širším časovém rámci a při větším projektu je Big Bang neudržitelný.

[2], [3], [1 stránky 39-46]

1.3 Projektový trojúhelník

Jako prvotní model pro práci s rozsahem informačního systému a pro jeho definování lze použít takzvaný projektový trojimperativ. Jedná se o známé dogma definující limity projektu v třísloužkovém trojúhelníku.



Obrázek 3: Projektový trojimperativ (zdroj vlastní)

Cílem je najít těžiště mezi složkami. Často je jedna část prioritizována nad ostatními, pak je nutné přizpůsobit zbylé složky – ubrat v jejich rámci na nárocích, či je naopak posilnit, jiným slovy najít požadovaný balanc. V ploše trojúhelníku se dle priorit lze posouvat k jednotlivým vrcholům – čím blíží, tím důležitější aspekt projektu. Abychom pokryly komplexně všechny limity, je možné vytvořit vnitřní trojúhelník s vrcholy korespondujícími vnějšímu hraničnímu modelu a projektovat tak upřednostňování dílčích složek v projektu ve všech třech směrech. Důležitým ukazatelem modelu je kvalita. Zaměřením pouze na jeden vrchol budou adekvátně trpět protilehlé body.

Složky trojúhelníku jsou:

1.3.1 Rozsah

Rozsah představuje definici množství práce a její kvality. Dobrou formou uchopení je dělení rozsahu na drobné, dílčí body, které lze lépe specifikovat a analyzovat. Přímá úměra funguje se zdroji a časem: pokud budou vysoké nároky na kvalitu a rozsah práce, bude potřebovat adekvátně více času a zdrojů (pracovníků, peněz...).

1.3.2 Zdroje

Zdroje určují, na kolik je ochota do projektu investovat. Nemusí se jednat čistě o finanční prostředky, i když mnohé podsložky se jimi dají vyjádřit. Jak bylo zmíněno, jednat se může o pracovní sílu, tak i vybavení. Pokud se bude šetřit na zdrojích, najme se jen jeden programátor

jako jedno z možných řešení, vývoj bude trvat déle, nebo se bude muset ubrat z rozsahu práce. Při porušení závislostí zase kvalita cílů projektu výrazně upadne.

1.3.3 Čas

Vymezení doby trvání jak celého projektu, tak dílčích milníků a úkolů. Nedostatek času opět vede k úpravě rozsahu práce a zdrojů. Na čas navazuje také jeho plánování. Neefektivní využití času vede k problémům prodlužování nebo problémům korelujících složek trojúhelníku.

[4], [5 str. 10]

2 ITIL 4

ITIL (z anglického IT Infrastructure Library) je rámec pro řízení projektů a služeb ve sféře informačních technologií. Jedná se v kostce o sbírku celosvětově uznávaných praktik pramenící ze zkušeností napříč časem a vytvořené největšími hráči, společnostmi, dominující na poli služeb IT. Označeným anglickým slovem „Library“ není rámec nadarmo. ITIL se skládá z mnoha publikací nejen o řízení, ale také o systému jeho školení a certifikací.

2.1 Seznámení

Implementace rámce ITIL poskytuje pro firmy jednu z možností jako lepší řízení rizik, obecnou správu, vztahy se zákazníky a zúčastněnými stranami, finanční kontrolu a obecně komplexní pokrytí jednotlivých oddělení a fází IT služeb, aby proces zůstal flexibilní, unesitelný, efektivní a vůbec úspěšný. Také je nutno podotknout, že se nejedná o striktní scénář, ale o praktiky, které lze vzít jednotlivě a aplikovat je tak, aby vyhovovali prostředí a záměru.

2.2 Historie

Podnětem pro vytvoření určitého standardu pro práci v oblasti IT služeb byl nadmíru rychlý vývoj informačních technologií, jehož trend nemá tendenci stagnovat ani dnes, a také velká poptávka po digitalizaci a poskytování služeb ve virtuálním prostředí. Jelikož chaos a obecně neuspořádanost v IT inklinovaly k velkému růstu společně s rozšiřováním služeb, začlo se tlačit na udržitelnou architekturu procesu a životního cyklu.

Prvotní ITIL spatřil světlo světa již koncem osmdesátých let dvacátého století. Základní kámen položila britská vládní agentura pro počítače a telekomunikace – Central Computer and Telecommunications Agency (CCTA). Jednalo se postupně o 30 publikací, které měly za cíl vmísit i do soukromého sektoru ověřené postupy v IT dle dosavadních poznatků shromážděné od subjektů z celého světa, nicméně v té době se jimi řídily primárně jen vládní složky.

V roce 2001 přejali iniciativu z úřadu OGC – Úřad pro vládní obchod, který zastřešil další agentury společně s CCTA, pro přijetí doposud funkčních postupů a osvěty veřejnosti. S tímto vychází i druhá verze ITIL, která je značně vytříbená vůči rozsáhlým svazkům minulých dob.

V roce 2007 byly zveřejněny oficiální základy pro verzi ITIL v3 pod úřadem OGC, které pomohly utvořit normu ISO/IEC 20000 - Management služeb pro IT z roku 2005, jíž zároveň rámec splňoval. Původní verze ITIL zachovala své pojmosloví, ale důležitým rozšířením byly

procesy a bylo zavedeno plný cyklus služeb s další publikací v roce 2011 pod hlavičkou Cabinet Office.

Rokem 2013 byl ITIL převzat společností Axelos, jež zároveň spravuje aktualizace a licence a to dodnes. Dále se o tři roky později objevila příručka „ITIL Practitioner“ jako možnost lepšího porozumění a zavedení rámce. Nejaktuálnější vydání označované ITIL v4 vyšlo v roce 2019 a přináší relativně velké přepracování původního konceptu tak, aby bylo flexibilnější a agilnější pro moderní podniky.

[6], [7], [8]

Klíčovými změnami verze 4 vůči svému předchůdci jsou:

- Procesy, kterých bylo 26, se transformovaly do 34 praktik.
- Původní verze specifikovala model životního cyklu. Ten byl nahrazen Systémem hodné služby (Service Value System).
- Devět principů se zúžilo na sedm.
- Návrh služeb zvaný „4 P“ (People, Products, Partners, Processes) přešel na čtyřdimenzionální model.
- Struktura osvědčení se reformovala z pěti na čtyři úrovně.

[6]

2.3 Pojmy a definice

Jednou z nejdůležitějších záležitostí pro jednotnou komunikaci je odborná terminologie, a to jak pro komunikaci v rámci IT týmu, tak pro správné pochopení se zadavatelem, jenž poptává službu nebo produkt. Proto se často ke specifikacím u metodik vývoje přidává i slovníček pojmů (glosář).

Následující část popisuje samotné fráze rámce ITIL4, které jsou dále zmiňovány. Z nich lze pochopit klíčová paradigma, jež definují strategický koncept řízení služeb.

Správa služeb (Service management) – jedná se o speciální způsoby, jak organizace poskytují služby, které mají pro zákazníky hodnotu. Aby organizace mohla rozvíjet specializované organizační schopnosti, musí porozumět tomu, co je hodnota, kdo jsou její zainteresované strany a jak služby vytvářejí hodnotu. Rozvoj specializovaných organizačních schopností vyžaduje, aby organizace měla jasnou představu o tom, co zákazníkům přináší hodnotu, jak jsou jejich zainteresované strany propojeny a jak mohou služby uspokojovat

potřeby zákazníků. Organizace, které chtějí rozvíjet specializované organizační schopnosti, musí být schopny porozumět komplexnímu systému hodnot, který zahrnuje zákazníky, zaměstnance, dodavatele a další zainteresované strany.

Hodnota – je jedno z klíčových slov metodiky. Jednoduše lze říct, že úkolem organizace je vytvářet produkty a služby, které mají pro její zainteresované strany hodnotu. Samotnou hodnotu lze vnímat tedy jako prostředek, který přináší třeba užitečnost, uspokojuje poptávky a tužby a je pro někoho důležitý. Samotný pojem je velice abstraktní a v různých odvětvích má naprosto jinou podobu. Důležité je, že je přínosem pro cílové osoby.

Spoluvytváření hodnot (Value co-creation) – na základě historického pohledu byl trend organizací poskytujících služby předat výše uvedenou hodnotu zákazníkovi, ale ten sám na konci řetězce nebyl brán v potaz jako možný článek pro úpravu hodnot, které by se mohly svým potenciálem dále vyvíjet. Proto přišlo na řadu uvědomění, i kvůli konkurenceschopnosti, pro začlenění uživatelů a zákazníků, kteří následně a nepřímou budou spoluúčastníky řetězce tvorby hodnot, tím pádem se posílí jak produkt, služba, tak vztah k odběratelům.

Organizace – seskupení lidí, popř. jedna osoba, kteří zastávají odpovědné funkce, k nimž se vážou pravomoce a vztahy k dosažení cílů. Je to jeden ze zainteresovaných subjektů v rámci řízení služeb. Její velikost a komplexnost se může silně lišit, a to v počtu osob samotných, tak i v rámci role, kterou zastává.

Poskytovatelé služeb (Service providers) – jedná se o subjekt, organizaci nabízející hodnotovou službu. Lze si ji představit jako interní oddělení firmy IT. Poskytování služeb může být rozšířeno i do volného trhu, kde zákazníci zahrnují jednotlivce i malé a střední podniky. V takovém případě je důležité, aby organizace pečlivě vyhodnotila své cílové trhy a efektivně je rozdělila podle potřeb zákazníků. Další možností pro poskytovatele je zapojení do servisní aliance, spojení dvou nebo více organizací, které sdílejí zdroje a dovednosti k poskytování lepších služeb zákazníkům. Ať už je organizace interní nebo externí, klíčové je mít jasný obraz o zákaznících a jejich potřebách. Důležité je také porozumět konkurenčnímu prostředí a rozdílům mezi poskytovateli na trhu. Schopnost vyvíjet a provádět procesy řízení služeb je pro organizaci klíčová. To zajišťuje vysokou kvalitu a dodání služeb včas a v souladu s rozpočtem. Měření výkonu a identifikace oblastí pro zlepšení jsou rovněž důležité. Nakonec je důležité

udržovat dobré vztahy se zákazníky. To znamená efektivní komunikaci, řešení problémů a poskytování excelentních služeb.

Spotřebitelé služeb (Service consumers) – subjekty, které odebírají poskytovanou službu. Spotřebitel služby zadává požadavky na službu, využívá její výstupy a poskytuje feedback poskytovateli služby. Je nutné si uvědomit, že spotřebitelé nabývají sami dalších dílčích vztahů. Zákazník je typ spotřebitele služby, který za službu platí. Zákazník služby je obvykle definován jako osoba nebo organizace, která má smlouvu se poskytovatelem služby a je oprávněna využívat službu. Zákazník služby může být také ten, kdo platí za službu bez uzavření smlouvy, když má službu zdarma nebo když jej služba sponzoruje poskytovatel služby. Uživatel je typ spotřebitele služby, který službu využívá. Uživatel služby může být zákazníkem služby, ale také nemusí. Ne všichni uživatelé služeb musí platit za službu, třeba když služba poskytuje volný přístup nebo když službu sponzoruje poskytovatel služby. Sponzor je typ spotřebitele služby, který podporuje poskytování služby. Sponzor služby může být zákazníkem služby, uživatelem služby nebo jinou osobou nebo organizací. Sponzor služby může poskytovateli služby poskytnout finanční prostředky, zdroje nebo jinou podporu.

Další zúčastněné strany (Stakeholders) – jedná se o další účastníky v řetězci tvorby hodnotové služby. Důležité je všimnout si a udržovat vztahy mezi nimi. Klíčovou složkou pro příklad jsou samotní zaměstnanci, nebo dílčí dodavatelé a investoři, akcionáři.

Tabulka 1: Příklad zainteresovaných stran a hodnot [8 str. 11]

Zúčastněná strana	Příklad hodnoty pro zainteresovanou stranu
Zákazníci služeb	Dosažení výsledků a optimalizace nákladů a rizik
Poskytovatel služeb	Financování od zákazníka, obchodní rozvoj, zlepšení obrazu
Zaměstnanci poskytovatele služeb	Finanční a nefinanční motivace, kariéerní a profesní rozvoj, smysluplná práce
Společenství a komunita	Zaměstnanost, daně, příspěvky organizací ke rozvoji komunity
Charitativní organizace	Finanční a nefinanční příspěvky od jiných organizací
Akcionáři	Finanční výhody, jako jsou dividendy, pocit jistoty a stability

Zákazníci služeb získávají hodnotu z poskytování služeb tím, že dosahují požadovaných výsledků a optimalizují své náklady a rizika. A zákazník, který používá IT služby, může dosáhnout požadovaných výsledků, jako je zvýšení produktivity nebo zlepšení zákaznického servisu. Zákazník může také optimalizovat své náklady a rizika tím, že používá služby, které jsou efektivní a spolehlivé.

Poskytovatelé služeb získávají hodnotu z poskytování služeb tím, že získávají financování od zákazníků, rozvíjejí své podnikání a zlepšují svůj obraz. Poskytovatel IT služeb může získat financování od zákazníků tím, že jim poskytuje vysoce kvalitní služby, které splňují jejich potřeby. Poskytovatel IT služeb může také rozvíjet své podnikání tím, že rozšiřuje své portfolio služeb nebo vstupuje na nové trhy. Poskytovatel IT služeb může zlepšit svůj obraz tím, že buduje si dobrou pověst mezi zákazníky a veřejností.

Zaměstnanci poskytovatelů služeb získávají hodnotu z poskytování služeb tím, že mají finanční a nefinanční motivaci, příležitosti pro kariérní a profesní rozvoj a smysluplnou práci. Zaměstnanci poskytovatele IT služeb mohou mít finanční motivaci v podobě mzdy, bonusu nebo akcií. Mohou také mít nefinanční motivaci v podobě dobrého pracovního prostředí, příležitostí pro vzdělávání a rozvoj nebo pocitu, že jejich práce má smysl.

Společenství a komunity získávají hodnotu z poskytování služeb tím, že vytvářejí zaměstnanost, generují daně a podporují rozvoj komunity. Organizace, která poskytuje sociální služby, může vytvářet zaměstnanost pro místní obyvatele, generovat daně pro místní vládu a podporovat rozvoj komunity tím, že poskytuje služby potřebným.

Charitativní organizace získávají hodnotu z poskytování služeb tím, že získávají finanční a nefinanční příspěvky od jiných organizací. Například charitativní organizace, která poskytuje pomoc lidem v nouzi, může získat finanční příspěvky od vládních organizací, firem nebo jednotlivců. Charitativní organizace může také získat nefinanční příspěvky, jako je dobrovolnická práce nebo poskytnutí materiálních zdrojů.

Akcionáři získávají hodnotu z poskytování služeb tím, že mají finanční výhody, jako jsou dividendy, a pocit jistoty a stability. Akcionáři společnosti, která poskytuje IT služby, mohou mít finanční výhody v podobě dividend vyplácených společností. Akcionáři mohou také mít pocit jistoty a stability, pokud společnost poskytuje vysoce kvalitní služby, které jsou žádané zákazníky.

Služby – umožňují spoluvytváření hodnot se zákazníkem, jelikož mu zjednodušují dosažení chtěných výsledků, aniž by se musel starat o rizika a náklady, jejichž otěže převzal poskytovatel.

Produkt – je hodnota vytvořená pro spotřebitele utříbením zdrojů organizace. Produkty slouží k uspokojení poptávky cílové skupiny, proto je důležité brát zřetel na jejich potřeby. Produkt je dost často škálovatelný, aby se pokrylo a oslovilo více potenciálních spotřebitelů. Zákazník často nevidí všechny aspekty spojené s produktem.

Nabídka služeb – jedná se o souhrn služeb, které organizace poskytuje, pevně definuje a předkládá svým zákazníkům. Nabídka může nabývat několika základních kategorií. Lze uvést příklad s restauračním zařízením. Základní služby sestávají z vaření a servírování. Doplňkové služby mohou nabízet, dnes již rozšířenou možnost, rozvážky nabízeného stravování přepravní službou až ke dveřím. Zatímco fyzické produkty jako jedna složka služeb může být prodej balených zákusků s sebou. Další formou je přístup ke zdrojům, kde zdroj může zastupovat v restauračním případě rezervaci prostorů pro větší skupinu lidí. V poslední řadě je servisní činnost, jež jako služba může brát podobu reklamaci objednávky ve stravovacím zařízení.

Servisní vztah – rozlišuje se mezi stranami, které službu vytváří a těmi, kteří službu konzumují. Napříč účastníky hraje roli vzájemná spolupráce a řízení vztahů a spoluvytváření hodnot na základě domluvených nabídkových služeb. Nemusí se vždy jednat o vztah dvou oddělených subjektů, ale dost často vztah nastává mezi vnitřním uspořádáním organizace – mezi odděleními.

Výstup a výsledek – ačkoliv by se tato slova zdála jako synonyma, lze v tomto případě každé z nich významově odlišit. Výstupem se definuje završení činnosti tvořící hodnotu. Výsledek oproti tomu je realizován výstupy. Z pohledu konzumenta a producenta si lze představit rozdíl na případu z předchozího odstavce. Výstupem poskytovatele, restaurace, je dobré a dostupné jídlo a příjemné prostředí. Z pohledu zákazníka, na které cílíme, je výsledkem spokojenost a vysoká šance návratu návštěvníka do zařízení.

Náklady – prostředky, většinou finanční, které se musí vynaložit pro realizaci činnosti, či přístupu ke zdroji. Pro spotřebitele služeb může jít o náklady, které jednak spotřebiteli náklady ušetří (čerpám cizí služby a nemusím pro ně platit zaměstnance, které je realizují), a na druhou stranu náklady vznikající z čerpání služby, kde je pro představu zapotřebí nakoupit zařízení,

z kterých se bude připojovat na pronajatý přístup k internetu od poskytovatele internetu (spotřební služba typu přístupu ke zdroji).

Rizika – jedná se o pravděpodobnost úspěchu, či v opačném případě neúspěchu, která může nastat v procesu dosahování cílů jak dílčích, tak celkových. Rizika jsou ukazatelem nejistoty pro výsledky. Spotřebitel služeb může registrovat následující stavy: ušetření rizik při samotné realizaci služby jiným poskytovatelem.

Možná rizika:

- Nedostatek kvalifikovaných IT specialistů, kteří by zajišťovali chod IT služeb spotřebitele
- Výpadky elektřiny, které by mohly ovlivnit fungování IT služeb spotřebitele
- Selhání serveru, úložiště nebo jiných komponent, které spotřebitel používá pro své vlastní IT služby.

Druhý stav rizik se týká přímého čerpání služby, převedení odpovědnosti na poskytovatele.

Možná rizika:

- Ztráta kompatibility – riziko, že daná služba nebude kompatibilní s novými technologiemi nebo s jinými službami, které spotřebitel používá.
- Narušení bezpečnosti – riziko, že dojde k hackerskému útoku nebo jinému narušení bezpečnosti, které by mohlo ohrozit data spotřebitele.
- Ukončení činnosti poskytovatele služby – riziko, že poskytovatel služby ukončí svou činnost a spotřebitel tak ztratí přístup k dané službě.

Spotřebitel sám by měl být proaktivně zainteresován v řízení rizik pro snížení pravděpodobnosti neúspěchu a to následovně:

- Spotřebitel by měl jasně definovat, co od dané služby očekává a jakých výsledků chce dosáhnout. To zahrnuje specifikaci funkčních i nefunkčních požadavků, jako je dostupnost, výkonnost, bezpečnost a kompatibilita. Čím jasněji jsou definovány požadavky, tím menší je riziko, že služba nebude splňovat očekávání spotřebitele.
- Spotřebitel by měl identifikovat faktory (CSF), které jsou pro úspěšné užívání služby klíčové. To může zahrnovat dostupnost personálu s potřebnými znalostmi, kompatibilitu s existujícími systémy a dodržování specifických regulačních požadavků. Spotřebitel by měl také jasně sdělit poskytovateli služby případná omezení, která se na

ni vztahují. To může zahrnovat omezený rozpočet, časové limity nebo specifické požadavky na licencování.

- V některých případech bude pro poskytování služby nutné, aby měl poskytovatel přístup k datům, systémům nebo infrastruktuře spotřebitele. Spotřebitel by měl zajistit, aby tento přístup byl dostupný po celou dobu trvání služby. To může zahrnovat zřízení VPN připojení, sdílení přístupových údajů nebo fyzický přístup k prostorám spotřebitele.

Mezi hlavní pojmy se také řadí **užitečnosti** a **záruky**. Užitečnost a záruky lze asociovat s funkčními a nefunkčními požadavky. U užitečnosti zkoumáme, jak přesně služba funguje, co může poskytnout a jestli je pro vhodná. Záruky zas více inklinují k nefunkčním požadavkům. Jedná se o ujištění, jaké specifikace má služba samotná ve vztahu její konzumace. Pro představu se může jednat o: dostupnost 99,9 %, kapacita pro zpracování 10 000 transakcí za sekundu, úroveň zabezpečení dle standardu PCI DSS nebo kontinuitu provozu i v případě výpadku napájení. I zde se rozhoduje o vhodnosti a přijetí služby samotné. Vše se musí zvážit, vytknout klady a zápory a zjistit, jestli se váha na pomyslných vahách přiklání spíše k pozitivnímu přínosu pro tvorbu výsledků a hodnot se zohledněním rizik, ba naopak k zápornému poměru, že služba by přinesla více starostí než hodnot.

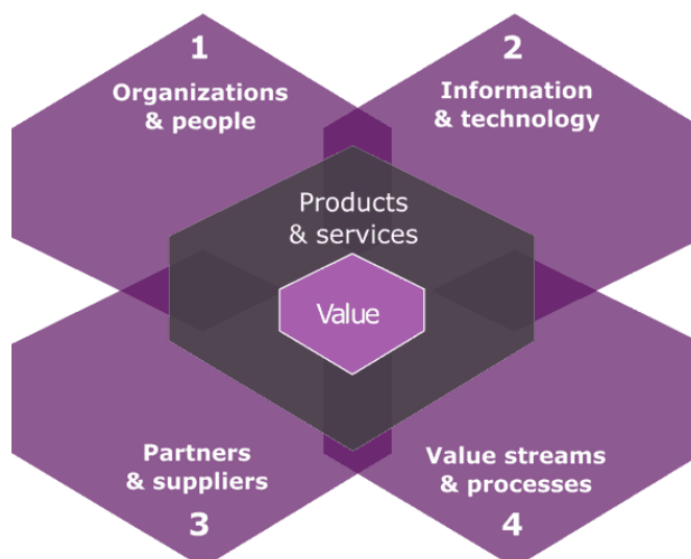
[8 stránky 6-18]

2.4 Řízení služeb přes čtyři dimenze

Efektivní fungování organizace vyžaduje komplexní přístup k řízení služeb, který zohledňuje všechny relevantní aspekty. V praxi se však organizace často zaměřují pouze na jeden aspekt, čímž riskují suboptimální výsledky a nízkou efektivitu.

Pro lepší představu lze uvést následující případy:

- zlepšení procesů bez ohledu na lidi – zavedení efektivnějších procesů může selhat, pokud nebudou zohledněny potřeby a dovednosti pracovníků,
- implementace technologií bez ohledu na procesy – nové technologie nemusí přinést očekávaný benefit, pokud nebudou integrovány do stávajících procesů a nebudou pro ně adekvátně vyškoleni uživatelé.



Obrázek 4: Čtyři dimenze správy služeb [9]

Proto se rámec ITIL4 zaměřil na pokrytí všech účastněných stran a aspektů a vytyčil následující dimenze:

2.4.1 Organizace a lidé

Tato dimenze se nezabývá pouze strukturou a hierarchií, ale také kulturou organizace a schopnostmi jejích zaměstnanců. Zdravá organizační kultura podporuje cíle organizace a motivuje zaměstnance k žádoucímu chování.

Vedení organizace by mělo aktivně prosazovat hodnoty, které povedou k pozitivnímu pracovnímu prostředí. Postupem času se z těchto hodnot a postojů stává nedílná součást kultury organizace. Jedním z podob může být podpora kultury důvěry a transparentnosti umožňuje zaměstnancům upozorňovat na problémy a jejich včasnou eskalaci. Tím se předchází negativním dopadům na zákazníky.

2.4.2 Informace a technologie

Efektivní řízení služeb vyžaduje silnou dimenzi informací a technologií. Zahrnuje to informace a znalosti potřebné pro provozování a zlepšování služeb, stejně jako technologie, které tyto služby umožňují. Pro představu personální služba spoléhá na přesné informace o zaměstnancích, zatímco správa sítě vyžaduje data o síťových připojeních. Organizace by měly pečlivě zvažovat otázky týkající se informačního řízení, jako je ochrana dat a dodržování předpisů. Technologie by měla být kompatibilní s celkovou architekturou a splňovat požadavky

na životaschopnost a dovednosti k její podpoře. Rozvoj technologií jako umělá inteligence a cloudových výpočtů zásadně ovlivňuje řízení služeb, a proto je důležité vybírat technologie, které nejen splňují aktuální potřeby, ale nabízejí i možnosti pro budoucí rozvoj.

Jelikož služby a informační technologie mají v dnešní době velice blízko a v rámci tématu práce je také potřeba rozvést specifičtější technologickou dimenzi, následující otázky pomáhají v rozhodování ohledně služeb v IT:

Integrace s existující infrastrukturou:

Technická kompatibilita

- Lze technologii bezproblémově integrovat do stávající softwarové a hardwarové architektury organizace a klientů?
- Existují kompatibilní rozhraní a protokoly pro bezproblémovou komunikaci s existujícími systémy?
- Jaké úpravy a modifikace stávající architektury by implementace technologie vyžadovala?

Organizační kompatibilita

- Je implementace technologie v souladu s procesy, standardy a bezpečnostními politikami organizace?
- Jaké změny v organizačních procesech a workflow bude nutné provést pro efektivní využití technologie?
- Jaká úroveň podpory a školení bude pro uživatele technologie nutná?

Interoperabilita s technologiemi partnerů:

Kompatibilita produktů

- Jsou technologické produkty používané partnery a zainteresovanými stranami kompatibilní s novou technologií?
- Existují kompatibilní standardy a protokoly pro výměnu dat a spolupráci mezi systémy?

Spolupráce a sdílení dat:

- Jakým způsobem bude technologie usnadňovat spolupráci a sdílení dat mezi organizací a jejími partnery?

- Jaké jsou bezpečnostní a právní aspekty sdílení dat mezi různými systémy?

Dopady nově vznikajících technologií:

Analýza trendů

- Jaké jsou trendy v oblasti strojového učení, umělé inteligence a internetu věcí?
- Jaké jsou potenciální dopady těchto technologií na odvětví, ve kterém organizace působí?

Strategická adaptace

- Jaké kroky může organizace podniknout pro proaktivní využití nově vznikajících technologií?
- Jaké jsou potenciální hrozby a rizika spojená s ignorováním těchto technologií?

[8 str. 28]

2.4.3 Partneri a dodavatelé

Spolupráce mezi firmami se vyznačuje širokou škálou propojení a formálnosti. Může se jednat o striktně definované smluvní vztahy s jasně danými úkoly a zodpovědnostmi, nebo o volná partnerství založená na sdílených cílech a rizicích, kde firmy úzce kooperují na dosažení společných výsledků.

Pozice firmy v tomto spektru závisí na její strategii a cílech v daném partnerství. Pokud se firma zaměřuje na stabilitu a minimalizaci rizik, bude preferovat formální smluvní spolupráci. Naopak firma, která klade důraz na flexibilitu a inovace, bude spíše usilovat o partnerská uspořádání.

Důkladná analýza variability v typech vztahů a zvážení vlastních cílů a strategie pomůže firmám vybudovat efektivní a prosperující partnerství.

2.4.4 Hodnotové toky a procesy

Dimenze hodnotové toky a procesy se specifikuje na samotné činnosti uvnitř i vně společnosti a analyzuje, jak úkony spolupracují, jestli se neduplikují, jestli efektivně spolupracují a integrují se pro vytyčenou tvorbu hodnot všech účastníků.

Proces lze chápat v kontextu ITIL a organizací jako posloupnost a zřetězení kroků, procedur pro přijetí, zpracování a předání vstupů další proceduře. Dost obdobně lze specifikovat hodnotový tok, ale z jiného úhlu pohledu. Rozdíl si je uveden níže.

Hodnotový tok se zaměřuje na mapování kroků a činností nutných k přeměně vstupů (požadavků zákazníků, surové informace, ...) na výstupy (hodnotné služby) pro zákazníky a další zainteresované strany. Hodnotový řetězec organizace provázáním jeho činností definuje zmíněný tok. Přehled nad procesy a toky je klíčovým aspektem pro zdokonalování a optimalizaci, proto z hlediska ITIL by měl mít každý výstup, služba či produkt... svůj identifikovaný tok.

Lze si představit proces instalace softwaru na počítači uživatele. Z pohledu procesu se jedná o sled kroků, které zahrnují stažení instalačního souboru, spuštění instalace, zadání licenčních informací a konfiguraci nastavení. Naproti tomu hodnotový tok vnímá instalaci softwaru z perspektivy uživatele. Začíná rozpoznáním potřeby daného softwaru a končí jeho bezproblémovým používáním pro splnění požadovaných úkolů. Hodnotový tok se zaměřuje na optimalizaci celého procesu z hlediska uživatelské zkušenosti, případy jako:

- Zjednodušení stažení a instalace.
- Poskytnutí jasných pokynů a návodů.
- Automatická konfigurace softwaru.
- Nabídnutí online podpory a řešení případných problémů.

Výše specifikované rozměry definují úhly pohledu na veškeré prvky systému hodnotového řetězce služeb (SVS). Opomínání jedné s dimenzí může následně vyústit v problémy typu: nízká kvalita výstupů, neefektivní využívání zdrojů a plýtvání, prodlužování termínů, prodražování, rozkol ve strategii organizace.

[8], [10]

2.5 Systém hodnot služeb

Systém hodnot služeb (SVS) v ITIL představuje klíčový koncept, který umožňuje organizacím efektivně vytvářet a dodávat hodnotu pro širokou škálu zainteresovaných stran, ať už se jedná o interní zákazníky (oddělení v rámci organizace), externí zákazníky (klientskou základnu) nebo další partnery. Jak bylo zmíněno, výstupem je hodnota, naproti tomu vstupy do systému jsou obecně příležitosti a poptávky. Zatímco poptávka nabývá přímočaré podoby ve tvaru potřeby/tužby spotřebitelů po službách a produktech, příležitost má mnoho podob, jako třeba:

- **Zlepšení efektivity procesů:** Automatizace manuálních úkolů implementací robotické automatizace procesů (RPA) a workflow nástrojů zefektivňuje opakující se úkoly a

procesy schvalování. Optimalizace workflow analyzuje a redesignuje procesy pro odstranění zbytečných kroků a duplicit, zatímco implementace štihlé výroby zefektivňuje výrobu.

- **Inovace produktů a služeb:** Průzkum trhu a design thinking pomáhá identifikovat potřeby a přání zákazníků, na jejichž základě se vyvíjí inovativní produkty a služby. Sběr zpětné vazby od zákazníků a implementace programů neustálého zlepšování umožňuje identifikovat a implementovat oblasti pro zlepšení stávajících služeb.
- **Rozšíření trhu:** Analýza trhu identifikuje potenciální trhy, pro které se produkty a služby adaptují na specifické požadavky. Implementace marketingových kampaní a využití online marketingu a sociálních médií oslovuje nové cílové skupiny a rozšiřuje dosah.
- **Zlepšení zákaznické zkušenosti:** Redesign webových stránek a aplikací pro intuitivnější používání a zavedení samoobslužných portálů usnadňuje zákazníkům používání služeb. Využití dat o zákaznících personalizuje nabídky a produkty a implementace programů loajality odměňuje věrné zákazníky.
- **Snížení nákladů:** Analýza a redesign procesů a implementace lean managementu umožňuje najít úspory a zefektivnit výrobu. Provádění auditů nákladů a vyjednávání o lepších cenách s dodavateli identifikuje oblasti, kde lze ušetřit.

SVS se zaměřuje na holistický přístup, kde všechny komponenty a aktivity organizace spolupracují jako systém v rámci ekosystému, ve kterém se propojuje s ostatními subjekty.

Systém se skládá z následujících částí, na které se pohlíží ještě přes zmíněné dimenze, tvořící komplexních celek:

- **Hlavní, klíčové zásady (Guiding principles)**
Hlavní, klíčové zásady představují základní principy, které usměřují rozhodování a jednání organizace v rámci poskytování služeb. Tyto zásady jsou jako kompas, který pomáhá organizaci směřovat své úsilí k dosahování svých cílů a dodávání hodnoty zákazníkům.
- **Řízení organizace (Governance)**
Řízení organizace zahrnuje rámec a procesy, které umožňují a řídí poskytování služeb. Je to způsob, jakým organizace definuje a uplatňuje pravidla, rozhoduje o prioritách a zajišťuje odpovědnost za dosažení svých cílů.
- **Hodnotový řetězec služby (SVC)**

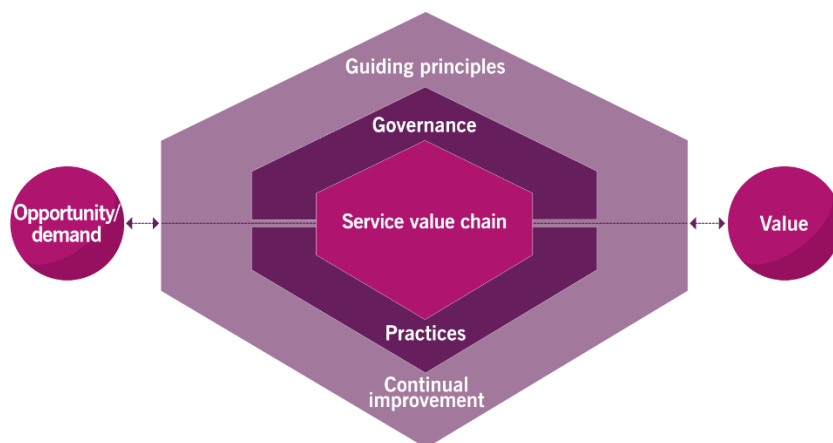
Hodnotový řetězec služby popisuje celkový proces poskytování hodnoty zákazníkům v rámci organizace. Jedná se o sérii aktivit, které vedou k vytvoření a dodávání produktů a služeb, které zákazníci potřebují a ocení.

- **Praktiky (Practices)**

Praktiky představují osvědčené postupy a přístupy k plnění určitých úkolů nebo dosahování cílů v rámci poskytování služeb. Tyto praktiky jsou založeny na odborných znalostech a zkušenostech a slouží jako nástroj pro dosahování optimálních výsledků.

- **Neustálé zlepšování (Continual improvement)**

Neustálé zlepšování je proces systematického hodnocení, analýzy a aktualizace všech aspektů poskytování služeb s cílem dosáhnout vyšší úrovně kvality, efektivity a efektivitu. Je to klíčový prvek pro udržení konkurenceschopnosti a úspěchu organizace v neustále se měnícím prostředí.



Obrázek 5: Systém hodnot služeb [11]

V současném světě, který je charakterizován neustálými změnami a výzvami, se pro dosažení úspěchu organizace stávají klíčovými agilita a odolnost, na které se i zaměřuje samotný koncept systému hodnot služeb v rámci ITIL. Agilita, schopnost flexibilně a rychle reagovat na vnitřní změny, je pro organizaci jako dýchání vzduchu. Pro ilustraci, když dojde k fúzi nebo akvizici, agilní organizace může efektivně sladit a integrovat nové struktury a procesy. Když technologie nebo obchodní modely rychle mění podobu, agilita umožňuje organizaci přizpůsobit se novým podmínkám bez zbytečného zpomalení nebo ztráty konkurenční výhody.

Na druhou stranu je odolnost klíčovým prvkem pro zajištění trvanlivosti a úspěchu v nestálém prostředí. Organizace, která je odolná, má schopnost předvídat a připravit se na vnější změny, a to i ty nepředvídatelné. Když dojde k politickým nebo ekonomickým změnám, odolná organizace je schopna efektivně minimalizovat jejich negativní dopady nebo je dokonce využít jako příležitost k růstu. Stejně tak, když se objeví nové technologické hrozby nebo environmentální výzvy, odolnost organizace znamená schopnost adaptace a inovace, aby se s těmito změnami vypořádala.

Kombinace agility a odolnosti poskytuje organizaci komplexní schopnost prosperovat v rychle se měnícím a náročném prostředí. Zatímco agilita umožňuje rychlou a efektivní reakci na vnitřní změny, odolnost poskytuje pevný základ, který umožňuje organizaci přežít, a dokonce se rozvíjet i za nepříznivých podmínek. Tyto atributy se stávají základem pro úspěch organizace ve 21. století.

2.5.1 Hlavní zásady

Hlavní zásady jsou jako pevné pilíře, na kterých organizace staví svou strategii a provozní modely v oblasti poskytování služeb. Tyto zásady jsou navrženy tak, aby poskytovaly organizační rámec, který je odolný vůči změnám, a přesto dostatečně flexibilní, aby reflektoval aktuální potřeby a trendy v odvětví. Jedním z klíčových principů ITIL SVS je orientace na hodnotu pro zákazníka. To znamená, že veškeré aktivity a rozhodnutí organizace jsou zaměřeny na dodání maximální hodnoty pro zákazníka. Tato zásada může být realizována prostřednictvím pravidelného sběru zpětné vazby od zákazníků a neustálého vylepšování služeb tak, aby lépe odpovídaly jejich potřebám a očekáváním.

Hlavními zásadami jsou:

1. Zaměření na hodnotu

Zaměření na hodnotu je klíčovým pilířem, který posouvá organizace směrem k úspěchu prostřednictvím vytváření hodnoty pro své zákazníky a uživatele služeb, ať už jsou interní nebo externí. Tato zásada je jakýmsi základním kamínkem celého procesu poskytování služeb, neboť klade důraz na to, že každý krok a rozhodnutí organizace by mělo mít za cíl maximalizovat užitek a spokojenost těch, kteří využívají poskytované služby.

2. Začněte tam, kde jste

Zásada Začněte tam, kde jste připomíná, že při snaze o zlepšení služeb a procesů je důležité nejprve pečlivě zhodnotit a využít stávající aktiva a zdroje organizace. Tato zásada klade důraz na to, že cesta k efektivnímu vylepšení začíná právě tam, kde se organizace momentálně nachází, a postupuje odtud.

Co se týče služeb a procesů, i když mohou existovat nedostatky, je důležité si uvědomit, že stávající služby a procesy mohou obsahovat cenné prvky, které lze dále rozvíjet a optimalizovat.

Lidé, konkrétně zaměstnanci s jejich znalostmi, zkušenostmi a dovednostmi, jsou dalším klíčovým faktorem. Jejich zapojení a motivace jsou nezbytné pro dosažení trvalého úspěchu. Organizace by měla aktivně vyhledávat a využívat know-how svých zaměstnanců, ať už jde o jejich odborné znalosti nebo nápady na zlepšení procesů.

Pokud jde o nástroje a technologie, i stávající infrastruktura může být užitečným zdrojem pro zlepšení služeb a procesů. Modernizace a integrace s novými řešeními mohou přinést značné výhody. Organizace může využít existující softwarové nástroje a technologie k automatizaci rutinních úkolů nebo k vylepšení sledování a řízení procesů.

3. Postupujte iterativně se zpětnou vazbou

Zásada Postupujte iterativně se zpětnou vazbou zdůrazňuje důležitost postupného a pružného přístupu při implementaci změn a zdokonalování služeb. Tento přístup umožňuje organizaci lépe se přizpůsobit dynamickým potřebám a očekáváním uživatelů a zároveň minimalizuje rizika spojená s implementací změn.

Snížení rizika je jedním z hlavních benefitů tohoto přístupu. Postupné a iterativní zavádění změn umožňuje lépe identifikovat a řešit problémy, které se mohou vyskytnout v průběhu procesu. Tím se minimalizuje riziko neúspěchu projektu a snižují se negativní dopady na uživatele a obchodní operace organizace. Pokud organizace implementuje nový software, postupné testování a zavádění jednotlivých funkcionalit umožní identifikovat a řešit problémy v raných fázích, což vede k menším komplikacím a rizikům.

Lepší adaptace je dalším klíčovým prvkem tohoto přístupu. Pravidelná zpětná vazba od uživatelů a zainteresovaných stran umožňuje organizaci pružně reagovat na změny potřeb a preferencí. Tím se zajišťuje, že implementovaná řešení co nejlépe odpovídají

skutečným potřebám uživatelů a podnikovým cílům. Pokud organizace nasadí novou službu, průběžná zpětná vazba od uživatelů umožní identifikovat nedostatky a možnosti vylepšení, které lze následně rychle integrovat do služby.

Udržitelný úspěch je dalším cílem tohoto přístupu. Iterativní přístup podporuje kulturu kontinuálního zlepšování a učení se z chyb. Tím organizace vytváří prostředí, ve kterém je inovace a efektivita neustále podporována. Dlouhodobý úspěch implementace je tak zajištěn tím, že organizace je schopna pružně reagovat na změny a neustále zdokonalovat své služby a procesy.

4. Spolupráce a podpora viditelnosti

Spolupráce a podpora viditelnosti je základním principem, který zdůrazňuje nutnost spolupráce a transparentnosti mezi všemi stakeholdery organizace při implementaci a zdokonalování služeb. Tato spolupráce a transparentnost jsou klíčové pro dosažení úspěchu a optimalizaci poskytovaných služeb.

Spolupráce mezi všemi stakeholdery, včetně managementu, IT oddělení, uživatelů služeb a dodavatelů, je základním kamenem pro úspěšnou implementaci a provoz služeb. Pokud organizace plánuje nasadit novou službu, je důležité, aby všechny relevantní strany spolupracovaly při definování požadavků, plánování implementace a poskytování zpětné vazby během provozu služby. Tímto způsobem je zajištěno, že všechny potřeby a očekávání stakeholderů jsou brány v úvahu a že všichni pracují na společném cíli.

Transparentnost je dalším důležitým aspektem této zásady. Informace o poskytovaných službách, procesech a cílech by měly být dostupné a srozumitelné pro všechny stakeholdery. To zahrnuje pravidelné aktualizace o stavu projektů, výsledcích provozu služeb a plánech na budoucí vývoj. Transparentnost umožňuje stakeholderům lépe porozumět situaci a přispívat k rozhodování na základě relevantních informací.

Sdílení znalostí a zkušeností je podstatné pro podporu učení a neustálého zlepšování v organizaci. Stakeholderi by měli aktivně sdílet své znalosti a zkušenosti s ostatními členy týmu, aby se všichni mohli učit z úspěchů a selhání. Tímto způsobem organizace posiluje svou schopnost adaptace a inovace, což přispívá k dlouhodobému úspěchu a udržitelnosti.

5. Přemýšlejte a pracujte komplexně

Zásada číslo pět zdůrazňuje nutnost zaměřit se na celkový kontext a propojenost různých částí organizace při poskytování služeb a dosahování cílů. Tato zásada podporuje integraci a koordinaci všech relevantních procesů, oddělení a technologií, aby mohly efektivně spolupracovat a poskytovat co nejlepší hodnotu pro zainteresované strany.

Integrovaný přístup je klíčovým prvkem této zásady. Organizace by měly aktivně pracovat na integrování všech aspektů své činnosti, aby fungovaly jako harmonický celek. To znamená, že různá oddělení, procesy a technologie by měly být navzájem propojeny a koordinovány, aby organizace byla schopna efektivněji reagovat na potřeby a požadavky svých uživatelů a zákazníků. Když organizace implementuje novou IT službu, je důležité, aby byly integrovány všechny potřebné procesy a systémy, aby byla zajištěna maximální efektivita a kvalita služeb.

Pochopení vzájemných závislostí mezi různými částmi organizace je dalším důležitým aspektem této zásady. Organizace by měly mít jasné povědomí o tom, jak různé části organizace vzájemně souvisejí a jaké jsou jejich vzájemné vlivy. To umožňuje organizaci lépe předvídat a řídit důsledky změn v jedné části na ostatní části organizace. Změna v jednom procesu může mít vliv na další procesy nebo na kvalitu poskytovaných služeb, a proto je důležité, aby organizace měla povědomí o těchto závislostech a byla schopna adekvátně reagovat.

Zaměření na výsledky je třetím klíčovým prvkem této zásady. Namísto zaměření pouze na dílčí cíle jednotlivých oddělení by se organizace měly soustředit na dosažení konečných výsledků a hodnoty pro všechny stakeholdery. To znamená, že veškeré úsilí a rozhodnutí by měly být podřízeny dosažení širších podnikových cílů a přinášení skutečné hodnoty pro uživatele a zákazníky. Jako ukázka, pokud organizace zavádí novou marketingovou kampaň, cílem by nemělo být pouze zvýšení prodeje, ale také budování dlouhodobých vztahů se zákazníky a zvýšení jejich spokojenosti.

6. Jednoduše a prakticky

Zásada Jednoduše a prakticky klade důraz na to, že organizace by měly přistupovat k řešením a procesům s cílem dosáhnout co nejefektivnějších a pragmatických výsledků,

kteřé skutečně přinášejí užitek a hodnotu. Tato zásada zdůrazňuje potřebu minimalizovat složitost a maximalizovat užitek pro zainteresované strany.

Uplatňuje se zde minimalizace kroků jako jeden z principů. Organizace by měly navrhovat své procesy a služby tak, aby obsahovaly pouze nezbytné kroky k dosažení stanoveného cíle. To znamená, že by se měly vyhýbat nadbytečným a zbytečně komplikovaným krokům, které by mohly zpomalovat proces nebo způsobovat zbytečné náklady. Pro instanci, při návrhu procesu zpracování objednávek by organizace měla eliminovat veškeré nadbytečné kroky a zdokonalit pracovní a technologické postupy tak, aby byl co nejefektivnější.

Zaměření se na výsledky hraje podstatnou roli v zásadě pro jednoduchou a praktickou cestu k výstupům. Řešení by se měla zaměřovat na dosažení konkrétních a měřitelných výsledků, které jsou relevantní pro potřeby organizace a jejích zákazníků. Namísto zbytečných komplikací by organizace měla klást důraz na to, jakým způsobem dané řešení přispívá k dosažení stanovených cílů a jakou hodnotu přináší. Jeden z případů, místo komplikovaných a rozsáhlých projektů by organizace mohla preferovat malé a agilní kroky, které jsou zaměřené na konkrétní vylepšení a dosažení měřitelných úspěchů.

Eliminace neefektivnosti je posledním zásadním prvkem této zásady. Organizace by měly identifikovat a odstranit veškeré procesy a aktivity, které nepřinášejí přidanou hodnotu. To znamená, že by měly být eliminovány veškeré zbytečné činnosti, které zpomalují procesy nebo zvyšují náklady, aniž by přispívaly k dosažení stanovených cílů. Pokud organizace identifikuje procesy, které jsou zastaralé nebo neproduktivní, měla by je nahradit efektivnějšími a účinnějšími alternativami.

7. Optimalizace a automatizace

Optimalizace a automatizace se zaměřuje na využití lidských a technických zdrojů co nejefektivněji za účelem dosažení maximální hodnoty pro organizaci i její zákazníky.

Optimalizace procesů je jedním ze základních pilířů této zásady. Organizace by měly systematicky analyzovat své procesy s cílem identifikovat a odstranit zbytečné kroky a neefektivní postupy. Tímto způsobem mohou zvýšit efektivitu svých operací a dosáhnout lepších výsledků s menší námahou. Pokud organizace zjistí, že určitá fáze

procesu je zbytečně složitá nebo nedostatečně optimalizovaná, může ji zjednodušit nebo automatizovat, čímž zlepší celkový průběh procesu a sníží náklady.

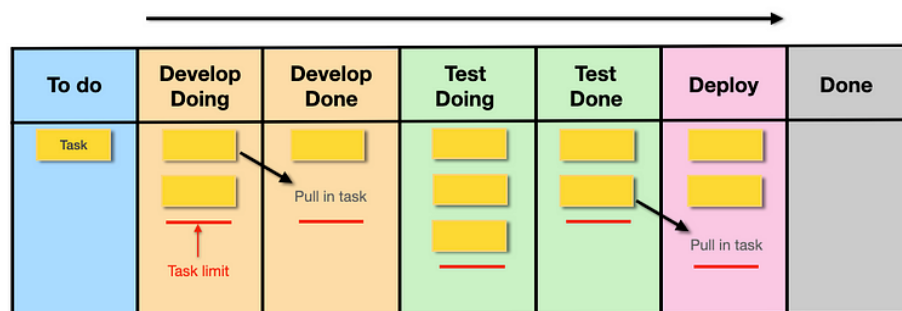
Pro samotnou optimalizaci lze využít postupů ITIL, ale také jiných, jako jsou:

- **Kanban**

Kanban je jako digitální nástěnka rozdělená na sloupce, z nichž každý reprezentuje určitou fázi v pracovním procesu, typické jsou „Ke zpracování“, „Zpracovááno“ a „Hotovo“. Na každém lístku, který může být buď fyzický nebo virtuální, jsou zapsány jednotlivé úkoly nebo úkoly a ty se pohybují mezi sloupci v souladu s tím, jak se na nich pracuje.

Tento systém vizuálně organizuje práci a umožňuje jednotlivým členům týmu vidět, na čem se právě pracuje, a jaké úkoly jsou připraveny k provedení. To pomáhá zajistit, že nikdo není přetížen nebo nepracuje na zbytečných úkolech, a umožňuje lepší spolupráci a plánování.

Díky Kanbanu může tým lépe koordinovat svou práci, snadněji identifikovat případné bloky nebo zpoždění a optimalizovat svůj pracovní postup. Také tento systém pomáhá omezit množství rozpracovaných úkolů najednou, což vede k větší produktivitě a rychlejšímu dokončení projektů.



Obrázek 6: Vizualizace Kanban [12]

[13]

- **Lean**

Lean je metodika, která se zaměřuje na maximalizaci hodnoty pro zákazníka a minimalizaci plýtvání v procesu výroby či poskytování služeb. Na rozdíl od Kanbanu, který se soustředí na organizaci práce, Lean se snaží identifikovat a odstranit veškeré aktivity, které nepřidávají žádnou hodnotu z pohledu zákazníka.

To znamená, že Lean se zaměřuje na zjednodušení procesů a eliminaci překážek, které brání toku hodnoty. Jednou z hlavních myšlenek je princip kontinuálního zlepšování, který podporuje průběžné hledání a odstraňování zbytečných kroků či činností.

Pro představu, v kontextu výroby automobilů by Lean identifikoval a odstranil nadbytečné kroky v procesu výroby, jako jsou zbytečné pohyby pracovníků nebo nadměrná spotřeba materiálů. Tím se sníží náklady a zkrátí se čas potřebný k výrobě automobilu, což umožňuje rychlejší a efektivnější dodávku vozidel zákazníkům.

Dalším případem může být Lean přístup v oblasti služeb. Zde by Lean identifikoval zbytečné činnosti v procesu poskytování služeb a zaměřil by se na jejich eliminaci, což by vedlo k rychlejšímu a efektivnějšímu poskytování služeb zákazníkům.

Principy štíhlé (Lean) výroby se snaží jednoduše vysvětlit, jak maximalizovat hodnotu pro zákazníka a minimalizovat plýtvání v procesu výroby nebo poskytování služeb.

Definice hodnoty

Hodnota je to, za co je zákazník ochoten zaplatit, a co pro něj představuje skutečný užitek. Je důležité nejen se zaměřit na vlastnosti produktu či služby, ale také na to, jak řeší potřeby zákazníka. To lze dosáhnout prostřednictvím průzkumů, rozhovorů a analýzy dat.

Zmapovat hodnotový tok

Tento krok zahrnuje mapování všech kroků, které jsou potřebné k vytvoření produktu nebo poskytnutí služby. Je důležité identifikovat kroky, které nepřidávají hodnotu konečnému produktu a odstranit je. Mezi typické formy plýtvání patří čekání, zbytečné pohyby a nadměrná výroba.

Vytvořit tok

Cílem je vytvořit plynulý tok práce, který minimalizuje zpoždění a úzká místa v procesu. To lze dosáhnout rozdělením složitých úkolů na menší a jednodušší kroky a posílením flexibility zaměstnanců.

Zavedení systému Pull

Namísto masové produkce předem se zaměřte na výrobu pouze toho, co je potřeba, když je to potřeba. Tento systém, známý jako "just-in-time," snižuje plýtvání a zajistí, že zboží zůstane čerstvé.

Usilování o dokonalost

Štíhlost není pouze cíl, ale spíše neustálá cesta ke zlepšení. Je důležité podporovat kulturu neustálého učení a hledání způsobů, jak eliminovat plýtvání a být každý den lepší. Tento proces by měl zapojit všechny zaměstnance ve firmě.

[14]



Obrázek 7: Pět principů metodiky Lean [14]

- **DevOps**

DevOps je moderní přístup k vývoji softwaru, který spojuje vývojáře a IT operátory do jednoho týmu, aby společně řešili výzvy spojené s vývojem, testováním a nasazováním aplikací. Tento koncept se odvíjí od tradice, kdy vývoj a provoz byly odděleny, což často vedlo k problémům s komunikací a koordinací. DevOps se snaží tento rozpor odstranit prostřednictvím integrace a automatizace procesů.

Lze si představit tým pracující na vývoji webové aplikace. Tradičně by vývojáři vytvářeli kód a poté předávali tento kód operátorům, kteří by se postarali o jeho nasazení na produkční server. Tento proces mohl trvat dlouho a mohl být náchylný k chybám při ručním nasazování. S DevOps přístupem by vývojáři a operátoři spolupracovali již od začátku vývoje, aby zajistili, že aplikace je navržena s ohledem na provoz a je snadno nasaditelná. Automatizované nástroje by pak umožnily rychlé a opakované nasazování aplikace bez ručního zásahu.

[15]

Zásada sedm také klade důraz na automatizaci úkolů. Opakující se a rutinní úkoly, které vyžadují malou nebo žádnou lidskou intervenci, by měly být automatizovány. Tím se uvolní lidské zdroje pro komplexnější a strategické úkoly, které vyžadují lidskou kreativitu a rozhodování. Realizace automatizace může být za pomoci automatické generování a odesílání reportů, správa IT infrastruktury pomocí nástrojů pro automatizaci, nebo automatické zpracování a klasifikace dat.

Vyvážený přístup k automatizaci je dalším faktorem pro úspěšnou implementaci této zásady. Automatizace by měla být prováděna s rozvahou, s ohledem na náklady, benefity a zachování lidského dohledu v klíčových oblastech. Je důležité najít správnou rovnováhu mezi automatizací a lidskou interakcí, aby organizace dosáhla maximální efektivity a zachovala kontrolu nad kritickými procesy. I když je automatizace některých úkolů výhodná pro organizaci, je stále důležité zajistit, aby bylo zachováno lidské dohledu v oblastech, které vyžadují lidský rozum a rozhodování, jako je řízení rizik nebo komunikace se zákazníky.

[8 stránky 39-56]

2.5.2 Řízení

V rámci ITIL SVS je řízení klíčovou funkcí, která má za úkol zajistit dosažení strategických cílů organizace a dodržování principů ITIL. Tato funkce zahrnuje celou řadu činností a procesů, které pomáhají organizaci efektivně řídit své operace a dosahovat požadovaných výsledků. Podrobněji se lze podívat na některé z hlavních prvků řízení v rámci ITIL SVS:

- **Hodnocení:** Řízení neustále hodnotí stav organizace a provádí analýzy s ohledem na měnící se potřeby stakeholderů a vnější prostředí. To zahrnuje pravidelné revize

strategií a cílů organizace, aby byla zajištěna relevance a adaptace na aktuální podmínky. Pro ilustraci, pokud se tržní podmínky nebo požadavky zákazníků změní, řízení může provést revizi strategických plánů a priorit, aby organizace zůstala konkurenceschopná a reagovala na nové výzvy.

- **Přímý vliv:** Řízení má také za úkol definovat strategii a politiky organizace a dohlížet na jejich realizaci. Strategie stanovuje směřování a priority organizace v dlouhodobém horizontu, zatímco politiky stanovují pravidla chování a postupy pro všechny stakeholdery. Například, pokud organizace stanoví politiku v oblasti bezpečnosti dat, řízení je odpovědné za zajištění toho, aby tato politika byla implementována a dodržována ve všech částech organizace.
- **Monitorování:** Další důležitou funkcí řízení je pravidelné monitorování výkonnosti organizace, včetně procesů, produktů a služeb. Cílem je zajistit soulad s principy a pokyny ITIL a identifikovat případné nedostatky nebo oblasti, které vyžadují zdokonalení. Pomocí metrik a klíčových ukazatelů výkonnosti (KPIs) může řízení sledovat úspěšnost implementace procesů ITIL a identifikovat oblasti, které vyžadují další opatření či zlepšení.

[8 str. 57]

KPI

Klíčové ukazatele výkonnosti (KPI) jsou klíčovými metrikami používanými v rámci ITIL k měření a hodnocení efektivity procesů a služeb organizace v průběhu času. Tyto KPI pomáhají organizacím dosahovat následujících cílů.

Posouzení celkové výkonnosti IT a jednotlivých služeb: KPI poskytují organizacím kvantitativní a kvalitativní data, která umožňují posoudit, jak efektivně IT infrastruktura a služby podporují potřeby a cíle organizace. Pro instanci, KPI může měřit dostupnost IT služeb, dobu řešení incidentů nebo spokojenost uživatelů s poskytovanými službami.

Identifikace oblastí pro zlepšení: Analyzováním KPI může organizace identifikovat slabé body a oblasti, ve kterých lze procesy a služby zlepšit. Pokud třeba KPI ukazují vysoký průměrný čas řešení incidentů, může to naznačovat potřebu vylepšit procesy řízení incidentů nebo investovat do lepších nástrojů a technologií.

Monitorování trendů a vývoje v čase: KPI umožňují sledovat změny a trendy výkonnosti procesů a služeb v čase. To organizacím umožňuje identifikovat dlouhodobé

trendy a potenciální problémy, které by mohly vyvstat v budoucnosti. Jako demonstrace, pokud KPI ukazují postupné zhoršování dostupnosti IT služeb, může to naznačovat potřebu provést rozsáhlejší revizi infrastruktury nebo strategií správy.

Řízení procesů a služeb na základě dat: KPI poskytují objektivní data, na základě kterých může organizace efektivně řídit své procesy a služby. Tato data umožňují rozhodování založené na faktech a pomáhají organizaci lépe porozumět jejím potřebám a prioritám. Organizace může na základě KPI upravit prioritní seznam změn nebo alokaci zdrojů tak, aby lépe odpovídala aktuálním potřebám a cílům.

[16]

Zástupcem budiž MTTR (Mean Time to Repair) - KPI, který měří průměrný čas potřebný k řešení incidentů od jejich nahlášení až po jejich uzavření. Tento ukazatel je důležitý pro hodnocení efektivity IT týmu v reakci na výpadky a problémy v IT službách. Nižší hodnoty MTTR obvykle znamenají rychlejší a efektivnější reakci na incidenty, což přispívá k menším přerušením v poskytování služeb a vyšší spokojenosti uživatelů.

Pro realizaci řízení jsou v aplikaci, konkrétně pro monitoring, připravené přehledné tabulky pro klíčové oblasti. Finanční stránku vizualizuje analýza prostředků a nákladů vůči vyčleněným penězům pro fázi i projekt. Dalším ukazatelem může být forma histogramu úkolů, v jakém stavu se aktuálně nachází, nebo odhadovaná doba trvání. Pokud jde o hodnocení, jak již bylo zmíněno, aplikace umožňuje upravení cílů, a tak reagovat na potenciální změnu situace.

2.5.3 Hodnotový řetězec služeb

Hodnotový řetězec služeb (Service Value Chain nebo SVC) představuje jádrové uspořádání a provozní model v rámci ITIL SVS, který detailně popisuje klíčové činnosti nutné pro reakci na poptávku a usnadnění realizace hodnoty pro zákazníky a organizaci jako celek.

Tento model je založen na předpokladu, že každá služba prochází určitým řetězcem činností, které přispívají k vytvoření, dodání a správě této služby, formálněji a v terminologii tento sled formuje hodnotový tok. SVC se skládá z jednotlivých článků, které spolu úzce souvisejí a vzájemně se ovlivňují za pomoci vstupů a výstupů.

Mezi klíčové činnosti, které SVC zahrnuje, patří:

- plánování,
- zlepšování,
- zapojení,
- návrh a přechod,
- získat a postavit,
- dodat a podporovat.

Tyto činnosti ke zpracování potřebují své organizační zdroje, které mohou nabývat různých podob. Může se jednat o spolupráci s dalšími organizacemi, dodavateli nebo partnery může poskytnout přístup k dalším zdrojům a znalostem. Organizace může vytvářet partnerství pro sdílení zdrojů, inovaci a vzájemnou podporu při poskytování služeb. Na místě jsou také znalosti, dovednosti a kompetence pracovníků jsou nezbytné pro úspěšné provádění činností. Organizace může investovat do školení a rozvoje svých zaměstnanců nebo najmout externí odborníky s potřebnými dovednostmi a zkušenostmi pro provádění specifických úkolů. Nadále se jedná o používání speciálních nástrojů a technologií může být klíčové pro efektivní provádění činností. To může zahrnovat automatizaci procesů pomocí softwarových nástrojů, správu služeb pomocí ITSM (IT Service Management) platformy nebo využití analytických nástrojů pro monitorování výkonnosti a identifikaci oblastí pro zlepšení. V poslední řadě to nejvíce základní, což jest organizační zdroje a procesy v podobě vlastních týmů, oddělení a procesy k provádění různých činností. To může zahrnovat interní týmy IT specialistů, správu projektů, kanceláře služeb, nebo jiná oddělení zodpovědná za poskytování služeb. Interní zdroje mají často hlubší znalosti o prostředí organizace a mohou být lépe integrovány do jejích procesů.

Plánování

Plánování je důležitá činnost v rámci hodnotového řetězce služeb (SVC) ITIL, která pomáhá organizacím stanovit, kam směřují a jak dosáhnou svých cílů. Při plánování se berou v úvahu různé faktory, jako jsou potřeby zákazníků, výkonnost služeb, nové produkty a služby, ale i požadavky na dodržování předpisů a spolupráce s externími dodavateli služeb. Výstupem plánování jsou strategické, taktické a operativní plány, které organizaci pomáhají orientovat se a rozhodovat o tom, které produkty a služby budou nabízeny, jak budou implementovány a jak budou řízeny. Tento proces také zahrnuje stanovení architektury a politik pro nové služby a produkty a určení oblastí, které je třeba zlepšit. Důležitým aspektem je také sjednání podmínek

s externími dodavateli služeb, pokud jsou zapojeni. Celkově plánování umožňuje organizaci dobře se připravit na budoucnost a zajistit, že veškeré aktivity směřují k dosažení stanovených cílů a potřeb zákazníků.

Zlepšování

Činnost „Zlepšování“ hodnotového řetězce služeb je klíčová pro kontinuální zdokonalování produktů, služeb a procesů v organizaci. Tato činnost je zaměřena na posun vpřed ve všech oblastech poskytování služeb a ve všech aspektech řízení služeb. K dosažení tohoto cíle je nutné mít k dispozici různé informace.

Začíná se přehledem o výkonnosti produktů a služeb, aby vznikl jasný obraz o tom, jak to vypadá ve srovnání s očekáváními a standardy. Dále je potřeba znát názory zákazníků, aby šlo zjistit, co je jim důležité a co by chtěli vidět zlepšené. To poskytuje užitečný náhled na oblasti, ve kterých je třeba se zlepšit. Důležité jsou také informace o tom, jak probíhají interní procesy a jak efektivní jsou. Tyto poznatky pomáhají identifikovat možnosti pro efektivnější postupy a odstranění případných překážek.

V rámci zlepšování se berou v potaz i novinky a změny, které se v organizaci dějí. To zahrnuje nové technologie, produkty nebo služby, které mohou přinést nové příležitosti k inovaci. Důležité jsou také informace o výkonnosti služeb poskytovaných třetími stranami, pokud jsou zapojeny, aby se posoudil jejich přínos a případné zlepšení.

Na základě těchto informací pak lze plánovat konkrétní kroky ke zlepšení. To může zahrnovat aktualizaci procesů, inovace produktů nebo služeb, ať už na základě interních nebo externích zpětných vazeb. Dále se zaměřuje na monitorování a sledování pokroku v implementaci těchto zlepšení, aby se zajistilo, jestli se postupuje správnou cestou. Celkově je cílem činnosti zlepšování neustále se vyvíjet a přizpůsobovat se měnícím potřebám a požadavkům, což umožňuje poskytovat stále lepší hodnotu zákazníkům a zlepšovat efektivitu organizace.

Zapojení

Činnost hraje klíčovou roli v budování důvěry a efektivní komunikaci mezi organizací a všemi zúčastněnými stranami. Cílem této činnosti je zajistit:

- **Dobré porozumění:** To znamená, že organizace musí pečlivě poslouchat potřeby a očekávání všech, kteří jsou dotčeni poskytovanými službami. Pro představu, pokud

organizace poskytuje softwarové služby, musí být citlivá na potřeby uživatelů těchto softwarových produktů.

- **Transparentnost:** Organizace by měly být otevřené a transparentní ohledně poskytovaných služeb a produktů. To zahrnuje poskytování jasných informací o funkcionalitě, cenách a podmínkách používání služeb.
- **Neustálé zapojení:** Je důležité, aby všechny strany byly neustále zapojeny do procesu tvorby a zlepšování služeb. To znamená aktivně naslouchat zpětné vazbě od zákazníků a uživatelů a zapojovat je do rozhodovacích procesů.
- **Dobré vztahy:** Je klíčové budovat a udržovat dobré vztahy se všemi stakeholdery, včetně zákazníků, partnerů a dodavatelů. Silné vztahy mohou vést k dlouhodobé spolupráci a vzájemnému prospěchu.

Klíčovými vstupy pro činnost jsou různé informace, které organizaci pomáhají lépe porozumět potřebám a očekáváním zúčastněných stran. To zahrnuje zpětnou vazbu od zákazníků, informace o produktech a službách třetích stran, požadavky na služby a produkty od uživatelů a mnoho dalšího.

Na základě těchto vstupů organizace může plánovat další kroky a činnosti. To může zahrnovat vytváření nových produktů a služeb, řešení stížností uživatelů, zlepšování procesů nebo jednání o nových smlouvách a dohodách s dodavateli.

Klíčovými výstupy jsou konkrétní akce a dokumenty, které vyplývají z interakce s různými zainteresovanými stranami. Tyto výstupy mohou zahrnovat požadavky na změny, úkoly pro týmy odpovědné za poskytování služeb, a smlouvy s externími dodavateli. Důležitým prvkem jsou také zprávy o výkonu služeb, které umožňují organizaci zhodnotit, jak dobře plní své závazky vůči zákazníkům.

Návrh a přechod

Tato činnost v ITIL zajišťuje, že nové nebo změněné produkty a služby odpovídají potřebám zákazníků a jsou kvalitní, cenově dostupné a uváděny na trh včas. To zahrnuje vytváření plánů a specifikací pro vývoj, uzavírání smluv s dodavateli a monitorování výkonnosti. Organizace sbírá informace od různých zdrojů, jako jsou požadavky zákazníků, výkonnostní data a smlouvy. Poté podle těchto informací plánuje a provádí kroky potřebné k vývoji a uvedení nových produktů a služeb na trh. Na konci této činnosti jsou definovány požadavky na produkty, uzavřeny smlouvy s dodavateli a představeny nové nebo změněné produkty a služby.

Získat a postavit

Činnost „Získat a postavit“ má za cíl zajistit, že komponenty služby jsou k dispozici včas a na správném místě a splňují dohodnuté specifikace. To zahrnuje přijímání produktů a služeb od externích i interních dodavatelů a partnerů, vývoj vlastních komponent služeb a jejich zlepšování. Organizace přijímá vstupy, jako jsou architektonické plány, smlouvy a dohody s dodavateli, specifikace a požadavky, a na základě těchto informací provádí akce potřebné k získání nebo vytvoření komponent služeb. Na konci této činnosti jsou k dispozici potřebné komponenty služeb pro další fáze v hodnotovém řetězci, a to jak pro poskytování služeb zákazníkům, tak pro interní použití.

Dodat a podporovat

Zde se hodnotový řetězec soustředí na zajištění dodávání služeb v souladu s dohodnutými specifikacemi a poskytování podpory, která splňuje očekávání stakeholderů. To znamená, že organizace musí zajistit, že služby jsou doručovány včas a ve správné kvalitě, aby uspokojily potřeby zákazníků a uživatelů. Klíčové vstupy pro tuto činnost zahrnují nové a změněné produkty a služby, které vznikly v předchozích fázích, stejně jako informace o možnostech zlepšení a plnění úkolů uživatelské podpory. Výstupy z této činnosti zahrnují samotné poskytování služeb zákazníkům a uživatelům, informace o výkonnosti produktů a služeb pro další zlepšení, žádosti o změny a požadavky na smlouvy a dohody pro další spolupráci.

[8 stránky 61-65]

Je nutné si uvědomit, že prvky, ať už jde o praktiky nebo třeba zmíněné činnosti, dominují vysokou úrovní abstrakce tak, aby pokryly většinou část případů s holistickým přístupem. Konkrétní výklady a implementace budou vždy odlišné s danou organizací. Výstupní aplikace je proto také vystavěna pro vyšší obecnost užití. Důležitým aspektem je nezapomínat na čtyři dimenze, přes které se na celý SVS koukáme. Tato podpora v rámci aplikace nechybí třeba v podobě rolí a přístupů. Základní funkce pro tvorbu samotného toku je realizována možnostmi pro definici specifikací produktu v činnosti plánování, zapojení stran (pozvání do projektu), návrh řešení přes úkoly a fáze, i za pomoci nahrávání souborů, předání požadavků vývojovému týmu a paralelní zpracování úkolů, doplnit hotové výstupy a neustálá možnost zpětné vazby k dílčím instancím zúčastněných stran a průběžné monitorování stavu projektu pro další zlepšování.

2.5.4 Praktiky

Řídicí postupy v ITIL jsou jakýmsi průvodcem pro organizace, jak lépe řídit své IT služby. Jsou to jakoby návody, které říkají, jak postupovat, aby vše fungovalo co nejlépe. Existují různé druhy těchto postupů. Některé jsou obecné a pomáhají v plánování, financích a řízení projektů. Jiné jsou zaměřené přímo na správu IT služeb, jako řešení problémů nebo změn. A pak tu jsou ještě ty, které se starají o technickou stránku, jako je udržování hardwaru a softwaru. Důležité je, aby se všechny tyto postupy vzájemně doplňovaly a aby byly upraveny tak, aby co nejlépe vyhovovaly potřebám konkrétní organizace. Používání těchto postupů pomáhá organizacím zlepšit kvalitu poskytovaných služeb, udržet náklady pod kontrolou a zajistit spokojenost uživatelů.

Tabulka 2: Praktiky řízení ITIL [8 str. 76]

Obecné postupy řízení (14)	Postupy řízení služeb (17)	Postupy technického řízení (3)
Řízení architektury	Řízení dostupnosti	Správa nasazení
Neustálé zlepšování	Obchodní analýza	Správa infrastruktury a platformy
Řízení znalostí	Řízení kapacity a výkonnosti	Vývoj a správa softwaru
Řízení bezpečnosti informací	Řízení změn	
Měření a podávání zpráv	Správa incidentů	
Řízení organizačních změn	Správa IT majetku	
Řízení portfolia	Monitorování a správa událostí	
Řízení projektů	Správa problémů	
Řízení rizik	Řízení vydání	
Řízení vztahů	Správa katalogu služeb	
Finanční řízení služeb	Správa konfigurace služeb	
Řízení strategie	Správa kontinuity služeb	
Řízení dodavatelů	Návrh služeb	
Řízení pracovní síly a talentů	Service desk	
	Správa úrovně služeb	
	Správa požadavků na služby	
	Validace a testování služeb	

Není nutné zavádět všechny praktiky do procesů organizace. Všechny praktiky a SVS samotný podléhají obecné pružnosti pro vlastní potřeby. To, avšak nikterak nevyhovává o jejich

samotné důležitosti. Pro daný účel může být relevantnější jedna nad druhou. Pro potřebu řízení informačního projektu jsou autorem vybrané pouze ty nezákladnější.

- **Správa požadavků na služby**

Správa požadavků na služby v ITIL 4 je stěžejním prvkem, který organizacím pomáhá efektivně reagovat na potřeby uživatelů ohledně poskytovaných služeb. Tato praktika zahrnuje několik důležitých kroků, včetně sběru, analýzy a řízení požadavků. Zástupcem pro přístup k procesu řízení je metodika Scrum, jež je moderním a rozsáhlým zástupcem v agilní rovině. Z hlediska rigorózního rozhodnutí vedení projektu je praktika obzvlášť důležitá a podcenění této vstupní fáze může vyvolat problémy až do dokončení práce, jestli vůbec.

Prvním krokem je sběr požadavků, kde uživatelé mohou své potřeby sdělit prostřednictvím různých kanálů, jako jsou online formuláře, helpdesk nebo e-mail. Následuje analýza a hodnocení požadavků, kdy jsou tyto požadavky pečlivě posuzovány a zjišťuje se jejich důležitost a naléhavost. Důležitým aspektem je také schválení požadavků, kde autorizované osoby rozhodují o jejich realizaci. Poté následuje řízení požadavků, kde jsou požadavky sledovány a řešeny až do jejich uzavření.

Pro úspěšnou implementaci Správy požadavků na služby je důležité mít vhodný systém pro sběr a správu požadavků, který umožní uživatelům snadno sdělit své potřeby. To může zahrnovat vytvoření online formulářů nebo „helpdeskových“ systémů. V dnešní digitální době se nabízí formát softwarového produktu, nicméně klasická tužka a papír umožní důstojně zastoupit realizaci praktiky, jelikož hlavním faktorem úspěšnosti je důslednost. Kromě toho je klíčové mít jasně definované procesy pro analýzu a hodnocení požadavků, aby bylo možné určit jejich prioritu a začlenit je do plánu dodávek služeb. Takový přístup zajišťuje, že požadavky uživatelů jsou správně pochopeny a adekvátně adresovány.

- **Neustálé zlepšování**

Praktika neustálého zlepšování v rámci ITIL 4 je faktorem pro trvalé a systematické zkvalitňování produktů, služeb a procesů v IT organizaci. Cílem této praktiky je zajistit, aby fungování IT bylo neustále přizpůsobováno měnícím se potřebám podniku, což je zásadní pro dosažení trvalého úspěchu.

Principy této praktiky zahrnují aktivní zapojení vedení organizace, které má za úkol podporovat a propagovat kulturu neustálého zlepšování. Dále je důležité vyvíjet a implementovat vhodné metody a techniky pro identifikaci, analýzu a implementaci zlepšení a zajistit, aby praktika neustálého zlepšování byla integrována do celkové strategie organizace. Závazek k neustálému zlepšování musí být zakořeněn ve všech oblastech a na všech úrovních organizace, což vyžaduje vyčlenění adekvátního času a finančních prostředků.

Činnosti spojené s touto praktikou zahrnují aktivní podporu neustálého zlepšování ze strany vedení organizace, zajištění dostatečných zdrojů v podobě času a financí pro aktivity v oblasti zlepšování, systematickou identifikaci příležitostí pro zlepšení prostřednictvím shromažďování dat a zpětné vazby, analýzu a prioritizaci identifikovaných oblastí zlepšení, vytváření obchodních případů pro investice do klíčových zlepšení, plánování a efektivní implementaci těchto zlepšení, sledování a analýzu výsledků změn a koordinaci aktivit v rámci celé organizace pro dosažení maximálního efektu.

Základem trvalého zlepšování je cyklus reflexe, implementace a hodnocení. Pravidelná retrospektivní setkání po dokončení iterace či sprintu slouží k reflexi nad provedenou prací, identifikaci oblastí pro zlepšení a nalezení inovativních řešení. Implementace nástrojů pro sběr zpětné vazby, jako jsou průzkumy spokojenosti nebo online formuláře, umožňuje shromažďovat cenné informace od uživatelů a stakeholderů.

Pro podporu inovací je důležité vytvořit prostředí, ve kterém se snadno sdílí nápady a diskutuje o možných zlepšeních. Toho lze dosáhnout například vytvořením online platformy pro sdílení myšlenek a návrhů.

Identifikované příležitosti ke zlepšení je nutné zhodnotit a prioritizovat podle jejich dopadu a náročnosti implementace. Následně se vytvoří plán s přidělením potřebných zdrojů.

Průběžné měření a vyhodnocování implementovaných změn a jejich dopadu na projekt je klíčové pro identifikaci úspěšných strategií a případnou úpravu plánů v oblastech s menší efektivitou.

- **Správa incidentů**

Správa incidentů v představuje praktiku zaměřenou na efektivní a včasnou reakci na incidenty s cílem minimalizovat jejich dopad na uživatele a kontinuitu provozu.

Každý incident je pečlivě zdokumentován s podrobným popisem, zahrnujícím informace o jeho dopadu na uživatele a prioritě pro další řešení. Pro usnadnění prioritizace a efektivního řešení jsou incidenty dále kategorizovány podle typu, závažnosti a naléhavosti.

Pro každý typ incidentu jsou stanoveny cílové doby řešení, které odpovídají očekáváním uživatelů a závažnosti daného incidentu. Tyto cílové doby jsou transparentně sdíleny s uživateli a stakeholdery, čímž se zajišťuje realističnost očekávání a budování důvěry.

Incidenty jsou prioritizovány na základě jejich dopadu na business a naléhavosti jejich řešení. Pro prioritizaci se využívají předem definovaná kritéria a klasifikace, které umožňují správně určit, které incidenty je nutné řešit jako první a alokovat jim adekvátní zdroje.

V případě, že incident nelze vyřešit v rámci stanovené cílové doby, je eskalován na vyšší úroveň podpory. Proces eskalace je jasně definovaný a transparentní pro všechny zúčastněné strany, čímž se zajišťuje rychlá a účinná reakce na vážné incidenty a minimalizace jejich dopadů.

Uživatelé jsou průběžně informováni o stavu incidentu a o plánovaných krocích k jeho vyřešení. Komunikace s uživateli je jasná, transparentní a včasná. To pomáhá udržovat důvěru uživatelů v poskytované IT služby, zvyšuje jejich spokojenost a podporuje celkovou efektivitu řešení incidentů.

- **Řízení změn**

Řízení změn je komplexním přístupem k organizaci a optimalizaci procesů transformace v IT prostředí. Jeho hlavním cílem je zajistit, aby veškeré změny, ať už se týkají infrastruktury, aplikací, dokumentace, procesů, vztahů s dodavateli nebo jiných aspektů, probíhaly efektivně a s maximálním prospěchem pro organizaci a uživatele.

Oproti řízení organizačních změn, které se soustředí na lidský faktor a podporu zaměstnanců při adaptaci na nové prostředí, se řízení změn v ITIL zaměřuje na technické aspekty a zajišťuje plynulý průběh implementací s minimalizací rizik.

Důležitým principem je nalezení rovnováhy mezi inovací a ochranou uživatelů. Všechny změny jsou pečlivě posuzovány z hlediska rizika a přínosu a teprve po schválení autorizovanými osobami jsou implementovány do produkčního prostředí.

Efektivní řízení změn v ITIL zahrnuje několik klíčových prvků. Zapojení stakeholderů zajišťuje aktivní účast všech dotčených stran při plánování a implementaci změn. Důležitá je také jasná a transparentní komunikace o všech aspektech změn a jejich dopadu na uživatele. Řízení rizik zahrnuje identifikaci, analýzu a minimalizaci rizik spojených s implementací změn. Před nasazením do produkce je důležité důkladné testování změn v testovacím prostředí a jejich schválení. Monitorování a vyhodnocování změn umožňuje průběžně sledovat jejich dopad a efektivitu, s možností následných úprav pro optimalizaci procesu.

[17 str. 223]

Změny můžou spadat do následujících kategorií: standartní (známé, bez potřeby vyššího zásahu a autorizace), běžné (k naplánování), nouzové (urgentní).

- **Měření a podávání zpráv**

Praktika ITIL Měření a vykazování je zaměřena na poskytování relevantních informací, které jsou nezbytné pro informované rozhodování a trvalé zlepšování. Cílem této praktiky je snížit míru nejistoty a poskytnout organizacím data o klíčových aspektech IT, aby mohly efektivně řídit a optimalizovat své procesy.

Měření a vykazování neznámá pouze sběr číselných údajů, ale i jejich interpretaci a prezentaci tak, aby poskytovaly smysluplný kontext pro rozhodování. Organizace by měly identifikovat relevantní ukazatele výkonnosti (KPIs), které odrážejí klíčové aspekty fungování IT služeb a procesů. Ty mohou měřit dobu, kterou trvá vyřízení incidentů, míru dostupnosti IT systémů nebo úroveň spokojenosti uživatelů.

- **Finanční řízení služeb**

Finanční řízení služeb má za úkol nejen podporovat, ale i usnadnit informované rozhodování managementu a vedení organizace v otázkách alokace finančních zdrojů. To zahrnuje jak rozpočtování a kalkulaci nákladů, tak i účetní činnosti související s produkty a službami IT.

Principy spočívají v jeho integraci s politikami a procesy organizace. To znamená, že praktika musí být pečlivě sladěna s ostatními aspekty řízení portfolia, projektů a vztahů. Důležitou roli hraje také srozumitelná komunikace. Finance totiž slouží jako společný jazyk, který umožňuje efektivní komunikaci se všemi stakeholdery.

Co se týče konkrétních odpovědností, ty zahrnují: rozpočtování, účetnictví a účtování.

- **Řízení dodavatelů**

Praktika Řízení dodavatelů se zaměřuje se na efektivní správu vztahů s externími dodavateli. Cílem této praxe je zajistit, aby dodavatelé poskytovali kvalitní produkty a služby v souladu s potřebami a očekáváními organizace.

Pro dosažení tohoto cíle je důležité budovat a udržovat silné a důvěryhodné vztahy s dodavateli. To zahrnuje pravidelnou komunikaci, vzájemnou spolupráci a transparentnost ve vztazích. Organizace by měla aktivně pracovat na budování důvěry a porozumění mezi sebou a svými dodavateli, což může vést k vytvoření dlouhodobých partnerských vztahů.

- **Validace a testování služeb**

Testování hraje nezbytnou roli v zajištění kvality IT služeb a produktů. Pomáhá odhalit chyby a nedostatky, čímž umožňuje jejich včasnou opravu a prevenci před negativním dopadem na uživatele.

Testování se řadí do následujících kategorií:

- Funkční: jednotkové, systémové, integrační a regresivní testy.
- Nefunkční: výkonostní, bezpečnostní, provozní, akceptační a záruční testy.

- **Řízení vydání**

V ITIL se termín "verze" používá k identifikaci a organizaci změn implementovaných do IT prostředí s cílem poskytnout nové funkce nebo vylepšení. Verze může obsahovat různé komponenty, které mohou zahrnovat infrastrukturu, aplikace, dokumentaci, školení, procesy, nástroje a další součásti IT služeb.

Když se hovoří o nové verzi softwaru, ta může obsahovat aktualizace nebo opravy chyb, nové funkce nebo vylepšení v uživatelském rozhraní. Pokud jde o infrastrukturu, nová

verze může zahrnovat aktualizace hardwaru nebo softwaru, které zlepšují výkon nebo bezpečnost systému.

- **Vývoj a správa softwaru**

Vývoj a správa softwaru má za cíl zajistit, že softwarové aplikace splňují požadavky všech zúčastněných stran, včetně interních uživatelů organizace a externích klientů a partnerů. Klíčovými faktory jsou funkčnost, spolehlivost, udržitelnost, shoda s normami a auditovatelnost softwaru.

V dnešních IT organizacích jsou softwarové aplikace klíčové pro poskytování hodnoty zákazníkům. Bez ohledu na to, zda se jedná o vývoj interně nebo ve spolupráci s partnery a dodavateli, efektivní vývoj a správa softwaru jsou nezbytné pro to, aby aplikace splňovaly svůj účel a přispívaly k dosahování strategických cílů.

Hlavní činnosti v této praxi zahrnují:

- Architektura řešení: Navrhování celkové struktury a koncepce softwarového systému.
- Návrh řešení: Detailní specifikace uživatelského rozhraní, uživatelské zkušenosti, designu služeb a dalších aspektů.
- Vývoj softwaru: Implementace softwarové aplikace podle definovaných specifikací.
- Testování softwaru: Ověření funkčnosti, spolehlivosti a bezpečnosti softwaru na různých úrovních, včetně jednotkového testování, integračního testování a uživatelského akceptačního testování.
- Správa depozitářů: Udržování integrity kódu a souvisejících artefaktů v centrálních depozitářích.
- Vytváření balíčků: Příprava softwaru pro efektivní a bezproblémové nasazení do produkčního prostředí.
- Správa verzí: Kontrola a koordinace změn v kódu a dokumentaci.
- Průběžná správa: Udržování a optimalizace softwaru v průběhu jeho životního cyklu, včetně aktualizací a úprav na základě potřeb uživatelů.

[8 stránky 75-165]

2.5.5 Neustálé zlepšování

Model neustálého zlepšování ITIL SVS je jako průvodce pro organizace v oblasti IT, který je provází na jejich cestě k trvalému úspěchu. Tento model se skládá ze sedmi klíčových kroků, které organizaci pomáhají pochopit, kde se nachází, kam směřuje a jak se tam dostane.

Prvním krokem je definování vize. Organizace si musí jasně vytyčit cíl, kterým chce dosáhnout, a představit si, jak by mělo IT vypadat a fungovat v budoucnosti. Například může mít za cíl zefektivnit IT procesy nebo zvýšit spokojenost uživatelů.

Druhým krokem je analýza současného stavu, kde organizace zhodnotí své aktuální postavení. To může zahrnovat měření klíčových ukazatelů výkonnosti, sběr zpětné vazby od stakeholderů nebo mapování procesů.

Třetím krokem je stanovení cílového stavu, kde organizace vytvoří specifické, měřitelné, dosažitelné, relevantní a časově ohraničené cíle (SMART). Tyto cíle slouží jako směrnice pro budoucí akce.

Čtvrtým krokem je vytvoření akčního plánu, který popisuje konkrétní kroky, které je potřeba podniknout k dosažení stanovených cílů. Tento plán může zahrnovat implementaci nových procesů, školení uživatelů nebo investování do nových technologií.

Pátým krokem je samotná implementace akčního plánu, kdy organizace provádí plánované změny a sleduje jejich pokrok.

Šestým krokem je pravidelné měření a vyhodnocování výsledků, aby organizace mohla sledovat svůj pokrok a zjistit, zda dosahuje stanovených cílů.

Sedmým a posledním krokem je udržování dosažených výsledků a neustálé hledání nových a inovativních způsobů, jak dále zlepšovat IT procesy a služby. Tento proces je cyklický, což znamená, že organizace se neustále vrací k prvním krokům a opakuje je, aby udržely tempo a zůstaly konkurenceschopné na trhu.

[8 stránky 66-73]



Obrázek 8: Model neustálého zlepšování [8 str. 66]

Tento kruhový přístup lze uplatnit v informačních projektech, v jeho vývoji a udržitelnosti. Samotná výstupní aplikace nezaručí dodržování postupu, ve většině případů bude vždy a pouze záviset na lidech, jak moc zodpovědný postoj vedení převezme. Aplikace ale tento přístup umí evidovat, ať už se uživatelé rozhodnou vést projektové řízení pečlivě, nebo ne.

3 Analýza konkurenčních řešení webové aplikace

Internet charakterizuje ohromná volatilita, dynamika v neustálých změnách, poskytované služby mizí ze scén a zanikají v hlubinách zapomnění, na druhé straně se každý den, každou chvíli valí do veřejného digitálního prostoru nové a nové služby, které se derou do povědomí lidí za pomoci velkých investic do marketingu, jenž má zajistit vstupenku do úspěšného virtuálního businessu. Konkurence v prostředí bez limitů nabývá velkého množství, a tak kvalitní projekty, byť se značným potenciálem, mohou jednoduše bez dravého přístupu propadnout do stínu větších služeb a značek. Služby popsané v následujících řádcích proto zdaleka nejsou všechny, jen průřez těch, které se uchytily a které se ve větší komunitě používají.

3.1 YouTrack

Řešení YouTrack vytvořila velká společnost JetBrains zaměřující se na projekty pro vývojáře, typicky se jedná o vývojová prostředí pro různé programovací jazyky (PHP, Java, C, C# ...) s širokou škálou nástrojů využívající i umělou inteligenci pro podporu psaní kódu a analýzu. Moderním trendem je vytvářet aplikace v cloudovém provedení přes přístup webového prohlížeče. Ani zde JetBrains nevybočil, a tak vytvořil interaktivní, dynamickou stránku-aplikaci pro řízení informačních projektů, která se pohybuje na trhu již od roku 2009. Chytré telefony nebyly ochuzeny o ani o mobilní verzi aplikace.

[18]

YouTrack, jako komplexní nástroj pro správu projektů a úkolů, nabízí širokou škálu funkcí a možností, které zvyšují efektivitu práce a zlepšují komunikaci v týmu. Jednou ze zásadních vlastností je sledování problémů, kde vývojáři mohou rychle a efektivně řešit různé problémy. Lze využít klávesových zkratk a příkazů pro změnu stavů problémů či zodpovědných osob.

Intuitivní vyhledávání je další důležitou funkcí, která umožňuje snadno najít požadované problémy díky chytrému vyhledávání s automatickým doplňováním. Tato funkce zrychluje proces identifikace a řešení problémů.

Aplikace také nabízí integraci s různými vývojářskými prostředími, což umožňuje pracovat s problémy přímo z preferovaného IDE, což minimalizuje nutnost přepínání mezi aplikacemi a zlepšuje plynulost práce.

V oblasti správy úkolů poskytuje YouTrack detailní popis úkolů pomocí pokročilého editoru. Tento editor umožňuje vytvářet podrobné popisy problémů, což přispívá k jasné

definici jednotlivých úkolů. Dále můžete vytvářet vlastní pole s konfigurovatelnou validací vstupních dat a využívat sdílené uložené filtry a hromadné úpravy.

V rámci spolupráce v týmu nabízí YouTrack agilní nástěnky pro vizuální přehled o pracovním postupu a zmínky a reakce, které usnadňují komunikaci a sdílení zpětné vazby mezi členy týmu.

Dále aplikace disponuje databází znalostí, která funguje jako úložiště ve stylu wiki, což umožňuje vytvářet, sdílet a diskutovat obsah týkající se projektů a řešení problémů.

Funkce Helpdesk Projects umožňuje vytvořit servisní prostor pro interní uživatele nebo podporu pro externí zákazníky, což zlepšuje poskytovanou podporu a udržuje efektivní komunikaci s uživateli.

YouTrack dále nabízí možnost automatizace opakujících se úloh pomocí přednastavených i vlastních pracovních postupů, což vede k efektivnější práci a úspoře času.

Vizuální přehled a reporting jsou další důležité aspekty aplikace. Nástěnky poskytují přehledný a vizuálně atraktivní souhrn důležitých informací o projektu, zatímco interaktivní Ganttovy diagramy umožňují sledovat časový harmonogram projektu a identifikovat případné problémy s termíny.

Kromě toho YouTrack umožňuje vestavěné sledování času a poskytuje širokou škálu reportů k analýze výkonu a pokroku v projektu. Lze také integrovat s dalšími nástroji od JetBrains i s nástroji třetích stran, což zvyšuje flexibilitu a možnosti vývojářů.

Známou a rozšířenou alternativou pro tento projekt může být Jira. Oba nástroje poskytují placený i neplacený program.

Mezi nevýhody může patřit rychlost prostředí a omezenost otevřené integrace vůči dalším nástrojům. Občasné neintuitivní ovládání vyvolává zmatky při práci s nástrojem, ale takové označení se může lišit se subjektivním názorem každého uživatele.

[19]

3.2 Trello

Trello je oblíbený nástroj pro řízení projektů, který se vyznačuje svou jednoduchostí a vizuálním přístupem k organizaci úkolů a spolupráci v týmu. Jeho základními stavebními kameny jsou nástěnky a karty, které umožňují strukturovat projekty a sledovat postup

pracovních úkolů. Opět jako jiné aplikace, Trello realizuje svoji službu přes webové rozhraní, ale i v mobilní verzi.

Nástěnky si lze představit jako virtuální tabule, na kterých se organizuje práce. Uživatelé mohou vytvářet nové nástěnky přímo z hlavní nabídky nebo z oddílu "Nástěnky" a přiřazovat jim názvy odpovídající konkrétním projektům nebo úkolům. Dále mohou do těchto nástěnek přidávat sloupce, které představují různé fáze pracovního procesu. Například sloupce „Nezapočato“, „Rozpracováno“ a „Dokončeno“ pomáhají organizovat úkoly podle jejich aktuálního stavu. Uživatelé mohou sloupce také přizpůsobovat specifickými termíny a daty, což usnadňuje plánování a monitorování časových lhůt. V kostce jde o implementaci přístupu Kanban.

Dalším klíčovým prvkem jsou karty, které reprezentují jednotlivé úkoly a nápady. Tyto karty umožňují uživatelům detailně popsat úkoly, přidávat k nim soubory, obrázky, checklisty a další informace, což usnadňuje spolupráci a sdílení potřebných informací v týmu. Dále lze karty filtrovat pomocí barevných štítků a nastavovat jim termíny, což pomáhá organizovat a prioritizovat úkoly. Nechybí ani podpora klávesových zkratk.

Trello podporuje integraci s dalším nástroji (Slack, Jira ...). Vnitřní obchod nabízí spousty „Power-Ups“ rozšíření funkcionality a lze si vybrat z široké škály dle svých potřeb.

Služba je založena na jednoduchém přístupu. Aplikace nedomnuje velkou škálou funkcí, ale to tak bylo a je zamýšleno. V tom spočívá největší přínos a zároveň úskalí platformy. Trello se tak více než aplikace pro řízení projektů označuje za nástroj podporující části projektového řízení.

[20]

3.3 Any.do

Any.do je alternativním řešením služby Trello. Platforma se stala významným hráčem mezi aplikacemi pro správu úkolů, který poskytuje uživatelům efektivní prostředek k organizaci jejich závazků, ať už se jedná o soukromé či pracovní povinnosti. Dominantou se stalo opět jednoduché ovládání a k němu příslušná funkcionality tak, aby uživatel nebyl zahlcen zbytečnými prvky a nepřehledným prostředím.

Vytváření a správa úkolů jsou nejdůležitějšími aspekty Any.do. Po registraci lze okamžitě začít s vytvářením úkolů. Stačí kliknout na tlačítko "Vytvořit úkol" na hlavní obrazovce

aplikace a jednoduše napsat, co je třeba udělat. Tento proces je rychlý a intuitivní, což uživatelům usnadňuje začlenění nových úkolů do svého plánu. Nechybí také kategorizace, přílohy a veškeré pokročilé možnosti specifikace. Zajímavou funkcí je rozdělení úkolů hierarchicky na dílčí podúkoly. Integrace s jinými aplikacemi je samozřejmostí.

Moderní aplikace pro řízení projektů dost často drží obdobný přístup. Některé berou v potaz širší aspekty kontextu projektu, některé se zas specifikují na jednu zásadní oblast tak, aby uživatelé v této rovině nepotřebovali nic víc, než je třeba díky optimalizacím. Opakujícím se klíčovým prvkem je organizace práce a pracovních postupů. Ten se realizuje především přes seznamy a nástěnky. Další rozšiřující základní dimenzí je čas a jeho řízení. Dodržování lhůt, využití kalendářů, návazností, integrací, grafů ... je primární fundament ve struktuře projektu z hlediska času. Úkoly se sdílejí napříč zúčastněnými stranami. Řízení týmů a lidí se stalo nedílnou součástí projektů společně s jejich začleněním do dílčích částí a možností neustále interakce. Jako další důležitý aspekt lze zmínit řízení zdrojů, nicméně to se v aplikacích vyskytuje již méně. Obecně je při tvorbě projektových nástrojů cílem zpřehlednit procesy, není tedy žádoucí tvořit složité nástroje, ale jednoduchá prostředí s intuitivními ovládacími prvky. Účastníci projektu tak mají jasný přehled o tom, co se děje prostřednictvím „dashboardů“, oznámení o změnách a čitelných a strukturovaných informací, jež se jich dotýkají v řetězci procesů tvorby hodnot.

[21]

3.4 Nextcloud

Nextcloud se sám sice neprezentuje jako nástroj projektového řízení, ale z osobní zkušenosti autora se jedná o velice flexibilní řešení, které se může velice příjemně sestavit pro holistický přístup k projektům. Produkt primárně poskytuje prostředí pro pokročilé sdílení souborů a dat. Jeho provoz není omezen platformou, produkt pouze vyžaduje databázovou instanci. Velice rychlé nasazení se může provést přes podporovanou kontejnerovou virtualizaci nástrojem Docker. Produkt samotný je dostupný pod otevřenou licencí a z finančního hlediska je k dostání zdarma. Nextcloud tak lze provozovat na vlastní infrastruktuře, tak i u partnerů projektu za poplatek pro bezstarostné řízení serverového provozu.

Nextcloud v základu nabízí, jak bylo zmíněno, vlastní prostor pro sdílení souborů. Software disponuje i aplikacemi pro mobilní či desktopové zařízení, přes které lze pohodlně synchronizovat práci a přistupovat k dalším funkcím. Aplikace se připojí na soukromí server

přes URL adresu a přístupové údaje, jinak primární ovládání probíhá přes webový prohlížeč. Nextcloud disponuje velkou komunitou a možnostmi si pořídit zadarmo nespočet rozšíření dle svých potřeb. V kostce může mít pro cloud každý uživatel svůj účet, pro který lze nastavit oprávnění, limity a vlastní přizpůsobení v podobě stylů a rozložení... Samotné sdílení souborů sice nezní jako lákadlo, ale jejich použití v hierarchické struktuře složek, řízení přístupu, komentování a přidělování k dalším modulům je velice ambiciózní. V Nextcloud lze využít modul pro poznámky, úkoly, kalendář, chat či „Deck“ (realizace nástěnky metodiky Kanban) a další. Většina modulů se umí propojit a sama o sobě disponují velikou škálou možností. V kalendáři pro představu pak lze vidět definované úkoly. Za zmínku stojí i online editace dokumentů typu Word pomocí Collabora Office. Všechna tato rozšíření s víceuživatelským přístupem a sdílením vytváří ohromný potenciál a možnosti pro řízení jakéhokoliv projektu ve virtuálním prostoru.

[22]

4 Doporučení pro aplikaci

Základní kameny projektu, které nelze opomenout a s kterými se musí pracovat, jsou čas, zdroje a rozsah. Na základě tohoto trojimperativu se aplikace zaměří na evidenci každého příslušného atributu pro pokrytí tří dimenzí. Zdroje bude aplikace evidovat v podobě financí a měnové jednotky a fyzických účastníků projektu. Čas se bude měřit pro entity v univerzálním čase a prezentující se uživateli v jeho nastavené zóně. Rozsah bude evidován jak časově pro úkoly a fáze, tak jako množství uživatelsky definované dílčí práce v kolekcích.

Na základě představeného rámce ITIL se následně autor zaměřil na jeho specifikace a doporučení a vybral tyto postupy pro zvážení a implementaci do výstupní cloudové aplikace pro řízení IT projektů.

Prvotní důležité pohledy na systém hodnot jsou s konkrétními zohledněními tato:

- **Organizace a lidé & Partneři a dodavatelé:** Tyto dvě dimenze aplikace přijme v podobě evidence samotné společnosti a k ní spjatých zaměstnanců/pracovníků a jejich hierarchického uspořádání – privilegovaného přístupu k jednotlivým oblastem. V projektu dle ITIL je potřeba zohlednit všechny zúčastněné strany, proto s příslušnými rolmi je nutno přizvat i partnery a dodavatele.
- **Informace a technologie:** Aplikace samotná zprostředkuje technologickou vrstvu a správu informací a dat pro řízení projektu. Další informační zpracování v jiných technologiích vyžaduje otevřenou a čitelnou formu integrace a komunikace s výstupní aplikací. To zaručí API a exportní funkce.
- **Hodnotové toky a procesy:** Na základě generického životního cyklu projektu aplikace umožní správu příslušných fází. Každá organizace disponuje jiným „know-how“, proto je důležité, aby uživatelské prostředí umožnilo pokrýt interní procesy organizace na abstraktnější vrstvě s širšími možnostmi evidence příslušných atributů.

Aplikace zohlední také zásady systému hodnot. Zaměření na hodnotu řešení neumí vynutit, ale strukturovaný proces nástroje pomůže s přehledným uspořádáním v definici dílčích artefaktů. Projekt se nemusí nacházet pouze v počátcích, tak aplikace umožní evidovat již běžící proces. Fundamentální iterativní přístup bude projektovat dílčí změny s možností komentování a schvalování. To samotné vede k dalšímu bodu zásad. Tento přístup podpoří většinou

spolupráci napříč zúčastněnými stranami. Aplikace musí fungovat na komplexní rovině, nicméně tento fakt nesmí zastínit jednoduchost.

Co se týče praxe správy požadavků na službu, jedná se o jeden z nejdůležitějších vstupů informačního projektu. Základem jsou řádné požadavky neopomíjející důležitá specifika. Aplikace umožní a začíná projekt mimo jiné položením zásadních požadavků do evidenčního seznamu, které mohou specifikovat účastníci a schvalovat zadavatelé.

Praktika finanční správy se překrývá již s dimenzí projektového trojúhelníku, proto není dále potřeba specifikovat podrobnosti. Důležité je neopomenout rozměr zdrojů a důsledně ho monitorovat a plánovat.

Ve spojení s praktikami a čtyřmi dimenzemi se bude realizovat následující abstrakt:

- V aplikaci se přistupuje k zásadě následujícím způsobem. Do jednotlivých oblastí projektu lze přidávat hotové instance, nebo instance rozpracované či plánované. Časově lze určit, kde jsme teď, co se utvořilo, s jakými výsledky a možnostmi přizpůsobit tomu další dění.
- Jednotlivé úkoly lze časově rozložit, přidat závislost na úkol, který musí být splněn předtím, a postupovat tak průběžně se zpětnou vazbou a stavem. Sdružování úkolů do fází taktéž podporuje tuto zásadu. Stav fáze vychází ze zpracování dílčích úkolů jí přiřazených. Pro ilustraci, nemůže se označit fáze vývoje za dokončenou, pokud předtím nebyly jasně ukončeny úkoly spadající do tohoto období.
- Aplikace podporuje role, jež se mohou distribuovat mezi zúčastněné strany a jež se mohou také přizpůsobit potřebám. Roli samotnou lze definovat jako seskupení oprávnění a navenek reprezentuje pohled na stav projektu z určité strany s konkrétními pravomocemi. Přístup aplikace tak umožňuje pozvat osoby do projektu, do kterého mohou přispívat nejen správci, ale všichni stakeholderi, a tak usměrňovat agilním způsobem chod projektu a včas reagovat na změny.
- Jelikož se projektu může účastnit více stran, vzniká zde prostor pro průnik více pohledů. Každý z nich může přispět v procesu řízení. K jednotlivým oblastem se lze vyjádřit třeba komentářem nebo výstupem jako jednou z funkčních komponent.
- Jednoduchý a praktický přístup aplikace umožňuje sledovat a analyzovat průběh zpracování všeho podstatného, jako jsou peníze, rozsah a čas, a tak podporovat rozhodnutí vedoucích rolí projektů.

5 Aplikace pro projektové řízení

Následující kapitoly uvedou konkrétní realizaci výstupní webové aplikace práce pro řízení IT projektů v cloudovém prostředí na základě doposud specifikovaných a rozebraných teoretických poznatků.

5.1 Funkční a nefunkční požadavky

Na základě zkušeností autora z doposud podstoupených pracovních příležitostí, zkušeností ze školních předmětů o vývoji a řízení projektu a z analýzy rozebraných dogmat teoretické části došel autor k následujícím požadavkům pro výstupní webovou aplikaci, která je v celém rozsahu pokrývá. Požadavky byly vypracovány i na základě úzké spolupráce s paralelně zpracovávanou diplomovou prací pana Bc. Tomáše Dokoupila, jenž se zaměřoval v obdobné problematice na jiné aspekty a pohledy řízení procesů, primárně přes modelovací jazyk BPMN a realizovanou podporu projektů na mobilní platformě.

Funkční požadavky:

Řízení přístupu

- FunReq1 Systém bude ověřovat a řídit přístup ke všem funkcionalitám aplikace na základě oprávnění. To se týká i všech funkčních požadavků.

Přihlašování

- FunReq2 Uživatel se bude moci přihlásit přes email a heslo k účtu.
- FunReq3 Uživatel si bude moci vytvořit uživatelský účet za pomoci emailové adresy, jména a hesla.
- FunReq4 Uživatel si bude moci obnovit heslo pomocí emailové adresy.

Administrace

- FunReq5 Administrátor bude moci vidět aktuální vzdálenou adresu připojení k API.
- FunReq6 Administrátor bude moci přidat roli, která bude projektová, či systémová.
- FunReq7 Administrátor bude moci roli smazat.
- FunReq8 Administrátor bude moci role zobrazovat.
- FunReq9 Administrátor bude moci k roli přiřazovat a odebírat oprávnění jednotlivých operací aplikace.
- FunReq10 Administrátor bude moci zobrazit, smazat a přidat oprávnění.

- FunReq11 Administrátor bude moci synchronizovat existující oprávnění, a tím vytvořit nová, dle reflexe – dostupných konečných bodů na komunikačním rozhraní.
- FunReq12 Administrátor bude moci zobrazit, vyhledávat uživatele aplikace a spravovat jejich role.
- FunReq13 Administrátor bude moci zobrazit aplikační logy změn na datové vrstvě společně s autorem změn.

Projekt

- FunReq14 Účastník projektu se bude moci přihlásit k prostředí projektu přes kombinaci organizace, kterou prezentuje, a výběrem projektu samotného, k jemuž má přijatou pozvánku.
- FunReq15 Uživatel bude moci vidět nástěnku projektů, organizací a úkolů, které jsou pro něj relevantní.
- FunReq16 Přihlášený účastník bude moci vidět přehled projektu (jméno, popis, náklady, odhadovaný čas dokončení, datum vytvoření a uživatele, jež vlastní projekt)
- FunReq17 Přihlášený účastník bude moci vidět poslední změny v projektu.
- FunReq18 Přihlášený účastník bude moci vidět základní analýzy projektu pomocí grafů (celkové finanční využití prostředků, poměrový stav všech definovaných úkolů).
- FunReq19 Přihlášený účastník bude moci upravit nastavení projektu (jméno, popis, měna).
- FunReq20 Přihlášený účastník bude moci vidět přehlednou projekci zasazenou v čase ohledně definovaných fází projektu společně s časovými detaily.
- FunReq21 Přihlášený účastník bude moci filtrovat vizualizaci fází na základě jejich aktuálních stavů.
- FunReq22 Přihlášený účastník bude moci rozkliknout jednotlivé fáze pro detailnější pohledy a ovládání.
- FunReq23 Přihlášený účastník bude moci v detailu fáze vidět přehled (jméno, období, stav, popis, rozpočet, náklady, odhadovaný čas, počet dní od započetí, počet dní do konce fáze).
- FunReq24 Systém bude projektovat náklady a odhadovaný čas na základě sumy atributů přiřazených úkolů fázi.

- FunReq25 Přihlášený účastník bude moci v detailu fáze spravovat asociované úkoly.
- FunReq26 Přihlášený účastník bude moci v detailu fáze spravovat asociované soubory.
- FunReq27 Přihlášený účastník bude moci v detailu fáze komentovat.
- FunReq28 Přihlášený účastník bude moci v detailu fáze spravovat výstupy.
- FunReq29 Přihlášený účastník bude moci v detailu fáze spravovat samotné atributy fáze (jméno, rozpočet, popis, časové období, stav).
- FunReq30 Systém dovolí přiřadit rozpočet pouze tehdy, pokud částka nepřesáhne volné prostředky.
- FunReq31 Systém dovolí změnit stav fáze na „dokončeno“ pouze tehdy, pokud jsou vyřešené asociované úkoly (stavy).
- FunReq32 Přihlášený účastník bude moci spravovat požadavky projektu a atributy: jméno, popis, stav.
- FunReq33 Systém bude zobrazovat pokrytí požadavků, jestli ke každému požadavku existuje implementující úkol.
- FunReq34 Přihlášený účastník bude moci spravovat dílčí výstupy.
- FunReq35 Přihlášený účastník bude moci spravovat úkoly s atributy: název, časové období, stav, priorita, odhadovaný čas, náklady, popis, asociovaná fáze, implementační skupiny, pokryté (schválené) požadavky a požadovaný dokončený úkol).
- FunReq36 Přihlášený účastník bude moci zobrazit pouze své úkoly.
- FunReq37 Přihlášený účastník bude moci řadit úkoly, a to hlavně přes prioritu a čas.
- FunReq38 Přihlášený účastník bude moci spravovat týmy, přiřazovat účastníky do týmů.
- FunReq39 Přihlášený účastník bude moci spravovat další účastníky (změna role, odebrání účastníka).
- FunReq40 Přihlášený účastník bude moci spravovat dílčí výstupy.
- FunReq41 Přihlášený účastník bude moci spravovat pozvánky uživatelů do projektu.
- FunReq42 Přihlášený účastník bude moci spravovat rozpočty projektu inkrementálním způsobem. Celkový budget počítán jako suma dílčích příjmů.

- FunReq43 Přihlášený účastník bude moci spravovat (nahrát, smazat, upravovat, zobrazovat) soubory.
- FunReq44 Uživatel bude moci dostávat notifikace o změnách v projektu.
- FunReq45 Systém bude pro důležité popisy entit projektu umožňovat formátovat text pomocí tzn. „Rich Text“ editoru.

Účet a nastavení

- FunReq46 Uživatel si bude moci vybrat a nastavit aplikaci v tmavém a světlém provedení.
- FunReq47 Uživatel si bude moci vybrat jazykovou lokalizaci: čeština, angličtina.
- FunReq48 Uživatel si bude moci vybrat a nastavit, v jaké zóně se budou zobrazovat časové údaje.
- FunReq49 Uživatel si bude moci změnit heslo po vložení svého starého.
- FunReq50 Uživatel si bude moci prohlédnout pozvánky do projektu a organizací včetně jejich odmítnutí/přijetí.
- FunReq51 Uživatel si bude moci zobrazit organizace, v kterých disponuje nějakou rolí.
- FunReq52 Uživatel si bude moci vytvořit vlastní organizaci.
- FunReq53 Uživatel si bude moci prohlédnout vlastní profil s informacemi o profilovém obrázku, popisku, jméně, evidovaném emailu a datu registrace.
- FunReq54 Uživatel si bude moci nastavit profilový popisek.
- FunReq55 Uživatel si bude moci nahrát profilový obrázek.
- FunReq56 Systém bude zobrazovat velké počáteční písmeno jména, pokud nebude obrázek uživatele dostupný.

Nefunkční požadavky:

Technologie

- NonFunReq1 Projekty budou součástí jednoho řešení Visual Studio 2022.
- NonFunReq2 Projekty budou psané v jazyce C#.
- NonFunReq3 Front-end projekt bude implementovaný pomocí webového rámce Blazor ASP.NET 8.
- NonFunReq4 Back-end projekt bude implementovaný pomocí Web API ASP.NET 8.

- NonFunReq5 Front-end projekt bude využívat primárně knihovny MudBlazor pro vizuální komponenty.
- NonFunReq6 Back-end projekt bude využívat Microsoft SQL Server pro datovou vrstvu.
- NonFunReq7 Webová aplikace bude nasazena na cloudovou platformu Microsoft Azure.

Přístup

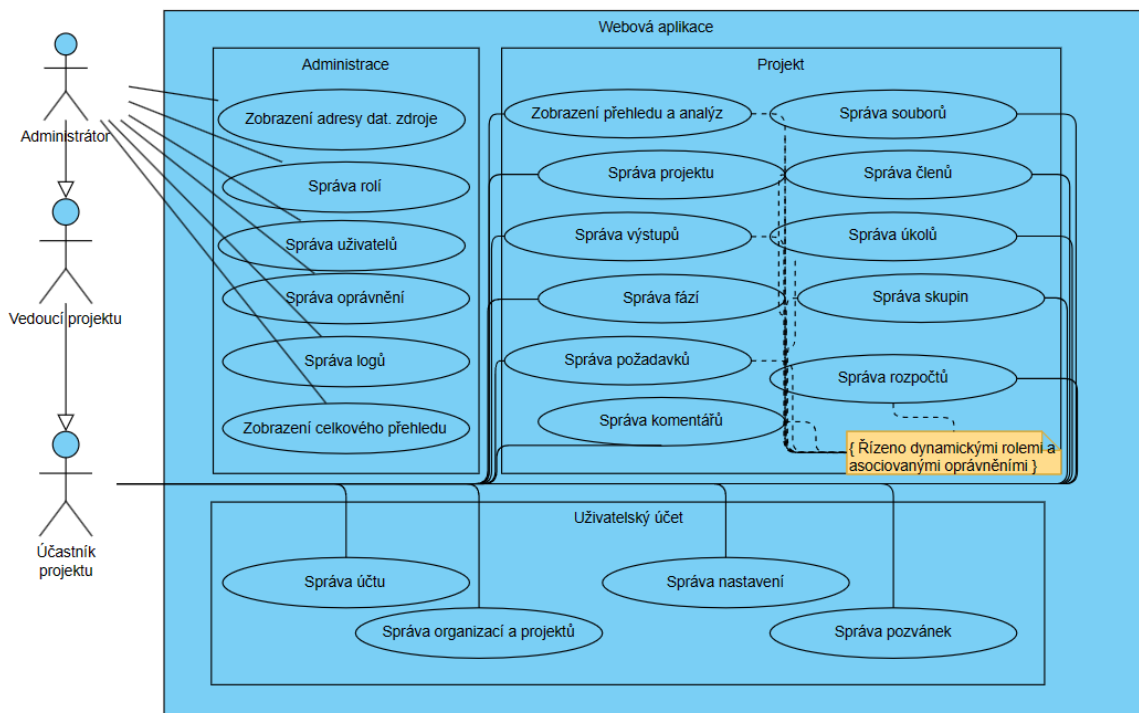
- NonFunReq8 Webová aplikace bude veřejně dostupná ve webovém prohlížeči na specifické URL adrese.

Optimalizace a bezpečnost

- NonFunReq9 Webová aplikace bude dostupná pod platným certifikátem zabezpečené vrstvy SSL v rámci komunikace.
- NonFunReq10 Webová aplikace bude dodržovat jazykové standardy a praktiky (Clean Code).
- NonFunReq11 Webová aplikace bude ověřovat autorizace a autentizace pomocí JWT.
- NonFunReq12 Operace v aplikaci budou ověřovány vůči oprávněním rolí.

5.2 Příklad užití

Obrázek níže specifikuje všechny obecné případy užití v systémech výstupní webové aplikace.



Obrázek 9: Use Case diagram aplikace (vytvořeno pomocí Visual Paradigm Online, zdroj vlastní)

Do systému vstupují tři základní aktéři:

- Administrátor – role, která se stará o běh aplikace se všemi oprávněními,
- Vedoucí projektu – vždy musí existovat alespoň jedna kompetentní osoba vedoucí projekt, která ho spravuje,
- Účastník projektu – obecná role, která může nabývat různých oprávnění na základě postavení v projektu. Může jít o analytika, vývojáře nebo třeba dodavatele.

Aplikace se rozděluje na tři základní části:

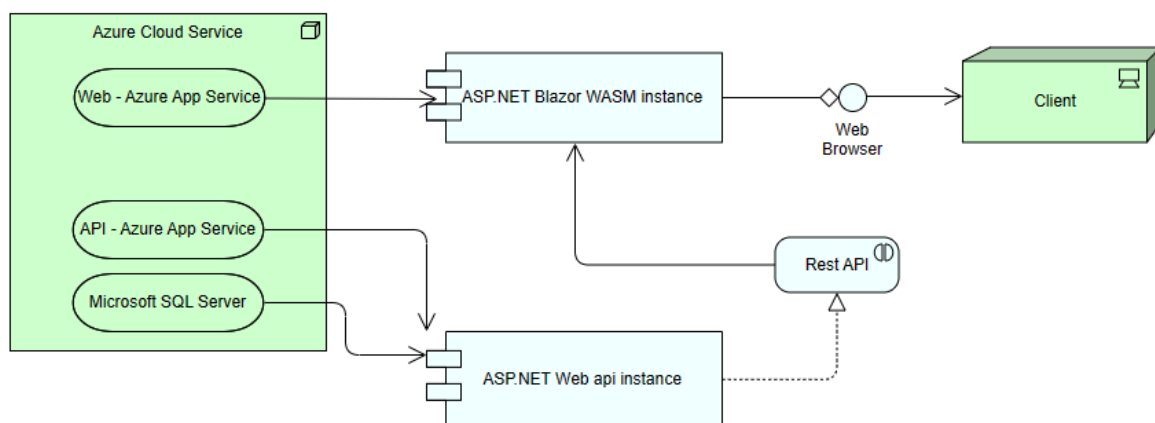
- Administrace – administrátor spravuje další systémy řízení,
- Projekt – jádro aplikace pro projektové řízení,
- Uživatelský účet – prostor pro uživatelská nastavení, editaci účtu a správu pozvánek.

5.3 Architektura systému

Projekt aplikace je realizován pomocí vývojového prostředí Visual Studio 2022 s následující strukturou řešení:

- **Web** – Projekt implementuje čistě front-end část aplikace v rámci Blazor WebAssembly.

- **API** – Projekt slouží pro serverovou službu rozhraní komunikace mezi klientskou částí a datovou/logickou vrstvou. Posílání zpráv realizuje přístup REST přes http protokol.
- **SharedLibrary** – Projekt obsahuje datové struktury, které se sdílí napříč řešením. Názorným zástupcem můžou být třídy pro přenos dat a definici jejich datových typů zvané DTO. Nacházejí se tu i služby pro registraci v rámci Dependency Injection.
- **ComponentSharedLibrary** – Projekt uchovává komponenty pro webovou aplikaci, které mohou být znovu použitelné. Jedná se o rozložení, formuláře, tabulky a další.



Obrázek 10: Architektura aplikace v jazyce ArchiMate (vytvořeno pomocí Visual Paradigm Online, zdroj vlastní)

Jak ukazuje model, uživatelské zařízení za pomoci webového prohlížeče načte při příchodu na webovou stránku strojový kód aplikace do webového rozhraní, který komunikuje přes rest API s databází SQL Server a další aplikační logikou v komponentě rozhraní pro programování aplikace. Vše umožňují služby Microsoft Azure Cloudu, které poskytují výpočetní výkon a uložení pro zpracování klientských požadavků.

Verzování změn implementuje distribuovaný systém Git na platformě GitHub od Microsoft v soukromém depozitáři, a tak vývoj mohl probíhat bez problému na vícero zařízeních se synchronizací zdrojových souborů.

5.4 Použité technologie

Kapitola o použitých technologiích představí klíčové prostředky v podobě rámců, knihoven a hostingu, o které se opřela výstupní cloudová webová aplikace.

5.4.1 Azure

Azure je cloudová platforma pro poskytování služeb ve světě IT. Výpočetní řešení lze jednoduše přesunout mimo vlastní infrastrukturu. Zodpovědnost za provoz tak přechází do profesionálních rukou. Zálžitosti jako bezpečnost část vyžadují znalosti a zkušenosti, a tak se takovým potenciálním problémům lze vyhnout, aby se zaručila stabilita aplikace. Cíle a benefity platformy jsou mnohostranné. Azure pomáhá organizacím modernizovat jejich IT infrastrukturu, migrovat aplikace do cloudu a vyvíjet inovativní řešení s využitím nejnovějších technologií, jako je umělá inteligence, strojové učení a internet věcí (IoT). Díky Azure mohou firmy rychle a snadno škálovat své IT zdroje na základě měnících se potřeb, což zvyšuje agilitu a flexibilitu v reakci na dynamické tržní podmínky. Cloudový model "pay-as-you-go" v Azure umožňuje organizacím platit pouze za ty zdroje, které skutečně využívají, což snižuje kapitálové výdaje a optimalizuje provozní náklady. Díky široké škále nástrojů a služeb Azure mohou organizace zefektivnit procesy a zvýšit produktivitu svých zaměstnanců. Azure také splňuje nejvyšší standardy zabezpečení a dodržování předpisů, což zajišťuje ochranu dat a aplikací organizací.

Hlavní vlastnosti Azure zahrnují širokou škálu služeb, flexibilní modely nasazení, podporu hybridních a více cloudových řešení, globální dostupnost a otevřenost a integraci s existujícími systémy. Služby Azure pokrývají všechny klíčové oblasti, jako je výpočet, úložiště, sítě, databáze, analýza, IoT, umělá inteligence a vývoj softwaru. Tyto služby poskytují uživatelům široký výběr možností pro vytváření a správu jejich IT prostředí v cloudu.

[23]

Webová aplikace, stejně jako Web API a databáze Microsoft SQL Server, je nasazena na platformě Azure. Velice přívětivou možností je způsob integrace přímo do vývojových prostředí. Publikace změn lze provést nastavením profilu a potom již stačí jedním tlačítkem propagovat změny do veřejné služby. Uživatel opravdu nemusí vynaložit velké úsilí pro provoz informačních projektů. Na druhou stranu si Microsoft rád za vše nechá zaplatit, to je potřeba také brát v úvahu, takže i pro představu, proces logování může být zpoplatněn. Také nastává stav, že se správci vzdávají v určitých rovinách možnosti komplexní flexibility tak, aby si vše nastavili na míru dle svých potřeb.

5.4.2 ASP.NET

ASP.NET je rozsáhlý framework vyvinutý pro vývoj webových aplikací v jazyce C#, VB.NET a F#. Jeho komplexnost spočívá v tom, že není pouze programovacím jazykem, ale spíše celou sadou knihoven, nástrojů a technologií, které vývojářům usnadňují a zrychlují proces tvorby webových stránek a aplikací. Vývojáři se setkávají s řadou běžných problémů při vytváření webových aplikací, a ASP.NET poskytuje hotová řešení pro mnoho z nich.

ASP.NET nabízí robustní funkce pro zabezpečení webových aplikací, včetně ochrany před útoky a neoprávněným přístupem. Dále umožňuje snadnou implementaci autentizace uživatelů a autorizace přístupu k datům a funkcím aplikace. S integrovanou podporou pro práci s databázemi mohou vývojáři snadno ukládat a načítat data z různých databázových systémů. ASP.NET poskytuje také komponenty pro správu formulářů na webových stránkách, což usnadňuje tvorbu a validaci formulářů.

Kromě těchto základních funkcí nabízí ASP.NET řadu dalších nástrojů a funkcí, jako je routing, caching, správa sezení a mnoho dalších. Díky této široké škále funkcí je ASP.NET vhodný pro různé typy projektů, od malých webových stránek až po rozsáhlé webové portály a podnikové aplikace.

Výhody používání ASP.NET spočívají v jeho produktivitě, udržovatelnosti, bezpečnosti, škálovatelnosti a podpoře komunity vývojářů. Vývojáři mohou psát méně kódu a dosáhnout rychlejšího vývoje aplikací, díky strukturálnímu a modulárnímu kódu je aplikace snadněji udržovatelná, a robustní funkce pro zabezpečení zajistí ochranu aplikace před různými druhy útoků. Navíc ASP.NET disponuje velkou a aktivní komunitou vývojářů, kteří si navzájem pomáhají a sdílejí své znalosti, což znamená, že vývojáři mohou využívat různé zdroje a materiály pro svou práci.

ASP.NET již na trhu nějakou chvíli působí, s tím se i časem měnily různé přístupy. Starším řešením byl rámec WebForms, obdoba WinForms, s cílem programovat aplikaci podobným způsobem pro webovou verzi přes funkční ovládací komponenty a jejich skládáním na plátno s přidáním logiky na pozadí. Novějším a dnes stále populárním způsobem se stal MVC dostupný i v Core verzi. Architektura MVC rozděluje logiku aplikace na tři základní logické prostory. M pro model stanovuje strukturu řídicí datovou vrstvu, V pro view, pohled, je určeno pro strukturu zobrazující se koncovému uživateli a C pro kontrolery, které řídí většinou aplikační logiku. Každá část aplikace tak má svoji zodpovědnost a funkci/e. A posledním

trumfem novodobé verze ASP.NET je přívlastek Core, jenž převádí rámce, knihovny do otevřené podoby a implementace aplikací již dostala multiplatformní kabát. Tato technologie primárně využívá proprietární server IIS, nicméně Nginx a Apache lze nastavit taktéž pro interpretaci aplikace.

[24]

5.4.3 Blazor WebAssembly

Blazor, jako inovativní framework pro vývoj webových stránek, přináší vývojářům webových technologií z balíčku .NET nové možnosti, které dříve nebyly možné. Jeho klíčovou vlastností je schopnost vytvářet interaktivní uživatelská rozhraní pomocí jazyka C# a strukturální přístup přes modulární skládání komponent, místo tradičního JavaScriptu. Tento přístup přináší řadu výhod, neboť vývojáři se mohou soustředit na jeden programovací jazyk a využít existující ekosystém knihoven a nástrojů.

Díky této technologii nabízí Blazor několik zásadních výhod. Patří mezi ně vysoký výkon a nízká latence díky nativnímu kódu běžícímu v prohlížeči, kompatibilita s moderními prohlížeči a snadná sdílení kódu díky využití jazyka C# a ekosystému .NET.

Blazor poskytuje dva hlavní režimy: Blazor WebAssembly a Blazor Server. V režimu WebAssembly je kód Blazoru kompilován do WASM a spouští se přímo v prohlížeči, zatímco v režimu Server běží kód na serveru a výsledný HTML je odeslán do prohlížeče. Komunikace dle volby pak probíhá buď přes volání rest API, WebSocket, nebo pro serverovou verzi proprietární SignalR, který umožňuje efektivní posílání zpráv a jednoduchou implementaci logiky pro real-time aplikace.

[25]

Klíčovým stavebním prvkem je komponenta, která se jako třída definuje převážně do jednoho souboru. Struktura obsahuje část pro značkovací jazyk, jako je HTML, kde lze vnořovat další komponenty nebo elementy hypertextu samotného. Do druhé části vývojář specifikuje aplikační logiku pro komponenty, načítání dat atd... Specifikace kódu lze uvést přímo do razor souboru, nebo rozdělit logiku do rozšiřující třídy. Komponenty drží přístup tříd, a tak je pro ně dostupná třeba i dědičnost. Často se využívá injekce závislostí registrovaných služeb přímo v souboru za pomoci DI.

```

<PageTitle>Counter</PageTitle>
<h1>Counter</h1>
<p role="status">Current count: @currentCount</p>
<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
@code {
    private int currentCount = 0;
    private void IncrementCount()
    {
        currentCount++;
    }
}

```

Zdrojový kód 1: Ukázka struktury komponenty Razor v Blazor [26]

5.4.4 MudBlazor

MudBlazor je populární knihovna připravených komponent, která si našla cestu k mnoha vývojářům využívající rámec Blazor. Alternativní volbou mohou být Telerik, Syncfusion, Blazorise. Na rozdíl od svých konkurentů, MudBlazor je k použití zdarma jako Open Source vydání. Největší předností se stal rozsáhlý katalog dostupných, funkčních a čistých komponent. Stejně tak knihovna přichází s pravidelnými aktualizacemi a rozšířeními. Cílem je poskytnout co nejvíce ovládacích prvků tak, aby vývojář nemusel psát vlastní komponenty, a už vůbec ne samotný JavaScript. MudBlazor disponuje sadou tříd CSS, ikon, rozložení a komponent. Nechybí základní prvky pro tlačítka, formuláře, responzivní tabulky nebo dialogy. Pokročilejší komponenty pokrývají Carousel, záložky, Drag&Drop boxy, našeptávač, časová osa a mnoho dalších.

5.4.5 Web API

Z dalších otevřených produktů balíčku .NET je čisté řešení pro jednoduchou implementaci projektu komunikačního rozhraní Web API. Rámec mapuje http dotazy na obslužné metody kontrolerů, kde již rutina v pohodlí jazyka C# zpracuje požadavek a vrátí odpověď uživateli, nejčastěji v podobě strojově čitelného formátu JSON. Velká část dodatečných nastavení lze specifikovat pomocí anotací u samotných metod. To platí i pro velké množství dostupných rozšiřujících knihoven. Hlavní zástupce tohoto přístupu je implementace autentizace a autorizace, jež se může specifikovat přívětivým způsobem následovně:

```
[Authorize(Users="John,Alice")]
public class StorageController : ApiController
{
}
[Authorize(Roles="Administrator")]
public class AdminController : ApiController
{
}
```

Zdrojový kód 2: Ukázka anotace pro řízení přístupu Web API (zdroj vlastní)

5.4.6 EF Core

Entity Framework Core je označení pro rámec podporující objektově relační mapování (ORM) v rodině .NET. Jedná se o vrstvu mezi datovou a aplikační. Přístupovat lze shora, či zdola. To znamená, že lze vytvářet automaticky aplikační modely na základě existujících databázových tabulek, nebo obráceně vytvářet automaticky databázovou strukturu z nadefinovaných tříd a kontextu. ORM přístup dost často volí vývojáři kvůli jednoduchosti nasazení, práce s daty, bezpečnosti a možnosti zrychlit přístup k datům využitím cachování. Změny se propagují přes migrační soubory, které na sebe navazují. Většina aspektů databáze nejsou opomenuty a bere se na ně zřetel i při využívání této abstraktní vrstvy. Příkladem může být indexace, vazby a integritní omezení.

EF Core pracuje jednotně s daty na základě dotazů jazyka LINQ vedle C#. Základní přístupy jsou přes rozšiřující metody, které více tíhnou dotazováním se objektovým způsobem s využitím anonymních funkcí, jež jsou známé i v jazyce Java. Druhým přístupem se stala klíčová slova LINQ tvořící strukturovaný dotaz podobný SQL. O typ uložení a způsob komunikace přes příslušný databázový dotazovací jazyk se postarají podporované ovladače.

Přístup k datům určuje speciální kolekce „DbSet“ deklarovaná v kontextovém souboru s generickým parametrem modelu reprezentující datovou entitu uložení. Nad touto vlastností instance kontextu se následně provádějí CRUD operace a rámec sleduje změny, které na pozadí převádí na příslušné dotazy, jenž se aplikují v technologickém řešení uložení.

5.5 Implementace klíčových aplikačních složek

Následující důležité interní systémy a funkcionality realizované v aplikaci byly vybrány autorem pro přiblížení a detailní rozebrání. Přístupy lze brát v obecné rovině pro použití

v jiných systémech, nestahují se pouze na tento konkrétní případ implementace v rámci výstupní aplikace práce.

5.5.1 Logování a notifikace

Aplikace podobným způsobem jako soft delete realizuje zaznamenávání změn datové vrstvy. Nedochozí k ní přes spouštěče v databázi, ale přes přetížení metody `SaveChangesAsync`, která je dostupná v kontextu díky Fluent API pro EF Core mapování. Změny dat se potvrzují právě přes tuto metodu, kterou rozšiřuje rutina pro zaznamenávání. Ta využívá vlastnosti předka kontextu, která obsahuje detaily o změnách záznamů a typu operace nad nimi. Informace se transformují do struktury kolekce abstraktních datových typů tabulek obsahující v klíči atribut entity a v hodnotě pole dvou řetězců reprezentující starou a novou hodnotu záznamu. K nim se připojí informace o přihlášeném uživateli a název oblasti změn a uloží se do tabulky v databázi.

K veškerému logování přistupuje administrátor přes oprávněnou sekci ve webové aplikaci, kde lze přehledně vidět veškeré sledované změny. Na základě logů se konstruuje notifikační zpráva pro uživatele, jež ho mají za úkol informovat o ději v projektech, na kterých uživatel spolupracuje. Záznamy rozšiřuje i příznak o přečtení. Ve výsledné projekci uživatel vidí číslo jako počet nepřečtených upozornění v odznáčku ikony obálky, která po rozkliknutí zobrazí seznam detailů.

5.5.2 Soft delete

Soft delete neboli „jemné smazání“ je oblíbená metoda přístupu k mazání dat na datové vrstvě. Klasické odebírání „hard delete“ zapříčiní trvalé odstranění záznamu, které lze obnovit jen z celistvých záloh aplikace, pokud jsou vůbec dostupné. Soft delete naopak data neodstraní, pouze je označí určeným příznakem, a pak se při operacích nad daty tyto záznamy, většinou automaticky, ignorují. Jednou z osvědčených příznaků smazání se stal nulovatelný atribut datumu, který zároveň nese informaci o tom, kdy k odstranění došlo, pokud je vyplněn. Velkou nevýhodou jemného mazání se stává fakt, že odebráním dat se neredukuje místo na disku, paradoxně se uchovávají meta data ještě navíc. Zjevná výhoda ochrany dat přístupu proti ztrátě není potřeba vysvětlovat.

Soft delete implementuje i samotná aplikace. Za pomoci přetížení metody Fluent API v EF Core pro ukládání změn datových modelů lze dosáhnout toho, že se entita nesmaže při jeho

odebírání, ale nastaví se příznak smazání tak, že vlastnost „DeletedAt“ se nastaví na aktuální hodnotu času v UTC.

```
if (entry.Entity is ISoftDelete entity)
{
    switch (entry.State)
    {
        case EntityState.Deleted:
            entry.State = EntityState.Modified;
            entry.CurrentValues["DeletedAt"] =
DateTime.Now.ToUniversalTime();
            break;
    }
}
```

Zdrojový kód 3: Přetížení metody SaveChangesAsync pro implementaci Soft Delete (zdroj vlastní)

5.5.3 Autentizace a autorizace

K ověřování práv v aplikaci je využita implementace JWT tokenu. Tento token se skládá z hlavičky, kde se nachází informace o struktuře, primárně je tu uložen algoritmus pro hashování. V struktuře následuje tělo, data označovaná jako „Payload“, kam se ukládají potřebné informace, v případě autentizace se může jednat o identifikátor uživatele. Struktura pokračuje poslední částí, která definuje podpis jako hash všech uvedených čitelných dat společně s privátním klíčem, kterým se pak ověřuje integrita tokenu. Token se zapisuje ve formátu JSON, konkrétně jako dvojice objektů a podpis oddělené tečkou.

Ve výstupní aplikaci se do tokenu primárně ukládá identifikátor uživatele v podobě emailové unikátní adresy a dostupná práva pro uživatelský účet. Token se přiloží ke každému uživatelskému dotazu do hlavičky na rozhraní serveru. Zde se data rozeberou a za pomoci vlastní bezpečnostní anotace u metody obslužného kontroleru, která specifikuje potřebné oprávnění, se v middleware ověří, zda uživatel takovým oprávněním disponuje.

```
[PermissionAuthorize("Requirements.GetProjectRequirement")]
[HttpGet("{requirementId}")]
public async Task<StatusResponse<Requirement>> GetProjectRequirement(long
requirementId)
```

Zdrojový kód 4: Ukázka anotace pro potřebné oprávnění (zdroj vlastní)

```

var permissions =
JsonSerializer.Deserialize<HashSet<string>>(stringToken.Claims.FirstOrDefault(
c => c.Type == "Permissions")?.Value ?? "{}");
if (!permissions.Contains(actionAttributes.Permission))
{
    context.Response.StatusCode = 403;
    return;
}

```

Zdrojový kód 5: Ověření práv uživatele v middleware (zdroj vlastní)

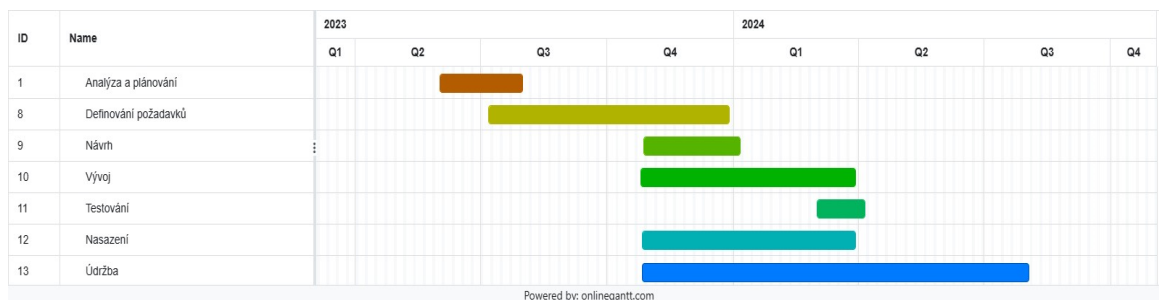
5.6 Ganttův diagram vývoje aplikace

Ganttovy diagramy jsou vynikajícím nástrojem pro vizualizaci časového harmonogramu projektů, avšak jejich využití je mnohem širší než jenom pro plánování. Poskytují široké spektrum možností, které ocení jak projektoví manažeři, tak i programátoři a portfolioví manažeři.

Základní funkce Ganttových diagramů poskytují užitečné nástroje pro efektivní správu projektů. Přehledná časová osa umožňuje zobrazit projektové úkoly v chronologickém pořadí, což poskytuje okamžitý přehled o délce trvání jednotlivých úkolů, jejich vzájemných závislostech a celkovém harmonogramu projektu. Dále umožňují přiřadit úkoly konkrétním členům týmu, což jasně vymezuje zodpovědnost a usnadňuje koordinaci práce. Také umožňují stanovit termíny dokončení a odhadnout náklady na realizaci úkolů, což podporuje efektivnější plánování zdrojů a identifikaci rizik. Důležitou funkcí je také sledování pokroku projektu, což se děje aktualizací stavu jednotlivých úkolů a vizualizací reálného průběhu projektu v porovnání s původním plánem.

Rozšířené funkce Ganttových diagramů dávají uživatelům možnost provádět pokročilé analýzy a plánování projektů. Například je možné definovat závislosti mezi úkoly, což umožňuje lépe plánovat kritickou cestu projektu a optimalizovat časový harmonogram. Další možností je vyznačení klíčových milníků projektu, které slouží jako orientační body pro sledování průběhu projektu a dosahování dílčích cílů. Ganttovy diagramy lze také filtrovat a zobrazovat podle různých kritérií, což umožňuje zaměřit se na specifické aspekty projektu a získat relevantnější informace. Navíc je možné integrovat Ganttovy diagramy s dalšími nástroji pro řízení projektů, jako jsou kanban nástěnky, reporty, časové rozvrhy a nástroje pro týmovou spolupráci, což umožňuje komplexní a efektivní správu projektu v jednom prostředí.

[27]



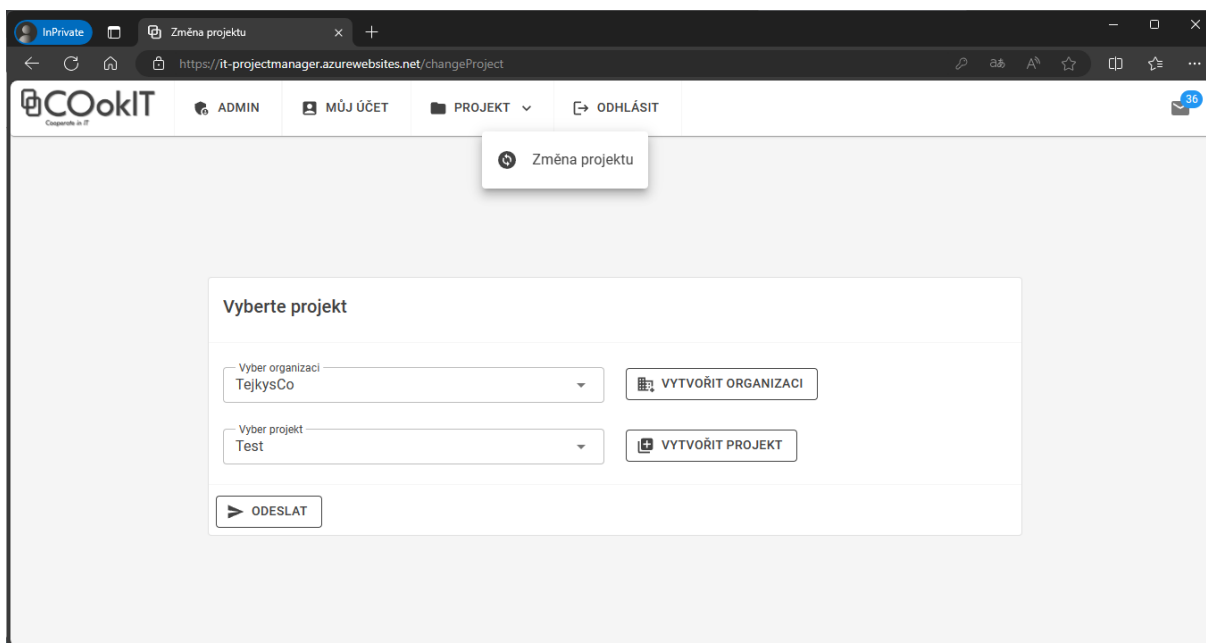
Obrázek 11: Životní cyklus vývoje webové aplikace v Ganttově diagramu (zdroj vlastní)

Obrázek výše specifikuje zobrazení cyklu projektu výstupní aplikace práce do časové dimenze. Lze vidět, že prvotní analýza pro sběr požadavků, rozvržení projektu a celkovým konceptem započal již v první polovině roku 2023. Během fáze odstartovalo definování požadavků na funkčnost aplikace. Další fáze pro návrh, vývoj, nasazení a údržbu se více méně odehrávali paralelně, jelikož průběžné přírůstky se propagovaly navenek pro validaci práce primárně z hlediska konzultací s vedoucím práce. Se zpětnou vazbou se měnily požadavky, či opravovaly chyby. Rozvržení práce by s velkou pravděpodobností vypadalo jinak, pokud by šlo o širší tým pracovníků.

5.7 Základní průvodce užití aplikace

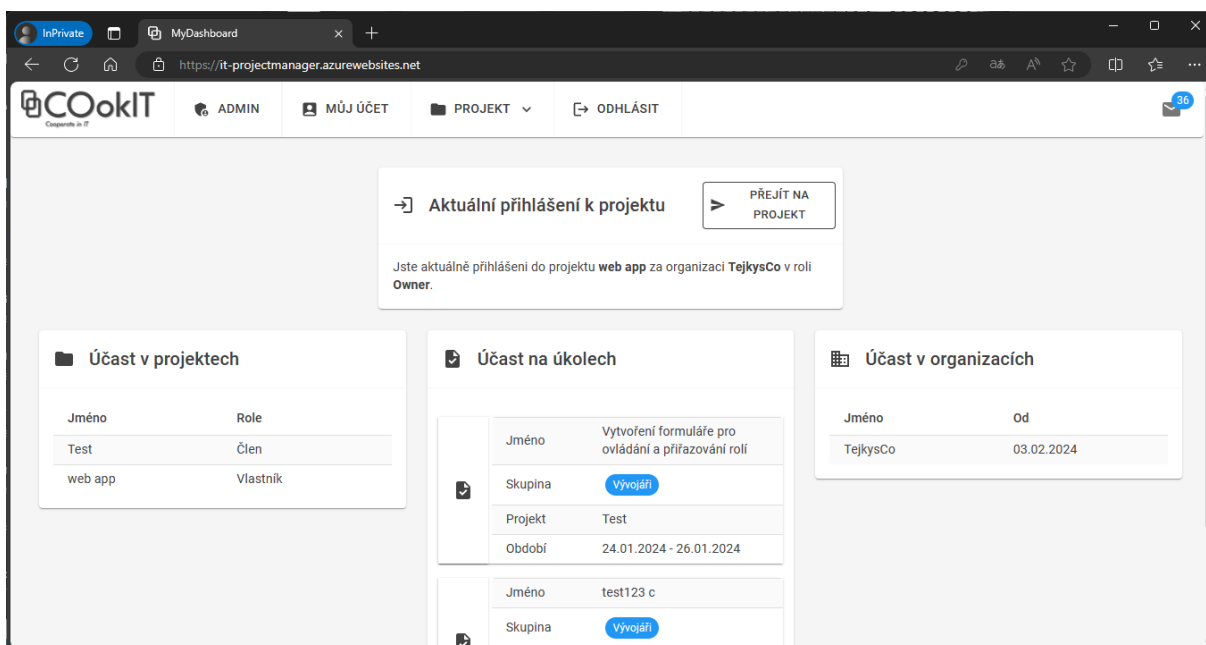
Následující odstavec představí jeden z možných scénářů v roli projektového manažera, který chce vytvořit a řídit komplexní informační projekt přes výstupní aplikaci práce.

Manažer zadá URL adresu webové aplikace. Jelikož nikdy aplikaci nepoužíval, po načtení stránky přejde na záložku „Registrace“ a zadá své požadované údaje pro vytvoření účtu. Pokud mu systém odpoví, že existující emailová adresa již existuje, přejde na záložku přihlášení, pokud si nepamatuje heslo, tak na záložku „Zapomenuté heslo“, kam zadá svůj email a systém vygeneruje klíč platný 1 hodinu a zašle ho do schránky. Následně manažer zkopíruje klíč z emailového klienta, přejde na záložku „Mám token“ a zadá ho s dalšími údaji a novým heslem do formuláře. Poté může konečně přejít k přihlášení. Jelikož není nikde v systému evidován, v menu „Projekt“ a „Změna projektu“ vytvoří organizaci, pod jakou se bude projekt evidovat. Následně obdobně pokračuje na tvorbu projektu, kde specifikuje požadovanou organizaci, jméno projektu, měnovou jednotku a popis.



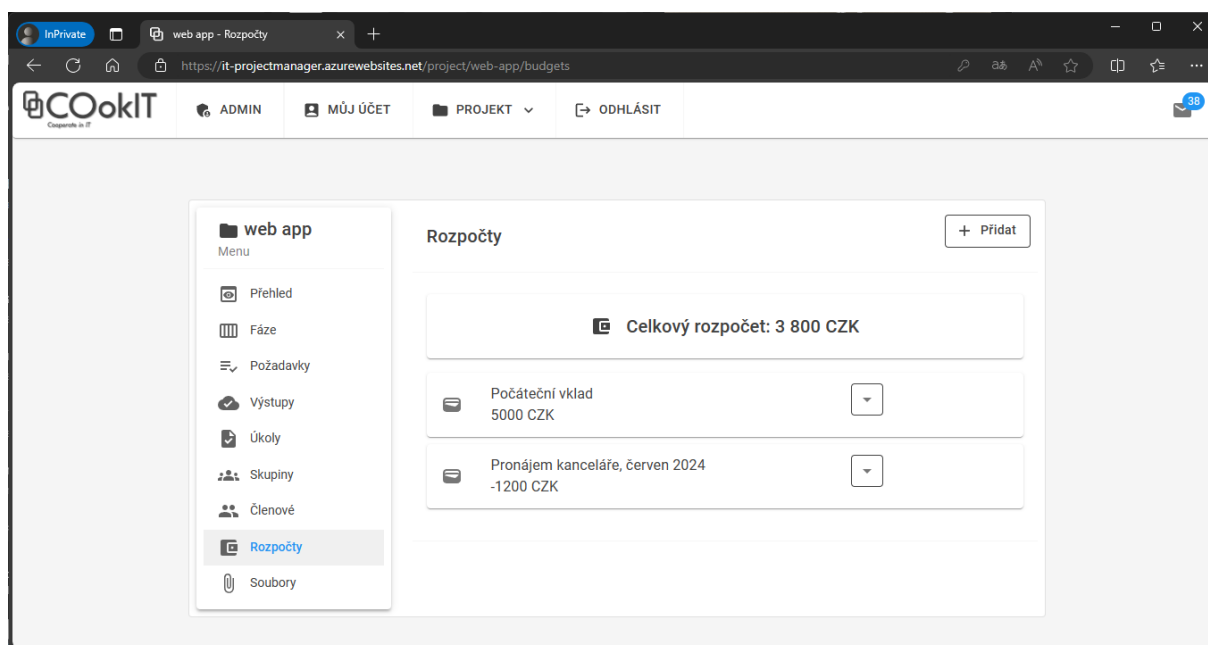
Obrázek 12: Ukázka aplikace – vstup do projektu (zdroj vlastní)

Na stránce „Změna projektu“ se již manažer přihlásí touto kombinací do řídicího prostředí. K projektu lze přejít tlačítkem na hlavním stránce, nebo v horním menu. Po kliknutí bude přesměrován na rozhraní projektu, přesněji na stránku přehledu. Jelikož manažer už obdržel požadavky od projektanta, který je pečlivě vykomunikoval se zákazníkem, požadavky manažer vloží v příslušném modulu projektu s dostupnými specifikacemi.



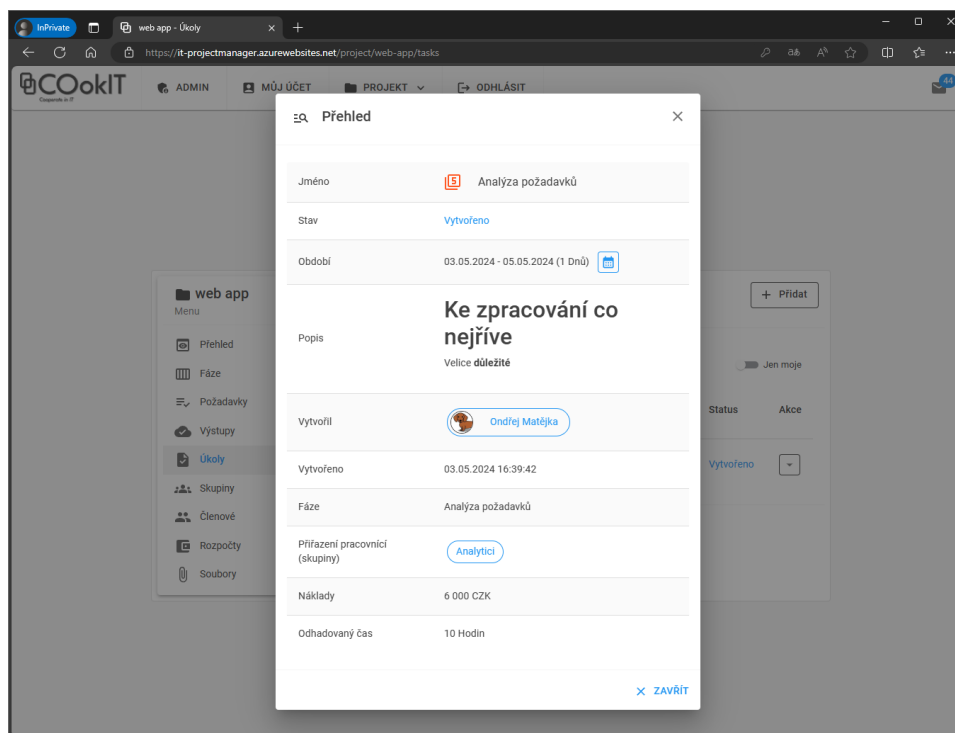
Obrázek 13: Ukázka aplikace – hlavní dashboard (zdroj vlastní)

Jelikož projekt disponuje počáteční investicí, vloží manažer tuto částku do modulu „Rozpočty“. Aby také rozdělil projekt na dílčí logické časové úseky, které lze lépe řídit a validovat, v modulu „Fáze“ manažer vytvoří tato období, milníky, na základě životního cyklu projektu.



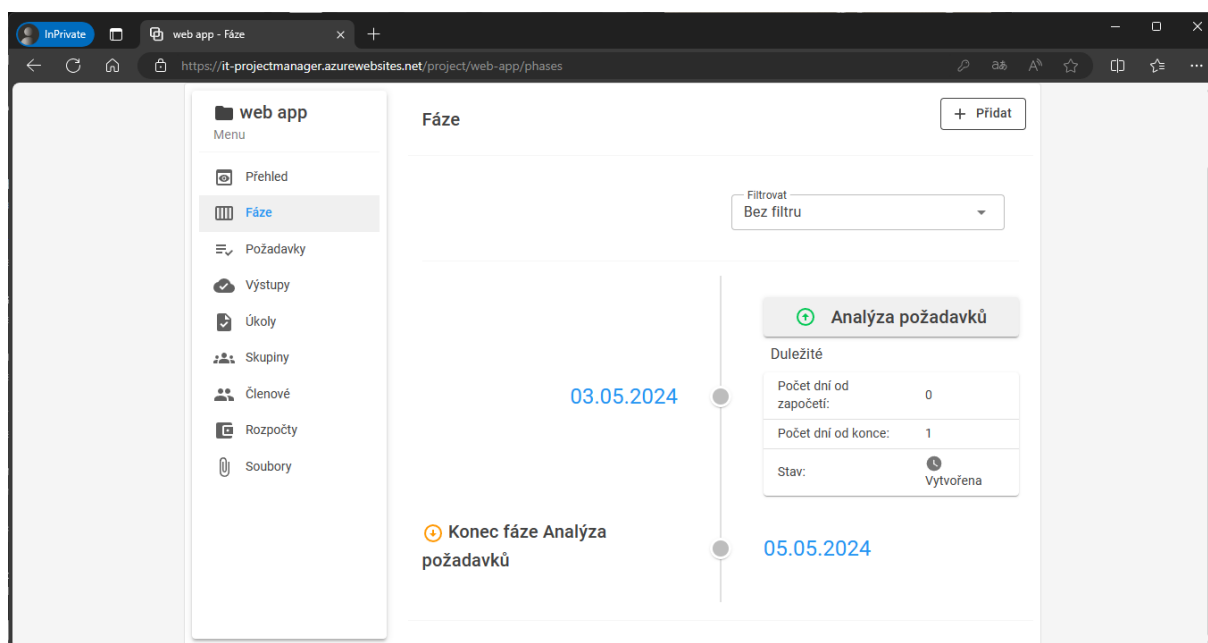
Obrázek 14: Ukázka aplikace – správa rozpočtu (zdroj vlastní)

K fázím dle finanční analýzy rozdělí dostupný rozpočet specifikovaný dříve. Fáze se mu budou následně projektovat do časové osy. Každou fázi lze přes hlavičku otevřít kliknutím. Tam manažer na záložce „Úkoly“ postupně přidá úkoly, které reflektují požadavky.



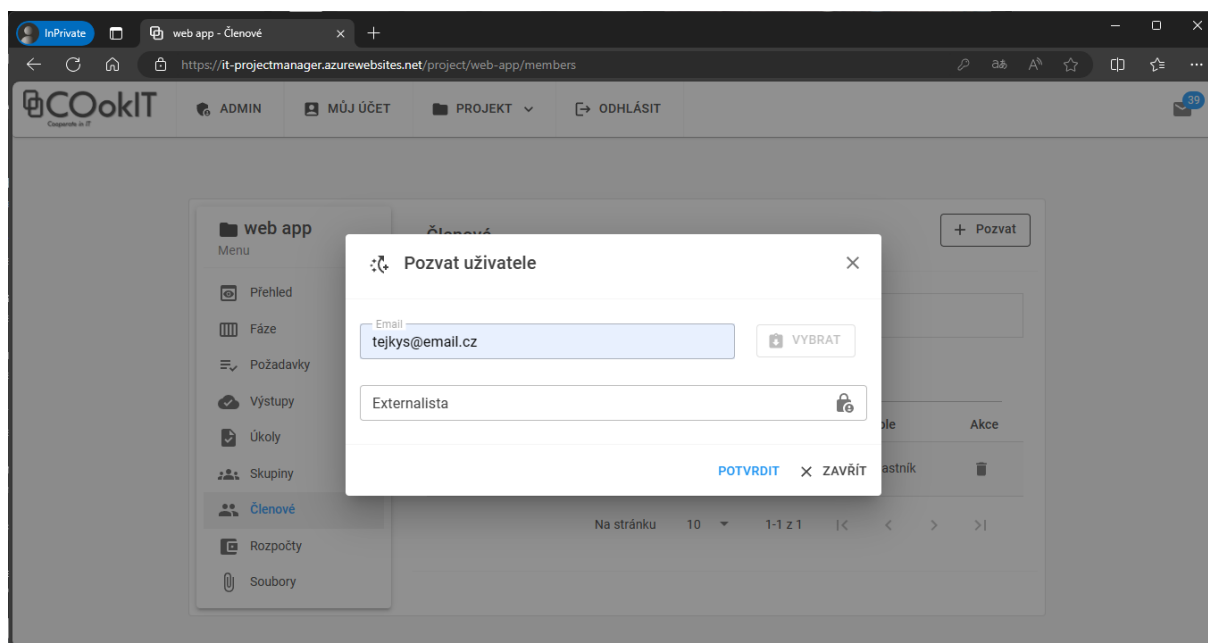
Obrázek 15: Ukázka aplikace – specifikace úkolu (zdroj vlastní)

Vedoucí uvede ke každému úkolu název, předpokládaný čas, náklady, prioritu, detailní popis s jasnými instrukcemi pro vývojáře a na záložce souborů přiloží dokument, kterým disponuje, s texty pro projektované komponenty.



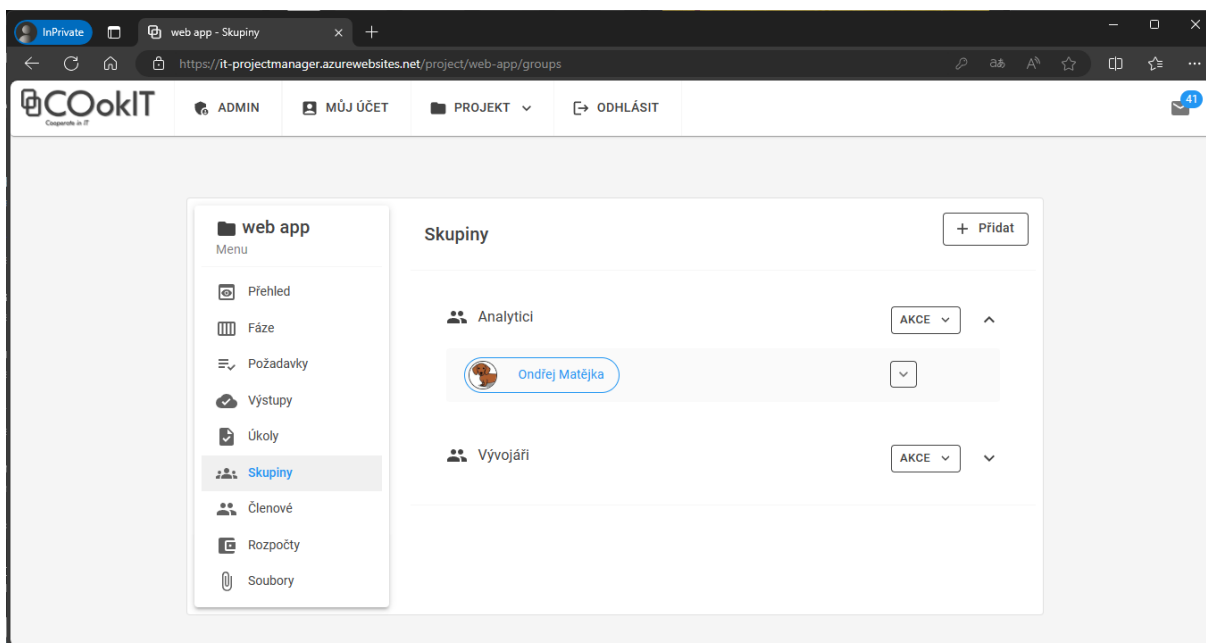
Obrázek 16: Ukázka aplikace – projekce fází projektu (zdroj vlastní)

Manažer nejdřív potřebuje zařadit interní pracovníky do své organizace. Přejde proto do „Můj účet“ a „Moje organizace“, kde vybere příslušnou organizaci, do které následně pozve zaměstnance přes emailovou adresu.



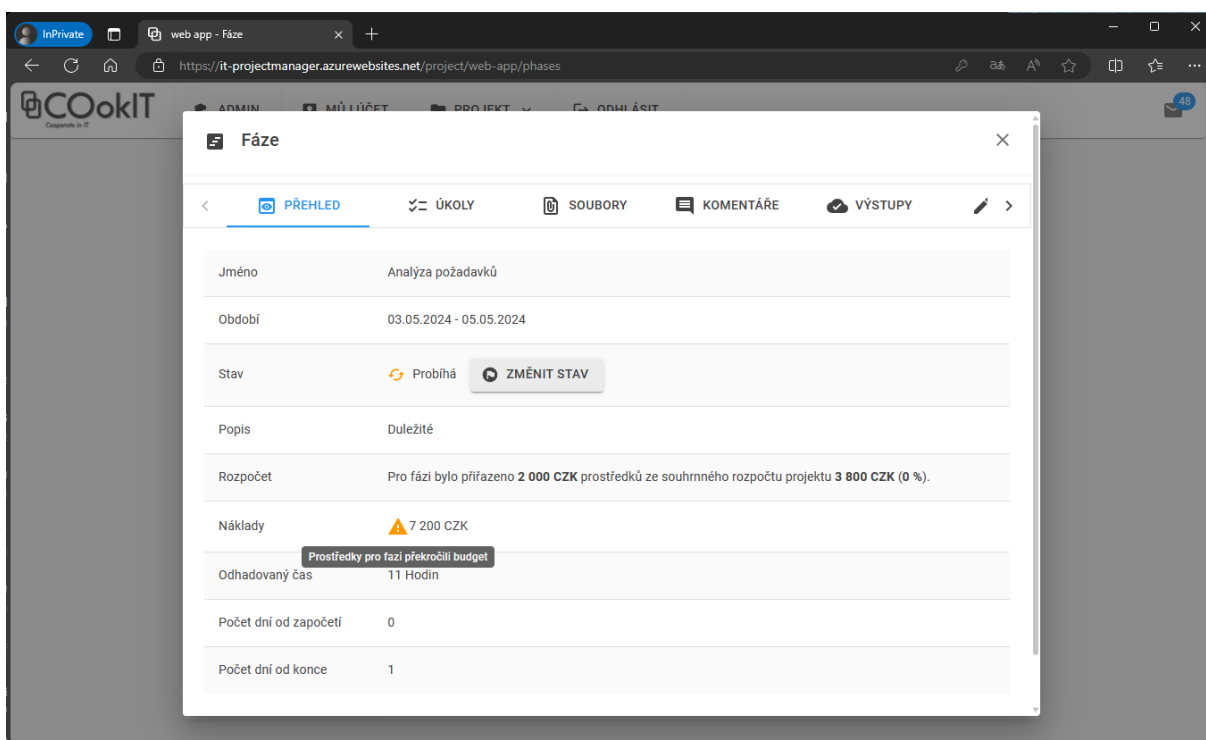
Obrázek 17: Ukázka aplikace – pozvání uživatele do projektu (zdroj vlastní)

Aby začlenil všechny účastníky do projektu, manažer přejde do modulu „Členové“ v projektu, kde pozve uživatele na základě jejich emailu a organizace, a tak jim přidělí roli-právo přejít na projekt. Nezapomene ani na zadavatele projektu, který je externistou. Ve stejném modulu manažer specifikuje role podle zodpovědností uživatelů v projektu. Jelikož potřebuje pro úkoly přidělit pracovní skupiny, které budou úkol realizovat, musí vedoucí v modulu skupin vytvořit toto seskupení lidí. Vytvoří skupinu „Vývojáři“ a do ní přiřadí příslušné účastníky. Následně manažer přejde buď přes fázi nebo položku projektového menu k úkolům a k nim dle rozdělení přidá skupiny zodpovědné za zpracování.



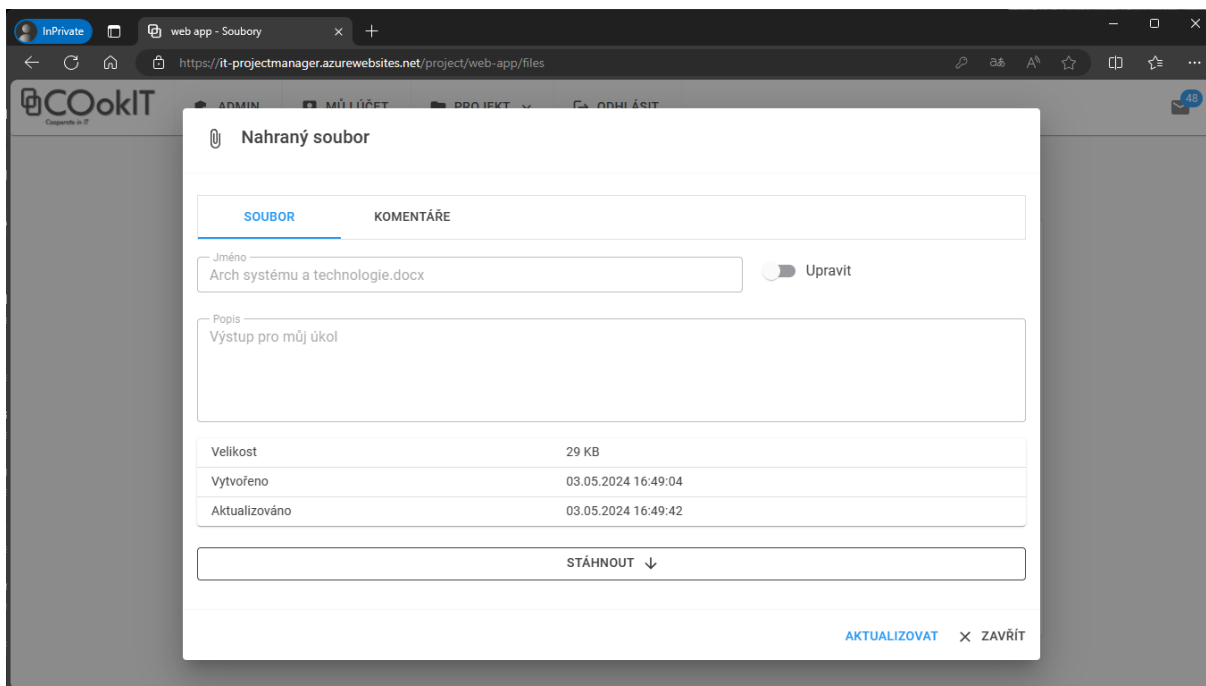
Obrázek 18: Ukázka aplikace – skupinové uspořádání (zdroj vlastní)

Po zpracování všech úkolů fáze manažer či zadavatel uzavírá fázi jako dokončenou, přidávají se dosažené cíle ve výstupech a v přehledu fáze se následně konzultují finanční a časové statistiky, jestli se vše vešlo do stanovených mezí.



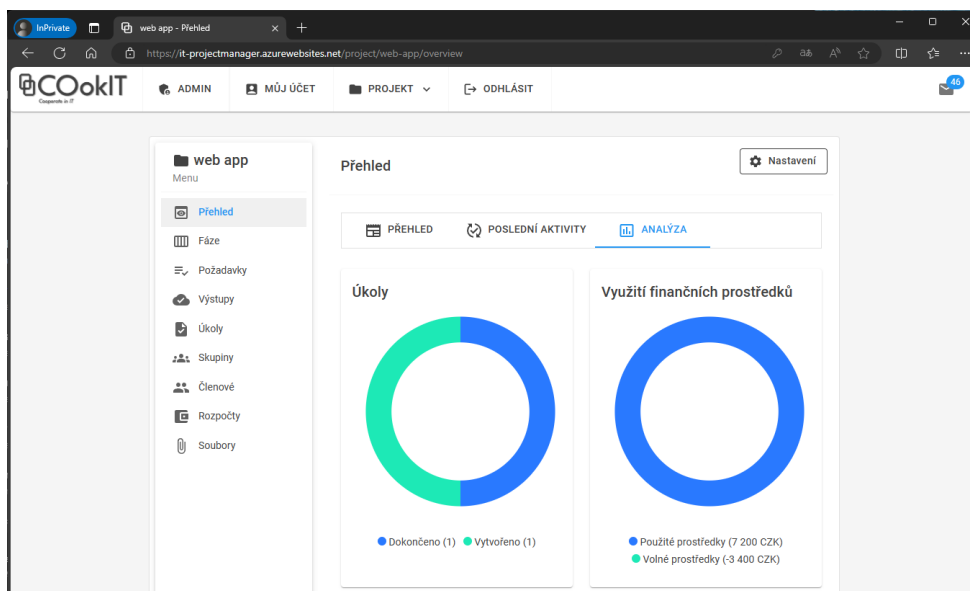
Obrázek 19: Ukázka aplikace – fáze (nejdůležitější průsečík částí projektu, zdroj vlastní)

Zadavatel přidává připomínky do komentářů fáze a manažer nahrává detailnější report do souborů.



Obrázek 20: Ukázka aplikace – správa datových souborů (zdroj vlastní)

Proces pokračuje a opakuje se dle fází, vedoucí sleduje aktivitu členů v přehledu projektu společně s grafy vizualizující stavy úkolů a finančních zdrojů.



Obrázek 21: Ukázka aplikace – vizualizace projektových prostředků (zdroj vlastní)

5.8 Budoucí rozšíření a zlepšení aplikace

Aplikace má potenciál růst a rozšiřovat se. Doposud byla vyvíjena v horizontu několika měsíců s omezenými pracovními zdroji. A jelikož projektové řízení nemá hranic, další rozšíření jsou jen otázkou kreativity a poptávky potenciálních klientů. Rámec ITIL pokrývá 34 praktik a jejich implementace v aplikaci není zdaleka kompletní.

Nabízí se rozšíření pro vizuální interpretace a analýzy. Jednou takovou implementací může být přímo Ganttův diagram. Dále řízení incidentů by mohlo usnadnit evidenci událostí s přesně danou strukturou pro detailnější řešení. Help Desk komponenta by rozšířila komunikaci mezi stranami, a to jak na straně aplikace samotné, tak z hlediska interního projektu. Řízení bezpečnosti by zavedlo dvoufázové ověřování a šifrování uložených dat jako jedno z možných rozšíření. Řízení dat by implementovalo export do známých standardizovaných formátů. Synchronizace a integrace dat by otevřela další možnosti propagace a extenze funkčnosti. Jednu z implementací by mohla zastat služba pro automatické nahrávání datumových údajů do oblíbeného kalendáře s popiskem případu. Tento přístup je již částečně vyhotoven, nicméně data se exportují do souboru ve formátu iCalendar, ten je potřeba ručně importovat do služeb kalendáře jako je Microsoft integrovaný do systému Windows, nebo Google, jež se interně propojuje s mobilním telefonem na operačním systému Android.

V krátkém horizontu by si aplikace zasloužila řadu re-designování, jelikož web se nachází ve výchozím vzhledu komponent MudBlazor s lehkými úpravami. To by se mohlo týkat jak tlačítek, tabulek, formulářů a obecné palety barev, tak i rozložení webu a komponent. Některá formulářová ovládání by šla nahradit přívětivějšími možnostmi, jako je Drag&Drop.

Vstupní stránka (takzvaná landing page) aktuálně pro aplikaci chybí. Zde by se angažovaly strukturované a výstižné informace o funkcionalitách a návodech. Takto uvedený úvodní obsah by pak šel propagovat ve webových vyhledávačích a podrobit procesu SEO.

Zlepšit by se musel i tarif pronájmu infrastruktury, na které aplikace běží. Aktuálně je software nasazen a provozován na cloudu Azure s nejmenšími parametry pro vývoj, nikoliv pro živý provoz. Zde se nabízí přechod na vlastní platformu, či jiného cloudového provozovatele, kde by byl dostupný větší prostor a flexibilita pro implementaci například vlastního monitorovacího subsystému bez nutnosti finančního vynaložení prostředků.

ZÁVĚR

Práce nejdříve přiblížila a zhodnotila přístupy k informačním projektům, ze kterých se následně vycházelo ve zpracování praktické úlohy. Cloudová webová aplikace pro řízení IT projektů byla vytvořena na základě analýzy a zpracování požadavků. Vývoj by šel asociovat se spirálovým modelem, kdy v rámci identifikace cílů na začátku projektu se identifikovaly hlavní cíle a požadavky aplikace. Zároveň se provedla první analýza rizik a určily se klíčové oblasti, které se budou pozorně sledovat během celého vývoje. Ve fázi analýzy a plánování se detailněji specifikovali požadavky a provedla se hlubší analýza rizik spojených s jednotlivými částmi aplikace. Zároveň se vytvořil plán pro další fáze vývoje. Během fáze implementace a testování se nasadily a testovaly jednotlivé části aplikace v rámci krátkých iterací. Každá iterace zahrnovala implementaci nových funkcionalit a následné testování a ladění. Po každé iteraci se provedlo hodnocení výsledků a reakce testera a vedoucího práce. Na základě těchto zpětných vazeb se rozhodlo o dalších krocích vývoje, kdy se buď pokračovalo v další iteraci, nebo se provedly úpravy na základě zjištěných nedostatků. Tento cyklus se opakoval po celou dobu vývoje, což umožnilo flexibilně reagovat na změny a neustále zlepšovat kvalitu aplikace. Spirálový model poskytl kvalitní rámec pro systematické řešení rizik a postupné zdokonalování aplikace v průběhu celého vývoje.

Vypracování softwarového řešení probíhalo bez zásadnějších problémů. Jedno z větších přitěží nastalo při zpracování komponenty pro RichText, která je řešena JavaScript knihovnou, jelikož Blazor by čistou implementaci v jeho rámci neumožnil. Synchronizace inicializace komponent v životním cyklu rámce se ale dostala do komplikací, když v době, kdy měla být komponenta textového pole již připravená dle dokumentace pomocí JS pro navázání obslužných rutin na funkční elementy, tomu tak nebylo. Řešením pro problém se stal callback, který čekal na iniciaci komponenty a následně dal vědět rámci, že je vše připraveno pro další zpracování.

Povedlo se úspěšně implementovat uživatelsky přívětivou cloudovou webovou aplikaci v moderním prostředí na základě specifikovaných požadavků, a tak splnit vytyčený cíl diplomové práce. V závěrečné kapitole autor popisuje možnosti dalšího rozvoje cloudové webové aplikace, kdy existuje potenciál růstu.

POUŽITÁ LITERATURA

1. **BREWER, Jeffrey L.** *Methods of It Project Management, Fourth Edition.* West Lafayette, Indiana : Purdue University Press, 2022. ISBN 9781612497907.
2. **CLARK, Hannah.** The Software Development Life Cycle (SDLC): 7 Phases and 5 Models. *The Product Manager.* [Online] 30. říjen 2023. [Citace: 23. listopad 2023.] <https://theproductmanager.com/topics/software-development-life-cycle/>.
3. **POLANSKÁ, Hana.** Trendy v přístupech k životnímu cyklu vývoje informačního systému (SDLC). [Online] 2008. [Citace: 23. listopad 2023.] https://dk.upce.cz/bitstream/handle/10195/29797/PolanskaH_Trendy%20v%20pristupech_PJ_2008.pdf?sequence=1.
4. **Freelo team.** FreeloCZ. *Co Je Projektový trojimperativ a jak ho použít V praxi.* [Online] 21. únor 2023. [Citace: 19. listopad 2023.] <https://www.freelo.io/cs/co-je-projektovy-trojimperativ-a-jak-ho-pouzit-v-praxi>.
5. **GUCKENHEIMER, Sam a PEREZ, Juan J.** *Efektivní softwarové projekty.* Brno : Zoner Press, 2007. ISBN 978-80-86815-62-6.
6. **MIRKHABIBOVA, Aziza, Bc.** COBIT 2019 a ITIL v4: Vzdalují se nebo přibližují? [Online] 2020. [Citace: 7. prosinec 2023.] https://vskp.vse.cz/81175_cobit_2019_aitil_v4_vzdujuji_se_nebo_priblizuji.
7. **WHITE, Sarah K. a GREINER, Lynn.** What is ITIL? Your guide to the IT Infrastructure Library. *CIO.* [Online] 16. březen 2022. [Citace: 7. prosinec 2023.] <https://www.cio.com/article/272361/infrastructure-it-infrastructure-library-til-definition-and-solutions.html>.
8. **AXELOS.** *ITIL® Foundation: ITIL 4 edition.* Norwich : TSO, 2019. 9780113316076.
9. **HELPDESK, Carlos.** The ITIL4 four dimensions and incident management. *Medium.* [Online] 6. prosinec 2023. [Citace: 8. březen 2024.] <https://medium.com/@carloshelpdesk/the-til4-four-dimensions-and-incident-management-1dc68958f887>.
10. **The Knowledge Academy.** Four Dimensions of ITIL 4. *Medium.* [Online] 31. červenec 2023. [Citace: 10. březen 2024.] <https://medium.com/@theknowledgeacademy/four-dimensions-of-til-4-47ee4df7f99b>.
11. **QRPI International.** What is itil? itil definition, the SVS and the 4 dimensions. [Online] 14. duben 2022. [Citace: 13. březen 2024.] <https://www.qrpinternational.be/blog/it-governance-and-service-management/what-is-til/>.
12. **LIN, Hause.** Implement Kanban Principles to Improve Your Workflow and Productivity. *Medium.* [Online] 17. červen 2021. [Citace: 18. březen 2024.] <https://medium.com/an-idea/implement-kanban-principles-to-improve-your-workflow-and-productivity-dc3a97bcd188>.

13. **Aneta.** Co je Kanban a jak s ním efektivně pracovat. *Lamael*. [Online] [Citace: 18. březen 2024.] <https://www.lamael.cz/co-je-kanban-a-jak-s-nim-efektivne-pracovat/>.
14. **DO, Doanh.** The Five Principles of Lean. *The Lean Way*. [Online] 5. srpen 2017. [Citace: 20. březen 2024.] <https://theleanway.net/The-Five-Principles-of-Lean>.
15. **Green Fox Academy.** Vše, co musíš vědět o DevOps. [Online] 22. srpen 2022. [Citace: 18. březen 2024.] <https://www.greenfoxacademy.cz/post/vse-co-musis-vedet-o-devops>.
16. **FLOWii.** Co je to KPI a jak jej nastavit. [Online] 4. březen 2021. [Citace: 20. březen 2024.] <https://www.flowii.com/cz/blog/co-je-to-kpi-a-jak-jej-nastavit>.
17. **HUGHES, Bob.** *Project Management for IT-Related Projects: 3rd edition*. Wiltshire : BCS Learning and Development Ltd, 2019. ISBN 9781780174860.
18. **Wikimedia Foundation.** YouTrack. *Wikipedia*. [Online] 9. březen 2024. [Citace: 2024. duben 6.] <https://en.wikipedia.org/wiki/YouTrack>.
19. **CORRALES, Enrique.** YouTrack Review. *Developer.com*. [Online] 5. srpen 2023. [Citace: 6. duben 2024.] <https://www.developer.com/project-management/youtrack-review/>.
20. **DRAKE, Nate a PROBST, Collin.** Trello review. *TechRadar*. [Online] 3. únor 2022. [Citace: 6. duben 2024.] <https://www.techradar.com/reviews/trello>.
21. **Any.do Team.** A to-do list you'd actually use. *Any.do*. [Online] © 2024. [Citace: 3. květen 2024.] <https://www.any.do/to-do-list/>.
22. **Nextcloud GmbH.** About Nextcloud. *NextCloud*. [Online] © 2022 - 2024. [Citace: 2. květen 2024.] <https://nextcloud.com/about/>.
23. **BIGELOW, Stephen J.** What is Microsoft Azure and How Does It Work? *TechTarget*. [Online] 19. říjen 2022. [Citace: 14. duben 2024.] <https://www.techtarget.com/searchcloudcomputing/definition/Windows-Azure>.
24. **HARTINGER, David.** ITnetwork.cz. *Lekce 1 - Úvod do ASP.NET*. [Online] © 2024. [Citace: 14. duben 2024.] <https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net>.
25. **Pragimtech.** What is Blazor WebAssembly. *Pragim Technologies*. [Online] © 2020. [Citace: 15. duben 2024.] <https://www.pragimtech.com/blog/blazor-webAssembly/what-is-blazor-webassembly/>.
26. **Learn.** ASP.NET Core Blazor. *Microsoft Learn*. [Online] 9. únor 2024. [Citace: 15. duben 2024.] https://learn.microsoft.com/cs-cz/aspnet/core/blazor/?view=aspnetcore-8.0&WT.mc_id=dotnet-35129-website.
27. **ProjectManager.** What Is a Gantt Chart? *ProjectManager*. [Online] © 2024. [Citace: 13. duben 2024.] <https://www.projectmanager.com/guides/gantt-chart>.

SEZNAM PŘÍLOH

PŘÍLOHA A: Zdrojový kód aplikace

PŘÍLOHA A: Zdrojový kód aplikace

Příloha v podobě formátu ZIP obsahuje implementační řešení celé aplikace pro Visual Studio 2022 s dílčími projekty.