



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

# NÁVRH A IMPLEMENTACE AUTOMATIZACE TESTOVÁNÍ V SOFTWAREM VÝVOJI

IMPLEMENTATION OF TESTING AUTOMATION IN SOFTWARE DEVELOPMENT

## DIPLOMOVÁ PRÁCE

MASTER'S THESIS

## AUTOR PRÁCE

AUTHOR

Bc. Pavla Panáková

## VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lukáš Novák, Ph.D.

BRNO 2021

# Zadání diplomové práce

Ústav:	Ústav informatiky
Studentka:	<b>Bc. Pavla Panáková</b>
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Informační management
Vedoucí práce:	<b>Ing. Lukáš Novák, Ph.D.</b>
Akademický rok:	2020/21

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

## **Návrh a implementace automatizace testování v softwarovém vývoji**

### **Charakteristika problematiky úkolu:**

Úvod  
Vymezení problému a cíle práce  
Teoretická východiska práce  
Analýza problému a současné situace  
Vlastní návrhy řešení, přínos návrhů řešení  
Závěr  
Seznam použité literatury  
Přílohy

### **Cíle, kterých má být dosaženo:**

Cílem práce je analyzovat potřeby a požadavky na testování softwaru pro vybranou organizaci, vybrat vhodné řešení a navrhnout postup implementace.

### **Základní literární prameny:**

GALIN, Daniel a Anna HAVLÍČKOVÁ. Software quality assurance: průvodce testováním. 1. vyd. New York: Pearson Education Limited, 2004. 590 p. ISBN 02-017-0945-7.

PATTON, Ron. Testování softwaru. Praha: Computer Press, 2002. ISBN 80-7226-636-5.

ROUDENSKÝ, Petr a Anna HAVLÍČKOVÁ. Řízení kvality softwaru: průvodce testováním. Brno: Computer Press, 2013. ISBN 978-80-251-3816-8.

SVOZILOVÁ, Alena. Projektový management. 2., aktualiz. a dopl. vyd. Praha: Grada, 2011. ISBN 978-80-247-3611-2.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

---

Mgr. Veronika Novotná, Ph.D.  
ředitel

---

doc. Ing. Vojtěch Bartoš, Ph.D.  
děkan

## **Abstrakt**

Tato diplomová práce se zabývá zavedením testovacího procesu do informačního oddělení společnosti IREKS ENZYMA s.r.o. V rámci práce jsou nejdříve představena teoretická východiska potřebná pro pochopení dané problematiky. Následně je provedena analýza současného stavu společnosti s důrazem na testování softwaru. Ve třetí části je pak proveden výběr testování, jehož implementace je popsána pomocí technik projektového managementu.

## **Klíčová slova**

Testování softwaru, projekt, IT, automatizace, implementace

## **Abstract**

This Master's thesis deals with the introduction of the testing process into the information department of IREKS ENZYMA s.r.o. In this thesis is firstly introduced the theoretical foundation necessary for the understanding of this topic and subsequently is analysed the current state of the company with an emphasis on the software testing. The third part of the thesis focuses on the selection of the testing automation and its implementation, which is described using the techniques of project management.

## **Key words**

Software testing, project, IT, automation, implementation



### **Bibliografická citace**

PANÁKOVÁ, Pavla. Návrh a implementace automatizace testování v softwarovém vývoji [online]. Brno, 2021 [cit. 2021-05-04]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/133710>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Lukáš Novák.

### Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracovala jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušila autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 30.4.2021

.....

**Bc. Pavla Panáková**

## **Poděkování**

V první řadě bych chtěla poděkovat panu Ing. Lukáši Novákovi, Ph.D. za obětavou pomoc a cenné odborné rady při tvorbě diplomové práce. Dále bych chtěla poděkovat společnosti IREKS ENZYMA s.r.o., která mi poskytla informace potřebné pro tvorbu práce. A v neposlední řadě samozřejmě svým nejbližším za velkou podporu, při psaní i v průběhu celého studia.

# OBSAH

<b>ÚVOD.....</b>	<b>12</b>
<b>CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ.....</b>	<b>13</b>
Cíl práce.....	13
Metody a postupy zpracování.....	13
<b>1    TEORETICKÁ VÝCHODISKA PRÁCE.....</b>	<b>15</b>
1.1    Definice softwaru.....	15
1.2    Testování softwaru.....	15
1.2.1    Softwarové chyby.....	16
1.2.2    Proč testovat.....	17
1.3    Testovací techniky.....	17
1.3.1    Manuální a automatické testování.....	17
1.3.2    Statické a dynamické testování.....	19
1.3.3    Testování černé a bílé skříňky.....	19
1.3.4    Konfirmační a regresní testování.....	20
1.4    Úrovně testování.....	20
1.4.1    Testování jednotek.....	21
1.4.2    Integrační testování.....	22
1.4.3    Systémové testování.....	22
1.4.4    Akceptační testování.....	22
1.4.5    Regresní testování.....	23
1.5    Typy testů.....	23
1.6    Analytické prostředky.....	25
1.6.1    PESTLE analýza.....	25
1.6.2    Porterův model konkurenčních sil.....	25

1.6.3	Analýza „7S“ .....	26
1.6.4	Organizační struktura.....	28
1.6.5	SWOT analýza.....	29
1.7	EPC diagram .....	30
1.8	Metody projektového managementu.....	31
1.8.1	Logický rámec .....	32
1.8.2	WBS.....	32
1.8.3	RACI matice .....	33
1.8.4	Metoda RIPRAN.....	33
1.8.5	Genttovy diagramy .....	33
1.8.6	Rozpočet .....	34
<b>2</b>	<b>ANALÝZY SOUČASNÉHO STAVU .....</b>	<b>35</b>
2.1	Představení společnosti .....	35
2.1.1	Formální údaje společnosti .....	36
2.1.2	Organizační struktura.....	36
2.2	Analýza vnějšího prostředí společnosti.....	37
2.2.1	PESTLE analýza .....	37
2.2.2	Porterův model konkurenčních sil společnosti .....	39
2.3	Analýza vnitřního prostředí společnosti.....	41
2.3.1	Analýza 7S.....	41
2.4	SWOT analýza .....	44
2.5	Analýza vývojového procesu .....	44
2.6	Zhodnocení současného stavu.....	46
<b>3</b>	<b>VLASTNÍ NÁVRH ŘEŠENÍ.....</b>	<b>47</b>
3.1	Výběr testovacího softwaru.....	47
3.1.1	Specifikace požadavků .....	47

3.1.2	HP QuickTest Professional .....	48
3.1.3	Selenium WebDriver .....	48
3.1.4	Conformiq Designer .....	49
3.1.5	TestComplete .....	50
3.1.6	NUnit .....	50
3.1.7	Souhrnné hodnocení .....	52
3.2	Vývojový proces s využitím NUnit.....	53
3.3	Identifikační listina projektu .....	54
3.4	Milníky projektu.....	55
3.5	Logický rámec.....	56
3.6	WBS projektu.....	58
3.7	Projektový tým (RACI).....	62
3.7.1	RACI matice projektu .....	62
3.8	Analýza rizik projektu.....	64
3.8.1	Identifikace a zhodnocení rizik.....	65
3.8.2	Opatření pro snížení rizik .....	66
3.8.3	Shrnutí analýzy rizik.....	67
3.9	Časová analýza projektu .....	67
3.10	Ekonomické zhodnocení projektu .....	71
3.10.1	Návratnost investice.....	72
3.11	Přínosy projektu.....	72
	<b>ZÁVĚR .....</b>	<b>74</b>
	<b>SEZNAM ZDROJŮ .....</b>	<b>76</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>79</b>
	<b>SEZNAM TABULEK.....</b>	<b>80</b>
	<b>SEZNAM GRAFŮ .....</b>	<b>81</b>

<b>SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ .....</b>	<b>82</b>
---	-----------

## ÚVOD

V posledních několika letech se zvyšuje nejen počet webových a mobilních platforem, ale i jejich implementační náročnost. Aktuální statistika počtu webových stránek ve světě udává celkem přes 1,8 bilionů webových stránek a přes 4 miliony mobilních aplikací [1]. Zvyšující tlak na rychlost vývoje a nabírání juniornějších vývojářů/vývojářek sebou přináší nová rizika. Jedním z nich je počet programátorských chyb v programu nebo aplikaci.

Testování softwaru předchází tomu, aby se tyto chyby dostaly do finálního produktu. Testování je nepostradatelnou součástí každého životního cyklu vývoje softwaru. Byly vytvořeny různé testovací postupy, které popisují, jak zlepšit kvalitu vyvíjených systémů.

Tím, jak se vyvíjí nové technologie a programovací jazyky, tak se rozvíjí a mění i testovací techniky softwaru. Testovací techniky se mohou rozdělit do několika kategorií, například podle fáze vývoje, ve kterém se software vyskytuje, nebo podle toho, kterou část softwaru chceme testovat. Každé softwarové odvětví se zabývá jinými typy testů, podle toho, jak jsou nastaveny požadavky na kvalitu systému.

V posledních letech se rozvíjí zejména testování mobilních aplikací, testování využívající cloud, virtualizace nebo využívání automatizačních nástrojů při testování.

Když se určitá část testování zautomatizuje, je práce ve vývojovém prostředí daleko rychlejší, snadnější a účinnější. Pro automatizované testování existuje několik nástrojů, které budou dále popsány v diplomové práci.



# **CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ**

## **Cíl práce**

Hlavním cílem této diplomové práce je výběr a zavedení vhodného automatizovaného testování softwaru do vývojového oddělení pro společnost, která vyrábí pekárenské směsi a sídlí v Brně, tak aby splňoval veškeré požadavky a měl pro společnost nějaký přínos.

## **Metody a postupy zpracování**

V rámci diplomové práce jsou zpracovány tři zásadní oblasti, tj. teoretická východiska práce, analýza současného stavu a vlastní návrh řešení.

V první zpracované oblasti jsou popsána teoretická východiska potřebná pro pochopení problematiky probírané v rámci práce. Tyto informace se týkají převážně testování softwaru, dále jsou zde popsány analytické prostředky a prostředky projektového managementu využívané v následujících dvou kapitolách.

V další fázi je společnost analyzována. Před zahájením samotné analýzy proběhly schůzky s představitelem společnosti, na kterých byl prodiskutován současný stav společnosti ve všech potřebných směrech. Na základě informací získaných z těchto schůzek mohla být zpracována konkrétní analýza.

V první řadě jsou popsány obecné informace o společnosti, jako např. sídlo, vedení, obor podnikání, organizační struktura apod.

Pro analýzu vnějšího prostředí společnosti jsou použity analýzy PESTLE a Porterův model konkurenčních sil. Prostřednictvím analýza „7S“ je provedena vnitřní analýza společnosti. SWOT analýza pak shrnuje jednotlivé výstupy z těchto analýz. Kromě analýz je zde popsána procesní mapa vývojového procesu.

Poslední fázi je možné rozdělit do dvou etap. Nejprve v první etapě je představeno a posouzeno několik testovacích metod od různých dodavatelů. Tyto testovací systémy jsou mezi sebou následně porovnány a s ohledem na požadavky společnosti a vhodnost použití je vybrán systém pro implementaci.

V druhé etapě je popsána samotná implementace vybraného řešení s využitím metod a postupů projektového managementu, jako je např. logický rámec, WBS, RACI matice, Ganttův diagram apod.

V závěru práce je projekt zhodnocen z ekonomického hlediska a také jsou hodnoceny celkové přínosy pro společnost. V návaznosti na to jsme schopni říct, zda byl cíl splněn.

# 1 TEORETICKÁ VÝCHODISKA PRÁCE

V následující kapitole jsou popsány základní teoretické poznatky, které souvisí s oblastmi popsanými v této práci. Všechny tyto poznatky jsou důležité pro správné pochopení zpracované problematiky.

## 1.1 Definice softwaru

Při charakteristice pojmu software vycházíme z definice, která byla uveřejněna v roce 1991 institucí IEEE. Za software jsou považovány počítačové programy a jeho související dokumentace. Takové produkty jsou vytvářeny nejen pro konkrétního zákazníka, ale také pro širokou veřejnost [2].

Tuto definici posléze přejímá a rozšiřuje Mezinárodní organizace pro normalizaci a vyskytuje se ve všech publikovaných normách, vztahujících se k softwaru.

Software je rovněž označován za programové vybavení. Záměrně není programové vybavení vázané na počítače, jelikož se v dnešní době se programové vybavení váže i na jiné zařízení jako je notebook, tablet, mobil a dokonce i tzv. informační technologičtí nositelé, mezi které řadíme chytré brýle, hodinky či sluchátka. Ve všech těchto zařízeních je jistý druh softwaru, který spolu s hardwarem tvoří informační technologie [2].

Software lze členit na dvě základní kategorie. První kategorie tvoří systémový software, který zajišťuje samotný chod zařízení, zpravidla se jedná o operační systém. Druhou kategorií tvoří aplikační software, se kterým se můžeme setkat v podobě kancelářských, grafických či vývojových programů [2].

## 1.2 Testování softwaru

Pod pojmem testování softwaru se často vybaví manuální procházení aplikace a klikání na nabídky za účelem odhalení defektu. Správné testování je složitější a je pro něj zavedeno několik definic.

Testování softwaru se provádí na základě specifikace produktu, která je řádně prodiskutována a zdokumentována vývojovým týmem. Specifikace podrobně popisuje produkt, např. jak bude vypadat, jak se bude chovat, co bude nebo naopak nebude dělat. Je důležité brát v potaz především koncové uživatele a jejich požadavky na produkt.

Testování slouží jako jeden z nástrojů k získávání informací o kvalitě softwaru, a to zejména k reportování nalezených problémů – tzv. softwarových chyb [3].

Cílem testování je tedy co v nejkratším čase odhalit co nejvíce softwarových chyb a zjistit, zda se daný produkt chová podle produktové specifikace, zda je schopen čelit nestandardním podmínkám, a to jak uživatelským, tak i softwarovým, hardwarovým i systémovým [4].

### 1.2.1 Softwarové chyby

Softwarovou chybu definoval Ron Patton následovně:

*„O softwarové chybě hovoříme právě tehdy, pokud je splněna jedna nebo více z následujících podmínek:*

- *Software nedělá něco, co by podle specifikace produktu dělat měl.*
- *Software dělá něco, co by podle údajů specifikace produktu dělat neměl.*
- *Software dělá něco, o čem se produktová specifikace nezmiňuje.*
- *Software nedělá něco, o čem se produktová specifikace nezmiňuje, ale měla by se zmiňovat.*
- *Software je obtížné srozumitelný, těžko se s ním pracuje, je pomalý, nebo – podle názoru testera softwaru – jej koncový uživatel nebude považovat za správný.“ [4]*

Význam a využívání softwarových chyb vždy závisí na konkrétním firemním prostředí.

Proto bylo zavedeno několik obecných pojmů, které se dodržují:

- **Error (Chyba)** – Příčinou je lidská činnost. Může se jednat o chybu v kódu, jako je například chyba v sintaxi nebo ve volání proměnné. Taktéž se může jednat o špatně nastaveném připojení k databázi.
- **Defect (Závada)** – Defekt je formálním názvem bugu (softwarové chyby). K defektu dochází z důvodu nějaké chyby v kódu nebo v dokumentu, která způsobí odchylku od nečekaného výsledku. Defekt je tedy nalezená chyba, pokud je potvrzena vývojářem nebo manažerem, stává se Bugem.
- **Failure (Selhání)** – Selhání je neschopnost systému nebo jeho součástí vykonávat požadovanou funkci. Čili se jedná o závadu u koncového uživatele. Může být způsobeno vlivem prostředí jako je radiace, magnetismu nebo i třeba vlhkost [5].

### 1.2.2 Proč testovat

V dnešní době je testování nezbytnou součástí každého životního cyklu vývoje softwaru. Vzhledem k velkému rozsahu a složitosti softwaru by jeho selhání mohlo mít až katastrofické následky. Je potřeba eliminovat veškerá rizika na minimum a tím zajistit požadovanou kvalitu softwaru.

Žádný software není perfektní, jelikož je napsán lidmi a lidé často dělají chyby. Příklady můžeme najít kdekoliv, stačí se podívat do historie, kolik pádů letadel, selhání družic, dopravních nehod bylo způsobeno kvůli softwarové chybě.

Konkrétním případem je například raketa Ariane 5 z roku 1996, která nesla čtyři družice, jež měly studovat interakci magnetického pole Země se solárním větrem. Bohužel kvůli chybě v naváděcím a řídicím systému, který se po pár vteřinách letu sám vypnul, předal tento systém řízení záložnímu systému, který nesl taktéž tuto chybu. Raketa nemohla dál letět a zřítla se do moře [6].

Softwarová chyba tedy může vést od jistého nepohodlí při používání přes nadbytečné finanční náklady až ke ztrátám na životech [4].

## 1.3 Testovací techniky

V této podkapitole bude vyjmenováno několik přístupů, jak lze k testování přistoupit. Testování lze rozdělit dle různých atributů.

### 1.3.1 Manuální a automatické testování

Testy lze rozlišit podle toho, zda jsou prováděny softwarem, nebo člověkem.

#### Manuální testování

Vhodnější využít manuální testování se vyplácí, pokud si test vyžaduje lidské ohodnocení a úsudek. Při testování bere tester na sebe roli koncového uživatele ve snaze odhalit nějaké nežádoucí chování nebo přímo bug v aplikaci.

Výhody

- Nižší krátkodobé náklady.
- Větší pravděpodobnost odhalení problémů reálného uživatele.
- Flexibilita – změny v návrhu testu se dají otestovat rychleji a jednodušeji [7].

## Nevýhody

- Některé úlohy se obtížněji provádějí manuálně – například simulování více uživatelů anebo testování rozhraní na nízké úrovni.
- Testování může být nudné a stále se opakující – při vyplňování stále stejných dat a formulářů může tester ztratit soustředění a stát se více náchylným na přehlédnutí chyby.
- Manuální testy nelze znovu použít – pokud je provedena nějaká změna v aplikaci, musí se test provést znovu. To je značná ztráta času [7].

## **Automatické testování**

Při automatickém testování tester vytváří automatické skripty a využívá různé softwarové nástroje pro testování. Proces automatického testování zahrnuje automatizaci nějakého manuálního testovacího procesu. Automatizace testů umožňuje jejich opětovné spuštění a rychlejší provedení.

## Výhody

- Rychlost – nástroje pro automatizované testování dokážou provádět testy až tisíckrát rychleji než manuální testeři.
- Efektivita – díky zkrácení času nezbytného k provádění testů zbývá více času na plánování testů a psaní nových testových případů.
- Správnost a přesnost – po ručním vyzkoušení několika stovek testových případů se zákonitě oslabí testerova pozornost a ten pak může začít dělat chyby ve své práci. Testovací nástroj provede stejné testy pokaždé naprosto přesně.
- Neúnavnost – testovací nástroje mohou klidně běžet celou noc, zatímco tester spí.
- Nákladově výhodné – v krátkodobém hledisku jsou nástroje automatického testování nákladné, dlouhodobě však náklady šetří [7; 4].

## Nevýhody

- Nástroje mohou být drahé – počáteční investice do nákupu nástrojů automatického testování může být vysoká.
- Testování může být stále velmi časově náročné.

- Omezení nástrojů – přestože automatické testy dokážou odhalit většinu bugů v systému, mají své omezení. Nemohou například testovat vizuální aspekty jako barva obrázku anebo velikost písma. Nežádoucí změny těchto parametrů mohou být odhaleny pouze manuálním testováním [4; 7].

### **1.3.2 Statické a dynamické testování**

Na základě toho, zda je k testování nutné spuštění aplikace, rozlišujeme statické a dynamické testy.

#### **Statické testování**

Statické testování nevyžaduje běh aplikace. Využívá se zejména v prvních fázích vývoje, kdy ještě není k dispozici funkční aplikace. Vhodné využití statických testů je např. na revizi programového kódu nebo pro kontrolu a ověřování specifikace požadavků [4].

#### **Dynamické testování**

Dynamické testování na rozdíl od statického vyžaduje spuštění aplikace. Využívají se v pozdějších fázích vývoje, kdy už je k dispozici spustitelná aplikace. Ověřuje, zda program běží správně. Testuje chování programu a porovnává vstupy s očekávanými výstupy [4].

### **1.3.3 Testování černé a bílé skřínky**

Obecně jsou uznávané 3 základní přístupy k testování software. Pojmy Black-box a White-box definuje již Ron Patton. Pojem Grey-box je pak určitou kombinací prvních dvou zmíněných přístupů [4].

#### **Černá skříňka (Black box)**

Při testování černé skřínky se vyznačuje tím, že tester nemá přístup k programovému kódu. Software si lze představit jako černou skříňku, jejíž obsah není zvenčí viditelný. Jak systém přesně pracuje s daty neznáme, ale pouze pozorujeme, jaký výsledek získáme po vložení vstupních dat. Testování softwaru v tomto případě spočívá na testovacích scénářích, které definují, jak má tester postupovat při testování [4].

Výhodou je rychlost a snadnost, jelikož tester nemusí disponovat znalostmi konkrétního programovacího jazyka. Možnou nevýhodou je testovací scénář, který nemusí pokrýt potenciální chyby [4].

### **Bílá skříňka (White box)**

U testování technikou bílé skříňky naopak známe vnitřní strukturu softwaru a máme k dispozici zdrojový kód. Testování je značně lepší, neboť můžeme testovat všechny průchody zdrojovým kódem, zadání neočekávaných vstupních hodnot nebo odhalit nežádoucí části kódu [4].

Bílá skříňka bývá náročnější z pohledu analýzy zdrojového kódu, neboť je důležité mu porozumět. Naopak díky tomu získáme vyšší kvalitu kódu výsledného produktu [4].

### **Šedá skříňka (Grey box)**

Testování šedé skříňky je přístup kombinující oba předchozí. Tester při návrhu a provádění testů zná částečně vnitřní strukturu programu. Princip testování je velmi jednoduchý, neboť tester ví, jak produkt funguje uvnitř a poté ho může lépe otestovat zvenku [8].

#### **1.3.4 Konfirmační a regresní testování**

##### **Konfirmační testy**

Konfirmační testování slouží pro kontrolu nových funkcí nebo vlastností softwaru. Konfirmační testy se využívají ve všech etapách testování. Nutné je zavedení dokumentace, která popisuje nové funkce a vlastnosti softwaru a je důležitá pro správné provedení testů [3].

##### **Regresní testy**

Regresní testy se využívají pro opakované testování funkcí a vlastností softwaru. Ideálně jsou prováděny vždy se změnou softwaru. Slouží k ověření, zda se při provedení změny nebo implementaci nových vlastností softwaru nevyskytly žádné další chyby v dosud fungujících částech aplikace. Regresní testy jsou velmi rozšířeny a je vhodné je automatizovat [3].

### **1.4 Úrovně testování**

Testovací fáze životního cyklu vývoje softwaru jsou taktéž nazývány testovací úrovně. Název jednotlivých úrovní napovídá, o jaké testování se jedná a na jaké problémy se nejčastěji zaměřuje.



Mezi nejzákladnější úrovně testování se řadí:

- testování jednotek (Unit testing),
- integrační testování (Integration testing),
- systémové testování (System testing),
- akceptační testování (Acceptance testing) [3].

Zmíněné testování prováděné na jednotlivých úrovních je dle V-modelu. V-model vychází z modelu vodopádu. Důležitým rysem V-modelu je účast testování od počátku projektu, neboť výstup z každé úrovně prochází ověřením [3; 4].



**Obrázek č. 1: V-model** (Zdroj: vlastní zpracování dle 8)

#### 1.4.1 Testování jednotek

Testování jednotek lze chápat jako proces testování co možná nejmenší testovatelné součásti softwaru – tzv. jednotky. Unitní testování provádí ověření, zda je napsaný kód spustitelný. Kromě jiného provádí i kontrolu shody se specifikacemi. Nejčastěji testování provádí sám programátor, který si ověří správnost své implementace. Nalezené chyby se v této fázi nezaznamenávají. Cílem je otestovat každou jednotku nezávisle na ostatních a ověřit, zda její zpracování a chování je správné [3].

Výhodou unitního testování je, že chyby jsou nalezeny velmi brzy a snižují se tedy náklady na jejich opravu. Nevýhodou může být fakt, že programátor testuje svůj vlastní kód, a tudíž je možné, že příslušný test navrhne tak, aby vyhovoval napsanému kódu, ale ne nutné specifikacím. Ovšem dá se to vyřešit druhým programátorem, který není autorem kódu [3].

#### **1.4.2 Integrovační testování**

Jakmile jsou všechny jednotky napsány, další etapa spočívá v sestavení systému. Toto sestavení se nazývá integrace a zahrnuje budování celku z menších částí. Cílem integračních testů je ověřit, jestli spolu jednotlivé moduly či komponenty softwaru spolupracují správně [3].

Integrovační testy lze provádět na různých úrovních. Obecně se rozlišuje intra systémové a inter systémové integrovační testování. U intra systémového integrovačního testování jde především o integraci modulů do funkčního systému, zatímco v druhém případě jde o integraci dvou a více systémů do jednoho celku [3].

#### **1.4.3 Systémové testování**

Systémové testování prověřuje aplikaci jako celek. Zda odpovídá požadavkům a specifikaci zákazníka. Jde o poslední možnost nalézt a opravit chyby dřív, než by je objevil uživatel. Tyto testy bývají prováděny až v pozdějších fázích vývoje softwaru. Simuluje se běžná práce uživatele se systémem. Jsou zde prováděny rozsáhlé testovací scénáře procházející skrz celý systém. Toto testování už převážně vykonává specializovaný tým [3].

Systémové testování je první fází, kde je aplikace testována jako celek. Aplikace je otestována důkladně k ověření, že splňuje funkční i nefunkční požadavky. Aplikace je testována v prostředí, které je velmi blízké produkčnímu, kde bude aplikace nasazena [3; 9].

#### **1.4.4 Akceptační testování**

Akceptační testování je poslední etapa v procesu testování před uvedením systému do provozu a je prováděno na straně zákazníka. Cílem akceptačních testů je zjistit, zda produkt splňuje všechna kritéria, která jsou předem dohodnutá mezi zákazníkem a dodavatelem [3].

- **Alfa testování** – Probíhá ve vývojovém prostředí, testy však neprovádí vývojáři. Ti jen poskytují případnou podporu.
- **Beta testování** – Probíhá v prostředí zákazníka a chyby jsou reportovány vývojářům dohodnutou cestou pomocí připravených formulářů [3].

Druhým označením pro alfa a beta testování je raná (alfa) a pozdější, stabilnější (beta) verze systému [3].

#### 1.4.5 Regresní testování

V průběhu jednotkových, integračních a systémových testů probíhá tzv. regresní testování. Účelem regresního testování je ověřit, že změněná část softwaru neovlivní dosavadní funkčnost ostatních částí softwaru. Regresní testování je důležité z důvodů:

- Testování změny k ověření, že změna nezpůsobila nežádoucí jev v jiné části aplikace.
- Zvyšuje pokrytí testy, bez ohrožení časových uzávěrek.
- Zvyšuje rychlé uvedení produktu na trh [9].

Tabulka č. 1: Srovnání testování (Zdroj: vlastní zpracování)

Testování	Rozsah	Způsob	Zodpovědnost
Unitní	Jednotlivé moduly	White box	Programátor
Integrační	Propojení více modulů	White box, black box	Programátor
Systémové	Celý produkt	Black box	Tester
Akceptační	Celý produkt	Black box	Zákazník

### 1.5 Typy testů

Testy software jsou rozděleny do mnoha kategorií a typů. Samotnému dělení se věnuje mnoho článků a knih a každý autor se na problematiku dívá odlišně. Následuje výčet několika základních a obecně zažitých typů, částečně definovaných Ronem Pattonem a uznávaných také například odbornými servery [4; 10].

Toto rozdělení je spíše teoretické. V praxi se jedná skoro vždy různé kombinace těchto typů. Jejich uvedení je nicméně vhodné pro lepší představu o rozmanitosti softwarového testování a množství pohledů na jeho problematiku.

V rámci testování se setkáváme s mnoha typy specializovaných testů. Zde je výčet těch nejpoužívanějších:

- **Regresní testy** – Zjišťují, jestli chyby odhalené v předcházejících testech byly skutečně opraveny a jestli důsledkem provedených změn nebyly do softwaru zaneseny nové chyby.
- **Přírůstkové testy** – Ověřují, zda nově přidaný modul nezpůsobuje chyby při integraci se stávajícími moduly.
- **Smoke testy** – Zajišťují připravenost softwaru na hloubkové systémové testy. Tyto testy je důležité provádět, aby při systémových testech nedocházelo k pádu.
- **Zátěžové testy** – Testují chování systému pod zátěží. Neklade se důraz na funkčnost nebo správnost systému, ale na jeho rychlost a stabilitu při nadstandardní interní zátěži.
- **Stresové testy** – Testují chování systému pod zátěží. Testovaná aplikace se zatěžuje nedostatečným hardwarem. Například se testuje se slabou operační pamětí a pomalým procesorem.
- **Zotavovací testy** – Testují rychlost vzpamatování aplikace po pádu systému, hardwarové chybě, výpadku proudu nebo jiných podobných problémech.
- **Bezpečnostní testy** – Slouží k zajištění celkové bezpečnosti softwaru. Zajišťují, jak a zda se systém chrání před neautorizovaným přístupem, jakým způsobem jsou ukládána hesla, jak je řízen přístup atd. Slouží k nalezení nežádoucího kódu, a bezpečnostních chyb.
- **Výkonnostní testy** – Sledují výkon aplikace při velkém počtu různých požadavků. Lze vysledovat, kterým částem systému je potřeba věnovat větší pozornost a provést v ní příslušné optimalizace.
- **Testy použitelnosti** – Testují uživatelské rozhraní, zda je intuitivní, konzistentní, flexibilní, užitečné, a především pohodlné pro uživatele.

- **Instalační testy** – Testují průběh instalování nebo odinstalování aplikace na dané platformě.
- **Testy kompatibility** – Kontrolují, zda testovaný software korektně spolupracuje a sdílí informace s jiným softwarem, hardwarem, operačním systémem nebo síťovým prostředím.
- **Srovnávací testy** – Porovnávají stabilitu i slabiny s předešlými verzemi softwaru nebo s podobnými produkty.
- **Dlouhodobé testy** – Hledají chyby, které se zpravidla vyskytnou až po delší době používání aplikace [4; 11; 10; 12].

## 1.6 Analytické prostředky

V této podkapitole jsou popsány analytické prostředky, které jsou využity při zpracování této práce, zejména pak v rámci analýzy společnosti.

### 1.6.1 PESTLE analýza

*„PESTLE analýza je analytická technika sloužící ke strategické analýze okolního prostřední organizace.“ [13]*

Zkratka PESTLE je složen z počátečních písmen vnějších faktorů:

- **P** – politické – působení politických vlivů,
- **E** – ekonomické – působení ekonomiky, ať už světové či lokální,
- **S** – sociální – sociální i kulturní vlivy na organizaci,
- **T** – technologické – vliv a dopad technologií na společnost,
- **L** – legislativní – vliv legislativy a jejich změn,
- **E** – ekologické – vliv a problematika životního prostředí [13].

### 1.6.2 Porterův model konkurenčních sil

*„Porterův model konkurenčních sil je způsob analýzy odvětví a jeho rizik. Podstatou metody je prognózování vývoje konkurenční situace ve zkoumaném odvětví na základě odhadu možného chování subjektů a objektů působících na daném trhu.“ [14]*

Porterův model pracuje s pěti základními prvky:

- stávající konkurenti – schopnost konkurence ovlivňovat nabídku a cenu výrobků,
- potenciální konkurenti – možnost, že vstoupí na trh noví konkurenti a tím ovlivní cenu a nabídku výrobků,
- dodavatelé – jejich možnost ovlivnit cenu a nabídku na základě dodávaných vstupů pro výrobky,
- kupující – jakým způsobem mohou ovlivnit cenu a nabízené množství,
- substituty – výrobky a služby, které jsou schopny alespoň částečně nahradit určitý výrobek nebo službu [14].



Obrázek č. 2: Porterův model konkurenčních sil (Zdroj: Vlastní zpracování podle 15)

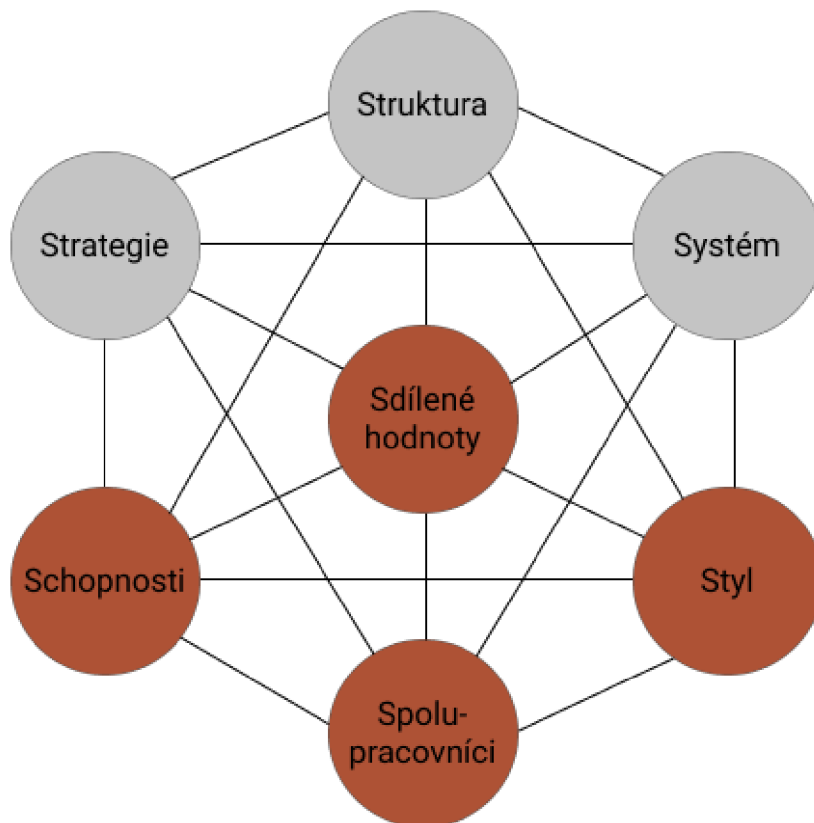
### 1.6.3 Analýza „7S“

„*McKinsey 7S je analytická technika používaná pro hodnocení kritických faktorů libovolné organizace. Patří mezi modely kritických faktorů úspěchu.*“ [15]

Jak již název napovídá jedná se o faktory, které v anglickém znění začínají na písmeno S. Můžeme je rozdělit na tvrdé (vyznačené šedě) a měkké (vyznačené hnědě). V rámci této interní analýzy je popsáno 7 následujících faktorů:

- strategie (strategy) – popis toho, jak organizace dosahuje svých vizí a jakým způsobem dokáže reagovat na hrozby a příležitosti ve svém oboru podnikání,

- **struktura** (structure) – popis organizační struktury podniku,
- **systemy** (systems) – informační systém podniku, společně s řízením a monitorováním procesů,
- **styl** (style) – jakým způsobem je přistupováno k řízení a k problémům,
- **sdílené hodnoty** (shared values) – důležité je, aby všichni v rámci organizace věděli, čeho chce organizace dosáhnout a proč, zahrnujeme sem vizi, poslání a firemní kulturu,
- **spolupracovníci** (staffs) – jakým způsobem je pracováno s lidskými zdroji v rámci organizace,
- **schopnosti** (skills) – znalosti, schopnosti a dovednosti, které se vyskytují v rámci organizace [15].



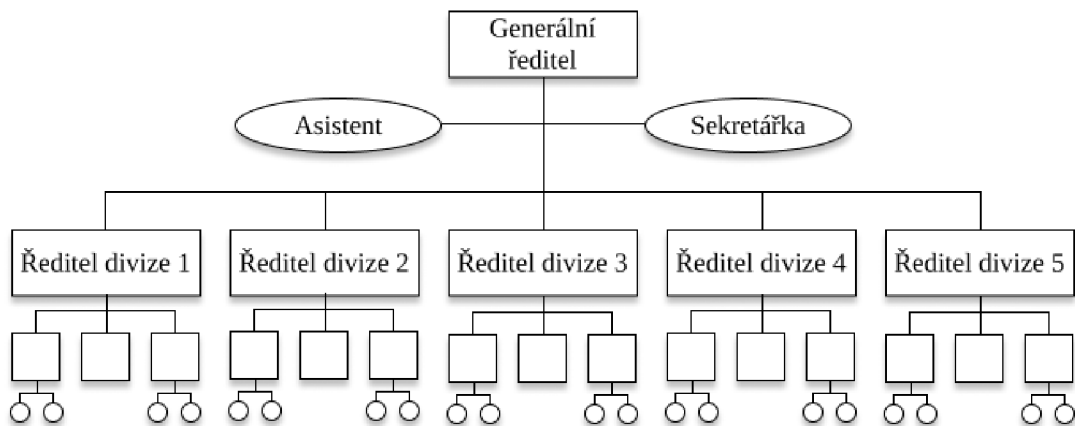
**Obrázek č. 3: Model 7S** (Zdroj: vlastní zpracování podle 16)

#### 1.6.4 Organizační struktura

**Organizační struktura** je hierarchické uspořádání vztahů mezi jednotlivými pracovními místy v rámci organizačních útvarů a vztahů mezi útvary v rámci organizace. Zahrnuje vztahy nadřízenosti a podřízenosti a řeší vzájemné pravomoci (kompetence), vazby a odpovědnost [16].

S ohledem na typ organizace, zaměření, složitost práce, prostorové rozmístění vedoucích a podřízených i počet zaměstnanců a schopnost jejich komunikace, je dobré zvolit optimální rozpětí řízení firmy. A to buď úzké nebo široké. V návaznosti na to lze pak vybrat i vhodnou organizační strukturu. Mezi základní typy organizačních struktur patří:

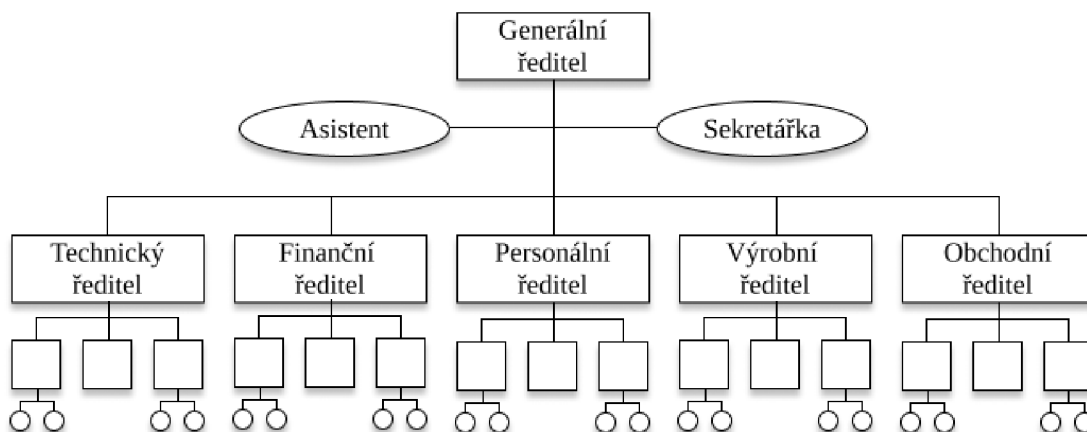
- **funkcionální** – vhodná do výrobních podniků, jednoznačně rozdělené pravomoci pro konkrétní oblasti,



Obrázek č. 4: Funkcionální organizační struktura (Zdroj: vlastní zpracování podle 17)

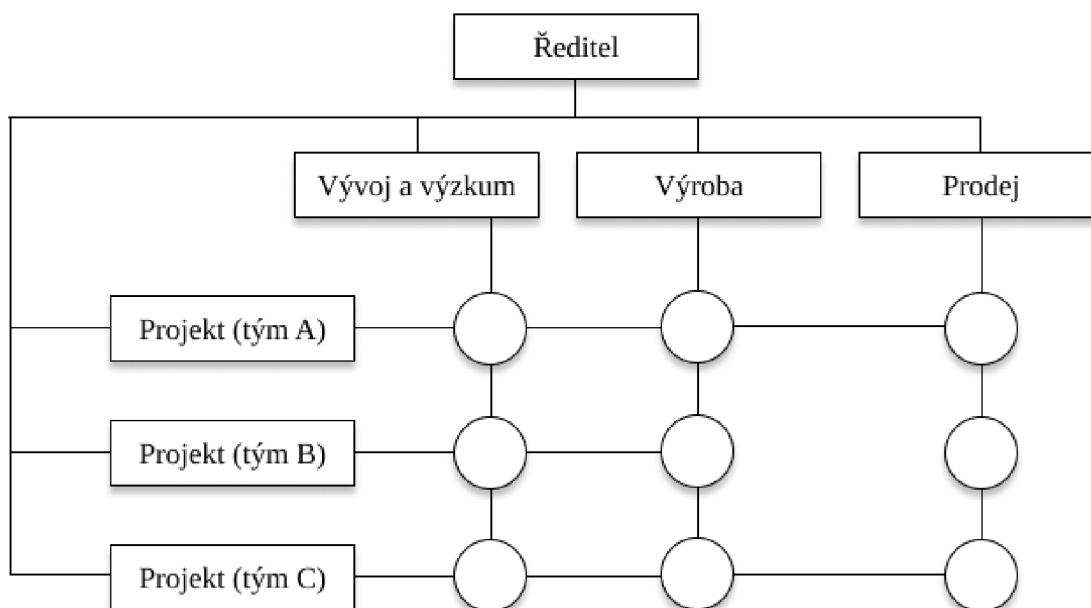
- **divizionální** – každá divize má pod sebou stejné útvary a je relativně samostatná, často na základě územního rozdělení (např. Praha, Brno, Ostrava),





**Obrázek č. 5: Divizionální organizační struktura** (Zdroj: vlastní zpracování podle 17)

- **maticová** – v podstatě propojená funkcionální a divizionální organizační struktura, zaměstnanec je součástí určitého oddělení a zároveň i projektového týmu [16; 17].



**Obrázek č. 6: Maticová organizační struktura** (Zdroj: vlastní zpracování podle 17)

### 1.6.5 SWOT analýza

*„Matice SWOT představuje koncepční rámec pro systematickou analýzu, který usnadňuje porovnání vnějších hrozeb a příležitostí s vnitřními silnými a slabými stránkami organizace, týmu, či projektu.“ [18]*

Zkratka SWOT je odvozena od anglických názvů, které zároveň označují jednotlivé kvadranty matice:

- **silné stránky** (strenghts) – přednosti, výhody, co se organizaci daří,
- **slabé stránky** (weaknesses) – nedostatky a slabiny, čím se organizace sama ohrožuje,
- **příležitosti** (opportunities) – co se organizaci samo nabízí,
- **hrozby** (threats) – nežádoucí ohrožení, to, co by mohlo organizaci ohrožovat v rozvoji [18].

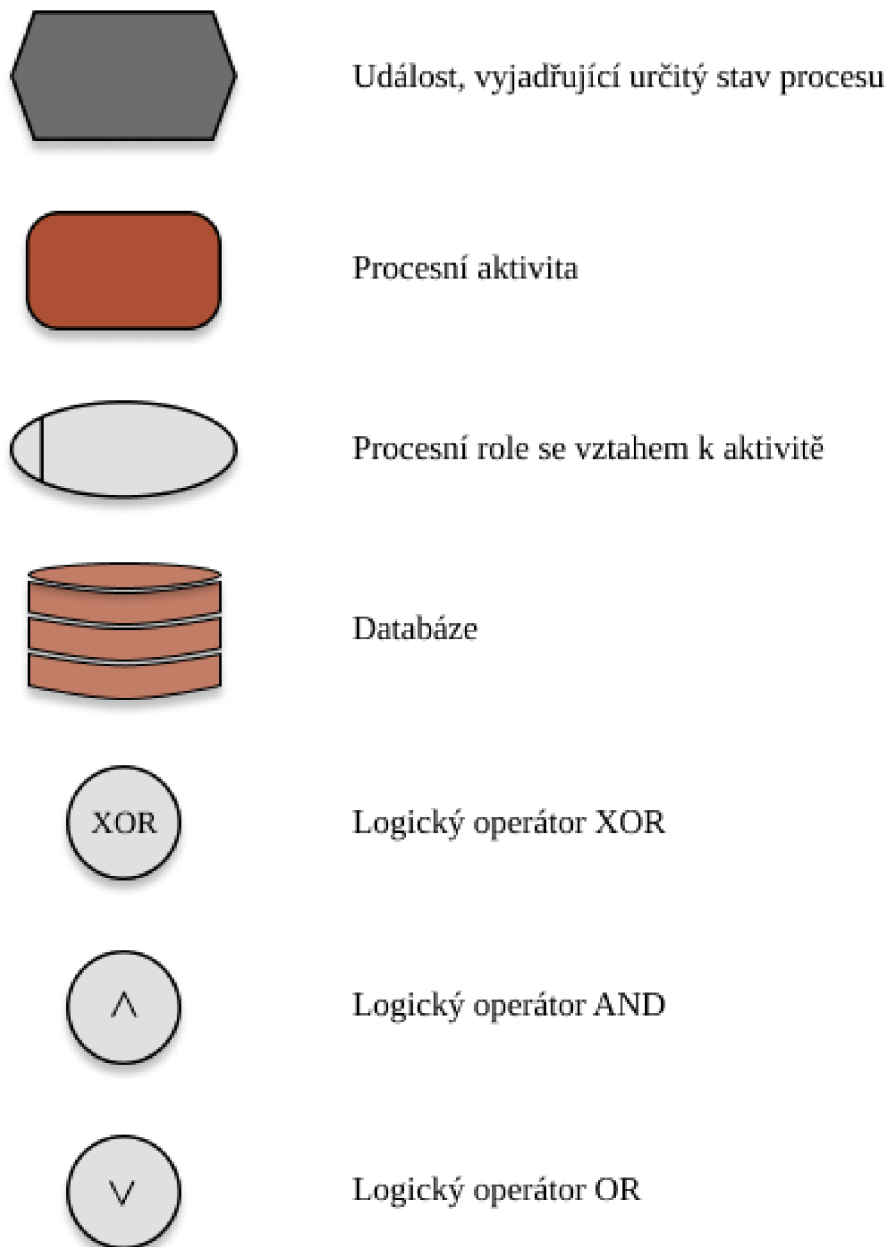
Výstupem SWOT analýzy je v podstatě externí a interní analýza podniku čili to, na co se má podnik zaměřit, co zlepšit a na co si dávat pozor.



Obrázek č. 7: SWOT analýza (Zdroj: vlastní zpracování podle 19)

## 1.7 EPC diagram

EPC (Event-driven Process Chain) diagram ilustrují grafické znázornění procesů. Díky těmto ilustracím lze lépe naplánovat zdroje a identifikovat možná vylepšení obchodního procesu [19].



**Obrázek č. 8: Značky EPC diagramu** (Zdroj: vlastní zpracování podle 20)

## 1.8 Metody projektového managementu

V této podkapitole jsou popsány metody projektového managementu, které jsou využity při zpracování této práce, zejména pak v rámci vlastního návrhu.

### 1.8.1 Logický rámec

Pan Jan Doležal definoval logický rámec jako: „*Metoda logického rámce slouží jako pomůcka při stanovování cílů projektu a jako podpora k jejich dosahování. Hlavním principem je fakt, že základní parametry projektu jsou vzájemně logicky provázány. Dalšími použitými principy je potřeba měřitelnosti výsledků, práce v týmu a systémový přístup.*“ [20, s. 67]

Jednotlivými prvky logického rámce jsou:

- záměr – odpovídá na otázku, co přesně daný projekt svou realizací přinese,
- cíl – odpovídá na otázku, proč má být projekt realizován, velmi často je využívána metoda SMART,
- výstupy – určuje, jaké konkrétní výstupy lze z projektu očekávat,
- aktivity – jednotlivé aktivity směřující k dosažení výstupů projektu [20, s. 68].

Logický rámec obsahuje ještě další 3 oblasti, které jsou potřebné pro komplexnost výstupu:

- objektivně ověřitelné ukazatele – měřitelné ukazatele na jejichž základě můžeme říct, zda byl záměr, cíl či výstup splněn,
- způsob ověření – zdroje, ze kterých je možné čerpat informace pro zhodnocení splnění cíle, záměru a výstupů,
- rizika a předpoklady – předpoklady potřebné pro naplnění očekávání projektu, rizika, která by tato očekávání mohla ohrozit [20, s. 68].

### 1.8.2 WBS

WBS paní Svozilová definovala: „*WBS svou strukturou odpovídá rozpisu dílčích cílů projektu a rozepisuje požadovaný produkt projektu do logické hierarchie úloh. Podrobný rozpis prací je jeden z klíčových dokumentů projektu, který je důležitým východiskem pro řízení projektu.*“ [21, s. 123]

Účelem této struktury je vytvořit přehledný plán celého projektu a zjistit jaké výstupy je třeba dostat. Existuje tzv. pravidlo 100 %, které říká, že co není ve WBS, nebude realizováno [21].

### 1.8.3 RACI matice

„**RACI matice** (*matice odpovědnosti*) je sestava řádků a sloupců, jejímž prostřednictvím jsou přiřazeny projektovým úsekům a úkolům ti, kteří mají schopnosti a odpovědnost pro jejich realizaci.“ [21, s. 155]

Raci matice je vytvářena na základě činností vyllynulých z WBS. Při vytváření RACI matice jsou využívány následující stavy:

- **R** – responsible – člověk fyzicky zodpovědný za vykonání úkolu,
- **A** – accountable – člověk zodpovědný za vykonání úkolu a jeho výsledek, ke každé činnosti může být pouze u jednou,
- **C** – consulted – člověk, se kterým má být činnost konzultována, který může poskytovat rady,
- **I** – informed – člověk, který má být informován o průběhu činnosti [22].

### 1.8.4 Metoda RIPRAN

„**Metoda RIPRAN** (*RIsk PROject ANalysis*) představuje empirickou metodu pro analýzu rizik projektů, zvláště je velmi vhodná pro střední a velké projekty.“ [23]

Skládá se z následujících kroků.

1. Příprava analýzy rizik projektu,
2. identifikace rizik projektu,
3. kvantifikace rizik projektu,
4. návrh opatření snižujících nebo eliminujících vliv rizik na projekt,
5. celkové zhodnocení rizikovosti projektu,
6. sledování a vyhodnocování rizik v průběhu projektu [23].

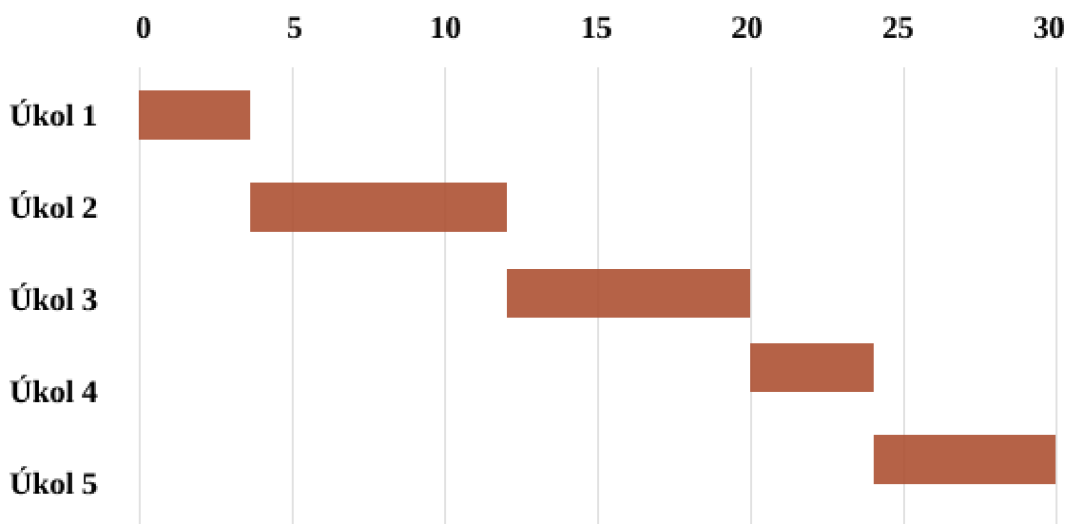
Metodu RIPRAN je možné využívat jak před začátkem projektu, tak i ve všech ostatních fázích projektu k identifikaci nových rizik projektu [23].

### 1.8.5 Ganttovy diagramy

Ganttovy diagramy velmi jednoduše znázorňují posloupnosti naplánovaných činností v časovém intervalu a jejich začátky a konce. Časová osa projektu bývá zobrazována na horizontální úrovni, zatímco jednotlivé činnosti jsou postupně zobrazovány pod sebou.

Ganttovy diagramy mají určitou slabinu a to, že nezobrazují závislosti mezi činnostmi [21, s. 138].

V současné době se Ganttovy diagramy využívají jako nástroj pro komunikaci a jednání, proto je lze vytvářet v mnoha různých softwarech [21, s. 139].



Obrázek č. 9: Příklad Ganttového diagramu (Zdroj: vlastní zpracování podle [24])

### 1.8.6 Rozpočet

„*Rozpočet projektu je souborem parametrů a číselných údajů, které dávají do souvislosti časová, množství a finanční kvanta, která souvisí s plánem a realizací dílčích elementů projektu.*“ [21, s. 159]

Rozpočet je standardně sestavován v konceptuální fázi projektu, jako podklad pro jednání o ceně a uzavření smlouvy. Jedná se o tzv. předběžný rozpočet projektu. V plánovací fázi je vytvořen závazný rozpočet projektu, který může být aktualizován i v realizační fázi projektu, ale pouze v souladu s předem danými pravidly [21, s. 160].

V rámci rozpočtu jsou zohledňovány 3 druhy nákladů:

- přímé náklady,
- nepřímé (režijní) náklady,
- ostatní náklady – slouží zejména ke krytí neočekávaných nákladů projektu [21, s. 160].

## 2 ANALÝZY SOUČASNÉHO STAVU

V této kapitole je provedena analýza současného stavu společnosti IREKS ENZYMA s. r. o., pro kterou bude vybírán určitý druh testování softwaru. V rámci kapitoly je nejdříve stručně představena společnost jako taková, následně jsou provedeny analýzy vnějšího a vnitřního prostředí, které jsou vstupem pro SWOT analýzu.

V závěru kapitoly jsou pak shrnuty veškeré výstupy zpracovaných analýz, které budou sloužit jako primární podklad pro zpracování vlastního návrhu řešení této práce.

### 2.1 Představení společnosti

Společnost IREKS ENZYMA s.r.o. byla založena na počátku roku 1991. Za nedlouho poté obsadila vedoucí pozici na trhu s přípravky, směsi a speciálními surovinami pro pekárny a cukrárny v tehdejší Československu. I po rozdělení si nově vzniklé státy tuto vedoucí pozici stále drží [25].

Nejprve se firma zabývala výrobou podle licenčních výrobků, ale postupem času začala si vytvořit vlastní recepty podle, kterých vyráběla produkty. Kvůli náhlému rozšiřování sortimentu se postupně kladly na výrobu čím dál větší nároky. V roce 1995 byl proto postaven nový výrobní závod v Kroměříži a později roku 1997 v Prešově [25].

K celosvětové skupině IREKS se firma připojila roku 1999 a současně se stala důležitou součástí koncernu. Z původního názvu ENZYMA se přejmenovala dle koncernového vzoru na IREKS ENZYMA s.r.o. Díky štedré investici v roce 2004 bylo postaveno nové a první sídlo firmy v historii. Budova je rozdělena na dvě části:

- administrativní,
- zákaznické, vývojové a technologické centrum [25].

Za léta, co se firma pohybuje na trhu, si vybudovala loajální a kompaktní tým zaměstnanců, který dbá na neustálý vývoj a zdokonalování svých dovedností.

Společnost IREKS je známá především ve světě pekařství, kam přináší své pekárenské přísady v nejvyšší kvalitě a kreativní výrobní nápady. Působí ve více než 90 zemích a poskytuje rady a služby pekařům a cukrářům. Primárním cílem je tedy výroba prvotřídních pekařských směsí z nejlepšího obilí a surovin. Řadí se mezi přední výrobce pivovarnických sladů pro řadu významných pivovarů. V posledních letech se angažují v oblasti cukrářství, kde vyvíjí specializovaná aromata a zmrzlinové výrobky [25].

Řídí se heslem: „Z přírody to nejlepší.“ [25]



Obrázek č. 10: Logo firmy (Zdroj: 26)

### 2.1.1 Formální údaje společnosti

<b>Obchodní firma:</b>	IREKS ENZYMA s.r.o.
<b>Sídlo:</b>	Brno – Horní Heršpice, Kšírova 668/257, PSČ 619 00
<b>Právní forma:</b>	Společnost s ručením omezeným
<b>IČO:</b>	155 27 212
<b>Datum vzniku:</b>	30. leden 1991
<b>Počet zaměstnanců:</b>	130
<b>Statutární orgán:</b>	Ing. Jaromír Gec
<b>Předmět podnikání:</b>	pekárenská výroba, cukrárenská výroba, výroba homogenizovaných potravinářských přípravků, obchod a služby, mlynářství [26]

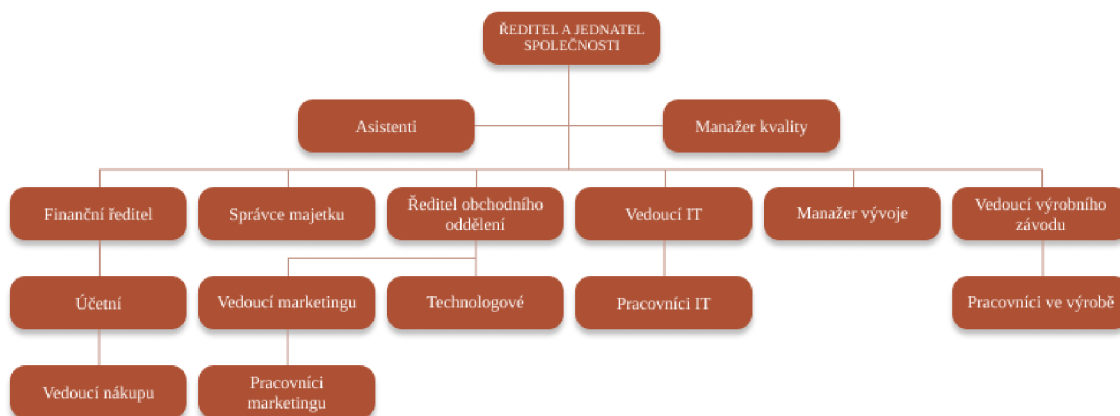
### 2.1.2 Organizační struktura

Organizační struktura firmy IREKS ENZYMA není nijak složitá, vzhledem k tomu, že patří mezi střední podniky, tj. podniky s méně jak 250 zaměstnanci. Jedná se o funkcionální organizační strukturu, kdy jednatel společnosti má pod sebou pracovníky odpovědné za chod jednotlivých oblastí (účetní, vedení marketingu, správa IT apod.) a ti následně mají pod sebou ostatní pracovníky společnosti.

Přestože je organizační struktura formálně rozdělena tímto způsobem, pracovníci společnosti mezi sebou velmi často spolupracují a vzájemně si vypomáhají.

Následující obrázek zobrazuje detailnější pohled na organizační strukturu firmy.





**Obrázek č. 11: Organizační struktura** (Zdroj: vlastní zpracování)

## 2.2 Analýza vnějšího prostředí společnosti

V rámci analýzy vnějšího prostředí jsou provedeny dvě analýzy, konkrétně analýza PESTLE a Porterův model konkurenčních sil. Díky tomu je možné zjistit, jak si společnost stojí v rámci svého odvětví, mezi konkurenty, jak ji ovlivňují vnější podmínky apod.

### 2.2.1 PESTLE analýza

#### Politické faktory

Politická situace v naší zemi je již mnoho let stabilní a nevyznačuje se nijak zásadními změnami. Obor podnikání společnosti je především zaměřen na výrobu pekárenských pochutin, které s politikou nejsou úzce spjaty.

Možným ovlivnitelem je daňová politika, ovšem vývoj DPH se za poslední roky příliš nezměnil. A posledním bodem je zahraniční politika, kde je potřeba dát pozor na případné sankce [27].

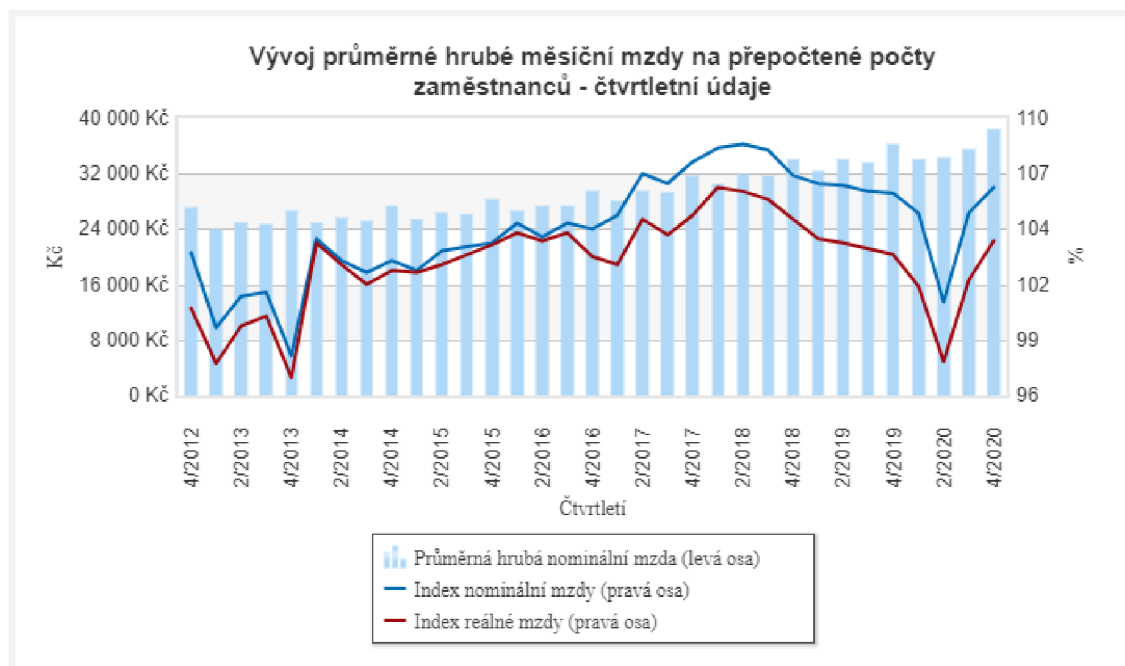
#### Ekonomické faktory

Ekonomické hledisko má naopak na fungování společnosti celkem zásadní dopad. To, v jakém stavu se momentálně nachází ekonomika státu, popř. celosvětová ekonomika totiž přímo ovlivňuje příjmy populace či nezaměstnanost. Mezi hlavní ekonomické ukazatele patří hlavní makroekonomické veličiny, jako je nezaměstnanost a inflace.

Inflace má neblahý vliv na výrobu, s rostoucí inflací poroste i cena surovin potřebných na výrobu, čímž by se zásadně musela změnit celková cena výrobků, a odběratelé by tuto cenovou nabídku nemuseli akceptovat.

Nezaměstnanost v případě krátkého časového horizontu ovlivňuje výšku nominálních mezd a inflaci. Tím se významně ovlivňuje počet pracovníků v podniku. Kvůli celosvětové pandemii a důsledcích veškerých vládních opatření, nezaměstnanost nevyhnutelně roste a růst bude nadále. V lednu 2021 byla nezaměstnanost na 4,3 %, což je o 1,3 % více než v lednu roku 2020 [28].

Graf č. 1 popisuje vývoj průměrné měsíční hrubé mzdy v ČR za posledních několik let, což je z hlediska prostředků pro nákup výrobku důležitý údaj.



Graf č. 1: Vývoj průměrné hrubé měsíční mzdy (Zdroj: 30)

### Sociální faktory

Sociální faktory, co se týče potravinářského průmyslu, nemají velký vliv na chod firmy. Zajímavým bodem může být zejména růst celkového počtu populace, což by mělo za následek případné zvětšení poptávky po zboží od firmy IREKS ENZYMA.

### Technologické faktory

Vliv technologií v potravinářském odvětví hraje velkou roli. Společnost ve své výrobě používá nespočet strojů, které by jejich „upgradováním“ mohli zefektivnit celkovou výrobu, a tím i výrazně ovlivnit cenu výrobků a teoreticky i trh po výrobcích.

Můžeme tedy říct, že vývoj nové technologie by mohl pozitivně ovlivnit chod celé firmy.

Kybernetické útoky obecně pro firmy představují velké riziko, a proto je zapotřebí se před těmito útoky a datovými podvody chránit. Důležitým krokem je zvyšování informovanost zaměstnanců a úroveň ochranných programů. Konkrétním případem kybernetického útoku je útok přes chybu v produktech společnosti Microsoft, který se stal v březnu 2021. Firma musela ihned reagovat, aby nedošlo k úniku dat.

### **Legislativní faktory**

Legislativa je jedním ze zásadnějších faktorů pro chod společnosti. Jakákoli legislativní změna týkající se potravinářského průmyslu má na společnost vliv. Společnost má právní formu podnikání jako společnost s ručením omezeným, proto se řídí zákonem č. 90/2012 Sb. – Zákon o obchodních korporacích. Pohybuje se v potravinářském průmyslu, tudíž se řídí zákonem č. 110/1997 Sb. – Zákon o potravinách a tabákových výrobcích, a o změně a doplnění některých souvisejících zákonů [29].

Dalšími zákony, které společnost musí dodržovat, jsou zákony o hygienických předpisech, zákony související s obchodováním, účetnictvím, zákony o ochraně osobních údajů, nařízení a směrnice Evropské unie.

Se zavedeným nařízením o ochraně osobních údajů podle GDPR je nutné, aby firma zacházela opatrně s osobními údaji, a to jak ve firmě, tak i v prostředí internetového obchodování, které je nově k dispozici [29].

### **Ekologické faktory**

Ekologické faktory mohou být pro firmu zásadní. V posledních letech se Česká republika potýká s obrovskými suchy, což má obrovský vliv na zemědělství. Následky těchto dopadů by se opět odrazily v ceně výrobků.

## **2.2.2 Porterův model konkurenčních sil společnosti**

### **Stávající konkurenti**

Z hlediska odvětví, ve kterém společnost podniká, je stávající konkurence velká, protože potravinový průmysl je potřebný pro každodenní život.

Jelikož si firma nepřeje zveřejňovat informace o konkrétních konkurentech, tak v rámci diskrétnosti zůstanou tyto údaje utajeny.

V Brně se momentálně nenachází příliš mnoho konkurentů, které by mohli firmu ohrozit. Není důležité, kde se firma nachází, jelikož neprodává výrobky koncovým uživatelům, nýbrž jiným firmám, které je dále zpracují a prodávají v kamenných obchodech. Proto raději řekněme, že v rámci celé České republiky se zde objevuje kolem deseti konkurenčních firem.

Výhodou společnosti v konkurenčním boji je možnost zboží objednávat přes internetový obchod, což firmám umožní komfortnější a rychlejší nákup.

### **Potenciální konkurenti**

I přesto, že je trh s potravinářskými výrobky celkem nasycen, je potřeba nadále sledovat trh kvůli vzniku nové konkurence. Pravděpodobnost výskytu nové konkurence je sice nízká, ale to procento tu je a je potřeba ho brát na vědomí.

Určitou hrozbou by mohla být společnost, která by vyráběla stejně kvalitní výrobky a služby, ale za nižší cenu. Ale vybudování takové firmy trvá roky, proto není nutné mít zatím obavy.

Nicméně by bylo dobré, aby společnost udržela stávající úroveň kvality svých výrobků a služeb a do budoucna je posunovala na vyšší úroveň.

### **Dodavatelé**

Vyjednávací síla dodavatelů se do určité míry liší v závislosti na produktu a na velikosti dodavatele. Pokud je dodavatel dominující je pochopitelné, že vyjednávací síla bude velká, u méně dominujícího dodavatel je to přesně naopak. Pak se vyjednávací síla liší i u produktů, které jsou produkovány v omezeném množství. U těchto produktů je vyjednávací síla dodavatelů poměrně vysoká. Na druhou stranu u běžných produktů již tato vyjednávací síla není až tak veliká.

Pokud se jedná o dodavatele softwarového či hardwarového vybavení společnosti, je jejich vyjednávací síla malá, jelikož i tento trh je hodně nasycen.

Při poskytování služeb zaměstnanci je společnost v podstatě sama dodavatelem služby, a má vůči sobě naprosto specifickou vyjednávací schopnost.

## **Kupující**

Vyjednávací síla kupujících je v tomto oboru podnikání poměrně velká. Jak již bylo zmíněno hlavními kupujícími jsou další firmy a podniky, proto zde hraje velkou roli jejich postavení na trhu i jejich velikost. Velká firma znamená pro společnost větší vyjednávací sílu, ale taktéž i větší zisky. Je důležité si takové firmy uhlídat, aby neodešly ke konkurenci. S malými firmami je vyjednávací síla o něco menší.

Výběr prodejce velmi často ovlivňuje prostředí, přístup obchodníka, a především nejdůležitější roli hraje cena výrobku. U e-shopů může být rozhodující zejména přehlednost stránek, celkový dojem nebo doba dodání zboží.

## **Substituty**

Jelikož si firma nepřeje zveřejňovat informace o konkrétních substitutech, tak v rámci diskrétnosti zůstanou tyto údaje utajeny.

Konkurenční výhodou společnosti je bezesporu kvalita výrobků, díky čemuž se snaží být těmi nejlepšími na trhu. Další bodem je poskytování kvalitních služeb, a to zejména předávání „know how“ těm, kteří o to opravdu stojí. V neposlední řadě je určitou výhodou i přizpůsobování se potřebám kupujícím, jelikož ne všichni kupující mají stejné vybavení jako IREKS ENZYMA a občas je potřeba nějaké postupy upravit.

## **2.3 Analýza vnitřního prostředí společnosti**

V rámci analýzy vnitřního prostředí společnosti je provedena analýza 7S. Díky této analýze je možné zjistit jaký je momentální stav uvnitř společnosti, co ji činní problémy, a naopak v čem je silná.

### **2.3.1 Analýza 7S**

#### **Strategie**

Hlavní strategií společnosti je poskytovat zákazníkům kvalitní produkty za přijatelnou cenu. Snaží se vybudovat ve firmě příjemné a přátelské prostředí, které bude mít za následek větší motivaci pracovat, odvádět lepší výsledky a celkově zvyšovat kvalitu práce. Další bodem strategie bylo vybudování elektronického obchodování a usnadnění nákupu stálým zákazníkům.

## **Struktura**

Organizační struktura společnosti IREKS ENZYMA byla popsána již dříve a kapitole 2.1.2 Organizační struktura.

## **Systém**

Ke každodenní práci pracovníci jak v administrativě, tak ve vývojové části, používají informační systém, který vyvíjí pracovníci v IT oddělení. Systém obsahuje tisíce položek a dat o výrobcích, recepturách, partnerech a zaměstnancích. Veškeré tyto data se ukládají do interní databáze obsahující recepty produktů, položkovou dokumentaci, stavy skladů, docházku a další informace potřebné ke správnému chodu firmy.

Mezi další ze základních softwarových vybavení patří klasický kancelářský balíček MS Office, který se využívá pro tvorbu různých podpůrných dokumentů potřebných při fungování společnosti.

Pro vzájemnou komunikaci mezi jednotlivými zaměstnanci je využíván e-mail či pracovní telefon. V současné době kvůli světové pandemii byla zakoupena licence od aplikace Microsoft Teams, která je využívána ke komunikaci především na „home office“.

V určitých částech společnosti se užívá program JIRA, který slouží k lepšímu plánování a kontrole projektů v týmu. Do budoucna firma plánuje rozšířit používání JIRA i do jiných oddělení.

Mezi systémy, které používá především IT oddělení, patří MS Visual Studio, MS SQL Server, Indigo Studio, BPMN Studio, Adobe Creative Suite, Internet Information Services, testovací server a sdílené adresáře.

## **Styl**

Vzhledem k tomu, že se jedná o střední podnik, převládá demokratický styl vedení, kdy ředitel společnosti bere v potaz návrhy ostatních vedoucích, ale rozhodující slovo má vždy on. Jelikož se jedná o firmu, která spadá pod německou firmu IREKS, je potřeba zohlednit i jejich návrhy a požadavky.

## **Spolupracovníci**

Ve společnosti v současné době pracuje 130 zaměstnanců. Jak již bylo zmíněno, firma se snaží vytvořit přátelské prostředí. Firma je na trhu pár desítek let a díky tomu je zde spousta zaměstnanců, kteří byli u vzniku, a proto se většina z nich zná navzájem a mají spolu dobré vztahy. Díky tomu jsou zde velmi loajální pracovníci, kteří se mají za sebou léta zkušeností a znalostí a vykazují velkou kvalitu své práce.

Komunikace mezi zaměstnanci z hlediska osobního zaměření je na výborné úrovni, jak již bylo zmíněno výše, ovšem co se týče komunikace v pracovním smyslu, tak zde je na ní potřeba zapracovat. A to zejména v přidělování úkolů, zaměstnanci mají problém správně definovat zadání úkolu na formální úrovni, který je potřebný pro následnou dokumentaci.

Při najímání nových zaměstnanců se společnosti snaží získat především mladší populaci, která by přinesla nový a svěží pohled na stereotypní práci všeho druhu. Samozřejmě musí projít pohovorem a mít určitou znalost v oblasti práce, o kterou mají zájem.

## **Schopnosti**

Firma si zakládá na výrobě kvalitních produktů, tudíž je důležité vzdělání a schopnosti všech zaměstnanců. Každý nový zaměstnanec se po úspěšném pohovoru účastní workshopů a různých školení, aby si své dosavadní zkušenosti dále prohloubil.

Pro komunikaci s obchodními partnery a zákazníky je dobrá komunikační schopnost a osobní projev. Jelikož partneři jsou většinou rakouské národnosti, oceňuje se znalost německého jazyka.

## **Sdílené hodnoty**

Jak již bylo řečeno hlavním posláním společnosti je poskytovat kvalitní produkty, všem, kteří o ně mají zájem, a celkově se rozvíjet.

Společnost se zejména specializuje na výrobu pekárenských směsí a dodávání do pekárenských firem, nicméně se postupně rozvíjí a nabízí i další výrobky do jiných odvětví, jako je například cukrářství.

## 2.4 SWOT analýza

Poznatky zjištěné v rámci analýz vnějšího a vnitřního prostředí společnosti vedly k sestavení SWOT analýzy společnosti, která shrnuje silné a slabé stránky společnosti a zároveň také možné příležitosti a hrozby, které by mohly společnost postihnout. SWOT analýza se především zaměřuje na aspekty týkající se testovacího procesu.



Obrázek č. 12: SWOT analýza společnosti (Zdroj: vlastní zpracování)

## 2.5 Analýza vývojového procesu

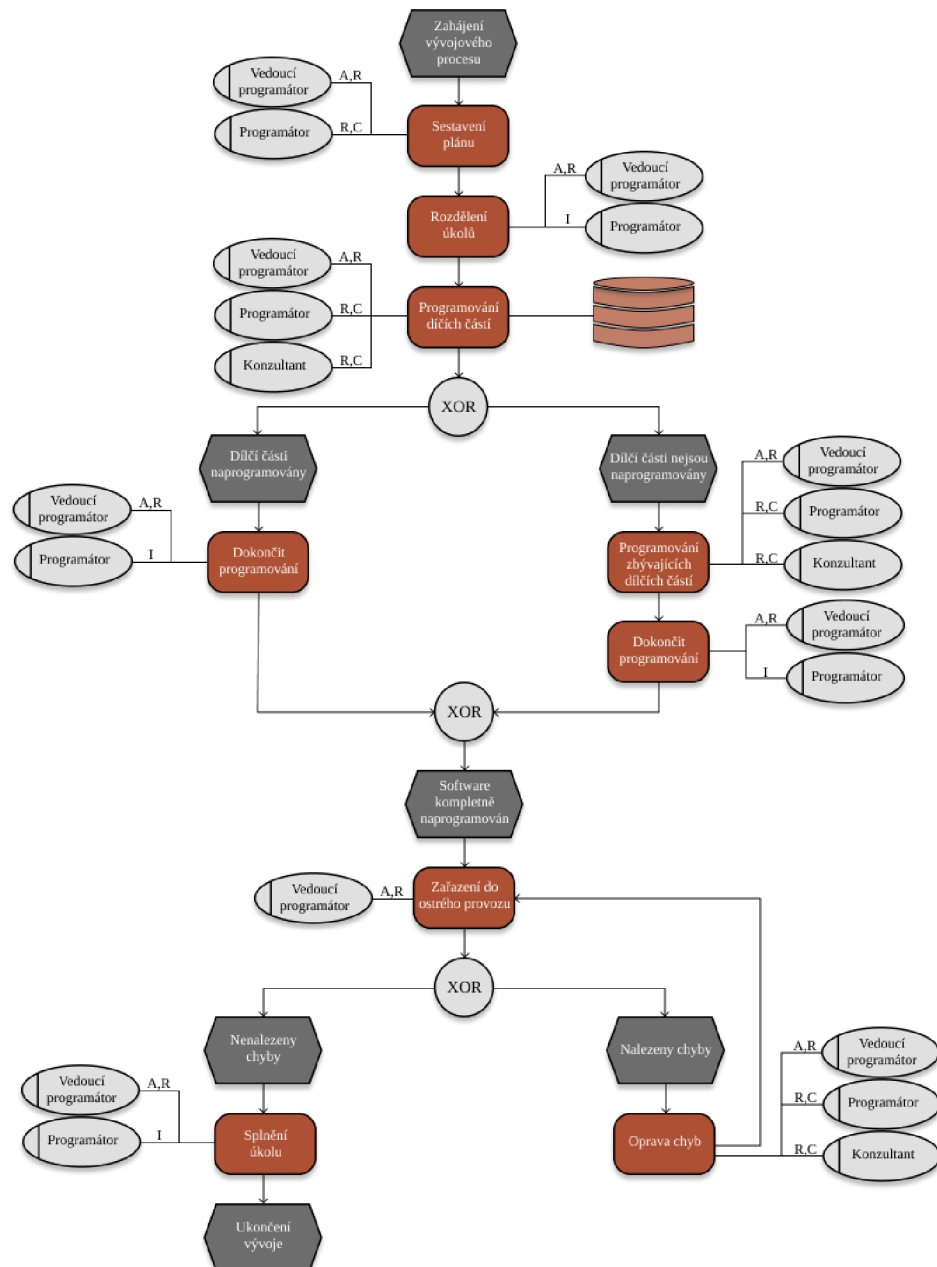
V této kapitole bude popsán hlavní proces vývoje softwaru v IT oddělení. Pro lepší popis bude tento proces ztvárněn za pomoci EPC diagramu.

Co se týká softwarového vývoje tak ten je většinou vždy stejný pro každý software. Nejprve je vedoucím programátorem zahájena porada IT oddělení, kde se projednává, jaké problémy je potřeba vyřešit. Na poradě se sestaví plán projektu, rozdělí se dílčí úkoly a osoba, která za ně bude zodpovědná. Všechny aktivity budou provedeny vedoucím programátorem za pomoci podřízeného programátora. Pokud by nastaly nejasnosti prodiskutují možnosti s konzultantem.



Po poradě se podle stanoveného plánu začne s programováním. Jakmile je vše naprogramováno, tak výsledný software jde hned do ostrého provozu a teprve poté se řeší příčné nedostatky. Pokud je nalezena chyba, tak se programátoři snaží co nejrychleji chybu opravit a následně software spustí do ostrého provozu. Po nalezení chyb je vývojový proces ukončen.

Takový postup je příliš zdlouhavý a může se stát, že tím mohou ovlivnit celý chod firmy a systém bude nefunkční.



Obrázek č. 13: Vývojový proces softwaru (Zdroj: vlastní zpracování)

## **2.6 Zhodnocení současného stavu**

Společnosti IREKS ENZYMA v současné době nemá velké potíže. Má vlastní informační systém, má spolehlivé zaměstnance a výbornou pověst.

Na co by bylo vhodné se více zaměřit je IT oddělení, které vyvíjí nové softwary nebo předělává stávající software na lepší, a je potřeba tyto softwary zkontrolovat předtím, než je začnou používat zaměstnanci. Tyto softwary jsou vyvíjeny a kontrolovány stejnými lidmi a někdy se stane, že na některé chyby se vůbec nepřijde.

Velkou nevýhodou je dosavadní testování softwaru, které ve společnosti zcela chybí. Ve firmě se nenachází nikdo, kdo by se věnoval čistě testování. Veškerý vývoj softwaru je přenechám pouze programátorům. Jakmile programátoři vyvinou nový software, ihned se zařadí do normálního chodu společnosti a teprve poté zkouší jeho funkčnost. Veškeré chyby mohou omezit funkčnost celého systému, a tím omezit i efektivitu práce zaměstnanců, kteří systém využívají.

Proto by bylo výhodné zaměřit se na ono zmiňované testování, které ušetří jak práci, tak i čas a vyvíjený produkt může být velmi rychle funkční.

## 3 VLASTNÍ NÁVRH ŘEŠENÍ

V následující kapitole je zpracován vlastní návrh řešení. V rámci kapitoly je nejdříve proveden výběr konkrétního testování softwaru na základě potřeb a požadavků společnosti IREKS ENZYMA s. r. o., které vyplynuly z analýzy současného stavu.

V další části je pak již popsána implementace systému, včetně jednotlivých fází. V závěru je pak shrnut celkový přínos řešení pro společnost.

### 3.1 Výběr testovacího softwaru

V této kapitole jsou nejdříve popsány požadavky, které by měl skladový systém splňovat, a to jak s ohledem na výstupy analýzy, tak i s ohledem na aktuální potřeby společnosti. V návaznosti na to jsou popsána tři různá řešení skladových systémů od tří společností. V závěru kapitoly je pak jedno z těchto řešení vybráno pro následnou implementaci.

#### 3.1.1 Specifikace požadavků

S ohledem na provedenou analýzu současného stavu byly po konzultaci se společností IREKS ENZYMA stanoveny požadavky na testování. Tyto požadavky jsou rozděleny do dvou skupin podle důležitosti pro společnost, a to klíčové a ostatní požadavky.

Klíčovými požadavky jsou myšleny ty požadavky, které jsou pro společnost naprostou nezbytností. Vedlejší požadavky pak shrnují požadavky, které by si společnost přála, nicméně by se bez nich i dokázala obejít.

Klíčové požadavky

- Zrychlení softwarového vývoje.
- Zrychlení hledání chyb při vývoji.
- Přijatelná cena a licenční podmínky.
- Automatizované testování.
- Zrychlení efektivity práce.
- Testy prováděné testerem.
- Cílový programovací jazyk C# a integrace do Visual Studia.

Vedlejší požadavky

- Možnost zavedení bez nutnosti pořízení velkého množství dalšího hardwarového vybavení.

- Možnost přístupu do systému odkudkoli.
- Co možná nejkratší doba zavedení systému.

### 3.1.2 HP QuickTest Professional

HP QuickTest Professional je jedním z nejpoužívanějších nástrojů pro automatizované testování. Byl vyvinut firmou Mercury Interactive a na trh byl uveden v roce 2001 jako verze 5.5. QuickTest Professional patří mezi GUI nástroje (Graphical User Interface – grafické uživatelské rozhraní), to znamená, že jej můžeme ovládat pomocí interaktivních grafických ovládacích prvků (menu, ikony, tlačítka, posuvníky, formuláře a podobně) a tudíž je i velice uživatelsky přátelský. Je postaven na skriptovacím jazyku VBScript, který je součástí základní instalace [30].

HP QuickTest Professional podporuje automatizaci funkčních a regresních testů, která řeší každé hlavní prostředí softwarových aplikací. Toto řešení využívá koncept testování na základě klíčových slov ke zjednodušení vytváření a údržby testů. Umožňuje testerům vytvářet funkční a regresní testovací případy zachycením toků přímo z obrazovek aplikací pomocí specializované technologie snímání. Odborníci na testování mají také plný přístup k základním vlastnostem testu a objektu prostřednictvím integrovaného skriptovacího a ladicího prostředí [30].

Cena licence se odlišuje podle období používání. Licence na 1 rok stojí 3 200 \$ [30].

### 3.1.3 Selenium WebDriver

Selenium je bezplatný (open-source) automatizovaný testovací rámec používaný k ověření webových aplikací v různých prohlížečích a platformách. K vytvoření testovacích skriptů selenu můžete použít více programovacích jazyků, jako je Java, C #, Python atd. Testování prováděné pomocí nástroje pro testování selenu se obvykle označuje jako testování selenu [31].

Software Selenium není jen jeden nástroj, ale sada softwaru, přičemž každý kus vyhovuje různým potřebám organizace pro testování kvality Selenium. Zde je seznam nástrojů:

- Selenium Integrated Development Environment (IDE),
- Selenium Remote Control (RC),
- WebDriver,

- Selenium Grid.

WebDriver se v mnoha aspektech osvědčil jako lepší než Selenium IDE a Selenium RC. Implementuje modernější a stabilnější přístup k automatizaci akcí prohlížeče. WebDriver se na rozdíl od Selenium RC nespolehá na JavaScript pro testování automatizace selenu. Ovládá prohlížeč přímou komunikací s ním [31].

Podporované jazyky:

- Java,
- C#,
- PHP,
- Python,
- Perl,
- Ruby [31].

Selenium je bezplatný open source a zdarma, ale používání automatizačního řešení není bezplatné. Roční licence stojí 2 500 \$.

#### **3.1.4 Conformiq Designer**

Conformiq Designer je nástroj pro vytváření testovacích skriptů pomocí modelů testovaných systémů. Tester vytvoří model předpokládaného chování systému a nástroj tento model převede v sadu testovacích skriptů i se vstupními a výstupními daty, které se pak dají aplikovat na vyvíjený software [32].

Modely se vytvářejí v jazyce UML, k tomu společnost Conformiq nabízí nástroj Modeler zcela zdarma, ale podporuje i jiné nástroje k tomu určené jako například IBM Rational nebo může být model pouze textový vytvořený použitím Conformiq Modelling Language (QML). QML je jazyk odvozený od jazyka Java vytvořený společností Conformiq. Nástroj sám testy pouze generuje a dokáže testy exportovat do jednoho z podporovaných formátů nebo rovnou vytvořit spustitelné balíčky testů. Nástroj je dobrý ke generování funkčních, integračních a systémových testů [32].

Cena nástroje Conformiq Designer pro komerční užití se stanoví až po jednání se společností Conformiq. Je možnost bezplatné licence na nástroje společnosti Conformiq pro akademické uživatele nebo pro ty, kteří jsou ochotni zveřejnit své modely [32].

Nástroj je stále společností podporován, ale společnost se nejvíce zaměřuje na podporu svých zákazníků a veřejnosti nevěnuje moc pozornosti [32].

### 3.1.5 TestComplete

TestComplete je jeden z nástrojů bez kódu, o kterém jsme mluvili dříve. Je to proprietární software, který může automatizovat testování webových, desktopových a mobilních aplikací. Nástroj TestComplete vyvíjí firma SmartBear Software. Testy lze nahrávat zaznamenáváním akcí, psát pomocí klíčových operací nebo vytvářet scripty manuálně [33].

Práce se softwarem je velmi snadná. Testovací scénáře lze jednoduše nahrávat pomocí integrovaného recordéru. Při exekuci testů nechybí možnost porovnávání obrazovek s uloženými screenshoty. Vytvořené testy lze automatizovaně spouštět a chyby zaznamenávat do logu. Software lze také rozšířit o některé pluginy.

TestComplete lze využít na:

- funkční (nebo GUI) testování,
- regresní testování (změna v jedné části ovlivňuje druhou),
- unit testování (testování jednotlivých funkcí),
- distribuované Testování,
- load Testing (zátěžové),
- web Testování,
- coverage Testování (white box),
- data-Driven Testování (databáze),
- manual Testing,
- keyword testing.

Podporované skriptovací jazyky jsou VBScript, JScript, DelphiScript, C++Script a C#Script [33]. Cena licence se pohybuje okolo 4 600 \$.

### 3.1.6 NUnit

NUnit je open source software, který je určen pro tvorbu jednotkových testů pro .NET framework a lze ho bez omezení využívat v bezplatných i komerčních aplikacích nebo knihovnách. Na jeho vývoji se podílí Charlie Poole, Rob Prouse.

Nejdůležitější vlastnosti frameworku:

- spouštění testů z konzole nebo ve Visual Studiu přes Test Adapter,
- testy mohou běžet paralelně,
- silná podpora řízení testů,
- každý testovací případ může být přidán do jedné nebo více kategorií, což umožňuje selektivní spouštění [34].

Testovací metody a třídy jsou pro účely testování aplikace označeny pomocí speciálních atributů. Mezi používané atributy patří:

- Test,
- TestFixture,
- SetUp,
- TearDown.

Dalším důležitým prvkem je třída Assert, která hraje významnou roli v jednotkovém testování. Třída Assert zahrnuje obsáhlou sadu statických metod, které ověřují pravdivost tvrzení. Jednotlivá tvrzení jsou předávána metodám pomocí parametrů. Jestli tvrzení selže, pak je nahlášena chyba [34].

Výsledné jednotkové testy lze spustit v konzoli nebo v aplikaci s grafickým rozhraním.

- **Console Runner** – Program zobrazuje textový výpis a používá příkazový řádek pro běh vytvořených testů. Tento způsob je výhodný pro integraci do jiných systémů. Výsledky jsou uloženy ve formátu XML. Takový formát umožňuje vytvářet reporty a dále výsledky zpracovávat.
- **Visual Studio Test Adapter** – Test Adapter umožňuje spouštět testy uvnitř Visual Studia. Visual Studio s podporou Test Adapteru zobrazuje seznam jednotkových testů. Dále nabízí uživatelům celou řadu funkcí pro správu testů, jako je spouštění, filtrování, vyhledávání nebo vyhodnocování [34].

Roční licence stojí 1 999 \$.

### 3.1.7 Souhrnné hodnocení

Na základě informací popsaných výše je potřeba vybrat automatizované testování, které bude nejlépe vyhovoval všem požadavkům společnosti, zejména těm klíčovým. Základem pro rozhodnutí bude tedy bodový zisk z těchto oblastí.

Tabulka č. 2 zobrazuje souhrnné bodové hodnocení všech pěti testování. Klíčové požadavky jsou ohodnoceny na stupnici od 1 do 5, kdy 1 znamená nejméně vyhovující a 5 nejvíce vyhovující. Každý požadavek má zároveň i určitou váhu pro společnost. Čím vyšší je získané skóre, tím více splňuje konkrétní systém požadavky společnosti IREKS ENZYMA.

Tabulka č. 2: Souhrnné hodnocení (Zdroj: vlastní zpracování)

	Váha	QT	WD	CD	TC	NU
Zrychlení softwarového vývoje	3	4	4	4	4	4
Zrychlení hledání chyb při vývoji	3	5	5	5	5	5
Přijatelná cena a licenční podmínky	2	4	3	3	1	4
Automatizované testování	1	5	5	5	5	5
Zrychlení efektivity práce	2	4	4	2	3	4
Testy prováděné testerem	1	5	5	5	5	5
Cílový programovací jazyk C# a integrace do Visual Studia	2	2	2	1	3	5
<b>Celkové hodnocení</b>		<b>60</b>	<b>58</b>	<b>50</b>	<b>54</b>	<b>68</b>

Vysvětlivky:

- QT – HP QuickTest Professional,
- WD – WebDriver,
- CD – Conformiq Designer,
- TC – TestComplete,
- NU – NUnit.



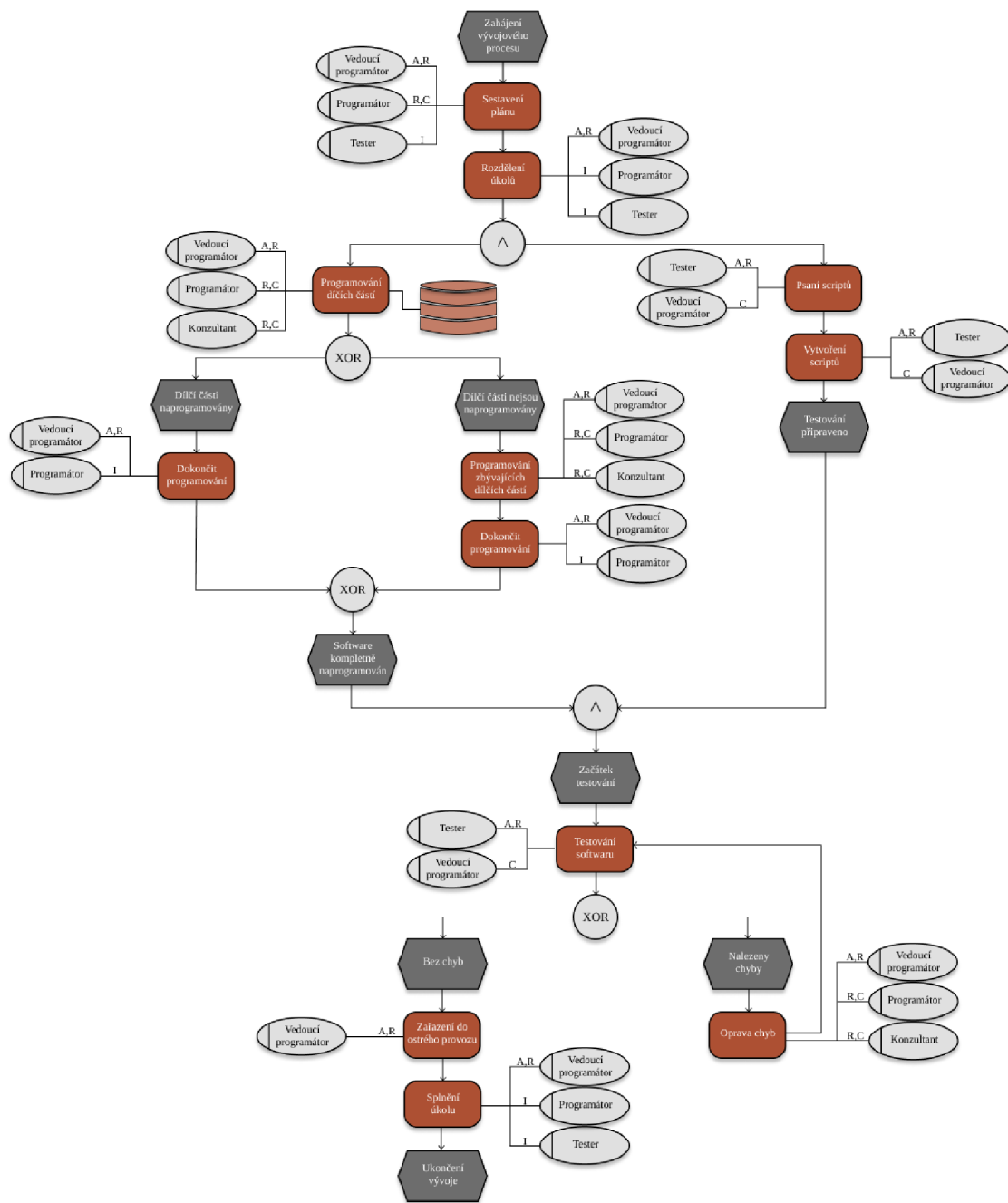
Na základě celkového hodnocení vyplývajícího z tabulky č. 2 lze říci, že nejvíce vyhovujícím testováním je testování jednotek NUnit, které získalo v hodnocení vyšší skóre než zbylé čtyři testování.

### **3.2 Vývojový proces s využitím NUnit**

Vývojový proces IT týmu zůstane stejný jen se do něj přidá automatizované testování.

Opět je vedoucím programátorem zahájena porada IT oddělení, kde se projedná, jaké softwary je potřeba vyřešit. Sestaví se plán projektu, rozdělí se dílčí úkoly a osoby, které za ně budou zodpovědní. Následně se začne s programováním. Tester začne psát a vytvářet skripty pro testování. Pro to, aby mohl napsat vhodné skripty, potřebuje znát kód, který je napsán programátory. Kvůli tomu pomáhá testerovy vedoucí programátor. Až programátoři a tester budou mít dokončenou práci, tak testovací skripty se spustí a provedou kontrolu kódu. Díky aplikaci Visual Studio Test Adapter se chyby zobrazí přímo programátorům ve Visual Studiu.

Pokud nebyly nalezeny chyby, může se výsledný kód spustit do ostrého provozu. Pokud budou nalezeny chyby, tak dojde k následné opravě a celé testování se provede znovu. Jakmile bude sepsaný kód zcela bez chyb, může dojít k ukončení procesu.



Obrázek č. 14: Vývojový proces softwaru s testováním (Zdroj: vlastní zpracování)

### 3.3 Identifikační listina projektu

Identifikační listina projektu zobrazuje ty nejdůležitější informace potřebné k realizaci plánovaného projektu. Mezi nejdůležitější informace patří: jaký je cíl projektu, termíny realizace, náklady a kdo se bude podílet na projektovém týmu. Identifikační listina je znázorněna níže v tabulce č. 3.

**Tabulka č. 3: Identifikační listina projektu** (Zdroj: vlastní zpracování)

IDENTIFIKAČNÍ LISTINA PROJEKTU	
Název projektu	Automatizované testování pro IREKS ENZYMA
Cíl projektu	Implementace automatizovaného testování, který splňuje požadavky zadavatele, při dodržení rozpočtu 200 000 Kč a dokončení do 15.11.2021
Účel projektu	Zefektivnění, zrychlení a zajištění menší chybovosti v IT vývoji
Termín zahájení projektu	1.7.2021
Termín ukončení projektu	15.11.2021
Plánované náklady	200 000 Kč
Vedoucí projektu	Pavla Panáková
Projektový tým	J.P. – vedoucí programátor J.P. – programátor A.M. - konzultant

### 3.4 Milníky projektu

V následující tabulce jsou zaznamenány důležité termíny týkající se zavedení automatizovaného testování.

Milníky slouží jako stručný a jednoduchý přehled všech hlavních termínů.

**Tabulka č. 4: Milníky projektu** (Zdroj: vlastní zpracování)

MILNÍKY PROJEKTU	
Zahájení projektu	1.7.2021
Získání informací	1.7.2021
Alokace zdrojů na projekt	22.7.2021
Návrh testování	27.7.2021
Obsazení pracovní pozice tester	3.8.2021
Implementace testování	4.10.2021
Školení	4.10.2021
Testování systému	25.10.2021
Předání systému	28.10.2021
Ukončení projektu	15.11.2021

### **3.5 Logický rámec**

V této podkapitole je uvedený logický rámec projektu. Obsahuje veškeré informace o projektu, proč jej realizujeme, jaké jsou podmínky a z jakých hlavních bodů se realizace skládá.

Tabulka č. 5: Logický rámec projektu (Zdroj: vlastní zpracování)

	Popis	Objektivně ověřitelné ukazatele	Způsob ověření	Předpoklady/Rizika
<b>Záměr</b>	<ol style="list-style-type: none"> <li>Zefektivnění vývojové práce o 10 %</li> <li>Zrychlení nalezení chyb o 15 %</li> <li>Získání podkladů pro analýzy</li> <li>Snížení chybovosti softwarů</li> <li>Zlepšení výrobního servisu</li> </ol>	<ol style="list-style-type: none"> <li>Počet hotových softwarů</li> <li>Počet nalezených chyb v 1 týdnu</li> <li>Více vypovídající analýzy</li> <li>Počet nalezených chyb</li> <li>Lepší zpětná vazba od zaměstnanců</li> </ol>	<ol style="list-style-type: none"> <li>Výkonnostní reporty</li> <li>Informační systém</li> <li>Čtvrtletní hodnocení společnosti</li> <li>Informační systém</li> <li>Výkonnostní reporty</li> </ol>	
<b>Cíl</b>	<ol style="list-style-type: none"> <li>Implementovat automatizované testování, které splňuje požadavky zadavatele, při dodržení stanoveného rozpočtu 200.000 Kč do 15.11.2021</li> </ol>	<ol style="list-style-type: none"> <li>Nepřekročení rozpočtu</li> <li>Dodržení plánovaného termínu</li> </ol>	<ol style="list-style-type: none"> <li>Dokumentace projektu, faktura</li> <li>Dokumentace projektu, smlouva</li> </ol>	<ul style="list-style-type: none"> <li>Neúspěšná implementace</li> <li>Špatné proškolení uživatelů</li> <li>Odmítnutí systému uživateli</li> <li>Časté chyby systému</li> <li>Změna požadavků na systém</li> </ul>
<b>Výstupy</b>	<ol style="list-style-type: none"> <li>Získání informací</li> <li>Alokace zdrojů na projekt</li> <li>Návrh testování</li> <li>Obsazení pracovní pozice tester</li> <li>Implementace testování</li> <li>Školení</li> <li>Testování systému</li> </ol>	<ol style="list-style-type: none"> <li>Rozpočet projektu nepřesáhne 200.000 Kč</li> <li>Systém bude zavedený do podzimu 2021</li> <li>Systém budou používat 3 lidé</li> <li>Zdroje nepřesáhnou částku 150.000 Kč</li> <li>Kontrolní schůzky na týdenní bázi</li> <li>Nový zaměstnanec</li> <li>Checkpointy na týdenní bázi</li> <li>Uživatelé jsou schopni začít samostatně pracovat</li> <li>Počet úspěšných testů</li> </ol>	<ol style="list-style-type: none"> <li>Faktura za projekt</li> <li>Časový plán projektu, smlouva</li> <li>NUnit</li> <li>Rozpočet projektu, faktura za projekt</li> <li>Zápis ze schůzky</li> <li>Pracovní smlouva</li> <li>Zápis z checkpointů</li> <li>Zápis ze školení</li> <li>Visual Studio</li> </ol>	<ul style="list-style-type: none"> <li>Nedostatečné informace</li> <li>Nedodržení termínu projektu</li> <li>Nedodržení rozpočtu projektu</li> <li>Ztráta finančních prostředků</li> <li>Špatný odhad plánovaných veličin</li> <li>Dodatečná změna požadavků na systém</li> <li>Nedodržení termínu pro dosažení pozice</li> <li>Nedodržení smlouvy 3. stranou</li> <li>Odmítnutí převzetí systému</li> <li>Špatná dokumentace</li> <li>Nedostatečná komunikace</li> </ul>
<b>Klíčové aktivity</b>	<ol style="list-style-type: none"> <li>Provedení analýzy</li> <li>Sepsání analýzy</li> <li>Stanovení termínů</li> <li>Podpis smlouvy</li> <li>Získání lidských zdrojů</li> <li>Získání finančních zdrojů</li> <li>Stanovení požadavků na systém</li> <li>Návrh konkrétního testování</li> <li>Výběr testování</li> <li>Hledání testera</li> <li>Výběrové řízení</li> <li>Vytvoření smlouvy</li> <li>Zajištění hardware vybavení</li> <li>Zakoupení licence</li> <li>Propojení s dosavadním systémem</li> <li>Programování scriptů</li> <li>Konzultace testů</li> <li>Zajištění školitelů</li> <li>Školení uživatelů systému</li> <li>Školení testera</li> <li>Testování testů</li> <li>Testování softwaru</li> </ol>	<ol style="list-style-type: none"> <li>16 hodin</li> <li>40 hodin</li> <li>20 hodin</li> <li>24 hodin</li> <li>12 hodin</li> <li>24 hodin</li> <li>4 hodiny</li> <li>16 hodin</li> <li>16 hodin</li> <li>240 hodin</li> <li>56 hodin</li> <li>16 hodin</li> <li>40 hodin</li> <li>8 hodin</li> <li>24 hodin</li> <li>80 hodin</li> <li>4 hodiny</li> <li>8 hodin</li> <li>8 hodin</li> <li>16 hodin</li> <li>12 hodin</li> <li>8 hodin</li> </ol>	<ol style="list-style-type: none"> <li>2 MD</li> <li>5 MD</li> <li>2,5 MD</li> <li>3 MD</li> <li>1,5 MD</li> <li>3 MD</li> <li>0,5 MD</li> <li>2 MD</li> <li>2 MD</li> <li>30 MD</li> <li>7 MD</li> <li>2 MD</li> <li>5 MD</li> <li>1 MD</li> <li>3 MD</li> <li>10 MD</li> <li>0,5 MD</li> <li>1 MD</li> <li>1 MD</li> <li>2 MD</li> <li>1,5 MD</li> <li>1 MD</li> </ol>	<ul style="list-style-type: none"> <li>Nedostatečná analýza</li> <li>Vznik dodatečných požadavků</li> <li>Nedostatek lidských zdrojů</li> <li>Nedostatek finančních zdrojů</li> <li>Vzájemné nepochopení smluvních stran</li> <li>Špatný odhad zdrojů</li> <li>Špatný časový odhad</li> <li>Neodsouhlasení návrhu</li> <li>Nedostatečné školení</li> <li>Nekvalitní dokumentace systému</li> </ul>
<b>Co projekt neřeší</b>	<ol style="list-style-type: none"> <li>Podmínky následné údržby implementovaného systému</li> <li>Další rozvoj systému po ukončení projektu</li> </ol>		<b>Předběžné podmínky</b>	<ol style="list-style-type: none"> <li>Prostor ve firmě pro nového zaměstnance</li> </ol>

### 3.6 WBS projektu

Tabulka č. 6 zobrazuje WBS projektu. Při vytváření WBS byly identifikovány veškeré činnosti, které je nutné vykonat pro úspěšnou realizaci celého projektu. Pro větší přehlednost byla WBS zpracována i graficky.

Tabulka č. 6: Kroky k implementaci (Zdroj: vlastní zpracování)

IMPLEMENTACE	
Úkol	Popis úkolu
1.	Získání informací
1.1	Provedení analýzy
1.2	Sepsání analýzy
1.2.1	Představení analýzy
1.2.2	Schválení analýzy
1.3	Stanovení termínů
1.3.1	Návrh termínů
1.3.2	Odsouhlasení termínů
1.4	Podpis smlouvy
1.4.1	Představení smlouvy
1.4.2	Připomínkování smlouvy
1.4.3	Podpis smlouvy
2.	Alokace zdrojů na projekt
2.1	Získání lidských zdrojů
2.1.1	Získání interních lidských zdrojů
2.1.2	Získání externích lidských zdrojů
2.1.3	Stanovení projektového týmu
2.2	Získání finančních zdrojů
2.2.1	Stanovení rozpočtu
2.2.2	Stanovení rezervy
3.	Návrh testování
3.1	Stanovení požadavků na testování

<b>Úkol</b>	<b>Popis úkolu</b>
3.1.1	Shrnutí zásadních požadavků
3.1.2	Odsouhlasení zásadních požadavků
3.2	Návrh konkrétního testování
3.2.1	Hledání informací o testování
3.2.2	Hledání vhodného řešení
3.3	Výběr testování
3.3.1	Představení testování a jeho funkčnosti
3.3.2	Výběr a odsouhlasení nejvhodnějšího testování
4.	Obsazení pracovní pozice tester
4.1	Vystavení inzerátu
4.1.1	Vytvoření inzerátu
4.1.2	Vložení inzerátu na internet
4.2	Výběrové řízení
4.2.1	Pohovory uchazečů
4.2.2	Výběr vhodného adepta
4.2.3	Přijmutí uchazeče
4.3	Vytvoření smlouvy
4.3.1	Sepsání smlouvy
4.3.2	Podpis smlouvy
4.4	Zajištění hardware vybavení
4.4.1	Výběr hardwaru
4.4.2	Koupě hardwaru
5.	Implementace testování
5.1	Zakoupení licence
5.1.1	Zakoupení licence NUnit
5.1.2	Nainstalování Visual Studio Test Adapter
5.2	Propojení s používaným informačním systémem
5.3	Psaní scriptů
5.3.1	Psaní scriptů

<b>Úkol</b>	<b>Popis úkolu</b>
5.3.2	Programování scriptů
5.4	Konzultace testů
5.4.1	Konzultace, zda vše je, jak má
6.	Školení uživatelů
6.1	Zajištění školitelů
6.2	Školení uživatelů
6.3	Školení testera
7.	Testování systému
7.1	Testování testů
7.2	Testování softwaru





Obrázek č. 15: Grafické znázornění WBS (Zdroj: vlastní zpracování)

### 3.7 Projektový tým (RACI)

Pro realizaci projektu byl sestaven tým, který se skládá z 5 členů. Každý z těchto členů má na projektu na starosti určitou oblast, za kterou je zodpovědný.

Jednu z největších zodpovědností má vedoucí projektu, jehož základním úkolem je dohlížet na realizaci projektu a zajišťovat jeho hladký průběh.

Dalším členem týmu je konzultant, který je schopen poskytovat cenné rady, jak vedoucímu projektu, tak také ostatním členům týmu. Dále se zodpovídá za zaškolení uživatelů.

Třetím členem týmu je tester. Primárním úkolem testera na projekt je správné naprogramování testování a jeho zakomponování do systému. Dále je zodpovědný za funkčnost testování.

Čtvrtým členem je vedoucí programátor, který má na starost celkový návrh testování, jeho výběr a implementaci. S jeho funkcí souvisí i výběr vhodného testera. Dalším úkolem je programování v návaznosti na UX/UI implementovaného řešení.

Posledním členem týmu je programátor, jehož úkolem je podílet se společně s vedoucím programátorem na výběru vhodného testování a v rámci této oblasti poskytovat vedoucímu rady.

#### 3.7.1 RACI matice projektu

V RACI matici projektu jsou zobrazeni jednotliví členové projektového týmu a konkrétní činnosti, za které jsou zodpovědní. Tato matice je znázorněna v tabulce č. 7.

RACI matice jsou označena následující pravidla:

- R – Responsible – osoba, která je odpovědná za vykonání svěřeného úkolu,
- A – Accountable – osoba, která je odpovědná za celý úkol, ale nemusí jej vykonávat přímo tato osoba,
- C – Consulted – osoba, která může poskytnout cennou radu či konzultaci,
- I – Informed – osoba, která má být informována o průběhu úkolu [22].

Tabulka č. 7: RACI matice projektu (Zdroj: vlastní zpracování)

Úkol	Vedoucí projektu	Konzultant	Tester	Vedoucí programátor	Programátor
1.1	A, R			C	
1.2.1	A, R			C	
1.2.2	A	I		R	
1.3.1	A, R	R		C	C
1.3.2	A	I		R, C	I
1.4.1	A, R	I		I	I
1.4.2	A, R	C		C, I	I
1.4.3	A, R	I		R	
2.1.1	A, R	C		C	C
2.1.2	A, R	C		C, I	C, I
2.1.3	A, R	I		I	
2.2.1	A	R		I	
2.2.2	A, R	C		I	
3.1.1	A, R	I		I	I
3.1.2	A, R	C		C	
3.2.1	A, I	C		R	R
3.2.2	A, I	C		R	R
3.3.1	A, R	I		I	I
3.3.2	A, R	I		R	
4.1.1	A, R	C		C, I	
4.1.2	A, R	I		I	
4.2.1	A		I	R	R
4.2.2	A	C	I	R	R
4.2.3	I, C		I	A, R	I
4.3.1	I, C		I	A, R	
4.3.2	I, C		R	A, R	

Úkol	Vedoucí projektu	Konzultant	Tester	Vedoucí programátor	Programátor
4.4.1	A	C		R	C
4.4.2	A			R	I
5.1.1	I	C		R, A	
5.1.2	I	C		R, A	
5.2	I	C		A, R	C
5.3.1	I		R	A	C
5.3.2	I		R	A	C
5.4.1	I	C	R	A	I
6.1	A, R	C		I	I
6.2	A, I	R		R	R
6.3	A, I	R	R	I	I
7.1	I	C	R	A, R	C
7.2	I	C	R	A, R	C

### 3.8 Analýza rizik projektu

V této podkapitole je popisována analýza rizik projektu pomocí metody RIPRAN. Při vyhodnocování velikosti rizika bylo postupováno podle tabulky č. 8.

**Tabulka č. 8: Kritéria vyhodnocení rizik** (Zdroj: vlastní zpracování)

	Malý dopad (MD)	Střední dopad (SD)	Velký dopad (VD)
Nízká pravděpodobnost (NP)	Nízká hodnota rizika (NHR)	Nízká hodnota rizika (NHR)	Střední hodnota rizika (SHR)
Střední pravděpodobnost (SP)	Nízká hodnota rizika (NHR)	Střední hodnota rizika (SHR)	Vysoká hodnota rizika (VHR)
Vysoká pravděpodobnost (VP)	Střední hodnota rizika (SHR)	Vysoká hodnota rizika (VHR)	Vysoká hodnota rizika (VHR)

### 3.8.1 Identifikace a zhodnocení rizik

Nejdříve byly identifikovány všechny hrozby, které by svým působením mohly způsobit incidenty. Následně byla ke každému riziku stanovena pravděpodobnost jeho výskytu a také dopad, který by mělo. Na základě těchto údajů byla vyhodnocena velikost rizika. Všechny identifikovaná rizika zobrazuje tabulka č. 9. Písmeno P je zkratkou pro pravděpodobnost a písmeno D pro dopad.

**Tabulka č. 9: Rizika projektu** (Zdroj: vlastní zpracování)

Č.	Hrozba	Scénář	P	D	Hodnota rizika
1	Výběr špatných členů týmu	Nespolehlivost, špatně odvedená práce, potřebná oprava a prodloužení činností	SP	VD	VHR
2	Špatně provedená analýza	Absence důležitých funkcionalit	SP	VD	VHR
3	Nedodržení smluvních podmínek	Odstoupení od smlouvy	NP	VD	SHR
4	Nedodržení termínů	Zpoždění konečného termínu projektu	SP	SD	SHR
5	Nedostatek finančních prostředků	Zrušení nebo pozastavení projektu	NP	VD	SHR
6	Nedostatek zdrojů pro implementaci	Prodloužení doby implementace, nutné získat dodatečné zdroje	NP	SD	NHR
7	Špatná komunikace v týmu	Neshody v týmu, prodloužení projektu	SP	SD	SHR
8	Špatný výběr testera	Nutná výměna testera nebo proškolení	SP	VD	VHR
9	Špatně naprogramované testy	Prodloužení trvání projektu	SP	VD	VHR
10	Ztráta dokumentace, dat	Prodloužení některých činností projektu	NP	SD	NHR
11	Nekontrolování kritických činností	Prodloužení trvání projektu	SP	SD	SHR
12	Absence některého z členů týmu	Prodloužení projektu	SP	SD	SHR
13	Nesplnění předběžných podmínek	Posunutí začátku využívání systému	NP	VD	SHR
14	Nedostatečné proškolení zaměstnanců	Nízká efektivita práce se systémem	NP	VD	SHR
15	Velká chybovost systému	Systém často padá a je nefunkční	SP	VD	VHR

### 3.8.2 Opatření pro snížení rizik

V další tabulce jsou uvedeny návrhy na opatření k jednotlivým rizikům projektu. Pro všechny identifikované hrozby byla následně zvolena relevantní opatření, která vedla ke snížení buď pravděpodobnosti výskytu rizika nebo snížení jeho dopadu a tím také k celkovému snížení hodnoty rizika. Písmeno P je zkratkou pro pravděpodobnost a písmeno D pro dopad.

**Tabulka č. 10: Návrhy opatření rizik a nová hodnota rizik** (Zdroj: vlastní zpracování)

Č.	Návrh na opatření	P	D	Nová hodnota rizika
1	Dbát na kvalitní a rozumný výběr členů týmu, především na základě dosavadních zkušeností	NP	VD	SHR
2	Prezentace analýzy a následně její odsouhlasení	NP	SD	NHR
3	Kvalitní provedení analýzy, precizní plán projektu	NP	VD	SHR
4	Vytvoření precizního plán projektu	NP	SD	NHR
5	Zaplacení zálohy před zahájením realizace, mít rezervu	NP	VD	SHR
6	Reorganizace zdrojů, nabírání dalších zdrojů	NP	SD	NHR
7	Lepší komunikace při řešení problému, mluvit o všem	NP	SD	NHR
8	Zlepšení výběrového řízení	NP	VD	SHR
9	Preciznost při programování	NP	VD	SHR
10	Vyhotovení a zakončení dokumentace po každé činnosti, pravidelná záloha	NP	SD	NHR
11	Správná kontrola činností a hlášení jejich zpoždění	NP	SD	NHR
12	Zvolit zástupce za chybějícího člena týmu, možnost práce i z domu	NP	SD	NHR
13	Představení nezbytných požadavků	NP	MD	NHR
14	Důkladné školení a poskytnutí školících materiálů	NP	VD	SHR
15	Důkladné testování, výkonný HW a správný výběr systému	NP	VD	SHR

### 3.8.3 Shrnutí analýzy rizik

Celkem se identifikovalo 15 rizik, které nejvíce ohrožují projekt.

Po zavedení opatření došlo ke snížení hodnot všech rizik, nicméně i tak zůstalo 7 scénářů s hodnotou „Střední hodnota rizika (SHR)“. Na tyto rizika je při realizaci projektu brát větší zřetel a více se jim věnovat.

Při realizaci projektu je nutné nadále sledovat všechny stávající rizika, ale i sledovat, zda se nevyskytly nová rizika, která by musela být dodatečně zahrnuta do analýzy.

## 3.9 Časová analýza projektu

Tato podkapitola se zabývá časovou náročností celého projektu. U každé činnosti, která musí být vykonána pro úspěšnou realizaci projektu, byla nejdříve stanovena doba trvání v MD, její předchůdci a v návaznosti na to také datum zahájení i dokončení této činnosti.

Pro lepší zobrazení časové posloupnosti jednotlivých činností byl vytvořen Ganttův diagram, který zobrazuje obrázek č. 16.

Tabulka č. 11: Doby trvání činností projektu (Zdroj: vlastní zpracování)

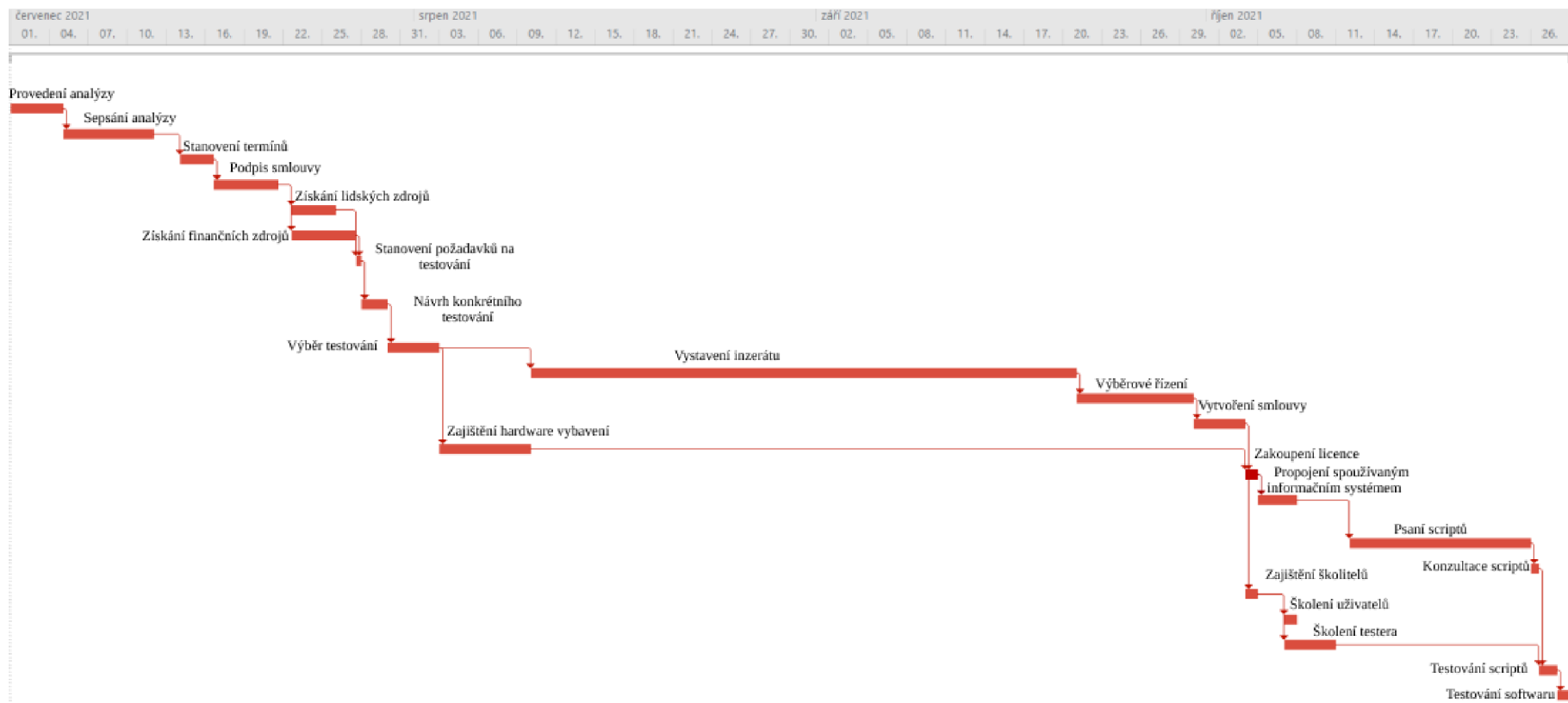
Úkol	Název úkolu	Doba trvání (MD)	Zahájení	Dokončení	Předchůdci
<b>1.</b>	<b>Získání požadavků od zákazníka</b>		<b>01.07. 2021</b>	<b>21.07. 2021</b>	
1.1	Provedení analýzy	2 dny	01.07. 2021	05.07. 2021	
1.2	Sepsání analýzy	5 dní	05.07. 2021	12.07. 2021	1.1
1.3	Stanovení termínů	2,5 dny	14.07. 2021	16.07. 2021	1.2
1.4	Podpis smlouvy	3 dny	16.07. 2021	21.07. 2021	1.3
<b>2.</b>	<b>Alokace zdrojů na projekt</b>		<b>22.07. 2021</b>	<b>27.07. 2021</b>	
2.1	Získání lidských zdrojů	1,5 dne	22.07. 2021	26.07. 2021	1.4
2.2	Získání finančních zdrojů	3 dny	22.07. 2021	27.07. 2021	1.4
<b>3.</b>	<b>Návrh testování</b>		<b>27.07. 2021</b>	<b>03.08. 2021</b>	

3.1	Stanovení požadavků na testování	0,5 dne	27.07. 2021	28.07. 2021	2.1, 2.2
3.2	Návrh konkrétního testování	2 dny	28.07. 2021	30.07. 2021	3.1
3.3	Výběr testování	2 dny	30.07. 2021	03.08. 2021	3.2
<b>4.</b>	<b>Obsazení pracovní pozice tester</b>		<b>03.08. 2021</b>	<b>25.10. 2021</b>	
4.1	Vystavení inzerátu	30 dní	10.08. 2021	21.09. 2021	3.3
4.2	Výběrové řízení	7 dní	21.09. 2021	30.09. 2021	4.1
4.3	Vytvoření smlouvy	2 dní	30.09. 2021	04.10. 2021	4.2
4.4	Zajištění hardware vybavení	5 dní	03.08. 2021	10.08. 2021	3.3
<b>5.</b>	<b>Implementace testování</b>		<b>04.10. 2021</b>	<b>26.10. 2021</b>	
5.1	Zakoupení licence	1 den	04.10. 2021	05.10. 2021	4.3, 4.4
5.2	Propojení s používaným informačním systémem	3 dny	05.10. 2021	08.10. 2021	5.1
5.3	Psaní scriptů	10 dní	12.10. 2021	26.10. 2021	5.2
5.4	Konzultace testů	0,5 dne	26.10. 2021	26.10. 2021	5.3
<b>6.</b>	<b>Školení uživatelů</b>		<b>04.10. 2021</b>	<b>11.10. 2021</b>	
6.1	Zajištění školitelů	1 den	04.10. 2021	05.10. 2021	4.4
6.2	Školení uživatelů	1 den	07.10. 2021	08.10. 2021	6.1
6.3	Školení testera	2 dny	07.10. 2021	11.10. 2021	6.1
<b>7.</b>	<b>Testování systému</b>		<b>26.10. 2021</b>	<b>29.10. 2021</b>	
7.1	Testování scriptů	1,5 dne	26.10. 2021	28.10. 2021	5.4, 6.3
7.2	Testování softwaru	1 den	28.10. 2021	29.10. 2021	7.1



Z časové analýzy vyplynulo, že celková časová náročnost projektu je 86 MD. Pokud bude projekt zahájen podle plánu 1.7.2021 měl by být dokončen 29.10.2021. Vzhledem k tomu, že cílem projektu je zavedení systému do 15.11.2021 je zde ještě rezerva 11 pracovních dní.

Jak lze vidět z Ganttova diagramu projektu velká část činností leží na kritické cestě. Z tohoto důvodu je projekt poměrně náchylný na vznik zpoždění, stačí aby se některá z činností na kritické cestě opozdila a dojde ke zpoždění celého projektu. Proto je velmi důležité klást velký důraz na dodržování termínů, zahajovat činnosti ihned jak je to možné a v případě nutnosti navýšit kapacity, tak aby nedošlo ke zpoždění.



Obrázek č. 16: Ganttův diagram (Zdroj: vlastní zpracování)

### 3.10 Ekonomické zhodnocení projektu

V této části se zabírám ekonomickým zhodnocením výběru a implementace NUnit v rámci projektu popsaného v předchozích kapitolách.

Mezi celkové náklady patří náklady na vybavení pracoviště pro Testera, které zahrnují vybavení nábytku a hardware. Další položkou je zaplacení inzerátu hledající uchazeče na pozici testera. Tester dostane bonus pro přijetí. Následně je potřeba zaplatit školitele, kteří dostatečně proškolí uživatele testování a testera.

Automatizované testování NUnit prodává svou licenci buď na 3, 6 nebo 12 měsíců. IREKS ENZYMA zvolila licenci na rok a postupně ji bude prodlužovat. Pro lepší pracování je potřeba získat i program Visual Studio Test Adapter, který umožňuje spouštět testy uvnitř Visual Studia. Tato položka však nic nestojí, jelikož program je zdarma.

Do nákladů se nezapočítávají mzdové náklady projektového týmu.

**Tabulka č. 12: Celkové náklady** (Zdroj: vlastní zpracování)

Název položky	Cena
Vybavení kanceláře – nábytek	20 000 Kč
Hardware vybavení	60 000 Kč
Webový inzerát	2 000 Kč
Školení	15 000 Kč
NUnit licence	1 999 \$/ 42 000 Kč
Visual Studio Test Adapter	0 Kč
Tester	20 000 Kč
Finanční rezerva	20 000 Kč
<b>Celkem</b>	<b>197 000 Kč</b>

Celkové náklady na implementaci řešení tedy činí 197 000 Kč. Mezi náklady je započtena i finanční rezerva ve výši 20 000 Kč.

### 3.10.1 Návratnost investice

Pro zjištění prosté doby návratnosti investice bude použit vzorec pro výpočet průměrné doby návratnosti [35]:

$$T_s = \frac{IN}{CF}$$

$T_s$ ...doba návratnosti

IN...počáteční investice

CF...průměrná roční úspora nákladů

Podle údajů získaných z výroční zprávy za rok 2016, 2017, 2018 a 2019 je průměrný roční výnos 756 287 Kč. Výsledná rovnice vypadá následovně [36].

$$T_s = \frac{197\ 000}{756\ 287} = 0,26 \text{ roku}$$

Doba návratnosti investice do projektu se vrátí za 0,26 roku, v přepočtu přibližně za 3,1 měsíce.

### 3.11 Přínosy projektu

Očekávanými přínosy zavedení automatizovaného testování do softwarového vývoje bude minimalizování nebo případně i odstranění nedostatků, které se momentálně ve vývoji nachází.

Hlavním přínosem je zavedení řízeného testování softwaru, které ve firmě doposud chybělo nebo bylo prováděno jen když bylo potřeba, a to ještě manuálně. Velkým benefitem jsou automatizované testy, které zjednoduší celkové testování. Budou vytvořeny skripty, podle kterých bude software otestován a zkontrolován rychleji, než tomu bylo doposud. Chyby se tak mohou snáz odhalit a opravit.

Dalším přínosem je přibrání nového člena týmu na pozici testera. Bude mít na starost celou oblast testování softwaru a správu nástroje pro automatizované testování. Ostatní členové týmu budou mít více prostoru pro samotný vývoj softwaru a minimalizují jeho chybovost.

V neposlední řadě se díky zdokonalení vývoje může zvýšit i výroba. Jelikož dojde ke zlepšení vývojového prostředí v IT oddělení, které vyvíjí softwary i pro výrobní oddělení,

bude mít tento dopad pozitivní vliv na efektivitu práce ve výrobě. Následkem toho může být větší počet objednávek.

Celkově by tedy po zavedení automatizovaného testování mělo dojít ke zjednodušení a zefektivnění práce v IT oddělení. Změna by se měla stát příjemnější pro všechny pracovníky.

## ZÁVĚR

Hlavním cílem této diplomové práce bylo seznámit se srozumitelnou formou s testováním softwaru, zařadit testování do životního cyklu vývoje softwaru, popsat možnosti automatizovaného testování, včetně výběru vhodných nástrojů a jejich následná implementace do softwarového vývoje firmy IREKS ENZYMA s.r.o.

Celá práce byla rozdělena do tří hlavních částí. První část se věnovala teoretickým východiskům pro testování softwaru a okrajově projektového managementu.

Druhá část je analytická. Zde byla provedena analýza současného stavu firmy pomocí analýzy vnějšího a vnitřního prostředí. Na základě těchto analýz vyplynulo, že ve firmě úplně chybí testování softwaru. Z toho důvodu bylo také rozhodnuto o výběru a implementaci automatizovaného testování do softwarového vývoje.

Výběrem a implementací se zabývá část třetí, kde bylo postupně popsáno pět variant testování: HP QuickTest Professional, Selenium WebDriver, Conformiq Designer, TestComplete a NUnit. U všech pěti variant byla zvážena nejzásadnější kritéria a požadavky společnosti, na základě kterých bylo následně vybráno nejvhodnější testování NUnit.

Nový proces vývoje s využíváním automatizovaného testování bude umožňovat zvýšení kvality a efektivity práce programátorů.

V rámci implementace NUnit byl vypracován projektový plán, který naplánoval celkovou časovou náročnost na 86 MD. Projekt by začal 1.7.2021, jak je plánováno a skončil by 29.10.2021, s tím že plánované dokončení je 15.11.2021, čili projekt má rezervu 11 pracovních dní.

V rámci projektového plánu byla provedena také analýza rizik, ze které vyplynulo 15 nalezených hrozeb, které by mohli zkomplikovat průběh implementace projektu. K těmto rizikům byla následně sepsána opatření, která snížila jejich rizikovou hodnotu.

Finanční náklady na projekt činí 197 000 Kč, ve kterých je zahrnuta i finanční rezerva ve výši 20 000 Kč. Doba návratnosti investice byla vypočtena na přibližně 3 měsíce.

Hlavní cíl práce byl dosažen splněním dílčích úkonů. Do firmy bude implementováno automatizované testování softwaru pomocí projektového managementu. Byl vytvořen kompletní návrh projektu, kterým se firma bude řídit a díky kterému dosáhne požadovaného cíle.

## SEZNAM ZDROJŮ

- [1] *Total number of Websites - Internet Live Stats* [online]. 2021 [cit. 2021-05-15]. Dostupné z: <https://www.internetlivestats.com/total-number-of-websites/#trend>
- [2] GALIN, Daniel a Anna HAVLÍČKOVÁ. *Software quality assurance: průvodce testováním*. 1. vyd. New York: Pearson Education Limited, 2004, 590 p. ISBN 02-017-0945-7.
- [3] ROUDENSKÝ, Petr a Anna HAVLÍČKOVÁ. *Řízení kvality softwaru: průvodce testováním*. Brno: Computer Press, 2013. ISBN 978-80-251-3816-8.
- [4] PATTON, Ron. *Testování softwaru*. Praha: Computer Press, 2002. Programování. ISBN 80-722-6636-5.
- [5] *Manual Testing Interview Question: Difference Between Error, Defect, and Failure* [online]. Copyright © 2021 Abode QA, 2021 [cit. 2021-04-06]. Dostupné z: <https://abodeqa.com/manual-testing-interview-question-difference-between-error-defect-and-failure/>
- [6] *Deset největších softwarových chyb v historii lidstva - Root.cz* [online]. 2021 [cit. 2021-05-14]. Dostupné z: <https://www.root.cz/clanky/deset-nejvetsich-softwarovych-chyb-v-historii-lidstva/>
- [7] *Manual testing the pros and cons* [online]. Main St., ME: ©Base36, Inc. [cit. 2021-04-06]. Dostupné z: <http://www.base36.com/automatedvs-manual-testing-the-pros-and-cons-of-each>
- [8] *Grey box test - CleverAndSmart Management Consulting* [online]. Čermák, 2010 [cit. 2021-05-14]. Dostupné z: <https://www.cleverandsmart.cz/grey-box-test/>
- [9] *Software testing: Levels of testing* [online]. Kavuri Hills: © Copyright 2020., 2020 [cit. 2021-04-06]. Dostupné z: [https://www.tutorialspoint.com/software\\_testing/levels\\_of\\_testing](https://www.tutorialspoint.com/software_testing/levels_of_testing)
- [10] *Types of Software Testing: Different Testing Types with Details* [online]. © COPYRIGHT SOFTWARETESTINGHELP 2021, 2021 [cit. 2021-04-06]. Dostupné z: <https://www.softwaretestinghelp.com/types-of-software-testing/>
- [11] *Testování SW - CleverAndSmart Management Consulting* [online]. ČR: © 2008 - 2021, 2009 [cit. 2021-04-06]. Dostupné z: <https://www.cleverandsmart.cz/testovani-sw/>
- [12] *Typy testování - CleverAndSmart Management Consulting* [online]. Čermák, 2011 [cit. 2021-05-14]. Dostupné z: <https://www.cleverandsmart.cz/typy-testu/>



- [13] *PESTLE analýza - ManagementMania.com* [online]. ČR: Copyright © 2011-2016, 2015 [cit. 2021-04-06]. Dostupné z: <https://managementmania.com/cs/pestle-analyza>
- [14] *Analýza pěti sil 5F (Porter's Five Forces)* [online]. Copyright © 2011-2016, 2016 [cit. 2021-04-07]. Dostupné z: <https://managementmania.com/cs/analyza-5f>
- [15] *Tesco McKinsey 7S Model - Research-Methodology: Tesco McKinsey 7S Model* [online]. Necessary knowledge to conduct a business research, 2019 [cit. 2021-04-06]. Dostupné z: <https://research-methodology.net/tesco-mckinsey-7s-model/>
- [16] *Typy organizačních struktur a jejich členění | Ostatní účelové organizační struktury | BusinessInfo.cz* [online]. ČR: © 1997-2021 CzechTrade, 2020 [cit. 2021-04-06]. Dostupné z: <https://www.businessinfo.cz/navody/typy-organizacnich-struktur-cleneni/>
- [17] KOPFOVÁ, Alena. *Organizační struktury*. Brno, 2020. Dostupné také z: [https://is.muni.cz/el/1422/podzim2012/MV502K/um/org\\_struktury\\_bez\\_obrazku.pdf](https://is.muni.cz/el/1422/podzim2012/MV502K/um/org_struktury_bez_obrazku.pdf). Prezentace. Muni.
- [18] *SWOT analýza*. Brno, 2020. Dostupné také z: [http://user.mendelu.cz/xbadal/Studijni%20opory/Hospodarska%20informatika/Stud\\_mat/SWOT%20anal%FDza.pdf](http://user.mendelu.cz/xbadal/Studijni%20opory/Hospodarska%20informatika/Stud_mat/SWOT%20anal%FDza.pdf)
- [19] *EPC Diagram - BPMN Diagrams - Unified Modeling Language Tool* [online]. [cit. 2021-05-04]. Dostupné z: <https://www.visual-paradigm.com/VPGallery/bpmodeling/epc.html>
- [20] DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO. *Projektový management podle IPMA. 2., aktualiz. a dopl. vyd.* Praha: Grada, 2012. Expert (Grada). ISBN 978-80-247-4275-5.
- [21] SVOZILOVÁ, Alena. *Projektový management. 2., aktualiz. a dopl.* Praha: Grada, 2011. ISBN 978-80-247-3611-2.
- [22] *RACI matice - PM Consulting* [online]. Praha: © PM Consulting [cit. 2021-04-06]. Dostupné z: <https://www.pmconsulting.cz/slovníkovy-pojem/raci-matice/>
- [23] *RIPRAN - Metoda pro analýzu projektových rizik* [online]. Praha, 2020 [cit. 2021-04-07]. Dostupné z: <https://ripran.cz/>
- [24] *Prezentace dat v Ganttově diagramu v aplikaci Excel* [online]. Microsoft, 2021 [cit. 2021-04-07]. Dostupné z: <https://support.microsoft.com/cs-cz/topic/prezentace-dat-v-ganttov%20c4%9b-diagramu-v-aplikaci-excel-f8910ab4-ceda-4521-8207-f0fb34d9e2b6?ui=cs-cz&rs=cs-cz&ad=cz>

- [25] *IREKS ENZYMA | Suroviny a technologie pro profesionální pekaře a cukráře.* [online]. Brno: © Copyright 2021 by IREKS GmbH, 2021 [cit. 2021-04-05]. Dostupné z: <http://www.ireks-enzyma.cz/Cesky.htm>
- [26] *Veřejný rejstřík a Sbirka listin - Ministerstvo spravedlnosti České republiky* [online]. ČR: © 2012-2015 Ministerstvo spravedlnosti České republiky, c2012-2015 [cit. 2021-04-05]. Dostupné z: <https://or.justice.cz/ias/ui/rejstrik-firma.vysledky?subjektId=462999&typ=UPLNY>
- [27] *Jak se vyvíjelo DPH v Čechách od roku 1993* [online]. 2016 [cit. 2021-04-18]. Dostupné z: <https://mladypodnikatel.cz/vyvoj-dph-v-cechach-od-1993-t3893>
- [28] *Nezaměstnanost v ČR, vývoj, rok 2021, Míra nezaměstnanosti v %* [online]. Praha: Kurzy.cz, AliaWeb, 2021 [cit. 2021-04-07]. Dostupné z: <https://www.kurzy.cz/makroekonomika/nezamestnanost/>
- [29] *Předpisy v roce 2021* [online]. 2021 [cit. 2021-04-18]. Dostupné z: <https://www.zakonyprolidi.cz/cs/rocnik/2021>
- [30] *HP QuickTest Professional* [online]. © 2012 PowerTest Software Solutions, 2012 [cit. 2021-04-27]. Dostupné z: <https://www.powertest.com/software-functional-testing-hp-quicktest-professional.html>
- [31] *What is Selenium? Introduction to Selenium Automation Testing* [online]. 2021 [cit. 2021-04-27]. Dostupné z: <https://www.guru99.com/introduction-to-selenium.html#10>
- [32] *CONFORMIQ DESIGNER* [online]. Conformiq, 2021 [cit. 2021-04-27]. Dostupné z: <https://www.conformiq.com/products/conformiq-designer/>
- [33] *TestComplete* [online]. 2021 [cit. 2021-04-27]. Dostupné z: <http://testovanisofwaru.cz/automatizovane-testovani/testcomplete/>
- [34] *What Is NUnit?* [online]. Charlie Poole, Rob Prouse, 2019 [cit. 2021-04-27]. Dostupné z: <https://nunit.org/>
- [35] *Doba návratnosti: vysvětleno na příkladu zateplení | Projekty INKAPO* [online]. České Budějovice [cit. 2021-04-28]. Dostupné z: <https://www.projekty-inkapo.cz/doba-navratnosti/>
- [36] *Veřejný rejstřík a Sbirka listin - Ministerstvo spravedlnosti České republiky* [online]. © 2012-2015 Ministerstvo spravedlnosti České republiky, c2012-2015 [cit. 2021-04-28]. Dostupné z: <https://or.justice.cz/ias/ui/vypis-sl-firma?subjektId=462999>

## SEZNAM OBRÁZKŮ

Obrázek č. 1: V-model.....	21
Obrázek č. 2: Porterův model konkurenčních sil.....	26
Obrázek č. 3: Model 7S .....	27
Obrázek č. 4: Funkcionální organizační struktura .....	28
Obrázek č. 5: Divizionální organizační struktura .....	29
Obrázek č. 6: Maticová organizační struktura .....	29
Obrázek č. 7: SWOT analýza .....	30
Obrázek č. 8: Značky EPC diagramu.....	31
Obrázek č. 9: Příklad Ganttového diagramu.....	34
Obrázek č. 10: Logo firmy .....	36
Obrázek č. 11: Organizační struktura .....	37
Obrázek č. 12: SWOT analýza společnosti .....	44
Obrázek č. 13: Vývojový proces softwaru.....	45
Obrázek č. 14: Vývojový proces softwaru s testováním .....	54
Obrázek č. 15: Grafické znázornění WBS.....	61
Obrázek č. 16: Ganttův diagram .....	70

## SEZNAM TABULEK

Tabulka č. 1: Srovnání testování.....	23
Tabulka č. 2: Souhrnné hodnocení .....	52
Tabulka č. 3: Identifikační listina projektu.....	55
Tabulka č. 4: Milníky projektu .....	56
Tabulka č. 5: Logický rámec projektu .....	57
Tabulka č. 6: Kroky k implementaci .....	58
Tabulka č. 7: RACI matice projektu.....	63
Tabulka č. 8: Kritéria vyhodnocení rizik .....	64
Tabulka č. 9: Rizika projektu.....	65
Tabulka č. 10: Návrhy opatření rizik a nová hodnota rizik .....	66
Tabulka č. 11: Doby trvání činností projektu .....	67
Tabulka č. 12: Celkové náklady .....	71

## **SEZNAM GRAFŮ**

Graf č. 1: Vývoj průměrné hrubé měsíční mzdy .....	38
--	----

## SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

Apod.	A podobně
Atd.	A tak dále
ČR	Česká republika
DPH	Daň z přidané hodnoty
EPC	Event-Driven Process Chain
GDPR	General data protection regulation
GUI	Graphical User Interface
IČO	Identifikační číslo osoby
IEEE	Institute of Electrical and Electronics Engineers
IT	Informační technologie
MD	Man-day
RIPRAN	RISk PROJect ANALYSIS
Sb.	Sbírka
s.r.o.	Společnost s ručením omezeným
UI	User interface
UX	User experience
WBS	Work breakdown structure
QML	Qt Modelling Language
XML	eXtensible Markup Language