

**Česká zemědělská univerzita v Praze**  
**Provozně ekonomická fakulta**  
**Katedra informačního inženýrství**



**Diplomová práce**

**Návrh a otestování řídicího systému pro zlepšování  
kvality software**

Autor Práce: Bc. Michal Fišer

Vedoucí práce: Ing. Josef Pavlíček, Ph.D.

© 2018 ČZU v Praze

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Michal Fišer

Informatika

Název práce

**Návrh a otestování řídicího systému pro zlepšování kvality software**

Název anglicky

**Design and testing of a software quality management system**

---

### Cíle práce

Hlavním cílem práce je navrhnout UI specifikaci systému pro řízení kvality software. Součástí práce bude tvora uživatelských testovacích scénářů. Ty budou mít za úkol odhalit nedostatky systému. Po ukončení kvalitativního šetření UI specifikace budou výsledky vyhodnoceny. Na základě analýzy výsledků budou definovány závěry a doporučení k realizaci.

### Metodika

Metodika práce je založena na analýze odborné literatury – rešerše, analýze existujících softwarových řešení a konzultacích s profesionály z oboru řízení kvalit software. Na základě nabytých znalostí a poznatků bude vytvořena UI specifikace, která bude zahrnovat:

- 1) Motivaci
- 2) Definici cíle
- 3) Personifikaci
- 4) Vzorové případy užití (UseCase)
- 5) Scénáře
- 6) Logický design

Na základě definovaných případů užití budou vytvořeny testovací scénáře pro vyhodnocení kvalitativního šetření práce. Kvalitativní šetření bude provedeno za pomoci min. 8 participantů, kteří budou charakteristicky odpovídat definovaným personám. Na základě výsledků kvalitativního šetření budou formulovány závěry práce a doporučení k realizaci.

## Doporučený rozsah práce

63

## Klíčová slova

UI specifikace, Řízení kvality software, Personifikace, UseCase, Testovací scénář, Testování software, Systém

---

## Doporučené zdroje informací

- BLACK, Rex. Managing the testing process. Redmond, WA: Microsoft Press, 1999. ISBN 073560584X.
- BUREŠ, Miroslav, Miroslav RENDA, Michal DOLEŽEL, Peter SVOBODA, Zdeněk GRÖSSL, Martin KOMÁREK, Ondřej MACEK a Radoslav MLYNÁŘ. Efektivní testování softwaru: klíčové otázky pro efektivitu testovacího procesu. Praha: Grada, 2016. Profesionál. ISBN 978-80-247-5594-6.
- LEWIS, William E. Software testing and continuous quality improvement. Boca Raton: Auerbach, c2000. ISBN 08-493-9833-9.
- NIELSEN, Jakob a Hoa. LORANGER. Prioritizing Web usability. Berkeley, Calif.: New Riders, c2006. ISBN 978-032-1350-312.
- PATTON, Ron. Testování softwaru. Praha: Computer Press, 2002. Programování. ISBN 80-722-6636-5.
- STEPHENS, Matt a Doug ROSENBERG. Testování softwaru řízené návrhem. Brno: Computer Press, 2011. ISBN 978-80-251-3607-2.
- WATKINS, John. Testing IT: an off-the-shelf software testing process. New York: Cambridge University Press, 2001.

---

## Předběžný termín obhajoby

2018/19 LS – PEF

## Vedoucí práce

Ing. Josef Pavlíček, Ph.D.

## Garantující pracoviště

Katedra informačního inženýrství

---

Elektronicky schváleno dne 24. 1. 2019

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

---

Elektronicky schváleno dne 24. 1. 2019

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 17. 03. 2019

## **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci "Návrh a otestování řídicího systému pro zlepšování kvality software" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 25.3.2019

---

## **Poděkování**

Rád bych touto cestou poděkoval Ing. Josefu Pavlíčkovi, Ph.D. za vedení mé diplomové práce, katedře informačního inženýrství a Ing. Petře Pavlíčkové, Ph.D. za velmi přínosné konzultace.

Rovněž děkuji všem zúčastněným respondentům, bez kterých by tato práce nemohla vzniknout a naplnit své cíle.

# Návrh a otestování řídicího systému pro zlepšování kvality software

## Abstrakt

Tato diplomová práce se zabývá studií použitelnosti, ve které navrhuji systém pro řízení kvalit software. Vyvíjený prototyp systému je vytvořen na základě uživatelských poznatků, nabytých znalostí z konzultací s profesionály a odborné literatury.

Navrhovaný prototyp je určen pro úzkou skupinu uživatelů, kteří pracují v oboru dnes nejčastěji známém jako testování software. Navrhovaný systém by měl zvládat každodenní rutinní činnosti pracovníků a zároveň musí být použitelný jak v korporátním, tak i v prostředí malých podniků.

V této práci se čtenář dočte, jakým způsobem je prototyp navržen, jak je otestován pomocí kvalitativního šetření, jak jsou nashromážděná data analyzována a interpretovány konečné výsledky.

Použité metody a poznatky ze studií jsou zaznamenány v kapitole teoretických východisek a slouží jako podklad pro vypracování praktické části práce.

Konkrétní zpracování práce je zaznamenáno v kapitole praktických výstupů, kde kromě konkrétního zpracování prototypu čtenář nalezne i výsledky kvalitativního šetření včetně analytického rozpracování.

**Klíčová slova:** UI specifikace, Řízení kvality software, Personifikace, UseCase, Testovací scénář, Testování software, Systém

# Design and testing of a software quality management system

## Abstract

This thesis deals with the usability study in which I propose a software quality management system. The developed prototype of the system is based on user knowledge, acquired knowledge from consultations with professionals and literature.

The proposed prototype is designed for a narrow group of users working in the field most commonly known as software testing. The proposed system should cope with the day-to-day routine of workers and at the same time be applicable in both corporate and small business environments.

In this diploma, the reader will learn how the prototype is designed, tested it using a qualitative survey, analyzed the collected data, and interpreted the final results.

Used methods and findings from the studies are recorded in the chapter of the theoretical background and serve as a basis for the elaboration of the practical part of the thesis.

Specific work is recorded in the chapter of practical outputs, where, in addition to the specific prototype processing, the reader will also find the results of a qualitative survey including analytical elaboration.

**Keywords:** UI specification, Software quality assurance, Personification, UseCase, Testing scenario, Software testing, System

# Obsah

<b>1 Úvod.....</b>	<b>13</b>
<b>2 Cíl práce a metodika .....</b>	<b>14</b>
2.1 Cíl práce .....	14
2.2 Metodika .....	14
<b>3 Teoretická východiska .....</b>	<b>15</b>
3.1 Teoretická východiska ke zpracování UI Specifikace .....	15
3.1.1 Interakce člověk a počítač.....	15
3.1.2 User Experience Design.....	16
3.1.3 Interakce člověk a počítač vs User Experience Design .....	17
3.1.4 Uživatel v oboru informatika.....	17
3.1.5 Uživatelské rozhraní .....	18
3.1.6 UI Specifikace.....	20
3.1.7 Cílová skupina .....	20
3.1.8 Personifikace.....	22
3.1.9 Use Case – Příklad užití.....	22
3.1.10 Scénář.....	24
3.1.11 User Story .....	25
3.1.12 Logický design – Wireframe (drátěný model).....	25
3.2 Teoretická východiska z oboru testování software .....	27
3.2.1 Proces nazývaný V Model .....	27
3.2.2 Analýza požadavků .....	28
3.2.3 Funkční požadavky .....	29
3.2.4 Mimofunkční požadavky .....	30
3.2.5 Základní role v běžném procesu testování.....	32
3.2.6 Struktura přípravy testů .....	34
3.2.7 Rozdělení efektů snižujících kvalitu softwaru.....	36
3.2.8 Závažnosti efektů snižujících kvalitu software.....	38
3.3 Teoretická východiska ze studií použitelnosti .....	39
3.3.1 Testovací scénáře pro participanty .....	39
3.3.2 Interview .....	41
3.3.3 Počty participantů pro studie použitelnosti.....	42
3.3.4 Průběh testování s participantem .....	44
3.3.5 Charakteristiky pro vyhodnocení kvalitativního šetření.....	45
3.4 Vybrané charakteristiky kvalitativního šetření .....	46



3.4.1	Měřitelnost stanovených charakteristik .....	46
3.4.2	Hodnocení charakteristik .....	46
3.4.3	Interpretace výsledků pomocí kvantifikace .....	48
3.5	Celkový náhled na proces studie použitelnosti v praxi .....	51
<b>4</b>	<b>Praktické výstupy .....</b>	<b>52</b>
4.1	Motivace.....	52
4.2	Definice cíle UI Specifikace .....	52
4.3	Personifikace – praktický výstup .....	52
4.3.1	Persona 1 – Tester (Martin Zkoumavý) .....	53
4.3.2	Persona 2 – Test Analytik (Lucie Pečlivá) .....	54
4.3.3	Persona 3 – Test Manažer (Jan Generál) .....	55
4.4	Sběr informací a podkladů pro vypracování UI specifikace .....	56
4.4.1	Řízený rozhovor s profesionály z praxe .....	56
4.5	User Story.....	57
4.5.1	Přehled funkčních rolí pro User Story .....	57
4.5.2	Seznam hlavních funkčních prvků:.....	58
4.5.3	User Story z pohledu rolí přístupu:.....	59
4.6	Hlavní vybrané usecase.....	60
4.6.1	UC01 – Vytvořit Nástěnku .....	60
4.6.2	UC02 – Vytvořit Analýzu Požadavků .....	61
4.6.3	UC03 – Naplánovat testovací scénáře .....	62
4.6.4	UC04 – Vyhledat analýzu požadavků .....	63
4.6.5	UC05 – Vytvořit testovací analýzu.....	64
4.6.6	UC06 – Vytvořit první testovací scénář s testovacími kroky .....	65
4.6.7	UC07 – Exekuce testovacího scénáře .....	66
4.6.8	UC08 – Zadat chybu .....	67
4.6.9	UC09 – Vyhledat chybu .....	68
4.7	Hlavní scénáře k vybraným usecase .....	69
4.7.1	SC01 – Vytvořit nástěnku.....	69
4.7.2	SC02 – Vytvořit Analýzu požadavků .....	69
4.7.3	SC03 – Naplánovat testovací scénáře .....	70
4.7.4	SC 04 – Vyhledat analýzu požadavků .....	70
4.7.5	SC05 – Vytvořit testovací analýzu .....	70
4.7.6	SC06 – Vytvořit první testovací scénář s testovacími kroky.....	71
4.7.7	SC07 – Exekuce testovacího scénáře.....	72
4.7.8	SC08 – Zadat chybu.....	72
4.7.9	SC09 – Vyhledat chybu .....	73

4.8	Kvalitativní šetření .....	74
4.8.1	Výběr testovacích cílů .....	74
4.9	Finanční analýza – Návrh systému v praxi .....	75
<b>5</b>	<b>Výsledky a diskuse .....</b>	<b>77</b>
5.1	Očekávané výsledky.....	77
5.2	Rekapitulace hodnocení kvalitativního šetření .....	78
5.3	Celkové výsledky kvalitativního šetření .....	79
5.4	Analytické rozpracování dílčích výsledků .....	80
5.4.1	Analytický rozpad charakteristiky Intuitivnost.....	81
5.4.2	Analytický rozpad charakteristiky Úspěšnost.....	81
5.4.3	Analytický rozpad charakteristiky Čas .....	82
5.4.4	Analytický rozpad charakteristiky Překlikatelnost .....	82
5.5	Poznátky získané od participantů.....	83
<b>6</b>	<b>Závěr.....</b>	<b>85</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>88</b>
<b>8</b>	<b>Seznam použitých zdrojů obrázků:.....</b>	<b>91</b>
<b>9</b>	<b>Přílohy .....</b>	<b>92</b>

## Seznam obrázků

Obrázek 1 - User experience design [1].....	16
Obrázek 2 - Příkazový řádek v systému MacOS.....	18
Obrázek 3 - Grafické rozhraní systému MacOS.....	19
Obrázek 4 - Logický design [2] .....	26
Obrázek 5 - V model [2] .....	28
Obrázek 6 - Křivka získaných informací vs celkové počty respondentů [3].....	43
Obrázek 7 - Zastupitelnost počtu získaných informací vs celkové počty respondentů [4] .	43
Obrázek 8 - Ukázkový interval nekonkrétního cíle testu [5].....	49
Obrázek 9 - Ukázková vizualizace na pavučinovém grafu.....	50
Obrázek 10 - Proces studie použitelnosti v praxi [6].....	51
Obrázek 11 - UC01 – Vytvořit Nástěnku .....	60
Obrázek 12 - UC02 – Vytvořit Analýzu Požadavků .....	61
Obrázek 13 - UC03 – Naplánovat testovací scénáře .....	62
Obrázek 14 - UC04 – Vyhledat analýzu požadavků.....	63
Obrázek 15 - UC05 – Vytvořit testovací analýzu.....	64
Obrázek 16 - UC06 – Vytvořit první testovací scénář s testovacími kroky .....	65
Obrázek 17 - UC07 – Exekuce testovacího scénáře .....	66
Obrázek 18 - UC08 – Zadat chybu .....	67
Obrázek 19 - UC09 – Vyhledat chybu.....	68
Obrázek 20 - Ganttův diagram studie použitelnosti v praxi .....	76
Obrázek 21 - Interval celkového ohodnocení systému .....	79
Obrázek 22 - Pavučinový graf celkového zhodnocení systému .....	80
Obrázek 23 - Interval analytického rozpadu charakteristiky Intuitivnost.....	81
Obrázek 24 - Interval analytického rozpadu charakteristiky Úspěšnost.....	81
Obrázek 25 - Interval analytického rozpadu charakteristiky Čas .....	82
Obrázek 26 - Interval analytického rozpadu charakteristiky Překlikatelnost .....	82

## **Seznam tabulek**

Tabulka 1 - Kaskádová struktura rozložení testů.....	35
Tabulka 2 - Ukázková součtová tabulka .....	49
Tabulka 3 - Rozbor mzdových sazeb.....	75
Tabulka 4 - Tabulka zdrojů pro finanční analýzu.....	75
Tabulka 5 - Plánování jednotlivých iterací studie použitelnosti v praxi.....	76
Tabulka 6 - Sumarizace celkových bodů kvalitativního šetření .....	79

# 1 Úvod

V oblasti testování software jsou každodenním cyklem vývoje testovacích scénářů, exekuce neboli spouštění testovacích scénářů a jejich vyhodnocení, plánování testů a prezentace sjednocených výsledků za určité období. K těmto úkonům jsou povoláni lidé s na pozicích test analytik, tester a test manažer.

Všichni mají společné především to, že spolu komunikují a každá z těchto pozic potřebuje pro svou práci výstupy, které generují jiné dvě pozice. Způsob předávání informací nemusí být vždy jednoduchým úkolem a hned v tomto kroku vzniká riziko nekonzistence dat. Kromě tohoto hrozí i ztráty dat a další hrozby.

Tyto hrozby byly vyřešeny centralizovanými nástroji, které svým vývojem buď ztratily původní vlastnosti, které byly užitečné pro testery nebo přidaly další funkce, které snižují přehlednost systému. V neposlední řadě tyto nástroje navýšily jejich prodejní cenu, která znemožňuje menším společnostem tyto nástroje používat.

V této práci je především zaměřena na vývoj nástroje pro řízení testů a testovacích nástrojů. Jsou vybrány nejdůležitější prvky na základě analýzy s experty z praxe a bude snahou je navrhnout jednoduché, intuitivní, a především lehké ovladatelné.

Dalším dílčím cílem systému je, že po konečné realizaci by neměl nabývat velké nominální hodnoty a měl by být dostupný i menším společnostem.

Teoretická východiska jsou zaměřena na definování základních kroků pro tvorbu UI specifikace. Dále jsou definovány základní informace potřebné k testování software. Informace o tom, jak by měl vypadat řízený rozhovor s participanty a zaměření na výběr charakteristik pro měření použitelnosti navrhované aplikace.

Praktická část je zaměřena na vývoj UI specifikace, která je podložena nastudovanými či analyzovanými údaji z teoretické části práce. Na základě usecase se stanovují testovací scénáře, resp. cíle pro participanty, kteří pomocí definovaných charakteristik hodnotí použitelnost navrženého systému.

V Závěru lze nalézt výsledky kvalitativního šetření, doporučení pro realizaci a vylepšení práce. Kromě poznatků získaných během práce v závěru se lze dočíst i zhodnocení této práce s doporučením k realizaci.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Hlavním cílem práce je navrhnout UI specifikaci systému pro řízení kvality software. Součástí práce bude tvora uživatelských testovacích scénářů. Ty budou mít za úkol odhalit nedostatky systému. Po ukončení kvalitativního šetření UI specifikace budou výsledky vyhodnoceny. Na základě analýzy výsledků budou definovány závěry a doporučení k realizaci.

### **2.2 Metodika**

Metodika práce je založena na analýze odborné literatury – rešerše, analýze existujících softwarových řešeních a konzultacích s profesionály z oboru řízení kvalit software. Na základě nabytých znalostí a poznatků bude vytvořena UI specifikace, která bude zahrnovat:

- 1) Motivaci
- 2) Definici cíle
- 3) Personifikaci
- 4) Vzorové případy užití (UseCase)
- 5) Scénáře
- 6) Logický design

Na základě definovaných případů užití budou vytvořeny testovací scénáře pro vyhodnocení kvalitativního šetření práce. Kvalitativní šetření bude provedeno za pomoci min. 8 participantů, kteří budou charakteristicky odpovídat definovaným personám. Na základě výsledků kvalitativního šetření budou formulovány závěry práce a doporučení k realizaci.

## **3 Teoretická východiska**

### **3.1 Teoretická východiska ke zpracování UI Specifikace**

V kapitole teoretických východisek se čtenář seznámí s termíny, které jsou důležité k pochopení práce. Termíny jsou úzce spojeny s vědním oborem interakce člověk a počítač a zároveň jsou hlavními pilíři k vykonání vlastní práce.

#### **3.1.1 Interakce člověk a počítač**

Interakce člověk a počítač je vědní obor, který se zabývá vývojem technologií, které jsou specializované na užívání počítačů lidmi neboli uživateli. Převážně se jedná o návrhy a design počítačů, či jen software, který je spouštěn v počítači. [1]

Tento obor vznikl kolem 80. let 20. století, kdy se na trhu objevily první osobní počítače. V tuto dobu se počítače začali dostávat převážně do kanceláří, ale i do domácností. V této době byly počítače používány spíše jako účetní pomůcky, textové editory či herní nástroje, což bylo zapříčiněno tím, že počítače nebyly tak rozměrné a nezabírali celou místnost jen pro jejich uskladnění. [1]

Na základě akademických šetření se právě tento vědní obor stal fenoménem při vývoji softwaru, ale i počítačů samotných. Začala se popularizovat myšlenka, že práce uživatele s počítačem je jakýsi dialog. Aby takovýto dialog mohl být umožněn je nejprve potřebné, aby uživatel počítači porozuměl. Proto se nejprve začaly upravovat pracovní metody s počítači, ale jak přibývaly technologie, jako jsou například internet, smartphoney a jiné, tak se začaly zavádět metody přístupu uživatele k počítači či jen samotnému systému. [2]

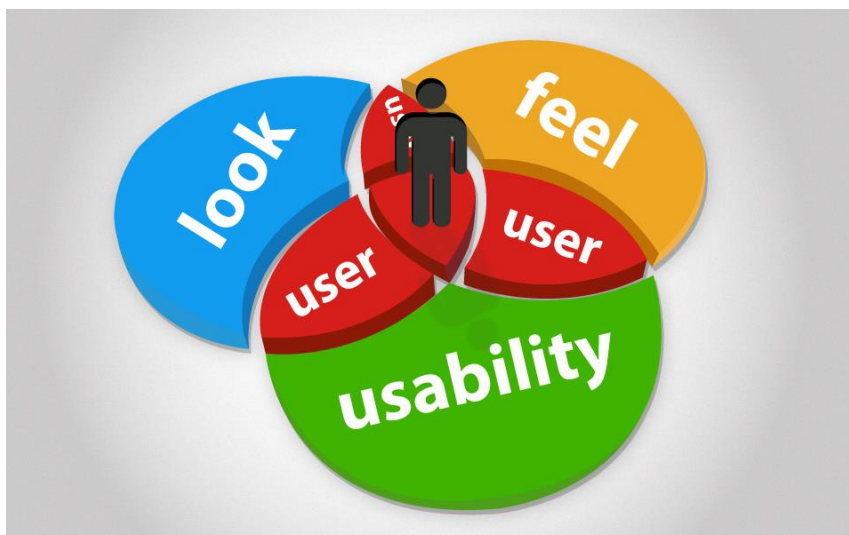
### 3.1.2 User Experience Design

UX Design je proces tvoření design aplikací a systémů na základě uživatelských zkušeností.

Tento proces je často zaměňován s pojmy použitelnost, návrh uživatelských rozhraní a podobně. Tento fakt není zas až tak daleko od skutečnosti, ale pravdou je, že tyto pojmy jsou brány jak podmnožinou samotného UX návrhářství.

V kostce se návrhář UX design se zaměřuje na celou škálu aspektů, které vychází právě z uživatelských zkušeností. Přednostně návrhář analyzuje procesy, které má systém vykonávat, následně tvoří přívětivé prostředí pro specifikovanou uživatelskou skupinu.

Návrhář se snaží pracovní procesy navrhnout takovým způsobem, aby navrhovaný systém odpovídal požadavkům uživatele a zároveň byl přehledný a lehce dostupný uživatelům, kteří nemají s podobnými systémy zkušenosti. Výsledný systém by měl přinést efektivitu práce a radost z jeho používání, což je známkou dobře navrhnutého systému. [4]



Obrázek 1 - User experience design [1]



### 3.1.3 Interakce člověk a počítač vs User Experience Design

Na rozdíl od samotného UX designu se Interakce člověk a počítač zaměřuje spíše na empirické chápání uživatele. Interakce člověk a počítač je bráno spíše z akademického hlediska. UX Design je chápán spíše jako proces návrhu systémů, který má být přístupný uživatelům cílových skupin. Jsou to spíše samotné pracovní postupy návrhů, jak přiblížit systém či aplikaci koncovým uživatelům zpřehlednit, a především zjednodušit jejich práci s navrhovaným systémem. [5]

### 3.1.4 Uživatel v oboru informatika

Uživatel je fyzická osoba, která přistupuje k počítači a jeho software, se kterým vykonává duševní práci. Tento člověk přistupuje k systému pomocí uživatelského rozhraní a v dnešní době nejčastěji pomocí tak zvaného grafického uživatelského rozhraní. Tuto práci vykonává za získání nějakého užitku, například můžeme uvést zisk dat ze systému či nabytí požadované informace. [6]

V informatice se rovněž za uživatele považuje člověk, který bez komplexních technologických znalostí počítače je schopen pracovat se systémem, a to za pouze omezených znalostí vztahujících se k danému systému.

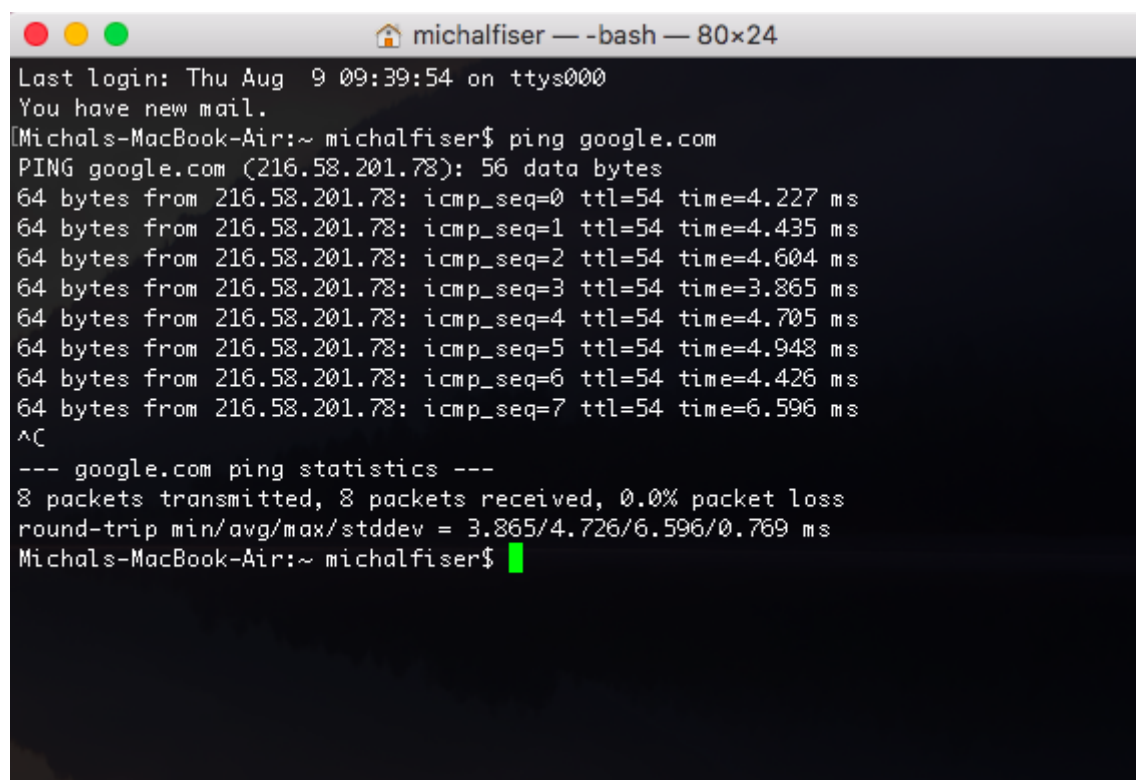
V neposlední řadě je dobré zmínit, že uživatel je osoba, pro kterou je systém navrhován a vyvíjen. [6]

Uživatelé pak můžeme rozdělit z hlediska používání na dvě základní skupiny:

1. Koncový uživatel – takový uživatel, který používá systém pouze omezeně za účelem splnění své práce.
2. Administrátor – uživatel, který drží správu systému v pořádku a snaží se být oporou koncovým uživatelům. Takový uživatel je velmi často nazýván slangy jako jsou „podpora“ nebo prostěji „ajták“. [7]

### 3.1.5 Uživatelské rozhraní

Ve vědním oboru informatika se uživatelské rozhraní chápe jako nástroj, který umožní uživateli pracovat nebo jinak řečeno vytvářet interakci mezi uživatelem a počítačem. Základním uživatelským rozhraní je tak zvaná příkazová řádka. Příkazová řádka se v dnešní době u běžných uživatelů používá velmi zřídka. Vypadá jako běžný text a funguje na principu příkazů. Uživatel zadá příkaz a systém na uvedený příkaz odešle odpověď.

A screenshot of a macOS terminal window. The title bar shows the user 'michalfiser' and the shell '-bash' with a resolution of '80x24'. The terminal output shows a successful ping to google.com. The user has entered 'ping google.com' and the system has responded with 8 successful pings, each 64 bytes in size, with varying round-trip times between approximately 3.865 ms and 6.596 ms. The terminal also shows a notification for new mail and a summary of the ping statistics.

```
Last login: Thu Aug 9 09:39:54 on ttys000
You have new mail.
[Michals-MacBook-Air:~ michalfiser$ ping google.com
PING google.com (216.58.201.78): 56 data bytes
64 bytes from 216.58.201.78: icmp_seq=0 ttl=54 time=4.227 ms
64 bytes from 216.58.201.78: icmp_seq=1 ttl=54 time=4.435 ms
64 bytes from 216.58.201.78: icmp_seq=2 ttl=54 time=4.604 ms
64 bytes from 216.58.201.78: icmp_seq=3 ttl=54 time=3.865 ms
64 bytes from 216.58.201.78: icmp_seq=4 ttl=54 time=4.705 ms
64 bytes from 216.58.201.78: icmp_seq=5 ttl=54 time=4.948 ms
64 bytes from 216.58.201.78: icmp_seq=6 ttl=54 time=4.426 ms
64 bytes from 216.58.201.78: icmp_seq=7 ttl=54 time=6.596 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.865/4.726/6.596/0.769 ms
Michals-MacBook-Air:~ michalfiser$
```

Obrázek 2 - Příkazový řádek v systému MacOS

Tato metoda komunikace mezi uživatelem a počítačem je už dnes využívána spíše ve vývojových strukturách společností.

Pro běžné uživatele bylo vyvinuto tak zvané grafické rozhraní. Toto uživatelské rozhraní bylo navrženo pro běžné uživatele, kteří nejsou natolik technicky zdatní, aby ovládali celý systém příkazy. Grafické uživatelské rozhraní umožňuje uživateli komunikovat s počítačem skrze předinstalované funkce, které se spustí pomocí jednoduchých interakcí jako je kliknutí myši, či stisknutí klávesových zkratk.

### Grafické rozhraní systému MacOS – menu aplikací



Obrázek 3 - Grafické rozhraní systému MacOS

Dalšími užitečnými vlastnostmi grafického rozhraní jsou například zobrazování fotografií, přehrávání multimediálních souborů, tvoření prezentačních materiálů nebo hraní her, které se v posledních 10 letech velmi rozšířilo. [8]

### **3.1.6 UI Specifikace**

UI Specifikace je technický dokument, na jehož základě se stanovují a specifikují požadavky pro vývojáře aplikací. Tento dokument nejčastěji píše a tvoří software analytik se software designerem. Vznik tohoto technického materiálu je projednávám s tak zvanou business organizační strukturou podniku. Materiál by měl obsahovat všechny logické prvky, use case a scénáře aktivit, které lze s vyvíjeným systémem provádět. Rovněž dokument musí zahrnovat detailní popis budoucích potenciálních uživatelů. Procesu navrhování a analýze potenciálních uživatelů se říká personifikace, o které se čtenář dozví více informací později.

Dle této specifikace pak vzniká „tvář“ aplikace, kterou uživatel využívá. Jinými slovy se této uživatelské vrstvě říká front-end. UI Specifikace nijak nespecifikuje, jak by měl vypadat back-end, tedy služby nebo vrstvy služeb, které na základě front-end interakce člověka přenášejí data či je transformují pro požadovaný výstup.

Kromě této stránky UI Specifikace, je tento dokument i standartní technickou pomůckou v oblasti vědy a výzkumu. Na základě tohoto materiálu je možné určit, zda navrhovaná aplikace bude použitelná pro budoucí uživatele a které prvky je potřebné zlepšit či opravit. Jedná se o proces testování aplikace, o kterém bude zmíněno v dalších kapitolách. [9]

### **3.1.7 Cílová skupina**

Cílová skupina nejčastěji představuje charakteristiky a vlastnosti potenciálních zákazníků, které se snaží prodejce oslovit na trhu. Cílová skupina se nejčastěji definuje v rámci marketingových kampaní, a to na základě socio demografických parametrů. Námi stanovené parametry nám utváří množinu hranic, kterých se při budování marketingové kampaně musíme držet. Avšak nejde jen o marketingovou kampaň ale i o službu či výrobek samotný. [10]

**Geografické zaměření:**

Zaměřujeme se na zákazníky, kteří se nachází v geografické blízkosti našeho trhu nebo přímo v něm. Máme v úmyslu oslovit zákazníky například z Prahy, a proto nešíříme reklamu v Paříži ale v Praze.

**Věkové rozmezí:**

Spousty služeb a statků, lze limitovat podle věku. Například nebudeme dědečkovi v 90 letech nabízet kolečkové brusle. Tento faktor bývá v praxi jedním z nejčastěji užívaným.

**Pohlaví:**

Rozdíly mezi pohlavími jsou příběhem již od stvoření světa, a tak když cílíme na skupinu extravagantních žen se šperkovou kolekcí, tak rozhodně nemůžeme zapomínat i oslovit muže, kteří jsou ochotní nabízenou šperkovou kolekci své dámě zakoupit.

**Vzdělání a společenské postavení:**

Tato rozdělení jsou velmi často spojována s cenou. Ideologicky je jasné, že člověk s minimální mzdou si nepůjde koupit nové auto každý rok. A zase manažersky postavený úspěšný člověk si nepůjde koupit salám v akci. V tomto případě je velmi důležité zjistit, co za společensky postavené lidi v daném trhu zajímá a jestli se vůbec vyplatí produkt či službu nabízet.

**Životní styl:**

Každý žije svůj život jinak a je potřebné se zaměřit na hlavní charakteristiky moderních trendů životních stylů. Kritérium životního stylu je nejvíce výrazné u produktů a služeb, které se zaměřují na koníčky a zábavu. Někdo žije sportem nabitý život a někdo zase raději sedí doma a hraje hry. [11]

### 3.1.8 Personifikace

Personifikace je metoda vytvoření fiktivní osoby neboli osoby, která svým archetypem a vlastnostmi zapadá do definovaných hranic stanovených v cílové skupině. Persona umožňuje návrhářům lépe porozumět požadavkům systému a udává jim určitý směr, kterým by se měli při návrhu systému ubírat. Běžná persona by měla být profilem osoby, a tudíž musí zahrnovat následující atributy:

- 1) Jméno a příjmení – ideálně takové, které ji vystihuje
- 2) Běžný den
- 3) Historie
- 4) Vzdělání
- 5) Koníčky

A další volitelné atributy, které by mohly designera nasměrovat k vytvoření definovaných cílů.[12]

### 3.1.9 Use Case – Příklad užití

Use Case neboli případ užití v oboru interakce člověk a počítač je metodou popisu procesů, které by měl systém v návaznosti na lidskou interakci vykonávat. Jednotlivé procesy jsou zapisovány sekvenčně po jednotlivých krocích. Jedná se o případu užití například běžně používaných bankomatů, nebo otevření souborů či jiné práce se software nástroji.

Takovýto případ užití lze zakreslit i pomocí tzv. UML (Unified Modeling Language) diagramu, který popisuje uživatelské akce v návaznosti na reakce systému, se kterým uživatel pracuje. Tato forma zápisu je dnes velmi rozšířená a v praxi se s ní setkáváme velmi často při modelování systémů. Nicméně, ke každému diagramu je dobré mít i slovně psanou formu nebo menší dokumentaci, která může vypadat následovně.

Pro příklad si uveďme prosté přihlášení do e-mailu na webovém portále. Avšak berme tento příklad s rezervou, protože již samotný proces autorizace by se mohl napsat v samostatném případu užití.

Předpoklad: Předpokládáme, že uživatel má již zobrazený webový portál v prohlížeči.

1. Krok) Systém zobrazuje textová pole „Username“, „Password“ a tlačítko „Login“.
2. Krok) Uživatel zanesse své uživatelské jméno do textového pole „Username“
3. Krok) Uživatel zanesse své uživatelské heslo do textového pole „Password“
4. Krok) Uživatel pomocí kurzoru stiskne tlačítko „Login“
5. Krok) Systém zobrazí kontrolní hlášku „Čekejte prosím, ověřuji Vaše osobní údaje“
6. Krok) Uživatel čeká
7. Krok) Systém zobrazí domovskou stránku a seznam e-mailů

Takovéto případy užití velmi často vychází z bussines zadání, které sestavují návrháři aplikace se zákazníky. Jsou tedy často velmi obecná a nepopisují programovací logiku systému. Vytvořené případy užití mají mít spíše informativní charakter o prvcích vyskytujících se v systému a způsobu zacházení s nimi. [13]

Na základě připravených UseCase vzniká drátový nebo také logický design, který je samotným prototypem systému. O logickém designu se více dozvíme v dalších kapitolách. Jednotlivé UseCase jsou po navržení logického designu testovány na základě uživatelských testů, které mají za úkol odhalit nedostatky systému z pohledu uživatele. [14]

### 3.1.10 Scénář

Scénář v problematice uživatelských případů udává reakci systému na uživatelské vstupy. Je tedy systémovou reakcí toho, co uživatel zadává na vstupu. V uživatelských případech je definováno, co uživatel očekává od systému, ale naopak ve scénářích systém čeká na uživatelův vstup.

Stejně jako v případech užití autor musí předpokládat s počátečním předpokladem pro vznik takového scénáře. Každý scénář by měl mít svůj případ užití, neboť je jeho systémovým obrazem. Pro úplnost přehledu navážeme na příklad z podkapitoly Use Case – Případ užití.

Předpoklad: Předpokládáme, že uživatel má již zobrazený webový portál v prohlížeči.

1. Krok) Systém zobrazuje textová pole „Username“, „Password“ a tlačítko „Login“
2. Krok) Systém čeká až uživatel zanesse své uživatelské jméno do textového pole „Username“
3. Krok) Systém validuje uživatelské jméno
4. Krok) Systém čeká až uživatel zanesse své uživatelské heslo do textového pole „Password“
5. Krok) Systém validuje přihlašovací heslo
6. Krok) Systém čeká až uživatel pomocí kurzoru stiskne tlačítko „Login“
7. Krok) Systém zobrazí kontrolní hlášku „Čekejte prosím, ověřuji Vaše osobní údaje“
8. Krok) Systém zobrazí domovskou stránku [14]



### 3.1.11 User Story

User Story, jak již sám název naznačuje, je uživatelský příběh, který definuje, jaké informace uživatel může v modelovaném systému zobrazit a ke kterým úkolům se dostane. Nejčastějším příkladem uživatelských příběhů jsou tzv. Uživatelská oprávnění. Definují, jaké akce může uživatel nebo spíše uživatelské role vykonávat v interakci se systémem.

[15] [16]

Jednoduchým příkladem může být:

User Story – Běžný uživatel

- 1) uživatel se může přihlásit
- 2) uživatel může napsat komentář do příspěvku
- 3) uživatel komentář nemůže smazat
- 4) uživatel se může odhlásit

User Story – Administrátor

- 1) uživatel se může přihlásit
- 2) uživatel může napsat komentář do příspěvku
- 3) uživatel komentář může smazat
- 4) uživatel se může odhlásit

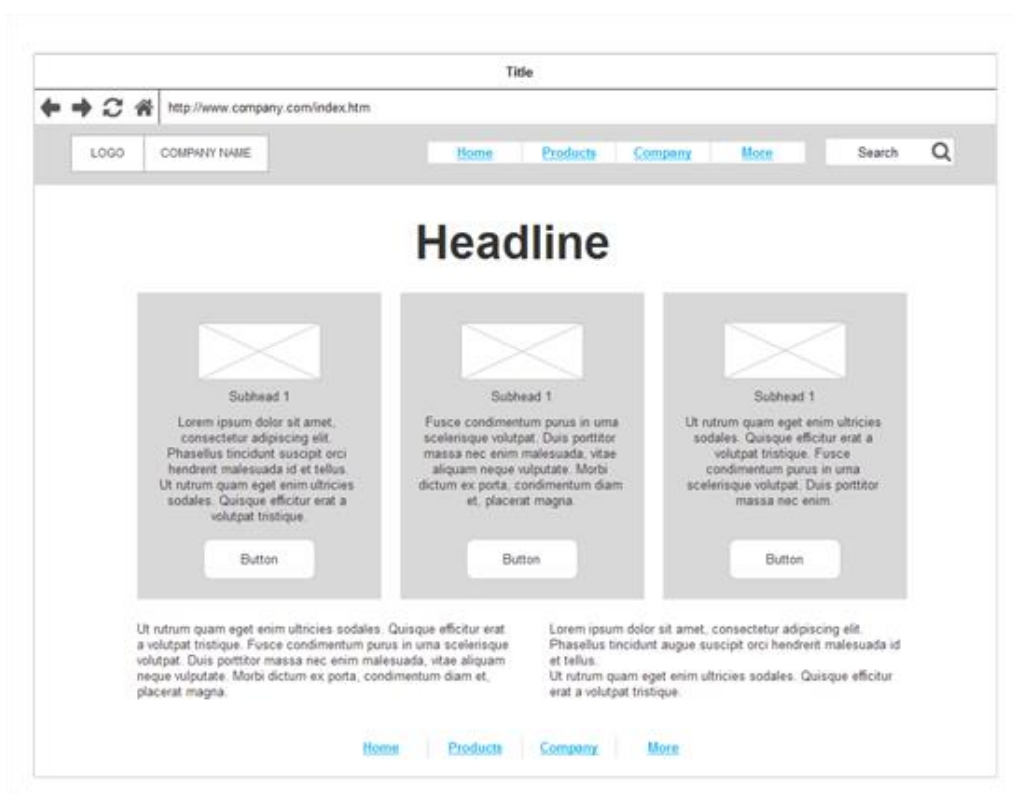
### 3.1.12 Logický design – Wireframe (drátěný model)

Logický Design, wireframe nebo také drátěný model představuje návrh aplikace na té nejzákladnější úrovni. Tato metoda navrhování aplikací slouží k navržení tzv. kostry systému či aplikace. Drátěné modely se vytváří v samotných začátcích vývoje a slouží jako jedna z prvních předloh či lépe řečeno dokumentací pro samotné developery. Běžným způsobem použití této metody je vytýčení hlavních či základních funkcí aplikace, které

by měly zohledňovat konkrétní uživatelské požadavky, jež byly stanoveny pro začátek vývoje systému.

Jednou z velkých předností drátěných modelů je především jeho jednoduché pochopení. Oproti diagramům, které například vychází z UML je i pro laického pozorovatele jednoduché se v drátovém modelu zorientovat. Jestliže zadavatel práce se nevyzná v prostředí UML, tak je mnohem jednodušší se domluvit na vývoji prostřednictvím drátěného modelu. Ve zkratce řečeno je drátěný model mnohem intuitivnější pro vyjednání zadání. Další výhodou je, že již navržený model, který je akceptovaný všemi stranami využít v některých nástrojích jako šablonu kódu. Principiálně vývojář nechá vygenerovat kód na základě návrhu a už ho může jen upravit či integrovat na již existující systémy.

Drátěný model rozhodně neobsahuje složité grafické prvky, jde pouze o prosté zobrazení funkčních požadavků systému bez grafické podpory. Vizualizace systému a grafické zpracování systému je požadováno v tzv. grafickém modelu, který nebude součástí této diplomové práce, a proto se k němu již nebudeme vracet. [17]



Obrázek 4 - Logický design [2]

## 3.2 Teoretická východiska z oboru testování software

Co je to řízení kvality software nebo také moderněji řečeno testování software? Testování software je proces, který začíná o něco později, než samotný vývoj softwaru následně pak běží paralelně s ním a samotné konec zakončují společně při odevzdání finálního produktu.

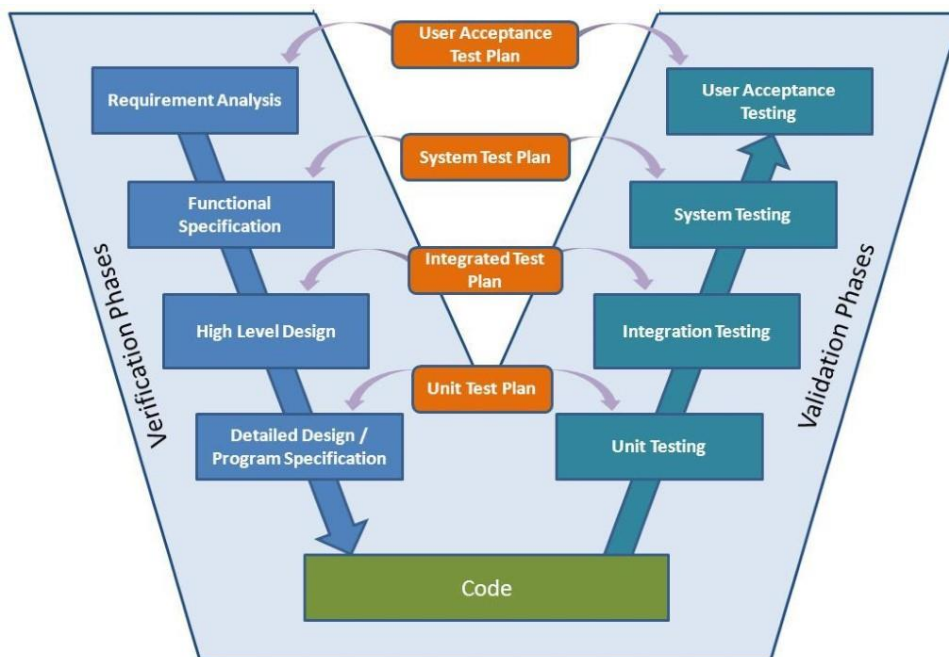
Testování je takový proces, při kterém testovací tým získává podklady pro testování a následně vykonává testy pro samotný software. Tým odhaluje chyby a následně je konzultuje s vývojáři, kteří testovaný software vylepšují či opravují na základě podkladů od testovacího týmu.

### 3.2.1 Proces nazývaný V Model

V Model je proces testování softwaru, jehož hlavní myšlenkou je, aby jednotlivé části softwaru byly testovány dle implementačních částí. Například u vodopádového modelu se velmi častým problémem stávalo, že odhalování jednotlivých chyb v systému bylo buď opožděné nebo nedostačující, a proto byl navržen tzv. V Model, který má potenciální nežádoucí vlastnosti, či chyby odstranit.

Nyní jsme si sdělili hlavní důvod, proč byl V Model vynalezen. Nyní je potřeba si říci, proč se tento proces nazývá V modelem, a co je jeho hlavní myšlenkou?

Hlavní myšlenkou V Modelu je, že jednotlivé fáze vývoje by měly být patřičně otestovány dle doplňující dokumentace. Například ve verifikační části lze najít pojem „detailed design / program specification“, který je šipkou spojený s „unit tests“ přes „unit test plan“. To znamená, že existuje specifikace natolik podrobná, která definuje jednotlivé funkcionality jednotek systému, a proto je možné, a tudíž i potřebné je otestovat, na základě rozvrženého testovacího plánu a analýzy. [18]



Obrázek 5 - V model [2]

Requirement Analysis vs User Acceptance Testing

Functional Specification vs System Testing

High Level Design vs Integration Testing

Detailed Design vs Unit Testing

### 3.2.2 Analýza požadavků

Analýza požadavků je jednou z prvních částí budování nového systému. Aby bylo možné rozdělit práci mezi analytiku, architektu, vývojáře a také testery, tak je nutné mít přesně zpracovanou analýzu požadavků. Analýza požadavků vychází převážně z byznysových zadání, která nemusí být vždy reálně proveditelná. Zákazník si může malovat vzdušné zámky, které nebudou realizovatelné. V případě, že dodavatel podepíše kontrakt se zákazníkem bez analýzy požadavků, tak se může stát, že se dodavatel na vlastní náklady ocitne v pasti a bude nucen vykonat nevykonatelnou práci. Z tohoto důvodu musí být požadavky vždy: proveditelné, měřitelné, testovatelné. [19]

Tato analýza nejčastěji představuje dokument uváděný ve formě tabulek, kde se zaznamenávají konkrétní údaje o požadavcích:

- 1) ID požadavku
- 2) Název požadavku
- 3) Popis požadavku
- 4) Typ nebo kategorie
- 5) Priorita
- 6) Složitost
- 7) A další...

Dalším využitím analýzy požadavků je evidence již realizovaných a plánovaných požadavků. Téměř při každém projektu se stává, že zákazník oponuje při výsledném produktu a nabízí argumenty typu “toto mělo být jinak” a “ takhle jsme se nedohodli”. V těchto případech, kdyby nebyla detailně sepsána analýza požadavků, tak by zákazník mohl dodavatele využít přes čáru rozpočtu a dodavatel by to musel podstoupit.

V případě, že dodavatel již analýzu má detailně zpracovanou a zákazník přijde s podobným argumentem, tak je tento tzv. Změnový požadavek naopak výhodou pro dodavatele, protože i za změnové požadavky, může účtovat větší objem peněz, než bylo původně stanoveno. [16]

### **3.2.3 Funkční požadavky**

Funkční požadavky jsou právě ty požadavky, které definují konkrétní funkcionality systému. Nezaobírají se designem systému, rychlostí či jeho přístupností ale definují, co přesně má systém vykonávat.

### **Definice funkčních požadavků:**

Požadavkem funkčnosti v oblasti softwaru a systémového inženýrství je prohlášení o zamýšlené funkci systému a jeho součástí. Na základě funkčních požadavků určuje inženýr chování (výstup), které má zařízení nebo software očekávat v případě určitého vstupu. Návrh systému je časnou formou funkčního požadavku. [20]

Typické příklady funkčních požadavků:

- 1) Obchodní pravidla
- 2) Opravy, úpravy a rušení transakcí
- 3) Administrativní funkce
- 4) Sledování auditu
- 5) Certifikační požadavky
- 6) Požadavky na reportování stavu dat
- 7) Udržení historických dat
- 8) Právní a regulační požadavky
- 9) A další... [21]

### **3.2.4 Mimofunkční požadavky**

Nefunkční nebo také mimofunkční požadavky se nepřímo váží k samotnému kódu aplikace. Jedná se o požadavky, které vystihují všechny ostatní aspekty aplikace, kromě právě těch funkčních, které zkráceně řečeno, jsou směřovány na funkčnosti aplikace. V nefunkčním testování se testeři zaměřují například na výkon, použitelnost a spolehlivost aplikace. Velmi často se k měření cílů nefunkčních požadavků používají specializované automatizované nástroje.

Jednou z nejčastějších charakteristik nefunkčního testování je počet uživatelů pracujících, nebo – li vykonávajících rutinní činnosti ve stejné chvíli po nějakou dobu. Dejme tomu, že uvádíme na trh novou aplikaci pro internetové bankovníctví, a tak chceme, aby současně bylo přihlášeno 3000 uživatelů a pracovali po dobu jedné hodiny.

Mimofunkční testování je také velmi důležité zejména marketingu. Ještě žádný zákazník neřekl “to je ale pěkný kód”. Každý zákazník se totiž dívá spíše na design a rychlost odpovědi zakupované aplikace, když tedy zjistí, že vybraná aplikace splňuje jeho požadavky na funkčnosti.

### **Konkrétní typy nefunkčních požadavků:**

- 1) Bezpečnost – požadavky jsou většinou cíleny na konkrétní zranitelnosti, které například definuje instituce pro bezpečnost IT OWASP
- 2) Spolehlivost – software je spolehlivý do té doby, dokud správně vykonává všechny své funkcionality
- 3) Schopnost přežít – tento parametr představuje schopnosti samo-uzdravení systému. V případě, že nebude dostupná služba/funkcionalita systému, je systém sám schopný toto selhání zaznamenat a napravit
- 4) Dostupnost – udává kolik paralelně uživatelů může paralelně vykonávat určitý druh práce, dokud systém nezačne selhávat
- 5) Použitelnost – jedná se o měření, kdy uživatel musí být schopen do nějaké časové jednotky splnit úkoly nebo odhalit funkcionality, které mu software nabízí. Když tento časový milník přesáhne, tak můžeme říct, že software není intuitivní a je potřeba ho upravit.
- 6) Škálovatelnost – je to spíše charakteristika, která ukazuje, jak moc můžeme rozšířit zpracovatelskou kapacitu, aby uspokojila nárůst poptávky. Příklad, je vydána nová online hra a poskytovatel počítá s 5 000 uživateli. Jenže reklama způsobí, že hru přijde vyzkoušet 10 000 uživatelů a on musí rozšířit kapacity.

7) Interoperabilita – je také spíše charakteristika. Udává, v jaké míře je možné integrovat rozhraní softwaru na jiný software. Příkladem může být integrace vykazovacího systému do účetního systému.

8) Účinnost – udává po jakou dobu je systém schopen odpovídat v rozumné míře v plné kapacitě přístupů.

9) Flexibilita – toto testování je spíše pro korporátní účely jako je Microsoft a Apple. Testuje se, zda systém je schopen pracovat na různých variacích hardware či software platformách.

10) Přenositelnost – software je schopný fungovat jak na stávajícím software a hardware řešení, tak i na odlišeném. [22]

### **3.2.5 Základní role v běžném procesu testování**

Testování software je proces, který vyžaduje na profesionální úrovni rozdělení práce mezi jednotlivé vrstvy pracovníků a následné řízení. V praxi vždy existují alespoň tři základní rozdělení, a to na test manažera, testového analytika a testera.

#### **Test manažer**

Práce test manažera nebo také někdy nazývaná jako test architekta je velmi úzce spojena s plánováním a řízením samotných aktivit. Tento manažer zodpovídá při vývoji za svůj testovací tým, kterému přiděluje práci a plánuje testování či analýzy dle časových možností. Jeho největší zodpovědností je tedy plánování, druhou největší zodpovědností je sbírat informace s hierarchicky výše postavenými lidmi a komunikovat je se svým týmem. Směr předávání informací není jednosměrný, a proto také musí zvládat přebírat informace od svého týmu, který mu dodává podklady o stavu testů či stavu analýz.

Třetí nejdůležitější zodpovědností test manažera, je výběr pracovníků do týmu dle určitých kritérií, které jsou definovány vyvíjeným systémem a kapacitou zdrojů testovaného projektu.



## **Test analytik**

Práce test analytika musí být vždy přesná a podrobná. Analytik musí zvládat vstřebávat mnoho informací a pokládat je do správných souvislostí, aby dokázal vytvořit takové testovací scénáře, které budou odpovídat zadání a zároveň pokryjí nejméně 90% požadavků na vyvíjený systém. Hlavní zodpovědností test analytika je tedy přebírání informací a dokumentů od svého manažera. Analytik se musí řídit plánem, který navrhnul jeho manažer a jestliže není v zadání něco jasné nebo případně dořešené, je jeho okamžitou odpovědností to jít prodiskutovat manažerem nebo s příslušným článkem, který za danou oblast odpovídá.

Další z jeho odpovědností je psaní, a tedy i příprava testů pro tzv. test exekuci. Test analytik musí dle analýzy napsat takové testy, aby kdokoliv je bude testovat byl schopen rozpoznat, a tudíž i otestovat danou problematiku. Často se také uvádí, že by analytik měl také připravovat data pro testery, což je velkým omylem. Analytik data může připravit, ale jeho povinností je pouze data definovat.

## **Tester**

Práce testera nebývá vždy prostá či přímo řízená. Tester se ve své pracovní náplni vždy musí potýkat s mnohými překážkami a správně je musí identifikovat. Překážky, se kterými se tester může potýkat mohou být snad jakéhokoliv charakteru. Nejčastějšími překážkami jsou selhání systému, bug, nelogický chyba v dokumentaci nebo také nesprávně napsané testy, dle kterých se tester řídí. Hlavními odpovědnostmi testera jsou tedy exekuce testů, odhalování chyb a jejich popis, urgence opravy chyb a jejich opětné re-testování.

Toto byly odpovědnosti běžného testera, který vykonává převážně manuální testy. Existuje rovněž tester, který vykonává automatizované testy, a to hned z několika důvodů, které jsou definovány v kapitole mimo funkčních požadavků. Tomuto testerovi dále pak přibývají zodpovědnosti jako jsou analýza provedených automatizovaných testů, reportování výsledků a případně návrh na zlepšení systému.

### **3.2.6 Struktura přípravy testů**

Do jedné z nejběžnějších metodik pro tvorbu testovací analýzy patří i vytváření testovacích sad, scénářů a kroků. Jejich hierarchické rozdělení má svůj velký význam, který je vysvětlen níže. Pro laickou veřejnost je nejjednodušší vysvětlit toto rozdělení na množinách. Testovací sada obsahuje více testovacích scénářů a testovací scénář obsahuje více testovacích kroků.

#### **Testovací sada – Test set**

Testovací sada neboli Test Set vyjadřuje množinu nebo také sdružení testovacích scénářů, které mají společné charakteristiky. Jde o analytické rozdělení například podle funkcionalit, modulů, systémů, integrací, dat a tak dále. Každý testovací set nebo sada by měli obsahovat právě ty testovací scénáře, které spolu souvisí. Níže je uvedený příklad, kde testovací sadu představuje zelená barva v tabulce. Jedná se o testovací sadu „Přihlášení“, která má za úkol testovat přihlašovací metody do internetového bankovníctví. Jednotlivé scénáře pak budou popisovat, jak metodicky postupovat. Každá testovací sada by měla obsahovat id, název, stručný popis testovací sady a pole na poznámky pro budoucí testery.

#### **Testovací scénář – Test case**

Testovací scénář neboli test case je podmnožinou testovací sady a obsahuje testovací kroky. Každý testovací scénář by měl být konkrétně zaměřený na testovanou oblast, nikoliv obecně jako u testovací sady. Testovací scénáře by měly být definovány buď jako pozitivní, tedy že si kladou za cíl pozitivní výsledek anebo, negativní, tedy že si kladou za cíl negativní výsledek. Každý testovací scénář by měl obsahovat id, jednoznačný název, stručný popis scénáře, finální výsledek a poznámky pro budoucí testery.

## Testovací krok – Test step

Testovací krok neboli test step je podmnožinou nebo také součástí testovacího scénáře. Testovací krok má dvě základní části. V první části definuje, co má tester vykonat a ve druhé části, co je očekávaným výsledkem. Každý krok musí být srozumitelný pro každého testera, aby tester, který bude testovat konkrétní scénář nemusel shánět autora testovacího scénáře. Dále testovací krok musí být atomický. Udává jednoznačnou informaci o akci a reakci.

Testovací krok musí obsahovat mnohem více informací než sada nebo scénáře. Jedná se například o id, název kroku, detailní popis kroku, očekávaný výsledek, poznámku pro aktuální stav, výsledek kroku (splněno / nesplněno / odloženo atp.) a spoustu dalších volitelných atributů.

TestSet_Name	TestCase_name	TestStep_name	Step_description	Expected_result
Přihlášení				
	<u>Přihlášení</u> <u>certifikátem</u>			
		Precondition	Jsou připraveny klientské certifikáty k autorizaci	Splněno
		Step - 1	Zobraz si stránku internetového bankovníctví	Zadáno, výsledky nalezeny.
		Step - 2	Vlož klientský certifikát do pole s titulkem autorizovat	Vloženo

Tabulka 1 - Kaskádová struktura rozložení testů

### 3.2.7 Rozdělení efektů snižujících kvalitu softwaru

Během testování nebo jakéhokoliv procesu zlepšování kvality software je nutné zaznamenávat nežádoucí vlivy, které by mohly nějakým způsobem snižovat kvalitu software. Každý tento nepříznivý efekt je potřebné nejprve detekovat a pak analyzovat. Velmi často jsem se v praxi setkal s tím, že kolegové zaznamenají nežádoucí chování, ale už většinou nemají možnost přesně popsat a definovat objevený nález.

#### **Chyba**

Chyba je velmi často označována jakožto nežádoucí výsledek nebo chování software, které dále nelze za použití dostupných nástrojů identifikovat. Chybový nález je prakticky cokoliv, co není uvedeno v požadavcích a zároveň je nežádoucí.

#### **Bug**

Bug neboli z převzatého anglického jazyka „brouk“ je takový nežádoucí jev, který v softwaru nastává při zanešení chyby programátorem do kódu. Takovýto bug můžeme vnímat hned z několika hledisek. Nejčastějšími to jsou:

- 1) Syntaktické bugy: Tento bug nastává při nedodržení správnosti syntaxe jazyka. Velmi často se stává, že programátor začíná programovat v jiném programovacím jazyce, než je zvyklý, a tudíž omylem dodržuje syntaxi jazyka předešlého.
- 2) Sémantické bugy: Již každému začínajícímu programátorovi se stalo, že vytvářel rekurzivní algoritmus, který se mu tak zvaně zacyklil. Kompilátor jazyka nemá problém s přeložením napsaného kódu, nicméně logika algoritmu je nedostatečná, a tak se stane, že program využije všechnu RAM paměť a poté se v lepší případě ukončí.
- 3) Neočekávané události: Nastávají při běhu programu, kdy se autor programu například zmýlil ve znaménku a očekávaný výstup ihned získává neočekávanou hodnotu.

## **Defekt**

Jako defekt se velmi často označuje synonymum pro chybu. Jelikož software se velmi často vyvíjí na odlišných prostředích, tak také vznikají i odlišné chyby, které se mohou jevit na první pohled stejně, nicméně jejich příčina může být různá. Defekt se tedy používá pro odfiltrování chyb mezi prostředími. Nejčastěji tento filtrační vztah bývá mezi prostředím produkčním a testovacím nebo také vývojovým.[23]

## **Selhání**

Selhání je velmi častým efektem nepřízně v procesu zkvalitňování softwaru. Selhání se často zaměňuje za bug, a to především z důvodu nedostatečné analýzy nežádoucího efektu nebo také z nedostatečných nástrojů a zdrojů pro vykonání analýzy nad nálezem.

Selhání lze definovat jako nežádoucí chování softwaru za příčiny vnějších vlivů. Jedná se například o situaci, kdy jiný software nebo i člověk zasílá nevalidní údaje ke zpracování a podle nich je také nevalidní i výsledek. Nejde ale jen o vstupní data, může se jednat i hmotné prvky jako je například pomalá komunikace mezi síťovými prvky, magnetická pole, bouřky nebo také selhání hardware zdrojů, které jsou důležité pro provozování software.

## **Riziko**

Riziko je reportovaná událost, která je do budoucna přínosná a pomůže předcházet nežádoucím jevům. Asi nejčastějšími riziky jsou rizika z pohledu nefunkčních požadavků. Většinou to jsou odhalení běžným průchodem aplikace při zatížení uživateli. Jedná se tedy incidenty na kapacitní požadavcích, které nastanou příčinou nedostatečného hardware nebo strukturou kódu, která neprošla refactoringem.

[24][25]

### **3.2.8 Závažnosti efektů snižujících kvalitu software**

Závažností nebo také cizím slovem severitou se označuje závažnost dopadu nežádoucího efektu na testovaný software. Tyto závažnosti musí být dle analytického hlediska rozděleny do kategorií, které poté určují závažnost nepříznivého dopadu. V dnešní době stále neexistuje metodika, která by přesně stanovovala, jak má být nežádoucí dopad posouzen. A proto zde uvádím hodnocení, které je publikováno na největším množství zdrojů.

#### **Kritická závada (Critical)**

Jako kritická závada se uvažuje nedostupnost systému nebo jeho extrémní kolísání. Jde také o stav, kdy systém neodpovídá na vstupní parametry či se s nimi neumí vypořádat dle náležitých funkčních požadavků. Dle definice ISTQB jde o závadu na kritických funkcích či kritických datech. Z hlediska testerského, jde také o vadu, která znemožňuje dokončení rozpracovaného testovacího scénáře.

#### **Vysoká závada (High)**

Závada s vysokou závažností ovlivňuje důležité funkce nebo data programu. Může jít například o nekompatibilitu s jiným modulem či nesprávné zpracování části vstupních údajů. Z testerského hlediska je tato chyba závažná ale měla by dovolit projetí na další krok testovacího scénáře.

#### **Střední závada (Medium)**

Závada se středním dopadem do systému je taková závada, která není příliš významná a jde o efekt, který není hrozbou pro systém jako takový, ale spíše je viditelný pro uživatele, kterého může dráždit. V takovémto případě se závada zaznamenává a odkládá na později nebo přímo na konec vývoj systému. Z testerského hlediska se nejedná o velkou závažnost a určitě neblokuje přechod na další krok testovacího scénáře.

### **Nízká závada (Low)**

Nízké závady jsou většinou informativního charakteru a nijak nepoškozují data či běh systému. Převážně se jedná o závadu grafické či gramatické. Z testerského hlediska je tato závada velmi banální a určitě dovoluje přejít na další krok testovacího scénáře.[30]

## **3.3 Teoretická východiska ze studií použitelnosti**

Testování či studie použitelnosti je v praxi velmi často zanedbávaná, a to ať z důvodů velké nákladnosti či pracnosti, tak i z důvodů, že si lidé myslí, že není tolik užitečná, jak by ve skutečnosti mohla být. V tomto případě je to velký omyl. Uživatelská testování použitelnosti nejsou sice levnou záležitostí při správném provádění, za to však mohou přinést výsledky, které zdokonalí nově vznikající produkt a přiblíží ho ke koncovému zákazníkovi, který si ho pro jeho intuitivnost oblíbí a tím vznikne další poptávka po produktu. Je velmi důležité, aby se systémy přizpůsobovali lidem, a nikoliv lidé systémům.

### **3.3.1 Testovací scénáře pro participanty**

Existují dva doporučení styly testovacích scénářů. Nazývají se Cognitivní a Heuristický. Dle pravidel těchto stylů sestavíme scénáře, které pak předložíme participantům a sledujeme jejich chování a práci se systémem.

#### **Heuristický styl**

Scénář pro Heuristický styl by měl vypadat jako sled kroků, které chceme testovat. Takový scénář definuje cíl, kterého má testovaný subjekt dosáhnout a zároveň i jeho cestu. Scénář se tvoří za pomoci definovaných Use Case. Takovýto scénář je velmi dobrý pro pozorovací metodu „Eye Tracking“, která pomocí kamery upevněné k obrazovce sleduje pohyb očí testovaného subjektu. [26]

Po vykonání testu Heuristickým stylem následuje vyhodnocení, které se skládá ze tří částí:

- 1) Popis špatných / nevhodných / neintuitivních Use Case.
- 2) Návrhy nových Use Case nebo oprav, které vychází z bodu jedna.
- 3) Doporučení na další úpravy a vylepšení.

#### **Příklad Heuristického scénáře:**

1. Zaregistruj se
  - a. Otevři si naši webovou stránku
  - b. Na hlavní stránce klikni na účet a přejdi na stránku registrace
  - c. Zaregistruj se
2. Oznam, že jsi našel dron
  - a. Otevři si naši webovou stránku
  - b. Zhruba ve středu stránky najdi tlačítko „našel jsem dron“
  - c. Oznam, že jsi našel dron
3. Zjisti v článcích, co se píše nového
  - a. Otevři si naši webovou stránku
  - b. V horním menu obrazovku najdi sekci články a klikni na ní
  - c. Zjisti, co se nového píše v článcích

#### **Cognitivní styl**

Cognitivní styl na rozdíl od Heuristického udává v testovacím scénáři participantovi pouze cíl, kterého má dosáhnout. Scénář je sestavený tak, aby testovaný subjekt mohl libovolně procházet Use Casy. V průběhu testu tyto cesty zaznamenáváme a občas se můžeme zeptat proč tuto cestu participant zvolil. [27]



### **Po konci testu nás zajímá ve vyhodnocení:**

- 1) Zhodnocení intuitivně nejlepších, průměrných a slabých Use Case.
- 2) Popis špatných / nevhodných / neintuitivních Use Case.
- 3) Návrhy nových Use Case a odstranění chybných nedostatků.
- 4) Doporučení na vylepšení Use Case a změny pro pracovní postupy k dosažení cíle.

### **Příklad Cognitivní scénáře:**

1. Zaregistruj se
2. Oznam, že jsi našel dron
3. Zjisti v člancích, co se píše nového

### **3.3.2 Interview**

Po dokončení testování přichází nejdůležitější část testů a to interview. Rozhovor s participantem by měl být veden přátelsky ale kromě toho by měl participant cítit nadšení či ochotu pomoci při vývoji systému a stát se na chvíli součástí vývojového týmu. Tento pocit by v něm měl nadchnout vedoucí pohovoru. Předpokládejme tedy, že participant je již připraven dle našich očekávání na rozhovor o testech a testovaném systému a ptáme se ho na otázky typu:

- 1) Likes – Co se ti v systému nejvíce líbilo? Co bylo nejvíce zřejmé a proč?
- 2) Dislikes – Co se ti v systému vůbec nelíbilo? Proč se ti to nelíbilo?
- 3) Likes again – Znovu se ptáme, co se participantovi líbilo, participant si vzpomene ještě maximálně na jednu až dvě věci, ale především se rozpomene, co bylo špatné.
- 4) Recommendations – Ptáme se participanta, co by na systému napravil a jak by to udělal.

[27]

Definované otázky jsou definovány pouze pro teoretická východiska. V praktické části by otázky měly být zaměřeny racionálně a na praktické věci. Konkrétním příkladem může být: Kolik zde očekáváte řádků v tabulce?

Mezi odpověďmi participanta vždy dodržujeme interval alespoň 10 sekund a rozhodně neskáčíme participantovi do řeči. Tento krátký interval vyvolá v participantovi malý pocit nejistoty či nervozity z tich a nechá mu prostor pro vyjádření dalších informací.

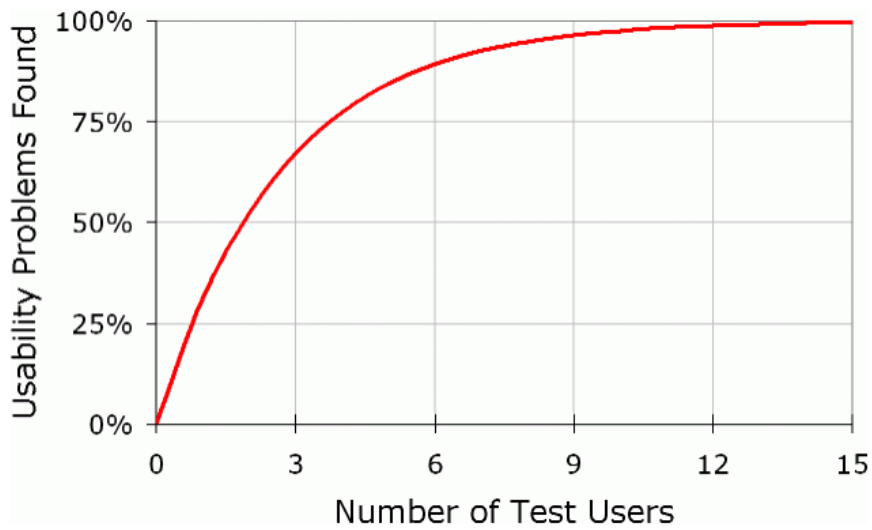
Další prostředkem pro získání informací od participanta jsou tak zvané emoční karty. Po ukončení rozhovoru dejte participantovi karty s různými pocity a emocemi a požádejte ho, aby Vám vybral 5 nejlépe vystihujících slov k Vašemu systému.[28]

### **3.3.3 Počty participantů pro studie použitelnosti**

Dle doktora Jacoba Nielsena pro dostačující výsledek uživatelských testů při studiích použitelnosti stačí, aby se zúčastnilo pět participantů. Těchto pět participantů dle následující křivky odhalí dostatečný počet problémů v použitelnosti systému, což je 75%. Dle studií doktora Nielsena a Toma Landauera je vztah křivky následující:

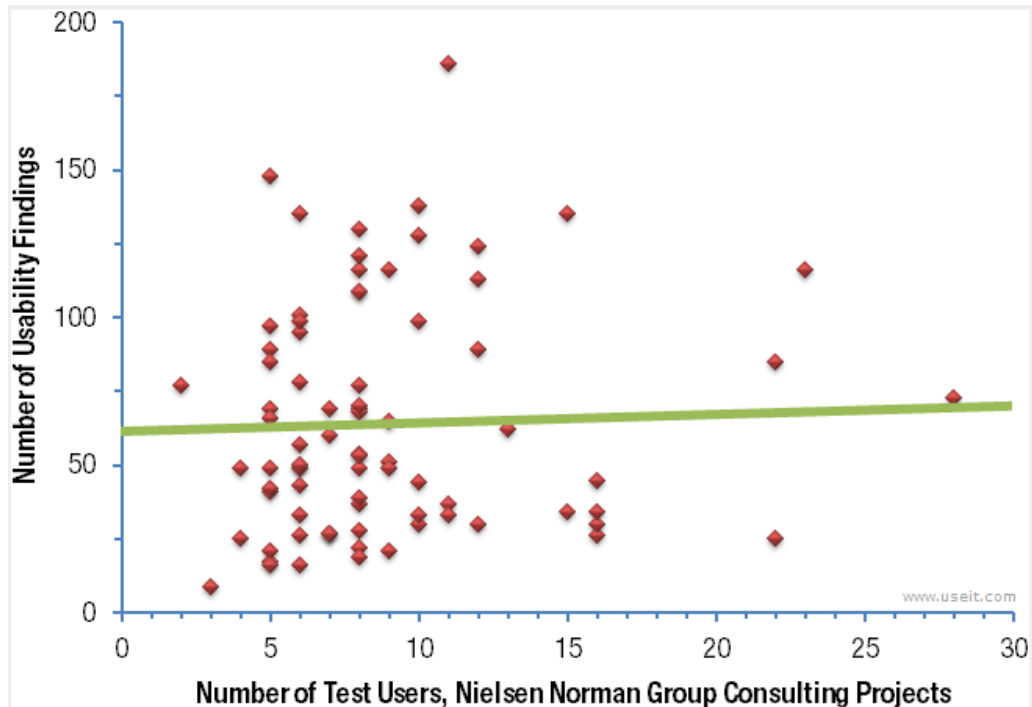
$$N(1-(1-L)^n)$$

N představuje celkový počet problémů a L je poměr problémů použitelnosti zjištěných při testování jednoho participanta. Hodnota L typicky nabývá kolem 31%, což bylo zjištěno studováním velkého počtu projektů. Průměrně tedy jeden participant vždy odhalil cca 31% problémů s použitelností a proto vztah uvedený výše vykresluje křivku zde na grafu.



Obrázek 6 - Křivka získaných informací vs celkové počty respondentů [3]

Na níže uvedeném grafu doktor Jacob Nielsen zobrazuje počty nálezů ku počtu testovaných subjektů.



Obrázek 7 - Zastupitelnost počtu získaných informací vs celkové počty respondentů [4]

Testování uživatelů bylo prováděno na více projektech. V tomto sledování doktor Nielsen a jeho tým dokázali, že největší vypočítané množství uživatelských nálezů je při počtu osmi participantů v poměru mezi přínosy a výdaji. Jde o to, že největší počet nových nalezených výskytů je zaznamenán do deseti uživatelů a nejvyšší zaznamenaný počet je v případě, kdy uživatelů bylo 8. Každý participant stojí náklady, které nesouvisí jen s jeho pořízením ale i s jeho hledáním a vybíráním, aby odpovídal definovaným personám. Proto doktor Nielsen doporučuje provádět uživatelské testování použitelnosti na 8 uživatelích.[29]

### **3.3.4 Průběh testování s participantem**

I participant je lidská bytost a žádá si určitá zacházení, a proto je nutné dodržet několik následujících pravidel, abychom dostali potřebné výsledky a participant se cítil dobře.

#### **1 Informujte participanta, co vše ho čeká**

Při první příležitosti přivítejte testera a představte mu blízké okolí, které se testování rovněž účastní. Nabídněte testerovi něco k pití a například sladký bonbón, který sníží napětí, které by participant mohl pociťovat. Sdělte testerovi, že je dobré, když bude přemýšlet nahlas, abyste si mohli dělat poznámky. Ukažte testerovi připravené scénáře a ujistěte ho, že v testu nemůže udělat žádnou chybu. Tedy, že všechna rozhodnutí jsou správná. Sdělte mu rovněž jak dlouho, která akce bude cca trvat a ať nikam nespěchá, neboť mu dopřejete času, kolik bude potřeba. To je velmi důležité, protože tester, který spěchá dělá chyby.

#### **2 Testování konkurence vyvíjeného systému**

Je potřebné, aby tester postupoval dle připraveného scénáře. Jestliže tester, nějaký úkol nemůže udělat z jakéhokoliv důvodu, tak víte, že tento prvek není dostatečně intuitivní. Testerovi se poděkuje a sdělí se mu, že tento prvek je špatně navržen a přejdete na další úkol. Je zbytečné nechat testera se stresovat s úkolem, který pro něj není lehce splnitelný.

### **3 Interview**

Po ukončení testování se vyptejte na předem stanovené otázky. Budete-li muset improvizovat, tak se držte stanovených metodik pro interview, které jsou definované výše v kapitole o interview. V tento moment Vám participant doplní poznámky ze svých pocitů a vy získáváte nejdůležitější data. Po interview je dobré použít metodu emočních karet, která zabere pouhých pár minut.

### **4 Rozloučení**

V závěru setkání participantovi poděkujete a předáte odměnu. Rovněž se pobavíte o jeho zkušenostech s jiným uživatelským testováním, a tak získáte i své osobní ohodnocení. Toto je i příležitost navázat s participantem další spojení, které můžete využít pro další testování.[28]

#### **3.3.5 Charakteristiky pro vyhodnocení kvalitativního šetření**

Aby bylo možné správně ohodnotit výstupy kvalitativního šetření, je potřeba si pro to stanovit charakteristiky či měřitelné jednotky, kterými pak objektivně výstupy šetření ohodnotíme.

Dle doktora Jacoba Neilsena je jednoduché si stanovit charakteristiky, dle požadovaných kritérií výstupů, ale horší částí je jejich sběr a vyhodnocení.

Typickými charakteristikami bývají:

- 1) Úspěšnost – kolik úkolů respondent zvládl dokončit
- 2) Čas – kolik času trvalo respondentovi, aby dokončil úkol
- 3) Chybovost – protiklad splnění cílů
- 4) Uživatelský subjektivní pohled [31]

### **3.4 Vybrané charakteristiky kvalitativního šetření**

Ve své praktické části budu chtít sledovat následující charakteristiky čas, překlikatelnost, intuitivnost, úspěšnost, naučitelnost.

#### **3.4.1 Měřitelnost stanovených charakteristik**

Aby charakteristika byla měřitelná je potřebné si definovat, jak budeme jednotlivé charakteristiky hodnotit. Každá charakteristika může mít dle subjektivního hlediska jiné ohodnocení. Nicméně, aby bylo měření, co nejpřesnější, stanovil jsem, že každá z uvedených charakteristik bude nabývat maximálně tří hodnot, které následně budou numericky interpretovány na ose či pavučinovém grafu v bodovém ohodnocení 1–3, kdy 1 je nejnižší a 3 nejvyšší známka úspěchu. Procesu převádění slovních výsledků se nazývá kvantifikace výsledků.

- Čas: rychlý bez váhání, středně rychlý s malým zaváháním, pomalý se zaváháním nebo s pomocí
- Překlikatelnost: nepřeklikaný, středně překlikaný, překlikaný
- Naučitelnost: rychle naučitelný, naučitelný, nenaučitelný
- Intuitivnost: intuitivní, středně intuitivní, neintuitivní
- Úspěšnost: splnil 3 cíle, splnil 2 cíle, splnil 1-0 cílů

#### **3.4.2 Hodnocení charakteristik**

##### **Intuitivnost**

Intuitivnost se často považuje za agregovaný výsledek sbíraných údajů. Tento parametr hodlám použít, jakožto subjektivní hledisko testovaného subjektu na konkrétní testovaný cíl. Jednotky tohoto parametru budou intuitivní, středně intuitivní a neintuitivní. Jednotlivé výsledky budou sbírány pomocí odpovědí od respondenta.

## **Úspěšnost**

Úspěšnost je jedním z nejlépe hodnocených parametrů, jelikož udává, kolik testovaných cílů respondent úspěšně splnil. Nabývá hodnot „splnil 3 cíle“, „splnil 2 cíle“, „splnil 1-0 cílů“.

## **Čas**

Jako čas vnímáme jednotku, která lze měřit a ihned matematicky interpretovat. Já si tento parametr volím opět jako subjektivní jednotku. Nechci při testování mít zapnuté stopky, které budou měřit respondentovi ubíhající čas. Takové šetření by mohlo mít velmi zkreslené výsledky, například respondentovou nervozitou.

Časový parametr bude nabývat hodnot „rychlý bez váhání“, kde respondent bude okamžitě vědět, jakou operaci a kde má vykonat. Další hodnotou je „středně rychlý s malým zaváháním“, kdy respondent bude vědět, co má dělat, jen bude chvíli přemýšlet. Poslední jednotkou je „pomalý se zaváháním nebo s pomocí“, tato jednotka bude udávat, že respondent nemohl rychle splnit požadovaný cíl. Jednotlivé výsledky budou sbírány v průběhu testu, který bude respondent absolvovat.

## **Překlikatelnost**

Překlikatelnost je dalším ze subjektivních parametrů, který bude vyhodnocován obdobně jako Intuitivnost. Překlikatelnost bude nabývat hodnot nepřeklikaný, středně překlikaný, překlikaný. Jednotlivé hodnoty mají význam subjektivního charakteru získaného od respondenta. Jde v podstatě o to, jak dlouho a zároveň v jaké pohodlnosti se respondentovi podařilo dosáhnout cíle. Sběr těchto hodnot bude opět pomocí řízeného rozhovoru.

## **Naučitelnost**

Naučitelnost je jedním z nejobtížnějších parametrů z pohledu získávání dat. Tuto charakteristiku budu měřit pomocí opakování cílů. Pokud testovaný subjekt, nedosáhne alespoň jednoho cíle, a to buď v krátkém čase nebo vůbec, tak tento cíl na konec šetření bude opakován. Je třeba dodat, že testovanému subjektu bude cesta k dosažení cíle ukázána po neúspěšném prvním pokusu.

Pokud si testovaný subjekt bude cestu stále pamatovat a zvládne dosáhnout cíle v krátkém čase, bude tento pokus ohodnocen jako rychle naučitelný. V ostatních případech dle subjektivity naučitelný či nenaučitelný.

### **3.4.3 Interpretace výsledků pomocí kvantifikace**

#### **Kvantifikace výsledků**

Výsledky budou zapsány do tabulek a následně převedeny na číselné hodnoty. Následně pomocí bodovací metody ohodnoceny, od nejlepších pro nejhorší. To znamená, že nejhorší výsledek bude ohodnocen jedním bodem a nejlepší výsledek bude ohodnocen maximálním možným výsledkem tři.

#### **Interpretace pomocí čísel**

Samotná interpretace již kvantifikovaných výsledků bude vynesena do tabulek, číselných intervalů a pavučinových grafech vždy ve třech fázích.

Příklad pro vyhodnocení jednoho testovacího subjektu.

Předpokládejme, že testovaným subjektem je respondent v roli osoby tester a má za cíle dosáhnout 3 funkčních prvků pomocí hlavních nebo vedlejších usecase.



## 1) Fáze 1: Kvantifikované vyhodnocení

Charakteristik	Cíl 1	Cíl 2	Cíl 3	Součet	Max body
Intuitivnost	2	3	3	8	9
Úspěšnost	3	3	3	9	9
Čas	2	2	3	7	9
Překlikatelnost	1	2	3	6	9
Naučitelnost	3	3	3	9	9
<b>Suma bodů</b>	<b>11</b>	<b>13</b>	<b>15</b>	<b>40</b>	<b>45</b>

Tabulka 2 - Ukázková součtová tabulka

Zdroj: vlastní tvorba

## 2) Fáze 2: Vynešení výsledků na intervaly

Pro každý cíl je interval rozsahu 1-15 uzavřený.

Number line:

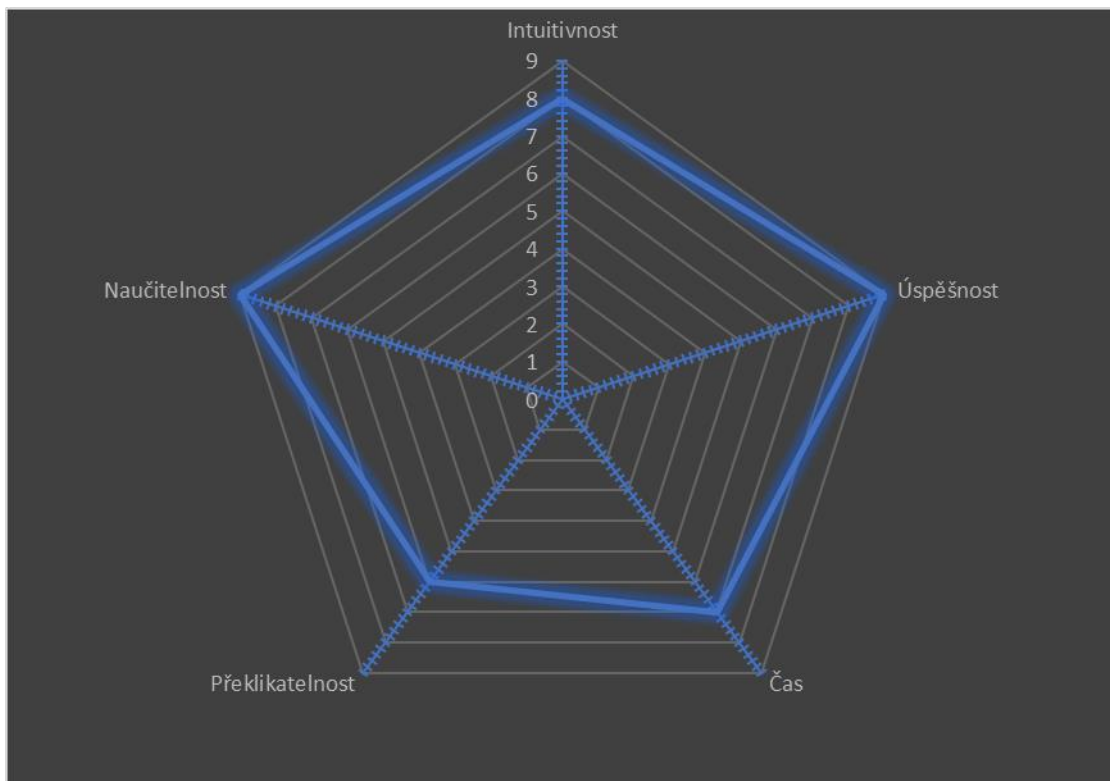


Obrázek 8 - Ukázkový interval nekonkrétního cíle testu [5]

- Pro celkové vyhodnocení jednoho respondenta je interval 15-45 uzavřený.
- Pro celkové vyhodnocení respondentů jedné osoby je 45-135 uzavřený (při předpokládaném počtu respondentů 3 na osobu).
- Pro celkové vyhodnocení respondentů všech osob a tedy vyhodnocení za systém jako celek je interval 135-405 uzavřený (předpokládáme 3 osoby násobeno třemi respondenty zastupujícími každou osobu).
- Procentuální hodnocení budu měřit jako počet bodů ponížený o minimumální hodnotu interval, kterou budu dělit maximumální hodnotou interval poníženou o minimumální. Výsledek budu násobit stem, abych dostal procenta úspěchu.

### 3) Fáze3: Vizualizace dat na pavučinovém grafu

Tento pavučinový graf je konkrétně vizualizací dat, které jsou vyplněny v tabulce u vyhodnocovací fáze 1. Jedná se o data, která jsou simulací jednoho kvalitativního šetření pro celkový součet bodů z celkového šetření.



Obrázek 9 - Ukázková vizualizace na pavučinovém grafu

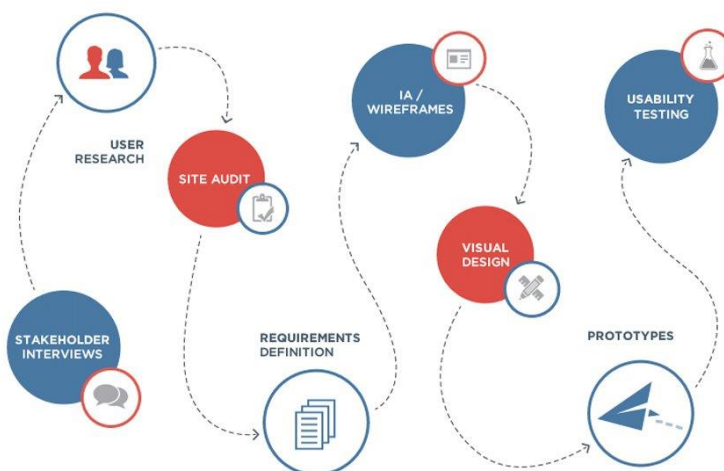
### 3.5 Celkový náhled na proces studie použitelnosti v praxi

Výzkum studie použitelnosti, jak ho známe se dělá již pro konkrétní statek či službu, které budou mít své zákazníky. V této práci se zaměřuji na návrh software, a tak se budu v dalších větách zaměřovat právě na software.

V praxi se postupuje následujícím způsobem:

- a. Nejprve zjistíme, co si zákazník přeje dodat za software.
- b. Definujeme osoby, dle kterých bude možné získat požadavky na systém.
- c. Zjistíme požadavky zákazníka a prokonzultujeme s potenciálními uživateli
- d. Vytvoříme jednoduché rámy obrazovek, kvůli zjištění návaznosti mezi nimi.
- e. Zkontrolujeme se zákazníkem, zda jsou návaznosti v pořádku.
- f. Vytvoříme funkční prototyp aplikace či systému.
- g. Provádím uživatelské testování s participanty odpovídající personám.

[32]



Obrázek 10 - Proces studie použitelnosti v praxi [6]

## **4 Praktické výstupy**

V kapitole o teoretických východiscích jsem nastudoval a zapsal podkladové informace k vypracování vlastních praktických výstupů.

V této kapitole pojmenované jako „Praktické výstupy“ se budu snažit splnit všechny cíle, které patří do praktických výstupů, objektivním pohledem zhodnotit výstupy mé práce pomocí kvalitativního šetření a napsat závěr.

### **4.1 Motivace**

Motivací k vypracování této diplomové práce jsou fakta týkající se nástrojů na správu testovacích scénářů k testování software a jejich vyhodnocování. Z vlastních zkušeností vím, že dostupné nástroje v dnešní době jsou velice drahé nebo velmi nepřehledné, anebo také velmi nedostačující po funkční stránce.

Většinou menší společnosti si nemohou dovolit nákladné nástroje, které by byly dostačující, a tak používají jako náhradu například nástroje MS Office, které jsou rozhodně nedostačující a snižují kvalitu testování. Naopak společnosti, které si drahé nástroje mohou dovolit pak velmi často nevyužívají velkou většinu naprogramovaných funkcí, jelikož nástroje jsou velmi složité a lidsky řečeno přepřácané.

### **4.2 Definice cíle UI Specifikace**

Na základě teoretických východisek a pozorování navrhnu takový systém, který bude možné prodávat za cenu, kterou si mohou dovolit i menší firmy. Svou použitelností bude jednoduchý nicméně se všemi potřebnými funkcemi.

### **4.3 Personifikace – praktický výstup**

V diplomových pracích se osoby používají ke stavbě dvou základních pilířů UI specifikace. Prvním pilířem je sběr dat a informací k položení základních kamenů funkčních prvků UI specifikace a druhým pilířem je výběr respondentů, kteří se budou účastnit kvalitativního šetření a podají tak objektivní pohled na použitelnost aplikace.

### 4.3.1 Persona 1 – Tester (Martin Zkoumavý)

<b>Kariéra a vzdělání</b>	<b>Koníčky</b>	<b>Osobní údaje</b>
Gymnázium	Přístrojové potápění	Martin Zkoumavý
VŠ s inženýrským zaměřením	Horolezectví	Věk 27 let
2 roky praxe jako tester	Četba dobrodružných románů	Nezadaný
Certifikace ISQTB	Hraní akčních videoher	Pracující
		Žijící v Praze

#### **Charakteristika osobnosti**

Martin je mladý aktivní člověk, který nemá problémy postavit se novým výzvám. Jak již jeho koníčky napovídají, tak je malým sportovním dobrodruhem, přestože jeho tělesná kondice zrovna nepřipomíná atletickou postavu. Je to velký detailista a stejně jako u potápění, tak i v zaměstnání je to jednou z jeho největších výhod. Martin je zároveň i velmi hravý, a proto je také na pozici testera velmi ceněný, protože svou hravostí zkouší různé alternativní scénáře a metody testování, kterými často odhalí nečekané nedostatky.

#### **Běžný den**

Když Martin zrovna nezažívá nějaké dobrodružství, tak jako asi každý pracující člověk vstává ráno do práce kolem sedmé hodiny. Vyčistí si zuby, vysprchuje a odchází do práce. Jelikož nechce ztrácet čas snídáním v práci nebo doma, tak po cestě do práce si zakoupí vždy svůj oblíbený obložený chlebiček a poklidným tempem chůze ho sní. V práci rozbalí své pracovní nástroje a kontroluje zbývající a nově přiřazenou práci. Martina při práci často brzdí nedostatečné informace k testování a velké množství funkcí v nástrojích pro řízení testů. Martin si často vyžádá od test analytiků surové testy napsané v tabulkových editorech, které jsou pro něj přehlednější.

### 4.3.2 Persona 2 – Test Analytik (Lucie Pečlivá)

<b>Kariéra a vzdělání</b>	<b>Koníčky</b>	<b>Osobní údaje</b>
Obchodní Akademie	Zdravá výživa	Lucie Pečlivá
VŠ ekonomické zaměření	Nakupování	Věk 32 let
3 roky jako business analytik	Fitness	Vdaná
2 rok jako test analytik	Procházky	Pracující
	Aranžování	Žijící v Praze

#### **Charakteristika osobnosti**

Lucie je velmi pečlivý člověk. K její osobnostní pečlivosti také přibývá fakt, že je žena a, jak to u většiny žen bývá, tak je od přírody zaměřená na detail. K jejím zálibám jako je aranžování a fitness patří také aranžování. Lucie si dává opravdový pozor na detail až se může zdát, že to někdy přehání a úzce se zaměřuje na jednu věc a nestíhá pak ostatní.

Lucie má již za sebou pár let pracovních zkušeností v korporátním prostředí, a tak není divu, že veškeré informace má pořád střežené a pokud jí není něco jasné, tak se okamžitě ptá a bádá po skutečnostech.

#### **Běžný den**

Lucie vstává brzy ráno, aby se mohla vypravit do nedalekého fitness centra. Poté co si zacvičí, tak se doma osprchuje a oblékne do práce. Nerada snídá za chůze, a proto si připravuje krabičky s potravinami do práce. Jednou z prvních úloh v práci je, že si zapne svůj pracovní počítač a přečte příchozí emaily, které by jí mohly pomoci v práci. Po klidnějším ránu většinou nadchází krušnější dopoledne, kdy za ní chodí testeři a ptají se na detaily analýzy, kterou zpracovala nebo zpracovává. Odpoledne má vyhrazený čas na schůzky a sběr informací k právě tvořeným test analýzám. Lucie kromě toho, že píše testovací analýzy a často připravuje testovací data, tak také navrhuje zlepšení procesů společnosti, pro kterou právě pracuje.

### 4.3.3 Persona 3 – Test Manažer (Jan Generál)

<b>Kariéra a vzdělání</b>	<b>Koníčky</b>	<b>Osobní údaje</b>
SŠ Elektrotechnická	Lodní plavby	Jan Generál
VŠ Manažerské informatiky	Tenis	Věk 35 let
5 let jako test analytik	Auta	Rozvedený
2 roky jako test manažer	Elektronika	Pracující
	Programování v Javě	Žijící v Praze

#### **Charakteristika osobnosti**

Jan je velmi cílevědomý muž, který objektivně hodnotí názory ostatních a spravedlivě rozhoduje. Jeho velkou výhodou jeho charismatický hlas, kterým dokáže své podřízené nasměrovat tou správnou cestou a umožnit mu klidné vyjednávání se svými oponenty. Jan je rozvážný a při své kariéře se už několikrát setkal se zajímavými lidmi, kteří ho dokázali nasměrovat na manažerskou kariéru.

#### **Běžný den**

Jan příliš brzo nevstává a rád si přispí. Vypije ranní kávu a vyráží do práce. Kolem deváté hodiny ranní se již většinou nachází na svém pracovišti a zkoumá počty chyb, odpracovaných testů a kontroluje, jestli jde vše podle plánu. Po obědě pořádá s týmem krátké schůzky, kde se všichni dozví, co mají na plánu, co splnili, nebo také, co je jejich aktuální překážkou v práci. Pokud se taková překážka vyskytne, tak Jan je odpovědný jí řešit. V průběhu dne většinou nemá moc času na své výstupy, a tak využívá čas k tomu, aby vedl tým správným směrem. Jan často zůstává v práci déle nebo alespoň odchází později, kdy ostatní už odešli. Tento čas využívá na své vlastní výstupy. Jan má velké zkušenosti v oboru testování software a ví, že jednou z hlavních věcí, které musí dělat je plánování, které mu jde s čísly dobře. Nicméně k plánování potřebuje kromě čísel i grafy a značnou část jeho pracovního času stráví tím, že dohledává jednotlivé údaje, zapisuje je do tabulek a následně z nich generuje grafy, které pak prezentuje na schůzkách.

## 4.4 Sběr informací a podkladů pro vypracování UI specifikace

Sběr informací byl proveden pomocí řízených rozhovorů s profesionály z praxe. Aby výsledky nabývaly relevantních hodnot, tak bylo vynaloženo úsilí nevybírat respondenty pouze z jedné instituce. Respondenti odpovídali stanoveným personám.

### 4.4.1 Řízený rozhovor s profesionály z praxe

Rozhovor byl veden v přátelské atmosféře na téma funkčních uživatelských prvků pro nástroj na řízení kvality software. Pro rozhovor jsem vybral 3 testery, 3 test analytiky a 3 test manažery.

Bylo očekáváno, že nejvíce podkladů do práce přinesou tzv. test manažeři, kteří mají letité zkušenosti s řízením týmů. Nicméně se ukázalo, že méně zkušení testeři, kteří pracují s nástroji na řízení kvalit software denně, a to několik hodin, přinesli do mé práce více podstatných detailů než samotní test manažeři s bohatšími zkušenostmi.

#### **Během rozhovoru byly pokládány otázky:**

- 1) Jaká je Vaše denní náplň? Případně, mohl/a byste popsat Váš běžný den v práci?
- 2) Jaký systém pro správu používáte?
- 3) Jakou vlastnost či funkcionalitu systému nejčastěji navštěvujete?
- 4) Co je na této vlastnosti nevhodné? Případně, jak byste to zlepšil/a?

#### **Shrnutí výsledků rozhovorů:**

Z výsledků řízeného rozhovoru byly získány informace dle předpokladu. Funkční prvky test exekuce, test analýza, správa chyb. Kromě předpokládaných prvků bylo zjištěno, že u běžně používaných nástrojů chybí plánovací vlastnosti z čehož vyplývá další prvek Test Plán.

Od testerů byly získány další poznatky, že by bylo dobré mít souhrnné informace na hlavní stránce. Tento prvek byl označen jako Nástěnka. Další funkční prvky, které jsou popsány v kapitole o funkčních prvcích, vychází z osobních praktických zkušeností.



Ke čtvrté otázce respondenti odpovídali ve vztahu k nízké intuitivnosti systémů a rovněž ke složitosti editace či vytváření samotných testovacích scénářů.

Největším postihem je tedy zanášení testovacích scénářů do systému, na kterém se shodli všichni test analytici. Kontakty na konkrétní osoby byly uloženy a rovněž byla a dohodnuta další setkání pro vyhotovení závěrečného výzkumu této diplomové práce.

Na základě zpracování tohoto šetření jsou stanoveny funkční prvky, dle nejvyššího bodového ohodnocení. Data byla získána pomocí doporučení, která byla pořízena při řízených rozhovorech. Následně bylo vybráno 5 funkčních prvků s nejvyšším bodovým ohodnocením:

- 1) Test exekuce – 8 hlasů
- 2) Test analýza – 8 hlasů
- 3) Test plán – 5 hlasy
- 4) Nástěnka – 6 hlasů
- 5) Správa chyb – 8 hlasů

Na těchto funkčních prvcích, které vznikly z výsledků řízených rozhovorů, byly založeny podklady pro vývoj prototypu systému, jsou obohaceny o vlastní zkušenosti a poznatky. Jednou z obohacených funkcionalit je řízení přístupů dle uživatelských rolí, které jsou dále popsány v kapitole o User Story.

## **4.5 User Story**

### **4.5.1 Přehled funkčních rolí pro User Story**

Z technického hlediska může každý uživatel vlastnit jednu nebo více rolí. Z hlediska uživatelského je tato vlastnost nežádoucí, avšak pro menší firmy či projekty, kdy například test analytik provádí exekuci testů, je tato vlastnost naopak žádoucí.)

**Seznam rolí:**

- Administrátor
- Test Manažer
- Test Analytik
- Tester
- Host

(Hlavní funkční prvky představují sekce, do kterých lze připustit uživatele s přiřazenou rolí. Více podrobných informací v následující kapitole o User Story.)

**4.5.2 Seznam hlavních funkčních prvků:**

- Test exekuce
- Test analýza
- Test plán
- Analýza požadavků
- Správa účtu
- Nástěnka
- Správa chyb
- Správa uživatelských rolí

(User Story z pohledu přístupu rolí udává, jací uživatelé mají do hlavních funkčních prvků přístup, a jakých akcí v těchto funkčních prvcích mohou využít.)

### 4.5.3 User Story z pohledu rolí přístupu:

#### 1) Administrátor

- Lze nahlížet a editovat funkční prvky: Test exekuce, Test analýza, Test plán, Analýza požadavků, Správa účtu, Nástěnka, Správa chyb, Správa uživatelských rolí

#### 2) Test Manažer

- Lze nahlížet na funkční prvky: Test exekuce, Test analýza, Test plán, Analýza požadavků, Správa účtu, Nástěnka, Správa chyb
- Lze editovat funkční prvky: Test plán, Analýza požadavků, Správa účtu, Nástěnka, Správa chyb

#### 3) Test Analytik

- Lze nahlížet na funkční prvky: Test exekuce, Test analýza, Test plán, Analýza požadavků, Správa účtu, Nástěnka, Správa chyb
- Lze editovat funkční prvky: Test Analýza, Analýza požadavků, Správa účtu, Nástěnka, Správa chyb

#### 4) Tester

- Lze nahlížet na funkční prvky: Test exekuce, Analýza požadavků, Správa účtu, Nástěnka, Správa chyb
- Lze editovat funkční prvky: Test exekuce, Správa účtu, Nástěnka, Správa chyb

#### 5) Host

- Lze nahlížet na funkční prvky: Test exekuce, Analýza požadavků, Správa účtu, Nástěnka, Správa chyb

## 4.6 Hlavní vybrané usecase

### 4.6.1 UC01 – Vytvořit Nástěnku

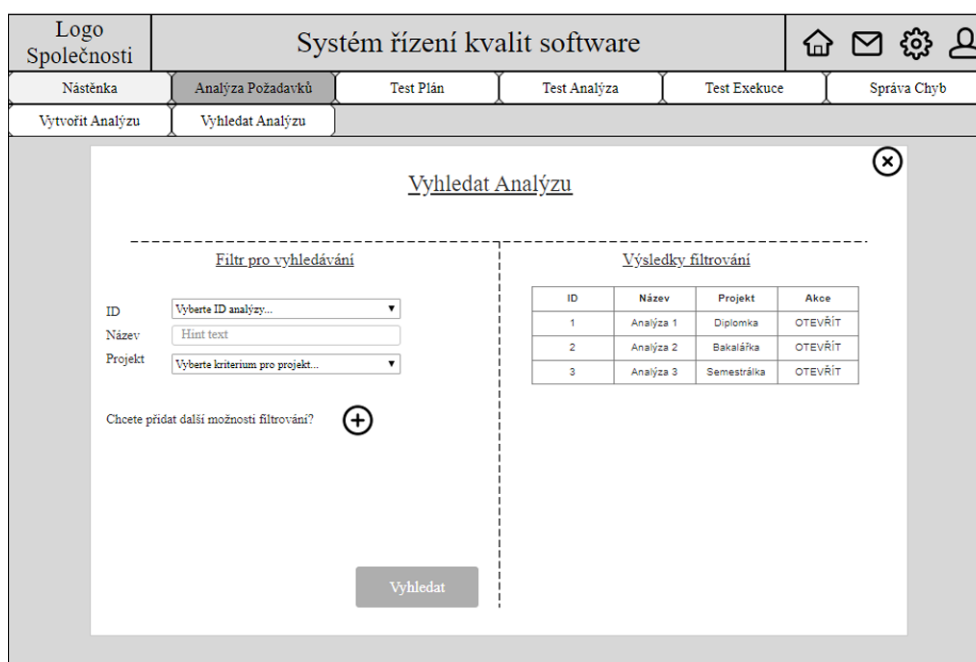
Obrázek 11 - UC01 – Vytvořit Nástěnku

#### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou hlavní stránku / nástěnku

- 1) Uživatel očekává, že po kliknutí na „Vytvořit Nástěnku“ na obrazovce „Nástěnka“ se mu zobrazí formulář pro vyplnění parametrů budoucí nástěnky.
- 2) Uživatel očekává, že při vybrání možnosti „Projektová Nástěnka“ se budou do vytvořené nástěnky zobrazovat data celého vybraného projektu. U sprintu očekává, že se mu budou zobrazovat na nástěnce pouze data vybraného sprintu z vybraného projektu
- 3) Uživatel očekává, že po vybrání možnosti „Vytvořit“ se vytvoří nástěnka na hlavní stránce s daty, které odpovídají vybraným atributům formuláře při vytváření nástěnky.

## 4.6.2 UC02 – Vytvořit Analýzu Požadavků



Obrázek 12 - UC02 – Vytvořit Analýzu Požadavků

### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku „Analýza požadavků“

- 1) Uživatel očekává, že po vybrání volby „Vytvořit Analýzu“ se mu zobrazí formulář pro vytvoření dokumentu analýza požadavků s možností zadat konkrétní požadavky včetně jejich řešení.
- 2) Uživatel očekává, že po kliknutí na šipku v pravé dolní části obrazovky, se přesune na předchozí nebo na následující požadavek, který touto možností vytvoří, jestliže již není vytvořen.
- 3) Uživatel očekává, že po stisknutí tlačítka „Uložit a ukončit“ se vytvoří dokument „Analýza požadavků“ s konkrétně vyplněnými požadavky dle vyplněného formuláře.

### 4.6.3 UC03 – Naplánovat testovací scénáře

The screenshot shows the 'Naplánovat testy' (Schedule tests) form within the 'System Quality Management' software. The interface includes a top navigation bar with 'Logo Společnosti' and 'System Quality Management' text, and a secondary navigation bar with icons for home, mail, settings, and user profile. Below these are tabs for 'Nástěnka', 'Analýza Požadavků', 'Test Plán', 'Test Analýza', 'Test Exekuce', and 'Správa Chyb'. The 'Test Plán' tab is active, and the 'Naplánovat testy' sub-tab is selected. The main form area contains several dropdown menus for 'Název test plánu' (TP 1), 'Název sprintu v test plánu' (Sprint 1), 'Název testovacího setu' (Test Set 1), and 'Název testovacího scénáře' (All...). There are also expand/collapse icons (+/-) and a checkbox for 'Všechny scénáře z tohoto setu?' (checked). A 'Uložit a ukončit' button is at the bottom right.

Obrázek 13 - UC03 – Naplánovat testovací scénáře

#### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku Test Plán. Zároveň počítá již s existencí vytvořeného test plánu se sprinty a vytvoření testovací sady naplněné testovacími scénáři pro nadcházející testování.

- 1) Uživatel očekává, že po vybrání volby naplánovat test se mu zobrazí formulář s možnostmi plánovat testy.
- 2) Uživatel očekává, že při výběru testovací sady a zaškrtnutí volby „Všechny scénáře z tohoto setu“ budou naplánovány všechny testovací scénáře ze zvolné sady.
- 3) Uživatel očekává, že při kliknutí na symbol „plus“ se mu rozšíří formulář o další atributy. V případě symbolu „mínus“ se mu formulář zkrátí.
- 4) Uživatel očekává, že při stisknutí tlačítka „Uložit a ukončit“ se naplánují testovací scénáře pro zvolaný test plán a sprint.

#### 4.6.4 UC04 – Vyhledat analýzu požadavků

Logo Společnosti

System řízení kvalit software

Nástěnka Analýza Požadavků Test Plán Test Analýza Test Exekuce Správa Chyb

Vytvořit Analýzu Vyhledat Analýzu

### Vyhledat Analýzu

Filtr pro vyhledávání

ID:

Název:

Projekt:

Chcete přidat další možnosti filtrování?

Výsledky filtrování

ID	Název	Projekt	Akce
1	Analýza 1	Diplomka	OTEVŘÍT
2	Analýza 2	Bakalářka	OTEVŘÍT
3	Analýza 3	Semestrálka	OTEVŘÍT

Obrázek 14 - UC04 – Vyhledat analýzu požadavků

#### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku „Analýza požadavků“

- 1) Uživatel očekává, že po vybrání volby „Vyhledat Analýzu“ se mu zobrazí formulář pro vyhledávání dokumentu analýzy požadavků s možností zadat konkrétní identifikátory pro filtraci.
- 2) Uživatel očekává, že po zadání identifikačních parametrů v levé části formuláře a následném stisknutí tlačítka „Vyhledat“ se mu zobrazí v pravé části obrazovky výsledky s možností prokliku na konkrétní detail analýzy požadavků
- 3) Uživatel očekává, že po stisknutí symbolu „plus“ se mu rozšíří nabídka zadání identifikačních parametrů pro přesnější filtrování analýz požadavků.

## 4.6.5 UC05 – Vytvořit testovací analýzu

The screenshot shows a web application interface for creating a test analysis. The top navigation bar includes 'Logo Společnosti', 'System řízení kvalit software', and icons for home, mail, settings, and user. Below this is a secondary navigation bar with tabs: 'Nástěnka', 'Analýza Požadavků', 'Test Plán', 'Test Analýza' (selected), 'Test Exekuce', and 'Správa Chyb'. A third navigation bar contains 'Vytvořit TA' and 'Vyhledat TA'. The main content area is titled 'Vytvořit Test Analýzu' and contains two columns of form fields. The left column is for 'Název test set' and the right for 'Název test case'. Both columns have a 'Název' field, a 'Projekt' dropdown, and a 'Detailní popis' text area. At the bottom, there are navigation arrows and a 'Uložit a ukončit' button.

Obrázek 15 - UC05 – Vytvořit testovací analýzu

### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku „Test Analýza“

- 1) Uživatel očekává, že po kliknutí na možnost „Vytvořit TA“ se mu zobrazí formulář na vytvoření testovací analýzy.
- 2) Uživatel očekává, že po vyplnění formulářového atributu „Projekt“ bude testovací analýza viditelná pouze pro uživatele, kteří mají přiřazenou roli ve vybraném projektu.
- 3) Uživatel očekává, že po stisknutí symbolu „plus“ se mu zobrazí širší nabídka atributů formuláře.
- 4) Uživatel očekává, že po stisknutí šipky se přesune na další test set nebo test case (zaleží na jakou šipku) a to ve směru zpět nebo dopředu. V případě poslední instance test setu nebo test casu a stisknutí šipky v před se vytvoří nová instance test set nebo test case.
- 5) Uživatel očekává, že po stisknutí tlačítka „Uložit a ukončit“ se vytvoří dokument s testovací analýzou.



## 4.6.6 UC06 – Vytvořit první testovací scénář s testovacími kroky

Popis akce	Očekávaný výsledek	Aktuální výsledek	Poznámka	Akce
Hint text	Hint text	Hint text	Hint text	+
Hint text	Hint text	Hint text	Hint text	- +
Hint text	Hint text	Hint text	Hint text	- +

Obrázek 16 - UC06 – Vytvořit první testovací scénář s testovacími kroky

### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku „Test Analýza“. Je splněn UC05. Uživatel si již vyhledal vytvořenou test analýzu dle UC05.

- 1) Uživatel očekává, že po kliknutí na možnost „Přidat testovací kroky do test case“ dle UC05 se mu zobrazí formulář na přidání testovacích kroků do konkrétního testovacího scénáře.
- 2) Uživatel očekává, že po kliknutí na symbol „plus“ se přidá další řádek s atributy, které odpovídají testovacím krokům. Symbol „mínus“ je ubírá/maže.
- 3) Uživatel očekává, že po kliknutí na šipku se posune na další nebo předešlý testovací scénář. V případě, že je na poslední testovacím scénáři se nový scénář nevytvoří.
- 4) Uživatel očekává, že po kliknutí na tlačítko „uložit“ se testovací kroky uloží a testovací scénáře budou naplněny kroky, které uživatel zadal.

#### 4.6.7 UC07 – Exekuce testovacího scénáře



Obrázek 17 - UC07 – Exekuce testovacího scénáře

#### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku „Test Exekuce“. Uživatel si vyhledal konkrétní testovací scénář a spustil ho tlačítkem „spustit test“

- 1) Uživatel očekává, že po editaci v textových polích „Aktuální výsledek“ a „Poznámka“ se celkový stav testu uloží jako instance s posledním datem a časem exekuce.
- 2) Uživatel očekává, že po editaci rozbalovací položky „stav“ se změní stav kroku ale i samotného scénáře a celkový stav se uloží jako instance s posledním datem a časem exekuce.
- 3) Uživatel očekává, že při kliknutí na šipku se posune na další nebo předchozí testovací scénář. V exekuci testů není možné editovat, takže při výběru šipky vpřed, kdy se uživatel nachází na posledním testovacím scénáři se nevytváří nový testovací scénář a ani se nepřechází na testovací scénář následujícího testovacího setu.
- 4) Uživatel očekává, že při kliknutí na tlačítko „Zadat chybu“ se mu zobrazí předvyplněný formulář pro zadání chyby ke scénáři, kde byla akce vyvolána.

## 4.6.8 UC08 – Zadat chybu

The screenshot shows a web application interface for reporting a bug. The top navigation bar includes a logo, the title 'Systém řízení kvality software', and icons for home, mail, settings, and user. Below this is a secondary navigation bar with tabs: 'Nastěnka', 'Analýza Požadavků', 'Test Plán', 'Test Analýza', 'Test Exekuce', and 'Správa Chyb'. The 'Zadat chybu' form is the active view, featuring a title bar with a close button. The form fields are organized into two columns. The left column contains: 'Název chyby' (text input), 'Severita' (dropdown), 'Projekt' (dropdown), 'Sprint' (dropdown), 'Charakter' (dropdown), 'ID Testovacího scénáře' (text input with hint), 'ID Testovacího kroku' (text input with hint), 'Aplikace' (text input with hint), and 'Komponenta' (text input with hint). The right column contains: 'Detailní popis chyby' (text area with hint), 'Vlastník' (dropdown), and 'Řešitel' (dropdown). A plus sign icon is located at the bottom left of the form area, and a 'Uložit a ukončit' button is at the bottom right.

Obrázek 18 - UC08 – Zadat chybu

### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku „Správa chyb“. Uživatel kliknul na možnost „Zadat chybu“ a zobrazil se mu formulář pro zadání chyby.

- 1) Uživatel očekává, že po zadání ID testovacího scénáře se mu automaticky doplní ID testovacího kroku, pokud tento scénář již byl spuštěn a na daný krok je označen v poslední exekované instanci stavem „failed“.
- 2) Uživatel očekává, že po stisknutí symbolu „plus“ se mu zobrazí nabídka pro zadání dalších parametrů, které mohou napomoci lepší identifikaci chyby. Například přidání souborů.
- 3) Uživatel očekává, že pokud nevyplní pole „Vlastník“ a „Řešitel“ a stiskne tlačítko „Uložit a ukončit“, tak systém automaticky tyto pole vyplní údaji zadávajícím uživatelem. Zároveň uloží chybu a zavře formulář.

## 4.6.9 UC09 – Vyhledat chybu

Logo Společnosti

System řízení kvality software

Nástěnka | Analýza Požadavků | Test Plán | Test Analýza | Test Exekuce | Správa Chyb

Zadat chybu | Vyhledat chybu

**Vyhledat chybu**

**Filtr pro vyhledávání**

ID: Zadej ID chyby  
Název: Zadejte název chyby  
Projekt: Vyberte kritérium pro projekt...  
Sprint: Vyberte sprint...  
Severita: Vyberte severitu...  
Charakter: Vyberte charakter...

**Výsledky filtrování**

ID	Název	Projekt	Akce
1	Chyba 1	Diplomka	OTEVŘÍT
2	Chyba 2	Bakalářka	OTEVŘÍT
3	Chyba 3	Semestrálka	OTEVŘÍT

Cheete přidat další možnosti filtrování? +

Vyhledat

Obrázek 19 - UC09 – Vyhledat chybu

### Usecase

Předpoklad: Uživatel je přihlášen a má zobrazenou obrazovku „Správa chyb“. Uživatel kliknul na možnost „Vyhledat chybu“ a zobrazil se mu formulář pro Vyhledávání chyb.

- 1) Uživatel očekává, že po vyplnění parametrů v levé části obrazovky bude možné stisknout tlačítko „Vyhledat“.
- 2) Uživatel očekává, že po stisknutí tlačítka „Vyhledat“ systém zobrazí množinu výsledků odpovídající filtrovacím parametrům, které uživatel zadal v levé části obrazovky.
- 3) Uživatel očekává, že po stisknutí symbolu „plus“ systém zobrazí nabídku dalších parametrů, které mohou upřesnit vyhledávací algoritmus a zobrazit tedy užší množinu vyhledávaných výsledků.
- 4) Uživatel očekává, že u množiny vyhledávaných výsledků bude u každého výsledku možné kliknout na možnost „Otevřít“, která uživateli zobrazí konkrétní instanci již založené chyby.

## **4.7 Hlavní scénáře k vybraným usecase**

### **4.7.1 SC01 – Vytvořit nástěnku**

#### **Scénář**

Předpoklad: Systém již uživatele autorizoval a čeká na reakci uživatele na hlavní stránce

- 1) Systém čeká až uživatel zvolí možnost „Vytvořit Nástěnku“ a poté, zobrazí formulář pro vytvoření nástěnky
- 2) Systém čeká na výběr uživatelských parametrů a podle toho bude dotahovat do nástěnky data dle zvolených atributů ve formuláři.
- 3) Systém čeká až uživatel zvolí možnost „Vytvořit“ a poté vytvoří nástěnku vyplněnou daty (pokud již existují), dle zvolených atributů ve formuláři pro vytvoření nástěnky.

### **4.7.2 SC02 – Vytvořit Analýzu požadavků**

#### **Scénář**

Předpoklad: Systém již uživatele autorizoval a čeká na reakci uživatele na obrazovce

„Analýza Požadavků“

- 1) Systém čeká až uživatel zvolí volbu „Vytvořit Analýzu“ a zobrazí formulář pro vytvoření analýzy požadavků s možnostmi vyplnit údaje a požadavcích včetně jejich konkrétních řešení.
- 2) Systém čeká až uživatel klikne na šipku v dolní pravé části obrazovky. Dle zvolené šipky (pravá nebo levá) systém zobrazí buď předchozí nebo následující požadavek. V případě, že je uživatel na posledním požadavku a klikne na šipku vpravo, tak systém vytvoří nový požadavek.
- 3) Systém čeká na stisknutí tlačítka „Uložit a ukončit“ se vytvoří dokument „Analýza požadavků“ s konkrétně vyplněnými požadavky dle vyplněného formuláře.

### **4.7.3 SC03 – Naplánovat testovací scénáře**

#### **Scénář**

Předpoklad: Systém uživatele autorizoval a uživatel se nachází na obrazovce „Test Plán“. Zároveň počítá již s existencí vytvořeného test plánu se sprinty a vytvoření testovací sady naplněné testovacími scénáři pro nadcházející testování.

- 1) Systém čeká až uživatel zvolí volbu naplánovat test a zobrazí mu formulář s možnostmi plánovat testy.
- 2) Systém při uživatelově výběru testovací sady a zaškrtnutí volby „Všechny scénáře z tohoto setu“ automaticky zvolí všechny scénáře z konkrétně vybrané testovací sady.
- 3) Jestliže uživatel kliknutím zvolí symbol „plus“, tak systém rozšíří formulář o další atributy. V případě symbolu „mínus“ mu formulář zkrátí.
- 4) Systém čeká, až uživatel stiskne tlačítko „Uložit a ukončit“. Systém uloží vyplněné nastavení

### **4.7.4 SC 04 – Vyhledat analýzu požadavků**

#### **Scénář**

Předpoklad: Systém uživatele autorizoval a uživatel se nachází na obrazovce „Analýza Požadavků“.

- 1) Systém čeká vybrání volby „Vyhledat Analýzu“. Po jejím vybrání zobrazí formulář pro vyhledávání dokumentu analýzy požadavků s možností zadat konkrétních identifikátorů pro filtraci.
- 2) Systém čeká na stisknutí tlačítka „Vyhledat“. Poté zobrazí v pravé části obrazovky výsledky dle zadaných vstupních identifikátorů s možností prokliku na detail konkrétní analýzy požadavků.
- 3) Systém čeká na stisknutí symbolu „plus“ aby mohl zobrazit širší nabídku pro zadávání identifikačních parametrů pro přesnější filtrování mezi analýzami požadavků.

### **4.7.5 SC05 – Vytvořit testovací analýzu**

#### **Scénář**

Předpoklad: Systém uživatele autorizoval a uživatel se nachází na obrazovce „Testovací Analýza“.

- 1) Systém zobrazí formulář na vytvoření testovací analýz po kliknutí na možnost „Vytvořit TA“.
- 2) Po vyplnění formulářového atributu „Projekt“ Systém zobrazí analýzu pouze uživatelům s rolí přiřazenou na zvolený projekt.
- 3) Systém zobrazí širší nabídka atributů formuláře po stisknutí symbolu „plus“.
- 4) Systém po stisknutí šipky zobrazí další test set nebo test case, a to ve směru zpět nebo dopředu (podle zvolené šipky). V případě poslední instance test setu nebo test casu a stisknutí šipky v před se vytvoří nová instance test set nebo test case.
- 5) Po stisknutí tlačítka „Uložit a ukončit“ systém vytvoří s testovací analýzu.

#### **4.7.6 SC06 – Vytvořit první testovací scénář s testovacími kroky**

##### **Scénář**

Předpoklad: Systém uživatele autorizoval a uživatel se nachází na obrazovce „Test Analýza“. Je splněn UC05. Systém si již zobrazil vytvořenou test analýzu dle UC05.

- 1) Systém čeká na zvolení možnosti „Přidat testovací kroky do test case“ dle UC05 s zobrazí formulář na přidání testovacích kroků do již zobrazeného testovacího scénáře.
- 2) Systém po kliknutí na symbol „plus“ přidá další řádek s atributy, které odpovídají testovacím krokům. Symbolem „mínus“ je ubírá/maže.
- 3) Systém po kliknutí na šipku zobrazí další nebo předešlý testovací scénář. V případě, je zobrazen poslední testovacím scénář, tak nový scénář nevytvoří.
- 4) Systém po kliknutí na tlačítko „uložit“ uloží testovací kroky a testovací scénáře budou naplněny kroky, které uživatel zadal.

#### **4.7.7 SC07 – Exekuce testovacího scénáře**

##### **Scénář**

Předpoklad: Systém uživatele autorizoval a uživatel se nachází na obrazovce „Test Exekuce“.

Systém zobrazil konkrétní testovací kroky a kliknutím na tlačítko „spustit test“

- 1) Systém po editaci v textových polích „Aktuální výsledek“ a „Poznámka“ se celkový stav testu uloží jako instance s posledním datumem a časem exekuce.
- 2) Systém po editaci rozbalovací položky „stav“ se změní stav kroku ale i samotného scénáře a celkový stav se uloží jako instance s posledním datumem a časem exekuce.
- 3) Systém po kliknutí na šipku zobrazí další nebo předchozí testovací scénář. V exekuci testů není možné editovat, takže při výběru šipky vpřed, kdy je zobrazen poslední testovací scénář se nevytváří nový testovací scénář a ani se nepřechází na testovací scénář následujícího testovacího setu.
- 4) Systém po kliknutí na tlačítko „Zadat chybu“ zobrazí předvyplněný formulář pro zadání chyby ke scénáři, kde byla akce vyvolána.

#### **4.7.8 SC08 – Zadat chybu**

##### **Scénář**

Předpoklad: Systém uživatele autorizoval a uživatel se nachází na obrazovce „Správa chyb“.

Systém zobrazuje možnost „Zadat chybu“ s formulářem pro zadání chyb.

- 1) Systém po zadání ID testovacího scénáře automaticky doplní ID testovacího kroku, pokud tento scénář již byl spuštěn a na daný krok je označen v poslední exekované instanci stavem „failed“.
- 2) Systém po stisknutí symbolu „plus“ zobrazí nabídku pro zadání dalších parametrů, které mohou napomoci lepší identifikaci chyby. Například přidání souborů.
- 3) Systém při nevyplnění pole „Vlastník“ a „Řešitel“ a stiskem na tlačítko „Uložit a ukončit“, automaticky tyto dvě pole vyplní údaji zadávajícím uživatelem. Zároveň uloží chybu a uzavře formulář.



#### 4.7.9 SC09 – Vyhledat chybu

##### Scénář

Předpoklad: Systém uživatele autorizoval a uživatel se nachází na obrazovce „Správa chyb“. Systém po kliknutí na možnost „Vyhledat chybu“ a zobrazil formulář pro vyhledávání chyb.

- 1) Systém po vyplnění parametrů v levé části obrazovky zaktivuje tlačítko „Vyhledat“. Než uživatel vyplní alespoň jeden parametr pro vyhledávání, tak tlačítko bude neaktivní.
- 2) Systém po stisknutí tlačítka „Vyhledat“ zobrazí množinu výsledků odpovídající filtrovacím parametrům, které uživatel zadal v levé části obrazovky.
- 3) Systém po stisknutí symbolu „plus“ zobrazí nabídku dalších parametrů, které mohou upřesnit vyhledávací algoritmus a zobrazit tedy užší množinu vyhledávaných výsledků.
- 4) Systém u množiny vyhledávaných výsledků zobrazí u každého výsledku možnost kliknout na aktivní pole „Otevřít“, které uživateli automaticky zobrazí konkrétní instanci již založené chyby.

## 4.8 Kvalitativní šetření

Tato kapitola se věnuje definici cílů pro testované subjekty, vyhodnocení informací nasbíraných při uživatelské testování použitelnosti a agregaci výsledků pro vizuální zobrazení nashromážděných dat a jejich interpretaci.

Je předpokládáno, že systém bude intuitivní ale bude obsahovat lehké vady. Jde o nástroj, který je navrhován pro určitou skupinu uživatelů a jelikož testování softwaru má několik metodik pro výkon práce, tak ne všichni se drží té jedné a té samé. Proto je předpokladem, že z výsledků vyjdou i takové poznatky, které budou v rozporu s již nasbíranými informacemi, dle kterých byl prototyp systému navržen.

### 4.8.1 Výběr testovacích cílů

V rámci kvalitativního šetření je potřebné definovat testované cíle, které se participanti budou snažit splnit. Cíle jsem vybíral dle definovaných usecase pro každý typ osoby zvlášť. Přesněji řečeno byly vybrány 3 cíle pro každou osobu, což odpovídá součtinu 9.

#### **Cíle pro participanty osoby test manažer**

- 1) Vytvořit nástěnku projektu se sloupcovým grafem s daty o testovacích scénářích
- 2) Vytvořit analýzu požadavků a korektně vyplnit první požadavek jako nefunkční
- 3) Naplánovat testovací scénáře pro již existující test plán

#### **Cíle pro participanty osoby test analytik**

- 1) Vyhledat analýzu požadavků pro projekt „Diplomka“
- 2) Vytvořit testovací analýzu
- 3) Vytvořit první testovací scénář s testovacími kroky

#### **Cíle pro participanty osoby tester**

- 1) Spustit testovací scénář a označit první krok stavem „failed“
- 2) Zadat chybu k testovacímu scénáři
- 3) Vyhledat chybu Dle ID a názvu chyby

## 4.9 Finanční analýza – Návrh systému v praxi

Pomocí nástroje MS Project jsou naplánovány akce nutné pro vytvoření finanční analýzy. Finanční analýza se zaměřuje na plánování procesu studie použitelnosti, pracovních pozic

Údaje zpracované v tabulce níže představují reálná ohodnocení pracovních pozic. Výběr dat je proveden ze získaných informací na portálech jobs.cz a práce.cz, bylo vybráno 5 nejčtetnějších hodnot, ze kterých byla vypočtena průměrná měsíční hrubá mzda. Dále jsem data jsou rozpočítána na jednotlivé hodinové sazby, které jsou použity pro plánování nákladů v nástroji MS Project.

Pracovní pozice	Nabízené platy za měsíc v tis. Kč					Měsíční sazba Kč	Denní Sazba Kč	Hodinová Sazba Kč
Projektový manažer	40	50	60	45	65	52000	2600	325
IT Analytik	60	40	60	40	50	50000	2500	312,5
UI Designer	45	55	50	40	40	46000	2300	287,5
Business Analytik	50	35	60	55	40	48000	2400	300

Tabulka 3 - Rozbor mzdových sazeb

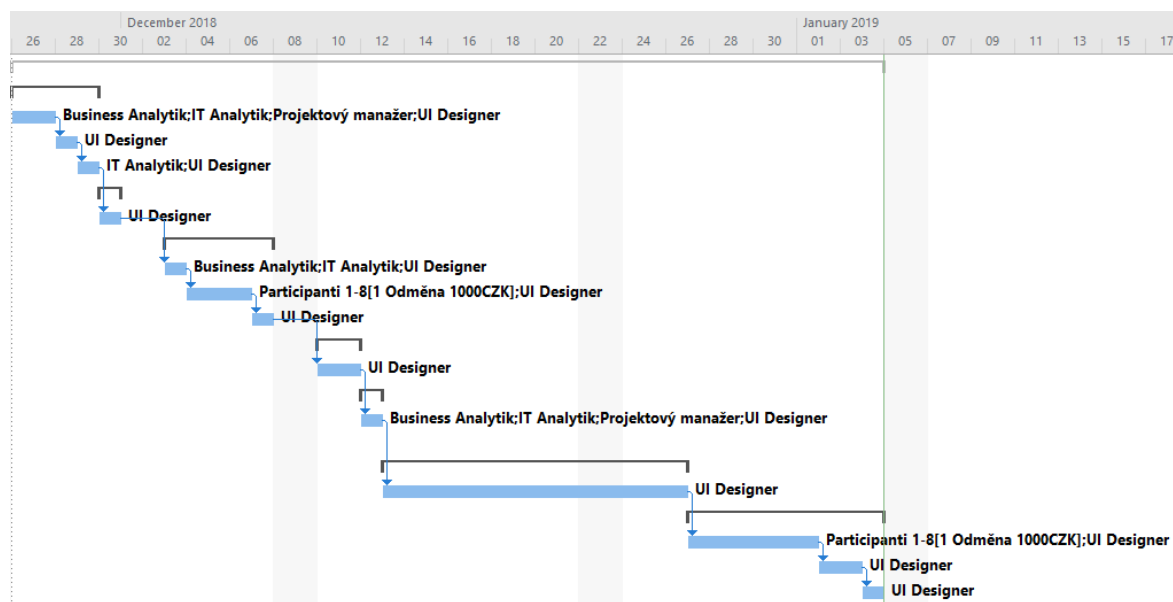
Název Zdroje	Typ Zdroje	Poznámka	Standartní Sazba
Projektový manažer	Práce		325 Kč/hr
IT Analytik	Práce		313 Kč/hr
UI Designer	Práce		288 Kč/hr
Business Analytik	Práce		300 Kč/hr
Participant 1-8	Materiál	Odměna 1000CZK	8 000 Kč

Tabulka 4 - Tabulka zdrojů pro finanční analýzu

Projekt je naplánován s reálnými odhady na 30 pracovních dní. V těchto pracovních dnech jsou zahrnuty i náklady na participanty, které činní 1000czk na jednoho. Celkové náklady činní 131 020 Kč včetně zahrnutých 16 000 Kč na participanty.

Název úkolu	Doba trvání	Start	Konec
<b>Studie použitelnosti v praxi</b>	<b>30 days</b>	<b>Mon 26.11.18</b>	<b>Fri 04.01.19</b>
<b>Rozhovor s účastníky projektu</b>	<b>4 days</b>	<b>Mon 26.11.18</b>	<b>Thu 29.11.18</b>
Schůze o předmětu projektu	2 days	Mon 26.11.18	Tue 27.11.18
Zběžná analýza požadovaného systému	1 day	Wed 28.11.18	Wed 28.11.18
Vyhledání podobných řešení	1 day	Thu 29.11.18	Thu 29.11.18
<b>Uživatelský výzkum</b>	<b>1 day</b>	<b>Fri 30.11.18</b>	<b>Fri 30.11.18</b>
Definování person	1 day	Fri 30.11.18	Fri 30.11.18
<b>Získání dat pro vytvoření prototypu</b>	<b>5 days</b>	<b>Mon 03.12.18</b>	<b>Fri 07.12.18</b>
Zajištění podrobnějších požadavků na systém	1 day	Mon 03.12.18	Mon 03.12.18
Rozhovor s potenciálními uživateli – získání dat	3 days	Tue 04.12.18	Thu 06.12.18
Analýza nashromážděných dat	1 day	Fri 07.12.18	Fri 07.12.18
<b>První náhled na systém</b>	<b>2 days</b>	<b>Mon 10.12.18</b>	<b>Tue 11.12.18</b>
Vytvoření papírového modelu	2 days	Mon 10.12.18	Tue 11.12.18
<b>Konzultace prvních náhledů</b>	<b>1 day</b>	<b>Wed 12.12.18</b>	<b>Wed 12.12.18</b>
Konzultace návaznosti obrazovek, dle papírového modelu	1 day	Wed 12.12.18	Wed 12.12.18
<b>Tvorba funkčního prototypu</b>	<b>10 days</b>	<b>Thu 13.12.18</b>	<b>Wed 26.12.18</b>
Tvorba funkčního prototypu	10 days	Thu 13.12.18	Wed 26.12.18
<b>Testování použitelnosti s uživateli</b>	<b>7 days</b>	<b>Thu 27.12.18</b>	<b>Fri 04.01.19</b>
Testování s participanty	4 days	Thu 27.12.18	Tue 01.01.19
Vyhodnocení výsledků	2 days	Wed 02.01.19	Thu 03.01.19
Doporučení pro realizaci / úpravu	1 day	Fri 04.01.19	Fri 04.01.19

**Tabulka 5 - Plánování jednotlivých iterací studie použitelnosti v praxi**



**Obrázek 20 - Ganttův diagram studie použitelnosti v praxi**

## **5 Výsledky a diskuse**

### **5.1 Očekávané výsledky**

Vzhledem k tomu, že prototyp je navrhovaný pro konkrétní skupinu uživatelů, kterou spojují zkušenosti z oboru testování software a zároveň je navrhován na podkladech, které byly zjištěny na základě řízených rozhovorů s odborníky z praxe, tak očekáváno, že kvalitativní šetření přinese následující výsledky:

#### **Intuitivnost**

Navrhovaný systém bude dostatečně intuitivní i pro začínající. Dále je předpokládáno, že v kvalitativním šetření budou zaznamenána lehká zaváhání, a to zejména u testovacích subjektů, kteří představují testery.

#### **Úspěšnost**

U úspěšnosti cílů je odhad poněkud skeptický. Každý cíl je splnitelný, avšak ne všichni mohou být stejného přesvědčení jako já. Je očekáváno, že participantů budou mít lehká zaváhání nebo dostatečně nesplní cíl, který bude po nich požadován.

#### **Čas**

Předpokladem je, že čas bude vždy rychlý. Aplikace je navrhována tak, aby byla co nejjednodušší pracující subjekt s ní mohl rychle a jednoduše interagovat.

#### **Překlikatelnost**

U překlikatelnosti nebyl odhad stanoven. Porovnání by mělo vycházet ze zkušeností participantů s ostatními systémy.

#### **Naučitelnost**

Aplikace bude jednoduše naučitelná. Opačný výsledek by byl překvapivý.

## 5.2 Rekapitulace hodnocení kvalitativního šetření

Aplikace byla rozdělena pro kvalitativní šetření na 3 části, a to dle modulů, které nejlépe odpovídají činnostem person. Každý participant byl zařazen do kategorie dle jeho nejlepších zkušeností a testoval pouze 3 vybrané cíle kategorie, do které nejlépe zapadal.

Výběr kategorie je proveden dle aktuálních pracovních zařazení participantů a dle subjektivních pocitů, každého z participantů.

Dále je potřeba zdůraznit bodová ohodnocení:

### **Minima:**

- Minimální dosažitelný počet bodů charakteristiky za blok testovaných subjektů je 9 (3 subjekty \* 3 cíle \* 1 min bod).
- Minimální dosažitelný počet bodů za charakteristiku celkem je 27 (9 subjektů \* 3 cíle \* 1 bod)
- Maximální dosažitelný počet bodů za systém 135 (27\* 5 charakteristik)

### **Maxima:**

- Maximálně dosažitelný počet bodů charakteristiky za blok testovaných subjektů je 27 (3 subjekty \* 3 cíle \* 3 max body).
- Maximální dosažitelný počet bodů za charakteristiku celkem je 81 (9 subjektů \* 3 cíle \* 3 body)
- Maximální dosažitelný počet bodů za systém 405 (81\* 5 charakteristik)

### 5.3 Celkové výsledky kvalitativního šetření

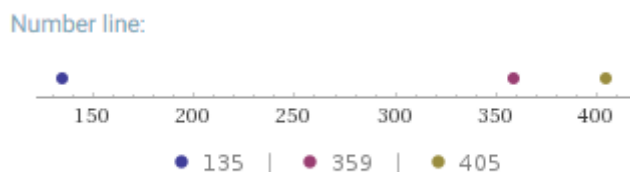
V tabulce níže jsou zapsány celkové výsledky převedené pomocí bodovací metody z naměřených či jinak získaných dat od testovaných subjektů na body v již dříve definovaném rozsahu. Body představují součty celkově získaných ohodnocení za jednotlivé osoby.

Charakteristika	Tester Součet	Analytik Součet	Manažer Součet	Celkový Součet
Intuitivnost	24	22	24	<b>70</b>
Úspěšnost	25	24	23	<b>72</b>
Čas	25	23	25	<b>73</b>
Překlikatelnost	25	22	18	<b>65</b>
Naučitelnost	27	27	25	<b>79</b>
<b>Suma bodů</b>	<b>126</b>	<b>118</b>	<b>115</b>	<b>359</b>

Tabulka 6 - Sumarizace celkových bodů kvalitativního šetření

Z výsledků v tabulce lze vyčíst například celkové hodnocení za systém jako takový, které nás v této podkapitole výsledků zajímá nejvíce. Ostatní data budou interpretována v pozdějších kapitolách kvůli analytickému rozdělení.

Jak jsme si připomněli v předešlé kapitole, tak minimum celkového hodnocení systému je 135 a maximum 405. Do tohoto uzavřeného intervalu položím naměřenou hodnotu 359.



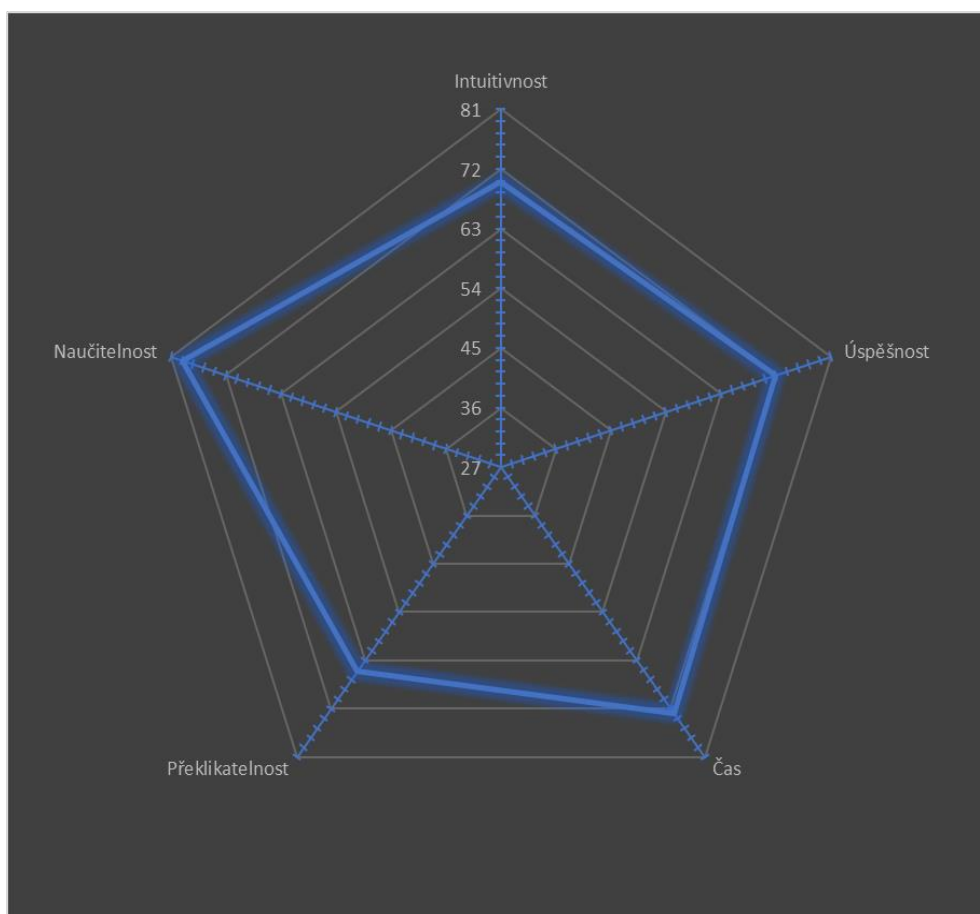
Obrázek 21 - Interval celkového ohodnocení systému

Z tohoto vizualizovaného intervalu lze vyčíst, že navrhovaný systém se blíží 100% splnění cíle použitelnosti. Splnění cíle dopadlo na 88.89%, což lze považovat za úspěch.

## 5.4 Analytické rozpracování dílčích výsledků

Jelikož výsledky dopadly téměř se 100% úspěšností, což je předpokládáno, tak je potřené se podrobněji podívat na nashromážděná data a provést analýzu, která odhalí příčiny či nedostatky navrhovaného systému.

Pro tuto analýzu byly použity paprskový nebo také jinak nazývaný pavučinový graf, který výborně vizualizuje měřené charakteristiky navrhovaného systému



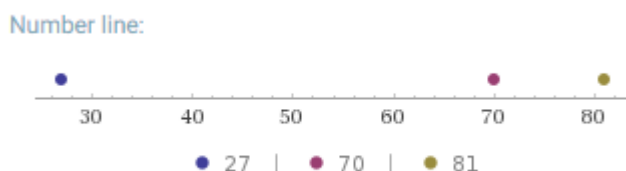
Obrázek 22 - Pavučinový graf celkového zhodnocení systému

Jak lze na grafu pozorovat, tak charakteristika Naučitelnost dopadla s 96,3% hodnocením, a tudíž jí dále nebudeme analyticky rozpracovávat. Za to ostatní charakteristiky dopadly o něco hůř a budou předmětem dalšího zkoumání.



### 5.4.1 Analytický rozpad charakteristiky Intuitivnost

Ve výsledcích je zaznamenáno 70 bodů z 81 možných a 27 minimálních. To znamená, že Intuitivnost nabyla hodnoty 79,63%.



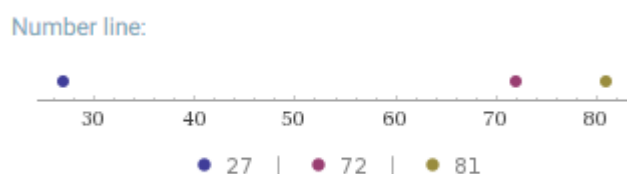
**Obrázek 23 - Interval analytického rozpadu charakteristiky Intuitivnost**

Tato čísla výrazně nesnižují hodnotu celkového výsledku a je možné říci, že aplikace se testovaným subjektům se zdá být intuitivní.

Procentuální hodnotu nejvíce snížily výsledky Analytiků ve třetím cíli. Cílem bylo vytvořit testovací scénář s vyplněnými kroky. Je patrné, že tento cíl nemusí být dostatečně intuitivní a bude doporučen na drobné úpravy.

### 5.4.2 Analytický rozpad charakteristiky Úspěšnost

Ve výsledcích je zaznamenáno 72 bodů z 81 možných a 27 minimálních. To znamená, že Úspěšnost nabyla hodnoty 83,33%.



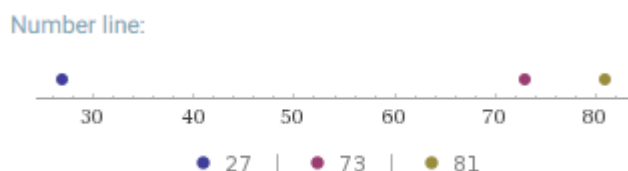
**Obrázek 24 - Interval analytického rozpadu charakteristiky Úspěšnost**

Tato čísla výrazně nesnižují hodnotu celkového výsledku a je možné říci, že pro testovací subjekty bylo převážně jednoduché cíle splnit.

Procentuální hodnotu nejvíce snížily výsledky Analytiků opět ve třetím cíli. Cílem bylo vytvořit testovací scénář s vyplněnými kroky. Je patrné, že tento cíl nemusí být dostatečně intuitivní a bude doporučen na drobné úpravy.

### 5.4.3 Analytický rozpad charakteristiky Čas

Ve výsledcích je zaznamenáno 73 bodů z 81 možných a 27 minimálních. To znamená, že Čas nabyl hodnoty 85,19%.



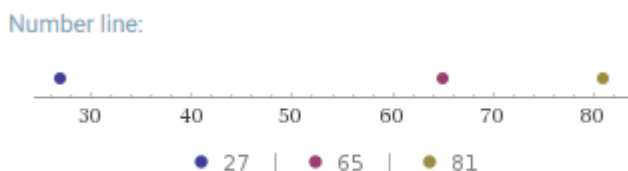
**Obrázek 25 - Interval analytického rozpadu charakteristiky Čas**

Tato čísla výrazně nesnižují hodnotu celkového výsledku a je možné říci, že pro testovací subjekty bylo převážně jednoduché cíle splnit v rychlém čase.

Procentuální hodnotu nejvíce snížily výsledky Analytiků opět ve třetím cíli. Cílem bylo vytvořit testovací scénář s vyplněnými kroky. Je patrné, že tento cíl nemusí být dostatečně intuitivní a bude doporučen na drobné úpravy.

### 5.4.4 Analytický rozpad charakteristiky Překlikatelnost

Ve výsledcích je zaznamenáno 65 bodů z 81 možných a 27 minimálních. To znamená, že Překlikatelnost nabyla hodnoty 70,37%.



**Obrázek 26 - Interval analytického rozpadu charakteristiky Překlikatelnost**

Čísla snižují hodnotu celkového výsledku a je možné říci, že pro testovací subjekty se navrhovaný systém může zdát překlíkaný.

Procentuální hodnotu nejvíce snížily výsledky Manažerů ve třetím cíli. Cílem bylo naplánovat již existující testovací scénáře pro již existující test plán. V tomto případě jsem si již při samotném testování všiml nejistoty participantů. Prvky, které se týkají třetího cíle manažerů budou určitě doporučeny na přepracování a opětné testování.

## 5.5 Poznatky získané od participantů

Během toho, co participantů plnily své úkoly, zmiňovaly své poznatky, které byly zapisovány a po dokončení cílů s nimi konzultovány. Po těchto konzultacích přišla na řadu poslední část, ve které byl procházen celý navržený systém a veden řízený rozhovor.

Některé záporné výsledky byly zapisovány přímo v průběhu testování. Záporné výsledky, u kterých si například participant delší dobu nebyl schopen poradit s cílem nebo obtížně.

*Poznámka: Některé výsledky se opakovaly, a proto jsou níže zmíněny pouze jednou.*

### **Participantů jako výhody / kladné / vhodné stránky systému uvedli:**

- a. Uživatelsky přijatelné. Jednoduché a nepřelácané funkcemi.
- b. Přehlednost prvků. Vše, co člověk hledá vidí na první pohled.
- c. Rozpis testů v exekuci.
- d. Rozdělení prvků kaskádově.
- e. Samotný průběh testování byl zábavný.
- f. Jednoduchost ovládání a snadné hledání.
- g. Kaskádový styl, na který je zvyklý.
- h. Snadno naučitelné a rychlé.

### **Participantů jako nevýhody / záporné / nevhodné stránky systému uvedli:**

- a. Tlačítko "přidat testy" není dostatečně viditelné.
- b. Formuláře obsahují hodně informací.
- c. Tlačítko "přidat kroky" je matoucí.
- d. Modul Analýza požadavků nebyl dostatečně srozumitelný. Očekával tento modul v Test Analýze.
- e. Tlačítka by měla představovat intuitivní symboly místo textu. Uživatel nechce číst.

**Participanti doporučili:**

- a. Přecházet do ostatních projektů pomocí nástěnky.
- b. Pohledy – tester vidí pomocí role jen to co má vidět.
- c. Modul pro vývojáře.
- d. Když u testovacího scénáře zadám stav "failed", tak by bylo dobré, aby vyskočil combo box s již reportovanými chybami ke scénáři.
- e. Zatím, neexistuje UC pro vybrání projektu.
- f. Označit povinná pole.
- g. Přidat pole, kam se budou dát nahrát přílohy (soubory atd.).
- h. Možnost exportovat testovací kroky při zakládání chyby.
- i. Modul Výběr exekuce – test sety například vložit do záložek. Bylo by vhodné jiné uspořádání.
- j. Přidal modul "Rizika" do Test Analýzy a Analýzy Požadavků.
- k. Naplánovat testy podle priorit.
- l. Legenda k nástěnce.

## 6 Závěr

Hlavním cílem této diplomové práce bylo navržení UI specifikaci systému pro řízení kvality software. Dílčími cíli, které bylo potřeba zrealizovat pro dosažení hlavního cíle bylo navržení vhodných testovacích scénářů, vyhodnocení kvalitativního šetření a sepsání závěru a doporučení k realizaci.

Aby bylo možné začít s navrhováním jednotlivých funkčních prvků, tak bylo potřebné stanovit svou uživatelskou skupinu, která představuje potenciální uživatele aplikace. Dosáhnu toho bylo tak, že jsem byla vydefinována skupiny, kterou odborně nazýváme persony. Tito fiktivní potenciální uživatelé odpovídají jednotlivým uživatelským rolím, které by měli systém používat. Na základě těchto person byla vybrána skupina respondentů, kteří mi na základě řízeného rozhovoru podali výchozí informace pro navržení základních funkčních prvků systému.

Byla vytvořena základní kostra prototypu systému na základě sesbíraných informací jsem, byla obohacena o osobní poznatky nabyté z literatury, internetových zdrojů a z přednášek a cvičení, absolvována v rámci výuky na ČZU. Tímto je splněn hlavní cíl práce, který bylo následně potřebné kvalitativně otestovat.

Předtím, než bylo možné začít testovat navržený prototyp systému, bylo potřebné zjistit, jak takové testování správně provést, a především s kým. Opět jsem bylo využito již definovaných person a na základě jich byly vybráni participanti, kteří se účastnili kvalitativního šetření.

Dalším předpokladem pro zahájení kvalitativního šetření bylo zajištění testovacích scénářů. Testovací scénáře jsou definovány formou cílů, tedy dle cognitivního stylu, kterých mají participanti dosáhnout. Výběr cílů je založen, na již prvním řízeném rozhovoru, který definoval základní funkční prvky prototypu.

Participanti byly rozděleni do tří skupin pro práci na kvalitativním šetření, dle charakteristik, které jsou definovány v personifikační části práce.

Aby bylo možné sesbírat data z výsledků kvalitativních šetření s participanty, je potřebné stanovit jednotlivé charakteristiky studie použitelnosti, které jsou definovány v rešeršní části. Těmito charakteristikami jsou čas, intuitivnost, naučitelnost, překlikatelnost a úspěšnost. Při schůzkách s participanty, které se týkaly kvalitativního šetření byly získány zpětné vazby, ve

kterých se participanti vyjadřovali velmi kladně k průběhu kvalitativního šetření, ze kterého odcházeli šťastni a s dobrým pocitem, že napomohli k vývoji navrženého systému.

Po sběru výsledků a dat z kvalitativního šetření bylo potřebné provést analýzu, která je dílčím cílem práce a hlavním krokem kvalitativního šetření. Data jsou zpracována do tabulky, ze které byla zformována vizualizace, pomocí paprskového nebo také jinak známého pavučinového grafu. Již na první pohled bylo zřejmé, že navrhovaný systém si ve studii použitelnosti vede velmi dobře a bude tedy použitelný. S takovýmto výsledkem ale nebylo možné se spokojit, a proto bylo rozhodnuto, že výsledky se musí analyzovat více do hloubky.

V detailních analýzách byla zkoumána a hodnocena procentuální hodnota zkoumaných charakteristik z pohledu na celkový systém.

Procentuální ohodnocení charakteristik z pohledu na celkový návrh systému:

- 1) Intuitivnost – 79,63%
- 2) Úspěšnost – 83,33%
- 3) Čas – 85,19%
- 4) Překlikatelnost – 70,37%
- 5) Naučitelnost – 96,3%

Jelikož výsledky dopadly převážně lépe, než bylo očekáváno, tak všechny charakteristiky kromě naučitelnosti byly zkoumány podrobněji. Z naučitelnosti je totiž zřejmé, že participanti neměli problém se s navrhovaným systémem naučit pracovat. Ostatní charakteristiky jsou rozpracovány v kapitole Analytické rozpracovány do dílčích výsledků.

Na základě rozpracovaných analýz výsledků kvalitativního šetření a získaných poznatků od participantů (klady, zápory, doporučení), které jsou uvedeny v kapitole „Poznatky získané od participantů“, bylo usouzeno, že prototyp systému je vhodný pro budoucí realizaci. Tímto konečným výsledkem je splněn poslední cíl práce.

### **Doporučení pro potenciální realizátory systému**

Pokud, by se někdo kromě mě pokoušel práci realizovat, je mu doporučeno si projít podrobné poznatky od participantů, udělat drobné úpravy, které vychází z již několikrát zmíněných podrobných analýz a následně znovu realizovat kvalitativní šetření. To by mělo realizátorům projektu dát dostatečné informace k porovnání nově přepracovaných funkčních prvků s původním, které byly navrženy v této diplomové práci.

## 7 Seznam použitých zdrojů

- 6) M. CARROLL, John. *What is Human-Computer Interaction (HCI)?* [online]. [cit. 2018-09-09]. Dostupné z: [www.interaction-design.org/literature/topics/human-computer-interaction](http://www.interaction-design.org/literature/topics/human-computer-interaction)
- 7) ŠTĚPÁNKOVÁ, Zuzana Alexandra. *INTERAKCE ČLOVĚK – POČÍTAČ (HCI)* [online]. [cit. 2018-09-09]. Dostupné z: [www.slideplayer.cz/slide/12745329](http://www.slideplayer.cz/slide/12745329)
- 8) PAVLÍČEK, Josef. *INTERAKCE ČLOVĚK A POČÍTAČ 1* [online]. [cit. 2018-09-09].
- 9) NORMAN, Don. *What is User Experience (UX) Design?* [online]. [cit. 2018-09-10]. Dostupné z: [www.interaction-design.org/literature/topics/ux-design](http://www.interaction-design.org/literature/topics/ux-design)
- 10) M. CARROLL, John. *A Brief History of Human-Computer Interaction* [online]. [cit. 2018-09-10]. Dostupné z: [www.interaction-design.org/literature/topics/human-computer-interaction](http://www.interaction-design.org/literature/topics/human-computer-interaction)
- 11) *End User* [online]. [cit. 2018-09-10]. Dostupné z: [www.techterms.com/definition/enduser](http://www.techterms.com/definition/enduser)
- 12) *Uživatel* [online]. [cit. 2018-09-10]. Dostupné z: [www.it-slovník.cz/pojem/uzivatel](http://www.it-slovník.cz/pojem/uzivatel)
- 13) ROUSE, Margaret. *User Interface (UI)* [online]. [cit. 2018-09-10]. Dostupné z: [www.searchmicroservices.techtarget.com/definition/user-interface-UI](http://www.searchmicroservices.techtarget.com/definition/user-interface-UI)
- 14) FROHN, Jessica. *UI Specification Document* [online]. [cit. 2018-09-15]. Dostupné z: [www.behance.net/gallery/36283417/Example-of-UI-Specification-Document-Project-A](http://www.behance.net/gallery/36283417/Example-of-UI-Specification-Document-Project-A)
- 15) *Cílová skupina* [online]. [cit. 2018-09-15]. Dostupné z: [www.podnikatel.cz/pruvodce/obchodni-znacka-brand/cilova-skupina-target-group/](http://www.podnikatel.cz/pruvodce/obchodni-znacka-brand/cilova-skupina-target-group/)
- 16) HEROUT, Tomáš. *Co je to cílová skupina?* [online]. [cit. 2018-09-15]. Dostupné z: [www.helpmark.cz/slovníkpojmu/494-cilova-skupina](http://www.helpmark.cz/slovníkpojmu/494-cilova-skupina)
- 17) PAVLÍČEK, Josef. *INTERAKCE ČLOVĚK A POČÍTAČ 2* [online]. [cit. 2018-09-09].
- 18) ROUSE, Margaret. *Use Case* [online]. [cit. 2018-09-15]. Dostupné z: [www.searchsoftwarequality.techtarget.com/definition/use-case](http://www.searchsoftwarequality.techtarget.com/definition/use-case)



- 19) BITTNER, Kurt a Ian SPENCE. *Use case modeling*. Boston, MA: Addison Wesley, 2003. ISBN 978-0201709131. [cit. 2018-09-15].
- 20) *User Story* [online]. Dostupné z: [www.inflectra.com/ideas/topic/use-cases.aspx](http://www.inflectra.com/ideas/topic/use-cases.aspx)
- 21) PAVLÍČKOVÁ, Petra. *Řízení IT Projektů 5* [online]. [cit. 2018-09-15].
- 22) *What is wireframing?* [online]. [cit. 2018-09-15]. Dostupné z: [www.experienceux.co.uk/faqs/what-is-wireframing/](http://www.experienceux.co.uk/faqs/what-is-wireframing/)
- 23) *V Model For Software Development Life Cycle | Verification & Validation* [online]. [cit. 2018-09-15]. Dostupné z: [www.softwaretestingstudio.com/v-model-software-development/](http://www.softwaretestingstudio.com/v-model-software-development/)
- 24) WIEGERS, Karl Eugene a Joy BEATTY. *Software requirements*. Third edition. Redmond, Washington: Microsoft Press, s division of Microsoft Corporation, [2013]. ISBN 978-0735679665. [cit. 2018-09-16].
- 25) *Functional Requirement* [online]. [cit. 2018-09-16]. Dostupné z: [www.techopedia.com/definition/19508/functional-requirement](http://www.techopedia.com/definition/19508/functional-requirement)
- 26) *Functional vs Non Functional Requirements* [online]. [cit. 2018-09-16]. Dostupné z: [www.reqtest.com/requirements-blog/functional-vs-non-functional-requirements/](http://www.reqtest.com/requirements-blog/functional-vs-non-functional-requirements/)
- 27) *Non Functional Requirements* [online]. [cit. 2018-09-16]. Dostupné z: [www.guru99.com/non-functional-testing.html](http://www.guru99.com/non-functional-testing.html)
- 28) *Software bugs cost US economy dear* [online]. [cit. 2018-09-16]. Dostupné z: [www.nist.gov](http://www.nist.gov)
- 29) *Defect* [online]. [cit. 2018-09-16]. Dostupné z: [www.softwaretestingfundamentals.com/defect/](http://www.softwaretestingfundamentals.com/defect/)
- 30) *Certifikovaný tester – učební osnovy pro základní stupeň* [online]. [cit. 2018-09-16]. Dostupné z: [www.castb.org/wp-content/uploads/2017/01/ISTQB\\_CTFL\\_Syllabus\\_v2011-CZ\\_1\\_0\\_0.pdf](http://www.castb.org/wp-content/uploads/2017/01/ISTQB_CTFL_Syllabus_v2011-CZ_1_0_0.pdf)
- 31) PAVLÍČEK, Josef. *INTERAKCE ČLOVĚK A POČÍTAČ 3* [online]. [cit. 2018-09-16].
- 32) *What is usability testing?* [online]. [cit. 2018-10-20]. Dostupné z: [www.experienceux.co.uk/faqs/what-is-usability-testing/](http://www.experienceux.co.uk/faqs/what-is-usability-testing/)

- 33) *Uživatelské testování krok za krokem* [online]. [cit. 2018-10-20]. Dostupné z:  
[www.pojdmetestovat.cz/file/16](http://www.pojdmetestovat.cz/file/16)
- 34) NIELSEN, Jacob. *How Many Test Users in a Usability Study?* [online]. [cit. 2018-09-16]. Dostupné z: [www.nngroup.com/articles/how-many-test-users/](http://www.nngroup.com/articles/how-many-test-users/)
- 35) *Defect Severity* [online]. [cit. 2018-09-16]. Dostupné z:  
[www.softwaretestingfundamentals.com/defect-severity/](http://www.softwaretestingfundamentals.com/defect-severity/)
- 36) NIELSEN, Jacob. *Usability Metrics* [online]. [cit. 2018-10-20]. Dostupné z:  
[www.nngroup.com/articles/usability-metrics/](http://www.nngroup.com/articles/usability-metrics/)
- 37) BHUSARI, Anand. *User experience design process* [online]. [cit. 2018-10-20].  
Dostupné z: [www.netsolutions.com/insights/user-experience-design-process](http://www.netsolutions.com/insights/user-experience-design-process)

## **8 Seznam použitých zdrojů obrázků:**

- 1) [online]. [cit. 2018-12-19]. Dostupné z: <https://praxent.com/services/ux-design>
- 2) [online]. [cit. 2018-12-19]. Dostupné z: <https://www.smartdraw.com/website-wireframe/>
- 3) [online]. [cit. 2018-12-19]. Dostupné z: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- 4) [online]. [cit. 2018-12-19]. Dostupné z: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- 5) [online]. [cit. 2018-12-19]. Dostupné z: [www.wolframalpha.com](http://www.wolframalpha.com)
- 6) [online]. [cit. 2018-10-20]. Dostupné z: [www.experienceux.co.uk/faqs/what-is-usability-testing/](http://www.experienceux.co.uk/faqs/what-is-usability-testing/)

## 9 Přílohy

Kompletní prototyp systému naleznete v systému Office365 pro ČZU na níže uvedeném odkazu.

[https://czuvprazemy.sharepoint.com/personal/xfism009\\_studenti\\_czu\\_cz/\\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fxfism009\\_studenti\\_czu\\_cz%2FDocuments%2FMichal%20Fi%C5%A1er%20-%20Diplomov%C3%A1%20Pr%C3%A1ce](https://czuvprazemy.sharepoint.com/personal/xfism009_studenti_czu_cz/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fxfism009_studenti_czu_cz%2FDocuments%2FMichal%20Fi%C5%A1er%20-%20Diplomov%C3%A1%20Pr%C3%A1ce)

Pokud budete mít potíže s nahlížecími právy, je možné kontaktovat vedoucího této práce, aby Vám práva udělil.

Adresář, na který je odkazováno, obsahuje zdrojové soubory prototypu i exportované HTML soubory pro snadné zobrazení v lokálním prohlížeči.