

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA

HORNICKO-GEOLOGICKÁ FAKULTA

Katedra geoinformatiky

DIPLOMOVÁ PRÁCE

Ostrava 2019

Bc. Veronika Kocourková

**VYSOKÁ ŠKOLA BÁŇSKÁ –
TECHNICKÁ UNIVERZITA OSTRAVA**

Hornicko–geologická fakulta

Katedra geoinformatiky

**Testování zásuvného modulu pro podporu
pátrání po pohřešované osobě**

Diplomová práce

Autor:

Bc. Veronika Kocourková

Vedoucí diplomové práce:

Ing. Jan Růžička, Ph.D.

Konzultant:

doc. Ing. Miroslav Bureš, Ph.D.

Ostrava 2019

VŠB - Technická univerzita Ostrava
Hornicko-geologická fakulta
Katedra geoinformatiky

Zadání diplomové práce

Student: **Bc. Veronika Kocourková**
Studijní program: N3654 Geodézie, kartografie a geoinformatika
Studijní obor: 3608T002 Geoinformatika
Téma: **Testování zásuvného modulu pro podporu pátrání po pohřešované osobě**
Testing of Plugin for Support of Investigation of Missing Person
Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je realizovat testování zásuvného modulu pro podporu pátrání po pohřešované osobě.

Úkoly:

1. Popište možnosti testování zásuvného modulu pro Quantum GIS.
2. Navrhněte způsoby testování zásuvného modulu Pátrač.
3. Realizujte testování zásuvného modulu Pátrač.
4. Zpracujte doporučení pro další testování.

Rozsah grafických prací:
dle potřeby

Rozsah původní zprávy:
50 - 70 normostran textu

Formální náležitosti diplomové práce stanoví směrnice děkana HGF HGF_SME_15_001 Pokyny pro zpracování závěrečných prací, zveřejněné na webových stránkách fakulty - https://www.hgf.vsb.cz/cs-old/portal-iso/interni/platne-dokumenty/sme/HGF_SME_15_001_A.pdf.

Seznam doporučené odborné literatury:

* Campbell J.E., Shin M. Geographic Information System Basics. 2012. Available at: <http://2012books.lardbucket.org/pdfs/geographic-information-system-basics.pdf>

* Veselovský E. Přehled nástrojů pro automatické testování aplikací. Bakalářská práce. ZČU. 2014.

Veronika Kocourková: Testování zásuvného modulu PÁTRAC

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Růžička, Ph.D.**

Datum zadání: 31.10.2018

Datum odevzdání: 30.04.2019



doc. Ing. Michal Kačmařík, Ph.D.
vedoucí katedry



prof. Ing. Vladimír Slivka, CSc., dr.h.c.
děkan fakulty

Prohlášení autora diplomové práce

- *Celou diplomovou práci včetně příloh, jsem vypracovala samostatně a uvedla jsem všechny použité podklady a literaturu. Byla jsem seznámena s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo.*
- *Beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užít (§ 35 odst. 3).*
- *Souhlasím s tím, že jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o diplomové práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové práce, budou zveřejněny v informačním systému VŠB-TUO.*
- *Souhlasím s tím, že diplomová práce je licencována pod Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported licencí. Pro zobrazení kopie této licence, je možno navštívit <http://creativecommons.org/licenses/by-nc-sa/3.0/>*
- *Bylo sjednáno, že s VŠB-TUO, v případě zájmu o komerční využití z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.*
- *Bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu komerčnímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).*

V Ostravě dne 22. 4. 2019

.....
Bc. Veronika Kocourková

Poděkování

Velmi děkuji panu Ing. Janu Růžičkovi, Ph.D., za odborné vedení mé diplomové práce, cenné rady a velkou dávku trpělivosti. Dále děkuji panu doc. Ing. Miroslavu Burešovi, Ph.D., odbornému konzultantovi z ČVUT v Praze, za mimořádnou ochotu při poskytování jeho praktických zkušeností z oblasti testování systémů.

Rovněž děkuji zástupcům svého zaměstnavatele, Ministerstva vnitra, za poskytnutou podporu a motivaci při studiu, jmenovitě panu Ing. Romanu Vrbovi, řediteli odboru eGovernmentu, a paní RNDr. Evě Kubátové, vedoucí oddělení geoinformatiky.

Anotace

Diplomová práce se věnuje problematice testování zásuvného modulu pro podporu pátrání po pohřešované osobě, zkráceně nazývaného „PÁTRAČ“, který je vyvíjen v rámci projektu Využití vyspělých technologií a čichových schopností psů pro zvýšení efektivity vyhledávání pohřešovaných osob v terénu.

V teoretické části práce jsou popsány metodiky testování, základní principy testování systémů a komerční i volně dostupné nástroje, které se pro tyto účely využívají.

Praktická část diplomové práce se zabývá návrhem metodiky pro potřeby testování softwaru Pátrač, vyvíjeného nad platformou Quantum GIS, a výběrem vhodných nástrojů pro automatizované generování a správu testovacích scénářů. Práce dále obsahuje zdokumentované výsledky manuálních testů, které byly provedeny v uživatelském prostředí, a doporučení pro další vývoj a testování softwaru.

Klíčová slova: metodika testování, testování software, manuální testování, automatizované testování, QGIS, PÁTRAČ, TMap, ISTQB.

Summary

My diploma thesis deals with testing of plug-in module for missing person searching whose abbreviated name is „SEARCHER“. The module is being developed within the project of „The Use of Advanced Technology and Olfactory Capabilities of Dogs to Increase the Efficiency of Missing Persons Searching in the field“.

Testing methodologies, basic principles of system testing and commercial and freely available tools used for these purposes are described in the theoretical part of the thesis.

The practical part of the thesis deals with the design of the methodology for testing the Searcher software developed over the Quantum GIS platform and selection of suitable tools for automated generation and test scenarios management. The thesis also contains documented results of manual tests that have been performed in the user environment and recommendations for further software development and testing.

Key words: Testing Methodology, Software Testing, Manual Testing, Automated Testing, QGIS, Searcher, TMap, ISTQB.

Obsah

1	Úvod	1
1.1	Cíl práce	1
2	Kynologické pátrací týmy k vyhledávání pohřešovaných osob v terénu	2
2.1	Definice pojmů	2
2.2	Možnosti využití kynologických týmů	4
2.3	Povolání kynologických týmů k zásahu	5
2.4	Práce kynologických pátracích týmů na místě zásahu	6
2.5	Úkoly kynologických týmů po ukončení dílčího pátrání	9
2.6	Postup při nálezu pohřešované osoby nebo stop po jejím pohybu	9
3	Zásuvný modul pro podporu pátrání po pohřešované osobě	11
3.1	Uživatelské požadavky zásuvný modul PÁTRAC	11
3.2	Architektura zásuvného modulu PÁTRAC	12
3.3	Proces vytvoření projektu pátrací akce v modulu PÁTRAC	13
4	Testování software	14
4.1	Pozice testování v projektu	16
4.1.1	Metodiky pro vývoj, testování a provoz softwaru	17
4.2	Proč je testování důležité – cena za kvalitu	19
4.3	Základní principy testování	20
4.4	Kvalita softwaru	21
4.5	Typy testů	22
4.5.1	Podle fáze vývoje softwaru	22
4.5.2	Podle závislosti vnitřní struktury	23
4.5.3	Podle realizace testování	24
5	Techniky a nástroje pro testování zásuvného modulu pro Quantum GIS ...	26
5.1	Metodiky pro obecnou podporu testovacího cyklu	26

5.1.1	TMap.....	26
5.1.2	ISTQB	27
5.2	Nástroje pro obecnou podporu testovacího cyklu.....	28
5.2.1	Správa scénářů, reportování chyb.....	28
5.2.2	Reportování chyb	30
5.2.3	Nástroje pro automatizaci testů.....	31
5.3	Techniky pro vytvoření testovacích scénářů.....	33
6	Metodika testování zásuvného modulu Pátrač	34
6.1	Plánování a příprava testů	34
6.1.1	Definice strategie testování.....	35
6.1.2	Specifikace rozsahu testů	35
6.1.3	Definice plánu testů	36
6.1.4	Příprava testovacích sad.....	37
6.1.5	Plánování testovacích běhů.....	37
6.1.6	Tvorba/ úprava dokumentace.....	37
6.1.7	Příprava testovacího prostředí.....	37
6.1.8	Přijetí aplikace do testování	38
6.2	Provedení a vyhodnocení testů.....	38
6.2.1	Provedení testů	38
6.2.2	Typy testů a jejich výstupy	38
6.2.3	Reporting chyb	39
6.2.4	Retesty opravených chyb	39
6.2.5	Vyhodnocení testů.....	39
7	Návrh testovacích scénářů pro zásuvný modul Pátrač	40
7.1	Analýza parametrů úlohy	40
7.1.1	Terén.....	40
7.1.2	Parametry dle uživatele.....	40
7.2	Identifikované parametry testované úlohy	41
7.2.1	Členitost terénu.....	41
7.2.2	Výskyt neprůchozích částí v terénu	42

7.2.3	Poslední spatření	42
7.2.4	Směr pohybu, zda ano/ ne	42
7.3	Výchozí testovací plán	42
7.3.1	SET 1: Členitost terénu	43
7.3.2	SET 2: Uživatel	45
8	Realizace vybraných testů	46
8.1	Příprava manuálních testů	46
8.1.1	Specifikace rozsahu manuálních testů	46
8.1.2	Definice plánu manuálních testů	46
8.1.3	Plánování testovacích běhů testu Projekt	47
8.2	Provedení a vyhodnocení testů manuálních testů	47
8.2.1	Realizace testu Projekt a reporting chyb	48
9	Doporučení pro další testování	55
10	Závěr	57
	Seznam použité literatury	58
	Seznam použitých zkratk	61
	Seznam obrázků	63
	Seznam tabulek	63
	Seznam příloh	64

1 ÚVOD

Projekt Využití vyspělých technologií a čichových schopností psů pro zvýšení efektivity vyhledávání pohřešovaných osob v terénu (dále jen „PÁTRAC“) je realizovaný z programu bezpečnostního výzkumu České republiky 2015 – 2022 (dále jen „Program“), který je zpracován v souladu s koncepčními a strategickými dokumenty České republiky v oblasti výzkumu, experimentálního vývoje a inovací a v oblasti bezpečnosti.

Posláním projektu PÁTRAC je zrychlit a zefektivnit rozhodování a řízení v procesech při zahájení a provádění pátracích akcí s využitím nově vyvíjeného softwaru komunikujícího s technikou pro přesnou lokalizaci a navigaci v terénu, a zároveň připravit nástroj umožňující generování dokumentace o průběhu pátracích akcí, jež bude vstupem pro následnou analýzu získaných statistických dat.

Průběžně získávaná statistická data umožní optimalizovat přípravu kynologických pátracích týmů zkoumáním vlivu způsobu práce kynologických pátracích týmů na únavu, vyčerpání a stres psů a jakým způsobem tyto faktory ovlivňují spolehlivost psů při pátrání.

Dalším podstatným úkolem projektu PÁTRAC je připravit využitelné výsledky pro složky Integrovaného záchranného systému (dále jen „IZS“), které realizují pátrání po pohřešovaných osobách v terénu (mj. Hasičský záchranný sbor České republiky (dále jen „HZS ČR“), Policie České republiky, Zdravotnická záchranná služba, Horská služba, Městské policie, občanská sdružení apod.) na základě zmocnění zákona č. 239/2000 Sb., o integrovaném záchranném systému a o změně některých zákonů.

1.1 Cíl práce

Cílem práce je realizovat testování zásuvného modulu pro podporu pátrání po pohřešované osobě. V rámci zpracování diplomové práce budou popsány možnosti testování zásuvného modulu PÁTRAC pro platformu Quantum GIS, navrženy způsoby testování zásuvného modulu PÁTRAC, dle navržené metodiky testování budou realizovány vytipované automatické testy. Závěrem práce bude zhodnocení průběhu testování a zpracování doporučení pro další testování.

2 KYNOLOGICKÉ PÁTRACÍ TÝMY K VYHLEDÁVÁNÍ POHŘEŠOVANÝCH OSOB V TERÉNU

Pro pochopení konceptu modulu PÁTRAC je nutné definovat základní pojmy z oblasti vyhledávání pohřešovaných osob v terénu, na které se budu v dalších kapitolách diplomové práce odvolávat, a dále popsat stanovené a doporučené postupy pro použití kynologických pátracích týmů při vyhledávání pohřešovaných osob v terénu. Uvedené postupy jsou plně v souladu s platnou legislativou České republiky, s interními akty řízení základních složek integrovaného záchranného systému (dále jen „IZS“ – pozn. Zákon č. 239/ 2000 Sb., o integrovaném záchranném systému a o změně některých zákonů) a respektovaly současnou úroveň poznání a moderních technologií.

Zdrojem informací pro charakteristiku činností složek IZS při realizaci pátrání po pohřešované osobě, jak je popsáno níže v této kapitole, byly interní materiály Policie České republiky (dále jen „PČR“) a konzultace s pověřeným zástupcem PČR.

2.1 Definice pojmů

Pod následujícími pojmy se rozumí:

- **kynologickým atestem** – osvědčení o splnění stanovených kvalifikačních požadavků (pozn. Pokyn generálního ředitele HZS ČR a náměstka MV č. 48/2003, kterým se stanoví kvalifikační požadavky na psovoda se psem předurčeného k nasazení v rámci záchranných prací) na psovoda se psem předurčeném k nasazení v rámci záchranných prací při společném zásahu složek IZS,
- **kynologickým týmem** – psovod se speciálně vycvičeným psem k vyhledávání osob,
- **kynologickým pátracím týmem** (dále jen „KPT“) – kynologický tým na vyhledávání pohřešovaných osob v přírodním terénu, který je držitelem kynologického atestu MV se specializací „P“ – plošné vyhledávání, který může být doplněn o další členy dle potřeby,
- **mapovými podklady** – topografická mapa prostoru pátrací akce v měřítku,
- **místem zásahu** – prostor, ve kterém probíhají pátrací práce v terénu řízené velitelem zásahu složek IZS za účelem vyhledání pohřešované osoby nebo pohřešovaných osob,

- **plošným vyhledáváním** – vyhledávání pohřešované osoby kynologickým pátracím týmem v přírodním terénu,
- **pohřešovanou osobou** – je dle §111 odst. 1 písm. d) zákona č.273/2008 Sb., o Policii České republiky fyzická osoba, o níž se lze důvodně domnívat, že je ohrožen její život nebo zdraví, místo jejího pobytu není známo a Policií České republiky po ní bylo vyhlášeno pátrání,
- **velitelem kynologické skupiny** – zpravidla příslušník PČR, do jehož podřízenosti určí velitel zásahu kynologické pátrací týmy na místě zásahu,
- **psem výjezdové skupiny** – pes PČR, který splňuje požadavky k výkonu služby ve výjezdové skupině služby kriminální policie a vyšetřování v souladu s interními akty řízení PČR (pozn. Pokyn policejního prezidenta č. 145 ze dne 8. července 2014, kterým se upravuje činnost služební kynologie),
- **psovodem** – fyzická osoba jako dobrovolník nebo příslušník bezpečnostního nebo záchranného sboru, která se zabývá výcvikem a použitím psů a která prokázala požadovanou odbornost kvalifikační zkouškou podle některého z interního aktu řízení PČR (pozn. Pokyn policejního prezidenta č.145/2014, kterým se upravuje činnost služební kynologie a služební hipologie; Pokyn ředitele Ředitelství služby pořádkové policie č.119/2017, o provádění služební kynologie) nebo HZS ČR (pozn. Pokyn generálního ředitele HZS ČR a náměstka MV č. 48/2003),
- **sektorem** – prostorově ohraničená část území určená k prohledání pátracími prostředky za účelem nalezení pohřešované osoby,
- **shromaždištěm** – prostor určený k soustředění sil a prostředků vyčleněných k provádění pátracích prací v terénu,
- **stupněm naléhavosti** – označení závažnosti neboli neodkladnosti řešení případu třemi stupni podle získaných informací o pohřešované osobě a okolnostech jejího pohřešování,
- **sutinovým vyhledáváním** – vyhledání osob kynologickými týmy na sutinové vyhledávání pod vrstvou trosek staveb, sutě, kamení či jiného materiálu, který mohl pohřešovanou osobu zasypat, kromě vyhledávání zasypaných osob

v lavinách (pozn. Pokyn generálního ředitele HZS ČR a náměstka MV č. 48/2003),

- **velitelem zásahu** – je dle §19 zákona č. 239/2000 Sb., o IZS zpravidla příslušník PČR, který může být současně i velitelem pátrací akce prováděné podle interního aktu řízení; velitel zásahu koordinuje činnost zasahujících složek IZS, včetně osob poskytujících osobní a věcnou pomoc a na místě řídí jejich součinnost; v případě potřeby může zřídit štáb velitele zásahu,
- **štábem velitele zásahu** – výkonný orgán řízení zásahu složený ze zástupců zasahujících složek případně speciálních služeb a skupin, který při zásahu většího množství sil a prostředků nebo při organizačně složitě pátrací akci v souladu s rozhodnutími velitele zásahu organizuje činnost na místě zásahu; štáb nepřebírá oprávnění a povinnosti velitele zásahu a je využíván zejména pro koordinaci složek IZS,
- **týlový prostor** – prostor určený k odpočinku a občerstvení osob účastnících se vyhledávání pohřešované osoby.

2.2 Možnosti využití kynologických týmů

Policii ČR téměř vždy při pátrání po pohřešované osobě, která se ztratila v rozsáhlém přírodním terénu, chybí pro okamžité zahájení a provedení pátrací akce, za účelem jejího včasného nalezení, dostatečné síly a prostředky. Z tohoto hlediska je velmi vhodné použití kynologických týmů, které mají vysoký poměr efektivity na zúčastněnou osobu na místě zásahu.

Případy, kdy je vhodné povolát kynologické pátrací týmy:

- na základě vyhodnocených informací nelze vyloučit přítomnost pohřešované osoby v přírodním terénu,
- na základě vyhodnocených informací nelze vyloučit její přítomnost pod vrstvou sutě, trosek staveb či jiného materiálu (zasypání, propadnutí apod.) – v tomto případě je účelné použití kynologických týmů, které jsou držiteli kynologického atestu MV se specializací „S“ (sutinové vyhledávání).

2.3 Povolání kynologických týmů k zásahu

Velitel zásahu stanoví potřebný počet kynologických týmů a jejich specializaci podle okolností případu zejména na základě velikosti plochy určené k prohledání, podle počtu objektů a ploch se sutinami v prohledávané oblasti.

Kynologické týmy se specializací sutinového vyhledávání velitel zásahu v případě potřeby vyžaduje podle metodiky sutinového vyhledávání vydaného Generálním ředitelstvím hasičského záchranného sboru České republiky (dále jen „GŘ HZS ČR“).

Pro stanovení způsobu a rozsahu provedení pátrací akce je nutné stanovit stupeň naléhavosti řešení daného případu, který určuje některé parametry pátrací akce. Velitel pátrací akce s ohledem na odhadovaná rizika ohrožení života a zdraví pohřešované osoby může zvolit mezi třemi stupni naléhavosti případu:

- I. stupeň – nelze vyloučit, že se pohřešovaná osoba nachází v terénu a riziko ohrožení jejího života a zdraví je vzhledem ke známým okolnostem velmi nízké,
- II. stupeň – dle dostupných informací je pravděpodobné, že se pohřešovaná osoba nachází v terénu a existuje zde ale reálné riziko ohrožení jejího života a zdraví vzhledem ke známým okolnostem,
- III. stupeň – pohřešovaná osoba se s velmi vysokou pravděpodobností nachází v terénu a riziko ohrožení jejího života a zdraví je vzhledem ke známým okolnostem velmi vysoké.

Dostupnost KPT a jejich povolání na místo zásahu velitel zásahu provádí prostřednictvím mobilní aplikace „PÁTRAC“ nebo integrovaného operačního střediska PČR u územně příslušného krajského operačního a informačního střediska HZS ČR (dále jen „KOPIS“).

V případě vyžádání KPT cestou operačního střediska postupuje velitel zásahu následovně:

- ověří počet dostupných KPT a jejich dojezdové časy prostřednictvím integrovaného operačního střediska PČR dotazem na KOPIS; dostupnost KPT zjišťuje KOPIS dotazem u příslušných kynologů na základě aktualizovaného seznamu držitelů kynologických atestů MV; získané informace předává na integrovaného operačního střediska PČR, které je

následně předává veliteli zásahu předem domluveným způsobem (radiokomunikace, SMS zpráva, e-mail, informačním systémem JITKA apod.),

- vyžádá na místo zásahu potřebný počet KPT prostřednictvím integrovaného operačního střediska PČR u KOPIS; při vyžádání KPT předá popis pohřešované osoby, kontaktní osobu na místě zásahu, kontaktní telefon na tuto osobu, popis místa srazu včetně souřadnic UTM; integrované operační středisko PČR veliteli zásahu potvrzuje přibližnou dobu příjezdu na místo zásahu.

Po příjezdu KPT na místo zásahu se psovod ohlásí u určené kontaktní osoby, pokud není tato osoba určena, tak u velitele kynologické skupiny, pokud není ještě ustanoven, tak u velitele zásahu. Psovod KPT se prokáže platným průkazem držitele kynologického atestu MV. Údaje o KPT a jejich čase příjezdu kontaktní osoba zaznamená do stanovené dokumentace pátrací akce, která je podkladem pro zprávu o zásahu.

2.4 Práce kynologických pátracích týmů na místě zásahu

KPT na místě zásahu postupují podle „Metodiky vyhledávání pohřešovaných osob kynologickými pátracími týmy v terénu“. KPT se specializací na vyhledávání osob v sutinách postupují přiměřeně podle „Metodiky sutinového vyhledávání“.

Komunikace mezi jednotlivými KPT a vedoucím kynologické skupiny je standardně zajišťována přenosnými radiostanicemi v analogové radiové síti IZS. Pro vzájemnou komunikaci účastníků pátrací akce mohou být ve specifických případech používány i komunikační signály. Pro zajištění nepřetržitého radiového spojení, orientace v terénu a bezpečnosti celého týmu je vhodné, aby každý KPT byl složen minimálně ze dvou fyzicky zdatných osob (psod se psem a minimálně jeden další doprovodný člen týmu).

KPT jsou přednostně používány k prohledání částí terénu, které nelze prohledat nebo lze prohledat jen velmi obtížně jinými pátracími prostředky (např. velmi členitý nebo nepřehledný terén), v místech ukončení nebo ztráty stopy psem výjezdové skupiny, za snížené viditelnosti jako je mlha nebo tma mají KPT výrazně vyšší efektivitu oproti ostatním pátracím prostředkům.

Rychlost pátrání KPT je závislá na členitosti terénu, hustotě vegetace a na povětrnostních podmínkách, které zásadně ovlivňují šíření pachu v prostoru. Průměrná rychlost pátrání KPT se ve středně členitém terénu a za běžných povětrnostních podmínek pohybuje okolo 10 ha/hod. Extrémní povětrnostní podmínky (např. silný nárazový vítr, bezvětří apod.) nebo výrazná členitost terénu (např. rokle, polom apod.) průměrnou rychlost pátrání výrazně snižuje.

Z důvodu efektivního využití KPT je vhodné plánovat jednotlivé pátrací úkoly v minimální délce 2 hodin a maximální délce 4 hodiny, přestávky na shromaždišti mezi jednotlivými pátracími pracemi KPT jsou zpravidla plánovány v rozmezí 30 minut až 2 hodin podle terénních a povětrnostních podmínek.

Při déle trvající pátrací akci je nutné v rámci každých 24 hodin nasazení všem členům KPT zajistit přiměřený odpočinek, a to v délce minimálně 8 hodin.

Vedoucí kynologické skupiny je na místě zásahu přímo podřízen přidělenému veliteli, kterého mu určí velitel zásahu, na místě zásahu se seznámí se situací a okolnostmi případu, zejména s popisem a zdravotním stavem hledané osoby, místy a časy jejího ověřeného výskytu, nebezpečími a riziky v prostoru zásahu, riziky hrozící od hledané osoby, přehledem o pohybu ostatních pátracích prostředků v blízkosti sektorů určených pro práci KPT, plánem spojení včetně náhradního způsobu spojení s přiděleným velitelem, způsobem určeným pro předávání poznatků a přebírání úkolů, místem vyhrazeným pro odpočinek KPT a způsobem přepravy KPT. Vedoucí kynologické skupiny předkládá návrhy přidělenému veliteli na využití KPT v daných podmínkách, informuje přiděleného velitele o aktuálním počtu psovodů se psy na místě zásahu včetně jejich specializace, případně o jejich speciálním vybavení.

Vedoucí kynologické skupiny dále v součinnosti s přiděleným velitelem zajišťuje doplnění KPT doprovodnými členy KPT, pokud psovod nemá vlastního člena týmu, zajišťuje doplnění nebo výměnu vybavení KPT, je ve stálém spojení se všemi KPT na místě zásahu, přebírá poznatky a výsledky práce od KPT, provádí jejich analýzu a výsledky předává přidělenému veliteli, vede písemnou dokumentaci o čase příjezdu psovodů na místo, o všech časech převzetí a splnění úkolů, o době nasazení jednotlivých KPT a o čase odjezdu psovodů z místa zásahu, zajišťuje specifické požadavky potřebné

pro účelné použití psů (např. odstranění překážek a rušivých vlivů, které mohou negativně ovlivnit práci psa apod.), zajišťuje nezbytné požadavky na veterinární péči.

Pokud mu to rozsah akce a objem činností vyplývajících z funkce vedoucího kynologické skupiny umožní, může být sám členem některého kynologického týmu provádějícího pátrání.

Vedoucí kynologické skupiny pro splnění úkolu a zvýšení efektivity práce rozhoduje zejména o způsobu a pořadí použitých metod vyhledávání podle podmínek na místě zásahu a dostupných sil a prostředků KPT, rozdělení přiděleného prostoru na jednotlivé sektory a určení pořadí průzkumu jednotlivých částí, zastavení práce kynologických týmů, jestliže dojde ke zvýšení ohrožení života členů kynologických týmů, nebo v případě značné únavy psů, provedení opakovaného vyhledávání (tzv. překrytí) jiným kynologickým týmem, při pochybách o činnosti psa nebo v jiných sporných případech, vystřídání kynologického týmu v případech, kdy nemůže tento tým dokončit stanovený úkol, nezbytných přestávkách určených k odpočinku kynologických týmů.

Přidělený velitel předává KPT informace o pohřešované osobě a o okolnostech případu, nebezpečích a rizicích na místě zásahu, plánu spojení a náhradním plánu spojení (např. radiový kanál, volací znaky, telefonní kontakty apod.), jejich úkolu včetně časového plánu, pohybu ostatních pátracích prostředků v prostoru určeném pro KPT nebo v jeho blízkosti, aktuálním počasím a předpokladu jeho vývoje na dobu minimálně 2krát delší, než je plánována doba práce KPT v terénu.

Nasazení KPT je možné až po likvidaci bezprostředního nebezpečí (např. požáru, úniku plynu z poškozených rozvodů, elektrická instalace pod napětím, zamezení volného pohybu psů a jiných zvířat, které by mohly ohrozit práci kynologických týmů apod.).

Pátrací činnosti KPT je možné při dodržení všech bezpečnostních zásad a pravidel provádět i za snížené viditelnosti, tj. v noci a za mlhy.

Ostatní pátrací týmy a skupiny postupují podle standardních taktických postupů určených pro jejich činnosti při pátrání po pohřešovaných osobách v terénu dle pokynů velitele zásahu.

2.5 Úkoly kynologických týmů po ukončení dílčího pátrání

Všichni členové KPT provedou po každém ukončení dílčího pátrání kontrolu kompletnosti a funkčnosti své výstroje a výbavy, v případě potřeby zajistí jejich opravu, doplnění či regeneraci (např. dobítí zdrojů apod.).

Psovod zkontroluje zdravotní stav psa a kompletnost a funkčnost jeho výstroje.

Pokud KPT nebo KPT na vyhledávání osob v sutinách pohřešovanou osobu přímo nenalezl, předá vedoucímu kynologické skupiny informace o výsledcích své práce v terénu, a to zejména do přidělených mapových podkladů označí modrou barvou území, které je spolehlivě prohledáno, oranžovou barvou označí území, které nebylo možné spolehlivě prohledat, zdůvodní proč a případně navrhne možnosti propátrání tohoto území jinými pátracími prostředky a dále do přidělených mapových podkladů označí a popíše místa, která neodpovídají informacím na mapách. V případě použití přístroje GPS poskytne vedoucímu kynologické skupiny záznam prošlé trasy a záznam trasových bodů ze svého přístroje, do přidělených mapových podkladů vyznačí všechna místa získaných poznatků a nálezů stop.

Po ukončení práce na místě zásahu zpracuje každý psovod písemný záznam o svém nasazení a předá jej vedoucímu kynologické skupiny, který do 10 dní od ukončení nasazení doručí na GŘ HZS ČR odbor IZS a služeb celkovou zprávu o činnosti kynologické skupiny na místě zásahu.

Kynologické týmy provedou výše uvedené úkoly i v případech, kdy celé dílčí pátrání nedokončí z důvodu zranění, ukončení pátrací akce apod.

2.6 Postup při nálezu pohřešované osoby nebo stop po jejím pohybu

Nalezne-li KPT pohřešovanou osobu a může se bezpečně dostat až k ní, provede dle možnosti její transport na bezpečnější místo, jestliže se nalezená osoba nachází na místě, na kterém může být ohrožena na životě nebo zdraví. Zjistí její zdravotní stav, neprodleně provede potřebná opatření k záchraně jejího života, pokud je pohřešovaná osoba ohrožena na životě a její stav vyžaduje poskytnutí první pomoci. V případě, že pohřešovaná osoba nejeví známky života, zahájí neprodleně resuscitaci, a co nejdříve si vyžádá pomoc dalších záchranářů, od resuscitace může upustit pouze v případech, kdy není pochyb o tom, že osoba již nežije (např. postavení částí těla neslučitelná se životem,

posmrtné skvrny apod.). Po provedení činností, které nesnesou odkladu (např. poskytnutí první pomoci nebo provedení transportu osoby na bezpečnější místo), oznámí vedoucímu kynologické skupiny přesné místo nálezů pohřešované osoby s uvedením rovinných souřadnic UTM. V případech, kdy je nalezená osoba zraněna nebo nejeví známky života, na místě nálezů a v jeho bezprostřední blízkosti zamezí pohybu nepovolaných osob a pohyb záchranářů co nejvíce minimalizuje a pokud je to možné pohyb všech osob na místě nálezů zaznamená pro potřeby vyšetřování případu. Vedoucímu kynologické skupiny předává požadované informace (např. nejvhodnější přístupová cesta pro další záchranné týmy, vhodné místo pro přistání vrtulníku apod.). Prostřednictvím radiové sítě předává informace o zdravotním stavu pohřešované osoby pouze na výslovnou žádost velitele zásahu, nebo na uzavřeném radiovém kanálu určeném pro předávání těchto informací.

Nalezne-li kynologický tým pohřešovanou osobu a nemůže se bezpečně dostat až k ní, oznámí vedoucímu kynologické skupiny případně přidělenému veliteli přesné místo nálezů pohřešované osoby s uvedením rovinných souřadnic UTM a popíše situaci na místě nálezů, zejména důvod, proč se nemůže k osobě dostat. Pokusí se odhadnout zdravotní stav pohřešované osoby, a pokud osoba komunikuje, tak jí uklidňuje a průběžně zjišťuje její zdravotní stav.

Nalezne-li kynologický tým stopu (např. věcnou nebo trasologickou) po pohybu pohřešované osoby, podle možností zamezí pohybu nepovolaných osob v okolí místa nálezů do vzdálenosti cca 50 m, věcných stop se nedotýká, nepřemisťuje je a vzdálí se od nálezů do vzdálenosti cca 25 m po své příchodové trase, tak aby měl místo nálezů pod dohledem. Oznámí vedoucímu kynologické skupiny druh nálezů a jeho přesné místo s uvedením rovinných souřadnic UTM a dále postupuje podle instrukcí vedoucího kynologické skupiny.

3 ZÁSUVNÝ MODUL PRO PODPORU PÁTRÁNÍ PO POHŘEŠOVANÉ OSOBĚ

V předchozí kapitole 2 byly definovány základní pojmy z oblasti KPT a jen velmi stručně popsány činnosti KPT při pátrání po pohřešovaných osobách jako úvod do problematiky, kterou se zabývá projekt PÁTRAC.

KPT se při plnění úkolů (viz kapitola 2) potýkají s čtyřmi základními problémy, kterými jsou:

- pátrací akci již z podstaty nepředvídatelné události nelze dopředu naplánovat a jedná se tedy o situaci, kterou je nutné operativně zorganizovat,
- bývá pravidlem, že je v ohrožení zdraví a život pohřešované osoby, hlavní roli v nastalé situaci hraje čas,
- obvykle je jen velmi málo dostupných informací o pohybu pohřešované osoby v terénu, a pokud už nějaké jsou, mívají různou míru věrohodnosti,
- z důvodu velké neurčitosti v informacích o pohřešované osobě, je nutné propátrat rozsáhlá území.

Výše uvedené problémy při pátrání v terénu je možné zobecnit konstatováním, že KPT mají mnoho úkolů, ale nedostatek času a prostředků.

Prvotní motivací byla tedy snaha vytvořit nástroj pro zvýšení efektivity vyhledávání pohřešovaných osob v terénu a tím vyřešit stávající situaci, která pro KPT není komfortní, komplikuje přípravu pátracích akcí i samotnou práci KPT v terénu a v neposlední řadě neumožňuje sbírat cenná statistická data z realizovaných pátracích akcí pro následné využití v analýzách.

Vize projektu PÁTRAC si klade za cíl nechat dělat stroj, v čem je rychlejší a efektivnější než člověk.

3.1 Uživatelské požadavky zásuvný modul PÁTRAC

V projektové dokumentaci vyvíjeného modulu PÁTRAC byly definovány mimo jiné základní uživatelské požadavky na nový software, kterými jsou například:

- na základě zadání posledního místa spatření pohřešované osoby, PÁTRAC vytipuje a zobrazí na mapě oblast předpokládaného výskytu pohřešované osoby s určitou mírou pravděpodobnosti,

- PÁTRAC navrhne všechny druhy prostředků sil a prostředků, které jsou v rámci IZS pro tyto činnosti alokovány,
- PÁTRAC navrhne využití daných druhů sil a prostředků podle typu pátracího prostoru,
- PÁTRAC vypočte potřebný čas k propátrání prostoru dle dostupných sil a prostředků,
- PÁTRAC prověří dostupnost a časové možnosti KPT, zadá lokalitu pátrací akce a následně odešle dotaz dostupným KPT
- PÁTRAC provede povolání KPT pomocí mobilní aplikace,
- PÁTRAC zadá lokalizaci shromaždiště sil a prostředků, informace o kontaktní osobě, informace o pohřešované osobě a odešle KPT a na KOPIS,
- PÁTRAC rozdělí vygenerovanou oblast předpokládaného výskytu pohřešované osoby na sektory dle požadovaných kritérií (členění dle druhu, sektory dělí pomocí přirozených hranic s dodržáním maximální velikosti sektoru 30 hektarů),
- PÁTRAC exportuje sektory do GPS přístrojů a mobilních telefonů povolaných KPT a dále exportuje mapové výstupy pro tisk,
- PÁTRAC komunikuje se souběžně vyvíjenou mobilní aplikací,
- PÁTRAC zaznamenává prošlé trasy, přehled KPT, dokumentuje postupy při pátrání a vyhodnocuje poznatky,
- a další doplňkové funkcionality dostupné v dokumentaci projektu PÁTRAC.

3.2 Architektura zásuvného modulu PÁTRAC

Zásuvný modul PÁTRAC je vyvíjen jako off-line pracující desktopové řešení postavené na open source nástroji Quantum GIS doplněné o mobilní aplikaci, která pracuje na operačním systému Android a HTTP Serveru.

Modul PÁTRAC je vyvíjený ve skriptovacím jazyce Python a pro analytickou část využívá rozšíření Quantum GIS o nástroj GRASS GIS.

Nástavbová mobilní aplikace v programovacím jazyce Java s Android SDK.

3.3 Proces vytvoření projektu pátrací akce v modulu PÁTRAC

Modul PÁTRAC nabízí uživateli dva panely, a to „Průvodce“ a „Expert“.

Pro vytvoření nového projektu uživatel zadá v panelu „Průvodce“ název obce, ve které bude realizována pátrací akce, a výběr potvrdí. V mapovém okně proběhne přiblížení na zvolené území.

V následujícím kroku uživatel zadá místo, případně místa posledního spatření pohřešované osoby, buď kliknutím do mapového okna, nebo zadáním známých GPS souřadnic, a ke každé lokalizaci místa posledního spatření doplní čas spatření a váhu důvěryhodnosti svědectví v intervalu 0 – 9 (0 – nedůvěryhodné, 9 – potvrzené), následně potvrdí volbu.

Ve třetím kroku vybere z nabízených možností typ pohřešované osoby a výběr potvrdí. PÁTRAC na základě zadaných parametrů provede výpočet oblasti možného výskytu pohřešované osoby a zobrazí ji v mapovém okně.

Ve čtvrtém kroku uživatel stanoví očekávanou procentuální míru pravděpodobnosti výskytu pohřešované osoby ve vypočtené oblasti, přičemž platí, čím vyšší procento pravděpodobného výskytu pohřešované osoby v oblasti, tím větší oblast pro pátrání. V následujícím kroku zadá dostupné síly a prostředky pro pátrací akci a volbu potvrdí.

Dále je uživatel vyzván k doplnění dostupných sil a prostředků pro pátrání. Po vyplnění a potvrzení údajů PÁTRAC provede překryvnou operaci rastrové vrstvy obsahující vypočtenou oblast předpokládaného výskytu pohřešované osoby s vektorovou vrstvou, která obsahuje předgenerované pátrací sektory.

Nakonec PÁTRAC reportuje průnik jednotlivých sektorů s vypočtenou oblastí možného výskytu pohřešované osoby v podobě mapových výstupů ve formátu PDF pro tisk a GPX do GPS přijímačů. Po dokončení pátrací akce je nutné, aby uživatel doplnit výsledek pátrání pomocí integrovaného formuláře. Data získaná z realizovaných pátracích akcí budou sloužit pro následné analýzy a strojové učení modulu PÁTRAC.

Panel „Expert“ nabízí uživateli možnost zpětné editace již vytvořeného projektu, například pomocí správy míst posledního spatření, ručního výběru sektorů, přečíslování sektorů, ručního exportu sektorů, vytvoření reportů, případně editovat typ pohřešované osoby apod.

4 TESTOVÁNÍ SOFTWARE

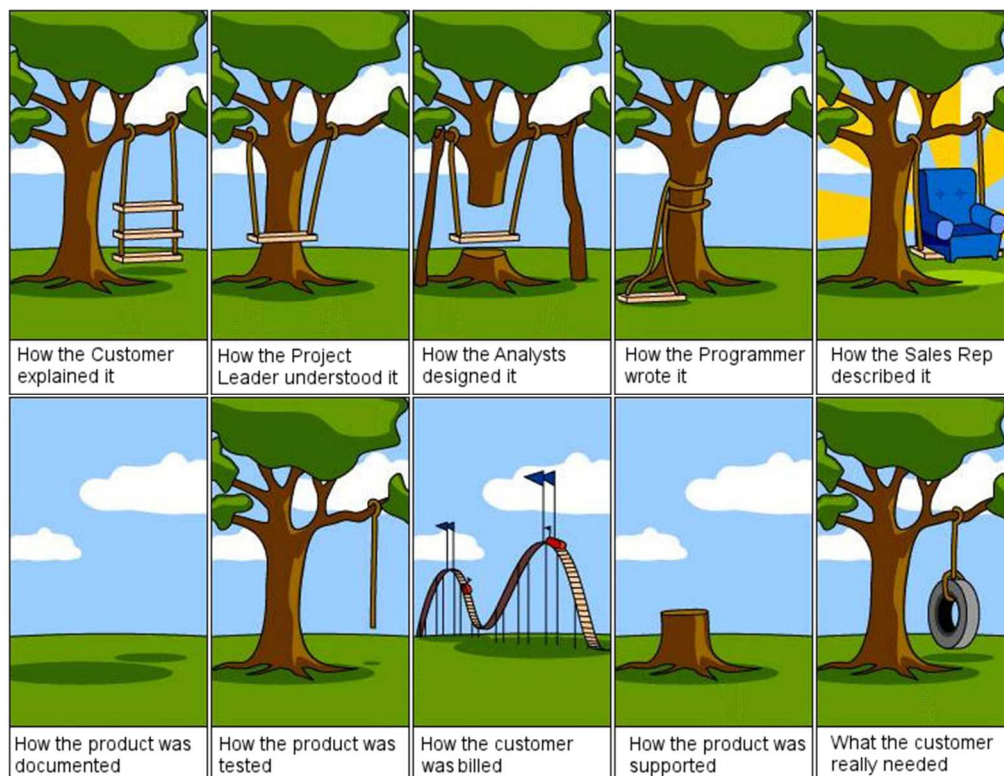
V současné digitální době je důraz na testování softwaru nutností ale i zábavou. [1]

Co ale to testování softwaru je?

Definice testování softwaru se v literatuře zabývající se kvalitou softwaru nachází mnoho, jelikož testování se vyvíjí stejnou rychlostí, jakou postupuje rozvoj systémů. Jako příklad lze uvést definici G. J. Myerse, který říká: *„Testování je proces spouštění programu se záměrem nalézt vadu.“* Standard ISEB testování definuje následovně: *„Testování obsahuje všechny aktivity, jejichž cílem je měřit a kontrolovat kvalitu softwaru.“*

Pro názornost je možné proces vývoje softwaru a testování přirovnat k plánu výstavby rodinného domu. Investor se stavební firmou uzavře smlouvu o dílo, jejímž předmětem bude projekt a výstavba rodinného domu dle požadavků investora. Stavební firma bude k výstavbě potřebovat architekta, stavební dělníky (programátory), stavební dozor (testery). Architekt připraví ve spolupráci s investorem projekt. Stavební dělníci začnou podle stavební dokumentace stavět a stavební dozor kontroluje, zda stavba odpovídá zadání jak architekta, tak i požadavkům investora. Při vývoji softwaru je proces prakticky totožný. Zákazník představí své potřeby softwarovému architektovi, který na základě analýzy potřeb zákazníka navrhne požadovaný software. Programátoři dle podkladů o softwarového architekta naprogramují výsledný produkt a testéři jako stavební dozor u stavby rodinného domu ověřují, zda vytvořený software odpovídá potřebám zákazníka, které byly definovány v zadání. [1]

Nejlépe celou problematiku vystihuje velmi známý obrázek návrhu houpačky (viz Obrázek 1), který v 10 krocích názorně ukazuje, jak snadno může dojít k nepochopení mezi zadavatelem (investorem), softwarovým architektem a programátory. [2]



Obrázek 1: Problém s vývojem softwaru [2]

Krok 1: Znázorňuje, jak zákazník vysvětlil požadavek. Zákazník požadoval třívrstvou houpačku po celé větvi, jak je vidět na obrázku níže. V následujících krocích je naznačeno, jak jednotliví lidé chápali jeho požadavky.

Krok 2: Ukazuje, jak byl zákazníkům požadavek pochopen vedoucím projektu.

Krok 3: Na základě požadavku analytik navrhl software.

Krok 4: Znázorňuje, jak byl software na základě požadavku vývojářem naprogramován.

Krok 5: Ukazuje, jak vyvíjený software popsal obchodní konzultant.

Krok 6: Projektová dokumentace zcela chybí.

Krok 7: Popisuje průběh testování produktu.

Krok 8: Naznačuje náklady, které byly zákazníkovi fakturovány.

Krok 9: Popisuje poskytovanou zákaznickou podporu.

Krok 10: Ukazuje, co zákazník opravdu chtěl. Požadavek zákazníka byl jednoduchý, a to houpačka přes větev stromu.

Modelová situace na obrázku může sice vypadat vtipně a s jistou mírou nadsázky, nicméně skutečnost je opravdu taková. Když si různé skupiny lidí předávají informace pro zajištění úkolů v rámci projektu, vždy při dialogu vzniká určitý šum vedoucí ke ztrátě části informace. Zmíněný šum v komunikaci mezi zákazníkem a projektovým týmem může ve svém důsledku vést až k tomu, že zákazník získává produkt, který zdaleka není tím, co potřeboval. [2]

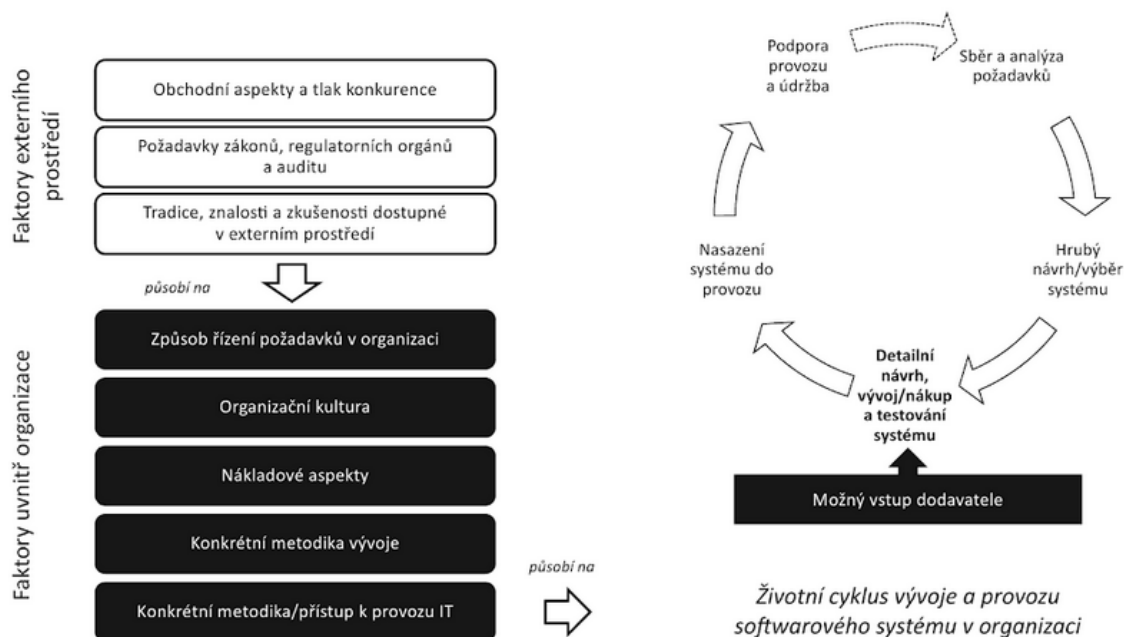
Jak zmíněný problém řešit? Nemohou být vytvářeny produkty, které zákazníkovi nepřinesou žádnou přidanou hodnotu. "Kvalita" produktů nebo služeb, které jsou zákazníkům nabízeny, je nanejvýš důležitá. [2]

Hlavní podstatou testování softwaru je tedy potvrzení kvality pomocí pozitivních testů nebo naopak simulování selhání systému za pomoci negativních testů. To znamená, že tester musí ověřit, že požadované funkce nastanou a nechtěné situace nenastanou. [1]

4.1 Pozice testování v projektu

V minulosti se k testování přistupovalo až v poslední fázi vývoje softwaru a jako takovému nebyla přikládána patřičná váha. Dnes je ale testování již nedílnou součástí všech fází vývoje nového softwaru a lze o něm hovořit jako o sofistikovaném procesu, který se skládá z mnoha částí a je do značné míry ovlivňován kontextem konkrétního projektu a společnosti. Zmíněný kontext určuje specifickou filosofii testování konkrétní organizace, jež zahrnuje používanou metodiku testování a komplexnost přístupu k testování v dané společnosti. [3]

Danou problematiku je možné zobecnit pomocí modelu obecného životního cyklu vývoje systémů začleněného do organizačních souvislostí subjektu, jak je možné vidět na Obrázek 2, který představuje dlouhodobé aktivity vykonávané v rámci vývoje a provozu daného informačního systému. Pravá strana znázorňuje jednotlivé činnosti od sběru a analýzy požadavků, přes konceptuální návrh, vývoj a testování až po nasazení systému do produkčního provozu a údržbu. [3]



Obrázek 2: Model průběžného rozvoje systému [3]

Levá část Obrázek 2 schematicky ilustruje jednotlivé faktory, jež mají zásadní vliv na životní cyklus vývoje systému, ty zahrnují [3]:

- obchodní stránky a sílu konkurence,
- regulatorní zátěž legislativy, orgánů veřejné moci a audit,
- zvyklosti, vědomosti a praxe vně organizace,
- podoba řízení potřeb,
- firemní kulturu,
- výdajová a investiční hlediska,
- konkrétní metodiky pro vývoj a provoz softwaru.

4.1.1 Metodiky pro vývoj, testování a provoz softwaru

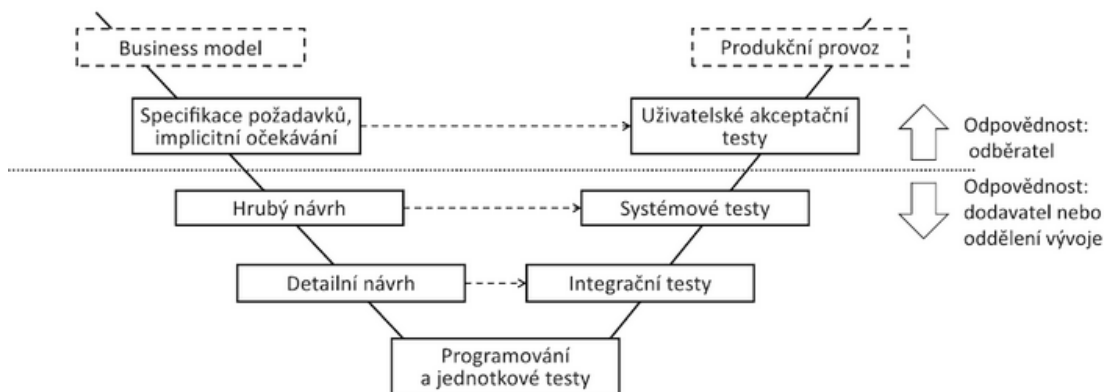
Ve spojitosti s výstavbou, testováním a nasazováním do provozu nových systémů jsou obvykle předmětem diskusí metodiky. Obecně se metodikou rozumí návody a instrukce, které pomáhají sjednotit postup v konkrétních situacích. [3]

Formálně pojem metodika definuje Jason Charvat takto: „*Metodika je souhrn vodítek a principů, které mohou být přizpůsobeny a aplikovány na řešení určité situace. Může jít jak o jednoduchý soupis akcí, které se mají vykonat, tak o specifický přístup, šablony, formuláře, popřípadě kontrolní seznamy.*“

Pro potřeby mé diplomové práce budou metodiky uvažovány ve smyslu konkrétních návodů, jak postupovat při řešení aktivit v rámci celého projektu.

Metodiky jsou interními dokumenty organizací, jež jsou připravovány s přihlédnutím k jejich specifickým potřebám. Ve většině případů ale se opírají o standardní modely vývoje systémů, mezi které patří například vodopádový nebo agilní přístup. [3]

V-model popsany již v druhé polovině 80. let 20. století je relativně oblíbeným modelem životního cyklu vývoje systému. Ve své době šlo o revoluční myšlenku, která změnila pohled na testování a pozdvihla jeho význam na úroveň návrhu a programování. Stalo se tak doplněním klasického vodopádového modelu o testovací fáze, jak je vidět na Obrázek 3. [3]

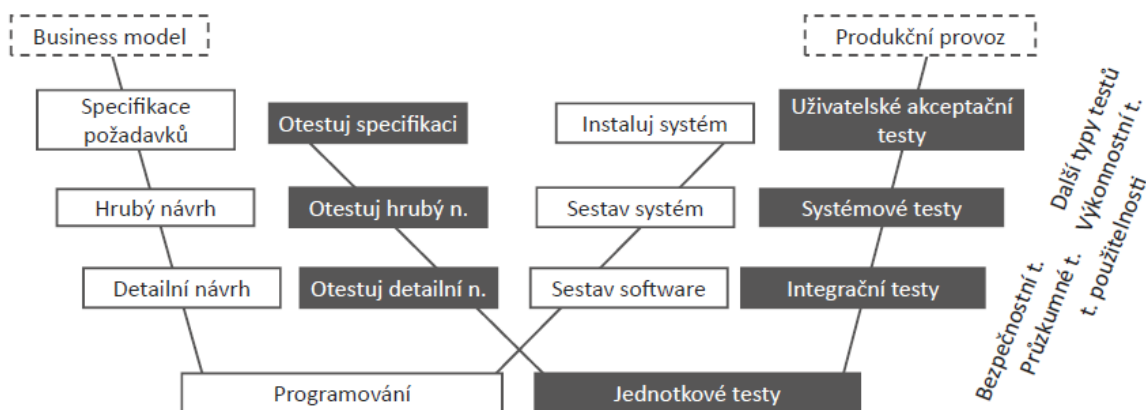


Obrázek 3: V-model [3]

V-model (na Obrázek 3) popisuje snahu vytvářet korektně navržené a vyvíjené systémy, a to realizací určitých sad testů pro danou etapu životního cyklu. [3]

Jednou z obecně známých nevýhod V-modelu je, že testování specifikací a statické testování se provádí až na konci procesu testování v rámci uživatelských akceptačních testů, což vede k pozdnímu odhalení chyb ze špatné specifikace a ve fázi akceptačního testování jsou opravy těchto chyb již velmi nákladné. [3]

V reakci na zmíněné nedostatky V-modelu je v literatuře popisován jako alternativa **W-model** znázorněný na Obrázek 4. [3]



Obrázek 4: W-model [3]

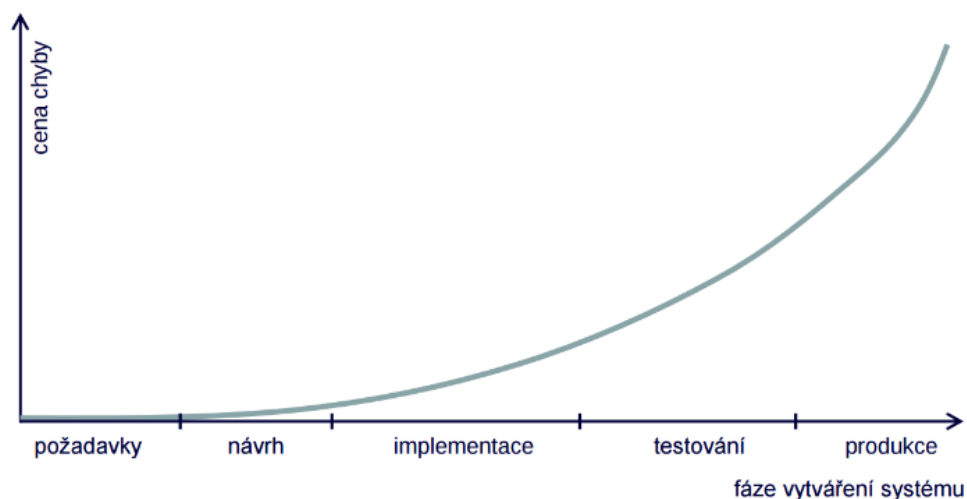
V pojetí W-modelu se v životním cyklu vývoje softwaru klade důraz na zařazení testů již ve fázi specifikace požadavků, jak je vidět na levé straně W-modelu (Obrázek 4). Na pravé straně W-modelu je pak znázorněn přímý vztah vývojových a testovacích aktivit, W-model tím podstatně otevírá příležitost podívat se na testování z širšího úhlu pohledu namísto pouhého ověřování činnosti jednotlivých komponent softwaru při spouštění. [3]

4.2 Proč je testování důležité – cena za kvalitu

Pokud by někdo netestoval vyvíjený software, nebyl by schopen odhalit značné množství chyb, které byly do softwaru zaneseny v průběhu vývoje. Zmíněná situace by způsobila značné finanční ztráty, podkopala by důvěru zákazníka a mohlo by to vést až ke ztrátám na lidských životech, jako např. v leteckém průmyslu. [1]

Téměř všechny IT projekty jsou proto provázeny diskuzemi mezi investorem, dodavatelem a manažerem testování na téma, do jaké hloubky provádět testování a s tím související otázka kolik prostředků je žádoucích, potřebných a možné vynaložit na testování. [3]

Jednoznačně je ale možné konstatovat, že nedocení testovací fáze způsobí vyšší náklady při vývoji systému a důvod je prostý. Jestliže chyby zanesené do vývoje systému již ve fázi specifikace požadavků a návrhu nejsou odhaleny co možná nejdříve, je jejich oprava tím dražší, čím později je chyba odhalena, jak definuje Boehmův první zákon (viz Obrázek 5). [4]



Obrázek 5: Boehmův první zákon [4]

4.3 Základní principy testování

Testování může ukázat, že ve vyvíjeném softwaru se nacházejí defekty, ale nemůže prokázat neexistenci defektů. Testováním se však snižuje pravděpodobnost, že v softwaru zůstanou neobjevené defekty, ale nenalezení žádného defektu stále není důkazem bezchybnosti. [5]

Vyčerpávající testování není možné, jelikož ověření všech možných kombinací vstupů a vstupních podmínek není realizovatelné s výjimkou triviálních případů. Namísto vyčerpávajícího testování by měly být k identifikování primárního předmětu testovacího úsilí použity analýza rizik a stanovení priorit. [5]

Testovací aktivity by měly v rámci životního cyklu vývoje softwaru nebo systému začít v co nejranější fázi a měly by se zaměřovat na definované cíle. [5]

Efekt shlukování defektů je možné pozorovat u malého množství modulů, které obsahují většinu defektů zjištěných v průběhu testování před uvolněním do provozu, nebo jsou zodpovědné za nejvíce provozních selhání. [5]

Pesticidním paradoxem je nazýván stav, kdy jsou stále opakovány tytéž testy, až časem stejný soubor testovacích případů nenalezne žádné nové defekty. K překonání „pesticidního paradoxu“ je nutné existující testovací případy pravidelně revidovat a upravovat a dále je potřeba napsat nové, odlišné testy na vykonání jiných částí softwaru nebo systému. [5]

Testování je závislé na kontextu, proto se testy provádějí odlišně pro různé kontexty, např. software kritický z pohledu zabezpečení se testuje jiným způsobem než firemní webové stránky. [5]

Představa o neexistenci omylů je falešná, jelikož nalezení a opravení defektů nic nevyřeší, pokud je vytvořený systém nepoužitelný a nesplňuje potřeby a očekávání uživatelů. [5]

4.4 Kvalita softwaru

Kvalitu softwaru je možné posuzovat z různých pohledů, které jsou definovány normou ISO/IEC 25010:211. Jak již bylo naznačeno výše, ověření funkčnosti je velmi důležité, ale kvalita má podstatně více rozměrů, na které není dobré zapomínat [1]:

- **funkčnost** (functionality) – jedná se o správné chování všech funkcí systému, jak je definováno funkční specifikací,
- **použitelnost** (usability) – popisuje, zda a jak lze dosáhnout požadovaného cíle, zda je systém uživatelsky přívětivý, zda se s ním dobře pracuje; přestože uživatel uvádí, že se software nechová “správně”, stále může jít o “správné chování” podle funkčních specifikací, v takovém případě může být chyba v samotných funkčních specifikacích a smlouvě, kterou je software vázán,
- **spolehlivost** (reliability) – udává, zda se software chová korektně za všech okolností, obzvláště po přetížení, po výpadku či chybě, zda tyto stavy umí odhalit a reportovat,
- **výkon** (performance) – popisuje, zda software je dostatečně rychlý a zvládne požadovaný počet současně pracujících uživatelů, nebo zda nepřiměřeně nevyužívá systémové zdroje,
- **podporovatelnost** (supportability) – uvádí, jak dobře software instaluje aktualizace, zda nemá problémy s hardwarovými a softwarovými nastaveními na straně uživatele a další vlastnosti související s údržbou a dalším vývojem softwaru,
- **bezpečnost** (security) – podává informace, zda je software i data v něm obsažená bezpečný a může být využíváný pouze oprávněnými osobami,
- **kompatibilita** (compatibility) – udává, se kterými softwary a systémy je daný produkt kompatibilní,

- **přenositelnost** (portability) – informuje, zda je daný software přenositelný mezi různými operačními systémy nebo jinými typy hardwaru.

4.5 Typy testů

Typy testů stejně jako kvalita se kategorizují podle různých pohledů. Následující rozdělení vychází z vodopádového přístupu k testování softwaru.

4.5.1 Podle fáze vývoje softwaru

Vývojářské testy (Unit testy) jsou automatizované testy obvykle vytvářené samotnými programátory v první fázi vývoje softwaru, kterými se ověřují základní komponenty ve zdrojovém kódu. Na vývojářské testy je kladen důraz především v Test Driven Development (TDD podrobněji popsáno v podkapitole 5.1.1). Na úrovni vývojářských testů je výhodné používat tzv. „test čtyř očí“, při kterém je zdrojový kód testován jiným programátorem v rámci organizace, tím je zajištěna dvojí kontrola zdrojového kódu před další fází testování. „Test čtyř očí“ tedy předchází ztrátám, jelikož opravy na úrovni vývojářských testů jsou nejlevnější. [6]

Modul testy jsou dalším typem automatizovaných vývojářských testů. Modul testy se využívají především při vývoji velkých informačních systémů pro ověření funkčnosti jednotlivých komponent systému. [1]

Integrační testy jsou vytvářené testovacími týmy po dokončení programátorských prací na zdrojovém kódu a sérii vývojářských testů nad jednotlivými částmi systému. Úkolem integračních testů je ověřit, zda bezchybně funguje komunikace mezi vnitřními moduly vyvíjeného softwaru, mezi softwarem a operačním systémem, hardwarem a dále prověřit kompatibilitu komunikačního rozhraní mezi vyvíjeným softwarem a vnějšími systémy. [6]

Funkční testy obvykle zkoumají činnost systému, tedy všechny funkce implementované v softwaru a porovnávají je s funkčními požadavky konečného uživatele. Funkční testy se dále dělí na smoke testy, regresní testy a testy použitelnosti. [1]

Regresní testy nacházejí své uplatnění především při retestování systému. Jedná se o ověřování funkcionality a chování systému, například po provedených opravách, změnách nebo po integraci nových komponent softwaru. Regresní testy se tedy zaměřují především na kontrolu činnosti těch částí systému, které nebyly předmětem úprav, a kontrolují, zda je

provedené změny negativně neovlivnily. Z podstaty regresních testů vyplývá, že jsou poměrně neměnné a pravidelně spouštěné, proto jsou tedy vhodnými kandidáty na automatizování. [6]

Smoke testy se využívají v poslední fázi integračních testů, kdy je systém již spustitelný. Jde o krátké testy kontrolující stav přípravy softwaru pro další fázi testování. Ve většině případů se provádí pouze základní kontrola implementace a instalace všech částí systému, pokrývají tedy jen stěžejní funkcionality softwaru a nejsou opakovaně aktualizovány. Z tohoto pohledu jsou stejně jako regresní testy ideálními kandidáty na automatizování. Smoke testy jsou v procesu testování velmi důležité, protože jejich zdárné dokončení je předpokladem zahájení fáze systémového testování. [6]

Systémové testy se připravují po úspěšném ukončení integrační fáze a je nutné ověřit funkčnost systému jako celku dle uživatelem definovaných funkčních potřeb. Systémové testy se provádějí na základě předem připravených testovacích scénářů, kdy se pomocí simulací ověřuje, jak se systém chová v situacích, které se mohou v průběhu provozu vzniknout. Testování systému probíhá v cyklech, po každé sérii testů jsou reportovány nalezené chyby, po jejichž opravení se spouští další série testů. Fázi systémového testování je dle její povahy možné považovat za výstupní kontrolu softwaru, z toho důvodu je systémovému testování přikládána přiměřená váha prakticky při všech procesech testování. Pokud by došlo k podcenění systémové fáze testování, byla by zásadně ohrožena spolehlivost vyvíjeného systému. [6]

Uživatelské akceptační testy (User acceptance test – UAT) jsou prováděny testovacími týmy uživatele. Po korektním dokončení testovacích fází na straně dodavatele se vyvíjený systém předává konečnému uživateli, který si v rámci svého testovacího týmu připravuje testovací scénáře a na jejich základě provádí v testovacím prostředí akceptační testy porovnáváním specifikací s dodaným softwarem. Chyby nalezené při provádění uživatelských akceptačních testů se reportují dodavateli systému, který zajistí opravu zjištěných závad a v co možná nejkratší lhůtě dodá uživateli opravenou verzi k dalšímu testování. [6]

4.5.2 Podle závislosti vnitřní struktury

Při testování pomocí tzv. „černé skříňky“ (Black-box testing) nemá tester k dispozici zdrojový kód softwaru, černou skříňku, jejíž obsah ve formě zdrojového kódu

není zvenčí viditelný. Testování pomocí černé skříňky se používá pro sledování získaných výsledků po vložení vstupních dat bez znalosti, jakým způsobem systém s daty pracuje. [6]

Naopak při testování s tzv. „**bílou skříňkou**“ (White-box testing) má testovací tým zdrojový kód k dispozici. Zná tedy vnitřní strukturu softwaru a může lépe připravit testování, pokud možno všech průchodů zdrojovým kódem, ověřit reakci systému při zadání vstupních hodnot mimo očekávaný interval a jiné testy vycházející ze znalosti zdrojového kódu. Nevýhodou testování pomocí bílé skříňky může být situace, kdy tester na základě znalosti vnitřní struktury kódu nevědomky napíše testy na míru vyvíjenému softwaru a nemusí být odhaleny některé funkční chyby dle specifikace. [6]

Občas je možné se setkat s testy tzv. „**šedé skříňky**“ (Grey-box testing), které kombinují testovací přístup černé a bílé skříňky. Princip testů šedé skříňky je možné přiblížit na příkladu, kdy vyvíjený systém je testován v uživatelském rozhraní, tedy pomocí černé skříňky, a získané výstupy z testů jsou ověřovány dotazy na server. [6]

4.5.3 Podle realizace testování

Manuální testování systému je většinou první, co si nezasvěcení spojí s pojmem testování softwaru a ve většině případů manuální testování zjednoduší pod pojem „klikání“. Podobného omylu se často dopouštějí manažeři, kteří se v problematice testování neorientují natolik, aby dokázali odhadnout kategorie testů potřebné pro daný software. I při manuálním testování systému musí být vybrán vhodný model životního cyklu systému. [6]

Automatizované testování systému přispívá k úsporám prostředků a zvyšování kvality vyvíjeného softwaru, jelikož vyvíjené systémy jsou v posledních letech čím dál robustnější a komplexnější, není možné celou funkcionalitu moderních softwarů pokrýt pouze manuálními testy, jednalo by se o velmi neefektivní a finančně náročnou činnost. Aktuálním trendem v technologickém vývoji je všechny pravidelně se opakující aktivity s minimem modifikací automatizovat. Zmíněný trend výrazně ovlivňuje i oblast testování. [6]

Automatické testy jsou vytvářeny jako krátké programy ve skriptovacích jazycích s cílem zautomatizovat výpočetně náročné testovací scénáře, které je potřeba provádět opakovaně a za stejných podmínek jako například smoke testy, regresní testy a nefunkční testy. [6]

V současnosti získává problematika automatizovaného testování stále větší oblibu a stává se čím dál vyhledávanější alternativou k všeobecně známým manuálním testům, jelikož automatizací je již možné pokrýt nejen provádění jednotlivých testů, ale i některé komponenty, případně celý testovací proces vyvíjeného softwaru. Na trhu jsou již dostupné nástroje (viz podkapitola 5.2), které umožňují porovnávat výsledky prováděných testů, instalovat a konfigurovat vyvíjený software nebo vytvářet testovací scénáře na základě provedených analýz. [6]

Realizace automatizovaných testů vyžaduje zajištění základních podmínek v procesu testování již v průběhu návrhu a vývoje systému, protože vytvářet automatizované testy pro software až v provozní fázi je problematické. [6]

5 TECHNIKY A NÁSTROJE PRO TESTOVÁNÍ ZÁSUVNÉHO MODULU PRO QUANTUM GIS

5.1 Metodiky pro obecnou podporu testovacího cyklu

5.1.1 TMap

TMap je od roku 1995 předním testovacím přístupem společnosti Sogeti, která drží krok s neustále se měnícími podmínkami v podnicích a technologiích. V souladu s touto politikou inovací se produkt TMap vyvinul v sadu TMap Suite. [7]

TMap Suite je znalostní báze pro profesionální testování budoucnosti, která poskytuje veškeré návody potřebné pro plnění testovacích a kvalitativních strategií v prostředí specifických informačních technologií. [7]

Nejnovějším přírůstkem v sadě TMap Suite je TMap HD – Human Driven. Tento přístup k testování je řízen kvalitou, aby splňoval politiky moderních agilních organizací. TMap HD je podrobně popsán v publikaci „Neil's Quest for Quality“. [7]

Kromě tištěných publikací společnosti Sogeti vytvořila dynamičtější formu TMap Suite v podobě webových stránek TMap.net, které obsahují všechny informace o testování se sadou TMap Suite, mj. stavební bloky pro sestavení vlastní testovací metody, ale také blogy, šablony a mnoho souborů ke stažení. [7]

Společnosti Sogeti se pomocí komunitní platformy na TMap.net snaží o centralizaci aktuálních poznatků z oblasti testování na jedno místo, aby byl zajištěn kontinuální rozvoj testování a požadavků na kvalitu. [7]

TMap NEXT je přístup ke strukturovanému testování publikovaný již v roce 2006, v průběhu let se stal standardem testování pro mnoho procesně orientovaných organizací. V současné době je realizován ve stovkách institucí po celém světě. Síla TMap může být do značné míry ovlivněna sdílením mnoha praktických zkušeností z reálné praxe, což může být cenným nástrojem pro současné i budoucí výzvy v oblasti testování. [8]

Využívání TMap NEXT přístupu k testování má následující výhody [8]:

- poskytuje přehled o všech rizicích, pokud jde o kvalitu testovaného systému,
- detekuje závady v rané fázi vývoje,
- eliminuje závady,

- zkracuje celkovou dobu realizace, jelikož testování na kritické cestě celkového vývoje je co nejkratší,
- testované produkty (např. testovací případy) jsou opakovaně použitelné,
- testovací proces je srozumitelný a zvládnutelný pro celý projektový tým.

Specifický obsah TMap NEXT strukturovaného testovacího přístupu lze shrnout do čtyř základních pilířů, a to [8]:

- je založen na přístupu business-driven test managementu (dále jen „BDTM“),
- popisuje strukturovaný testovací proces,
- obsahuje kompletní sadu nástrojů,
- je metodou adaptivního testování.

První pilíř přímo souvisí se skutečností, že obchodní příležitosti informačních systémů jsou pro organizace stále důležitější. Přístup BDTM poskytuje obsah celého procesu, proto může být považován za hlavní vodítko pro druhý pilíř, tedy strukturovaný testovací proces, k jehož popisu se v TMap používá model životního cyklu. Zároveň musí být nastaveny různé aspekty v oblasti infrastruktury, technik a organizace tak, aby proces testování probíhal korektně. Třetím nezbytným pilířem TMap je kompletní sada nástrojů pro testování ve formě příkladů, kontrolních seznamů, popisů technik, postupů, organizační struktury testů a testovací prostředí. Čtvrtý neméně důležitý pilíř TMap je charakteristický flexibilitou, jež umožňuje implementaci nezávislou na fázi vývoje systému. Jinými slovy TMap lze implementovat, jak ve fázi vývoje nového systému, údržby systému nebo doplněného modulu i pro outsourcing testovacího procesu. [8]

5.1.2 ISTQB

International Software Testing Qualifications Board (dále jen „ISTQB“) je mezinárodní organizací založenou v roce 2002 za účelem sjednotit skupinu expertů v oblasti testování systémů v celosvětově uznávanou certifikační autoritu. ISTQB k prosinci 2018 spravuje více než 830 000 zkoušek a vydalo více než 605 000 certifikací ve více než 120 zemích světa. [5]

Systém ISTQB se opírá o znalostní bázi a po celém světě jednotně uplatňovaná zkušební pravidla, přičemž studijní materiály a zkoušky jsou k dispozici v mnoha jazycích. Jednou z aktivit ISTQB je publikace terminologického slovníku z oblasti testování

systémů, který je mezinárodně uznávaným standardem. Dále zajišťuje certifikaci profesionálních testerů ve třech úrovních, a to základní, pokročilá a expertní. [5]

Základní certifikace je zaměřena na profesionály, kteří potřebují prokázat praktické znalosti základních konceptů testování softwaru, jako jsou testovací designéři, analytici testů, testovací inženýři, testovací poradci, testovací manažeři, testovací uživatelé a IT profesionálové. Dále je vhodná pro ty, kteří potřebují základní znalosti o testování softwaru, jako jsou projektoví manažeři, manažeři kvality, manažeři vývoje softwaru, obchodní analytici, IT ředitelé a konzultanti v oblasti řízení. [5]

Pokročilá úroveň se zaměřuje na profesionály, kteří potřebují hlubší porozumění testování softwaru, jako jsou analytici testů, testovací inženýři, testovací poradci, testovací manažeři, vývojáři softwaru, projektoví manažeři, manažeři kvality a manažeři vývoje softwaru. Pro získání pokročilé úrovně certifikace musí uchazeči vlastnit certifikát základní úrovně a vyhovět zkušební komisi, která zkoumá, zda mají dostatečné praktické zkušenosti, aby mohli být kvalifikováni jako pokročilí. [5]

Expertní úroveň rozšiřuje znalosti a zkušenosti získané na pokročilé úrovni poskytováním detailních, prakticky orientovaných certifikací v řadě různých testovacích případů. [5]

V České republice je ISTQB zastoupeno neziskovou organizací Czech and Slovak Testing Board (dále jen „CaSTB“), která činnost ISTQB podporuje především překlady studijních materiálů a slovníku ISTQB. [5]

5.2 Nástroje pro obecnou podporu testovacího cyklu

V následující podkapitole jsou uvedeny příklady tří skupin nástrojů, kterými jsou nástroje pro správu scénářů, reportování chyb a nástroje pro automatizaci testů.

5.2.1 Správa scénářů, reportování chyb

Na trhu je dostupných více nástrojů pro správu scénářů a reportování chyb, ať už se jedná o komerční nebo open source produkty. Pro bližší představení možností správy scénářů a hlídání chyb jsem vybrala dva zástupce komerčních systémů, a to HP ALM a QAComplete.

HP ALM (Application LifeCycle Management) je komerčním a poměrně finančně náročným produktem společnosti Hewlett Packard Enterprise distribuovaným jako sada

aplikačních nástrojů pro správu a automatizaci testovacích úloh a reportování chyb, která je vhodná především pro správu rozsáhlých projektů. Pomocí nástroje HP ALM lze efektivně řídit celý životní cyklus softwaru od plánování projektu, specifikaci požadavků až po testování, implementaci a provoz. [9]

Nasazení HP ALM je praktické ve velkých a středních organizacích pro účelnou koordinaci početných týmů, ať už se jedná o interní, distribuované či outsourcované týmy vývojářů, a pro správu dynamicky se vyvíjejících systémů. Technologie HP ALM je postavena na standardizované architektuře, kterou je možné přizpůsobit a rozšířit pomocí modulů dle potřeb jednotlivých organizací. [9]

Mezi hlavní výhody řízení požadavků pomocí HP ALM patří [9]:

- propojení a strukturování testovacích scénářů na importované modely obchodních procesů a vyhodnocování pokrytí cesty,
- využívání konfigurovatelných šablon,
- vytváření testovacích scénářů v HP ALM pomocí editoru formátovaného textu, který připomíná uživatelsky známé prostředí Microsoft Word,
- prohlížení seznamu testovacích scénářů v jednom pohledu, což umožňuje jejich rychlou editaci,
- sledování vztahů mezi testovacími scénáři, pokrytím testy, procesy a defekty,
- propojení všech testovacích scénářů s testy, odhalenými chybami a reportovanými úlohami vývojářů.

SmartBear QAComplete je dalším komerčním nástrojem vyvíjeným společností SmartBear pro správu testovacího procesu, který umožňuje plánovat, koordinovat a shromažďovat všechny testovací scénáře, testy a reporty odhalených chyb související s vývojem a provozem, a to všechno na jednom místě. Jako webová aplikace může přistupovat k datům kdekoli, kde je připojení k internetu. [10] [11]

Nástroj QAComplete usnadňuje správu testovacích scénářů díky integraci s běžně používanými nástroji pro vývoj a testování systémů, například Visual Studio, JIRA Software, HP UFT, Bugzilla, Jenkins, GitHub a jinými oblíbenými ručními i automatizovanými testovacími nástroji. [10] [11]

QAComplete umožňuje variabilní nastavení pracovních postupů dle individuálních potřeb jednotlivých organizací. Používáním integrovaných šablon nebo vytvořením

vlastních pracovních postupů umožňuje spravovat, organizovat a reportovat výsledky manuálních i automatizovaných testů na jednom centrálním místě. A dále pak vytvářet libovolnou hierarchii projektů, včetně verzí, sprintů a sestav. [10] [11]

QAComplete obdobně jako výše popsany nástroj HP ALM propojuje testovací scénáře s požadavky, odhalenými chybami a reportovanými úkoly pro zajištění přehlednosti a kontroly o aktuálním stavu jednotlivých požadavků ze strany všech zainteresovaných. [10] [11]

QAComplete je dostupný jako softwarový nástroj, cloudová služba nebo tzv. „On-Premises“ pro ty, kteří chtějí hostování webové aplikace a databáze na vlastních serverech. [10] [11]

Mezi hlavní výhody řízení požadavků pomocí QAComplete patří [10] [11]:

- přizpůsobení nástroje konkrétní metodice vývoje,
- funkce centrální křižovatky pro celý proces testování,
- koordinuje manuální i automatizované testy připravované ve více prostředích,
- pomocí konektoru pro JIRA přímá aktualizace v JIRA,
- rychlejší testování díky identifikaci kandidátů na automatizování testů,
- koordinace testovacích verzí a cyklů.

5.2.2 Reportování chyb

Ze skupiny nástrojů pro správu a reportování chyb jsem zvolila tři následující příklady, z toho jeden komerční a dva open source.

JIRA je softwarovým nástrojem, který přivedla na trh australská společnost Atlassian, používaným pro odhalování chyb, problémů a řízení projektů. [12]

JIRA byl vytvořen v programovacím jazyce Java a je kompatibilní s programy pro řízení zdrojů, jako jsou Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial a Perforce. Dodává se s různými překlady včetně angličtiny, japonštiny, němčiny, francouzštiny a španělštiny. [9]

Software JIRA nabízí mnoho užitečných funkcionalit pro sledování a koordinaci činností, rozdělování práce v týmu a kontrolu postupu týmových aktivit. [12]

Mezi hlavní výhody sledování a reportování chyb pomocí JIRA patří [9]:

- odhalení a přehledná organizace chyb,
- rozdělení práce a dozor nad činnostmi v týmech,
- možnost prioritizace úkolů a přizpůsobení fronty aktuálním požadavkům,
- možnost propojení nástroje JIRA s GitHub Enterprise pomocí volného konektoru DVCS.

Mantis BUG TRACKER (dále jen MantisBT) je dalším nástrojem pro reportování chyb vyvíjený stejnojmennou společností jako open source produkt. MantisBT je webovým systémem postaveným na PHP s podporou Linux, Windows i macOS na straně serveru. MantisBT je kompatibilní s běžně využívanými webovými prohlížeči jako například Firefox, Chrome, Internet Explorer 10+, Opera a Safari. [13]

MantisBT si získal velkou oblibu mezi uživateli především pro svou vyváženost mezi výkonem a jednoduchostí implementace a provozu. [13]

Redmine je další z řady open source webových nástrojů pro správu chyb a řízení projektů. Nabízí funkcionality pro správu více projektů, řízení aktivit a přístupů na základě přidělených rolí v projektu. Podrobné informace, tipy a triky, jak pracovat s Redmine je možné získat například z publikací Mastering Redmine, Redmine Plugin Extension and Development nebo Redmine Cookbook. [14]

5.2.3 Nástroje pro automatizaci testů

Nástrojů pro automatizaci testů je celá řada jak komerčních, tak distribuovaných jako open source. Z toho důvodu byl pro účely této práce použit jako vodítko k výběru vhodných nástrojů elektronický článek Top 10 nástrojů pro automatizaci testů 2019 publikovaný 11. ledna 2019 na stránkách www.testbytes.net, který stručně představuje žebříček deseti nejpoužívanějších systémů pro automatizaci testů. [15]

Z deseti jmenovaných nástrojů v žebříčku Top 10 nástrojů pro automatizaci testů 2019 jsem vybrala následující tři, které by mohly být využity pro potřeby testování modulu PÁTRAC.

TestComplete je komerčním produktem společnosti SmartBear, který umožňuje automatizovat testy při vývoji webových, mobilních i desktopových systémů.

TestComplete je možné zakoupit jako samostatný nástroj nebo součást balíčku společnosti SmartBear, který poskytuje komplexní řešení pro podporu testování. [15]

Software podporuje řadu skriptovacích jazyků mimo jiné JavaScript, VBScript, Python a další. Vytváření testů v TestComplete je poměrně intuitivní a jednotlivé testy lze dokumentovat krok za krokem, pomocí prioritních operací nebo psaním skriptů. Nástroj obsahuje integrovaný rekordér sloužící k nahrávání testovacích scénářů. [6]

Připravené testy se spouští manuálně nebo automatizovaně, průběh testu je zaznamenáván printscreeny pro následné porovnání a odhalené chyby se zapisují do logů. [6]

Nástroj TestComplete je možné rozšířit o nabízené moduly, např. modul pro identifikaci komponent dynamického uživatelského rozhraní, jehož využití je účelné především u aplikací s rychle se vyvíjejícím uživatelským rozhraním. [15]

Ranorex Studio nabízí komplexní sadu nástrojů pro vytváření sofistikovaných automatizovaných testů desktopových, webových případně mobilních aplikací. Práce s nástroji Ranorex je poměrně jednoduchá a nevyžaduje mimořádné programátorské dovednosti. Součástí softwaru Ranorex je integrovaná metodika. [16]

Dále Ranorex poskytuje aktuální informace o průběhu testování, včetně grafů a podrobných reportů odhalených chyb, a to ve srozumitelné formě pro testery, vývojáře i projektové managery. Součástí zakoupené licence Ranorex Studio je profesionální podpora a zajištění školení dle potřeb organizace. [16]

HPE Unified Functional Testing (dále jen „UFT“) je mezi řadou odborníků považován za standard automatizace funkčního testování podporující nejrozličnější platformy jako například SAP, Java, Delphi, Oracle, Net, Mobile, PowerBuilder, Visual Basic a mnoho dalších. UFT nabízí velmi sofistikovaný přístup k automatizaci testů, a to pomocí opětovně využitelných částí automatizovaných testů, inteligentní identifikace objektů, automatického generování dokumentace a efektivní správy chyb. [15]

UFT je postaven pro podporu vývoje software nad Windows a používá Visual Basic Scripting Edition k evidenci testovacích procesů a řízení objektů. UFT umožňuje automatizovat testování vyvíjeného systému, ať už se jedná o desktop, webový software, mobilní aplikaci. [15]

5.3 Techniky pro vytvoření testovacích scénářů

Pro vytváření testovacích scénářů je možné využít řadu technik, které jsou popsány v odborné literatuře, například v publikaci TMap NEXT® - For Result-Driven Testing, případně v Introduction to software testing. [8] [17]

Mezi těmito technikami je možné identifikovat dvě základní, které do jisté míry představují i základ ostatních testovacích technik. V první základní technice se jedná o vytváření sledů akcí v testovaném systému, kterou v této práci budu nazývat technikou procesního testování. Druhou ze základních technik je vytváření kombinací testovacích dat nebo situací pro testování, které budu v této práci říkat kombinatorické techniky.

Příklady dalších technik je možné najít například ve dvou uvedených publikacích, kterými jsou TMap next: for result-driven testing a Introduction to software testing. [8] [17]

Principem procesního testu je vykonání určité sekvence akcí v testovaném systému. Aby bylo vynaložené úsilí optimální, je vhodné vytvořit model testovaného procesu a testovací scénáře automaticky vygenerovat. K tomu je možné využít různé moderní algoritmy, které jsou detailně popsány například v odborném článku More efficiency in testing of business processes and workflows. [18]

Principem kombinatorického testu je vytvoření optimálního počtu kombinací testovacích dat, které poskytují co nejvyšší pravděpodobnost odhalení důležitých chyb. Přehled používaných metod poslední generace je možné najít v publikacích Constrained Interaction Testing: A Systematic Literature Study a Introduction to combinatorial testing. [19] [20]

Metoda pracuje s identifikací parametrů, které spolu vytváří v systému vzájemnou vazbu, a proto tyto související parametry je připravována vyšší intenzita testovacích kombinací. Podobně jako v případě procesních testů je nejvhodnější tyto kombinace generovat automaticky vhodným algoritmem. V této oblasti se často využívá metod umělé inteligence. Příklady konkrétních algoritmů lze nalézt v práci A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problém. Výsledky metody byly několikrát nezávisle ověřeny. [20] [21]

6 METODIKA TESTOVÁNÍ ZÁSUVNÉHO MODULU PÁTRAČ

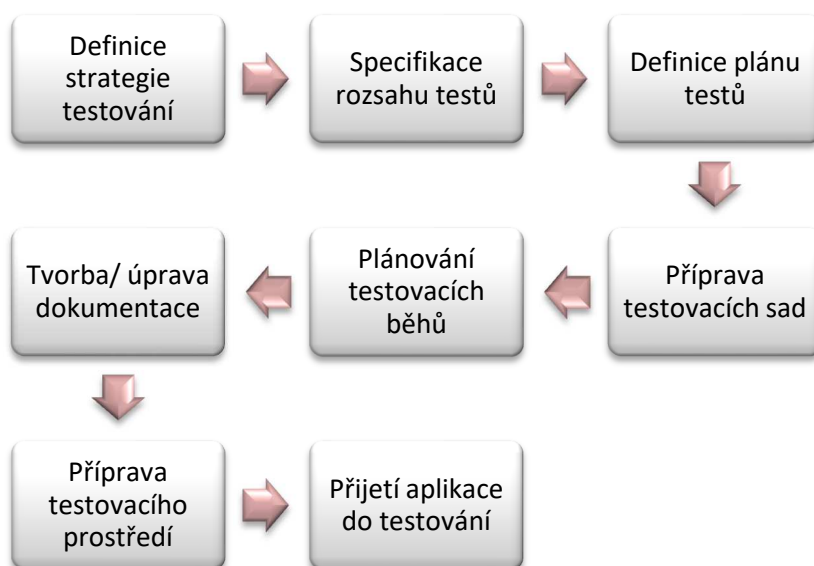
Hlavním cílem vytvářené metodiky testování modulu PÁTRAČ je stanovit strukturovaný model aktivit a odpovědností, který povede k zajištění kvality, tj. mimo jiné připravit plán testování. Tento model zahrne požadavky na přípravu a ověření shody a provedení akceptačních testů a umožní sledovat a kontrolovat provedení a výsledky příslušných aktivit. V rámci předložené metodiky budou stanoveny postupy a pravidla, jak pro interní testy, tak i pro ověření shody a akceptační testy.

Prakticky se základní kroky procesu testování dají shrnout do dvou základních celků:

- plánování a příprava testů,
- provedení a vyhodnocení testů.

6.1 Plánování a příprava testů

Proces plánování a přípravy testů je možné shrnout do osmi základních fází znázorněných diagramem na Obrázek 6.



Obrázek 6: Fáze plánování a přípravy testů

6.1.1 Definice strategie testování

Strategie testování definuje priority a způsob testování modulu PÁTRAC. V rámci této aktivity jsou stanoveny klíčové ukazatele pro následné hodnocení testování. Součástí této fáze je i analýza rizik, která pomáhá identifikovat funkce nebo vlastnosti modulu PÁTRAC, které nesou nejvyšší míru rizika při nasazení do provozu. Na základě těchto poznatků se následně plánuje rozsah testů a priority testovacích aktivit.

Strategie testování obsahuje zejména:

- cíle testů,
- specifikace oblastí testování, vč. negativního vymezení,
- předpoklady plánu testů, vč. specifikace součinnosti:
 - technické (připravenost),
 - organizační (součinnost),
- produkty plánu testů:
 - manažerské (řídící dokumenty plánu testů),
 - testovací prostředí a konfigurace,
 - testovací nástroje a konfigurace,
 - definice rizik pro testování,
 - přístup k přípravě testovacích scénářů a testovacích dat,
 - automatizované testy
- nástroje:
 - správa manažerských produktů plánu testů,
 - správa uživatelských a testovacích scénářů a dat,
 - správa automatizovaných testů,
 - evidence a řízení životního cyklu defektů (neshod kvality),
- matice odpovědnosti jednotlivých komponent plánu testů.

6.1.2 Specifikace rozsahu testů

Během specifikace rozsahu testů je stanoveno, jak jsou požadavky na testy realizovány. V rámci této aktivity jsou pro potřeby modulu PÁTRAC navrženy konkrétní typy testů, které budou prováděny a je určeno jaké nástroje, data a prostředí jsou použity pro ověření daných kritérií.

6.1.3 Definice plánu testů

Účelem definice plánu testů je jasná deklarace testovacích aktivit pro ověření kvality vyvíjeného modulu PÁTRAC.

V rámci plánu testování bude vypracován detailní plán testovacích aktivit. Tento plán zahrne zejména:

- rozvrh jednotlivých fází testování a testovacích aktivit,
- rozpad testovacích aktivit do jednotlivých činností podle jejich charakteru (například pokud je vykonávají různé role),
- rozsah testů (typy testů),
- přiřazení pracovníků k jednotlivým plánovaným aktivitám,
- analýzu závislostí mezi plánovanými aktivitami (například když konec jedné aktivity je předpokladem pro začátek druhé).

Každý z testů je popsán následujícím způsobem:

- ID testovacího scénáře v návaznosti na katalog požadavků,
- název testu,
- priorita testu (kategorie A, B, C),
- shrnutí testu
- popis testu (scénář),
- specifické vstupní a výstupní podmínky testu, pokud existují,
- případné předpoklady a rizika testovacího scénáře,
- testovací data,
- informace o tom, zda v rámci daného testu probíhá komunikace s okolními systémy,
- očekávaný výstup,
- datum vytvoření/ modifikace testovacího scénáře,
- verze testovacího scénáře,
- příloha k testovacímu scénáři (například vlastní testovací skript, proprietárně připravená data apod.)

a řídí se strategií testování.

6.1.4 Příprava testovacích sad

Příprava testovacích sad bude spočívat ve vytváření testovacích sad pro jednotlivé typy testů. Pod pojmem testovací sada rozumíme skupinu testovacích scénářů, které se provádějí se specifickým cílem a souvisí s konkrétní skupinou funkcí modulu PÁTRAC, oblastí pokrytí nebo problematikou. Testovací sada může obsahovat například soubor testovacích scénářů pro ověření správné funkce konkrétního procesu v aplikaci. Testovací sady mohou být součástí manuálně nebo automatizovaně prováděných testů.

6.1.5 Plánování testovacích běhů

Plánování testovacích běhů zahrnuje definici podmínek pro spouštění jednotlivých testů modulu PÁTRAC, včetně data a času jejich provedení.

6.1.6 Tvorba/ úprava dokumentace

Pro danou testovací techniku (manuální funkční testování, automatizované funkční testování) jsou určeny požadavky na testování a následně je vytvářena testovací dokumentace (testovací scénáře, automatizované testovací skripty apod.) na základě, které budou vykonávány. V rámci vytváření testovací dokumentace budou určena pravidla spouštění testů. Součástí bude také příprava testovacích dat.

Cílem je připravit testovací scénáře, které sníží pravděpodobnost existence chyby aplikace na minimum. Pro každý funkční aplikační požadavek a relevantní nefunkční požadavek bude vytvořen testovací scénář.

Prvním krokem přípravy testovací dokumentace je seznámení se s analytickou dokumentací aplikace.

Druhým krokem je vytvoření testovacích scénářů na základě požadavků na aplikaci. Pro každý funkční aplikační požadavek a relevantní nefunkční požadavek je vytvořen testovací scénář.

6.1.7 Příprava testovacího prostředí

Před každým spuštěním testů je zapotřebí vybrat a připravit testovací prostředí (podle specifikovaných požadavků, softwarové architektury, nastavení produkčního prostředí apod.), připravit testovanou aplikaci k testování (rezervace, instalace příslušných aplikací, konfigurace systémů apod.).

6.1.8 Přijetí aplikace do testování

Modul PÁTRAC se do testování přebírá na základě procesu testování, který je uveden v plánu testů.

6.2 Provedení a vyhodnocení testů

6.2.1 Provedení testů

V rámci této aktivity dojde k provedení všech naplánovaných testů a zaznamenání výsledků.

6.2.2 Typy testů a jejich výstupy

Navrhované typy testů pro potřeby vyvíjeného modulu PÁTRAC lze rozdělit do následujících základních typů:

- interní testy, které budou v závislosti na charakteru testované části zahrnovat:
 - vývojové, jednotkové testy,
 - funkční testy (manuální a automatizované),
 - testy komponent (řeší i testy výjimek a hraniční stavy),
 - vývojové integrační testy,
- systémové testy budou zahrnovat:
 - funkční testy,
 - integrační testy,
 - regresní testy,
 - smoke testy,
- akceptační testy budou zahrnovat:
 - funkční testy,
 - integrační testy,
 - bezpečnostní testy,
 - testy odolnosti proti výpadku,
 - testy obnovy systému,
 - testy migrace,
- testy po nasazení do produkčního prostředí budou zahrnovat:
 - regresní testy,
 - smoke testy.

6.2.3 Reporting chyb

V případě zjištění odlišností od očekávaného výsledku testů dojde k zaznamenání zjištění do nástroje pro evidenci defektů.

6.2.4 Retesty opravených chyb

Defekty, které byly označeny za opravené, jsou ověřeny, znovu otestovány, zda byly skutečně opraveny. Opravou a retestem všech chyb není zaručeno, že je výsledný systém bez chyb. Opravou defektů mohou být do vyvíjené aplikace zavlečeny nové chyby. Pokud je tedy opravených chyb více, je prováděno opakované spuštění kompletní sady testů.

6.2.5 Vyhodnocení testů

U každého cyklu testování modulu PÁTRAC bude provedeno vyhodnocení výsledků testů na základě, kterého se rozhodne o ukončení testování nebo zahájení dalšího testovacího cyklu. Vyhodnocování výsledků testů bude spočívat v analýze reportů a výsledků testů po ukončení jednotlivých testovacích běhů. Pro jednotlivé kroky se bude hodnotit, zda je výsledek testů úspěšný nebo neúspěšný.

7 NÁVRH TESTOVACÍCH SCÉNÁŘŮ PRO ZÁSUVNÝ MODUL PÁTRAČ

Po dohodě s vedoucím práce jsem obecnou testovací strategii prezentovanou v této práci rozvedla do detailu v jedné konkrétní oblasti, a to určení jednotlivých obecných testovacích scénářů (variant situací), jež je potřeba při pátrání otestovat a které budou předmětem zájmu této kapitoly. Testovací scénáře budou automaticky generovány pomocí aplikace kombinatorických technik popsaných v podkapitole 5.3.

7.1 Analýza parametrů úlohy

V přípravné fázi byly identifikovány dva základní typy vstupních parametrů:

7.1.1 Terén

Na základě použitých statistických dat bylo konstatováno, že pohřešovaná osoba se ve volném terénu pohybuje nejpomaleji při chůzi do kopce, rychleji po rovině a nejvyšší rychlosti dosahuje při chůzi z kopce.

Sklon svahu, který se pro účely algoritmu použitého pro výpočty pohybu pohřešované osoby v terénu považuje za neprůchozí, je 45° .

Při výpočtech výše uvedeného algoritmu se zároveň považují za nepřekonatelné překážky dálnice, silnice 1. třídy a budovy.

V následujících studiích bude nutné zahrnout analýzy vodních ploch a toků z hlediska prostupnosti. V aktuální verzi PÁTRAČE se s vodou pracuje jako s neprůchodnou překážkou. Zároveň by dle mého názoru stálo za zvážení pracovat v analýzách pohybu i s prostupností vegetačního porostu, a to ve smyslu přidělení vah dle náročnosti průchodu jednotlivými druhy vegetace.

7.1.2 Parametry dle uživatele

Jedním ze základních vstupů do PÁTRAČE je lokalizace posledního spatření pohřešované osoby. Zaznamenávanými informacemi o posledním spatření jsou GPS souřadnice jednoho nebo více míst spatření, čas a váha svědectví o posledním spatření na stupnici 0 – 9 (0 – nedůvěryhodné, 9 – potvrzené).

Volitelným parametrem pro výpočet oblasti možného výskytu pohřešované osoby je směr pohybu. Velitel zásahu má na základě dostupných informací o lokalizaci

posledního spatření pohřešované osoby možnost zvolit, zda bude do výpočtu směr pohybu zahrnut nebo se s ním dále nebude pracovat.

Velitel zásahu na základě vyhodnocení dané situace zahrne do výpočtu parametr míry pravděpodobnosti, že se pohřešovaná osoba nalezne v oblasti, kterou PÁTRAČ vygeneruje. Při zadání vyšší pravděpodobnosti nálezu pohřešované osoby (např. 90 %) se výrazně zvyšuje rozloha PÁTRAČEM generované oblasti, kterou je nutné propátrat.

Velitel zásahu volí jako další parametr pro pátrání po pohřešované osobě dostupné prostředky, které jsou aktuálně k dispozici, jmenovitě se jedná o počet psů s psovody, počet osob do rojnice a eventuálně o potápěče. Do následující verze PÁTRAČE bude mezi dostupné prostředky pro pátrání doplněn vrtulník. Na základě dostupných prostředků systém odhaduje dobu, která bude nezbytná pro propátrání oblasti pravděpodobného výskytu pohřešované osoby.

Mohou ale nastat situace, kdy je v zájmu ochrany života a zdraví nutné pohřešovanou osobu nalézt v co možná nejkratší době, jako příklad lze uvést pátrání po malém dítěti nebo diabetikovi, který se ztratil a je předpoklad, že u sebe nemá inzulín. Pro řešení této mimořádné události se musí postupovat opačným způsobem, než bylo popsáno v předchozím odstavci, kdy známe dostupné prostředky. Je tedy znám čas, do kdy musí být pohřešovaná osoba nalezena, aby se předešlo újmě na zdraví, a PÁTRAČ spočítá, jaké prostředky jsou potřebné k nalezení pohřešovaného například do 4 hodin od nahlášení.

7.2 Identifikované parametry testované úlohy

Analýzou parametrů vyplývajících z charakteru terénu a dalších parametrů záviselých na rozhodnutí uživatele, které ovlivňují rozdělení území na jednotlivé sektory a generování výsledné oblasti určené pro pátrání, byly identifikovány následující klíčové parametry, jež budou předmětem testování.

7.2.1 Členitost terénu

V podkapitole 7.1.1 bylo popsáno, že na základě statistických analýz rychlost pohybu pohřešované osoby ve volném terénu závisí na svažitosti terénu. Pro potřeby testování je sklon terénu rozdělen do tří intervalů, a to:

- za směr z kopce dolů je považován sklon od -8° do -45° ,
- za rovinu je považován terén se sklonem od -7° do 7° ,

- za směr nahoru do kopce je považován sklon od 8° do 45°.

7.2.2 Výskyt neprůchozích částí v terénu

V podkapitole 7.1.1 byly identifikovány překážky, které jsou pěší chůzí v terénu nepřekonatelné, a PÁTRAC by průchod přes tyto překážky měl vyloučit. Neprůchozí části v terénu jsou velmi podstatným vstupem do výpočtů, proto budou zahrnuty do testování, jedná se o:

- sklon svahu menší nebo větší než 45°,
- dálnice,
- silnice 1. třídy,
- budovy,
- vodní plochy,
- vodní toky.

7.2.3 Poslední spatření

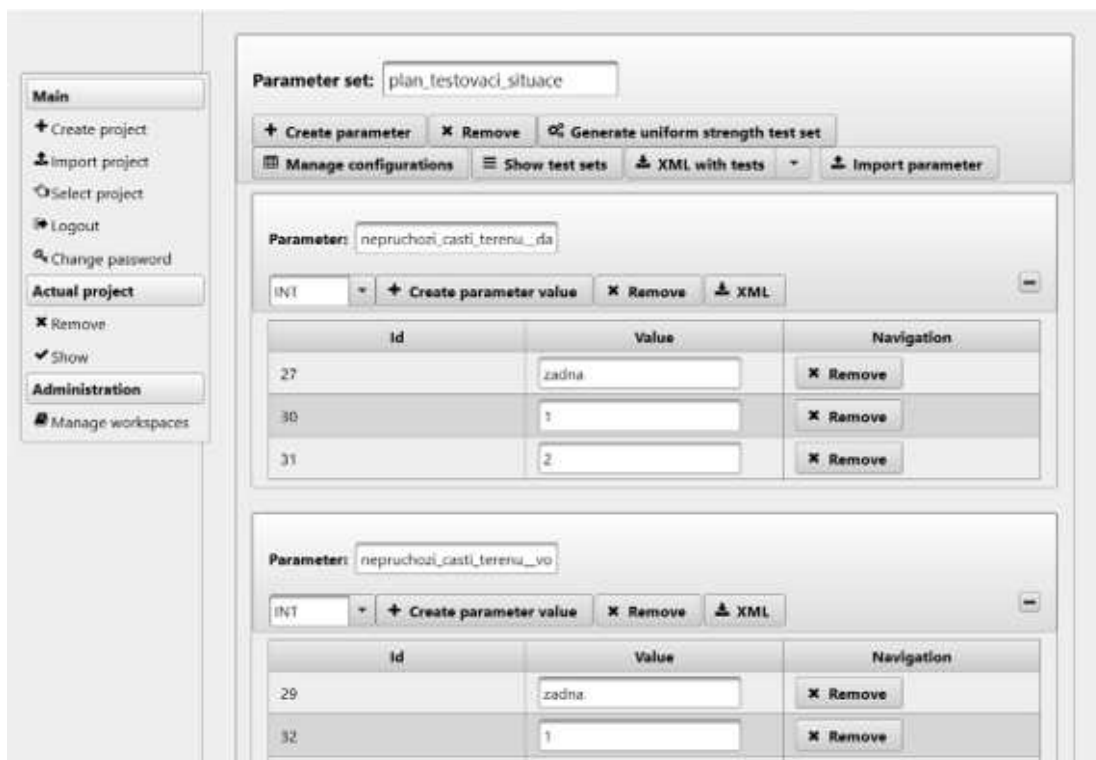
Jedním ze základních vstupů do modulu PÁTRAC je lokalizace posledního spatření pohřešované osoby. Zaznamenávanými informacemi o posledním spatření jsou GPS souřadnice jednoho nebo více míst spatření, čas a přidělená váha svědectví o posledním spatření na stupnici 0 – 9 (0 – nedůvěryhodné, 9 – potvrzené).

7.2.4 Směr pohybu, zda ano/ ne

Velitel zásahu má na základě vyhodnocení aktuálních informací o pohřešované osobě možnost použít volitelný parametr, zda má být do výpočtu oblasti možného výskytu pohřešované osoby zahrnut směr pohybu nebo se může rozhodnout, že například ze dvou lokalizací posledního spatření nelze odhadovat směr pohybu a vyloučí jej z výpočtu.

7.3 Výchozí testovací plán

Pro vytvoření testovacího plánu s využitím kombinatorických technik byl použit nástroj Combinatorial testing tool (dále jen „CTT“) vyvíjený na univerzitě ČVUT v Praze. Ukázka testovacího prostředí je na Obrázek 7.



Obrázek 7: Combinatorial testing tool

Generování testovacích scénářů probíhalo ve dvou etapách nazvaných SET 1 a SET 2. Pro obě etapy byly vygenerovány sady testovacích scénářů ve třech intenzitách síly testů.

7.3.1 SET 1: Členitost terénu

V první etapě byly pro výpočet možných kombinací vloženy parametry vztahující se k terénu krajiny s následujícími hodnotami:

- plocha malých kopců $8^\circ - 20^\circ$: žádný, 0 % – 20 %, 20 % – 40 %, 40 % – 60 %,
- plocha velkých kopců $21^\circ - 45^\circ$: žádný, 0 % – 20 %, 20 % – 40 %,
- neprůchozí části terénu:
 - kopce nad 45° : žádný, 1, 2, 3, 4,
 - dálnice – žádná, 1, 2,
 - silnice 1. třídy: žádná, 1, 2,
 - vodní plocha – žádná 1, 2,
 - budovy – žádná, 1, 2, 3, 4, 5.

Výsledky pro SET 1:

V první etapě byly do kombinací zahrnuty pouze parametry vycházející z členitosti terénu, a přesto nástroj CTT vypočítal 9 720 možných kombinací parametrů, pokud by nebyly využity kombinatorické techniky.

Tabulka 1: SET 1 – střední intenzita síly testů

Neprůchozí části terénu					Členitost terénu	
dálnice	vodní plocha	silnice 1. třídy	svah $\pm 45^\circ$	budovy	plocha malých kopců $8^\circ - 20^\circ$ v %	plocha velkých kopců $21^\circ - 45^\circ$ v %
1	1	1	žádný	žádná	0 - 20	0
2	2	2	1	žádná	20 - 40	20 - 40
žádná	žádná	žádná	2	žádná	40 - 60	0 - 20
1	2	žádná	3	žádná	0	0
2	žádná	1	4	žádná	0 - 20	20 - 40
žádná	1	2	žádný	1	20 - 40	0 - 20
1	žádná	1	1	1	40 - 60	0
2	1	žádná	2	1	0	20 - 40
žádná	2	1	3	1	0 - 20	0 - 20
1	žádná	2	4	1	20 - 40	0
2	2	žádná	žádný	2	40 - 60	20 - 40
žádná	1	2	1	2	0	0 - 20
1	2	1	2	2	0 - 20	0
2	žádná	2	3	2	20 - 40	20 - 40
žádná	1	žádná	4	2	40 - 60	0 - 20
1	žádná	1	žádný	3	0	20 - 40
2	1	žádná	1	3	0 - 20	0 - 20
žádná	2	2	2	3	20 - 40	0
žádná	1	2	3	3	40 - 60	0
1	2	2	4	3	0	20 - 40
1	žádná	2	žádný	4	0 - 20	0 - 20
2	1	žádná	1	4	20 - 40	0
žádná	2	1	2	4	40 - 60	20 - 40
2	2	žádná	3	4	0	0 - 20
2	2	2	4	4	40 - 60	0
1	žádná	žádná	žádný	5	0 - 20	0 - 20
2	1	1	1	5	20 - 40	0
žádná	2	2	2	5	40 - 60	20 - 40
2	2	1	3	5	0	0 - 20
2	1	žádná	4	5	40 - 60	20 - 40

Při zvolení střední intenzity síly testů, byl původní počet kombinací snížen na výsledných 30 testovacích scénářů uvedených v Tabulka 1.

V případě potřeby redukce počtu prováděných testů, je možné snížit intenzitu jejich síly. Zde je postup výpočtu následující: z každého parametru je vybrána nejméně jedna hodnota a následně kombinatorický algoritmus optimalizuje počet testovacích scénářů na minimum kombinací. Daný postup je vhodný zejména pro regresní testy, případně smoke testy. Výsledný počet testovacích scénářů pro nejnižší intenzitu je 6 (viz příloha č. 1).

Pro potřeby velmi intenzivní síly testů je možné postupovat následovně: pro každou možnou trojici parametrů jsou vygenerovány všechny možné kombinace a následně je provedena jejich optimalizace. Výsledný počet testovacích scénářů pro nejvyšší intenzitu je 133 (viz příloha č. 2).

7.3.2 SET 2: Uživatel

Ve druhé etapě byly k parametrům terénu krajiny přidány uživatelsky zvolené parametry, a to:

- lokalizace posledního spatření váhy: 0, 3, 6, 9,
- směr pohybu: ano, ne,
- pravděpodobnost výskytu v oblasti: 50 % – 59 %, 60 % – 69 %, 70 % – 79 %, 80 % – 89 %, 90 % – 99 %, 100 %
- typ pohřešované osoby: dítě do 3 let, dítě 3 – 6 let, dítě 7 – 12 let, dítě 13 – 15 let, deprese, psychická nemoc, retardovaný, alzheimer, turista, demence.

Výsledky pro SET 2:

Pokud by nebyly využity kombinatorické techniky, nástroj CTT vypočítal 2 332 800 možných kombinací parametrů.

Při zvolení střední intenzity síly testů, byl původní počet kombinací redukován na 62 (viz příloha č. 3).

Při další redukci počtu prováděných testů, je možné snížit intenzitu jejich síly. Výsledný počet testovacích scénářů pro nejnižší intenzitu je 10 (viz příloha č. 4).

V případě zvolení nejvyšší intenzity síly testů dosáhl výsledný počet testovacích scénářů 320 (viz příloha č. 5).

8 REALIZACE VYBRANÝCH TESTŮ

Z metodiky testování představené v kapitole 6 vyplývá, že pro zajištění požadavků na vyvíjený modul PÁTRAC je nutné realizovat celou řadu testů.

S ohledem na aktuální potřeby testování modulu PÁTRAC byla nejvyšší priorita přidělena dvěma kategoriím testů, a to:

- testy variant možných situací při plánování pátrací akce,
- testy grafického uživatelského rozhraní (dále jen „GUI“).

Rozpracování testů variant možných situací při plánování pátrací akce až na úroveň testovacích scénářů se věnovala kapitola 7, ale vzhledem k náročnosti zpracování těchto typů testů není jejich realizace předmětem této diplomové práce. Aktuálně se z vytvořených testovacích scénářů připravuje testovací plán pro další fázi testování.

Po dohodě s vedoucím práce jsem se tedy zaměřila na realizaci sady manuálních testů z pohledu uživatele pomocí GUI, které popisuje tato kapitola.

8.1 Příprava manuálních testů

8.1.1 Specifikace rozsahu manuálních testů

Sadou manuálních uživatelských testů provedených přes GUI modulu PÁTRAC budou ověřeny deklarované funkcionality panelu „Průvodce“.

Pro potřeby provedení uživatelských testů budou použita zdrojová data za Plzeňský a Královéhradecký kraj.

8.1.2 Definice plánu manuálních testů

V rámci uživatelských testů budou kontrolovány požadované funkcionality pomocí následujících aktivit:

- vytvoření nového projektu,
- zadání názvu obce,
- automatický zoom na zvolené území v mapovém okně,
- vložení místa posledního spatření kliknutím do mapového okna,
- doplnit ID ke každé lokalizaci místa posledního spatření, čas a váhu důvěryhodnosti svědectví v intervalu 0 – 9,
- výběr typu pohřešované osoby,

- zobrazení vypočtené oblasti možného v mapovém okně,
- zadání očekávané procentuální míry pravděpodobnosti výskytu pohřešované osoby ve vypočtené oblasti,
- zadání dostupných sil a prostředků pro pátrací akci,
- zobrazení pátracích sektorů v mapovém okně,
- report v podobě mapových výstupů ve formátu PDF a GPX do GPS,
- doplnění výsledku pátrání pomocí integrovaného formuláře.

V rámci plánování uživatelských testů jsem sestavila testovací scénář zobrazený v Tabulka 2.

Tabulka 2: Testovací scénář

Parametr	Obsah
ID testu	P_01
Název testu	Projekt
Priorita testu	A
Shrnutí testu	Založení projektu v PÁTRACI pomocí „Průvodce“
Popis testu	Tester projde všechny kroky v panelu „Průvodce“ vložení požadovaných údajů dle uživatelské dokumentace a ověří tím, zda všechny etapy procesu pracují dle očekávání.
Specifické vstupní a výstupní podmínky testu (pokud existují)	
Předpoklady a rizika	Předpokladem úspěšného dokončení testu je korektní průchod jednotlivými etapami.
Testovací data	
Vazba na okolní IS	
Očekávaný výstup	Tester získal mapové výstupy v PDF a GPX
Datum vytvoření/ revize	2018-06-11
Verze	1
Přílohy	

8.1.3 Plánování testovacích běhů testu Projekt

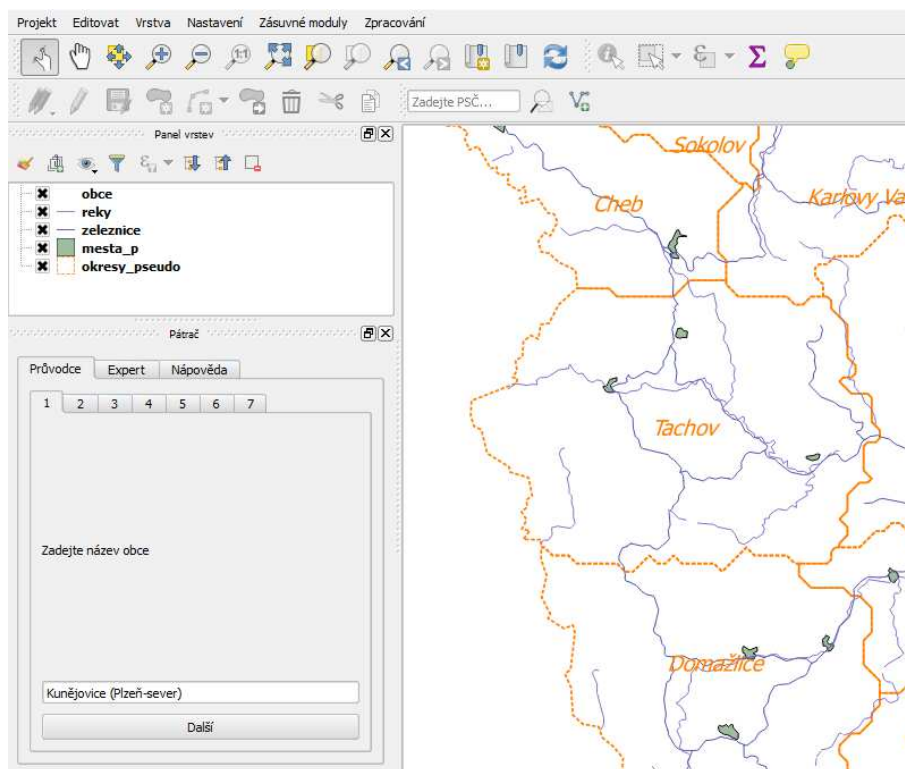
Připravené testy jsou prováděny po každé avizované aktualizaci modulu PÁTRAC pro ověření, že opravou nevznikly neočekávané chyby.

8.2 Provedení a vyhodnocení testů manuálních testů

V následující podkapitole dokumentuji průběh první sady testů, pro demonstrování odhalených chyb, které jsem reportovala vývojáři. V aktuální verzi modulu PÁTRAC jsou již výsledky provedených testů zpracovány.

8.2.1 Realizace testu Projekt a reporting chyb

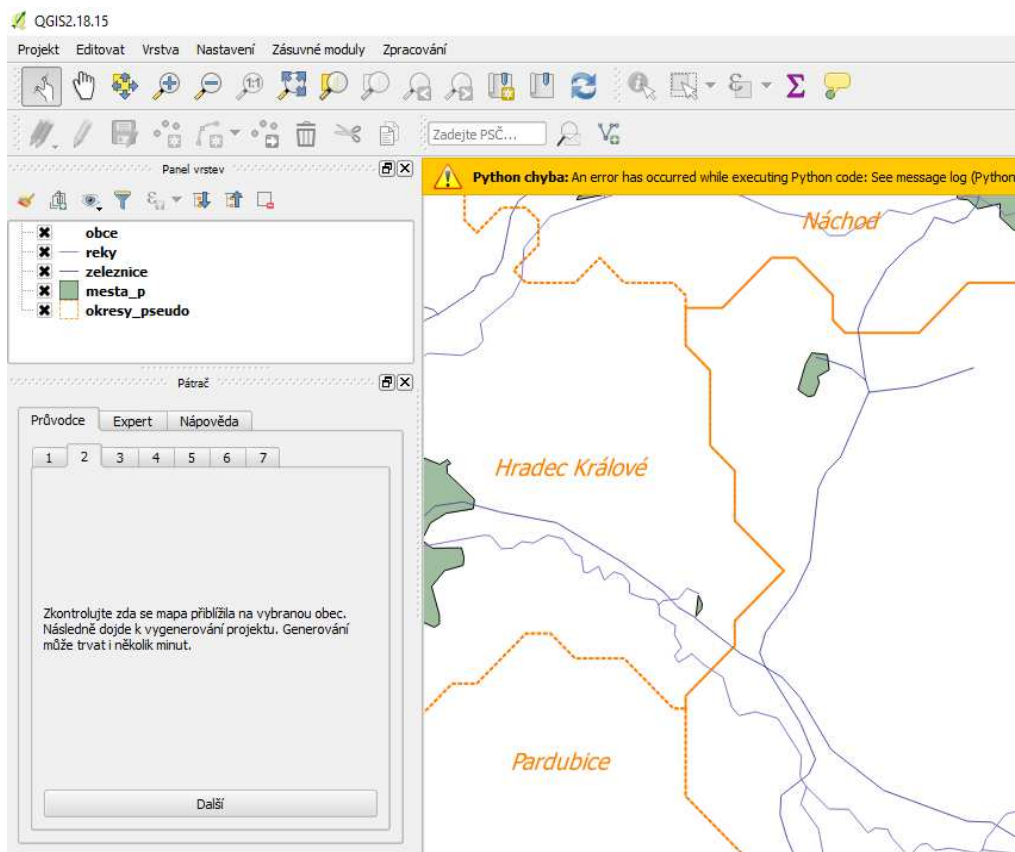
Krok č. 1: Tester v panelu „Průvodce“ vyplnil název obce a potvrdil výběr – v mapovém okně se objevil náhled dané lokality (viz Obrázek 8) a panel se automaticky posunul na záložku 2.



Obrázek 8: Zadání obce

Krok č. 2: Tester provedl vizuální kontrolu a výsledek potvrdil.

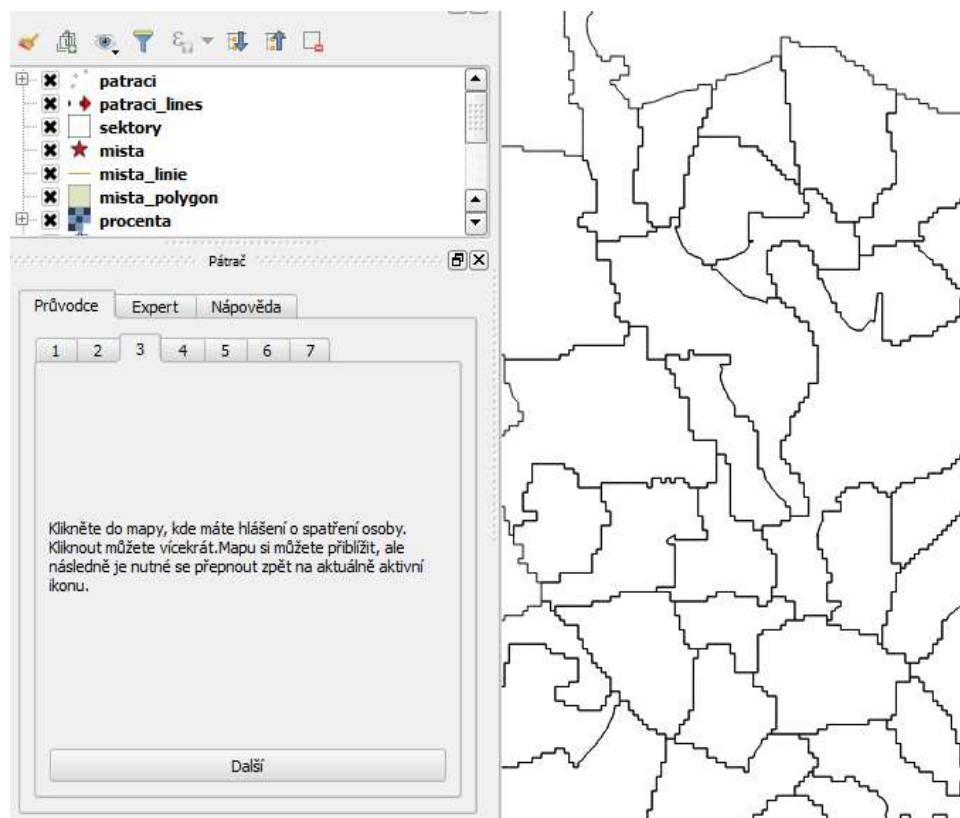
První provedený test v tomto bodě skončil chybou, protože po načtení vybrané oblasti nedošlo k automatickému posunu na záložku 3, jak bylo očekáváno. Pro pokračování v testu je nutné se ručně přepnout na záložku 3. Byla tedy simulována situace, kdy PÁTRAC zobrazí vybranou lokalitu, ale uživatel si není jistý, zda akce proběhla a na záložce 2 provede opětovné potvrzení. V takovém případě se v mapovém okně ztratí požadovaná lokalita a zoom se přesune mimo požadované území (viz Obrázek 9).



Obrázek 9: Odhalená chyba v panelu „Průvodce“

Krok č. 3: Tester kliknutím do mapového okna zadá místo posledního spatření a doplní čas spatření a váhu svědectví (viz Obrázek 10).

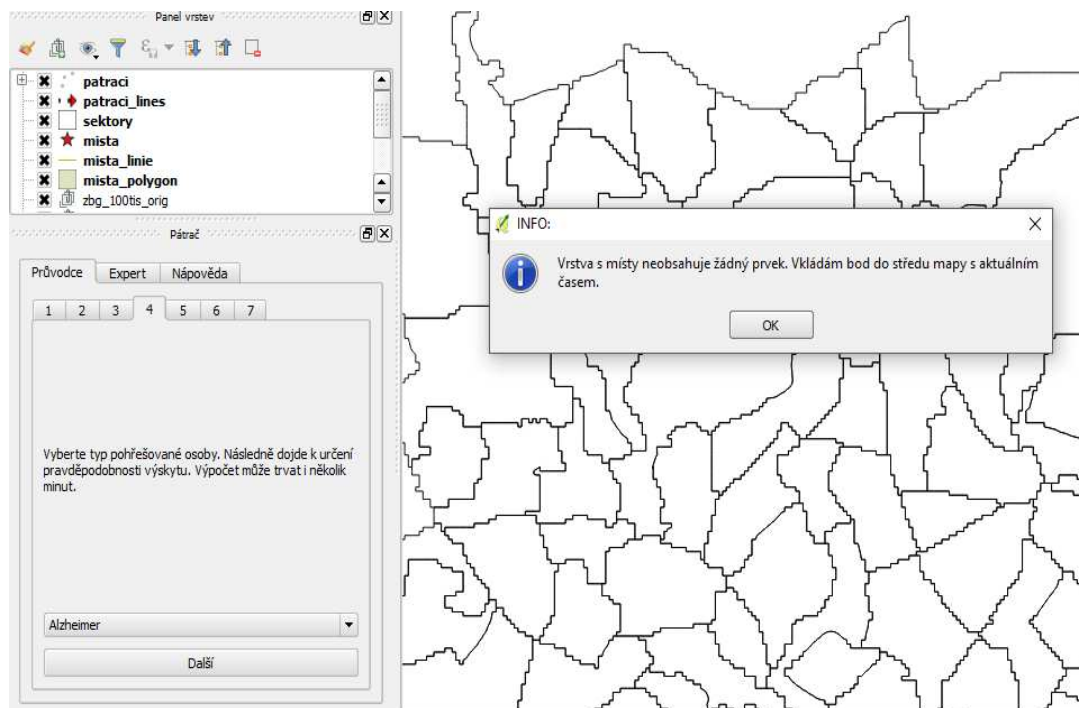
Při provádění první série testu „Projekt“ jsem odhalila, že nelze kliknout do mapového pole pro zadání místa posledního spatření, nalezená chyba byla následně reportována vývojáři systému.



Obrázek 10: Zadání místa posledního spatření

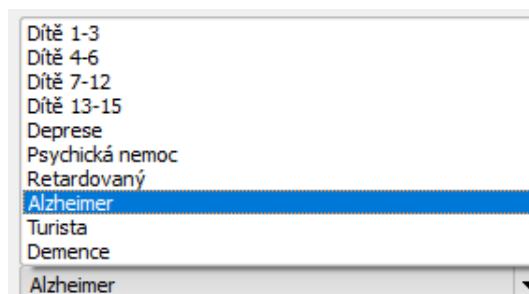
Při pokusu potvrdit výběr bez zadání místa posledního spatření se objeví dialogové okno, které informuje, že jako místo posledního spatření PÁTRAC doplní na střed mapového okna s aktuálním časem (viz Obrázek 11) a automaticky se přepne na záložku 4.

Dle očekávání by nemělo být uživateli umožněno přejít na další krok bez zadání místa posledního spatření, jelikož se jedná o důležitý parametr pro výpočet oblasti možného výskytu.



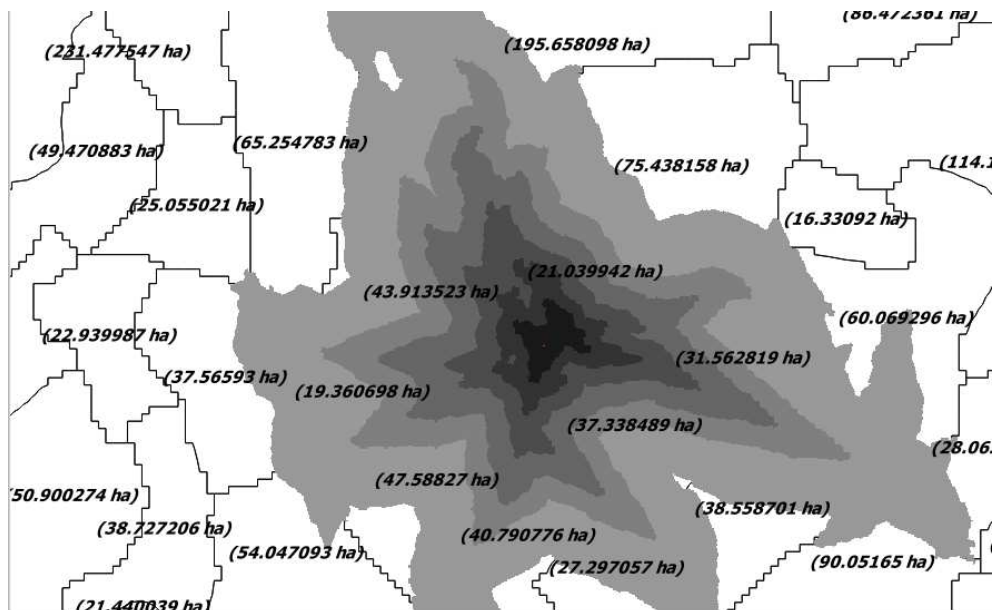
Obrázek 11: Informace o umístění místa posledního spatření do středu mapového pole

Krok č. 4: Tester provede výběr ze seznamu jednotlivých typů pohřešovaných osob. Zadání proběhlo dle očekávání (viz Obrázek 12).



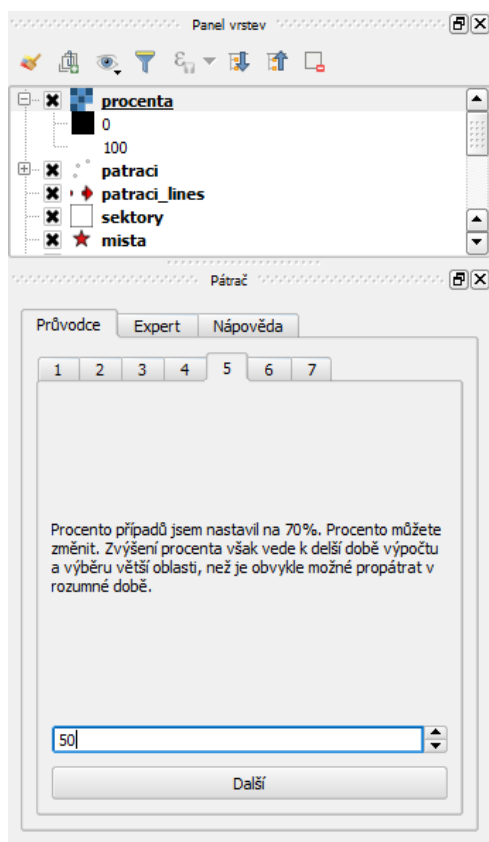
Obrázek 12: Nabídka typů pohřešovaných osob

PÁTRAC na základě zadaných parametrů provede výpočet oblasti možného výskytu pohřešované osoby a zobrazí ji v mapovém okně. Proběhlo dle očekávání (viz Obrázek 13).



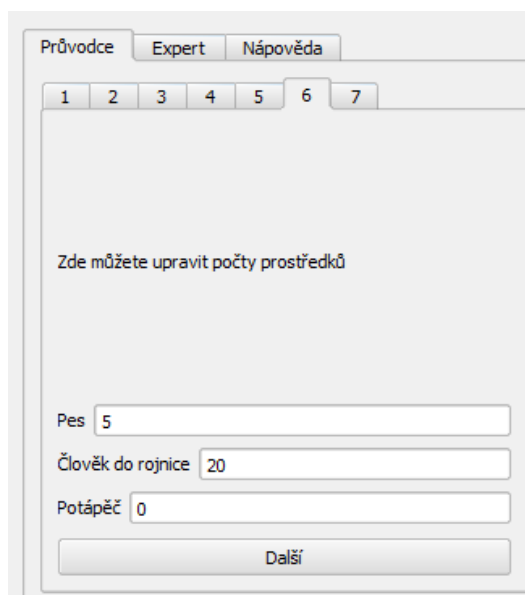
Obrázek 13: Oblast předpokládaného výskytu pohřešované osoby

Krok č. 5: Tester stanoví očekávanou procentuální míru pravděpodobnosti výskytu pohřešované osoby ve vypočtené oblasti. Proběhlo dle očekávání (viz Obrázek 14).

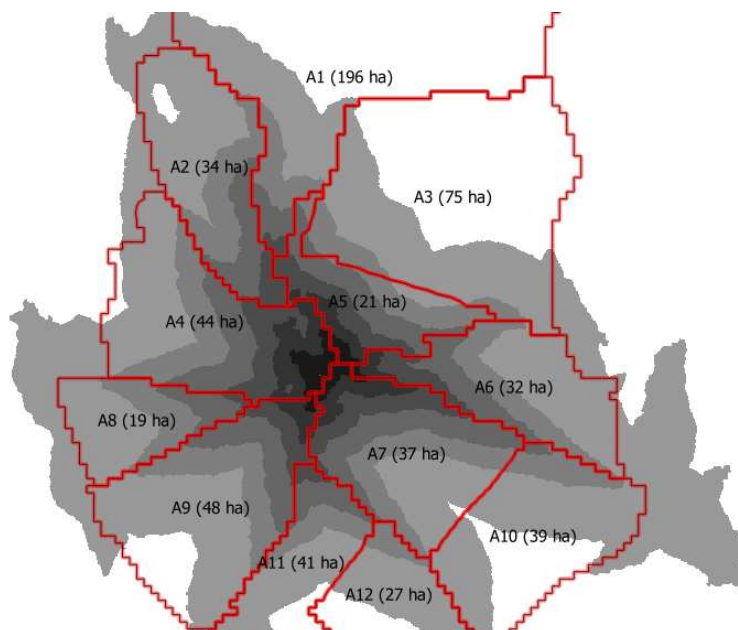


Obrázek 14: Stanovení procentuální míry pravděpodobnosti výskytu pohřešované osoby

Dále tester doplní dostupné síly a prostředky pro pátrání (viz Obrázek 15). Po vyplnění a potvrzení údajů PÁTRACĚ zobrazí pátrací sektory (viz Obrázek 16). Obrázek 15).



Obrázek 15: Zadání údajů o dostupných silách a prostředcích pro pátrání



Obrázek 16: Vygenerované pátrací sektory

Krok č. 6: V době provádění této sady testů nebyly dostupné nástroje pro generování mapových výstupů. Je dostupný pouze report v podobě viz Obrázek 17.

REPORT

Doba pro pátrání

Pro propátrání se počítá 3 hodiny jedním týmem

K dispozici je 5 KPT

Oblast prohledají přibližně za 14.0 hodin

Vhodné nasadit vzdušný pátrací tým (APT). Helikoptéru nebo dron. Prostor obsahuje volné plochy bez porostu.

K dispozici je 20 lidí pro PT

Oblast prohledají přibližně za 42.0 hodin

K dispozici není žádný potápěč. Je nutné nějaké zajistit.

GPX a PDF pro pátrání

Pro propátrání referenční plochy (cca 30 ha) se počítá 3 hodiny jedním týmem.



SEKTOR A1 (196 ha) [Typy povrchu](#)

- volný schůdný bez porostu: 7.06 %
- volný schůdný s porostem: 3.95 %
- volný obtížně schůdný: 0 %
- porost lehce průchozí: 82.84 %
- porost obtížně průchozí: 0 %
- zastavěné území měst a obcí: 1.59 %
- městské parky a hřiště s pohybem osob: 0 %
- městské parky a hřiště bez osob: 0 %
- vodní plocha: 1.26 %
- ostatní plochy: 3.31 %

Nasazení **6 KPT, 2 PT, 1 VPT, 1 APT** [Podrobnosti](#)

- Vhodné nasadit 6 Kynologických pátracích týmů (KPT) k propátrání do 3 hodin
- Je možné nahradit 6 KPT 6.0 PT
- Vhodné nasadit 2 Pátracích týmů (PT) s dvaceti členy k propátrání do 3 hodin
- Vhodné nasadit 1 Vodních pátracích týmů (VPT) k propátrání do 3 hodin
- Vhodné nasadit 1 Vzdušný pátrací tým (APT). Helikoptéru nebo dron. Prostor obsahuje volné plochy bez porostu.



Obrázek 17: Výsledný report

9 DOPORUČENÍ PRO DALŠÍ TESTOVÁNÍ

Vzhledem k složitosti problému a množství různých kombinací, které je nutné otestovat se jako nejvhodnější ukazuje použít automatizované testy.

V těchto testech není nezbytně nutné přistupovat k testovanému systému přes jeho GUI. Stačí vytvořit testy, které využívají programové jednotky testovaného systému na nižší úrovni. Jako příklad takových testů může sloužit speciální model, který nasimuluje pohyb pohřešované osoby v terénu. K tomuto modelu je možné vytvořit sadu automatizovaných testů, která bude kontrolovat vygenerovaný pátrací plán v tom smyslu, zda vytvořený pátrací plán dokáže simulovanou pohřešovanou osobu najít efektivně z pohledu vynaložených prostředků.

Další možností je vytvoření jiné sady automatizovaných testů, která bude kontrolovat základní parametry vygenerovaného pátracího plánu například, zda neobsahuje výrazné chyby nebo výrazné rozpory.

Z výsledků testování je patrné, že algoritmus použitý v modulu PÁTRAC pro výpočty pohybu pohřešované osoby v terénu vyhodnocuje svah se sklonem 45° jako neprůchozí, nezahrnuje ho tedy do výpočtu. Nabízí se tedy otázka, jak ošetřit případ, kdy pohřešovaná osoba mohla spadnout ze srázu dolů. V tuto chvíli je možné zmíněnou situaci popsat v uživatelské dokumentaci, že přirozenou hranici generovaných sektorů typu sráz je nutné při pátrání samém také pečlivě prohledat jako jedno z možných míst výskytu pohřešované osoby.

Totožný problém vyvstává u budov v prohledávané oblasti, které modul PÁTRAC obdobně jako u svahů se sklonem nad 45° považuje za nepřekonatelné překážky, ale přesto se pohřešovaná osoba může nacházet například v lesní chatě. Z těchto důvodů nelze při pátrání po pohřešované osobě tyto stavby opomenout.

Rovněž je potřeba zohlednit druhy vegetace, která určuje rychlost postupu v krajině a analýzy vodních ploch a toků z hlediska prostupnosti. V aktuální verzi PÁTRAC se s vodou pracuje jako s neprůchodnou překážkou. Zároveň by dle mého názoru stálo za zvážení pracovat v analýzách pohybu i s prostupností vegetačního porostu, a to ve smyslu přidělení vah dle náročnosti průchodu jednotlivými druhy vegetace.

V neposlední řadě je třeba zvážit provedení citlivostní analýzy, která by měla odhalit vliv změny parametrů na generování testovacích scénářů.

10 ZÁVĚR

Při realizaci testování zásuvného modulu pro podporu pátrání po pohřešované osobě byly popsány možnosti testování zásuvného modulu PÁTRAC pro platformu Quantum GIS, navrženy způsoby testování zásuvného modulu PÁTRAC, dle navržené metodiky testování. Zároveň byla vytvořena sada testovacích scénářů, které slouží jako vstup pro vytvoření vytipovaných automatických testů.

Testování probíhalo bez výrazných komplikací, odhalené chyby při první fázi testování pomocí GUI byly v další verzi modulu PÁTRAC odstraněny, proto hodnotím tento modul pro výše popsané účely jako velmi přínosný.

SEZNAM POUŽITÉ LITERATURY

- [1] Testování softwaru. KITNER [online]. Radek Kitner [cit. 2019-01-07]. Dostupné z: <https://kitner.cz>
- [2] Software Testing. SlideShare [online]. KernelTraining.com, 2015 [cit. 2019-04-04]. Dostupné z: <https://www.slideshare.net/kerneltraining/software-testing-demo-presentation>
- [3] 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.
- [4] BOEHM, Barry W., Robert K. MCCLEAN a D. E. URFRIG. Some experience with automated aids to the design of large-scale reliable software. 1. IEEE Transactions on Software Engineering, 1975, 125-133. DOI: 10.1109/TSE.1975.6312826.
- [5] ISTQB. International Software Testing Qualifications Board [online]. ISTQB, 2016 [cit. 2019-04-04]. Dostupné z: <https://www.istqb.org>
- [6] TESTOVÁNÍ SOFTWARE [online]. Tomáš Hlava [cit. 2019-01-07]. Dostupné z: <http://testovanisoftwaru.cz>
- [7] Testing with the TMap Suite. TMap [online]. Sogeti [cit. 2019-04-04]. Dostupné z: <http://www.tmap.net>
- [8] KOOMEN, Tim, Leo VAN DER AALST, Michiel VROON a Bart BROEKMAN. TMap NEXT® - For Result-Driven Testing. Sogeti, 2014, 710 s. ISBN 9789075414806.
- [9] All About Software Testing Tools. QATestingTools.com [online]. Yuval

- Ben Hur [cit. 2019-04-10]. Dostupné z: <https://www.qateestingtools.com>
- [10] QAComplete. SMARTBEAR QAComplete [online]. SmartBear, 2019 [cit. 2019-04-06]. Dostupné z: <https://qacomplete.com>
- [11] QAComplete. SMARTBEAR [online]. SmartBear, 2018 [cit. 2019-04-06]. Dostupné z: <https://support.smartbear.com>
- [12] Guru99 [online]. Guru99 [cit. 2019-04-07]. Dostupné z: <https://www.guru99.com>
- [13] MantisBT [online]. mantis BUG TRACKER [cit. 2019-04-08]. Dostupné z: <https://www.mantisbt.org>
- [14] Redmine [online]. Redmine [cit. 2019-04-28]. Dostupné z: <https://www.redmine.org>
- [15] Top 10 Automation Testing Tools 2019. Testbytes Making Auality a Habit [online]. New York: testbytes Making Auality a Habit, 2019 [cit. 2019-04-10]. Dostupné z: <https://www.testbytes.net/blog/top-10-automation-testing-tools-2019>
- [16] Ranorex [online]. Graz: Ranorex, 2019 [cit. 2019-04-28]. Dostupné z: <https://www.ranorex.com>
- [17] AMMANN, Paul a Jeff OFFUTT. Introduction to software testing. Edition 2. New York, NY, USA: Cambridge University Press, 2017. ISBN 978-110-7172-012.
- [18] BURES, Miroslav; CERNY, Tomas; KLIMA, Matej. Prioritized process test: More efficiency in testing of business processes and workflows. In: International Conference on Information Science and Applications. Springer, Singapore, 2017. p. 585-593.
- [19] Ahmed, B. S., Zamli, K. Z, Afzal, W., Bures, M. "Constrained Interaction Testing: A Systematic Literature Study." IEEE Access, 5, 2017, 25706-25730. [Impact Factor = 3.244].

- [20] KUHN, D. Richard; KACKER, Raghu N.; LEI, Yu. Introduction to combinatorial testing. CRC press, 2013.
- [21] Zamli, K. Z., Din, F., Ahmed, B. S., Bures, M. . A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem. PloS one, 13(5), 2018, e0195675 [Impact Factor = 2.77].

SEZNAM POUŽITÝCH ZKRATEK

ACTS	Automated Combinatorial Testing for Software
ALM	Application Life Cycle Management
BDTM	Business-driven test management
CaSTB	Czech and Slovak Testing Board
CTT	Combinatorial testing tool
ČVUT	České vysoké učení technické v Praze
GIS	Geografický informační systém
GPS	Globální polohový systém (Global Positioning System)
GŘ HZS ČR	Generální ředitelství hasičského záchranného sboru České republiky
GUI	Grafické uživatelské rozhraní
HP ALM	Hewlett Packard Application Lifecycle Management
HZS ČR	Hasičský záchranný sbor České republiky
IEC	Mezinárodní elektrotechnická komise (International Electrotechnical Commission)
IEEE	Institut pro elektrotechnické a elektronické inženýrství (Institute of Electrical and Electronics Engineers)
ISEB	Zkušební výbor informačních systémů (Information Systems Examination Board)
ISO	Mezinárodní organizace pro normalizaci (International Organization for Standardization)
ISTQB	International Software Testing Qualifications Board
IT	Informační technologie
IZS	Integrovaný záchranný systém
KOPIS	Územně příslušné krajské operační a informační středisko HZS ČR
KPT	Kynologický pátrací tým
MantisBT	Mantis BUG TRACKER
MV	Ministerstvo vnitra
PČR	Policie České republiky
TDD	Test Driven Development

UAT	Uživatelské akceptační testy (User acceptance test)
UFT	HPE Unified Functional Testing
UTM	Univerzální transversální Mercatorův systém souřadnic (universal transversal mercator)
VŠB-TUO	Vysoká škola báňská – Technická univerzita Ostrava

SEZNAM OBRÁZKŮ

Obrázek 1: Problém s vývojem softwaru [2]	15
Obrázek 2: Model průběžného rozvoje systému [3]	17
Obrázek 3: V-model [3]	18
Obrázek 4: W-model [3]	19
Obrázek 5: Boehmův první zákon [4]	20
Obrázek 6: Fáze plánování a přípravy testů	34
Obrázek 7: Combinatorial testing tool	43
Obrázek 8: Zadání obce	48
Obrázek 9: Odhalená chyba v panelu „Průvodce“	49
Obrázek 10: Zadání místa posledního spatření	50
Obrázek 11: Informace o umístění místa posledního spatření do středu mapového pole	51
Obrázek 12: Nabídka typů pohřešovaných osob	51
Obrázek 13: Oblast předpokládaného výskytu pohřešované osoby	52
Obrázek 14: Stanovení procentuální míry pravděpodobnosti výskytu pohřešované osoby	52
Obrázek 15: Zadání údajů o dostupných silách a prostředcích pro pátrání	53
Obrázek 16: Vygenerované pátrací sektory	53
Obrázek 17: Výsledný report	54

SEZNAM TABULEK

Tabulka 1: SET 1 – střední intenzita síly testů	44
Tabulka 2: Testovací scénář	47

SEZNAM PŘÍLOH

Příloha č. 1: SET 1 – nejnižší intenzita

Příloha č. 2: SET 1 – nejvyšší intenzita

Příloha č. 3: SET 2 – střední intenzita

Příloha č. 4: SET 2 – nejnižší intenzita

Příloha č. 5: SET 2 – nejvyšší intenzita