**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY

BAKALÁRSKA PRÁCA

FRANTIŠEK CABADAJ

**Vývoj automatizovaného testovacieho systému pre kontrolu komunikácie medzi SAP a externými úložiskami**

Vedúci práce: doc. Ing. Marek Kvet, PhD.

Registračné číslo: 556425

Žilina, 2020

**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA ......................................

BAKALÁRSKA/DIPLOMOVÁ/

DIZERTAČNÁ PRÁCA

ŠTUDIJNÝ ODBOR:

MENO PRIEZVISKO

**Názov práce**

Žilinská univerzita v Žiline

Fakulta ..................

Školiace pracovisko..............

Žilina, 2016

#### Úvodné pokyny pre použitie šablóny

Väčšina nadpisov šablóny je previazaná na dokument Pokyny\_pre\_vypracovanie\_ZP.docx. Sú v ňom podrobnejšie informácie o vypracovaní záverečnej práce. Linky budú správne fungovať vtedy a len vtedy keď si uložíte dokument so záverečnou prácou do toho istého priečinka, v ktorom sa nachádza uvedený dokument. Priamo z tejto šablóny nefungujú.

#### (Pri vytváraní svojho dokumentu pokyny pre použitie šablóny vymažte!)

#### [Čestné vyhlásenie](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Čestné_vyhlásenie)

Podľa národnej legislatívy nie je v záverečnej práci povinné.

#### [Poďakovanie](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Poďakovanie)

Nie je povinné, vysvetlenie a príklad poďakovania

[ABSTRAKT V ŠTÁTNOM JAZYKU](C:\\Users\\DonChameleón\\Desktop\\Pokyny_pre_vypracovanie_ZP.docx" \l "Abstrakt)

CABADAJ, František: Vývoj automatizovaného testovacieho systému pre kontrolu komunikácie medzi SAP a externými úložiskami. [Bakalárska práca] – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky;

PRIEZVISKO, Meno: *Názov témy záverečnej práce*. [Bakalárska/ Diplomová/ Dizertačná práca]. – Názov univerzity. Názov fakulty; Názov školiaceho pracoviska. – Školiteľ/Vedúci: (tituly pred menom) Meno a priezvisko (tituly za menom) – Stupeň odbornej kvalifikácie: bakalár/magister/iný. – Mesto: skratka fakulty a univerzity, rok predloženia. Počet strán (napr. 35 s.)

[Vysvetlenie a príklad.](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Príklad_abstrakt)

[**Kľúčové slová**](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Kľúčové_slová)**:**  (jednoslovné alebo viacslovné termíny, ktoré charakterizujú vecný obsah práce)

[ABSTRAKT V CUDZOM JAZYKU](C:\\Users\\DonChameleón\\Desktop\\Pokyny_pre_vypracovanie_ZP.docx" \l "Abstrakt_CJ)

[Vysvetlenie.](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Abstrakt_CJ)

**Key words:**

[Obsah](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Obsah)

Zoznam obrázkov 7

Zoznam tabuliek 8

Zoznam skratiek 9

Úvod 10

1 Súčasný stav riešenej problematiky doma a v zahraničí 12

2 Teoretické východiská 13

2.1 Automatizované testovanie https://testguild.com/automation-testing/ 13

2.1.1 Manuálne testovanie 13

Why Automated Testing? (https://testguild.com/automation-testing/) 14

2.2 SAP 16

2.2.1 Jazyk ABAP 16

2.3 Externé úložiská 16

2.3.1 Hadoop 16

2.3.2 Iné 16

3 Ciele práce 17

4 Praktická časť 18

4.1 Návrh automatizovaného testovacieho systému 18

4.1.1 Architektúra 18

4.2 Implementácia testovacieho systému 18

4.3 Spracovanie výsledkov 18

5 Metodika práce a metódy skúmania 19

6 Výsledky práce a diskusia 20

6.1 Výsledky práce 20

6.2 Diskusia 20

Záver 21

Zoznam použitej literatúry 22

Zoznam príloh 23

Prílohy 24

Príloha A: Názov prílohy 25

Príloha B: Obsah DVD 26

# [Zoznam obrázkov](C:\\Users\\DonChameleón\\Desktop\\Pokyny_pre_vypracovanie_ZP.docx" \l "Zoznam_obrázkov)

**Nenašli sa žiadne položky zoznamu obrázkov.**

# [Zoznam tabuliek](C:\\Users\\DonChameleón\\Desktop\\Pokyny_pre_vypracovanie_ZP.docx" \l "Zoznam_obrázkov)

**Nenašli sa žiadne položky zoznamu obrázkov.**

# [Zoznam skratiek](C:\\Users\\DonChameleón\\Desktop\\Pokyny_pre_vypracovanie_ZP.docx" \l "Zoznam_skratiek)

ANSA Automated Network Simulation and Analysis

BFD Automated Network Simulation and Analysis

Cisco IOS Cisco Internetwork Operating System

# [Úvod](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Úvod)

Úvodný text sa píše v štýle odseku textu Normálny. Ten je v tejto šablóne nastavený a neodporúčame ho meniť.

Zásady spracovania záverečnej práce sú uvedené [TU](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Zásady_spracovania_ZP).

Formálna úprava záverečných prác je uvedená [TU](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Formálna_úprava_ZP)

#### Pokyny pre používanie šablóny

V šablóne sú pripravené štyri typy nadpisov:

* **Nadpis 1** je číslovaný a začína vždy na novej stránke (dá sa priradiť klávesom **F5**)
* **Nadpis 2** nadväzuje číslovaním na Nadpis 1 (**F6**)
* **Nadpis 3** nadväzuje číslovaním na predchádzajúce nadpisy (**F7**)
* **Nadpis 4** je bez čísla (**F8**)

V šablóne sú pripravené ďalšie štýly odseku textu, pri ktorých odporúčame nemeniť ich nastavené vlastnosti:

* **Normálny** je určený na písanie bežného textu (**F4**)
* **Obrázok** je zarovnaný na stred a určený pre obrázky a ich popis (**F10**)
* **Skratky** je určený na písanie zoznamu skratiek použitých v dokumente
* **Literatúra** je určený na písanie zoznamu literatúry (ešte nie je pripravený)
* **Tabuľka** je určený na vkladanie textu do tabuliek
* **Typ práce** je určený na vypísanie typu práce do hlavičky

#### Pokyny pre prácu s ilustráciami

Ilustrácie sú obrázky obsahujúce grafy, diagramy, mapy, schémy a pod. Nie je potrebné rozlišovať rozličné typy ilustrácií, stačí, ak sa všetky označia ako „Obrázok”.

#### Vloženie ilustrácie

Obrázky a tabuľky vkladáme v tejto šablóne takto:

1. Karta **Vložiť** → **Obrázok**
2. Na obrázku klikneme na pravé tlačidlo myši → **Vložiť popis**
3. Do poľa **Popis** dopíšeme názov obrázku
4. Na obrázok a jeho popis použijeme štýl **Obrázok** (**F10**)

V texte sa na vytvorené obrázky a tabuľky odkazujeme krížovými odkazmi. Krížový odkaz na obrázky a tabuľky vytvoríme takto:

1. Nastavíme sa kurzorom tam, kde sa má objaviť odkaz,
2. Na karte **Vložiť** → **Krížový odkaz**
3. V dialógovom okne pre **Krížový odkaz** v poli **Typ odkazu** vyberieme **Tabuľka** alebo **Obrázok**
4. V poli **Vložiť odkaz** vyberieme **Iba menovka a číslo**
5. Zo zoznamu popisov vyberieme tabuľku alebo obrázok a zaškrtneme **Hypertextový odkaz**
6. Klikneme na **Vložiť**

#### Aktualizácia krížových odkazov a popisov obrázkov

Pokiaľ v texte vymažete alebo pridáte ilustráciu, tak je potrebné prečíslovanie ilustrácií a krížových odkazov:

1. Vyberte celý dokument (**Ctrl + A**)
2. Stlačte kláves **F9**

Obrázky by mali byť kreslené v rovnakom štýle s popisom v jazyku práce. Podrobnejší popis pre prácu s ilustráciami nájdete [tu](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Ilustrácie).

#### Pokyny pre prácu s tabuľkami

Každá tabuľka musí mať poradové číslo a titulok, umiestnený zvyčajne nad tabuľkou, pričom titulok je zarovnaný na pravú stranu tabuľky. Tabuľky by mali byť rovnako naformátované s rovnakým vzhľadom buniek.

Podrobnejší popis pre prácu s tabuľkami nájdete [tu](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Tabuľky).

#### Zoznam použitej literatúry

Bibliografické odkazy vkladajte cez kartu **Referencie,** zoskupenie nástrojov **Citácie a bibliografia**.

# [Súčasný stav riešenej problematiky doma a v zahraničí](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Súčasný_stav_riešenia)

# Teoretické východiská

## Automatizované testovanie <https://testguild.com/automation-testing/>

Automatizované testovanie je technika, ktorá sa používa na zvýšenie rýchlosti overovania alebo iných opakovaných úloh v životnom cykle vývojového oddelenia.

Because of this, to save time, many companies try to take their manual test cases and convert them to an automated test case. An automated test tool then executes the test steps automatically without human intervention.

Z tohto dôvodu, aby ušetrili čas, sa veľa spoločností snaží zobrať manuálne testovacie scenáre a previesť ich na automatizované testovacie scenáre. Automatizovaný testovací nástroj potom vykoná kroky testovacieho scenára automaticky bez ľudského pričinenia.

Also, automated tools use a programming approach to emulate a user interacting with an application and verifying test steps using programming assertions.

Taktiež automatizovaný testovací nástroj používa programátorský prístup aby napodobnil interakciu používateľa s aplikáciou a overil kroky testu použitím programátorských výrokov (tvrdení)??!!.

The term “‘automation testing” can also be controversial with many folks preferring to use the term [automated checking](https://testguild.com/automation-checking/) or automation in testing.

In any case, before we take a look at automated testing, let's touch on some issues with manual testing.

Automatické testovanie by malo byť využívané pre uľahčenie práce vývojárov, takže predtým ako sa ponoríme hlbšie do témy automatických testov, pozrime sa najskôr na problémy manuálneho testovania.

### Manuálne testovanie

There are a few reasons why [manual testing](https://testguild.com/2016/08/25/modern-testing-21st-century-tester-testers-screwed/) may be problematic:

* It uses a lot of resources
* It’s time-consuming
* It sometimes lacks proper coverage
* Due to their repetitiveness, testers may get bored and miss steps when executing manually, leading to possible inconsistencies.

Like automated testing “manual testing” can also be a controversial term. In fact, [Michael Bolton](http://www.developsense.com/blog/2017/11/the-end-of-manual-testing/) has gone as far as to say that [manual testing does not exist](https://testguild.com/196-rapid-software-testing-michael-bolton/).

*Automation testing* should be used to help with your testing efforts.

But does automation replace all your tests?

During [regression testing](https://testguild.com/2016/10/25/improve-testing-efficiency/), a manual tester will take an existing test case procedure and execute it step by step. This can be time-consuming since it is a manual process done by hand.

V priebehu regresného testovania, zisťovania či nedávne zmeny v kóde neporušili alebo nezničili už existujúcu funkcionalitu systému, zoberie manuálny tester existujúci testovací scenár pracovného postupu a krok po kroku ho vykoná.

Je niekoľko dôvodov prečo manuálne testovanie môže byť problematické:

* Používa veľa prostriedkov – na manuálne testovanie potrebujeme testovací tím, ktorý môže byť finančne náročný
* Je časovo náročné – celkové pokrytie testovacích scenárov manuálne zaberá veľa času
* Niekedy nemusí pokrývať celý testovaný scenár – určité scenáre vyžadujú schopnosti programátorov
* Kvôli opakujúcej sa povahe, testeri môžu pri manuálnom vykonávaní zanedbať alebo prehliadnuť niektoré dôležité kroky čo môže viesť k nekonzistentnému stavu testu.

Existujú však prípady, kedy automatické testovanie nedokáže pokryť všetky scenáre. Preto pri vývoji nového softvéru vo firmách, ktoré nemajú testovací tím, robia vývojári manuálne testy predtým ako sa nový softvér dostane na produkčný systém.

## Why Automated Testing? (<https://testguild.com/automation-testing/>)

Prečo testovať automaticky?

So zvýšenou rýchlosťou, ktorou sa softvér v dnešnej dobe vyvíja, je automatické testovanie nevyhnutnou súčasťou vývoja nového softvéru. Praktiky ako priebežná integrácia, vývojárska praktika ktorá od vývojárov vyžaduje integrovať kód do zdieľaného úložiska niekoľko krát denne, vyžadujú testy ktoré bežia rýchlo a spoľahlivo. Veľa manuálneho overovania zabráni dosiahnutiu požadovanej rýchlosti vývoja softvéru.

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. (<https://www.thoughtworks.com/continuous-integration> )

**Průběžná integrace** (*angl. Continuous Integration*) je souhrnem různých vývojářských nástrojů a metod k urychlení vývoje [softwaru](https://cs.wikipedia.org/wiki/Software) a [spolupráce týmů](https://cs.wikipedia.org/w/index.php?title=Spolupr%C3%A1ce_t%C3%BDm%C5%AF&action=edit&redlink=1). Jedná se o součást metodik [extrémního programování](https://cs.wikipedia.org/wiki/Extr%C3%A9mn%C3%AD_programov%C3%A1n%C3%AD" \o "Extrémní programování). Slouží mimo jiné k urychlení nalezení nedostatků a chyb u softwarových [projektů](https://cs.wikipedia.org/wiki/Projekt" \o "Projekt) ve fázi vývoje.

I’d go so far as to say that in today’s modern development environment, we couldn’t succeed without automation.

Je takmer isté, že v dnešnom modernom vývojovom prostredí, by sme bez automatizovania neuspeli.

Although the main reason teams try to create automation is to save the company both time and money, it’s also important to give developers quick feedback so that when they check in code, they are alerted as soon as possible that the change they checked in broke something.

Hoci hlavným dôvodom prečo sa tímy snažia testovať automatizovane je aby ušetrili spoločnosti čas a peniaze, dôležité je tiež dodať vývojárom spätnú väzbu aby boli pri odovzdaní kódu čo najskôr informovaný o tom, že zmena kódu ktorú odovzdali niečo pokazila. Ďalšie dôvody pre automatizované testovanie sú:

* Verify newer versions of software
* Free testers up to focus on more exploratory-type testing
* Automated scripts are more repeatable
* Data population
* Accurate benchmarking
* Less false failure due to human error
* Greater test coverage
* Reusability
* Quicker release of software
* Get fast feedback to your developer on failing checked in software
* Saves time
* Ability to leverage programming capabilities
* Run for every new build of your application to act like a safety net
* Overenie novších verzií softvéru
* Oslobodenie testerov aby sa mohli sústrediť na viac komplexnejšie testovacie scenáre
* Väčšie pokrytie testov
* Znovupoužiteľnosť
* Rýchlejšie nasadenie softvéru
* Šetrí čas
* Dodáva rýchlu spätnú väzbu pre vývojárov ohľadom chýb v softvéri

The theory is that an automated test will save the company both time and money. But it seems that many folks fail to factor in the amount of time and money it takes to [maintain automated test suites](https://testguild.com/2016/12/19/test-automation-in-agile/). Some other reasons for automated testing are:

Teória je taká, že automatizované testy ušetria spoločnostiam čas a peniaze. Vyzerá to však tak, že veľa ľudí neberie do úvahy čas a peniaze, ktoré treba investovať do udržania stabilného testovacieho systému.

So are there any downsides to creating automation?

Existujú teda nevýhody pri vytváraní automatizovaných testov?

Since automated testing relies on programming languages for their creation, automation becomes a full-blown development effort.

Keďže na tvorbu automatických testov sa používajú programovacie jazyky, automatizácia sa stáva samostatným projektom na vývoj.

As a matter of fact, what you are doing is developing a piece of software to test another piece of software.  Treat your automated code just like your development code. Follow the same processes and best practices you would use for any other software development project.

Čo sa v skutočnosti robí je, že sa vyvíja časť softvéru na testovanie inej časti softvéru, preto je potrebné investovať rovnaké úsilie do vývoju automatických testov. Dodržujú sa rovnaké procesy a osvedčené postupy ako by sa použili pri vývoji akéhokoľvek iného projektu.

Automation testing is difficult and complicated, just like most other development software projects. It also presents many of the same issues other software programs do. So the same best practices for developing software also applies to automation as well.

Automatizované testovanie je náročné a komplikované, presne ako väčina ostatných projektov na vývoj softvéru. Taktiež prezentuje veľa rovnakých problémov výziev.

Treating your automation as a second-class development project will cause maintenance and reliability issues in the long run.

Ak sa k automatizovanému testovaniu bude pristupovať ako k druhotriednemu projektu, bude mať problémy s údržbou a spoľahlivosťou z dlhodobého hľadiska.

Proces automatizovaného testovania môžeme zhrnúť do týchto krokov:



**Príprava**

Najskôr je potrebné pripraviť sa a pochopiť ciele funkčného testovania, aké testovacie dáta sú potrebné a čo treba overiť.

### Write

Turn the requirements into an automated solution. Know what the start and end conditions are for each test. Tests should be completely independent of other tests. Add proper assertion checks to ensure your application is behaving according to your specifications. Each test should have a particular purpose.

**Písanie**

Zmena požiadaviek na automatizované riešenie. Určenie počiatočného a koncového stavu pre každý test. Testy by mali byť úplne nezávislé jeden od druhého. Je potrebné pridať dostatočné kontroly na uistenie sa, že sa aplikácia správa podľa zadaných špecifikácií. Každý test by mal mať konkrétny účel.

**Realizácia**

Realizácia testov by mala byť spoľahlivá. Odporúča sa spustiť každý test aspoň tri krát za sebou predtým ako sa skontroluje kód.

### Execute

Execute your tests should be reliable. Run each test at least three times in a row before checking in code.

**Vyhodnotenie**

Overenie, či automatizované testy robia to, čo sa od nich čaká. Pomocou manuálnych testov sa overí, či sa testy správajú podľa požiadaviek.

### Evaluate

verify that the automated test is doing what you expect it to do. Have manual tests verify that it is working as expected. Remember — if it’s not asserted, it’s not checked.ev

**Komunikácia**

Každý člen tímu by mal byť informovaný o výsledkoch testov. Pochybné testy by mali byť opravené čo najskôr, inak vzniká riziko, že tím bude výsledky testov ignorovať.

### Communicate

Be sure that everyone on your team is aware of the results. Flaky tests should be fixed ASAP, or you'll risk your team ignoring your test results.

**Opakovanie/Spravovanie**

Ak sa ukáže pochybný test, je potrebné ho prerobiť aby sa stal viac spoľahlivým. Najdôležitejšie je však vymazať akékoľvek testy, ktoré nie sú spoľahlivé a neboli opravené v danom časovom rámci.

### Repeat/Refactor

if you notice a flaky test, refactor it to make it more reliable. Most importantly, deletes any tests that are not reliable and haven’t been fixed within a given time frame.

## SAP

## What is SAP? <https://www.guru99.com/what-is-sap-definition-of-sap-erp-software.html>

**SAP** stands for Systems Applications and Products in Data Processing. SAP, by definition, is also the name of the ERP (Enterprise Resource Planning) software as well as the name of the company. SAP Software is a European multinational, founded in 1972 by Wellenreuther, Hopp, Hector, Plattner, and Tschira. They develop software solutions for managing business operations and customer relationships.

**SAP** je skratka pre systém aplikácie a produkt v spracovaní dát. SAP, podľa definície, je taktiež meno pre **ERP** softvér ako aj meno spoločnosti. SAP Software je európska medzinárodná spoločnosť založená Dietmarom Hoppom, Hans-Wernerom Hectorom, Hassom Plattnerom, Klausom Tschiraom a Clausom Wellenreutherom v roku 1972. Spoločnosť vyvíja softvérové riešenia pre riadenie obchodných operácií a vzťahov so zákazníkmi.

SAP system consists of a number of fully integrated modules, which covers virtually every aspect of business management.

SAP systém sa skladá z niekoľkých plne integrovaných modulov, ktoré pokrývajú prakticky každý aspekt riadenia podniku.

SAP is #1 in the ERP market. As of 2010, SAP has more  than 140,000 installations worldwide, over 25 industry-specific business solutions and more than 75,000 customers in 120 countries

SAP je jednotka na ERP trhu. Celosvetovo od roku 2010 má SAP viac ako 140 000 inštalácií, viac než 25 podnikových riešení pre špecifické priemyselné odvetvia a viac ako 75 000 zákazníkov v 120 krajinách.

Other Competitive products of SAP Software in the market are  Oracle, Microsoft Dynamics, etc.

## Externé úložiská

### Hadoop

### Iné

# Ciele práce

Cieľom bakalárskej práce je vytvoriť funkčný systém automatizovaných testov, pomocou ktorého sa bude kontrolovať komunikácia systému SAP s vybranými externými úložiskami dát (Apache Hive, AWS Redshift, Oracle , MSSQL)

Obsah práce možno zhrnúť nasledovne:

1. Analýza vybraných úložísk a identifikácia vhodných scenárov na automatizáciu testovania,

2. Zber a vyhodnotenie požiadaviek na vytváraný systém automatického testovania,

3. Návrh a implementácia softvérového riešenia v jazyku ABAP tak, aby sa jednotlivé testy vykonávali vo vopred určenom čase a výsledky testov aby sa uchovávali na vopred definovanom mieste a v presne určenej forme,

4. Návrh a implementácia testov, tvorba testovacích scenárov,

5. Vyhodnotenie dosiahnutých výsledkov,

6. Dokumentácia riešenia vo forme UML.

# [Praktická](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Ciele_práce) časť

## Návrh automatizovaného testovacieho systému

### Architektúra

## Implementácia testovacieho systému

## Spracovanie výsledkov

# [Metodika práce a metódy skúmania](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Metodika_práce)

# [Výsledky práce a diskusia](C:\\Users\\DonChameleón\\Desktop\\Pokyny_pre_vypracovanie_ZP.docx" \l "Výsledky_práce)

## Výsledky práce

Podrobný popis postupov podľa metodiky riešenia.

## Diskusia

Výsledky, ktoré boli dosiahnuté riešením.

# [Záver](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Záver)

# [Zoznam použitej literatúry](file:///C:\Users\DonChameleón\Desktop\Pokyny_pre_vypracovanie_ZP.docx#Zoznam_použitej_literatúry)

# [Zoznam príloh](C:\\Users\\DonChameleón\\Desktop\\Pokyny_pre_vypracovanie_ZP.docx" \l "Prílohy)

**Príloha A** Názov

**Príloha B** Názov

# 

# Prílohy

## Príloha A: Názov prílohy

Každá ďalšia príloha začína na novej strane.

## Príloha B: Obsah DVD

Priložené DVD obsahuje:

* Práca v elektronickej podobe (formát PDF)