

Zabezpečenie kvality softvérových riešení

Jana Bajerovska
Matúš Balko

IBM Consulting

Kto sme ?

Jana Bajerovska

V IBM od 2014

Rola : Senior Test Manažér / Test Analytik

Venujem sa :

Test manažment, Test analýza, funkčný testing (Test FE aj BE). V minulosti test automation java, selenium.

Matúš Balko

V IBM od 2018

Rola : Application Developer/ Test Automation Engineer

Venujem sa :

Hľadanie riešení/toolov a príprava architektúry pre automatizovanie testovania, integrácia do CI/CD a nonfunctional testing (Performance, Parallel, Accesibility)

Obsah - Testing

- Čo si ľudia o testingu myslia – a nie je to tak úplne pravda
- Čo je to testing?
- Testing z pohľadu jednotlivých modelov vývoja
- Testovací proces a testovacie techniky
- Statický vs. Dynamický testing
- Test level a Test Typ

Čo si ľudia o testingu myslia – a nie je to tak úplne pravda 😊

- Testing je „ len klikanie „ (dynamické vykonávanie) pripravených testov.
- Je len čisto verifikačná činnosť – Kontrolujem či mi sedí aktuálny stav s tým, ktorý je zadaný v dokumentácii
- Tester nepotrebuje žiadne technické znalosti
- Tester nemusí vedieť programovať

Čo je to testing?

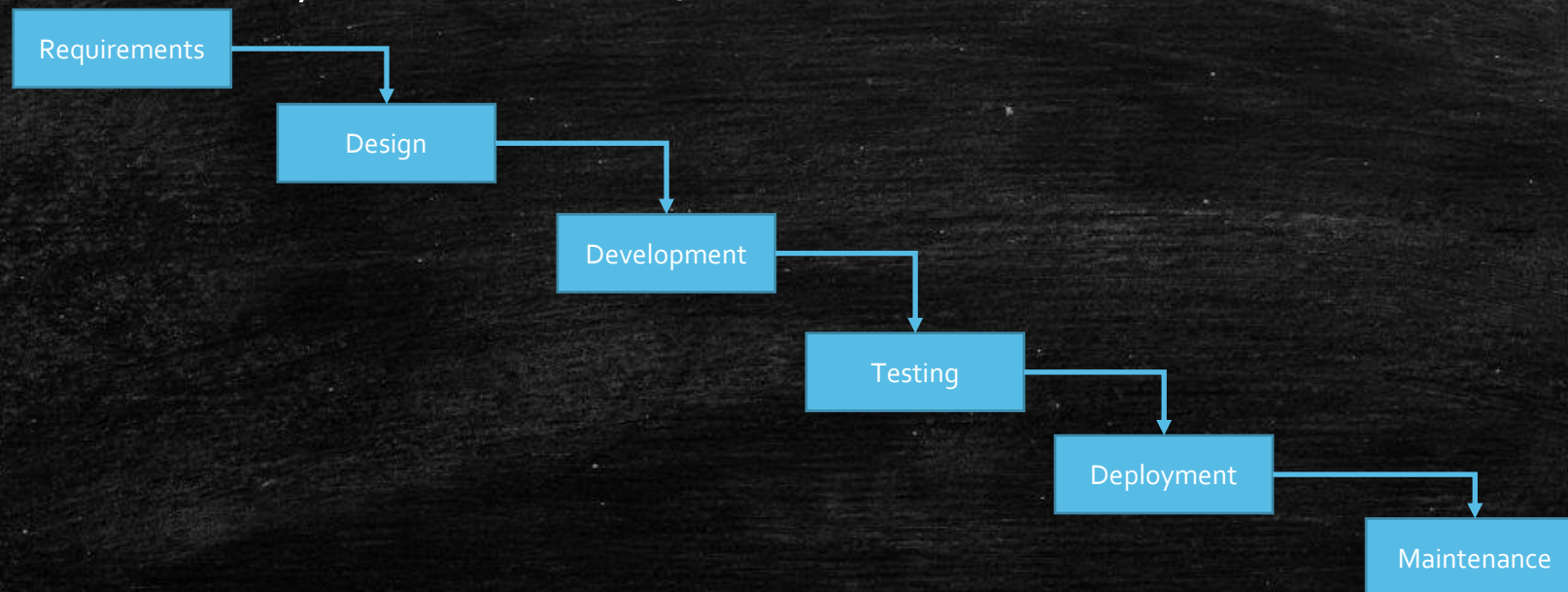
- Chybný softvér, ktorý nefunguje správne a podľa očakávaní užívateľov môže spôsobiť mnohé problémy ako sú :
 - Strata peňazí
 - Strata času
 - Strata dobrej povesti
 - Zranenia alebo smrť
- Správnym testovaním dokážeme znížiť riziko zlyhania softvéru

Testing je proces

- ISTQB – International Software Testing Qualifications Board (medzinárodný certifikačný systém v oblasti testovania softvéru) definuje testing ako:
- *„The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of a component or system and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.“*

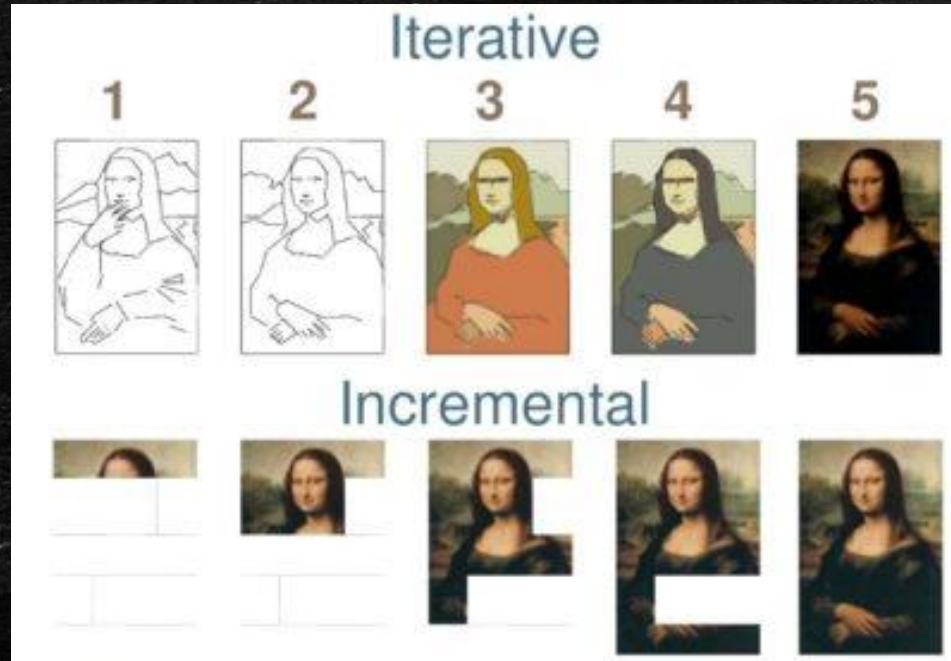
Testing z pohľadu jednotlivých modelov vývoja

- Sekvenčné vývojové modely
 - Lineárny / sekvenčný tok činností
 - Nová fáza môže začať až po ukončení tej predošlej
 - Na konci poskytujeme ucelený softvér , čo môže trvať mesiac alebo roky
 - Príklady : Waterfall model, V model



Testing z pohľadu jednotlivých modelov vývoja

- Iteratívne a inkrementálne vývojové modely
 - Vývoj po menších častiach
 - Príklady inkrementálneho vývoja : RAD
 - Príklady iteratívneho vývoja : RUP, Scrum, Kanban, Spiral model

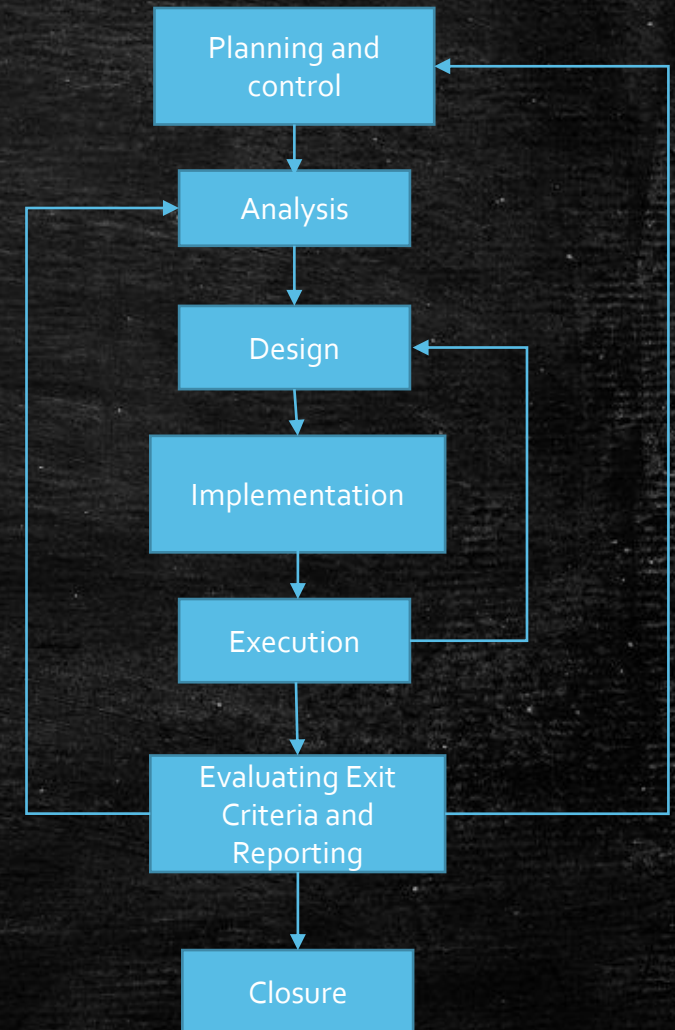


Testovacie Role podľa ISTQB

- Test Manager – riadi testovací process
- Test Analyst – analyzuje funkčné požiadavky
- Technical test analyst – analyzuje nefunkčne požiadavky

Testovací proces

- Testovací proces pozostáva činnosti :
 - Plánovanie testovania – Test manažér
 - Monitorovanie a riadenie testovania – Test manažér
 - Testovacia analýza – Test analytik
 - Dizajn / Návrh testov – Test analytik
 - Implementácia testov – Test analytik
 - Vykonávanie testov – Test analytik
 - Dokončenie testovania – Test manažér



Testovacie Techniky

- Testovacie techniky nám majú pomôcť identifikovať :
 - Testovacie podmienky
 - Testovacie prípady
 - Testovacie dáta

- Testovacie techniky závisia od :
 - Zložitosť komponentov alebo systému
 - Regulačné normy
 - Požiadavky zákazníka alebo zmluvné požiadavky
 - Úroveň a typy rizík
 - Dostupná dokumentácia
 - ...

Testovacie techniky

- ***Black-box – test techniques (behavioral or behavior-based)***
 - Tester analyzuje podklady pre testovanie (napr. dokumentáciu rozhrania, REQ, prípady použitia...) Platí pre funkčné aj nefunkčné testovanie Sústrediť sa na vstupy a výstupy testovaného objektu (bez sústredenia sa na vnútornú štruktúru)
 - *Equivalence Partitioning , Boundary Value Analysis , Decision Table Testing , State transition Testing , Use Case Testing*
- ***White-box test techniques (structural or structural-based techniques)***
 - Tester analyzuje architektúru, podrobný návrh, vnútornú štruktúru alebo kód testovaného objektu sústrediť sa na štruktúru a spracovanie v rámci testovacieho objektu
 - Statement Testing and coverage, Decision testing and coverage

Testovacie techniky

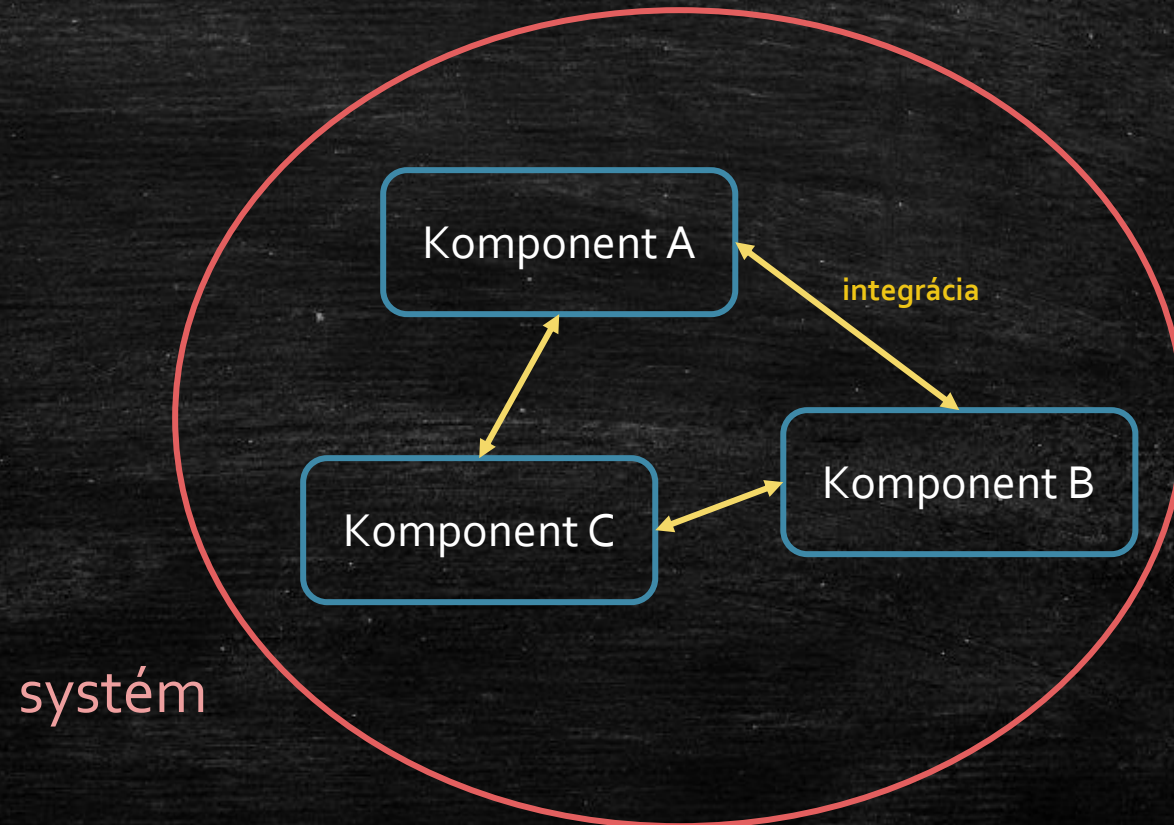
- ***Experience-based test techniques***
 - Na návrh testov potrebuje skúsenosti vývojárov, testerov a používateľov.
 - Error guessing, Exploratory Testing Checklist-based Testing

Statický vs. Dynamický testing

- Dynamický testing
 - je potrebné aby sme softvér spustili
- Statický testing
 - nie je potrebné spustenie softvéru - ide o skúmanie jednotlivých výstupov
 - Manuálne – napr. review
 - Pomocou nástrojov –napr. statická analýza
 - Dôležitý pri safety-critical systémoch a pri security testingu

Test level

- Testovací level je definovaný ako skupina súvisiacich testovacích činnosti, ktoré sú organizované a riadené spoločne
- Poznáme
 - Testing Komponentov
 - Integračný testing
 - Systémový testing
 - Akceptačne testovanie



Typy testingu

- Functional testing
 - Testujeme funkciu a hľadáme odpoveď na otázku : Čo daný softvér robí?
 - Na všetkých testovacích leveloch
- Non-functional testing
 - Testujeme charakteristiku systému a hľadáme odpoveď na otázku : Ako sa náš softvér chová ?

Obsah – automatizovaný testing

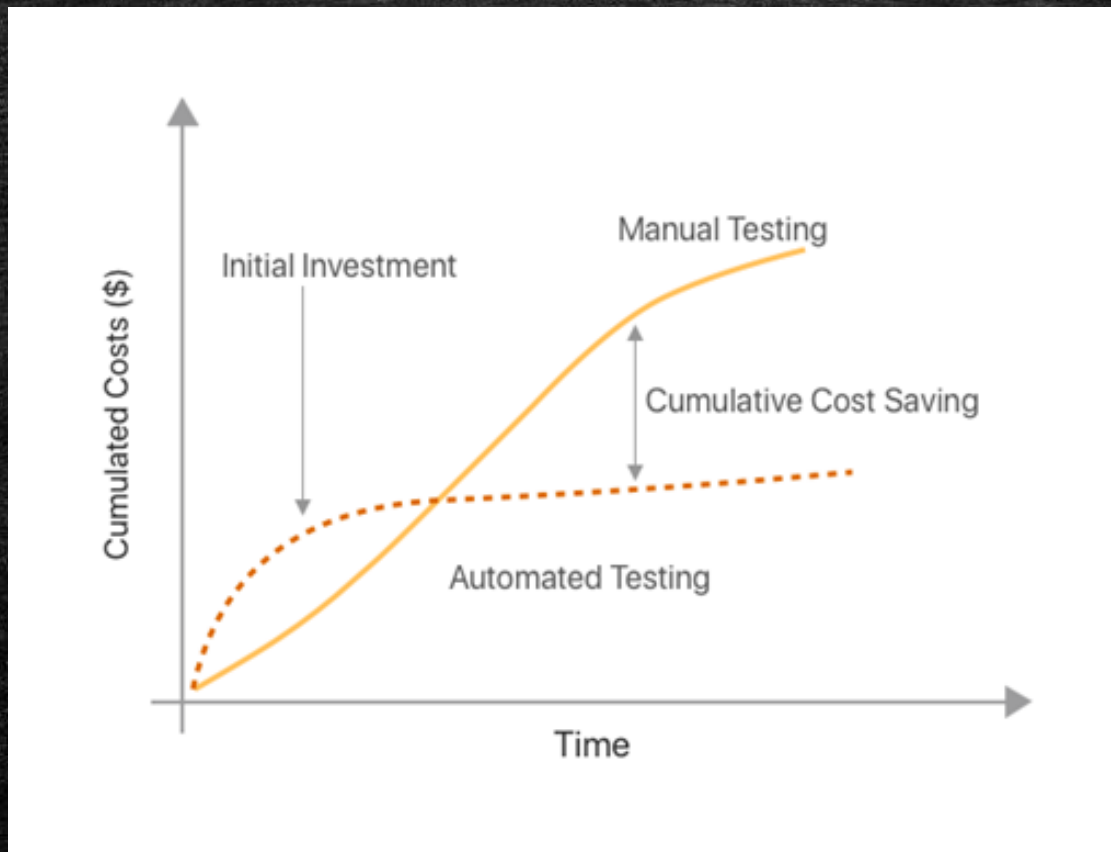
- Čo je to automatizovaný testing?
- Reálne benefity na projekte
- Výhody, nevýhody, podmienky pre úspech
- Čo patrí k povinnostiam TAE
- Výskum
- Nástroje
- Riešenia na mieru a Architektúra
- CI/CD, Príklady a Reporty

Čo je to automatizovaný testing?

Podľa definície ISTQB-TAE

- Automatizačný testing je softvérova testovacia technika pre porovnanie aktuálnych a očakávaných výsledkov.
- Urýchľuje kontrolu a nastavovanie predpokladov pre úspešné testovanie
- Urýchľuje spúšťanie testov

K čomu to je dobré?



Najväčšia starosť klientov sú budget, zdroje a úspešný produkt (zabezpečenie kvality)

Výhody, nevýhody, podmienky pre úspech

- Zvýšená efektivita testov
- Veľké pokrytie funkcionality
- Znížene náklady na testing
- Non-functional testing (Paralelné testy, Závažové testy, Stabilita ...)
- Možné častejšie spúšťanie testov

Pros & Cons

Čo patrí k povinnostiam TAE

1. Výskum
2. Nástroje
3. Riešenia na mieru, Architektúra a Dokumentácia
4. CI/CD

Výskum

- Ktoré komponenty systému potrebujeme zautomatizovať?
- Len backend? Len GUI(graphical user interface)?
 - Ktoré časti, aké procesy, akú máme kontrolu nad týmito časťami
- Aké endpointy máme k dispozícii?
- Čo klient vyžaduje ako súčasť AT riešenia?

Nástroje

- Aký budget máme k dispozícii?
- Splňuje požiadavky manažmentu a klienta “dá sa to predat”?
- Dostupnosť generovania reportov a logovania
- Aký skill majú jednotliví členovia AT tímu
 - Kvôli zvoleniu developmentu:
 - Linear (record-playback) – Katalon, Selenium ...
 - Keyword-driven – (skripty, definície) Robot Framework, Eclipse-red, RIDE...
 - Structural – (linear+libraries) JSON-LD, Yandex.Webmaster...
 - data-driven – Excel, Smartbear...
 - proces-driven/model-based (logic, data, scenarios) – gherkin cucumber

Riešenia na mieru a Architektúra

- Test automation solution TAS(Riešenie na mieru)
- Test automation architecture TAA(Architektúra)
- Využitie dostupných pluginov zvoleného toolu/IDE
- Dostupnosť knižníc a ich možná aplikácia/Nutnosť vývoja nových knižníc
- Konvencie
- Vyhodnotenie efektivity riešenia

Dokumentácia je nevyhnutnosť!!

CI/CD – continuous integration/continuous delivery

- Aplikácia automatu do pravidelného deploymentu
- Testovanie produktových inovácií a ich vplyv na stabilitu
- Podpora inkrementálneho vývoja a development modelov
- Využitie maximálneho potenciálu cloudu (pipeline, healthcheck, delivery support)

Redhat, IBM cloud, AWS, Azure ...

Príklady

- Keyword-driven – Python based – Robotframework

Resources > CustomLibraryExample.py > ...

```
1 from robot.api import logger
2
3
4 def print_me_this(str):
5     if str == 'abracadabra':
6         logger.console("Do some Magic")
7
```

28 ... Close browser

29

Run | Debug | Run in Interactive Console



Selenium Fluent Wait

```
31 ... [Documentation] ... lorem ipsum [https://www.google.com | Open page from report]
32 ... Open Browser ... https://www.google.com ... chrome
33 ... Wait And Click Element ... //button[@id='L2AGLb']
34 ... Wait And Input Text ... //input[@type='text'] ... Lorem
35 ... Press Keys ... //input[@type='text'] ... ENTER
36 ... Wait And Click Element ... //h3[contains(text(),'Lorem Ipsum Generator, Origins and Meaning')]
37 ... Wait And Click Element ... //button[text()='AGREE']
38 ... Wait And Click Element ... //span[contains(text(),'Images')]
39 ... Print me this ... msg
40 ... Sleep ... 5s
41 ... Close Browser
```

42

43 *** Keywords ***

Load in Interactive Console

44 Wait And Click Element

45 ... [Arguments] ... \${xpath}

46 ... Wait Until Keyword Succeeds ... 5x ... 2 seconds ... Click Element ... \${xpath}

47

Load in Interactive Console

48 Wait And Input Text

49 ... [Arguments] ... \${xpath} ... \${text}

50 ... Wait Until Keyword Succeeds ... 5x ... 2 seconds ... Input Text ... \${xpath} ... \${text}

Report

LOG

Test Details

All
Tags
Suites
Search

Status: 3 tests total, 2 passed, 1 failed, 0 skipped

Total Time: 00:00:58.041

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
exampleWait . Selenium Explicit Wait			FAIL	Element with locator 'xpath=//p[contains(text),'OK']' not found.	00:00:14.024	20220427 19:17:49.651 20220427 19:18:03.675
exampleWait . Implicit Wait Type	Variables taken from yml file		PASS		00:00:04.041	20220427 19:17:45.606 20220427 19:17:49.647
exampleWait . Selenium Fluent Wait	lorem ipsum Open page from report		PASS		00:00:39.976	20220427 19:18:03.678 20220427 19:18:43.654

	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
exampleWait	2	1	0	00:00:58	<div style="width: 100%; height: 10px; background: linear-gradient(to right, #90ee90 66%, #f08080 66%, #f08080 100%);"></div>

exampleWait

total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
3	2	1	0	00:01:02	<div style="width: 100%; height: 10px; background: linear-gradient(to right, #90ee90 66%, #f08080 66%, #f08080 100%);"></div>

Test Execution Log

- SUITE** exampleWait

Full Name: exampleWait

Source: C:\Users\MatusBalko\Desktop\RFPlayground\tests\exampleWait.robot

Start / End / Elapsed: 20220427 19:17:42.019 / 20220427 19:18:43.659 / 00:01:01.640

Status: 3 tests total, 2 passed, 1 failed, 0 skipped

00:01:01.640

 - TEST** Implicit Wait Type

00:00:04.041

 - TEST** Selenium Explicit Wait

00:00:14.024

 - TEST** Selenium Fluent Wait

00:00:39.976

Zdroje :

ISTQB foundation level

ISTQB Test Automation Engineer -

<https://www.istqb.org/certifications/test-automation-engineer>

ISTQB Glossary - <https://glossary.istqb.org/en/search/>

Otázky



?