

**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII**  
**Universitatea Tehnică a Moldovei**  
**Facultatea Calculatoare Informatică și Microelectronică**  
**Departamentul Ingineria Software și Automatică**

**Admis la susținere**  
**Șef de departament:**  
**Fiodorov I. dr., conf. univ.**

-----

**Analiza metodelor de testare a proceselor din cadrul sistemelor  
informaționale cu arhitectura orientată pe servicii**

**Teză de master**

**Student:** \_\_\_\_\_ **Ciorba Varvara, TI-211M**

**Coordonator:** \_\_\_\_\_ **Cojuhari Irina, dr., conf. univ.**

**Consultant:** \_\_\_\_\_ **Cojocaru Svetlana, asist. univ.**

**Chișinău, 2022**

## REZUMAT

**La teza de master: „Analiza metodelor de testare a proceselor din cadrul sistemelor informaționale cu arhitectura orientată pe servicii”, elaborat de Ciorba Varvara, Chișinău, 2023**

**Cuvinte cheie:** sistem orientat pe servicii, testarea software, testarea eficientă.

Prezentă lucrare urmărește scopul de cercetare a subiectelor din domeniul tehnologiei informație, precum sistemele informaționale cu arhitecturi orientate pe servicii și testarea software. Lucrarea este divizată în două părți. Prima parte este dedicată cercetării acestor subiecte, iar a doua parte reprezintă produsul cercetării și anume algoritmul de testare a proceselor din cadrul sistemelor cu arhitectura orientată pe servicii.

Componenta teoretică este formată din două capitole, unde fiecare capitol reprezintă unul din subiectele propuse spre cercetare. În primul capitol are loc analiza, cercetarea și concluzionarea celor studiate din tematica arhitecturii orientate pe servicii. Are loc evidențierea importanței a sistemelor cu arhitectura dată și descrierea acestora din trei puncte de vedere, precum utilizabilitatea, fiabilitatea și vulnerabilitatea. Al doilea capitol reflectă analiza, cercetarea și concluzionare a tehnicilor și metodelor de testare software. Se analizează momentele importante ce țin de testarea eficientă, factorii de determină dacă testarea este eficientă ori nu, se clasifică tehnicile și metodele de testare după mai multe categorii și descrierea fiecărei metode de testare, precum și mecanismele actuale de testare eficientă.

Componenta practică este compusă de un singur capitol. În urma cercetării aprofundate a subiectelor din partea teoretică, în capitolul dat are loc descrierea importanței creării algoritmului, prezentarea unei statistici comparative, descrierea algoritmului propriu-zis, demonstrarea algoritmului pe un proiect real cu descrierea eficienței algoritmului.

## ABSTRACT

**Master's thesis: "Analysis of process testing methods within information systems with service-oriented architecture", elaborated by Ciorba Varvara, Chisinau, 2023**

**Keywords:** service-oriented system, software testing, effective testing.

This paper aims to research information technology topics such as information systems with service-oriented architectures and software testing. The work is divided into two parts. The first part is dedicated to the research of these topics, and the second part represents the product of the research, namely the algorithm for testing processes within systems with service-oriented architecture.

The theoretical component consists of two chapters, where each chapter represents one of the topics proposed for research. In the first chapter, the analysis, research and conclusion of what was studied from the topic of service-oriented architecture takes place. It highlights the importance of systems with the given architecture and describes them from three points of view, such as usability, reliability and vulnerability. The second chapter reflects the analysis, research and conclusion of software testing techniques and methods. The important moments related to effective testing are analyzed, the factors that determine whether testing is effective or not, the techniques and methods of testing are classified according to several categories and the description of each testing method, as well as the current mechanisms of effective testing.

The practical component is composed of a single chapter. Following the in-depth research of the topics from the theoretical side, in the given chapter there is a description of the importance of creating the algorithm, a presentation of comparative statistics, a description of the algorithm itself, a demonstration of the algorithm on a real project with a description of the efficiency of the algorithm.

## CUPRINS

<b>LISTA TABELELOR, FIGURILOR, DIAGRAMELOR ȘI SCHEMELOR.....</b>	<b>8</b>
<b>INTRODUCERE.....</b>	<b>9</b>
<b>1 ANALIZA ȘI CERCETARE A SISTEMELOR ORIENTATE PE SERVICII .....</b>	<b>11</b>
1.1 Importanța sistemelor orientate pe servicii.....	12
1.2 Utilizabilitatea, fiabilitatea și vulnerabilitatea sistemelor orientate pe servicii.....	15
<b>2 ANALIZA ȘI CERCETAREA TEHNICILOR ȘI METODELOR DE TESTARE SOFTWARE</b>	<b>18</b>
2.1 Eficiența testării.....	20
2.2 Clasificarea tehnicilor și metodelor de testare.....	22
2.3 Mecanisme actuale de testare.....	33
2.4 Importanța creării algoritmului.....	39
<b>3 METODA DE TESTARE A PROCESELOR DIN CADRUL SISTEMELOR CU ARHITECTURA ORIENTATĂ PE SERVICII .....</b>	<b>42</b>
3.1 Descrierea algoritmului.....	43
3.2 Demonstrarea pe un exemplu concret.....	47
3.3 Eficiența algoritmului.....	50
<b>CONCLUZII.....</b>	<b>52</b>
<b>BIBLIOGRAFIE.....</b>	<b>53</b>
<b>ANEXE.....</b>	<b>54</b>
Anexa A (Suită de teste).....	54

## **LISTA TABELELOR, FIGURILOR, DIAGRAMELOR ȘI SCHEMELOR**

Figura 1.1 - Arhitectura orientată pe servicii

Tabelul 1.1 - Avantajele și dezavantajele ale sistemelor orientate către servicii

Figura 1.2 - Caracteristica arhitecturii orientate pe servicii

Figura 1.3 - Statistica preferințelor stilurilor arhitecturale a sistemelor informaționale

Tabelul 2.1 - Metodologiile de management al proiectelor

Figura 2.1 – Factorii care afectează eficiența testării

Figura 2.2 – Clasificarea după obiectul testării

Figura 2.3 – Clasificarea testelor după cunoștințe despre sistem

Figura 2.4 – Clasificarea testelor după gradul de automatizare

Figura 2.5 – Clasificarea testelor după gradul de izolare a componentelor

Figura 2.6 – Testarea de componente și de integrare

Figura 2.7 – Clasificarea testelor după momentul testării

Figura 2.8 – Clasificarea testelor după tipul de scenariu

Figura 2.9 – Clasificarea testelor după gradul de pregătire

Tabelul 3.1 - Stratouri de sistem

Figura 3.1 – Lucrul estimativ de testare a serviciilor

Figura 3.2 – Lucrul estimativ de raportare a defectelor

Figura 3.3 – Lucrul estimativ de fixare a defectelor

Figura 3.4 - Șablonul pentru tabel

Figura 3.5 - Exemplu de aplicare a algoritmului

Figura 3.6 - Testarea serviciului

Figura 3.7 - Testarea serviciului (continuare)

## INTRODUCERE

Arhitectura orientată pe servicii este deseori aplicată în crearea sistemelor informaționale de rang mare, pentru companii și întreprinderi mari, iar siguranța acestor sisteme este punctul cheie în alegerea stilului arhitectural. Se utilizează pentru a realiza servicii fiabile, mai bune, injectabile și reutilizabile. Sistemele orientate pe servicii, simplu, reprezintă o colecție de servicii slab cuplate și integrate într-o aplicație distribuită. Cu alte cuvinte, este o construcție sau principiul de construcție a sistemelor informaționale care are scop de asamblarea serviciilor într-o construcție și livrarea lor către clienți independent, sigur și calitativ.

Prezenta lucrare a fost elaborată în conformitate cu următoarele obiective propuse spre rezolvare, cum ar fi analiza și cercetarea sistemelor orientate pe servicii, analiza și cercetarea tehnicilor și metodelor de testare existente a produselor program și a celor cu arhitectura orientată pe servicii, principiile de testare a proceselor din cadrul sistemelor cu arhitectura orientată pe servicii, expunerea și justificarea creării noului algoritm care are ca scop creșterea eficienței a procesului de testare a sistemelor cu arhitectura de dezvoltare orientată pe servicii, îndeplinind parțial sau integral criteriile, precum financiar, uman, cronologic, calitatea produsului dezvoltat, astfel încât, asumându-le pe toate are loc atingerea așteptărilor ale clientului, ceea ce înseamnă că echipa obține succesul.

Algoritmul propus spre realizare reprezintă o metodă de testare a proceselor din cadrul sistemelor cu arhitectura orientată pe servicii. Metoda include o nouă organizare a procesului de testare prin crearea tabelului de vizualizare în timp real și continuu a situației de validare a serviciilor dezvoltate în cadrul echipei de dezvoltatori pas după pas, având mai multe avantaje, precum disciplinarea și organizarea activității ale specialiștilor de testare, implicarea activă a managerilor de proiect și a echipei de dezvoltatori, iar descrierea proceselor testate pe puncte de decizie mărește randamentul și productivitatea.

Cercetarea sistemelor orientate pe servicii are ca scop evidențierea importanței utilizării acestor arhitecturi în dezvoltarea produselor program, găsirea și explicarea punctelor forte și slabe din punctul de vedere a fiabilității și, bineînțeles, stabilirea în urma cercetării a vulnerabilității acestor sisteme.

Analiza și cercetarea tehnicilor și metodelor de testare software a avut ținta de aprofundare în materia clasificărilor metodelor și tehnicilor de testare, evidențierea importanței testării eficiente a produselor program și de expunere a celor mecanisme care influențează direct în eficientizarea întregului proces de testare, verificare și validare.

Importanța cercetării a domeniilor sistemelor orientate pe servicii și testarea software se conține în rezultatul studierii acestora, adică în dezvoltarea algoritmului de testare a sistemelor orientate pe servicii, care are ca scop mărirea eficacității întregului proces de testare. În lucrarea dată este descris algoritmul dezvoltat în urma cercetării, demonstrarea pe un exemplu concret și anume pe un sistem guvernamental real și, desigur, analiza și concluzionarea rezultatelor obținute.

Pentru a atinge scopul cercetării și elaborării a lucrării aferente prezentului raport au fost utilizate mai multe metode de cercetare relevante temei propuse, și anume metodele descriptivă și comparativă. Prin intermediul metodei descriptive au fost analizate materialele existente corelate cu tematica lucrării. Sursele de colectare a datelor au fost biblioteci digitale. O altă metodă utilizată este metoda comparativă, deoarece a fost considerat de o esență majoră de a compara tehnici și mecanisme de testare actuale și de formulat concluzia dacă este necesar și relevant de creat un nou mecanism de testare software.

În prezent există mai multe metode și tehnici de testare software care sunt aplicate cu devotament de specialiști, dar eficiența lor este pusă sub semnul întrebării din punctul de vedere al timpului de execuție și calității oferite. Răspunsul la această întrebare va fi punctul de pornire spre crearea și justificare noului mecanism de testare eficientă a sistemelor orientate pe servicii, care se bazează pe principiul de crearea a planului corect de testare, care va permite eficientizarea procesului de testare. E de menționat faptul că mecanismul de testare eficientă este apreciat de criterii, precum timp și calitate, sau cu alte cuvinte, numărul de defecte găsite, raportate, fixate și închise într-o unitate de timp. Cu cât mai multe defecte au fost raportate și remediate în cel mai scurt timp cu atât mai eficient este procesul de testare și de dezvoltare a produsului. Precum că, testarea urmărește obiectivul de validarea calității a produsului și influențează în mod direct reușita echipei de dezvoltatori, iar apoi și a companiei.

În această ordine de idei, crearea și utilizarea noului algoritm de testare va urma o serie de îmbunătățiri în cadrul ciclului de viață al produsului. Toate acestea vor fi descrise și explicate în următoarele capitole aferente prezentului proiect.

## Bibliografie

1. Software Development Community, What Is Service-Oriented Architecture? [citat 01.09.2022]. Disponibil:<https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec>
2. Ministerul Tehnologiei Informației și Comunicațiilor, Structura ministerului. [citat 05.09.2022]. Disponibil: <https://old.mtic.gov.md/ro/structura-0>
3. JetBrains s.r.o, Microservices. [citat 05.09.2022]. Disponibil: <https://www.jetbrains.com/lp/devecosystem-2021/microservices/>
4. Dorota Huizinga, Adam Kolawa, Automated Defect Prevention: Best Practices in Software Management. [citat 05.09.2022]. Disponibil: <https://www.wiley.com/enus/Automated+Defect+Prevention%3A+Best+Practices+in+Software+Management-p-9780470042120>
5. Software Testing Class, What is Test Effectiveness? . [citat 10.09.2022]. Disponibil: <https://www.softwaretestingclass.com/what-is-test-effectiveness-and-test-efficiency/#:~:text=Test%20Effectiveness%20can%20be%20defined,that%20meets%20the%20customer%20requirement.>
6. Thomas Hamilton, What is Compatibility Testing? Forward & Backward Example. [citat 10.09.2022]. Disponibil: <https://www.guru99.com/compatibility-testing.html>
7. Thomas Hamilton, Test Documentation in Software Testing (Example). [citat 15.09.2022]. Disponibil: <https://www.guru99.com/testing-documentation.html>
8. Katalon, Top 10 Continuous Testing Tools | 2022 Updated. ). [citat 15.09.2022]. Disponibil: <https://katalon.com/resources-center/blog/continuous-testing-tools#>