

ȘCOALA DOCTORALĂ INTERDISCIPLINARĂ

Facultatea: INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

Ing. Ligia Georgeta GUȘEILĂ

**INTEGRAREA SERVICIILOR DE TIP CLOUD
COMPUTING ÎN CENTRELE DE PRELUCRARE A
INFORMAȚIILOR**

**INTEGRATION OF CLOUD COMPUTING
SERVICES IN DATACENTERS**

REZUMAT / ABSTRACT

Conducător științific

Prof.dr.ing. Sorin-Aurel MORARU

BRAȘOV, 2019

D-lui (D-nei)

COMPONENȚA

Comisiei de doctorat

Numită prin ordinul Rectorului Universității Transilvania din Brașov

Nr. din

PREȘEDINTE:

Conf.dr.ing. Carmen GERIGAN
Universitatea Transilvania din Brașov

CONDUCĂTOR ȘTIINȚIFIC:

Prof.dr.ing. Sorin-Aurel MORARU
Universitatea Transilvania din Brașov

REFERENȚI:

Prof.dr.ing. Theodor BORANGIU
Universitatea Politehnica din București
Prof.dr.ing. Decebal POPESCU
Universitatea Politehnica din București
Conf.dr.ing. Liviu PERNIU
Universitatea Transilvania din Brașov

Data, ora și locul susținerii publice a tezei de doctorat: 20.09.2019, ora, sala

Eventualele aprecieri sau observații asupra conținutului lucrării vor fi transmise electronic, în timp util, pe adresa ligia.guseila@unitbv.ro

Totodată, vă invităm să luați parte la ședința publică de susținere a tezei de doctorat.

Vă mulțumim

Cuprins

	Pag. ează	Pag. rezumat
Listă de figuri	6	
Listă de tabele	7	
1. Introducere	8	3
1.1 Importanța și actualitatea temei	8	
1.2 Obiectivele tezei	10	4
1.3 Prezentarea capitolelor tezei	12	
2. Stadiul actual al cercetărilor în domeniu	14	6
2.1 Cloud Computing – fundament al transformării digitale	14	6
2.2 Cloud Computing	16	6
2.3 Modele de implementare Cloud Computing	18	7
2.4 Modele de livrare a serviciilor	20	7
2.5 Rolurile și modul de interacțiune în Cloud Computing	23	
2.6 Tehnologia	25	8
2.7 Securitatea în Cloud Computing	26	9
2.8 Caracteristicile Cloud Computing	28	10
2.9 Interoperabilitatea în Cloud	31	
2.10 Concluzii	32	11
3. Arhitectura Cloud Computing și modele de proiectare	33	11
3.1 Arhitectura orientată pe servicii	33	11
3.2 Virtualizarea	36	12
3.3 Containerele	38	12
3.4 Microserviciile	43	14
3.5 Serviciile REST-ful	48	16
3.6 Infrastructura definită software	50	16
3.7 Arhitectura aplicațiilor fără server	52	
3.8 Livrarea software automată	54	
3.9 Concluzii	55	

4.	Strategia de analiză, de dezvoltare și de implementare a serviciilor software	56	18
4.1	Tiparele de arhitectură și de proiectare pentru aplicațiile Cloud	56	18
4.2	Arhitectura de referință Cloud	61	19
4.3	Provocări în proiectarea aplicațiilor Cloud	64	
4.4	Aplicații tradiționale în domeniul IT	66	20
4.5	Strategia de migrare în Cloud	67	20
4.6	Analiza de maturitate Cloud pentru aplicații tradiționale	69	21
4.7	Procesul de migrare în Cloud Computing pentru aplicațiile tradiționale	75	22
4.8	Studiu de caz	77	23
4.9	Concluzii	85	26
5.	Automatizarea procesului de integrare a serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor	87	27
5.1	Managementul ciclului de viață în dezvoltarea serviciilor software	87	27
5.2	Modele de dezvoltare software	88	27
5.3	Ciclul de livrare software agil	92	27
5.4	Analiza comparativă a ciclului de dezvoltare tradițional și agil	95	
5.5	Cloud și DevOps	97	27
5.6	Adoptarea DevOps în dezvoltarea aplicațiilor software	100	
5.7	Procesele DevOps	103	29
5.8	Cerințele de dezvoltare pentru implementarea automată în Cloud	106	
5.9	Tehnologia	106	
5.10	Instrumentele	107	
5.11	Modelarea procesului de dezvoltare software	111	29
5.12	Analiza DevOps – studiu de caz	113	29
5.13	Soluția de implementare	122	33
5.14	Implementarea CI/CD	124	35

5.15	Implementarea modului de testare automată	126	36
5.16	Procesul automat de publicare și de implementare software	130	39
5.17	Monitorizarea sistemului software și a infrastructurii	132	40
5.18	Scalarea sistemului	133	41
5.19	Beneficiile implementării soluției propuse	133	41
5.20	Concluzii	135	42
6.	Optimizarea sistemului propus folosind inteligența artificială	137	44
6.1	Monitorizarea performanței aplicației	137	44
6.2	Automatizarea proceselor de business	141	45
6.3	Învățarea automată	143	46
6.4	Învățarea în adâncime	145	47
6.5	Concluzii	148	49
7.	Concluzii generale, realizări și contribuții originale, direcții viitoare de cercetare și diseminare	150	50
7.1	Concluzii generale	150	50
7.2	Contribuții originale	152	52
7.3	Direcții viitoare de cercetare	154	54
7.4	Lista publicațiilor	155	54
7.5	Valorificarea rezultatelor cercetării	156	56
	Bibliografie	157	57
	Anexe	161	59
	Anexa 1. Rezumat/Abstract	161	59
	Anexa 2. Curriculum Vitae	162	60
	Anexa 3. Declarație de autenticitate	166	

1. Introducere

Globalizarea aduce cu sine noi provocări pe care organizațiile de astăzi încearcă să le soluționeze. Necesitatea de a accesa putere de calcul crescută și de a procesa volume mari de date reprezintă astăzi o provocare pentru organizații și tocmai această abilitate de a procesa volume mari de date este văzută ca un avantaj față de competitorii existenți. Modelul tradițional de operare a serviciilor IT prin accesarea centrelor de date devine, în ritmul evoluției volumului de date aflat în continuă creștere, un model extrem de costisitor.

Cloud Computing vine ca o soluție de business la provocările lumii moderne enunțate mai sus. Cloud Computing a cunoscut în ultimele două decenii o dezvoltare extraordinară, reușind să capteze atât interesul specialiștilor din domeniul IT, dar și a numeroase altor persoane din diferite domenii de activitate. Avantajele aduse de către tehnologia Cloud Computing sunt date de ofertarea resurselor IT sub forma unor servicii, de care beneficiarii se pot bucura la momentul, în forma, în cantitatea și la calitatea de care aceștia au nevoie, independent de poziția geografică a beneficiarului sau a furnizorului de servicii Cloud.

Lucrarea este structurată în 7 capitole, conținând 160 de pagini care cuprind 50 de figuri, 9 tabele și 75 de referințe bibliografice.

Obiectivul general al lucrării vizează evidențierea conceptelor pentru integrarea serviciilor software de tip Cloud Computing în centrele de prelucrare a informațiilor, adoptarea unei soluții viabile de arhitectură pentru folosirea serviciilor în Cloud, identificarea tehnologiilor actuale și a mecanismelor ce pot incorpora caracteristicile conceptelor definite.

Lucrarea include informații despre detaliile tehnice referitoare la orientarea tehnologică și metode folosite în implementarea arhitecturii și infrastructurii necesare serviciilor în Cloud Computing.

Capitolul 1 prezintă introducerea în domeniu, prin evidențierea importanței utilizării tehnologiilor noi de tip Cloud Computing și introducerea acestora în centrele de prelucrare a informațiilor.

Capitolul 2 realizează o introducere în stadiul actual al cercetării în Cloud Computing cu evidențierea modelelor folosite în implementarea Cloud-ului, arhitectura care stă la baza Cloud, dar și caracteristicile care asigură utilizarea de servicii Cloud.

În capitolul 3, am realizat un studiu pentru serviciile software utilizate în Cloud, prin evidențierea arhitecturii și a tehnologiilor care stau la baza Cloud Computing.

Capitolul 4 prezintă strategia de adoptare a serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor din cadrul unei organizații. S-a propus și s-a realizat o metodă de evaluare a aplicațiilor de tip *legacy* în vederea transformării și migrării lor în Cloud, s-au evidențiat procesele și pașii de urmat în cadrul acestei transformări și s-a propus un mecanism de evaluare standardizat.

În capitolul 5 se prezintă ciclul de viață al aplicațiilor software și se evidențiază necesitatea utilizării metodologiei DevOps în procesul de dezvoltare al aplicațiilor software folosind un model de colaborare între departamentele organizației. Prin folosirea metodologiei DevOps se urmărește automatizarea procesului de dezvoltare și de publicare a codului software în mediile de producție de tip Cloud.

Capitolul 6 concentrează direcții tehnologice noi prin integrarea de mecanisme avansate în vederea optimizării soluției propuse.

În capitolul 7 sunt prezentate concluziile generale ale acestei teze, precum și contribuțiile originale aduse, lucrările publicate în care au fost diseminate rezultatele cercetării efectuate pe parcursul anilor de studiu, dar și o scurtă prezentare a viitoarelor direcții de cercetare.

1.1 Obiectivele tezei

Portabilitatea aplicațiilor este un factor cheie în cadrul platformelor, care asigură suport pentru o varietate de infrastructuri locale. Lumea noastră se află într-o continuă schimbare, iar satisfacerea cerințelor de business devine strâns legată de migrarea activității către unul din modelele de Cloud. În consecință, identificarea și adoptarea modelelor eficiente pentru procesarea volumului mare de date este de mare interes.

Motivația lucrării de cercetare pentru tema aleasă în vederea integrării serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor, a fost de a analiza și de a integra cele mai noi tendințe de tehnologie și de arhitectură în dezvoltarea aplicațiilor software. Această cercetare a luat în calcul temele emergente, care vin în sprijinul cercetării realizate și includ:

- Modificări culturale cu scopul de a promova colaborarea și empatia între oameni, precum și înlăturarea fenomenului de „silozii” din cadrul aceleiași organizații – există numeroase echipe, care nu colaborează cu alte departamente din cadrul aceleiași organizații, ținând toate cunoștințele și informațiile interne departamentului din care fac parte. Acest obiectiv poate fi atins prin furnizarea unor mijloace îmbunătățite de colaborare ce asigură interacțiuni audio-video în timp real, punându-se un accent deosebit pe clădirea empatiei între colegi și împărtășirea informațiilor.
- Automatizarea infrastructurii la nivel fizic, de platforma și software, precum și a fluxului de date în aplicații, pentru a asigura elasticitatea și reutilizarea serviciilor.
- Nevoia existenței unei platforme de management hibride, care facilitează publicarea și livrarea aplicațiilor și a funcționalităților noi în Cloud, luând în calcul schimbarea continuă a cerințelor de business sau ale utilizatorilor finali cu privire la procesul de business folosit.
- Mijloace de verificare și de implementare de tehnologie nouă, care poate face față unor medii distribuite, care asigură un grad ridicat de elasticitate a serviciilor, se bazează pe automatizare și care includ metrici organizaționale pentru a susține procesul de adopție a serviciilor de tip Cloud Computing.

Lucrarea își propune să abordeze subiectele enumerate mai sus, propunând și detaliind modelele de implementare Cloud, analizând și urmărind implementarea serviciilor de tip Cloud Computing pe parcursul întregului proces de transformare digitală. Mai mult decât atât, rezultatele aplicării acestor modele vor fi analizate și interpretate în vederea enunțării unor concluzii pe baza studiilor de caz și a implementării experimentale realizate.

Obiectivul general al lucrării vizează evidențierea conceptelor pentru integrarea serviciilor software de tip Cloud Computing în centrele de prelucrare a informațiilor, adoptarea unei soluții viabile de arhitectură pentru folosirea serviciilor în Cloud, identificarea tehnologiilor actuale și a mecanismelor ce pot incorpora caracteristicile conceptelor definite. De asemenea se urmărește validarea aspectelor teoretice prin implementarea unei soluții software automate pentru publicarea aplicațiilor software în Cloud. Modelul propus este conceput pe o platformă de tip *open-source* și permite gestionarea resurselor, pentru o mai bună interconectare a sistemelor IT din mediul Cloud.

O serie de soluții originale sunt valorificate în cadrul proiectului “NOT Alone at Home – NOAH” - Cooperare Europeana și Internațională AAL - Active Assistive Living, și al proiectul PN-III-P2-2.1-PTE-2016-0064 ”CON-INTEL – Măsurarea consolidată și transmiterea parametrilor energetici spre punctele de colectare” - Finanțat de UEFISCDI, [Perniu et al, 2019].

De asemenea, această teză de doctorat are următoarele **obiective specifice**:
OS1. Primul obiectiv specific constă în analiza stadiului actual tehnologic din domeniul IT pentru utilizarea Cloud Computing în centrele de prelucrare a informațiilor, având ca specific

identificarea de modele de implementare a serviciilor în Cloud Computing. Este evaluată tehnologia necesară, caracteristicile unei implementări de servicii de tip Cloud în centrele de prelucrare a informațiilor, dar și avantajele sau provocările, care trebuie luate în calcul în vederea implementării Cloud Computing.

OS2. Al doilea obiectiv specific are în vedere definirea, evaluarea și soluționarea unui model de arhitectură, care să susțină modelul de livrare de servicii software de tip Cloud Computing. Trebuie luate în calcul interoperabilitatea serviciilor, tehnologiile noi necesare, care stau la baza implementării unei arhitecturii pentru servicii distribuite, dar și utilitățile necesare pentru adoptarea modelului optim de Cloud. În cadrul acestui obiectiv se va realiza o evaluare a Cloud Computing din punct de vedere arhitectură și infrastructură tehnică prin prisma abordării unei soluții de livrare a serviciilor de tip Cloud Computing. Obiectivul propune definirea arhitecturii și identificarea cerințelor de tehnologie necesare pentru implementarea Cloud ca soluție inovativă. Va introduce modele hibride pentru a susține diferite necesități de business și va propune modelarea aplicațiilor Cloud pentru a facilita utilizarea serviciilor, independent de schimbările de infrastructură, de platformă sau de cerințe business în Cloud.

OS3. Al treilea obiectiv al acestei lucrări presupune definirea unei strategii de adopție și de implementare a aplicațiilor software în Cloud Computing. Se vor lua în calcul atât modelarea și dezvoltarea aplicațiilor Cloud, cât și procesele de business existente, dar și strategia de evaluare a aplicațiilor într-o organizație pentru migrarea aplicațiilor existente în Cloud. Soluția trebuie să ofere caracteristicile care stau la baza Cloud pentru oferirea de servicii flexibile, disponibile oricând, scalabile în funcție de necesitatea business și trebuie să ofere o viteză de răspuns ridicată la cererea utilizatorilor. Integrarea de servicii și funcționalități în Cloud trebuie să vizeze atât aplicații Cloud, cât și aplicații existente, să prezinte și să verifice procesul de adopție Cloud Computing și să asigure calitatea și disponibilitatea serviciilor.

OS4. Al patrulea obiectiv este analiza și identificarea posibilităților de optimizare a proceselor de dezvoltare software, dar și de publicare a serviciilor în Cloud având în vedere că publicarea și dezvoltarea aplicațiilor software în model tradițional duce la o disponibilitate întârziată a funcționalităților în mediile de producție.

OS5. Al cincilea obiectiv este automatizarea procesului de migrare și de adopție a serviciilor Cloud prin stabilirea unor metode, procese și metodologii de dezvoltare software, precum și prin automatizarea proceselor de livrare a aplicațiilor sau funcționalităților noi în Cloud cu o viteză ridicată. Acest proces automatizat de dezvoltare trebuie să conțină o platforma automată, hibridă, care să asigure calitatea codului software produs, publicat la orice moment în Cloud. Este necesară identificarea arhitecturii platformei hibride, implementarea și evaluarea acesteia, dar și testarea automată a aplicațiilor înainte de publicarea aplicațiilor în Cloud. Se va implementa un mecanism automat de livrare a aplicațiilor software în Cloud, care este independent de limbajul de programare în care aplicația software a fost dezvoltată.

OS6. Cel de-al șaselea obiectiv constă în analiza modelului propus și a capacității acestuia de a satisface necesitățile organizațiilor de dezvoltare continuă. Se vor căuta metode de optimizare a soluției propuse de integrare a aplicațiilor software în Cloud și metode de optimizare a sistemului implementat prin utilizarea inteligenței artificiale.

Lucrarea include informații despre detaliile tehnice referitoare la orientarea tehnologică și metode folosite în implementarea arhitecturii și infrastructurii necesare serviciilor în Cloud Computing. Soluția propusă va crește productivitatea sistemelor din centrele de date existente sau a celor în curs de dezvoltare, va reduce costurile prin implementarea Cloud-ului bazat tehnologiile inovative propuse, va optimiza și va reutiliza consumul de resurse prin virtualizarea infrastructurii și va reduce timpul de procesare a informațiilor prin creșterea puterii de calcul.

2. Stadiul actual al cercetărilor în domeniu

2.1 Cloud Computing – fundament al transformării digitale

Apariția Cloud Computing-ului a fost dictată, și totodată reușește să îmbine caracteristicile esențiale ale celor patru sisteme de calcul: sistemele utilitare („Utility Computing”), sistemele la cerere („On-demand Computing”), sistemele de calcul independent („Autonomic Computing”) și IT-ul verde („Green IT”).

Evoluția către Cloud Computing a venit spre sfârșitul anilor 1980 odată cu apariția conceptului de Grid Computing, când pentru prima dată, un număr mare de sisteme de calcul a fost implicat în soluționarea unei singure probleme de natură științifică, și a cărei soluționare a cerut un nivel înalt de procesare a datelor într-un regim sincron și paralel.

Sistemele de calcul la cerere, uneori numite și sisteme utilitare, se axează pe alocarea dinamică sau pe elasticitatea a resurselor de calcul, pentru a satisface eficient cererile în continuă schimbare ale clienților. Resursele sistemelor de calcul la cerere pot fi localizate în cadrul unei organizații sau pot fi externalizate către un terț, denumit și furnizor de servicii.

Cloud Computing a luat naștere ca o evoluție firească de la formele inițiale de calcul de volum mare de date. Printre aceste se remarcă:

Răspândirea fenomenului relativ nou de transformare digitală constrânge organizațiile IT să își transforme modelele de business actuale pentru a putea adăuga plus-valoare serviciilor oferite clienților lor. Deși această transformare a creat oportunități pentru definirea unor modele de business noi și inovative, companiile existente sunt deseori forțate să opereze în cadrul unor modele învechite. Acest lucru reprezintă o piedică majoră în fața capacității de a răspunde rapid la cerințele utilizatorilor, ducând într-un final la deprecierea valorii organizației de piață.

Având în vedere aceste date, organizațiile au înțeles că trebuie să fie capabile de o restructurare rapidă și de o remodelare a modelelor de business pentru a putea răspunde la dinamica cerințelor utilizatorilor sau la diferiți factori externi. Modelul de Cloud reprezintă bazele unei noi lumi dinamice de business. Este platforma ideală pentru implementarea unui ciclu de dezvoltare agil al aplicațiilor. Infrastructura bazată pe Cloud reprezintă cheia furnizării unui acces flexibil, și la cerere al resurselor, piloni pe care se sprijină noile oferte de business. Totodată le permite organizațiilor să poată extinde nivelul de infrastructură pentru a face față dinamicii priorităților de business, reducând riscurile de a irosi resurse IT și înlăturând astfel unul din principalele impedimente ce stăteau în calea investițiilor în noile servicii digitale.

În contextul noii economii digitale din zilele noastre, companiile sunt dependente de utilizarea eficientă a tehnologiei, nu doar pentru a asigura procesele de business curente, dar și pentru generarea de noi surse de venit și care să ofere o diferențiere față de competitorii de pe piață.

2.2 Cloud Computing

De-a lungul anilor în industria IT au fost prezentate mai multe definiții ale Cloud Computing în încercarea de a înțelege ce reprezintă Cloud-ul. Cele mai multe dintre acestea fac trimitere la rețelele de Internet interconectate care sunt reprezentate sub forma unui nor.

Există trei modele acceptate de implementare Cloud Computing: Cloud public, Cloud privat și Cloud hibrid. Accesul la Cloud poate oferi servicii la nivel de infrastructură,

platformă sau aplicație software, oferind posibilitatea utilizatorilor să beneficieze de cele cinci caracteristici esențiale ale acestuia.

Cloud Computing poate fi văzut ca un model de business, prin care tehnologia este oferită ca un serviciu online, utilizatorii accesând astfel aplicațiile software prin intermediul Internetului, fără a deține licența asupra lor, în schimbul unei taxe lunare.

Sistemul Cloud îmbunătățește viteza de calcul a informațiilor din centrele de date tradiționale, și duce la creșterea performanței și a nivelului de securitate din sistemele IT.

Cloud Computing este modelul de livrare a resurselor din domeniul informației tehnologice sub forma de serviciu prin intermediul conexiunii la rețea. Cloud Computing este un mecanism de procesare a informației, în care resursele hardware sunt virtualizate și alocate dinamic, fiind ulterior oferite clienților spre utilizare ca serviciu IT prin intermediul conexiunii la Internet.

Cloud-ul este văzut ca un sistem de calcul distribuit la nivel geografic, care face posibilă partajarea resurselor, selecția și agregarea dinamică a puterii de calcul sub forma unor servicii "autonome" la momentul cererii utilizatorului. În Cloud are loc agregarea resurselor la momentul execuției prin verificarea și validarea resurselor în funcție de disponibilitatea serviciilor, capacitatea acestora, performanța și costul aferent utilizării acestora.

Beneficiile date de utilizarea Cloud Computing sunt evidențiate de caracteristicile acestuia: accesul facil la serviciile Cloud, elasticitatea serviciilor oferite, care pot fi accesate la cerere, și cuantificarea serviciilor în funcție de consum.

2.3 Modele de implementare Cloud Computing

Cloud Computing permite patru modele de implementare în funcție de necesitate.

- Cloud-ul privat

În mod obișnuit un Cloud privat este proiectat și condus de către departamentul IT al unei organizații, cu scopul de a livra servicii organizației respective. Cloud-ul privat poate fi adăpostit în centrul de date intern al organizației sau într-un centru de date anexat organizației, sub stricta supraveghere și respectare a proceselor și procedurilor existente în organizație.

- Cloud-ul public

Resursele computaționale ale Cloud-ului public sunt puse la dispoziția publicului de larg consum. Cloud-ul public poate fi deținut, condus și întreținut de către o organizație guvernamentală, academică sau business sau o combinație a acestora. Poate exista numai pe premisa unui furnizor de Cloud. Serviciile și aplicațiile dintr-un Cloud public sunt puse la dispoziția publicului, aplicațiile Cloud-ului accesând o infrastructură mixtă a mai multor furnizori de Cloud.

- Cloud-ul de comunitate

În cazul Cloud-ului de comunitate, resursele computaționale sunt puse în comun de către două sau mai multe organizații cu scopul de a atinge anumite obiective de business, sau cerințe specifice. Echipamentele IT sunt localizate fie într-un centru de date intern, sau unul extern, sub atenta supraveghere a partenerilor asociați. Cloud-ul de comunitate prezintă avantajele reducerii costurilor de mentenanță și operaționale și reușește să limiteze potențialele riscuri de securitate.

- Cloud-ul hibrid

Serviciile oferite de către Cloud-ul hibrid reprezintă o compunere a două sau mai multe tipuri de Cloud distincte - privat sau public - care există ca entități unice, dar care sunt interconectate prin tehnologii standard sau specifice, și care permit portabilitatea datelor sau a aplicațiilor.

2.4 Modele de livrare a serviciilor

Cloud Computing are la bază arhitectura orientată pe servicii. Arhitectura orientată pe servicii presupune distribuirea funcționalității aplicației în unități mai mici, distincte, numite servicii.

Serviciile care pot fi distribuite într-o rețea și pot fi utilizate împreună pentru a crea aplicații și funcționalități destinate beneficiarilor.

- Software ca serviciu (SaaS)

În cazul modelului de livrare Software ca un serviciu (SaaS), serviciile sunt accesate de către beneficiar prin intermediul unui portal web sau a unei aplicații, care este vizibilă utilizatorului. Acesta este singurul nivel, care interacționează direct cu beneficiarul de servicii Cloud.

Serviciul la nivel software este cel mai familiar model de livrare Cloud pentru utilizatorii de servicii Web. Modelul oferă aplicații software utilizatorilor, care de obicei nu cunosc și nu au control asupra infrastructurii sau asupra platformei. Utilizatorii nu dețin aplicația software pe care o utilizează. Aplicațiile care rulează în Cloud sunt furnizate la cerere sub formă de servicii utilizatorilor, fără a fi necesară achiziționarea componentelor hardware sau a licențelor aferente produselor.

- Platforma ca un serviciu (PaaS)

Metodă de livrare Cloud, platforma ca un serviciu, oferă toate resursele necesare pentru a construi aplicații și servicii disponibile prin Internet, fără a fi necesară descărcarea sau instalarea de aplicații suplimentare.

PaaS îi are ca țintă pe dezvoltatorii de aplicații, fiind un nivel poziționat între infrastructura virtualizată și aplicațiile software. Soluțiile bazate pe PaaS sunt ușor de implementat deoarece utilizatorului i se oferă o platformă și uneltele necesare pentru dezvoltarea aplicațiilor, fără a ține cont de problemele legate de infrastructura hardware ce stă la bază. Platforma susține securitatea, elasticitatea, stocarea, administrarea și versionarea aplicației.

- Infrastructura ca serviciu (IaaS)

Nivelul de bază al Cloud-ului este nivelul de infrastructură livrat ca un serviciu. Acesta se bazează pe o serie de resurse fizice, cum ar fi serverele, echipamentele de rețea sau sistemele de stocare, etc., care sunt oferite ca un serviciu consumatorului.

În mod asemănător cu modelul de livrare PaaS, virtualizarea este o metodă folosită frecvent pentru furnizarea resurselor hardware. Serviciile de infrastructură sunt componentele care echipează centrul de date prin asigurarea unei puteri crescute de calcul. În plus, datorită tehnicilor de virtualizare folosite la nivel de infrastructură, costurile cu infrastructura pot fi reduse considerabil.

2.5 Tehnologia

Cloud Computing utilizează într-un mod optim alocarea dinamică a resurselor cu scopul de a livra utilizatorilor un număr mare de servicii diverse. Utilizatorii se pot concentra pe dezvoltarea aplicațiilor, externalizând rularea și execuția proceselor în sistem către furnizorii de Cloud.

- Virtualizarea

Tehnologia principală care stă la bazele Cloud Computing este virtualizarea și a fost dezvoltată de către IBM în anii 1960 prin folosirea sistemelor *mainframe*.

Recent, modelul de livrare Cloud Computing cunoscut ca software ca un serviciu (Software as a Service) a ridicat virtualizarea la nivel de aplicație, ca un model de business prin care plata serviciilor se realizează în funcție de resursele consumate și în funcție de tipul aplicației software utilizate.

Bazele virtualizării sunt reprezentate de emularea mai multor calculatoare pe aceeași mașină fizică. Acest tip de virtualizare este cunoscut drept virtualizare completă întrucât permite instalarea unui sistem de operare în cadrul izolat al unui mediu virtual.

Conceptul de virtualizare nu se limitează doar la emularea unui calculator în ansamblul său. Există numeroase tehnologii de virtualizare specifice altor tipuri de resurse, cum ar fi mediile de stocare, bazele de date, memoria sistemului, echipamentele de rețea. Fiecare din ele adaugă un strat de abstractizare între resursele virtualizate și resursele fizice. Un sistem bazat pe Cloud poate utiliza mai multe tehnologii de virtualizare în cadrul său, fiecare din ele având interfețe specifice definite pentru managementul resurselor virtualizate.

- Containere

În modelul de Cloud, virtualizarea reprezintă tehnica de bază pentru atingerea parametrilor de elasticitate și a resurselor distribuite la scară largă [Mell & Grance, 2011]. Așa cum mașinile virtuale reprezintă elementele de bază pentru infrastructura în Cloud, containerele permit o virtualizare la nivel mai mic, prin compunerea caracteristicilor lor ca și pachete de aplicații provenite din imagini individuale. Ele necesită un consum redus de resurse și de timp. Containerizarea se bazează pe capabilitatea de a dezvolta, testa și implementa aplicațiile pentru funcționare. Containerele se utilizează la nivelul PaaS din Cloud.

- Automatizarea

Automatizarea activităților de mentenanță și operaționale ale serviciilor Cloud reprezintă o caracteristică importantă pentru furnizorul de servicii Cloud. Automatizarea se referă la sistemele de management, de operațiuni și de mentenanță ale resurselor din Cloud. Procesul de management al resurselor în Cloud trebuie să fie complet automatizat și să aibă un timp de răspuns minim.

- Monitorizarea

Resursele și serviciile Cloud trebuie supravegheate constant în vederea asigurării disponibilității acestora. Pentru acestea sunt folosite diferite utilitare automate care pot asigura în permanență monitorizarea infrastructurii, a platformei și a datelor sau a proceselor de business. Monitorizarea automată este necesară în livrarea serviciilor Cloud la calitatea așteptată de către beneficiar.

2.6 Securitatea în Cloud Computing

În ultimii ani s-a evidențiat strategia multor organizații de a migra procesele de business în Cloud. Acestea pot include adeseori și procese ce utilizează date financiare, personale sau cu caracter senzitiv.

Securitatea datelor și a informațiilor în Cloud este și astăzi una dintre temele cele mai provocatoare în adoptarea soluției Cloud Computing. Înainte de a externaliza date sau informații în mediul public, orice organizație trebuie să se asigure că toate măsurile de securitate, de integritate și de consistență a datelor au fost luate.

În acest sens furnizorul de Cloud trebuie să asigure siguranța și securitatea informațiilor la fiecare nivel în Cloud, de la nivelul de infrastructură și până la cel de servicii software.

Se disting astfel măsurile la nivelele Cloud Computing:

- Securitatea la nivelul infrastructurii Cloud
Întrucât puterea de procesare este compusă din resursele de infrastructură a unuia sau mai multe centre de date, este importantă asigurarea securității serviciilor, a componentelor și a căilor de comunicare prin care datele sunt transmise.
- Securitatea la nivel de platformă Cloud
Pentru a asigura securitatea datelor la nivel de platformă, este esențial ca sistemele de gestionare și de procesare a datelor să fie menținute la versiunea actuală, iar traficul de date suspicios să fie întrerupt imediat. Aici sunt definite la nivel de organizație diverse politici și regulamente de securizare a datelor.
- Securitatea la nivel de software în Cloud
Furnizorul trebuie să protejeze întregul portofoliu de aplicații împotriva amenințărilor interne și externe, începând cu faza de proiectare și până când aplicațiile sunt lansate în producție. Securitatea trebuie asigurată pe durata întregului ciclu de viață. De asemenea, este important ca politicile de securitate și procesele definite să nu devină piedici în fața funcționalităților de business și să nu introducă alte riscuri sau dificultăți pentru clienți și furnizori.

2.7 Concluzii

Cloud Computing reprezintă fundamentul transformării digitale pentru orice organizație a zilelor noastre. Cloud-ul vine nu numai cu soluții la nivel tehnologic, dar și cu politici de preț atractive. Utilizarea aplicațiilor mobile, dar și creșterea dispozitivelor interconectate vor duce la o adoptare exponențială a tehnologiilor Cloud.

Majoritatea implementărilor de Cloud vor utiliza sub o formă sau alta modele hibride, în care sistemele Cloud privat vor fi integrate, fie cu sisteme similare, fie cu modele de implementare de tip Cloud public. Înțelegerea ambelor perspective, a furnizorului dar și cea a utilizatorului de Cloud, dar și a modelelor de dezvoltare și de livrare a serviciilor prin prezentarea avantajelor și a limitărilor Cloud este esențială în vederea realizării unei implementări de succes.

Componentele de infrastructură, de platformă și serviciile software trebuie să asigure un grad ridicat de elasticitate și trebuie să poată răspunde cerințelor interne și externe ale organizațiilor. Prin utilizarea de servicii Cloud Computing este permis accesul la resurse computaționale nelimitate, disponibile la cerere și foarte flexibile.

3. Arhitectura Cloud Computing și modele de proiectare

Capitolul 3 va pune accentul pe arhitectura software în implementarea serviciilor de tip Cloud Computing și se va concentra pe modelele de design pentru crearea de aplicații software Cloud durabile și fiabile.

Conceptele esențiale în crearea mediilor Cloud Computing eterogene se bazează pe arhitectura orientată pe servicii și pe tehnologia virtualizării. Astfel, lucrarea va introduce conceptele de proiectare și beneficiile raportate prin utilizarea acestora în vederea implementării mediului Cloud. Proiectarea de Cloud se va realiza folosind cele mai noi tehnologii de pe piața IT în utilizarea Cloud-ului.

În zilele noastre, principalul motiv pentru a adopta Cloud Computing față de implementarea IT tradițională a unui centru de date este stabilitatea, flexibilitatea, fiabilitatea și elasticitatea sistemelor IT odată cu furnizarea rapidă a serviciilor.

Cloud Computing se bazează ca model de arhitectură în livrarea serviciilor software pe arhitectura orientată spre servicii, arhitectură care partajează și reutilizează funcționalități ale aplicațiilor prin intermediul accesului la rețea.

Caracteristica de a pune la dispoziție serviciul IT prin intermediul rețelei de Internet unei alte componente interconectate la rețea, ajută la reutilizarea serviciilor.

Calitatea serviciilor în Cloud Computing este definită de toate caracteristicile funcționale și nefuncționale ale serviciilor software, care aflate în uz pot fi evaluate și măsurate în timpul furnizării serviciilor.

3.1 Arhitectura orientată pe servicii

Timp de câteva decenii, dezvoltarea de software a necesitat utilizarea elementelor funcționale modulare, care execută o anumită activitate. Având în vedere că integrarea aplicațiilor erau dependente de infrastructură, stabilirea mediului în care va avea loc execuția aplicațiilor a reprezentat o problemă, ce a necesitat soluționare. S-a identificat nevoia definirii unui model de dezvoltare a aplicațiilor, bazat pe niște proceduri de bune practici pentru utilizarea componentelor aflate la distanță și distribuite geografic. Soluția la această problemă a fost enunțată prin redefinirea și partajarea vechiului model de livrare al serviciilor dintr-o structură mare și complexă în structuri mici, unitare, ce pot fi puse la dispoziție unei aplicații, prin intermediul infrastructurii distribuite..

Arhitectura care a permis partajarea sistemelor de mari dimensiuni în servicii modulare pentru a sprijini re-utilizarea serviciilor în condiții de securitate și de conectivitate, a fost denumită arhitectura orientată pe servicii (*Software Oriented Architecture, SOA*). Serviciile sunt distribuite în rețea și pot fi accesate împreună cu scopul de a construi aplicații software.

Arhitectura orientată pe servicii este un model de dezvoltare software pentru componentele distribuite ale aplicațiilor, care asigură descoperirea serviciilor în rețea, controlul accesului la date, securizarea datelor și asigură îndeplinirea cerințelor de securitate.

Într-o arhitectură orientată spre servicii, serviciile pot fi combinate cu alte servicii disponibile în rețea prin intermediul procedurii de orchestrare a serviciilor. Un serviciu este reutilizabil, specific și poate fi descoperit dinamic în întreaga întreprindere, în sistemele partenere sau în mediile distribuit de tip Cloud Computing. Aceste caracteristici permit ca serviciile să fie cuplate în mod liber, rezultând noi aplicații proiectate.

Există două părți implicate în comunicarea la nivelul arhitecturii orientate pe servicii: utilizatorul de servicii și furnizorul de servicii.

Arhitectura orientată pe servicii este un tip de arhitectură, concepută structural pentru implementarea proceselor de business incluse ca servicii pe tot parcursul ciclului de viață al aplicațiilor. Serviciile sunt dezvoltate agil și se pot adapta cu ușurință la schimbările cerute de organizație. Toate acestea pot fi utilizate pentru a spori eficiența unei organizații, pentru a aduce produsele software mult mai repede pe piață, pentru a sprijini eficiența costurilor prin reutilizarea serviciilor existente.

Întrucât, Cloud Computing reprezintă trecerea de la computerele tradiționale la medii informatice distribuite pentru a furniza resurse flexibile și scalabile, arhitectura orientată pe servicii reprezintă metoda de proiectare cea mai bună, care realizează conexiunea optimă dintre utilizatori și serviciile Cloud.

3.2 Virtualizarea

Datorită creșterii rapide a volumului de date utilizate în centrele de date, se înregistrează o cerere crescută a resurselor fizice. Toate acestea duc la o creștere continuă a consumului de energie și putere în centrele de date.

Virtualizarea este tehnologia, care permite operarea simultană a două sau mai multe sisteme de operare pe o singură mașină fizică. Managerul mașinii virtuale (*Virtual Machine Manager, VMM*), sau *hypervisor-ul*, este un mecanism software, care permite platformei să găzduiască mai multe sisteme de operare, care rulează simultan pe același server fizic. Virtualizarea împarte serverul fizic într-un număr mai mare de resurse cu scopul furnizării de servicii utilizatorilor finali.

Virtualizarea permite abstractizarea și izolarea funcționalităților la nivel fizic și a componentelor la nivel hardware. Aceasta permite portabilitatea funcțiilor la nivelele superioare și partajarea și / sau agregarea resurselor fizice. Virtualizarea ajută la utilizarea la nivel optim de încărcare a componentelor fizice existente în centrele de date.

Industria IT se concentrează asupra virtualizării ca soluție IT pentru a susține adoptarea Cloud Computing, în timp ce utilizarea calculatoarelor virtuale și a tehnologiei de virtualizare oferă avantaje ca maximizarea utilizării resurselor de calcul, și reușește în același timp să reducă cheltuielile asociate de mentenanță a echipamentelor, a consumului de energie și de întreținere, dar și a spațiului fizic destinat depozitării echipamentelor de infrastructură.

3.3 Containerele

Un container împachetează codul software și toate fișierele de configurare al acestuia, astfel încât aplicația să poată rula rapid și fiabil într-un mediu de calcul. Containerul este un pachet software ușor de utilizat, care conține toate elementele necesare pentru ca o aplicație să poate fi executată: limbaj cod, proces de execuție, unelte de sistem, biblioteci de sistem și setări. Un container poate opera aplicații Linux și Windows, indiferent de infrastructură.

Mașinile virtuale pot utiliza mai multe resurse de sistem, deoarece fiecare mașina virtuală rulează, nu doar o copie completă a unui sistem de operare, ci o copie virtuală a întregului hardware pe care sistemul de operare este executat. Acest lucru oferă viteză prin accesul direct la memoria și la procesorul sistemului. Utilizarea containerelor poate fi văzută ca o metodă optimizată față de execuția aceluiași proces pe o mașină fizică. Trebuie însă luată în calcul necesitatea fiecărei aplicații software.

Containerele, necesită resurse minime în execuția acestora. Spre deosebire de mașinile virtuale, containerele nu au nevoie de un sistem de operare dedicat, care să fie instalat în container și nu au nevoie de o copie virtuală a hardware-ului serverului gazdă. Containerele

pot opera cu cantitatea minimă de resurse pentru a îndeplini sarcina pentru care au fost proiectate.

Din punct de vedere al software-ului și al securității acestuia, containerele oferă un mare avantaj în asigurarea stabilității sistemului de operare de bază.

În funcție de obiectivele de proiectare și de punere în aplicare a acestora, containerele clasificate în containere de aplicație și containere de sistem.

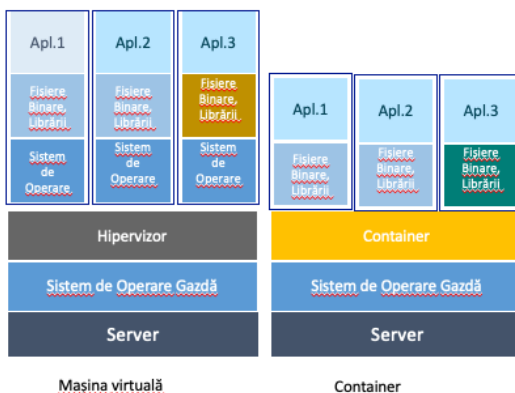


Fig. 3.3.1 Mașina virtuală (stânga) vs. Container (dreapta)

Orchestrarea containerului se referă atât la pornirea sau oprirea aplicațiilor, cât și la mutarea acestora de la un server la altul. Procesul de orchestrare a containerului constă în construirea și gestionarea continuă a aplicațiilor sau a unor grupuri distribuite de aplicații software folosind containere.

Instrumentele de orchestrare ale containerelor pot fi adoptate în general pentru a oferi un cadru de integrare și gestionare a containerelor la scară largă

Kubernetes și Docker Swarm sunt cele mai frecvent utilizate platforme pentru orchestrarea containerelor. Acestea sunt capabile să orchestreze infrastructura de calcul, de rețea și de stocare în numele utilizatorilor.

Tabelul 3.3.2 prezintă o diferențiere clară între mașinile virtuale și containere.

Tabel 3.3.1 Comparația dintre mașinile virtuale și containere

Mașinile virtuale	Containerele
Greoaie în utilizare	Facile în utilizare
Performanță limitată	Performanță nativă Cloud
Fiecare MV rulează pe propriul sistem de operare	Toate containerele împart sistemul de operare gazdă
Virtualizarea se face la nivel de hardware	Virtualizare la sistem de operare
Viteză de execuție scăzută	Viteză de execuție crescută
Alocă memoria necesară	Necesită mai puțin spațiu de memorie
Complet izolate, securitate ridicată	Izolare la nivel de proces, securitate scăzută

În configurația ideală, ambele tehnologii, atât virtualizarea cât și containerele pot fi incluse în procesul de arhitectură și de design al aplicațiilor, luând în calcul flexibilitatea ridicată a mașinilor virtuale, dar și cerințele minime de resurse necesare containerelor. Atât mașinile virtuale, cât și containerele pot funcționa împreună și oferi medii de execuție stabile distribuite și securizate pentru aplicațiile software de tip Cloud.

3.4 Microserviciile

Utilizarea microserviciilor în proiectarea și în dezvoltarea aplicațiilor reprezintă o nouă tendință de arhitectură pentru serviciile software din domeniul IT.

Microserviciile reprezintă o tehnică de dezvoltare a aplicațiilor software, care structurează o aplicație ca o colecție de servicii cuplate în mod liber. În arhitectura bazată pe microservicii, serviciile sunt descompuse în unități mai mici pentru a îmbunătăți modularitatea lor. Acestea sunt dezvoltate, testate și implementate ca module singulare de funcționalitate a aplicațiilor. Întrucât, acestea sunt organizate în jurul proceselor de business și sunt dezvoltate independent, ele devin în mod liber cuplate și trebuie interconectate.

Capacitatea microserviciilor de a fi executate independent unele de altele este folosită mai ales în livrarea continuă de software programat. Microserviciile pot fi scalate independent de alte servicii și pot fi implementate pe hardware-ul asignat nevoilor acestora. Mai mult decât atât, datorită dimensiunii lor reduse, acestea sunt mai ușor de întreținut și sunt mai tolerante la erori, deoarece defectarea unui serviciu nu va întrerupe execuția întregului sistem, ca în cazul unui sistem monolitic [Lewis & Fowler, 2014].

Industria IT corelează arhitectura bazată pe microservicii cu tehnologiile containerelor de aplicații, care simplifică implementarea automată la nivel de aplicație. Containerele pot prezenta încredere pentru construirea unor astfel de unități de arhitectura bazată pe microservicii prin execuția autonomă a serviciilor la momentul rulării serviciului.

Pentru a beneficia de caracteristicile arhitecturii bazate pe microservicii, trebuie rezolvate câteva aspecte importante la momentul proiectării și implementării arhitecturii software. Printre acestea se enumeră orchestrarea și coordonarea serviciilor, strategiile de implementare ale microserviciilor și stocarea datelor.

Avantajele implementării microserviciilor în medii virtuale sau în containere asigură scalabilitatea serviciului, făcând sistemele flexibile la utilizarea unui număr mare de microservicii.

Managementul decentralizat și gestionarea datelor permit serviciilor să fie independente și să evite nevoia de a standardiza scrierea codului sursa într-un singur limbaj de programare.

Caracteristica de livrare a codului software în regim independent, izolarea față de celelalte servicii și funcțiile ușor de implementat și de mentenanță a serviciilor, cât și funcțiile de scalare și de auto-vindecare a acestora, fac microserviciile să devină un model de arhitectură adoptat la scară largă în vederea dezvoltării aplicațiilor de tip Cloud Computing.

De asemenea, ușurința de a menține și de a refolosi microserviciile dezvoltate, sunt principalele beneficii pentru care ele sunt abordate în dezvoltarea aplicațiilor de tip Cloud. Apariția tiparelor de implementare este un prim pas către o strategie de dezvoltare, de integrare și de implementare a aplicațiilor software folosind metodologia DevOps. Publicarea microserviciilor trebuie făcută prin intermediul utilizării containerelor sau a mașinilor virtuale.

3.5 Serviciile REST-ful

Transferul de stat reprezentativ (REST) este un stil arhitectural software care definește un set de caracteristici, care trebuie utilizate pentru crearea serviciilor Web. Serviciile Web care sunt conforme cu stilul arhitectural REST, denumite servicii Web RESTful (RWS), asigură interoperabilitatea între sistemele informatice legate la rețeaua de Internet. Serviciile Web RESTful permit sistemelor care accesează serviciile să manipuleze reprezentările textuale ale resurselor Web prin utilizarea unui set uniform și predefinit de operațiuni. Alte tipuri de servicii Web, cum ar fi serviciile SOAP Web, expun propriile seturi de operații arbitrare.

Resursele web au fost definite pentru prima dată pe World Wide Web ca documente sau fișiere identificate prin URL-urile lor.

Stilul arhitectural REST este proiectat să utilizeze un protocol de comunicare, de tip HTTP. În stilul arhitecturii REST, clienții și serverele fac schimb de reprezentări ale resurselor utilizând o interfață și un protocol standardizat.

Prin utilizarea unui protocol și a unor operațiuni standard, sistemele RESTful urmăresc o performanță ridicată a sistemului, fiabilitatea și capacitatea de creștere a sistemului prin reutilizarea componentelor. Aceste componente pot fi gestionate și actualizate în orice moment fără a afecta sistemul în ansamblu. Aceste modificări pot fi făcute, chiar și în timp ce sistemul este în funcțiune.

Într-o arhitectură RESTful, clienții și serverele sunt ușor integrate și implementate pe platforme principale. Mai mult decât atât, serviciile Web RESTful sunt cunoscute ca fiind foarte scalabile.

Principalele avantaje ale utilizării serviciilor RESTful se datorează scalabilității, flexibilității și portabilității în diferite medii, dar și datorită independenței serviciilor. Protocolul REST se distinge prin scalabilitatea acestuia. Datorită separării dintre client și server, produsul poate fi scalat de o echipă de dezvoltare fără mari dificultăți.

Flexibilitatea, scalabilitatea și portabilitatea sunt asigurate prin separarea dintre client și server. Se poate efectua o migrare de la un server la altul, sau se pot efectua modificări în baza de date în orice moment, fără a se afecta al doilea participant de comunicare, deoarece transferul de date este asigurat. Separarea se realizează mai ușor la nivel de servere, ceea ce face ca aplicațiile să fie mai flexibile la momentul execuției.

Întrucât serviciile de tip Cloud Computing sunt distribuite în rețea, utilizarea de către clientul final a API-urilor RESTful bazate pe servicii web reprezintă o soluție optimă pentru consumul de la distanță de servicii și de prelucrare a datelor.

REST facilitează microserviciile colaborarea și comunicarea între servicii și sunt astăzi cea mai bună practică în proiectarea soluțiilor la nivel SaaS în Cloud.

În timp ce arhitectura bazată pe microservicii implică mai multe aplicații dezvoltate individual pentru a lucra împreună, REST este servicii web bazat pe HTTP care permite comunicarea între aplicații. Astfel, se consideră că acest tip de arhitectura este utilizat frecvent în dezvoltarea aplicațiilor native web dedicate Cloud-ului.

Prin adoptarea acestor noi paradigme, orice organizație va putea beneficia de beneficiile transformării digitale susținute de Cloud, oferind flexibilitate și adaptabilitate schimbărilor de business.

3.6 Livrarea software automată

Livrarea continuă este o practică de dezvoltare software, în care schimbările de cod sunt pregătite automat pentru lansarea în producție.

Livrarea continuă (*Continuous Delivery*, CD) este procesul de construire, de testare, de configurare și de implementare a codului software dintr-un mediu de dezvoltare într-un mediu de producție. Există mai multe medii de testare sau de integrare pentru a crea o conductă de publicare DevOps și pentru a automatiza crearea infrastructurii și implementarea uneia noi. Mediile succesive suportă activitățile de integrare, de integrare și de testare, dar și de aprobare a codului de către utilizatorii finali.

Practicile de livrare continuă pot economisi mult timp și pot reduce costurile de livrare software în proiectele de dezvoltare. Echipele de dezvoltare pot fi mai sigure că software-ul produs a fost testat și pregătit corespunzător pentru mediul de producție, atunci când testarea și implementarea automată a avut loc cu succes, și conducta DevOps a livrat codul în mediile de producție. Un avantaj al utilizării livrării continue este faptul că un client poate testa codul la fiecare fază de dezvoltare software, astfel încât acesta poate fi mult mai devreme implicat în procesul de dezvoltare și de livrare al produsului software. Livrarea continuă poate fi implementată în etape, procesele de suport în vederea automatizării și orchestrării au cel mai mare impact în vederea publicării aplicațiilor în mediile de producție și oferă cea mai mare rentabilitate pentru banii și timpul investit în implementare. Abordarea și implementarea conducerii DevOps va fi analizată în continuare și prezentată în capitolul 5.

Utilizarea aplicațiilor Cloud-native ale serviciilor cuplate în mod liber prin intermediul microserviciilor se recomandă să se facă prin intermediul mecanismului de livrare continuă.

Cloud Computing are un impact major asupra multor domenii ale tehnologiei și ale arhitecturii IT. Livrarea continuă face parte din această schimbare, încurajând mai puține erori în sistemele de producție, un client mai implicat, care poate oferi feedback imediat cu privire la rezultatele de acceptanță a funcționalităților noi software, dar și un mai bun control, o bună conformitate și calitatea ridicată a serviciilor livrate. Modelele prezentate sunt o necesitate de urmat pentru a oferi aplicații Cloud de o calitate înaltă a serviciilor.

4. Strategia de analiză, de dezvoltare și de implementare a serviciilor software

Arhitecturile și procesele de software moderne pot face organizațiile să devină mai flexibile și mai pregătite pentru a răspunde acestor schimbări, în timp ce le ajută să rămână competitive în industria în care activează.

În capitolele anterioare s-au prezentat caracteristicile Cloud Computing, arhitectura Cloud și cele mai noi tehnologii folosite în domeniul IT cu scopul de a aduce o referință de bază în procesul de proiectare, de analiză și de implementare a serviciilor software de tip Cloud Computing.

În acest capitol, se va realiza un mecanism de evaluare a aplicațiilor software existente în mișcarea strategică de adopție Cloud. Se va construi un utilitar care va ajuta organizațiile în evaluarea portofoliului de aplicații existente în transformarea digitală folosind Cloud Computing.

Strategia de adopție a aplicațiilor de tip Cloud se va realiza după arhitecturi software moderne. În plus se va descrie o modalitate de modernizare a aplicațiilor existente, de tip *legacy* pentru a putea funcționa în Cloud. Se vor utiliza de microservicii, containere, se va planifica optimizarea proceselor încât flexibilitate dorită și automatizarea sa fie posibilă la momentul integrării aplicațiilor în Cloud.

4.1 Tiparele de arhitectură și de proiectare pentru aplicațiile Cloud

Termenul de aplicație Cloud nativă a început să aibă importanță după 2012 și a fost văzută ca o continuare a arhitecturii bazate pe microservicii și a aplicațiilor bazate pe containere. Există mai mulți cercetători care identifică aplicațiile potrivite Cloud ca fiind „native” Cloud. Aplicațiile native de tip Cloud au ca și caracteristică de bază aplicații bazate pe microservicii și containere. Aceste microservicii asigură elasticitatea și adaptabilitatea rapidă a aplicațiilor software la necesitățile clienților aflate în continuă schimbare.

Principiile care trebuie luate în calcul în vederea creării de aplicațiile native Cloud trebuie să poată fi executate pe platforme automate, să fie programate și să pună în mod automat infrastructura necesară rulării aplicației la dispoziție și să poată oferi elasticitate sistemului. Aplicațiile dezvoltate Cloud trebuie să poată opera pe mai multe tipuri de platforme. În acest scop, aplicațiile Cloud trebuie să fie concepute ținând cont de infrastructura care stă la bază și de componentele *middleware*.

Arhitectura unui serviciu Cloud trebuie să se bazeze pe containere și microservicii. Aceste două elemente pot să furnizeze proprietăți de flexibilitate, de elasticitate și reziliență în mediul Cloud.

Operarea aplicațiilor Cloud necesită o platformă existentă și flexibilă. Aici majoritatea IT-iștilor sugerează folosirea Kubernetes sau Docker Swarm ca platformă pentru implementare aplicațiilor Cloud.

Proiectarea aplicațiilor native Cloud se face în jurul microserviciilor și prin utilizarea de tehnologii noi bazate pe containere, ambele fiind factori importanți în implementarea aplicațiilor Cloud.

Designul aplicației Cloud se află în strânsă legătură cu aceste modele. Platformele de automatizare sunt adesea menționate alături de subiecte arhitecturale folosind microservicii, de compoziția serviciului software și componentele de implementare.

Proiectarea unei arhitecturi sustenabile pentru aplicația Cloud este influențată de modelele de implementare ale acesteia. Microserviciile sunt la rândul lor construite pe unități

de implementare de sine stătătoare, de tip container, care să fie disponibile atunci când sunt executate.

Autorul [Fehling et al., 2014] propune ca o aplicație Cloud să fie creată după principiul **IDEAL**, astfel încât aplicația să aibă o **stare (i)zolată**, să fie **(d)istribuită** prin natura ei în rețea, să fie **(e)lastică** și scalabilă pe orizontală, operată prin intermediul unui sistem de **gestionare (a)utomat. Servicii(1)e** sale trebuie să fie **independente** unele de altele [Fehling et al., 2014].

S-a identificat necesitatea existenței unei arhitecturi ale aplicației Cloud native care să permită livrarea de soluții de software mai rapide. Aplicație Cloud trebuie să fie tolerantă la erori și stabilă, să permită automatizarea și scalarea aplicației în funcție de cererile utilizatorilor. Aplicație Cloud trebuie să se poată adapta diferitelor tipuri de platformă și de sisteme existente pe piață.

Pentru implementarea aplicațiilor native Cloud, arhitectura aplicațiilor Cloud și infrastructura utilizată trebuie să îndeplinească următoarele cerințe:

- Trebuie să utilizeze microservicii. Microserviciile pot fi utilizate la descompunerea sistemelor monolitice în servicii implementate independent, care fac „un singur lucru bine”;
- Principalul mod de interacțiune între serviciile, care folosesc arhitecturi native Cloud, este prin intermediul API-urilor publicate și versionate. Se realizează colaborarea între servicii prin API. API-urile sunt adesea bazate pe protocoale HTTP și urmează un stil de arhitectură de tip REST, dar și alte protocoale și formaturi de serializare pot fi folosite;
- Unitățile de implementare singulare ale arhitecturii sunt create și interconectate conform unei colecții de modele specifice Cloud.

Aplicând elementele prezentate în dezvoltarea de aplicații software, organizațiile pot trece la implementarea unor medii Cloud. Aplicațiile native Cloud pot fi distribuite, asigură scalabilitatea serviciilor pe orizontală, sunt sigure și flexibile în conformitate cu nevoile business.

4.2 Arhitectura de referință Cloud

Arhitectura Cloud implementată integrează soluțiile de tip container, cum ar fi Docker, mediile intermediare de infrastructură Cloud, cum ar fi medii de rulare în Cloud bazate pe containere și scalabile de genul Kubernetes și se va realiza dezvoltarea de aplicații folosind microservicii. Toate aceste componente de proiectare în vederea implementării aplicațiilor Cloud native identificate sunt importante pentru a construi soluții tehnologice care creează independență de furnizor.

Dat fiind că arhitectura aplicațiilor Cloud începe cu nivelul inferior, la nivel de infrastructură, primul nivel arhitectural va cuprinde serverele fizice și containerele.

Serverele fizice au fost utilizate în modelul tradițional ca și medii de implementare pentru aplicații. Pentru aplicațiile de tip Cloud Computing serverele vor să fi virtualizate și distribuite în rețea. La nivelul stratului inferior se vor utiliza containere, în cadrul cărora vor fi executate aplicațiile.

- **Nivelul 1, infrastructura fizică**

Nivelul de infrastructură fizică face analogia dintre Cloud Computing cu modelul fizic OSI. În Cloud accesul la acest nivel este disponibil numai furnizorului de Cloud.

- **Nivelul 2, infrastructura virtuală**

Nivelul 2 furnizează mecanismul de virtualizare pentru infrastructura Cloud. Stratul pune la dispoziție mașini virtuale, care sunt conectate la rețeaua furnizorului de Cloud.

Mașinile pot fi configurate pentru a fi accesibile prin Internet folosind adrese de IP și porturi configurabile. Mașinile gestionează spațiul pe disc, puterea de calcul, etc. furnizate de infrastructura Cloud.

- **Nivelul 3, containerului gazdă**

În majoritatea cazurilor, sistemele de operare standard sunt instalate pe mașini virtuale pentru a opera și pentru a instala și configura pachete software adiționale. Sistemele de operare tipice utilizate în Cloud Computing sunt adesea sistemele Linux/Unix, dar pot fi posibile și sisteme de operare arbitrare. Cu toate acestea, există o tendință de a folosi tehnologii de tip container pentru a simplifica implementarea și pentru a minimiza incompatibilitățile date de diferite aplicații care rulează pe aceeași mașină. Soluțiile de tip container, de obicei Docker, furnizează un mediu de execuție autonom, format din imaginea și din sistemul de *build* al containerului.

Integrarea containerului trebuie luată în considerare pentru a putea manevra mai multe containere ca pe un singur nod de calcul logic. Acest tip de tehnologii reprezintă adesea coloana vertebrală tehnologică pentru soluțiile la nivel de platformă în Cloud și pentru mediile portabile de execuție. Containerele sunt portabile deoarece ele pot fi implementate pe orice tip infrastructură Cloud.

- **Nivelul 4, cluster-ul**

La nivelul cluster-ului apar în mediul IT mai multe substraturi. Acestea sunt programatorul, rețeaua și integratorul. Ei sunt foarte interconectați și comunică între ei cu scopul de a furniza un singur nivel cluster-ului de containere. Aceste tehnologii sunt utilizate ca și platforme de infrastructură agilă și furnizează un mediu de rulare în portabil pentru aplicațiile Cloud Native.

La nivelul de **PaaS**, aplicațiile bazate pe microservicii sunt folosite. Microserviciile sunt implementate și operate de noduri singulare. Cu toate acestea, toată gestionarea containerelor este complexă și necesită ca aceste noduri singulare să fie integrate într-un cluster cu scopul de a furniza o platformă de servicii, PaaS.

La nivelul straturilor **SaaS**, aplicațiile Cloud sunt vizibile clientului final. Aplicațiile Cloud sunt compuse din servicii mai mici, operate în cluster-e, formate din noduri de calcul și sisteme stocare unice. Modelul de furnizare Cloud la nivel de aplicație, SaaS, face parte din această categorie.

Nivelul 5 este optimizat pentru colaborarea și comunicarea sistemelor prin intermediul API cu scopul de a livra aplicația utilizatorului final.

- **Nivelul 5, Serviciul Software**

Stratul de serviciu software constă în punerea la dispoziție a funcționalităților aplicațiilor pentru a livra capacități funcționale ca și serviciu. Accesul la acest nivel se face prin intermediul API. Serviciile de la acest nivel pot fi servicii analitice, de învățare automată, etc.

Următoarea funcție facilitează punerea la dispoziție a serviciului software prin simplificarea aplicației și a comunicării. Se pune accent pe furnizarea de servicii Cloud cu modele de interacțiune scalabile. Există, de asemenea, și servicii de stocare de date, accesate prin intermediul API bazat pe modele arhitecturale de tip REST.

- **Nivelul 6, Aplicația**

Nivelul aplicației este centrat pe serviciul și funcționalitățile livrate către utilizatorul final prin intermediul interacțiunii cu aplicația proiectată. Acest strat furnizează interfața grafică programatorilor sau specialiștilor IT.

Modelul arhitectural de referință Cloud arătat poate fi văzut ca un ghid în vederea implementării de tehnologie și în vederea standardizării aplicațiilor pentru dezvoltarea de servicii native Cloud.

4.3

Aplicații tradiționale în domeniul IT

Centrele de date IT ale organizațiilor constau în mare parte din aplicații tradiționale, unele dintre ele folosind tehnologii învechite, cunoscute sub denumirea de aplicații *legacy*. Aceste aplicații au fost proiectate cu mult timp în urmă și nu se concentrează pe crearea de experiențe digitale utilizatorului.

În ingineria software, o aplicație monolit descrie un program software dezvoltat pe un singur nivel în care interfața cu utilizatorul și codul de acces la date sunt combinate într-un singur program software. O aplicație monolit este autonomă și independentă de alte aplicații. Ea poate conține câteva sute de funcționalități adunate într-o singură aplicație.

Aplicațiile de tip monolit sunt greu de întreținut, de modernizat și uneori chiar și de construit. Întrucât aplicațiile tradiționale folosesc tehnologii vechi, sunt necesare măsuri de siguranță sporite pentru a asigura funcționalitatea lor în timp. Marconi [Marconi et al., 2016] menționează în lucrarea sa că procesul de dezvoltare al unei aplicații de tip monolit este de regulă un proces îndelungat, greoi și poate introduce erori la momentul dezvoltării acestei.

Aplicațiile native de tip Cloud reprezintă soluția ideală pentru aceste cazuri, întrucât ele pot asigura disponibilitatea serviciilor în regim 24 x 7, permite schimbări de implementare la nivel de serviciu fără a necesita întreruperea întregului proces de business și sunt extrem de scalabile.

4.4

Strategia de migrare în Cloud

Calea către aplicațiile Cloud necesită ca echipele de dezvoltare și operațiuni IT să evolueze într-o direcție comună pentru a construi și pentru a implementa aplicații mai repede și mai eficiente. Indiferent de industria în care o organizație activează, de dimensiunea acesteia, compania trebuie să ia în considerare o gama largă de activități, de tehnologii, și de procese pentru a se putea transforma.

Din cauza complexității de a gestiona sisteme distribuite, aplicații învechite foarte personalizate și care procesează un volum mare de date, într-o eră marcată de inovație și rapiditate, schimbarea către Cloud poate fi dificilă pentru unele organizații.

Sunt necesare modele de maturitate pentru evaluarea organizației în vederea realizării trecerii la tehnologie modernă și pentru a evalua progresul realizat la trecerea către Cloud.

Adoptarea cu succes a unui model de tip Cloud depinde de etape de planificare în proiectele aflate în curs.

Forrester definește în manualul „Cloud Computing Playbook” pașii de urmat în vederea tranziției către Cloud și pune la dispoziție un plan alcătuit din patru faze.

Faza 1. Definirea potențialului Cloud

Primul pas al drumului spre Cloud este înțelegerea modurilor în care Cloud poate să ajute organizația de a deveni mai flexibilă față de cerințele clientului, să acționeze într-un mod intuitiv, să fie mai rapidă și să rămână conectată la necesitățile business.

Faza 2. Planul de migrare Cloud

Această fază acoperă analiza stadiului actual al mediului IT împreună cu definirea unor pași de migrarea a sistemelor IT în cloud.

Faza 3. Optimizarea proceselor

Lumea Cloud necesită noi aptitudini, un nou mecanism de livrare a serviciilor Cloud către utilizatorii finali. Optimizarea proceselor implică automatizarea și managementul aplicațiilor software. Acțiunile identificate pentru a fi schimbate pe parcursul analizei vor fi implementate în cadrul acestei etape.

Faza 4. Procesul de optimizare Cloud

Ultima fază a transformării face referire la optimizarea și îmbunătățirea continuă a practicilor utilizate în timpul fazei de proiect. În acest scop este necesară implementarea unui ciclu continuu de măsurare, de analiză și de ajustare.

La adoptarea unei strategii de implementare Cloud Computing organizațiile nu ar trebui să se concentreze numai pe o nouă metodă de dezvoltare software, ci trebuie să evalueze tehnologia utilizată în cadrul organizației, procesele afectate, nivelul de pregătire al personalului.

4.5 Analiza de maturitate Cloud pentru aplicații tradiționale

O aplicație Cloud este o aplicație construită pentru a beneficia de caracteristicile Cloud Computing cu scopul de a pune la dispoziție servicii IT în timp scurt, oferă flexibilitate și calitate a serviciilor livrate.

Se ia în considerare o analiză completă a ciclului de viață al aplicației. În acest stadiu, toate criteriile și subcriteriile fiecărei componente trebuie identificate în mod iterativ pentru a construi structura ierarhică.

Unele studii iau în considerare compatibilitatea aplicațiilor pentru Cloud numai din punct de vedere tehnic [Frey, 2011]. Celelalte studii iau în considerare migrarea spre Cloud doar din punct de vedere financiar [Misra, 2011].

Pentru a stabili strategia necesară adopției Cloud, factorii de decizie au nevoie de un studiu complet al tuturor domeniilor cheie și posibile dezavantaje sau riscuri. De aceea, analiza are nevoie să se integreze și să furnizeze perspective de utilizare ale serviciilor Cloud, procese de business, ciclului de viață al produsului software, *tool*-urile și tehnologiile folosite.

În continuare, criteriile de analiză trebuie definite. Metoda de analiză Cloud utilizează patru arii de evaluare pentru a măsura și descrie compatibilitatea Cloud pentru o aplicație software. Aceste domenii de evaluare sunt considerate a fi niște clase pentru ordonarea aplicațiilor software în baza compatibilității acestora cu migrarea în Cloud. Clasificarea aplicației necesită compararea valorii cumulate a unei aplicații cu niște valori de referință, care ajută la stabilirea categoriei din care aplicația face parte.

Profilul de referință pentru fiecare categorie e determinat de factorul de decizie al organizației în baza nevoilor de business ale respectivei organizații. Majoritatea abordărilor în vederea deciziei de adopție Cloud Computing sunt bazate pe o abordare calitativă. Chiar și abordările cantitative existente evaluează strategia de migrare într-o astfel de manieră încât riscul sau costul unui criteriu este compensat de beneficiul adus de un alt criteriu. Există cazuri în care deficiența unui criteriu nu poate fi compensată beneficiul a unuia sau a mai multor criterii. O aplicație poate fi evaluată ca având un scor de compatibilitate ridicat prin funcționalitatea existentă a aplicației sau prin valoarea adăugată adusă proceselor de business și cu toate acestea poate să nu fie un bun candidat pentru migrarea Cloud. Acesta poate fi stabilită în funcție de gradul de expunere al aplicației la riscuri cunoscute pe care organizația nu este dispusă să-l accepte.

Luând în considerare aceste aspecte, criteriile folosite pentru a analiza compatibilitatea unui sistem de aplicații software pentru migrarea în Cloud pot fi clasificate, ca și criterii de referință sau de evaluare. Criteriile de referință fac parte dintr-o categorie de criterii ale căror limitări nu ar trebui contrabalansate de impactul altor criterii. Într-un astfel de caz, pentru ca o aplicație să fie potrivită pentru migrare, o limită minimă de criterii trebuie menținut sau nu trebuie să depășească limita maximă a criteriilor de cost. Criteriile de evaluare fac parte dintr-o categorie de caracteristici utilizate pentru a analiza compatibilitatea Cloud a

unui sistem software, a cărui valoare poate fi contrabalansată de impactul avut de către celelalte criterii.

De vreme ce, fiecare dimensiune și criteriile respective au un efect decisiv mutarea unui sistem software existent în Cloud, revizuirea importanței sau greutateii de a muta aplicația în Cloud poate fi diferită. De aceea, o astfel de greutate trebuie stabilită în baza nevoilor de business ale organizației. Există mai multe metode pentru a evalua și pentru a valida maturitatea aplicației. Procesul ierarhic analitic este una dintre cele mai folosite metode de a evalua și de a valida fiecare caracteristică prin comparația a două câte două caracteristici în cadrul metodei de evaluare cu mai multe criterii.

E posibil ca unele criterii să nu aibă standarde de măsurare pentru securitatea instanțelor aplicațiilor. Pentru genul acesta de criterii este necesară stabilirea unei scale de măsurare pentru evaluarea criteriilor.

Utilizând metoda de evaluare a maturității aplicației în vederea adoptării Cloud, se analizează ciclul de viață al aplicației software, care utilizează setul descris de criterii și caracteristici. Acestea permit clasificarea aplicației folosind diverse discipline, cum ar fi statistica, inteligența artificială sau obiective de tip operațional. Evaluarea propusă ia în considerare beneficiile operaționale și de business, ca și criteriile fundamentale.

Modelul propus va atribui o valoare aplicației la nivelul fiecărui criteriu de validare și apoi se va înmulți cu un factor de greutate prestabilit pentru a clasifica aplicația, prin utilizarea metodei de clasificare în baza mai multor criterii. Aceasta va indica dacă aplicația trebuie rescrisă pentru migrare, dacă rămâne în starea curentă sau dacă poate fi transferată în Cloud.

4.6 Procesul de migrare în Cloud Computing pentru aplicațiile tradiționale

Cadrul descris oferă un punct de plecare pentru analiza aplicațiilor în vederea migrării Cloud.

În teza de față, soluția propusă pentru evaluarea Cloud este automată și poate fi realizată cu ușurință de către specialiștii IT. Implementarea poate furniza rezultate imediate în vederea trecerii la etapa de implementare.

- **Mutarea în Cloud**

Cel mai simplu mod de a muta o aplicație existentă în Cloud, care să satisfacă condițiile pentru arhitectura aplicației Cloud, este de a o dezvolta în conformitate cu tehnologiile și principiile Cloud. Acest lucru se întâmplă de obicei în cazul aplicațiilor nou dezvoltate, care au fost proiectate folosind arhitecturi moderne și tehnologii specifice Cloud.

- **Relocarea**

Relocarea presupune execuția aplicațiilor pe infrastructura furnizorului de Cloud. Această abordare necesită optimizarea unor elemente de bază ale aplicației pentru a funcționa cu succes în Cloud.

- **Modernizarea aplicațiilor pentru adoptarea Cloud**

Multe aplicații existente sunt critice pentru îndeplinirea obiectivelor de business și sunt critice pentru organizații. Din aceste motive ele nu pot fi retrase din regimul de funcționare curent. Acest tip de aplicații trebuie integrate în noile norme de proiectare specifice Cloud. Această soluție presupune reconstruirea sau implementarea anumitor funcționalități ale aplicației, utilizând tehnologii Cloud la nivel PaaS. Soluția este determinată de necesitatea business de a adăuga noi caracteristici, de a scala sau de a crește performanța aplicației.

- Dezactivarea din mediul de producție

În unele cazuri, rezultatul analizei indică faptul că aplicațiile sunt învechite sau depășite din punct de vedere tehnologic de la momentul la care au fost proiectate sau dacă obiectivul pentru care au fost proiectate nu mai este necesar organizației. Acest tip de aplicații reprezintă candidați pentru retragerea din regimul de funcționare și permit înlocuirea acestora cu aplicații noi, care sunt proiectate ca fiind aplicații moderne de tip Cloud.

În plus, un alt beneficiu este faptul că retragerea aplicațiilor depășite, permite resurselor IT să fie reutilizate și să reducă riscurile de securitate cauzate de către aplicațiile vechi.

4.7 Studiu de caz

Există posibilitatea ca prin intermediul adopției Cloud modele existente de business și aplicații noi să poată fi create. Indiferent de intenția sau de necesitatea de business, toate organizațiile aflate în procesul de adopție Cloud au obiective legate de o flexibilitate, de a răspunde mai repede schimbărilor și cerințelor noi de business cu scopul de a se transforma într-o organizație digitală.

S-a dorit modernizarea mediului IT tradițional spre servicii la cerere, disponibile oricând, adaptate pe nevoile specifice ale clienților. Mișcarea spre Cloud a fost văzută ca un mijloc de inovare în cadrul organizației cu scopul de a ține pasul cu noile tehnologii moderne de pe piață.

Fig 4.8.1 prezintă abordarea proiectului pentru tranziția în Cloud a mediului IT al organizației. Această metodă se bazează pe evaluarea aplicațiilor în patru pași.

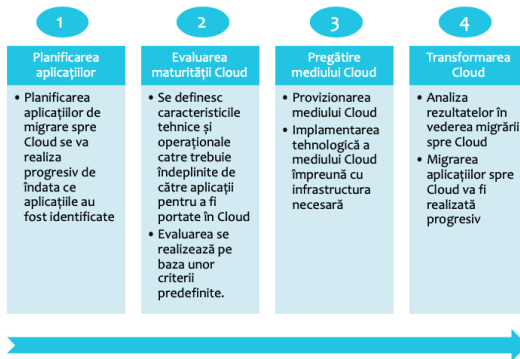


Fig. 4.7.1 Prezentarea strategică a transformării Cloud

Pentru a obține metricile pe baza cărora să se va lua decizia de migrare, trebuie realizată o analiză calitativă și cantitativă minuțioasă. Dacă rezultatele satisfac așteptările și nevoile companiei, migrarea va avea loc chiar și în cazul în care compania va trebui să facă față altor provocări derivate din arhitectura și constrângerile generate de Cloud Computing. De aceea, una dintre primele provocări pentru echipa implicată în proiect a fost să definească și să propună o abordare de analiză a modernizării aplicațiilor. Studiul trebuie să ofere vizibilitate asupra caracteristicilor aplicației existente din două perspective: tehnică și de business, urmată apoi de o analiză de fezabilitate tehnică și o analiză de fezabilitate financiară și de business.

Înainte de a trece la migrarea efectivă spre Cloud, trebuie analizate cu atenție performanța sistemului, stabilitatea și fiabilitatea aplicației, iar echipa de migrare trebuie să corecteze toate problemele identificate.

Majoritatea aplicațiilor care deserveau departamentul de business rula pe o infrastructură și platformă învechită din punct de vedere tehnologic și ar fi putut fi migrate pe servere virtuale. Datele și procesele care erau folosite în organizație erau de impact major pentru derularea activităților în organizație, așadar modelul propus trebuia să evite potențiale riscuri de nefuncționalitate ale sistemelor.

Deoarece mediul IT este proiectat folosind diferite tehnologii și are diverse implementări arhitecturale și mai puțină standardizare, metoda și abordarea migrării în Cloud a fost declarată o provocare cu impact major.

Proiectul a avut ca scop investigarea aplicațiilor și a soluțiilor tehnice existente în vederea mutării aplicațiilor existente într-un Cloud de tip privat. Aceste soluții vor fi proiectate în așa fel încât să dezactiveze orice dependență de furnizorul de Cloud.

De aceea, abordarea modernizării trebuie să acopere:

- strângerea de informații din două perspective: aspecte tehnice, cum ar fi limbajul de programare în care aplicația a fost scrisă, arhitectură, baza de date, etc., dar și aspecte de business cum ar fi procesul de business susținut, SLA-uri, costuri de licențiere, strategii de livrare, etc., ale aplicației;
- Identificarea unui set de metrici și indicatori în vederea caracterizării aplicației existente și a companiei din perspectiva ambelor dimensiuni: tehnică și de business;
- Alinierea la strategia de migrare în Cloud;
- Furnizarea unor modalități de optimizare în conformitate cu metricile definite;

Pentru implementarea proiectului, prima fază a presupus înțelegerea obiectivelor strategice de business, urmat de alcătuirea echipei de proiect, a planului de proiect și de comunicare. S-a realizat analiza aplicațiilor și a infrastructurii, s-au identificat aplicațiile pentru analiză și s-a stabilit arhitectura software necesară pentru aplicațiile determinate.

Pentru o standardizare a cadrului de evaluare, a fost dezvoltată o aplicație care să identifice informații relevante specifice procesului de business.

Instrumentul de evaluare pus la dispoziție a ajutat părțile implicate să capete încredere în proiect și au fost capabili să decidă mult mai repede care aplicații să fie primele cu care să se înceapă migrarea aplicațiilor în Cloud. În vederea identificării aplicațiilor propuse spre evaluare s-a utilizat un arbore decizional, care să faciliteze evaluarea cu privire la tipul aplicațiilor, fig. 4.7.2..

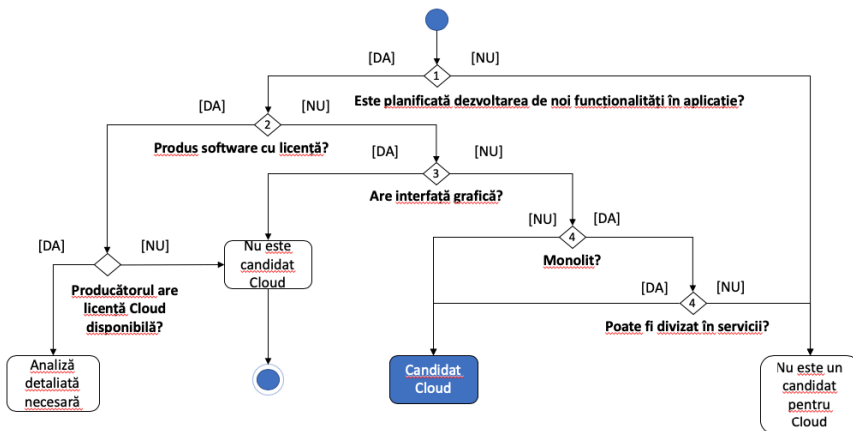


Fig. 4.7.2 Selecția aplicației pentru analiza Cloud

În a doua fază de proiect, a fost utilizată analiza propusă de evaluare a maturității aplicației. Aceasta a ajutat la examinarea infrastructurii existente, a arhitecturii și tehnologiilor utilizate de aplicații. În acest stadiu s-au analizat dependențele aplicațiilor, datele și informațiile relevante, procesele și ciclurile de viață ale aplicațiilor. Apoi s-a definit infrastructura propusă pentru implementarea Cloud, s-au agreat arhitectura software și tehnologiile folosite în migrare.

Analiza Cloud a permis determinarea fezabilității și a potențialelor riscuri. Din acest stadiu pot fi furnizate un plan și estimări generale pentru implementarea proiectului. În vederea analizei de maturitate a aplicațiilor software, s-a dezvoltat un utilitar care evaluează adaptabilitatea aplicației la mediul Cloud.

Pentru analiza gradului de maturitate al aplicațiilor se va verifica opt domenii pentru migrarea spre mediile Cloud. Aceste domenii fac referire la aspectele tehnologice, la procesele business și la tipul de date procesate de către sistemele evaluate.

Rezultatele agregate din fig. 4.7.3 indică rezultatul evaluării aplicațiilor cu privire la maturitatea lor de a fi mutate în Cloud sunt calculate automat în baza referințelor standardizate. Aceasta permite optimizarea sistemelor în mod sistematic folosind automatizare.



Aggregated Cloud maturity level results for Application1

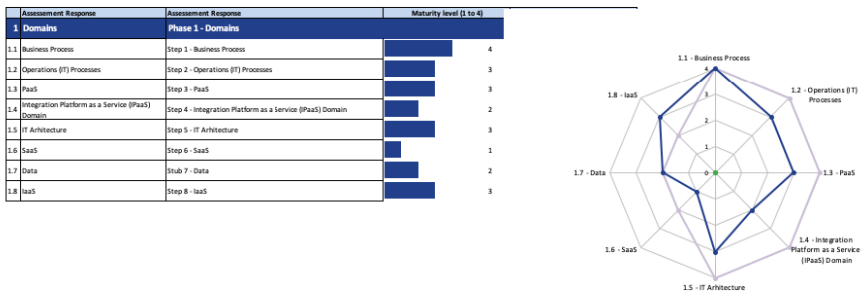


Fig. 4.7.3 Rezultatele agregate ale evaluării aplicațiilor

Rezultatele obținute ca urmare a analizei de fezabilitate tehnică și a celei de business vor ajuta echipele de management în verificarea progresului realizat pentru adoptarea strategică Cloud.

Analiza de maturitate Cloud dezvoltată pentru acest proiect a permis o verificare atentă a tuturor pașilor: evaluare tehnică și de business, condițiile esențiale pentru consolidarea de informații în vederea construirii analizei de fezabilitate tehnică și de organizație.

Analiza Cloud implementată permite observarea a două perspective, analiza situației curente (inițială) și a celei dorite (finală) a unei aplicații, ce urmează să fie migrată în Cloud.

Tool-ul implementat se axează pe caracterizarea metricilor și indicatorilor, metrii calculați și combinați în evaluările dimensiunilor abordate. Pentru realizarea analizei sunt necesare informații legate nu numai de codul sursă și arhitectură, de procesul de dezvoltare, sau de arhitectura dorită, ci și de informații referitoare la infrastructură, la cerințele non-funcționale, etc.

Aplicațiile Cloud sunt proiectate pentru a putea fi mutate în diferite medii de execuție distribuite. Livrarea continuă și DevOps sunt metode utilizate pentru a automatiza procesul de programare, de testare și de integrare ale serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor.

Rezultatul analizei de afinitate pentru mediul Cloud este realizată în cazul primului set aplicații evaluate și prezentate în fig. 4.7.4.

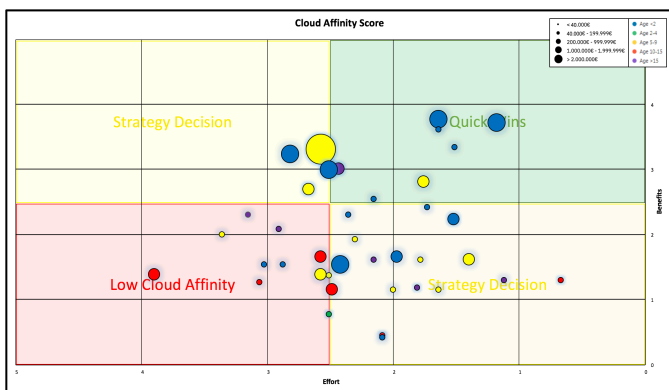


Fig. 4.7.4 Rezultatele evaluării de migrare ale aplicațiilor în Cloud

Rezultatul analizei împreună cu modelul arhitectural stabilit pentru aplicațiile Cloud au ajutat organizația să construiască și să modeleze noua infrastructură la nivel de platformă Cloud Computing.

Beneficiile soluției produsului software implementat pentru evaluarea maturității mediului IT furnizează numeroase avantaje:

- Oferă informații relevante și de calitate managerilor pentru adoptarea strategică Cloud;
- Oferă un sistem consolidat în vederea evaluării compatibilității aplicației pentru adoptarea Cloud în cadrul organizației;
- Oferă o privire de ansamblu cu privire la ciclul de viață al aplicației;
- Oferă un bun indicator (nivelul de maturitate sau capacitatea a serviciului) în privința calității unui serviciu Cloud solicitat;
- Reduce eforturile manuale pentru evaluarea maturității aplicațiilor în vederea migrării în Cloud;
- Face posibilă o evaluare standardizată pe baza aceluiași criterii pentru întreaga organizație începând cu tehnologia, procesele de business, securitatea, etc.;
- Oferă specialiștilor IT o vedere de ansamblu bună asupra produselor lor, permițându-le să îmbunătățească serviciile și calitatea produselor Cloud;

4.8 Concluzii

În acest capitol a fost descris un model de referință pentru a ajuta la implementarea și dezvoltarea de servicii Cloud Computing. A fost prezentat în studiul de caz proiectarea unui sistem informațional care analizează maturitatea aplicațiilor existente în organizație și prezintă pașii de optimizare a acestora în vederea migrării ei spre Cloud. Sunt evidențiate beneficiile pentru o organizație digitală ce pune la dispoziție servicii moderne de tip Cloud Computing.

Capitolul a prezentat o abordare strategică de modernizare a aplicațiilor pentru a fi aliniată cu tehnologia Cloud Computing. S-a realizat un cadru standardizat de migrare spre Cloud, cu scopul de a consilia specialiștii IT pentru a putea decide dacă o aplicație existentă

poate fi migrată în sistemele Cloud, dacă aceasta trebuie modernizată sau este recomandată retragerea ei și înlocuirea cu un nou serviciu mai modern. Se prezintă informații cu privire la pașii necesari în vederea transformării digitale și se evaluează domeniile afectate de schimbare împreună cu obiectivele ce trebuie atinse. Se oferă informații cu privire la procesele organizaționale, ce trebuie optimizate ca și consecință a migrării.

S-a introdus un mecanism automat de analiză a maturității aplicațiilor pentru a putea stabili compatibilitatea acestora la nivel de tehnologie și de proces în vederea migrării ei spre mediile Cloud Computing. Studiul de caz furnizează informații valoroase cu privire la tehnologia utilizată, arhitectura software, procesele și metodele de adoptare Cloud pentru produsele software tradiționale existente în organizațiile de astăzi. Modelul și sistemul propus va fi optimizat pentru a automatiza procesul de dezvoltare ale serviciilor Cloud Computing și de livrarea de noi funcționalități utilizatorilor săi.

5. Automatizarea procesului de integrare a serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor

În vederea implementării modelului automatizat de integrare a produselor software în centrele de prelucrare a informațiilor de tip Cloud Computing, se va evalua procesul de dezvoltare a aplicațiilor software și se vor arăta fundamentele, procedurile și metodele de dezvoltare software. De asemenea, se vor verifica și se vor valida tehnologiile actuale de pe piața IT, utilitățile necesare proiectării și se va prezenta metoda de implementare automată a serviciilor în Cloud. Soluția implementată va asigura calitatea serviciilor Cloud.

5.1 Managementul ciclului de viață în dezvoltarea serviciilor software

Procesul de dezvoltare a aplicațiilor software și ale sistemelor IT este dictat de diferite metodologii de dezvoltare. O metodologie de dezvoltare software se referă la direcțiile de proiectare utilizate pentru a planifica, pentru a gestiona și pentru a controla procesul de dezvoltare a unui sistem IT. Ciclul de viață în dezvoltarea aplicațiilor software este o procedură, care descrie activități efectuate în fiecare fază a procesului de dezvoltare. Managementul ciclului de viață în dezvoltarea aplicațiilor descrie detaliat modul în care are loc codarea, mentenanța și înlocuirea aplicațiilor software. Acesta este cunoscut și sub denumirea de proces de dezvoltare a serviciilor software.

5.2 Modele de dezvoltare software

Există multe modele de dezvoltare software și multe organizații creează modele noi, pe care ulterior le utilizează. Alegerea unui model de dezvoltare software are un impact major asupra procesului de testare.

- **Modelul Waterfall**

Cunoscut și ca modelul de dezvoltare tradițional, acesta este un model cu ciclu de viață secvențial și linear. Aici toate sarcinile sunt realizate în ordine secvențială și o activitate va începe doar atunci când cea precedentă a fost încheiată și aprobată.

- **Modelul incremental**

Modelul incremental divide produsul în unități de *build*, în care unități de dezvoltare sunt create și testate separat. Această abordare identifică erori în cerințele utilizatorului, deoarece feedback-ul e solicitat la fiecare etapă și codul este testat încă din etapele inițiale de programare.

- **Modelul iterativ**

Modelul iterativ este modelul care vine cu o metodă nouă de dezvoltare a aplicațiilor, Modelul iterativ poate să furnizeze rezultate rapide, necesită un minim de comunicare în avans și oferă multă flexibilitate. În cazul dezvoltării iterative, proiectul este împărțit în etape mai mici. Acest lucru permite echipei de dezvoltare să demonstreze rezultate rapide în cadrul etapele de programare și să obțină feedback valoros din partea utilizatorilor sistemului.

Metodele agile au luat locul metodelor tradiționale pentru a depăși rigiditatea modelului tradițional. Metoda agilă urmează o abordare dinamică a dezvoltării de servicii software. Aceasta este o metodă interactivă, bazată pe o echipă, care lucrează împreună pentru a programa o aplicație software într-o perioadă scurtă de timp.

5.3 Cloud și DevOps

Odată cu utilizarea Cloud Computing ca și serviciu și cu transformarea industriei IT, procesul de dezvoltare software a devenit mai atractiv și a modelat felul în care hardware-ul IT este conceput și este utilizat.

În vreme ce Cloud facilitează accesul la aplicațiile software prin intermediul rețelelor de calculatoare, Agile și DevOps ca și practici, ajută inginerii IT să îmbunătățească și să automatizeze gestionarea ciclului de viață al unui produs software.

Lucrarea propune un mecanism cu o arhitectură orientată proceselor DevOps, care are la bază instrumente agile folosite împreună în implementarea unui sistem automat de livrare a codului software. În plus, modulul de testarea continuă acoperă cele mai bune practici de testare și metode, ce pot fi implementate la scară largă în orice mediu Cloud.

Cloud și DevOps sunt interconectate și aceasta se datorează:

- naturii caracteristice Cloud Computing care prin utilizarea automatizării puse la dispoziție de practicile DevOps prin intermediul unei platforme standardizate și centralizate realizează testarea, lansarea și publicarea aplicațiilor în regim automat. Dat fiind că în trecut sistemele distribuite nu se potriveau bine cu dezvoltarea de software centralizată, prin utilizarea unei platforme Cloud, multe dintre problemele legate de complexitatea distribuției sistemelor sunt soluționate;
- Automatizarea DevOps impulsionează dezvoltarea de software în medii Cloud. Majoritatea furnizorilor publici și privați de Cloud Computing susțin DevOps din punct de vedere sistematic pe platforma lor, incluzând module de integrare continuă și instrumente de dezvoltare continuă. Această integrare strânsă scade costurile legate de tehnologia de automatizare DevOps în centrul de date al organizației și furnizează conducere și control organizat pentru procesul DevOps;
- DevOps și Cloud micșorează nevoia de a justifica resursele folosite. Cloud facilitează utilizarea resurselor în funcție de cantitatea utilizată și reușește să aloce cantitatea de resurse neconsumate către aplicație, dezvoltator, utilizator sau consumatorul de date, etc. Sistemele tradiționale nu furnizează acest serviciu în mod obișnuit. Când se facilitează folosirea resurselor aflate în Cloud, este mult mai simplă urmărirea costurilor legate de resursele de dezvoltare și de operare.

Acest capitol se axează pe crearea unei platforme automate de livrare a aplicațiilor software în mediile Cloud utilizând componente de integrare continuă și publicare continuă, în timp ce asigură calitatea serviciului software utilizând module de testare automate.

Sistemele de tip Cloud Computing și mai ales, la nivel de infrastructură din Cloud Computing oferă un mediu de programare versatil și flexibil și furnizează putere de calcul, de stocare și de rețea. Acestea fac posibilă automatizarea completă a furnizării de servicii software într-un mediu Cloud.

Metodologia DevOps propune un model de colaborare bazat pe comunicare și pe colaborarea dintre dezvoltatori și specialiștii IT ale echipelor de operațional, cu scopul de a îmbunătăți calitatea și viteza de publicare în sistemele de producție.

Automatizarea este utilizată pentru a mări ciclurile de publicare, îmbunătățind timpii de lansare pe piață, reducând rata de erori produse în dezvoltarea produsului software și totodată încurajează, prin intermediul ciclului de feedback continuu, adoptarea de noi funcționalități în cel mai scurt timp posibil

5.4 Procesele DevOps

Procesele DevOps (Fig. 5.7.1) definite vor reprezenta punctul de pornire în stabilirea procesului de implementare pentru soluția automată. O înțelegere sumară a proceselor sale principale trebuie stabilită în scopul unei implementări cu succes a unui sistem automat de migrare a serviciilor Cloud.

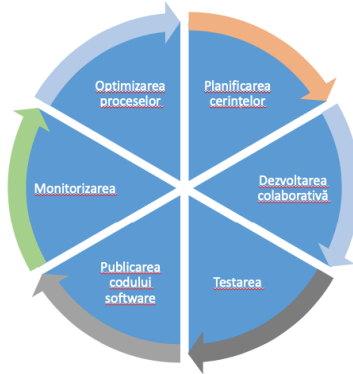


Fig. 5.7.1 Procesele DevOps

În vederea definirii unei arhitecturi stabile pentru implementarea automatizată a serviciilor în Cloud, se va pleca de la procesele DevOps prezentate, urmând a se identifica tehnologiile și utilitățile optime în crearea ansamblului de migrare automată a aplicațiilor în Cloud.

Se vor analiza tehnologiile actuale și se vor identifica posibilitățile de implementare.

5.5 Modelarea procesului de dezvoltare software

Deoarece din ce în ce mai multe companii se bazează pe software pentru dezvoltarea și furnizarea unor produse și servicii adecvate, procesele de dezvoltare software devin parte integrată a proceselor organizațiilor. La fel ca și în cazul proceselor de business, procesele de dezvoltare software pot varia semnificativ de la o companie la alta datorită caracteristicilor unice ale acesteia.

Aceste procese pot fi reconfigurate în mai multe moduri pentru a ține cont de schimbările și de particularitățile sau de cerințele specifice ale organizației.

Modelarea proceselor de business și metodele folosite în procesele de dezvoltare indică direcția de proiectare a sistemelor software luând în calcul perspectivele multiple ale organizației și ajută la alegerea tehnologiei necesare. Configurația procesului software este prezentată cu scopul implementării conceptelor DevOps. Prin modelarea arhitecturii proceselor de business se descrie și se analizează configurațiile arhitecturii proceselor luând în calcul toate scenariile de proiectare posibile.

În decizia configurației procesului DevOps se iau în considerare obiectivele organizației și beneficiile finale acesteia. Configurația de design a proceselor este o decizie, care se ia în funcție de cerințele funcționale sau nefuncționale din departamentul business și ajută echipele de programatori în uniformizarea procesului de dezvoltare și livrare software și îmbunătățește procesul de dezvoltare.

Proiectarea modelului de business a fost conceput în baza scenariului de utilizare prezentat. Obiectivul final este de a defini metodele în care etapele ciclului de dezvoltare DevOps trebuie implementat sau adaptat încât sa deservească unui număr mare de programatori. Procesul nou dezvoltat trebuie să susțină cerințele de business existente în cadrul unei organizații sau să susțină o gamă largă de procese de business pentru diferite organizații, care activează în domenii diferite.

Procesul de business modelat pentru dezvoltarea softwaresusține abordarea de dezvoltare agilă, utilizează automatizarea și are capacitatea de se adapta frecvent la schimbare. Procesul de dezvoltare poate fi ajustat după necesități în funcție de specificul aplicațiilor, de dezvoltarea produsului sau de procesele operaționale, toate în funcție de cerințele existente.

5.6 Analiza DevOps – studiu de caz

În prezent, majoritatea companiilor IT discută despre transformarea digitală și adoptarea metodologiilor noi de dezvoltare software, în vreme ce se confruntă cu probleme de integrare și buna funcționare a sistemelor existente pe piață pentru implementarea și testarea aplicațiilor software. Problematika este dată de utilizarea diverselor *tool-uri* pentru realizarea integrării codului sursă, configurarea mediilor de testare și validare, dar și asigurarea unei bune funcționalități a întregului sistem, de stabilitatea și de performanța sistemelor actuale.

Studiul de caz a fost realizat în vederea implementării serviciilor Cloud în organizație și a fost realizat pe parcursul a 4 luni de zile. Acesta a plecat de la necesitatea organizației de a rămâne competitivă în industria în care activează, dar și datorită motivației de a integra cerințele venite din departamentele business mult mai repede în procesele de dezvoltare, de a utiliza produse software de actualitate din punct de vedere tehnologic, stabile și performante.

În acest sens organizația s-a orientat în abordarea ei strategică de a deveni mai aproape de clientul final, de a fi mult mai flexibilă și agilă în răspunsul la adresa cererilor din departamentele business și de a livra clientului final produse de calitate superioară.

Strategia de migrare spre Cloud utilizând DevOps cuprinde comunicarea viziunii de transformare în organizație, pregătirea proceselor, a tehnologiilor și efectuarea modificărilor culturale organizatorice în vederea adoptării DevOps ca model funcțional prin implementarea de proiecte mai mici la început și ulterior extinderea către întreaga organizație.

Cele patru faze de implementare sunt definite în cadrul hărții de transformare DevOps [Gușeală et al., 2019], așa cum este reprezentată în fig. 5.6.1.

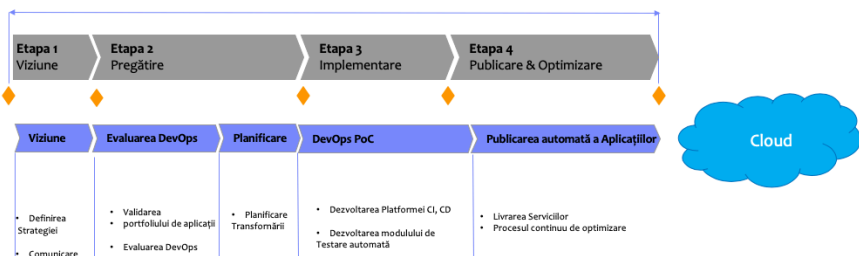


Fig. 5.6.1 Planul strategic de transformare DevOps

După ce strategia de business este agreată și stabilită, comunicată și clarificată cu toate părțile afectate, pasul de pregătire include utilizarea unui cadru pentru a evalua aplicațiile din portofoliu în vederea adoptării DevOps.

Un model continuu de maturitate a livrării permite adoptarea de practici de livrare continue într-o manieră pragmatică și graduală. Această abordare este creată pentru a îmbunătăți calitatea serviciilor și pentru a spori timpul de lansare pe piață.

Abordarea nu afectează numai procesul de dezvoltare software, dar poate de asemenea avea ca rezultat recomandări pentru îmbunătățiri aduse arhitecturii software sau schimbări în aspectul organizațional al procesului de dezvoltare software.

Cu această abordare scopul este de a reduce timpul de livrare a produselor pe piață și eliminarea riscurilor din procesul de publicare.

Tool-ul de evaluare a aplicațiilor software trebuie să permită analiza stadiului actual al sistemului IT, dar să și prezinte elemente de optimizat, domenii propuse spre schimbare pentru a atinge maturitatea și agilitatea sistemelor dorite [Gușeală et al., 2019].

În vederea obținerii și consolidării informațiilor business și tehnice la nivel de portofoliu de aplicații și apoi la nivel granular de aplicație, interviurile tehnice au verificat întregul ciclu de viață al aplicațiilor.

În timpul unei astfel de pregătiri, întregul ciclu de viață al unei aplicații este evaluat din faza de cerință de business și până la momentul dezvoltării funcționalității, publicarea productivă și până la obținerea feedback-ului după livrarea în mediul Cloud.

În studiul de caz s-a evaluat un sistem software compus din peste 50 de aplicații IT destinate procesului de proiectare din inginerie. Complexitatea acestei evaluări și soluții de optimizare a fost dată de faptul ca aplicațiile în cauză foloseau tehnologii și limbaje de programare diferite. Așadar utilitarul creat a fost soluționat pe parcursul proiectului colectând toate informațiile despre aplicații fără să depindă de tehnologia, interfețele sau limbajul de programare în care aplicația evaluată a fost dezvoltată.

Aplicațiile analizate se bazează pe limbaje de programare diferite, plecând de la aplicații dezvoltate în Java C++, Oracle SQL, Linux, .NET, etc. Așadar complexitatea de soluționare a fost de a identifica un mecanism optim de evaluare a aplicațiilor independent de limbajul de programare folosit în dezvoltarea a aplicațiilor.

Tool-ul creat a permis consolidarea informațiilor tehnice și de business fiecărei aplicații plecând de la infrastructura actuală, platforma *middleware*, procesul de dezvoltare software, verificarea procesului de test și au oferit o vedere detaliată a componentelor.

Utilitarul de evaluare a maturității aplicațiilor vine în sprijinul transformării și migrării aplicațiilor în mediile Cloud, prezentând ariile de optimizare în procesul de dezvoltare a aplicațiilor software, validează la nivel de proces DevOps aplicațiile și verifică componentele tehnice ale acesteia.

În a doua etapă se trece la evaluarea aplicației de către specialiștii IT prin răspunsurile oferite de către aceștia. Aceste răspunsuri sunt folosite în metoda de calcul prezentată în utilitar. Răspunsul este cântărit și evaluat automat în funcție de importanța definită de către managementul organizației pentru procesul evaluat. Se recomandă atingerea nivelului superior pentru a beneficia de avantajele procesului de transformare în organizație.

Sistemul de evaluare este aliniat proceselor DevOps definite anterior și integrate în *tool*-ul de stabilire a maturității DevOps pentru aplicațiile software.

Rezultatele obținute prin utilizarea *tool*-ului propus, indică poziționarea aplicației evaluate pe scara de maturitate DevOps și oferă specialiștilor IT o direcție de optimizare a serviciilor.

Raportarea portofoliului de aplicații la metricile de maturitate DevOps oferă transparență managerilor de departamente prin posibilitatea raportării stării actuale pentru aplicațiile de

care răspund, și permite stabilirea mijloacelor de optimizare în vederea împlinirii direcției strategice de abordare a strategiei DevOps pentru organizație.

Rezultatul analizei de adopție DevOps în aplicațiile IT indică evaluarea maturității unei aplicații, în care clasificarea se realizează cu o notare de la 1 la 4, unde 4 este nota cea mai mare ce indică confirmarea maturității, fig. 5.6.2

Utilitarul va calcula și va indica nivelul de maturitate actual al aplicației, va arăta care domenii trebuie luate în calcul pentru îmbunătățire, va oferi indicații detaliate specifice domeniului identificat și acțiunile necesare atingerii unui prag optim pentru transformare.

Results for Application1				
Maturity model for Phase 1 - Prepare				
Step	Response (0-4)	Maturity level (0-15)	Weighted score (0-15)	Evidence supplied
Step 1: Continuous Business Planning				
1.1.01	How the requirements are captured	2	2	4. Requirements tools are integrated with CSM/ALM 2. Requirements are captured using tools like JIRA 2. Requirements are in tools like Rational, Service Now 2
1.1.02	What is the granularity of the business reqs ?	4	4	1. Requirements are captured in a structured manner 2. Requirements are captured in a structured manner 3. Requirements are captured in a structured manner 4. Requirements are captured in a structured manner
1.1.03	How the requirements are prioritized	4	4	1. Prioritization is done in a structured manner 2. Prioritization is done in a structured manner 3. Prioritization is done in a structured manner 4. Prioritization is done in a structured manner
1.1.04	How are the project activities planned	4	4	1. Project planning is done in a structured manner 2. Project planning is done in a structured manner 3. Project planning is done in a structured manner 4. Project planning is done in a structured manner
1.1.05	Are there any metrics defined?	4	4	1. Metrics are well defined and used consistently 2. Metrics are well defined and used consistently 3. Metrics are well defined and used consistently 4. Metrics are well defined and used consistently
1.1.06	How are requirements tracked to the production release?	4	4	1. Requirements are tracked to the production release 2. Requirements are tracked to the production release 3. Requirements are tracked to the production release 4. Requirements are tracked to the production release
1.1.07	Do you have integrated view of the code promotion process from Dev to QA to Ops?	4	4	1. Integrated view of the code promotion process is available 2. Integrated view of the code promotion process is available 3. Integrated view of the code promotion process is available 4. Integrated view of the code promotion process is available
1.1.08	How do you capture the decisions from the planned release calendar?	4	4	1. Decisions are captured in a structured manner 2. Decisions are captured in a structured manner 3. Decisions are captured in a structured manner 4. Decisions are captured in a structured manner
Step 2: Collaborative Development				
1.2.01	Can you see the configuration items (CI) under version control. Use the criteria to identify it as a configuration item (CI)	3	3	4. All artifacts in the project are under version control 3. Requirements are not versioned but other artifacts are versioned 2. Build scripts, deployment scripts, database scripts are not under version control 1. No clear identification of CI in version system is in place (CI metrics not defined for CI in version system control is used)
1.2.02	Is the software configuration management system integrated with Development, Test and Deployment?	3	3	4. IDE, Test Stages, Production, Test Management System are integrated with SCM system 3. Test Management is not integrated with SCM system 2. Production and Test stages are not integrated with SCM system 1. SCM Dev IDE is not integrated with SCM system (No integration of any system with SCM system)
1.2.03	How are the baselines defined? What is the baseline strategy? What is the brand/owner/branch/namespace?	2	2	4. Definition in place for "When and how baseline is created". Baseline is created in all stages in a continuous manner for all types of builds. 3. Baselines are created during major releases and only when testing is successful for each of the intermediate stages. 2. Baselines are created only during major releases and only for production system. 1. Define SCM tool baseline is followed (Baselines strategy is not defined. Baseline is not practices)
1.2.04	How do you manage remotely located teams for effective delivery of the product? Explain the collaboration tools implemented if any.	1	1	4. Teams are distributed geographically and SCM system has capabilities to replicate and merge the code frequently 3. Merging is automatic but every morning rather than frequent 2. Merging is automatic but person dependent. Needs to review the code manually and merge it 1. Merging and code replication is dependent on person's availability. It is done manually (Merging and code replication is dependent on person's availability. It is done manually)

Fig. 5.6.2 Rezultatele evaluării de maturitate

Valorile rezultate în urma efectuării calculului folosind utilitarul descris sunt agregate. În cazul aplicației prezentate în fig. 5.6.3 trebuie să își optimizeze procedurile de dezvoltare software în mediul distribuit, să optimizeze procesele de testare automată și să investească semnificativ în mecanismele de monitorizare.

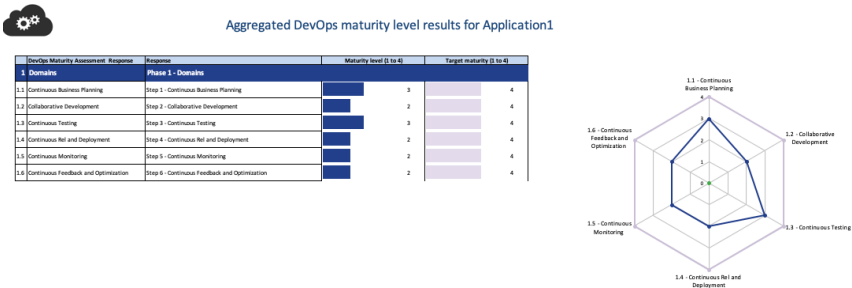


Fig. 5.6.3 Rezultatul analizei de maturitate DevOps

Analiza de maturitate stabilește gradul de adaptabilitate a conceptelor DevOps la fiecare etapă a procesului de dezvoltare și evaluează procesul curent intern al organizației,

tehnologia folosită, mecanismele de livrare software, utilitățile și susține optimizarea continuă a produsului și inovația.

Maturitatea aplicației sau a portofoliului de aplicații se realizează pe o scară de la 1 la 4 și oferă și o referință la care aceasta evaluare se raportează.

În mod similar, o astfel de analiză ar permite echipei IT să decidă dacă e necesară o planificare de scoatere din uz a aplicației, când acest lucru va fi posibil, în condițiile în care nu se mai prevede dezvoltarea aplicației sau includerea de noi funcționalități. Evaluarea de maturitate poate indica și faptul ca o aplicație software folosește tehnologii învechite, ceea ce reprezintă un risc de securitate pentru organizație. În aceste circumstanțe se recomandă înlocuirea ei cu o aplicație mai modernă.

Analiza de maturitate DevOps permite validarea necesității business a unei aplicații, precum și a unui întreg portofoliu, prezentând zonele sensibile pentru fiecare dintre ele, zone care trebuie gestionate în timpul transformării DevOps. Evaluarea este realizată pentru fiecare dintre aplicațiile în discuție, independent de tehnologia ce stă la baza programării aplicației. Rezultatele au permis evaluarea cu privire la adoptarea proceselor agile, a mediului de lucru conform normelor DevOps pentru întreg portofoliul de aplicații. Acestea au dus la identificarea acelor procese din întreg ciclul de viață al aplicațiilor, care aduc beneficii pe termen scurt și care aduc valoare adăugată prin procesul de transformare. S-a verificat procesul de dezvoltare, de implementare a codului software și testarea acestuia. Conceptul dezvoltat a fost optimizat pe parcursul desfășurării și dezvoltării proiectului și a permis, odată ce procesele au ajuns la maturitate, scalarea proiectului pilot la întregul portofoliu de aplicații software din departament.

Conceptul și implementarea propusă a fost folosită ulterior la publicarea aplicațiilor în mediile Cloud de aplicațiile evaluate, stabilind un fundament stabil în dobândirea experienței de migrare a aplicațiilor în sisteme de tip Cloud Computing.

Soluția de analiză a maturității pentru aplicațiile planificate include și diverse informații din punct de vedere al procesului de dezvoltare, de testare și de publicare în mediile Cloud. Aceasta permite și identificarea unor potențiale probleme în inter-conectivitatea sistemelor la nivel de platformă sau de ordin organizatoric și social din echipele participante la proces.

Pentru validarea rezultatelor obținute, studiul de maturitate expus a fost realizat și în cadrul altor departamente. S-a constatat ca soluția propusă este un bun premergător al adopției serviciilor Cloud în întreaga organizație.

Pentru vizualizarea globală de poziționare a organizației în procesul de transformare se definesc cele patru stadii existente în procesul de schimbare, Tabel 5.6.1. Obiectivul este de a aduce procesele existente de livrare software la nivelul avansat.

Tabel 5.6.1 Tabelul de referință DevOps

Modelul de maturitate pentru livrarea continuă a aplicatilor software	Incipientă	Premegătoare	Intermediară	Avansată
Sistemul de versionare a codului sursa	Source code	Testare unitară	Baza pe de baze de date relationale	Conține configurațiile sistemelor și de platformă
		Bazat pe fișiere de configurare	Conține un proces de testare funcțională și de asigurare a datelor pentru test. Există un mecanism de ramificare și integrare a codului provenit de la diferite echipe de programatori.	Permite publicarea continuă a codului software în sistemele de producție împreună cu documentația aferentă versiunii de publicare
			Există un mecanism recursiv de a aduce codul software la starea anterioară pentru a putea efectua analiza de cod.	
Procedura de integrare a codului	Manual sau rezoluție prin unități de cod neintegrate	Procese de integrare automate ce sunt programate să integreze codul sursă în fiecare noaptea și cu o inter-dependență ridicată în proces	Integrarea continuă a codului sursă și testare automată la comiterea codului în sistemul de versionare	Componentă automată de integrare a codului Existența unui proces de build
			Depozit de artefacte	Asigurarea infrastructurii ca modul integrat în platforma DevOps Existența unui modul automat de testare în cadrul platformei DevOps
Procesul de verificare a calității codului software și testare	Testare manuală	Există o primă fază dedicată managementului testării care se bazează pe testarea automată unitară și teste automate de integrare	Există un proces de execuție automată a testelor funcționale Posibilitatea analizei statice de cod	Sisteme automate de testare funcțională și nefuncțională a componentelor (validarea performanței sistemelor în regim de încălzire)
				Componentă automată de testare
Modelul de publicare	Manual	Bazat pe fișiere și programe executabile	Există o conductă automată de publicare a codului de tip DevOps	Mecanisme de publicare a codului în sisteme cloud automate în sistemele de producție printr-un singur click Sisteme CI/CD automatizate
	Schimbarea manuală a Schemelor din Baza de date relațională	Utilizează același proces pentru diferite medii de implementare		
Infrastructură	Fizică	Virtuală	Cloud	Sisteme automate de alocare a infrastructurii în funcție de necesitate în Cloud
	Configurarea sistemului este rezoluție manuală	Disponibilitatea sistemului este asigurată prin mecanisme de backup și redundanță	Utilizarea mecanismelor de programare a infrastructurii ca și Cod	
Monitorizarea	Nu există monitorizare	Monitorizare automată la nivel de sistem	Monitorizare la nivel de serviciu software sau aplicație Sistem central de stocare a logurilor și de monitorizare a aplicațiilor	Monitorizarea performanței la nivel de aplicație Permite soluționarea automată a erorilor detectate de acestea în sistem
Arhitectura	Componentele software sunt complet integrate	Bazată pe servicii flexibile și autonome	Orientată pe servicii	Bazată pe microservicii și containere
		Asigurat prin intermediul testării unitare	Proiectată în concordanță cu necesitatea actuală a departamentelor de business. Asigurat testarea automată a componentelor și monitorizarea automată	
Organizarea echipelor de dezvoltare	Model Waterfall Organizarea echipelor de programatori în echipe distincte	Folosind metoda de dezvoltare iterativă cu focus pe colaborarea directă a echipelor de programatori dar și pe documentarea cerințelor business în vederea programării și implementării acestora	Agile Lean	Echipe DevOps mature

5.7 Soluția de implementare

În contextul integrării continue, echipele de dezvoltare și cele de operațional (DevOps) au fuzionat, cu scopul de a produce și de a asigura livrarea de produse software de calitate ridicată în cel mai scurt timp.

Soluția tehnică prezentată se bazează pe metodele, procesele și utilitățile descrise și reușește să implementeze cu succes o conductă de livrare software.

În vederea implementării mecanismului de automatizare a serviciilor în Cloud, autorul a luat în considerare avantajele aduse de către metodologia de dezvoltare agilă și propune pentru optimizarea proceselor de dezvoltare și de integrare a serviciilor software, utilizarea proceselor DevOps.

Dezvoltarea codului include procesele de integrare continuă, de livrare continuă și de publicare în mediile Cloud și constau în dezvoltarea codului, versionarea acestuia, compilarea, asamblarea codului prin împachetarea codului compilat cu librăriile și elementele de configurație necesare rulării acestuia, testarea și publicarea în diferite medii de execuție de tip Cloud.

Este nevoie de prioritizarea muncii înainte ca procesul de dezvoltare să înceapă. De aceea, primul obiectiv este de a avea tot codul sursă produs și configurațiile aplicațiilor versionate în *source code repository*. În această etapă se stabilesc practicile de versionare a

codului, se definesc și se implementează o strategie de *tagging* și *branching*, care conferă mai multă transparență asupra activităților realizate de către echipelor de dezvoltare.

Al doilea obiectiv este definirea procesului de integrare continuă. Prin utilizarea de aplicații integrate și automate sunt necesare câteva componente, cum ar fi, un server pentru integrare continuă, un depozit de artefacte și un server pentru asigurarea calității codului sursă. Se va monitoriza în mod continuu calitatea tehnică a codului software și se va acționa în zonele identificate cu erori tehnice.

Acest pas oferă posibilitatea de a defini dependențele dintre aplicații și poate folosi tehnici software de asigurare a calității, ca de exemplu testele de funcționalitate, testele de integrare, etc. care vor aduce valoare adăugată business-ului.

La pasul al treilea se realizează implementarea automată de integrare software, conectând serverul de depozitare artefacte de mediile de publicare ale aplicațiilor.

A patra etapă constă în furnizarea de infrastructură și gestionarea configurării. Acest pas este utilizat pentru a programa mediile și aplicațiile implicate în proces, pentru a implementa configurațiile în cod programat, ceea ce permite furnizarea în mod automat de medii complet configurate în orice moment în care dezvoltatorul are nevoie de ele. Se va trece la testarea codului creat și la publicarea codului în mediul de producție.

În ultima fază se va stabili monitorizarea continuă și se va configura sistemul destinat gestionării auditării tranzacțiilor din sistem. Acest pas este ușurat prin adăugarea unui sistem de monitorizare care verifică starea aplicației și colectează informații cu privire la resursele folosite. Aceleași principii se aplică gestionării de auditare sau de securitate, care vor facilita accesul securizat numai echipelor de dezvoltare la sistemul DevOps.

Soluția automată propusă pentru livrarea continuă a aplicațiilor în mediile Cloud are la bază procesul de construcție, de integrare și de testare a aplicațiilor bazat pe o suită de componente de tip *open source*, ce alcătuiesc platforma DevOps. Integrarea componentelor este realizată cu ajutorul instrumentului de orchestrare, Jenkins, care la momentul în care identifică unități de cod nou adăugat în sistemul de versionare, va declanșa etapele de construire a pachetelor de cod și va realiza integrarea, testarea automată și publicarea în mediul de integrare. Toate aceste procese se vor declanșa și executa automat.

Prin simpla apăsare a unui buton de salvare a codului programat în sistemul de versionare, sistemul creat va trece prin fazele descrise și va publica codul dezvoltat în mediul Cloud. Codul produs se va construi prin integrarea tuturor bibliotecilor și fișierelor aferente într-un pachet software, va fi testat automat și va fi pregătit pentru o lansare în sistemele productive.

Configurațiile de infrastructură sunt rulate în mediul de execuție la fiecare iterație de publicare a unei noi versiuni de cod, care declanșează procesul de *build*, procesul de integrare și procesul de publicare software.

Fig. 5.7.1 indică execuția unui ciclu de publicare în Cloud.

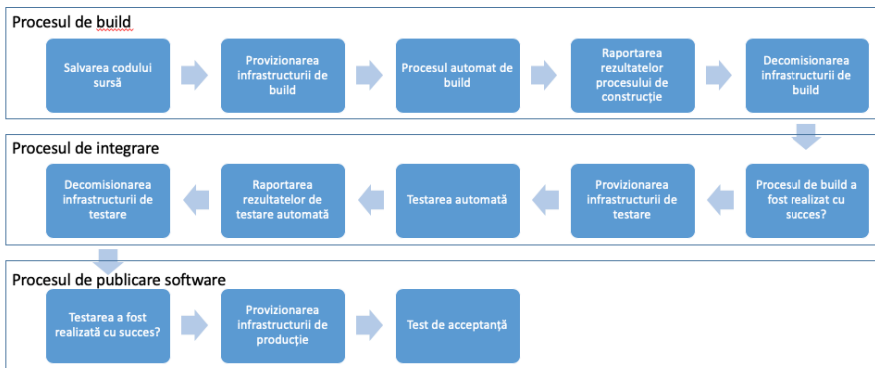


Fig. 5.7.1 Pașii de implementare DevOps

Testarea automată începe odată cu confirmarea realizării pachetului software, moment în care testele funcționale și nefuncționale sunt declanșate automat. Rezultatele testelor vor fi afișate în sistemul de management al defectelor și vor trebui reparate de către dezvoltatori înainte de publicarea software în mediul productiv.

Practicile DevOps sunt concepute pentru a crește eficiența și eficacitatea procesului de livrare software, pentru a crește viteza de publicare a produselor în mediile productive și pentru a scădea costurile și riscurile de producere a aplicațiilor. Pentru o furnizare eficientă de cod, DevOps se bazează pe conceptul unei conduite de livrare software automată. Acest lucru pune în balanță eliminarea erorilor și blocajele întâmpinate la fiecare pas al procesului, pentru a obține o procesare mai rapidă. Metodele de îmbunătățire ale procesului, cum ar fi feedback-ul continuu, reprezintă o parte integrată și importantă a metodei DevOps. Eliminarea eforturilor manuale prin introducerea automatizării și reducerea timpilor de așteptare reprezintă alte criterii, pentru care specialiștii IT optează pentru o conductă automată DevOps. Pentru o gestionare eficientă a procesului de furnizare, DevOps se bazează pe proiectarea unor etape de proiectare flexibile, bazate pe conceptul de agilitate și permite integrarea feedback-ului în oricare etapă, reușind să pună accent pe calitatea și stabilitatea sistemelor. Principiul agil al unui produs software aflat în curs de dezvoltare este extins cu fiecare iterație la un produs software în curs de publicare într-un mediu de producție. Testele de calitate, în special testele de integrare, poziționate încă în faza inițială de dezvoltare a codului, reduc erorile și ajută la împlinirea sintagmei de integrare continuă și de testare continuă.

Este esențială implementarea unui nivel sporit de automatizare în procesul de testare pentru realizarea unui proiect de succes DevOps, dar mai ales pentru menținerea calității codului produs și realizează reducerea semnificativă a timpilor necesari etapelor de livrare software. Măsurătorile indicatorilor de performanță sunt necesari pentru optimizarea continuă a proceselor definite. Punerea în aplicare a analizei și a tehnologiilor cognitive la nivelul sistemului implementat permite îmbunătățirea sistemului și a serviciilor livrate. Soluțiile bazate pe module de inteligență artificială au un impact semnificativ în execuția proceselor DevOps atunci când sunt combinate cu o utilizare a componentelor de monitorizare a performanței aplicațiilor și a tranzacțiilor. DevOps are implicații în diferite zone ale unei organizații plecând de la cultură business și până la mediul arhitectural IT sau componentele software și hardware.

Soluția automată de livrare software propusă se axează pe metodologia DevOps, asigurând în același timp calitatea producerii software, integrarea codului software și testarea automată.

5.8 Implementarea CI/CD

Cadrul construit este o platformă de dezvoltare software care poate fi folosită pentru o varietate largă de aplicații software dezvoltate în limbaje diferite de programare, cum ar fi Java, SQL, C++ sau oricare altă tehnologie de programare. Platforma implementată se caracterizează printr-un grad ridicat de adaptabilitate la orice mediu destinat execuției, ce poate fi un Cloud public, privat sau hibrid. Soluția prezentată are abilitatea configura și personaliza automat mediile de execuție, așa cum este dorit de către programatori sau indicat în codul aplicației dezvoltate.

Conducta DevOps este realizată ținând cont de proceselor de business prezentate DevOps. Aceasta este construită ca un sistem integrat de componente pentru publicarea automată de software. Platforma utilizează tehnologii și instrumente moderne și pentru colectarea cerințelor, pentru gestionarea versionării codului software, instrumente pentru integrarea aplicațiilor, utilitare pentru testarea automată și pentru monitorizarea sistemului, [Gușeală et al., 2019].

Conducta declanșează pașii de execuție la momentul la care dezvoltatorul a salvat cod software nou în sistemul de versionare prin comanda *commit*. Odată ce conducta este implementată, aceasta poate să execute ori de câte ori se realizează modificări de cod software cu costuri de producție minime. Schimbările de cod produs nu impactează disponibilitatea aplicațiilor productive, deoarece calitatea codului produs este testată la fiecare pas, și mai mult decât atât, publicarea codului dezvoltat în mediile de producție se face numai cu aprobarea managerului de publicare. Acest lucru permite echipelor să aducă mici îmbunătățiri suplimentare produselor lor fără să se teamă de potențiale probleme de *downtime* necesare sistemelor de producție.

Conducta DevOps propusă, conține intrări software care promovează sau împiedică automat trecerea unor artefacte create într-o nouă versiune. Dacă protocolul de lansare nu e onorat, porțile software-ului rămân închise, iar conducta anulează execuția în curs. Sunt generate notificări, ce sunt trimise unei liste de distribuție care include membrii ai echipei de dezvoltare pot accesa platforma și stemele integrate și care pot remedia eroarea întâmpinată. Acest lucru garantează faptul că bucată de cod implementată în sistemelor de producție este testată, întregul proces fiind log-at și auditabil.

Cadrul DevOps oferă controlul sursei, construiește pachetele software și asigură testarea automată a codului software produs. Platforma constă dintr-o serie de componente, așa cum este prezentat în Fig. 5.8.1.

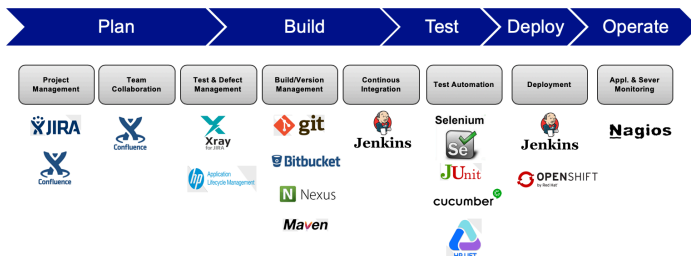


Fig. 5.8.1 Componentele conduitei DevOps

- Modulul de orchestrare a integrării continue

Aceasta este componenta principală a conduitei DevOps care permite ca alte componente să fie gestionate și integrate cu scopul de a produce pachetul software, de a-l testa și, în cele din urmă, de a-l publica în mediul Cloud. Modulul de integrare continuă conține propriul proces de *build*, un sistem de analiză a calității, de testare automată și o unitate de implementare automată a codului.

- Sistemul de versionare al codului

Acesta e reprezentat de o componentă, care permite stocarea codului în *source repository*.

- Modulul automat de testare

Acesta este o componentă a soluției DevOps, care permite administrarea scenariilor de testare funcțională și susține executarea unui plan de testare automatizat, odată ce codul a fost salvat în sistemul de versionare. După execuție, rezultatele sunt făcute transparente dezvoltatorilor și se furnizează feedback imediat către echipa de implementare.

- Modul de planificare a cerințelor

Această componentă contribuie și oferă transparență asupra cerințelor de business pentru toate părțile implicate în procesele de implementare și cele de mentenanță.

- Componenta de feedback

Această componentă autorizează departamentul IT să primească feedback constant și imediat de la utilizatorii platformei, permițând reacții rapide și la timp, în cazul unor probleme sau solicitări.

- Modulul de monitorizare

Aceasta este platforma destinată echipelor de suport care vor evalua, înregistra și le permite să reacționeze rapid și proactiv în cazul producerii de incidente în sistemele de producție.

- Sistemul de management al codului

Aceasta este o componentă din cadrul soluțiilor DevOps, care permite stocarea într-o locație centralizată a pachetelor necesare procesului de implementare împreună cu fișierele binare asociate. Ele se vor stoca într-o locație centralizată.

5.9 Implementarea modului de testare automată

Livrarea continuă de software implică dezvoltarea de software și publicarea acestuia în mediile de producție în orice moment.

Teza introduce o conductă de livrare software oferind un sistem de livrare complet automat de integrare a serviciilor software în centrele de prelucrare a informațiilor. Toate acestea sunt realizate în mod automat, având și un modul de testare implementat în sistemul de livrare software.

Arhitectura se bazează pe un sistem de gestionare a codului pentru integrarea codului și împachetare, pentru configurare și testare automată.

În modelul propus, infrastructura este creată folosind Jenkins, care asigură executarea activităților programate în mod automat la fiecare etapă. Sistemul conține două mașini Jenkins, una dintre ele fiind serverul principal de *build* care servește procesului de integrare continuă (CI), ce reprezintă totodată și interfața cu utilizatorul, iar cealaltă mașină furnizează resurse de calcul adiționale pentru rularea proceselor de *build*. O abordare similară este folosită și pentru testarea automată, având o mașină pentru crearea joburilor de teste și o a doua pentru rularea joburilor de test. Pentru testarea de integrare și de acceptare, software-ul e împachetat și instalat în medii de integrare dedicate destinate testării funcționalităților aplicației de către utilizatorii business.

Fiecare componentă software rulează în conducta CI și oferă feedback programatorilor instant în caz de eroare. Codul produse de către mai multe echipe de

dezvoltatori este folosit la integrarea și împachetarea acestuia, este integrat de către componenta de CI a platformei, este împachetat, testat și ulterior lansat. Atunci când o componenta de integrare continuă a verificat o codul în toate mediile de testare, codul este etichetat ca fiind gata pentru incrementarea produsului și este planificat pentru următoarea etapă de publicare în Cloud, [Gușeală et al., 2019]. Fig. 5.9.1, ilustrează pașii procesului de livrare software și a instrumentelor asociate.

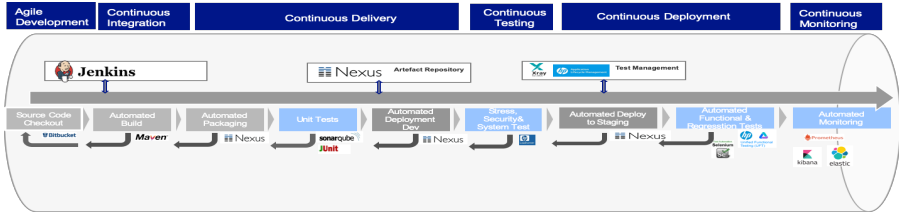


Fig. 5.9.1 Platforma automată de livrare software

Instrumentele folosite în implementare sunt:

- Bitbucket ca și mediu colaborativ de stocare și gestionarea a codului sursă;
- Maven pentru setarea și implementarea produsului software
- Nexus este depozitul care gestionează artefactele binare;
- SonarQube analiza de calitate a codului;
- JUnit pentru testarea unitară;
- Selenium și HP UFT pentru testare automată;
- Jenkins pentru orchestrarea joburilor din conducta CI/CD;
- Ansible pentru a crea toată infrastructura și pentru a aproviziona infrastructura necesară la momentul execuției;
- Monitorizarea automată e implementată folosind Kibana și ElasticStack.

Componentele furnizează funcționalitate separată și facilitează procesul de creare a unui pachet de livrare testat, care în final este implement în mediul de producție.

Pentru a asigura calitatea serviciului pachetului livrat, activitățile de testare trebuie să fie planificate la fiecare pas.

Acestea trebuie să includă:

- scenarii de test pentru cod software, infrastructură și de configurare a componentelor critice din punct de vedere business;
- scenarii de test care sunt executate repetat;
- scenarii de test care sunt complexe și dificil de efectuat manual;
- scenarii de test care necesită mult timp de execuție;
- toate componentele și procesele de business critice pentru organizație;
- scenarii care conțin o cantitate mare de date;
- funcționalități obișnuite ale aplicațiilor;
- componentele care asigură fezabilitatea tehnică a aplicațiilor, de ex. testele de sistem, testele de încărcare și funcționare, etc.).

Testarea este un proces important în dezvoltarea de aplicații și necesită analiza sistemelor încă din fazele incipiente de implementare al codului. Testarea software poate reduce costurile de mentenanță și de dezvoltare software. Testarea software oferă garanții de stabilitate în dezvoltarea de noi caracteristici. Procesul de testare oferă asigurări, că o caracteristică funcționează, așa cum a fost proiectată, iar utilizatorii sistemului nu se vor confrunta cu erori.

În scenariul propus, gestionarea și execuția testului sunt făcute folosind produse dedicate de gestionare a testelor. O eroare identificată este automat raportată în sistemul de management al defectelor. Pentru lansările de cod în producție, pot fi implementate practicile uzuale de management de test, unde adăitional conduitei automate este adăugat un proces decizional, ce va asigura că implementarea în sistemul de producție a fost autorizată și auditată în prealabil. Fig.5.9.2 prezintă o potențială implementare a modului de testare automată în platforma CI/CD.

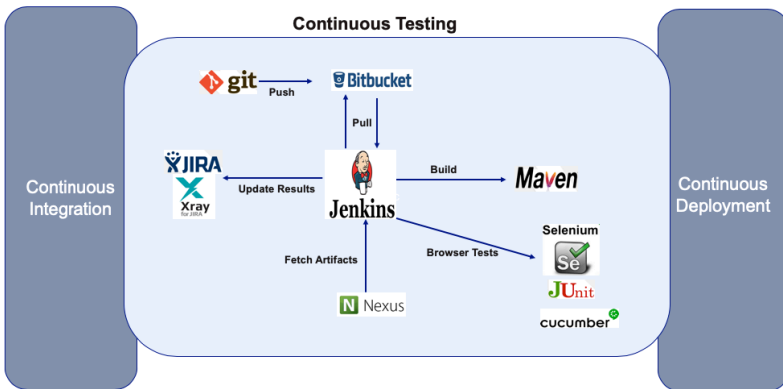


Fig. 5.9.2 Modulul de testare automată

În faza de scriere de cod, instrumentele folosite iau ca valoare de intrare, orice schimbare în sistemul de gestionare al cod, împachetează codul identificat și inițiază rularea automată a testelor.

Prima fază de testare în testarea continuă este reprezentată de testările unitare, care construiesc baza de testare pentru validarea codului produs. Testările unitare sunt dezvoltate de programatori, și reprezintă cea mai rapidă metodă de a detecta erori de calitate în cod într-o fază incipientă.

Este necesar ca programatorii să creeze scenarii de testare unitară pentru a acoperi întreaga paletă de cod și mai ales pentru a acoperi un procent de 70% din procesul de testare, din punct de vedere durată de timp, obiectivele de testare și numărului de scenarii de testare existente. Toate scenariile de testare trebuie întreținute și actualizate, pentru a asigura calitatea produsului software lansat. Orice eroare identificată trebuie rezolvată imediat.

Analiza codului static este făcută după realizarea testării unitare și de build. Acest lucru contribuie la identificarea erorilor de cod sau a breșelor de securitate, și asigură faptul că normele de programare au fost respectate. Orice problemă identificată trebuie rezolvată înainte ca pachetul software să fie introdus în mediul de integrare. În cazul proceselor de build care sunt executate noaptea, se recomandă planificarea mai multor testări automate și analize de cod statice, prin verificare cerințelor de bază.

În faza de testare, mediul de test este creat pentru a oglindi mediul de producție. Rularea procesului de integrare automată, verificarea componentelor și a validarea sistemului prin testare reprezintă a doua fază a testării continue. Aceasta asigură disponibilitatea infrastructurii și a mediului de execuție al sistemului. În vreme ce testele de integrare compară automat interfețele componentelor cu design-ul sistemului, pentru a verifica robustețea și integritatea aplicației software, testarea componentelor verifică compatibilitatea dintre componente și procesarea de date. Testarea de sistem verifică întregul sistem *end-to-end* de la interfața grafică, la API și la sistemele *backend*.

Testele de performanță automate cum ar fi testele de performanță, de securitate și de conformitate, sunt rulate automat în timpul aceleiași faze de testare automată. Scopul testului de performanță este să verifice promptitudinea sistemului sub o încărcare ridicată, în timp ce măsoară atributele calității, cum ar fi viteza de procesare, utilizarea resurselor, scalabilitatea, etc. Aceste teste sunt folosite împotriva criteriilor de standardizare a aplicației.

Testele funcționale automate, testele de acceptanță ale utilizatorului, sunt realizate în mediile de integrare și asigură faptul că cerințele funcționale au fost implementate în conformitate cu descrierea dată de business. Această fază validează fluxul de business *end-to-end*.

Studiul a arătat faptul că este necesar un proces de dezvoltare agil pentru a elabora și a furniza un sistem dedicat mediului Cloud cu componente distribuite. Lucrarea a prezentat procesul, metodele și instrumentele necesare introducerii automatizării pentru integrarea continuă, pentru implementare codului și pentru testare automată în dezvoltarea de software de tip Cloud.

În acest capitol, este documentată și detaliată o arhitectură de referință pentru implementarea de software automat și testarea automată, odată cu pașii necesari implementării în vederea creării unei arhitecturi de referință pentru livrarea automată de servicii software în mediile Cloud.

Se evidențiază atingerea următoarelor obiective în implementarea sistemului automat de livrare software:

- Standardizarea întregului proces de livrare software luând în calcul activitățile manuale executate de către programatori. Pentru acest pas, cele mai bune practici sunt exemplificate;
- Identificarea componentelor care construiesc o soluție de arhitectură, care susține procesul de integrare automată și implementarea codului produs în regim continuu;
- Identificarea de scenarii de test și fazele de testare necesare. Este prezentat la modul de asigurare a calității a produsului software produs prin intermediul folosirii unei conducte de tip DevOps utilizând cele mai bune practici.

Integrarea de instrumente și de tehnologii moderne este esențială pentru livrarea de servicii Cloud. Platforma pentru CI/CD prezentată, împreună cu modulele implementate pentru testare automată se aliniază îndeaproape cu practicile și metodologia de dezvoltare agilă. Abordarea prezentată pentru testarea continuă furnizează o viziune clară asupra calității software-ului livrat și susține furnizarea de produse robuste, stabile și de înaltă calitate către client.

5.10 Procesul automat de publicare și de implementare software

Odată cu nevoile de business în permanentă schimbare, procesul de gestionare a livrării include multiple sprint-uri pentru a face ca produsul să fie disponibil în sistemele de producție.

Metodele agile prezentate, integrarea de software continuă, implementarea și testarea joacă un rol esențial în gestionarea mai rapidă a codului produs și în securizarea calității produselor.

Utilizând o conductă de livrare software pentru automatizarea integrării codului, pentru construirea acestuia și pentru testarea automată a produsului livrat se poate îmbunătăți considerabil calitatea software-ului și viteza de livrare a acestuia.

Crearea și standardizarea procesului de livrare și implementarea prin utilizarea mecanismelor automate pentru livrarea software au rolul de a simplifica transferurile de informații între participanții la procesul DevOps.

Odată ce codul produs a fost validat și acceptat, urmează validarea punerii acestuia în sistemele de producție de către management. Noul cod dezvoltat devine disponibil pentru toate serverele de producție și utilizatorii aplicației.

Instrumentele de monitorizare ale sistemului verifică și validează comportamentul funcțional și nefuncțional al sistemului, a componentelor acestuia disponibilitatea interfețelor și corectitudinea procesării datelor.

Utilizând o platformă de management pentru implementările tehnice de aplicații software în mediile de producție, de gestionare a defectelor tehnice și de validare a codului, este posibilă gestionarea procesului de publicare, ce devine transparent pentru toate părțile implicate în proces.

Prin activarea procesului de integrare software, codul este împachetat, construit și pregătit pentru livrarea în mediile de Cloud. Cu ajutorul *job*-urilor create de Jenkins în procesele de integrare, de construcție și de testare, procesul de livrare și publicare software este gestionat automat. Rezultatele de testare sunt vizibile echipelor de dezvoltare și de suport, astfel încât ele pot interveni la momentul apariției unei erori de natură tehnică.

Aceeași transparență a rezultatelor de testare este oferită și utilizatorilor business, care sunt implicați în procesul de acceptanță a codului software pentru mediile productive.

Fig. 5.10.1 prezintă procesul de livrare agil în sistemele de producție și afișează o viziune de ansamblu a procesului de publicare, împreună cu actorii implicați în ciclul de viață de producere a programelor software, [Gușeală et al., 2019].

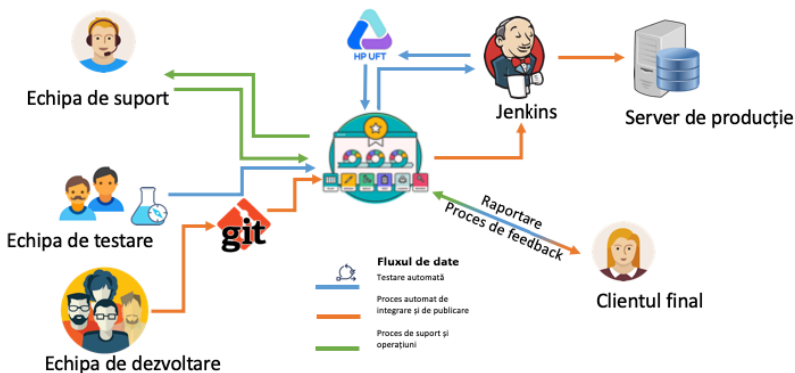


Fig. 5.10.1 Procesul de publicare si implementare

Procesul de asigurare a calității software trebuie să asigure corectitudinea proceselor de business și execuția corectă a aplicației. Testarea programelor software este realizată pentru a aduce asigurări că sistemele tehnice produc rezultatele așteptate și produsul software se comportă așa cum a fost proiectat.

Testarea continuă necesită un mod de abordare specific bunelor practici de testare, utilizând instrumente de testare moderne. Succesul implementării constă în feedback-ul rapid obținut de la toate părțile implicate în procesul de livrare software și integrarea acestuia în etapele de dezvoltare în cel mai scurt timp.

În cazul apariției de erori de natură tehnică sau funcțională, sistemul permite crearea, gestionarea, prioritizarea și urmărirea erorilor până când acestea sunt rezolvate de către echipele de programatori.

Procesul de testare automată implică executarea de teste funcționale și nefuncționale pe parcursul livrării software cu scopul de a obține un feedback imediat cu privire la calitatea programului software produs.

Conducta de livrare software implementată poate asigura integrarea automată a codului programat, construcția automată și publicarea codului, și conține totodată un modul de testare automată pentru a asigura o calitate a produsului software livrat în Cloud.

Utilizând soluția propusă, calitatea codului este asigurată și optimizată, produsele software noi dezvoltate sunt livrate rapid și buna funcționalitate a aplicațiilor este asigurată. Sistemul central de gestionare a job-urilor oferă transparență pe parcursul întregului proces de livrare software.

5.11 Monitorizarea sistemului software și a infrastructurii

Odată implementată în Cloud, soluția este în permanență monitorizată folosind tool-uri moderne, ce asigură stabilitatea sistemului și verifică performanța componentelor.

În mediul IT, erorile identificate în sistemele de producție, pot apărea dacă dezvoltarea software nu a fost realizată în conformitate cu specificațiile de proiectare sau de codare, sau dacă testarea codului nu a fost suficientă, precum și de configurări greșite ale componentelor [Bass et al., 2013], [Hamilton, 2007].

Există soluții tehnice, ce încorporează utilitare de analiză de date și de proces în componentele de monitorizare cu scopul identificării pro-active de comportamente neadevrate în utilizarea a software-ului implementat.

În lucrarea sa, autorul accentuează necesitatea implementării utilităților de monitorizare, care detectează erori sau probleme de performanță și, care sunt cu atât mai valoroase la repornirea proceselor încheiate în eroare pentru a asigura redundanța sistemului [Fazal-Baqaie et al., 2017]. În cazul în care mecanismul automat nu reușește repornirea componentelor defecte, atunci echipă de suport este alertată pentru a rezolva incidentele din sistemul de producție.

Mecanismele de monitorizare, de verificare și de alertare, în caz de defecțiune tehnică, au un rol esențial în asigurarea stabilității componentelor sistemului implementat.

5.12 Scalarea sistemului

Scalarea sistemului se face utilizând containere. Containerele simplifică procesul de testare și de implementare software. Acestea permit împachetarea codului software într-un singur container portabil, care poate fi mutat ușor de pe un server pe altul pe tot parcursul ciclului de viață al implementării. Containerele facilitează integrarea și livrarea continuă furnizând medii de aplicații stabile, sigure și eficiente de dimensiuni mai mari sau mai mici.

Odată configurate, containerele permit rularea imediată a aplicațiilor în mediul de producție și permit echipelor de dezvoltare să înceapă procesul de testare pentru a valida impactul cauzat de schimbările efectuate. Acest lucru permite echipelor DevOps să repare și remedieze imediat erorile tehnice identificate, ceea ce reduce efortul operațional și grăbește procesul de dezvoltării. Adoptând dezvoltarea aplicației folosind containere, se poate practica integrarea și livrarea continuă de software mult mai eficient.

Asocierea elementelor de CI/CD împreună cu containerele de aplicații are potențialul de a automatiza complet procesul de livrare software, permite testarea eficientă a codului și implementarea acestuia în mediile Cloud. Scalarea aplicațiilor folosind containerele este foarte ușor de gestionat, iar Cloud oferă mediul optim pentru balansarea componentelor, pentru optimizarea resurselor și pentru a asigura stabilitatea serviciilor.

5.13 Beneficiile implementării soluției propuse

Spre deosebire de modelul tradițional de implementare a aplicațiilor software în sistemele de producție, modelul propus este unul care folosește conceptele metodologiei agile, dar și elemente DevOps prin implementarea modulelor de dezvoltare continuă a funcționalităților aplicațiilor. Soluția folosește și integrează tehnologii noi și inovative și asigură integrarea automată a serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor.

Aplicațiile și noile funcționalități sunt puse la dispoziția utilizatorilor în scurt timp și respectă totodată termenii de calitate și de stabilitate ale aplicațiilor.

Metoda propusă elimină “zidul de conflict” între utilizatorul de business, echipa de dezvoltare și cea operațională, printr-o metodă, care pune accent pe valoarea adăugată organizației și elimină erorile de natură tehnică și timpul îndelungat de pregătire a aplicațiilor pentru punerea în producție.

Un alt beneficiu al soluției implementate este automatizarea și standardizarea întregului proces de dezvoltare software și publicarea serviciilor în Cloud. Implementarea propusă înlătură potențiale erori umane ce pot apărea la configurarea sistemelor și duce la eficientizarea întregului ciclu de dezvoltare al serviciilor Cloud asigurând o calitate ridicată a serviciilor.

Automatizarea este o practică esențială utilizată în vederea standardizării proceselor și nu numai pentru adoptarea DevOps. Automatizarea poate furniza echipelor de suport un mijloc sigur, transparent și sigur cu privire la starea de funcționalitate a sistemului implementat.

Studiul arată că aplicația sau funcționalitățile nou programate pot fi publicare în medii Cloud la intervale scurte de timp, sistemul creat asigură calitatea serviciilor livrate prin testarea automată și arată că tendința de a aduna mai multe schimbări de cod într-un singur ciclu de publicare, efectuat la intervale mari de timp este învechită și riscantă.

Pentru a face procesul de publicare transparent echipelor de dezvoltare software și de suport, automatizarea a fost construită astfel încât este posibilă afișarea de mesaje și notificări cu referire la progresul procesului de livrare software. Cloud și DevOps creează mediul optim de inovație prin folosirea de tehnologii și utilitare moderne, care sporesc abilitatea de plasare a produselor noi pe piață.

Întrucât organizațiile de astăzi se axează pe adoptarea DevOps în vederea realizării unei livrări de servicii mai rapide și mai fiabile către clienții săi, soluția tehnică propusă vine să susțină abordarea strategică a unei organizații de transformare pentru a furniza servicii în Cloud.

5.14 Concluzii

Companiile de astăzi utilizează software în desfășurarea activităților lor. Ele devin dependente de folosirea soluțiilor IT moderne, dar și de realizarea de noi mijloace software adaptate propriilor necesități. Dezvoltarea software are un rol esențial în procesul de creare și de furnizare de valoare prin intermediul produselor și serviciilor lansate pe piață.

Introducerea cu succes a modificărilor tehnologice menite să răspundă la permanenta nevoie de schimbare a cerințelor clienților, sau la schimbările date de noile direcții strategice din organizație, este esențială în livrarea de produse și servicii IT. Pentru aceasta se are în vedere implementarea unei soluții automate pentru generarea de medii configurate și personalizate în procesul de dezvoltare software.

În studiu s-a arătat ca abordarea DevOps pentru crearea de soluții software în Cloud aduce beneficii atât la nivel de organizație, cât și în relația cu clienții. DevOps are totodată și un impact semnificativ asupra modelului de lucru din punct de vedere organizatoric.

Autorul explorează metode, instrumente și tehnici din diferite zone ale domeniului de inginerie software, arată cerințele de implementare și de arhitectură software necesară și studiind literatura de specialitate contribuie la definirea unui sistem IT pentru gestionarea procesului de dezvoltare software agil folosind automatizarea.

Autorul dezvoltă un model de implementare a serviciilor software Cloud ce poate fi utilizat de către organizațiile mici, medii și mijlocii. S-a arătat procesul de dezvoltare al aplicațiilor software folosind metode tradiționale și agile și s-a evidențiat necesitatea folosirii metodei agile de lucru în combinație cu utilitățile DevOps cu scopul de a extinde durata de viață a produselor software.

Produsul software creat va încorpora atribute pentru sporirea flexibilității și agilității în procesele de dezvoltare software. În final, autorul a prezentat un sistem ce permite dezvoltarea, integrarea și publicarea automată de aplicații, ce au la bază tehnologii diverse sau care au aplicabilitate domeniilor diferite.

Autorul a arătat rolul central pe care atât DevOps, cât și Cloud îl dețin în vederea transformării digitale. Furnizarea de servicii inovative este prezentată ca bazându-se pe un set de metodologii și de practici care completează sistemele hardware și software existente în organizație în vederea livrării de produse și servicii software.

Analiza de portofoliu a aplicațiilor existente este introdusă ca și cadru pentru pregătirea mediului IT în vederea adoptării DevOps și Cloud.

Metodele DevOps prezentate în acest capitol permit integrarea răspunsurilor venite de la utilizatorii de business, măresc calitatea software-ului creat, reduc timpul de lansare pe piață al produselor IT și reduc costurile asociate producerii acestora.

prezentată arată metode de optimizare a calității și a vitezei de livrare a serviciile furnizate.

Lucrarea evidențiază următoarele aspecte importante pentru integrarea de servicii de tip Cloud Computing în centrele de prelucrare a informațiilor:

- Standardizarea întregului proces de livrare software cu privire la sarcinile manuale executate într-un anumit mediu sau sistem. Pentru acest pas cele mai bune practici sunt exemplificate;
- Identificarea componentelor necesare, care construiesc o arhitectură robustă pentru îndeplinirea integrării și construcția codului software;
- Identificarea de procese, scenarii și faze de testare necesare în vederea integrării acestora în procesul de dezvoltare software;

- O analiză a cercetărilor în domeniu cu privire la procesul și etapele de dezvoltare software pentru servicii Cloud, plecând de la metodele de programare waterfall și agile;
- O analiză sistematică a cercetărilor existente cu privire la metodologia DevOps;
- Caracteristicile esențiale și tool-urilor de dezvoltare software;
- Identificarea și proiectarea unei arhitecturi, a unor utilitare care sunt necesare la implementarea unei conduite DevOps și un mecanism de integrare al acestora;
- Implementarea unei conduite DevOps pentru livrare automată de aplicații software în medii Cloud;
- Implementarea se bazează numai pe utilitare de tip *open-source*, ceea ce presupune costuri de licențiere sau dependența de un anumit producător de produse software;
- Sistemul asigură calitatea serviciilor furnizate, susține standardizarea proceselor și ajută la reducerea erorilor umane cu ajutorul automatizării.

Studiul a arătat că este necesar un proces de dezvoltare software agil, care să permită interacțiunea și stabilitatea componentelor Cloud aflate într-un mediu distribuit.

Lucrarea a prezentat procesul, metodele și instrumentele necesare introducerii automatizării pentru livrarea de aplicații software de tip Cloud, ținând cont de etapele de integrare continuă, de implementare și de testare automată în dezvoltarea unui program software de calitate.

Integrarea de instrumente și tehnologii este esențială pentru livrarea de servicii Cloud. Platforma CI/CD prezentată, împreună cu modul de testare automată se aliniază îndeaproape cu practicile și metodologia de programare agilă. Abordarea prezentată pentru testarea continuă furnizează o viziune clară asupra calității produselor software livrate și ajută la furnizarea de produse robuste, stabile și de înaltă calitate clientului final.

Acest capitol furnizează un fundament pentru adoptarea de soluții inovative în livrarea de servicii IT din mediile Cloud. Implementarea propusă este creată în strânsă legătură cu principiile și procesele DevOps, care pot fi ușor utilizate și integrate în etapele de dezvoltare software.

Soluția prezentată poate fi utilizată la integrarea de servicii software în toate mediile Cloud Computing, fie ca acestea sunt de tip public, hibrid sau privat.

Lucrarea prezentată va fi extinsă în capitolul următor pentru a include sisteme inteligente de monitorizare și alertare la implementarea de servicii inovative Cloud. Se utilizează în acest sens tehnologia și capacitatea inteligenței artificiale de a verifica procesele existente, se va implementa un mecanism de analiză și identificare a erorilor tehnice în etapele de dezvoltare a aplicațiilor software, astfel încât intervenția umană să poată fi redusă la minim.

6. Optimizarea sistemului propus folosind inteligența artificială

Automatizarea este un factor important în vederea eliminării eforturilor manuale. Indiferent că este vorba despre software IT sau automatizare de infrastructură, procesul de automatizare joacă un rol important în accelerarea livrării de aplicații Cloud prin eliminarea activităților manuale.

Managementul IT și utilitățile de automatizare creează procese repetabile, standardizate și norme de utilizare, care pot înlocui sau reduce intervenția factorului uman. Aceste mecanisme de automatizare pot fi extinse mai departe în direcția utilizării unor tehnologii specifice, ca de exemplu, la utilizarea containerelor, sau a unor metodologii de lucru, cum ar fi DevOps. De asemenea, ea poate fi utilizată și în domenii extinse, de tip Cloud Computing prin automatizarea securității, automatizarea testării, a monitorizării și sistemului de alertare în caz de eroare.

Prin urmare, automatizarea este elementul central de optimizare a proceselor IT, care ajută organizațiile în implementarea de tehnologii inovative, ce duc la adoptarea transformării digitale la nivel organizațional.

Din acest motiv, autorul se va concentra pe cele mai utilizate patru abordări legate de optimizarea continuă a produselor software în Cloud:

- Monitorizarea performanței aplicației (*Application Performance Monitoring, APM*);
- Automatizarea proceselor de business;
- Învățarea automată în scopul identificării problemelor de natură tehnică în sistem;
- Inteligența artificială în scopul auto-remedierii erorilor de natură identificate.

6.1 Monitorizarea performanței aplicației

Pentru majoritatea organizațiilor, monitorizarea performanței aplicațiilor este văzută ca un element esențial în activitățile efectuate la nivel operațional în procesul de management al aplicațiilor. Monitorizarea performanței reprezintă, prin intermediul folosirii DevOps, o metodă de colaborare a echipelor de operațional cu cele responsabile de dezvoltarea aplicațiilor, îmbinând astfel IT-ul și procesul business în vederea digitalizării organizației.

În domeniul tehnologiei informației și a managementului de sisteme IT, monitorizarea performanței unei aplicații reprezintă monitorizarea și gestionarea componentelor aplicațiilor software și a proceselor în vederea asigurării unei disponibilități ridicate a serviciilor software pentru utilizatori.

Când vine vorba de măsurarea performanțelor aplicației, există doi parametri critici care sunt de menționat.

Primul set de metrici de performanță definește performanța experimentată de utilizatorii finali ai aplicației. Aceasta este caracterizată de timpul mediu de răspuns al aplicației în condiții de sarcină maximă. Componentele care indică criteriile de măsurare a performanței unei aplicații includ încărcarea la sarcină ridicată și măsurarea timpilor de răspuns ai aplicației.

Al doilea set de metrici de performanță măsoară resursele de calcul utilizate de către aplicație în regim de sarcină, și indică dacă există capacitate adecvată în sistem pentru a susține sarcina. Acesta poate indica primele locații, unde vor apărea probleme de funcționare în aplicație în regim de sarcină. Analiza performanței poate da primele indicii de probleme în funcționalitatea aplicației în regim de sarcină.

Analiza de performanță a aplicației este utilizată pentru a stabili cauza inițială a problemei, care poate fi cauzată de un volum de date crescut generat la momentul rulării aplicației. Scopul acestei analize este de a anticipa mai bine și a face analiza potențialelor probleme, ce ar putea apărea la utilizarea aplicației de către utilizator.

La implementarea tool-urilor APM pentru aplicațiile Cloud este necesară identificarea nevoilor organizației, deoarece piața IT pune la dispoziție o gama largă de produse. Cu toate acestea există o suită de provocări, atunci când vine vorba despre implementarea utilităților de tip APM.

În vederea soluționării primei situații, procesul de management al aplicațiilor propune analiza având ca subiect central aplicația, unde vizualizarea metricilor cu impact asupra performanței business-ului reprezintă obiectivul principal.

A doua problemă prezentă în aplicațiile distribuite, virtualizate și care rulează în Cloud, se prezintă ca o provocare și are o soluție mai complexă, deoarece în vederea monitorizării performanței aplicației, majoritatea componentelor de sistem nu mai sunt găzduite de un singur server, ci de cele mai multe ori aceste componente sunt distribuite în rețea. Fiecare componentă e acum proiectată în rețea și este oferită ca un serviciu prin intermediul conexiunii la Internet. În Cloud însă, în cazul apariției unor probleme de genul acesta, aplicațiile pot fi migrate de pe un sistem pe un altul, reușind astfel să fie menținute obiectivele contractuale de livrare ale serviciilor și să preîntâmpine posibilele întreruperi de funcționare a aplicațiilor.

Dimensiunile principale introduse de Gartner fac referire la:

- Experiența utilizatorului final

Experiența utilizatorului presupune măsurarea traficului de date de la momentul lansării cerinței utilizatorului și până la obținerea unui răspuns. Procesul se referă la captarea experienței utilizatorului final (End User Experience, EUE). Rezultatul acestei măsurători este dat de monitorizarea aplicației în timp real și este o măsurătoare de sus în jos. Acest tip de monitorizare are două componente una pasivă și alta activă.

- Tranzacțiile de business

Managementul tranzacțiilor de business este cunoscut și ca monitorizarea proceselor de business și reprezintă numărul de tranzacții efectuat de către aplicații, sau numărul de tranzacții efectuate de către utilizator. Managementul tranzacțiilor de business arată în mod transparent fluxul de tranzacții la nivel de arhitectură a aplicației și prezintă o mapare dinamică asupra componentelor aplicației.

- Analiză și raportare

Este important să existe un set comun de metrici, care să colecteze și să raporteze date pentru fiecare aplicație, apoi să standardizeze datele plecând de un element comun și să prezinte datele legate de performanța aplicației. Colectarea de date neprelucrate din celelalte utilitare de analiză a modelului APM oferă transparență în raport cu funcționalitatea aplicației.

6.2 Automatizarea proceselor de business

Automatizarea proceselor de business poate avea un impact mare asupra abilității organizației de a schimba și de a menține o calitate ridicată a serviciilor livrate, dar și de a reduce costurile. Automatizarea proceselor de business constă în analiza, documentarea, optimizarea, standardizarea și ulterior, automatizarea proceselor de business. Aceasta se bazează pe eliminarea procedurilor redundante, reduc eforturilor manual din activitățile de zi cu zi și ajută la implementarea de aplicații software stabile în organizație prin intermediul

standardizării. Automatizarea proceselor de business asigură fiabilitatea sistemelor și reușește să optimizeze la nivel global procesele organizației.

- **Automatizarea procesului de business**

Automatizarea procesului de business este automatizarea realizată prin intermediul tehnologiei a proceselor de business complexe. Aceasta poate eficientiza un proces business simplificându-l și împărțindu-l în unități mai mici, care ulterior vor fi standardizate și automatizate.

Această metodă crește calitatea serviciilor livrate, îmbunătățește livrarea serviciilor și poate reduce costurile de mentenanță cu aplicațiile și sistemele IT. Automatizarea procesului de business constă în integrarea aplicațiilor într-un sistem, care este capabil să susțină procesele business prin intermediul tehnologiei și reușește să reducă cheltuielile de personal. Automatizarea proceselor prin folosirea roboților este un domeniul emergent în cadrul automatizării procesului de business și folosește inteligența artificială.

Monitorizarea automată este esențială pentru ca fiecare proiect să fie unul de succes. Este important de menționat că aplicația sau serviciul pus la dispoziție clientului final trebuie să fie disponibil și să funcționeze conform criteriilor definite în contractele de furnizare a serviciilor IT. Majoritatea aplicațiilor încearcă să furnizeze o disponibilitate de aproape 99.9%, lucru care nu este mereu ușor de realizat. Monitorizarea automată a procesului de business este metoda cea mai sigură de a asigura funcționarea permanentă a aplicațiilor.

Întrucât așteptările utilizatorilor de servicii Cloud sunt de a beneficia de o disponibilitate ridicată a serviciilor, specialiștii IT trebuie să ia în calcul la definirea arhitecturii sistemelor software și hardware de implementarea de sisteme de monitorizare, care măsoară timpii de răspuns ai aplicației.

Sistemele de monitorizare trebuie să ofere un răspuns la fiecare câteva minute, indiferent de locația de unde este oferit serviciul.

Beneficiile utilizării monitorizării automate a proceselor de business pot optimiza per total funcționalitatea sistemelor și a aplicațiilor IT. Ele ajută la menținerea unor procese de business robuste și stabile și asigură o experiență de utilizare a aplicațiilor software mai bună pentru utilizatorul final.

Un alt beneficiu constă în abilitatea de a identifica probleme de natură tehnică la nivel de aplicație sau la nivel de proces business, înainte ca aceste erori să fie sesizate de către utilizator.

În această fel, orice eroare, care ar putea duce la întreruperea sau buna funcționare a sistemelor poate fi evitată.

Un sistem de monitorizare implementat trebuie să poată identifica tendințele, care ar putea duce la apariția unei probleme tehnice în viitorul apropiat. Soluționarea erorilor tehnice înainte ca ele să apară prin analiza a datelor existente în sistem poate reprezenta un criteriu esențial în menținerea proceselor de business la un nivel stabil și performant.

6.3 Învățarea automată

Învățarea automată se ocupă cu studiul algoritmilor și modelelor statistice utilizate de către sistemele computaționale pentru a realiza o anumită activitate fără a folosi instrucțiuni detaliate. Mecanismul de învățare se bazează pe modelele de interpretare a datelor.

Învățarea automată de învățare este văzută ca un domeniu ce ține de inteligența artificială. Algoritmii de învățare automată construiesc un model matematic bazat pe date simple, cunoscute sub denumirea de date de testare cu scopul de a face predicții sau de a lua decizii fără a fi fost programat în prealabil pentru execuția acelei activități. Algoritmii de învățare automată sunt utilizați într-o gamă largă de aplicații de date nestructurate.

Învățarea automată ca metodă de calcul se bazează pe statistică, și realizează prognoze prin intermediul resurselor de calcul. Modelul matematic furnizează metode, baze teoretice și domenii de aplicații pentru învățarea automată. Extragerea informațiilor din cadrul învățării automate se realizează pe analiza datelor experimentale și este cunoscut ca învățarea nesupravegheată. În aplicarea acestuia la diferite probleme de business, învățarea automată este cunoscută și sub denumirea de analiză predictivă.

De obicei, modelele de învățare automată necesită un volum mare de date pentru a putea funcționa în mod optim. În general, la formarea unui model de învățare automată este nevoie de colectarea unui eșantion mare de date reprezentative pentru un anumit exercițiu de învățare. Datele pot varia de la corpurile unui text și până la o colecție de imagini sau de date colectate de la utilizatorii unui serviciu. Este de dorit să se evite încărcarea sistemului cu prea multe informații la momentul formării unui model de învățare automată.

- **Învățarea supervizată**

Algoritmii de învățare supervizată construiesc un model matematic format dintr-un set de date, care conține atât informațiile, cât și rezultatele dorite. Datele sunt cunoscute ca și date de intrare și sunt formate dintr-un set de eșantion de date de testare. Fiecare eșantion de date de testare este reprezentat printr-o serie de numere sau de un vector, numit vector caracteristic.

- **Învățarea semi-supervizată**

În cazul algoritmilor de învățare semi-supervizată, există elemente cărora le lipsesc etichetele de antrenament la momentul începerii procesului de învățare. Acestea însă pot fi utilizate pentru a îmbunătăți calitatea modelului.

Folosind învățarea semi-supervizată, algoritmii dezvoltă modele matematice din date incomplete la momentul pornirii antrenamentului. Datele furnizate în procesul de învățare nu au etichete.

- **Învățarea nesupervizată**

Algoritmii de învățare nesupervizată folosesc un set de date, care conțin informații, și găsesc o metodă de a structura acele date, prin gruparea punctelor din datele furnizate la începutul procesului de învățare. Astfel algoritmii învață din datele de testare care nu au fost etichetate, clasificate sau categorisite. Algoritmii învățării nesupervizate identifică similitudinile dintre date și răspund în prezența sau în absența unor astfel de similitudini prin livrarea de noi grupuri de date.

Analiza unui grup de date este realizată prin divizarea unui set de observații în subseturi (numite clustere), astfel încât observațiile din același cluster sunt procesate similar și în conformitate cu unul sau mai multe criterii prestabilite.

În învățarea nesupervizată, algoritmul construiește un model matematic dintr-un set de date care conține doar informații, nu și rezultate dorite. Algoritmii de învățare nesupervizată sunt folosiți pentru a găsi structură în date. Învățarea nesupervizată poate descoperi șabloane în datele furnizate și poate grupa informațiile pe categorii, așa cum este cazul în învățarea de noi funcționalități. Diminuarea dimensiunii de învățare este procesul de reducere a numărului de elemente sau informații dintr-un set de date.

- **Învățarea asistată**

Învățarea asistată este o arie a învățării automate, care se ocupă cu modul în care agenții software trebuie să acționeze într-un sistem pentru a maximiza noțiunea de recompensă. Mulți algoritmi de învățare asistată folosesc tehnici de programare recursive. Algoritmii de învățare asistată nu își asumă cunoașterea unui model matematic exact, dar pot fi folosiți atunci când modelul de date nu este structurat.

În învățarea asistată, algoritmii accesează etichetele de învățare dorite ca pe un set limitat de informații și de date. Algoritmul poate fi optimizat prin alegerea automată a

informațiilor pentru care se face acumularea de etichete în faza de antrenare. Când sunt utilizate interactiv, acestea pot fi prezentate unui utilizator pentru etichetare.

6.4 Învățarea în adâncime

Învățarea în adâncime este un subdomeniu al învățării automate, care conține algoritmi de învățare și care acționează similar funcțiilor creierului. Acestea sunt numite rețele neurale artificiale. Învățarea în adâncime face parte dintr-o familie mai mare a metodelor învățării automate bazată pe rețele neurale artificiale. Învățarea poate fi supervizată, semi-supervizată sau nesupervizată. Majoritatea modelelor de învățare în adâncime se bazează pe rețele neurale artificiale.

Învățarea în adâncime are nevoie de multe etape de antrenament în cadrul activității de învățare. Cloud Computing este un sistem de calcul în care resursele de calcul sunt furnizate și alocate la cerere. Cloud-ul reprezintă o alegere ideală ca platformă pentru sistemele de învățare aprofundată, deoarece oferă servere, spații de stocare și acces la rețea la cerere. Aceste caracteristici oferă elasticitate sistemului, procesare rapidă a datelor, capacitate de stocare a datelor și poate furniza resurse adiționale necesare procedurii de analiză.

Acest tip de învățare automată este specific sistemelor, care pot învăța din execuțiile efectuate anterior și pot astfel acumula abilități și date adiționale fără implicarea umană.

În cazul învățării în adâncime, fiecare nivel de învățare va transforma datele primite la nivel de intrare într-o reprezentare mai abstractă. Algoritmii învățării în adâncime, pot fi aplicați în cazul sarcinilor de învățare nesupervizată. Acesta oferă un beneficiu în lumea datelor, deoarece volumul de date neetichetate în sistemele IT este cu mult mai mare decât volumul datelor etichetate.

Având în vedere că învățarea în adâncime se referă la numărul straturilor utilizate, această tehnică necesită mai mult spațiu de stocare pentru volumul mare de date necesar în timpul antrenamentului. Întrucât cerințele de procesare de date sunt în continuă creștere și aceste activități consumă un număr mare de resurse computaționale.

Învățarea în adâncime se bucură de multă atenție în zilele noastre și asta datorită rezultatelor remarcabile care nu puteau fi atinse de către calculatoarele tradiționale, [Perniu et al, 2019].

Învățarea în adâncime reușește să atingă o acuratețe foarte bună de recunoaștere la nivele superioare față de cele inferioare. Aceasta ajută echipamentele electronice de larg consum să satisfacă cerințele utilizatorului. Utilizarea acestui tip de sistem este foarte importantă la aplicațiile destinate protecției, securității și siguranței datelor, fiind des utilizate în scenariile de testare ale autovehiculelor autonome [Bratu & Gușeală, 2019].

Cloud Computing este platforma optimă de analiză pentru învățarea în adâncime, deoarece acest tip de arhitectură oferă elasticitate, poate stoca volume mari de date atât structurate, cât și nestructurate și deține resurse nelimitate.

În vederea optimizării sistemului implementat și implementat la capitolul 5, se va propune la model conceptual utilizarea unei soluții inovative bazate pe un modul de inteligență artificială. Modul este responsabil de monitorizarea preventivă a sistemului și de identificarea posibilelor erori tehnice ce pot apărea în regim de funcționare.

Analiza se realizează pe baza colectării datelor din sistem, prin analiza proceselor de business, analiza log-urilor din sistem și trebuie să alerteze echipele de suport atunci când o valoare prestabilită a fost depășită. Alertarea se face prin crearea unui ticket către grupele de suport, folosind un mesaj standardizat și indicând locația unde sistemul are probleme.

Metoda este cunoscută ca mentenanță preventivă a sistemului și trebuie să alerteze înainte ca eroarea să cauzeze un impact asupra proceselor business, și mai ales înainte ca eroarea să fie resimțită de către utilizatorul final.

Aceste date nestructurate sunt analizate prin intermediul instanței cognitive, oferă funcții de raportare, astfel încât specialiștii din echipele de suport să poată evalua și interveni pro-activ în menținerea unui sistem stabil. Fig. 6.4.1 prezintă sistemul de monitorizare preventivă propus pentru sistemul automat de dezvoltare software.

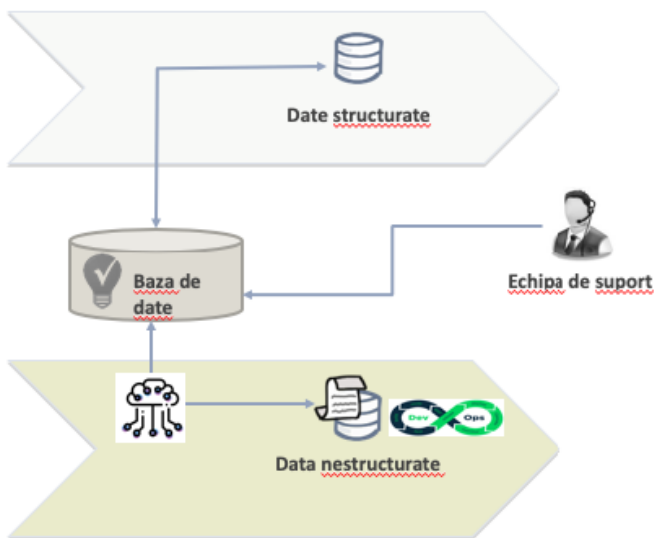


Fig. 6.4.1 Sistem de monitorizare preventivă

Modulul implementat se va integra în sistemul automat DevOps și trebuie să asigure prevenția erorilor de natura tehnica în sistem. Astfel componenta de învățare se concentrează pe evaluarea sistemului în vederea transmiterii de alerte când valoarea prestabilită de control a fost depășită.

Soluția propusă va acționa similar unui sistem inteligent de monitorizare, care va scrie în mod regulat informații într-o bază de date sub forma de date nestructurate din sistem. Sistemul creat va permite monitorizarea proceselor de CI/CD și echipamentele acestuia.

Fig. 6.4.2 prezintă un exemplu de monitorizare și de raportare a incidentelor apărute în sistem într-o perioadă predefinită de timp. Raportul este utilizat de către specialiștii echipelor de suport pentru analiza erorilor tehnice apărute în sistem în vederea soluționării lor, înainte ca acestea să aibă un impact major asupra procesului din organizație.

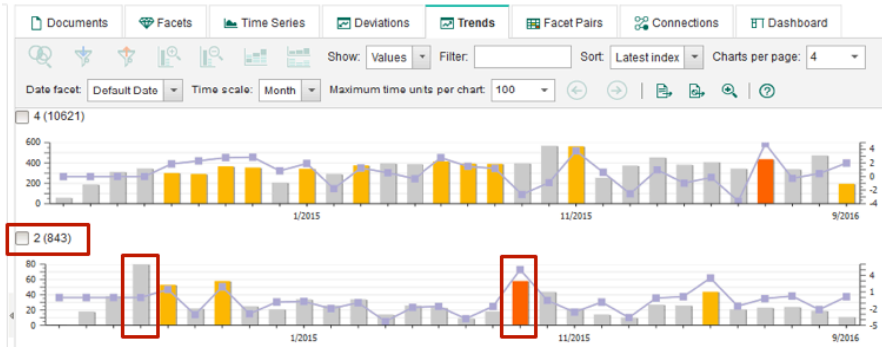


Fig. 6.4.2 Analiza predictivă

6.5 Concluzii

Sistemul automat de livrare software propus la capitolul 4 este extins cu un modul automat de inteligență artificială ce permite monitorizarea sistemului folosind tehnologii moderne. Modulul de învățare stochează informații din sistem, evaluează procesele executate și implementează un mecanism de prevenție și alertare bazat pe informațiile disponibile în sistem. Bazându-se pe învățarea automată, modulul cognitiv realizează predicții cu privire la erorile ce ar putea apărea în sistem.

Elasticitatea sistemelor de tip Cloud Computing este caracteristica de bază care face sistemele distribuite medii optime pentru procesarea volumelor mari de date. Proprietățile și serviciile Cloud Computing susțin procesul de analiză a volumelor mari de date cu care operează mecanismele de învățare automate.

Modelele Cloud sunt implementate cu succes ca resurse de stocare pentru sistemele distribuite și oferă o variantă eficientă din punct de vedere cost, asigurând totodată accesul la tehnologii inovative pentru prelucrarea unui volum mare de date, [Perniu et al, 2019].

Utilizarea serviciilor Cloud Computing pentru implementări de analiză de date scade necesitatea de a procesa volume mari de date în centrele de date private și încurajează migrarea datelor și procesarea lor în Cloud cu o mai mare ușurință.

Combinăția dintre programele software de analiză avansată și disponibilitatea puterii de procesare la cerere fac din Cloud mediul ideal pentru execuția și analiza volumelor mari de date folosite în procesul de învățare automat. Sistemele de inteligență artificială pot utiliza puterea de calcul distribuită disponibilă în Cloud Computing pentru execuția de procese complexe folosind.

Prin utilizarea celor două tehnologii, cea de învățare automată și Cloud Computing, organizațiile sunt ajutate în demersurile lor de adoptare a tehnologiilor inovative pentru împlinirea obiectivelor strategice de transformare digitală.

7. Concluzii generale, realizări și contribuții originale, direcții viitoare de cercetare și diseminare

7.1 Concluzii generale

Cloud Computing vine ca un răspuns bazat pe tehnologie la necesitatea companiilor de astăzi de a accesa putere de calcul crescută și de a procesa volume mari de date. Întrucât modelul tradițional de operare a serviciilor IT prin accesarea centrelor de date existente în cadrul companiei, licențierea și mentenanța hardware și software implică costuri ridicate pentru companii, serviciile Cloud reprezintă mijlocul ideal pentru organizații de a lansa servicii IT de calitate pe piață. Cloud Computing oferă acces la resursele IT sub forma de servicii, asigură disponibilitatea la cerere a serviciilor, și plata acestora în funcție de consum.

Recent s-a observat că din ce în ce mai multe organizații încearcă implementarea, dezvoltarea dar și integrarea proceselor de business accesând resurse Cloud în vederea optimizării cheltuielilor. Organizațiile de astăzi trebuie să pună la dispoziție servicii de înaltă calitate și trebuie să mențină pasul cu noile tendințe tehnologice specifice transformării digitale.

Cloud Computing devine astfel unul dintre cele mai importante mijloace tehnologice de transformare a organizațiilor în vederea optimizării proceselor de business și este folosit în vederea îndeplinirii obiectivelor strategice de business. Cloud aduce o suită de servicii inovative, de înaltă calitate și disponibile, accesibile imediat utilizatorului final. Astfel, furnizorii de servicii Cloud Computing oferă soluții de infrastructură, soluții software și hardware, dar și acces la platforma *middleware*. Componentele și serviciile de tip Cloud, reușesc să optimizeze utilizarea de servicii IT, disponibilitatea și accesul la acestea.

Utilizarea tehnologiilor Cloud ajută la creșterea eficienței serviciilor publice sau private oferite utilizatorilor odată cu creșterea eficienței operaționale și optimizarea serviciului de livrare de resurse computaționale. Unul dintre obiectivele majore ale acestei lucrări este de a aduce o referință practică în adoptarea unei strategii “Cloud Computing” pentru soluționarea provocărilor din domeniul informatic.

Întrucât, implementarea de tehnologii Cloud Computing reprezintă numeroase avantaje pentru organizații, este necesară o bună înțelegere a Cloud Computing-ului în vederea implementării și utilizării serviciilor acestuia. De asemenea este necesară asimilarea de cunoștințe cu privire la nivelele care stau la baza implementării Cloud Computing (IaaS, PaaS și SaaS), caracteristicile Cloud, avantajele dar și provocările unei astfel de implementări. Tehnologiile inovative folosite de Cloud asigură implementarea unui model de succes, ele însă trebuie evaluate, sintetizate și prezentate în lucrări de specialitate.

Prin intermediul acestei teze s-a dorit identificarea conceptelor, care stau la baza sporirii agilității organizațiilor prin intermediul utilizării produselor IT de ultimă generație. Pentru aceasta s-a analizat literatura de specialitate, s-a identificat și s-a implementat soluția optimă de creare a unui sistem flexibil, configurabil și care asigură o performanță ridicată. Sistemul propus este adaptat la standardele tehnologice impuse de Cloud Computing.

În lucrare s-a abordat și s-a prezentat un model arhitectură pentru soluționarea și implementarea aplicațiilor software de tip Cloud. Astfel, cercetarea prezintă un model de proiectare în vederea dezvoltării de soluții software de tip Cloud durabile și fiabile.

S-au identificat și s-au caracterizat diferite modele de arhitectură pentru utilizarea de servicii Cloud și s-a propus soluții apelând la bunele practici de proiectare în vederea dezvoltării aplicațiilor software de tip Cloud.

Modelul de arhitectură propus poate fi văzut ca un catalog arhitectural pentru implementarea și dezvoltarea soluțiilor software Cloud. Pentru elaborarea soluției tehnice s-

au analizat avantajele și dezavantajele diferitelor modele de design al serviciilor software în funcție de tipul lor de implementare. Scopul creării acestui catalog arhitectural este de a veni în sprijinul specialiștilor IT în identificarea de modele de soluții optime pentru integrarea de servicii de tip Cloud Computing în centrele de procesare a informațiilor.

Întrucât, majoritatea proceselor de business încă se bazează pe aplicații tradiționale sau sunt adeseori aplicații învechite din punct de vedere tehnologic, s-a propus o strategie de evaluare a acestor aplicații în vederea migrării în Cloud. Strategia descrisă oferă posibilitatea oricărei organizații să înceapă transformarea digitală prin implementarea de tehnologii de tip Cloud Computing.

A fost descris un model de referință care ajută la implementarea și integrarea de servicii Cloud în centrele de date. A fost prezentată o abordare inițială a cadrului de analiză de modernizare a aplicațiilor tradiționale cu scopul de a măsura impactul unei migrări în Cloud.

Identificarea și evaluarea aplicațiilor existente, care stau la baza proceselor de business, a fost realizată în mai mulți pași. Pentru început s-a arătat și implementat arhitectura și componentele unei aplicații native Cloud, ulterior s-au evidențiat caracteristicile unei aplicații de tip *legacy* și s-au prezentat constrângerile celei din urmă care nu permit funcționarea ei în Cloud. Mai apoi, s-a arătat strategia de adoptare Cloud Computing pentru organizație, urmată de metoda de evaluare și de standardizarea a evaluării rezultatelor. S-a dezvoltat un instrument, care verifică maturitatea aplicațiilor, caracteristicile ei și ariile de optimizare în vederea integrării aplicației în Cloud. Acest mecanism de evaluare poate fi utilizat indiferent de tehnologia de programare care a stat la baza codării aplicațiilor.

S-a implementat o soluție software de evaluare a maturității aplicațiilor în cadrul studiului de caz, s-a testat și s-a validat funcționalitatea acesteia prin evaluarea unei suite de aplicații și s-au identificat zonele unde trebuie intervenit la nivel tehnologic, de proces sau de business pentru a putea migra aplicațiile evaluate în Cloud. Aplicația dezvoltată poate fi folosită ca un standard în migrarea aplicațiilor existente în Cloud.

Din ce în ce mai multe companii devin foarte dependente de software și de procesul de dezvoltare al serviciilor IT, pentru a putea crea și oferi valoare adăugată produselor create pentru utilizatorul final. Introducerea cu succes a unor mecanisme automate în procesul de dezvoltare software, ca răspuns la schimbările permanente ale cerințelor de business sau ale direcției strategice dintr-o organizație pot avea un impact important asupra livrării de produse și servicii IT. De aceea, în aceasta teză s-a introdus o soluție automată, care susține livrarea continuă de aplicații sau funcționalități noi în Cloud. S-a introdus astfel, ciclul de viață al aplicațiilor, s-au evidențiat diferențele metode și metodologii de dezvoltare și s-a concluzionat că pentru mișcarea strategică de adoptare Cloud, DevOps reprezintă metodologia fundamentală de dezvoltare și implementare a aplicațiilor software în Cloud.

În general, utilizarea Cloud Computing reprezintă soluția tehnologică principală, care ajută companiile să realizeze transformarea digitală. În același timp Cloud permite și altor tehnologii, cum ar fi inteligența artificială, analiza volumelor mari de date, automatizarea și utilizarea principiilor de dezvoltare agile ca software-ul să fie mai repede produs și pus la dispoziție utilizatorului final.

În vreme ce Cloud facilitează accesul la aplicațiile software prin intermediul rețelelor de Internet, dezvoltarea agilă și DevOps, ca și metodologii, ajută inginerii IT să îmbunătățească și să automatizeze gestionarea ciclului de viață al unui produs software.

În consecință, lucrarea a arătat procesul, metodele și instrumentele necesare introducerii automatizării pentru dezvoltarea continuă de aplicații software, de implementare și de testare automată în dezvoltarea de aplicații și funcționalități noi dedicate mediilor Cloud.

A fost propusă o arhitectură de referință pentru implementarea automată de software în Cloud și pentru testarea continuă. A fost implementat sistemul de integrare și de livrare continuă pe baza unor utilitare de tip *open-source*, s-au testat și s-au arătat avantajele utilizării acestora în integrarea serviciilor software în Cloud.

Platforma pentru CI/CD introdusă împreună cu modulele implementate pentru testarea automată se aliniază îndeaproape cu practicile și metodologia de dezvoltare agilă. Abordarea prezentată pentru testarea continuă furnizează o viziune clară asupra calității software-ului livrat și susține livrarea de produse software robuste, stabile și de înaltă calitate către client. Mai mult decât atât, soluția implementată poate fi adoptată în orice mediu de tip Cloud, fie că acesta este de tip public, hibrid sau privat. Avantajul soluției propuse este dat de ușurința cu care sistemul poate fi configurat în funcție de necesitățile existente.

În continuare, s-a validat implementarea și s-au identificat metode de optimizare a sistemului propus prin introducerea unui modul de inteligență artificială. Acest modul permite monitorizarea proceselor cu scopul oferirii de informații predictive pentru a asigura stabilitatea și disponibilitatea sistemului. Se utilizează în acest sens tehnologia și capacitatea inteligenței artificiale de a verifica preventiv potențiale erori de programare sau de configurare. Se face apel la inteligența artificială pentru a implementa un mecanism de analiză ce oferă posibilitatea de corecție, în cazul problemelor detectate pe parcursul întregului proces de implementare software, ce presupune intervenția limitată umană.

Lucrarea prezintă conceptele cheie pentru a adopta cele mai bune soluții tehnice utilizând servicii Cloud, plecând de la alegerea componentelor, arhitectura sistemului și aplicațiile software, utilitățile necesare pentru implementarea automată de servicii de tip Cloud Computing în centrele de procesare a informațiilor.

Pentru aceasta s-a verificat, s-a validat și s-a soluționat arhitectura serviciilor software de tip Cloud nativ, s-a identificat necesitatea de a utiliza containere în proiectarea aplicațiilor, de a dezvolta aplicațiile software noi folosind arhitectura bazată pe microservicii sau arhitecturi fără server. Acesta din urmă permite programatorilor să creeze aplicații care sunt executate pe infrastructura generată prin cod programabil. Toate acestea duc la îmbunătățirea procesului de dezvoltare software în Cloud, ajută la optimizarea resurselor și facilitează lansarea în mediile de producție a produselor și serviciilor IT pe piață în timp record.

În lucrare, s-a prezentat strategia, dar și pașii și evaluare de urmat de către organizații în vederea implementării Cloud Computing. S-a arătat că exista posibilitatea de a îmbunătăți aplicațiile *enterprise* existente, astfel încât să nu fie complet retrase din regimul de funcționare, odată cu implementarea Cloud Computing în organizație. Pentru publicarea aplicațiilor s-a implementat un mecanism automat de dezvoltare, de integrare și de publicare a serviciilor software în Cloud. Mecanismul automat propus oferă independență față de furnizorul de platformă, de infrastructură, sau de tipul Cloud utilizat.

Prin toate cele prezentate, avantajele Cloud-ului pot fi extinse la dezvoltarea rapidă de produse software rezistente, ușor gestionabile și scalabile la nivel global, ce vor ajuta organizațiile să își îndeplinească obiectivele de adoptare a transformării digitale.

7.2 Contribuții originale

Obiectivele definite la inițierea cercetării și descrise în capitolul 1 al lucrării au fost îndeplinite în întregime, atât la nivel teoretic, cât și practic.

- S-a realizat o analiză complexă a stadiului actual tehnologic în domeniul IT pentru utilizarea Cloud Computing în centrele de prelucrare a informațiilor. Analiza a avut ca specific

identificarea de modele de implementare a serviciilor în mediile de tip Cloud Computing. S-a arătat tehnologia necesară, dar și caracteristicile și limitările de arhitectură în cazul unei implementări de servicii Cloud. Aici au fost prezentate limitările tehnologice în vederea alcătuirii unui model de arhitectură de referință pentru aplicațiilor software de tip Cloud;

- S-au identificat atât avantajele, cât și provocările și riscurile specifice implementării serviciilor software în Cloud, dar și o serie de măsuri care trebuie luate în calcul în vederea implementării Cloud Computing în centrele de prelucrare a informațiilor.
- S-a realizat o analiză complexă în vederea definirii, evaluării și soluționării unui model de arhitectură care să susțină modelul de livrare de servicii software de tip Cloud Computing;
- S-a luat în calcul necesitatea de interoperabilitate și de comunicare a serviciilor în Cloud, s-au analizat tehnologiile inovative existente pe piață care stau la baza implementării unei arhitecturi pentru dezvoltarea serviciilor software, s-au verificat tehnologiile necesare cum ar fi necesitatea utilizării containerelor la nivel de platformă în Cloud Computing și a componentelor necesare în alcătuirea unui mediu de implementare;
- S-a propus utilizarea tehnologiilor ca programarea infrastructurii, arhitectura bazată pe microservicii și automatizarea integrării serviciilor software în Cloud;
- S-a identificat un model de proiectare arhitectural pentru dezvoltarea și implementarea de aplicații software Cloud;
- S-a prezentat necesitatea de automatizare a livrării serviciilor în mediile de producție Cloud;
- S-a arătat necesitatea de a avea un instrument IT standardizat ce permite evaluarea aplicațiilor existente în vederea migrării acestora în Cloud.
- S-a propus și s-a identificat o serie de măsuri, de nivel strategic, pentru evaluarea maturității aplicațiilor în vederea implementării lor în Cloud;
- S-a realizat un mecanism automat de evaluare a aplicațiilor software existente în vederea migrării acestora în mediile Cloud Computing;
- S-a definit o serie de criterii necesare în evaluarea maturității aplicațiilor pentru migrarea în Cloud;
- S-a prezentat o serie de caracteristici de îndeplinit de către echipele de specialiști în vederea migrării aplicațiilor în Cloud;
- S-a aplicat metoda și soluția propusă în cazul studiului de caz pentru migrarea aplicațiilor unui departament IT cu scopul validării acesteia. S-au arătat și implementat măsurile necesare pentru implementarea aplicațiilor în Cloud;
- Rezultatele studiului de caz au arătat necesitatea folosirii automatizării în vederea integrării serviciilor în centrele de prelucrare a informațiilor de tip Cloud Computing.
- S-a evaluat ciclul de viață al unei aplicații prin evidențierea caracteristicilor celor două metode existente de dezvoltare, cu accentuarea procesului de dezvoltare de cod tradițional și agil;
- S-au identificat limitările producției de software în model tradițional, care duc la o disponibilitate întârziată a noilor funcționalități în mediile de producție și care generează costuri ridicate de mentenanță și operațional pentru organizații;
- S-au analizat și s-au arătat posibilitățile de optimizare ale proceselor de dezvoltare software prin utilizarea metodologiilor DevOps și a automatizării întregului ciclu de dezvoltare software.
- S-a realizat modelarea procesului de gestionare a serviciilor software și s-a arătat mecanismul de dezvoltare software curent;

- S-a implementat o platforma de tip *open-source*, care să permită livrarea automată de cod și testarea aplicației în regim automat;
- S-a prezentat un model de verificare și de testare a codului software produs, care să asigure calitatea aplicațiilor dezvoltate;
- S-a propus și s-a implementat un mecanism pentru introducerea, publicarea și distribuția în orice moment a serviciilor software în Cloud;
- S-a documentat și s-a detaliat o arhitectura propusă pentru implementarea de software în Cloud, s-a implementat un modul de testare automată și s-au descris pașii necesari pentru a configura platforma furnizată pentru a implementa un mecanism de livrare software complet automatizat *end-to-end*;
- A fost prezentat procesul, metodele și instrumentele necesare integrării automate de cod în Cloud folosind integrarea continuă, publicarea și testarea automată în dezvoltarea de aplicații software;
- S-a realizat standardizarea întregului proces de livrare software prin reducerea dependențelor de factorul uman;
- S-au identificat tipurile de scenarii de test și fazele de testare în care testarea automată este realizată folosind conducta automată de furnizare a codului software;
- S-a realizat integrarea cu succes a diferitelor componente și tehnologii pentru a construi platforma de livrare a serviciilor software în Cloud. Platforma pentru CI/CD prezentată, împreună cu modulele implementate pentru testarea automată se aliniază îndeaproape cu practicile și metodologia de dezvoltare agilă. Abordarea prezentată pentru testarea continuă furnizează o viziune clară asupra calității software-ului livrat și susține distribuția de produse software robuste, stabile și de înaltă calitate către client;
- Conducta de implementare software prezentată este o platformă utilă pentru toate aplicațiile existente, fie că acestea sunt aplicații mobile sau web, care rulează în Cloud și susține dispozitivele interconectate în vederea trecerii la următoarea generație de dispozitive folosite în IoT (*Internet-of-Things*);
- Modelul propus a fost prezentat și validat în cadrul celui de-al doilea studiu de caz realizat, care a integrat un număr de peste 100 de aplicații ingineresti în mediul Cloud.
- S-a realizat analiza modelului implementat și a capacității acestuia de a satisface necesitățile organizațiilor de dezvoltare continuă, care să susțină procesele business;
- S-au propus metode de optimizare a platformei dezvoltate prin utilizarea modulelor de inteligența artificială și componentelor de învățare automată.

În aceasta lucrare am explorat metode, instrumente și tehnologii din diferite zone de inginerie software și am luat în calcul cerințele de implementare specifice unui mediu Cloud de tip hibrid. Printre acestea menționez arhitectura software necesară, ce stă la baza furnizării serviciilor software, platforma hibridă de dezvoltare software și mecanismele de automatizare cu scopul de a contribui la definirea unui cadru pentru gestionarea aplicațiilor software în mediile Cloud Computing. Am dezvoltat astfel un model pentru implementarea aplicațiilor Cloud în centrele de prelucrare a informațiilor și am arătat necesitatea de folosire a metodologiei agile pentru dezvoltarea aplicațiilor software. Toate acestea prelungesc ciclul de viață al unui produs software în cadrul organizației și reușesc să ofere flexibilitate și agilitate serviciilor software de tip Cloud integrate în centrele de procesare a informațiilor.

În final, am arătat că metodele prezentate au fost validate și testate derulând o serie de experimente și studii de caz pentru diverse tipuri de aplicații software.

7.3 Direcții viitoare de cercetare

Lucrarea furnizează un fundament în integrarea serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor cu scopul de a oferi flexibilitate și agilitate întregului proces de livrare software în mediile Cloud. Prin intermediul tehnologiilor inovative utilizate, a arhitecturii software bazată pe micro-servicii și prin intermediul mecanismelor de automatizare prezentate la nivel de platformă hibridă obiectivele acestei lucrări au fost îndeplinite cu succes.

Pe parcursul cercetărilor efectuate și ca urmare a analizei rezultatelor prezentate, am identificat o serie de noi direcții de studiu, care pot constitui subiectul unor cercetări viitoare:

- Optimizarea procesului automatizat de livrare continuă a aplicațiilor software în Cloud prin folosirea unor mecanisme de inteligență artificială, care permit auto-vindecarea la întâmpinarea erorilor tehnice în procesul de livrare. Aceasta permite reducerea intervenției umane la zero;
- Elaborarea unei soluții pentru generarea de medii configurate și personalizate utilizând inteligența artificială;
- Utilizarea tehnologiei și capacitatea inteligenței artificiale de a rula testări automate în procesul de dezvoltare software;
- Extinderea platformei de livrare continuă a aplicațiilor software în Cloud și utilizarea acesteia pentru studiul de interoperabilitate a mai multor tipuri de implementări Cloud (de ex. Azure, Amazon, etc.). Astfel, am în vedere dezvoltarea platformei propuse cu un modul ajustabil, care poate decide pe care tip de platformă Cloud are loc publicarea codului software în funcție de specificații.

7.4 Lista publicațiilor

Diseminarea rezultatelor cercetării doctorale

Rezultatele activității de cercetare desfășurate pe întregul parcurs al studiilor doctorale au fost diseminate, validate și prezentate, atât prin participări la diferite conferințe internaționale de specialitate, cât și prin publicarea de articole în reviste de specialitate naționale și internaționale. Dintre acestea, o serie de lucrări au fost publicate ca prim autor sau coautor. De asemenea, prin participarea la o serie de workshop-uri de specialitate am realizat și prezentat rezultatele din perioada de pregătire doctorală.

Dintre acestea enumăr:

- Articole în volumele unor conferințe internaționale IEEE:
 1. **Gușeală, L.G.**, Moraru, S.A., Bratu, D., „Continuous Testing in the Development of IoT Applications” in International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019, IEEE Instrumentation & Measurement Society, Lisbon, Portugal, 2019;
 2. **Gușeală, L.G.**, Moraru, S.A., Bratu, D., „DevOps Transformation for Multi-Cloud IoT Applications” in International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019, IEEE Instrumentation & Measurement Society, Lisbon, Portugal, 2019;
 3. Bratu, D., **Gușeală, L.G.**, Moraru, S.A., „A Performance Comparison between Deep Learning Network și Haar Cascade on an IoT device” in International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019, IEEE Instrumentation & Measurement Society, Lisbon, Portugal, 2019;

4. Perniu, L., Mosoi, A.A., Sandu, F., Moraru, S-A., Ungureanu, D.E., Kristaly, D.M., **Gușeală, L.G.**, „Cloud services for an active assisted living platform using wireless sensors and mobile devices” in International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019, IEEE Instrumentation & Measurement Society, Lisbon, Portugal, 2019.

- Participări la workshop-uri și conferințe de specialitate prin sintetizarea și susținerea unor lucrări în cadrul Companiei IBM, atât interne, cât și internaționale, pe teme centrale de interes cum ar fi metodologia DevOps, implementarea tehnologiilor Cloud Computing în centrele de prelucrare a informațiilor, dar și transformarea livrării de servicii dezvoltate în regim tradițional spre proiecte cu livrare de software folosind metodele agile, digitalizarea și automatizarea sistemelor IT, unele dintre acestea fiind menționate mai jos:

1. Elaborarea unei platforme automate pentru dezvoltarea continuă, integrarea și publicarea software a aplicațiilor ingineresti folosind mecanisme avansate de automatizare pentru implementarea în Cloud Computing, München, septembrie 2018;

2. L. Gușeală, "Transformarea și etapele transformării DevOps în domeniul Automotive", München, octombrie 2017;

3. L. Gușeală, „Știința, managementul și ingineria serviciilor în Cloud Computing”, Brașov, februarie 2018;

4. Participarea la evenimentul IT IBM AMS Automation University Event, organizat de către IBM Germania pe tema automatizării și optimizării proceselor de business prin intermediul Inteligenței Artificiale Watson, Roboti UiPath în vederea integrării DevOps în Organizațiile IT, München, aprilie 2018;

5. Premiul Best of IBM, Eveniment IT organizat de către IBM pe tema Digitalizării în IT, Inteligența Artificială, Roboți și Mecanisme de Automatizare în IT, Nassau, februarie 2018;

6. L. Gușeală, "Inițierea în transformarea mediului IT într-un mediu digital plecând de la aplicații, sisteme și produse IT în aplicațiile ingineresti de design (CAD)", Brașov, februarie 2018;

7. Participarea la Workshop-ul IBM "Innovation Shaping Tomorrow's Technology", București, februarie 2018;

8. L. Gușeală, "Think future(@Automotive Industry)", IBM Automation Event Workshop, București, aprilie 2018;

9. Participarea la Workshop-ul IBM: "Modele de Consum pentru Proiecte Agile si DevOps", München, aprilie 2018;

10. Participarea la conferința IBM: "Around the World with Open Artificial Intelligence and Standard Dev. Organizations", Jeffrey Borek, Jennifer Skeivik, Susan Malaika, Kaoutar El Maghraoui, Steve Holbrook, Zhong TIAN, Tony Holland, mai 2018;

11. L. Gușeală, et al., Cercetarea, sintetiza și alcătuirea unui model de concept pentru interogarea datelor, prezentare și arhivare utilizând tehnologii OpenShift în domeniul Automotive, München, iulie 2018;

12. L. Gușeală, „Procesul de Transformare DevOps, automatizarea și optimizarea proceselor de business utilizând Jenkins”, „Testarea automată folosind UiPath RPA”, IT Workshop, Bangalore, mai 2018;

13. L. Gușeală, „Digitalizarea și Abordarea Agile în proiecte IT, modele de colaborare în organizații IT distribuite global”, Bangalore, mai 2018;

14. L. Gușeală, „Definirea unei direcții strategice de modernizare a aplicațiilor ingineresti în vederea transformării DevOps”, München, septembrie 2018;

15. L. Gușeală, „Implementarea unui mecanism automat de evaluarea a maturității aplicațiilor în vederea integrării în Cloud Computing”, München, octombrie 2018;
16. L. Gușeală, „Automatizarea folosind mecanisme ca RPA UIPATH, Inteligența Cognitivă ca mijloace de digitalizare în aplicațiile ingineresti”, München, iulie 2018;
17. L. Gușeală, „Definirea unui Framework DevOps în proiecte de dezvoltare agile folosind metode de Design Thinking”, München, august 2018;
18. L. Gușeală, „Transformarea DevOps pentru aplicațiile ingineresti folosind sisteme avansate de automatizare”, Conferința Automotive IT, München, octombrie 2018;
19. L. Gușeală, Susținerea lucrării „Implementarea mecanismelor automate și optimizarea proceselor în dezvoltarea software folosind platforma HP ALM Octane”, Conferința Automotive IT, München, octombrie 2018;
20. Participarea la evenimentul IT „IBM AMS Automation University Event”, organizat de către IBM pe tema tehnologiilor și sistemelor inovative de testare automate în dezvoltarea aplicațiilor software prin intermediul roboților UiPath si Blueprism, Frankfurt, aprilie 2019.

- Rapoartele științifice elaborate în cadrul programului de doctorat și susținute în cadrul Departamentului Automatică și Tehnologia Informației al Universității Transilvania din Brașov:

1. L. Gușeală, „Tehnologii de creștere a eficienței energetice în centrele de prelucrare a informațiilor”, Brașov, mai 2013;
2. L. Gușeală, „Aplicații pentru managementul ciclului de viață a produselor”, Brașov, septembrie 2013.

7.5 Valorificarea rezultatelor cercetării

O serie de soluții originale sunt valorificate în cadrul proiectului “NOT Alone at Home – NOAH” - Cooperare Europeană și Internațională AAL - Active Assistive Living, și al proiectul PN-III-P2-2.1-PTE-2016-0064 ”CON-INTEL – Măsurarea consolidată și transmiterea parametrilor energetici spre punctele de colectare” - Finanțat de UEFISCDI.

Bibliografie selectivă

1. Alhamazani, K., Ranjan, R., Mitra, K., Rabhi, F., Jayaraman, P., Khan, S., Guabtni, A., Bhatnagar, V., “An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art”, in *Computing*, vol. 97, no. 4, pp. 357-377, 2014;
2. Alkhalil, A., Sahandi, R., John, D., “Migration to Cloud Computing: A Decision Process Model”, in 10.13140/RG.2.1.4382.0569, 2016;
3. Allspaw, J., “10+ Deploys Per Day: Dev and Ops Cooperation at Flickr”, in <http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr/>, 2009;
4. Alonso, J., Orue-Echevarria Arrieta, L., Escalante, M., Gorroñogoitia, J., Presenza, D., “Cloud modernization assessment framework: Analyzing the impact of a potential migration to Cloud”, pp. 64-73, in 10.1109/MESOCA.2013.6632736, 2013;
5. Andrikopoulos, V., Fehling, C., Leymann, F., “Designing for CAP - The Effect of Design Decisions on the CAP Properties of Cloud-native Applications”, 2012
6. Babar, Z., Lapouchnian, A., Yu, E., “Modeling DevOps Deployment Choices Using Process Architecture Design Dimensions” in 10.1007/978-3-319-25897-3_21, pp. 322-337, 2015;
7. Bratu, D., **Gușciă, L.G.**, Moraru, S.A., “A Performance Comparison between Deep Learning Network și Haar Cascade on an IoT device”, in *International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019, IEEE Instrumentation, Measurement Society, Lisbon, Portugal, 2019*;
8. Callanan, M., Spillane, A. “DevOps: Making It Easy to Do the Right Thing”, in *IEEE Software*. 33. 1-1. 10.1109/MS.2016.66, 2016;
9. Cappelli, W., Ganguli, S., De Silva, F., “Magic Quadrant for Application Performance”, Gartner, 2018;
10. Cohen, D., Lindvall, M., Costa, P., “Agile software development”, in *DACS SOAR Report*, pp. 11, 2003;
11. Cohn, M., “Succeeding with Agile”, Pearson, 2015;
12. Collins, E., Neto, A., Lucena Jr, V., “Strategies for Agile Software Testing Automation: An Industrial Experience”, pp. 440-445, in 10.1109/COMPSACW.2012.84, 2012;
13. Costello, K., “Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17.3 Percent in 2019”, Gartner, STAMFORD, Conn., 2018;
14. De Silva, F., Haight, C., “Magic Quadrant for Application Performance Monitoring Suites”, in *Gartner Research (ID Number=G00298377)*, 2016;
15. Deb, B., “Assess enterprise applications for cloud migration Using the Analytic Hierarchy Process to evaluate apps for the cloud”, 2010;
16. Debbiche, A., Diener, M., Berntsson Svensson, R., “Challenges When Adopting Continuous Integration: A Case Study”, in *Product-Focused Software Process Improvement, Vol. 8892 of Lecture Notes in Computer Science*, pp. 17–32, Springer International Publishing, 2014;
17. Diagram of the Scrum process management method, Wikimedia Commons, accessed March 15, 2009, http://en.wikipedia.org/wiki/File:Scrum_process.svg.

18. Dragich, L., “APM Conceptual Framework on Prioritizing Gartner's APM Model”, APM Digest, 2012;
19. Edwards, D., “Integrating DevOps Tools into a Service Delivery Platform”, 2012;
20. Elberzhager, F., Arif, T., Naab, M., Süß, I., Koban, S., “From Agile Development to DevOps: Going Towards Faster Releases at High Quality – Experiences from an Industrial Context”, in Proc.9th International Conference on Software Quality. Complexity and Challenges of Software Engineering in Emerging Technologies, Switzerland: Springer International Publishing, pp. 33–44, 2017;
21. Fang, L., Jin, T., Jian, M., “NIST Cloud Computing Reference Architecture”, in Special Publication (NIST SP) - 500-292, 2011;
22. Fehling, C., Leymann, F., Retter, R., Schupeck, W., Arbitter, P., “Cloud Computing Patterns”, Springer, 2014;
23. Feitelson, D., Frachtenberg, E., Beck, K., “Development and deployment at facebook”, in IEEE Internet Computing, vol. 17, no. 4, pp. 8–17, <http://dx.doi.org/10.1109/MIC.2013.25>, 2013;
24. Fitz, T., “Continuous Deployment”, in <http://timothyfitz.com/2009/02/08/continuous-deployment/>, 2009;
25. Garcia-Gomez, S., Escriche-Vicente, M., Arozarena, P., Jiménez-Gañán, M., Lelli, F., Taher, Y., Bíró, J., Momm, C., Spriestersbach, A., Vogel, J., Le Jeune, G., Giessmann, A., Junker, F., Dao, M., Carrie, P., Niemöller, J., Mazmanov, D., “4CaaS: Comprehensive Management of Cloud Services through a PaaS”, in 10.1109/ISPA.2012.72, pp. 494-499, 2012;
26. **Guşeilă, L.G.**, Moraru, S.A., Bratu, D., “Continuous Testing in the Development of IoT Applications”, in International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019, IEEE Instrumentation, Measurement Society, Lisbon, Portugal, 2019;
27. **Guşeilă, L.G.**, Moraru, S.A., Bratu, D., “DevOps Transformation for Multi-Cloud IoT Applications”, in International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019, IEEE Instrumentation, Measurement Society, Lisbon, Portugal, 2019;
28. Hammond, J.S., Rymer, J.R., Mines, C., Heffner, R., Bartoletti, D., Tajima, C., Birrell, R., “How To Capture The Benefits Of Microservice Design”, Forrester Research, Tech. Rep., 2016;
29. Humble, J., Farley, D., “Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation”, 1st Edition, Addison-Wesley Professional, 2010;
30. Khan, A., “Practicing Continuous Integration and Continuous Delivery on AWS Accelerating Software Delivery with DevOps”, 2017;
31. Khan, R., Srivastava, “Agile approach for Software Testing process” pp. 3-6, in 10.1109/SYSMART.2016.7894479, 2016;
32. Kowall, J., Cappelli, W., “Magic Quadrant for Application Performance Monitoring”, Gartner, 2013;
33. Moldovan, D., Copil, G., Truong, H.L., Dustdar, S., “QUELLE – A Framework for Accelerating the Development of Elastic Systems”, in 3rd Service-Oriented and Cloud Computing (ESOCC), Manchester, United Kingdom. pp.93-107, 10.1007/978-3-662-44879-3_7.1318277, 2014;
34. Mukherjee, J., “What is a continuous delivery pipeline?”, Atlassian, in <https://www.atlassian.com/continuous-delivery/pipeline;>

35. Newman, S., "Building microservices: designing fine-grained systems", 2015;
36. Odonell, G., "Cloud Services accelerate your pursuit of customer obsession," in *The Cloud Computing Playbook*, Forrester, 2018;
37. Ogigau-Neamtiu, "Cercetări privind securizarea informației în sistemele Cloud Computing", 2018;
38. Perniu, L., Mosoi, A.A., Sandu, F., Moraru, S.A., Ungureanu, D.E., Kristaly, D.M., **Gușeală, L.G.**, "Cloud services for an active assisted living platform using wireless sensors and mobile devices", in *International Symposium on Sensors and Instrumentation in IoT Era – ISSI 2019*, IEEE Instrumentation, Measurement Society, Lisbon, Portugal, 2019;
39. Petersen, K., Wohlin, C., Baca, D., "The waterfall model in large-scale development", in *International Conference on Product-Focused Software Process Improvement*, pp. 386–400, Springer, 2009;
40. Satzger, B., Hummer, W., Inzinger, C., Leitner, P., Dustdar, S., "Winds of Change: From Vendor Lock-In to the Meta Cloud", in *IEEE Internet, Computing*, vol. 17, no. 1, pp. 69-73, 2013;
41. Stoica, M., Mircea, M., Ghilic-Micu, B., "Software Development: Agile vs. Traditional", in *Informatica Economica* 17, pp. 64-76, 10.12948/issn14531305/17.4.2013.06, 2013;
42. Vijayasathy, L.R., "Agile Software Development: A survey of early adopters", in *Journal of Information Technology Management Volume XIX*, Number 2, 2008;
43. Workineh, M., Garcia, N., Midekso, D., "Cloud Suitability Assessment Method for Application Software" in 10.5220/0006365206260631, pp. 626-631, 2017.

Anexe

Anexa 1. Rezumat/Abstract

Rezumat

În cadrul tezei de doctorat intitulată „Integrarea serviciilor de tip Cloud Computing în centrele de prelucrare a informațiilor” se abordează tematica implementării Cloud Computing la nivel de organizație utilizând cele mai inovative componente și elemente de arhitectură de pe piață. Cloud Computing vine ca o evoluție tehnologică a procesului de industrializare din domeniul IT a ultimilor decenii. Cloud permite eficientizarea proceselor de business prin intermediul tehnologiei, prin standardizarea și automatizarea proceselor existente, ajută la modelarea noilor procese de business și oferă un grad ridicat de disponibilitate și de flexibilitate a serviciilor independent de locația geografică a utilizatorului final. Cloud Computing este văzut ca o componentă esențială în curentul actual de digitalizare a industriilor, permite și susține integrarea de tehnologii inovative, cum ar fi inteligența artificială, Blockchain sau Big Data, ce stau la baza transformării digitale pentru organizații. Deși Cloud Computing se bucură de mare atenție și oferă numeroase avantaje organizațiilor, care utilizează serviciile Cloud. Cu toate acestea, o alegere incorectă a componentelor, a tehnologiilor sau integrarea acestora în mod necorespunzător poate îngreuna, sau chiar compromite transformarea digitală a proceselor de business din organizație. Lucrarea analizează critic posibile modele de implementare Cloud, componentele și tehnologia necesară în implementarea unui model de succes de tip Cloud Computing în centrele de prelucrare a informațiilor. În teză, autorul a propus un model de arhitectură software bazat pe soluții tehnologice noi pentru dezvoltarea aplicațiilor software Cloud native, cât și o metodă de migrare a aplicațiilor existente în Cloud. A fost propus un model automat de implementare a serviciilor software în Cloud, folosind principii de dezvoltare agile și metodologii DevOps. Implementarea propusă este optimizată utilizând module de monitorizare a performanței aplicațiilor software, dar și o componentă cognitivă ce permite soluționarea erorilor din sistem la momentul apariției unei erori în sistem. Rezultatele cercetării au fost verificate și validate prin intermediul studiilor de caz, ceea ce a permis definirea și alcătuirea unei soluții optime de integrare a serviciilor software de tip Cloud Computing în centrele de prelucrare a informațiilor.

Abstract

In the doctoral thesis “Integration of Cloud Computing services in data centers” the author addresses the implementation and integration of Cloud Computing services in the data centers by using ultimate innovative components and architectural elements available on the market. Cloud Computing is seen as a technological evolution of the industrialization process in the IT area over the last years. Cloud enables the optimization of an organization’s business processes using new technologies, using standardization and automation of existing processes, helps in the creation of new business models and offers an increased degree of availability and flexibility of services regardless of the end-user’s geographic location. Cloud Computing is seen as an integral part of the current digitalization trend of industries, enables and supports the use of new technologies like artificial intelligence, Blockchain or Big Data, which are the main pillars of the digital transformation for organizations. Although Cloud Computing is very popular nowadays and is able to provide a huge amount of advantages by using it, it must be understood that an improper implementation and integration of its technological components can slower an organization on its path towards the digital transformation. Therefore, the aim of the study is to critically analyze the possible deployment models of Cloud Computing for an organization, the underlying components and used technology for the successful implementation of a Cloud Computing model. In this paper, the author proposes

an architectural software model based on innovative technologies for the development of Cloud native applications and also a model for the evaluation and migration of existing applications to the Cloud environment. In the thesis an automated model is implemented for the continuous integration of software services to the Cloud, using agile and DevOps principles. The proposed model is enhanced using automated application performance monitoring and cognitive components to ensure the self-healing of applications in case of system errors. The results of this research were tested and validated during the case studies, which allowed the author to define and build an optimized technical solution for the automated integration of Cloud Computing software services in the data centers.

Anexa 2. Curriculum Vitae

INFORMAȚII
PERSONALE

LIGIA GEORGETA GUȘEILĂ



EXPERIENȚA PROFESIONALĂ

Aprilie 2017 – prezent

Delivery Project Executive
IBM, München – Germania
Conducerea și coordonarea echipelor de proiect pentru
livrarea de servicii IT.

Ianuarie 2016 – Martie
2017

Senior Service Delivery Manager
IBM, München – Germania
Managementul livrării de produse și de servicii IT în
conformitate cu obligațiile contractuale.

August 2012 –
Decembrie 2015

Service Delivery Manager
IBM, Brașov - România
Gestionarea livrării serviciilor de aplicații pentru un Cluster
operațional.

EDUCAȚIE ȘI FORMARE

2010 – 2012

Diplomă de masterat, Sisteme Avansate în Automatică și
Tehnologia Informației, Universitatea Transilvania, Brașov

2006 – 2010

Diplomă de licență, Inginerie Electrică și Știința
Calculatoarelor, Universitatea Transilvania, Brașov

2002 - 2006

Colegiul Național Doamna Stanca, Făgăraș
Profil: Matematică-Informatică cu predare în limba germană

COMPETENȚE PERSONALE

Limbi străine

Engleză – avansat, Germană – avansat

Competențe de comunicare

- Abilități de comunicare și gestionare a proiectelor dovedite pe parcursul anilor.
- Gândire creativă și comunicare eficientă față de clienți și de echipele de proiect.
- Experiență multiculturală dovedită.

Competențe organizaționale/manageriale

- Abilități excelente de Project Management, People Management și IT Service Management, dobândite în decursul a mai mult de 9 ani de activitate în domeniul aplicațiilor și serviciilor software.
- Bun coordonator de echipe, orientat pe rezultate cu abilități bune de analiză a competențelor oamenilor și de a desemna sarcinile în funcție de aptitudinile lor.
- Dețin capacitatea de a lucra sub presiune și de a lua decizii în medi cu factori necunoscuți cu termene scurte de implementare.

INFORMAȚII SUPLIMENTARE

Distincții

Premiul Best of IBM (Mai 2018)

Premiul IBM Eminence and Excellence (Nov 2016)

Premiul IBM Eminence and Excellence Award (Dec 2011)

PERSONAL INFORMATION

LIGIA GEORGETA GUȘEILĂ



WORK EXPERIENCE

April 2017 – Present

Delivery Project Executive

IBM, Munich – Germany

Providing account leadership to the service delivery management team and direct teams to develop project and business strategies in order to meet contractual scope.

January 2016 – March 2017

Senior Service Delivery Manager

IBM, Munich – Germany

Leading the team of service delivery managers (SDMs) and acting as Delivery Project Executive stand-in for the delivery of the engineering application services.

August 2012 –
December 2015

Service Delivery Manager

IBM, Brașov – Romania

Managing the application services delivery for the Geometrical Design Tower

EDUCATION AND TRAINING

2010 – 2012

Master of Science, (M.Sc.), Advanced Control and IT System Engineering

Transilvania University of Brasov

2006 – 2010

Bachelor's degree, (B.Sc.), Electrical Engineering and Computer Science

Transilvania University of Brașov

2002 - 2006

High School, Mathematics and Science – German
Colegiul Național Doamna Stanca, Făgăraș

PERSONAL COMPETENCIES

Foreign Languages

English – advanced; German - advanced

Communication skills

- Creative thinking and effective communication in a multicultural sensitive environment.
-

Organizational / Managerial skills

- Excellent Project Management, People Management and IT Service Management skills acquired throughout
- 9+ years of Application Services experience
- People and Team Management
- Leadership in a multicultural project team environment
- Full Financial Responsibility over revenue, costs & gross profit.

Additional Information

Awards

- Best of IBM (May 2018)
- IBM Eminence and Excellence Award (Nov 2016)
- IBM Eminence and Excellence Award (Dec 2011)