

CONTINUOUS QUALITY ASSURANCE PROCESSES FOR SOFTWARE PRODUCTS

ASIGURAREA ÎN REGIM CONTINUU A CALITĂȚII PRODUSELOR SOFTWARE

SECU Andrei, masterand

Programul: Management Informațional

Academia de Studii Economice din Moldova

Republica Moldova, Chișinău, str. Bănulescu-Bodoni 61, www.ase.md

e-mail autor: andrei.s3cu@gmail.com

Abstract. Nowadays the art of assuring the quality of a software product comes like a mandatory measure in order to go into production. Lack of quality can bring critical impacts on the business the product is representing. Therefore, developers all over the world started to investigate and come up with clear solutions for ensuring the quality standards of the products they develop. It's not a secret that most digital solutions are delivered in an iterative way, starting with the minimum valuable product. Looking for feedback in near stages, allows the business to take correct decisions based on the user's behavior and requests. From a high-level perspective, this approach is great, because the team is focused on building the correct product without spending resources and effort on developing some features that will not bring value to the feature. Even if the approach is attractive from development, management, and business perspective, we cannot say the same for quality assurance aspects, that require modeling and reorganization. Each time the team is delivering a new software product or at least makes some code changes, the quality assurance team has to make sure that the previously implemented features are working perfectly and added one that corresponds to the provided business specifications. This being said, the research analyzes the way quality assurance procedures are performed at different software products and provides an original practical implication, that will allow us to see the benefits and challenges of applying continuous testing during the software development life cycle.

Keywords: quality assurance, quality, standards, continuous testing, testing, automation testing, quality management, levels of quality, automation testing framework.

JEL CLASIFICATION: O30, O31, O33

INTRODUCERE

Situația actuală, fără precedent a economiei internaționale denotă valoare informației ca fiind una dintre cele mai importante resurse existente pe fața globului pământesc. Dacă în cursul timpului erau considerați bogați cei ce dețineau pământ, aur, proprietăți, etc. Astăzi, aprecierea de bogăție revine celor care dețin informații și pot da dovadă de un management iscusit în administrarea lor. Suntem martorii unei evoluții incredibile a domeniului tehnologiilor informaționale, singurul în care după cum spune Rober Kiyosaki [1] ”niște adolescenții care nu au nimic, pot deveni miliardari într-o scurtă perioadă de timp”, iar drept confirmare avem exemplul companiilor Zoom, Club House, Instagram, etc. Analizate din punct de vedere calitativ toate aceste afaceri de succes au câteva lucruri în comun. Principalele dintre ele fiind aspectul tehnologic al afacerilor, având în vedere că toate exemplele enumerate sunt cunoscute de publicul larg mai degrabă în calitate de produse software decât afaceri. Un alt aspect comun elucidat în urma analizei a fost faptul că la baza lor se află un management iscusit al informațiilor deținute de către fondatorii companiilor de succes. La întrebarea de ce ei și nu alții? Obținem un răspuns direct din aceeași sursă scrisă de R. Kyosaky ”deoarece nu contează cantitatea informațiilor cât calitatea acestora prin prisma deținătorului lor”. Așadar, calitate este cea care face diferența între succes și insucces. Abordările față de definiția conceptului de calitate sunt diferite și pe alocuri subiective, cu toate acestea ISO 9000 oferă o definiție extinsă pentru a înțelege nivelul de aplicabilitate și interpretare al calității ce constă într-un set de proceduri ”în raport cu obiectivele calității, pentru satisfacerea necesităților, așteptărilor și cerințelor părților interesate, după caz.”. Prin urmare, pentru fiecare dintre noi calitate poate avea valori și însemnătate diferită, la fel cum pentru

fondatorii afacerilor enumerate anterior, prin calitative am apreciat informațiile utilizate pentru satisfacerea nevoilor de a crește afaceri, câștiga bani.

Având în vedere cele spuse despre conceptul de calitate, cercetarea în cauză urmărește analiza procedurilor de asigurarea a calității continue a produselor software. Pentru a putea oferi o definiție clară a necesității de respectare a standardelor de calitate pentru produsele software trebuie punctăm destinația și scopul acestor produse. Instant, se evidențiază doi actori principali care beneficiază de pe seama produselor software, aceștia sunt prestatorul de servicii (cel care creează, administrează și menține aplicația) și beneficiarul final, care adesea este întâlnit în ipostaza utilizatorului de aplicație software. Ambii actori au motive separate și comune în materie de asigurare a calității, dacă vorbim despre administrator, printr-un nivel admisibil de calitate el se asigură că afacerea nu falimentează, având în vedere evoluția imaginii companiei sale și a veniturilor înregistrate. Mai mult, acesta poate utiliza calitate produselor software propuse pe piață drept un avantaj concurențial prin care să-și consolideze sau chiar mărească numărul de clienți în raport cu concurenții acestuia. În postura de utilizator, calitate produsului are însemna satisfacerea nevoilor fără a suferi din urma utilizării produsului software. Totodată, utilizatorul este cel care prin intermediul calității serviciilor obținute poate aprecia nivelul de deservire oferit de administrator. Analizând aceste necesități și tendințe, putem confirma ceea ce spune Elon Musk în ultima sa carte [2]: ”succesul unei produs depinde în primul rând de calitatea lui”.

Așadar, dinamica evenimentelor și analizei de mai sus, incită pozitiv ideea precum că calitate devine următorul factor important de apreciere al lucrurilor. Iar după cum au demonstrat impactul calității este unul major pentru toți actorii implicați într-un proces. Având în vedere complexitatea produselor software dezvoltate în ziua de astăzi, putem cu certitudine sublinia calitatea drept factorul determinant de succes. Drept exemplu, putem oferi lansarea telefoanelor din gama Note a companiei Samsung, în perioada anului 2017. Din cauza lipsei unui nivel de calitate satisfăcător, bateriile unor telefoane au explodat în timp ce proprietarii acestora le alimentau la sursa de energie electrică. Acest moment a cauzat prejudicii colosale, atât în aspect financiar cât și încredere. Cu siguranță exemple există foarte multe, dar pentru a rămâne consecvenți, cercetarea în cauză va analiza un studiu de caz independent, care constituie un sistem automatiza de administrare a parcarilor cu plată.

Până a trece definitiv la partea aplicativă, este necesar să stabilim contextul problematicii și obiectivele certării, prin prisma cărora vor fi analizate rezultatele.

CONTEXTUL PROBLEMATICII

Pornind de la importanța satisfacerii unui nivel acceptat de calitate al produselor software, se face referire la etapele de dezvoltare a unui produs software în raport cu activitățile necesare pentru asigurarea coeficienților admisibili de calitate. Fiecare produs are particularitățile sale, în dependență de ramura de business pe care o reprezintă, respectiv ar fi greșit să interpretăm un singur model de dezvoltare a produselor software cu scopul de a stabili procedurile de asigurare a calității în cadrul ciclului de viață al acestuia. Se disting următoarele metodologii de construire a unui produs software:

- Agile
- Lean
- Waterfall
- Iterative
- Spiral

Fiecare are un anumit set de avantaje și direcții de întrebuițare. Important de menționat este faptul că procesele de testare sunt distincte pentru fiecare, dar nu și scopul acestor procese, care se rezumă la asigurarea unui nivel de calitate optim acceptat de către toți actorii implicați. Aparent dificil, dar cu ajutorul articolului scris de Rober Half, la data de 24 mai, 2019, putem analiza activitățile de testare ca fiind parte a tuturor activităților de dezvoltare prin prisma fazelor comune ale tuturor metodologiilor. Așadar, distingem următoarele faze comune de dezvoltare a produselor software:

- Analysis – analiza sistemelor deja existente pe piață, formarea viziunii;
- Design – etapa de planificare, schițare și creare a specificațiilor;

- Implementation – faza de scriere a codului cu scopul de a reda funcționalului o structură și formă logică;
- Testing – etapă dedicată exclusiv analizei nivelului de calitate al lucrului deja făcut;
- Evaluation – aprecierea produsului prin prisma utilizării sale.

Aparent, faza de testare este dedicată exclusiv procedurilor de verificare și validare a funcționalului deja implementat. Cu toate acestea, privit prin prisma exemplelor de afaceri oglindite în partea introductivă, Facebook sau Instagram nu reprezintă echipa care ar putea să oprească toate procesele cu scopul de a verifica și valida funcționalul implementat. Asta deoarece se pierde resurse colosale de timp și financiare, programatorii fiind nevoiți să aștepte pentru a putea purcede la alte activități. Mai mult, o dilemă iminentă în acest caz o servește și descoperirea defectelor abia în cea de a treia etapă de dezvoltare, ceea ce efectiv nu e rezonabil pentru echipa de dezvoltare conform principiilor stabilite de către ISTQB (International Software Testing Qualifications Board) care afirmă ”defectele depistate în fază inițială sunt mai ieftin și mai ușor de fixat”.

Luând în considerare cele enunțate, dar și faptul că în condițiile unei evoluții ce depășește ritmurile exponențiale ale tehnologiilor informaționale, majoritatea produselor software necesită constant actualizări, prin care produsul fie este înzestrat cu noi funcționalități pentru a face față concurenței de piață, fie este optimizat pentru a oferi o mai bună experiență utilizatorilor. În toate situațiile și fazele evoluției, termenul de calitate constituie o prioritate, asta deoarece în ritmuri înalte livrare a actualizărilor necesare să nu fie scăpate erori care pot duce direct la falimentarea business-ului sau în cel mai rău caz – pierderi umane. Necesitatea implicării procedurilor de asigurare a calității la fiecare fază de dezvoltare al produsului software este privită drept unica oportunitate de menținere a ritmului înalt de livrare. Cu toate acestea, atât timp cât procedurile de testare sunt executate în mod manual, acestea sunt privite drept un mod de a reține viteza de dezvoltare la ritmuri inadmisibile. Respectiv, implicarea procedurilor de testare în incinta fazelor de dezvoltare este numită asigurare a calității continuă, dar acesta are un efect relativ redus atât timp cât nivelul de automatizare al sarcinilor este privit sceptic.

Din acest motiv, companiile mari, care posedă informații calitative și tind să construiască produse menite să schimbe lumea sunt interesate să asigure prin intermediul automatizării un nivel acceptat de performanță și funcționalitate al produselor și serviciilor lor.

OBIECTIVELE CERCETĂRII

Scopul și obiectivele lucrării rezultă din provocările întâlnite de echipele de dezvoltare a produselor software de îndată ce încearcă să asigure un nivel optim de calitate ținând cont de ritmul accelerat de livrare în mediul de producție.

Scopul principal este determinarea mijloacelor aplicative de optimizare a resurselor alocate procedurilor de asigurare a calității prin a face față ritmului rapid de dezvoltare și menținere a unui nivel constant și satisfăcător de calitate.

Pornind de la contextul problematicii, dar și de la scopul major al lucrării, obiectivele primordiale ale cercetării sunt:

- Analiza metodologiei de lucru în ceea ce privește dezvoltarea unui proiect ce necesită actualizării frecvente;
- Optimizarea resurselor destinate satisfacerii nivelului necesar de calitate al produsului software;
- Minimizarea timpului de livrare al actualizărilor prin prisma proceselor de asigurare a calității;
- Crearea unui model/sistem aplicativ care să servească drept suport pentru îndeplinirea activităților de asigurare a calității în regim automatizat.

Cumulativ, atingerea obiectivelor de mai sus vor facilita acoperirea scopului printr-o analiză cantitativă și calitativă a rezultatelor analizate până la abordarea soluției practice oferite de către lucrarea în cauză și rezultatele obținute după. Drept urmare, evaluarea actualității cercetării în cauză va fi efectuată pe baza rezultatelor colectate.

STUDIUL DE CAZ

Compania X, localizată pe teritoriul Regatului Unit – Marea Britanie, a fost fondată în anul 2000 și este specializată în prestarea serviciilor de parcare automatizată. Serviciile oferite de către aceasta vizează direct folosirea resurselor digitale, cu scopul de a captura imagini cu mașinile care intră, iar mai apoi părăsesc zonele predestinate parcarilor de vehicule. Într-un final, scopul principal al softului este de a procesa informațiilor și a emite o decizie, urmată de o factură de plată către proprietarul vehiculului, în cazul în care acesta a încălcat normele prestabilite de zona de parcare.

Primul produs software, cu ajutorul cărei echipa companiei X a pornit la drum a fost dezvoltat în aproximativ 2 ani de zile, ca mai apoi în decursul timpului aceasta să vină pe piață cu alte 16 versiuni actualizate în aproximativ 18 ani. În mediu, compania a lucrat câte aproximativ un an de zile pentru fiecare actualizare a produsului. Important de menționat este faptul că evaluând timpul necesar pentru lansarea unei actualizări în producție, acesta mereu a înregistrat o creștere. Lucru ușor și obiectiv explicat de către CEO al companiei, care spune: ”Odată cu creșterea funcționalului, pentru a merge în producție cu o versiune nouă, echipa noastră trebuie să se asigure că precedentul funcțional funcționează conform standardelor în vigoare prestabilite de către politica companiei, iar funcționalul nou adăugat se comportă exact așa cum a fost planificat. Aceste volume sunt mereu în creștere, motiv pentru care și resursele alocate sunt mereu revăzute și după necesitate suplinite, pentru a nu suferi extinderi de timp nedorite.”. Intenția directorului general de-a respecta nivelul de calitate este una evidentă și justificată, având în vedere că zilnic, compania procesează între 2 și 8 milioane de imagini capturate, iar o eventuală eroare ar putea crea un haos total și pierderi nu mai mici de 1 milion de lire sterline pe zi.

CONSTATĂRI

Analizând modul în care au fost livrate toate cele 16 actualizări ale produsului software propus de către compania X, putem identifica cu ușurință următoarele aspecte predestinate optimizării:

- Dezvoltarea a fost făcută în baza principiului cascada, ceea ce duce la costuri suplimentare și un timp de livrare extins;
- Procedurile de dezvoltare nu erau însoțite de către acțiuni de asigurare a calității, în cele din urmă majoritatea erorilor nu puteau fi identificate în faza primară, drept urmare fixarea lor prevedea cheltuieli de ordin financiar sporite;
- Activitățile de asigurare a calității erau executate exclusiv manual, cu scopul de a face față volumului mare de lucru, compania a triplat numărul resurselor umane prevăzute pentru asigurarea normelor de calitate.

Așadar, aceste fapte fiind elucidate, s-a văzut necesar și oportun crearea produsului de la 0, prin utilizarea abordărilor moderne cu scopul de a face față concurenței acerbe de pe piață. În calitate de decizie prioritară s-a pus necesitatea rescrierii întregului produs software prin utilizarea unui limbaj funcțional de programare, cum este Scala. Mai mult s-a propus expres implicarea procedurilor de verificare și validare la fiecare nivel posibil de dezvoltare, astfel încât chiar și cele mai mici modificări de cod să fie verificate înainte de validare.

Drept rezultat, în urma cercetării și constatărilor de rigoare menționate mai sus, s-a stabilit crearea unui sistem independente care ar putea:

- Asigura calitatea modificărilor executate de către echipa de dezvoltare la cel mai inferior nivel (schimbări mici de cod);
- Verificare funcționalului deja implementat odată ce o nouă funcționalitate este adăugată;
- Asigurarea unui nivel optim de compatibilitate al sistemului de management automatizat al calității cu sistemul dezvoltat, supus testării;
- Luarea deciziilor de validare fără implicarea resurselor umane.

PARTEA APLICATIVĂ

Acestea fapte fiind elucidate, s-a văzut necesar și oportun crearea produsului de la 0, prin utilizarea abordărilor moderne ale metodologiei Agile ce permite dezvoltarea iterativă a proiectului de creare a produsului software prin implicarea procedurilor de testare la nivelul tuturor fazelor de dezvoltare. Ca răspuns al solicitărilor efectuate de către constatările elucidate, sistemul automatizat de testare al produsului software este creat pe baza limbajului de programare Java și are o arhitectură schematică similară modelului din figura 1. Prin utilizare tehnologiilor avansate de automatizare a procedurilor de testare, cum este Cucumber, s-a reușit scrierea tuturor testelor manuale în format automatizat. Mai mult, biblioteca tehnologiei GERKIN a oferit posibilitatea interpretării testelor automatizate într-un model de business, astfel încât la baza oricărui test este formula de bază (GIVEN-WHEN-THEN), prin care se poate exprima orice acțiune, pornind de la ce avem, ce facem și ce dorim să obținem.

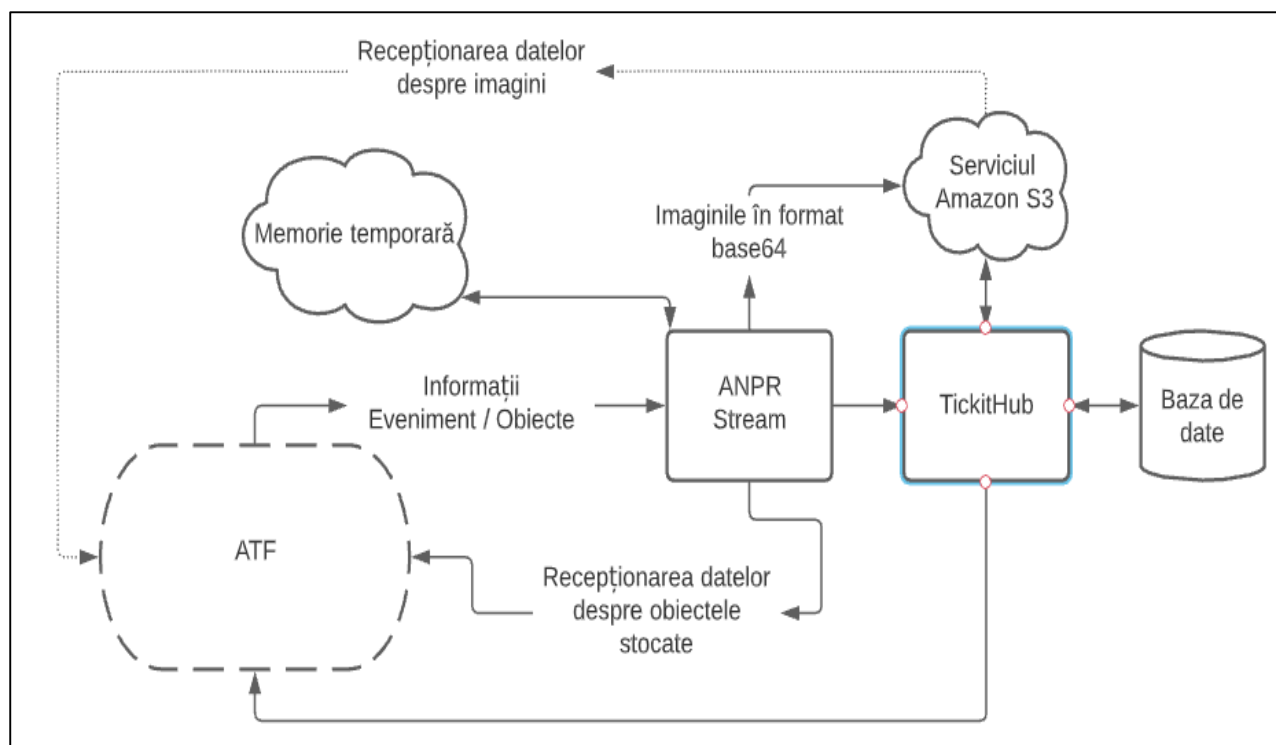


Figura 1. Arhitectura generală a sistemului de gestionare al calității automatizată destinat entității economice

În cele din urmă, după cum se poate observa în figura 1, sistemul independent de asigurare a calității are menirea de a replica activitatea mașinilor din parcare prin furnizarea de informații cu privire la intrarea și ieșire mașinilor, iar mai apoi prin conexiunea cu aplicația de bază, poate verifica veridicitatea și validitatea funcțiilor de bază.

În momentul scrierii acestui raport, au trecut 8 luni de zile de la începutul cercetării, în care cu ajutorul a două resurse umane au fost puse în practică constatările elucidate și create scenarii automatizate de testare pentru întreg funcționalul existent. Astăzi, odată ce un membru al echipei face o modificare a codului, în rezultatul căreia cel puțin unul din cele 568 de scenarii de testare ar cădea, schimbarea acestuia va fi refuzată, el fiind nevoit să cizeleze codul astfel încât scenariul care nu a trecut să fie etichetat cu statutul de succes.

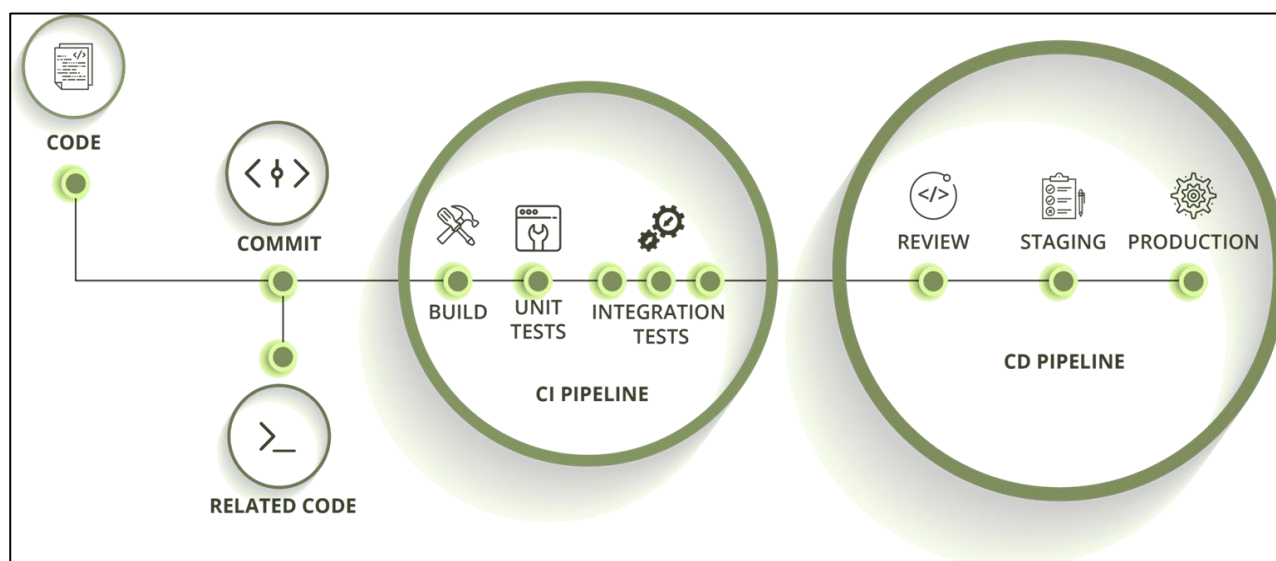


Figura 2. Schema generală de interconectare a sistemului automatizat de asigurare a calității cu conductele de integrare continuă

În aspect general, procedurile cap-coadă au loc conform schemei reflectate în figura 2, unde datorită intercalării sistemului automatizat de asigurare a calității cu conducta de integrare a codului deciziile de validare sunt luate în regim automatizat. Sumar luate, toate acțiunile ilustrate mai sus ne permit obținerea unor rezultate care pot fi supuse analizei comparative, după cum urmează în secțiunea următoare – rezultatele.

REZULTATE

Având în vedere constatările efectuate și partea aplicativă care a furnizat informații despre evoluția performanței echipei până la implementarea testării continue automatizate și după, pot fi evidențiate următoarele rezultate:

Tabelul 1. Analiza comparativă a rezultatelor obținute în urma cercetării: până la implementarea sistemului automatizat de testare continuă și după

	Defecte depistate în faza primară	Timpu necesar pentru livrarea primei versiuni a softului	Timpu necesar rulării scenariilor de verificare și validare	Defecte depistate în producție
Până	Aproximativ 10%	2 ani	2-3 săptămâni	Aproximativ 35%
După	Aproximativ 85%	5 luni	Până la 10-15 minute	Aproximativ 5%

După cum și s-a menționat în faza de analiză a sistemului existent, precum și a oportunităților existente, rezultatele înregistrate în urma implementării soluției aplicative a cercetării s-au soldat cu un succes impresionant, care poate fi evaluat vizual fără dificultate. Eficiența sistemului, se mai caracterizează și printr-o frecvență sporită de actualizări ale produsului software în mediul de producție, având în vedere că în mai puțin de un an au fost livrate deja 4 actualizări.

CONCLUZII

Abordarea calității la nivelul superior constituie una din prioritățile afacerilor de succes. Odată ce companiile operează cu norme de calitate sporite și tehnologiile respective de asigurare a standardelor necesare, valoarea returnată este incomparabilă cu pierderile care pot fi suportate din cauza unui defect în mediul de producție. După cum a fost demonstrat, automatizarea testării continue sporește randamentul eficacității echipelor de dezvoltare și permite afacerilor să fie competitive pe piața acerbă dictată de evoluția fascinantă a tehnologiilor informaționale. Important de menționat este că, odată puse la punct din aspectul organizatoric, abordarea de testarea continuă aduce o returnare de investiții

în permanentă creștere, întrucât necesitatea și importanța acestor proceduri aplicative sporește odată cu creșterea și dezvoltarea produsului software.

REFERINȚE BIBLIOGRAFICE:

1. ROBERT KIYOSAKI, *Rich Dad's Increase Your Financial IQ: Get Smarter with Your Money* (2008). ISBN 0-446-50936-1SA 540
 2. ASHLEE VANCE, ELON MUSK: *Tesla, SpaceX, and the Quest for a Fantastic Future* (2015). ISBN 9780062301253
 3. NICOLE FORGREN, JEZ HUMBLE, GENE KIM, *Accelerate Building and Scaling High Performing Technology Organizations*, It Revolution Press, 2018, 257 pp. ISBN: 1942788339
 4. SURESH CHANDRA SATAPATHY et al. *Automated Software Engineering: A Deep Learning-Based Approach*, Springer, 1st ed., 2021, 132 pp. ISBN: 3030380084202
-

Coordonator științific: ZGUREANU Aureliu, dr., conf. univ.

Academia de Studii Economice din Moldova,
Republica Moldova, or. Chișinău, str. Mitropolit Bănulescu-Bodoni, 61 www.ase.md
e-mail: zgureanu.aureliu@ase.md