**Ministerul Educației și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**RAPORT**

Sarcina individuală

la cursul „Testarea Produselor Program”

Tema: „Integrarea Continuă”

Student : TI-214 Reguș Ruslan

Coordonator: asist. univ. Crîjanovschi Adriana

**Chișinău 2024**

Introducere

În lumea dinamică a dezvoltării software, eficiența și calitatea sunt elemente fundamentale pentru succesul unui produs. În acest context, integrarea continuă (CI) a devenit o practică esențială în ciclul de viață al dezvoltării software. CI este o metodă prin care spațiile de lucru ale dezvoltatorilor sunt unificate într-un depozit comun de mai multe ori pe zi. Scopul său principal este de a preveni și de a gestiona provocările asociate cu integrarea codului, cunoscute sub numele de „integration hell”, care pot apărea în proiectele de dimensiuni mari sau în echipele distribuite.

Această introducere oferă o privire de ansamblu asupra conceptului de integrare continuă și a relevanței sale în domeniul testării produselor software. În continuare, vom explora în detaliu beneficiile integrării continue, procesul de implementare a acesteia în ciclul de dezvoltare a software-ului, instrumentele și tehnologiile utilizate, precum și provocările și soluțiile asociate cu implementarea acestei practici în testarea produselor software.

Prin intermediul acestui raport, ne propunem să oferim o înțelegere profundă a integrării continue și să evidențiem importanța sa în asigurarea calității și eficienței în procesul de dezvoltare a produselor software.

Vom explora beneficiile pe care le aduce integrarea continuă în procesul de testare a produselor software, arătând cum această practică poate contribui la îmbunătățirea calității, la reducerea riscurilor și la accelerarea ciclului de dezvoltare.

Unul dintre avantajele majore ale integrării continue în testarea produselor software este capacitatea sa de a identifica și de a remedia problemele de integrare într-un stadiu incipient al ciclului de dezvoltare. Prin integrarea frecventă a modificărilor într-un depozit comun și prin rularea automată a testelor, echipele pot detecta și rezolva conflictele și erorile de cod înainte ca acestea să devină probleme majore. Acest lucru conduce la reducerea timpului și efortului necesar pentru remedierea problemelor, evitând astfel întârzierile și costurile suplimentare asociate cu erorile descoperite tardiv în procesul de dezvoltare.

De asemenea, integrarea continuă facilitează o abordare mai agilă și mai iterativă în testarea produselor software. Prin implementarea testelor automate în cadrul procesului de integrare, echipele pot obține un feedback rapid asupra calității modificărilor aduse codului, permițându-le să identifice și să remedieze problemele într-un mod eficient și reactiv. Acest ciclu rapid de feedback sprijină dezvoltarea incrementală și îmbunătățește colaborarea între membrii echipei, contribuind la accelerarea ritmului de livrare a funcționalităților noi și la îmbunătățirea experienței utilizatorilor.

Conceptul de Integrare Continuă

Integrarea continuă (CI) este o practică esențială în domeniul dezvoltării software, care constă în unificarea spațiilor de lucru ale dezvoltatorilor într-un depozit comun de mai multe ori pe zi. În esență, CI implică integrarea automatizată și frecventă a modificărilor aduse codului într-o linie principală de dezvoltare, urmată de construirea și testarea automată a întregului sistem. Această abordare permite echipei să detecteze și să remedieze prompt orice probleme de integrare sau erori de cod, asigurând o evoluție fluidă a proiectului.

Principiile de bază ale integrării continue sunt următoarele:

Integrare Frecventă: Dezvoltatorii încarcă și integrează modificările în repozitoriul comun de cod de mai multe ori pe zi. Această practică asigură actualizarea constantă a codului și prevenirea acumulării de modificări mari, care ar putea duce la conflicte și probleme de integrare.

Cuprins

Level1 1

Level1 2

Level1 3

Level1 4

Level1 5

Level1 6

Introducere

În lumea dinamică a dezvoltării software, eficiența și calitatea sunt elemente fundamentale pentru succesul unui produs. În acest context, integrarea continuă (CI) a devenit o practică esențială în ciclul de viață al dezvoltării software. CI este o metodă prin care spațiile de lucru ale dezvoltatorilor sunt unificate într-un depozit comun de mai multe ori pe zi. Scopul său principal este de a preveni și de a gestiona provocările asociate cu integrarea codului, cunoscute sub numele de „integration hell”, care pot apărea în proiectele de dimensiuni mari sau în echipele distribuite.

Această introducere oferă o privire de ansamblu asupra conceptului de integrare continuă și a relevanței sale în domeniul testării produselor software. În continuare, vom explora în detaliu beneficiile integrării continue, procesul de implementare a acesteia în ciclul de dezvoltare a software-ului, instrumentele și tehnologiile utilizate, precum și provocările și soluțiile asociate cu implementarea acestei practici în testarea produselor software.

Prin intermediul acestui raport, ne propunem să oferim o înțelegere profundă a integrării continue și să evidențiem importanța sa în asigurarea calității și eficienței în procesul de dezvoltare a produselor software.

Vom explora beneficiile pe care le aduce integrarea continuă în procesul de testare a produselor software, arătând cum această practică poate contribui la îmbunătățirea calității, la reducerea riscurilor și la accelerarea ciclului de dezvoltare.

Unul dintre avantajele majore ale integrării continue în testarea produselor software este capacitatea sa de a identifica și de a remedia problemele de integrare într-un stadiu incipient al ciclului de dezvoltare. Prin integrarea frecventă a modificărilor într-un depozit comun și prin rularea automată a testelor, echipele pot detecta și rezolva conflictele și erorile de cod înainte ca acestea să devină probleme majore. Acest lucru conduce la reducerea timpului și efortului necesar pentru remedierea problemelor, evitând astfel întârzierile și costurile suplimentare asociate cu erorile descoperite tardiv în procesul de dezvoltare.

De asemenea, integrarea continuă facilitează o abordare mai agilă și mai iterativă în testarea produselor software. Prin implementarea testelor automate în cadrul procesului de integrare, echipele pot obține un feedback rapid asupra calității modificărilor aduse codului, permițându-le să identifice și să remedieze problemele într-un mod eficient și reactiv. Acest ciclu rapid de feedback sprijină dezvoltarea incrementală și îmbunătățește colaborarea între membrii echipei, contribuind la accelerarea ritmului de livrare a funcționalităților noi și la îmbunătățirea experienței utilizatorilor.

Conceptul de Integrare Continuă

Integrarea continuă (CI) este o practică esențială în domeniul dezvoltării software, care constă în unificarea spațiilor de lucru ale dezvoltatorilor într-un depozit comun de mai multe ori pe zi. În esență, CI implică integrarea automatizată și frecventă a modificărilor aduse codului într-o linie principală de dezvoltare, urmată de construirea și testarea automată a întregului sistem. Această abordare permite echipei să detecteze și să remedieze prompt orice probleme de integrare sau erori de cod, asigurând o evoluție fluidă a proiectului.

Principiile de bază ale integrării continue sunt următoarele:

Integrare Frecventă: Dezvoltatorii încarcă și integrează modificările în repozitoriul comun de cod de mai multe ori pe zi. Această practică asigură actualizarea constantă a codului și prevenirea acumulării de modificări mari, care ar putea duce la conflicte și probleme de integrare.

Integrarea continua:

• Integrarea continuă (Continuous integration -

CI) este practica/tehnologia/modelul din ingineria

software de unificare a spațiilor de lucru ale

dezvoltatorilor într-un depozit comun de mai multe

ori pe zi.

• Scopul său principal este cel de a evita

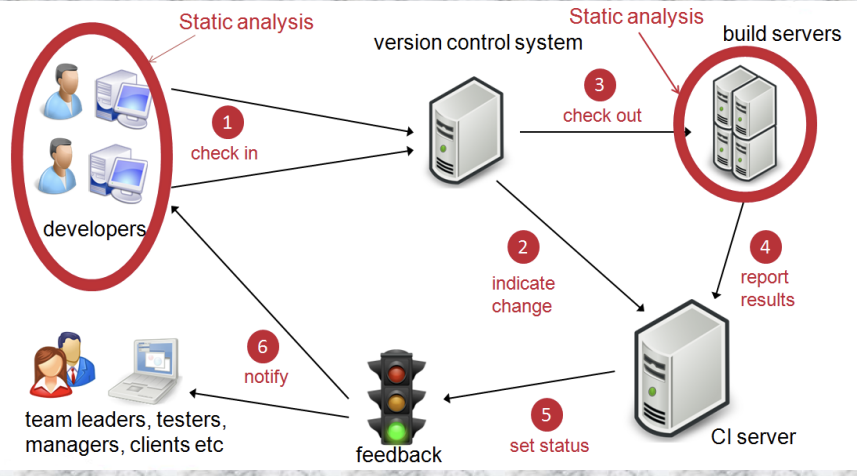
problemele de integrare, denumite „integration

hell”. CI poate fi văzut ca o intensificare a

practicilor de integrare periodică promovate de

metodele deja publicate ale dezvoltării de software

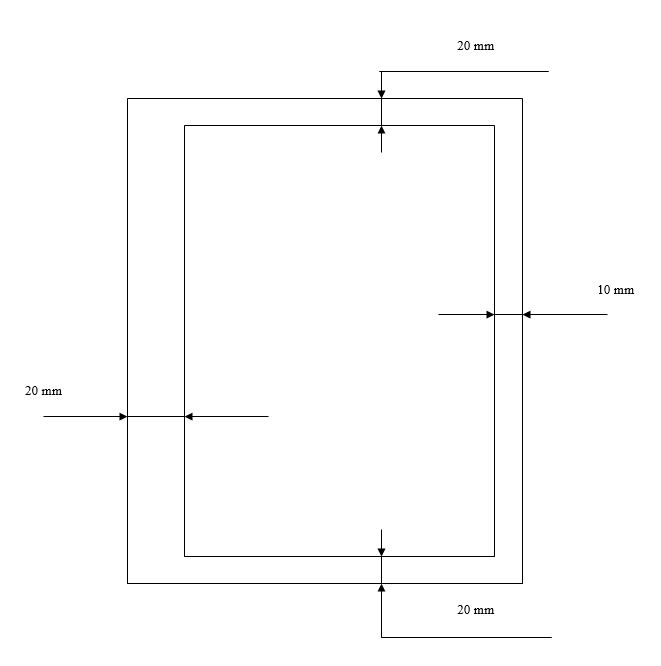
prin proces incremental-iterativ



**Aspecte generale de redactare:**

* lucrarea va fi scrisă în stil impersonal (nu se va utiliza persoana I-a) și se redactează în limba de instruire;
* textul de bază se culege cu fontul Times New Roman, Normal, Justify – 12pt, interval – 1,5, culoarea Fontului – Negru și cu cu următoarele alinieri faţă de margini (fără chenar şi inscripţii): dreapta (Righ) - 10mm; sus (Top) - 20mm; stînga, jos (Left, Buttom) - 20 mm.
* capitolele și subcapitolele se numerotează cu cifre arabe, cu excepția capitolului introductiv și a concluziilor (nu se scrie cuvîntul capitol și fără punct). Exemplu: 1, 2, 3, etc., sau 1.1, 1.2, 1.3, etc… sau 1.1.1, 1.1.2 (se acceptă până la al treilea nivel);
* codul sursă se editează utilizând fontul Courier New cu o dimensiune de 10pt și un interval de 1, dacă codul sursă scris pe pagină ocupă mai mult de 50% din pagină, acesta va fi inclus într-o anexă și se va face referire la el în textul lucrării;
* abrevierile și cuvintele care nu fac parte din limba de instruire, cum ar fi cele în limba engleză, dar sunt utilizate în context, trebuie să fie înscrise în Lista de Abrevieri și Definiții, această listă trebuie să fie plasată înaintea introducerii și inclusă în cuprinsul lucrării;
* fiecare titlu de capitol se scrie din pagina nouă și cu majuscule, cu fontul Times New Roman, Bold, Center – 13pt;
* fiecare titlu de subcapitol se scrie cu fontul Times New Roman, Bold– 12pt aliniere la stânga;
* denumirile de capitole, subcapitole nu se scriu la sfârșitul paginii;
* în cadrul paragrafelor textul va fi aliniat între marginile din stânga și dreapta (Justify);
* primul rând al fiecărui paragraf va avea un alineat de 1,25 cm;
* distanța dintre capitol și text trebuie să fie nu mai mare de 1 interval;
* volumul unui subcapitol trebuie să fie minim de o pagină;
* paginația lucrării va fi realizată jos, la mijloc, fără punct, Foaia de titlu și alte foi până la Introducere sunt incluse în numărul de pagini dar nu se indică paginația;
* este recomandat ca ponderea paginilor cu caracter teoretic să nu depășească 40% din totalul paginilor;
* tipărirea se va face pe format A4, doar pe prima pagină (față) a fiecărei foi pentru teza de licență/master, pentru alte tipuri de lucrări se acceptă și tipărirea pe ambele părți;
* lucrarea se va lega prin spiralare, într-un singur exemplar pentru teza de licență, lucarea de practica și de lucrarea de an la dorință copertați, lucrarea de laborator nu se copertează.

Se va folosi un limbaj simplu și clar. Informațiile vor fi comunicate într-o manieră directă și inteligibilă, într-o structurare logică și coerentă. Lucrarea trebuie să reflecte integral munca autorului. Sursele bibliografice utilizate vor fi menționate în lista bibliografică și la acestea vor fi făcute referințe în text.



**ENUMERARE ÎN CONTEXT**

Dacă în text sunt **enumerări simple** acestea se numerotează prin liniuță, se începe cu litera mică, punct și virgulă la sfârșit după ultima enumerare punct.

*Exemplu:*

- unu;

- doi;

- trei.

Dacă în text la enumerări se face referință, atunci acestea se numerotează prin litere latine (minuscule) cu paranteză.

Dacă în text sunt **enumerări multiple**, acestea se numerotează prin litere latine (minuscule) cu paranteză (dacă enumerările conţin enumerări, acestea se numerotează prin cifre arabe cu paranteză).

*Exemplu:*

а) \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

b) \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

* \_\_\_\_\_\_\_\_\_\_\_;
* \_\_\_\_\_\_\_\_\_\_\_;

2) \_\_\_\_\_\_\_\_\_\_\_\_\_\_;

c) \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Nu se admit alte semne.**

***În enumerare nu pot fi descrieri de mai mult de o propoziție.***

**FIGURI**

Figurile se numerotează în ordine crescătoare pentru fiecare capitol: prima cifră va indica numărul capitolului, iar a doua – numărul de ordine al elementului în respectivul capitol ca “Figura 1.1”.

Figura și Cuvântul “Figura 1.1” și denumirea ei sunt amplasate pe centru, fără punct la sfârșit, se editează cu bold. Dacă imaginea ocupă mai mult de 75% din pagină, aceasta va fi inclusă într-o anexă și se va face referire la ea în textul lucrării.

Figurile pot fi dispuse pe pagină în format peisaj (landscape) dacă este necesar, cu condiția ca acestea să fie enumerate în Anexă și să li se facă referire în textul lucrării.

Către toate figurile trebuie să fie date referinţe și figura se plaseaza dupa prima referință la ea.

*Exemplu*:

Conform figurei 1.1 ....

**Figura 1.1 – Denumirea figurii**

**Conform subcapitolelor figurile nu se numerotează. Denumirile de figuri nu se scriu la sfârșitul paginii.**

**TABELE**

Tabele se numerotează în ordine crescătoare pentru fiecare capitol: prima cifră va indica numărul capitolului, iar a doua – numărul de ordine al elementului în respectivul capitol ca “Tabelul 1.1”.

Denumirile tabelelor se scriu deasupra acestora, fiind aliniate la dreapta. Textul în tabel se culege cu fontul – 11pt, interval – 1.

Tabelele pot fi dispuse pe pagină în format peisaj (landscape) dacă este necesar, cu condiția ca acestea să fie enumerate în Anexă și să li se facă referire în textul lucrării.

Către toate tabele trebuie să fie date referinţe și tabelul se plaseaza dupa prima referință la el..

*Exemplu:*

Conform datelor din tabelul 1.1 se recomandă.........

**Tabelul 1.1 - Denumire tabel**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **D** |
| 123 | 125 | 255411 | 553321 |
| 123569 | 2548 | 123355 | 1233 |

La continuarea tabelului pe altă pagină, denumirea se scrie în partea dreaptă și se indică “Continuarea tabelului 1.1” în partea dreaptă a tabelului.

Exemplu:

**Continuare tabelul 1.1**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **D** |
| 1235555 | 12588 | 25541188 | 55332188 |
| 1235696778 | 25488 | 12335588 | 123388 |

Titlurile rubricilor tabelului, în cazul divizării acestuia pe două sau mai multe pagini, se indică în cadrul ambelor pagini.

Denumirile de tabele nu se scriu la sfârșitul paginii.

**FORMULE**

Formulele se numerotează cu cifre arabe în ordine crescătoare pentru fiecare capitol: prima cifră va indica numărul capitolului, iar a doua – numărul de ordine al elementului în respectivul capitol, vor fi centrate, iar numerotarea acestora se va plasa la sfârșitul rândului.

Explicaţiile simbolurilor şi coeficienţilor din formule trebuie prezentate sub formulă, în acea ordine, cum sunt date în formulă.

Către toate formulele trebuie să fie date referinţe și formula se plaseaza dupa prima referință la ea.

*Exemplu:*

Conform formulei (1.1) se calculează.....

*Ax+B=C (1.1)*

**ANEXE**

Anexele trebuie să fie menționate în textul proiectului/tezei și se prezintă în ordine crescătoare (de exemplu, Anexa A, Anexa B etc.), fiecare anexă începe din pagină nouă și are un titlu, enunțat sub inscripția Anexa, centrat;

Formulele, tabelele și figurile din anexe, se numerotează, luând în considerare numărul anexei, de exemplu, Figura A.1.

În cazul când anexa conține o singură figură sau un singur tabel, denumirea anexei respective poate să coincidă cu denumirea tabelului sau a figurii;

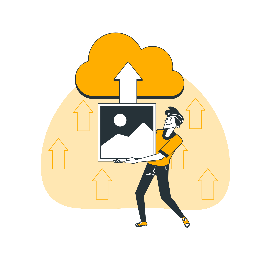
*Exemplu:*

**Anexa A**

**Denumire anexă**

**Tabelul A.1-Denumire tabel**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |



**Figura A.1- Denumire figură**

**REFERINȚE BIBLIOGRAFICE**

Referinţele bibliografice sunt prezentate într-o succesiune numerică, corespunzătoare ordinii citărilor lor în text. Citarea, forma scurtă a referinţei inserată în text între paranteze pătrate, permite identificarea publicaţiei din care s-a extras citatul sau ideea comentată etc. şi indicarea localizării în cadrul publicaţiei sursă.

Numerele, prezentate sub forma de indici între paranteze pătrate, înserate în text, trimit la document în ordinea în care ele sunt citate pentru prima dată. Citările următoare primesc acelaşi număr ca şi prima citare. Dacă sunt citate numai anumite părţi ale unui document, după numărul respectiv poate fi dată şi paginaţia, de exemplu [8, p. 231].

**Bibliografia în teză trebuie de redactat în aplicația [Zotero](https://www.zotero.org/) și anume stilul IEEE**