Proiect 2

În acest proiect echipa va dezvolta **parțial** o aplicație (cerințele sunt la alegerea echipei) folosind **metodologia SCRUM** [1,21]. Dezvoltarea va consta dintr-un singur Sprint, având 5 zile lucrătoare (40 de ore). Se recomandă ca aceste 5 zile sa fie distribuite de-a lungul a 5 săptămâni (o zi / săptămână). **Accentul în acest proiect pe este management și organizare, nu pe dezvoltare!**

**Roluri**: Unul dintre membrii echipei își va asuma rolul de Product Owner, un altul își va asuma rolul de Scrum Master, iar ceilalți vor fi Dezvoltatori. În echipă nu vor exista Testeri dedicați, generarea de teste fiind inclusă în procesul de implementare, iar testarea de acceptanță va fi făcută încrucișat (de către un alt Dezvoltator decât cel care a realizat implementarea).

**Cerințe:**

1. Se pornește de la un scenariu propriu care descrie cerințele sistemului ce urmează a fi dezvoltat (Requirements Document). Acesta poate cuprinde atât cerințe funcționale cât și cerințe nefuncționale și alte constrângeri. Cerințele sistemului vor fi alese astfel încât din acest document să rezulte cel puţin 15 User Stories [1-5]. (Responsabilitate: Product Owner)
2. Din Requirements Document vor fi derivate User Stories inițiale. (Responsabilitate: Product Owner)
3. User Stories de la punctul anterior vor fi rafinate astfel încât să satisfacă cele 6 criterii: Independent, Negotiable, Valuable to users or customers, Estimable, Small, Testable (INVEST) [1-3]. Să se justifice modificările făcute. Fiecărei User Story i se va atribui o prioritate, care va fi justificată. User Stories rezultate vor alcătui Project Backlog. (Responsabilitate: Product Owner și Dezvoltatorii)
4. Să se estimeze efortul (în story points) pentru fiecare User Story folosind Planning Poker [7,8] și să se justifice estimarea. (Responsabilitate: Dezvoltatori)
5. Se completeaza Sprint Backlog in Planning Meeting (care are loc in prima zi a Spint-ului). Observatie: În mod normal, **Sprint Backlog nu va cuprinde toate User Stories din Project Backlog;** din contră, având în vedere durata redusă a sprint-ului și dimensiunea echipei, este de așteptat ca doar o parte din ele să poată fi incluse în Sprint Backlog, selecția fiind făcută în funcție de estimarea efortului și priorități. (Responsabilitate: Dezvoltatori, Scrum Master, Product Owner)
6. Se va folosi un tool (la alegerea echipei) pentru project management și issue tracking [5,12]. (Responsabilitate: Dezvoltatori, Scrum Master, Product Owner)
7. Să se implementeze User Stories:

User Story-urile au fost implementate în limbajul C#, folosind framework-ul .NET, versiunea 6.

* 1. aproximativ jumătate dintre acestea folosind metoda **Test-Driven Developmen**t [1,19].
  2. iar restul în mod traditional, însă folosind tehnica **Pair Programming** [22].
  3. Să se descrie procesul şi evoluţia testelor.

Pentru implementarea folosind metoda TDD, am urmărit schema de dezvoltare a acesteia, care implică scrierea testelor înainte de implementarea propriu-zisă. În acest caz, dezvoltarea este realizată astfel încât în urma testului să nu apară erori.

Testul trebuie sa verifice o funcționalitate, astfel trebuie sa eșueze în stadiul inițial, apoi urmează redactarea codului pentru a face testul sa treacă parametrii. Acest pas este urmat de rularea din nou a testului pentru a ne asigura că va trece cu succes. În următoarea etapă evaluăm codul și dacă este necesar, modificăm scrisul cu scopul de a avea un cod cât se poate de eficient.

Dacă finalizăm cu succes primul test, trecem la următorul test pentru altă funcționalitate sau extindem testul deja făcut.

Acești pași formează împreună un ciclu testare – implementare – refactorizare care asigură prin repetarea acestuia faptul că toate testele vor trece cu succes.

* 1. Să se scrie cazurile de testare rezultate folosind un utilitar adecvat (e.g. JUnit pentru limbajul Java [17]).

Utilitarul folosit este NUnit, adecvat pentru limbajul de programare C#. Testarea a fost realizată având în vedere toate scenariile care pot apărea în urma utilizării unei funcționalități.

* 1. Să se stabileasca nivelul de acoperire realizat de fiecare dintre seturile de teste folosind un utilitar de code coverage precum cele prezentate in [20].

A screenshot of a computer

Description automatically generated

* 1. Să se comenteze avantajele si dezavantajele Test-Driven Development față de dezvoltarea tradițională.

Avantaje:

* Erorile sunt detectate instant: acestea pot fi identificate într-o fază incipientă, cât să nu afecteze proiectul într-un stadiu mult mai avansat, atunci când daunele ar fi mult mai mari.
* Contribuie la implementarea corectă a codului, implicând crearea unor scenarii, ceea ce asigură aplicabilitatea codului pentru situații concrete.
* Codul este mai lizibil, mai ușor de întreținut, ușurând procesul de dezvoltare pe viitor, mai ales atunci când sunt mai multe persoane care lucrează la același proiect.
* Implică realizarea unui cod care este testabil în mod automat, ceea ce ajută pe viitor la dezvoltarea mai multor funcționalități.

Dezavantaje:

* Este greu de realizat într-un mod corect și necesită un timp îndelungat de lucru, mai ales pentru începători, fiind nevoie de un nivel de experiență mai mare.
* TDD trebuie să fie aplicat în timpul implementării, nu după finalizarea acesteia – în acest caz nu se va beneficia de toate avantajele TDD (unii programatori pot ignora cu ușurință această regulă, în acest caz aplicarea metodei nu va avea aceeași valoare).
* Deoarece TDD este aplicat în timpul implementării, fazele clasice de dezvoltare a unui proiect sunt mai greu de delimitat (de exemplu: planificarea proiectului, implementarea codului și testarea).
  1. Să se comenteze avantajele si dezavantajele Pair Programmingfață de programarea individuală.

La prima vedere, metoda Pair Programming poate duce la un cod de nivel de calitate ridicat pentru că mai multe persoane urmăresc dezvoltarea acelorași idei. În schimb, prin prezența mai multor programatori pot apărea neînțelegeri cu privire la abordarea anumitor probleme. Prin atenția mai multor persoane, apariția erorilor și a bug-urilor este redusă. Totuși, dacă doar un dezvoltator scrie cod, iar ceilalți nu, nivelul de atenție al celorlalți poate scădea pe termen lung.

1. Să se utilizeze un sistem de control al versiunilor, de exemplu SVN (Subversion), CVS (Concurrent Version System), GIT etc [13,14]. Să se folosească o platformă (cu găzduire free) pentru repository-ul proiectului dezvoltat, să se instaleze şi utilizeze diferiți clienți pentru sistemul de control al versiunilor ales (de exemplu SmartSVN, TurtoiseSVN, Subclipse pentru Subversion). (Responsabilitate: Dezvoltatori, Scrum Master)

Sistemul de control al versiunilor pe care l-am utilizat este GIT, fiind cunoscut de către întreaga echipă. Acesta implică numeroase avantaje precum: popularitatea sa, viteza, performanța, alături de posibilitatea de a reveni la orice versiune a proiectului din istoric. Platforma pentru repository-ul proiectului dezvoltat este GitHub deoarece este ușor de folosit și permite încărcarea unui proiect public sau privat în mod gratuit. Clientul pentru sistemul de control al versiunilor este Visual Studio, care are Git integrat în mediul de dezvoltare, dar și clientul Git Bash.

1. Sprint Daily Standup Meetings vor avea loc în fiecare zi din Sprint, iar raportul prezentat la sfârșitul proiectului va consemna principalele impedimente apărute. (Responsabilitate: întreaga echipa)

SĂPTĂMÂNA 4.12.2023-10.12.2023:

Sprint Daily Standup Meeting 1 (4.12.2023):

Acest meeting include discuții despre obiectivele Sprint-ului, organizând User Story-urile și împărțindu-le ca task-uri pentru membrii echipei. Se estimează efortul pentru implementarea User Story-urilor. Se stabilește sistemul de control al versiunilor (Git) și se crează un repository al proiectului pe GitHub. De asemenea, se alege clientul pentru sistemul de control al versiunilor, mai exact Visual Studio.

Sprint Daily Standup Meeting 2 (5.12.2023):

Pentru Sprint Daily Standup Meetings, Scrum Master-ul ales este Radu, care a facilitat intâlnirile și a ghidat echipa pentru a înțelege și aplica Scrum Framework-ul.

Se începe dezvoltarea proiectului cu User Story-urile Create User Account si Edit Profile. Acestea au fost testate utilizând TDD. În urma încercării de implementare cu TDD pentru Create Usder Account, s-a decis a se utiliza metoda clasică, folosirea framework-ului pentru acest User Story însemnând o pierdere de timp pentru implementare.

Cosmin si Radu:

* Implementarea funcționalităților pentru Create User Account.
* Probleme întâlnite: folosirea framework-ului pentru TDD a îngreunat timpul de dezvoltare, s-a renunțat la implementarea cu TDD.

Denisa si Alex:

* Implementarea funcționalităților pentru Edit Profile.
* Probleme întâlnite: nu s-au întâmpinat probleme la implementare.

Sprint Daily Standup Meeting 3 (6.12.2023):

Se continuă dezvoltarea proiectului cu User Story-urile Create Product Listing și Product and Search Filtering.

Cosmin si Radu:

* Implementarea funcționalităților pentru Create Product Listing.
* Probleme întâlnite: nu s-au întâlnit probleme la implementare.

Denisa si Alex:

* Implementarea funcționalităților pentru Product and Search Filtering.
* Probleme întâlnite: nu au fost probleme la implementare.

Sprint Daily Standup Meeting 4 (7.12.2023):

Se continuă dezvoltarea proiectului cu User Story-urile View Product Details și Shopping Cart Management. Pentru Checkout nu s-a realizat testarea prin TDD deoarece momentan lipsește interfața grafică a aplicației.

Cosmin si Radu:

* Implementarea funcționalităților pentru View Product Details.
* Probleme întâlnite: nu s-au întâlnit probleme la implementare.

Denisa si Alex:

* Implementarea funcționalităților pentru Shopping Cart Management.
* Probleme întâlnite: nu s-a realizat testarea prin TDD pentru Checkout (lipseste interfata grafica).

Sprint Daily Standup Meeting 5 (8.12.2023):

Se continăa dezvoltarea proiectului cu User Story-ul Multiple Payment Options. Nu s-a reușit testarea prin TDD deoarece momentan lipsește interfața grafică a aplicației.

Cosmin si Radu:

* Implementarea funcționalităților pentru Multiple Payment Options.
* Probleme întâlnite: nu s-a realizat testarea prin TDD ( lipsește interfața grafică).

Denisa si Alex:

* Verificarea implementărilor realizate până în momentul de față și rezolvarea eventualelor erori sau bug-uri apărute.

În anumite cazuri de erori sau bug-uri, Product Owner-ul (Ionut) a fost prezent pentru a-și exprima părerea și a ajuta la rezolvarea acestora.

1. Implementarea (fiecărei User Story în parte) va fi prezentată (și validată/ invalidată de către Product Owner) în Review Meeting, care va avea loc in ultima zi a Sprint-ului. Tot în ultima zi va avea loc și o foarte scurtă (maxim 20 minute) Retrospective Meeting. (Responsabilitate: întreaga echipa, facilitată de Scrum Master)

Review Meeting:

Implementarea fiecărui User Story este prezentată de către dezvoltatori către Product Owner, care efectuează validarea. Participanții (echipa de dezvoltare, manageri) adresează întrebări specifice și clarifică eventualele nelămuriri. Față de cantitatea de munca propusă, în momentul actual a fost realizată 70% din aceasta, însă la un nivel de calitate superior.

Cosmin si Radu:

* Ce am realizat:
  + Implementarea următoarelor User Story: Create User Account, Create Product Listing, View Product Details, Multiple Payment Options.
* Ce probleme am întâlnit:
  + Folosirea framework-ului pentru TDD a îngreunat timpul de dezvoltare.
  + Pentru anumite User Story, din lipsa interfeței grafice, nu s-a aplicat TDD.
* Pașii urmatori:
  + Implementarea următoarelor User Story.
  + Imbunătățirea continuă a ceea ce s-a realizat.

Denisa si Alex:

* Ce am realizat:
  + Implementarea următoarelor User Story: Edit Profile, Product and Search Filtering, Shopping Cart Management.
* Ce probleme am întâlnit:
  + Pentru anumite User Story, în lipsa interfeței grafice, nu s-a aplicat TDD.
* Pașii urmatori:
  + Implementarea următoarelor User Story.
  + Imbunătățirea continuă a ceea ce s-a realizat.

Retrospective Meeting:

Product Owner-ul evaluează procesul și modul de dezvoltare al proiectului în timpul acestui Sprint ca fiind unul eficient, cu un nivel de comunicare și colaborare în echipă ridicat.

Cosmin si Radu:

* Ce a mers bine:
  + Am reușit implementarea User Story-urilor fără apariția bug-urilor
  + Colaborarea în echipă a fost îmbunătățită de Sprint Daily Standup Meetings
* Ce poate fi îmbunătățit:
  + Estimarea efortului pentru fiecare User Story deoarece nu s-a reușit realizarea în totalitate a obiectivelor propuse

Denisa si Alex:

* Ce a mers bine:
  + Am reușit implementarea User Story-urilor fără apariția bug-urilor
* Ce poate fi îmbunătățit:
  + Comunicarea între echipe pentru a evita apariția bug-urilor de implementare în viitor

1. Să se precizeze (ca procentaj, la sfârșitul raportului) contribuția fiecărui membru al echipei la realizarea proiectului.

Echipa este formată din următorii membri:

* Ionescu Ioan (Product Owner) – 25%
* Ioan Radu (Dezvoltator si Scrum Master) -16.66%
* Rinu Alexandru (Dezvoltator) – 25%
* Hongu Cosmin (Dezvoltator) – 16.66%
* Bocu Denisa (Dezvoltator) – 16.66%