Proiect 1

* User Stories

Pentru ilustrarea tehnicii de dezvoltare Test-Driven Development, au fost alese 2 User Stories ce fac parte din Proiectul 2. În cadrul acestui proiect, s-a dorit realizarea unei aplicații web de comerț, ce servește drept platformă pentru utilizatorii ce doresc să vândă sau să cumpere produse online, permițând răsfoirea catalogului de produse, gestionarea conturilor, adăugarea de noi produse și achiziționarea acestora prin intermedierea plății. Cele două User Story sunt: Create Product Listing și Product and Search Filtering.

Utilitarul folosit pentru TDD 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.

În cazul metodei tradiționale, aceasta s-a realizat folsind documentația generată în Swagger.

* Ce este TDD?

TDD este o metoda de dezvoltatre software care implică scrierea testelor înainte de implementarea propriu-zisa, fiind opusul testării tradiționale. În acest caz, dezvoltarea este realizată astfel încât în urma testului să nu apară erori.

* Ce este metoda tradițională?

Metoda tradițională implică implementarea propriu-zisa urmată de testare. În urma acestei etape, în cazul în care se întâmpină erori, programatorul se va întoarce la faza de implementare și va aplica schimbările necesare.

* Avantajele și dezavantajele folosirii TDD.

În primul rand, erorile sunt detectate instant: acestea pot fi identificate într-o fază incipentă, cât să nu afecteze proiectul într-un stadiu mult mai avansat, atunci când daunele ar fi mult mai mari. De asemenea, TDD 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 Folosirea TDD implică realizarea unui cod care este testabil în mod automat, ceea ce ajută pe viitor la dezvoltarea mai multor funcționalități.

Pe de altă parte, TDD 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. Acesta 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 neavând aceeași valoare). Deoarece 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).