Testni slučajevi i jedinični (unit) testovi

PREDAVANJE 11

Testiranje

Testni slučajevi

Jedinični testovi

Testiranje korisničkog sučelja

Ponavljanje: Zašto testirati?

Kako bismo na vrijeme otkrili pogreške i nepravilnosti u programskoj podršci (softwareu)

Tražimo **bugove**

- Kakve?

Kako sve možemo testirati?

Različiti tipovi grešaka

Greške u sintaksi

Greške u izvođenju

Neobrađene iznimke

Sinkroznizacijski problemi

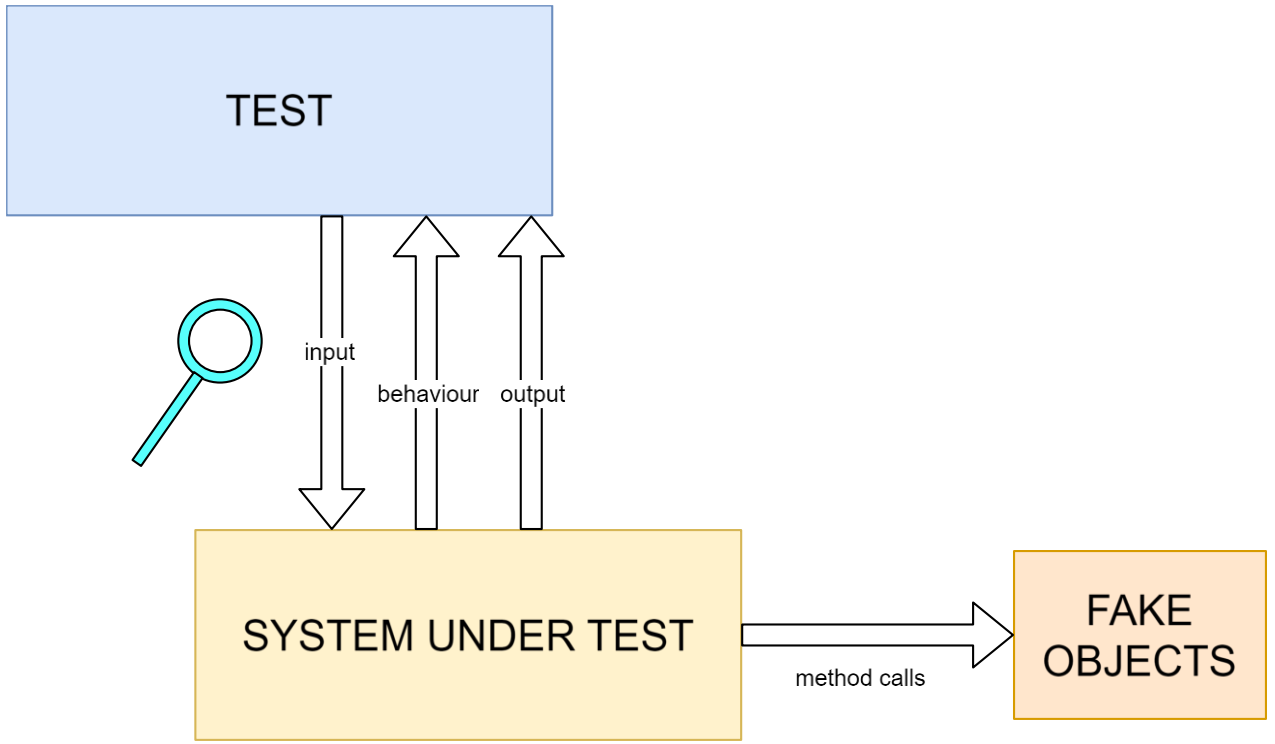
Memory leaks

Beskonačne petlje

Logičke greške

Testiranje

Testiranje možemo formalizirati kao provjeru ponašanja i/ili rezultata SUT-a *(system under test)* za dane ulazne podatke



Testni slučajevi

TEST CASES

Testni slučajevi

Osnovni formalni raspis testova

Opisati slučaj koji testiramo

Atomizirani dio funkcionalnosti

Definiramo set akcija koje je potrebno izvršiti kako bismo dobili neki rezultat

Provjera dobivenog rezultata sa očekivanim rezultatom

Primjer

Testni slučaj: **Korisnik se može prijaviti koristeći valjane vjerodajnice**

Koraci:

1. Posjetiti login stranicu
2. Popuniti polje korisničkog imena
3. Popuniti polje zaporke
4. Pritisnuti ”Login” tipku

Očekivani rezultat:
Korisnik je uspješno prijavljen u sustav i presumjerenu je na glavnu stranicu.

Primjer

Prethodni primjer definira **pozitivni test**. Moguće je također definirati varijantu testa koja provjerava ponašanje i/ili rezultate funkcionalnosti ukoliko korisnik pogrešno koristi funkcionalnost.

To nazivamo **negativan test**.

Primjer – negativan test

Testni slučaj: **Korisniku je onemogućena prijava uz nevaljane vjerodajnice**

Koraci:

1. Posjetiti login stranicu
2. Popuniti polje korisničkog imena
3. Popuniti polje zaporke
4. Pritisnuti ”Login” tipku

Očekivani rezultat:
Korisnik **nije** prijavljen u sustav te mu se pokazuje poruka o grešci.

Kompleksniji primjer

....

Naziv: Kao nastavnik, želim uploadati materijale za kolegij

Preduvjeti:

- Korisnik je prijavljen kao nastavnik
- Nastavnik ima pristup kolegiju za kojeg prenosi materijale

Koraci:

1. Posjetiti stranicu kolegija (kao nastavnik)
2. Odabrati opciju „Prijenos materijala”
3. Popuniti naslov datoteke: „Predavanje 1"
4. Odabrati datoteku: „Predavanje01.pdf"
5. Stisnuti tipku „Prenesi”

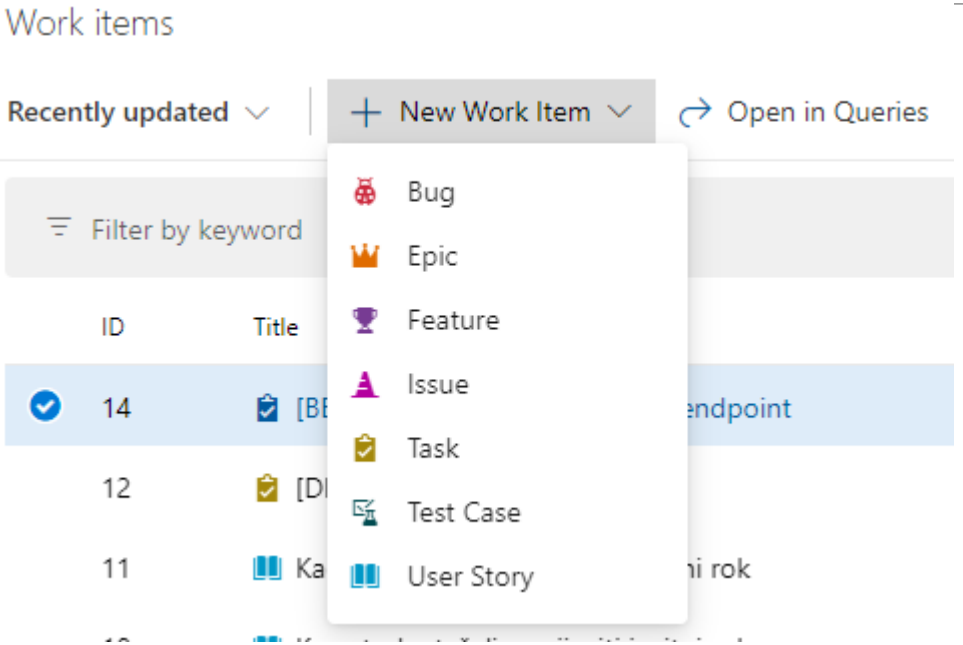
Očekivani rezultat:

- Materijal „Predavanje 1” se nalazi u listi materijala kolegija
- Prikazana je poruka o uspješnoj akciji

Uvjeti:

Datoteka „Predavanje01.pdf„ je spremljena u sustavu i dostupna je svim studentima upisanim na kolegij.

Gdje ih dodati



Jedinični testovi

UNIT TESTS

Jedinični testovi (*Unit tests*)

Testiraju se osnovne jedinice programske podrške, **metode**

Pomažu u razvoju novih značajki sa ranim otkrivanjem grešaka

Osiguravaju razvoj novih značajki bez narušavanja postojećih

xUnit

Skup radnih okvira koji se temelje na SUnit okviru razvijenom za programski jezik SmallTalk

- JUnit (Java)
- XUnit / NUnit (C#)
- unittest (Python)

…

Arhitektura xUnit-a

1. Test
2. Tvrdnje
3. Kolekcija testova
4. Kontekst testa
5. Izvođač testova

Test

Test testira jednu metodu

Svaki test trebao bi biti izoliran i neovisan o drugim testovima

Svaki test je jedna metoda koja testira danu jedinicu

Atribut **[Test]** (NUnit) ili **[Fact]** (xUnit)

Anotacija **@Tes**t (JUnit)

Tvrdnje (Assertions)

Logičke tvrdnje koje jedinični test treba zadovoljiti kako bi jedinica bila ispravna

- Je li očekivana vrijednost jednaka dobivenoj
- Je li dobivena lista prazna
- Sadrži li dobivena lista određeni element
- Baca li poziv zadane metode iznimku

Tvrdnje - xUnit.net (C#)

```
// Standardno
Assert.Equal(expectedString, actualString);
Assert.NotEqual(expectedValue, actualValue);
Assert.Null(actualValue);
// Kolekcije
Assert.Contains(expectedThing, collection);
Assert.DoesNotContain(expectedThing, collection);
Assert.Empty(collection);
// Iznimke
Assert.Throws<T>(() => sut.Method());
```

Tvrdnje - JUnit

```
// Standardno
assertEquals(expectedString, actualString);
assertNotEquals(expectedValue, actualValue);
assertNull(actualValue);
// Kolekcije
assertTrue(collection.contains(expectedThing));
assertFalse(collection.contains(expectedThing));
assertTrue(collection.isEmpty());
// Iznimke
assertThrows(MyException.class, () -> sut.method());
```

Kolekcija testova

Skupina jediničnih testova povezanih u klasi / modulu

Kontekst testova

- Test fixture
- Priprema konteksta za izvođenje jediničnog testa
- Pokretanje procesa, stvaranje pomoćnih baza podataka, instanciranje lažnih objekata, itd.

Izvođač testova – Test Runner

Program koji izvršava definirane jedinične testove i prikazuje rezultate

Integriran u Visual Studio ili dostupan preko naredbe

```
dotnet test
```

AAA

1. Arrange
2. Act
3. Assert

GWT

1. Given
2. When
3. Then

Unit test - primjer

```
public class Item
{
    public string Name { get; set; }
    public float Price { get; set; }

    public Item(string name, float price)
    {
        Name = name;
        Price = price;
    }

    public void ApplyDiscount(float percentage)
    {
        Price -= percentage * Price;
    }

}
```

Unit test - primjer

```
[Fact]
public void ApplyDiscount_ShouldCorrectlyDecreasePrice()
{
    // Arrange
    Item item = new("test", 100f);

   // Act
    item.ApplyDiscount(0.25f);

   // Assert
    Assert.Equal(75f, item.Price);
}
```

Unit test - primjer

Razmisliti o mogućim scenarijima i rubnim slučajevima.

Kada upotrijebiti različite tvrdnje?

Kako pokrenuti testove?

Testiranje korisničkog sučelja

Potrebno kada razvijamo desktop, mobilna ili web korisnička sučelja

Omogućuju provjeru tijeka radnje zadanih funkcionalnosti aplikacije

Simuliraju ponašanje korisnika

Temeljene na testnim slučajevima

Radni okviri

- Playwright
  - Preporuke!

- Selenium