

Uputstvo za rad na μ Projektima (Softversko inženjerstvo)

Način rada

Polaznici dizajniraju i implementiraju softver pridržavajući se principa softverskog inženjerstva. Zbog toga se izrada projekata deli na 3 faze:

1. Dizajn
2. Implementacija
3. Testiranje

Dizajn

Cilj ove faze je da polaznici preko UML dijagrama modeluju klase koje će da koriste u projektu kao i neke osnovne tokove rada softvera. U zavisnosti od projekta, posavetovati se sa mentorom ili ostalim saradnicima o specifičnim dijagramima neophodnim za projekat. Dijagrami će biti osnova za dalji rad i implementaciju, s obzirom na činjenicu da će se vaš razvoj softvera zasnivati na onome šta dizajnirate, tako da treba voditi računa da ova faza bude dobro urađena.

Implementacija

U ovoj fazi polaznici se pridržavaju dizajna koji su napravili u prethodnom koraku, ili da izmene dizajn ako primete propuste tokom implementacije. Ova faza bi trebalo da traje najduže i u ovoj fazi izrađujete softver.

Testiranje

Ono što razlikuje inženjera od developera je ova faza. Polaznici treba da nauče kako da testiraju svoj softver i vide da li se ponaša onako kako su zamislili. Kreirate testove koji će pokriti implementaciju i proveriti njenu ispravnost.

Potreban softver

Python 3.12.2

VSCode

Instalirati sledeće ekstenzije za rad u VSCode: Python, PythonDebugger, Pylance, GitLens

Git for Windows

StarUML

Konfigurisanje virtuelnog okruženja (venv)

Instalirati python svom računar. Na Windows instalirajte Anaconda-u, a na Linuxu je podrazumevano instaliran.

Korak 1: Otvorite Terminal ili Command Prompt

Za korisnike Windows-a: Pritisnite **Win + R**, otkucajte **cmd**, i pritisnite Enter.

Za korisnike Linux-a: Možete koristiti **Ctrl + Alt + T** da otvorite terminal.

Korak 2: Kreirajte Novi Direktorijum za Vaš Projekat (Opciono)

Navigirajte do direktorijuma gde želite da kreirate svoj projekat. Možete preskočiti ovaj korak ako već imate direktorijum za svoj projekat.

bash

Za Linux

cd /putanja/do/vas/direktorijum

Za Windows

cd C:\putanja\do\vas\direktorijum

Korak 3: Kreirajte Virtuelno Okruženje

Sada, kreirajmo virtuelno okruženje koristeći Python-ov ugrađeni modul **venv**.

bash

Za Windows i Linux

python -m venv moje_okruzenje

Zamenite **moje_okruzenje** imenom koje želite da dodelite svom virtuelnom okruženju.

Korak 4: Aktivirajte Virtuelno Okruženje

Koraci za aktivaciju se malo razlikuju između Windows-a i Linux-a.

Za Windows:

bash

moje_okruzenje\Scripts\activate

Za Linux:

bash

```
source moje_okruzenje/bin/activate
```

Korak 5: Proverite Aktivaciju

Primetićete da se vaša komandna linija promenila tako da sadrži ime vašeg virtuelnog okruženja. To znači da je virtuelno okruženje sada aktivno.

Korak 6: Instalirajte Zavisnosti

Sada možete instalirati pakete i zavisnosti specifične za vaš projekat bez uticaja na globalno Python okruženje.

bash

```
pip install ime_paketa
```

Zamenite `ime_paketa` imenom paketa koji želite da instalirate.

Korak 7: Deaktivirajte Virtuelno Okruženje

Kada završite sa radom na vašem projektu, možete deaktivirati virtuelno okruženje.

bash

```
Deactivate
```

Šta je i kako se koristi GitHub Project?

GitHub Project je funkcionalnost koja vam omogućava da organizujete, pratite i upravljate radom na vašim GitHub repozitorijumima koristeći tablu sa karticama.

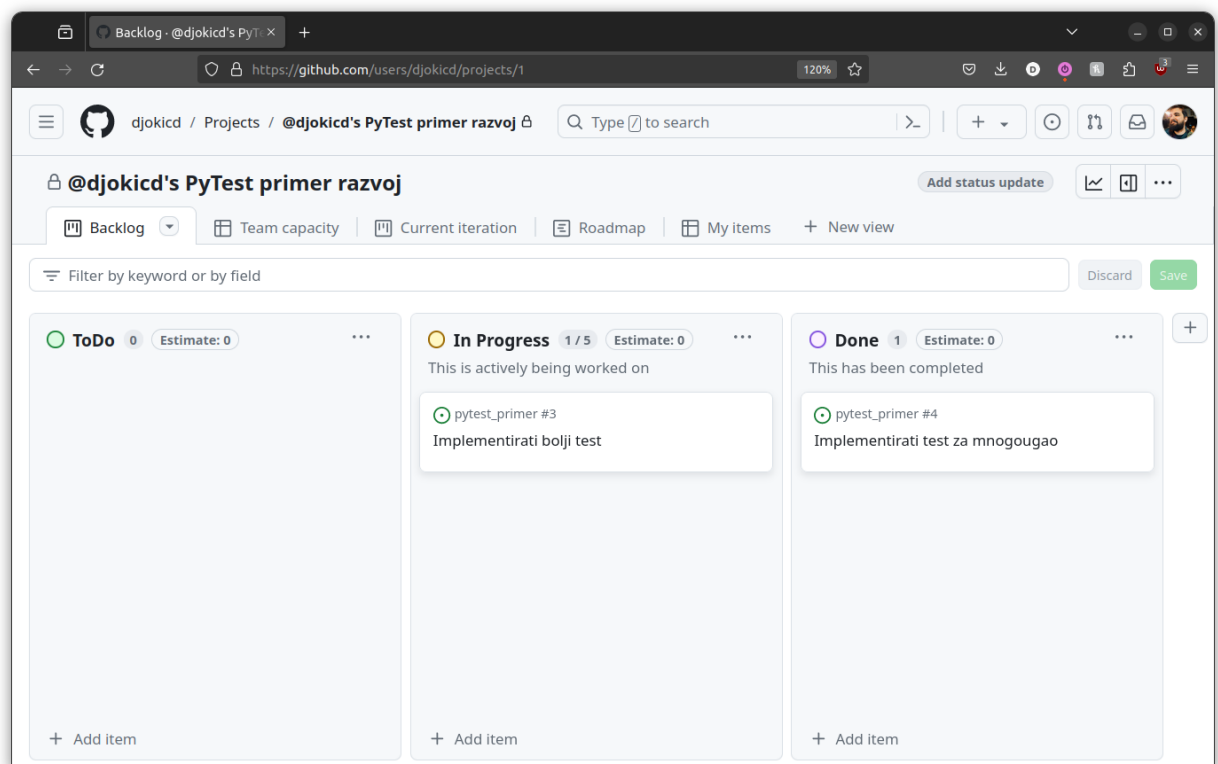
Kako kreirati GitHub Project?

1. **Otvorite repozitorijum:** Počnite tako što ćete posetiti stranicu željenog repozitorijuma na GitHub-u.
2. **Izaberite opciju "Projects":** Na vrhu repozitorijumove stranice, kliknite na karticu "Projects".
3. **Kreirajte novi Project:** Ako već ne postoji projekat, pritisnite dugme "New project", i kreirajte projekat sa odgovarajućim nazivom.
4. **Izaberite šablon (opcionalno):** GitHub nudi nekoliko šablona za Projects. Za sada, najbolje je za naše potrebe izabrati prazan projekat.

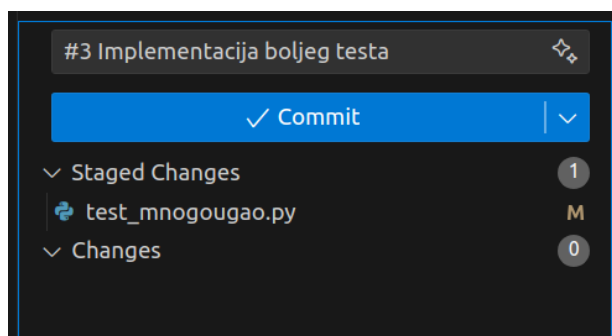
Kako koristiti GitHub Project?

Nakon što kreirate Project:

1. **Dodajte kolone:** Kliknite na "Add a column" kako biste kreirali kolone. Kolone omogućavaju organizaciju po fazama rada. Predefinisane faze, koje ćemo mi koristiti, jesu "To do", "In progress", "Done".
2. **Dodajte kartice zadatka:** U svakoj koloni, dodajte kartice koje predstavljaju konkretne zadatke. Kliknite „+“ i unesite naziv zadatka. Naziv zadatka započnite sa „#“ da bi se kartica vezala za projekat. Dodavanjem imena taska kreira se novi Issue koji je vezan za taj task. Na primer kao na slici:



Ukoliko se onda commit prefixuje npr sa #3, onda će on biti vezan za issue #3 čime je vezan za task i biće moguće pratiti napredak na pojedinačnim taskovima. Na primer, ukoliko načinimo i pushujemo commit sa slike



Takav commit će se videti u okviru Issue i u okviru odgovarajuće task kartice:

Implementirati bolji test #3

Open



djokicd/pytest_primer

Public

djokicd opened this issue 20 minutes ago

No description provided.

added this to @djokicd's PyTest primer razvoj 20 minutes ago

moved this to **ToDo** in @djokicd's PyTest primer razvoj 20 minutes ago

moved this from **ToDo** **to** **In Progress** in @djokicd's PyTest primer razvoj 6 minutes ago

added a commit that references this issue 1 minute ago

[#3](#) Implementacija boljeg testa

c4de675

3. **Organizujte kartice:** Pomerajte kartice između kolona kako biste pratili napredak zadatka. Organizacija se najakše izvodi prevlačenjem kartica između kolona.
4. **Dodajte detalje zadatka (opciono):** Kliknite na karticu zadatka kako biste dodali dodatne informacije kao što su opis zadatka, oznake, ili eventualno dodatne podzadatke
5. **Dodelite zadatke članovima tima:** Ako radite u timu, dodelite zadatke odgovornim članovima klikom na karticu i odabirom opcije "Assignees".

Održavajte ažurirane kartice kako bi se reflektovalo trenutno stanje projekta.