

Sissejuhatus tarkvaraarendusse

Martti Raavel

martti.raavel@tlu.ee

Neljas loeng

- Meenutame eelmist loengut
- Pull Request
- Forkimine
- Dokumentatsioon
- Mittefunktsionaalsed nõuded

Pull Request

`Pull Request` on mehhanism mis võimaldab arendajatel teavitada teisi arendajaid oma muudatustest, mis on tehtud koodi baasis, ning paluda neil üle vaadata ja lisada need muudatused teise koodi harusse (sageli peamisse harusse).

Fork

Fork imine tähendab olemasoleva repositooriumi kopeerimist enda Githubi kontole.

Fork itud repositoorium säilitab ühenduse originaalrepositooriumiga, mis võimaldab algse repositooriumi omanikul vaadata ja kopeerida muudatusi, mis on tehtud forkitud repositooriumis.

Fork imine annab võimaluse teha muudatusi repositooriumis, millele sul ei ole kirjutamisõigusi (läbi **Pull Request** i).

Fork vs Clone

Mis vahe on repositooriumi `fork` imisel ja `clone` imisel?

Dokumentatsioon

Tarkvaraarenduse dokumentatsioon viitab kirjalike või visuaalsete materjalide loomise ja korrastamise protsessile, mis kirjeldavad tarkvaratoote või -süsteemi disaini, arendust ja kasutamist.

Tehnilised kirjeldused

Dokumendid, mis kirjeldavad tarkvarasüsteemi, sealhulgas andmestruktuuride, algoritmide ja liideste kavandamist ja rakendamist.

Kasutusjuhendid

Dokumendid, mis kirjeldavad tarkvaratoote kasutamist, sealhulgas installi-, konfiguratsiooni- ja kasutusjuhiseid.

Väljalaskemärkmed

Dokumendid, mis kirjeldavad tarkvaratoote iga versiooni muudatusi ja täiustusi.

Koodikommentaarid

Kommentaarid, mis lisatakse tarkvaratoote lähtekoodile, et kirjeldada, kuidas koodi erinevad osad töötavad ja kuidas need on kasutamiseks mõeldud.

API dokumentatsioon

Dokumendid, mis kirjeldavad tarkvarasüsteemi poolt avaldatavaid liideseid, sealhulgas iga liidese sisendeid, väljundeid ja käitumist.

Dokumentatsiooni loomine

Tarkvaraarenduse dokumentatsiooni koostamine võib olla keeruline protsess, kuna olenevalt projekti vajadustest võib olla vaja koostada palju erinevat tüüpi dokumentatsiooni. Siin on aga mõned üldised sammud, mida saab tõhusa tarkvaraarenduse dokumentatsiooni loomiseks järgida

Tehke kindlaks dokumentatsiooni eesmärk ja ulatus

Enne dokumentatsiooni loomise alustamist on oluline kindlaks teha, millist tüüpi dokumentatsiooni on vaja, kes on sihtrühm ja millised on dokumentatsiooni eesmärgid.

Dokumentatsiooni kavandamine

See samm hõlmab dokumentatsiooni struktuuri, vormingu ja sisu määratlemist. Peaksite kaaluma dokumentatsiooni (nt tekst, diagrammid või videod) parimat vormingut ja otsustama, milline sisu tuleb kaasata.

Dokumentatsiooni kirjutamine

Kui planeerimine on lõpetatud, võite alustada dokumentatsiooni kirjutamist. Kasutage selget ja ülevaatlikku keelt ning veenduge, et dokumentatsioon oleks kergesti loetav ja arusaadav. Peaksite lisama ka kõik asjakohased diagrammid, pildid ja viited, mis aitavad sisu selgitada.

Dokumentatsiooni testimine ja ülevaatamine

Oluline on dokumentatsiooni testida, et olla kindel, et see on täpne ja hõlpsasti jälgitav. See võib hõlmata seda, et teised arendajad, sidusrühmad või lõppkasutajad vaatavad tagasiside andmiseks dokumentatsiooni üle.

Dokumentatsiooni värskendamine ja hooldamine

Tarkvaraarendus on pidev protsess, mistõttu on oluline dokumentatsiooni projekti edenedes uuendada ja säilitada. See võib hõlmata uue teabe lisamist, olemasoleva teabe värskendamist ja aegunud teabe eemaldamist.

Näpunäiteid

- Kasutage kogu dokumentatsioonis ühtset vormingut ja stiili
- Keeruliste mõistete selgitamiseks kasutage diagramme ja pilte
- Hoidke dokumentatsioon ajakohasena
- Kasutage selget ja kokkuvõtlikku keelt
- Kaaluge sihtrühma vajadusi ja vaatenurki

Mittefunktsionaalsed nõuded

Funktsionaalsed vs mittefunktsionaalsed nõuded?

Mittefunktsionaalsed nõuded

Kriteeriumid, mis määravad süsteemi kvaliteedi, toimivuse ja töökindluse, kuid ei ole seotud süsteemi konkreetsete funktsionaalsete omadustega

Mittefunktsionaalsed nõuded projekte

- Jõudlus
- Skaleeritavus
- Turvalisus
- Usaldusväärsus
- Hooldatavus
- Kasutatavus
- Ühilduvus

Mittefunktsionaalsed nõuded koodile

- Koodi loetavus
- Koodi struktuur ja modulaarsus
- Dokumentatsioon
- Testitavus
- Versioonihaldus
- Koodi optimeerimine
- Turvalisus
- Koodi ühilduvus
- Koodi läbipaistvus ja lähtekoodi avatus
- Arendusstandardite ja kodeerimisstiilide järgimine

Kodune töö

- Lisage oma viimasele Programmeerimine I kodutööle dokumentatsioon läbi `Pull Requesti` ja määrake sellele ülevaatajaks sama üliõpilane, kellega koos tegite eelmist kodust tööd
- Kui olete ise määratud ülevaatajaks, siis tehke ülevaatus (`Approve` või `Request changes`) ja tehke `Merge` kui kõik on korras

Lisatud dokumentatsioon peab sisaldama vähemalt:

- Autori nimi
- Mis rakendusega tegemist on
- Kuidas rakendust käivitada
- Kuidas rakendust kasutada