**2 namų darbas**

PS kūrimo procesas: užduočių programavimas ir testavimas

**Tikslas**

Įgyvendinti PS kodo realizavimo ir testavimo procesus

Užduotys:

1. Naudoti kodo paskirstymo versijų valdymo sistemas GIT (pasirinktinai). Kurti versijas, išsišakojimus/apjungimus ir peržiūras. Parengti kodo testavimui reikiamus vienetų testus (ang. Unit). Įvertinti kodo pagendiąm testais.
2. Taikyti TDD kodo rašymo techniką (arba sukurti reikalavimų testavimą taikant BDD techniką).
3. Taikyti integracinį arba regresinį testavimą.
4. Parengti procesų gerinimo išvadas

**Reikalavimai:**

**1\_užduotis:**

1. Susikurti kodo saugyklą pasirinktoje GIT aplinkoje
2. Sukurti šakojimus (angl. branches): master, *develop*, feature. Kiekvienas komandos narys turi feature išsišakojimą.
3. Kiekviena šaka turi commit’us su komentarais kas buvo atnaujinta/pakeista/neveikia. Komenrtare paminėti BackLog item #. Kiekvienas komandos narys sukuria bent 5-n commit’us, kurie
   1. Sukuria funkcionalumą/taisyklę/apribojimą
   2. Ištaisyta klaida/os
   3. Atliktas kodo pertvarkymas (refactoring)
4. Pateikti network diagramą arba naudoti įrankį vizualiam commit’ų ir išsišakojimų pakeitimams parodyti
5. Mokėti komentuoti visą Git Flow, peržiūrėti bet kurį commit, merge ir pan. Turi atsipindėti komandinis darbas.
6. IDE aplinkoje įdiegtas įrankis, leidžiantis fiksuoti kodo pagengiamumą testais (code coverage)
7. Kiekvienas komandos narys sukuria vienam savo funkciniam commit’ui Unit testus. Pasiekti kodo padengiamumą daugiau kaip 90%

**2\_užduotis: 1 variantas**

1. Kiekvienas komandos narys pasirinktam reikalavimui arba jo priėmimo kriterijui pritaiko TDD kodo rašymo techniką.
2. Taikant TDD galime realizuoti pasirinktą reikalavimą ar jo priėmimo kriterijų. (testuojamas kodas skiriasi nuo 1 užduoties 7 punkto)
3. Unit testų kiekis padengia bent 90% kodo
4. Pasirinktam kodo fragmentui:
   1. Sukurti fail testą
   2. Parašyti kodą
   3. Sukurti pass testą
   4. Pertvarkyti kodą (refactoring)
   5. Kartoti c ir d etapus
5. TDD įvykių eigą pateikti confluence, komentuotame kode arba įrašyti video

**2\_užduotis: 2 variantas**

1. Kiekvienas komandos narys sukuria scenarijus taikant BDD testavimo techniką pasirinktam reikalavimui:
   1. Aprašyti bent yra bent 2-n scenarijai, kurie padengia pasirinktą reikalavimą su priėmimo kriterijais;
   2. Sukurti scenarijaus automatizuotą įvykdymą

**3\_užduotis**

1. Kiekvienas komandos narys sukuria bent 1 integracinį testą su minimum 2-n testiniais atvejais tarp dviejų sistemos komponentų/lygių.
2. Testiniai atvejai padengia fukcionalumą ir nefunkcinių reikalavimų išpildymą.
3. Galima rašyti integracinį unit testą arba taikyti spiacializuotus tam skirtus įrankius kaip Postman.

**4\_užduotis**

1. Užduočių programavimo pertvarkymo komandoje iššūkiai. Darbas su GIT. Kokia GIT struktūrą ir komandos taisyklės efektyvintų užduočių programavimo ir testavimo procesą
2. Unit testo ir testinių atvejų kūrimo principai, efektyvūs kodo padengimas testais siekiant išpildyti reikalavimą ir priėmimo kriterijus
3. Pasiūlyti taisykles (2-5), kurių laikantis mažėtų klaidų tikimybė, kylanti iš kodo rašymo komandoje ir testavimo procesų.
4. Pateikti dokumente arba confluence

Vertinimas

|  |  |  |
| --- | --- | --- |
| Vertinimo objektas | Kriterijai (pagal reikalvimai) | Svoris |
| 1 užduotis | Išpildyti reikalavimai ir geba apginti atliktas užduotis | 3 balai |
| 2 užduotis | Išpildyti reikalavimai ir geba apginti atliktas užduotis | 2.5 balai |
| 3 užduotis | Išpildyti reikalavimai ir geba apginti atliktas užduotis | 2.5 balai |
| 4 užduotis | Išvados minėtais punktais | 2 balai |