

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Programų sistemų kūrimo metodų tyrimas

Testavimo planas

Atliko:	3 kurso 5 grupės studentai	
	Simonas Mikulis	(parašas)
	Mantas Petrikas	(parašas)
Darbo vadovas:	Dr. Vytautas Valaitis	(parašas)

Vilnius – 2017

TURINYS

1. TESTAVIMO PLANO IDENTIFIKATORIUS	2
2. DOKUMENTAI IR NUORODOS Į JUOS	3
3. ĮŽANGA	4
4. TESTAVIMO OBJEKTAI	5
5. PROGRAMINĖS ĮRANGOS RIZIKOS	6
6. TESTUOJAMOS SAVYBĖS	7
7. NETESTUOJAMOS SAVYBĖS	8
8. TAKTIKA / PRIĖJIMAS	9
9. SĖKMINGO / NESĖKMINGO TESTAVIMO KRITERIJAI	10
10. SUSTABDYMO IR PRATĖSIMO KRITERIJAI	11
11. PLANUOJAMI DARBO PRODUKTAI	12
12. LIKUSIOS TESTAVIMO UŽDUOTYS	13
13. REIKALINGOS APLINKOS	14
14. KOMPETENCIJŲ IR MOKYMŲ POREIKIAI	15
15. ATSAKOMYBĖS	16
16. GRAFIKAS	17
17. RIZIKOS IR NENUMATYTI ATVEJAI	18
18. PATVIRTINIMAI	19
19. ŽODYNĖLIS	20

1. Testavimo plano identifikatorius

Šio testavimo plano unikalus ID: PST.TP.1. Versija - 1.1.

Plano autorius: Simonas Mikulis (Programų Sistemos, 3 kursas 5 grupė) Mantas Petrikas (Programų Sistemos, 3 kursas 5 grupė)

Kontaktinė informacija: Elektroninis paštas: simonas.mikulis@mif.stud.vu.lt Elektroninis paštas: mantas.petrikas@mif.stud.vu.lt

2. Dokumentai ir nuorodos į juos

Užduoties pobūdis:

Užduočių valdymo sistema, skirta planuoti ir vykdyti užduotis vadovaujantis „Agile“ metodika Dokumento versija: 1.0

Agile principų specifikacija <http://www.agilemanifesto.org> Dokumento versija: 1.0

Kanban „Lean“ metodo specifikacija <http://www.everydaykanban.com/what-is-kanban>
Dokumento versija: 1.0

3. Įžanga

Šio darbo tikslas sudaryti planą užduočių valdymo sistemos veikimo testavimui atlikti. Pagal šį planą ištestuoti programinę įrangą ir užfiksuoti rastus defektus. Plano lygmuo – „Master“. Planas apima pasirinktos programos testavimą, atitikimą reikalavimams. Biudžeto šiai sistemai nėra, resursai – du žmonės, programos kūrėjai yra šio plano autoriai. Testavimas pagal šį planą bus vykdomas nuo sistemos kūrimo pradžios. Sistemai sukurti bus skirta 20 iteracijų po 2 savaites.

4. Testavimo objektai

Bus testuojama tai kas pateikiama vartotojui naudotis:

- Vartotojo sąsaja
- Sistemos vientisumas naudojantis keliems vartotojams
- Klaidų apdorojimas
- Našumas

5. Programinės įrangos rizikos

Kadangi užduočių valdymo sistema nėra unikalus produktas, reikės sukurti programinę įrangą, kuri savo funkcionalumu bei intuityvia vartotojo sąsaja nukonkuruotų varžoves. Išorinės integracijos, kurios bus naudojamos užduočių valdymo sistemoje, yra nusistovėjusios, todėl labai didelės rizikos nėra, tačiau sistemoje bus sukurti integracijų plėtimo taškai lengvam naujovių įdiegimui. Naudotojų duomenys bus saugomi duomenų bazėje, todėl yra nedidelė, bet egzistuojanti rizika, kad duomenys gali būti prarasti arba nutekinti tretiesiems asmenims.

6. Testuojamos savybės

Pagal defektų atsiradimo tikimybę programinės įrangos veikimo metu, rizikos lygmenys yra standartiškai suskirstyti į:

- Žemas (Ž)
- Vidutinis (V)
- Aukštas (A)

Iš naudotojo pusės bus testuojama:

- Greitaveika (lygis – V)
- Vartotojo sąsajos aiškumas (lygis – A)
- Sistemos patikimumas (lygis – A)
- Vartotojo sąsajos dizainas (lygis – Ž)

7. Netestuojamos savybės

Iš vartotojo pusės nebus testuojamos senesnės programinės įrangos pvz.: Interneto naršyklių, SSH klientų palaikymas. Tai bus daroma dėl to, kad kuriama sistema atitktų naujausius standartus bei gerąsias praktikas. Tuo pačiu nebus švaistomas brangus programos kūrimo laikas, kuris bus panaudotas programinės įrangos kokybės užtikrinimui

8. Taktika / Priėjimas

Pagal „Agile“ metodiką, nustatoma, kad sprintas truks 2 savaites. Kiekvieno sprinto pabaigoje bus įvykdomas demo ir retrospektyvos, kurių metu bus aptariamasi praėjusio sprinto teigiami ir neigiami aspektai. Programinės įrangos kurimo procese bus naudojama TDD praktika. Kadangi prie sistemos dirbs tik du žmonės kasdieniniai susitikimai bus pakeisti pokalbiais prie kavos. Už programos testų rašymą bus atsakingi programuotojai.

Sistemos kūrimui naudotina programinė įranga: IntelliJ IDEA Ultimate, Git, MongoDB, JavaScript, Windows 10 bei Ubuntu operacinės sistemos. Įrankiai sistemoms kūrėjams gerai žinomi ir išmanomi, todėl papildomų mokymų nereikės.

Programinės įrangos kurimo procese bus renkamos šios metrikos: funkcijų sudėtingumas, kodo padengimas testais. Sistemos konfigūracija bus valdoma konfigūraciniais failais. Kūrimo, testavimo ir produkcinėms aplinkoms bus sukurtos atskiros konfigūracijos, bus testuojama visose aplinkose. Sukurta sistema bus testuojama šiomis naršyklėmis: Mozilla Firefox, Google Chrome, Safari. Išleidus naują programos versiją bus atliekami regresiniai testai sistemos našumui, patikimumui ir ankstesnės sistemos versijos funkcionalumui. Naujam funkcionalumui testuoti bus rašomi vienetų testai.

Defektų valdymo proceso aprašas:

Testavimo metu randami defektai bus kaupiami ir registruojami. Kiekvienam defektui suteikiamas unikalus ID, jo kilimui nustatyti ir taisyti bus sudaromas defekto aprašas, poveikio lygmuo, testavimo atvejo ID, kurio metu defektas užfiksuotas, bei norima programos elgsena.

Pagal poveikį programinės įrangos veikimui, defektai yra suskirstyti į:

- Svarbus – defektas trukdo programai teisingai atlikti esmines funkcijas. Kilus šio tipo programos veikimo sutrikimui, programa dalimi atvejų veiks kitaip negu numatyta reikalavimuose, gali susigadinti dalis programos išsaugotų duomenų.
- Vidutinis – defektas trukdo programai teisingai atlikti pagalbines funkcijas. Kilus šio tipo programos veikimo sutrikimui, programa dalimi atvejų gali veikti neefektyviai, lėčiau negu turėtų arba užduotį galima įgyvendinti kitu scenarijumi.
- Žemas – defektas netrukdo vartotojui atlikti veiksmų, bet neatitinka programinės įrangos ar vartotojo lūkesčių specifikacijos.

9. Sėkmingo / Nesėkmingo testavimo kriterijai

Testavimo atvejis yra laikomas sėkmingu, kai jo vykdymo metu nekyla jokie defektai, tikimasis atvejo rezultatas sutampa su atvejo vykdymo rezultatu. Kitu atveju, laikoma nesėkmingu, fiksuojamas defektas.

Testavimo etapas laikomas sėkmingu, kai yra įvykdomi visi (100%) testavimo atvejai, 80% kodo padengta testais.

Projekto testavimas laikomas sėkmingu, kai yra pilnai įvykdyti visų etapų testavimo planai, neištaisytų defektų yra ne daugiau kaip 10% ir jų kritiškumas nėra svarbus. Kitu atveju, laikoma nesėkmingu, toliau šalinami aptikti defektai, vykdomi etapų testavimo planai.

10. Sustabdymo ir pratęsimo kriterijai

Stabdymas daromas, jei defektų kiekis testavimo etapo metu viršija 30% lyginant su testavimo atvejų kiekiu. Testavimas pratęsiamas jei defektų kiekis neviršija 10% visų testavimo atvejų.

11. Planuojami darbo produktai

Darbo metu planuojami:

- Sudarytas testavimo planas
- Sukurti testavimo atvejai, padengiantys bent 80% kodo ir visus testavimo objektus.
- Testavimo atvejų vykdymo, defektų aptikimo ir jų šalinimo rekomendacijų analizės žurnalas

12. Likusios testavimo užduotys

Testavimą vykdo 2 asmenys, kurie ir kuria testavimo planą bei sistemą, todėl likusias užduotis jie ir vykdys:

- Sukurti testavimo atvejus
- Įgyvendinti testavimo atvejus
- Užfiksuoti ir aprašyti aptiktus defektus
- Aprašyti defektų šalinimo rekomendacijas ir sistemos elgseną pašalinus defektus

13. Reikalingos aplinkos

Testavimui atlikti reikalingos aplinkos yra: operacinė aplinka: Windows 10, MacOS arba Linux distribucija, naršyklė Mozilla Firefox, Google Chrome arba Safari. Testavimui skirti failai pasiekiami per Git versijavimo sistemą.

14. Kompetencijų ir mokymų poreikiai

Sistemos kūrimui ir testavimui atlikti reikalinga žinoti ir mokėti dirbti su Java 9, Spring Framework, JUnit, MongoDB, JavaScript ES6, React 16, Karma, Mocha, Redux, React-router, Postman, Apache JMeter, Selenium Webdriver. Testuojamos sistemos ir įrankių naudojimo mokymai nebus organizuojami, nes jiems organizuoti nepakanka lėšų, individualiam tobulėjumui galima naudoti PluralSight.

15. Atsakomybės

Prie sistemos kūrimo ir testavimo prisidės du žmonės, jie bus atsakingi už visus pateikiamus kriterijus:

- Rizikų valdymą
- Testuojamų ir netestuojamų savybių parinkimą
- Testavimo taktiką
- Testavimo etapų nustatymą ir aprašymą
- Testavimo atvejų sukūrimą
- Aplinkų paruošimą testavimui
- Grafiko problemų sprendimą
- Kritinius sprendimus, neapibrėžtus testavimo plane
- Testavimo rezultatų rinkimą ir registravimą

16. Grafikas

Kadangi sistemos kurimui numatytos 40 savaitių, naujų funkcijų kūrimas turėtų pasibaigti 36 savaitę. Likusios savaitės bus skirtos galutiniam sistemos testavimui ir defektų šalinimui. Jeigu kritinės funkcijos nebus pabaigtos iki funkcijų kūrimo pabaigos, 2 savaitės gali būti skirtos šių funkcijų užbaigimui, lygiagrečiai vykdant sistemos testavimą. Anksčiau nei suplanuota pabaigus programavimo darbus, galutinį testavimą galima pradėti anksčiau.

- Iš užsakovo gautų pradinių reikalavimų peržiūra.
- Detaliosios reikalavimų specifikacijos sudarymas, pradedamas ne vėliau kaip diena po pradinių reikalavimų peržiūros.
- „Master“ lygmens testavimo plano sudarymas, pradedamas ne vėliau kaip po reikalavimų specifikacijos sudarymo.
- Testavimo etapų planavimas ir specifikavimas, pradedamas po pagrindinio testavimo plano sudarymo.
- Testavimo atvejų kūrimas ir analizė, pradedamas ne vėliau kaip diena po visų testavimo planų sudarymo.
- Vienetų testavimo vykdymas, daromas sistemos surinkimo metu ir po kiekvienos programavimo iteracijos.
- Integracijos, sistemos ir priėmimo testavimo vykdymas, diena po sistemos surinkimo pabaigos.
- Rezultatų rinkimas, registravimas ir kaupimas, vykdomas testavimo metu.
- Rezultatų vertinimas ir ataskaitų rengimas, pradedamas po rezultatų surinkimo pabaigos.

Nukrypęs nuo grafiko spartinimas grafiko vykdymas. Žingsniai, kurie turėtų būti pradėti po vėluojančio procesų pabaigos, pagal galimybes pradedami daryti lygiagrečiai. Stengiamasi išlaikyti nukrypimą kuo mažesnį, bei nenukrypti nuo grafiko dar labiau su tolimesnių testavimo proceso žingsnių vykdymu.

17. Rizikos ir nenumatyti atvejai

Jeigu dėl nenumatytų aplinkymų trūks ar nebus žmonių gebančių atlikti sistemos kūrimo ar testavimo darbus, suderinus su užsakovu paliekama galimybė darbus sustabdyti ar atidėti, kol bus gauta pakankamai žmogiškųjų išteklių sistemai pabaigti.

Jeigu trūks įrankių ar duomenų testi sistemos kūrimo ar testavimo darbus ir neturint galimybės per 2 savaites gauti visus reikiamas resursus, gali būti mažinama sistemos apimtis, sistemoje paliekant plėtimo galimybes pilnam reikalavimų įgyvendinimui vėlesnėse iteracijose, kai bus gauta pakankamai resursų pilnai įgyvendinti reikalavimus.

Jeigu keisis reikalavimai jau suderintoms projekto dalims, priklausomai nuo pakeitimų dydžio, gali būti atsisakyta dalies dar neįgyvendintų funkcijų. Tai daroma norint užtikrinti kad darbai bus pabaigti laiku ir būtų užtikrintas sistemos atitikimas visiems pasikeitusiems reikalavimams.

18. Patvirtinimai

Projekto korektiškumą ir išbaigtumą kiekvieno sprinto pabaigoje patvirtins programuotojų ir testuotojų susitikimas. Pagal programos kurėjų ir užsakovų poreikius, bet ne rečiau kaip kas mėnesį, bus organizuojami sistemos funkcionalumo pristatymai dalyvaujant užsakovams ir sistemos kūrėjams, kurių metu bus verfikuojamas ir validuojamas sukurtas funkcionalumas, derinami pakeitimai.

19. Žodynėlis

Sąvokos:

- „Master“ lygmuo – Testavimo plano lygmuo, nurodantis, kaip bus vykdomas visas testavimo planas bei kad bus kuriami testavimo etapų planai.
- „Judrioji“ metodika – programų sistemų kūrimo praktika, kai apibrėžtas produkto kūrimo kaina ir trukmė, bet dėl galimų reikalavimų pakitimų neapibrėžtas rezultatas. Paprastai produktas kuriamas ir diegimas etapais, kiekvieno metu prisitaikant prie paskitusių reikalavimų.
- Sprintas – viena fiksuoto laiko darbų iteracija.
- „Lean“ metodika – sunaudojant kuo mažiau resursų, pateikti kuo geresnį produktą.

Sutrumpinimai:

- Rizikų lygmenys:
 - Ž – žemas
 - V – vidutinis
 - A – aukštas