Testavimo ataskaita
Komanda 1

Turinys

1. Tikslas

Šio dokumento tikslas – pateikti išsamią “Mokyklos tvarkaraščių kūrimo” programos testavimo ataskaitą. Šiame dokumente paaiškinama programos strategija, eiga ir rezultatai.

2. Programos apžvalga

Mokyklos tvarkaraščių kūrimo programa yra įrankis, skirtas planuoti, sukurti, stebėti ir perplanuoti tvarkaraščius. Programa yra skirta Vilnius Techin mokymo įstaigos vadovui, sudarinėjančiam paskaitų tvarkaraščius kiekvienai grupei, pagal besikeičiantį grupių kiekį bei samdomų dėstytojų prieinamumą. Naudojantis tvarkaraščių kūrimo programa, tvarkaraščių kūrimo procesas tampa efektyvesnis, greitesnis, tvarkaraščiai yra pateikiami patogioje formoje. Taip pat, programa leidžia optimizuoti sudarytą tvarkaraštį minimizuojant "langus" bei efektyviai išnaudojant resursus. Priešingai nei tvarkaraštį sudarant rankiniu būdu Excel platformoje, turinčioje minimalias validacijas, šis produktas leidžia automatiškai validuoti užduotus apribojimus, eliminuoja klaidų tikimybę, yra sutaupoma laiko.

Programoje yra 8 klasifikatoriai, kurie buvo ištestuoti: Tvarkaraščiai, Mokytojai, Grupės, Moduliai, Dalykai, Pamainos, Klasės ir Programos. Šių klasifikatorių testavimo strategija, eiga ir rezultatai aprašyti toliau dokumente.

3. Testavimo apimtis

Testavimas apėmė

1. Funkcinį testavimą išvardintiems moduliams:
    - Tvarkaraščiai
    - Mokytojai
    - Grupės
    - Moduliai
    - Dalykai
    - Pamainos
    - Klasės
    - Programos

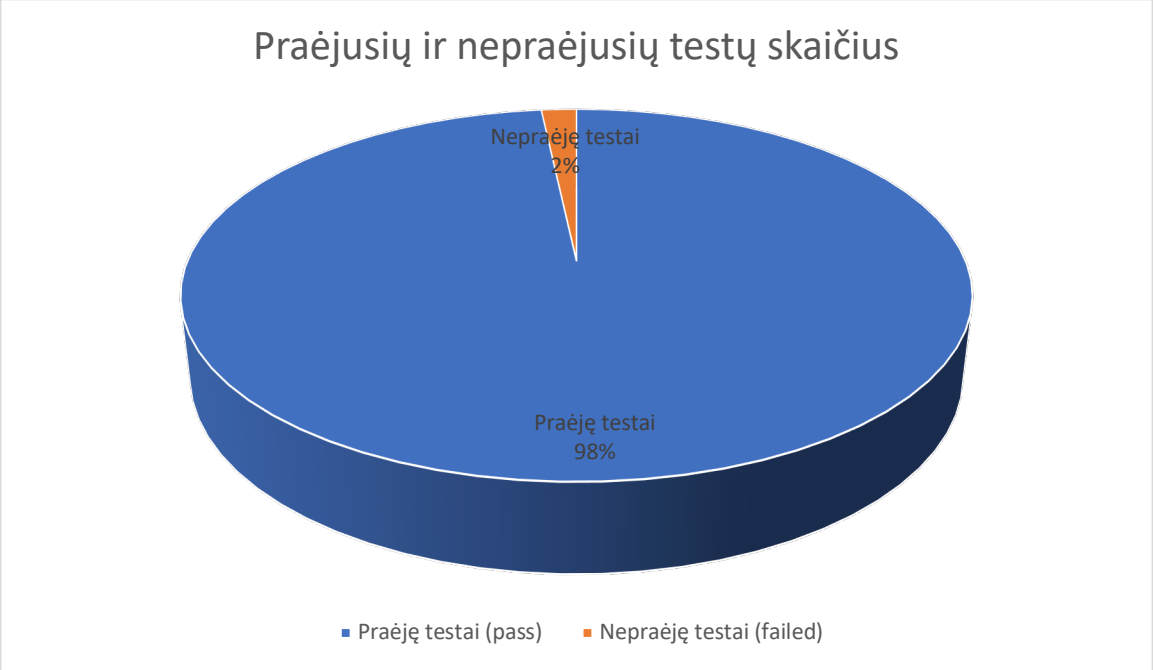
2. Nefunkcinį testavimą:
    - Suderinamumo (*compatibility*) – Mac OS ir Windows operacinėse sistemose, Microsoft Edge ir Safari interneto naršyklėse.
    - Lokalizacijos (*localizations*) – lietuvių kalba.

4. Metrikos

1. Parašytų ir įvykdytų testavimo scenarijų skaičius
2. Praėjusių ir nepraėjusių testų skaičius

Parašyti testavimo scenarijai	Įvykdyti testavimo scenarijai	Praėję testai (*pass*)	Nepraėję testai (*failed*)
234	234	230	6

Lentelė 1 - Parašytų ir įvykdytų testavimo scenarijų skaičius, praėjusių ir nepraėjusių testų skaičius

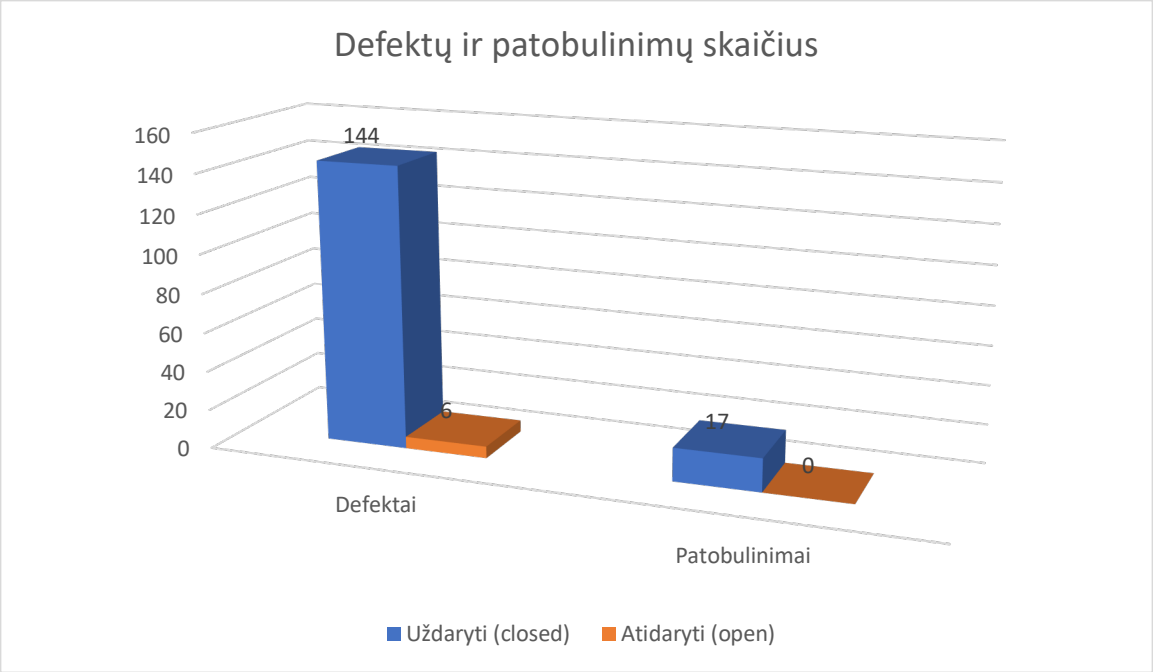


Pav 1 – Praėjusių ir nepraėjusių testų dalis

3. Defektų (*bugs*) ir patobulinimų (*improvements*) skaičius, jų statusas

	Defektai	Patobulinimai
Uždaryti (closed)	144	17
Atidaryti (open)	6	0

Lentelė 2 - Defektų (*bugs*) ir patobulinimų (*improvements*) skaičius, jų statusas

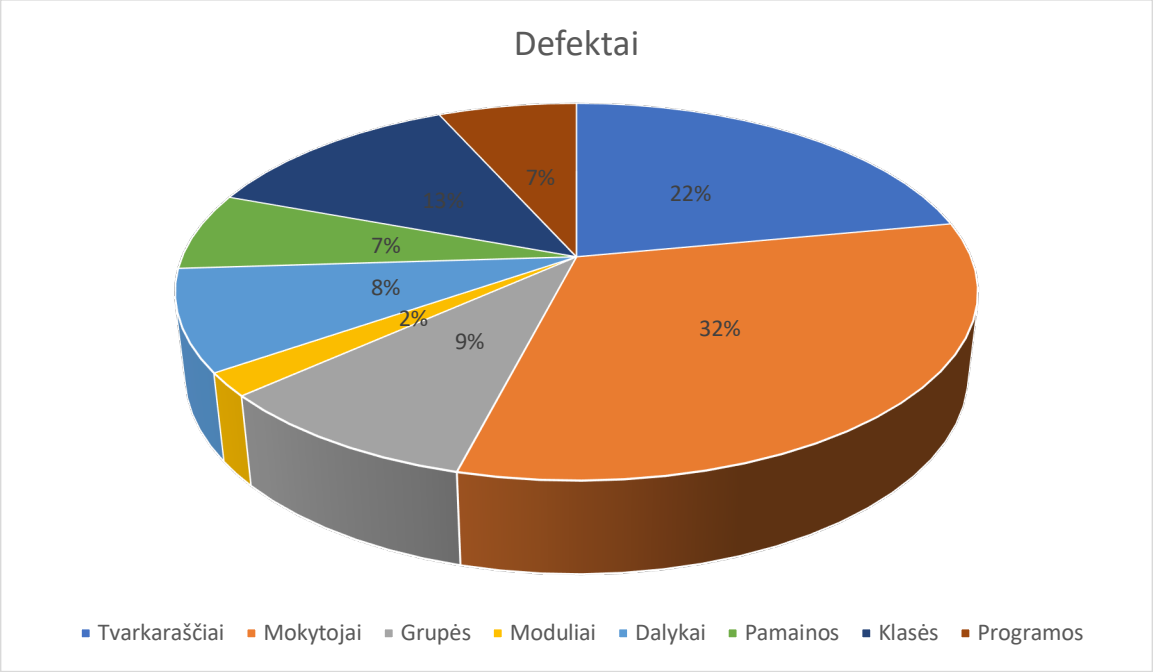


Pav 2 – Defektų ir patobulinimų skaičius

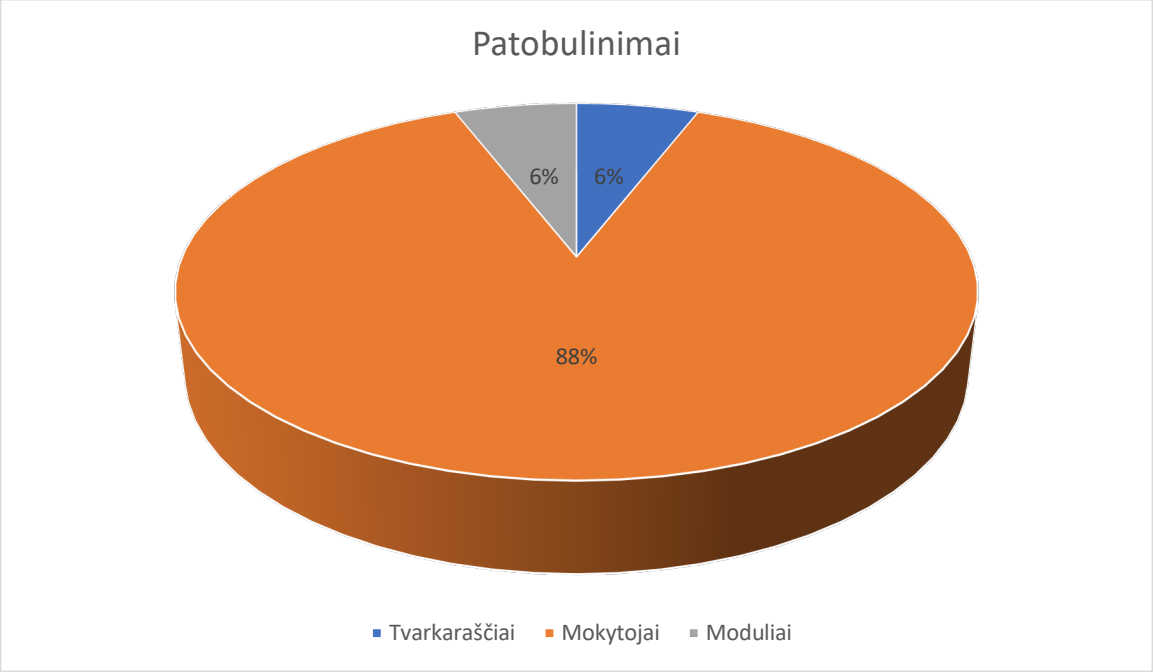
4. Defektų ir patobulinimų pasiskirstymas tarp klasifikatorių

	Tvarkaraščiai	Mokytojai	Grupės	Moduliai	Dalykai	Pamainos	Klasės	Programos	Viso
Defektai	33	48	14	3	13	10	19	10	150
Patobulinimai	1	15	0	1	0	0	0	0	17

Lentelė 3 - Defektų ir patobulinimų pasiskirstymas tarp klasifikatorių



Pav 3 – Defektų pasiskirstymas tarp klasifikatorių



Pav 4 – Patobulinimų pasiskirstymas tarp klasifikatorių

5. Atlikto testavimo tipai

1. Smoke testavimas (*Smoke testing*)
   Šis testavimas buvo atliekamas kiekvieną kartą, kai buvo gaunama nauja versija (*build*), siekiant įsitikinti, kad pagrindinės funkcijos veikia gerai ir testavimą galima pradėti. Siekiant pagreitinti testavimo procesą, smoke testavimo scenarijai buvo automatizuoti. Automatizuoti smoke scenarijai epėmė šiuos klasifikatorius: Klasės, Dalykai, Mokytojai, Moduliai, Programos, Pamainos. Parametrizuoti mokytojų automatiniai testai.
2. Regresinis testavimas (*Regression tesing*)
   • Regresinis testavimas buvo atliekamas kiekvieną kartą, kai buvo gaunama nauja versija (*build*), kurioje yra defektų pataisymų ir naujų patobulinimų, jei tokių yra.
   • Regresinis buvo testavimas atliekamas visai programai, o ne tik naujoms funkcijoms ir defektų pataisymui.
   • Šis testavimas užtikrina, kad esamos funkcijos gerai veiktų ištaisius defektus ir pridėjus naujų patobulinimų prie esamos programos.
   • Testavimo scenarijai (*test cases*) naujoms funkcijoms buvo pridedami prie esamų testavimo scenarijų ir įvykdomi.

6. Testavimo aplinka ir įrankiai

Programos URL	https://schedule-maker-production.up.railway.app/schedule-maker/#/
Serveris	127.0.0.1
Duomenų bazė	H2

7. Gerosios praktikos

Kaskart rankiniu būdu vykdyti testavimo scenarijus (pasikartojančias užduotis) užtrukdavo daug laiko. Todėl smoke testavimo scenarijai buvo automatizuoti, kas sutaupė laiko ir išteklių.

8. Baigiamosios nuostatos

1. Visi sukurti testavimo scenarijai yra įvykdyti.
2. Visi kritiniai, dideli, vidutinio sunkumo defektai patikrinti, dauguma jų uždaryti.
3. Mažiau reikšmingiems defektams – parengtas ištaisymo veiksmų planas.

9. Išvados

Kadangi ne visi kriterijai buvo įvykdyti ir patenkinti, kaip nurodyta 8 skirsnyje, šią programą testavimo komanda siūlo patobulinti prieš išleidimą. Taip pat, turėtų būti atliktas atitinkamas naudotojo/verslo priėmimo (*acceptance*) testavimas.