



Kauno technologijos universitetas

Informatikos fakultetas

**Unifikuota sistema automatiniam aparatinės įrangos
testavimui gamybos linijoje**

Baigiamasis magistro projektas

Audrius Andrijaitis

Projekto autorius

Doc. Šarūnas Packevičius

Vadovas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Unifikuota sistema automatiniam aparatinės įrangos testavimui gamybos linijoje

Baigiamasis magistro projektas

Programų sistemų inžinerija (6211BX011)

Audrius Andrijaitis

Projekto autorius

Doc. Šarūnas Packevičius

Vadovas

Doc. Tomas Blažauskas

Recenzentas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Audrius Andrijaitis

Unifikuota sistema automatiniam aparatinės įrangos testavimui gamybos linijoje

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Audrius Andrijaitis

Patvirtinta elektroniniu būdu



Kauno technologijos universitetas

Informatikos fakultetas

Baigiamojo magistro projekto užduotis

Projekto tema

Unifikuota sistema automatiniam aparatinės įrangos testavimui
gamybos linijoje

Reikalavimai ir sąlygos
(tikslinti pavadinimą
pagal poreikį)

Vadovas / Vadovė

(vadovo pareigos, vardas, pavardė, parašas)

(data)

Andrijaitis Audrius. Unifikuota sistema automatiniam aparatinės įrangos testavimui gamybos linijoje. Magistro baigiamojo projekto vadovas doc. Šarūnas Packevičius; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Programų sistemos.

Reikšminiai žodžiai: automatinio testavimo sistema, aparatinės įrangos testavimas, aparatinės įrangos funkcinis testavimas, aparatinės įrangos automatinis testavimas.

Kaunas, 2022. 50 p.

Santrauka

Darbe pristatoma unifikuota sistema automatiniam aparatinės įrangos testavimui gamybos linijoje. Analitinėje dalyje apžvelgiama sistemos taikymo sritis, jos tikslas, pagrindumas. Analizuojami automatinių bandymų metodai, rinkoje egzistuojančios sistemos. Projektinėje dalyje pristatoma veiklos sudėtis ir veiklos kontekstas, kurioje sistema realizuojama. Apžvelgiami iškelti reikalavimai, sistemos statinis vaizdas. Tyrimo dalyje analizuojama sukurtos sistemos kokybė, apžvelgiamas kokybės vertinimo procesas, tiriamas sistemos prižiūrimumas ir sudėtingumas, suformuojamos sistemos tobulinimo rekomendacijos. Eksperimentinėje dalyje pristatomi du atlikti tyrimai, kuriais siekiama nustatyti kaip sistema paspartina įmonės produkcijos testavimą ir kaip sistema optimizuoja inžinierių kuriančių gaminių testavimo įrankius darbą.

Andrijaitis Audrius. Unified Test Bench System for Automated Hardware Testing on the Production Line. Master's Final Degree Project supervisor doc. Šarūnas Packevičius; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Program systems.

Keywords: automated hardware testing, test bench systems, hardware defect-based testing, hardware functional testing.

Kaunas, 2022. 50 p.

Summary

This paper describes a unified test bench system for automated hardware testing on the production line. The analytical part reviews the scope, purpose, and validity of the system. Automated test bench methods and existing systems on the market are analysed. The project part presents the composition and the context of the activity in which the system is implemented. Requirements and static view of the system are reviewed. The research part analyses the quality of the developed system, reviews the quality assessment process, examines the maintainability and complexity of the system, and formulates recommendations for system improvement. In the experimental part, two studies are presented to determine how the system accelerates the testing of the company's products and how the system optimizes the work of engineers developing product testing tools.

Turinys

Lentelių sąrašas	9
Paveikslų sąrašas	10
Santrumpų ir terminų sąrašas	11
Įvadas.....	12
1. Analitinė dalis	13
1.1. Tikslas.....	13
1.2. Srities apžvalga.....	14
1.2.1. Aparatinės įrangos automatinių bandymų metodai	14
1.2.2. Rinkoje egzistuojančios testavimo sistemos	20
1.3. Pasirinkto testavimo metodo pagrindimas.....	21
1.3.1. Užsakovo reikalavimai, lemiantys metodo pasirinkimą.....	21
1.3.2. Pasirinkto testavimo metodo tinkamumo pagrindimas	21
1.3.3. Papildomi faktoriai	21
1.4. Egzistuojančių rinkoje metodų bei sprendimų palyginimas.....	21
1.5. Įgyvendinimo problemos.....	22
2. Projektinė dalis	23
2.1. Veiklos sudėtis.....	23
2.1.1. Veiklos kontekstas.....	23
2.1.2. Veiklos padalinimas	23
2.2. Sistemos sudėtis (panaudojimo atvejų modelis).....	24
2.3. Diegimo aplinka	24
2.4. Sistemai keliami reikalavimai	25
2.4.1. Funkciniai reikalavimai	25
2.4.2. Svarbiausi nefunkciniai reikalavimai	25
2.5. Sistemos statinis vaizdas	25
2.5.1. Apžvalga.....	25
2.5.2. Paketų detalizavimas	27
2.6. Išdėstymo (deployment) vaizdas	29
2.7. Duomenų vaizdas	30
3. Tyrimo dalis	31
3.1. Sukurtos sistemos kokybės analizės tikslai	31
3.1.1. Aptikti klaidas funkcionavime, logikoje, realizacijoje.....	31
3.1.2. Patikrinti ar programų sistema atitinka reikalavimų specifikaciją	31
3.1.3. Įsitikinti, ar programų sistema sukurta pagal standartus	31
3.1.4. Suformuoti sistemos tobulinimo rekomendacijas	31
3.2. Kokybės vertinimo procesas.....	31
3.2.1. Peržiūros	31
3.2.2. Formalios techninės peržiūros.....	32
3.2.3. Prižiūrimumo ir sudėtingumo charakteristikų tyrimas	32
3.3. Vertinimo rezultatai	33
3.3.1. Kokybės įvertinimas.....	33
3.3.2. Prižiūrimumo ir sudėtingumo charakteristikų tyrimo rezultatai	35

4. Eksperimentinė dalis	39
4.1. Įvadas.....	39
4.2. Pirmasis tyrimas	39
4.2.1. Tyrimo aprašas	39
4.2.2. Tyrimo prielaidos	39
4.2.3. Tyrimo duomenys.....	40
4.2.4. Tyrimo rezultatai	44
4.2.5. Tyrimo išvados	45
4.3. Antrasis tyrimas.....	45
4.3.1. Tyrimo aprašas	45
4.3.2. Tyrimo prielaidos	46
4.3.3. Tyrimo duomenys.....	46
4.3.4. Tyrimo rezultatai	47
4.3.5. Tyrimo išvados	47
Išvados	48
Literatūros sąrašas	49
Priedai.....	50
1 Priedas. Programų sistemos perdavimo ir aprobavimo aktas	50

Lentelių sąrašas

1 lentelė Testavimo metodų palyginimas.....	21
2 lentelė Rinkoje kuriamų sprendimų palyginimas.....	22
3 lentelė Veiklos įvykių sąrašas	23
4 lentelė Apklausos anketa.....	32
5 lentelė Vertinimas iš kūrėjo perspektyvos	34
6 lentelė Vertinimas iš naudotojo perspektyvos	34
7 lentelė Vartotojų apklausos rezultatai	35
8 lentelė <i>RUT950</i> produkto pirmos operacijos statistinės vertės	40
9 lentelė <i>RUT950</i> produkto antros operacijos (visi bandymai) statistinės vertės	41
10 lentelė <i>RUT950</i> produkto antros operacijos (sėkmingi bandymai) statistinės vertės	42
11 lentelė Konsoliduoti <i>RUT950</i> produkto testavimo operacijos (visi bandymai) statistiniai duomenys.....	42
12 lentelė Konsoliduoti <i>RUT950</i> produkto testavimo operacijos (sėkmingi bandymai) statistiniai duomenys.....	43
13 lentelė <i>RUT360</i> produkto testavimo operacijos (visi bandymai) statistinės vertės	43
14 lentelė <i>RUT360</i> produkto testavimo operacijos (sėkmingi bandymai) statistinės vertės.....	44
15 lentelė <i>RUT950</i> ir <i>RUT360</i> produktų testavimo operacijos (visi bandymai) atlikimo trukmių palyginimas.....	44
16 lentelė <i>RUT950</i> ir <i>RUT360</i> produktų testavimo operacijos (sėkmingi bandymai) atlikimo trukmių palyginimas.....	45
17 lentelė Užduočių trukmių palyginimas	47

Paveikslų sąrašas

1 pav. Bendras AOI sistemų veikimo principas [3]	14
2 pav. AOI sistemos iliustracija.	15
3 pav. Supaprastinta FCT veikimo schema.....	16
4 pav. Funkcinio testavimo sistemos iliustracija.....	17
5 pav. Ribinio skenavimo pavyzdys	18
6 pav. JTAG sistemos iliustracija	18
7 pav. ICT blokinė schema.....	19
8 pav. DUT prijungto prie testavimo adatų iliustracija.....	19
9 pav. Konteksto modelis.....	23
10 pav. PA modelis	24
11 pav. Diegimo diagrama	24
12 pav. Komponentų diagrama	26
13 pav. Sistemos išskaidymo į paketus vaizdas	27
14 pav. Paketų klasių diagrama.....	28
15 pav. Išdėstymo vaizdas.....	29
16 pav. Duomenų bazės modelio vaizdas	30
17 pav. Prižiūrimumo indekso reikšmės pagal klases.....	36
18 pav. Ciklomatinio sudėtingumo reikšmės pagal klases.....	36
19 pav. Metodų klasėje ciklomatinio sudėtingumo sumų reikšmės pagal klases	37
20 pav. Metodų darnos trūkumo reikšmės pagal klases.....	37
21 pav. Eilučių skaičiaus metrikos reikšmės pagal metodus	38
22 pav. RUT950 produkto pirmos operacijos trukmių grafikas.....	40
23 pav. RUT950 produkto antros operacijos (visi bandymai) trukmių grafikas	41
24 pav. RUT950 produkto antros operacijos (sėkmingi bandymai) trukmių grafikas	42
25 pav. RUT360 produkto testavimo operacijos (visi bandymai) trukmių grafikas.....	43
26 pav. RUT360 produkto testavimo operacijos (sėkmingi bandymai) trukmių grafikas	44
27 pav. Unifikuotos automatinio testavimo sistemos papildymo naujo gaminio testavimu užduotis	46
28 pav. Užduoties vykdymui skirto laiko ataskaita	46
29 pav. Automatinio testavimo (sena programinė įranga) kūrimo užduotis.....	46

Santrumpų ir terminų sąrašas

Santrumpos:

AOI (angl. Automatic Optical Inspection) – automatinė optinė inspekcija;

API (angl. Application Programming Interface) – programų sąsaja. Tai sąsaja, kurią suteikia kompiuterinė sistema, biblioteka ar programa tam, kad programuotojas per kitą programą galėtų pasiekti jos funkcionalumą ar apsiektų su ja duomenimis;

ATE (angl. Automated test environment) – automatizuota bandymų aplinka;

DUT (angl. device under test) – testuojamas gaminys;

FCT (angl. Functional testing) – funkcinis testavimas;

FTP (angl. File Transfer Protocol) – standartas failų perdavimui;

HTTP (angl. Hypertext Transfer Protocol) – protokolas, skirtas keistis informacija pasauliniame tinkle;

ICT (angl. In-circuit test) – grandinės bandymas;

JSON (angl. JavaScript Object Notation) – atviro standarto formatas, perduodantis duomenų objektus, sudarytus iš atributo ir reikšmės porų, lengvai skaitomame tekste;

JTAG (angl. Joint Test Action Group) – jungtinė bandymų veiksmų grupė;

PCB (angl. Printed Circuit Boards) – spausdintinė plokštė;

PĮ – programinė įranga;

REST (angl. Representational State Transfer) – architektūrinis stilius skirtas interneto servisų kūrimui;

SMT (angl. Surface-mount technology) – paviršinio montažo technologija;

TCP/IP (angl. Transmission control protocol/Internet Protocol) – standartinis duomenų perdavimo protokolų rinkinys, kurio pagrindu veikia internetas bei daugelis privačių komercinių tinklų;

TFTP (angl. Trivial File Transfer Protocol) – standartas failų perdavimui;

UML (angl. Unified Modelling Language) – grafinė sistemų projektavimo ir specifikacijų kūrimo kalba, skirta projektuoti objektines programų sistemas;

XML (angl. Extensible Markup Language) – bendros paskirties duomenų struktūrų bei jų turinio aprašymo kalba;

Įvadas

Įmonėje *Teltonika Networks* po pirminių naujos gaminių serijos projektavimo etapų tolimesnis žingsnis yra pasiruošimas masinei gamybai. Pradėjus masinę gamybą siekiama užtikrinti gaminamos produkcijos kokybę, tad kiekvienas gaminytis turi būti ištestuotas. Tai daroma siekiant eliminuoti gamybos broką. Taip pat kiekvienas gaminytis turi būti paruoštas naudojimui, turi būti įdiegta mikroprograma, įrašyti unikalūs gamintojo identifikaciniai parametrai, tokie kaip serijinis gaminio numeris, produkto kodas. Įmonė naujas gaminių serijas projektuoja ir paleidžia į gamybą nuolat. Kiekviena gaminių serija yra skirtinga ir jai reikia skirtingos testavimo sistemos. Šiems uždaviniams spręsti kuriama ši sistema.

Darbo tikslas – sukurti unifikuotą programinę įrangą skirtą automatiniam aparatinės įrangos testavimui gamybos linijoje. Ši programinė įranga turėtų efektyviai ir optimaliai aptikti aparatinės įrangos gaminių broką, įdiegti į gaminius programinę įrangą, unikalius gamintojo identifikacinius parametrus. Sistema turi veikti su skirtingomis gaminių serijomis. Tikslui pasiekti iškeliami šie uždaviniai:

1. Atlikti aparatinės įrangos testavimo technologijų analizę.
2. Atlikti reikalavimų analizę.
3. Sudaryti projektą kuriamai sistemai.
4. Realizuoti sistemą.
5. Atlikti sistemos testavimą, kokybės tyrimą.
6. Paruošti sistemos naudotojo vadovą.
7. Įdiegti sistemą įmonėje.

Darbas susideda iš kelių pagrindinių dalių: analitinės dalies, projektinės dalies, tyrimo dalies ir eksperimentinės dalies. Analitinėje dalyje apžvelgiama sistemos taikymo sritis, jos tikslas, pagrindumas. Analizuojami automatiniai bandymų metodai, rinkoje egzistuojančios sistemos. Projektinėje dalyje pristatoma veiklos sudėtis ir veiklos kontekstas, kurioje sistema realizuojama. Apžvelgiami iškelti reikalavimai, sistemos statinis vaizdas. Tyrimo dalyje analizuojama sukurtos sistemos kokybė, apžvelgiamas kokybės vertinimo procesas, tiriamas sistemos prižiūrimumas, sudėtingumas, suformuojamos sistemos tobulinimo rekomendacijos. Eksperimentinėje dalyje pristatomi du atlikti tyrimai, kuriais siekiama nustatyti kaip sistema paspartina įmonės produkcijos testavimą ir kaip sistema optimizuoja inžinierių kuriančių gaminių testavimo įrankius darbą. Darbo pabaigoje pateikiamos išvados.

1. Analitinė dalis

Darbas orientuotas į įmonėje *Teltonika Networks* išskylančias problemas, kai suprojektavus naują gaminių seriją ar atlikus pakeitimus jau egzistuojančioje aparatinėje įrangoje reikia paruošti aparatinės įrangos testavimo įrankius gamybos linijai. Įmonėje suprojektavus naują gaminių seriją (ar atlikus pakeitimus joje) toliau seka gamybos procesas. Gamybos linijoje pagaminus gaminį jį reikia ištestuoti, patikrinti ar veikia visos funkcijos, kurioms turi įtakos aparatinės įrangos brokas, įrašyti programinę įrangą (mikroprogramą), identifikacinius parametrus.

Įmonėje produkcija testuojama skirtingų operacijų metu, todėl produkciją reikia perkėlinėti iš vienos operacijos vykdymo vietos į kitą. Dėl šios priežasties veltui išekvojama daug laiko bei didėja gamybos išlaidos. Taip pat, skirtingų gaminių modelių skirtingoms operacijoms vykdyti yra naudojama labai daug skirtingos programinės įrangos, kurią reikia nuolat prižiūrėti ir tobulinti. Esant tokioms aplinkybėms realu, kad nėra pasiekiamas didžiausias galimas našumas. Įmonei nuolat augant ir parduodant vis daugiau produkcijos, išleidžiant skirtingus produktų modelius reikalingi didesni specialistų prižiūrinčių programinę įrangą resursai. Siekiant didesnio gamybos našumo nedidinant gamybos pajėgumų, reikia kuo daugiau sumažinti operacijų skaičių, eliminuoti nereikalingus produkcijos gabenimus tarp skirtingų darbo vietų ir kuo daugiau padidinti vykdomų operacijų efektyvumą.

Siekama, kad kuriama sistema spręstų minėtus įmonei kylančius iššūkius. Sistema turėtų iki galo patikrinti gaminamos aparatinės įrangos funkcijas (broko prevencija), paruošti aparatinę įrangą tolimesniam naudojimui, o tai apima mikroprogramos įdiegimą, unikalių gamintojo identifikacinių parametrų įrašymą į gaminius. Įmonei svarbu sekti kiekvieno gaminio produkcijos eigą, todėl sistema turėtų kaupti rezultatus apie kiekvieną gaminį ir atliktas operacijas su juo duomenų bazėje. Taip pat, jeigu visuose gaminiuose būtų vienoda programinė įranga, prireiktų daug mažiau kvalifikuotų specialistų jai prižiūrėti ir tobulinti, o tai dar labiau sumažintų įmonės kaštus.

Įmonei augant ir plečiantis, didinat gamybos pajėgumus ir gaminių asortimentą, tokia sistema tampa vis svarbesnė, nes ji gali daryti tiesioginę įtaką gamybos kaštams. Taip pat, svarbu paminėti, kad tokios sistemos kūrimas ir integravimas yra būtinas galvojant apie visišką gamybos proceso automatizavimą ir operatorių eliminavimą iš gamybos proceso. Norint pasiekti kuo didesnę įmonės efektyvumą ir pelną, būtina automatizuoti ir optimizuoti kuo daugiau procesų, automatizuojant sujungti mažesnes sistemas į bendras sistemas, taip priverčiant sistemas dirbti kartu vietoje žmonių.

1.1. Tikslas

Šio darbo tikslas yra sukurti sistemą, kuri visiems įmonėje gaminamiems produktams leistų efektyviai, optimaliai ir patikimai atlikti aparatinės įrangos funkcijų ir elektroninių parametrų testavimą, įrašytų reikiamus duomenis į gaminio atmintį, taip sumažindama testavimo operacijos atlikimo trukmę. Pagrindinis sistemos skirtumas nuo jos pirmtakų yra tai, kad sistema turi būti vienoda visiems įmonės gaminiams, turi būti pritaikyta nuolatiniame plėtimui naujais gaminiais. Tai turi reikalauti kuo mažiau resursų. Sistema taip pat turi sumažinti naujo gaminio paleidimo masinėje gamyboje kaštus, pagreitindama ir supaprastindama šį procesą. Labai svarbu paminėti ir tai, kad įdiegus šią sistemą turėtų sumažėti ir testavimo įrangos palaikymo kaštai, nes nebereikės palaikyti tiek sistemų, kiek yra produktų, bus galima palaikyti tik šią sistemą.

1.2. Srities apžvalga

Ši taikomoji sritis labiausiai analizuojama ir vystoma komercinėje aplinkoje. Tą nulemia tai, jog kiekvienas aparatinės įrangos gamintojas susiduria su problema, kai pagaminus elektronikos įrenginius reikia patikrinti jų kokybę. Gamintojai ieško būdų, kaip tai išspręsti: vieni perka paruoštus sprendimus iš įmonių, kurios juos kuria, kiti gamintojai stengiasi automatinio testavimo sprendimus pasidaryti patys. Dar vienas privalumas yra tai, jog automatizuojant kokybės tikrinimo sistemas, jas įdiegiant galima eliminuoti žmogiškųjų klaidų riziką.

Pasaulinėje rinkoje daugėjant elektronikos gaminių, augant šių prietaisų sudėtingumui jų testavimas tampa sudėtingesnis ir brangesnis [1].

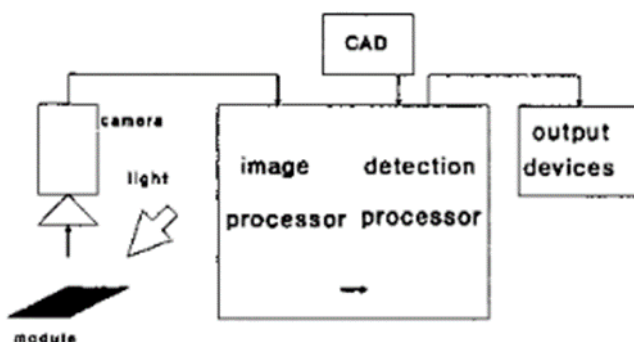
Šiame skyriuje bus apžvelgiami gerai žinomi ir rinkoje naudojami automatinės bandymų įrangos (ATE) tipai ir metodai.

1.2.1. Aparatinės įrangos automatinių bandymų metodai

Optinė inspekcija

Automatinė optinė inspekcija (AOI) yra vienas iš spausdintinių plokščių tikrinimo gamybos metu būdų. Šis tikrinimo būdas dažniausiai naudojamas su paviršinio montažo technologijos (SMT) gamybos būdu. Dėl SMT komponentų mažo dydžio ir tankesnio išsidėstymo gaminant gaminius gamybos procesas greitėja, bet didėja broko, susijusio su litavimo kokybe, tikimybė. Optinės inspekcijos tikrinimo būdas iš principo ir yra orientuotas į litavimo kokybės tikrinimą.

Šio bandymo tipo veikimo modelis yra grįstas tuo, kad pagamintoms PCB kamera autonomiškai daromos nuotraukos, kurios siunčiamos į aptarnaujantįjį kompiuterį, kuriame vykdoma nuotraukų analizė. Aptarnaujantysis kompiuteris analizės metu nustato gedimus ir rezultatus pateikia aptarnaujančiam personalui. Tai gan plačiai naudojamas metodas elektronikos įrenginių gamyboje, nes tai yra bekontaktis testavimo metodas, galintis greitai ir konkrečiai identifikuoti gedimus. Istoriskai AOI sistemos yra naudojamos po komponentų litavimo. Daugiausia todėl, kad AOI sistemos gali patikrinti, ar nėra daugumos defektų (komponentų išdėstymas, nenumatyti lydmetalių susijungimai, trūksta lydmetalių ir kt.) vienoje linijoje su viena sistema. Tokiu būdu iškart nustatomi gedimai, sugedusios plokštės taisomos, o kitos plokštės keliauja į kitą proceso etapą [2].



1 pav. Bendras AOI sistemų veikimo principas [3]

1 paveiksle pateiktą šio testavimo metodo veikimo principo blokinę diagramą galima padalyti į tris svarbiausias dalis: vaizdo atpažinimo, vaizdo apdorojimo ir išvesties įrenginio. Vaizdo atpažinimo dalis tai autonomiškai ir tiksliai virš visos testuojamos plokštės pagal iš anksto sudarytą planą judantis vaizdo jutiklis, kamera, kuri fotografuoja visus (kiek tai yra įmanoma) komponentus, padarydama kuo daugiau nuotraukų iš įvairių krypčių. Visa vaizdinė medžiaga yra perduodama į vaizdo apdorojimo įrenginį. Vaizdo apdorojimo įrenginyje visa vaizdinė medžiaga apdorojama lyginant vaizdo medžiagą su jau turimais duomenimis, pagal iš anksto nustatytas taisykles ar panaudojant kitus vaizdo atpažinimo algoritmus ar metodus. Gauti vaizdo apdorojimo rezultatai perduodami į išvesties įrenginį. Operatorius, atsakingas už defektų šalinimą, gali nuolat stebėti rezultatus, patikrinti problemines vietas, būtinu atveju turi galimybę jas pataisyti. Patogu yra tai, kad sistema gali tiksliai nustatyti įtariamo gedimo vietą.



2 pav. AOI sistemos iliustracija.

2 paveiksle pateikiama realios AOI sistemos iliustracija. Iliustracijoje matyti linijoje esantis gaminys virš kurio yra kamera, kuri fiksuoja komponentus, ir vaizduoklis, kuriame realiu laiku atvaizduojamas vaizdas, kurį fiksuoja kamera.

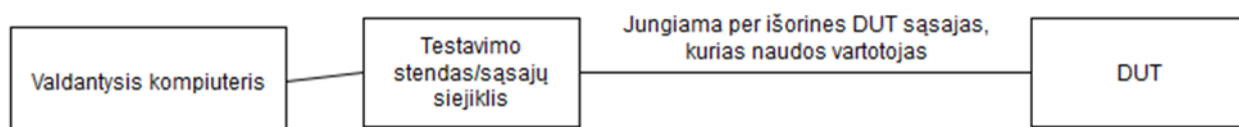
Pagrindiniai šio testavimo tipo trūkumai yra tai, kad toks testavimas gali nustatyti tik kai kuriuos litavimo defektus, bet negali iki galo patikrinti gaminio kokybės. Šiuo metodu galima tikrinti tik SMT pagaminti gaminiai, šiuo metodu tikrinant gaminį negalima į gaminio atmintį įrašyti programinės įrangos. Taip pat, šiam metodui reikalinga brangi įranga.

Funkcinis testavimas

Vienas iš daugiausiai galimybių suteikiantis bei lengvai integruojamas testavimo metodas yra funkcinis testavimas (FCT). Funkcinis testavimas apima visą gaminio funkcionalumo testavimą, patikrinamos visos funkcijos, kurios yra priklausomos nuo aparatinės įrangos. Dažniausiai gaminius testuojant šiuo metodu prie gaminio jungiamasi per išorines gaminio jungtis, siekiant patikrinti jų veikimą, bet pasitaiko ir atvejų, kai prie testuojamo gaminio jungiamasi ir panaudojant testavimo

adatas. Toks prijungimas žymiai greitesnis, bet didžiulis to trūkumas, kad išorinės jungtys lieka nepatikrintos [4].

Pagrindinis šio metodo veikimo principas yra tai, kad testavimo metu testuojamasis gaminys imituoja realų ar iš dalies realų veikimą, tam dažniausiai reikia pasitelkti papildomą aparatinę įrangą, kuri gali veikti kaip sąsajų siejiklis tarp testavimą valdančio kompiuterio ir testuojamo gaminio. 3 paveiksle pateikiama supaprastinta funkcinio testavimo veikimo schema. Schemoje matyti, kad gaminys su testavimo stendu (angl. test board) arba sąsajų siejikliu jungiasi per standartines sąsajas, kurios ateityje bus naudojamos gaminio eksploatacijos metu.



3 pav. Supaprastinta FCT veikimo schema

Testo vykdymo metu iš valdančiojo kompiuterio yra siunčiamos kontrolės komandos į testuojamą gaminį, atliekami įvairūs veiksmai, skaičiavimai testuojamo gaminio viduje, siunčiami fiktyvūs duomenys per visas testuojamojo gaminio sąsajas, siekiant patikrinti sąsajų veikimą. Funkcinio testavimo metu gaminio atmintyje gali būti išsaugomi reikiami duomenys.

Testuojant gaminius šiuo metodu kiekvienam unikaliam testuojamo gaminio modeliui reikia unikalios testavimo įrangos. Todėl kuriant funkcinio testavimo automatizuotas sistemas testavimo sistemų kūrėjams tenka projektuoti testavimo stendus, kurie galėtų apimti visas reikalingas testuojamo gaminio sąsajas arba kombinuoti kelis testavimo stendus, kad būtų galima taip vykdyti testavimą. Šiai problemai spręsti buvo pradėtos kurti moduliųjų testavimo stendų sistemos. Tokios testavimo sistemos iki šių dienų gan plačiai išsivystė ir yra plačiai naudojamos visame pasaulyje. Moduliųjų stendų sistemos turi net specifikuotą architektūrą modulių tarpusavio komunikacijai [5].

Neatsiejama šio testavimo metodo dalis yra programinė įranga. Visuose testavimo metoduose programinė įranga atlieka svarbų vaidmenį, bet šis testavimo metodas šiuo atžvilgiu išsiskiria iš kitų tuo, kad programinė įranga yra susijusi ne tik su testavimo įrangos valdymu, bet ir su konkretais testuojamo gaminio funkcijomis. Dėl šios priežasties tenka kurti sudėtingą į testuojamų gaminių modelių orientuotą programinę įrangą. Žinoma, daugelis funkcinio testavimo stendų / moduliųjų stendų kūrėjų naudoja programinės įrangos platformas, integruotas kūrimo platformas (angl. integrated development environments), kurios žymiai palengvina programavimo darbus [6].

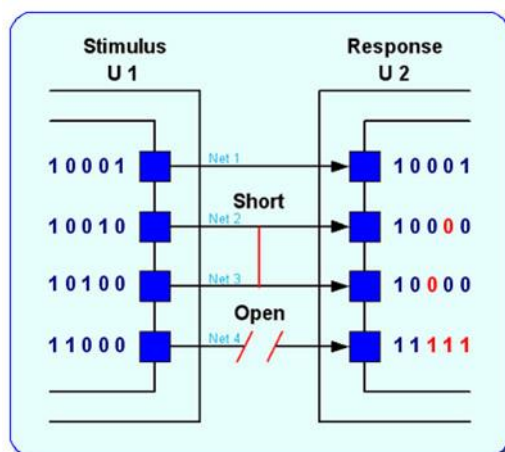
4 paveiksle pateikiama funkcinio testavimo sistemos iliustracija.



4 pav. Funkcinio testavimo sistemos iliustracija

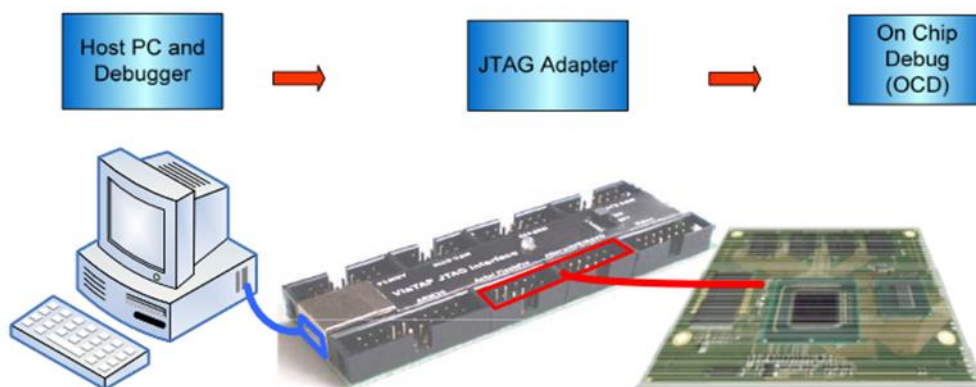
Ribinis skenavimas

Dar vienas metodas, skirtas įrenginių testavimui gamybos linijoje, yra ribinis skenavimas (angl. boundary scan). Šis metodas sukurtas JTAG ir aprašytas kaip IEEE Std 1149.1 standartas [7]. Nuo tada JTAG ir ribinis skenavimas minimi kaip sinonimai. Šis testavimo metodas suteikia galimybę tikrinti gaminio (PCB) viduje esančias įvairias jungtis, pavyzdžiui, tarp pagrindinio procesoriaus ir atminties, tarp įvairių kitų lustų, tarp loginių klasterių nenaudojant fizinių testavimo jungčių prie kiekvieno komponento [7]. Šio testavimo metodo veikimas reikalauja, kad testuojamas gaminy (DUT) būtų paruoštas šiam metodui, kad gaminyje integruoti lustai turėtų JTAG sąsają, kad gaminyje esantis grandynas būtų pritaikytas. Šio metodo veikimas yra grįstas tuo, kad prie testuojamo įrenginio prijungiamas JTAG siejiklis. Iš valdančiojo kompiuterio siunčiamos specialios komandos į testuojamo įrenginio lustus (kurie yra sujungti tarpusavyje grandynais). Lustai atlieka nurodytus veiksmus, siunčia skaitmeninius signalus ir tikrina ar gaunami numatyti rezultatai. Tokiu būdu gali būti patikrinamas visas testuojamos plokštės grandynas. 5 paveiksle pateikiamas ribinio skenavimo pavyzdys, iliustruojant kaip vyksta jungčių tarp lustų tikrinimas ir kaip yra nustatomos klaidos.



5 pav. Ribinio skenavimo pavyzdys

Dar viena funkcija, kurią suteikia JTAG naudojimas gaminio tikrinimo procese, tai gaminio atminties pasiekimas ir galimybė šio tikrinimo metu į gaminio vidinę atmintį įrašyti programinę įrangą [8].



6 pav. JTAG sistemos iliustracija

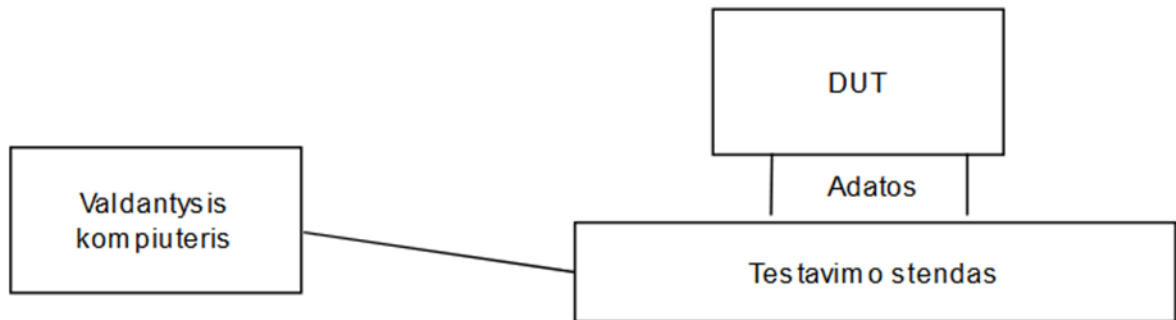
6 paveiksle pavaizduotas pilnas ribinio skenavimo testavimo metodo sprendimas.

Iliustracijoje matyti valdantis kompiuteris, JTAG sietuvas ir testuojamas gaminys. Šiam metodui įgyvendinti to ir užtenka.

Taigi šis testavimo metodas iš principo pigus, nereikalaujantis brangios specifinės įrangos, greit atliekamas, bet svarbu pabrėžti tai, kad šiam metodui būtina, kad testuojamas įrenginys būtų specialiai suprojektuotas ir pritaikytas.

Grandinės testavimas

Šis testavimo metodas (ICT) sukurtas elektriškai tikrinti PCB. Naudojant šį metodą gaminio grandynai yra tikrinami dalimis. Taip pat, yra tikrinami individualūs komponentai arba jų grupės. Tikrinant šiuo metodu yra siekiama patikrinti ar visi komponentai prilituoti teisingai, ar sulituoti tinkami komponentai, ar nėra trumpų jungimų tarp komponentų ir kiti elektroniniai parametrai [9].



7 pav. ICT blokinė schema

7 paveiksle pateikiama supaprastinta ICT sistemos blokinė schema, kuri atskleidžia esminį šios sistemos veikimo principą. DUT yra prijungiamas prie testavimo stendo, naudojant testavimo adatų plokštę (angl. bed of nails), o visą testavimo eigą kontroliuoja valdantysis kompiuteris. Testavimo adatos yra jungiamos prie visų DUT taškų, kurie bus tikrinami. DUT prijungto prie testavimo adatų iliustracija pateikiama 8 paveiksle.



8 pav. DUT prijungto prie testavimo adatų iliustracija

Apibendrinant šį testavimo metodą reiktų paminėti, kad norint testuoti gaminį šiuo metodu gaminys ir sistema turi būti suderinti tarpusavyje, testavimo adatų plokštė projektuojama kiekvienam naujam gaminio modeliui. Tai didina šio metodo savikainą. Taikant šį metodą nebegalimi didesni pakeitimai DUT aparatinėje įrangoje (kitu atveju reikalingi pakeitimai ir testavimo adatų plokštėje). Taikant šį metodą nėra galimybės įrašyti programinės įrangos į DUT atmintį, nebent ši testavimo sistema

kombinuojama su kitomis. Kadangi taikant šį testavimo metodą testavimo adatos jungiasi su daugybe taškų gaminyje, todėl atsiranda galimybė vienoje sistemoje kombinuoti grandinės testavimą, ribinį skenavimą ir funkcinį testavimą [10]. Tai atveria duris ir DUT atminties įrašymui ir daug platesniam bei detalesniam gaminio tikrinimui.

1.2.2. Rinkoje egzistuojančios testavimo sistemos

Šiame skyriuje aptarsiu rinkoje egzistuojančias sistemas, skirtas automatizuotam aparatinės įrangos testavimui. Rinkoje kuriamų sprendimų palyginimas pateiktas 2 lentelėje.

***Teradyne* [11]**

Teradyne – įmonė, kurianti testavimo įrangą elektronikos gamintojams. Įmonė kuria aparatinę ir programinę gaminių testavimui skirtą įrangą, universalius robotus, skirtus gamybos pramonei. *Teradyne* sukurtos sistemos plačiai naudojamos visame pasaulyje, teikia konkurencinį pranašumą pirmaujančioms pasaulio elektronikos kompanijoms. *Teradyne* kuria sistemas, kurios atlieka visiškai automatinį testavimo ciklą arba net visiškai automatinį gaminių testavimą gamybos linijoje. Įmonė suteikia galimybę jų testavimo sistemas tobulinti, pritaikyti savo reikmėms, taikyti su kitomis sistemomis. Įmonės produktai visiškai apima visą gaminių testavimą. Įmonės sukurtų sistemų pritaikymas, sukongūravimas ir paleidimas gamybos procese yra tokie sudėtingi, kad tai atlikti rengiami mokymai. *Teradyne* kuriamos sistemos yra kombinuotos, naudojančios kelis testavimo metodus - grandinių testavimą ir ribinį testavimą. Įmonė kartu su savo testavimo aparatine įranga pateikia ir integruotas programinės įrangos kūrimo platformas, kurias naudojant sistemas galima pritaikyti savo reikmėms, įsidiegti pilną sprendimą. Šios įmonės kuriami sprendimai yra serijiniai ir nėra taikomi konkrečioms klientams.

***Caliber Interconnect* [12]**

Caliber Interconnect – įmonė, kurianti ir diegianti testavimo įrangą elektronikos gamintojams. Įmonė kuria tiek pačią aparatinę testavimo įrangą, tiek programinę įrangą. Įmonė siūlo visiškai automatizuotą testavimo ciklą, operatoriui draugišką vartotojo sąsają, realaus laiko parametrų stebėjimą, testavimo rezultatų saugojimą duomenų bazėje, testavimo rezultatų statistikos sudarymą, grafinį statistikos atvaizdavimą. Testavimo sistema palaiko lygiagrečių įrenginių testavimą, palaiko reikiamos informacijos įrašymą į gaminio atmintį, taip pat yra galimybė konfigūruoti testavimo planą. Sistema neturi galimybės ją pildyti papildomu funkcionalumu, tokiu kaip operatoriaus atliktų operacijų registravimu, taip pat neturi sistemos API, kuri naudojant būtų galima sistemą įdiegti kitoje sistemoje. *Caliber Interconnect* kuria sistemas naudojant funkcinio testavimo metodą. Įmonės kuriami sprendimai nėra serijiniai, jie kuriami kiekvienam klientui atskirai.

***National Instruments* [13]**

National Instruments - rinkoje lyderiaujanti įmonė, kurianti sprendimus elektronikos gamintojams ir laboratorijoms. Įmonė kuria aparatinę ir programinę gaminių testavimui skirtą įrangą. *National Instruments* kuriami sprendimai yra universalūs, gaminami serijiniu būdu. Ši įmonė daugiausia dėmesio skiria funkcinio testavimo įrangai, bet turi sprendimų ir su ribinio testavimo metodu. Kuria modulinės sistemas, kurios remiasi PXI [5] architektūra. Tokios modulinės sistemos leidžia klientams, panaudojant skirtingus modulius, greitai susikurti reikiamą testavimo įrangą, kuri ateityje gali būti pakartotinai panaudota testuojant kitus gaminius.

***ALeader* [14]**

ALeader - viena iš įmonių, kuriančių automatinius optinės inspekcijos sprendimus. Įmonės kuriami produktai visiškai pritaikomi elektronikos gamintojams. Sprendimai gali būti montuojami į bendras gamybos linijas arba būti naudojami atliekant atskiras operacijas. Kartu su automatine optine inspekcija atliekančiomis sistemomis įmonė kuria ir programinę įrangą, kuri visiškai paruošta naudojimui gamyboje. Programinę įrangą telieka tik sukonfigūruoti, kad sistema prisitaikytų prie gaminamų gaminių.

1.3. Pasirinkto testavimo metodo pagrindimas

1.3.1. Užsakovo reikalavimai, lemiantys metodo pasirinkimą

Pagrindinis užsakovo keliamas reikalavimas, kad sukurta sistema galėtų iki galo patikrinti visas gaminio funkcijas, įskaitant jungtis. Užsakovas yra suprojektavęs ir pasigaminęs aparatinę įrangą, kuri gali būti naudojama kaip testavimo stendai, sąsajų sietuvai, todėl papildomos aparatinės įrangos testavimo reikmėms užsakovas pirkti nenori.

1.3.2. Pasirinkto testavimo metodo tinkamumo pagrindimas

Išanalizavus visus rinkoje egzistuojančius testavimo metodus bei užsakovo keliamus reikalavimus, nuspręsta kuriamoje sistemoje panaudoti funkcinį testavimą. Pagrindinis kitų metodų trūkumas yra tai, kad juos naudojant negalima visiškai užtikrinti gaminio kokybės, patikrinti visų gaminio funkcijų. Funkcinio testavimo metodas suteikia galimybę patikrinti visas gaminio funkcijas. Kuriant sistemą bus galima prisitaikyti prie jau turimų užsakovo testavimo stendų, ją bus galima nesunkiai plėsti, pritaikyti naujiems gaminiams.

1.3.3. Papildomi faktoriai

Taip pat, svarbu pabrėžti, kad užsakovas gamybos linijoje jau naudoja optinės inspekcijos testavimo metodą. Kiekvienas pagamintas gaminytis yra tikrinamas su AOI įranga.

1.4. Egzistuojančių rinkoje metodų bei sprendimų palyginimas

1 lentelėje pateikiamas analizuotų testavimo metodų palyginimas.

1 lentelė Testavimo metodų palyginimas

Kriterijus/ Metodas	Optinė inspekcija	Funkcinis testavimas	Ribinis skenavimas	Grandinės skenavimas
Testavimo greitis	Vidutinis	Lėtas	Greitas	Greitas
Galimybė įrašinėti DUT atmintį	Ne	Taip	Taip	Ne
Gaminys pritaikytas testavimui	Ne	Ne	Taip	Taip
Galimybė iki galo patikrinti DUT kokybę	Ne	Taip	Ne	Ne
Galimybė patikrinti išorines DUT jungtis	Ne	Taip	Ne	Ne
Įrangos brangumas	Brangi	Pigi	Pigi	Brangi

Palaikymo brangumas	Pigus	Brangus	Brangus	Vidutinis
DUT jungimas prie testavimo įrangos	Bekontaktis	Per išorines DUT jungtis	Per spec. jungtį	Per testavimo adatas

2 lentelėje pateikiamas analizuotų rinkoje kuriamų sprendimų palyginimas.

2 lentelė Rinkoje kuriamų sprendimų palyginimas

Kriterijus/ Sprendimas	Teradyne	Caliber Interconnect	National Instruments	ALeader
Testavimo metodas	ICT/JTAG	FCT	FCT	AOI
Galimybė pritaikyti su kitomis PĮ sistemomis	Taip	Ne	Taip	Ne
Galimybė diegti į gamybos liniją	Taip	Ne	Taip	Taip
Galimybė iki galo patikrinti DUT kokybę	Ne	Taip	Taip	Ne
Testavimo rezultatų saugojimas duomenų bazėje	Taip, jeigu klientas tai realizuoja	Taip	Taip, jeigu klientas tai realizuoja	Taip
Galimybė į DUT atmintį įrašinėti reikiamus parametrus	Taip	Taip	Taip	Ne
Gaminama serijiniu būdu	Taip	Ne	Taip	Taip

1.5. Įgyvendinimo problemos

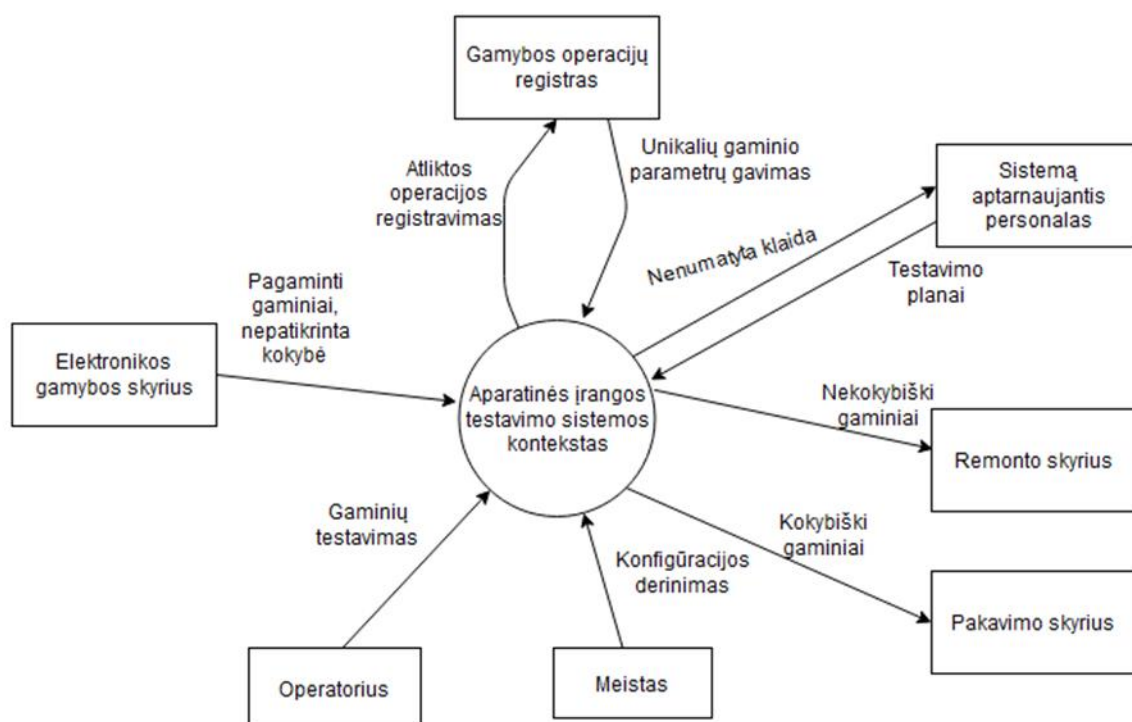
Kuriama sistema turi būti unifikuota ir pritaikyta visų užsakovo gaminamų gaminių testavimui, todėl atrodo, kad daugiausia problemų įgyvendinant turėtų kelti skirtingų testavimo stendų, skirtingų komunikacijos protokolų tarp valdančiojo kompiuterio ir stendo palaikymas, problemos susijusios su aparatinės ir programinės įrangos suderinamumu.

2. Projektinė dalis

2.1. Veiklos sudėtis

2.1.1. Veiklos kontekstas

Šis veiklos konteksto modelis atskleidžia dabartines veiklas, kurias vykdo įmonė ir kurias norima atnaujinti.



9 pav. Konteksto modelis

2.1.2. Veiklos padalinimas

Žemiau esančioje lentelėje pateikiamas veiklos įvykių sąrašas.

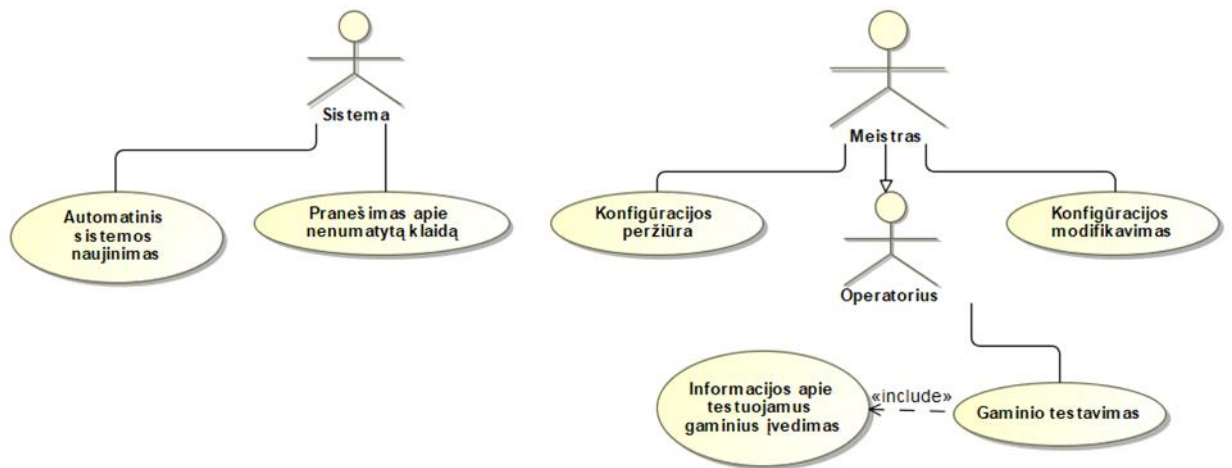
3 lentelė Veiklos įvykių sąrašas

Eil. nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Sistemos konfigūracijos derinimas	Atnaujinta konfigūracija (in)
2	Atliktų operacijų registravimas	Duomenys apie atliktą operaciją (out)
3	Unikalių gaminio parametrų nuskaitymas	Unikalūs gaminio parametrai (in)
4	Gaminių testavimas	Netestuoti gaminiai (in) Ištestuoti gaminiai(out)
5	Testavimo rezultatų saugojimas duomenų bazėje	Testavimo rezultatai (out)

6	Automatinis sistemos naujinimasis	Atnaujinti sistemos komponentai (in)
7	Atsakingų asmenų (sistemos aptarnaujančio personalo) informavimas apie įvykusią nenumatytą klaidą.	Pranešimas apie įvykusią nenumatytą klaidą (out)
8	Testavimo plano įtraukimas	Testavimo planas (in)

2.2. Sistemos sudėtis (panaudojimo atvejų modelis)

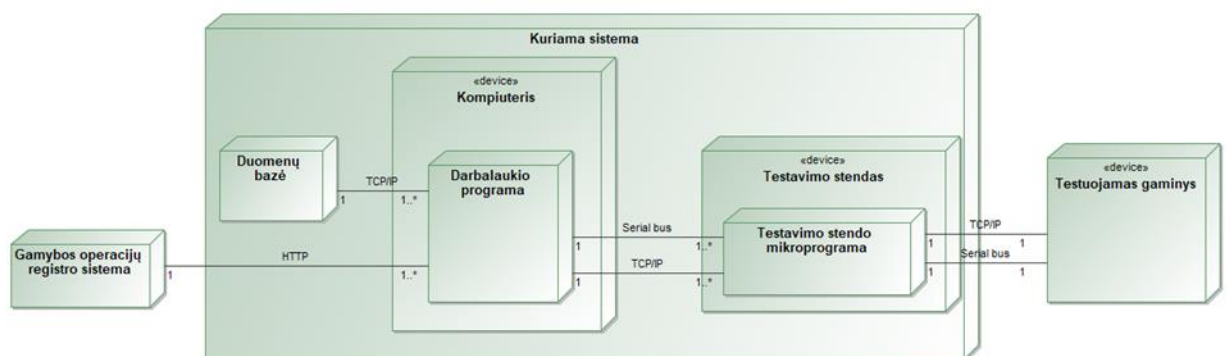
Žemiau pateikiama panaudos atvejų diagrama. Panaudos atvejų vizualiniam atvaizdavimui naudojamas UML panaudos atvejų modelis.



10 pav. PA modelis

2.3. Diegimo aplinka

Žemiau pateikiama sistemos diegimo diagrama.



11 pav. Diegimo diagrama

2.4. Sistemai keliami reikalavimai

2.4.1. Funkciniai reikalavimai

- Sistema turi naujintis automatiškai;
- Sistema turi išsiųsti pranešimus sistemos aptarnaujančiam personalui apie įvykusias nenumatytas klaidas;
- Sistema turi teikti galimybę peržiūrėti sistemos nustatymus;
- Sistema turi teikti galimybę modifikuoti sistemos nustatymus;
- Sistemos nustatymus peržiūrėti gali tik meistrai;
- Sistemos nustatymus modifikuoti gali tik meistrai;
- Prieš pradėdamas darbą operatorius turi įvesti informaciją apie gaminius, kurie bus testuojami;
- Sistema turi vykdyti iš anksto sudarytą, konkrečiam gaminiui skirtą testavimo planą;
- Sistema testavimo rezultatus turi saugoti duomenų bazėje;
- Sistema turi registruoti atliktas gamybinės operacijas;
- Unikalių gaminio parametrų nuskaitymas iš gamybos operacijų registro;

2.4.2. Svarbiausi nefunkciniai reikalavimai

Išvaizdos reikalavimai

Atvaizduojant testavimo rezultatus turi būti naudojamos trys spalvos: raudona – testavimas baigėsi nesėkmingai, žalia – testavimas baigėsi sėkmingai, oranžinė – testavimas vykdomas.

Reikalavimai veikimo sąlygoms

- Sistema turi tinkamai veikti „Windows 10“ operacinę sistemą naudojančiame kompiuteryje;
- Sistema turi būti suderinta su testavimo stendu, kurio mikrokompiuteris yra „Raspberry Pi 4“, o operacinė sistema „OpenWRT“;

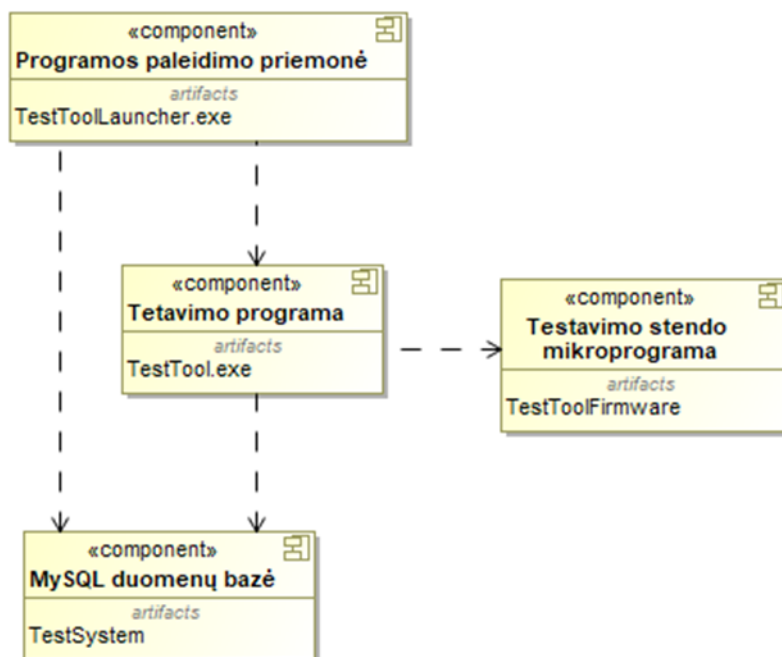
Reikalavimai saugumui

Jokie sistemos komponentai negali būti pasiekiami iš pasaulinio interneto tinklo (angl. wide area network).

2.5. Sistemos statinis vaizdas

2.5.1. Apžvalga

Žemiau pateikiama sistemos komponentų diagrama.



12 pav. Komponentų diagrama

Visa kuriama sistema susideda iš 4 komponentų.

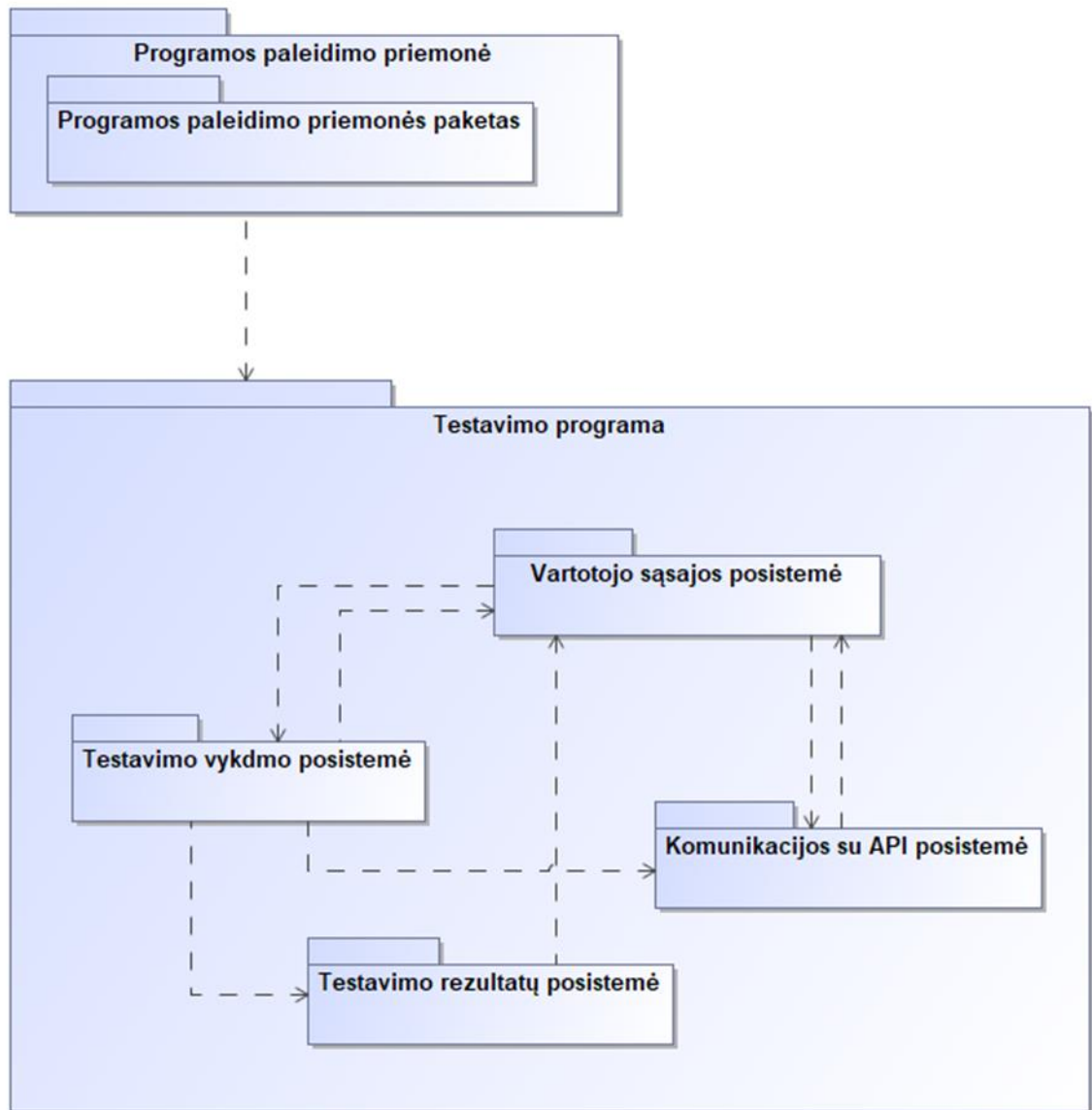
„Programos paleidimo priemonė“ – tai taikomoji programa, kurios tikslas yra patikrinti duomenų bazėje ar yra naujesnė testavimo programos versiją, jei reikia ją parsisiųsti ir paleisti testavimo programą.

„Testavimo programa“ – tai pagrindinė sistemos dalis. Taikomoji programa, kuri atlieką pagrindines sistemos funkcijas, parenka reikiamą testavimo planą, jį vykdo, valdo testavimo stendą, atlikinėja įrašus į duomenų bazę.

„Testavimo stendo mikroprograma“ – tai mikroprograma, kuri įrašinėjama į testavimo stendą. Šios mikroprogramos paskirtis užtikrinti komunikaciją tarp testavimo programos ir testavimo stendo, sėkmingai valdyti testavimo stendo aparatinę įrangą.

„MySQL duomenų bazė“ – tai sistemos duomenų bazė, kurioje kaupiami testavimo rezultatai ir saugomos sistemos komponentų naujesnės versijos.

Taip pat žemiau pateikiamas ir sistemos išskaidymo į paketus vaizdas.

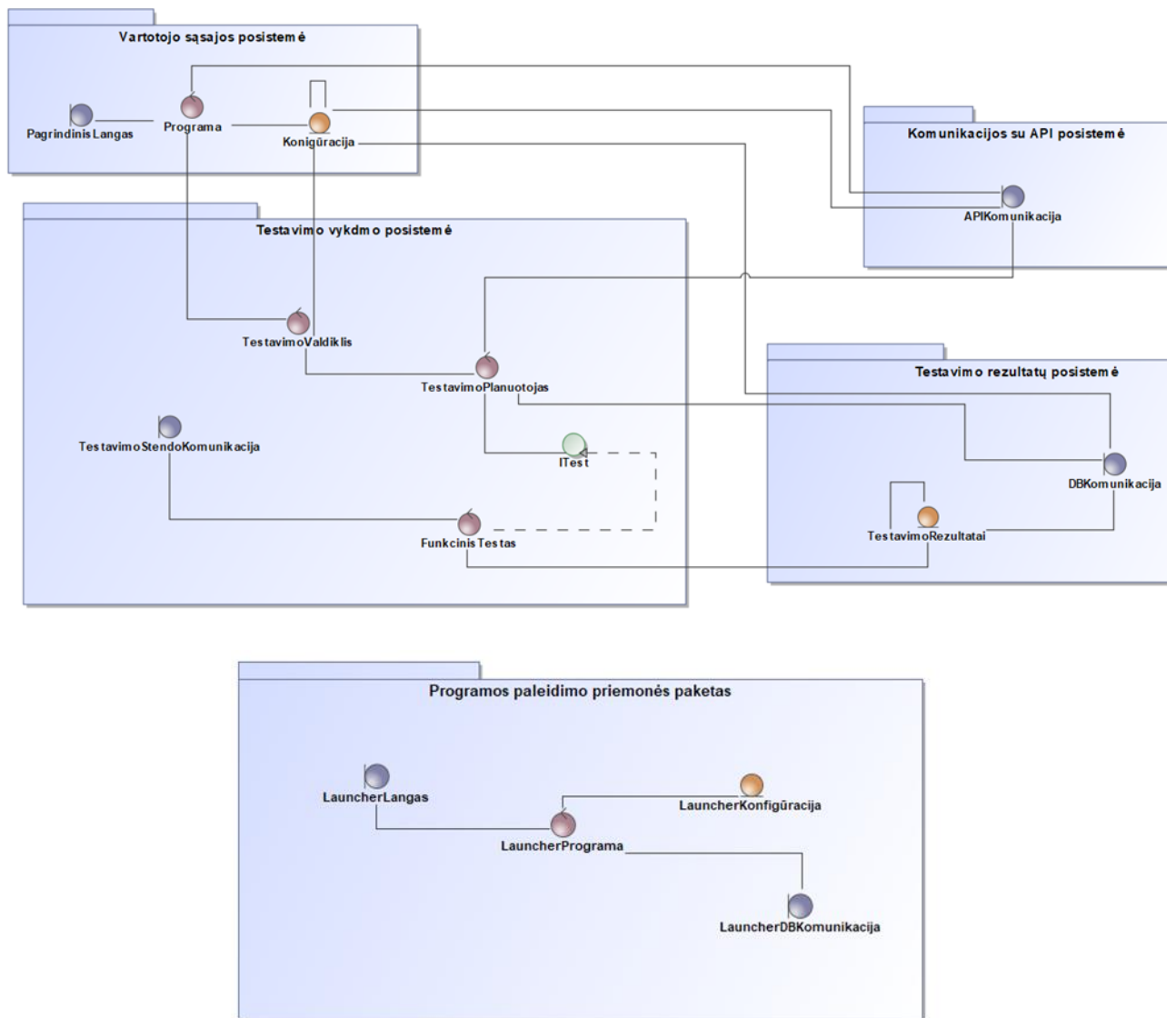


13 pav. Sistemos išskaidymo į paketus vaizdas

Sistemos išskaitymo į paketus vaizde pateikiami dviejų komponentų paketai. „Programos paleidimo priemonė“ paketas vaizduoja „programos paleidimo priemonės“ komponentą, o paketas „testavimo programa“ vaizduoja „testavimo programos“ komponentą. Visi šie paketai detalizuojami ir apžvelgiami išsamiau kitame skyrelyje („2.5.2 Paketų detalizavimas“).

2.5.2. Paketų detalizavimas

Žemiau pateikiamos sistemos paketų klasių diagramos ir kiekvieno paketo trumpi aprašymai.



14 pav. Paketų klasių diagrama

„Programos paleidimo priemonės paketas“ – tai sistemos dalis, kuri atsakinga už tai, kad visad būtų paleista naujausia testavimo programos versija. Paleidinėjant sistemą pirmiausia paleidžiama būtent ši dalis. Ji sudaro komunikaciją su duomenų baze ir patikrina ar duomenų bazėje yra naujesnė sistemos versija. Jeigu duomenų bazėje nėra naujesnės sistemos versijos negu dabar įdiegta kompiuteryje sistema tiesiog paleidžiama, priešingu atveju naujausia versija parsiuočiama į kompiuterį taip užtikrinant, kad kompiuteryje nuolat veiks naujausia sistemos versija.

„Vartotojo sąsajos posistemė“ - tai testavimo programos dalis, kuri atsakinga už programos grafinę vartotojo sąsają, vartotojo įvedamų duomenų nuskaitymą ir apdorojimą, sistemos konfigūracijos nuskaitymą iš konfigūracijos failų, šiuose failuose duomenys saugomi XML formatu. Ši posistemė paruošia visą sistemą tolesniam darbui, „testavimo vykdymo“ posistemėi perduoda visą reikiamą informaciją apie testuojamą gaminį, kad ši galėtų sėkmingai inicijuotis. Suteikia galimybę visoms kitoms posistemėms prieigą prie konfigūracijos.

„Testavimo vykdymo posistemė“ – tai testavimo programos dalis atsakinga už testavimo plano vykdymą. Ši posistemė gavusi reikiamą informaciją iš „vartotojo sąsajos“ posistemės paruošia

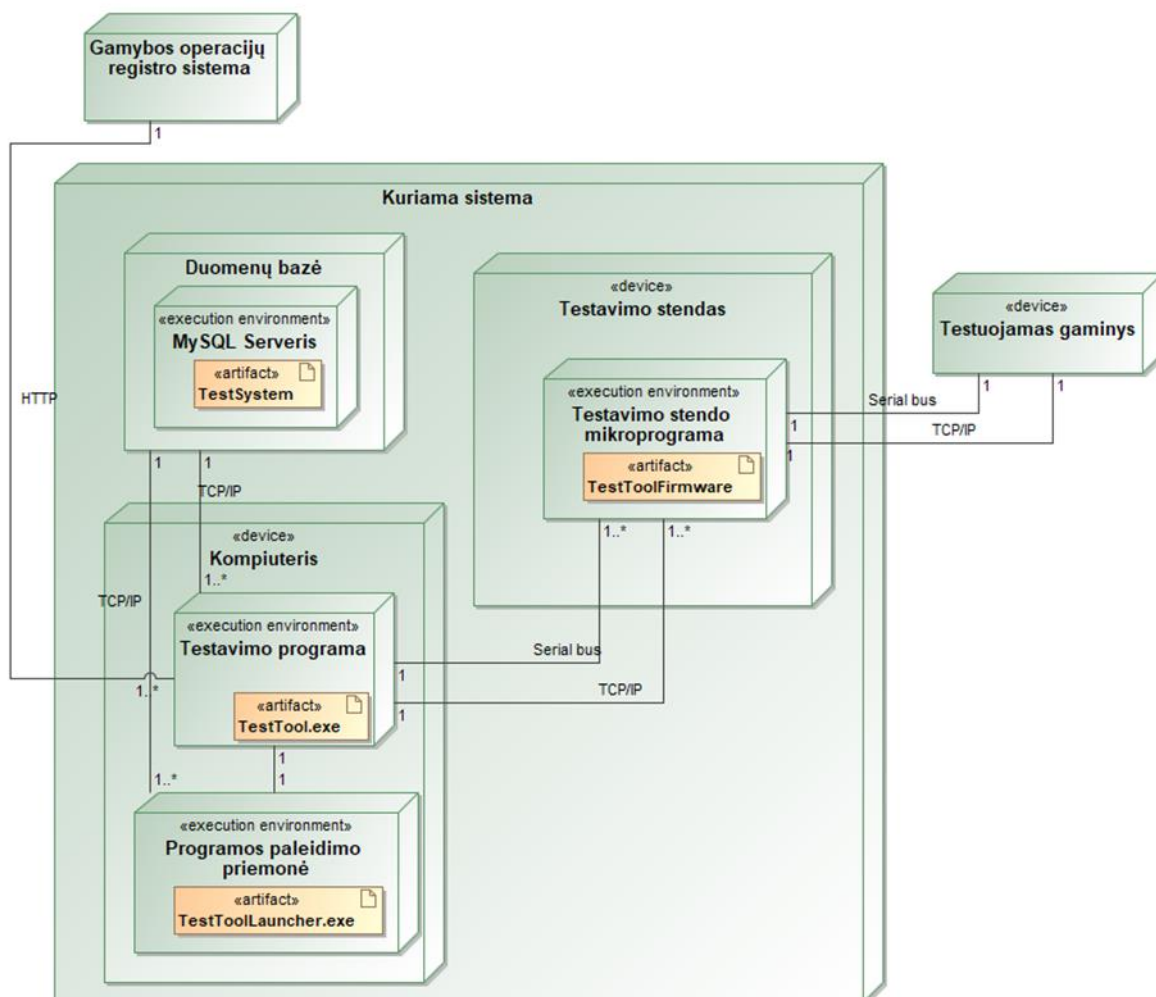
testavimo planą skirtą konkrečiam gaminiui ir pradeda jį vykdyti (atsižvelgiant į tai kiek testavimo stendų bus naudojama). Ši posistemė taip pat atsakinga ir už komunikaciją su testavimo stendais ir už konkrečių testų vykdymą.

„Komunikacijos su API posistemė“ – tai testavimo programos dalis atsakinga už komunikaciją su vidiniu įmonės REST API kuriame saugomi duomenys apie kiekvieną gaminį ir atliekamos atliktų operacijų registracijos. Duomenų apsikeitimas su šiuo REST API vykdomas naudojant JSON formatą.

„Testavimo rezultatų posistemė“ – tai testavimo programos dalis atsakinga už komunikaciją su duomenų baze, testavimo rezultatų įrašymu į duomenų bazę. Ši posistemė iš „testavimo vykdymo“ posistemės gauna duomenis apie įvykdytus testus ir testų rezultatus ir juos įrašo į duomenų bazę.

2.6. Išdėstymo (deployment) vaizdas

Žemiau pateikiama sistemos išdėstymo vaizdas kartu su sistemos komponentais, kurie išdėstyti skirtinguose techninės įrangos mazguose.



15 pav. Išdėstymo vaizdas

Siekiant užtikrinti korektišką sistemos veikimą techninė įranga, kurioje bus išdėstyta ir veiks sistema turi atitikti šias specifikacijas:

Kompiuteris:

- Bent 4 GB darbinės atminties (RAM);
- Bent 1 GB laisvos disko vietos;
- Procesorius Intel® Core®, 2,5 GHz arba analogiškas (spartesnis);
- Dvi tinklo plokštės;
- Windows 10 operacinė sistema;
- .NET 4.5 karkasas (arba naujesnė versija);
- FTP servisas;
- TFTP servisas;

Testavimo stendas:

- Raspberry Pi 4 mikrokompiuteris;
- OpenWRT operacinė sistema;

2.7. Duomenų vaizdas

Žemiau pateikiamas duomenų bazės modelio vaizdas.



16 pav. Duomenų bazės modelio vaizdas

Kuriamoje sistemoje naudojama duomenų bazė, kurioje yra dvi lentelės. Lentelėje „TestavimoRezultatai“ saugomi gaminių testavimo rezultatai, kitoje lentelėje „Naujiniai“ saugomi sistemos konfigūracijos ir sisteminiai failai, kurie naudojami automatinio sistemos naujinimosi proceso metu.

3. Tyrimo dalis

3.1. Sukurtos sistemos kokybės analizės tikslai

3.1.1. Aptikti klaidas funkcionavime, logikoje, realizacijoje

Siekiant nustatyti ar kuriamos sistemos funkcionavime, logikoje ir realizacijoje nėra klaidų buvo sudarytas testavimo planas, kuriame išskirti testavimo tikslai, sukurta testavimo strategija. Vykdamas sistemos testavimą vadovaujantis sistemos testavimo planu aptiktos ir klaidos, jos iškart ištaisytos. Taip pat siekiant aptikti kuo daugiau klaidų sistemoje sudaryti automatiniai testai.

3.1.2. Patikrinti ar programų sistema atitinka reikalavimų specifikaciją

Siekiant nustatyti ar kuriama programų sistema atitinka reikalavimų specifikaciją į testavimo plano dalį „Priėmimo testavimas“ įtrauktas papildomas procesas. Šio proceso metu patikrinta ne tik visas sistemos funkcionavimas, bet ir tikrinta ar sistema atitinka visas specifikacijas ir reikalavimus aprašytus sistemos reikalavimų specifikacijoje.

3.1.3. Įsitikinti, ar programų sistema sukurta pagal standartus

Siekiant nustatyti ar programų sistema sukurta pagal standartus į testavimo plano dalį „Priėmimo testavimas“ įtraukta papildomas procesas. Šio proceso metu patikrinta ne tik visas sistemos funkcionavimas, bet ir tikrinta ar sistema sukurta pagal iš anksto numatytus standartus. Numatytieji standartai įtraukti į reikalavimų specifikaciją.

3.1.4. Suformuoti sistemos tobulinimo rekomendacijas

Siekiant suformuoti sistemos tobulinimo rekomendacijas bei nustatyti programos modulius reikalaujančius atlikti peržiūros operacijas į kokybės vertinimo procesą įtrauktas sistemos prižiūrimumo ir sudėtingumo charakteristikų (metrikų) tyrimas.

3.2. Kokybės vertinimo procesas

3.2.1. Peržiūros

Projekto pabaigoje vykdytojų komandą peržiūri projektą. Peržiūros tikslas surinkti informaciją apie projektą ir pateikti apžvalgą, kurioje pateikti projekto sprendimų apžvalgą. Apžvalgoje pateikiamos tinkami projekto sprendimai, kurie galės būti panaudoti kituose projektuose. Taip pat, aprašomi ir netinkami sprendimai, kuriuos reiktų patobulinti ar pakeisti tinkamesniais sprendimais vykdomam projektui. Projekto apžvalga buvo atliekama po kiekvienos didesnės sistemos dalies realizavimo.

Interviu su užsakovu

Interviu su užsakovu metu buvo apsvaistyta galimybė ateityje praplėsti sistemą pridedant šiuos pakeitimus:

- Vidinės sistemos „AVS“ palaikymą;
- Sukurti įrankį patogiai sistemos veikimo rezultatų peržiūrai;

Projektavimo komandos narių peržiūrų aprašymas

Peržiūrų metu buvo aptarinėjamos sistemos komponentės, sistemos architektūra. Buvo tariamasi kaip sistema turėtų būti testuojama.

Rolės ir atsakomybės

- Projekto vadovas – Sistemos analitikas, patarėjas;
- Studentas – Sistemos projektuotojas, programuotojas, testuotojas;

Apklausų anketos

Apklausa skirta sistemos naudotojams apklausti apie jų patirtį susijusią su sistema. Pagal apklausos rezultatus sprendžiama ar sistema išpildė jai keltus reikalavimus (tuos, kurių negalima pamatuoti objektyviai). Anketa pateikiama žemiau:

4 lentelė Apklausos anketa

„UnifiedTestApp“ sistemos apklausa skirta naudotojams po pilotinio diegimo	
Ši apklausa skirta žmonėms, kurie turėjo galimybę naudotis šia sistema. Jeigu jūs sistema nesinaudojote prašome šios anketos nepildyti. Pildantiems anketą, prie visų klausimų pateikite įvertinimą nuo 1 iki 5 (1 – visiškai nesutinku, 5 – visiškai sutinku).	
Ar sistemą vertinate kaip intuityvią (greit randamos norimos funkcijos, aišku kur ir ką spausti)?	
Ar pavyko naudotis sistema (galėjote atlikti jums paskirtas užduotis)?	
Ar sistemoje naudojama leksika jums buvo aiški (naudojami paaiškinamieji žodžiai, užrašai, tekstai)?	
Ar naudojant sistemą pastebėjote išryškintą (ar kitaip dėmesį atkreipiančią) svarbią informaciją?	
Ar klaidos pranešimo metu sistema jums aiškiai nurodo kaip turėtumėte pasielgti (ar nurodomi tolimesni veiksmai)?	

3.2.2. Formalios techninės peržiūros

Formalios techninės peržiūros buvo atliekamos nuolat, viso kūrime metu naudojant automatinius statinės kodo analizės įrankius.

Vykdam statinę kodo analizę, taikyti šie kriterijai:

- Turi būti pašalintas nevykdomas programinis kodas;
- Turi būti panaudoti visi paskelbti kintamieji;
- Turi būti realizuoti visi klasių metodai;
- Programoje turi būti taikomas vienodas programinio kodo stilius;
- Kitos pastabos susijusios su programiniu kodu;

Dar vienas formalios techninės peržiūros metodas, kuris buvo naudojamas tai vienetų testavimas. Vienetų testavimas atliekamas po sistemos sukūrimo. Vienetų testavimas padengia pagrindines sistemos funkcijas.

3.2.3. Prižiūrimumo ir sudėtingumo charakteristikų tyrimas

Tyrimui naudoti įrankiai

Tyrimo atlikimui pasirinkta programinė įranga *Designite* (<https://www.designite-tools.com/>). Šis įrankis skirtas C# ir programavimo kalba kurtų sistemų analizavimui, metrikų skaičiavimui.

Šis įrankis gali matuoti šias metrikas:

- NOF (angl. Number Of Fields);
- NOM (angl. Number Of Methods);
- NOP (angl. Number Of Properties);
- NOPF (angl. Number of Public Fields);
- NOPM (angl. Number of Public Methods);
- LOC (angl. Lines of Code);
- WMC (angl. Weighted Methods per Class);
- NC (angl. Number of Children);
- DIT (angl. Depth of Inheritance Tree);
- LCOM (angl. Lack of Cohesion of Methods);
- FANIN (angl. Fan-in);
- FANOUT (angl. Fan-out);
- CC (angl. Cyclomatic Complexity);
- PC (angl. Parameter count);

Taip pat tyrimui panaudojau ir *Visual Studio IDE*. Šio įrankio pagalba buvo pamatuotas prižiūrimumo indeksas (MI).

Tyrimui naudojamų programų charakteristikų (metrikų) aprašymas

Šiam tyrimui naudojamos šios metrikos:

- MI - Prižiūrimumo indeksas (angl. Maintainability Index), programų sistemos priežiūrai reikalingų pastangų matas;
- CC - Ciklomatinis sudėtingumas (angl. Cyclomatic Complexity), nepriklausomų kelių programos kodo grafe skaičius;
- WMC – Metodų klasėje ciklomatinio sudėtingumo suma (angl. weighted methods per class);
- LCOM - Metodų darnos trūkumas (angl. lack of cohesion of methods);
- LOC - Programinio kodo eilučių skaičius(angl. Lines of Code);

Tyrimo eiga

Tyrimas atliekamas įrankio pagalba, automatizuotu būdu išmatuojant pasirinktas programinio kodo metrikas. Įrankis su metrikų reikšmėmis taip pat nurodo ir rekomenduojamas metrikų reikšmes, nurodo reikšmes, kurios viršija ribas. Metrikos viršijančios rekomenduojamas reikšmes indikuoja apie problemines sistemos vietas. Tyrimo rezultatų analizavimo metu atsižvelgiant į realias ir rekomenduojamas metrikų reikšmes suformuojamos sistemos tobulinimo rekomendacijos, nustatomi sistemos moduliai reikalaujantys peržiūros operacijų.

3.3. Vertinimo rezultatai

3.3.1. Kokybės įvertinimas

Po interviu su užsakovu į programų sistemos kokybę pažvelgta (ir įvertinta) iš dviejų perspektyvų, kūrėjo ir naudotojo. Vertinimui pasitelkiamos vertinimo lentelės, kuriose atskirai vertinama kiekviena savybė ir jis pakomentuojam. Savybės vertinamos „labai blogai“, „blogai“, „patenkinamai“, „gerai“ arba „labai gerai“ apibendrinti įvertinamai pateikiami atskirai vertinimui iš kūrėjo perspektyvos ir atskirai iš naudotojo perspektyvos. Apibendrintas vertinimas atliekamas pagal savybę su žemiausiu įvertinimu.

Vertinimas iš kūrėjo perspektyvos

Sistemą vertinant iš kūrėjo perspektyvos fokusuojamasi ne į sukurta galutinę sistemą, bet į programinio kodo rinkinį.

5 lentelė Vertinimas iš kūrėjo perspektyvos

Savybė	Įvertinimas	Komentaras
Įvykdyti visi sutartiniai įsipareigojimai	Labai gerai	Sistema įgyvendino visus reikalavimus ir atitinka specifikaciją.
Naudojama šiuolaikinė ir populiari programavimo kalba	Labai gerai	Naudojama tinkamai parinkta, populiari programavimo kalba ir įrankiai
Programinis kodas lengvai plečiamas	Gerai	Programinis kodas lengvai plečiamas, bet tam reikia perprasti sistemos struktūra ir architektūrą.
Programinės įrangos struktūra, stilius aiškūs ir suprantami	Labai gerai	Struktūra, stilius aiškūs ir suprantami.
Programinės įrangos architektūra leidžianti programinės įrangos lengvą kūrimą iš atskirų atsiėtų komponentų	Labai gerai	PĮ architektūra išnaudoja objektinio programavimo principų, dizaino modelių privalumus, dėka to sistemos architektūra yra nesunkiai plečiama, modifikuojama.
Tinkama programinės įrangos kūrimo procesas	Gerai	PĮ kurta pagal „Agile“ ideologiją ir iš principo kūrimo procesui priekaištų nėra. Identifikuojama vienintelė probleminė vietai tai, kad sistemą kūrė vienas žmogus. Tai gali reikšti priimtus ne pačius teisingiausius sprendimus kūrimo eigoje.

Apibendrinant vertinimą iš kūrėjo perspektyvos sistema vertinama **gerai**.

Vertinimas iš naudotojo perspektyvos

Sistemą vertinant iš naudotojo perspektyvos fokusuojamasi į sukurta galutinę sistemą. Vertinimo lentelė sudaryta atsižvelgiant į ISO-25010 [15] standarto naudojamas charakteristikas.

6 lentelė Vertinimas iš naudotojo perspektyvos

Savybė	Įvertinimas	Komentaras
Funkcinis tinkamumas	Labai gerai	Sistema teikia visas numatytas funkcijas
Našumo efektyvumas	Labai gerai	Sistema veikia efektyviai, nenaudoja resursų daugiau nei jai skiriama, turi puikų atsako laiką.
Suderinamumas	Labai gerai	Sistema veikia teisingai su bendradarbiaujančiomis sistemomis
Tinkamumas naudoti	Labai gerai	Sistema gali naudotis visi numatyti vartotojai be jokių problemų
Patikimumas	-	Kol kas nevertinama. Tai gali būti vertinama tik po galutinio diegimo praėjus tam tikram laikui.
Saugumas	Labai gerai	Sistema išpildo saugumui keliamus reikalavimus, apsaugo informaciją numatytais metodais
Palaikomumas	Gerai	Vertinama atsižvelgiant į vertinimą iš kūrėjo perspektyvos.
Pernešamumas	Labai gerai	Sistema labai lengvai pernešama tarp skirtingos aparatinės įrangos.

Apibendrinant vertinimą iš naudotojo perspektyvos sistema vertinama **gerai**.

Įvertinimas po vartotojų apklausos

Sistemos vertinimas pagal vartotojų apklausą pateikiamas lentelėje.

7 lentelė Vartotojų apklausos rezultatai

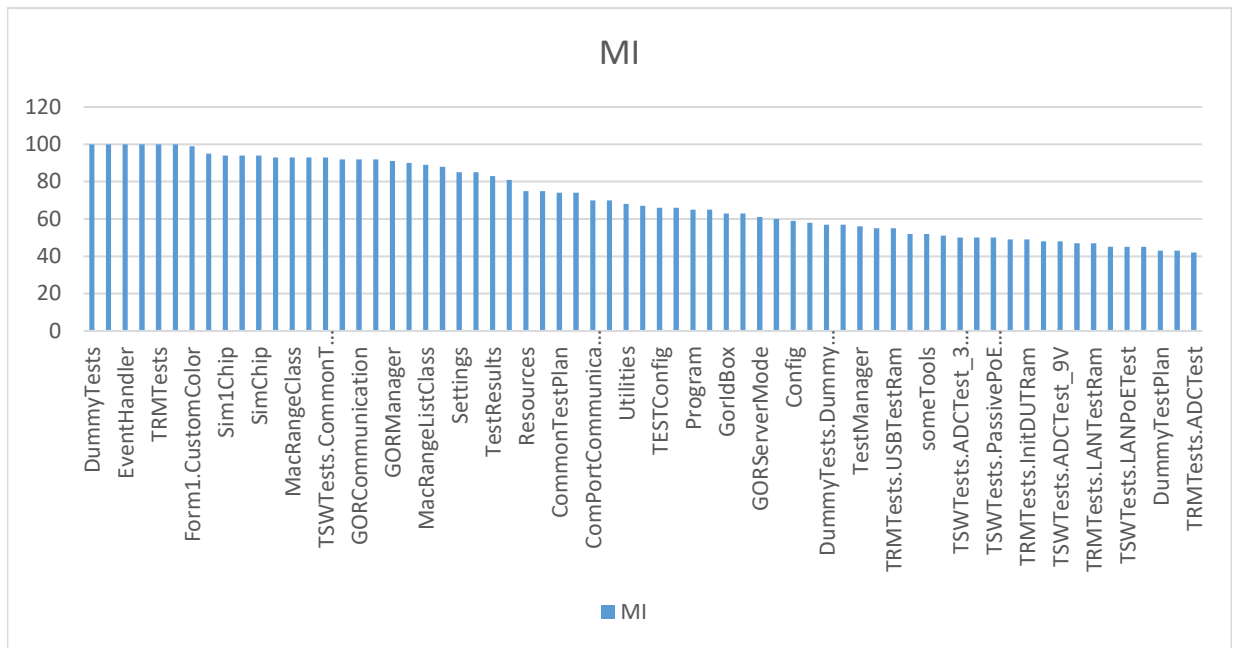
Anketos klausimas	Siekiami reikšmė	Vertinimai					Pasiiekta reikšmė
		5	4	3	4	5	
Ar sistemą vertinate kaip intuityvią (greit randamos norimos funkcijos, aišku kur ir ką spausti)?	70%	5	4	3	4	5	80%
Ar pavyko naudotis sistema (galėjote atlikti jums paskirtas užduotis)?	90%	4	5	4	4	5	100%
Ar sistemoje naudojama leksika jums buvo aiški (naudojami paaiškinamieji žodžiai, užrašai, tekstai)?	70%	4	5	5	4	4	100%
Ar naudojant sistemą pastebėjote išryškintą (ar kitaip dėmesį atkreipiančią) svarbią informaciją?	80%	4	4	3	4	4	80%
Ar klaidos pranešimo metu sistema jums aiškiai nurodo kaip turėtumėte pasielgti (ar nurodomi tolimesni veiksmai)?	80%	4	4	3	4	4	80%

Apibendrinant vertinimą po vartotojų apklausos galima teikti, kad sistema vertinama **gerai**, visi vertinami aspektai pasiekė keltus tikslus.

3.3.2. Prižiūrimumo ir sudėtingumo charakteristikų tyrimo rezultatai

Prižiūrimumo indeksas

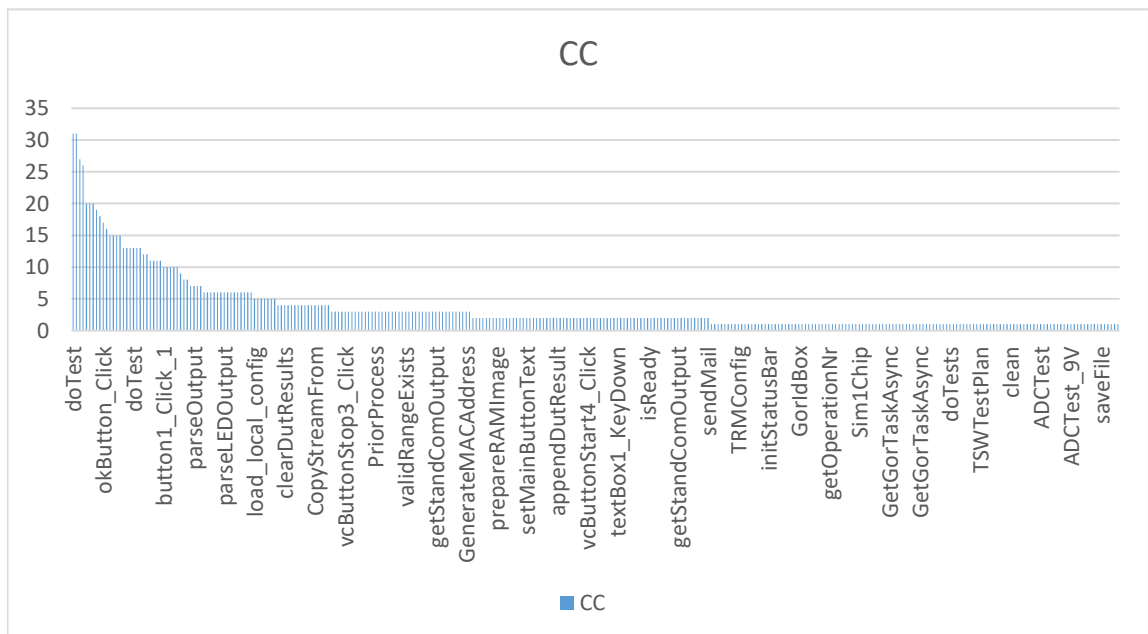
Bendras sistemos palaikomumo indeksas yra 77, tai rodo vidutinį sistemos palaikomumą. Pažvelgus daugiau į pavienes klases matoma, kad 29 iš 67 klasių (43%) yra vertinamos kaip blogo prižiūrimumo (vertė mažesnė nei 65), 14 iš 67 klasių (21%) yra vertinamos kaip vidutinio prižiūrimumo (vertė tarp 65 ir 85) ir 24 iš 67 klasių (36%) vertinamos kaip gero prižiūrimumo (vertė didesnė nei 85) [16]. Žemiau pateikiamas grafikas, kuriame atvaizduojamos visos klasės ir atitinkamai jų prižiūrimumo indeksas.



17 pav. Prižiūrimumo indekso reikšmės pagal klases

Ciklomatinis sudėtingumas

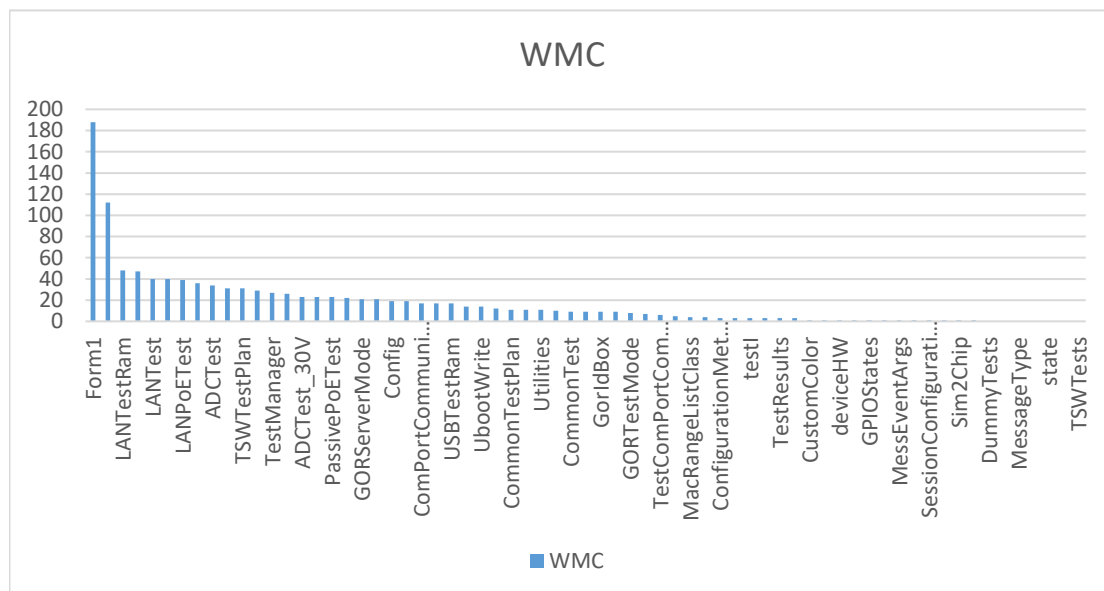
4 iš 312 (1.3%) metodai remiantis ciklomatinio sudėtingumo parametru (vertė rėžiuose tarp 21 ir 50) vertinami kaip sudėtingi ir turintys didelę klaidos tikimybę metode. 23 iš 312 (7.4%) metodai remiantis ciklomatinio sudėtingumo parametru (vertė rėžiuose tarp 11 ir 20) vertinami kaip vidurinio sudėtingumo, o klaidos tikimybė metode, vidutinė. Visi likusieji metodai remiantis ciklomatinio sudėtingumo parametru (vertė rėžiuose tarp 1 ir 10) vertinami kaip paprasti ir turintys mažą klaidos tikimybę [17]. Žemiau pateikiamas grafikas, kuriame atvaizduojamos visi metodai ir atitinkamai jų ciklomatinis sudėtingumas.



18 pav. Ciklomatinio sudėtingumo reikšmės pagal klases

Metodų klasėje ciklo matinio sudėtingumo suma

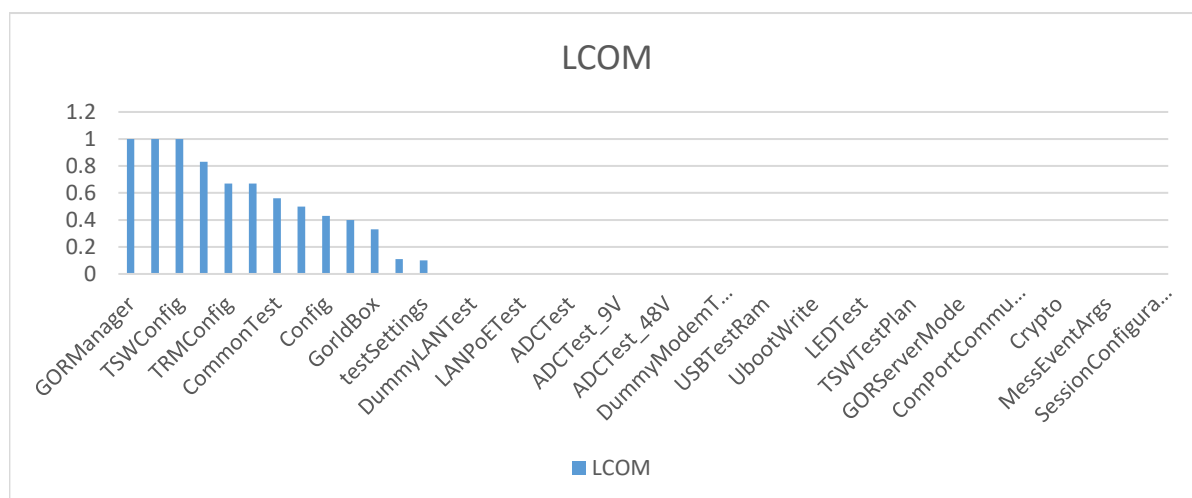
Remiantis įrankio *Designite* rekomendacijomis skirtomis programinės įrangos metrikoms [18] nustatyta, kad 2 iš 67 klasių (3%) viršija rekomenduojamą (100) metrikos reikšmę. Tai indikuoja, kad šios klasės yra per didelio sudėtingumo. Žemiau pateikiamas grafikas, kuriame atvaizduojamos visos klasės ir atitinkamai jų išmatuotos metrikos vertės.



19 pav. Metodų klasėje ciklo matinio sudėtingumo sumų reikšmės pagal klases

Metodų darnos trūkumas

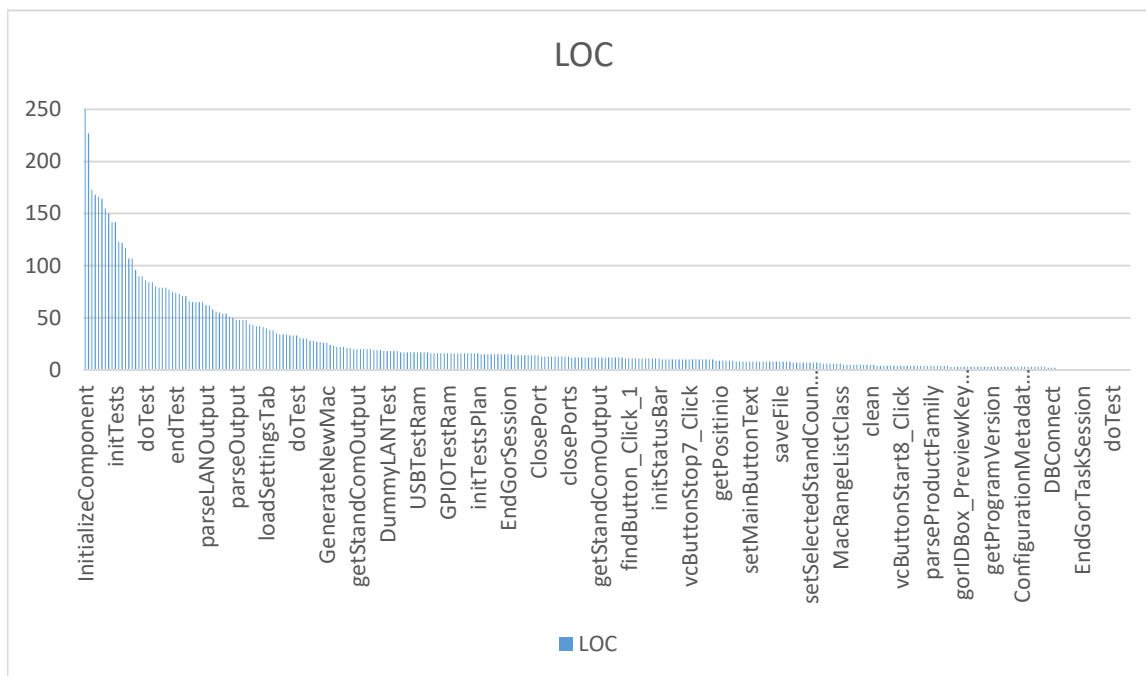
Remiantis įrankio *Designite* rekomendacijomis skirtomis programinės įrangos metrikoms [18] nustatyta, kad 4 iš 67 klasių (6%) viršija rekomenduojamą (0.8) metrikos reikšmę, o 24 klasėse šios metrikos nustatyti nebuvo galima. Rekomenduojamos metrikos reikšmės viršijimas indikuoja blogą sistemos dekompoziciją į klases. Žemiau pateikiamas grafikas, kuriame atvaizduojamos visos klasės ir atitinkamai jų išmatuotos metrikos vertės.



20 pav. Metodų darnos trūkumo reikšmės pagal klases

Programinio kodo eilučių skaičius

Remiantis įrankio *Designite* rekomendacijomis skirtomis programinės įrangos metrikoms [18] nustatyta, kad 15 iš 312 metodų (5%) viršija rekomenduojamą 100 eilučių per metodą skaičių. Tai parodo, kad šie metodai yra per dideli ir gali kelti nepatogumus juos prižiūrint. Žemiau pateikiamas grafikas, kuriame atvaizduojami visi metodai ir atitinkamai jų išmatuotos metrikos vertės.



21 pav. Eilučių skaičiaus metrikos reikšmės pagal metodus

Sistemos tobulinimo rekomendacijos

Atsižvelgiant į gautas sistemos programinio kodo metrikas, matyti, kad dalis sistemos komponentų yra ganėtinai didelio sudėtingumo, todėl siekiant paprastesnio sistemos palaikomumo ir siekiant sumažinti klaidų tikimybę reikėtų peržiūrėti sistemos metodus, kuriuose yra labai didelis ciklomatinis sudėtingumas ir klases, kuriuose metodų ciklomatinio sudėtingumo suma yra labai didelė. Metodų darnos trūkumo metrika tarsi rodo kelių sistemos klasių galimą nekokybiškumą ir ne visai gerą sistemos dekompoziciją į klases, bet atsižvelgiant į tai, kad metrikos rekomenduojamą reikšmę viršija tik kelios klasės, kurios sukurtos loginiam sistemos dalių atskyrimui ir yra skirtos tik pagalbiniams tikslams, naudojamos iš kitų sistemos komponentų galima teigti, kad tai nėra trūkumas. Programinio kodo eilučių skaičiaus rodo, kad dalis metodų viršija rekomenduojamą 100 eilučių per funkciją ribą. Siekiant patobulinti sistemą reikėtų peržiūrėti funkcijas viršijančias šią ribą ir esant prasmingumui funkcijas suskaidyti į mažesnes apjungiant vietas, kuriuose programinis kodas dubliuojasi.

4. Eksperimentinė dalis

4.1. Įvadas

Magistrinio darbo kūrimo metu buvo atlikti 2 eksperimentiniai tyrimai skirti nustatyti sukurtos sistemos kuriamą naudą įmonei.

Pirmasis tyrimas orientuotas į įmonės pagamintos produkcijos testavimo ir paruošimo naudojimui (mikroprogramos įrašymo, identifikacinių gamintojo parametrų įrašymo) proceso optimizavimą. Kaip ir buvo minėta anksčiau, įmonėje, kol nebuvo įdiegtos šios sistemos produkcijos testavimas buvo atliekamas skirtingų operacijų metu, todėl produkciją reikėdavo perkėlinėti iš vienos operacijos vykdymo vietos į kitą. Procesų vykdymas skirtingomis operacijomis priversdavo sugaišti pridėtinio laiko, vykdavo neoptimaliai. Įdiegus mano sukurtą sistemą situacija pasikeitė, produkcijos testavimas ir paruošimas naudojimui gali būti vykdomas vienos operacijos metu. Tyrimo metu nustatyta kiek ir kaip pasikeitė gaminio testavimo ir paruošimo naudoti procesas.

Antrasis tyrimas orientuotas į įmonės kvalifikuotų specialistų (programuotojų) resurso sąnaudas naudojantis sistema. Kol įmonė nenaudojo unifikuotos sistemos, kuriant įrankius naujų produktų testavimui, paruošimui naudojimui kiekvieną kartą buvo kuriamos naujos sistemos panaudojant prieš tai buvusių sistemų dalis. Įdiegus mano sukurtą sistemą situacija pasikeitė, specialistai, kurie yra atsakingi už gamybos procese naudojamos programinės įrangos kūrimą ir priežiūrą kurdami įrankius naujiems produktams testuoti gali naudoti mano sukurtą sistemą ir taip sutaupyti laiką. Tyrimo metu nustatyta kaip pasikeitė naujų produktų testavimo įrankių kūrimas.

4.2. Pirmasis tyrimas

4.2.1. Tyrimo aprašas

Visuose gamybinėse įmonėse, kurios gamina produkciją masiniu būdu yra ypač svarbu turėti kuo optimalesnius gamybos procesus, taupyti gaminio gamybos ciklo laiką. Tai gali tiesiogiai daryti įtaką produkcijos gamybos kaštams. Tyrimo metu bandoma nustatyti kaip pasikeitė testavimo proceso ciklo laikas lyginant su testavimo procesu iki kol mano kurta sistema nebuvo įdiegta ir naudojama. Tyrimo metu buvo analizuojami testavimo metu kaupiami duomenys bei duomenys gauti iš gamybos padalinio kolegų.

Tyrimo metu buvo lyginti dviejų produktų, *RUT950* [19] ir *RUT360* [20] testavimo ir paruošimo naudojimui procesai. *RUT950* produktas testuojamas ir ruošiamas naudojimui pagal nuo seno turimus metodus, vykdant dvi gamybinės operacijas. *RUT360* produkto testavimas ir ruošimas naudojimui buvo vykdytas su mano sukurta sistema, tai atliekama vienos gamybinės operacijos metu.

Atliekant tyrimą gaminių testavimo laikai buvo palyginti keliomis imtimis. Palyginta testavimo operacijos neatsižvelgiant į tai ar operacijos atliktos sėkmingos ar ne ir taip pat palyginta tik sėkmingai atliktos operacijos.

4.2.2. Tyrimo prielaidos

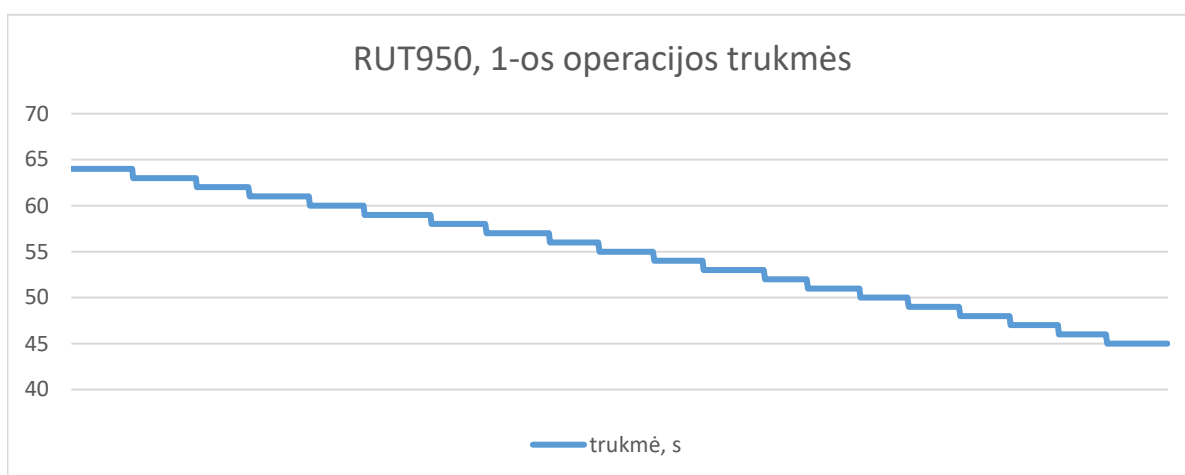
Atliekant tyrimą buvo daromos šios prielaidos:

- Abiejų tyrimui parinktų produktų funkcinis sudėtingumas panašus ir tai nedaro įtakos tyrimo rezultatams;

- Abiejų tyrimui pasirinktų produktų prijungimas prie testavimo įrangos ir atjungimas nuo jos trunka tiek pat laiko;
- Produkto *RUT950* gabenimo iš vienos darbo vietos į kitą laiko sąnaudos yra 10s. vienam gaminiui;

4.2.3. Tyrimo duomenys

Žemiau pateikiamas grafikas, kuriame atvaizduojama *RUT950* produkto 1 000 atliktų operacijų trukmės surikiuotos nuo didžiausios iki mažiausios.



22 pav. *RUT950* produkto pirmos operacijos trukmių grafikas

Žemiau esančioje lentelėje pateikiamos *RUT950* produkto pirmos operacijos statistinės vertės.

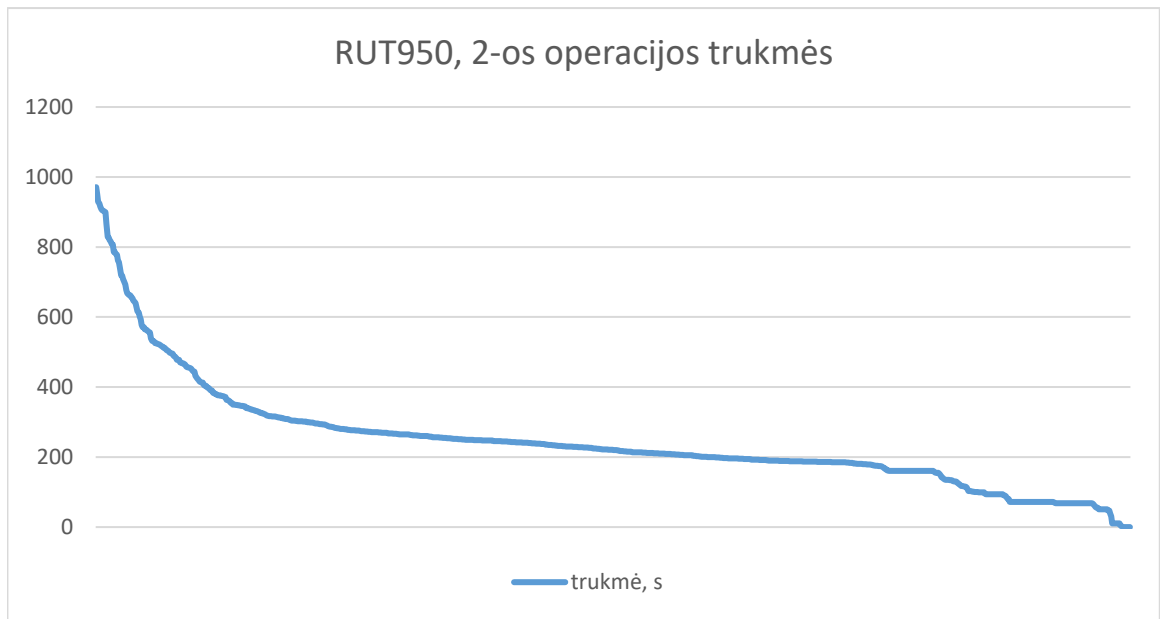
8 lentelė *RUT950* produkto pirmos operacijos statistinės vertės

Pavadinimas	Vertė
Didžiausia vertė, s.	64
Mažiausia vertė, s.	45
Vidurkis, s.	54.84
Mediana, s.	55
Bendra atliktų operacijų trukmė, s.	54 841

RUT950 pirmos operacijos vykdymo metu negalima nustatyti ar operacija įvyko sėkmingai ar ne, todėl ši operacija tiriama tik viena imtimi.

Pagal taikomą prielaidą, kad vieno gaminio pergabenimas iš pirmos operacijos darbo vietos į antros operacijos darbo vietą trunka 10s. apskaičiuojama, kad pergabenti 1 000 gaminių užtrunka 10 000s.

Žemiau pateikiamas grafikas, kuriame atvaizduojama *RUT950* produkto 1 000 atliktų operacijų trukmės surikiuotos nuo didžiausios iki mažiausios, neatsižvelgiant ar operacija buvo sėkminga ar ne.



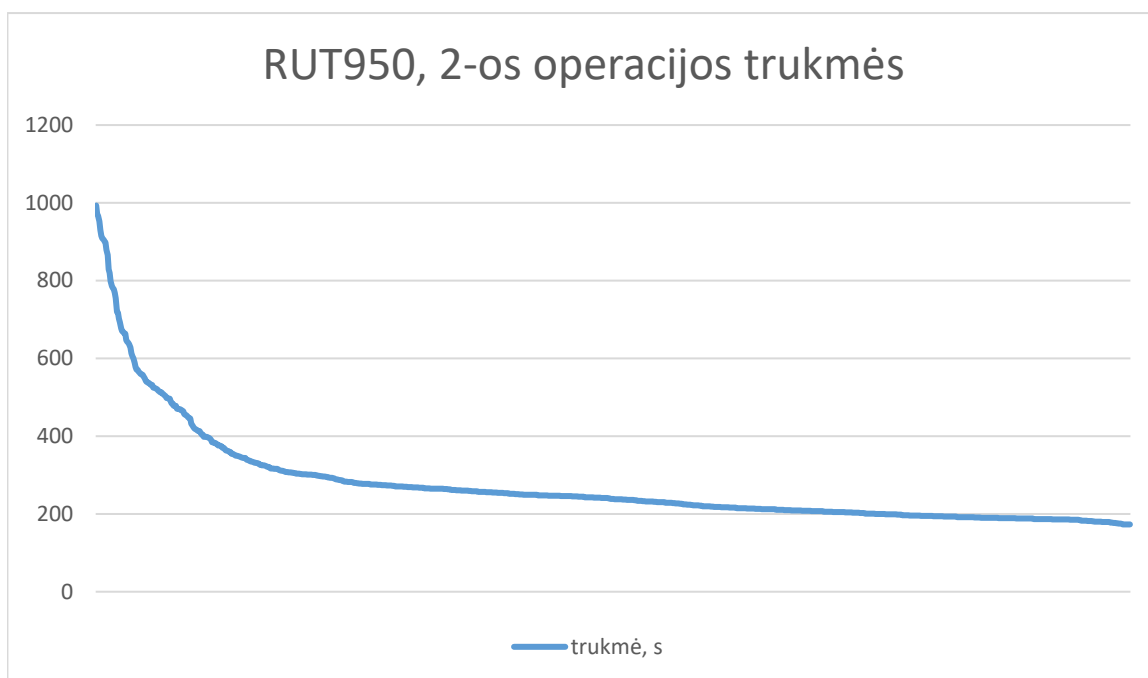
23 pav. RUT950 produkto antros operacijos (visi bandymai) trukmių grafikas

Žemiau esančioje lentelėje pateikiamos RUT950 produkto antros operacijos, neatsižvelgiant ar operacija buvo sėkminga ar ne, statistinės vertės.

9 lentelė RUT950 produkto antros operacijos (visi bandymai) statistinės vertės

Pavadinimas	Vertė
Didžiausia vertė, s.	971
Mažiausia vertė, s.	0
Vidurkis, s.	246.12
Mediana, s.	220.5
Bendra atliktų operacijų trukmė, s.	246 119

Žemiau pateikiamas grafikas, kuriame atvaizduojama RUT950 produkto 1 000 sėkmingai atliktų operacijų trukmės surikiuotos nuo didžiausios iki mažiausios.



24 pav. *RUT950* produkto antros operacijos (sėkmingi bandymai) trukmių grafikas

Žemiau esančioje lentelėje pateikiamos *RUT950* produkto antros operacijos (sėkmingi bandymai) statistinės vertės.

10 lentelė *RUT950* produkto antros operacijos (sėkmingi bandymai) statistinės vertės

Pavadinimas	Vertė
Didžiausia vertė, s.	993
Mažiausia vertė, s.	173
Vidurkis, s.	274.82
Mediana, s.	239
Bendra atliktų operacijų trukmė, s.	274 822

Žemiau esančioje lentelėje pateikiami konsoliduoti *RUT950* produkto testavimo operacijos (visi bandymai) statistiniai duomenys.

11 lentelė Konsoliduoti *RUT950* produkto testavimo operacijos (visi bandymai) statistiniai duomenys

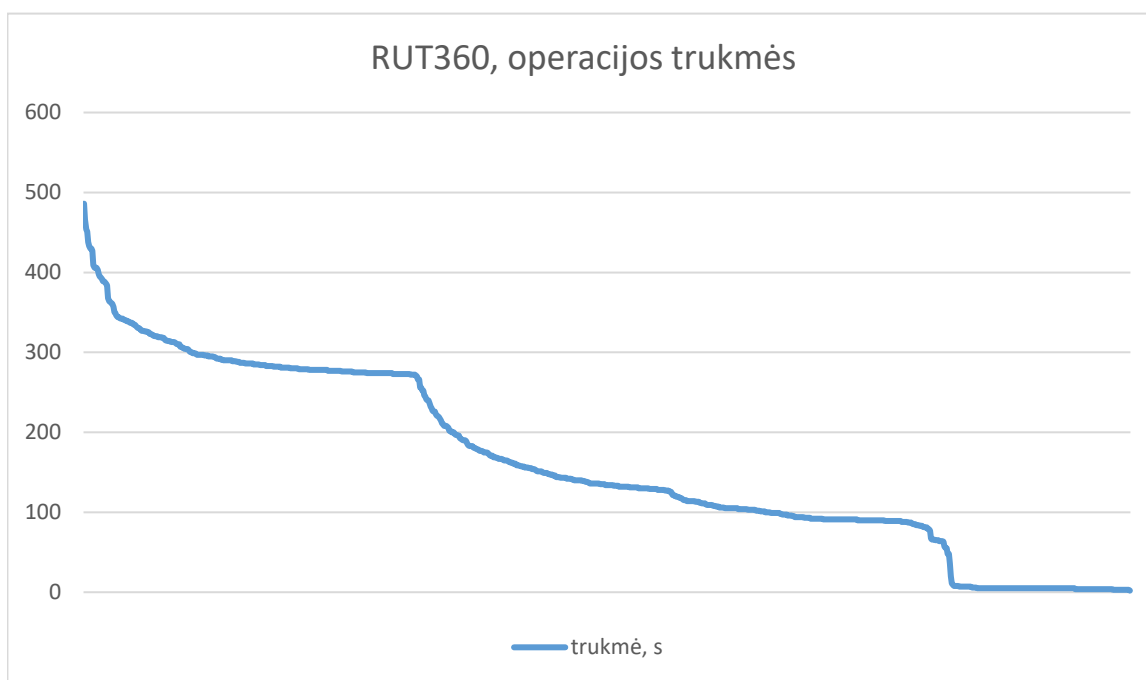
Pavadinimas	Vertė
Didžiausia vertė, s.	1 045 (64+10+971)
Mažiausia vertė, s.	55 (45+10+0)
Vidurkis, s.	310.96
Mediana (medianų suma), s.	285.5 (55+10+220.5)
Bendra atliktų operacijų trukmė, s.	310 960 (54 841+10 000+246 119)

Žemiau esančioje lentelėje pateikiami konsoliduoti *RUT950* produkto testavimo operacijos (sėkmingi bandymai) statistiniai duomenys.

12 lentelė Konsoliduoti RUT950 produkto testavimo operacijos (sėkmingi bandymai) statistiniai duomenys

Pavadinimas	Vertė
Didžiausia vertė, s.	1 067 (64+10+993)
Mažiausia vertė, s.	228 (45+10+173)
Vidurkis, s.	339.66
Mediana (medianų suma), s.	304 (55+10+239)
Bendra atliktų operacijų trukmė, s.	339 663 (54 841+10 000+274 822)

Žemiau pateikiamas grafikas, kuriame atvaizduojama RUT360 produkto 1 000 atliktų testavimo operacijų trukmės surikiuotos nuo didžiausios iki mažiausios, neatsižvelgiant ar operacija buvo sėkminga ar ne.



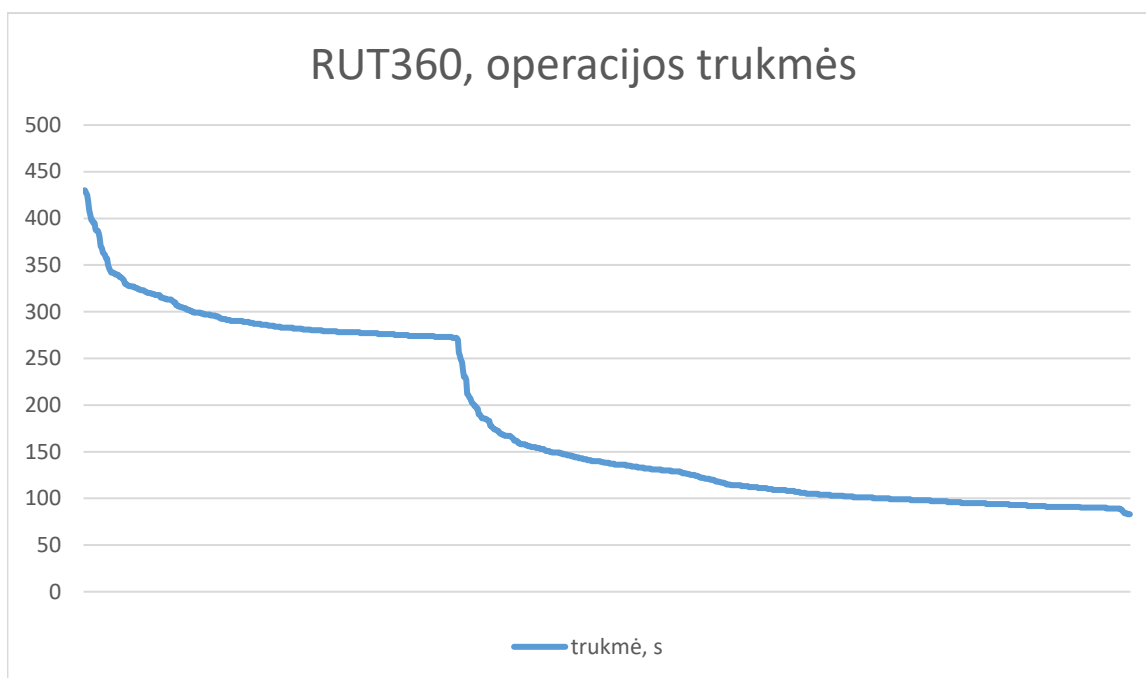
25 pav. RUT360 produkto testavimo operacijos (visi bandymai) trukmių grafikas

Žemiau esančioje lentelėje pateikiamos RUT360 produkto testavimo operacijos, neatsižvelgiant ar operacija buvo sėkminga ar ne, statistinės vertės.

13 lentelė RUT360 produkto testavimo operacijos (visi bandymai) statistinės vertės

Pavadinimas	Vertė
Didžiausia vertė, s.	486
Mažiausia vertė, s.	2
Vidurkis, s.	161.48
Mediana, s.	134
Bendra atliktų operacijų trukmė, s.	161 475

Žemiau pateikiamas grafikas, kuriame atvaizduojama RUT360 produkto 1 000 sėkmingai atliktų testavimo operacijų trukmės surikiuotos nuo didžiausios iki mažiausios.



26 pav. RUT360 produkto testavimo operacijos (sėkmingi bandymai) trukmių grafikas

Žemiau esančioje lentelėje pateikiamos RUT360 produkto testavimo operacijos (sėkmingi bandymai) statistinės vertės.

14 lentelė RUT360 produkto testavimo operacijos (sėkmingi bandymai) statistinės vertės

Pavadinimas	Vertė
Didžiausia vertė, s.	430
Mažiausia vertė, s.	83
Vidurkis, s.	182.59
Mediana, s.	138
Bendra atliktų operacijų trukmė, s.	182 585

4.2.4. Tyrimo rezultatai

Turint dviejų produktų testavimo operacijų (atliktų skirtingais metodais) atlikimo trukmių duomenis galima juos palyginti ir nustatyti kaip įdiegta sistema daro įtaką testavimo operacijos atlikimo laikui.

Žemiau pateikiamas RUT950 ir RUT360 produktų testavimo operacijos (visi bandymai neatsižvelgiant į tai ar operacija buvo atlikta sėkmingai ar ne) atlikimo trukmių palyginimas.

15 lentelė RUT950 ir RUT360 produktų testavimo operacijos (visi bandymai) atlikimo trukmių palyginimas

	RUT950	RUT360	Skirtumas	Skirtumas, proc.
Didžiausia vertė, s.	1 045	486	-559	-53,5%
Mažiausia vertė, s.	55	2	-53	-96,4%
Vidurkis, s.	310,96	161,48	-149,48	-48%
Mediana, s.	285,5	134	-151,5	-53%

Bendra atliktų operacijų trukmė, s.	310 960	161 475	-149 485	-48%
-------------------------------------	---------	---------	----------	------

Iš gauto duomenų palyginimo matyti, kad *RUT360* produkto testavimo operacija atliekama žymiai greičiau. Vidutinis vieno *RUT360* gaminio testavimas yra 149,48s. (~2,5 min.) spartesnis negu *RUT950* gaminio. Bendras 1 000 gaminių testavimas trunka 149 485s. (~41,5 h.) sparčiau. Testavimo laiko sumažėjimas siekia 48%.

Žemiau pateikiamas *RUT950* ir *RUT360* produktų testavimo operacijos (sėkmingi bandymai) atlikimo trukmių palyginimas.

16 lentelė *RUT950* ir *RUT360* produktų testavimo operacijos (sėkmingi bandymai) atlikimo trukmių palyginimas

	RUT950	RUT360	Skirtumas	Skirtumas, proc.
Didžiausia vertė, s.	1067	430	-637	-59%
Mažiausia vertė, s.	228	83	-145	-63,6%
Vidurkis, s.	339,66	182,59	-157,07	-46,2%
Mediana, s.	304	138	-166	-54,6%
Bendra atliktų operacijų trukmė, s.	339 663	182 585	-157 078	-46,2%

Iš gauto duomenų palyginimo matyti, kad *RUT360* produkto testavimo operacija atliekama žymiai greičiau net ir lyginant tik sėkmingai atliktus bandymus. Vidutinis vieno *RUT360* gaminio testavimas yra 157,07s. (~2,6 min.) spartesnis negu *RUT950* gaminio. Bendras 1 000 gaminių testavimas (skaičiuojant tik sėkmingus bandymus) trunka 157 078s. (~43,6 h.) sparčiau. Testavimo laiko sumažėjimas siekia 46.2%.

4.2.5. Tyrimo išvados

Atlikus tyrimą nustatyta, kad įdiegta sistema sumažino gamybinės operacijos atlikimo trukmę. Nustatyta, kad vieno gaminio testavimo ciklas sutrumpėjo vidutiniškai ~2,5 min. Pasiiektas 48% testavimo ciklo laiko sumažėjimas.

4.3. Antrasis tyrimas

4.3.1. Tyrimo aprašas

Programuotojai yra vieni iš brangiausiai apmokamų specialistų todėl ypač svarbu, kad programuotojai dirbtų efektyviai ir optimaliai, kad kiekvieną darbo valandą sukurtų kuo didesnę vertę įmonei. Efektyvus programinės įrangos inžinierių resurso panaudojimas gali tapti vienu iš svarbiausių veiksnių IT įmonės sėkmei. Tyrimo metu bandoma nustatyti kaip pasikeitė testavimo įrangos kūrimo proceso laikas lyginant su testavimo įrangos kūrimo procesu iki kol mano sukurta sistema nebuvo įdiegta ir naudojama. Tyrimo metu buvo analizuojami įmonės vidinių užduočių sekimo sistemų duomenys.

Tyrimo metu buvo lyginti dviejų užduočių atlikimo laikai. Viena užduotis apėmė įrankio skirtą naudoti gamybos procese, naujo produkto testavimui, kūrimą. Įrankis buvo kuriamas remiantis nuo seno įmonėje naudota programine įranga. Kita užduotis buvo sukurti įrankį naujo produkto testavimui gamybos linijoje papildant mano sukurta unifikuotą sistemą.

Tyrimo metu taip pat palygintas laikas skirtas užduoties atlikimo metu padarytiems pakeitimams įdiegti gamybos kompiuteriuose.

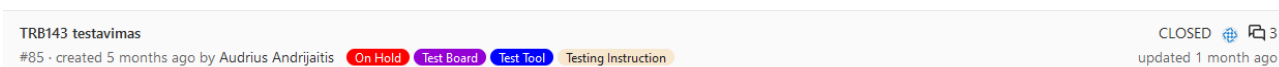
4.3.2. Tyrimo prielaidos

Atliekant tyrimą buvo daromos šios prielaidos:

- Inžinieriai atliekantys užduotis turi panašią kompetenciją;
- Abiejų kuriamų įrankių funkcinis sudėtingumas panašus;
- Sistema diegiama į vienodą kiekį gamybos kompiuterių;

4.3.3. Tyrimo duomenys

Žemiau esančiuose paveikslėliuose pateikiami duomenys iš vidinės įmonės užduočių sekimo sistemos. Paveikslėliuose pateikiama unifikuotos automatinio testavimo sistemos papildymo naujo gaminio testavimu užduotis ir šiai užduočiai skirto laiko ataskaita.



27 pav. Unifikuotos automatinio testavimo sistemos papildymo naujo gaminio testavimu užduotis

Time tracking report ×

Spent At	User	Time Spent	Summary / Note
March 25, 2022, 15:33 (UTC: +0300)	Audrius Andrijaitis	2w 3d 3h 15m	
		2w 3d 3h 15m	Summary / Note

28 pav. Užduoties vykdymui skirto laiko ataskaita

Atlikus užduotį su unifikuota sistema pakeitimai į gamybos kompiuterius diegiami panaudojant automatinio sistemos naujinimo funkcionalumą. Panaudojant šį funkcionalumą įrankis į gamybos kompiuterius įdiegiamas užtrunkant iki 5 min.

Žemiau esančiame paveikslėlyje pateikiami duomenys iš vidinės įmonės užduočių sekimo sistemos. Paveikslėlyje pateikiama užduoties, kurios metu buvo kuriamas įrankis automatiniam testavimui (remiantis nuo seno įmonėje naudota programine įranga), ataskaita.

Foto	Darbuotojas	Užduotis	Skirtos val.	Planuotos val.	%	Projektas	Planuotas Terminas	Realus Terminas	Rezultatas
		RUT950 automatinio testavimo įgyvendinimas gamyboje	142	100	100	RUT950	2020-11-30 22:00:00	2020-12-16 15:36	Atlikta Patvirtinta

29 pav. Automatinio testavimo (sena programinė įranga) kūrimo užduotis

Atlikus užduotį remiantis nuo seno įmonėje naudota programine įranga sistema į gamybos kompiuterius diegiama rankinių būdų. Tai gali užtrukti iki 30 min.

4.3.4. Tyrimo rezultatai

Turint dviejų užduočių atlikimo trukmių duomenis galima juos palyginti ir nustatyti kaip unifikuota sistema daro įtaką programuotojo užduoties atlikimo laikui. Taip pat palyginama kaip pasikeičia sugaištamas laikas įrankių diegimui gamybos kompiuteriuose.

Žemiau pateikiama lentelė, kurioje lyginami tyrimo duomenys.

17 lentelė Užduočių trukmių palyginimas

	Sena PĮ	Unifikuota sistema	Skirtumas	Skirtumas, proc.
Užduoties vykdymo trukmė, h.	142	107,25	-34,75	-24,5%
Diegimo trukmė, min.	30	5	-25	-83,3%

Iš gautų duomenų palyginimo matyti, kad unifikuotos sistemos papildymas naujo produkto testavimu atliekamas žymiai sparčiau. Užduoties atlikimo laikas sutrumpėjo 34,75 h. ir tai yra 24,5% sutrumpėjimas. Įrankio diegimas gamybos kompiuteriuose sutrumpėjo dar drastiškiau, sutrumpėjimas siekia 25 min., o tai yra 83,3% trumpiau.

4.3.5. Tyrimo išvados

Atlikus tyrimą nustatyta, kad sukurta unifikuota sistema leido sumažinti testavimo įrangos kūrimo proceso laikas. Nustatyta, kad naujo produkto testavimo įrangos kūrimo laikas sumažėjo ~34,75 h. Pasiektas 24,5% laiko sumažėjimas. Taip pat sutrumpėjo ir programinės įrangos diegimo gamybos kompiuteriuose laikas. Sutrumpėjimas siekia 25 min., o tai yra 83,3% sutrumpėjimas.

Išvados

1. Literatūros analizės metu išanalizuoti rinkoje egzistuojantys testavimo metodai, nustatyti pagrindiniai testavimo metodų skirtumai, privalumai, trūkumai. Apžvelgtos rinkoje egzistuojančios testavimo sistemos. Įgyvendinimui pasirinktas funkcinio testavimo metodas.
2. Unifikuota sistema automatiniam aparatinės įrangos testavimui gamybos linijoje sukurta ir įdiegta sėkmingai. Sistema įgyvendina visus užsakovo iškeltus reikalavimus.
3. Ištyrus ir išanalizavus sistemos kokybę nustatyta, kad sistema atitinka užsakovo kokybės reikalavimus, yra tinkama naudoti. Sistemos kokybės tyrimo metu taip pat ištirtas sistemos prižiūrimumas ir sudėtingumas, suformuotos sistemos tobulinimo rekomendacijos.
4. Pirmojo eksperimentinio tyrimo metu nustatyta, kad įdiegta sistema sumažino gamybinės operacijos atlikimo trukmę. Nustatyta, kad vieno gaminio testavimo ciklas sutrumpėjo vidutiniškai ~2,5 min., o tai yra 48% sutrumpėjimas.
5. Antrojo eksperimentinio tyrimo metu nustatyta, kad sukurta unifikuota sistema leidžia sumažinti testavimo įrangos kūrimo proceso laiką. Nustatyta, kad naujo produkto testavimo įrangos kūrimo laikas sumažėjo ~34,75 h. Pasiektas 24,5% laiko sumažėjimas. Taip pat sutrumpėjo ir programinės įrangos diegimo gamybos kompiuteriuose laikas. Sutrumpėjimas siekia 25 min., o tai yra 83,3% sutrumpėjimas.

Literatūros sąrašas

- [1] N. Gavrilenko, „What is Automated Hardware Test Today?“ <https://www.exsensio.com/blog/what-is-automated-hardware-test> (žiūrėta lapkr. 08, 2020).
- [2] P. Vitoriano, „Automatic Optical Inspection for Surface Mounting Devices with IPC-A-610D compliance“, Žiūrėta: lapkr. 10, 2020. [Interaktyvus]. Adresas: https://www.academia.edu/4616712/Automatic_Optical_Inspection_for_Surface_Mounting_Devices_with_IPC_A_610D_compliance.
- [3] O. Hecht ir G. Dishon, „Automatic optical inspection (AOI)“, 40th Conference Proceedings on Electronic Components and Technology, geg. 1990, p. 659–661 t.1, doi: 10.1109/ECTC.1990.122259.
- [4] A. Cort, „Functional Testing of PCBs“. <https://www.assemblymag.com/articles/83988-functional-testing-of-pcbs?v=preview> (žiūrėta lapkr. 13, 2020).
- [5] „PXI Systems Alliance - Home“. <http://www.pxisa.org/Default.aspx> (žiūrėta lapkr. 13, 2020).
- [6] NI, „Automated Test Software Suite“. <https://www.ni.com/en-us/shop/software/products/automated-test-software-suite.html> (žiūrėta lapkr. 13, 2020).
- [7] „IEEE Standard Test Access Port and Boundary-Scan Architecture“, IEEE Std 11491-1990, p. 1–139, geg. 1990, doi: 10.1109/IEEESTD.1990.114395.
- [8] R. Johnson ir S. Christie, „Development Tools Product Marketing Intel Corporation“, p. 26.
- [9] „Basic Things You Should Know About In-Circuit Testing - Accelerated Assemblies Inc.“ <https://www.acceleratedassemblies.com/blog/basic-things-you-should-know-about-in-circuit-testing> (žiūrėta lapkr. 11, 2020).
- [10] A. Albee ir T. Inc, „The Evolution of ICT“. https://www.circuitinsight.com/pdf/evolution_ict_changing_pcb_technologies_ipc.pdf (žiūrėta lapkr. 08, 2020).
- [11] „Offline Test Solutions“, Teradyne. <https://www.teradyne.com/pbt-offline-test-solutions/> (žiūrėta lapkr. 08, 2020).
- [12] „Functional Test | FCT | Low Cost Test Strategy| Caliber Interconnect Solutions“. <https://caliberinterconnect.net/pcba.php> (žiūrėta lapkr. 08, 2020).
- [13] „Automated test software suit“. <https://www.ni.com/en-us/shop/software/products/automated-test-software-suite.html> (žiūrėta lapkr. 08, 2020).
- [14] „About Us - ALeader-Europe provides a wide range of AOI and SPI systems“, ALeader-Europe. <https://aleader-europe.com/about-us/> (žiūrėta lapkr. 14, 2020).
- [15] „ISO/IEC 25010:2011“, ISO. <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/57/35733.html> (žiūrėta lapkr. 20, 2021).
- [16] D. Coleman, D. Ash, B. Lowther, ir P. Oman, „Using metrics to evaluate software system maintainability“, *Computer*, t. 27, nr. 8, p. 44–49, rugpj. 1994, doi: 10.1109/2.303623.
- [17] I. Heitlager, T. Kuipers, ir J. Visser, „A Practical Model for Measuring Maintainability“, *6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007)*, Lisbon, Portugal, rugs. 2007, p. 30–39. doi: 10.1109/QUATIC.2007.8.
- [18] „Understanding Software Metrics - Designite“. <https://www.designite-tools.com/blog/understanding-software-metrics> (žiūrėta bal. 23, 2022).
- [19] „RUT950 - 4G LTE Wi-Fi Dual-SIM Router“, *Teltonika Networks*. <https://teltonika-networks.com/product/rut950/> (žiūrėta bal. 27, 2022).
- [20] „RUT360 - LTE Industrial Router“, *Teltonika Networks*. <https://teltonika-networks.com/product/rut360/> (žiūrėta bal. 27, 2022).

Priedai

1 Priedas. Programų sistemos perdavimo ir aprobavimo aktas



Tel. +370 3 721 6110

www.teltonika-networks.com
networks@teltonika.lt

PROGRAMŲ SISTEMOS PERDAVIMO IR APROBAVIMO AKTAS

202 2 m. sausio 11 d.

Programų sistemos pavadinimas „Unifikuota sistema automatiniam aparatinės įrangos testavimui gamybos linijoje“

Kūrinio tipas Gamybinių procesų programinė įranga.

Programų sistemos sukūrimo data 202 1 m. gruodžio 10 d.

Kūrinio įteikimo UŽSAKOVUI data 202 1 m. gruodžio 13 d.

Užsakovo arba trečiojo asmens kūrinio aprobavimo rezultatas:

Programų sistema „Unifikuota sistema automatiniam aparatinės įrangos testavimui gamybos linijoje“ atitinka visus išskeltus reikalavimus, Sistema taip pat buvo teigiamai įvertinta testavimo metu ir bandomojo diegimo metu. **Sistema yra tinkama naudoti.**

Kūrinio aprobavimo data 202 2 m. sausio 11 d.

AUTORIAI:

Audrius Andrijaitis
(vardas, pavardė)

Aunt
(parašas)

UŽSAKOVAS:

Programuotojų skyriaus vadovas
Gintautas Beržunskis
(vardas, pavardė)

[Parašas]
(parašas)

UAB TELTONIKA NETWORKS
K. Baršausko g. 66, LT-51436
Kaunas, Lietuva

Įmonės kodas 305579419
PVM mokėtojo kodas LT100013223510

AB Swedbank
LT78 7300 0101 6274 0111
S.W.I.F.T. HABALT22

