

Padmarajs Nidagundi

UZ LEAN CANVAS MODEĻA BALSTĪTA PROGRAMMATŪRAS TESTĒŠANAS STRATĒGIJA

Promocijas darba kopsavilkums



RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte
Lietišķo datorsistēmu institūts

Padmarajs Nidagundi (*Padmaraj Nidagundi*)

Studiju programmas "Datorsistēmas" doktorants

UZ LEAN CANVAS MODEĻA BALSTĪTA PROGRAMMATŪRAS TESTĒŠANAS STRATĒGIJA

Promocijas darba kopsavilkums

Zinātniskie vadītāji:

profesore *Dr. sc. ing.*

MARINA UHANOVA

profesors *Dr. habil. sc.ing.*

LEONĪDS NOVICKIS

RTU Izdevniecība
Rīga 2022

Nidagundi P. Uz *Lean Canvas* modeļa balstīta programmatūras testēšanas stratēģija. Promocijas darba kopsavilkums. Rīga: RTU Izdevniecība, 2022. 29 lpp.

Publicēts saskaņā ar promocijas padomes "RTU P-07" 2021. gada 25. augusta lēmumu, protokols Nr. 21-7.

PROMOCIJAS DARBS IZVIRZĪTS ZINĀTNES DOKTORA GRĀDA IEGŪŠANAI RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ

Promocijas darbs zinātnes doktora (*Ph. D.*) grāda iegūšanai tiek publiski aizstāvēts RTU promocijas padomes atklātā sēdē 2022. gada 21. martā plkst. 14.30 tiešsaistē <https://rtucloud1.zoom.us/j/93987854650>.

OFICIĀLIE RECENZENTI

Profesors *Dr. sc. ing. Jānis Grabis*,
Rīgas Tehniskā universitāte

Profesors *Dr. sc. ing. Igors Lemberskis*,
Ventspils Augstskola, Latvija

Asoc. profesors *Ph. D. Dedžans Džokičs (Dejan Jokic)*,
Starptautiskā *BURCH* universitāte, Bosnija un Hercegovina

APSTIPRINĀJUMS

Apstiprinu, ka esmu izstrādājis šo promocijas darbu, kas iesniegts izskatīšanai Rīgas Tehniskajā universitātē zinātnes doktora (*Ph. D.*) grāda iegūšanai. Promocijas darbs zinātniskā grāda iegūšanai nav iesniegts nevienā citā universitātē.

Padmarajs Nidagundi _____ (paraksts)

Datums: _____

Promocijas darbs ir uzrakstīts angļu valodā. Tajā ir ievads, četras nodaļas, secinājumi, literatūras saraksts ar 187 atsauču avotiem un pielikumi. Darbā iekļauti 45 attēli un 23 tabulas, kopā 175 lappuses neskaitot pielikumus.

SATURS

1. DARBA VISPĀRĒJS RAKSTUROJUMS.....	5
2. GALVENIE KONCEPTI.....	10
3. TESTĒŠANAS STRATĒGIJAS VIZUALIZĀCIJAS ASPEKTI	12
3.1. Programmatūras testēšanas stratēģijas vizualizācija	12
3.2. <i>Lean Canvas</i> modelis	12
4. VIZUALIZĀCIJAS PIEEJAS IZSTRĀDE	14
4.1. <i>Test Activity Building Block (TABB)</i>	14
4.2. Prototipa sistēmas projektēšana.....	18
4.3. Jaunas stratēģijas adaptēšana.....	20
5. APROBĀCIJA UN <i>TEST ACTIVITY BUILDING BLOCK</i> STRATĒGIJA.....	22
5.1. <i>Test Activity Building Block (TABB)</i> novērtēšana.....	22
5.2. Izpētes joma	22
5.2.1. Drošības testēšanas gadījumu izpēte	23
5.3. Gadījumu izpētes rezultātu analīze	25
SECINĀJUMI UN TURPMĀKĀ DARBĪBAS JOMA.....	27
LITERATŪRAS SARAKSTS	28

1. DARBA VISPĀRĒJS RAKSTUROJUMS

Tēmas aktualitāte

Tehnoloģiju straujā attīstība ir devusi daudz priekšrocību cilvēku dzīvē, mūsdienās tehnoloģijas arvien vairāk tiek integrētas arī programmatūrā. Tās ir kļuvušas par neatņemamu sastāvdaļu ikdienas jautājumu risināšanā. Bleks (*Black*) [1] norādīja, ka programmatūras izstrāde ir virzījusies uz spējās izstrādes pieeju (*Agile*) un ka ikreiz, kad programmatūra tiek izstrādāta, tā ir jātestē, lai izlabotu defektus, pirms tā tiek nodota lietotājiem. Labi pārbaudīta programmatūra var nodrošināt klientu apmierinātību. Programmatūras testēšanai ir gara vēsture, un daudzi autori ir devuši savu ieguldījumu šīs tēmas izpētē, kā minēts [2], [3], [4] un [5]. Testēšanas stratēģija ir ietvars, kas raksturo programmatūras testēšanas tehniku programmatūras izstrādes dzīves ciklā. Tomēr *Lean Canvas* modeļa izmantošana programmatūras testēšanas procesā un testēšanas stratēģijas vizualizācija dažādos līmeņos nav pietiekami izpētīta [17] un [18].

Saskaņā ar Pasaules kvalitātes ziņojumiem [20] un [21], kas tiek publicēti katru gadu, programmatūras testēšanas nozarē pastāv pastāvīga prasmju atšķirība testēšanas stratēģiju izstrādē. Testa stratēģijas projekta priekšrocības ietver testēšanas procesu attīstību, lai paātrinātu dokumentācijas apriti, risinātu testēšanas pārvaldības jautājumus un samazinātu programmatūras izstrādes izmaksas un laiku. Programmatūras izstrādes uzņēmumi turpina uzskatīt testēšanas stratēģijas izstrādi par atsevišķu procesu. Testēšanas stratēģijas izstrādes prakse ne vienmēr tiek uzskatīta par ļoti svarīgu, bieži vien – tikai par aspektu, kas palielina izmaksas. Saskaņā ar [7] ir nepieciešams skaidrs un atbilstošs process vai testēšanas stratēģija. Šī pētījuma mērķis ir piedāvāt *Lean Canvas* modeļa vizualizēto dizaina rīku, izmantojot testa stratēģijas vizualizāciju, kas var kalpot šīm vajadzībām.

Pētījuma objekts ir programmatūras testēšanas stratēģijas vizualizācija. *Lean Canvas* modeļa stratēģijas ieviešana vizualizācijas procesā ļauj samazināt testēšanas procesa izmaksas, paātrināt dokumentācijas apriti un risināt testēšanas komunikācijas un pārvaldības jautājumus, kas rodas programmatūras izstrādes procesā. Tas palīdz samazināt izmaksas programmatūras izstrādē un testēšanas procesa optimizācijā, kā izklāstīts [19].

Pētījums piedāvā jaunu vizualizētu testēšanas stratēģiju programmatūras izstrādei, ko ātri var ieviest testa vadītājs, testa grupas vadītājs, testētājs, komandas dalībnieks vai uzņēmums. Šādā veidā ātri tiek radīta un pārvaldīta pārredzama testēšanas stratēģija, uzraugot programmatūras izstrādes dzīves ciklus. Ja programmatūras testēšana netiek veikta atbilstoši, klients saņem zemas kvalitātes programmatūru, kas rada klientu neapmierinātību ar gala produktu. Dažos gadījumos organizācijas saskaras ar ievērojamiem finansiāliem zaudējumiem, kas radušies programmatūras kļūdu dēļ.

Saskaņā ar [17] programmatūras izstrādes nozarei ir nepieciešama atbilstoša projektu testēšanas stratēģija atbilstoši projekta darbības jomai un prasībām. Saskaņā ar programmatūras testēšanas dokumentu *IEEE 29119* pareizas testēšanas stratēģijas izvēle ir noteicošais faktors programmatūras testēšanā. Vairumā gadījumu programmatūras testēšanas stratēģijas plāns ir testa plāna apakšsadaļa.

Pētījuma mērķis ir izstrādāt programmatūras testēšanas stratēģiju jaunas vizualizētas stratēģijas izstrādes ziņā. Tā pamatā ir lietojumprogrammu komponenti izstrādātajā programmatūrā vai izstrādes procesā, kas koncentrējas uz produkta funkcionalitāti.

Ierosinātajā stratēģijā galvenā uzmanība jāpievērš vispārējai projekta vizualizācijai. Tā nosaka testēšanas veidus, būtiskākās programmatūras sastāvdaļas, kas attiecīgi ir vissvarīgākās priekšrocības un vislielākais risks, tādējādi iezīmējot atšķirību, salīdzinot ar tradicionālajām testēšanas stratēģijām. Testēšanas stratēģijas izstrādes procesa ieviešana sniedz būtisku labumu programmatūras izstrādes dzīves ciklā. Saskaņā ar [20] un [21] programmatūras izstrādes uzņēmumi turpina uzskatīt “testēšanas stratēģijas izstrādes” praksi par nebūtisku procesa daļu, un, lai risinātu šo problēmu, ir jānovērš prasmju trūkums. Pētniecības procesā autors konstatēja, ka dažos uzņēmumos stratēģijas izstrādes process ir saistīts ar liela apjoma dokumentāciju, nav atjaunināts atbilstoši mainīgajām prasībām vai vispār nav

ieviests. Pētījuma autors plāno piedāvāt vizualizētu programmatūras testēšanas stratēģiju kā instrumentu, kas var kalpot šāda veida uzņēmumiem un arvien vairāk veicināt atbilstošas programmatūras testēšanas prakses izmantošanu.

Promocijas darba objekts un priekšmets

Pētījuma objekts ir programmatūras testēšanas stratēģijas vizualizācija.

Pētījuma priekšmets ir vizualizētās programmatūras testēšanas stratēģijas izstrāde dažādos programmatūras izstrādes dzīves ciklos un programmatūras testēšanas veidos.

Pētījuma hipotēze

Programmatūras testēšanas stratēģija ietver tādus posmus kā plāns, dizains, izpilde, analīze un ziņošana programmatūras izstrādes dzīves ciklā (*Software Development Life Cycle, SDLC*). Aizstāvēšanai izvirzāmās tēzes:

- vizualizēta programmatūras testēšanas stratēģija uzlabo testa procesu, darba pārredzamību, testa komandas resursu apstrādi un attālinātu komandas komunikāciju gan funkcionālā, gan nefunkcionālā testēšanā;
- ierosinātā pieeja palīdzēs samazināt ilgo dokumentēšanas darbu un isā laikā pārskatāmākā veidā radīt efektīvu testēšanas stratēģiju.

Hipotēze, kas jāpierāda promocijas darbā, ir šāda: lai pārvarētu iepriekš minētās problēmas, ir izstrādāta, ieviesta un novērtēta vizualizētā testēšanas stratēģija.

Promocijas darba mērķis

Promocijas darba mērķis ir izstrādāt vizualizētu programmatūras testēšanas stratēģiju, izmantojot *Lean Canvas* modeli. Saskaņā ar [30] programmatūras kvalitātes uzlabošana joprojām ir nozīmīgs izaicinājums dažādu veidu programmatūras izstrādes dzīves ciklos daudzām organizācijām, un tā kļūst vēl būtiskāka galvenokārt tad, ja sadalītās grupas strādā attālināti un testē kompleksus produktus. Programmatūras testēšanas process ietver daudzas darbības, tostarp dažādus testēšanas līmeņus un dažādus resursus. Autors plāno izpētīt un izprast programmatūras testētāju iespējas izstrādāt vizualizētu *Lean Canvas* modeļa pieeju un izpētīt tās ietekmi uz saziņas, sadarbības un dokumentācijas aprites procesu.

Promocijas darba uzdevumi

Lai sasniegtu promocijas darba galveno mērķi, ir jāizpilda šādi uzdevumi:

- izpētīt esošos programmatūras testēšanas procesus un testēšanas stratēģijas, analizēt vizualizācijas nepieciešamību stratēģijas izveidē;
- apzināt jaunāko programmatūru stratēģiju galvenās priekšrocības un trūkumus un analizēt stratēģijas vizualizāciju;
- izstrādāt vizualizētu testēšanas stratēģiju, pamatojoties uz dažādiem reāliem projektiem un programmatūras prasībām;
- izstrādāt kritērijus, lai novērtētu ierosināto vizualizēto testēšanas stratēģiju;
- ieviest vizualizēto programmatūras testēšanas stratēģiju programmatūras izstrādes projektos un apkopot atsauksmes;
- izpētīt un dokumentēt priekšrocības un ierobežojumus, pamatojoties uz praktiskajiem rezultātiem, apkopot citu projekta un grupas dalībnieku ieteikumus;
- noteikt ierosinātās stratēģijas iespējamo turpmāko attīstības jomu.

Pētījuma metodoloģija

Promocijas darbā izmantotas vairākas metodes.

Pētījums sākas ar esošo dažādo testēšanas stratēģiju un to dokumentācijas analīzi. Kvalitatīvās – izpētes un novērošanas – pētījuma metodes izmantotas, lai noteiktu programmatūras testēšanas stratēģijas vizualizācijas dažādu pieeju stiprās un vājās puses. Tika analizētas sešas dažādas stratēģijas vizualizācijas iespējas un noteikti ierobežojumi, kas aprakstīti 2.3. sadaļā. Pēc tam izstrādāta programmatūras testēšanas stratēģijas vizualizācija, izmantojot *Lean Canvas* testēšanas darbību pamatmoduļus, kas minēti 3.2. sadaļā.

Lai pētītu un izstrādātu prototipa sistēmu, šajā pētījumā autors apkopoja sistēmas projekta prasības, tehnoloģijas, sistēmas lietotājus un lietošanas piemērus. No apkopotajām prasībām 3.3. sadaļā tika simulēts tīmekļa portāls.

Iegūtie kvantitatīvie un kvalitatīvie dati (aptaujās balstītas anketas, tiešsaistes lietotāju atsauksmes, lietošanas gadījuma metode) tika izmantoti, lai noteiktu autora ierosinātās pieejas priekšrocības un ierobežojumus testēšanas stratēģijas vizualizēšanai. Kvantitatīvās pētījuma metodes izmantotas, lai analizētu dokumentācijas apjomu.

Visbeidzot, situācijas izpētes rezultātu analīzes rezultātā tika reģistrēts dokumentācijas apjoms lapusēs un stundās katrai programmatūras versijai pēc piedāvātas stratēģijas *Test Activity Building Block (TABB)* ieviešanas.

Daudzi teorētisko aprēķinu un rezultātu grafiskā attēlojuma procesi tika sasniegti, izmantojot dažādas programmatūras sistēmas, tostarp:

- *WordPress.com* (PHP sistēma satura pārvaldības sistēmas izstrādei);
- *MySQL* (lietotāju informācijas un satura pārvaldības sistēmas uzglabāšana);
- *GitHub.com* (pirmkoda uzturēšanas versiju kontrole);
- *Atlassian.com* (programmatūras izstrādes un sadarbības rīki problēmu un projektu izsekošanai);
- *Google* rasējumi (*TABB* izveide, uzglabāšana un atjaunināšana testēšanas darbībām);
- *Google* formu programmatūra (uzskaite un datu vākšana);
- *Microsoft Word* (esošo dokumentu analīze un jaunu dokumentu izveide).

Zinātniskais jaunieguvums

- Ir izstrādāta jauna pieeja testēšanas stratēģijas vizualizēšanai, lai īstenotu *Lean Canvas* modeli balstītu programmatūras testēšanas stratēģiju.
- Jauna pieeja – *TABB* stratēģija (*Test Activities Building Block*) tika izmantota, lai atrastu pareizās testēšanas metrikas *Lean Canvas* vizualizētajai shēmai, izmantojot definētos transformācijas modeļus.
- Jaunā stratēģija ir izvērtēta dažādos projektos, un dati ir apkopoti.

Praktiskā nozīme

Izstrādātā stratēģija tika ieviesta vairākos projektos, apkopotas konstruktīvas atsauksmes, lai izprastu pētījuma ietekmi. Autors ir apkopojis kritiku un statistikas datus *TABB* stratēģijas pilnveidošanai.

- Izstrādāti *TABB* stratēģijas kritēriji. Izskaidrots, kā noteikt efektīvu testēšanas metriku *Lean Canvas* shēmai.
- Veikti eksperimenti ar mērķi izmantot *Lean Canvas* modeļa testēšanas stratēģiju programmatūras izstrādes projektos programmatūras izstrādes uzņēmumos.
- Turpmāk programmatūras uzņēmumi var izmantot apkopotos praktiskos rezultātus, lai uzlabotu programmatūras izstrādes un testēšanas procesus, galvenokārt izmantojot *Lean Canvas* modeli balstītu *Test Activity Building Block (TABB)* stratēģiju.

- Izveidots atvērtā pirmkoda projekts, kur var izmantot integrētos stratēģijas modeļus.

Izpētes rezultātu aprobācija

Pētījuma rezultāti prezentēti astoņās starptautiskās konferencēs Čehijā, Ukrainā, Lietuvā, Austrijā un Latvijā.

1. Symposium for Young Scientists in Technology, Engineering and Mathematics, Kaunas, Lithuania, 28 April 2017.
2. Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference. Rezekne Academy of Technologies, Rezekne, Faculty of Engineering, Latvia, 29 April 2017.
3. Proceedings of the IRES International Conference, Vienna, Austria, 26 November 2017.
4. XII-th International Scientific and Technical Conference “Computer Science and Information Technologies”. Lviv, Ukraine, 6 September 2017.
5. CSOC 2017, Faculty of Applied Informatics, Tomas Bata University in Zlin, Czech Republic, 27 June 2017.
6. RTU 58th International Scientific Conference, Riga Technical University, Riga, Latvia, on 13 October 2017.
7. International Conference on Information Technologies, IVUS 2018; Kaunas; Lithuania, 27 April 2018.
8. RTU 59th International Scientific Conference, Riga Technical University, Riga, Latvia, 11 October 2018.

Pētījuma rezultāti ir publicēti 11 publikācijās.

1. Nidagundi P., Introduction to Investigation and Utilizing Lean Test Metrics. In: Agile Software Testing Methodologies. Int. Journal of Engineering Research and Applications. ISSN: 2248-9622, Vol. 6, Issue 4, (Part – 1) April 2016, pp. 13–16. (INDEXED: Google Scholar).
2. Nidagundi P., Novickis L. Possibilities about the design lean canvas model and its adaptation in the agile testing. Symposium for Young Scientists in Technology, Engineering and Mathematics, Volume 1853, SYSTEM 2017, Kaunas, Lithuania, 28 April, pp. 20–23. (INDEXED: SCOPUS, Google Scholar).
3. Nidagundi P., Novickis L. Introduction to Lean Canvas Transformation Models and Metrics. In: Software Testing, APPLIED COMPUTER SYSTEMS, Year 2016, pp. 30–36. (INDEXED: Google Scholar, Web of Science).
4. Nidagundi P., Lukjanska M. Introduction to adoption of lean canvas in software test architecture design. Computational Methods in Social Sciences, Volume 4, Number 2, 2017, pp. 23–31 (9). (INDEXED: Google Scholar, ProQuest, RePEc, DOAJ, EconPapers, Index Copernicus International).
5. Nidagundi P., Novickis L. Introducing Lean Canvas Model Adaptation in the Scrum Software Testing, Procedia Computer Science. Volume 104, 2017, pp. 97–103. (INDEXED: SCOPUS, Web of Science, ScienceDirect, Google Scholar).
6. Nidagundi P., Novickis L. Towards Utilization of Lean Canvas in the DevOps Software. Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference, ISSN 2256-070X, June 2017, pp. 107–111. (INDEXED: SCOPUS, Google Scholar).
7. Nidagundi P., Novickis L. Towards Utilization of a Lean Canvas in the Testing Extra-Functional Properties, Software Engineering Trends and Techniques in Intelligent Systems June 2017, pp. 349–354. (INDEXED: SCOPUS, Web of Science, Google Scholar).
8. Nidagundi P., Stepanova V. Survey on Software Test Strategy. Proceedings of the IRES International Conference, Austria, Vienna, November 2017, pp. 26–27. (INDEXED: Google Scholar, WORLD RESEARCH LIBRARY).
9. Nidagundi P., Novickis L. New method for mobile application testing using lean canvas to impro-

- ving the test strategy, XII-th International Scientific and Technical Conference “Computer Science and Information Technologies”. Sept. 2017. (INDEXED: IEEE Xplore digital library, SCOPUS, Web of Science, Google Scholar).
10. Nidagundi P., Novickis L. Towards Utilization of a Lean Canvas in the Biometric Software Testing. A JOURNAL OF MULTIDISCIPLINARY SCIENCE AND TECHNOLOGY, IIOABJ, 2017, Vol. 8 (Supple 3), pp. 32–36.
 11. Nidagundi P., Uhanova M. Software application security test strategy with lean canvas design, International Conference on Information Technologies. April 2018, pp. 50–53. (INDEXED: SCOPUS, Google Scholar).

Promocijas darba struktūra un apjoms

Promocijas darbu veido ievads, četras nodaļas, kas risina teorētiskus un praktiskus jautājumus, secinājumi, bibliogrāfija un pielikumi. Promocijas darba apjoms ir 175 lappuses, tajā ir 43 attēli un 24 tabulas, bibliogrāfijā ir 199 atsauces.

Promocijas darba kopsavilkumu veido piecu nodaļu ievadi, kur risināti teorētiskie un praktiskie jautājumi, secinājumi, bibliogrāfija un pielikumi.

1. nodaļā sniegts pētījuma vispārējs apraksts, t. i., norādīta risināmā problēma, definēts mērķis, uzdevumi un tēzes, kas jāaizstāv, kā arī izklāstīta pētniecības metodika un promocijas darba struktūra.

2. nodaļa veltīta galvenajām koncepcijām, stratēģijas veidiem un standartiem.

3. nodaļā iekļauts vizualizētas programmatūras testēšanas stratēģijas projekts, stratēģijas vizualizācija un *Lean Canvas* projekta raksturojums.

4. nodaļā atspoguļota izstrādes un vizualizētas testēšanas stratēģijas izstrāde. Ieviesta *TABB (Test Activities Building Block)* stratēģija, izveidots skaidrojums un sistēmas projekta lietošanas piemērs, kā arī sniegts prototipa sistēmas projekta *pārskats, jaunas stratēģijas ieviešana, ņemot vērā sistēmas dizaina nozīmi un tās ierobežojumus*.

5. nodaļā novērtēta *Test Activity Building Block* ieviešana, sniegts eksperimentu pārskats un situācijas izpētes rezultātu analīze.

Secinājumos un turpmāko pētījumu jomā sniegts pārskats par pētījuma rezultātiem, iegūtajiem atklājumiem un turpmāko pētījumu virzieniem.

2. GALVENIE KONCEPTI

Testēšanas stratēģijas definīcija ir šāda: testēšanas stratēģija ir ietvars, kas apraksta programmatūras testēšanas tehniku programmatūras izstrādes dzīves ciklā.

Tā ir paredzēta, lai informētu projektu vadītājus, komandas biedrus, testētājus un izstrādātājus par dažiem būtiskiem testēšanas procesa jautājumiem. Programmatūras izstrādes un testēšanas procesā bieži izmanto proaktīvas vai reaktīvas testēšanas stratēģijas [28].

- a) Proaktīvā stratēģija – šajā stratēģijā testēšanas process tiek iesākts, cik drīz vien iespējams, lai atklātu un novērstu programmatūras defektus pirms programmatūras struktūras izveides.
- b) Reaktīvā stratēģija – stratēģija, kurā testēšana netiek sākta, kamēr nav pabeigta lietotāja saskarņu projektēšana un kodēšana.

Lai izveidotu labas kvalitātes programmaproduktu, stratēģijas pieejas un kvalitātes nodrošināšanas mērķa noteikšana nodrošina netraucētu ieinteresēto personu, kvalitātes nodrošināšanas speciālistu un izstrādātāju saziņu. Stratēģijas izprašana ir pirmais solis, lai pieņemtu lēmumu par stratēģiju. Programmatūras inženierijas sākumposmā labas dokumentācijas radīšana ir vissvarīgākā pamatprakse, kas jāisteno programmatūras izstrādes laikā.

Ņemot vērā programmatūras testēšanas stratēģijas dokumentāciju, sākotnējā posmā tā izskatās maza, un vēlākos posmos tā ir jāmaina biežāk atbilstoši mainīgajām prasībām un projekta darbības jomai. Turklāt testētājs sāk pārbaudi un izpēti, un domā par plašākas informācijas atklāšanu laika gaitā. Testēšanas stratēģija attiecīgi pieaug. Laika gaitā ir ļoti grūti pārvaldīt tik liela apjoma dokumentāciju.

Saskaņā ar [29] un [30] pašreizējā nozares tendence ir spējās izstrādes manifests, kur galvenā uzmanība pievērsta darba programmatūras izstrādei, izmantojot visaptverošu dokumentāciju. Autora skatījumā, ir jādomā par iespējām vizualizēt pašas programmatūras testa stratēģiju. Vizualizētā stratēģija ir paredzēta, lai akcentētu testēšanas riskus, iespējamās testa stratēģijas dokumenta komponentus, kā arī palielinātu komandas saziņas ietekmi, lai samazinātu testēšanas izmaksas un dokumentācijas intensitāti.

Autors analizē esošās programmatūras testēšanas stratēģijas. Autors ir izveidojis 2.1. tabulu, kurā ir apkopoti stratēģijas veidi, riska faktori, biznesa mērķi, nepieciešamās testēšanas komandas prasmes, mērķi, noteikumi un vizualizētā testa stratēģija.

2.1. tabula

Stratēģiju veidi un standarti

Stratēģiju veidi	Riska faktori	Testēšanas veids	Testēšanas komandai nepieciešamās iemaņas	Darbības joma un mērķi	Noteikumi
Analītiskā	Jā	–	–	–	–
Veidota pēc paraugmodeļa	–	Jā	–	–	–
Metodiskā stratēģija	–	–	–	–	Jā
Procesiem vai standartiem atbilstošas stratēģijas	–	–	–	–	Jā
Dinamiskā stratēģija	Jā	–	–	–	–
Konsultatīvā vai pārvaldītā stratēģija	Nē	–	–	–	–
Pretragersa	Jā	–	–	–	–
Reaktīvā testa stratēģija	Nē	–	–	–	–
Standarta atbilstības testēšanas stratēģija	–	–	–	–	Jā

Promocijas darba autors iesaka problēmas atrisinājumu vizualizētā veidā. Autors risina šādas problēmas: kādi ir būtiskākie riski, ko produkta īpašnieks vēlas zināt, un kāds mērķis ir jāsasniedz. Šādā veidā stratēģija piedāvā risinājumu sarežģītai problēmai par to, kā apmierināt ieinteresēto personu vajadzības pēc informācijas un kā to efektīvi darīt. Ir daudz aspektu, kas jāapsver ceļā uz labāku programmatūras testēšanas stratēģiju.

3. TESTĒŠANAS STRATĒGIJAS VIZUALIZĀCIJAS ASPEKTI

Pētniecības problēmas konteksts ir šāds: programmatūras testēšana kļūst ļoti svarīga, kā arī problemātiska jebkuram programmatūras izstrādes uzņēmumam, mēģinot piegādāt bezkļūdu programmatūru gala klientiem. Tradicionālā stratēģija programmatūras lietojumprogrammu testēšanā tika pārņemta no uzņēmumiem, kas pastāv jau ilgu laiku. Saskaņā ar [6], [7], [8] un [9] daudzi pētniecības raksti un nozares apsekojumi pierāda, ka ir jāuzlabo programmatūras testēšana un ka ir jāpārdomā testēšanas stratēģija, testēšanas vadība, instrumenti un procesu pielāgošana.

3.1. Programmatūras testēšanas stratēģijas vizualizācija

Saskaņā ar [23], [24] un [25] programmatūras izstrādes projektiem ir tūlītēja piekļuve projektam būtiskai informācijai, kas nozīmē, ka tiem ir iespējas uzlabot programmēšanas un testēšanas ātrumu, gūt labāku izpratni par uzņēmējdarbības jomu, izmantot iespējas un iegūt konkurences priekšrocības tirgū. Bet līdz šim brīdim informācija ir dokumentu veidā, kam pašiem par sevi un bez pienācīgas analīzes var būt maza nozīme. Praksē atkarībā no dokumentācijas apjoma pati interpretācija var kļūt diezgan laikietilpīga vai sniegt ārkārtīgi sarežģītu rezultātu.

Stratēģijas vizualizācija ir nepieciešama, jo shēma var nekavējoties sniegt informāciju neaizmirstamā veidā. Vizualizācijas stratēģijas uzlabotā izpratne palīdz uztvert nozīmīgas vietas tekstā, apvienojot vairākas detaļas. Vizualizējot shēmu lietotāji tiek aicināti paust savas domas, vizualizēt tekstu un padarīt to par savu komunicēt spējīgu lēmumu. Tādējādi darbības tiek atvieglotas, izmantojot vizualizāciju, kas ļauj lietotājiem izprast attiecības starp programmatūras testēšanu un rezultātiem, mijiedarboties ar vizualizētu *Lean Canvas* shēmu, un ģenerēt jaunas idejas un diskusijas.

Grupas dalībnieki, testēšanas vadītājs, grupas vadītājs un testētājs izlasa dokumentu vienas lapas apjomā un izprot padziļinātu informāciju par konkrēto produktu vai projektu, rīkiem, procesu un stratēģiju. Vizualizējot testa stratēģiju jebkurā projektā, kopējais testēšanas process ir vienkāršots.

3.2. *Lean Canvas* modelis

Šajā pētījumā autors analizēja arī dažādas esošās iespējas stratēģijas vizualizēšanai, piemēram, zivs asakas (Išikava *fishbone diagram*) jeb cēloņu un seku shēmu, stratēģijas kartes shēmu, empātijas kartes dizainu, *Lean Canvas* modeli un biznesa procesu modeli, kā arī notācijas dizainu. Autors uzskatīja, ka turpmākai testēšanas stratēģijas vizualizācijas izpētei lielāka priekšrocība ir *Lean Canvas* modelim kā pamatmodelim [17].

Saskaņā ar [10], [12], [13] un [11] *Lean Canvas* modelis ir Aleksandra Ostervaldera 2008. gadā izveidotā uzņēmējdarbības modeļa audekla (*Business Model Canvas*) pielāgojums, šo modeli izveidoja Ašs Mauria (*Ash Maurya*) savā *Lean Start-up*. To izmanto stratēģiskai pārvaldībai, efektīvām jaunuzņēmumu veidnēm un esošo uzņēmējdarbības modeļu dokumentēšanai. 3.1. attēlā sniegts standarta *Lean Canvas* veidnes piemērs. Tas ir dokuments vienas lapas apjomā, kas vizualizē dažādas produkta vai uzņēmuma daļas, piemēram, vērtības piedāvājumu, infrastruktūru, klientus un finanšu sadaļu.

PROBLĒMA	RISINĀJUMS	UNIKĀLAS VĒRTĪBAS PIEDĀVĀJUMS	NEGODĪGAS PRIEKŠROCĪBAS	KLIENTU SEGMENTI
	GALVENIE RĀDĪTĀJI		KANĀLI	
IZMAKSU STRUKTŪRA		IEŅĒMUMU PLŪSMAS		

3.1. attēls. Standarta *Lean Canvas* veidnes piemērs.

Plašā skatījumā programmatūras testēšanā iespējams izmantot arī *Lean Canvas* modeli [18], [19]. To var pielāgot *scrum* programmatūras testēšanai, kas var tieši ietekmēt programmatūras kvalitātes uzlabošanu [17]. Autors vēlējas izpētīt, vai esošais *Lean Canvas* modelis ar deviņiem moduļiem ir pietiekams testēšanas stratēģijas vizualizēšanai, vai arī nepieciešamas turpmākas izmaiņas. Šajā procesā autors pēta lietotāja pieredzi ar *Lean Canvas* modeli no personas viedokļa, šī pētījuma gadījumā “programmatūras testētājs” ir lietotājs.

Pieredzē balstīts projekts lietotāja izpratnē neattiecas uz saskarnes darbībām. Tas attiecas uz cilvēku darbībām un tehnoloģiju atbilstošu pielāgošanu. Pirmajā posmā ir jānoskaidro, kas ir projektētājs. Projektētājs ir arhitekts, celtnieks, kurš rada izdomātus artefaktus, kas vēlāk kļūst par īstiem produktiem.

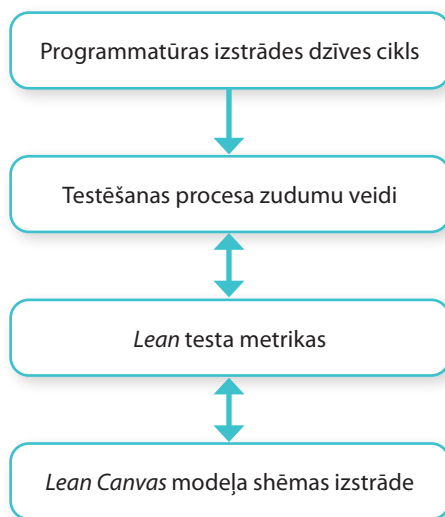
Māršs (*Marsh*) [14] apgalvo, ka darba grupa nodrošina to, ka digitālais ražojums ir izgatavots atbilstoši lietotāja vajadzībām un ieinteresēto personu prasībām. Lietotāja pieredzē balstīts projekts (*User Experience Design – UXD*) ir projektēšanas process, kura mērķis ir izstrādāt digitālās sistēmas, saskarnes un lietojumprogrammas, lai piedāvātu lietotājiem visefektīvākās un vienkāršākās ērtības. Tādējādi lietotāja pieredzē balstīts projekts ietver teoriju par daudzām disciplinām, piemēram, lietotāja saskarnes dizainu, lietojamību, pieejamību, informācijas arhitektūru un cilvēka un datora mijiedarbību.

Personāži ir izdomāti lietotāji, kuriem piemīt vispārējās to personu īpašības, kas izstrādā autora izpētīto testēšanas stratēģiju. Pirms *Lean Canvas* modeļa pārbaudes stratēģijas izstrādes ir nepieciešams noteikt personāžu vajadzības. Šajā pētījumā uzmanības centrā ir programmatūras testētājs, testa vadītājs, komandas loceklis un testa vadība. Lai radītu personāžus, autors izmantoja dinamisko pieeju. Grānts un Mitels (*Grant* un *Mittal*) [14], [16] diskutē par to, ka personāžs ir tipisks jeb standarta lietotājs, izdomāts iespējamo lietotāju attēlojums, un to izmanto prioritāšu noteikšanai un saskarņu izstrādes lēmumu vadīšanai.

Veicot izpēti attiecībā uz *Lean Canvas* modeļa pielāgojumu testēšanas stratēģijas projektam, autors pievērsās lietotāja pieredzē balstītam projektam. Šajā pētījumā autors secināja, ka stratēģijas vizualizācijai esošie deviņi apakšmoduļi nav pietiekami. Lai labāk izprastu problēmu, autors apkopoja personāžu viedokli. Nākamajā posmā autors vēlreiz izvērtēja risinājumu ar *Lean Canvas* modeļa projektu un novērtēšanas metodes. Tādējādi, vēlreiz izvērtējot idejas no personāžu viedokļa, apgūstot lietotāju pieredzē balstīta projekta un lietotāju saskarnes projekta pamatus, var formulēt ieteikumus un vadlīnijas, lai izstrādātu labāk vizualizētu *Lean Canvas* modeli balstītu pārbaudes stratēģiju.

4. VIZUALIZĀCIJAS PIEEJAS IZSTRĀDE

Programmatūras izstrādes dzīves ciklā komanda var saskarties ar dažādiem zudumu veidiem, šie zudumi rodas no dažādiem avotiem. Ja komanda nerisina ar zudumiem saistītās problēmas, projekts var aizņemt vairāk laika nekā plānots. Zudumiem var būt tieša vai netieša ietekme uz izstrādes un testēšanas dzīves ciklu, šajā situācijā programmatūras izstrādes dzīves cikla zudumu identifikācijai ir ļoti liela nozīme (4.1. att.). Zudumu klasifikācija ļāva atrast primāro galveno metanosaukumu *Lean Canvas* modeļa shēmai.



4.1. att. Programmatūras izstrādes dzīves cikla zudumu identifikācija.

4.1. Test Activity Building Block (TABB)

Ir svarīgi, lai būtu atbilstoši modeļa pamatmoduļu nosaukumi, katram modulim koncentrējoties uz atšķirīgu faktoru. Tie palīdz lietotājiem apskatoties ātri iegūt jēgpilnu informāciju. Šajā apakšnodaļā izskaidrota moduļu, apakšmoduļu, nosaukumu un subtitrēšanas procesa izveide.

Trīs galvenie posmi *Lean Canvas* modeļa testēšanas darbību projektēšanai, veidojot moduļus, ir šādi.

Pirmajā posmā tiek norādīts shēmas primārais galvenais metanosaukums. Lietotājs var izmantot esošo pārbaudes stratēģijas dokumentu vai sarīkot prāta vētru ar grupas dalībniekiem, lai izprastu izšķirošo shēmas metanosaukumu (piemēram, testēšanas rezultāti).

Otrajā posmā metavirsraksts tiek sadalīts mazākos. Izmantojot efektīvās domāšanas principus, tiek noskaidrots svarīgais metanosaukums un pievienots shēmai. Vēlāk tiek pievienots metaapraksts.

Trešajā posmā tiek modificēts metanosaukums vai metaapraksts. Metanosaukums vai metaapraksts tiek modificēts atbilstoši programmatūras izstrādes dzīves ciklam.

Katru posmu autors ir sīki aprakstījis vizualizētajā *Lean Canvas* modeļa testēšanas stratēģijas projektā.

Vizualizētas Lean Canvas modeli balstītas testa stratēģijas projekta posmi

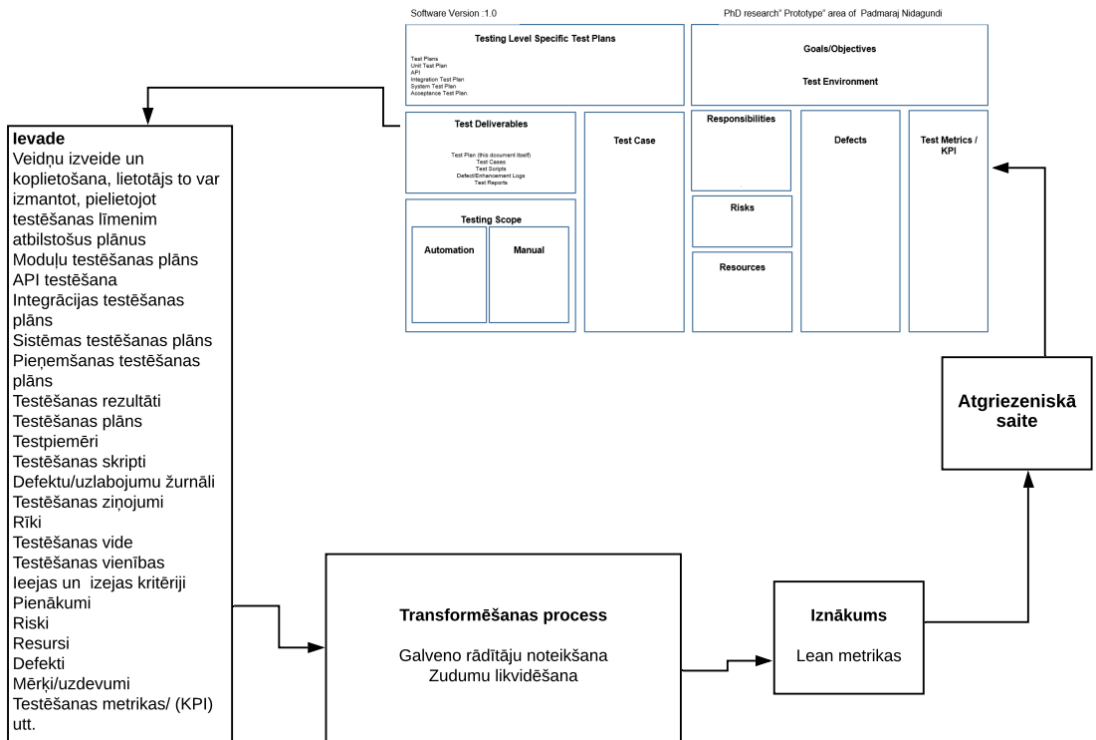
1. posms. Identificēt pašreizējo programmatūras izstrādes procesa veidu.
Šajā posmā tiek identificēts programmatūras izstrādes process (piemēram, *Agile*, *Waterfall*, *Lean*).
2. posms. Norādīt programmatūras testēšanas veidu.
Šajā posmā tiek noteikts programmatūras testēšanas veids (piemēram, regresijas tests, veikspējas tests).
3. posms. Izmantot esošo testēšanas stratēģiju vai testa dokumentāciju vai vadīt prāta vētras pasākumu kopā ar grupu, lai to izveidotu (piemēram, testēšanas plāna dokumentu, testēšanas stratēģijas dokumentu).
4. posms. Noteikt primāro galveno metanosaukumu shēmai, izmantot tukšu *Lean Canvas* veidni kopā ar shēmu un sākt apakšmoduļu pievienošanu (piemēram, testēšanas līmeņus, testēšanas apjomu, testēšanas rezultātus, risku).
5. posms. Sadalīt galveno metavirsrakstu mazākos un definēt attiecīgos galvenos metanosaukumus moduļiem un apakšmoduļiem (piemēram, manuālais, automatizācija).
6. posms. Sākt aizpildīt īso metaaprakstu apakšmoduļos (4.2. att.; piemēram, testēt darbības jomu apakšmoduļos kā testēšanas jomu ārpus darbības jomas).
7. posms. Vizualizētā testēšanas stratēģija ir gatava lietošanai.
8. posms. Ja nepieciešams, pēc katra sprinta vai izstrādes posma atkārtoti pārskatīt testēšanas stratēģiju un modificēt metanosaukumu vai metaaprakstu.

Programmas kritisko funkciju saraksts Pietiekšanās Atteikšanās Paroles atkopšana Pamatlapas pārļādēšana	Regrestestēšanas kontrolsaraksts Gadījumi ar biežiem defektiem Lietotājiem redzamās funkcijas Gadījumi, kuros pārbauda produkta pamatīpašības Funkcionalitātes gadījumi, kas nesen mainīti Robežvērtības testa gadījumi Veiksmīgas testēšanas gadījumu paraugs	Zināšanu bāzes saites	Pārļūka pārklājums <i>Chrome</i> <i>Firefox</i> <i>Edge</i>	Regresīvā testkomplekta automatizācijas rezultāti
			Regrestesta rezultāti Uzskaites pārbaude – derīgs Lietotāju plūsma – derīgs Failu sūtīšanas tests – derīgs	Riski Lietojumprogrammatūra Zināšanas Automatizācija Pārklājums
Testēšanas vide		Mērķis/uzdevumi Regrestestēšanas mērķis ir novērst nejausi ieviešanās kļūdas un pārliecināties, ka iepriekš novērstie defekti nevar atkārtoties. Šādā veidā regrestestēšana nodrošina, ka koda izmaiņas neietekmē esošās programmas funkcijas		

4.2. att. Virsrakstu izveidošana un aprakstu aizpildīšana.

Ņemot vērā 4.2. attēlā redzamo tabulu, pirmā moduļa nosaukumi nodrošina pārveidošanas procesu, un izvades atgriezeniskā saite tiek pārveidota par ievadi.

Testēšanas procesa zudumu klasifikācija *TABB* stratēģijas *Lean Canvas* modeļa shēmai redzama 4.3. attēlā.



4.3. att. TABB stratēģija Lean Canvas modeļa moduļiem un apakšmoduļiem.

Ievade. Svarīgu terminu apkopojums, ko izmanto esošajā testēšanas stratēģijā vai testēšanas procesā; ja nav vēsturisku pierādījumu testēšanas stratēģijai, tad grupa var tos iegūt prāta vētras laikā.

Pārveidošanas process. Šajā posmā ir jāatceļ nevēlamos metanosaukumus vai metaaparakstus atbilstoši efektivitātes (*lean*) principiem. Mērķis ir saglabāt svarīgus virsrakstus tikai shēmas platei [27]. Izmantojot pārveides procesu, lietotājiem ir jāievēro astoņi efektivitātes (*lean*) principi. Turklāt lietotājs var saņemt atsauksmes no grupas par to, ko viņi uzskata par būtisku nosaukumu, kas ir jāvizualizē sīkāk.

Izvide. Iegūtos galvenos rādītājus uzskata par vizualizētā *Lean Canvas* modeļa shēmas nosaukumu, apakšnosaukumiem vai īsu aprakstu.

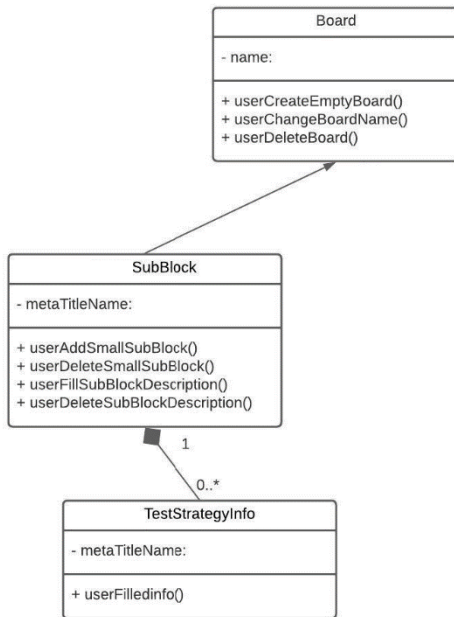
Atgriezeniskā saite. Tiek piedāvāti apkopotie galvenie rādītāji, turklāt to nosaukumi tiek atkārtoti izmantoti *Lean Canvas* modeļa shēmas platē.

Iepriekšējā sadaļā sniegts TABB apraksts *Lean Canvas* modeļa moduļu un apakšmoduļu dzīves ciklam, ko izmanto, lai izskaidrotu un vizualizētu *Lean Canvas* modeļa testēšanas stratēģiju astoņos posmos, kas ļauj saprast, kā to izmantot.

TABB formālā specifikācija

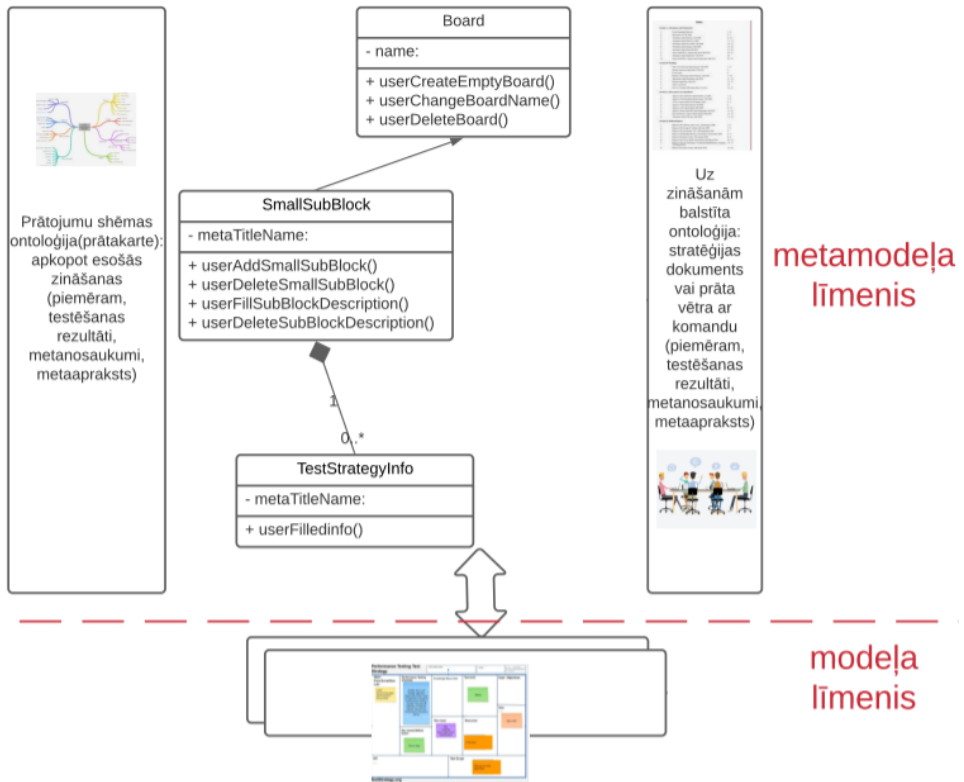
Autors novērš plaisu starp neformāliem modeļiem un formālu modeli, izmantojot vienotās modelēšanas valodas (UML) klases shēmu (4.4. att.).

Pirmajā posmā tiek izveidota tukša plate, un lietotājs pievieno atbilstošu metanosaukumu, kas balsītīs stratēģijas dokumentu vai ideju veidošanā kopā ar grupu (piemēram, testēšanas rezultāti). Vēlāk lietotājs var pievienot papildu metavirsrakstus vai noņemt tos. Lietotājs var pievienot aprakstu meta-modelim, kā jau skaidrots vizualizētajā *Lean Canvas* modeļa testēšanas stratēģijas projektā, veicot šādas darbības:



4.4. att. Formālā pieeja TABB, izmantojot UML klases shēmu.

Pētījuma rezultātu tālāk ir iespējams attīstīt kā konceptuālu risinājumu (4.5. att.). Formālās metodes ir izrādījušās efektīvas skaidrojošās sistēmas izstrādē.



4.5. att. Konceptuāls risinājums *TABB* ģenerēšanas grafiskajam attēlojumam.

Kā parādīts iepriekš, zināšanas ir metaformātā no dažādiem avotiem, tās ir atkarīgas no lietotāja, un vēlākā posmā informācija tiek ievietota *TABB* modeļa formātā. Ņemot vērā visus iepriekš minētos datus, autors paskaidro, ka prototipa sistēma ir īstenota kā *Web* lietojumprogramma un aprakstīta nākamajā apakšnodaļā.

4.2. Prototipa sistēmas projektēšana

Pētījums tika veikts, pievēršoties eksperimentālajiem pierādījumiem attiecībā uz pilotprojektu. Nosakot attiecīgās tehnoloģijas un veidojot testēšanas stratēģiju, testēšanas procesā tiek izmantots tīmekļa portāls ar lietošanai gatavu stratēģijas veidni. Vizualizēto testēšanas stratēģijas veidņu skaits ir pieejams www.teststrategy.org.

Pilotprojekts tika izveidots četros posmos:

1. posms sniedz būtiskus skaidrojumus par izmēģinājuma projekta iespējām. Šajā posmā, ņemot vērā hipotēzes testēšanas stratēģiju, tiek pētītas vizualizācijas iespējas.

2. posmā tiek izstrādātas izmēģinājuma projekta aktivitātes un iespējamie risinājumi. Šajā posmā tiek identificēti vairāki lietošanas gadījumi. Izmēģinājuma projektā tiek izstrādātas lietotāju plūsmas testēšanas stratēģijas vizualizācijai portālā www.teststrategy.org.

3. posmā tiek identificētas izmēģinājuma projekta priekšrocības un nepilnības. Šajā posmā pēc darbību īstenošanas tiek noteiktas vizualizētās programmatūras testēšanas stratēģijas priekšrocības un ierobežojumi.

4. posms sniedz īsu analīzi, apkopojumu un padomus, kā virzīties uz priekšu.

Autors ir noteicis sistēmas projekta prasības *TABB* izveidošanai. Veidojot pirmo sistēmas prototipu, autors saņēma vairākas norādes. Vēlākā posmā šīs atsauksmes tika izmantotas sistēmas un *TABB* uzlabošanai. Šajā procesā tika noteikti arī daži ierobežojumi. Triangulācijas metode parādīja, ka lietotājiem, piemēram, testētājiem, izstrādātājiem un testu vadītājiem, jāpievērš uzmanība noteiktam jautājumu kopumam, lai iegūtu atsauksmes par vizualizēto testa stratēģiju. Šos jautājumus uzdod *www.teststrategy.org* portāla individuālie lietotāji un adaptētāji.

Vizualizētās testēšanas darbību pamatmoduļu stratēģijas stiprumu un ierobežojumu izpētes procesam bija jāpiemēro pētnieciskās izpētes metode, lai attīstītu labāku izpratni par salīdzinoši jaunu pētniecības jomu. Šī metode būtībā ietver divus posmus: a) tiešsaistes lietotāju komentāru un atsauksmju analīzi; b) vairāku gadījumu izpēti, kas veikta aktuālos projektos. Autors izmantoja dažādas datu vākšanas metodes, rezultātu analīzei izmantojot triangulāciju (4.6. att.).



4.6. att. Vizualizētās testēšanas stratēģijas kvantitatīvā un kvalitatīvā triangulācija.

Pirmajā posmā tiek apkopoti un analizēti tiešsaistes lietotāju sniegtie komentāri un atsauksmes. Lielākā daļa lietotāju ir programmatūras testētāji; šie lietotāji ir agrīnie *www.teststrategy.org* lietotāji. 4.1. tabulā sniegts kvantitatīvās un kvalitatīvās intensitātes pārskats.

4.1. tabula

Vizualizētās testēšanas stratēģijas kvantitatīvā un kvalitatīvā triangulācija

Aptauja, kuras pamatā ir anketa	Kvantitatīvā izpēte prototipa izveidošanai un īstenošanai	Atlasītās dalībnieku grupas diskusija sākotnējā stadijā	Piedalījās divas grupas
		Atlasītās dalībnieku grupas diskusija pēc vizualizēta <i>Lean Canvas</i> modeļa ieviešanas	Piedalījās četras grupas
		Aptaujas jautājumi un atbildes	Piedalījās 33 personas
Tiešsaistes lietotāju recenzija un komentāri	Kvalitatīvs pētījums projekta uzlabošanai	Dažādas atsauksmes, kas saņemtas <i>www.teststrategy.org</i>	15 neatkarīgas atsauksmes par projektu
Lietošanas piemēru ieviešana	Sistēmas projekta un uzlabojumu apsvērumi	Vairāki lietotāji izmantoja gatavas <i>TABB</i> veidnes, kas tika izstrādātas un saglabātas <i>Gdrive www.teststrategy.org</i>	Daudzi projekti, kas ieviesti <i>www.teststrategy.org</i>

Otrajā posmā tika izmantots vairāku situāciju analīzes pētījums, kas veikts ar aktuāliem projektiem, lai analizētu iespējamās izmantošanas gadījumus un intervijas sniedzēju pieredzi saistībā ar vizualizēto *TABB* stratēģiju. Kopējā procesā rezultātu apkopošanai tika izmantota triangulācijas metode.

4.3. Jaunas stratēģijas adaptēšana

Saskaņā ar [22] jaunas stratēģijas veiksmīga adaptēšana ir atkarīga no mērķa, projekta/metodoloģijas, secinājumiem, praktiskajām sekām, oriģinalitātes/vērtības un galvenajām vērtībām. Šajā pētījumā autors pievērš uzmanību visam iepriekš minētajam. Autors iesaka izmantot piedāvātos posmus kā vadlīnijas, lai sāktu jaunu vai palaistu esošos programmatūras izstrādes vai testēšanas projektus. Lietotājs var pieņemt lietošanai gatavas *TABB* veidnes, ko piedāvā autors un kas pieejamas portālā *teststrategy.org* (4.7. att.).

- 1. posms. Vienoties vai piedāvāt savai komandai vizualizēt testa stratēģiju, izmantojot *Lean* veidni uz vienas lapas.
- 2. posms. Izpētīt esošo pārbaudes stratēģijas dokumentu vai testēšanas procesa dokumentu, lai iegūtu pārskatu, ja komandai tāda nav, tad uzdot jautājumus vai sarīkot prāta vētru kopā ar komandas biedriem jautājumu noskaidrošanai.
- 3. posms. Izmantot lietošanai gatavu veidni no vietnēm *teststrategy.org* vai *GitHub resource*.
- 4. posms. Pārdēvēt moduļa un apakšmoduļa metanosaukumu vai metaaprakstu atbilstoši savam projektam un testēšanas veidam.
- 5. posms. Ar grupu koplietot “*Test Activity Building Block Strategy Using Lean Canvas Visualised Test Strategy*” (Testēšanas darbību pamatmoduļu stratēģija, izmantojot *Lean Canvas* modeli balstītu vizualizēto testēšanas stratēģiju), lai to atjauninātu, kad mainās projekta dzīvescikls.

Smoke testing test strategy		Team member names	Version	Date: dd/mm/yyyy	Iteration #
Critical functionalities list Login Logout Password recover Main page load	Smoke testing checklist	Knowledge base links	Browser coverage	Smoke test suite automation results	
	Key issues/defects found <small>key activities you measure</small>				
Test environment <small>www.abod.com</small>		Test document links			

4.7. att. Parauga veidne no vietnes *teststrategy.org*.

Šajā pētījumā autors vēlas aprakstīt jauno ierosināto testēšanas stratēģiju *TABB*, lai turpmāk to izstrādātu detalizētāk. Tādējādi pētījuma posms, kas aprakstīts 4. nodaļā, tiek pabeigts ar tālāk minēto stipro pušu un ierobežojumu analīzi. 4.2. tabulā apkopotas vizualizētās programmatūras testēšanas stratēģijas stiprās pusēs un ierobežojumi.

4.2. tabula

Vizualizētās programmatūras testēšanas stratēģijas stiprās pusēs un ierobežojumi

Stiprās pusēs	Ierobežojumi
Acīmredzama stiprā puse ir vērtības centralitātes atzišana. Vizualizētā testa stratēģijas modeļa konstruēšanas mērķis ir noteikt un nodrošināt vērtību programmatūras testēšanas procesam.	Tika konstatēta liela vienprātība par tādu ārēju faktoru trūkumu, kā, piemēram, programmatūras pārbaudes procesa briedums, normatīva atbilstība īpašiem programmatūras projektiem. Nepieciešamība pēc prasmīgas komandas, komandas un darba attiecībām un detalizētas testēšanas centienu aplēses vizualizētajā testēšanas stratēģijā.
Tiešsaistes lietotāju apsekojuma un respondentu komentāru apstrādes laikā kā stiprās pusēs testēšanas procesā tika minētas veidnes vizuālais attēlojums, lietderība, izmantojamība un vienkāršība.	Tika izpētīti atsevišķi moduļi, tika konstatēti atkārtoti ierobežojumi. Visas intervējamās personas minēja neskaidrības par pamatmoduļiem, kas nav pietiekami, lai pievienotu lielāku informācijas kopumu.
Vizualizētās testēšanas stratēģijas spēks ir tas, ka tā darbojas kā saziņas rīks starp komandām, testēšanas vadītājiem, individuālajiem testētājiem, attālinātiem komandas dalībniekiem un partneriem. Vizualizētā formāta dēļ testa stratēģijas struktūra un vienkāršība veicina labāku komunikāciju starp dažādām disciplīnām.	Pēdējais ierobežojums ir vērsts uz komandu mijiedarbību un vizualizētās testa stratēģijas izveides vērtību. Jāņem vērā, ka spējajā testēšanā vizualizētajā testēšanas stratēģijas izstrādes procesā jāiesaista visi komandas dalībnieki.
Testēšanas stratēģijas vizuālais attēlojums ir labi orientēts uz iekšējiem faktoriem.	Tādu ārēju faktoru neievērošana, kā, piemēram, procesa briedums, prasmīga komanda, komandas un darba attiecības un testa pasākumu aplēse.
Vizualizētas testēšanas stratēģijas dažādu dimensiju pārklājums, piemēram, riski, mērķi/uzdevumi, konkrēti testēšanas līmeņi plāni, testēšanas rādītāji, galvenais veikspējās rādītājs.	Vizualizētā testēšanas stratēģija balstās dažādu abstrakcijas līmeņu pamatmoduļos. Tā rezultātā uzsvars tiek likts tikai uz pamatmoduļiem, piemēram, testa reģistru, galveno funkcionalitāšu sarakstu, gariem teksta aprakstiem.

5. APROBĀCIJA UN *TEST ACTIVITY BUILDING BLOCK* STRATĒGIJA

Šajā pētījumā autors ir cieši sadarbojies ar divām programmatūras izstrādes kompānijām – vidēja un liela izmēra kompāniju. Autors ir atradis platformu, lai ieviestu *TABB*, kas attiecas uz aktīvajiem programmatūras izstrādes projektiem. Autora izvēlētais uzņēmums ir vidēja lieluma uzņēmums, kas nodarbojas ar programmatūras produktu izstrādi gala klientam. Uzņēmumā tika izstrādāti vairāki programmatūras izstrādes projekti, izmantojot dažādus rīkus un dažādas izstrādes metodes. Autors ierosināja ideju stratēģijas izstrādes testēšanai. Šādā veidā autors apkopoja rezultātus un vērtīgas atsauksmes, kas tika dokumentētas apakšnodaļā. Autors ieguva iespēju arī strādāt šajā programmatūras izstrādes uzņēmumā. Tas ir viens no Eiropas vadošajiem mākoņprogrammatūras izstrādes produktu ražošanas uzņēmumiem. Visu produktu izstrādes dzīvesciklu atbilst spējai izstrādei (*Agile*). Pats uzņēmums ir ļoti liels, autors atrada mantotu projektu starp jauniem produktu izstrādes projektiem un ierosināja ideju, kas tika akceptēta un guva labas atsauksmes.

5.1. *Test Activity Building Block (TABB)* novērtēšana

Lean Canvas modeli balstīta *TABB* stratēģija ir novērtēta divos programmatūras izstrādes uzņēmumos vairākos spējās izstrādes projektos. Šajā procesā ir izvērtēti 22 projekti, promocijas darbā dokumentēti sešu projektu rezultāti. Situāciju izpēti analīzes procesā autors atklāja, ka katrs uzņēmums izmanto dažādas tehnoloģijas, rīkus, testēšanas procesu un unikālu biznesa domēna programmatūru.

5.2. Izpētes joma

Izpētes joma aptver testēšanas darbības un pamatmoduļu stratēģijas ieviešanu uzņēmumos, izmantojot *Lean Canvas* modeli. Lai gan projekta darbības joma ietvēra dažādas uzņēmējdarbības jomas, dažādus programmatūras lietojumu veidus un to dzīves ciklu attīstību, pētījumā uzmanība pievērsta dažādiem iespējamiem domēniem, lai aptvertu dažādus programmatūras izstrādes dzīves cikla testēšanas veidus (5.1. tab.).

**Projekta veids, biznesa domēns, izveidotās testēšanas stratēģijas veids
un izstrādes metodoloģijas veids**

Nr.	Projekta veids	Uzņēmējdarbības domēns	Izveidotās testēšanas stratēģijas veids	Izstrādes veids	Dokumentācijas lielums (līdz) lapām (WWWWWH testēšanas stratēģija)	Dokumentācijas izstrādes darbietilpība (līdz) katrai programmatūras versijai stundās
1.	Mikro serviss	Iepirkums	Testēšanas automatizācija	<i>Agile Kanban</i>	14	4
2.	Tīmekļa lietojumprogramma	Izklaide	Veiktspējas testēšana	<i>Agile Scrum</i>	12	4
3.	Tīmekļa lietojumprogramma	Izklaide	Drošības testēšana	<i>Agile Scrum</i>	7	4
4.	Tīmekļa lietojumprogramma	E-komercija	E2E AI testēšana	<i>Agile Scrum</i>	9	4
5.	Tīmekļa lietojumprogramma	Nodokļu deklarācija	Savietojamības testēšana	<i>Waterfall</i>	22	8
6.	Tīmekļa lietojumprogramma	E-komercija	DevOps testēšana	<i>DevOps testēšana</i>	14	4

- Dokumentācijas lielums (līdz) lapām (WWWWWH testēšanas stratēģija) – lappuses izmēri palielinās un samazinās atkarībā no dokumentā ietvertā apraksta.
- Dokumentācijas izstrādes darbietilpība katrai programmatūras versijai stundās – dokumentu apjoms stundās pieaug vai samazinās atkarībā no projekta plānošanas sesijā piešķirtā laika.

5.2.1. Drošības testēšanas gadījumu izpēte

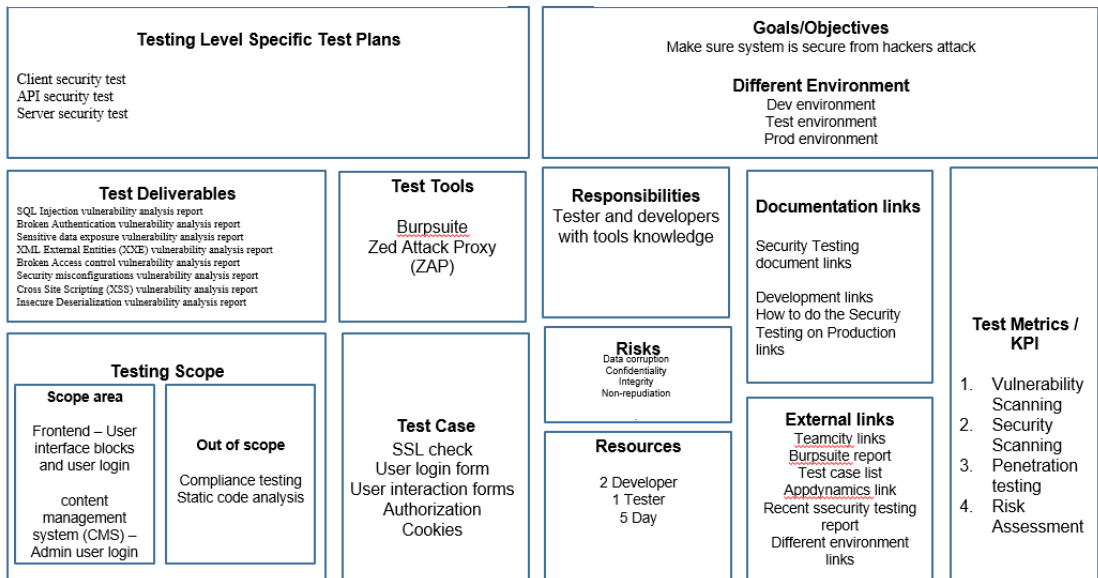
Šajā apakšnodaļā aplūkota *TABB* stratēģijas pieņemšana vienā no projektiem, kur tiek veikta drošības testēšana (5.1. tab.). Šis projekts ir tīmekļa lietojumprogramma, kas atbilst modelim *Agile Scrum*. Šajā projektā vizualizētā testēšanas aktivitāšu pamatmoduļu stratēģija ir izstrādāta tīmekļa lietojumprogrammu drošības testēšanai.

Ņemot vērā iepriekšējo izpēti saistībā ar projektu, kam tika lietota drošības testēšana, grupas dalībniekiem tika uzdoti vairāki jautājumi. Atbildes tika reģistrētas turpmākai testēšanas stratēģijas izstrādei (5.2. tab.).

Drošības testēšanas novērojumi, jautājumi un detalizēta informācija par projektu

Detalizēta informācija par projektu	Projekta nosaukums: <i>EveryHair</i> Komandas lielums: 10 SDLC veids: <i>Agile Lean</i> Testēšanas veids: drošības testēšana Grupas dalībnieki: analītiķis, attīstītājs, QA, arhitekts		
Jautājumi	1. jautājums Kādi testēšanas veidi tiek veikti? Funkciju testēšana, lietotāju akcepttestēšana (UAT)	2. jautājums Ar kādām problēmām tīmekļa lietojumprogrammu drošībai saskaras testēšanas komanda? Iesprauduzbrukums Uzlauzta autentifikācija Sensitīvu datu atklāšana XML ārējās vienības (XXE) Bojāta piekļuves kontrole Drošības kļūdas Starpvietņu skriptošana (XSS) Nedroša deserializēšana	3. jautājums Kas ir atbildīgs par programmatūras kvalitāti? Testētāji un daļēji izstrādes komanda
	4. jautājums Kādu dokumentāciju veido jūsu komanda? Testēšanas dokuments Drošības testēšanas ziņojums	5. jautājums Vai projektam ir pārbaudes stratēģija dokumenta vai vizualizētā formā? Nē	6. jautājums Komanda vēlētos vizualizēt vienas lapas apjoma testa stratēģiju. Jā
Novērojumi	<i>EveryHair</i> ir lietotāja iesaistes risinājums salona lietotājiem Projekta tehnoloģijas: <i>Azure</i> , <i>C#</i> , progresīvā tīmekļa lietojumprogramma, <i>AngularJs</i> , <i>SQL</i> , <i>Git</i> Testēšanas vadība: augsta līmeņa testēšanas dokuments, manuāla testēšana Testēšanas stratēģija: testēšanas dokuments un izlaidis kontrolsaraksts		

Ņemot vērā visus iepriekš minētos datus, esošā dokumenta analīze un *TABB* pieņemšana ļauj izveidot vizualizētu drošības pārbaudes stratēģiju (5.1. att.). Lietotāja vizualizētā testēšanas darbību stratēģija *TABB* ir izveidota tā, kā autors aprakstīja 4.1. apakšnodaļā. Tajā ir trīs posmi, lai izveidotu *Lean Canvas* testēšanas darbību pamatmoduli un vizualizētu *Lean Canvas* testēšanas stratēģijas projektu.



Copyright © 2018

5.1. att. Vizualizēta drošības testēšanas stratēģija.

Vizualizēta testēšanas aktivitāšu pamatmoduļu stratēģija atrisina daudzus jautājumus:

- atbilstoši apakšmoduļu nosaukumi padara tos viegli saprotamus visiem komandas dalībniekiem;
- tiek identificēti dažādi drošības pārbaudes veidi;
- tiek noteikti testēšanas ziņojumi un uzdevumi;
- testēšanas joma ir skaidri noteikta, un ir risināti jautājumi, kas nav saistīti ar darbības jomu;
- projektam ir noteikti drošības pārbaudes rīki pārklājuma pārbaudei;
- pārbaudes gadījums ir atzīts un izveidots atbilstoši prioritātēm;
- ir noteikts drošības pārbaudes mērķis, atšķirīga testēšanas vide un cilvēkresursi;
- tiek kartētas attiecīgās dokumentācijas saites un ārējās saites;
- darbības pamatrādītājus (*KPI*) izmanto, lai uzraudzītu testēšanas gaitu un sekas;
- šajā projektā ietilpst vienkārša dokumentācijas stratēģija;
- arī saistītās iekšējās grupas var izmantot drošības pārbaudes darbību veidošanas pamatmoduļu stratēģiju;
- arī netehniskais darba grupas dalībnieks var izmantot testēšanas darbību pamatbloku stratēģiju, lai sazinātos vai apkopotu rezultātus.

5.3. Gadījumu izpētes rezultātu analīze

Šajā apakšnodaļā sniegti seši ilustratīvi piemēri, lai parādītu, kā pielāgot un piemērot vizualizēto *TABB* stratēģiju. Gadījumi attiecas uz dažādiem programmatūras projektu veidiem, uzņēmējdarbības jomām, tehnoloģijām un programmatūras testēšanas veidiem. Sākotnējā posmā pirms testēšanas darbību pamatmoduļu stratēģijas sākšanas komandai tika uzdoti vairāki jautājumi. Paralēli tika pārskatīti esošie pārbaudes un izstrādes dokumenti, lai apkopotu projekta informāciju. Turklāt šajā apakšnodaļā tiek veidota vizualizētā testa stratēģija, kā arī apkopotas un apspriestas atsauksmes. 5.3. tabulā redzama

empīriskā pētījuma rezultātu analīze, ņemot vērā testēšanas veida, dokumentācijas lieluma [26], saziņas un vizualizācijas salīdzinājumu.

TABB pieņemšana nozīmē uzlabojumu, kas izteikts atbilstoši 5,0 un 0,6 stundu varianci. Samazinātās dokumentācijas lapas vidēji atbilst 53 % un 0,8 procentuālā variāncei.

Atbilstoši 5.3. tabulai:

- dokumentācijas lielums lappusēs pēc *TABB* pieņemšanas – lapušu daudzums palielinās vai samazinās atkarībā no dokumentā rakstītā apraksta. No iepriekš apkopotajiem datiem redzams, ka minimālais lapu skaits ir samazināts, salīdzinot ar *WWWWW* testēšanas stratēģijas lapām;
- dokumentācijas lielums katrai programmatūras versijai stundās – dokumentācijas izstrādes darbietilpība stundās pieaug un samazinās atkarībā no projekta plānošanas sesijā piešķirtā laika. Pamatojoties uz iepriekš apkopotajiem datiem, tika konstatēts, ka, salīdzinot ar *WWWWW* testēšanas stratēģiju, ir samazinātas minimālās dokumentēšanas slodzes stundas.

5.3. tabula

Empīriskā pētījuma rezultātu analīze

Nr.	Projekta veids	Dokumentācijas lielums lapās (<i>WWWWW</i> testēšanas stratēģija)	Dokumentācijas lielums lappusēs pēc <i>TABB</i> pieņemšanas	Samazinātās dokumentācijas lapas procentos	Delta lappuses	Dokumentācijas izstrādes darbietilpība katrai programmatūras versijai stundās	Dokumentācijas lielums lappusēs pēc <i>TABB</i> stundas
1	Mikro serviss [4.4.1]	14	5	64 %	9	4	2
2	Tīmekļa lietojumprogramma [4.4.2]	12	7	42 %	5	4	2
3	Tīmekļa lietojumprogramma [4.4.3]	7	3	57 %	4	4	2
4	Tīmekļa lietojumprogramma [4.4.4]	9	4	56 %	5	4	2
5	Tīmekļa lietojumprogramma [4.4.5]	22	13	41 %	9	8	4
6	Tīmekļa lietojumprogramma [4.4.6]	14	6	57 %	8	4	2

Autora pētījumi par vienas lapas apjoma vizualizētas testēšanas stratēģijas (*TABB*) izveidi nerada plaši pamanāmas sekas tādos testēšanas stratēģijas un testēšanas vadības problēmu gadījumos kā:

- kvalitātes dokumentācija; vizualizācijas testa stratēģijas deva neietekmē dokumenta izkropļojumu, koordināciju, kontroli, piegādi vai atbalstu, kas nepieciešams kvalitātes nodrošināšanai;
- komunikācija; vienas lapas apjoma testēšanas stratēģija vienkāršo starpkomandu saziņu un saziņu ģeogrāfiski sadalītā komandā;
- dokumentēšanas darbs; dokumentācijas izstrādes laiks un pasākumi ir saīsināti.

SECINĀJUMI UN TURPMĀKĀ DARBĪBAS JOMA

Promocijas darba mērķis ir izstrādāt vizualizētu programmatūras testēšanas stratēģiju, izmantojot *Lean Canvas* modeli.

Ceļā uz šī mērķa sasniegšanu ir iegūti šādi **rezultāti**:

- veikta testēšanas procesa zudumu klasifikācija, lai izveidotu labākus metanosaukumus;
- ontoloģijā ir analizētas prāta kartes zināšanu vadības sistēmai; nākamajā posmā metavirsrakstu izveidei izmantota prātojamu shēmu ieguve;
- attiecībā uz sistēmu izstrādi ir apkopoti izmantošanas gadījumi un izstrādāta tiešsaistes tīmekļa lietojumprogramma, ņemot vērā tiešsaistes tīmekļa ierobežojumus;
- lai iegūtu labāku ieskatu, analizēti gan primārie, gan sekundārie pētījumu dati par *TABB*;
- ieskicēti pasākumi, kas jāveic, lai izveidotu *TABB* sistēmu;
- dokumentētas esošās testēšanas stratēģijas *WWWWWH* un *STD (Story, Development, Testing)*, ko pieņēmusi nozare;
- vispārējie apstiprinājuma rezultāti ļauj noteikt, ka *TABB* vizualizētā testēšanas stratēģija ir veiksmīgi pieņemta un izpētīta funkcionālos un nefunkcionālos projektos;
- autora piedāvātā *TABB* testēšanas stratēģija ir salīdzināta ar citām testēšanas stratēģijām;
- veikta dažādu gadījumu pētījumu analīze testu automatizācijai, veiktspējas testēšanai, drošības testēšanai, *E2E* testēšanai, savietojamas programmatūras izstrādei un *DevOps* projektiem, rezultāti ir novērtēti un dokumentēti;
- vispusīgi novērtēta izstrādātā metodoloģija un metodes, apvienojot eksperimentus, gadījumu pētījumus un ekspertu intervijas.

Secinājumi

- *TABB* pieejas pieņemšana samazināja dokumentācijas lapas vidēji par 53 % un laiku, lai izstrādātu dokumentāciju, aptuveni par 50 %.
- Galvenie drošības pārbaudes elementi ir mērķi/uzdevumi, testēšanas vide, riski, testēšanas joma, testēšanas līmeņa testēšanas plāni, ārējās saites un resursi.
- *TABB* pielāgojams testa automatizācijas, veiktspējas testēšanas, drošības testēšanas, *E2E AI* testēšanas, savietojamības testēšanas un *DevOps* testēšanas jomā. Tas ir pielāgojams gan funkcionālajā, gan nefunkcionālajā testēšanā.
- *TABB* vizualizētā testēšanas stratēģija ir pielāgojama *Agile (Kanban, Scrum)*, *Waterfall*, *DevOps* un citu metodoloģiju projektos.
- *TABB* adaptācija ir analizēta un novērtēta, tā nerada sekas, kas saistītas ar plaši pamanītām testēšanas stratēģijas un testa vadības problēmām, piemēram, komunikāciju, vizualizāciju, iekšējo komandu konfliktiem, programmatūras defektiem, dokumentēšanas pasākumiem.

Promocijas darba rezultāti sniedz iespēju veikt turpmākus pētījumus vairākās jomās.

- *TABB* lietojamības paplašināšana, ieskaitot dažādu programmatūras izstrādes projektu novērtēšanu, piemēram, iegulto programmatūru, programmaparatūru utt.
- Rīku vai spraudņu izstrāde, lai integrētu *TABB* testa ziņošanas rīkus.
- Vienkāršāka tiešsaistes risinājuma izstrāde *teststrategy.org* un vizualizētas testēšanas stratēģijas izveide, izmantojot mašīnmācību un mākslīgo intelektu. Šajā procesā lietotājs iesniedz nepieciešamo projekta informāciju un iegūst vairākus lietošanai gatavus vizualizētus testēšanas stratēģijas modeļus.
- Radīt izpratni par “vizualizēto testēšanas stratēģiju” programmatūras testēšanas kopienā, lai palielinātu tās pielāgošanu. Pastāvīga atsauksmju apkopošana no kopienas un šo atsauksmju piemērošana labākam un jaunam modelim.

LITERATŪRAS SARAKSTS

1. Black, R.: Agile Testing Foundations: An ISTQB Foundation Level Agile Tester guide. 2017, pp. 12.
2. Spillner, A., Linz, T., Schaefer, H.: Software Testing Foundations. 2014, pp. 23.
3. Graham, D., Fewster, M.: Experiences of Test Automation. 2012, pp. 32.
4. Watters, A., Pinkster, I., Janssen, D.: Integrated Test Design and Automation. 200, pp. 34–76.
5. Black, R., Veenendaal, V., Graham, D.: Foundations of Software Testing: ISTQB Certification (3e). 2012, pp. 49.
6. Test Process Improvement. – Internet / <https://www.qualitestgroup.com/assets/test-process-improvement.pdf>.
7. Black, R.: Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional, 2016.
8. Robinson, J., Nanda, V.: Six Sigma Software Quality Improvement. 2011.
9. Fleurey, F., Haugen, Q., Johansen M. F.: A Survey of Empirics of Strategies for Software Product Line Testing. IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, 2011.
10. Maurya, A.: Running Lean: Iterate from Plan A to a Plan That Works (Lean Series), 2012.
11. IPC Group.: Eat. Sleep. Create. Entrepreneurs notebook Lean Canvas Business Ideas Journal, 2018.
12. Lin W.: Lean Startup Canvas Notebook, 2018.
13. Mirabelli V.: The 1 Page Project: A Practical Guide to Using the Lean Project Canvas to Focus Your Projects on What Matters, 2019.
14. Marsh J.: UX for Beginners: A Crash Course in 100 Short Lessons, 2016.
15. Grant W.: 101 UX Principles: A definitive design guide, 2018.
16. Mittal N. M.: Software Design Patterns 2.0: UI and UX Design Principles and Tips
17. Nidagundi, P., Novickis, L.: Possibilities about the design lean canvas model and its adaptation in the agile testing. Symposium for Young Scientists in Technology, Engineering and Mathematics, Volume 1853, SYSTEM 2017, Kaunas, Lithuania, 28 April, pp. 20–23.
18. Nidagundi, P., Novickis, L.: Introduction to Lean Canvas Transformation Models and Metrics in Software Testing, APPLIED COMPUTER SYSTEMS, Year 2016, pp. 30–36.
19. Nidagundi, P., Novickis, L.: New method for mobile application testing using lean canvas to improving the test strategy, XII-th International Scientific and Technical Conference “Computer Science and Information Technologies”, Sept 2017.
20. World Quality Report – Sogeti. – Internet / <https://www.sogeti.com/explore/reports/world-quality-report-2017-2018/>.
21. WQR – Sogeti. – Internet / https://www.sogeti.com/globalassets/global/wqr-201819/wqr-2018-19_secured.pdf.
22. Graner, M., Mißler-Behr., M: 22. Key determinants of the successful adoption of new product development methods, European Journal of Innovation Management 16 (3), Year 2013.
23. Visualizing the Complex Software Development Process. – Internet / <https://www.cmcrossroads.com/article/visualizing-complex-software-development-process>.
24. Visualize Strategy. – Internet / http://udleditions.cast.org/strategy_visualize.html.
25. Tan, K., Platts, K.: Strategy visualisation: knowing, understanding, and formulating.
26. Penella S.: Overcoming Challenges to Good Test Documentation – Internet / <https://www.stickyminds.com/article/overcoming-challenges-good-test-documentation>.
27. Nidagundi, P., Novickis, L.: Introduction to Lean Canvas Transformation Models and Metrics in Software Testing, APPLIED COMPUTER SYSTEMS, Year 2016, pp. 30–36.
28. Proactive Vs Reactive: Which Approach is Better to Attain Quality. – Internet / <https://www.qualityze.com/proactive-vs-reactive-approach-better-attain-quality/>.
29. Rubin K. S.: Essential. Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature): A Practical Guide To The Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)), 2012.

30. Chen, N.: std 829-2008 and Agile Process – Can They Work Together?, Year 2013.
31. C. Unger-Windeler, J. Klünder and K. Schneider.: “A Mapping Study on Product Owners in Industry: Identifying Future Research Directions,” 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP), 2019, pp. 135–144.



Padmarajs Nidagundi (*Padmaraj Nidagundi*) dzimis 1985. gadā Vidžajapurā, Karnatakas štatā, Indijā. Visvesvarajā Tehnoloģiskajā universitātē ieguvis bakalaura grādu informācijas zinātnes un inženierzinātnes nozarē (2010) un maģistra grādu datortehnikā (2014). No 2010. gada strādā dažādās programmatūras izstrādes kompānijās. Patlaban ir Rīgas Tehniskās universitātes pētnieks. Viņa veiksmīgākā darbības joma iepriekšējos gados ir bijusi programmatūras izstrāde un testēšana, kurā P. Nidagundi guvis nozīmīgu starptautisku darba pieredzi. Pēdējos gados bijis mācībspēks un piedalījies pētniecības projektos. Pētniecības intereses – programmatūras izstrāde, kvantu skaitļošana, kibernetiķība un mākslīgais intelekts.