

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**PROGRAMMATŪRAS TESTĒŠANAS PROCESU
KVALITĀTES NOVĒRTĒŠANA**

BAKALaura DARBS

Autors: **Laila Gutke**

Studenta apliecības Nr. lg09031

Vadītāja: M.dat. Dace Zvīdriņa

RĪGA 2013

ANOTĀCIJA

Bakalaura darbā „Programmatūras testēšanas procesu kvalitātes novērtēšana” tika veikta testēšanas metožu, testēšanas procesā sagatavojamās dokumentācijas, sistēmu izstrādes un ieviešanas konkursu nolikumu un testēšanas procesu aprakstošu starptautisku un Latvijas standartu analīze un testētāju viedokļa par testēšanas procesā izvirzītajām prasībām izpēte. Bakalaura darba ietvaros autore iepazīnās ar *LEAN* metodoloģiju (turpmāk *LEAN*) darba procesa optimizācijai.

Darba gaitā tika izstrādāta testēšanas procesa plūsmas sagatave un minimālais un ieteicamais dokumentu, kā arī izvērtējamo un pielietojamo testēšanas metožu kopums, balstoties uz *LEAN* pamatprincipiem, lai nodrošinātu maksimāli kvalitatīvu un apjomīgu sistēmas testēšanu un kvalitātes novērtēšanu.

Atslēgvārdi: Testēšana, *LEAN* metodoloģija, *LVS 70:1996*, *LVS 73:1996*, *ANSI/IEEE Std. 829-1998*, *ANSI/IEEE Std. 829-2008*, *ANSI/IEEE Std. 1008-1987*

ABSTRACT

In the report “Software testing process quality assesment” testing methods, testing documentation, tender rules of new system development and implementation, the testing process descriptive international and Latvian standards are analyzed. A research to determine the attitude of testers about requirements for the testing process from clients and internal needs of projects was done. During the creation of this report the author explored *LEAN* methodology that is used for process optimization.

As a result of the report a testing process organization guide was prepared - a minimum set of requested documentation to be prepared, recommended testing methods to do. The creation was based on the fundamental principles of *LEAN* methodology, to ensure the highest quality of the testing process with minimum unnecessary steps.

Keywords: Testing, *LEAN* methodology, *LVS 70:1996*, *LVS 73:1996*, *ANSI/IEEE Std. 829-1998*, *ANSI/IEEE Std. 829-2008*, *ANSI/IEEE Std. 1008-1987*

SATURS

APZĪMĒJUMU UN TERMINU SARAKSTS	1
IEVADS	3
1. LITERATŪRAS APSKATS	6
1.1. Testēšanas nepieciešamība	6
1.2. Testēšanas veidi.....	7
1.2.1. Strukturālā testēšana	7
1.2.2. Funkcionālā testēšana.....	7
1.2.3. Vienībtestēšana	8
1.2.4. Integrācijas testēšana.....	8
1.2.5. Apstrādājamā apjoma testēšana	8
1.2.6. Stresa testēšana	8
1.2.7. Lietojamības testēšana	9
1.2.8. Drošības testēšana	9
1.2.9. Veiktspējas testēšana.....	9
1.2.10. Atmiņas testēšana	9
1.2.11. Uztādījumu testēšana	9
1.2.12. Savietojamības, pārveidošanas testēšana.....	10
1.2.13. Atkopšanās testēšana.....	10
1.2.14. Akcepttestēšana	10
1.2.15. Instalācijas testēšana	10
1.2.16. Regresā testēšana	11
1.3. Testēšanas procesa kvalitātes kritēriji.....	11
1.4. Testējamo sistēmu kvalitātes metrikas.....	12
1.4.1. Koda kvalitāte.....	13
1.4.2. Sistēmas kvalitāte	13
1.4.3. Testēšanas dotais uzlabojums	14
1.4.4. Testēšanas efektivitāte	14
1.5. Testēšana standartos	14
1.5.1. LVS 70:1996	15
1.5.2. LVS 73:1996	19
1.5.3. ANSI/IEEE Std. 829-1998	19

1.5.4.	ANSI/IEEE Std. 829-2008	21
1.5.5.	ANSI/IEEE Std. 1008-1987	22
1.6.	ISTQB sertificēts testētājs	23
1.7.	LEAN metodoloģija	24
1.7.1.	Vērtības identificēšana	27
1.7.2.	Vērtību plūsmas sagatave	27
1.7.3.	Vērtību plūsmas izstrāde	27
1.7.4.	Vērtību plūsmas uzlabošana	28
1.7.5.	Vērtību plūsmas idealizācija	28
2.	UZDEVUMA IZPĒTE	30
2.1.	Veicamie darbi mērķu sasniegšanai	30
2.2.	Aptauja „Programmatūras testēšanas procesu kvalitātes novērtēšana”	31
2.2.1.	Aptaujas atbilžu par prasībām pret testētāju analīze	32
2.2.2.	Aptaujas atbilžu par LEAN metodoloģiju analīze	39
2.2.3.	Aptaujas rezultātu kopsavilkums	43
2.3.	Atvērtu konkursu nolikumu analīze	44
2.4.	Testēšanas process, izmantojot LEAN metodoloģiju	46
2.4.1.	Kļūdas simptomi	49
2.4.2.	Kļūdu procesa jautājumi	49
3.	RISINĀJUMS	52
3.1.	Minimālais dokumentācijas kopums	53
3.2.	Kritiskāko pielietojamo testēšanas metožu kopums	55
	REZULTĀTI UN DISKUSIJA	57
	SECINĀJUMI	60
	IZMANTOTĀ LITERATŪRA UN AVOTI	62
	PIELIKUMI	64
1.	Pielikums. Aptauja „Programmatūras testēšanas procesu kvalitātes novērtēšana”	64
2.	Pielikums. Aptaujas „Programmatūras testēšanas procesu kvalitātes novērtēšana” starprezultāti	69

APZĪMĒJUMU UN TERMINU SARAKSTS

Apzīmējumi

IT – Informāciju tehnoloģija jeb zināšanu, metožu, paņēmieni un tehniskā aprīkojuma kopums, kas ar datoru un sakaru līdzekļu starpniecību nodrošina jebkuras informācijas iegūšanu, glabāšanu un izplatīšanu.

ISO – Starptautiskā standartizācijas organizācija jeb Starptautiska organizācija, kas dibināta 1946.g. un kas nodarbojas ar sakaru un informācijas apmaiņas starptautisku standartu izstrādāšanu. Starptautiskā standartizācijas organizācija kopā ar Starptautisko elektrotehnikas komisiju (IEC) ir izveidojusi Apvienoto tehnisko komiteju (JTC-1), kuras uzdevums ir izstrādāt standartus informācijas tehnoloģijas jomā.

ISTQB – (*International Software Testing Qualifications Board* jeb Starptautiskā Programmatūras testēšanas kvalifikācijas padome, kas ir kļuvusi par pasaules mēroga līderi sertificētāju programmatūras testētājiem.

LEAN – (īpaš.v. tulkojumā *liess*, kas nozīmē satur maz taukvielu vai citu lieku vielu) metodoloģija, kas nosaka, ka rezultātu iegūšanas procesā nepieciešams darbības tikai ar pievienotu vērtību bez liekiem, nevajadzīgiem soļiem un pienākumiem.

PPS – programmatūras prasību specifikācija ir specifikācija atsevišķam programmatūras produktam, programmai vai programmu kopai, kas izpilda noteiktas prasības.

SCRUM – iteratīva spējās programmatūras izstrādes sistēma programmatūras izstrādes projektu pārvaldībai un produktu vai lietojumprogrammu izstrādei.

TFS – (*Visual Studio Team Foundation Server*) – sadarbības platforma, kas nodrošina Microsoft lietotņu dzīves cikla pārvaldību.

Termini

Agile - Informācijas sistēmu ātra izstrāde un ieviešana, ar pakāpeniskiem tuvinājumiem uzlabojot šīs sistēmas un iekļaujot tajās jaunākās lietotāju prasības.

„*Dūmu*” *testi* – pagaidu sistēmas pārbaudes, lai atklātu vienkāršas, bet pietiekoši smagas kļūmes tekošajā programmatūras piegādes pakotnē.

Hot fix jeb labojumfailu pakotne ir viena kumulatīva pakotne, kas sastāv no viena vai vairākiem failiem, kurus izmanto problēmas novēršanai sistēmā. Labojumfaila pakotne risina konkrētu klienta problēmsituāciju un to nedrīkst izplatīt ārpus šī klienta organizācijas.

Klients - Pakalpojumu vai preču pircējs, pasūtītājs, kas izmanto kādas iestādes, uzņēmuma vai organizācijas piedāvājumu.

Ūdenskrituma modelis – sistēmas izstrādes metodoloģija ar secīgām aktivitātēm neatkarībā no fāžu skaita un specifiskas sistēmas, kur katras aktivitātes rezultāts tiek izmantots par ieejas informāciju nākamajai aktivitātei.

„*V*” *modelis* – sistēmas izstrādes metodoloģija, kurā sistēmas izstrādes aktivitātes un fāzes ir savstarpēji atbilstošas: kreisais zars parāda progresu, sākot ar analīzi un projektēšanu, līdz kodēšanai un pieaugošam sistēmas sadalījumam komponentēs, bet labais V zars attēlo progresējošu sastāvdaļu salikšanu un testēšanu, beidzoties ar programmatūras produkta piegādi.

IEVADS

Mūsdienās programmatūras pakalpojumu klāsts un sarežģītība strauji pieaug, nodrošinot to nozīmīgumu biznesa procesū, klientu vajadzību realizācijai un sakārtošanai. Tas paver lietotājam daudz dažādas iespējas, piemēram, veikt personāla pārvaldību, norēķinu pārvaldību, dokumentācijas pārvaldību u.c. biznesa procesus elektroniski, efektīvi, ātri un lēti ilgtermiņā. Lai visas šīs darbības klients varētu veikt, nepieciešams izstrādāt konkurētspējīgu, kvalitatīvu, standartiem un likumdošanai atbilstošu, biznesa procesū un klienta vajadzību apmierinošu sistēmu.

Programmatūras izstrāde ir vairāku procesu (analīze, projektēšana, izstrāde un testēšana) mijiedarbība, kuri tiek veikti ar atšķirīgu intensitāti dažādos laika posmos. Neatkarīgi no projekta, testēšana ir tikpat svarīga un neatņemama sastāvdaļa kā pārējie procesi. Tieši testēšana ir process, kas pārbauda, vai iepriekš paveiktais ir nodrošinājis kvalitatīvas programmas izveidi. Testēšanas process nav programmatūras izstrādes noslēdzošā aktivitāte - tas ļoti bieži ir pagrieziena punkts, lai sāktos jauns izstrādes posms jeb iterācija - tiktu veiktas izmaiņas kodā, projektējumā vai prasībās.

Programmatūras izstrādē viena no kvalitātes problēmām saistās ar testēšanas spējām klientam sniegt pārlicību par to, ka izstrādātā sistēma ir pārbaudīta, pārklājot gan tās funkcionalitāti, biznesa procesus, gan datu drošību un korektumu, gan sistēmas veiktspēju un dzīves ciklu, tās intuitīvu lietošanu, gan programmatūras īpašības, kuras ir vēlēties klients.

Mūsdienās lielākā daļa izstrādātās programmatūras tiek pārbaudīta pēc dažādiem biznesa procesu scenārijiem, automatizētiem testiem, lieliem datu apjomiem u.c. metodēm, kuras tiek izvēlētas atbilstoši izstrādātās programmatūras īpašībām.

Testēšanas process tiek aprakstīts gan starptautiskos standartos, gan metodoloģiskos informācijas avotos, kurus izmanto testēšanas speciālistu izglītošanai un sertificēšanai. Mūsdienās programmatūras izstrādes un ieviešanas konkursos un piedāvājumos testēšanas darbs tiek vērtēts, balstoties uz šiem standartiem un metodoloģiskajiem avotiem, pielāgojot tos sistēmas īpašībām un klienta vajadzībām. Jaunajos projektos sistēmas izstrādātājiem testēšanas process ietver sevī gan gandrīz visas testēšanas dokumentācijas izstrādi, gan testēšanas veidu un metožu izmantošanu, neskatoties, cik tiešām šī dokumentācija un visas metodes ir vai nav lietderīgas un nepieciešamas projektam.

Ņemot vērā lielo standartu un metodoloģisko informācijas avotu apjomu un Latvijā esošo testētāju pieredzi darba autore ir nolēmusi veikt izpēti un izstrādāt palīgīdzekli projektu

vadītājiem, testēšanas nodaļu vadītājiem un pašiem testētājiem, kas palīdzēs noteikt, izmērīt, cik kvalitatīvi, apjomīgi, efektīvi ir veikta izstrādātās sistēmas testēšana pēc veiktajiem darbiem, izpildītajiem procesiem, testēšanas rezultātiem, salīdzinot testēšanas procesa kvalitātes kritērijiem un testējamo sistēmu kvalitātes metrikām. Palīglīdzeklis ir paredzēts kā materiāls „kur ieskatīties” testētājam ar obligāti izpildāmajām un ieteicamajām testēšanas metodēm projekta ietvaros, balstoties uz klienta prasībām, kā arī minimāli izstrādājamās dokumentācijas aprakstu atbilstoši Latvijas standartiem, iekļaujot tikai klientam un projektam nepieciešamās sadaļas. Pēc šī palīglīdzekļa testētājam ir ieteicams izveidot izstrādājamās dokumentācijas sagataves, lai optimizētu tieši dokumentu sagatavošanu.

Palīglīdzeklis ir paredzēts arī projekta vadītājam un klientam kā veikto darbu trasējamības materiāls, lai novērtētu, vai testētājs ir veicis visus minimāli nepieciešamos pasākumus sistēmas testēšanas procesā gan klienta vajadzībām, gan iekšējai projekta kārtības ievērošanai.

Palīglīdzeklis nedefinē, cik kļūdas testētājam ir jāatrod un cik kļūdas sistēmā vēl saglabāsies pēc testēšanas procesa. Palīglīdzeklis nosaka, ja testētājs ir veicis visus nepieciešamos pasākumus biznesa procesu, vienumu un funkciju testēšanai un atbilstoši ir dokumentējis problēmu ziņojumus un ievērojis testēšanas plānu, sistēmā ir notestēta maksimāli klientam nepieciešamā funkcionalitāte.

Lai efektīvāk izmantotu Latvijā esošo testēšanas speciālistu cilvēkresursus un veiktu testēšanas procesa padziļinātu izpēti un apkopošanu, tika izvirzīts darba mērķis:

-
- Izstrādāt testētāja palīglīdzekļa sagatavi testēšanas darbu organizācijai un testēšanas procesa kvalitātes un efektivitātes novērtēšanai.

Lai sasniegtu darbā izvirzīto mērķi, darbā tika izvirzīti šādi uzdevumi:

- Iepazīties ar Latvijā esošiem standartiem, kas apraksta testēšanas procesu;
- Iepazīties ar starptautiskiem standartiem, kas apraksta testēšanas procesu;
- Iepazīties ar metodoloģiskās informācijas avotiem, kas paredzēti testētāju sertificēšanai un zināšanu papildināšanai;
- Veikt laika posmā 2011. – 2013.gadam internetā brīvi pieejamu Latvijā izsludinātu konkursu nolikumos prasītās dokumentācijas un testēšanas metožu izpēti;
- Veikt esošo standartu analīzi, salīdzinot aprakstītos testēšanas procesus ar Latvijā strādājošu testētāju pieredzi.

Veikt Latvijā esošo testēšanas speciālistu aptauju par testēšanas procesiem to ikdienas pienākumos un projektos

Izvēlēto uzdevumu realizācijai darbā tika izmantotas šādas metodes:

- iepazīšanās un analīze Latvijā esošiem standartiem;

- iepazīšanās un analīze starptautiskiem standartiem;
- iepazīšanās un analīze aptauju rezultātiem;
- iepazīšanās un analīze Latvijā laika posmā 2011. – 2013. izsludināto konkursu nolikumiem.

Bakalaura darbs ir sadalīts trīs daļās. Literatūras apskatā ir dots ieskats testēšanas procesā pielietojamās metodēs un izstrādājamajā dokumentācijā atbilstoši Latvijas un starptautiskajos standartos sniegtai informācijai. Literatūras apskats ietver izklāstu par *ISTQB* sertifikāta apmācību materiālos aprakstīto testēšanas procesu un *LEAN* metodoloģiju. Uzdevuma izpētes daļā ir analizēta aptaujāto testētāju reālā pieredze un attieksme pret testēšanas procesā izpildāmajām testēšanas metodēm un izstrādājamo dokumentāciju, kā arī ir apkopots testētāju viedoklis par *LEAN* ieviešanu testēšanas procesā. Uzdevuma izpēte ietver analīzi par iepriekšējos gados izsludinātajos sistēmu ieviešanas un izstrādes konkursos ietvertajām vajadzībām un vēlmēm attiecībā pret testēšanas procesu. Darba rezultātā ir sagatavota testēšanas procesa plūsmas sagatave un izstrādāts minimālais dokumentācijas un testēšanas metošu kopums, kuru ieteicams izmantot testētājam visu sistēmu testēšanas procesos.

1. LITERATŪRAS APSKATS

Šajā nodaļā ir attēlots testēšanas process atbilstoši Latvijas un starptautiskiem standartiem, ISTQL sertifikāta apmācību materiāliem un *LEAN* metodoloģijai.

1.1. Testēšanas nepieciešamība

Mūsdienās biznesa procesu izpildē un dažādu funkciju veikšanā tiek izmantota programmatūra, kas ar gadiem paliek sarežģītāka dēļ gala lietotāju vēlmēm un vajadzībām, biznesa procesu maiņām, likumdošanas u.c. apstākļiem. Laikam ritot, programmatūras izstrādes procesā arvien lielāka loma tiek piešķirta arī testēšanai, jo klients un gala lietotājs vēlas saņemt kvalitatīvu, ātru, viegli lietojamu, intuitīvu un biznesa procesu aptverošu programmatūru, kas neapgrūtinās uzņēmuma darbību un neradīs papildus izmaksas. Lai šādu programmatūru nodrošinātu, testēšanai vajadzētu piešķirt svarīgu sistēmas dzīves cikla lomu, jo tā parasti tiek saistīta ar programmatūras kvalitātes nodrošināšanu. Tā kā testēšanas rezultātus precīzi neviens nevar izmērīt, arī jēdziens – kvalitatīvs produkts, ir jāuzskata par vismaz daļēji subjektīvu rādītāju, jo tas parasti tiek saistīts ar nemērāmiem lielumiem, piemēram, lietotāju apmierinātību. Šāda informācija parasti tiek iegūta no gala lietotājiem, uzņēmuma apmierinātības aptaujas formā. Sistēmas kvalitātes kritēriji parasti tiek iedalīti šādi [16]:

- apmierinātības līmenis, kas raksturo to, cik lielā mērā produkts atbilst lietotāja prasībām un vēlmēm;
- produkta vērtība attiecībā uz dažādajām ieinteresētajām pusēm un tā sniegtajām iespējām;
- rādītājs, kas raksturo cik lielā mērā produktam piemīt vēlamās īpašības;
- pakāpe, kas raksturo cik lielā mērā produkts strādā vidē, kurai tas bija paredzēts, nenovirzoties no sagaidītās uzvedības;
- programmatūras izstrādē iesaistīto procesu efektivitāte un to korektums.

Lai šādu programmatūru nodrošinātu, pasaulē ir izveidotas vairākas testēšanas metodes un veidi, dokumentācijas standarti un sertifikācijas, kas pasaka, ka testēšana nesastāv tikai no testu izpildes. Būtisku testēšanas daļu sastāda arī plānošana, kontrole un sagatavošanās testu izpildei. Daudzos uzņēmumos tiek pieļautas divas lielas galējības:

- uzskats, ka testu izpilde ir lielākā vai vienīgā testēšanas procesa aktivitāte, novedot projektu pie lielākām izmaksām testēšanas novērtējuma apjoma vai labojamo kļūdu

apjoma dēļ, kas pieteiktas no gala lietotāja, jo netiek pietiekami veikta testu plānošana un sagatavošanās.

- no testētājiem tiek prasīts sīki un detalizēti plānot, gatavoties testēšanai un rakstīt testpiemērus un citu dokumentāciju tā, ka testētājam neatliek pietiekami daudz laika reālai testu izpildei, pieļaujot lielākas projekta izmaksas un termiņu nobīdi gala lietotāja pieteikto kļūdu labošanai.

1.2. Testēšanas veidi

Programmas ir dinamiskas. Ja tika atklāta kāda kļūda un tā tika novērsta, tad vairs nav pārliecības, ka programma darbosies korekti uz testa piemēriem, kas tika izpildīti pirms šīs kļūdas izlabošanas. Tas nozīmē, ka testēšana ir jāveic atkārtoti pilnā apmērā. Lai izvairītos no šādiem gadījumiem, pasaulē ir izveidotas un pieņemtas vairākas testēšanas metodes un veidi, kas ir piemēroti dažādiem sistēmas izstrādes un uzturēšanas posmiem [19, 20].

1.2.1. Strukturālā testēšana

Strukturālā jeb sistēmas spēju testēšana ir testēšana, kura, atšķirībā no funkcionālās testēšanas, testē arī sistēmas vai tās komponentu iekšējo struktūru [13]. Šī ir baltās kastes testēšana, kura tiek lietota visos gadījumos, kad sistēma tiek attēlota algoritmiskā, projektēšanas vai prasību valodā. Strukturālā testēšana paredz atrast tādu testu kopu, kas veido pietiekamu pārklājumu sistēmas funkcionalitātei. Tā pievēršas programmas ceļu izvēlei, atbilstošo testa piemēru kopas veidošanai. Tiek salīdzināta programmas uzvedība attiecībā pret kodā paredzēto un apskata sistēmas struktūras loģiku, iekļaujot apgabalos, ko netestētu, ja veiktu tikai funkcionālo testēšanu. Strukturālā testēšana nenodrošina novērtējumu, ka sistēma atbilst prasībām, un testi neimitē reālās dzīves situācijas. Strukturālā testēšana nevar notestēt to, kas nav uzprogrammēts, jo tā ir piemērotāka teorētiskajiem pētījumiem un formalizēšanai. Baltās kastes testēšanu lieto programmētāji vai testētāji, kuriem ir pieejams programmas teksts. [19, 20]

1.2.2. Funkcionālā testēšana

Funkcionālā jeb funkciju testēšana ir testēšana, kas analizē tikai sistēmas atbildes reakciju uz noteiktām ieejas iedarbībām, bet ignorē tās iekšējo struktūru [13]. Šī ir melnās kastes testēšana, kas pārliecinās, ka sistēma izpilda izvirzītās prasības. Melnās kastes testēšanas procesā netiek analizēts, kā notiek process, bet gan, kāds rezultāts tiek izdots. Funkcionālajā testēšanā sistēma tiek uztverta kā „melna kaste”, kur iekšējās darbības precīzi nav zināmas. Šajā gadījumā sistēmai tie iedoti ievaddati un „ja rezultāts jeb izvaddati tiek iegūti kā gaidīts, tests ir izpildīts. Tā salīdzina programmas uzvedību pret specifikāciju. Funkcionālā testēšana simulē reālu sistēmas

lietošanu un neveido nekādus pieņēmumus par sistēmas struktūru. Funkcionālā testēšana pieļauj iespēju palaist garām loģikas kļūdas programmatūrā un pieļauj liekas testēšanas iespējamību. [19, 20]

1.2.3. Vienībtestēšana

Vienībtestēšana jeb sistēmas moduļu testēšana ir process, kas ietver testēšanas plānošanu, testu kopas iegūšanu un testējamās programmatūras vienības jeb moduļa mērīšanu, salīdzinot testējamo vienību ar tai noteiktajām prasībām, kas specificētas PPS. Sistēma netiek pārbaudīta kā viens kopums, bet gan uzmanība tiek vērsta uz sistēmas moduļiem. Vienībtestēšanu parasti veic programmētājs, kas izstrādājis konkrēto sistēmas moduli. Vienībtestēšanas laikā konstatētās kļūdas ir novēršamas daudz vieglāk un efektīvāk, salīdzinājumā ar brīdi, kad programmatūra izstrādāta pilnībā. [8]

1.2.4. Integrācijas testēšana

Integrācijas testos galvenais uzsvars tiek likts uz datu plūsmām un sadarbību starp sistēmas moduļiem vai citām sistēmām, aparatūru, testējot sistēmu kā vienu kopumu. Liela nozīme integrācijas testu veiksmīgai izpildei ir pareiza sistēmas izstrādes plānošana, ja sistēmas izstrādes dzīves ciklā tās moduļi, sasaiste ar citām sistēmām un aparatūru tiek piegādāta pakāpeniski. Šos testus parasti veic programmētāji. Integrācijas testos noteikti ir jāpārlicinās, ka pievienotās izmaiņas neievieš jaunas kļūdas esošajā sistēmā. [19]

1.2.5. Apstrādājamā apjoma testēšana

Slodzes testēšana ir testēšana, lai pārbaudītu komponentes vai sistēmas darbību, ja tiek palielināta slodze, piemēram, paralēlu lietotāju skaits, lai noskaidrotu, kā sistēma uz to reaģē. Šī testēšanas metode palīdz noteikt maksimālo operāciju apjomu, kā arī visas vājās vietas, un noteikt, kurš sistēmas vienums izraisa sistēmas pārslodzi, lēndarbību vai „pakāršanos”. Apstrādājamais apjoms parasti ir tik liels, ka sistēmas kļūda parasti tieši ir sagaidāmais rezultāts. Bieži vien apstrādājamā apjoma testēšana ir līdzīga stresa testēšanai. [19]

1.2.6. Stresa testēšana

Testēšanas veids, kuru izmanto, lai pārslogotu sistēmu un pārbaudītu tās izturēšanos pārslodzes gadījumos. Stresa testēšanu izmanto bufera pārpildīšanās kļūdām, lieliem datu apjomiem, garām transakciju ķēdēm, lieliem skaitam ierīču savienojumiem, nepietiekamas atmiņas gadījumos, vīrusiem u.c. Stresa testēšana parāda drošības riskus un trūkumus, kas parādās attiecīgajās situācijās. [19]

1.2.7. Lietojamības testēšana

Lietojamība ir sistēmas īpašība, kas raksturo, cik viegli lietotājs var apgūt tās izmantošanu, sagatavot tai ieejas datus un interpretēt tās izejas datus. Lietojamības testēšana ir testēšana, lai pārbaudītu, cik programmatūras produkts ir saprotams, viegli apgūstams un vizuāli pievilcīgs lietotājiem pēc noteiktiem nosacījumiem. To var uzskatīt par neaizstājamu praksi sistēmas lietošanā, jo pēc tās var noteikt, cik reāli lietotāji izmantos sistēmu. [19]

1.2.8. Drošības testēšana

Drošības testēšana ir testēšanas process, kuras laikā pārlicinās, vai informācijas sistēma aizsargā datus un uztur funkcionalitāti kā paredzēts. Sešas pamata drošības koncepcijas, kas ir jānosedz, izmantojot drošības testēšanu ir konfidencialitāte, integritāte, autentificēšanās, pieejamība, autorizācija un nenoliegšana. Drošības kļūda ir kļūda, kas palielina nesankcionētas piekļuves risku sistēmai, kas atļauj skatīt un mainīt failus, kam nav tiesību to darīt. [19]

1.2.9. Veiktspējas testēšana

Veiktspējas testēšana ir testēšana, kas novērtē sistēmas atbilstību specifikācijā noteiktajām veiktspējas prasībām. Tā var arī kalpot, lai izpētītu, noteiktu, apstiprinātu vai pārbaudītu citus kvalitātes atribūtus sistēmā, piemēram, mērogojamība, uzticamība un resursu izmantošana. Veiktspējas kļūda ir kļūda, kuras galvenais simptoms ir neapmierinoša un pazemināta veiktspēja (piemēram, zema caurlaides spēja vai aiztures palielināšanās). Veiktspējas testēšanu var izmantot dažādiem mērķiem:

- pārbaudīt, vai sistēma atbilst veiktspējas kritērijiem;
- salīdzināt sistēmas savā starpā, lai izzinātu, kura strādā labāk;
- pārbaudīt, vai palielināta slodze sistēmas darbību ietekmē negatīvi.

Daudzi veiktspējas testi tiek veikti bez īpaši nostādītiem izpildāmiem mērķiem. [19]

1.2.10. Atmiņas testēšana

Atmiņas testēšanas mērķis ir novērtēt, vai sistēma apmierina vai neapmierina noteiktās atmiņas prasības. Daudziem programmatūras produktiem ir noteiktas dažādas atmiņas prasības, kas nosaka, piemēram, atmiņas apjomu operatīvajā atmiņā un palīgatmiņā, ko programmatūra drīkst izmantot. [19]

1.2.11. Uzstādījumu testēšana

Uzstādījumu testēšanas mērķis ir pārbaudīt, vai kopējā sistēma ir spējīga darboties tai noteiktajās konfigurāciju iespējās. Šī testēšanas metode tiek pielietota, testējotādus programmatūras produktus kā operētājsistēmas, datu bāzu vadības sistēmas, ziņu apmaiņas

programmas, kas atbalsta dažādas aparatūras konfigurācijas, tai skaitā dažāda tipa ievada un izvada ierīces, saziņas līdzekļus un atmiņas lielumus. [19]

1.2.12. Savietojamības, pārveidošanas testēšana

Savietojamības, pārveidošanas testēšanas mērķis ir pārlicināties, vai jaunais produkts, versija, ražotāja „*hot fix*” jeb ātrais labojums ir savietojams ar veco sistēmu atbilstoši noteiktajām prasībām un sistēmas darbībai. Gadījumos, kad sistēma tiek papildināta ar jaunu versiju vai ātru labojumu no sistēmas platformas ražotāja, vai arī sistēma netiek veidota pilnībā no jauna, bet ir kāda cita nepilnīga vai novecojuša produkta aizstājēji, tad bieži šādai versijai, labojumam vai sistēmai ir specifiskas prasības attiecībā uz savietojamību un procedūru pārveidošanu no esošās sistēmas. [19]

1.2.13. Atkopšanās testēšana

Atkopšanās testēšanas mērķis ir pārlicināties, vai tādi programmatūras produkti kā operētājsistēmas, datu bāzu vadības sistēmas, attālās piekļuves datu apstrādes sistēmas atkopjas pēc sistēmas kļūdām, aparatūras kļūmēm vai datu problēmām atbilstoši iepriekš definētām atkopšanās prasībām. [19]

1.2.14. Akcepttestēšana

Akcepttestēšana ir formāla testēšana, kurā uzsvars tiek likts uz lietotāja vajadzībām un biznesa procesiem, lai pārbaudītu vai sistēma apmierina vai neapmierina akceptēšanas kritērijus un dotu iespēju lietotājiem, klientiem un citiem, kam ir autorizēta pieeja sistēmai, izlemt vai sistēma ir akceptējama. Akceptēšanas kritēriji ir izejas kritēriji, kurus komponentei vai sistēmai ir jāapmierina, lai to akceptētu lietotājs, klients vai cita autorizēta persona. Akcepttests ir programmatūras vai aparatūras galīgā pārbaude, kas parāda izstrādātā produkta funkcionālās iespējas. Šī pārbaude nosaka, vai pārbaudāmais objekts darbojas atbilstoši līgumā fiksētajām prasībām (specifikācijai), t. i., vai tas atbilst akceptēšanas kritērijiem. [19]

1.2.15. Instalācijas testēšana

Instalācijas testēšanas mērķis ir atrast kļūdas sistēmas uzstādīšanas procesā. Instalācijas testus izstrādā organizācija, kas izstrādā sistēmu, un tos piegādā, kā daļu no sistēmas. Tie ir jāizpilda, kad sistēma ir uzstādīta. Testa piemēri var pārbaudīt , piemēram, vai ir izvēlētas savietojamas konfigurācijas, vai sistēmai ir visas nepieciešamās daļas, vai visi nepieciešamie faili ir izveidoti un tiem ir nepieciešamais saturs. [19]

1.2.16. Regresā testēšana

Regresā testēšana ietver iepriekš izveidoto un pārbaudīto testu atkārtotu izpildīšanu. Regresa testēšanu pielieto, lai pārlicinātos, ka jaunas izmaiņas vai kļūdu labojumi:

1. neievieš jaunas kļūdas,
2. ka iepriekš izstrādātā funkcionalitāte joprojām strādā tāpat.

Regresā testēšana tomēr pārklāj visas sistēmas testēšanu, bet neietver sīku, individuālu vai specifisku gadījumu atkārtotu testēšanu, tāpēc tai ir savi trūkumi, kā, piemēram,

1. visu sistēmu nevar pakļaut regresā testēšanai;
2. ja tiek darbināti tikai esošie testi, testētājs neatradīs kļūdas, kuras varētu atrast citi testi;
3. esošu testu uzturēšana, atjaunošana var būt sistēmai finansiāli neizdevīgi.

Regresai testēšanai uzlabojums ir automatizētie lietotāja grafiskās saskarnes, kas nodrošina ērtu, ātru un efektīvu automātisku izpildi un iegūto rezultātu pierakstu, bet šie testi izslēdz kļūdas, kuras varētu atrast citi testi. [19]

1.3. Testēšanas procesa kvalitātes kritēriji

Daudzos standartos testēšanas procesu definē kā savstarpēji saistītu aktivitāšu kopu, kas pārvērš ievadi (informāciju, objektus) par izvadi. Tradicionāli procesu var iedalīt vai nu mazākos procesos, vai arī konkrētās izpildāmās darbībās jeb soļos. Kā izriet no nodaļā 1.1 minētajām testēšanas definīcijām un praksē pielietotajām testēšanas metodēm un veidiem, testēšanas procesam ir iespējamās dažāda tipa ievades un izvades (skat. 1.3-1 tabula). Konkrētais veids ir atkarīgs no testēšanas veida.

1.3-1 tabula

Testēšanas procesa iespējamās ievades un izvades

Ievade	Izvade
prasības	testu saraksts
projektējumi	testēšanas rezultātu pieraksts – testēšanas žurnāls
modeļi	problēmziņojumi
izmaiņu pieprasījumi	vērtējums par programmatūras kvalitāti
standarti	
programmas kods	
darbināma programmatūra	
projekta organizatoriskie nosacījumi	

Testēšanas procesa kvalitāte ir vērtējama pēc iegūstamo rezultātu (izvaddatu) atbilstības testētāju un testēšanas procesā izvirzītajām prasībām. Ir iespējams nodefinēt trīs šādas galvenās prasības, kuras var izmantot par kritērijiem testēšanas procesa kvalitātes novērtēšanai:

1. Informācijas pieejamība - visai procesa informācijai jābūt pieejamai konkrētu vienumu vai vienumu kopas analīzei.
2. Operativitāte - jābūt iespējai ieinteresētajām personām pēc atbilstošu aktivitāšu norises operatīvi iegūt izvades informāciju.
3. Pilnīgums - procesā iegūtajiem rezultātiem jābūt pilnīgiem attiecībā pret ievades informāciju un izmantotās metodes sagaidāmajiem rezultātiem.

Iepriekšminētās prasības var būt apmierinātas vienā no trīs līmeņiem:

- zems – prasība tiek ievērota minimāli vai nemaz;
- vidējs – prasība tiek ievērota daļēji;
- augsts – prasība tiek ievērota pilnībā.

Ņemot vērā to, ka minētie kritēriji neizvirza testēšanas procesam specifiskus saturiskos kritērijus, tie vienlīdz labi var būt attiecināmi arī uz citiem programmatūras izstrādes procesiem. Prasību vērtējuma līmeņus var raksturot ar noteiktu punktu skaitu – 0 (zems), 1 (vidējs) un 2 (augsts). Līdz ar to, testēšanas procesa kvalitāti ir iespējams raksturot arī skaitliski ar vērtību robežās no 0 līdz 6 (piemēru skat. *1.3-2 tabula*) [15].

1.3-2 tabula

Testēšanas kvalitāti raksturojoši lielumi

	Informācijas pieejamība	Operativitāte	Pilnīgums
Augsts	2	2	2
Vidējs	1	1	1
Zems	0	0	0

1.4. Testējamo sistēmu kvalitātes metrikas

Testēšanas procesa kvalitāti un testējamās sistēmas kvalitāti var mērīt pēc dažādiem kvantitātes aspektiem, piemēram, testpiemēru skaitīšana, patērētā laika mērīšana uz testpiemēru izpildi, atrasto, atrisināto kļūdu skaitīšana u.c. metodes, kas palīdz izdalīt četras kvalitātes metrikas:

1. izstrādātāju iedotā koda kvalitāte,
2. testētāju atdotās sistēmas kvalitāte,
3. testēšanas dotais sistēmas kvalitātes uzlabojums,
4. testēšanas efektivitāte defektu uzrādīšanā.

Šīs metrikas ir noderīgas, ja ir nepieciešami skaitļi, tomēr, pamatoties uz tiem, objektīvi nevar novērtēt testēšanas kvalitāti dažādos sistēmas izstrādes periodos.

1.4.1. Koda kvalitāte

Šī metrika izsaka sakarību starp koda apjomu un tajā atrastajām kļūdām. Atrastās kļūdas arī tiek apskatītas no kritiskuma viedokļa un prioritātes. Piemēram, ir pieļaujams, ka paliks drukas kļūdas saskarnē, vai mazas kļūdas mazsvarīgos moduļos, ja pa to laiku, kas ietaupīts, mēģinot likvidēt šīs mazās kļūdas, tiks atklātas daudz lielākas un būtiskākas kļūdas. Tāpēc ir jāapzinās un skaidri jāsaprot, kuri sistēmā ir kritiskie moduļi un attiecīgi jāplāno darbības, ņemot vērā informāciju par sistēmas svarīgākajiem un būtiskākajiem moduļiem.

Programmatūras koda nozīmīgums parasti tiek iedalīts trīs pakāpēs [17, 18]:

- Augsta riska kods – kods, kurš potenciāli spēj nodarīt neatgriezeniskas sekas. Piemēram, no tā būtu atkarīgi finanšu dati vai datu bāzu integritāte. Gala lietotājam šāda kļūda nozīmē avāriju, kurā gala lietotājs nevar izmantot kādu sistēmas modeli, lai veiktu savus darba pienākumus ar sistēmu.
- Vidēja riska kods – kods, kurā individuālas kļūdas varbūt nav kritiskas, taču pārāk vēla to atklāšana vai neatklāšana, varētu prasīt salīdzinoši lielas izmaksas to novēršanai. Gala lietotājs šādas kļūdas pamana, piemēram, datu nepareizībā, kad noapaļošana netiek veikta korekti, vai summas atšķiras par vienu vienību. Šādas kļūdas ir kritiskas, bet tās netraucē veikt darba pienākumus, jo ir iespējams sistēmas darbību apiet, labot datus.
- Zema riska kods – kods, kurā, ja arī ir kļūdas, tad tās nevar ietekmēt produkta nodošanas termiņus, datu drošību vai kaut ko citu, kas būtu svarīgs. Šādas kļūdas parasti tiek labotas pēc akcepttestēšanas, piemēram, neatbilstoši biznesa loģikai apzīmēts kāds sistēmas elements, vai ieviesusies drukas kļūda.

Metrikas mērījums tiek iegūts kā kļūdas, kas atrastas pirms nodošanas gala lietotājam, un kļūdu, kas atrastas pēc nodošanas gala lietotājam, summas dalījums ar mainīto vai jauno koda rindiņu skaitu tūkstošos.

Diemžēl nav noteiktas metodes, veida, algoritma, kā sadalīt kodu šajās pakāpēs, bet to var secināt no biznesa procesu nozīmīguma, kuru jāuztur un jāvar realizēt sistēmā.

1.4.2. Sistēmas kvalitāte

Sistēmas kvalitāte mēra sakarību starp kļūdām (to smagumu un novēršanas izmaksām) un sistēmas vai laidiena apjomu pēc sistēmas nodošanas gala lietotājam. Metrikas mērījums tiek iegūts kā kļūdu, kas atrastas pēc nodošanas ražošanā, skaita dalījums ar mainīto vai jauno koda rindiņu skaitu tūkstošos.

1.4.3. Testēšanas dotais uzlabojums

Testēšanas dotais uzlabojums izsaka sakarību starp testēšanas komandas atrastajiem defektiem un sistēmas apjomu. Šajā metrikā netiek ņemtas vērā gala lietotāja atrastās kļūdas. Jo vairāk un smagākas kļūdas tiek atrastas, jo lielāku sistēmas uzlabojumu ir devusi testēšana. Metrikas mērījums tiek iegūts kā testēšanas komandu pirms sistēmas nodošanas ražošanā atrasto kļūdu skaita dalījums ar mainīto vai jauno koda rindiņu skaitu tūkstošos.

1.4.4. Testēšanas efektivitāte

Testēšanas efektivitāte izsaka sakarību starp kļūdām, kas atrastas testēšanas laikā un visām kļūdām, kas sistēmā bijušas. Metrika tiek izteikta procentos un noder, ja nepieciešams novērtēt testu kopas kvalitāti, un palīdz izveidot „dūmu testu” kopu sistēmas uzturēšanas posmam. Metrikas mērījums tiek iegūts kļūdu, kuras atraduši testētāji visā sistēmas izstrādes laikā pirms un pēc nodošanas gala lietotājam, skaita dalījums ar kļūdu, kas atrastas pirms nodošanas gala lietotājam, un kļūdu, kas atrastas pēc nodošanas gala lietotājam, summu pareizinātu ar simts procentiem.

1.5. Testēšana standartos

ISO standartos ir aprakstīts, ka testēšana ir tehniska operācija, kas sastāv no dotā produkta īpašību noteikšanas, un ka tas ir arī process vai serviss, kas tiek izpildīts pēc konkrētās procedūras. Citiem vārdiem sakot, testēšana ir salīdzināšanas, plānošanas un novērtēšanas process, kas orientēts uz informatīvās sistēmas īpašību noteikšanu un atšķirību uzrādīšanu starp faktisko un prasīto.

Parasti tiek uzskatīts, ka testēšanas galvenais uzdevums ir noteikt produkta atbilstību īpašībām, kuras uzstādītas pret šo sistēmu, bet testēšanas mērķis ir noteikt uz uzrādīt programmatūras defektus, kas izraisa programmatūras kvalitātes pazemināšanos. Tādejādi dažādas testēšanas aktivitātes var iedalīt verifikācijā un validācijā. Verifikācija ir pierādīšana, pārliecināšanās par konkrēta uzdevuma korektu izpildi. Validācija ir apstiprināšana, ka sistēma darbojas atbilstoši tai uzstādītajām prasībām.

Tomēr testēšana nesastāv tikai no testu izpildes, tas ir sistēmas pārbaudes process, kas nav stingri noteikts pēc izstrādes. Testēšana parasti tiek veikta iteratīvi, pārbaudot izstrādātāja paveikto, kas ietver gan sistēmas izstrādi atbilstoši prasībām, gan labotās kļūdas, un veicot gan procesa, gan rezultātu dokumentēšanu. Testēšanas process ietver plānošanu, kontroli un sagatavošanos, testu izpildi un rezultātu dokumentēšanu. Rezultātā testēšana dod novērtējumu par programmatūras produkta kvalitāti, uzrādot atšķirības starp vēlamu stāvokli un faktiski esošo

stāvokli. Tādējādi testēšana nodrošina informāciju, lai varētu pieņemt lēmumu par sistēmas nodošanu produkcijā.

Testēšana ir svarīga sistēmas dzīves cikla sastāvdaļa. Tā parasti tiek saistīta ar programmatūras kvalitātes nodrošināšanu, bet testēšanas rezultātus un sistēmas kvalitāti precīzi neviens nevar izmērīt, jo parasti sistēmas kvalitātes kritēriji tiek saistīti ar nemērāmiem lielumiem, piemēram, lietotāju apmierinātība.

Testēšanu kā kvalitātes aktivitāti apstāsta virkne IT nozares standartu. Saskaņā ar standartu *ISO/IEC 12207:1995 Information technology - Software life cycle processes* [4] (jeb 2002.gadā apstiprināto Valsts standartu *LVS ISO/IEC 12207:2002 Informācijas tehnoloģija - Programmatūras dzīves cikla procesi*) testēšanas procesam ir jānotiek kopā ar programmas koda izstrādi, kā arī pie produkta nodošanas klientam un produkcijā. Detalizēti testēšanas prasības ir atspoguļotas arī standartā *J-Std-016-1995 EIA/IEEE Interim Standard for Information Technology - Software Life Cycle Processes - Software Development Acquirer-Supplier Agreement* [5]. *J-Std-016* ir daudzos Latvijas IT uzņēmumos kļuvis par praksē izmantoto standartu dokumentācijas komplekta satura un formas noteikšanai. Testēšanai tas definē trīs dokumentus: testēšanas plānu, testu aprakstu un testēšanas pārskatu. Standarts *1465-1998 IEEE (12119:1998 ISO/IEC) Information Technology - Software Packages - Quality Requirements and Testing* [6] izvirza specifiskās prasības tieši pakotņu programmatūrai.

Latvijā ir divi standarti, kas ir tiešā veidā attiecināmi uz IT sistēmu testēšanu. *LVS 70:1996 Informācijas tehnoloģija - Programminženierija - Programmatūras testēšanas dokumentācija* [7] un *LVS 73:1996 Informācijas tehnoloģija - Programminženierija - Programmatūras vienībtestēšana* [8]. Kas ir veidoti pēc standartiem: *ANSI/IEEE Std 829-1983* (pašlaik jaunākā versija - *829-1998*[9]) un *ANSI/IEEE Std 1008-1987* [10]. Tie definē pieeju tam, kādai ir jābūt testēšanas dokumentācijai programmatūras projektos, un nosaka vienībtestēšanas (programmas funkciju, moduļu pārbaudes, izstrādājamās dokumentācijas) procesu. Standarts *ISO 9000:2001* [11] apraksta darba kvalitātes uzlabošanu vispārējās kvalitātes sistēmas ieviešanā.

1.5.1. LVS 70:1996

Bieži vien testēšanas process tiek uztverts kā tikai testpiemēru izpilde, kļūdu reģistrēšana, pārtestēšana un paziņošana, ka sistēma ir bez kļūdām. Tomēr, lai testēšanas process būtu sakārtots, lai sniegtu atskaites klientiem, lai testēšanas procesu varētu atkārtot pēc nepieciešamības un izmantot iegūto pieredzi citu projektu izpildē, nepieciešams izstrādāt arī testēšanas dokumentāciju, kas nav tikai testpiemēru apraksti. Testēšanas dokumentācija ietver dokumentus kā testēšanas plāns, dažādi pārskati, ziņojumi un specifikācijas, kas pilnvērtīgi apraksta projektā paredzēto, paveikto un sasniegtos rezultātus. Šos dokumentus uzņēmums var

izmantojot testēšanas procesa pārvaldības, izpildes, iegūto rezultātu izpētei, analīzei, kvalitātes novērtēšanai un pieredzes saglabāšanai, pilnveidošanai.

Zemāk esošajās sadaļās darba autore apraksta minimālo dokumentu komplektu, kas jāizstrādā testēšanas procesa gaitā, kā arī uzskaita, kādus dokumentus papildus ir ieteicams izstrādāt projekta ietvaros, atbilstoši Latvijas standartam *LVS 70:1996* [7].

Standarta autori min, ka tā lietojamība nav ierobežota ar programmatūras veidu, apjomu vai sarežģītību, kā arī programmatūras izstrādes posmu. Tomēr katrā uzņēmumā, kas testēšanu veic izstrādes projektiem vai testēšanas projektiem, nepieciešams, balstoties uz standartiem vai klientu prasībām, specificēt, kādus testēšanas dokumentus, to saturu un detalizācijas līmeni ir paredzēts projekta ietvaros izstrādāt iekšējai lietošanai, piemēram, statistikas veidošanai, un kādus dokumentus nodos klientam un savstarpēji saskaņos.

1.5.1.1. Testēšanas plāns

Testēšanas plāns ir jāizstrādā testēšanas procesa gaitā. Pēc pielietojuma tas ir paredzēts projekta vadītājam un vecākajam testētājam, kas seko projekta realizācijas gaitai. Šis dokuments var tikt saskaņots un savstarpēji parakstīts ar klientu. Dokumentā nepieciešams aprakstīt darbības sfēru, testēšanas pieeju, paredzētos resursus un testēšanas pasākumu kalendāro plānu, identificējot testējamus vienumus, raksturiezīmes, kas jātestē, testēšanas uzdevumus, kas jāizpilda, personālu, kurš ir atbildīgs par katru uzdevumu un risks, kurš ir saistīts ar projekta testēšanas procesu.

Testēšanas plānam jāsaturs vismaz šāda informācija:

1. Testēšanas procesa pieeja,
2. Testēšanas nodevumi,
3. Testējamie vienumi,
4. Testējamās funkcijas,
5. Testējama vienuma /funkcijas novērtēšanas kritēriji,
6. Testēšanas vides vajadzības,
7. Personāla komplektēšanas un apmācīšanas vajadzības,
8. Kalendārais plāns,
9. Risks un neveiksmju sekas,
10. Apstiprinājumi.

Papildus testēšanas plānā var iekļaut ievadu, informāciju par netestējamiem vienumiem un funkcijām, testēšanas uzdevumus.

1.5.1.2. Testpiemēru specifikācija

Testpiemēru specifikācija definē testpiemērus par testējamajiem vienumiem un funkcijām atbilstoši testēšanas procesa pieejai. Šis dokuments var tikt saskaņots un savstarpēji parakstīts ar klientu, lai novērtētu, vai sistēmā ir iekļauti visi klientam nepieciešamie biznesa procesi, kurus sistēmā ir jāvar izpildīt.

Testpiemēru specifikācijā ir jāiekļauj šāda informācija:

1. Testējamie vienumi un funkcijas,
2. Ievaddati, kas nepieciešami, lai izpildītu testpiemēru,
3. Izvaddati, kas nepieciešami, lai izpildītu testpiemēru,
4. Veicamās darbības testpiemēra izpildei,
5. Testēšanas vides vajadzības jeb aparātūra, programmatūra utt.,
6. Ierobežojumi,
7. Saraksts ar saistītajiem testpiemēriem.

1.5.1.3. Testēšanas žurnāls

Testēšanas žurnāls paredzēts testēšanas procesa vēstures jeb hronoloģisku ierakstu glabāšanai, rezultātā ļaujot izvērtēt testēšanas apjomu. Dokuments var tikt saskaņots un savstarpēji parakstīts ar klientu.

Šajā dokumentā ir secīgi jāapraksta testēšanas procesā veiktās

1. Darbības,
2. Notikumi ar testēšanas vienumiem un funkcijām.

Papildus ieteicams iekļaut vienumu un funkciju versiju un revīziju, identificēt testēšanas vides apstākļus, kas ietekmē testēšanu, testēšanas darbību izpildīšanās laiku, procedūras rezultātus un informāciju par veiksmīgu vai neveiksmīgu testa izpildi. Ja tests tiek reģistrēts problēmu ziņojumos (skat. nodaļu 1.5.1.5), atsauci uz to nepieciešams norādīt šajā dokumentā.

1.5.1.4. Testēšanas kopsavilkuma pārskats

Testēšanas kopsavilkuma pārskats visprecīzāk projektu vadītājam un vecākajam testētājam palīdz veikt kvalitātes izvērtēšanu. Šis dokuments ir jāizstrādā un jāsaskaņo, jāapstiprina ar klientu, sniedzot testēšanas darbību kopsavilkumu un novērtējumu, bāzējoties uz iegūtajiem rezultātiem.

Dokumentā nepieciešams iekļaut šādu informāciju par katru testējamo vienumu un funkciju:

1. Novērtējuma kopsavilkums, identificējot notestētos vienumus, funkcijas, to versiju un revīziju, papildus norādot uz citiem izstrādātajiem dokumentiem,
2. Novirzes no testēšanas plāna, testu specifikācijas, projektējuma, minot novirzes cēloni,

3. Rezultātu kopsavilkums, identificējot visas atrisinātās un neatrisinātās problēmas,
4. Novērtējums, balstoties uz testa vienumu, funkciju novērtēšanas kritērijiem un riskiem,
5. Darbību kopsavilkums, norādot kopējos resursu, patērēto laiku testēšanas darbībām,
6. Apstiprinājumi.

1.5.1.5. Testa problēmu ziņojums

Testēšanas problēmu ziņojums ir paredzēts Testēšanas nodaļas iekšējai lietošanai, nodrošinot dažādu problēmsituāciju uzskaiti un aprakstus, kurām nepieciešama papildus izpēte.

Dokumentā nepieciešams iekļaut šādas nodaļas:

1. Testa kopsavilkums, kas apraksta īsi apraksta testējamus vienumus un funkcijas, sagaidāmo rezultātu,
2. Problēmas apraksts, kas sevī iekļauj ievaddatus, atkārtēšanas scenāriju, izvaddatus, testēšanas vides raksturojumu.

Dokumentā ieteicams iekļaut saistītas darbības un novērojumus, kas var palīdzēt izolēt un novērst problēmas cēloni, kā arī iespaidus, kā atrastā problēma iespaido testēšanas plānu.

1.5.1.6. Testējamo vienumu žurnāls

Testējamo vienumu žurnāls identificē testējamus vienumus, kas tiek nodoti testēšanai gan izstrādātāja testētājam, gan klienta testētājam, ja projektā ir paredzēta akcepttestēšana. Šajā dokumentā viens tiek saprasts kā piegādes pakotne, kurā paredzēts notestēt kādu sistēmas daļu vai sistēmu kopumā. Šis dokuments atbilst un ir ieteicams izstrādāt projektos, kas realizēti pēc iteratīvām projektu pārvaldības metodēm, definējot par piegādes pakotnēm atbildīgās personas, to fizisko atrašanās vietu un statusu.

Testējamā vienumu žurnālā nepieciešams iekļaut šādu informāciju:

1. Nosūtāmie vienumi
3. To fiziskā atrašanās vieta, piemēram, e-pasts, serveris utt.,
4. Statuss, piemēram, testēšanā, notestēts, piegādāts produkcijas vidē,
5. Atbildīgā persona.

1.5.1.7. Testu projektējuma specifikācija

Testu projektējuma specifikācija ir ieteicams izstrādāt liela apjoma projektos, lai specificētu detalizētas testēšanas pieejas un identificētu vienumus un funkcijas, kas jātestē saskaņā ar šo projektējumu ar tam atbilstošiem testiem. Šo dokumentu izmanto projekta testētāji un vecākais testētājs testēšanas gaitas un kvalitātes kontrolei.

Testu projektējuma specifikācijai jāsaturs vismaz šāda informācija:

1. Testējamie vienumi un funkcijas,

2. Pieejas detalizēšana,
3. Vienumu un funkciju novērtēšanas kritēriji.

Papildus dokumentā var iekļaut īsus aprakstus par katru testpiemēru un procedūru, kas saistīta ar konkrēto projektu un dokumentu.

1.5.2. LVS 73:1996

LVS 73:1996 standarts apraksta vienībtestēšanas darbības un dokumentāciju. Vienībtestēšanas mērķis ir censties noteikt implementācijas korektumu un pilnību attiecībā pret vienības prasībām un projektēšanas dokumentāciju, mēģinot atklāt defektus:

1. prasītās vienības raksturozīmēs saistībā ar atbilstošiem stāvokļiem (piemēram, neaktīvais, aktīvais gaidot ziņojumu, aktīvais apstrādājot ziņojumu),
2. vienības darbībā ar nepareiziem ievaddatiem,
3. jebkurā lietošanas vai operāciju procedūrā, kas saistīta tikai ar vienību,
4. vienības algoritmos vai iekšējo datu struktūrās, vai abos kopā,
5. vienības vadības loģikā.

Latvijas standarta *LVS 73:1996* [8] autori min, ka vienībtestēšanai obligāti izstrādājami dokumenti ir Testu projektējuma specifikācija (skat. nodaļu *1.5.1.7*) un Testēšanas kopsavilkuma pārskats (skat. nodaļu *1.5.1.4*). Darba autore uzskata, ka nepieciešams izstrādāt vai papildināt arī pārējo dokumentāciju atbilstoši veicamajām darbībām, lai Testēšanas nodaļa varētu sekmīgi veikt testēšanas procesa kvalitātes izvērtēšanu.

Vienībtestēšanā veicamās darbības:

1. Testēšanas plānošanas fāze (testēšanas plāns (skat. nodaļu *1.5.1.1*)),
2. Testu kopas iegūšanas fāze (testu projektējuma specifikācija (skat. nodaļu *1.5.1.7*), testpiemēru specifikācija (skat. nodaļu *1.5.1.2*)),
3. Testējamās vienības mērīšanas fāze (Testēšanas kopsavilkuma pārskats (skat. nodaļu *1.5.1.4*), testējamo vienumu žurnāls (skat. nodaļu *1.5.1.6*), testēšanas žurnāls (skat. nodaļu *1.5.1.3*), testa problēmu žurnāls (skat. nodaļu *1.5.1.5*)).

1.5.3. ANSI/IEEE Std. 829-1998

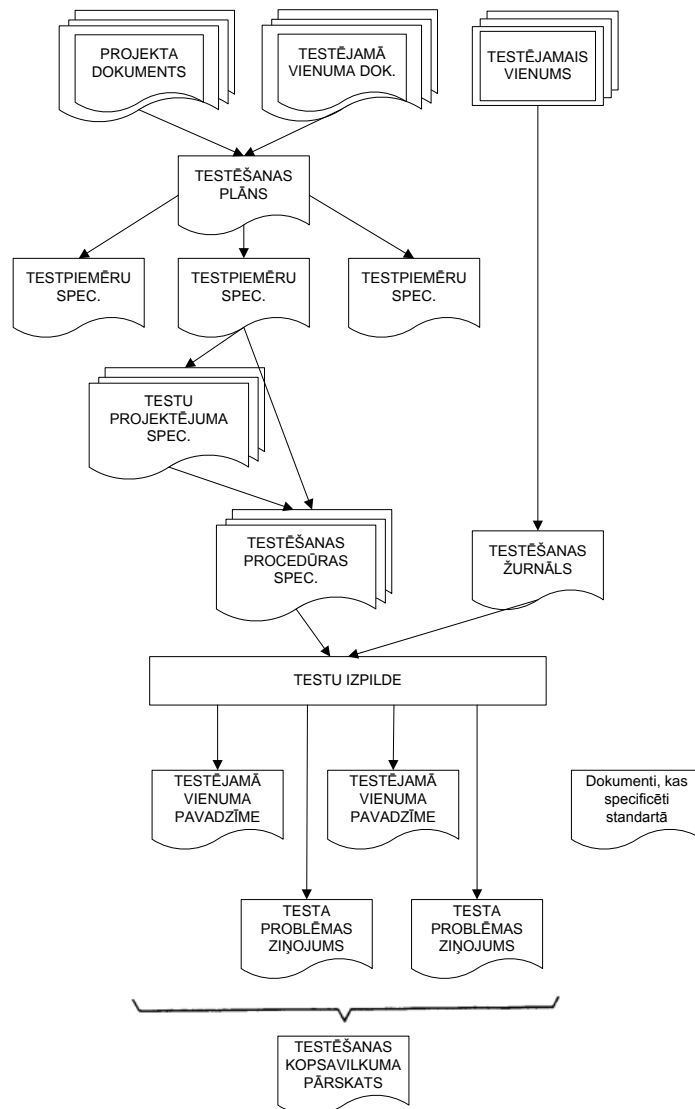
Šī standarta mērķis ir aprakstīt pamata programmatūras testēšanas dokumentāciju. Standartizētu testa dokumentācija var atvieglot komunikāciju, nodrošinot vienotu modeli (piemēram, klientam un piegādātājam ir vienādi definēts testa plāns). Satura definīcija standartizētai testa dokumentācijai var kalpot par kontrolsarakstu saistītajā testēšanas procesā. Standartizēts komplekts var nodrošināt pamatu, lai novērtētu pašreizējo testēšanas dokumentēšanas praksi. Daudzi uzņēmumi uzskata, ka standartā definēto dokumentu

sagatavošana būtiski palielina testēšanas vadāmību. Palielinot testēšanas procesa vadāmību, ievērojami jāpalielinās redzamībai katrā testēšanas procesa posmā.

Šis standarts nosaka formu un saturu individuālai testēšanas dokumentācijai. Tas nenosaka nepieciešamo komplektu testēšanas dokumentācijai. Uzņēmumos tiek pieņemts, ka nepieciešamas kopums testēšanas dokumentācijai tiek precizēts, kad standarts tiek piemērots uzņēmuma testēšanas procesam.

Testēšanas dokumentāciju standarts sadala 3 daļās – testēšanas plāns, testu specifikācijas un ziņojumi par testiem. Visu dokumentu mijiedarbība ir attēlota 1.5-1 attēls.

Pēc šī standarta ir veidots *LVS 70:1996 Informācijas tehnoloģija - Programminženierija - Programmatūras testēšanas dokumentācija* [7].



1.5-1 attēls Testēšanas dokumentu mijiedarbība

Standarts apraksta, ka testēšanas plāns nosaka darbības jomu, pieeju, resursus un testēšanas kalendāro grafiku. Testēšanas plāns identificē sistēmas modeļus, funkcijas un vienības, kuras ir jātestē, kādus testēšanas uzdevumus jāveic, testētāju kopu, kas atbild par saviem testēšanas uzdevumiem, un identificē riskus, kas saistīti ar testēšanas plānu.

Standarts nosaka, ka pie testēšanas specififikācijām pieder trīs dokumentu veidi:

- Testu projektējuma specififikācija viennozīmīgi nosaka darbības jomu un nosaka funkcijas, kuras testējot ir jāpārklāj, iekļaujot saistītās pārbaudes. Tā identificē testpiemērus un testēšanas procedūras, ja tādas nepieciešamas, lai veiktu pārbaudes un precizētu funkciju atbilstības / neatbilstības.
- Testpiemēru specififikācija dokumentē faktiskās ievades vērtības un iegūto rezultātu. Testpiemēri nosaka arī procedūru ierobežojumus. Testpiemēri tik atdalīti no testu projektējuma, lai tos varētu izmantot arī citās situācijās.
- Testēšanas procedūras specififikācija identificē visus vajadzīgos pasākumus, lai nodrošinātu sistēmas darbības testēšanu, balstoties uz testēšanas projektējuma specififikāciju.

Testu procedūras arī tiek atdalītas no testu projektējuma specififikācijas, lai to būtu vieglāk ievērot soli pa solim bez liekām sastāvdaļām. Ziņojumi par testiem tiek iedalīti četros dokumentu veidos:

- Testējamā vienuma dokuments identificē vienumu, kas paredzēts testēšanai, piesaistot vienumam nepieciešamās testu izpildes grupas.
- Testēšanas žurnālu izmanto visa testēšanas komanda, lai aprakstītu visu, kas noticis testu izpildes laikā.
- Testējamā vienuma pavadzīme apraksta visus notikumus, kas notiek visā testu izpildes laikā un prasa papildus izpēti.
- Testēšanas kopsavilkuma pārskatā tiek apkopotas testēšanas darbības, kas balstoties uz vienu vai vairākām testu projektējuma specififikācijām.

1.5.4. ANSI/IEEE Std. 829-2008

Šis standarts atbalsta visus programmatūras dzīves cikla procesus, tostarp, iegādes, izstrādes, ekspluatācijas un uzturēšanas. Šis standarts ir saderīgs ar visiem dzīves cikla modeļiem. Ne visi dzīves cikla modeļi izmanto visu, kas minēts šajā standartā. Sistēmas un programmatūras testēšanas process nosaka, vai tā atbilst paredzēto lietotāju vajadzībām un vēlmēm. Šis standarts ir paplašinājums standartam *IEEE Std, 829-1998* [9], kas ietver papildus jaunu dokumentāciju:

- Vadītāja testēšanas plāns (skat. nodaļu *1.5.4.1*);
- Testēšanas scenāriji (skat. nodaļu *1.5.4.2*);
- Testēšanas starposma statusa ziņojums (skat. nodaļu *1.5.4.3*).

1.5.4.1. Vadītāja testēšanas plāns

Vadītāja testēšanas plāna mērķis ir sniegt vispārēju testu plānošanas un testu pārvaldības dokumentāciju vairāku līmeņu testēšanai (vienam noteiktam projektam vai starp vairākiem

projektiem). Ņemot vērā programmatūras prasības un projekta kvalitātes nodrošināšanas plānu, Vadītāja testēšanas plāna izstrāde kā atsevišķa darbība sastāv no veicamo testēšanas darbību saraksta, nosakot katrai darbībai mērķus, nosakot darba dalīšanu (laiks, resursi) starp veicamajām darbībām, apzinot riskus, pieņēmumus un standartus, kas jāpielieto darbībām. Tas identificē integritātes līmeņa shēmas, testēšanas līmeņu skaitu, kopējos veicamos uzdevumus un prasību dokumentāciju.

1.5.4.2. Testēšanas scenāriji

Testēšanas scenāriju mērķis ir noteikt nepieciešamo informāciju kā izejošo un ienākošo informāciju programmatūrai, kas tiek testēta. Testēšanas scenāriji ietver visus scenārijus, kas pārklāj testēšanas plānu. Dokumenta detalizācija var atšķirties atkarībā no aprakstāmās testēšanas darbības. Dokumentā ir jāapraksta kopsavilkums par testēšanas statusu, kādas ir izmaiņas novirzes no testēšanas plāna un novirzes no Vadītāja testēšanas plānā definētajām darbībām.

1.5.4.3. Testēšanas starpposma statusa ziņojums

Šī dokumenta mērķis ir apkopot izvēles kārtībā noteiktu testēšanas darbību rezultātus un sniegt novērtējumus vai ieteikumus, pamatojoties uz šiem rezultātiem.

1.5.5. ANSI/IEEE Std. 1008-1987

Šis standarts saka, ka programmatūras vienībtestēšana ir process, kas ietver pārbaudāmo plānošanu, testa komplekta sagatavošanu un testēšanas vienību pārbaudes pret to prasībām. Šajā gadījumā mērīšana nozīmē izmantot vienkāršus izlases datus, lai izpildītu vienību un salīdzinātu iegūto rezultātu ar prasībās definēto uzvedību.

Šī standarta galvenais mērķis ir noteikt standarta pieeju programmatūras vienībtestēšanai. Otrais mērķis ir aprakstīt programmatūras inženierijas koncepcijas un testēšanas pieņēmumus, uz kuriem šis standarta pieeja balstās. Trešais mērķis ir nodrošināt nepieciešamos resursus un informāciju, lai palīdzētu īstenot šī standarta izmantošanu vienībtestēšanas pieejā.

Standartā ir definētas šādas vienībtestēšanas aktivitātes:

1. Testu plānošanas posms, kas ietver vispārējās pieejas, resursu un laika plānošanu, pārbaudāmo funkciju noteikšanu, vispārējā plāna precizēšanu.
2. Testēšanas komplekta definēšanas posms, kas ietver informāciju par pārbaudīto kopumu, īstenojot precizēto testēšanas plānu.
3. Vienībtestēšanas rezultātu analīzes posms, kas ietver testēšanas procedūru izpildi un pārbaudi, iegūto rezultātu, vienību un veikto darbību novērtēšanu.

Šis standarts ir izmantots *LVS 73:1996* [8] standarta sagatavošanā.

1.6. *ISTQB* sertificēts testētājs

ISTQB (Starptautiskā Programmatūras testēšanas kvalifikācijas padome) tika dibināta 2002.gada novembrī Beļģijā. *ISTQB* ir definējusi „*ISTQB Certified Tester*” jeb „*ISTQB sertificēts testētājs*” shēmu, kas ir kļuvusi par pasaules mēroga līderi sertificētāju programmatūras testētājiem.

ISTQB Certified Tester apraksta mācību kursu testētājiem, lai iegūtu *ISTQB* attiecīgā līmeņa sertifikātu. *ISTQB Certified Tester* ir pasaulē visizplatītākā testētāju kvalifikācijas sistēma, kā arī *ISTQB* mācību materiāli un vārdnīca ir kļuvuši par galveno kvalifikācijas noteicēju testētāju profesijā [21].

ISTQB apraksta 7 testēšanas principus un vispārīgas vadlīnijas testēšanai [14]:

1. Testēšana attēlo kļūdu esamību. Testēšana var parādīt vai šobrīd sistēma ir ar kļūdām vai nav. Testēšana samazina varbūtību neatklātām kļūdām, bet ja kļūdas netiek konstatētas testēšanas procesā, tas nav pierādījums, ka sistēma ir bez kļūdām.
2. Pilnīga testēšana ir neiespējama. Pilnīga sistēmas testēšana nav iespējama, izņemot triviālas lietas. Pilnīgas testēšanas vietā ir jāņem vērā risku analīze un sistēmas funkcionalitātes un elementu prioritātes.
3. Pirmie testi. Lai atrastu kļūdas pēc iespējas agrākā sistēmas izstrādes posmā, nepieciešams arī testēšanu uzsākt pēc iespējas ātrāk sistēmas izstrādes dzīves ciklā, koncentrējoties uz noteiktiem mērķiem.
4. Kļūdu apkopošana. Testēšanā veiktais darbs ir jāvērs uz atrasto kļūdu analīzi. Mazi sistēmas moduļi parasti sastāv no visvairāk kļūdām, kuras tiek atklātas sistēmas izstrādes dzīves cikla sākumā.
5. „Kaitēkļu paradokss”. Ja viens tests tiek atkārtots atkal un atkal, iespējams tā pati testpiemēru kopa vairs neatradīs jaunas kļūdas. Lai pārvarētu šo „kaitēkļu paradoksu”, testēšanas rezultāti ir regulāri jāanalizē un jāraksta jaunos testus, kas pārklāj arī citas sistēmas daļas, lai atrastu pēc iespējas vairāk kļūdas.
6. Testēšana ir atkarīga no situācijām. Dažādās situācijās testēšana tiek veikta atšķirīgi.
7. „Kļūdu prombūtne” ir maldi. Kļūdu atrašana un labošana nepalīdz, ja izstrādātā sistēma ir nelietoājama un neatbilst lietotāju vajadzībām un vēlmēm.

ISTQB saka, ka vispamanāmākā testēšanas daļa ir testu izpilde. Bet, lai būtu efektīvs un produktīvs, testēšanas plāniem ir jāiekļauj gan laiks, kas tiek pavadīts plānojot testus, izstrādājot testēšanas scenārijus, sagatavojot testēšanas vidi un novērtējot rezultātus.

ISTQB saka, ka pamatos testēšanas process sastāv no šādām galvenajām aktivitātēm [14]:

1. testu plānošana un kontrole;
2. testu analīze un projektēšana;

3. testu ieviešana un izpilde;
4. izejas kritēriju novērtēšana un ziņošana;
5. testēšanas aktivitāšu slēgšana.

ISTQB raksta, ka tipiskie testētāja uzdevumi iekļauj darbības kā [14]:

1. pārskatīt un pilnveidot testēšanas plānus,
2. analizēt, pārskatīt un novērtēt lietotāju prasības, specififikācijas un testēšanas moduļus,
3. izstrādāt testu specififikācijas,
4. sagatavot testēšanas vidi (bieži saskaņojot ar sistēmas administrēšanu un tīklu pārvaldību),
5. sagatavot un izmantot testu datus,
6. izmantot sagatavotos testpiemērus pa visiem testēšanas līmeņiem, izpildīt un pierakstīt testu izpildi, izvērtēt rezultātus un dokumentēt novirzes no gaidāmajiem rezultātiem,
7. izmantot testu administrācijas vai pārvaldības un uzraudzības rīkus, kurus pieprasa sistēmas vai projekta specifika.
8. automatizēt testus (to izstrādē var palīdzēt sistēmas izstrādātājs vai automatizēto testu speciālists),
9. mērīt veiktspēju sistēmas komponentēm,
10. pārskatīt citu testētāju veidotos testus.

Lai izstrādātu un pārskatītu pienākumos minētos dokumentus, *ISTQB* iesaka izmantot *IEEE STD 829-1998* standartā aprakstīto dokumentu struktūru.

1.7. LEAN metodoloģija

Tā kā mūsdienās ir liels piedāvājums testēšanas standartiem un metodēm, kas aptver maksimāli plašu testētāju pienākumu kopu, tad darbā tiek apskatīta *LEAN* metodoloģija (turpmāk *LEAN*), kas definē, ka rezultātu iegūšanas procesā nepieciešams darbības kārtot secīgos soļos, izmantojot tikai soļus ar pievienotu vērtību. Pievienotā vērtība ir darba pienākumu un veikto darbu tiešais labums projektam, klientam, projekta komandai, bez kura projekts, ir neveiksmīgs, nepilnvērtīgs. Procesā nedrīkst izmantot izšķērdīgus, nevērtīgus soļus, lai rezultāta iegūšanas procesā pēc iespējas mazāk patērētu laiku un resursus papildus “birokrātiskām” darbībām [1].

LEAN ir paredzēta uzlabot produktu vai pakalpojumu analīzes, izstrādes, testēšanas procesus, lai izšķērdīgus procesus pārvērstu vērtību pievienošos procesos, iegūstot izaicinājumu un regulāras atsauksmes no klientiem. *LEAN* nosaka, ka produkta pilnajā izstrādes procesā ir

jāatbrīvojas no visiem liekajiem, izšķērdīgajiem, pievienoto vērtību nedodošajiem procesiem, nodevumiem un darbiem. *LEAN* saka, ka izšķērdība ir projekta ienaidnieks [2].

LEAN autori saka, ka izmantot, patērēt vai darbināt nelietderīgus, bez pievienotas vērtības esošus procesus ir neproduktīvi, disfunkcionāli un potenciāli postoši jebkuram projektam. Tāpēc nepārtraukta uzlabošana, samazinot izšķērdīgus procesus jāklūst par prioritāti projektu ietvaros, lai uzņēmums sasniegtu augstākus rezultātus [3].

LEAN mērķis ir samazināt projektos izšķērdību un uzlabot augstāku produkta vērtību gala lietotājam. Izšķērdība ir jebkura darbība, kas izmanto resursus, bet neatgriež pievienotu vērtību, piemēram, papildus procesi, defekti, daļēji izdarīta darbība.

Daļēji pabeigti darbi var kļūt par novecojušiem un nelietojamiem. Risinājums – produkta izstrādi veikt iteratīvi, veicot koda sakārtošanu. Defekti ir korekcijas un funkciju atkārtošanās, ko var risināt ar testu kopas atjaunošanu un klientu atsauksmēm. Papildus procesi var būt, piemēram, lieka, neizmantojama dokumentācija vai papildus funkcionalitāte, kuru klients nav pasūtījis, prasījis izstrādāt. Šajā gadījumā risinājums ir veikt patstāvīgas dokumentācijas pārbaudes un nepārtraukt sadarbību ar klientu.

Pie izšķērdīgām darbībām tiek pieskaitīti draudi arī par multi-uzdevumu pildīšanu, kas prasa laiku gan uz uzdevumu pārslēgšanos, gan izpildes kvalitāti. *LEAN* iesaka tomēr pieturēties pie noteiktiem uzdevumiem katras produkta relīzes ietvaros. Kā arī visiem produkta izstrādes procesā iesaistītajiem ir jābūt vienmēr aizpildītiem ar darbiem, ļaujot piekļūt informācijai par produktu un ļaujot iesaistītajiem veikt arī savus lēmumus nepieciešamības gadījumos.

Programmatūras izstrādes procesā izšķērdība parasti tiek iedalīta 3 veidos [1]:

- izšķērdība produkta koda izstrādē,
- izšķērdība projekta pārvaldībā,
- izšķērdība darbaspēka potenciālā.

LEAN izveides aizsākumi ir meklējami 1450.gados, Venēcijā, kur Henrijs Fords bija pirmais cilvēks, kas patiesi uzsāka visu ražošanas procesu integrāciju, izsākot darbā ieviest plūsmas (secīgu darbību) ražošanu. H. Fords ierindoja detaļu ražošanas soļus procesa kārtībā, lai pēc iespējas vairāk izmantotu ražošanai paredzēto aparatūru detaļu apstrādei un sastiprināšanai kopā, nodrošinot, ka produkts tiek pagatavots dažu minūšu laikā bez liekiem papildus soļiem un gaidīšanas [2].

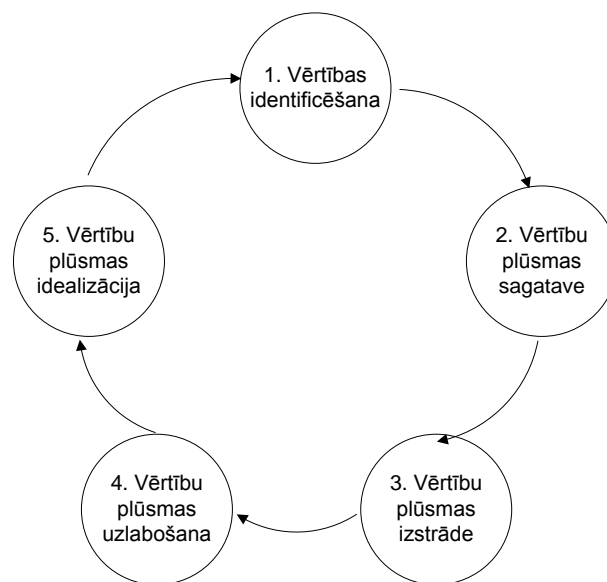
Forda ideja bija liels pavērsiens automašīnu izstrādes procesā, bet šāds produktu izstrādes modelis nespēja sniegt dažādību produktu klāstā. Produkti bija limitēti ne tikai pēc krāsojuma, bet arī funkcionāli, atbilstot vienai specifikācijai, bet patērētājs vēlējās dažādību.

Pēc Otrā pasaules kara automašīnu būves uzņēmums „Toyota” atgriezās pie H. Forda idejas, uzlabojot to. Rezultātā uzsvars produktu ražošanā tika likts nevis uz atsevišķu detaļu

plūsmas ražošanas pārvaldību, bet uz visa produkta izstrādes procesa pārvaldību. „Toyota” secināja, ka, lai palielinātu automašīnu faktisko ražošanas apjomu, nepieciešams ieviest paškontroles mašīnas, lai uzlabotu kvalitāti un ražošanu sakārtotu secīgi. Katra mašīnas bija jāveic neliela apjoma darbības katras detaļas apstrādei. Katrs paveiktais solis informēja iepriekšējo soli, ka tas var darboties tālāk. Kopā šāda mašīnu ražošanas pārvaldības metode nodrošināja zemas izmaksas, augstu kvalitāti un ātrus izskatīšanas termiņus, ļaujot ražotājam reaģēt uz mainīgajām klientu vēlmēm.

Mūsdienās *LEAN* programmatūru izstrādē tiek izmantota kopā ar spējās programmatūras izstrādes iteratīvu projekta pārvaldības metodoloģiju, definējot 5 principus, kā pielāgot savu izstrādes procesu vērtību ienesošam darbam (skat. *1.7-1 attēls*) [1]:

1. specificēt, saprast klienta gala lietotāju vajadzības un vēlmes par produktu;
2. identificēt visus vērtīgus soļus produkta izstrādei un likvidēt visus soļus, kuros tiks nelietderīgi tērēts laiks un netiks saņemta vērtība;
3. sagatavot produkta izstrādes procesu nepārtrauktu, izmantojot tikai soļus ar pievienoto vērtību, tā, lai produkts bez aizķeršanās nonāktu pie klienta;
4. līdz plūsma ir ieviesta, atļaut klientam pievienot vērtību pie nākamā posma aktivitātēm;
5. līdz klienta vērtība ir saņemta, vērtību plūsmas ir identificētas, izšķērdīgie soļi ir noņemti un plūsma ir ieviesta, sākt procesu no jauna un turpināt to līdz pilnības stāvoklis ir sasniegts bez izšķērdīgiem soļiem.



1.7-1 attēls **LEAN principi darba organizācijas uzlabošanai**

1.7.1. Vērtības identificēšana

Vērtību specificēšana ir funkciju noteikšana, kuras nepieciešams iestrādāt, notestēt, sagatavot produktā, balstoties uz klientu un gala lietotāju viedokļiem. Vērtība / funkcija tiek noteikta atbilstoši klienta specifiskajām vajadzībām, atbilstoši cenai un ieguldāmajam laikam. Konkrētiem produktiem vai pakalpojumiem tiek novērtēts, kuras vērtības ir jāpievieno. Vērtības var tikt iegūtas no produkta vai pakalpojuma gala lietotājiem vai vēlākos projekta procesos (skat. *1.7-1 attēls* pirmais posms) [3].

1.7.2. Vērtību plūsmas sagatave

Ja ir identificētas vērtības, tad ir identificētas arī veicamās darbības. Visu darbību secību sauc par vērtību plūsmu. Identificējot vērtību plūsmu, tiek pārbaudīts, kuras darbības neveicina vērtības produktam vai pakalpojumam. Nepieciešamās darbības tiek definētas kā priekšnoteikums citām pievienotajām vērtībām vai būtiska daļa no biznesa prasībām. Vērtības, kuras neveicina produkta vai pakalpojuma vērtības, ieteicams nepieņemt vērtību plūsmā vai izslēgt no procesa pilnībā (skat. *1.7-1 attēls* otrais posms) [3].

Lai varētu efektīvi identificēt vērtību plūsmu, nepieciešams veikt sekojošus soļus: sagatavot biznesa sistēmu, lai veicinātu *LEAN* domāšanu, apmācīt *LEAN* domāšanas ideju un prasmes gan darbiniekiem (mūsu gadījumā testētājiem), gan klientiem, lai ikviens varētu izveidot funkcionējošu vērtību plūsmu un apzināties izmaksas ikvienā no veicamajām vērtību plūsmas darbībām. Lēmumus pieņemt tikai pēc pilnīgas informācijas iegūšanas no klientam bet tā kā klients mēdz mainīt savas domas, ir jāvar risinājums piegādāt pietiekoši ātri, lai klients neuzspētu mainīt savas domas.

1.7.3. Vērtību plūsmas izstrāde

Kad vērtīgās un nevērtīgās darbības ir identificētas, var veikt uzlabojumus vērtību plūsmai. Uzlabošana ir paredzēta vērtību plūsmas darbību izpildes laika, izmaksu un kļūdu samazināšanai, fokusējoties uz klientu prasībām. Plūsma ir nepārtraukta produkta vai pakalpojuma plūsma, izmantojot sistēmas lietotāju vēlmes un vajadzības. Lielākie posmi tiek iedalīti plūsmas darbu rindā, partiju apstrādē un transportēšanā. Šie iedalījumi palēnina produkta vai pakalpojuma izstrādes uzsākšanas līdz nodošanas laiku. Šādi iedalījumi var tikt izmantoti citur organizācijā un nosegt sistēmas ierobežojumu ietekmi un darbību izšķērdēšanu (skat. *1.7-1 attēls* trešais posms) [3].

Plūsmas uzlabošana tiek veikta tikai gadījumos, kad gala lietotājs ir lūdzis pievienot nepieciešamās prasības un tās ir ar pievienoto vērtību. Uzlabot plūsmu vajag ne tikai noteiktām produkta sastāvdaļām, bet arī visam produktam kopumā.

1.7.4. Vērtību plūsmas uzlabošana

Ja izšķērdējošās darbības ir noņemtas no vērtību plūsmas, jācenšas klientam izpildīt produktu vai pakalpojumu, izmantojot procesu. Organizācijai ir jāizmanto produkts vai pakalpojums tikai sistēmas lietotāju vajadzībām, ne pirms, ne pēc (skat. *1.7-1 attēls* ceturtais posms).

1.7.5. Vērtību plūsmas idealizācija

Šie darbi ir paredzēti atkārtoti un patstāvīgu mēģinājumu veikšanai, lai atbrīvotos no izšķērdīgajām vērtībām, uzlabojot vērtību plūsmu un klientu apmierinātību piegādes vajadzībām. Kamēr *LEAN* fokusējas uz izšķērdēšanas novēršanu un vērtību plūsmas uzlabošanu, tas arī var radīt neizskaidrojamus defektus, ja pārlietu un nemākulīgi to izmanto. Kvalitāte ir uzlabota. Produkts vai pakalpojums mazāk laika pavada procesa izpildei, samazinot bojājumu iespēju vai novecošanos. Ja sistēma skatās uz visām darbībām dažādas variācijas, sistēmas ierobežojumi tiek noņemti (skat. *1.7-1 attēls* piektais posms) [3].

LEAN veido dažus pieņēmumus:

- cilvēki augstu vērtē vizuālu vērtību darbu plūsmas,
- izšķērdība ir galvenais ierobežojums uz rentabilitāti,
- daudzi mazi uzlabojumi ir vieglāk uztverami, nekā lieli analītiskas pētījumi.

LEAN ietver daudz cilvēkus vērtību plūsmā. Pārejot un vērtību plūsmu domāšanas metodiku, izraisa milzīgas pārmaiņas domām, kā cilvēks uztver savu lomu organizācijā un to attiecības ar produktu vai pakalpojumu (skat. *1.7-1 tabula*).

LEAN ietvaros tiek veikta produkta vai pakalpojuma konfigurācija, pārveidošana, uzlabojot sistēmas procesus [1]. *LEAN* saka, ka

- produkta vai pakalpojuma dizains būs pareizs,
- produkta vai pakalpojuma dizains ir visekonomiskākais,
- klienta vajadzības tiek apmierinātas ar šo dizainu,
- pašreizējā produktu un pakalpojumu konfigurācija atbilst funkcionālajām prasībām,
- vadības struktūra atbalsta un palīdz uzlaboties un mainīties.

1.7-1 tabula

***LEAN* metodoloģijas kopsavilkums**

Pazīme	<i>LEAN</i> metodoloģija
Teorija	Izšķērdības aizvākšana
Metodikas vadlīnijas	1. Identificēt vērtības 2. Identificēt vērtību plūsmu 3. Plūsma

Pazīme	LEAN metodoloģija
	<p>4. Papildinājumi</p> <p>5. Plūsmas idealizācija</p>
Fokuss	Fokusēts uz plūsmu
Pieņēmumi	Izšķērdīgo vērtību noņemšana uzlabos biznesa procesus. Daudz mazu uzlabojumu ir labāk nekā sistēmas analīze.
Primārā iedarbība	Reducēts plūsmas izpildes laiks
Sekundārā iedarbība	<p>Mazāk produkta vai pakalpojuma variāciju.</p> <p>Vienots pakalpojumu vai produktu izvads.</p> <p>Nepieciešams mazāk aprīkojuma plūsmas nodrošināšanai.</p> <p>Jauna uzskaites sistēma.</p> <p>Vērtību plūsma – veikspējas mērvienība menedžeriem.</p> <p>Uzlabota kvalitāte.</p>
Iebildumi	Statistika vai sistēmas analīze ir nevērtīgas darbības.

2. UZDEVUMA IZPĒTE

Šajā nodaļā ir aprakstīti darba mērķu sasniegšanai veiktie izpētes darbi.

2.1. Veicamie darbi mērķu sasniegšanai

Precizējot darba tēmu, tika nolemts veikt izpēti par sistēmas testēšanas procesā obligāti veicamajiem pienākumiem, kas atbilst testēšanas metodēm, testēšanas procesā sagatavojamai dokumentācijai, sistēmu izstrādes un ieviešanas konkursu nolikumiem, testēšanas procesu aprakstošiem starptautiskiem un Latvijas standartiem un testētāju viedoklim par testēšanas procesā izvirzītajām prasībām. Darba izpētes procesa realizācijai tika izvirzīti šādi uzdevumi:

- Veikt izpēti šobrīd pieejamām un realizējamām testēšanas metodēm, norādot to ieguvumus un priekšrocības projekta realizācijā;
- Apkopot viedokli par testēšanas procesa kvalitātes kritērijiem un metrikām;
- Analizēt Latvijā un Eiropā definētus standartus testēšanas procesam, atzīmējot testētājam izstrādājamās dokumentācijas apjomu un pielietojamās testēšanas metodes atbilstoši standartiem, ja tādas ir definētas;
- Analizēt *ISTQB* sertifikācijas programmas noteikumus un nepieciešamo zināšanu apjomu, lai iegūtu šādu sertifikātu un strādātu atbilstoši tajā definētajiem ieteikumiem;
- Analizēt *LEAN* metodoloģijas priekšnosacījumus, lai sistēmas testēšanas procesu varētu tai pielāgot;
- Noteikt minimālo dokumentācijas un pielietojamo testēšanas metožu kopu, kas atbilst *ISTQB* sertifikācijas programmas noteikumiem, Latvijā un Eiropā definētiem standartiem un *LEAN* metodoloģijai, lai spētu noteikt:
 1. Vai projekta ietvaros ir veiktas visas nepieciešamās darbības sistēmas testēšanas procesā.
 2. Vai projekta ietvaros izstrādātā dokumentācija atbilst starptautiski un Latvijā definētiem standartiem, bet neietver visu standartos prasīto informāciju, ietver tikai projektam un klientam būtisku informāciju.
- Izstrādāt programmatūras testēšanas procesu kvalitātes novērtēšanas kritērijus atbilstoši iepriekš noteiktai minimālai dokumentācijas un testēšanas metožu kopai.

2.2. Aptauja „Programmatūras testēšanas procesu kvalitātes novērtēšana”

Bakalaura darba ietvaros tika izstrādāta anonīma aptauja „Programmatūras testēšanas procesu kvalitātes novērtēšana” (skat. 1. pielikums), kuras mērķis ir iepazīt, izziņāt Latvijā esošu testētāju viedokli par ikdienā un projektos veicamajiem pienākumiem.

Aptaujas jautājumi apkopo vispārīgu informāciju gan par projektiem, projektu pārvaldības metodēm, kuros testētāji darbojas un kā tiek noteikti to pienākumi, gan informāciju, kādus dokumentus, kādas testēšanas metodes visbiežāk testētāji izmanto veicot testēšanas darbu, kā arī tiek apkopots testētāju viedoklis par *LEAN*.

Aptauja kopā ietver 16 jautājumus un ir publiski pieejama vietnē [visidati.lv](http://www.visidati.lv): <http://www.visidati.lv/aptauja/820056044/15/>. Kā personiskā informācija respondentiem ir jānorāda savs vecums un izglītības līmenis, kas respondentus ļauj sadalīt atbilstošās vecuma un izglītības līmeņu grupās.

Sagatavojot aptaujas jautājumus darba autore centās apmēram paredzēt, kādas atbildes lielākoties tiks saņemtas, balstoties uz darba pieredzi projektos un uzņēmumos. Liela daļa atbilžu (skat. 2. pielikums) atbilda sagaidītajam rezultātam, bet daļa atbilžu darba autori pārsteidza, piemēram, ir testētāji, kas godīgi atzina, ka testēšanas metodēm nepārzina latviskos terminus, attiecīgi nespēja atbildēt uz noteiktiem jautājumiem.

Kopā aptaujā piedalījās 23 respondenti no dažādiem Rīgas un Latvijas mēroga uzņēmumiem, kas sniedz sistēmas izstrādes, ieviešanas, uzturēšanas pakalpojumus un testēšanas pakalpojumu citiem uzņēmumiem. Respondentu vecums svārstījās no 23 līdz 52 gadiem, nodrošinot vidējo vecumu 28 gadi.

Lielākā daļa respondentu jeb 11 testētāji ir ar maģistra izglītību, 8 respondenti ir ieguvuši bakalaura izglītību, 3 respondentiem ir nepabeigta augstākā profesionālā izglītība.

Lielākā daļa respondentu jeb 22 testētāji darbojas ieviešanas un izstrādes projektos, no kuriem 13 respondenti darbojas arī uzturēšanas projektos. 6 respondenti nodrošina testēšanu, kā ārpalpojumu citiem uzņēmumiem, bet tikai 1 respondents darbojas 10 gadus vecā uzturēšanas projektā.

Lielākā daļa respondentu jeb 16 atbildētāji darbojas 1 – 3 gadus garos projektos, starp kuriem vienlīdzīgi pa 7 atbildēm ir norādīti arī 1 – 3 cilvēkmēnešus, 4 – 7 cilvēkmēnešus un 7 – 12 cilvēkmēnešus gari projekti.

Kā projektu pārvaldības metodes visvairāk respondentu vidū ir atzīmētas „Ūdenskritums”, „Spējās programmatūras izstrādes metodoloģija *Agile*” un „Nenoteikts”, kur zem „Nenoteikts” var saprast, ka tas ir abu augstākminēto projektu pārvaldības metožu krustojums. 7 respondenti ir atzīmējumi *SCRUM*. 3 respondenti zem atbildes varianta „Cits” ir ierakstījuši V-modeli kā

projektu pārvaldības metodi un viens respondents min, ka tiek izmantots uzņēmuma iekšēji izveidots programmatūras izstrādes process.

2.2.1. Aptaujas atbilžu par prasībām pret testētāju analīze

Galvenokārt, aptauja tika koncentrēta uz jautājumiem par testētāja darba pienākumiem, to sadali, organizēšanu, veicamajiem darbiem un to nepieciešamību.



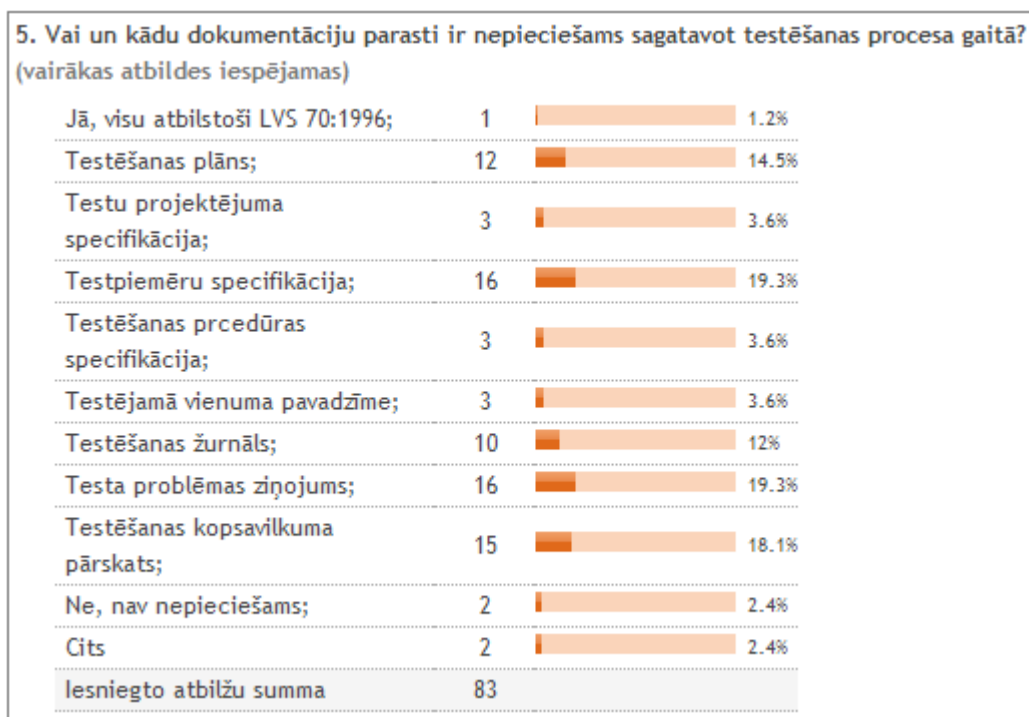
2.2-1 attēls Aptaujas 4. jautājums

Pēc testētāju atbildēm (skat. 2.2-1 attēls) seko, ka pusē uzņēmumu prasības pret testētāju ir aprakstītas uzņēmuma iekšējos normatīvajos aktos vai arī testētājs darbojas atbilstoši projekta vadītāja norādēm, bet 9 testētāji tomēr darbojas pēc saviem ieskatiem, kas liecina, ka ir daļa uzņēmumu, kas pat neapraksta testētāja profesijas pienākumus, vai arī projektu vadītāji nav ieinteresēti testētāju darbos, kas tomēr nozīmīgi ietekmē projekta un sistēmas kvalitāti, jo projekta vadītājam ir jāpārzina, kuras sistēmas daļas un pēc kādām metodēm tās tiks testētas. Viena piektā daļa testētāju jeb 5 respondenti ir atzīmējuši, ka uzņēmumā ir pieejams testētāja profesijas apraksts, kā arī uzņēmums piedāvā „rokasgrāmatu” par testēšanas procesa norisi šajā uzņēmumā. Viena ceturtda daļa respondentu jeb 6 testētāji ir atzīmējuši, ka projekta vadītājs seko projekta līgumā noteiktām prasībām, attiecīgi testētāja darba uzdevumi, kas ir gan sagatavojamā dokumentācija, gan veicamās testēšanas metodes, tiek veikti atbilstoši līgumam, iespējams, neveicot papildus pienākumus iekšējām projekta vajadzībām.

2 respondenti atzīmēja variantu „Cits”, norādot, ka prasības pret testētāju ir aprakstītas uzņēmuma kvalitātes sistēmā vai arī uzņēmums testētājam piedāvā „rokasgrāmatu” mutiskā

formā par testēšanas procesa norisi. Viens respondents min, ka testējamajam produktam uzņēmumā nav izstrādāta testēšanas metodika, kas liecina, ka sistēmai pat netiek piemērota parasta regresijas testēšana, kas ļauj jau esošajā sistēmas funkcionalitātē pārbaudīt, vai nav uzradušās jaunas vai bijušās kļūdas.

Rezultātā darba autore secina, ka lielākā daļa testētāju darba pienākumi tiek definēti rakstiskā formā, tas ir gan projekta līgumos, gan testētāja profesijas aprakstā, kvalitātes sistēmā vai uzņēmuma iekšējos normatīvajos aktos, bet ar laiku šie dokumenti noveco, nav aktuāli, jo netiek pilnveidoti balstoties uz katra projekta pieredzi. Šo dokumentu papildināšana, atjaunošana prasa no uzņēmuma papildus resursus, tāpēc tādas darbības parasti netiek veiktas, attiecīgi jaunajiem testētājiem, ienākot uzņēmumā, liels resursu patēriņš tiek patērēts savas veicamo pienākumu metodes izstrādei. Tā kā liela daļa respondentu kopā ir atzīmējuši, ka darbojas pēc saviem ieskatiem un atbilstoši projekta vadītāja norādēm, darba autore secina, ka ir uzņēmumi, kuros pie testētāja pienākumiem pieiet samērā elastīgi, nodrošinot iespēju, veicamos darbus jebkurā brīdī mainīt atbilstoši situācijai un nepieciešamībai, saglabājot testētāja minimālo pienākumu kopu, kas būtu jāveic katra projekta ietvaros.



2.2-2 attēls Aptaujas 5. jautājums

Pēc respondentu atbildēm (skat. 2.2-2 attēls) var izšķirt, ka visbiežāk projektu ietvaros tiek izstrādāti 5 dokumenti gan atbilstoši testētāja profesijas aprakstiem, gan iekšējiem normatīvajiem aktiem, gan pašu testētāju vai projektu vadītāju iniciatīvas ietvaros: Testēšanas plāns, Testpiemēru specifikācija, Testēšanas žurnāls, Testa problēmas ziņojums un Testēšanas kopsavilkuma pārskats.

Viens respondents ir atzīmējis, ka projektos nepieciešams izstrādāt pilnu standartā *LVS 70:1996* minēto dokumentu kopu. Šis respondents ir atzīmējis, ka darbojas 1-3 cilvēkmēnešus lielos ūdenskrituma modeļa projektos, kur projektu vadītājs seko līdzi līgumā noteiktām prasībām. Šajā gadījumā ir vērts pieminēt, ka respondents nākošajā jautājumā par aptuveni patērēto laiku dokumentu izstrādei no visa testēšanas procesa ir atzīmējis 70%, kas norāda, ka šajā gadījumā lielākā daļa darba tiek veltīta dokumentu izstrādei nevis maksimāli daudz nepieciešamo testu izpildei.

Viens respondents, kas darbojas ieviešanas un izstrādāšanas, kā arī ārpakalpojumu projektos ar ūdenskrituma projekta pārvaldības metodi ir atzīmējis visus jautājumā uzskaitītos dokumentus, bet ar piebildi, ka atkarībā no projekta specifikas, daļa dokumentu var tikt pavienota vienā dokumentā, izmantojot uzņēmumā izveidotas sagataves. Laika patēriņš dokumentu sagatavošanai netiek minēts, bet respondents raksta, ka laika patēriņš ir atkarīgs no uzņēmumā izveidotās sagataves. Ja tās ir kvalitatīvi izstrādātas, tad mazāk laika nepieciešams patērēt dokumentācijas izstrādei. Šis respondents, papildus atzīmējot atbilžu variantu „Citi”, raksta, ka dokumentācijas veidus un nepieciešamos standartus nosaka klienta prasības un uzņēmuma kvalitātes sistēmā aprakstītais minimālais nepieciešamais dokumentu komplekts. Respondents piezīmē, ka standarts *LVS 70:1996* ir novecojis, bet atzīst, ka ir gadījumi, ka klients prasa dokumentāciju atbilstoši tam. Šādi gadījumi parasti ir sastopami lielos projektos.

Darba autore secina, ka veicamo darbu pienākumus, īpaši izstrādājamo dokumentāciju projekta ietvaros bieži ietekmē klienta prasības, tāpēc 2.3 nodaļā darba autore ir veikusi pētījumu par pēdējo 2011. – 2013. gadu posmā izsludinātajiem konkursiem un tajos minēto izstrādājamās dokumentācijas kopu.



2.2-3 attēls Aptaujas 6. jautājums

Puse testētāju 20-30% no projektā paredzētā testēšanas laika izmanto dokumentācijas sagatavošanai (skat. 2.2-3 attēls). 1/4 testētāju izmanto 40% testēšanai paredzētā laika dokumentācijas sagatavošanai. 1/5 testētāju izmanto 50% un vairāk no testēšanai paredzētā laika dokumentācijas sagatavošanai. Testētāji, kuri ir atzīmējuši 40% un vairāk iepriekšējā variantā ir atzīmējuši vismaz 4 dokumentus, kas tiek sagatavoti projektu ietvaros. Bet testētāji, kas ir atzīmējuši 20-30% iepriekšējā jautājumā ir atzīmējuši maksimāli 3 dokumentus, kas tiek sagatavoti projekta ietvaros, gandrīz visos gadījumos tas ir Testēšanas kopsavilkuma pārskats, kas tiek izstrādāts arī pārējos gadījumos, retāk Testpiemēru specififikācija, Testa problēmas ziņojums vai Testēšanas plāns.

Darba autore secina, ka, ja testētājs izstrādās minimālo dokumentācijas kopu no 3 dokumentiem, izmantojot esošas uzņēmuma sagataves, laika patēriņš šīm darbībām būs līdz 20% no visā projektā paredzētā testēšanas laika, nodrošinot lielāku laika periodu dažādāku testpiemēru izpildei un testēšanas metožu pielietošanai. Bet, ja testētājs izstrādā 4 un vairāk dokumentus līdz pat pilnai dokumentācijas kopai, tad tikai dokumentācijas izstrāde testētājam var aizņemt no 40-70% laika no visa testēšanai paredzētā laika, kas testētājam nodrošina tikai minimālās testu kopas izpildi. Šādi testētājs nevarēs būt pārliecināts, ka sistēma ir kvalitatīvi un efektīvi notestēta, toties dokumentācija noteikti būs kvalitatīvi izstrādāta. Šajā gadījumā darba autore vēlas vērst uzmanību, ka testētāja galvenais pienākums ir darbināt, testēt sistēmu, nevis izstrādāt dokumentāciju. Liela apjoma dokumentācijas izstrāde neliecina, ka sistēma ir tik pat kvalitatīvi izstrādāta un notestēta.



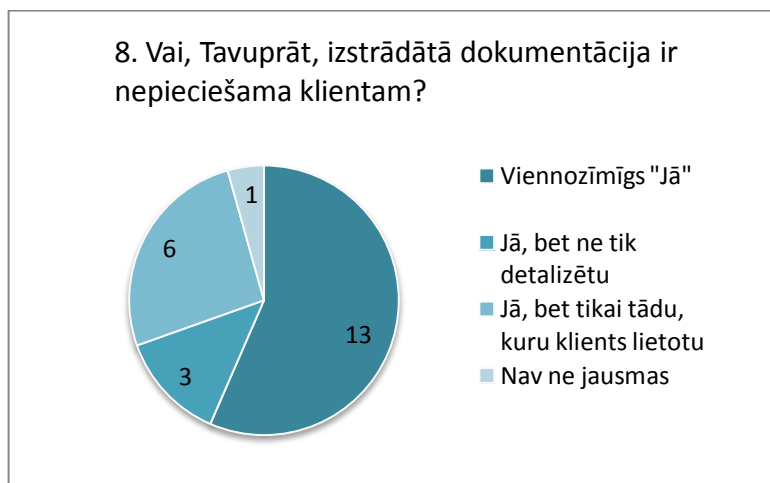
2.2-4 attēls Aptaujas 7. jautājums

2/3 respondentu ir atzīmējuši (skat. 2.2-4 attēls), ka testēšanas procesā izstrādātā dokumentācija tiek nodota klientam saskaņošanai un iekļauta nodevumā atbilstoši līgumam,

lielākoties komentāros kā nodevums tiek minēts testēšanas plāns un akcepttestu, testu scenāriju akts vai atskaites, kontrolsaraksti, problēmu pieteikumi. No šiem pašiem respondentiem 7 testētāji ir atzīmējuši, ka paralēli nododamajai dokumentācijai tiek izstrādāta arī iekšējā dokumentācija, kas ietver regresijas testpiemēru kopu, testēšanas scenārijus, testēšanas specifiku, testu piemēru prioritātes, testpiemēru validācijas pierādījumu materiāli no sākotnējām testēšanas stadijām. Kā redzams, klientam un iekšējai lietošanai tiek izstrādāta atsevišķa dokumentācija, kas arī palielina dokumentācijas izstrādei nepieciešamo laiku. Lietderīgāk būtu, ja iekšējai lietošanai un klientam varētu izstrādāt kopīgu dokumentāciju. Šie respondenti problēmu ziņojumus un pieteikumus apstrādā arī problēmu pieteikumu pārvaldības rīkos, piemēram, *JIRA*, *TestDirector*, *TestTrack*, *TFS* utt. Daļai no šiem rīkiem ir pieeja arī klientam. Viens respondents min, ka projektos, kur nodevuma kopa ir ļoti minimāla, tiek ievērotas arī uzņēmuma iekšējās procedūras, kas definē veidot dokumentāciju, kas paredzēta izstrādes / uzturēšanas fāzei, piemēram, regresijas testpiemēru kopa, tā parasti netiek nodota klientam saskaņošanai.

1/5 respondentu ir atzīmējuši, ka testēšanas procesā izstrādā dokumentāciju tikai iekšējām vajadzībām. Šie respondenti pie izstrādājamās dokumentācijas ir atzīmējuši ne vairāk kā 3 dokumentus no šiem – Testēšanas plāns, Testpiemēru specifiku, Testēšanas žurnāls, Testa problēmas ziņojums vai Testēšanas kopsavilkuma pārskats -, kā arī dokumentu izstrādes laiks nepārsniedz 30% no visa testēšanai paredzētā laika.

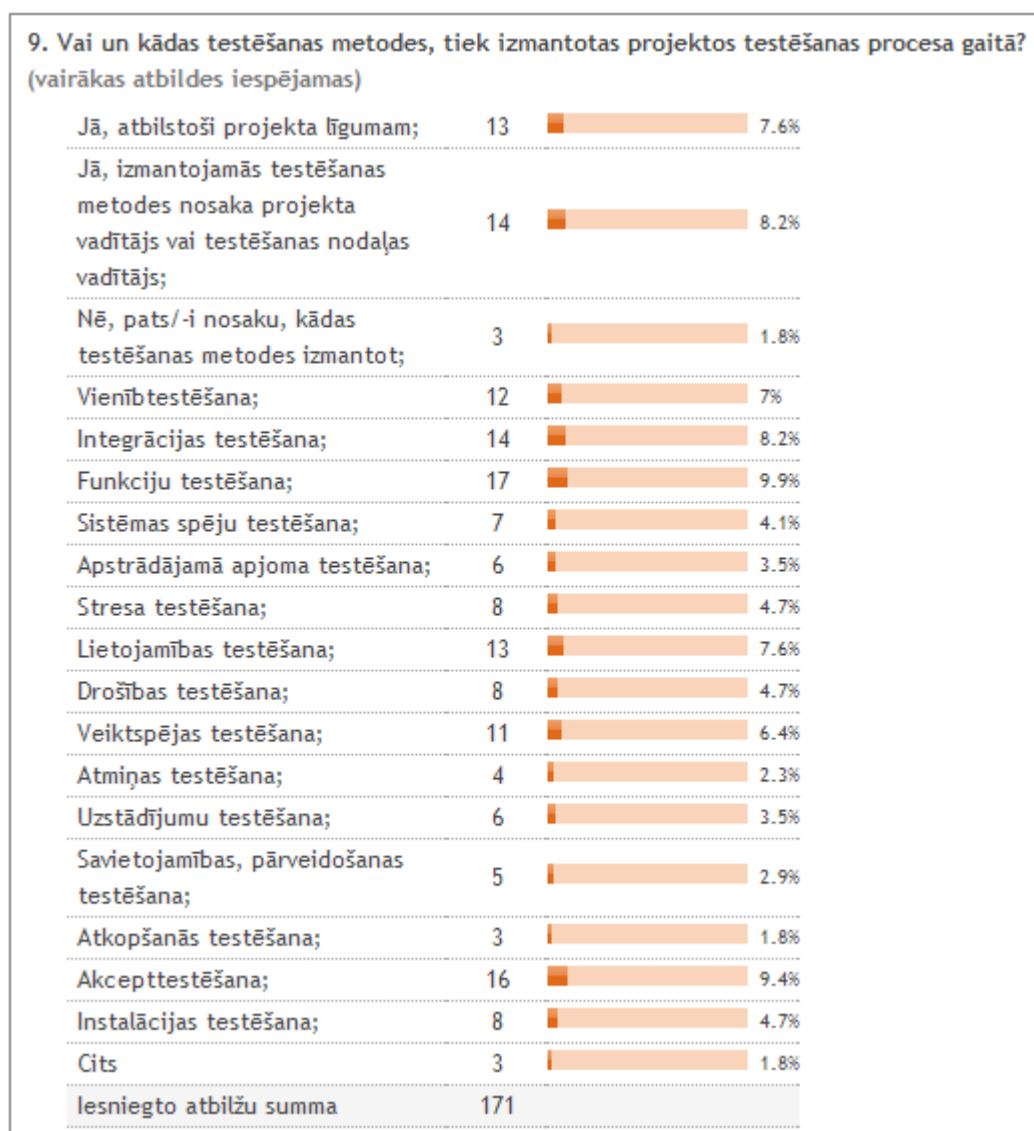
Darba autore secina, ka bez standartā *LVS 70:1996* uzskaitītās dokumentācijas mūsdienās tiek sagatavota un izstrādāta papildus jauna testēšanas dokumentācija, kā piemēram, akcepttestu akts, testpiemēru protokols, kā arī paralēli Testa problēmu ziņojumam vai Testēšanas žurnālam tiek uzturēts un izmantots kāds problēmu pieteikumu pārvaldības rīki. Darba autore uzskata, ka šajā gadījumā būtu lietderīgi pilnveidot un izmantot problēmu pieteikumu pārvaldības rīkus, lai iegūtu atbilstošas atskaites Testēšanas žurnālam, šo procesu darba autore iesaka automatizēt.



2.2-5 attēls Aptaujas 8. jautājums

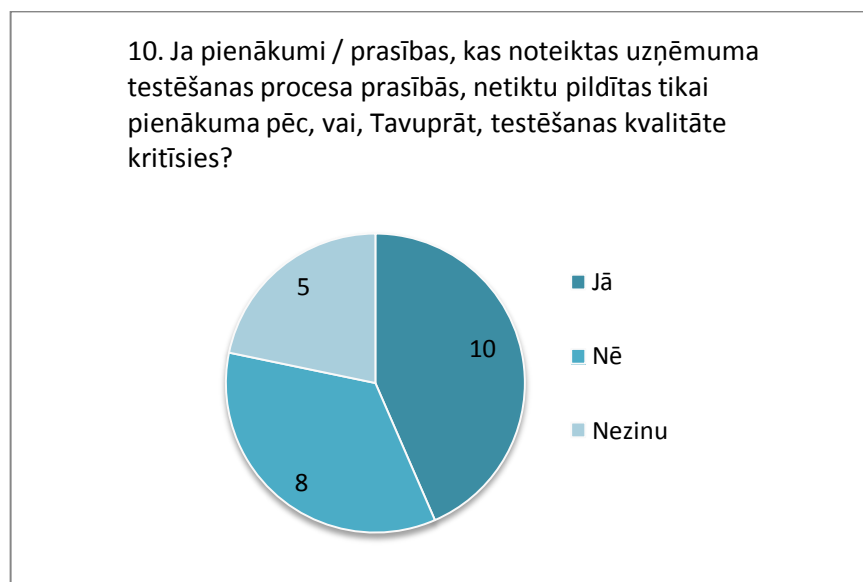
Visi respondenti (izņemot vienu) ir atbildējuši (skat. 2.2-5 attēls), ka klientam ir nepieciešama dokumentācija par testēšanu, tās rezultātiem, bet ar piebildēm, ka tai nav jābūt tik detalizētai kā parasti tiek sagatavota iekšējai lietošanai, jo klients šādu dokumentāciju nesaprot. ¼ respondentu piezīmē, ka klienta dokumentācijai ir jābūt lietojamai klienta vajadzībām un saprotamā valodā.

Darba autore secina, ka testēšanas posmā nepieciešams izstrādāt dokumentāciju klientam, bet tai nav jābūt tik tehniskai un detalizētai, kā tas ir nepieciešams projekta ietvaros. Dokumentam ir jābūt viegli uztveram un pārskatāmam, izceļot klientam būtiskākos jēdzienus. Darba autore uzskata, ka par dokumentu saturu ieteicams vienoties ar klientu pirms dokumentu izstrādes, pasniedzot klientam pilnu dokumenta sagatavi, kuru uzņēmums pats atjauno atbilstoši projektu vajadzībām, pieredzei un standartiem, lai klients pats norādītu, kuras dokumentu sadaļas tam ir būtiskākās, lai nebūtu jāizstrādā pilns dokuments kā rakstīts standartos.



2.2-6 attēls Aptaujas 9. jautājums

Jautājumā par izmantotākajām testēšanas metodēm (skat. 2.2-6 attēls) lielākā daļa respondentu ir atzīmējuši funkciju testēšanu un akcepttestēšanu, kā nākošās svarīgākās metodes tika atzīmētas integrācijas testēšana, lietojamības testēšana, vienībtestēšana un veikspējas testēšana. 14 respondenti ir atzīmējuši, ka projektā izmantojamās testēšanas metodes nosaka projekta vadītājs vai testēšanas nodaļas vadītājs, kā arī projektā izmantojamās testēšanas metodes tiek izmantotas atbilstoši projekta līgumam. Visretāk atzīmētas izmantotās testēšanas metodes ir atmiņas testēšana, atkopšanās testēšana, uzstādījumu testēšana un apstrādātā apjoma testēšana. Vidējs novērtējums kā projektos izmantojamām testēšanas metodēm ir sistēmas spēju testēšanai, stresa testēšanai, drošības testēšanai un instalācijas testēšanai. Viens respondents ir atzīmējis vērtību „Citi”, norādot, ka ir nepieciešams izmantot visas testēšanas metodes. Bet divi respondenti, atzīmējot vērtību „Citi”, ir pierakstījuši komentāru, ka testēšanas metožu izvēle ir atkarīga no projekta specifikas. Papildus šie respondenti ir atzīmējuši, ka savos projektos testēšanas metodes tiek izvēlētas atbilstoši projekta līgumam, ka arī tās nosaka projekta vadītājs vai testēšanas nodaļas vadītājs.



2.2-7 attēls Aptaujas 10. jautājums

Respondentiem domas daļās diezgan vienlīdzīgi par to, vai testēšanas kvalitāte kritīsies, ja uzņēmuma noteiktās testēšanas procesa prasības / pienākumi netiktu pildīti tikai pienākuma pēc. Šajā jautājumā (skat. 2.2-7 attēls) vērtīgi ir palasīt respondentu komentārus, kuros respondenti raksta, ka vadlīnijas ir nepieciešams ievērot, bet šīm vadlīnijām ir jābūt sakārtotām, patstāvīgi atjaunotām un balstītām uz testētāju pieredzi. Neievērojot minimālas uzņēmuma testētāju prasības, testēšanas kvalitāte kritīsies, jo pastāv risks konkrētai informācijas sistēmai nepamanīt būtiskas testēšanas metodes, izveidojot nepilnīgu sistēmas testēšanas pārklājumu. Šādas

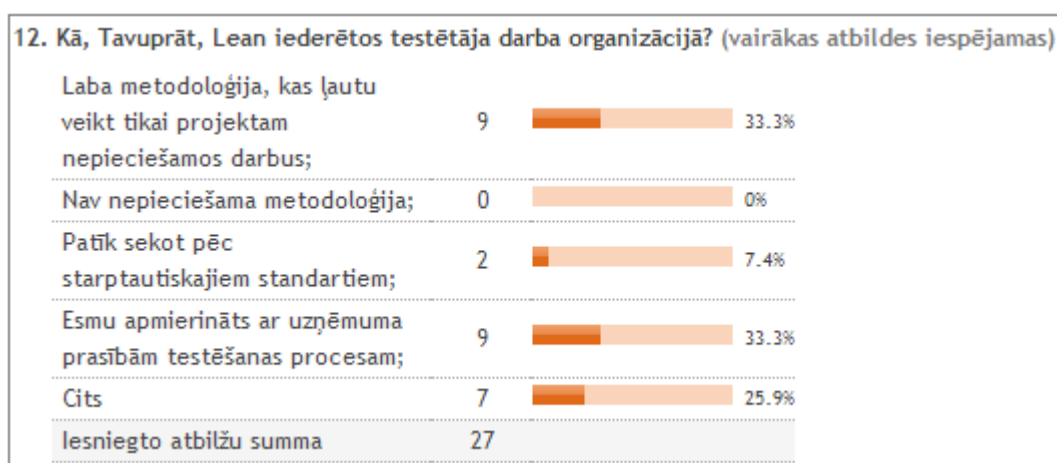
vadlīnijas ir nepieciešamas, jo mazāk pieredzējušie testētāji var nezināt, kas vēl bez parastas testu izpildes testētājam projektā ir jādara.

Visi respondenti saka, ka nevajag veikt tikai formālus pienākumus „ķeksīša” dēļ, rezultātā iegūstot tikai nevajadzīgi iztērētu laiku, bet ir jāveic projektam nepieciešamus pienākumus, kas nodrošinās klientu, projekta vadītāju un komandu ar informāciju, kas nodrošinās sistēmas kvalitatīvu darbību. Galvenais ir, lai testēšanas process būtu sakārtots, tad arī kvalitāte tiks nodrošināta.

2.2.2. Aptaujas atbilžu par *LEAN* metodoloģiju analīze

Aptaujas laikā respondentiem tika uzdoti jautājumi arī par *LEAN* metodoloģiju (turpmāk *LEAN*) ar mērķi, izzināt, vai šis domāšanas veids ir pazīstams, vai sabiedrībā esošie testētāji atbalsta *LEAN*, utt.

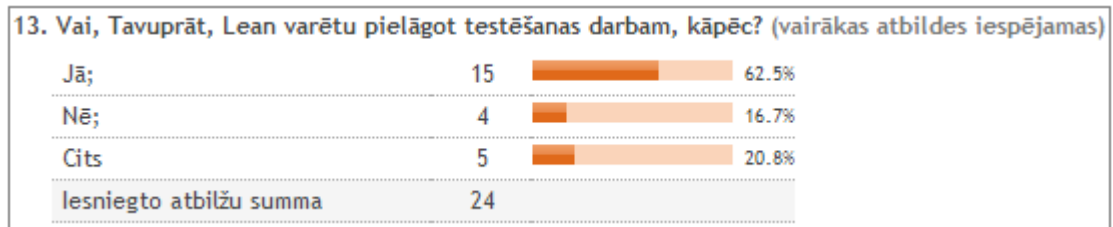
8 no respondentiem bija agrāk dzirdējuši, zinājuši par *LEAN*. 15 respondenti par *LEAN* nebija dzirdējuši. Viens no respondentiem ir atzīmējis, ka aptaujas laikā ir iepazinies ar *LEAN*, pateicoties aptaujā esošai informatīvai hipersaitei.



2.2-8 attēls Aptaujas 12. jautājums

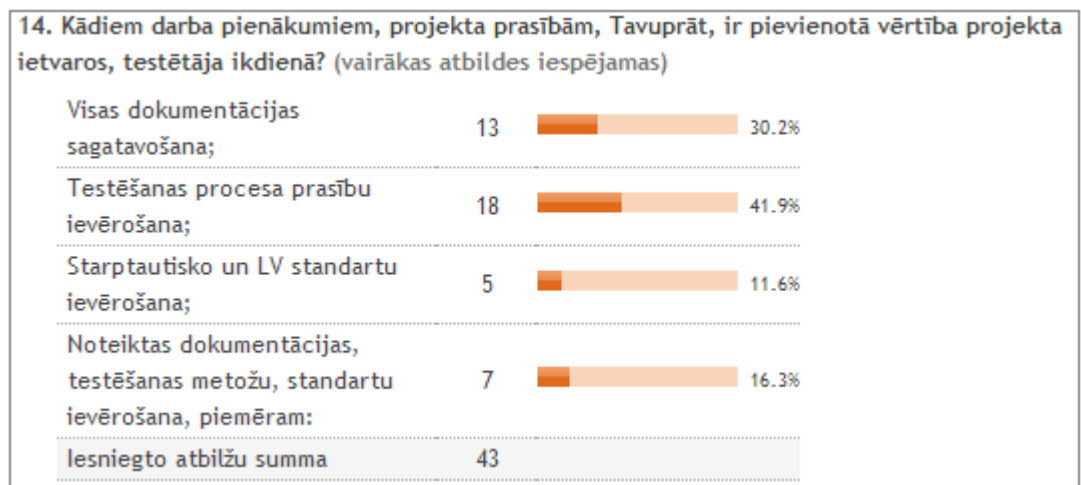
Liela daļa respondentu atzīst, ka *LEAN* ir laba metodoloģija (skat. 2.2-8 attēls), kas ļautu projektā veikt tikai nepieciešamos darbus. Arī palasot komentārus, kas aprakstīti zem atbilžu varianta „Cits”, tiek atzīmēts, ka, ja *LEAN* pamatā ir veikt tikai nepieciešamos darbus, kas veica ātrāku projekta izpildi nez lieku procesu veikšanas, tad tā varētu būt lietderīga. Vienam respondentam nav viedokļa, bet cits saka, ka projektos metodoloģija nedrīkst būt viena, katram projektam der sava. 9 respondenti atzīmē, ka ir apmierināti ar uzņēmumā esošajām prasībām testēšanas procesam, kā arī komentāros ir atzīmēts, ka šajos uzņēmumos ir savas vērtības, kuras zināmā mērā pārklājas ar *LEAN*, līdz ar to var teikt, ka šie uzņēmumi jau darbojas pēc *LEAN*. Divi respondenti atzīmē, ka *LEAN* metodoloģija, lai derētu testētāja darbam, varētu iekļaut tikai

svarīgāko no standartiem, lai nebūtu viss jāpilda „pēc grāmatas”. Neviens respondents nav atzīmējis, ka *LEAN* nav nepieciešama metodoloģija, bet tā ir tikai pareizi jāpielāgo testētāja darbam. 2 respondenti ir atzīmējuši, ka labprāt darbojas pēc zināmiem standartiem.



2.2-9 attēls Aptaujas 13. jautājums

3/5 respondentu atzīst, ka *LEAN* varētu pielāgot testētāja darbam (skat. 2.2-9 attēls). Viens respondents ir atzīmējis, ka arī noteikti ir nepieciešams *LEAN* pielāgot testētāja darbam, lai nodrošinātu sistēmas kvalitāti, bet tas nav izdarāms 100%, jo bieži darbus, kurus komandai ir jāveic, nosaka līgums ar klientu. 4 respondenti ir atzīmējuši, ka *LEAN* nav nepieciešams pielāgot testētāja darbam. Šie paši respondenti iepriekšējā jautājumā ir atzīmējuši, ka ir apmierināti ar uzņēmumā esošajām prasībām testētāja darbam. 3 respondenti ir atzīmējuši, ka nav kompetenti atbildēt uz šo jautājumu, jo nav iepazinušies ar *LEAN*.



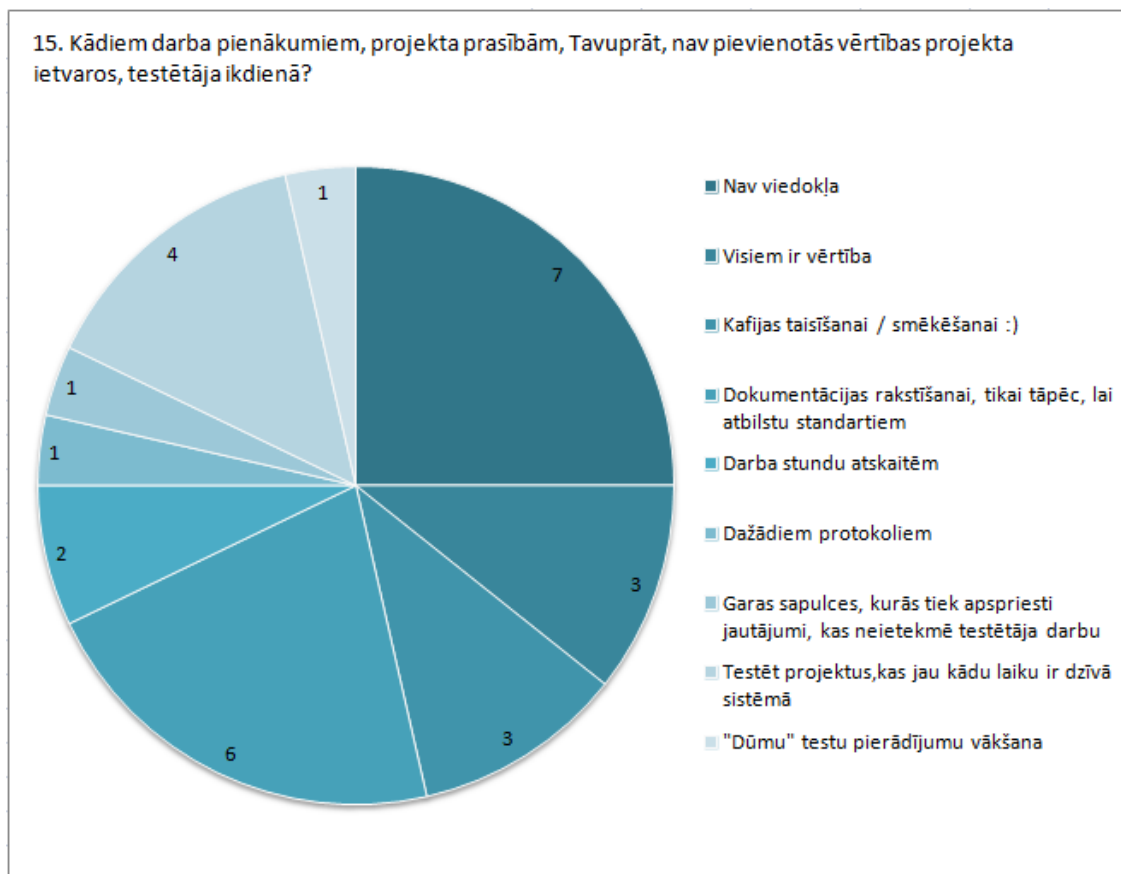
2.2-10 attēls Aptaujas 14. jautājums

5 respondenti nosaka, ka ir pilnīgi visiem testētāja darba pienākumiem, prasībām ir pievienotā vērtība projekta ietvaros (skat. 2.2-10 attēls). 4/5 respondentu atzīst, ka pievienotā vērtība ir uzņēmumā esoša testēšanas procesa prasību ievērošana, 8 no tiem atzīmēja arī visas dokumentācijas sagatavošanu. Tikai neliela daļa respondentu jeb 5 testētāji noteica, ka pievienotā vērtība ir:

- testēšanas protokoliem,
- testēšanas projekta plānošanai,

- kontrolesarakstiem,
- problēmu pieteikumu reģistrēšanai un apstrādei, izmantojot piemērotus rīkus,
- automātisko testu izpildei,
- sekošanai testēšanas plānam,
- visam, kas atbilst zem „Mācīties no kļūdām” un „Kāpēc mēs te kļūdījāmies?” (skat. nodaļu 2.4).

Viens respondents piebilst, ka, ja uzņēmums ir *ISO* sertificēts, tad uzņēmums pats apņemas sagatavot atbilstošu dokumentāciju.



2.2-11 attēls Aptaujas 15. jautājums

Diemžēl 1/3 respondentu nav viedokļa, kādiem testētāja darba pienākumiem, projekta prasībām nav pievienotās vērtības projekta ietvaros vai testētāja ikdienā (skat. 2.2-11 attēls). 3 respondenti uzskata, ka visiem darba pienākumiem vai projekta prasībām ir pievienotā vērtība. 6 respondenti atzīst, ka nav pievienotās vērtības dokumentācijas rakstīšanai, kas atbilst standartiem, jo standarti ir novecojuši un neatbilst vairs mūsdienu prasībām. Respondenti ir pieminējuši, ka pat, ja tiek sekots standartiem, tad šī dokumentācija tiek pārtaisīta, atstājot tikai standartos definētos nosaukumus un saturu. 4 respondenti uzskata, ka nav pievienotās vērtības jau ekspluatācijā esošu sistēmu testēšanai, jo gala lietotājs, strādājot ar sistēmu, jau veic neapzinātu

tās testēšanu, kā arī testētājs nespēj pārklāt visas sistēmā iespējās darbību kopas atbilstoši biznesa prasībām.

2.2-1 tabula

Aptaujas 16. jautājums

Respondents	Vienībtēšana	Integrācijas testēšana	Funkciju testēšana	Sistēmas spēju testēšana	Apstrādājamā apjoma testēšana	Stresa testēšana	Lietojamības testēšana	Drošības testēšana	Veiktspējas testēšana	Atmiņas testēšana	Uztādījumu testēšana	Savietojamības, pārveidošanas testēšana	Atkopšanās testēšana	Akcepttestēšana	Instalācijas testēšana
1										x			x		
2													x		
4	x			x		x	x				x			x	
5							x								
6		x				x	x	x	x			x		x	x
13			x				x	x							
15													x		
19					x	x						x	x		
KOPĀ:	1	1	1	1	1	3	4	2	1	1	1	2	4	2	1

8 respondenti atzīmēja noteiktas testēšanas metodes, kurām, viņuprāt, nav pievienotās vērtības jautājumā „16. Kādām testēšanas metodēm, Tavuprāt, nav pievienotās vērtības projektos?” (skat. 2.2-1 tabula). Pēc respondentu atbildēm redzams, ka visvairāk pievienotās vērtības nav lietojamības testēšanai, atkopšanās testēšanai, stresa testēšanai, drošības testēšanai, savietojamības, pārveidošanas testēšanai un akcepttestēšanai.

7 respondenti ir atzīmējuši, ka testēšanas metožu pievienotā vērtība ir atkarīga no sistēmas biznesa procesiem, tās arhitektūras un lietotāju skaita. 5 respondenti ir pateikuši, ka pievienotā vērtība ir visām testēšanas metodēm. Diemžēl 2 respondenti nepārzina latviskos nosaukumus testēšanas metodēm, tāpēc nespēja atbildēt uz šo jautājumu. Viens respondents ir atzīmējis, ka pievienotās vērtības nav protokolu sagatavošanai, kā arī minējis risinājumu, ka nepieciešams rīks, kas ģenerētu atskaites pēc izpildītajiem testiem.

Darba autore piekrīt, ka pievienotā vērtība ir visām testēšanas metodēm, bet to vērtības lielums tomēr ir atkarīgs no projekta specifikas un mērķa, sistēmas arhitektūras un lietotāju daudzuma. Bet tomēr ir iespējams laika gaitā, balstoties uz testētāju pieredzi izdalīt visbiežāk lietotās un obligāti izpildāmās testēšanas metodes, kuras būtu noteikti jāizmanto visos projektos.

2.2.3. Aptaujas rezultātu kopsavilkums

Apkopojot aptaujas rezultātus, tika iegūts testētāju viedoklis par darba pienākumiem un projektu prasībām, testēšanas metodēm un sagatavojamo dokumentāciju, kas pēc testētāju viedokļa ir vai nav ar pievienoto vērtību, par *LEAN* ieviešanu testētāja darbā.

Pēc aptaujas rezultātiem darba autore secina, ka testētāji lielākoties darbojas dažāda izmēra ieviešanas, izstrādes un uzturēšanas projektos, kas tiek organizēti pēc ūdenskrituma, V modeļa vai spējās programmatūras izstrādes metodoloģijas, bet lielai daļai projektu arī netiek pielietota noteikta projekta pārvaldības metode. Lielākajā daļā uzņēmumu ir iekšējie normatīvie akti, testētāju „rokasgrāmatas”, kā arī testētāja profesijas apraksts, bet šī informācija ir novecojusi, tāpēc ir daļa testētāju, kas darbojas pēc saviem ieskatiem vai arī seko tikai projekta līguma nosacījumiem.

Apkopojot aptaujas rezultātus par testēšanas procesā izstrādājamo dokumentācijas pievienoto vērtību projektam, darba autore secina, ka lielākā daļa testētāju izmanto apmēram 30-40% no visa testēšanas procesa dokumentācijas izstrādei. Kā arī anketas rezultāti apliecina, ka dokumentāciju ir nepieciešams sagatavot gan projektam, gan klientam, tikai klientam dokumentācijai nav jābūt tik detalizētai kā projekta vajadzībām, lai klients turpmāk to varētu gan saprast, gan izmantot. Darba autore uzskata, ka optimālais gadījums testēšanas procesa dokumentācijas izstrādei ir 20% no visa testēšanai paredzētā laika, tāpēc tika apkopots dokumentu saraksts, kas pēc testētāju viedokļa projektam sniedz pievienoto vērtību. Kā arī darba autore vēlas vērst uzmanību, ka uzņēmumos nepieciešams izstrādāt šo dokumentu sagataves, lai veicinātu vieglāku dokumentu aizpildīšanu un uzturēšanu. Anketas rezultātos kā visvairāk vērtību pievienojošie dokumenti tika atzīmēti Testēšanas plāns, Testpiemēru specifikācija, Testu problēmu ziņojums un Testēšanas kopsavilkuma pārskats, par kuriem tika teikts, ka tos nav obligāti jāraksta „pēc grāmatas”, bet ir nepieciešams iekļaut tikai klientam un projektam nepieciešamo informāciju.

Aptaujas rezultātos testētāji norādīja, ka mūsdienās visbiežāk tiek izmantotas un ir obligāti izpildāmas testēšanas metodes - vienībtestēšana, integrācijas testēšana, funkciju testēšana, veiktspējas testēšana un akcepttestēšana, bet kā neobligāti izpildāmās testēšanas metodes tika atzīmētas Stresa testēšana, Lietojamības testēšana, Drošības testēšana, savietojamības, pārveidošanas testēšana un atkopšanās testēšana. Vienībtestēšana un Funkciju testēšana aptauju rezultātu komentāros tika minētas arī kā šobrīd visbiežāk izmantotās testēšanas metodes, kuras ļauj visefektīvāk pārbaudīt tieši sistēmas vienumu un funkciju darbību atbilstoši sistēmas arhitektūras un sistēmas biznesa prasībām. Papildus lielākā daļa testētāju norādīja, ka ir nepieciešamas visas testēšanas metodes, bet tās ir jāpiemēro atbilstoši projekta specifikai un biznesa prasībām, arhitektūrai un lietotāju skaitam.

Aptaujas laikā lielākā daļa testētāju norādīja, ka vadlīnijas ir nepieciešamas testētāja darbam, bet tās patstāvīgi ir jāatjauno, jāuzlabo, kā arī jāizveido vai jāatjauno dokumentācijas sagataves, lai efektīvi un kvalitatīvi varētu strādāt jaunos un esošos projektos, kā arī apmācīt jaunos darbiniekus.

2.3. Atvērtu konkursu nolikumu analīze

Lai apzinātu un klasificētu klientam nepieciešamās, nododamās dokumentācijas apjomu, darba autore, nešķirojot projektu apjomus un klientus, meklēja internetā pieejamos atklāto konkursu projektu nolikumus, tehniskās specifikācijas un līgumu projektus par laika periodu 2011. – 2013. gads.

Kopā tika izskatīta 45 dažādu konkursu pieejamā dokumentācija - nolikumi, projektu līgumi un tehniskās specifikācijas, no kuriem tikai 27 konkursos tika prasīts izstrādāt dokumentāciju par testēšanu (skat. 2.3-1 tabula). 9 konkursu dokumentos tika minēts, ka dokumentācija ir jāizstrādā atbilstoši standartiem, šajos gadījumos atbilstoši *LVS 70:1996* standartam.

2.3-1 tabula

Atvērtu konkursu nolikumos minētās dokumentācijas kopsavilkums

	2011	2012	2013
Akcepttestu protokols / pārskats	1	2	1
Testēšanas plāns	2	6	2
Testpiemēri	2	5	1
Testēšanas scenāriji	2	6	2
Testēšanas rezultātu atskaite		4	
Testēšanas kopsavilkuma pārskats	3	5	1
Testpiemēru specifikācija		1	
Problēmziņojumu pārskats	1	1	
Akcepttestēšanas plāns		3	
Testēšanas žurnāls	1	2	1
Akcepttestēšanas scenāriji		3	1
Akcepttestēšanas akts		1	
Testēšanas protokols		3	3

Par 2011 gadu tika atrasti 4 dažādu valsts iestāžu konkursu nolikumi un izmantots 1 Valsts Ieņēmumu Dienesta konkursa nolikums, no kuriem tikai daži pieprasītie dokumenti atkārtojās. 2 konkursa nolikumos testēšanas dokumentācija netika prasīta. Valsts Ieņēmumu Dienesta

konkursiem katram gadam tika izmantots 1 nolikums, jo šobrīd tā ir vienīgā iestāde, kura visos konkursu nolikumos no testētāja pieprasa vienādu dokumentācijas kopu: Programmatūras testēšanas plāns (apraksts), Testēšanas scenāriji, Testpiemēri, Testēšanas žurnāls un Testēšanas (kopsavilkuma) pārskats.

Par 2012.gadu tika atrasti 14 dažādu iestāžu konkursu dokumenti, kuros testēšanas dokumentācija tiek pieprasīta dažādā sastāvā un 8 konkursu dokumentos analogiski 2011.gadam testēšanas dokumentācija netiek pieprasīta.

Par 2013.gadu tika atrasti 5 dažādu iestāžu konkursu dokumenti, kuros testēšanas dokumentācija tiek pieprasīta dažādā sastāvā un 3 konkursu dokumentos analogiski iepriekšējiem gadiem testēšanas dokumentācija netiek pieprasīta.

Kopā visos gados visvairāk tika pieprasīti trīs šādi dokumentu kopumi:

1. Testēšanas plāns, Testpiemēri, Testēšanas scenāriji, Testēšanas žurnāls un Testēšanas (kopsavilkuma) pārskats;
2. Testēšanas rezultātu atskaite, Akcepttestēšanas plāns un Akcepttestēšanas scenāriji;
3. Testēšanas protokols un Testēšanas scenāriji.

Darba autore, balstoties uz LVS70:1996 uzskaitīto testētājam izstrādājamo dokumentācijas kopumu, secina, ka klients no testētāja nesagaida tik lielu dokumentācijas apjomu kā definē LVS 70:1996, bet sagaida dokumentāciju, kas klientam palīdzētu veikt akcepttestēšanu.

Dažu konkursu dokumentos tika noteikts, ka testētājam noteikti ir jāveic veikspējas testus, regresijas testus un funkcionālos testus, to rezultātus aprakstot Testēšanas rezultātu atskaitēs vai Testēšanas protokolos.

Atlikušo 18 konkursu dokumentos tika minēts, ka Izpildītājam ir jāveic testēšana, jānodrošina testa, demonstrācijas un apmācību vides, bet netika prasīta testēšanas dokumentācija pat tik daudz kā testēšanas rezultātu atskaite vai akcepttestēšanas protokols. Šajos konkursos visplašāk tika aprakstītas sistēmas funkcionālās prasības. Tā kā konkursu dokumenti attēlo informāciju un darbības, par kurām klients ir gatavs maksāt, tad pēc dokumentācijas analīzes darba autore secina, ka Latvijā lielākoties konkursi tiek virzīti funkcionālo prasību realizācijai, un klienti paredz ietaupījumu uz testētāju darbu, attiecīgi uz sistēmas kvalitāti, kaut gan rezultātu tāpat pieprasa maksimāli kvalitatīvu.

Dažu konkursu dokumentos tika noteikts, ka izpildītājam ir jānodrošina pieejama problēmu pieteikumu sistēma problēmziņojumu dokumentēšanai un risināšanas gaitas atsekošanai.

Pēc konkursu dokumentācijas analīzes darba autore secina, ka 3/5 klientu nemaz budžetu neparedz kārtīgam testēšanas procesam, kas ietver gan testu izpildi, gan projekta specifikai atbilstošas dokumentācijas sagatavošanu. Diemžēl klients bieži tiek maldināts, ka testēšana ir tikai testu izpilde, kā arī šāda konkursu dokumentācija liek aizdomāties par to, ka klientam it kā

neinteresē kvalitatīva sistēma, kaut gan ikdienā nemaz tā nav. Bet, ja testētājs izmantos uzņēmumā definēto profesijas, kvalitātes sistēmas, iekšējo normatīvo aktu dokumentāciju un veiks nepieciešamo papildus darbību minimumu, lai dokumentētu testēšanas procesu, sistēmas kvalitāte manāmi uzlabosies un būs vieglāk atsekojams risināto problēmu kopums. Konkursu nolikumi, kuros netiek definēti obligāti izpildāmo dokumentu un testēšanas metožu apjoms, testētājam ļauj pašam brīvi izvēlēties projektam vislabāk un visprecīzāk veicamos darbus un pienākumus, bet šādos projektos no klienta puses nav nekādas kontroles par veiktajiem testēšanas darbiem un informācijas par sasniegtajiem rezultātiem, kas testētājam neuzliek pienākumu klientam atskaitīties par veiktajiem testēšanas darbiem, kā arī nenosaka, ka testēšana projektā vispār būtu jāveic. Darba autore uzskata, ka šobrīd esošo palīglīdzekļa sagatavi var izmantot arī klients, lai spētu saprast, vai projektā veiktais testēšanas process tiek pietiekami labi organizēts un tiek izpildītas visas projekta ietvaros nepieciešamās minimālās prasības sistēmas ērtākai uzturēšanai un kļūdu apstrādei.

2.4. Testēšanas process, izmantojot *LEAN* metodoloģiju

Bakalaura darba izstrādes laikā autore testētāja darba pienākumus iedalīja divos veidos atbilstoši *LEAN* domāšanas veidam un balstoties uz aptaujas rezultātiem: testēšanas procesam vērtību pievienojošie un nevērtīgie darba pienākumi. Rezultātā tika konstatēts, ka lielākā daļa testēšanas metožu un procesu, ieskaitot visas LVS 70:1996 standartā minētās dokumentācijas sagatavošanu, (pat dažos uzņēmumos visi) ir nepieciešams veikt testētāja pienākumos neatkarīgi no sistēmas arhitektūras un lietotāju skaita, bet nav nepieciešams sagatavot tik lielu apjomu dokumentu, kā nosaka standarti. Balstoties uz aptaujas un konkursu nolikumu rezultātiem autore secina, ka testētāji savos darba pienākumos vairāk vēlas vērst uzmanību uz kļūdu atrašanu un izzināšanu, nevis dokumentācijas sagatavošanu.

Izmantojot *LEAN* metodoloģijas pamatprincipus un aptaujas rezultātus, darba autore testēšanas metodes un izstrādājamo testēšanas dokumentāciju ir izdalījusi grupās, kas atbilst vērtību identificēšanai pēc *LEAN*. Pēc *LEAN* metodoloģijas izveidotāju vārdiem, programmatūru izstrādē *LEAN* ieteicams izmantot kopā ar iteratīvu programmatūras izstrādes metodoloģiju, regulāri analizējot iegūtos rezultātus un atbildot uz šādiem jautājumiem:

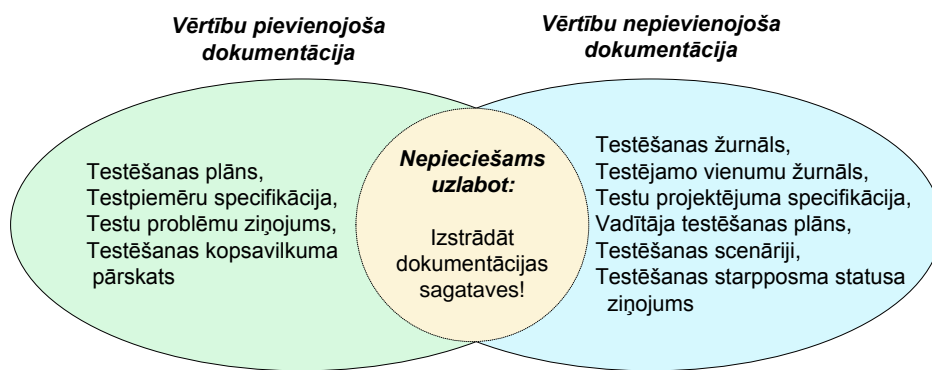
- vai veiktās darbības ir vai nav bijušas projektam vērtību pievienojošas,
- vai veiktajā iterācijā turpmāk ir nepieciešams noņemt, uzlabot vai optimizēt kādas darbības.

Izmantojot iteratīvu pieeju, pilna testēšanas procesa plūsmas izmantošanas reize var tikt pielietota gan noteikta sistēmas moduļa vai funkcionalitātes testēšanai, gan visas sistēmas

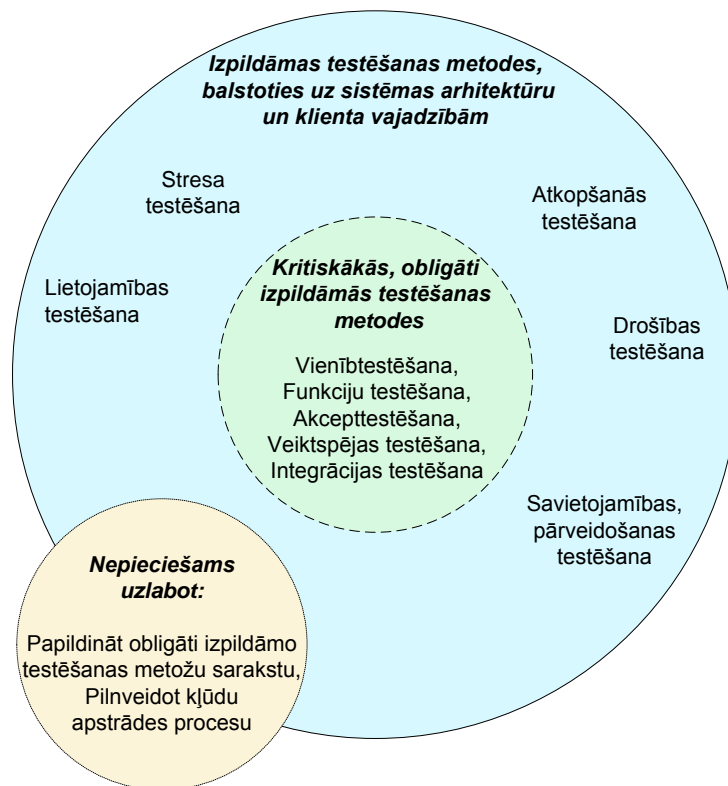
kopumā testēšanai. Testēšanas procesa plūsmas pielietošana testētāju neierobežo ar testējamo sistēmas apgabalu apjomu.

Lai vieglāk novērtētu veikto darbību efektivitāti, testētājs, projekta vadītājs vai klients var izmantot testēšanas procesa kvalitātes kritērijus (skat. nodaļu 1.3) un testējamo sistēmu kvalitātes metrikas (skat. nodaļu 1.4).

Diagrammā (skat. 2.4-1 attēls) vērtību pievienojoša dokumentācija ir dokumentācija, kas atbilst minimālās testēšanas procesā izstrādājamās dokumentācijas apjomam, lai testēšana process būtu maksimāli kvalitatīvi dokumentēts, lai testēšanas dokumentāciju varētu izmantot un saprast klients, lai testētājs maksimāli daudz no testēšanas procesa veltītu testu izpildei, nevis dokumentācijas izstrādei. Lai dokumentācijas sagatavošana notiktu efektīvāk, darba autore iesaka sagatavot un uzlabot atbilstošo dokumentu sagataves.

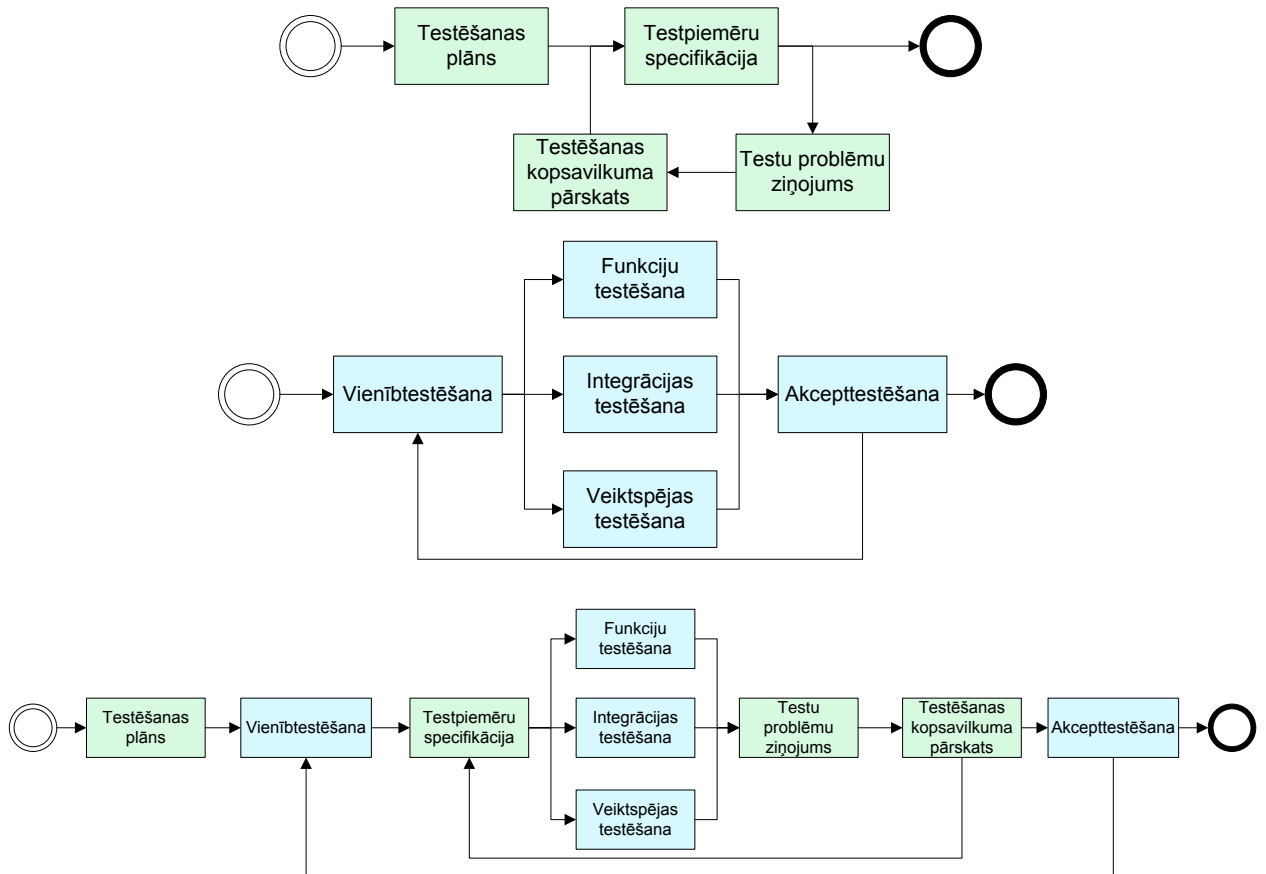


Vērtību pievienojošas testēšanas metodes



2.4-1 attēls Vērtību identificēšana, balstoties uz LEAN

Balstoties uz *LEAN* nākošo pamatprincipu- Vērtību plūsmas sagatave, darba autore ir izveidojusi sagataves līmenī vērtību pievienojošo, veicamo pasākumu plūsmu gan dokumentācijas sagatavošanai, gan testēšanas veikšanai atsevišķi, gan apvienojot (skat. 2.4-2 attēls) vienai iterācijai, diagramma var tikt izmantota atkārtoti katras iterācijas ietvaros, lai katrā jaunā iterācijā uzlabotu katru plūsmas posmu līdz idealizācijai.

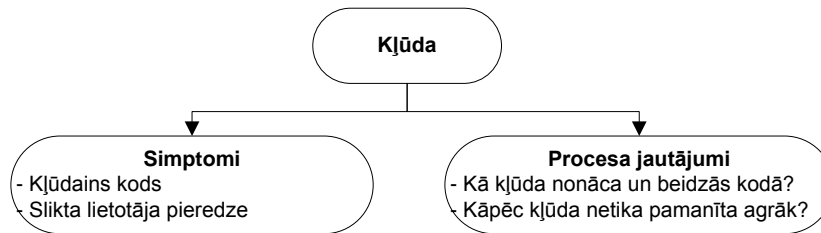


2.4-2 attēls Vērtību plūsmas sagatave, izmantojot *LEAN*

Šobrīd *LEAN* pamatprincipi - Vērtību plūsmas izstrāde un Vērtību plūsmas idealizācija – netiek realizēti, jo šobrīd darba autorei nav projekta, kurā varētu pielietot jaunās zināšanas, bet, balstoties uz pieredzi no projektiem, kuros šobrīd darbojas, darba autore secina, ka, balsoties uz *LEAN* pamatprincipu – Vērtību plūsmas uzlabošana – ir nepieciešams uzlabot:

- testēšanas procesā sagatavojamās dokumentācijas sagataves,
- testēšanas procesā pielietoto testēšanas metožu dažādību,
- kļūdu apstrādes, izpētes un labošanas procesu.

Lai uzlabotu kļūdu apstrādes, izpētes un labošanas procesu, autore iesaka kļūdu atrašanu sistēmā izdalīt divu veidu problēmās - simptomi un jautājumi, lai izdalītu projektam vērtību pievienojošus procesus. (skat. 2.4-3 attēls) Lai nodrošinātu, ka sistēma darbojas kvalitatīvi un ir ilgi dzīvojoša, nepieciešams risināt abas problēmas [12].

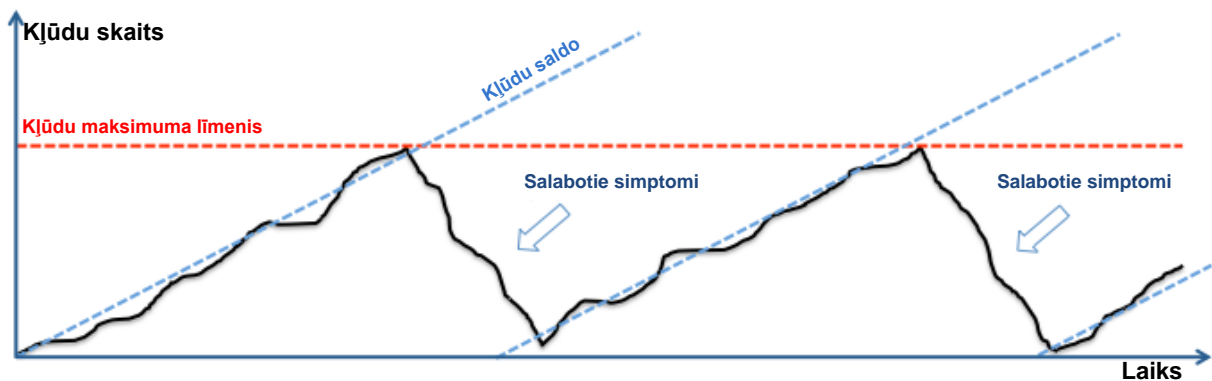


2.4-3 attēls Kļūdas simptomi un procesa jautājumi

2.4.1. Kļūdas simptomi

Simptomi ir tā kļūdas daļa, kas ir redzama cilvēka acīm, piemēram, negaidīta, nesaprotama sistēmas uzvedība, kas noved pie sliktas lietotāja pieredzes. Bet no izstrādes puses tas ir kļūdainais kods vai slikti nospecificēta / uzprojektēta funkcionalitāte [12].

Izlabojot kļūdaino kodu vai funkcionalitātes projektējumu, tiks atrisināti tikai kļūdas simptomi. Tā tiek darīts daudzos uzņēmumos, kur tiek novērots laika gaitā kļūdu skaita pieaugums līdzīgi zāģa robiem (skat. 2.4-4 attēls). Ja tiek pielietota tikai simptomu labošanas metode, tad izstrādes komanda diezgan bieži nonāk līdz kļūdu maksimuma līmenim jeb roba virsotnei, atkārtojoties šim procesam.



2.4-4 attēls Laika gaitā kļūdu skaita pieaugums, novēršot tikai simptomus

Kļūdu skaits šajā metodē pieaug, ja radīto kļūdu skaits ir lielāks par jau konstatēto un laboto kļūdu skaitu.

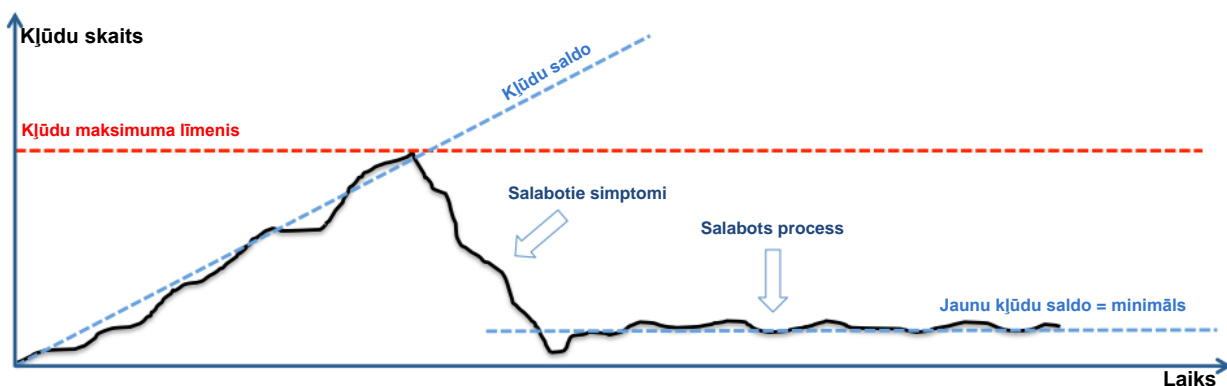
2.4.2. Kļūdu procesa jautājumi

Lai nodrošinātu kvalitatīvi un ilgi dzīvojošu sistēmu, tikai ar simptomu labošanu nav pietiekoši. Lai nodrošinātu kvalitatīvu un ilgi dzīvojošu sistēmu, nepieciešams pārlicināties, ka kļūdas neatkārtosies, izveidojot nebeidzamu esošo un jauno kļūdu labošanas virkni un risinot 2.4-3 attēls minētos problēmu jautājumus (skat. 2.4-1 tabula) [12].

Problēmu jautājumu risinājums

	Jautājums	Atbilde
1.	Kā kļūda nonāca un beidzās kodā?	Izstrādātājs apzināti netaisa kļūdas, tomēr nepieciešams vērst uzmanību, ko izstrādes grupa var pasākt, lai izvairītos no kļūdām nākotnē, piemēram, meklēt kļūdas rašanās cēloni kodā, nevis rast apkārtceļus, lai apietu noteiktu situāciju.
2.	Kāpēc kļūda netika pamanīta agrāk?	Piemēram, kļūda pēc visiem testēšanas procesiem tomēr ir nonākusi līdz produkcijas videi. Ko izstrādes grupa var darīt savādāk, lai efektīvāk un laicīgāk atrastu kļūdas izstrādes procesā?

Lai risinātu problēmas procesa jautājumus, nepieciešams projekta izstrādes grupai un testētājam mainīt procesu, kā tiek veikts darbs. Procesā ir jāiekļauj tikai vērtību pievienojoši uzdevumi un jāizņem visi laiku prasoši, bet nevērtīgi procesi. Starp pievienojamajiem uzdevumiem jāiekļauj kļūdas saknes meklēšana un risināšana, un kļūdainu situāciju samazināšana izstrādē esošai sistēmai, mācoties no savām kļūdām. Turpmākajā projekta izpildē, papildinot un turpinot sistēmas izstrādi, var parādīties kļūdas, bet to būs mazāk, jo, pirmkārt, tās nebūs vecās kļūdas, kuras ik pa laikam uzrodas kā „zāģa zobi”, un, otrkārt, tā kā vecajām kļūdām tiks atrisināta „saknes” problēma, tad jaunās kļūdas neradīsies veco kļūdu rezultātā. (skat. 2.4-5 attēls) Rezultātā būs lielāka iespējamība, ka jaunās kļūdas tiks laicīgi pamanītas, un mazāka iespējamība, ka jaunās kļūdas nonāks līdz sistēmas gala lietotājam.



2.4-5 attēls Laika gaitā kļūdu skaita pieaugums, novēršot arī kļūdu iemeslus

Šāda procesa ieviešana sistēmas izstrādē un uzturēšanā nebūs viegla, kā arī rezultāts netiks sasniegts vienā rāvienā. Procesu ieviešana ir vērtīga, neskatoties uz to, ka ieviešana var prasīt

vairākas iterācijas un sākumā „zāga zobi” nemainīsies, bet laika periodā tie kļūs mazāki un nolīdzināsies.

Šādas problēmu risināšanas metodes nekvalitatīvai sistēmai, kas nonāk līdz gala lietotājam, mērķis ir iegūt augstu kvalitāti un labu ātrumu sistēmas izstrādē, izmantojot kļūdu cēloņu analīzi.

3. RISINĀJUMS

Šajā nodaļā ir aprakstīts darbā sasniegtais problēmas risinājums – testētāja palīglīdzekļa sagatave atbilstoši *LEAN* metodoloģijas pamatprincipiem, aptaujas rezultātiem, klientu izsludinātajiem konkursu nolikumiem par laika periodu 2011. – 2013.gads. Palīglīdzekļa sagatave ietver testēšanas procesa aprakstu atbilstoši *LEAN* (skat. nodaļu 2.4) un testētāja minimālo izstrādājamo dokumentācijas (skat. nodaļu 3.1) un kritiskāko pielietojamo testēšanas metožu (skat. nodaļu 3.2) kopu. Testētāja palīglīdzekļa sagatavi ir paredzēts izmantot:

1. Testētāja darba organizācijai testēšanas projektos, nosakot
 - a. Testēšanas procesā veicamo pasākumu plūsmu pēc iteratīvas pieejas, regulāri analizējot un salīdzinot iegūtos rezultātus ar testēšanas procesa kvalitātes kritērijiem un testējamo sistēmu kvalitātes metrikām, un uzlabojot vai optimizējot plūsmas posmus,
 - b. minimālās izstrādājamās dokumentācijas kopas saturu atbilstoši klienta un iekšēji projektam būtiskākajām vajadzībām un LVS 70:1996 standartam,
 - c. kritiskāko pielietojamo testēšanas metožu kopu, kura testētājam obligāti ir jāizskata, vai vajag sistēmas testēšanas procesā, lai varētu pateikt, ka lielākā daļa sistēmas ir notestēta atbilstoši klienta un iekšējām projekta vajadzībām un klienta izvirzītam biznesa procesam, kuru sistēmai ir jāuztur un jārealizē.
2. Sistēmas testēšanas procesa kvalitātes novērtēšanai, pēc šādām pazīmēm:
 - a. Testētājs testēšanas procesu veic atbilstoši darba ietvaros izstrādātajai palīglīdzekļa sagatavei;
 - b. Sagatavotās dokumentācijas apjoms (ieteicams minimālais dokumentācijas kopums);
 - c. Dokumentācijas sagatavošanā patērētais laiks pret visu testēšanas procesu (ieteicams zem 30%);
 - d. Izmantoto testēšanas metožu dažādība (vismaz kritiski pielietojamo testēšanas metožu kopums);
 - e. Testēšanas procesa uzlabošana katras iterācijas ietvaros atbilstoši *LEAN*.

Turpmāk testētāja palīglīdzekļa sagatavi ir paredzēts papildināt ar:

- padziļinātu projektu raksturlielumu izpēti, lai noteiktu kuras testēšanas metodes ir jāpielieto tieši noteiktam projektam, balstoties uz projekta arhitektūru, klienta vajadzībām un sistēmā izstrādāto biznesa procesu;
- piedāvājumu minimālajā dokumentācijas kopā iekļauto dokumentu sagatavēm, lai testētājs ērtāk un efektīvāk varētu veikt arī testēšanas darba dokumentēšanu, izmantojot jau iepriekš sagatavotas dokumentācijas sagataves;
- brīvi pieejamu testēšanas procesa palīgriku izpēti, kas sniedz projekta grupai ērti lietojamas un klientam saprotamas atskaites / protokolus par veiktajiem un plānotajiem testēšanas darbiem.

Pētot testēšanu teorijā, no dažādiem literatūras avotiem tika izveidots ieskats testēšanas procesa kvalitātes kritērijos (skat. nodaļu 1.3) un testējamo sistēmu kvalitātes metrikās (skat. nodaļu 1.4), kā šobrīd tiek piedāvāts mērīt programmatūras testēšanas procesu kvalitāti. Darbā aprakstītajā risinājumā visam testēšanas procesam autore mudina novērtēt nevis testēšanas procesa efektivitāti, kas, piemēram, iekļauj salīdzinājumu, cik kļūdas tiek atrastas uz 100 koda rindiņām, bet gan novērtēt pašu testēšanas procesu - pielietoto testēšanas metožu dažādību un atbilstību sistēmas arhitektūrai, klienta vajadzībām un biznesa prasībām, izstrādātās dokumentācijas apjomu un nepieciešamību iekšēji projektam un klientam, pārbaudot cik tas ir kontrolēts, pilnīgs un drošs, vai ir iespējams un nepieciešams to uzlabot. Rezultātā testētājs, projekta vadītājs un klients varēs novērtēt vai testēšanas kvalitāte un efektivitāte ir slikta vai pasliktinājusies pēc noteiktu darbu pievienošanas vai izmaiņšanas iterācijas ietvaros. Testēšanas procesa kvalitātes kritērijus un testējamo sistēmu kvalitātes metrikas autore rosina izmantot tieši testēšanas iteratīvo procesu novērtēšanai, veicot procesa uzlabojumus un optimizāciju, lai salīdzinātu vai veiktie testēšanas procesa plūsmas posmi projektam sniedz ar vien lielāku vai mazāku pievienoto vērtību, līdz testēšanas procesa plūsma ietver tikai vērtību pievienojošus posmus un ir maksimāli pakārtota *LEAN* metodoloģijai, nodrošinot pēc iespējas mazāku dokumentācijas sagatavošanas darbu un pēc iespējas lielāku testu izpildes darbu.

3.1. Minimālais dokumentācijas kopums

Darba izpētes gaitā kā visvairāk vērtību projektā pievienojošie sagatavojamie testēšanas dokumenti tika atzīmēti:

- Testēšanas plāns (nepieciešams gan klientam, gan projektam iekšēji), jo tas nosaka projekta darbības sfēru, pārvaldības pieeju, paredzētos resursus un testēšanas pasākumu kalendāro plānu, kā arī identificē testējamus vienumus, kurus jātestē, testēšanas uzdevumus un riskus, kas saistīti ar plānu;

- Testpiemēru specifikācija (nepieciešams projektam iekšēji, klientam nepieciešams atkarībā no projekta prasībām), jo tā definē veicamos testpiemērus, ievaddatus un izvaddatus, ļaujot pārbaudīt sistēmai nodrošināmos biznesa procesus un akcepttestus;
- Testu problēmu ziņojums (nepieciešams gan klientam, gan projektam iekšēji), jo tas ietver jebkuru notikumu, kas gadījies testēšanas laikā, ja tas prasa pētīšanu. Vissvarīgāk dokumentā ir iekļaut neatrisinātās problēmas, kā arī visām problēmām ir jābūt prioritizētām pēc vienošanās ar klientu, lai ziņojumu saņēmējs spētu pilnvērtīgi izmantot iegūto informāciju;
- Testēšanas kopsavilkuma pārskats (nepieciešams gan klientam, gan projektam iekšēji), jo tas ir plānoto testēšanas darbību rezultātu kopsavilkums, kā arī novērtējums, balstoties uz minētajiem rezultātiem.

Par šo dokumentāciju testētāji aptaujas rezultātos min, ka to nav obligāti jāraksta „pēc grāmatas” t.i. pēc standarta *LVS 70:1996* (skat. nodaļu *1.5.1*), bet ir nepieciešams iekļaut tikai klientam un projektam nepieciešamo informāciju atbilstošā detalizācijas līmenī.

Testēšanas plānā (skat. nodaļu *1.5.1.1*) atbilstoši standartam *LVS 70:1996* noteikti ir jāiekļauj tāda informācija, kā

1. Testēšanas procesa pieeja un kalendārais plāns, pieminot arī testēšanas nodevumus, testēšanas vides vajadzības un atbilstību produkcijas videi,
2. Testēšanas vienumi / funkcijas un to novērtēšanas kritēriji, uzskaitot galvenos testēšanas vienumus un funkcijas tabulas formā un pretī norādot novērtēšanas kritērijus, piemēram, testējamais vienums ir noteikts funkcionalitātes apgabals vai un novērtēšanas kritērijs ir prasību dokuments, pēc kura tiek veikta testēšana,
3. Risks un neveiksmju sekas, kas noved pie iespējas nepamanīt kļūdas vai neiekļauties testēšanas procesam atvēlētajā projekta laikā, piemēram,
 - a. pietrūkstoša pieredze par biznesa procesiem, testēšanas metodēm,
 - b. testēšanas procesam atvēlēts maz laika,
 - c. problēmas ar testa vidi,
 - d. slikti sagatavota projekta un sistēmas dokumentācija, uz kuru balstīt testpiemēru sagatavošanu,
 - e. smagas sistēmas kļūdas, kas neļauj turpināt testēšanas procesu,
 - f. kļūdas, kuru salabošana rada jaunas kļūdas iepriekš testētos apgabalos.

Risikiem nepieciešams iedalīt atbildīgos un noteikt veicamos darbus risku iestāšanās gadījumos, lai varētu ātri un operatīvi novērst tos.

4. Apstiprinājumi, izstrādājot jaunu testēšanas plānu, kas iekļauj visu personu uzvārdus un amatus, kurām šis plāns jāapstiprina, kā arī parakstīšanas vietu un datumu.

Testpiemēru specififikācijā (skat. nodaļu *1.5.1.2*) atbilstoši standartam *LVS 70:1996* obligāti jāiekļauj veicamās darbības testpiemēra izpildei pa soļiem, iekļaujot informāciju par ievaddatiem un sagaidāmo rezultātu.

Testa problēmu ziņojumā (skat. nodaļu *1.5.1.5*) atbilstoši standartam *LVS 70:1996* nepieciešams informāciju strukturēt šādi:

1. Atkārtotā scenārijs,
2. Ievaddati,
3. Izvaddati,
4. Testēšanas vides raksturojums.
5. Sagaidāmais rezultāts.

Testēšanas kopsavilkuma pārskatā (skat. nodaļu *1.5.1.4*) atbilstoši standartam *LVS 70:1996* nepieciešams iekļaut šādu informāciju:

1. Rezultātu kopsavilkums, identificējot visas atrisinātās un neatrisinātās problēmas,
2. Novērtējums, balstoties uz testa vienumu, funkciju novērtēšanas kritērijiem un riskiem,
3. Novirzes no testēšanas plāna, norādot to cēloņus un riskus, kas iestājās,
4. Apstiprinājumi, kas iekļauj visu personu uzvārdus un amatus, kurām šis pārskats jāapstiprina, kā arī parakstīšanas vietu un datumu,.

3.2. Kritiskāko pielietojamo testēšanas metožu kopums

Darba izpētes gaitā kā visvairāk vērtību projektā pievienojošās testēšanas metodes tika atzīmētas:

- Vienībtestēšana (skat. nodaļu *1.2.3*),
- Integrācijas testēšana (skat. nodaļu *1.2.4*),
- Funkciju testēšana (skat. nodaļu *1.2.2*),
- Veiktspējas testēšana (skat. nodaļu *1.2.9*),
- Akcepttestēšana (skat. nodaļu *1.2.14*),

bet darba autore uzskata, ka ir nepieciešams izvērtēt, kuras testēšanas metodes ir nepieciešams veikt sistēmas testēšanas procesā, balstoties uz sistēmas specifiku, jo atņemot katru no testēšanas metodēm, mazinās testēšanas kvalitāte, ko var nodrošināt sistēmai. Tāpēc darba autore uzskata, ka ir nepieciešams samazināt testēšanas procesā izstrādājamās dokumentācijas

apjomu atbilstoši mūsdienu klienta un iekšējām projekta vajadzībām (skat. nodaļu 3.1), lai vairāk laika varētu testētājs veltīt tieši testu izpildei.

Vienībtestēšana un Funkciju testēšana aptauju rezultātu komentāros tika minētas kā šobrīd visbiežāk izmantotās testēšanas metodes. Funkcionālā testēšana pārlicinās, ka sistēma izpilda izvirzītās prasības, salīdzinot iegūto rezultātu ar sagaidāmo rezultātu, bet vienībtestēšana salīdzina testējamo vienumu ar tai noteiktajām prasībām, kas specificētas PPS. Šīs testēšanas metodes ir jāiekļauj minimālo pielietojamo testēšanas metožu kopumā.

Akcepttestēšana šobrīd tiek visbiežāk jautāta no klienta puses konkursu nolikumos, kuru veic pats klients, bet ir projekti, kur klientam tiek parādīta akcepttestēšanas prezentācija, kas neatbilst akcepttestēšanas metodes mērķim. Šī testēšanas metode ir jāiekļauj minimālo pielietojamo testēšanas metožu kopumā.

Veiktspējas testēšana ir nepieciešama projektos, kur klients ir paredzējis, ka sistēmu lietos liels apjoms lietotāju, un ir būtiska sistēmas ātrdarbība noteiktu darbību veikšanā lielas noslodzes apstākļos, kas ir raksturīgs projektos ar liela apjoma datu bāzēm, pakešapstrādes uzdevumiem, reāla laika sistēmām. Projektos ar maziem datu un lietotāju apjomiem veiktspējas testi nav nepieciešami. Veiktspējas testi nav būtiski arī projektos, kas ir paredzēti izklaidei, piemēram, www.draugiem.lv vienlaicīgi dienas laikā izmanto apmēram trīsdesmit viens tūkstotis lietotāju, bet veiktspēja sistēmā nav būtiska, jo sistēmas sasniedzamais rezultāts neietekmē lietotāju veselības stāvokli, drošību, naudas plūsmu, nav nepieciešams apstrādāt norēķinus, nav nepieciešams apstrādāt lielu apjomu dokumentu. Veiktspējas testi ir nepieciešami sistēmām, kas var nest briesmas dzīvībai vai milzīgus zaudējumus, piemēram, internetbanka.

Integrācijas testēšana ir nepieciešama projektos, kur sistēmai ir paredzēts integrēties ar citām klienta rīcībā esošām un biznesa prasību realizācijai nepieciešamās sistēmām, iegūt informāciju no tām vai nosūtīt informāciju uz tām, atbilstoši apstrādāt un saistīt to. Sistēmām, kurām nav paredzēts sadarboties ar citām klienta sistēmām integrācijas testēšana nav nepieciešama, bet sistēmām, kas, piemēram, iegūst adreses no Adrešu reģistra ir nepieciešama integrācijas testēšana, lai pārbaudītu vai visas reģistrētās adreses tiek nosūtītas uz klienta sistēmu un to turpmākā apstrāde sistēmā nav kļūdaina.

REZULTĀTI UN DISKUSIJA

Darba gaitā autore iepazīnās ar vizizplatītākajām ieviešanas, izstrādes un uzturēšanas projektos izmantotajām testēšanas metodēm, testētājam izstrādājamo dokumentāciju atbilstoši Latvijas standartiem. Darba autorei bija pieeja testēšanas darbā izmantojamiem starptautiskiem standartiem kā arī *ISTQB* sertifikāta apmācību materiāliem, kas gan definē testēšanas procesu, sistēmas kvalitāti, gan testēšanas procesā izstrādājamo dokumentāciju, un *LEAN* metodoloģijas literatūras avotiem. Izpētes rezultātā darba autore ieguva priekšstatu par testētāja darbā veicamo pienākumu un prasību apjomu neatkarīgi no projekta lieluma, klienta vajadzībām un vēlmēm vai citiem raksturlielumiem.

Lai darba sasniegtais rezultāts atbilstu mūsdienai Latvijā esošai situācijai, darba autore sagatavoja aptauju Latvijā esošo uzņēmumu testētājiem (skat. nodaļu 2.2). Aptaujas mērķis bija apkopot testētāju pieredzi par projektos visbiežāk nepieciešamajām testēšanas metodēm un izstrādājamo dokumentāciju un viedokli par testēšanas procesa optimizācijas nepieciešamību atbilstoši *LEAN* metodoloģijai. Apkopojot aptaujas rezultātus, darba autore ieguva, ka testēšanas procesu ir iespējams un pat ieteicams pielāgot *LEAN* metodoloģijai, testētāja veicamajos pienākumos iekļaujot tikai klientam un iekšēji projektam nepieciešamos pasākumus projekta veiksmīgai izpildei un pieredzes uzkrāšanai turpmākajiem projektiem. Aptaujas rezultātos (skat. nodaļu 2.2) kā visbiežāk un vēlamākās testēšanas metodes tika norādītas vienībttestēšana, integrācijas testēšana, funkciju testēšana, veiktspējas testēšana un akcepttestēšana, bet kā visnevērtīgākās testēšanas metodes tika atzīmētas stresa testēšana, lietojamības testēšana, drošības testēšana, savietojamības, pārveidošanas testēšana un atkopšanās testēšana. Aptaujas rezultātos darba autore izzināja, ka dokumentācijas sagatavošana testēšanas procesā vairumā aizņemt 30 – 40% no visa testēšanai atvēlētā laika. Bet darba autore uzskata, ka pilnīgi visa testēšanas procesa dokumentēšana nav nepieciešama, jo darba autore uzskata, ka sistēmas kvalitātei svarīgāk ir veikt pēc iespējas vairāk dažādus testus, kas atbilst sistēmas specifikai, un dokumentēt tikai problēmsituācijas, lai uzkrātu pieredzi un veiksmīgāk veiktu testus turpmākiem līdzīgiem projektiem. Kā arī, ja testi tiek veikti un dokumentēti ar noteiktiem rīkiem, piemēram, TFS, kas ir paredzēti sistēmas darba organizācijai, tad no šādiem rīkiem arī var iegūt gatavas atskaites, kas neprasa papildus dokumentēšanu. Pēc aptaujas rezultātiem, kas balstīti uz testētāju pieredzi un viedokli, darba autore secina, ka testētājam obligāti izstrādājami dokumenti ir testēšanas plāns, testpiemēru specifikācija, testa problēmas ziņojums un testēšanas kopsavilkuma

pārskats, nosakot, ka tos nav obligāti jāraksta pilnībā pēc standartiem jeb „pēc grāmatas”, bet ir nepieciešams iekļaut tikai klientam un projektam nepieciešamo informāciju.

Darba gaitā autore veica pētījumu iepriekšējo gadu sistēmu ieviešanas un izstrādes konkursu nolikumiem, apkopojot informāciju par klientu vajadzībām pret projektā izstrādājamo dokumentāciju un obligāti izpildāmajām testēšanas metodēm, kas visos konkursu nolikumos ir minēta kā akcepttestēšana. Izpētes rezultātā tika iegūts, ka klients konkursu nolikumos visbiežāk pie testēšanas nodevumiem no jau zināmās dokumentācijas iekļauj testēšanas plānu un testēšanas kopsavilkuma pārskatu, bet starp visbiežāk iekļauto parādās arī jauna izstrādājamā dokumentācija, kas nav atrunāta standartos – testpiemēri jeb testēšanas scenāriji, kas pēc uzbūves ir līdzīgi testpiemēru specifikācijai, un testēšanas rezultātu atskaite, kas pēc uzbūves ir līdzīga testēšanas kopsavilkuma pārskatam. Papildus izpētes rezultātos tika iegūta pavisam jaunas dokumentācijas kopa, kas tiek balstīta uz klienta vajadzību – akcepttestēšana. Šie dokumenti ir akcepttestu protokols / pārskats, akcepttestēšanas plāns, akcepttestēšanas scenāriji un akcepttestēšanas akts.

Darba gaitā autore sagatavoja testēšanas procesa plūsmas sagataves aprakstu balstoties uz aptaujas un konkursu nolikumu izpētes rezultātiem un *LEAN* metodoloģijas pamatprincipiem. Testēšanas procesa plūsmas sagatave ietver šobrīd darba gaitā identificētās projektam vērtību pievienojošas darbības gan no testēšanas procesā sagatavojamās dokumentācijas, gan no pielietojamām testēšanas metodēm, sakārtojot plūsmas posmus izpildes secībā. Izpētes rezultātos autore norāda, ka testēšanas procesa plūsma var tikt pielietota gan noteikta sistēmas moduļa funkcionalitātes testēšanai, gan visas sistēmas kopumā testēšanai, izmantojot iteratīvu testēšanas pieeju, lai katrā iterācijā varētu novērtēt veikto darbību lietderību projektam un nepieciešamības gadījumā veikt to optimizāciju, balstoties uz vērtēšanas rezultātiem.

Pēc šiem rezultātiem darba autore secina, ka mūsdienās testētājam noderētu kopīga, viegli pieejama un ērti lietojama rokasgrāmata ar veicamajiem pasākumiem veiksmīga projekta izpildei, balstoties uz klientu vajadzībām un testētāju pieredzi. Tāpēc darba rezultātā darba autore ir apkopojusi minimālo izstrādājamās dokumentācijas kopu un kritiskāko pielietojamo testēšanas metožu kopu, balstoties uz aptaujas un konkursu nolikumu izpētes rezultātiem, kā arī sagatavojusi testēšanas procesa plūsmas sagatavi, balstoties uz *LEAN* metodoloģijas pamatprincipiem, jo darba autore uzskata, ka sistēmai kvalitāti var nodrošināt ar pēc iespējas vairāk izpildītiem testiem, nevis maksimāli daudz sagatavotu dokumentāciju. Minimālais dokumentu kopums (skat. nodaļu 3.1) ietver testēšanas plāna, testpiemēru specifikācijas, testa problēmu ziņojuma un testēšanas kopsavilkuma pārskata rediģētu versiju balstoties uz *LVS 70:1996*, atstājot dokumentācijas saturā nodaļas, kuras pēc darba autores viedokļa ir nepieciešams iekļaut, lai nodrošinātu saprotamu informāciju klientam un paliekošu pieredzes

aprakstu sistēmas uzturēšanai un jaunu sistēmu ieviešanas un izstrādes projektiem. Kritiskāko pielietojamo testēšanas metožu kopums (skat. nodaļu 3.2) iekļauj testēšanas metodes, kuras testētāji ir norādījuši, kā visbiežāk izmantotās testēšanas metodes savā ikdienas darbā. Darba autore ir nosaukusi nodaļu par „*Kritiskāko* pielietojamo testēšanas metožu kopums”, jo darba autore uzskata, ka, ja sistēmas testēšanas procesā tiks mazināts pielietojamo testēšanas metožu kopums, mazināsies arī testēšanas kvalitāte, ko projekta grupa var nodrošināt sistēmai.

SECINĀJUMI

Par galvenajiem darba sasniegumiem autore uzskata:

- testēšanas procesa plūsmas sagataves izstrāde, izmantojot iteratīvu pieeju sistēmas testēšanai, balstoties uz *LEAN* metodoloģijas pamatprincipiem,
- minimālās dokumentācijas kopas sagatavošana, iekļaujot tikai tās *LVS 70:1996* nodaļas, kas ir atbilstošas klienta un iekšējām projekta vajadzībām, un *LEAN* metodoloģijai,
- kritiskās projektā pielietojamo testēšanas metožu kopas sagatavošana, atbilstoši testētāju pieredzei un viedoklim, kas noteikti ir jāizpilda visos projektos, lai nodrošinātu sistēmas kvalitāti.

Autore uzskata, ka darba mērķis ir sasniegts, un uzdevumi ir izpildīti, jo darba autore ir iepazinusies gan ar Latvijā esošiem standartiem, kas apraksta testēšanas procesu, gan ar starptautiskiem standartiem, kas apraksta testēšanas procesu, gan ar metodoloģiskās informācijas avotiem, kas paredzēti testētāju sertificēšanai, zināšanu papildināšanai un dažādu procesu uzlabošanai. Darba autore ir veikusi esošo standartu analīzi, salīdzinot aprakstītos testēšanas procesus ar Latvijā strādājošu testētāju pieredzi. Autore ir veikusi Latvijā esošo testēšanas speciālistu aptauju par testēšanas procesiem to ikdienas pienākumos un projektos. Darba rezultātā autore ir izstrādājusi palīglīdzekļa sagatavi testēšanas darbu kvalitātes un efektivitātes novērtēšanai. Darba autore uzskata, ka šāds palīglīdzeklis ir nepieciešams, jo izpētes rezultātā secināts, ka šobrīd pieejamie Latvijas standarti ir novecojuši un neatbilst klientu vajadzībām un vēlmēm, projektu pārvaldības metodēm un iekšējām projektu vajadzībām.

Iespējamie turpmākie darbi un neatrisinātās problēmas ir:

- veikt padziļinātu projektu raksturlielumu izpēti, lai noteiktu kuras testēšanas metodes ir jāpielieto tieši noteiktam projektam, balstoties uz projekta arhitektūru, klienta vajadzībām un sistēmā izstrādāto biznesa procesu;
- papildināt esošo palīglīdzekļa sagatavi ar piedāvājumu minimālajā dokumentācijas kopā iekļauto dokumentu sagatavēm, lai testētājs ērtāk un efektīvāk varētu veikt arī testēšanas darba dokumentēšanu;
- veikt pētījumu kā strādājot pēc izstrādātās palīglīdzekļa sagataves, mainās testēšanas procesa kvalitātes kritēriji (skat. nodaļu 1.3) un testējamo sistēmu kvalitātes metrikas (skat. nodaļu 1.4).

- izpētīt brīvi pieejamus testēšanas procesa palīgriekus, kas sniedz projekta grupai ērti lietojamas un klientam saprotamas atskaites / protokolus par veiktajiem un plānotajiem testēšanas darbiem.

Darba izstrādes gaitā autore iepazinās ar dažādām testēšanas metodēm un izstrādājamo dokumentāciju. Iegūtās zināšanas darba autore paredz izmantot darba vajadzībām, lai optimizētu savā projektā strādājošo testētāju darbu, un nodrošinātu vieglāku sistēmas kvalitātes novērtēšanu.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. **Dave Nave** *How To Compare Six Sigma, Lean and the Theory of Constraints* [e-grāmata]
Pieejams: <http://www.lean.org/Admin/KM/documents/76dc2bfb-33cd-4ef2-bcc8-792c5b4ef6a6-ASQStoryonQualitySigmaAndLean.pdf>
2. *LEAN* [tiešsaiste] , pieejams: <http://www.lean.org/whatslean/>
3. **James P. Womack, Daniel T. Jones** *Lean Thinking: Banish Waste and Wealth in Your Corporation* : New York - Free Press, 2003, [e-grāmata], pieejams :
http://books.google.lv/books/about/Lean_Thinking.html?id=2eWHaAyiNrgC&redir_esc=y
4. *ISO/IEC 12207:1995 Information technology - Software life cycle processes*, p 57, [e-grāmata]
Pieejams : <http://www.abelia.com/docs/12207cpt.pdf>
5. *J-Std-016-1995 EIA/IEEE Interim Standard for Information Technology - Software Life Cycle Processes - Software Development Acquirer-Supplier Agreement*, p 232, [e-grāmata]
Pieejams: http://www.abelia.com/docs/122_016.pdf
6. *1465-1998 IEEE (12119:1998 ISO/IEC) Information Technology - Software Packages - Quality Requirements and Testing*, p 22.
7. *LVS 70:1996 Informācijas tehnoloģija - Programminženierija - Programmatūras testēšanas dokumentācija*. Latvijas Nacionālais standartizācijas un metroloģijas centrs, 1996.
8. *LVS 73:1996 Informācijas tehnoloģija - Programminženierija - Programmatūras vienībtestēšana*. Latvijas Nacionālais standartizācijas un metroloģijas centrs, 1996.
9. *ANSI/IEEE Std 829-1998 IEEE Standard for Software Test Documentation*, [e-grāmata],
pieejams: <http://faculty.ksu.edu.sa/mohamedbatouche/SWE%20434/IEEE%20Std%20829%20-%201998.pdf>
10. *IEEE 1008-1987 IEEE Standard for Software Unit Testing*, p 28, [e-grāmata] pieejams:
<https://s3.amazonaws.com/akitaonrails/files/std1008-1987.pdf>
11. *ISO 9000:2001 Quality management systems -- Fundamentals and vocabulary* [tiešsaiste],
pieejams: http://www.ehow.com/facts_6954950_iso-9000_2001_.html
12. **Jan Grape** *Every bug means two problems* [tiešsaiste]
Pieejams: <http://blog.crisp.se/2013/04/08/jangrape/every-bug-means-two-problems>
13. *LZA Terminoloģijas komisija* [tiešsaiste], pieejams: <http://termini.lza.lv>
14. **Thomas Muller, Debra Friedenber, ISTQB WG Foundation Level ISTQB** “*Certified Tester*”
Foundation Level Syllabus, 2011.
15. *ISO/IEC 9126-1:2001 Software engineering -- Product quality*, 2001, [prezentācija]
Pieejams: http://is.uni-paderborn.de/fileadmin/Informatik/AG-Engels/Lehre/SS08/SQA/Folien/Chapter_3-Standards_Komplett.pdf

16. **Mārtiņš Gills** *Programmatūras testēšana un trasējamība* Rīga 2005
Pieejams: www3.acadlib.lv/greydoc/Gilla_disertacija/MGills_disertacija.doc
17. **Cem Kaner, Jack Falk, Hung Q. Nquyen** *Testing Computer Software, 2nd Edition*, Canada 1999, pieejams: <http://hfs1.duytan.edu.vn/upload/ebooks/3989.pdf>
18. **Andreas Spillner, Tilo Linz, Hans Scheafer** *Software Testing Foundations: A Study Guide for the Certified Tester Exam (Rockynook Computing), 3rd Edition*, Santa Barbara 2011, pieejams: <http://books.google.lv/books?id=cA9t91JTrsMC&printsec=frontcover&hl=lv#v=onepage&q&f=false>
19. **Boris Beizer** *Software Testing Techniques, 2nd Edition*, Dreamtech Press, 2003.
20. **Boris Beizer** *Black-Box Testing: Techniques for Functional Testing of Software and Systems*, Canada 1995.
21. *ISTQB International Software Testing Qualifications Board* [tiešsaiste],
Pieejams: <http://www.istqb.org/>

PIELIKUMI

1. Pielikums. Aptauja „Programmatūras testēšanas procesu kvalitātes novērtēšana”

Labdien, mani sauc Laila Gutke, esmu 4. kursa studente Latvijas Universitātes Datorikas Fakultātes Datorzinātņu bakalaura programmā. Šobrīd kursa ietvaros izstrādāju bakalaura darbu par tēmu „Programmatūras testēšanas procesu kvalitātes novērtēšana”. Bakalaura darbā viens no uzdevumiem ir aptaujas formā apkopot un analizēt Latvijas un starptautiskos IT uzņēmumos strādājošu testētāju pārstāvju viedokļus par testēšanas procesa norisi projektos un pildāmajiem pienākumiem.

Lai sasniegtu bakalaura darbā izvirzīto mērķi, lūgums atbildēt uz zemāk esošajiem jautājumiem.

Visos jautājumos var atzīmēt vairākas atbildes.

1. Kādos projektu veidos Tu piedalies?
 - a. Ieviešanas un izstrādes;
 - b. Uzturēšanas;
 - c. Testēšanas, kā ārpakalpojums citiem uzņēmumiem;
 - d. Cits _____

2. Cik lielos projektos Tu piedalies?
 - a. 1 – 3 c/m;
 - b. 4 – 7 c/m;
 - c. 7 – 12 c/m;
 - d. 1 – 3 gadi;
 - e. Cits _____

3. Pēc kādas projektu pārvaldības metodes tiek vadīti projekti, kuros Tu darbojies?
 - a. Agile;
 - b. Ūdenskritums;
 - c. SCRUM;
 - d. Nenoteikts;
 - e. Cits _____

4. Vai uzņēmumā / struktūrvienībā / projektā ir noteiktas kādas prasības, kā jānotiek testēšanas procesam?
- a. Jā, aprakstītas iekšējos normatīvos;
 - b. Jā, aprakstītas testētāja profesijas aprakstā;
 - c. Jā, projekta vadītājs seko projekta līgumā noteiktām prasībām;
 - d. Jā, uzņēmums testētājam piedāvā “rokasgrāmatu” par testēšanas procesa norisi;
 - e. Nē;
 - f. Darbojos pēc saviem ieskatiem;
 - g. Strādāju atbilstoši projekta vadītāja norādēm;
 - h. Cits _____
5. Vai un kādu dokumentāciju parasti nepieciešams sagatavot testēšanas procesa gaitā?
- a. Jā, visu atbilstoši LVS70:1996;
 - b. Testēšanas plāns;
 - c. Testu projektējuma specifikācija;
 - d. Testpiemēru specifikācija;
 - e. Testēšanas procedūras specifikācija;
 - f. Testējamā vienuma pavadzīme;
 - g. Testēšanas žurnāls;
 - h. Testa problēmas ziņojums;
 - i. Testēšanas kopsavilkuma pārskats;
 - j. Ne, nav nepieciešams;
 - k. Cits _____
6. Cik aptuveni laiku no visa testēšanas procesa aizņem dokumentācijas sagatavošana?
- _____
7. Vai izstrādātā testēšanas dokumentācija tiek nodota saskaņošanai klientam, vai tiek izstrādāta iekšējām vajadzībām? Piemēram?
- _____
8. Vai, Tavuprāt, izstrādātā dokumentācija ir nepieciešama klientam?
- _____
9. Vai un kādas testēšanas metodes tiek izmantotas projektos testēšanas procesa gaitā?

- a. Jā, atbilstoši projekta līgumam;
- b. Jā, izmantojamās testēšanas metodes nosaka projekta vadītājs vai testēšanas nodaļas vadītājs;
- c. Nē, pats/-i nosaku, kādas testēšanas metodes izmantot;
- d. Vienībtestēšana;
- e. Integrācijas testēšana;
- f. Funkciju testēšana;
- g. Sistēmas spēju testēšana;
- h. Apstrādājamā apjoma testēšana;
- i. Stresa testēšana;
- j. Lietojamības testēšana;
- k. Drošības testēšana;
- l. Veiktspējas testēšana;
- m. Atmiņas testēšana;
- n. Uzstādījumu testēšana;
- o. Savietojamības, pārveidošanas testēšana;
- p. Atkopšanās testēšana;
- q. Akcepttestēšana;
- r. Instalācijas testēšana;
- s. Cits _____

10. Ja pienākumi / prasības, kas noteiktas uzņēmuma testēšanas procesa prasībās, netiktu pildītas tikai pienākuma pēc, vai, Tavuprāt, testēšanas kvalitāte kritīsies? Kāpēc?

11. Bakalaura darbā kā viens no objektiem tiek apskatīts “*LEAN* domāšanas” veids (turpmāk *LEAN*), kas saka, ka rezultātu iegūšanas procesā nepieciešams darbības kārtot secīgos soļos, izmantojot tikai soļus ar pievienotu vērtību. Pievienotā vērtība ir darba pienākumu un veikto darbu tiešais labums projektam, klientam, projekta komandai, bet kura projekts, Tavuprāt, ir neveiksmīgs, nepilnvērtīgs. Procesā nedrīkst izmantot izšķērdīgus, nevērtīgus soļus, lai rezultāta iegūšanas procesā pēc iespējas mazāk patērētu laiku un resursus papildus “birokrātiskām” darbībām. Bakalaura darbu ir paredzēts virzīt atbilstoši šim domāšanas veidam. Vai Jūs zināt, vai esat dzirdējuši, kas ir *LEAN*? Sīkāk var skatīt: <http://www.lean.org/WhatsLean/>

- a. Jā;
- b. Nē;
- c. Cits _____

12. Kā, Tavuprāt, LEAN iederētos testētāja darba organizācijā?
- Labā metodoloģija, kas ļautu veikt tikai projektam nepieciešamos darbus;
 - Nav nepieciešama metodoloģija;
 - Patīk sekot pēc starptautiskajiem standartiem;
 - Esmu apmierināts ar uzņēmuma prasībām testēšanas procesam;
 - Cits _____
13. Vai, Tavuprāt, LEAN varētu pielāgot testēšanas darbam, kāpēc?
- Jā;
 - Nē;
 - Cits variants _____
14. Kādiem darba pienākumiem, projekta prasībām, Tavuprāt, ir pievienotā vērtība projekta ietvaros, testētāja ikdienā?
- Visas dokumentācijas sagatavošana;
 - Testēšanas procesa prasību ievērošana;
 - Starptautisko un LV standartu ievērošana;
 - Noteiktas dokumentācijas, testēšanas metožu, standartu ievērošana, piemēram:

15. Kādiem darba pienākumiem, projekta prasībām, Tavuprāt, nav pievienotā vērtība projekta ietvaros, testētāja ikdienā?
- Dokumentācijas sagatavošana,
 - Noteiktu testēšanas metožu izmantošana,
 - Noteiktu standartu ievērošana,
 - Cits variants: _____
16. Kādām testēšanas metodēm, Tavuprāt, nav pievienotās vērtības projektos?
- Vienībtēšana;
 - Integrācijas testēšana;
 - Funkciju testēšana;
 - Sistēmas spēju testēšana;
 - Apstrādājamā apjoma testēšana;
 - Stresa testēšana;
 - Lietojamības testēšana;
 - Drošības testēšana;
 - Veiktspējas testēšana;

- j. Atmiņas testēšana;
- k. Uzstādījumu testēšana;
- l. Savietojamības, pārveidošanas testēšana;
- m. Atkopšanās testēšana;
- n. Akcepttestēšana;
- o. Instalācijas testēšana;
- p. Cits _____

2. Pielikums. Aptaujas „Programmatūras testēšanas procesu kvalitātes novērtēšana” starprezultāti

Respondentu statistika:

Respondentu skaits	23
Vidējais vecums	28 gadi

Rezultātu kopsavilkums:

1. Kādos projektu veidos Tu piedalies? (vairākas atbildes iespējamās)

Ieviešanas un izstrādes;	22	52.4%
Uzturēšanas;	14	33.3%
Testēšanas, kā ārpalpojums citiem uzņēmumiem;	6	14.3%
Cits	0	0%
Iesniegto atbilžu summa	42	

2. Cik lielos projektos Tu piedalies? (vairākas atbildes iespējamās)

1 - 3 c/m;	7	18.9%
4 - 7 c/m;	7	18.9%
7 - 12 c/m;	6	16.2%
1 - 3 gadi;	16	43.2%
Cits	1	2.7%
Iesniegto atbilžu summa	37	

Citas atbildes:
10 gadu vecs projekts

3. Pēc kādas projektu pārvaldības metodes tiek vadīti projekti, kuros Tu darbojies? (vairākas atbildes iespējamās)

Agile;	9	21.4%
Udenskritums;	11	26.2%
SCRUM;	7	16.7%
Nenoteikts;	9	21.4%
Cits	6	14.3%
Iesniegto atbilžu summa	42	

Citas atbildes:
Isti nav skaidrs, kāda tā metode ir. Krustojum no Agile un Scrum.
V model
Mini ūdenskritums
V model
V-models
uzņēmuma iekšējais developemnt process

4. Vai uzņēmumā / struktūrvienībā / projektā ir noteiktas kādas prasības, kā jānotiek testēšanas procesam? (vairākas atbildes iespējamās)

Jā, aprakstītas iekšējās normatīvos;	11	22.4%
Jā, aprakstītas testētāja profesijas aprakstā;	5	10.2%
Jā, projekta vadītājs seko projekta līgumā noteiktām prasībām;	6	12.2%

Jā, uzņēmums testētājam piedāvā "rokasgrāmatu" par testēšanas procesa norisi;	5	10.2%
Nē;	0	0%
Darbojos pēc saviem ieskatiem;	9	18.4%
Strādāju atbilstoši projekta vadītāja norādēm;	10	20.4%
Cits	3	6.1%
Iesniegto atbilžu summa	49	

Citas atbildes:
Aprakstīts uzņēmuma kvalitātes sistēmā
Testējamajam produktam uzņēmumā isti nav izstrādāta testēšanas metodika
uzņēmums testētājam piedāvā "rokasgrāmatu" (mutiski) par testēšanas procesa norisi

5. Vai un kādu dokumentāciju parasti ir nepieciešams sagatavot testēšanas procesa gaitā? (vairākas atbildes iespējamās)

Jā, visu atbilstoši LVS 70:1996;	1	1.2%
Testēšanas plāns;	12	14.5%
Testu projektējuma specifikācija;	3	3.6%
Testpiemēru specifikācija;	16	19.3%
Testēšanas procedūras specifikācija;	3	3.6%
Testējamā vienuma pavadzīme;	3	3.6%
Testēšanas žurnāls;	10	12%
Testa problēmas ziņojums;	16	19.3%
Testēšanas kopsavilkuma pārskats;	15	18.1%
Ne, nav nepieciešams;	2	2.4%
Cits	2	2.4%
Iesniegto atbilžu summa	83	

Citas atbildes:
Dokumentācijas veidus un nepieciešamos standartus nosaka klienta prasības un uzņēmuma kvalitātes sistēmā aprakstītais minimālais nepieciešamais dokumentu komplekts. (LVS 70:1996 ir novecojis, bet gads, ka pasūtītājs prasa dokumentus atbilstoši tam)
atkarīgs no projekta specifikas, ir gadījumi, kur vairāki augstaakmeētie ir apvienoti vienā

6. Cik aptuveni laiku no visa testēšanas procesa aizņem dokumentācijas sagatavošana?

- 1) Maz, jo jāgatavo tikai viens dokuments un jāatraksta rezultāti pieteikumu portālā.
- 2) Pāris stundas. Tā principā ir atskaites sagatavošana, kuras izmērs ir atkarīgs no paveiktā darba apjoma.
- 3) 50%
- 4) 5-15%, atkarībā no projekta definētiem testēšanas nodevumiem.
- 5) puse
- 6) 40%
- 7) 1/3 daļa
- 8) Uz sāku projektu projekta pusē. Bija jāgatavo testēšanas scenāriji un testēšanas piemēri. Kā arī testa kopsavilkums. Dokumentācijas sagatavošana aizņem apmēram 40%.
- 9) Manā gadījumā tie ir 20%-30%
- 10) Vairāk kā 50%.
- 11) Aptuveni 30 - 40%
- 12) 40%
- 13) Aptuveni 40%
- 14) apmēram 20% no projekta - test lead time
- 15) 30-50%
- 16) 1/4
- 17) atkarīgs no taa cik efektīvas ir sagataves - ja taadas ir.

- jo kvalitatīva ir saakums, jo maza laika pateera papildinaashana
- 18) 20%
- 19) 30%
- 20) 70%
- 21) 30%
- 22) 20%
- 23) Katra projekta tas ir savadaks bet kopsumma nevar aizņemt vairak ka 10%

7. Vai izstrādātā testēšanas dokumentācija tiek nodota saskaņošanai klientam, vai tiek izstrādāta iekšējām vajadzībām? Piemēram?

- 1) Jā, iekšējās testēšanas protokols.
- 2) Tiek nodota saskaņošanai klientam.
- 3) Konfidencialitātes līgums neatļauj man izpaust šādu informāciju
- 4) Parasti tiek izstrādāti dokumenti, kas ir paredzēti nodevumiem (testēšanas plāns vai/un testpiemēri, iespējams testpiemēru specifikācija, atskaites). Viss, kas ir uz nodevumu, protams tiek arī saskaņots ar klientu, lai tiktu parakstīts akts. Gadās arī projekti, kur nodevums ir minimāls, bet uzņēmuma iekšējās procedūras definē veidot dokumentāciju, kas ir paredzēta izstrādes/uzturēšanas fāzei- piemēram regresijas testpiemēru kopa, tā parasti netiek nodota klientam.
- 5) Izstrādāta iekšējām vajadzībām, nodota klientam saskaņošanai.
- 6) Tiek nodots klientiem.
- 7) Ja klienta prasībās noteikts, kādu testēšanas dokumentāciju vēlas saņemt, tā tiek saskaņota ar klientu.

Ja nav precīzi definēti dokumenti / standarti, kādus izmantot, dokumentu komplekts tiek saskaņots projekta sākumā vai pēc nepieciešamības.

Iekšējām vajadzībām tiek izmantots minimālais dokumentu komplekts (testa plāns, kontrolosaraksti, problēmu pieteikumi), pēc iespējas izmantojot darba uzdevumu / problēmu pieteikumu pārvaldības rīkus (JIRA, TestDirector, TestTrack, TFS utt., utjpr.)

- 8) Ir vairāku veidu testēšanas dokumentācija. Tā kas, nododama klientam ir kādaļ ano nodevumu. Iekšējām vajadzībām tiek rakstīti scenāriji, kuri ir jāizpilda, lai varētu nodot programmas daļu.
- 9) Pagaidām, tikai iekšējām vajadzībām.
- 10) Reizēm tiek, bet ne vienmēr.
- 11) Gan tā, gan tā
- 12) atbilstoši līguma nosacījumiem
- 13) Tiek nodota klientam.
- 14) Gan, gan.

Regression test scope - izskatīts un apstiprināts ar pasūtītāju.

Testa piemēru prioritātes - iekšējām vajadzībām.

Gala atskaite - klientam un visām iesastītājām pusēm.

- 15) Tiek saskaņoti akcepttesti, pārējais nē.
- 16) Dažādi, atkarībā no projekta.
- 17) Atkarīgi no līgumaa mineetiem nosacījumiem, visi dokumenti, kas atziimeeti saskanjoshanai ir jaasaskanjo. Jo lielaaks projekts, jo lielaaka varbuutiiga, ka visi.

iznjeemums vareetu buit ieksheejais reglaments, tespiemeeru validaācijas pieraaditijumu materiaali no saakotneejaam testeeshanas stadijaam (specificki izstraades modelim) utml.

- 18) Pamatā tikai iekšējām vajadzībām. Klientam tiek sagatavota akceptances TC kopa.
- 19) nodod klientam
- 20) Jaa
- 21) Tiek nodota saskaņošanai klientam un arī tiek izstrādāta iekšējām vajadzībām - atkarībā no projekta, piemēram, testēšanas plāns, testpiemēru specifikācija
- 22) Pārsvārā tiek izstrādāta iekšējām vajadzībām, izņemot akceptances testpiemērus, kas tiek nodoti klientam
- 23) Ieksejam vajadzibam

8. Vai, Tavuprāt, izstrādātā dokumentācija ir nepieciešama klientam?

- 1) Jā.
- 2) Jā
- 3) Jā
- 4) Jebkura dokumentācija ir apliecinājums tam, ka tā tika izstrādāta vai/un pielietota. Respektīvi, jā-

nepieciešama.

- 5) Jā.
- 6) Protams
- 7) Nepieciešama, bet ne tādā apjomā, kā klienti to nereti prasa (un pēc tam pat nemēģina lasīt un izmantot).

Nepieciešams dokumentēt vienošanos par akcepttesta procesu un kritērijiem, par problēmu pieteikumu apstrādi, reizēm- testa piemēri/scenāriji.

8) Manuprāt, tā dokumentācija, kas ir iekļauta nodevumā, klientam nedod nekādu skaidrību.

Tiek veidoti Excel apkopojumi par to, kas funkcionē un kas nefunkcionē nodevumā programmas daļā, un pēc tiem var klienti vadīties, bet pēc sausiem testēšanas rezultātiem, īsti nesaskatu jēgu to nodošanā, kā tikai līguma punkta izpildīšanā.

- 9) Domāju ka jā, kontroles un situācijas apzināšanas vajadzībām.
- 10) Sarežģītai funkcionalitātei - noteikti.
- 11) Nav ne jausamas
- 12) atkarīgs no detalizācijas līmeņa
- 13) Jā, bet ne tik detalizēta, jo klienti ne vienmēr ir spējīgi to saprast.
- 14) Interesē:

Testa piemēri - idejiskā līmenī;

Testēšanas progresa atskaites;

Gala atskaite.

- 15) Ir nepieciešama, bet klientu grūti pierunāt lietot.
- 16) Reizēm jā, piemēram, PPA auditiem utml
- 17) Ja dokumenti ir gatavoti ir konkrētu meerkji, tad jaa.. jo dokumnets taisiits tikai "kjeksiisha peec" - pievienotas veertiibas shim pasaakumam vareetu arīi nebuut
- 18) Jā
- 19) jā un ne tikai (arī iekšēji tā strukturē darbu)
- 20) Protams
- 21) Testēšanas kopsavilkums - jā - atskaitē par padarīto darbu
- 22) Klientam dokumentāciju ir nepieciešama, lai saprastu, vai tiešām viņa prasības ir pareizi izprastas un sekojoši pareizi implementētas
- 23) Ja

9. Vai un kādas testēšanas metodes, tiek izmantotas projektos testēšanas procesa gaitā? (vairākas atbildes iespējamas)



Akcepttestēšana;	16	9.4%
Instalācijas testēšana;	8	4.7%
Cits	3	1.8%
Iesniegto atbīžu summa	171	

Citas atbildes:

visas augšupminētās

Testēšanas metodes nosaka projekta specifika, klienta prasības, testēšanas vadītājs/projekta vadītājs.

Project specific - atkarīgs no projekta apjoma un produkta, man ir bijusi taa laime stradaat ar masliava izstradeem - tur ir nepieciešams testeet produktu no dazjaadlem apsektiem.

10. Ja pienākumi / prasības, kas noteiktas uzņēmuma testēšanas procesa prasībās, netiktu pildītas tikai pienākuma pēc, vai, Tavuprāt, testēšanas kvalitāte kritīsies? Kāpēc?

- 1) Nē, jo vairāk tiks vēlēts laiks testēšanai nevis birokrātijai.
- 2) Nav mums tādu testēšanas procesa prasību, pats izdomāju kā organizēt procesu.
- 3) Pienākuma pēc strādā šoferā dēļ. Profesionālās kompānijās darbinieki dara savu darbu, lai sasniegtu kopēju rezultātu.

Šis jautājums ir pilnīgi neadekvāts.

- 4) Jā, jebkurš darbs, kas tiek pildīts nemotivēti, nav tik kvalitatīvs/ražīgs, kā gribētos.
- 5) Jā, kritīsies.
- 6) Jebkurā projektā ir jāstrādā pēc vadlīnijām un jābūt kādai plānošanai. Ja projekts nav sakārtots tad kvalitāte arī nebūs.
- 7) Testēšanas kvalitāte var kristies, izpildot formālas klienta prasības un nevajadzīgi tērējot laiku formālu dokumentu aizpildīšanai (to lietderīgāk var izmantot pašai testēšanai)

Minimālās uzņēmuma KVS (kvalitātes vadības sistēmā) noteiktās prasības nepildot, testēšanas kvalitāte var kristies, pastāv risks nepamatoti konkrētai IS būtiskas testēšanas metodes, tiek izveidots nepilnīgs pārklājums, mazāk pieredzējuši testētāji var nezināt, kas vēl jādara testētājam.

8) Manuprāt, nevajadzētu atņemt pilnīgi visus aprakstītos procesus, bet izskatīt un nebūtiskākos (birokrātiskākos) atņemt.

Jo nav liela jēga radīt čupu ar papīriem „kurus nodot. Svarīgi ir radīt kvalitatīvu produktu, ar kuru klients ir apmierināts un kurš nes uzņēmuma vārdu, kā uzticamu un atbildīgu pakalpojuma sniedzēju.

- 9) Uz doto brīdi, šo nevaru komentēt.
- 10) Noteikti kritīsies kvalitāte. Jo tad katrs varēs testēt pēc saviem ieskatiem, vadlīnijām noteikti jābūt un jābūt saskaņotam ar klientu ko u kā testēt.
- 11) nē
- 12) Kritīsies, jo palaujoties tikai uz savu intuīciju un programmatāju spējām var palaist garām dažādas kļūdas
- 13) Jā, iespējams. Jo nebūtu vairs pārskatāmība pār to, ko testētāji dara,
- 14) Nē. Prasībās noteiktais ir standarta testēšanas process, pēc kura tiek veikta testēšana.
- 15) Jā, kaut ko aizmirsīs notestēt.
- 16) Kritīsies.
- 17) Atkarīgi, no taa cik kvalitatīvas ir aprakstītas prasības. Ja aprakstītas prasības ir novecojušas un precīzē lietot arhaisku testēšanas rīku, kas nav efektīvs, piemēram skriptu rakstīšana aizņem daudz laika, vēl vairāk to uzturēšana no reliīzes līdz reliīzei, un tajaa pat laikaa testeetaaji jau ir atradūshi un veiksmīgi pielieto patēlciigaaku rīku - shaadas prasības nepildīšana kvalitaati negatiīvi neietekmees.

Tachu - lielaakoties - prasības ir paardomaatas un taas ir veidojushi cilvēeki ar pieredzi. Ja testeetaaas neredz uzlabojumus iisterminjaa, pastaav liela varbuutiiba, ka prasiiba ietekmee kvalitaate ilgterminjaa, taapeec to nedriikst neleveerot.

- 18) Atkarīgs no cilvēka
- 19) jā, pilnīga anarhija! nevar kontrolēt rezultātu & saņemt informāciju par progresu!
- 20) galvenais ir sekmīgs rezultāts
- 21) Pārāk sarežģīti formulēts jautājums.

Grūti pateikt

- 22) Manuprāt kvalitāte nekristos, tikai dažas kļūdas tiktu atrastas vēlāk
- 23) Ja kristos un jo gala produktu pasūtītājs saņem vji notestētu produktu.

11. Bakalaura darbā kā viens no objektiem tiek apskatīts "Lean domāšanas" veids (turpmāk Lean), kas saka, ka rezultātu iegūšanas procesā nepieciešams darbības kārtot secīgos soļos, izmantojot tikai soļus ar pievienotu vērtību. Pievienotā vērtība ir darba pienākumu un veikto darbu tiešais labums projektam, klientam, projekta komandai, bet kura projekts, Tavuprāt, ir neveiksmīgs, nepilnvērtīgs. Procesā nedrīkst izmantot izšķērtīgus, nevērtīgus soļus, lai rezultāta iegūšanas procesā pēc iespējas mazāk patērētu laiku un resursus papildus "birokrātiskām" darbībām. Bakalaura darbu ir paredzēts virzīt atbilstoši šim domāšanas veidam. Vai Jūs zināt, vai esat dzirdējuši, kas ir Lean? Stikāk var skatīt: <http://www.lean.org/WhatsLean/> (vairākas atbildes iespējamas)

Jā;	9	37.5%
Nē;	13	54.2%
Cits	2	8.3%
Iesniegto atbīžu summa	24	

Citas atbildes:

nu tagad jau zinu, jo ir taču links

Bet projekt avaditajs mīsu komandu organizē, tā lai vienmēr tiktu radīts pēc iespējas ar lielāku pievienoto vērtību rezultāts.

12. Kā, Tavuprāt, Lean iedarbības testētāja darba organizācija? (vairākas atbildes iespējamas)

Labā metodoloģija, kas ļāvu veikt tikai projektam nepieciešamos darbus;	9	33.3%
Nav nepieciešama metodoloģija;	0	0%
Patīk sekot pēc starptautiskajiem standartiem;	2	7.4%
Esmu apmierināts ar uzņēmuma prasībām testēšanas procesam;	9	33.3%
Cits	7	25.9%
Iesniegto atbīžu summa	27	

Citas atbildes:

Manā darbavietā ir savas vērtības, kas zināmā mērā pārklājas ar Lean'u, līdz ar to varētu teikt, ka mums tas ir liels, kamēr kopumā es piekriktu, ka tā būtu noderīga metodoloģija it īpaši valsts iestādēs.

metodoloģija nedrīkst būt viens, katram projektam der sava

Nav laika iedziļināties metodoloģijas pricipos, bet nu ja tās pamatā ir veikt nepieciešamos darbus, kas veicina ātrāku darba izpildi bez lieku procesu veikšanas, tad varētu būt lietderīga.

Lean ir autobūves piemērs, kas dod iestermīja risinājumu, tomēr nepareizi izmantojot, var radīt problēmas nākotnē laba metodoloģija sevī var iekļāst labāko no abām "passauleem". Standarti var šķīstīt morāli novecojuši pie pirmas skatu uzmetiena. Nav jāaplūda viss "by the book" ir jāņemties tieši piemērotā projekta, komandai un galvenokaart pasham produkt

Neesmu pazīstams ar Lean. Būtu jāizpēta un jāsaprot, vai un kā noder uzņēmumam. Bieži vien no katras metodes var paņemt vajadzīgo, netpildot visu pēc grāmatas.

Nav viedokļa

13. Vai, Tavuprāt, Lean varētu pielāgot testēšanas darbam, kāpēc? (vairākas atbildes iespējamas)

Jā;	15	62.5%
Nē;	4	16.7%
Cits	5	20.8%
Iesniegto atbīžu summa	24	

Citas atbildes:

es neiepazīnos ar lean

Dājēji, jo bieži, darbus, ko jāveic nosaka līgums.

Neesmu kompetents saistībā ar Lien metodoloģiju, līdz ar to nevaru komentēt šo jautājumu.

Visu var pielāgot un vajag pielāgot - produkta kvalitāte pirmajaa vietaa. Ja tests neatrod kļūdas - tas nozīmee, ka ir aspekts, kas nav testeets

Nezinu, nepieciešams tuvāk izpētēt metodoloģiju

14. Kādiem darba pienākumiem, projekta prasībām, Tavuprāt, ir pievienotā vērtība projekta ietvaros, testētāja ikdienā? (vairākas atbildes iespējamas)

Visas dokumentācijas sagatavošana;	13		30.2%
Testēšanas procesa prasību ievērošana;	18		41.9%
Starptautisko un LV standartu ievērošana;	5		11.6%
Noteiktas dokumentācijas, testēšanas metožu, standartu ievērošana, piemēram:	7		16.3%
Iesniegto atbilžu summa	43		

Citas atbildes:

Testēšanas protokoli
visam darbam ir pievienotā vērtība, kurš tad strādā par veilti?
Testēšanas procesa plānošana, kontrolsaraksti, problēmu pieteikumu reģistrēšana un apstrāde, izmantojot piemērotus rīkus
automātisko testu izveide
Sekošana testu plānam
Jebkas kas atbilst "lessons learned", "how we failed at this"
Atkarībā no projekta vai uzņēmuma prasībām - ja uzņēmums ir ISO sertificēts, tad uzņēmums pats apņemas sagatavot atbilstošo dokumentāciju

15. Kādiem darba pienākumiem, projekta prasībām, Tavuprāt, nav pievienotā vērtība projekta ietvaros, testētāja ikdienā?

- 1) Visai dokumentācijai, kas tiek noteikta standartos.
- 2) Nezinu, man tādu nav. Iepriekšējā darbā bija jāskaidra ar papīru rīņķī pa māju un jāvāc no visiem paraksti - tādām darbam nav pievienotās vērtības :)

Šeit neviens nenosaka kaut kādas prasības pret testēšanu neskaitot "notestējiet visu, ko vajag notestēt". Tālāk jau tas ir testētāju atbildībā paveikt visu nepieciešamo.

- 3) kafijas taisīšanai
- 4) Ja procedūra visu laiku tiek atjaunota, tad visam ir vērtība.
- 5) Plāno nepieciešamos tehniskos resursus
- 6) Kafijai/ smēķu lietošanai. :)
- 7) Dokumentācijas rakstīšana tikai tādēļ, lai atbilstu standartiem, šo dokumentu iesniegšana klientam drukātā, ne elektroniskā formā un šo dokumentu ignorēšana / klienta neieinteresētība par dokumentu saturu no pasūtītāja puses.

Pievienotās vērtības nav arī garām sapulcēm, kurā tiek apspriesti jautājumi, kas neietekmē testētāja darbu.

- 8) Darba stundu atskaitēm.

Dažādiem protokoliem.

- 9) Tā kā ar testēšanu nodarbojos tikai 6 mēnesi, tad pagaidām neesmu saskāries ar tādu konkrētu situāciju, kas projektam nedotu pievienoto vērtību.
- 10) Domāju, ka visiem ir.
 - 11) Nezinu
 - 12) Dokumentācijas izstrādei
 - 13) nodevumu sagatavošanā
 - 14) Testēt projektus, kas jau kādu laiku ir dzīvā sistēmā un ko jau ir notestējuši klienti.
 - 15) Man šķiet, viss ir nepieciešams
 - 16) -
 - 17) Veelme atrast kko kas veel nav paarbaidīts.

Testeetaajam ir jābūtu ar aicnaajumu. Ja komanda ir testeetaaji, kas naav "atklīshkjinaat" savu dienas devu - kvalitaate nodroshinaata netiks. labaakajaa gadiijumaa taa buus uztuureeshana ar meerkji, lai nepaliek sliiktaak.

- 18) Grūti uzreiz pateikt.
- 19) "smoke" testu pierādījumu vakšana
- 20) pashaī testeeshanaī
- 21) Nav viedokļa
- 22) šobrīd tādas nevaru iedomāties
- 23) -

16. Kādām testēšanas metodēm, Tavuprāt, nav pievienotās vērtības projekts? (vairākas atbildes iespējamas)

Vienībttestēšana;	2		3.6%
Integrācijas testēšana;	2		3.6%
Funkciju testēšana;	2		3.6%
Sistēmas spēju testēšana;	2		3.6%
Apstrādājamā apjoma testēšana;	2		3.6%
Stresa testēšana;	4		7.1%
Lietojamības testēšana;	5		8.9%
Drošības testēšana;	3		5.4%
Veiktspējas testēšana;	2		3.6%
Atmiņas testēšana;	2		3.6%
Uzstādījumu testēšana;	2		3.6%
Savietojamības, pārveidošanas testēšana;	3		5.4%
Atkopšanās testēšana;	5		8.9%
Akcepttestēšana;	3		5.4%
Instalācijas testēšana;	2		3.6%
Cits	15		26.8%
Iesniegto atbilžu summa	56		

Citas atbildes:

atkarīgs no projekta specifikas
Es te pusi nezinu pēc tiem latviskajiem nosaukumiem.
nesaprotams jautājums.
Metodes nepieciešamību vai prioritāti nosaka, tikai iepazīstot sistēmu, tās arhitektūru un lietotāju skaitu.
Kvalitātei vajag visus testēšanas procesus.
Uz šo jautājumu atbilde atšķiras atkarībā no projekta. Gadās, ka īpašas pievienotās vērtības snāv drošības un veiktspējas testiem, ja tos neveic vidē, kas maksimāli tuvināta pasūtītāja produkcijas vidē.
protokolu sagatavošanai. Vajag rīku, kas ģenerētu pēc izpildītajiem testiem, atskaites.
katrai no metodēm ir sava vērtība (atkarīgs no projekta).
Jebkura veida testēšanai ir pievienotā vērtība
visām ir
*
Visi ir vajadzīgi
Atkarīgs no projekta
atziļmeēju visu - lemesis - product specific - nav viennozīmīgas formulas ko vajag - vajag visu ko var atjauties un nodroshinaat, jo vairak jo labaak. nav runa par dokiem bet par paarklaajumu
Atkarībā no projekta visām testēšanas metodēm var būt pievienotā vērtība

Skaidrojums: Izvēlēto atbilžu summa ir iegūta saskaitot respondentu visas atzīmētās atbildes uz šo jautājumu. Katras atbildes procentuālā vērtība tiek iegūta konkrētās atbildes izvēlēto reižu skaitu dalot ar kopējo iesniegto atbilžu summu.

Bakalaura darbs „Programmatūras testēšanas procesu kvalitātes novērtēšana” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____ Laila Gutke

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītāja: profesore M.dat. Dace Zvīdriņa _____ 30.05.2013.

Recenzents: Dr.habil.dat., prof. Juris Borzovs

Darbs iesniegts Datorikas fakultātē 31.05.2013

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe _____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

_____. prot. Nr. _____

Komisijas sekretārs: _____