

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**PROGRAMMATŪRAS KVALITĀTES MĒRĪŠANA UN
SALĪDZINĀŠANA, IZMANTOJOT ISO/IEC 25022:2016
UN ISO/IEC 25023:2016 STANDARTUS**

MAĢISTRA DARBS

Autore: **Anete Kristīne Nesaule-Eriņa**

Studenta apliecības Nr. an11091

Darba vadītājs: prof. Dr.habil.dat. Juris Borzovs

RĪGA 2021

ANOTĀCIJA

Maģistra darbā tika pētīti kvalitātes mēri, kas aprakstīti standartos ISO/IEC 25022:16 [2] un ISO/IEC 25023:16 [3].

Darba mērķis bija izanalizēt standartos aprakstītos kvalitātes mērus, lai novērtētu, vai tie ir labi aprakstīti un ir pielietojami programmatūru lietojumkvalitātes un produkta kvalitātes novērtēšanā un salīdzināšanā.

Darba uzdevumi bija veikt standartos aprakstīto kvalitātes mēru analīzi, atrast un pielietot metodoloģiju programmatūras mēru novērtēšanai, kā arī veikt aptauju, lai noskaidrotu, kādus kvalitātes mērus izmanto programmatūras izstrādes nozarē Latvijā.

Darba gaitā tika secināts, ka, salīdzinot lietojumkvalitātes un produkta kvalitātes modeļa raksturlielumus no objektivitātes viedokļa, lietojumkvalitātes mēri ir vairāk subjektīvi, savukārt produkta kvalitātes mēri lielākoties ir objektīvi un savstarpēji salīdzināmi. Aptaujas rezultāti norāda, ka tikai divus mērus no standartiem izmanto vairāk nekā 80% aptaujāto uzņēmumu, bet vienu ceturtdaļu no mēriem izmanto mazāk nekā 20% aptaujāto uzņēmumu. Mēru novērtēšanai tika izvēlēta 12 soļu metodoloģija derīgiem programmatūras mēriem un pēc tās tika novērtēti mēri, kas starp aptaujātajiem uzņēmumiem tiek izmantoti visplašāk.

Atslēgas vārdi: programmatūras kvalitāte, lietojumkvalitāte, metrika, mērs, atribūts, ISO/IEC 25022:2016, ISO/IEC 25023:2016.

ABSTRACT

Title: Measurement and comparison of software quality using ISO/IEC 25022:2016 and ISO/IEC 25023:2016 standards

This master work focuses on finding information about the quality measures that are described in standards ISO/IEC 25022:16 [2] and ISO/IEC 25023:16 [3].

The main goal of the master work was to evaluate quality measures that are described in standards, in order to assess whether they are well described and applicable in the evaluation and comparison of software quality in use and product quality.

The tasks of the work was to analyse the quality measures described in the standards, to find and apply the methodology for evaluating software measures, as well as to conduct a survey to find out what quality measures are used in the software development industry in Latvia.

The research concludes that comparing the characteristics of the quality in use and product quality model from the point of view of objectivity, application quality measures are more subjective, while product quality measures are mostly objective and comparable. The result of the survey shows that only two measures of the standards are used by more than 80% of the surveyed companies, but one quarter of the measures are used by less than 20% of the surveyed companies. A 12 Steps methodology to Useful Software Metrics was chosen to evaluate the measures, and the measures that are most widely used among the surveyed companies were evaluated.

Keywords: software quality, quality in use, metric, measure, attribute, ISO/IEC 25022:2016, ISO/IEC 25023:2016.

AUTOREFERĀTS

Autore darba ietvaros ir veikusi standartu ISO/IEC 25022:2016 [2] un ISO/IEC 25023:2016 [3] izpēti un tajos aprakstīto mēru analīzi. Atbilstoši standartos aprakstītajam, mēri ir iedalīti vispārīgos un konkrētos mēros, kā arī pēc to rekomendācijas līmeņa. Papildus ir analizēts tas, cik mērs ir objektīvs vai subjektīvs. Tika veidota elektroniska aptauja atbilstoši standartos apkopotajiem raksturlielumiem un mēriem, kas tika nodota izpildei dažādiem uzņēmumiem Latvijā, kas ir saistīti ar IT nozari. Ir veikta aptaujas respondentu analīze un iegūto rezultātu izvērtēšana. Tika veikta astoņu dažādu zinātnisko rakstu izpēte, lai atrastu metodoloģiju mēru novērtēšanai, jo lai mēru varētu izmantot kvalitātes novērtēšanā, tam ir jābūt uzticamam un kvalitatīvam. Pēc rakstu izpētes tika izvēlēta un aprakstīta metodoloģija mēru novērtēšanai, un atbilstoši tai tika veikta mēru novērtēšana. Laika trūkuma dēļ padziļinātāk tika veikta deviņu mēru novērtēšana, kas pēc aptaujas rezultātiem ir izmantoti starp visvairāk uzņēmumiem.

Programmatūras kvalitātes novērtēšanai ir liela nozīme, un būtu jābūt vienotam veidam, kā to novērtēt. Starptautiskie standarti apraksta daudz un dažādus mērus tam, kā novērtēt lietojumkvalitāti un produkta kvalitāti. Taču šajos standartos aprakstītais ir izmantojams tikai tad, ja tas ir praktiski pielietojams un nozarē tiek vienādi izmantots, lai būtu iespējams salīdzināt dažādu programmatūru kvalitāti. Autore vēlas noskaidrot, vai ar šo standartu palīdzību var novērtēt programmatūras kvalitāti, un vai tas tiek praktiski pielietots IT nozarē Latvijā.

Avotu un izmantotās literatūras saraksts ir apkopots darba beigās. Maģistra darba terminoloģija saskaņota ar Akadēmisko terminu datubāzi *AkadTerm* [5]. Darba galvenie avoti ir starptautiskie standarti ISO/IEC 25010:2011 [1] par sistēmu un programmatūras kvalitātes prasībām un novērtēšanu, ISO/IEC 25022:2016 [2] par lietojumkvalitātes mērīšanu un ISO/IEC 25023:2016 [3] par sistēmu un programmatūru produkta kvalitātes mērīšanu. Darbā ir atsauce uz Dr. sc. comp. Vinetas Arnicānes, Dr.habil.dat. Jura Borzova un autores zinātnisko rakstu “*Do We Really Know How to Measure Software Quality*” [4]. Papildus tika izmantoti vēl astoņi zinātniskie raksti, kā arī citi tīmekļa resursi.

Līdz šim citos literatūras avotos biežāk tiek apskatīta tieši programmatūras koda kvalitāte un tā novērtēšana, bet mazāk pašas programmatūras vai produkta kvalitātes un lietojumkvalitātes novērtēšana. Tas apstiprinājās arī pēc veiktās literatūras izpētes darba ietvaros. Lai gan ir izstrādāti starptautiski standarti, kas ietver dažādus raksturlielumus un to mērus, kas paredzēti programmatūras kvalitātes novērtēšanai, svarīgi ir tas, vai tie tiek praktiski pielietoti nozarē. Plašāk pieejama informācija par Latvijas IT nozarē izmantotajām programmatūras kvalitātes novērtēšanas metodēm nav pieejama, tāpēc tika veikta aptauja, lai

uzzinātu, vai nozarē tiek izmantoti starptautiskie standarti. Darbā iegūtie rezultāti parāda to, ka standartos aprakstītie raksturlielumi un to raksturojošie mēri ir pielietojami kvalitātes novērtēšanā, taču lai tie praksē tiktu izmantoti vienādi, standartā pieejamā informācija būtu jāpapildina. Tas atbilst piebildei no standarta, ka tā nav galīgā versija, un mēri var tikt pārskatīti un pilnveidoti. No praktiskās puses, nozarē tiek izmantoti mēri kvalitātes novērtēšanai no starptautiskajiem standartiem, taču mazākā apjomā. Izmantot visus 122 standartos aprakstītos mērus būtu laikietilpīgi un dārgi. Kvalitātes raksturlielumus bieži vien var novērtēt pat ar vienu mēru, un tas sniedz pietiekamu kvalitātes novērtējumu.

Maģistra darba autore ir galvenā raksta “*Do We Really Know How to Measure Software Quality*” [4] autore un personīgi to prezentēja 14. starptautiskajā Baltijas konferencē par datu bāzēm un informācijas sistēmām (*Baltic DB&IS 2020*).

Darba teksts ir pārlasīts un pamanītās kļūdas ir izlabotas. Darbā ir izmantota latviešu valodā oficiāli pieņemtā nozares terminoloģija. Darbs atbilst maģistra darba metodiskajiem norādījumiem, kā arī ir ievērota maģistra darba struktūra un tehniskais noformējums. Darbā ir iekļautas atsauces uz izmantotajiem avotiem no literatūras saraksta.

SATURA RĀDĪTĀJS

Apzīmējumu saraksts.....	8
Ievads.....	10
1. Standarti ISO/IEC 25022:2016 un ISO/IEC 25023:2016	11
1.1. Kvalitātes modeļi.....	11
1.2. Vispārīgi un konkrēti mēri	12
1.3. Mēru rekomendācijas līmeņi.....	13
2. Mēru objektivitāte.....	15
2.1. Mēru objektivitāte	15
2.2. Kvalitātes noteikšana, izmantojot mērus no standartiem.....	16
3. Aptauja.....	20
3.1. Aptaujas mērķis.....	20
3.2. Aptaujas apraksts.....	20
3.3. Aptaujas metode.....	20
3.4. Respondentu analīze.....	21
3.5. Aptaujas rezultāti.....	22
4. Programmatūras kvalitātes noteikšana	24
4.1. Metodoloģija programmatūras mēru novērtēšanai.....	24
4.1.1. Literatūras izpēte	24
4.1.2. 12 soļi, lai iegūtu derīgus mērus programmatūras novērtēšanai	26
4.1.3. SMART novērtēšanas kritēriji	31
4.1.4. Programmatūras mēru novērtēšanas metodoloģija	32
4.2. Lietojumkvalitātes mēru novērtēšana.....	34
4.2.1. Riskbrīvība	34
4.2.2. Apmierinātība.....	38
4.2.4. Lietojumkvalitātes mēri.....	41
4.3. Programmatūras produkta kvalitātes mēru novērtēšana.....	42
4.3.1. Veiktspējas efektivitāte	43

4.3.2. Funkcionālā atbilstība	51
4.3.3. Uzticamība	54
4.3.5. Programmatūras produkta kvalitātes mēri.....	61
Rezultāti un secinājumi	63
Pateicības.....	65
Avotu un izmantotās literatūras saraksts	66
Pielikumi	68
1. pielikums. Zinātniskais raksts “Do We Really Know How to Measure Software Quality”	69

APZĪMĒJUMU SARAKSTS

- Apģūstamība** – no angļu valodas *Learnability*
- Adaptējamība** – no angļu valodas *Adaptability*
- Aizstāspēja** – no angļu valodas *Replaceability*
- Analizējamība** – no angļu valodas *Analysability*
- Apakšraksturlielums** – no angļu valodas *subcharacteristic*
- Apmierinātība** – no angļu valodas *Satisfaction*
- Atbildība** – no angļu valodas *Accountability*
- Atbilstības atpazīšana** – no angļu valodas *Appropriateness Recognizability*
- Atkārtota lietojamība** – no angļu valodas *Reusability*
- Atkopjamība** – no angļu valodas *Recoverability*
- Autentiskums** – no angļu valodas *Authenticity*
- Briedums** – no angļu valodas *Maturity*
- Darbināmība** – no angļu valodas *Operability*
- Defektpiecietība** – no angļu valodas *Fault Tolerance*
- Drošība** – no angļu valodas *Security*
- Efektivitāte** – no angļu valodas *Efficiency*
- Ekonomiskā riska mazinājums** – no angļu valodas *Economic risk mitigation*
- Ekstrēmās programmēšanas metodoloģija** – no angļu valodas *extreme programming*
- Ērtums** – no angļu valodas *Comfort*
- Funkcionālā atbilstība** – no angļu valodas *Functional suitability*
- Funkcionālā pareizība** – no angļu valodas *Functional correctness*
- Funkcionālā piemērotība** – no angļu valodas *Functional appropriateness*
- Funkcionālais pilnīgums** – no angļu valodas *Functional completeness*
- IEC** – Starptautiskā elektrotehniskā komisija (*International Electrotechnical Commission*)
- Instalējamība** – no angļu valodas *Installability*
- Integritāte** – no angļu valodas *Integrity*
- ISO** – Starptautiskā standartizācijas organizācija (*International Organization for Standardization*)
- IT** – informācijas tehnoloģija (*information technologies*)
- Jauda** – no angļu valodas *Capacity*
- Konfidencialitāte** – no angļu valodas *Confidentiality*
- Kontekstu pārklājums** – no angļu valodas *Context coverage*
- Kontekstu pilnīgums** – no angļu valodas *Context completeness*

Laikuzvedība – no angļu valodas *Time Behavior*

Lean metodoloģija – no angļu valodas *lean methodology*

Līdzāspastāvēšana – no angļu valodas *Co-existence*

Lietojamība – no angļu valodas *Usability*

Lietojumkvalitāte – no angļu valodas *Quality in use*

Lietotāju kļūdu aizsardzība – no angļu valodas *User Error Protection*

Lietotāju saskarnes estētika – no angļu valodas *User Interface Aesthetics*

Modularitāte – no angļu valodas *Modularity*

Nenoliedzamība – no angļu valodas *Non-repudiation*

Noderīgums – no angļu valodas *Usefulness*

Pārnēsamība – no angļu valodas *Portability*

Pārveidojamība – no angļu valodas *Modifiability*

Patika – no angļu valodas *Pleasure*

Pieejamība – no angļu valodas *Availability*

Piekļūstamība – no angļu valodas *Accessibility*

Pielāgojamība – no angļu valodas *Flexibility*

Prototipēšanas metodoloģija – no angļu valodas *prototype methodology*

Resursu izmantošana – no angļu valodas *Resource utilization*

Rezultatīvitate – no angļu valodas *Effectiveness*

Riskbrīvība – no angļu valodas *Freedom from risk*

ROI – saīsinājums peļņa no ieguldījuma no angļu valodas *Return on investment*

Sadarbspēja – no angļu valodas *Interoperability*

Saderība – no angļu valodas *Compatibility*

Scrum metodoloģija - no angļu valodas *scrum software development*

Spējā programmatūras izstrāde – no angļu valodas *agile software development*

Spirāles metodoloģija – no angļu valodas *spiral model*

Testējamība – no angļu valodas *Testability*

Ūdenskrituma programmatūras izstrāde – no angļu valodas *waterfall software development*

Uzticamība – no angļu valodas *Reliability*

Uzticēšanās – no angļu valodas *Trust*

Uzturamība – no angļu valodas *Maintainability*

Veiktspējas efektivitāte – no angļu valodas *Performance efficiency*

Veselības un drošības riska mazinājums – no angļu valodas *Health and safety risk*

Vides risku mazinājums – no angļu valodas *Environmental risk mitigation*

IEVADS

Mūsdienās programmatūru izstrāde ir ļoti aktuāla dažādās nozarēs. Ir svarīgi, lai izstrādātā programmatūra atbilst lietotāju prasībām un apmierina to vajadzības, bet bez tā programmatūrai ir jābūt arī ir kvalitatīvai un jāatbilst visām kvalitātes prasībām. Kā zināt, vai programmatūra ir pietiekami kvalitatīva, un vai tā ir kvalitatīvāka par kādu citu līdzīgu programmatūru? Ir pieejami standarti ISO/IEC 25022:2016 [2] un ISO/IEC 25023:2016 [3], kas apkopo lietojumkvalitātes un sistēmas/programmatūras produkta kvalitātes mērus. Kopā šajos divos standartos ir apkopoti 122 mēri, kas raksturo 13 raksturlielumus un 43 apakšraksturlielumus. Bet vai tie visi ir objektīvi, uzticami un reāli pielietojami? Maģistra darba ietvaros tika veikta standartos pieejamo mēru analīze, lai novērtētu, vai tie ir pielietojami programmatūru lietojumkvalitātes un produkta kvalitātes novērtēšanā un salīdzināšanā.

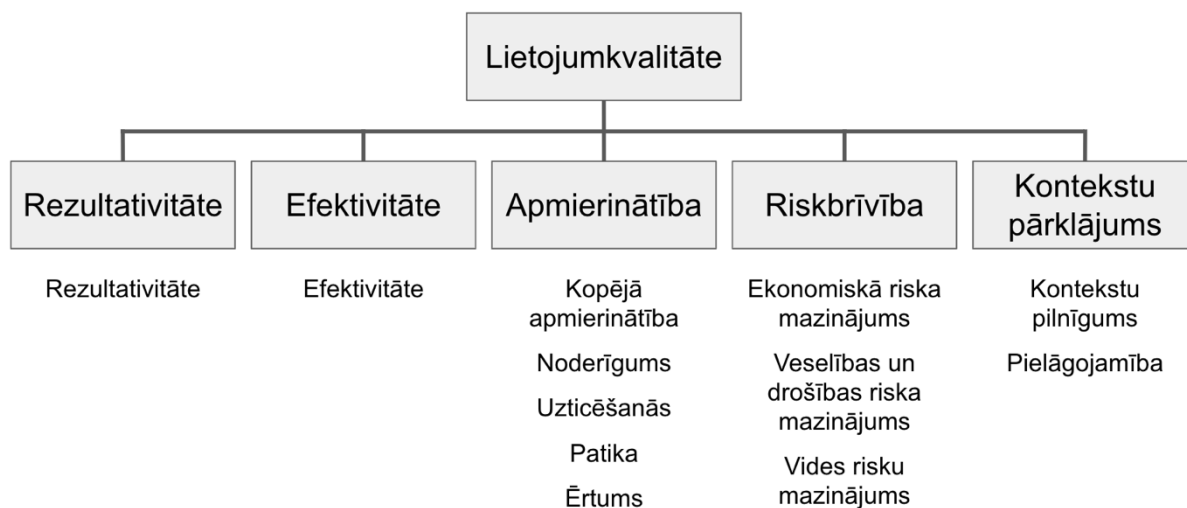
Darba laikā tika izpētīti ISO/IEC 25022:2016 un ISO/IEC 25023:2016 standartos pieejamie mēri un informācija par tiem. Tika veikta aptauja, lai noskaidrotu, kādus mērus izmanto IT nozarē Latvijā. Tas lika secināt to, ka standartos ir pieejami ļoti daudz un dažādi mēri, lai izmērītu programmatūras kvalitāti, bet tikai daļa no tiem tiek praktiski pielietota IT nozarē Latvijā. Lai noskaidrotu, kāpēc daļa mēru tiek izmantoti biežāk, un daļa netiek izmantota, tika veikta standartos pieejamo mēru novērtēšana. Nozīmīga šī maģistra darba rezultātu daļa tika prezentēta 14. starptautiskajā Baltijas konferencē par datu bāzēm un informācijas sistēmām (*Baltic DB&IS 2020*) ar rakstu “*Do We Really Know How to Measure Software Quality*” [4].

Maģistra darbs sastāv no 4 daļām. Tā pirmajā nodaļā ir aprakstīta standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 pieejamā informācija par mēriem. Otrajā nodaļā ir analizēta standartos pieejamo mēro objektivitāte. Trešajā nodaļā tiek aprakstīta veiktā aptauja un apkopoti aptaujā iegūtie rezultāti. Ceturtajā nodaļā ir aprakstīta metodoloģija, lai novērtētu mērus, ko izmantot programmatūras kvalitātes noteikšanai, kā arī veikta standartos pieejamo mēru analīze, lai novērtētu, vai tie ir labi aprakstīti un ir pielietojami dažādu programmatūru novērtēšanā un salīdzināšanā, radot līdzsvaru starp izmaksām, kvalitāti un laiku.

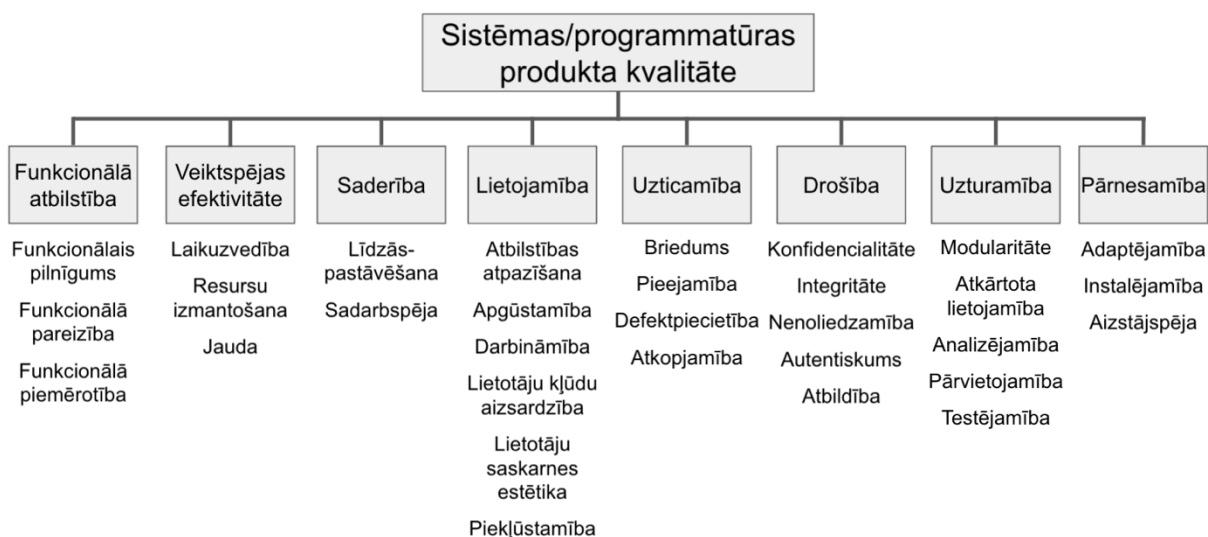
1. STANDARTI ISO/IEC 25022:2016 UN ISO/IEC 25023:2016

1.1. Kvalitātes modeļi

ISO/IEC 25022:2016 [2] un ISO/IEC 25023:2016 [3] ir daļa no starptautisko standartu sērijas no ISO/IEC 25000 līdz ISO/IEC 25099 ar nosaukumu “*Systems and software engineering – Systems and software Quality Requirements and Evaluation*” jeb “Sistēmu un programmatūras inženierija - Sistēmu un programmatūras kvalitātes prasības un novērtējums” ar saīsinājumu *SQuaRE*. Standartā ISO/IEC 25010:2011 [1] ir definēti divi kvalitātes modeļi – lietojumkvalitātes modelis (skatīt *1.1. att.*) un sistēmas/programmatūras produkta kvalitātes modelis (skatīt *1.2. att.*). Katrs no šiem modeļiem sastāv no kvalitātes raksturlielumiem un to apakšraksturlielumiem, tos, savukārt, raksturo dažādi mēri. Mērs standartā ISO/IEC 25010:2011 tiek definēts kā mainīgais, kuram mērīšanas rezultātā tiek piešķirta vērtība [1].



1.1. att. Lietojumkvalitātes modelis



1.2. att. Sistēmas/programmatūras produkta kvalitātes modelis

1.2. Vispārīgi un konkrēti mēri

Katram mēram standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 ir norādīts, vai tas ir konkrēts (*specific*) vai vispārīgs (*generic*) mērs. To apzīmē katra mēra identifikācijas koda pēdējais burts, kur “S” apzīmē konkrētu mēru, bet “G” – vispārīgu mēru (skatīt 1.3. att.), kur apkopoti standartos aprakstīto mēru identifikatori, tādā secībā kādā tie ir standartos un iekrāsoti atbilstoši sadalījumam. Vispārīgi mēri ir vispārēji piemērojami, tos var izmantot dažādās situācijās, savukārt, konkrēti mēri ir paredzēti speciāliem gadījumiem, kad tas ir nepieciešams.

Piemērs vispārīgam mēram ir “MRe-1-G aktīvu atkārtota lietojamība (*Reusability of assets*)”, šis mērs norāda to, cik daudz sistēmas aktīvu var tikt lietoti atkārtoti. Tas tiek aprēķināts kā attiecība starp visiem aktīviem, kas ir izstrādāti un ieviesti, lai tiktu atkārtoti lietoti un visiem sistēmas aktīviem.

Piemērs konkrētam mēram ir “REc-6-S tīmekļa vietņu apmeklētāji, kas kļuvuši par klientiem (*Website visitors converted to customers*)”, šis mērs attēlo konkrētas tīmekļa vietnes apmeklētāju proporciju, kas kļuvuši par klientiem. Tas tiek mērīts, izmantojot darījumanalītiku (*business analytics*).

Ef-1-G	SPI-1-G	FCp-1-G	CIn-2-G	UEp-1-G	SCo-2-G	MMd-2-G
Ef-2-S	SCo-1-G	FCr-1-G	CIn-3-S	UEp-2-S	SCo-3-S	MMd-3-S
Ef-3-G	REc-1-G	FAP-1-G	UAp-1-G	UEp-3-S	SIn-1-G	MTe-1-G
Ef-4-G	REc-2-G	FAP-2-G	UAp-2-S	UIn-1-S	SIn-2-G	MTe-2-S
Ef-5-G	REc-3-G	PTb-1-G	UAp-3-S	UAc-1-G	SIn-3-S	MTe-3-S
Ey-1-G	REc-4-G	PTb-2-G	ULe-1-G	UAc-2-S	SNo-1-G	PAd-1-G
Ey-2-S	REc-5-S	PTb-3-G	ULe-2-S	RMa-1-G	SAC-1-G	PAd-2-G
Ey-3-S	REc-6-S	PTb-4-G	ULe-3-S	RMa-2-G	SAC-2-S	PAd-3-S
Ey-4-S	REc-7-S	PTb-5-G	ULe-4-S	RMa-3-G	SAu-1-G	Pln-1-G
Ey-5-S	REc-8-G	PRu-1-G	UOp-1-G	RMa-4-S	SAu-2-S	Pln-2-G
Ef-6-S	RHe-1-G	PRu-2-G	UOp-2-G	RAv-1-G	MMo-1-G	PRE-1-G
Sus-1-G	RHe-2-G	PRu-3-G	UOp-3-S	RAv-2-G	MMo-2-S	PRE-2-S
Sus-2-G	RHe-3-G	PRu-4-S	UOp-4-S	Rft-1-G	MRe-1-G	PRE-3-S
Sus-3-G	REn-1-G	PCa-1-G	UOp-5-S	Rft-2-S	MRe-2-S	PRE-4-S
Sus-4-G	CCm-1-G	PCa-2-G	UOp-6-S	Rft-3-S	MAn-1-G	
Sus-5-G	CFI-1-S	PCa-3-S	UOp-7-S	RRe-1-G	MAn-2-S	
Sus-6-G	CFI-2-S	CCo-1-G	UOp-8-S	RRe-2-S	MAn-3-S	
STr-1-G	CFI-3-S	CIn-1-G	UOp-9-S	SCo-1-G	MMd-1-G	

Apzīmējumi:
Vispārīgs
Konkrēts

1.3. att. Vispārīgu un konkrētu mēru sadalījums

1.3. Mēru rekomendācijas līmeņi

Katrs mērs standartā ISO/IEC 25023:2016, kas atbilst sistēmas/programmatūras produkta kvalitātes modelim, var tikt klasificēts pēc rekomendācijas līmeņa. Standartā ISO/IEC 25023:2016 ir izdalīti trīs rekomendāciju līmeņi – ļoti ieteicams (*highly recommended*), ieteicams (*recommended*) un izmantojams pēc lietotāja ieskatiem (*used at user's discretion*) (skatīt 1.4. att.). Ļoti ieteicamos mērus iesaka izmantot vienmēr. Ieteicamos mērus iesaka izmantot tad, kad tas ir nepieciešams, taču standartā nav paskaidrots, kad ir lietderīgi izmantot katru no mēriem. Pēc lietotāja ieskatiem izmantojamus mērus izmanto kā atsauci, izstrādājot jaunu kvalitātes mēru, jo šiem mēriem nav zināma uzticamība (*unknown reliability*).

Piemērs ļoti ieteicamam mēram ir “SCo-1-G piekļuves vadāmība (*Access controllability*)”, šis mērs nosaka, kāda daļa konfidencialu datu vienumu ir aizsargāta pret nesankcionētu piekļuvi. Tas tiek aprēķināts kā attiecība starp visiem konfidencialajiem datu vienumiem, kuriem var piekļūt bez pilnvarošanas (*authorisation*), un datiem, kuriem ir nepieciešama piekļuves kontrole.

Piemērs ieteicamam mēram ir “RAv-2-G vidējais dīkstāves laiks (*Mean down time*)”, šis mērs nosaka to, cik ilgi sistēma paliek nepieejama, ja rodas atteice. Tas tiek aprēķināts kā attiecība starp kopējo dīkstāves laiku un novērotajām atteicēm.

Piemērs pēc lietotāja ieskatiem izmantojamam mēram ir “UAc-2-S atbalstīto valodu adekvātums (*Supported languages adequacy*)”, šis mērs nosaka, kāda proporcija nepieciešamo valodu tiek atbalstīta. Tas tiek aprēķināts kā attiecība starp faktiski atbalstītajām valodām un tām valodām, kuras būtu jāatbalsta.

Ef-1-G	SPI-1-G	FCp-1-G	CIn-2-G	UEp-1-G	SCo-2-G	MMd-2-G	
Ef-2-S	SCo-1-G	FCr-1-G	CIn-3-S	UEp-2-S	SCo-3-S	MMd-3-S	
Ef-3-G	REc-1-G	FAP-1-G	UAp-1-G	UEp-3-S	SIn-1-G	MTE-1-G	
Ef-4-G	REc-2-G	FAP-2-G	UAp-2-S	UIn-1-S	SIn-2-G	MTE-2-S	
Ef-5-G	REc-3-G	PTb-1-G	UAp-3-S	UAc-1-G	SIn-3-S	MTE-3-S	
Ey-1-G	REc-4-G	PTb-2-G	ULe-1-G	UAc-2-S	SNo-1-G	PAd-1-G	
Ey-2-S	REc-5-S	PTb-3-G	ULe-2-S	RMa-1-G	SAC-1-G	PAd-2-G	
Ey-3-S	REc-6-S	PTb-4-G	ULe-3-S	RMa-2-G	SAC-2-S	PAd-3-S	
Ey-4-S	REc-7-S	PTb-5-G	ULe-4-S	RMa-3-G	SAu-1-G	Pln-1-G	
Ey-5-S	REc-8-G	PRu-1-G	UOp-1-G	RMa-4-S	SAu-2-S	Pln-2-G	
Ef-6-S	RHe-1-G	PRu-2-G	UOp-2-G	RAv-1-G	MMo-1-G	PRE-1-G	
Sus-1-G	RHe-2-G	PRu-3-G	UOp-3-S	RAv-2-G	MMo-2-S	PRE-2-S	
Sus-2-G	RHe-3-G	PRu-4-S	UOp-4-S	RFt-1-G	MRe-1-G	PRE-3-S	
Sus-3-G	REn-1-G	PCa-1-G	UOp-5-S	RFt-2-S	MRe-2-S	PRE-4-S	
Sus-4-G	CCm-1-G	PCa-2-G	UOp-6-S	RFt-3-S	MAn-1-G		Apzīmējumi:
Sus-5-G	CFI-1-S	PCa-3-S	UOp-7-S	RRe-1-G	MAn-2-S		UD
Sus-6-G	CFI-2-S	CCo-1-G	UOp-8-S	RRe-2-S	MAn-3-S		R
STr-1-G	CFI-3-S	CIn-1-G	UOp-9-S	SCo-1-G	MMd-1-G		HR

1.4. att. Mēru rekomendāciju līmeņu sadalījums

2. MĒRU OBJEKTIVITĀTE

2.1. Mēru objektivitāte

Mērs var būt piemērots un uzticams tikai tad, ja tas ir objektīvi izmērāms. Autore kopā ar Dr. sc. comp. Vinetu Arnicāni un Dr.habil.dat. Juri Borzovu veica standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 pieejamo mēru objektivitātes noteikšanu (skatīt 2.1. att.).

Ef-1-G	SPI-1-G	FCp-1-G	Cln-2-G	UEp-1-G	SCo-2-G	MMd-2-G	
Ef-2-S	SCo-1-G	FCr-1-G	Cln-3-S	UEp-2-S	SCo-3-S	MMd-3-S	
Ef-3-G	REc-1-G	FAP-1-G	UAp-1-G	UEp-3-S	Sln-1-G	MTE-1-G	
Ef-4-G	REc-2-G	FAP-2-G	UAp-2-S	Uln-1-S	Sln-2-G	MTE-2-S	
Ef-5-G	REc-3-G	PTb-1-G	UAp-3-S	UAc-1-G	Sln-3-S	MTE-3-S	
Ey-1-G	REc-4-G	PTb-2-G	Ule-1-G	UAc-2-S	SNo-1-G	PAd-1-G	
Ey-2-S	REc-5-S	PTb-3-G	Ule-2-S	RMa-1-G	SAC-1-G	PAd-2-G	
Ey-3-S	REc-6-S	PTb-4-G	Ule-3-S	RMa-2-G	SAC-2-S	PAd-3-S	
Ey-4-S	REc-7-S	PTb-5-G	Ule-4-S	RMa-3-G	SAu-1-G	Pln-1-G	
Ey-5-S	REc-8-G	PRu-1-G	UOp-1-G	RMa-4-S	SAu-2-S	Pln-2-G	
Ef-6-S	RHe-1-G	PRu-2-G	UOp-2-G	RAv-1-G	MMo-1-G	PRE-1-G	
Sus-1-G	RHe-2-G	PRu-3-G	UOp-3-S	RAv-2-G	MMo-2-S	PRE-2-S	
Sus-2-G	RHe-3-G	PRu-4-S	UOp-4-S	Rft-1-G	MRe-1-G	PRE-3-S	Apzīmējumi:
Sus-3-G	RĒn-1-G	PCa-1-G	UOp-5-S	Rft-2-S	MRe-2-S	PRE-4-S	Subjektīvs
Sus-4-G	CCm-1-G	PCa-2-G	UOp-6-S	Rft-3-S	MAn-1-G		Objektīvs, ietekmē lietotāja kvalifikācijas
Sus-5-G	CFI-1-S	PCa-3-S	UOp-7-S	RRe-1-G	MAn-2-S		Objektīvs konkrētam produktam
Sus-6-G	CFI-2-S	CCo-1-G	UOp-8-S	RRe-2-S	MAn-3-S		Objektīvs, ja eksistē kritēriji
STr-1-G	CFI-3-S	Cln-1-G	UOp-9-S	SCo-1-G	MMd-1-G		Objektīvs, salīdzināms

2.1. att. Mēru objektivitāte

Mēri, kas nav novērtējami objektīvi, ir tādi mēri, kas tiek iegūti, piemēram, no aptaujām vai lietotāju atgriezeniskās saites, šādi mērījumi būs subjektīvi. Piemērs subjektīvam mēram ir “Sus-2-G apmierinātība ar raksturiezīmēm (*Satisfaction with features*)”, šis mērs nosaka lietotāja apmierinātību ar konkrētām sistēmas raksturiezīmēm. To var novērtēt izmantojot aptaujas, bet lietotāju sistēmas funkciju novērtējums var būt ļoti subjektīvs, atkarībā no lietotāja attieksmes pret pašu sistēmu, garastāvokļa, noguruma un citiem faktoriem.

Objektīvi iegūstamus mērus var iedalīt vairākos veidos, ņemot vērā to, vai rezultātus ietekmē lietotāja kvalifikācija un zināšanas, vai eksistē industrijas kritēriji, un vai šie mēri ir salīdzināmi starp dažādām sistēmām un produktiem. Piemēram, mērs “Ef-3-G kļūdas uzdevumā (*Errors in task*)”, kas nosaka kļūdu skaitu, ko izdara lietotājs uzdevuma veikšanas laikā, ir objektīvs mērs, tas tiek novērtēts pēc lietotāja izpildes rādītājiem, taču to ietekmē lietotāja kvalifikācijas, pieredze, zināšanas un prasmes. Tāpēc šis mērs nav piemērojams vienas sistēmas dažādu posmu, kā arī dažādu sistēmu salīdzināšanai.

Savukārt, mērs “Rec-2-G laiks, lai panāktu ieguldījumu atdevi (*Time to achieve return on investment*)”, kas parāda laiku, kas nepieciešams, lai panāktu ieguldījumu gaidāmo atdevi,

ko aprēķina veicot rūpīgu darījuma analīzi, būs objektīvs tikai konkrētam produktam, un salīdzināšana starp dažādiem produktiem būs tikai relatīva.

Vēl ir mēri, kas būs objektīvi tikai tad, ja eksistē industrijas kritēriji vai etaloni, piemēram, “MTe-1-G testa funkciju pilnīgums (*Test function completeness*)”, kas nosaka, cik pilnīgi ir implementētas testu funkcijas un iespējas. To var izteikt kā attiecību starp testa funkcijām, kas implementētas atbilstoši specifikācijai un nepieciešamajām testa funkcijām. Šis mērs ir objektīvi pielietojams tikai tad, ja ir kāds industrijas kritērijs, kas nosaka to, cik daudz testu funkcijas ir nepieciešamas, ja šāda kritērija nav, tad šis mērs būs tikai subjektīvs.

Visobjektīvākie mēri ir tie, kurus var izmērīt, un veikt to salīdzināšanu starp dažādiem produktiem un sistēmām, pamērām, “RAv-1-G sistēmas pieejamība (*System availability*)”, kas parāda to, kādu daļu no plānotā sistēmas darbības laika sistēma faktiski ir pieejama. To aprēķina kā attiecību starp faktisko sistēmas darbības laiku un sistēmas darbības laiku, kas noteikts sistēmas darbības grafikā. Šis mērs ir objektīvs un izmantojams salīdzināšanai ar citām sistēmām, jo aprēķinā izmantotie parametri tiek iegūti statistiski vai arī no sistēmas darbības grafika.

2.2. Kvalitātes noteikšana, izmantojot mērus no standartiem

Lai noteiktu, kuri mēri ir piemēroti sistēmas kvalitātes noteikšanai, ir praktiski jānosaka, kuri no mēriem ir objektīvi un uzticami. Apkopojot informāciju par mēru rekomendācijas līmeņiem no standarta un trīs personu viedokli par mēru objektivitāti, ar melnu krustu tiek iezīmēti tie mēri, kas ir bez zināmas uzticamības, savukārt, ar melnu rāmi – ļoti ieteicami un objektīvie mēri (skatīt 2.2. att.). Tikai 14 mēri no 122 ir ļoti ieteicami un objektīvi, no kuriem 12 ir vispārīgi izmantojami.

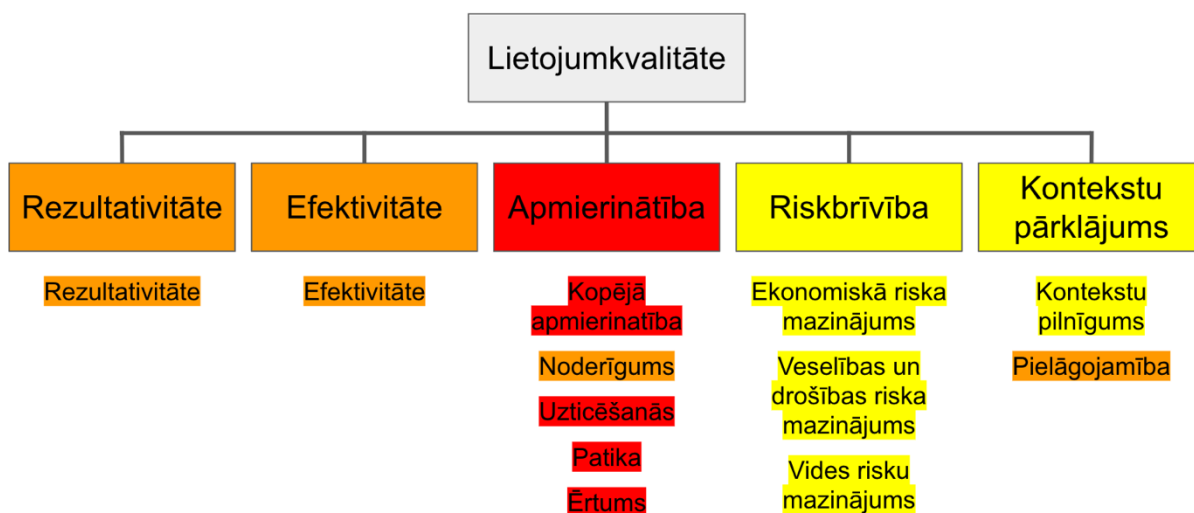
Ef-1-G	SPI-1-G	FCp-1-G	Cln-2-G	UEp-1-G	SCo-2-G	MMd-2-G
Ef-2-S	SCo-1-G	FCr-1-G	Cln-3-S	UEp-2-S	SCo-3-G	MMd-3-S
Ef-3-G	REc-1-G	FAP-1-G	UAp-1-G	UEp-3-S	SIn-1-G	MTe-1-G
Ef-4-G	REc-2-G	FAP-2-G	UAp-2-S	UIn-1-S	SIn-2-G	MTe-2-S
Ef-5-G	REc-3-G	PTb-1-G	UAp-3-S	UAc-1-G	SIn-3-S	MTe-3-S
Ey-1-G	REc-4-G	PTb-2-G	ULe-1-G	UAc-2-S	SNo-1-G	PAd-1-G
Ey-2-S	REc-5-S	PTb-3-G	ULe-2-S	RMa-1-G	SAC-1-G	PAd-2-G
Ey-3-S	REc-6-S	PTb-4-G	ULe-3-S	RMa-2-G	SAC-2-S	PAd-3-S
Ey-4-S	REc-7-S	PTb-5-G	ULe-4-S	RMa-3-G	SAu-1-G	Pln-1-G
Ey-5-S	REc-8-G	PRu-1-G	UOp-1-G	RMa-4-S	SAu-2-S	Pln-2-G
Ef-6-S	RHe-1-G	PRu-2-G	UOp-2-G	RAv-1-G	MMo-1-G	PRe-1-G
Sus-1-G	RHe-2-G	PRu-3-G	UOp-3-S	RAv-2-G	MMo-2-S	PRe-2-S
Sus-2-G	RHe-3-G	PRu-4-S	UOp-4-S	RFT-1-G	MRe-1-G	PRe-3-S
Sus-3-G	REn-1-G	PCa-1-G	UOp-5-S	RFT-2-S	MRe-2-S	PRe-4-S
Sus-4-G	CCm-1-G	PCa-2-G	UOp-6-S	RFT-3-S	MAAn-1-G	
Sus-5-G	CFI-1-S	PCa-3-S	UOp-7-S	RRe-1-G	MAAn-2-S	
Sus-6-G	CFI-2-S	CCo-1-G	UOp-8-S	RRe-2-S	MAAn-3-S	
STr-1-G	CFI-3-S	Cln-1-G	UOp-9-S	SCo-1-G	MMd-1-G	

Apzīmējumi:
~~bez zināšanas izstrāšanās~~
 ļoti ieteicams un objektīvs

2.2. att. Ieteicami un objektīvi mēri

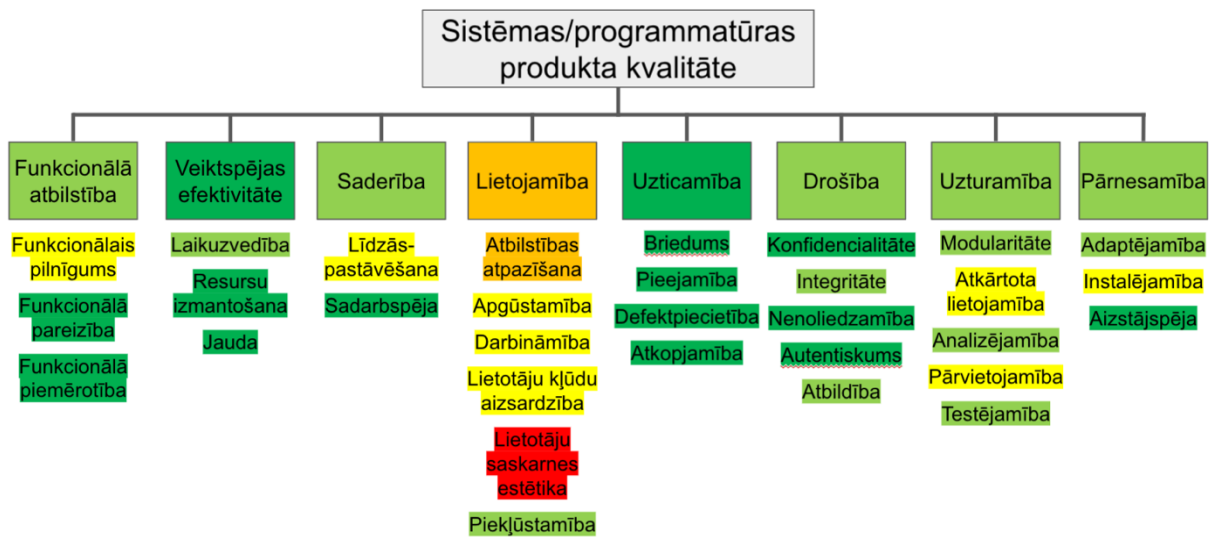
Lai noteiktu, cik lielā mērā standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 pieejamie mēri un to apakšraksturlielumi ir izmantojami produktu un sistēmas kvalitātes noteikšanā, tiek apskatīti kvalitātes modeļi, to mēri un apakšraksturlielumi.

Standartā ISO/IEC 25022:2016 lietojumkvalitātes mēriem netiek noteikts rekomendācijas līmenis. Lietojumkvalitātes modelī nav objektīvi mēri, kas būtu salīdzināmi starp dažādām programmatūrām. Ja mēri nav objektīvi, tad arī raksturlielumi nebūs objektīvi (skatīt 2.3. att.). Ar dzeltenu krāsu apzīmēti raksturlielumi, kas ir objektīvi, bet tikai konkrētam produktam un nav salīdzināmi ar citiem produktiem. Oranža krāsa apzīmē raksturlielumus, kas ietver mērus, kurus ietekmē lietotāja kvalifikācija un zināšanas, ar sarkanu krāsu apzīmēti raksturlielumi, kas ietver mērus, kas ir subjektīvi.



2.3. att. Lietojumkvalitātes modeļa raksturlielumu objektivitāte un uzticamība

Savukārt, sistēmas/programmatūras produkta kvalitātes modelī esošie raksturlielumi ir daudz objektivāki un rekomendējamāki (skatīt 2.4. att.). Standartā ISO/IEC 25023:2016 sistēmas/programmatūras produkta kvalitātes modeļa mēriem tiek noteikts rekomendācijas līmenis, kā arī liela daļa no tiem ir atzīstami par objektīviem mēriem. Ar sarkanu krāsu iekrāsoti apakšraksturlielumi, kas ir subjektīvi, bet ar oranžu krāsu iekrāsoti raksturlielumi un apakšraksturlielumi, kas ir objektīvi, taču tos ietekmē lietotāja kvalifikācija un zināšanas. Ar dzeltenu krāsu iekrāsoti apakšraksturlielumi, kas ir objektīvi, bet tikai konkrētam produktam un nav salīdzināmi ar citiem produktiem. Ar gaiši zaļu krāsu iekrāsoti apakšraksturlielumi un to raksturlielumi, kas ir objektīvi, ja eksistē industrijas kritēriji, ar tumšu zaļu krāsu apzīmēti raksturlielumi un to apakšraksturlielumi, kas ir objektīvi un salīdzināmi starp dažādām sistēmām un produktiem.



2.4. att. Sistēmas/programmatūras produkta kvalitātes modeļa raksturlielumu objektivitāte un uzticamība

3. APTAUJA

3.1. Aptaujas mērķis

Tika veikta aptauja, lai noskaidrotu, kādus mērus izmanto programmatūras izstrādes nozarē Latvijā. Aptaujai bija divi mērķi. Galvenais aptaujas mērķis bija noskaidrot, kādus kvalitātes raksturlielumus izmanto programmatūras izstrādātāji, testētāji un kvalitātes nodrošināšanas inženieri IT nozarē Latvijā, veicot programmatūras izstrādi un uzturēšanu. Otrs mērķis bija noskaidrot, vai visi standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 minētie kvalitātes raksturlielumi ir labi zināmi un izmantoti IT nozarē Latvijā.

3.2. Aptaujas apraksts

Tika aptaujāti augstākā līmeņa darbinieki, kas saistīti ar programmatūras izstrādi, testēšanu vai kvalitātes nodrošināšanu kādā programmatūras izstrādes uzņēmumā vai citas nozares uzņēmuma IT nodaļā.

Aptauja sastāvēja no trīs daļām. Pirmajā daļā bija aptaujas ievads, kas paskaidroja aptaujas mērķus un to, kā aptauja jāaizpilda.

Aptaujas otrajā daļā bija jautājumi par uzņēmuma lielumu, piemēram, kāds ir ar programmatūras izstrādi saistīto darbinieku skaits, kā arī cik no tiem ir testētāji. Tika uzdots jautājums, vai uzņēmums nodrošina programmatūras izstrādi kā pakalpojumu, kā arī tika jautāts, cik aktuālu IT projektu šobrīd ir uzņēmumā un kādas programmatūras izstrādes metodoloģijas tiek izmantotas projektos.

Pati svarīgākā trešā aptaujas daļa saturēja jautājumus par 43 apakšraksturlielumu 122 mēriem no standartiem ISO/IEC 25022:2016 un ISO/IEC 25023:2016. Katram mēram tika dots nosaukums, apraksts, formula, skaidrojums, kā arī piezīmes, ja tādas norādītas standartā. Katrā no 13 raksturlieluma mēru grupas beigām bija iespēja sniegt komentārus par konkrētu raksturlielumu.

Tā kā aptauja bija laukietilpīga, respondentiem bija iespēja aptauju aizpildīt daļēji un pēc tam to turpināt.

3.3. Aptaujas metode

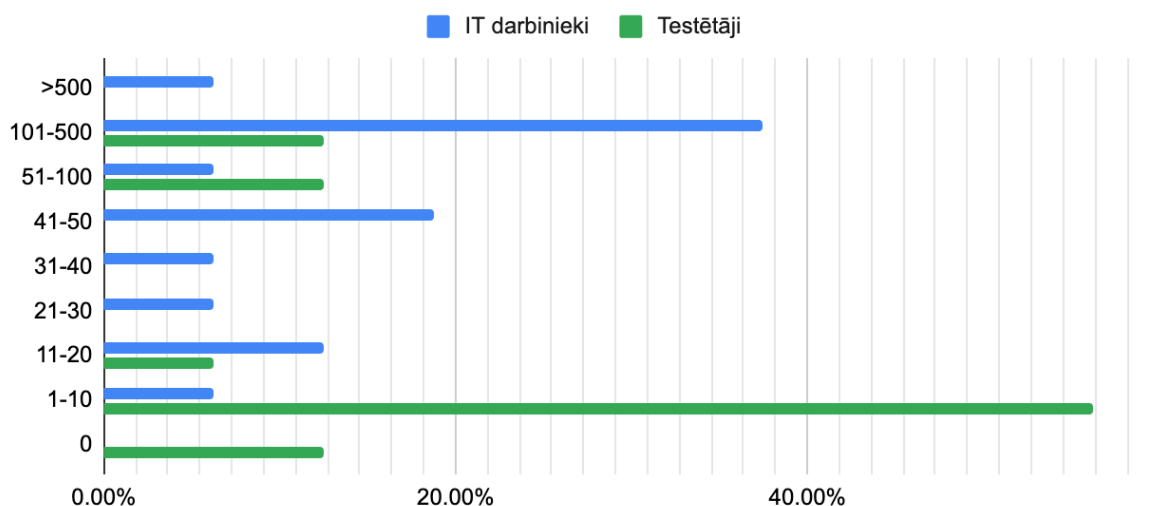
Respondenti tika aicināti aizpildīt aptauju tiešsaistē *Google Forms* tīmekļa vietnē. Katram jautājumam par kvalitātes mēru tika doti trīs atbilžu varianti: 1) Jā; 2) Nē; 3) Cits..., no kuriem bija jāizvēlas viens. Trešais atbilžu variants "Cits ..." bija atvērtā tipa atbilžu variants,

kas respondentiem ļāva norādīt konkrētu atbildi. Respondenti tika informēti, ka informācija aptaujas jautājumiem ir no standartiem ISO/IEC 25022:2016 un ISO/IEC 25023:2016. Katram mēram tika dota definīcija, izskaidrota formula un metode mēra aprēķināšanai, lai respondentiem būtu skaidrs termina skaidrojums. Tika pieņemts, ka aptaujas aizpildīšanai būs nepieciešamas aptuveni 30 minūtes. Visiem respondentiem un to uzņēmumiem tika nodrošināta konfidencialitāte un privātums.

3.4. Respondentu analīze

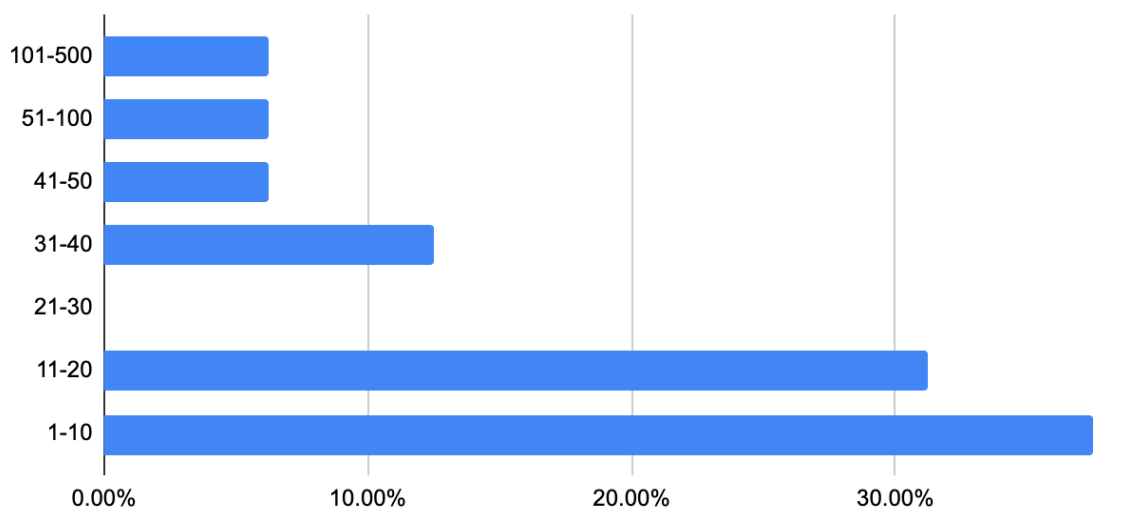
Aptaujā tika izvēlēti respondenti, kas ir darbinieki kādā no programmatūras izstrādes uzņēmumiem vai citas nozares uzņēmumā programmatūras izstrādes vai kvalitātes nodrošināšanas nodaļā. Galvenokārt tika izvēlēti testēšanas vadītāji, otrkārt - kvalitātes nodrošināšanas vai projektu vadītāji, kas pārzina programmatūras kvalitātes nodrošināšanas principus uzņēmumā.

Rezultātā tika saņemtas atbildes no 16 Latvijā esošiem uzņēmumiem, kas apkopo vairāk nekā 3000 IT darbiniekus, no kuriem gandrīz 500 ir testētāji. 50% respondentu norādīja, ka viņu uzņēmumos strādā vairāk nekā 50 IT darbinieku. Divos no aptaujātajiem uzņēmumiem nav testētāju, bet vairāk nekā pusē ir no viena līdz 10 testētājiem (skatīt 3.1. att.).



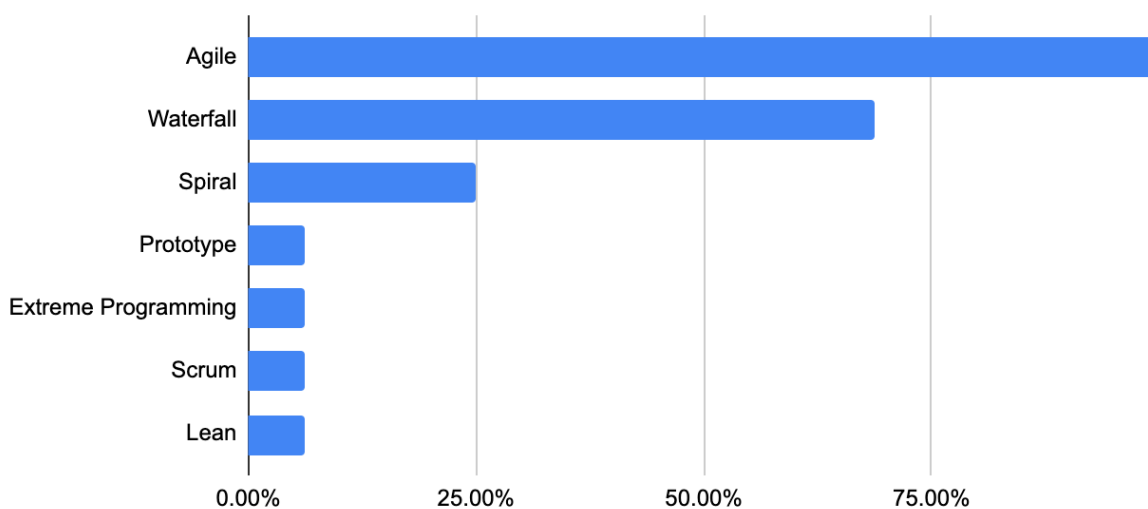
3.1. att. IT darbinieki un testētāji uzņēmumos

Tikai nedaudz vairāk nekā puse aptaujāto uzņēmumu piedāvā programmatūras izstrādi kā pakalpojumu. Gandrīz 70% no aptaujātajiem uzņēmumiem ir 1 – 20 aktuālu IT projektu (skatīt 3.2. att.).



321. att. Aktuālo projektu skaits uzņēmumā

Visos aptaujātajos uzņēmumos tiek pielietota spējā programmatūras izstrāde, un gandrīz 70% pielieto arī ūdenskrituma programmatūras izstrādi. Dažos no aptaujātajiem uzņēmumiem pielieto arī spirāles, prototipēšanas, ekstrēmās programmēšanas, scrum un lean metodoloģijas (skatīt 3.3. att.).



3.3. att. Programmatūras izstrādes metodoloģiju izmantošana

3.5. Aptaujas rezultāti

Aptaujas rezultāti parāda, ka standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 aprakstītie raksturlielumi netiek praktiski izmantoti Latvijas IT nozares uzņēmumos. Programmatūras izstrādātāju un testētāju izmantotie kvalitātes mēri ir atšķirīgi dažādos uzņēmumos, kā arī izmantotie mēri ne vienmēr ir objektīvi un ticami.

Aptaujas rezultāti ir apkopoti tabulā (skatīt 3.4. att.), kur baltā krāsā ir mēri, kas netiek izmantoti aptaujātajos uzņēmumos, bet ar sarkanu krāsu iekrāsoti mēri, kas tiek izmantoti reti - mazāk nekā 20% aptaujāto uzņēmumu. Vienu ceturto daļu no standartos apkopotajiem mēriem izmanto mazāk nekā 20% aptaujāto uzņēmumu. Ar oranžu krāsu iekrāsoti tie mēri, kurus izmanto 20 – 40% aptaujāto uzņēmumu, ar dzeltenu krāsu iekrāsoti tie mēri, kurus izmanto 41 – 59% aptaujāto uzņēmumu, ar gaiši zaļu krāsu iekrāsoti tie mēri, kurus izmanto 60 – 80% aptaujāto uzņēmumu, savukārt, tumši zaļā krāsā iekrāsoti mēri, kuri tiek izmantoti visbiežāk – vairāk nekā 80% aptaujāto uzņēmumu. Tikai 12% no standartos apkopotajiem mēriem izmanto vairāk nekā 60% uzņēmumu. Mēri, kurus aptaujātajos uzņēmumos neizmanto un kas attēlā ir baltā krāsā, ir 4, mēri, kurus izmanto reti un kas ir iekrāsoti sarkanā krāsā, ir 25, oranži – 47, dzelteni – 31, gaiši zaļi – 13 un tikai 2 ir iekrāsoti tumši zaļi un tiek izmantoti lielākajā daļā aptaujāto uzņēmumu.

Ef-1-G	SPI-1-G	FCp-1-G	CIn-2-G	UEp-1-G	SCo-2-G	MMd-2-G	
Ef-2-S	SCo-1-G	FCr-1-G	CIn-3-S	UEp-2-S	SCo-3-S	MMd-3-S	
Ef-3-G	REc-1-G	FAP-1-G	UAp-1-G	UEp-3-S	SIn-1-G	MTE-1-G	
Ef-4-G	REc-2-G	FAP-2-G	UAp-2-S	UIn-1-S	SIn-2-G	MTE-2-S	
Ef-5-G	REc-3-G	PTb-1-G	UAp-3-S	UAc-1-G	SIn-3-S	MTE-3-S	
Ey-1-G	REc-4-G	PTb-2-G	ULe-1-G	UAc-2-S	SNo-1-G	PAd-1-G	
Ey-2-S	REc-5-S	PTb-3-G	ULe-2-S	RMa-1-G	SAC-1-G	PAd-2-G	
Ey-3-S	REc-6-S	PTb-4-G	ULe-3-S	RMa-2-G	SAC-2-S	PAd-3-S	
Ey-4-S	REc-7-S	PTb-5-G	ULe-4-S	RMa-3-G	SAu-1-G	Pln-1-G	
Ey-5-S	REc-8-G	PRu-1-G	UOp-1-G	RMa-4-S	SAu-2-S	Pln-2-G	
Ef-6-S	RHe-1-G	PRu-2-G	UOp-2-G	RAv-1-G	MMo-1-G	Pre-1-G	
Sus-1-G	RHe-2-G	PRu-3-G	UOp-3-S	RAv-2-G	MMo-2-S	Pre-2-S	Apzīmējumi:
Sus-2-G	RHe-3-G	PRu-4-S	UOp-4-S	RFt-1-G	MRe-1-G	Pre-3-S	0%
Sus-3-G	REn-1-G	PCa-1-G	UOp-5-S	RFt-2-S	MRe-2-S	Pre-4-S	1- 20%
Sus-4-G	CCm-1-G	PCa-2-G	UOp-6-S	RFt-3-S	MAN-1-G		20-40%
Sus-5-G	CFI-1-S	PCa-3-S	UOp-7-S	RRe-1-G	MAN-2-S		41-59%
Sus-6-G	CFI-2-S	CCo-1-G	UOp-8-S	RRe-2-S	MAN-3-S		60-80%
STr-1-G	CFI-3-S	CIn-1-G	UOp-9-S	SCo-1-G	MMd-1-G		> 80%

3.4. att. Aptaujas rezultāti

Bieži izmantotie mēri, kas tiek izmantoti vairāk nekā 80% aptaujāto uzņēmumu, ir “Rec-1-G peļņa no ieguldījuma (*Return on investment (ROI)*)” un “PTb-1-G vidējais atbildes laiks (*Mean response time*)”. Peļņu no ieguldījuma mēra, veicot darījumanalīzi (*business analytics*). Vidējais atbildes laiks ir vidējais laiks, kas nepieciešams sistēmai, lai reaģētu uz lietotāja vai sistēmas darbību, to mēra kā vidējo aritmētisko no visiem laikiem, kas sistēmai bijuši nepieciešami, lai reaģētu uz konkrētu lietotāja vai sistēmas darbību.

4. PROGRAMMATŪRAS KVALITĀTES NOTEIKŠANA

4.1. Metodoloģija programmatūras mēru novērtēšanai

4.1.1. Literatūras izpēte

Lai veiktu lietojumkvalitātes un sistēmas/programmatūra produkta kvalitātes mēru novērtēšanu, ir nepieciešami novērtēšanas kritēriji. Sākotnēji tika veikta informācijas meklēšana tīmekļa meklētājprogrammā *Google Scholar*, kas veic meklēšanu zinātniskās literatūras avotos, kā arī meklētājprogrammā *Google*. Veicot meklēšanu tika izmantoti tādi atslēgas vārdi kā “*evaluation of software metrics*” un “*software metrics reliability*”.

K. *Chopra* un M. *Sachdeva* raksta “*Evaluation of Software Metrics for Software Projects*” [11] mērķis ir izveidot rīku, kas nosaka programmatūras koda kvalitāti un automātiski veic pirmkoda sarežģītības mērīšanu. Sākumā tiek izvēlēti programmatūras mēri, lai nodrošinātu programmatūras projekta kvalitāti, kas tiek iedalīti kategorijās – uz izmēru balstīti mēri, atkarību mēri, uz mantošanu balstīti mēri, mēri, kas parāda projekta lasāmību un mēri, kas nosaka projekta objektivitāti. Pēc tam ar rīku *NDepend*, kas ir statistiskās analīzes rīks .NET pārvaldītajam kodam, tiek veikta mēru aprēķināšana.

Dr. L. *Rosenberg*, T. *Hammer* un J. *Shaw* rakstā “*Software Metrics and Reliability*” [8] tiek apstiprināts tas, ka eksistē mēri uzticamības novērtēšanai, un tos var izmantot sākot jau ar prasību definēšanas posmu. Kā arī tiek uzsvērts tas, ka katrā projekta posmā ir iespējams veikt uzticamības novērtēšanu, bet jo agrākā posmā tiek atklātas nepilnības, jo vieglāk tās ir labot.

H. *Barkmann*, R. *Lincke* un W. *Löwe* darba “*Quantitative Evaluation of Software Quality Metrics in Open-Source Projects*” [9] ietvaros tiek izstrādāts rīks, lai savāktu lielu apjomu ar kvantitatīviem datiem no atvērtā pirmkoda Java projektiem. Otra lieta ir novērtējamo mēru skaita samazināšana, jo pastāv korelācija starp atsevišķiem mēriem, norādot, ka daži no tiem mēra vienas īpašības.

M. *Jah* rakstā “*Software Metrics – Usability and Evaluation of Software Quality*” [17] atsaucas gan uz ISO, gan IEEE standartiem. Šajā darbā ir pētīta, analizēta un pārskatīta programmatūras kvalitāte, programmatūras mēri un daži programmatūras kvalitātes mēru pielietojumi. Tiek parādīts ka, piemēram, programmas koda rindu skaits var noteikt programmatūras kvalitātes līmeni, aprēķinot kļūdu skaitu un koda rindu skaita attiecību. Jo mazāka attiecība, jo augstāka būs programmatūras kvalitāte. Raksta autors raksta, ka labam programmatūras mēram jābūt spējai paredzēt programmatūras izstrādes procesu. Tam ir jābūt vienkāršam, viegli definējamam, skaidram, derīgam un sapratīgam, kā arī tam ir jābūt mērķim.

Lielākoties atrastie materiāli apraksta to, kā novērtēt pašu programmatūras koda kvalitāti, bet ne kā novērtēt mērus, ko izmanto, lai novērtētu kopējo programmatūras kvalitāti. Turpinot meklēšanu, tika atrasta informācija arī par pašu mēru novērtēšanu.

M. F. *Bertoa* un A. *Vallecillo* raksts “*Usability metrics for software components*” [16] ir par lietojamības kvalitātes mēriem. Tie tiek definēti pamatojoties uz ISO 9126 kvalitātes modeli. Mēri pārsvarā ir netieši, definēti bez atsauces uz atribūtiem, ko tie mēra. Tie ir tikai definēti un bez pamatojuma piešķirti apakšraksturlielumiem. ISO 9126 mēri tiek piešķirti tikai vienam kvalitātes raksturlielumam, kaut gan teorētiski viens atribūts var ietekmēt vairākus kvalitātes raksturlielumus, piemēram, komponentes lielums ir saistīts gan ar uzturamību, gan apgūstamību. Darbā tiek aprakstīta programmatūras novērtēšanas koncepcija un lietojamības novērtēšanai nepieciešamie mēri. Rezultātā tiek definēts lietojamības novērtēšanas mēru kopums, nosakot komponentu atribūtus, ko tie mēra, kā arī to, kā tie ietekmē un nosaka dažādus lietojamības kvalitātes apakšraksturlielumus. Pēc autoru viedokļa un pieredzes tiek noteikta saistība starp atribūtiem un izmantojamības apakšraksturlielumiem, bet tas ir tikai sākums turpmākiem pētījumiem, lai noteiktu katra atribūta un apakšraksturlieluma pāra ietekmes pakāpi, un to saistību pārbaudītu statistiski.

Programmatūras novērtēšanā katrs pielietotais mērs ir jāapstiprina, lai tas būtu derīgs. K.P. *Srinivasan* un T. *Devi* rakstā “*Software Metrics Validation Methodologies in Software Engineering*” [13] tiek aplūkota programmatūras mēru novērtēšanas metodoloģija, kas sastāv no divu veida validācijām – teorētiskās un empīriskās. Teorētiskā validācija apstiprina, ka mērījums nepārkāpj mērīšanas elementam nepieciešamās īpašības, bet empīriskā validācija apstiprina to, ka atribūtu izmērītās vērtības atbilst vērtībām, kuras paredzētas modeļos, kas ietver atribūtus. Teorētiskās metodes uzdevums ir pierādīt to, ka mērs mēra to, ko tas mēra, bet empīriskā metode apstiprina mēra derīgumu vai nederīgumu. Teorētiskajā validācijā izmanto mērījumu reprezentatīvo teoriju un uz īpašībām balstītas pieejas, taču līdz šim vēl nav pieņemts standarts tam, kā teorētiski apstiprināt programmatūras mērus.

L. *Westfall* rakstā “*12 Steps to Useful Software Metrics*” [15] ir aprakstīts praktisks process, kā izveidot un pielāgot programmatūras mērus, kas koncentrējas uz mērķiem un informācijas vajadzībām. Process nodrošina praktisku un sistemātisku programmatūras mēru izvēles, noformēšanas un ieviešanas metodi no sākuma līdz beigām. Kā izvēlēties mērus, kas ir piemēroti uzņēmumam? Pirmajos četros soļos tiek identificētas ieinteresētās personas, kam mēri ir interesējoši un izmantota mērķa/jautājuma/mēra paradigma, lai izvēlētos programmatūras mērus, kas atbilst identificēto klientu vajadzībām. No 5. līdz 10. solim ir aprakstīts mēru projektēšanas un pielāgošanas process, kas ietver definīciju, modeļu, etalonu,

mērķu, ziņošanas mehānismu un papildu klasifikatoru izstrādes un pielāgošanas procesu. Pēdējie divi soļi ir saistīti ar ieviešanas problēmām, piemēram, datu vākšanu un cilvēcisko faktoru ietekmes samazināšanu.

Pēc vairāku rakstu analīzes, autoraprāt, mēru novērtēšanai vispiemērotākā ir rakstā “*12 Steps to Useful Software Metrics*” [15] aprakstītā novērtēšanas metodoloģija, un to var pielietot, lai novērtētu standartos aprakstītos mērus. Rakstā minēts, ka šī metodoloģija nodrošina praktisku un sistemātisku programmatūras mēru izvēles, noformēšanas un ieviešanas metodi. Lai gan standartos mēri jau ir izvēlēti un aprakstīti, ir svarīgi novērtēt to noformējumu un standartizēt to ieviešanu un izmantošanu. Šī 12 soļu metodoloģija ļauj uz mēriem paskatīties no citas puses, novērtējot to nepieciešamību un aprakstot to izmantošanu, gan novērtējot, gan salīdzinot dažādas programmatūras.

4.1.2. 12 soļi, lai iegūtu derīgus mērus programmatūras novērtēšanai

Pēc rakstu analīzes vispiemērotākais mēru novērtēšanai ir raksts “*12 Steps to Useful Software Metrics*” [15]. Rakstā ir aprakstīti divpadsmit soļi, lai izveidotu un pielāgotu mērus programmatūras novērtēšanai.

Pirmajā solī identificē personas, kam mērs būtu interesējošs. Tās ir personas, kas pieņem lēmumus vai rīkojas, pamatojoties uz mēriem, kā arī tie, kam mēru sniegtā informācija ir nepieciešama. Ieinteresētās personas ir projektu vadītāji, inženieri, testētāji, dažādi speciālisti, klienti un gala lietotāji. Mērījumu veikšana ir laikietilpīga un dārga, tāpēc mērus, bez ieinteresētām personām nav vērts mērīt un uzturēt.

Otrajā solī tiek noteikts viens vai vairāki izmērāmi mērķi. Mērķi tiek noteikti, lai tos izmantotu mērķis/jautājums/mērs paradigmā. Uzņēmuma līmenī parasti izvirza augsta līmeņa stratēģiskus mērķus, piemēram, uzturēt augstu klientu apmierinātības līmeni vai sasniegt paredzētos ieņēmumu vai peļņas mērķus. Projekta līmenī parasti izvirza mērķus, kas saistīti ar projekta vadību un kontroli, piemēram, programmatūras piegāde ar nepieciešamo kvalitātes vai veikspējas līmeni. Konkrēta uzdevuma līmenī tiek noteikti mērķi, kas uzsver uzdevuma veiksmes faktorus, ko var izteikt kā uzdevuma ieejas un izejas kritērijus. Programmatūras novērtēšanas mēriem ir jāsniedz informācija, kas nepieciešama programmatūras projektu pārvaldīšanai un programmatūras inženierijas procesu un pakalpojumu uzlabošanai. Uzņēmuma, projekta un uzdevuma mērķi tiek noteikti iepriekš, un mēri pēc tam tiek izvēlēti, pamatojoties uz iepriekš noteiktajiem mērķiem. Mērus izmantoto, lai noteiktu efektivitāti

mērķu sasniegšanā. Izvēloties mērus, ir jāuzklausā visu projekta ieinteresēto personu mērķi un problēmas, tas palīdzēs definēt mērus, kas nepieciešami uzņēmumam vai komandai.

Trešajā solī definē jautājumus, uz kuriem jāatbild, lai nodrošinātu katra mērķa sasniegšanu. Piemēram, ja mērķis ir piegādāt programmatūru bez defektiem, varētu definēt šādus jautājumus:

- a. Vai programmatūras produkts ir atbilstoši notestēts?
- b. Cik daudz defektu joprojām nav atklāti?
- c. Vai visi zināmie defekti ir novērsti?

Ceturtajā solī tiek izvēlēti mēri, kas sniedz nepieciešamo informāciju, lai atbildētu uz iepriekšējā solī definētajiem jautājumiem. Katram izvēlētajam mēram ir skaidrs mērķis atbildēt uz vienu vai vairākiem jautājumiem, kas jāatbild, lai noteiktu, vai uzņēmuma, projekta un uzdevuma mērķi tiek sasniegti. Izvēloties mērus, ir jādomā praktiski, reālistiski un pragmatiski. Sākuma labāk ir izvēlēties tādus mērus, ko iespējams iegūt uzreiz, kad tie ir apkopoti, var turpināt papildināt mērus ar jauniem. Ir svarīgi atcerēties, ka mēri neatrisina programmatūras problēmas, to dara cilvēki, mēri ir rādītāji, kas sniedz informāciju, lai cilvēki varētu pieņemt apzinātus lēmumus un veikt saprātīgas izvēles. Mērs veic četras funkcijas:

- a) palīdz izprast programmatūru,
- b) ļauj novērtēt programmatūru atbilstoši noteiktajiem standartiem un mērķiem,
- c) sniedz nepieciešamo informāciju, lai kontrolētu resursus un procesus, ko izmanto programmatūras izstrādei,
- d) ļauj prognozēt programmatūras entītijū atribūtus.

Visaptverošam programmatūras mēram būtu jāveic visas šīs funkcijas. Katra mēra prasības var definēt kā vienu no šīm četrām funkcijām. To var izteikt ar šādu šablonu: Izprast/novērtēt/kontrolēt/prognozēt [mērāmais mērs], lai sasniegtu [noteiktais mēra mērķis]. Piemēram novērsto zināmo defektu procentuālais daudzums varētu tikt definēts tā ka, lai novērtētu izstrādes laikā novērsto zināmo defektu procentuālo daudzumu, lai visi zināmie defekti tiktu novērsti pirms programmatūras piegādes. Ja mēram ir skaidri definētas prasības, tas sniedz vairākas priekšrocības:

- a) palīdz nodrošināt precīzi definētus mērus, kas balstīti uz ieinteresēto personu mērķiem,
- b) novērš pārpratumus par to, kā mērus paredzēts izmantot,
- c) dara zināmu mēru nepieciešamību, kas var palīdzēt iegūt resursus datu vākšanas un ziņošanas mehānismu ieviešanai,
- d) nodrošina pamatu mēra izstrādei.

Piektajā solī tiek precizētas definīcijas, lai novērstu interpretāciju atšķirības. Var izvēlieties definīcijas, kas atbilst uzņēmuma mērķiem, vai izmantot tās par pamatu jaunu definīciju izveidošanai. Tā pat var izmantot definīcijas, kas nozarē jau pastāv, piemēram, definīcijas no IEEE programmatūras inženierijas terminoloģijas vārdnīcas vai definīcijas, kas atrodamas programmatūras inženierijas literatūrā.

Sestajā solī tiek izvēlētas funkcijas mēru mērīšanai jeb tas, kā mēri tiks aprēķināti. Mērus iedala vienkāršos un atvasinātos mēros. Vienkārši mēri tiek aprēķināti tieši, un parasti tā aprēķināšanas funkcija satur vienu mainīgo. Atvasināti mēri tiek aprēķināti, izmantojot vienkāršus mērus vai citus atvasinātus mērus kombinācijā ar vienādojumiem vai algoritmiem. Programmatūras novērtēšanas funkcijām ir jābūt pragmatiskām, ir jāizvēlas svarīgākie aspekti. Mēra funkcijā nav jāiekļauj pilnīgi visi ietekmējošie faktori, lai tā nekļūtu pārāk sarežģīta. Pēc nepieciešamības, laika gaitā to var papildināt. Funkciju izvēlei ir divas metodes – izmantot jau esošu vai veidot jaunu. Bieži vien nav vajadzīgs radīt jaunu funkciju, bet, veicot nelielu izpēti, var atrast nepieciešamo, un to pielāgot konkrētajam gadījumam. Otra metode ir izveidot jaunu funkciju. Veidojot jaunu funkciju, ir jānodrošina, ka tā ir saprotama ieinteresētajām personām, un ir jāpierāda, ka tā ir derīga tam, ko nepieciešams izmērīt. Bieži vien derīgumu var pierādīt tikai ar statistikas metodēm. Piemēram, mērs neplānotu sistēmas pārtraukumu ilgumam varētu tikt aprēķināts kā vienkāršs mērs, nosakot pārtraukuma minūtes kalendārajā mēnesī, ja tiek novērtēta vienā vietā instalēta programmatūra. Savukārt, ja mērķis ir salīdzināt dažādas programmatūras versijas dažādās vidēs, varētu izmantot funkciju pārtraukuma minūtēm uz 100 darbības mēnešiem. Ja skatās uz to, kā neplānots sistēmas pārtraukuma ilgums ietekmē klientu vai lietotāju, tad to varētu atspoguļot kā funkciju, kas nosaka neplānotas sistēmas pārtraukuma minūtes gadā konkrētā vidē. Variantu ir daudz, tāpēc ir svarīgi noteikt konkrētu funkciju, katram mēram.

Septītajā solī mēru funkcijas tiek sadalīta pamata mēros, un tiek definēta mērīšanas metode, kas tiek izmantota, lai katram pamata mēram piešķirtu vērtību. Ir mērīšanas metodes, kurās tiek izmantotas standartizētas mērvienības, piemēram, metri, kilogrami, litri, eiro, stundas, dienas. Citu mērīšanas metožu pamatā ir skaitīšanas kritēriji, kas ir lietu ar noteiktām īpašībām skaits. Mērīšanas metodei var izmantot arī citus kritērijus, kas jau tiek izmantoti uzņēmumā.

Astotajā solī tiek noteikti rezultātu interpretēšanas nosacījumi. Kad ir zināms ko un kā mērīt, ir jānosaka tas, kā izmantot iegūtos rezultātus. Standartā ISO/IEC 15939 Sistēmu un programmatūru izstrādes – mērīšanas process (*Systems and software engineering — Measurement process*) rezultātu interpretēšanas nosacījumi ir sliekšņi, mērķi vai paraugi, ko

izmanto, lai noteiktu nepieciešamību rīkoties, turpinātu pētīt vai aprakstītu uzticamības līmeni iegūtajam rezultātam. Citiem vārdiem sakot, ir nepieciešams noteikt rezultātu interpretēšanas nosacījumus, kas palīdzēs interpretēt mēru iegūtos rezultātus. Kontroles tipa mēri tiek izmantoti, lai pārraudzītu programmatūras procesus, produktus un pakalpojumus un identificētu vietas, kurās nepieciešams veikt izmaiņas vai pārvaldību. Kontroles mēri brīdina, ja kaut kas nenotiek kā plānots, savukārt, ja viss notiek kā paredzēts, tad rezultāts tiek interpretēts tā, ka viss ir kārtībā, un nav jāveic papildus darbības. Kontroles tipa mēru rezultātu interpretēšanas nosacījumi ir, piemēram, sliekšņi, novirzes un robežas. Novērtēšanas tipa mēri tiek izmantoti, lai pārbaudītu un analizētu rezultātu informāciju. Tā ir daļa no rezultātu interpretēšanas procesa. Novērtēšanas tipa mēriem rezultātu interpretēšanas process palīdz definēt to, kas ir labi. Izprašanas un prognozēšanas tipa mēri ir uzticamības līmenis konkrētam rezultātam pēc ISO/IEC 15939 standarta definīcijas. Tas nosaka to, cik lielā mērā var ticēt iegūtajam rezultātam, ka tas atbilst realitātei un prognozētais rezultāts atspoguļo nākotnes vērtības. Lai to noteiktu, var aprēķināt statistisko ticamības intervālu, bet var iegūt arī subjektīvāku ticamības līmeni, piemērojot dažādus faktorus. Piemēram, kāda ir datu integritāte un precizitāte, ja darbinieki uzdevumam patērēto laiku aizpilda veicot uzdevumu vai tikai nedēļas beigās novērtē, cik daudz laika patērēts katram uzdevumam.

Devītajā solī tiek noteikts tas, kā tiks ziņots par iegūtajiem rezultātiem. Tas ietver pārskata formāta, datu iegūšanas un ziņošanas cikla, ziņošanas mehānisma un izplatīšanas un pieejamības noteikšanu.

Desmitajā solī mēriem tiek noteikti papildus klasifikatori. Labs mērs ir vispārīgs mērs, tas nozīmē, ka mērs ir derīgs visiem papildus klasifikatoriem. Piemēram, neplānotu pārtraukumu ilgumu var noteikt visai programmatūrai kopumā, konkrētam modulim vai konkrētai programmatūras versijai. Papildus klasifikatori sniedz papildus informāciju, kas nepieciešama dažādiem mērījumu rezultātu skatiem. Mērījumu rezultātiem jāsaturs ne tikai metriskie pamata dati, bet arī papildus dati, lai varētu nošķirt iegūtos rezultātus pēc papildus klasifikatoriem.

Vienpadsmitais solis nosaka datu iegūšanas procesu. Tas kādus datus nepieciešams vākt ir aprakstīts septītajā un desmitajā solī. Šajā solī ir jāapkopo dati, kas nepieciešami mēru pamata un papildus klasifikatoru nodrošināšanai. Vairumā gadījumu datu īpašnieks ir tas, kuram būtu jāievāc dati. Datu īpašnieks ir persona, kurai ir tieša piekļuve datu avotam, parasti tā ir arī atbildīga par datu ģenerēšanu. Datu īpašnieki var būt vadītāji, kas atbild par plānu un budžetu, inženieri, kas atbild par uzdevumam patērēto laiku, kontroles un defektu datiem un defektu cēloņiem, testētāji, kas atbild par testpiemēriem un problēmu ziņojumiem un citi. Lai datu

vākšana būtu veiksmīga, ir svarīgi cilvēkiem, kas ir atbildīgi par datu ievākšanu, tam atvēlēt pietiekami daudz laika, kā arī veikt apmācību tam, kā tieši tas jādara. Datiem ir jābūt objektīviem, tie katru reizi jāievāc vienai un tai pašai personai vienā un tajā pašā veidā. Datiem ir jābūt nepārprotamiem, ja divi dažādi cilvēki ievāks noteiktus datus, tiem ir jāsakrīt. Datiem ir jābūt piemērotiem, datu vākšanas procesam ir jābūt pietiekami vienkāršam, lai netraucētu citus darbus, un tam ir jābūt kā daļai no kopējā procesa. Datiem ir jābūt pieejamiem, piekļuvei datiem ir jābūt vieglai, lai dati būtu noderīgi un tiktu izmantoti. Pat ja dati tiek iegūti manuāli, tie ir jāiekļauj datu bāzē. Ja tas ir iespējams, datu vākšanas procesu var automatizēt. Tas samazina nepieciešamo laiku, kas tiek pavadīts, lai ievāktu datus, kā arī tiek izslēgtas cilvēciskās kļūdas, un dati ir pieejamāki, jo ir elektroniski pieejami.

Divpadsmitajā solī tiek ņemts vērā cilvēciskais faktors, jo bez tā mērs nevar būt pilnīgs. To vai mērs ir derīgs, katrs uzņēmums nosaka individuāli. Ja kaut kas tiek mērīts, tad automātiski tiek pieņemts, ka tam ir nozīme. Izvēloties mērus, ko pielietot, ir svarīgi noteikt kādu uzvedību vēlas veicināt kā arī jānosaka, kāda vēl uzvedība var rasties, lietojot mēru vai lietojot to nepareizi. Lai izvairītos no problēmām, ko var radīt cilvēciskais faktors var ievērot šādus pamatnoteikumus:

- a. Nemērīt indivīdus, individuālie produktivitātes mēri ir klasisks piemērs. Piemēram, labākajiem cilvēkiem tiek dots vissarežģītākais uzdevums, kā arī tiek sagaidīts, ka viņš konsultēs arī citus. Ja produktivitāte tiktu mērīta uzrakstīto koda rindiņu skaitā stundā, tad cilvēks koncentrēsies tikai uz savu darbu, tādā veidā kaitējot komandai. Vai var gadīties, ka tiks rakstītas nevajadzīgas koda rindiņas. Veicot programmatūras novērtēšanu, jākoncentrējas tieši uz procesiem un produktiem, nevis cilvēkiem.
- b. Neizmanto mēru kā "nūju", ja mērs tiks izmantots pret kādu cilvēku vai grupu, pēc tam vairs nebūs iespējams iegūt derīgus datus.
- c. Neignorēt datus, jo ignorējot datus, kad tiek pieņemti lēmumi, mērs tiek sabojāts.
- d. Neizmanto tikai vienu mēru, jo programmatūra ir sarežģīta un daudzšķautņaina. Koncentrēšanās uz vienu atsevišķu mēru var izraisīt situāciju, ka mērāmais atribūts uzlabojas, bet tā rezultātā pasliktinās kāds cits. Ir jābūt līdzsvaram starp izmaksām, kvalitāti un laiku, lai apmierinātu visas klienta vajadzības.
- e. Mēri jānosaka atbilstoši izvirzītajiem mērķiem. Tādā veidā visa uzmanība tiek veltīta tam, kas ir svarīgs.

- f. Sniegt atgriezenisko saiti komandai par datu izmantošanu un iegūtajiem rezultātiem. Tas parāda nepieciešamību vākt datus un izglīto komandu.
- g. Iegūt komandas atbalstu, iesaistot komandu mērķu un mēru noteikšanas procesā. Tas ļaus labāk izprast novērtēšanas procesu un uzlabot kvalitāti.

4.1.3. SMART novērtēšanas kritēriji

Novērtēšanas kritēriji SMART [14] ir saīsinājums no angļu valodas konkrēts, izmērāms, pievienot vērtību un rīkoties, reālistisks un piemērojams, savlaicīgs (*Specific, Measurable, Add value and actionable, Realistic and Relevant, Timely*). Katra kritērija apraksts un novērtējumi ir aprakstīti 4.1. tabulā.

4.1. tabula

SMART kritēriji

SMART	Apraksts	0	1	2	3
		Vāji	Vidēji	Labi	Teicami
S	Mēriem jābūt konkrētiem un mērķētiem uz mērīšanai paredzēto apgabalu.	Nav vērsti uz mērīšanai paredzēto apgabalu.	Neformāli adresē un aptver mērīšanai paredzēto apgabalu.	Adresē un aptver mērīšanai paredzēto apgabalu.	Pareizi adresē un aptver mērīšanai paredzēto apgabalu.
M	Mēriem jāļauj savākt precīzus un pilnīgus datus.	Mēru aprēķināšanai nepieciešamo elementu nepilnīga vai slikta definīcija.	Mēru aprēķināšanai nepieciešamo elementu formāla definīcija bez piemēriem un rīkiem.	Mēru aprēķināšanai nepieciešamo elementu definīcija ar piemēriem, bet bez rīku ieteikumiem.	Pilnīga un izsmeļoša mēru aprēķināšanas elementu definīcija ar vērtīgiem piemēriem un rīku ieteikumiem.
A	Mēriem jābūt viegli saprotamiem	Nav norādījumi par “slikto”	Pamata norādījumi par “slikto”	Vidēji norādījumi par “slikto”	Tendenču analīze skaidri

	un jānorāda “labais” un “sliktais” virziens, lai noteiktu, kad nepieciešams rīkoties.	un “labo” virzienu tendenču analīzē.	un “labo” virzienu tendenču analīzē.	un “labo” virzienu tendenču analīzē.	parāda “slikto” un “labo” virzienu.
R	Jāņem vērā tikai tie mēri, kas patiešām ir svarīgi un nepieciešami uzņēmumam.	Nesniedz noderīgus ieteikumus par saistītā procesa uzlabojumiem.	Sniedz nelielus ieteikumus par saistītā procesa uzlabojumiem.	Sniedz vidējus ieteikumus par saistītā procesa uzlabojumiem.	Sniedz kritiskus ieteikumus par saistītā procesa uzlabojumiem.
T	Mēri ir noderīgi, ja saistītā informācija tiek izmantota savlaicīgi.	Datu vākšanai noteiktais laiks neļauj veikt ziņošanas un datu analīzes uzdevumus, kā paredzēts.	Datu vākšanai noteiktais laiks ļauj veikt minimālu pārskatu sniegšanu un datu analīzi.	Datu vākšanai paredzētais laiks ļauj rentabli veikt pārskatus un datu analīzi.	Datu vākšanai noteiktais laiks viegli ļauj veikt pārskatus un datu analīzi.

4.1.4. Programmatūras mēru novērtēšanas metodoloģija

Standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 ir apkopoti mēri, to funkcijas, apraksti un definīcijas. Tas atbilst ceturtajam soli, kurā tiek izvēlēts mērs, piektajam solim, kurā tiek noteiktas definīcijas un sestajam un septītajam solim, kuros tiek noteikta mēra formula un mainīgie.

Atbilstoši divpadsmit soļu metodoloģijas pirmajam solim, mēram no standarta sākumā tiks identificētas personas, kam mērs ir interesējošs, svarīgs un noderīgs. Ja mēram ir iespējams norādīt ieinteresētās personas, tad var turpināt ar otro soli, jo mērs kādam ir nepieciešams, bet

ja mēram nav ieinteresētās personas, tad tas nav nepieciešams un to nav vērts pielietot programmatūras novērtēšanai, jo tā rezultāts nav vajadzīgs ieinteresētajām personām.

Kad ieinteresētās personas ir noteiktas, ir jānosaka mērķi un jautājumi, lai nodrošinātu mērķu sasniegšanu, kā aprakstīts otrajā un trešajā solī. Konkrētam mēram tiek definēts uzņēmuma, projekta vai uzdevuma mērķis, kas atbilst ieinteresēto personu vajadzībām, kā arī jautājumi, kas jāatbild, lai nodrošinātu mērķa sasniegšanu. Ja mēram ir iespējams definēt atbilstošu mērķi un jautājumus, var pāriet pie nākamā punkta, savukārt, ja mēram nav iespējams definēt mērķus, tad šāds mērs nav jāpielieto programmatūras novērtēšanā.

Nonākot pie ceturta soļa, jānosaka, vai mērs veic kādu no mēru funkcijām:

- palīdz izprast programmatūru,
- ļauj novērtēt programmatūru atbilstoši noteiktajiem standartiem un mērķiem,
- sniedz nepieciešamo informāciju, lai kontrolētu resursus un procesus, ko izmanto programmatūras izstrādei,
- ļauj prognozēt programmatūras entīciju atribūtus,

izsakot to kā apgalvojumu: Izprast/novērtēt/kontrolēt/prognozēt [mērāmais mērs], lai sasniegtu [noteiktais mēra mērķis]. Ja mērs veic kādu no funkcijām, tad var turpināt ar nākamo soli.

Piektajā solī tiek noteiktas definīcijas. Standartā jau katram mēram ir sava definīcija, tāpēc tā ir jānovērtē. Definīcijai ir jābūt atbilstošai, skaidri definētai, saprotamai un nepārprotamai.

Sestajā un septītajā solī, tiek analizēta mēra funkcija un tās mainīgie. Tā pat kā pašai mēra definīcijai, arī funkcijai un tās mainīgajiem jābūt saprotamiem un nepārprotamiem. Funkcijai ir jā satur nepieciešamie kritēriji, un ir jābūt skaidri noteiktam, kā tie tiek iegūti, kā arī ir jābūt noteiktām mainīgo mērvienībām.

Kad ir skaidrs, kā iegūt mēra rezultātu, ir jānosaka, vai iegūtais rezultāts ir labs vai slikts, un ko tas parāda. Astotajā solī tiek noteikts tas, kā iegūto rezultātu izmantot un interpretēt. Atkarība no mēra, nosaka, piemēram, sliekšņus, novirzes, robežas, definē, kad rezultāts ir labs vai nosaka uzticamības līmeni konkrētam rezultātam.

Devītais solis pieprasa aprakstīt rezultātu atskaites formu, datu iegūšanas veidu un biežumu, mērījumu veikšanas biežumu kā arī rezultātu pieejamību. Šajā darbā tiek analizēti paši mēri, tāpēc šis punkts sīkāk netiks apskatīts. Tā pat arī tas varētu mainīties atkarībā no konkrētā uzņēmuma pieņemtajiem standartiem, bet tam nebūtu jāietekmē mēra iegūto rezultātu derīgums.

Mēra funkcijas un tās elementu analīze tika veikta sestajā un septītajā solī. Desmitajā solī tiek noteikti nepieciešamie papildus klasifikatori, bet tā kā mērs un funkcija jau ir definēti

standartā, tie jau tika apskatīti iepriekš. Ja nepieciešami vēl kādi citi rezultātu skati, standartā pieejamo formulu var papilināt ar papildus klasifikatoriem.

Vienpadsmitajā solī tiek noteikts, vai funkcijai nepieciešamie dati, lai veiktu aprēķinus ir reāli iegūstami, un to vākšanai nepieciešamais laiks un resursi ir atbilstoši iegūtajiem rezultātiem. Vēl šis solis nosaka, kurš būs atbildīgs par datiem un to iegūšanu. Šī darba ietvaros varētu teikt, ka tā būs kāda no ieinteresētajām personām, kurai konkrētais mērs ir interesējošs, jo visdrīzāk katrā uzņēmumā tas var atšķirties.

Divpadsmitajā un noslēdzošajā solī tiek apskatīts cilvēcisks faktors. Tas var ietvert standartos pieejamo informāciju par to, vai mērs ir konkrēts vai vispārīgs, un vai tas ir ļoti ieteicams, ieteicams vai izmantojams pēc lietotāja ieskatiem. Tā pat var tikt ņemts vērā iepriekš noteiktais objektivitātes līmenis un iegūtie aptaujas rezultāti.

Papildus 12 soļu metodoloģijai, mēri tiks novērtēti pēc SMART kritērijiem. Tādā veidā vizualizējot atbilstību kritērijiem un novērtējot mēru. Labam mēram būtu visos kritērijos jāatbilst novērtējumam 2 vai 3.

Ja kādā punktā mērs neatbilst 12 soļu metodoloģijai, vai arī standartā trūkst informācija, bet to varētu uzlabot vai papildināt, autore papildinās konkrēto soli ar saviem ieteikumiem *slīprakstā*.

Turpmākajās apakšnodaļās tiks novērtēti populārākie lietojumkvalitātes un sistēmas/programmatūras produkta kvalitātes mēri pēc veiktās aptaujas rezultātiem.

4.2. Lietojumkvalitātes mēru novērtēšana

Lietojumkvalitāte ir pakāpe, kādā produktu vai sistēmu var izmantot konkrēti lietotāji, lai apmierinātu viņu vajadzības, lai sasniegtu konkrētus mērķus ar rezultativitāti, efektivitāti, apmierinātību un bez riska noteiktā lietošanas kontekstā [2]. Lietojumkvalitāti raksturo pieci raksturlielumi – rezultativitāte, efektivitāte, apmierinātība, riskbrīvība un kontekstu pārklājums.

4.2.1. Riskbrīvība

Riskbrīvība ir pakāpe, kādā produkta vai sistēmas kvalitāte mazina vai novērš iespējamus riskus ekonomiskajam stāvoklim, cilvēka dzīvībai, veselībai vai videi [2]. Standarts ISO/IEC 25022:2016 riskbrīvību iedala trīs apakšraksturlielumos – ekonomiskā riska mazinājums, veselības un drošības riska mazinājums un vides risku mazinājums.

4.2.1.1. Ekonomiskā riska mazinājums

Ekonomiskā riska mazinājums nosaka kvalitātes ietekmi uz ekonomiskiem mērķiem, kas ir saistīti ar finanšu stāvokli, efektīvu darbību, komerciālo īpašumu, reputāciju vai citiem resursiem, kas varētu būt apdraudēti vai sniegtu iespējas [2]. Riska mazināšana var tikt izmantota, lai mazinātu nepieņemamu ekonomisko rezultātu risku, kas var rasties sliktas programmatūras kvalitātes dēļ. Sagaidāmās mēru vērtības var novērtēt, pamatojoties uz faktiskajām vērtībām no vēsturiskajiem datiem. Ekonomiskā riska mazinājuma mēru rezultātus var būt sarežģīti interpretēt, jo var būt grūti izolēt produkta vai sistēmas kvalitātes ieguldījumu katrā mērā, jo daudzi citi faktori ietekmē šos mērus. Mēru galvenā vērtība var būt kvalitātes ietekmes uz ekonomiskiem mērķiem novērtēšana, pamatojoties uz iepriekšējo pieredzi par to, kā kvalitāte ietekmē ekonomiskos rezultātus. Finanšu mērus, kas ir cieši saistīti ar sistēmas/programmatūras produktu uzvedību, būs vieglāk interpretēt, piemēram, klientu apkalpošana (*Service to customers*), tīmekļa vietņu apmeklētāji, kas kļuvuši par klientiem (*Website visitors converted to customers*), ieņēmumi no katra klienta (*Revenue from each customer*) un kļūdas ar ekonomiskām sekām (*Errors with economic consequences*).

Ekonomiskā riska mazinājumu standartā raksturo astoņi dažādi mēri, viens no tiem ir “(Rec-1-G) peļņa no ieguldījuma (*Return on investment (ROI)*)”, ko, pēc aptaujas rezultātiem, izmanto vairāk nekā 80% aptaujāto uzņēmumu. Šī mēra definīcija sakrīt ar nosaukumu. Tā aprēķināšanai tiek izmantota formula (4.1), un to mēra, veicot darījumanalīzi. [2]

$$X = (A - B)/B \quad (4.1)$$

A – iegūtie papildu labumi

B – investētais apjoms

Standartā kā piemēri iegūtajiem labumiem ir minēti, piemēram, personāla izdevumu samazināšana, krājumu samazināšanās vai materiālu izmaksu samazināšanās, veicot koncentrētus pirkumus.

Peļņa no ieguldījuma mēra novērtēšana:

1. Peļņa no ieguldījuma ir svarīga tam, kurš maksā par programmatūras izstrādi un gūs peļņu no programmatūras ieviešanas. Tas visdrīzāk varētu būt programmatūras pasūtītājs.
2. Mērķis var būt, piemēram, 10% peļņa no ieguldījuma [12].
3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Cik daudz līdzekļu ir investēti programmatūras izstrādē un ieviešanā?
 - b. Kādi ir iegūtie papildu labumi no programmatūras ieviešanas?

4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast peļņu no ieguldījuma un sasniegt 10% peļņu no ieguldījuma,
 - b) novērtēt peļņu no ieguldījuma un sasniegt 10% peļņu no ieguldījuma,
 - c) kontrolēt peļņu no ieguldījuma un sasniegt 10% peļņu no ieguldījuma,
 - d) prognozēt peļņu no ieguldījuma un sasniegt 10% peļņu no ieguldījuma.
5. Peļņa no ieguldījuma standartā tā arī tiek aprakstīta un papildus apraksts vai definīcija nav dota, *taču apskatot mēra formulu, var teikt, ka peļņa no ieguldījuma ir attiecība starp investēto apjomu un iegūtajiem labumiem.*
6. Peļņu no ieguldījuma aprēķina pēc formulas (4.1), un tā atbilst tam, kā tiek aprēķināta peļņa no ieguldījuma, *bet lai rezultātu izteiktu procentos, būtu jāveic dalīšana ar 100%.*
7. Formula satur divus mainīgos: iegūtie papildu labumi un investētais apjoms. Standartā nav konkrēti aprakstīts, kas ietilpst šajās vērtībās un kā tās iegūt. Ir minēti piemēri tam, kas var būt iegūtie papildu labumi. Izmantojot šo informāciju, var izveidot mainīgo vērtību aprakstu vai definīciju. *Piemēram, investētais apjoms ir visas tās izmaksas, kas nepieciešamas, lai izstrādātu, ieviestu un uzturētu programmatūru. Iegūtie papildu labumi ir viss tas, kas tiks iegūts no programmatūras ieviešanas un izmantošanas, ieviešot programmatūru ieguvumi būs, piemēram, personāla izdevumu samazināšanās, materiālu izmaksu samazināšanās un citi ieguvumi, kā minēts paša standartā.*
8. Standarts nenosaka to, kā interpretēt mēra rezultātu, vai arī kādas vērtības būtu labas. *Loģiski domājot, iegūtajiem papildu labumiem būtu jābūt lielākiem par investēto apjomu, līdz ar to arī peļņai no ieguldījuma būtu jābūt pozitīvai jeb lielākai par 0%. To varētu uzskatīt par minimumu. Savukārt, laba peļņa no ieguldījuma katrai nozarei varētu būt atšķirīga, bet par labu peļņu no ieguldījuma varētu uzskatīt, piemēram, 10% un vairāk. Tālāk attiecīgi, jo lielāka ir peļņa no ieguldījuma, jo lielāku pievienoto vērtību dod izstrādātā programmatūra.*
9. –
10. *Kā papildus kritērijs šim mēram varētu būt laiks, jo ir noteicoši vai, piemēram, 10% peļņa no ieguldījuma tiek sasniegta viena mēneša vai viena gada laikā.*
11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs varētu būt, piemēram, produkta īpašnieks, pasūtītājs vai analītiķis, kas pārzina uzņēmuma darbību. Formulai (4.1) nepieciešamie dati ir reāli iegūstami, bet tas atšķirsies katrā*

uzņēmumā, ņemot vērā tā specifiku. Tā pat arī datu iegūšanai nepieciešamais laiks un resursi ir atbilstoši iegūstamajiem rezultātiem.

12. Cilvēciskais faktors

- a. Vispārīgs mērs.
- b. Objektīvs konkrētam produktam, jo nenosaka faktisko peļņu, bet tikai procentuālo daļu, kā arī nenosaka laika periodu, kādā tā tiek sasniegta.
- c. Izmanto vairāk nekā 80% aptaujāto uzņēmumu.
- d. Svarīgi būtu ņemt vērā laiku, kurā peļņa no ieguldījuma tiek sasniegta, jo ir nozīmīgi vai 10% tā tiek sasniegta viena mēneša, viena gada vai desmit gadu laikā. Standartā ir arī mērs (Rec-2-G) laiks, lai sasniegtu ROI (Time to achieve ROI), kas tieši to arī nosaka, kas nozīmē, ka mērs Peļņa no ieguldījuma būtu jāizmanto kopā ar mēru laiks, lai sasniegtu ROI.
- e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.

13. SMART kritēriju novērtējumu mēram peļņa no ieguldījuma skatīt 4.2. tabulā.

Standartā aprakstītais mērs labi atbilst kritērijam S, jo tas ir adresētu uz mērīšanai paredzēto apgabalu, šajā gadījumā peļņu. Kritērijam M mērs atbilst vidēji, nav precīzi aprakstīts kā savākt precīzus datus, jo mēra aprēķināšanai nepieciešamo elementu definīcijas nav pilnīgas. A kritērijs nav ievērots, jeb ir vājš, jo standartā nav norādījumi par “labo” un “slikto” virzienu, lai zinātu, kad nepieciešams rīkoties. Kritērijam R mērs atbilst labi, tas sniedz vidējus ieteikumus par saistīto procesu uzlabojumiem, piemēram, ja peļņa no ieguldījuma ir maza, tas norāda, ka nepieciešami uzlabojumi. Arī kritērijam T mērs atbilst labi, jo tiek pieņemts, ka mēram nepieciešamajai informācijas un datu iegūšanai tiek atvēlēts laiks, lai rentabli varētu veikt to analīzi.

4.2. tabula

SMART kritēriju novērtējums mēram peļņa no ieguldījuma

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs peļņa no ieguldījuma ir piemērots programmatūras novērtēšanai, tomēr būtu nepieciešams papildināt standartā pieejamo informāciju, lai mērs būtu viennozīmīgs un nepārprotams. Būtu nepieciešams papildināt informāciju par formulā (4.1) izmantotajiem mainīgajiem, kā arī noteikt kā mēru interpretēt un kādas vērtības ir labas. Tas arī atspoguļojas SMART kritēriju novērtējumā, kas kopumā ir labs, bet M un A kritēriju vērtējumi ir vāji un

vidēji. Šis mērs varētu būt aktuāls un interesējošs daudzos programmatūras projektos, bet iespējams būs tādi programmatūras projekti, kuros netiks mērīta peļņa no ieguldījuma, līdz ar to ne vienmēr būs iespējams šī mēra rezultātus salīdzināt starp dažādām programmatūrām.

4.2.2. Apmierinātība

Apmierinātība ir pakāpe, kādā tiek apmierinātas lietotāja vajadzības, izmantojot produktu vai sistēmu noteiktā lietošanas kontekstā [2]. Lietotājam, kurš tieši nesadarbojas ar produktu vai sistēmu, ir svarīga tikai mērķa sasniegšana un uzticēšanās. Apmierinātība ir lietotāja reakcija uz mijiedarbību ar produktu vai sistēmu, un tā ietver attieksmi pret produkta lietošanu. Lietotāji ietver primāros lietotājus, kas mijiedarbojas ar sistēmu, lai sasniegtu galvenos mērķus, lietotājus, kas nodrošina atbalstu un netiešos lietotājus, kuri saņem izvadi, bet nesadarbojas ar sistēmu. Lietotāju apmierinātību ietekmē programmatūru un datorsistēmu īpašību uztvere, piemēram, to, kuras mēra ar ārējiem mēriem, kā arī rezultativitātes, efektivitātes un riskbrīvības uztvere. Lietotāju apmierinātību var izmērīt jau izstrādes laikā, tā ir daļa no lietojamības testēšanas, un tas palīdz saprast, kā dizains var tikt uzlabots. Lietotāju apmierinātību var izmērīt, aptaujājot lietotājus, pēc jaunas versijas publicēšanas, lai saprastu, vai produkts atbilst lietotāju prasībām. Psihometriskajai skalai ir zināma uzticamība un derīgums. Standartā lietotāja vajadzības ietver vēlmes un cerības, kas saistītas ar produkta, sistēmas vai pakalpojuma izmantošanu. Vēlmju un cerību pārsniegšana ir līdzeklis, lai ievērojami palielinātu apmierinātību un uzlabotu lietotāja pieredzi. Standarts ISO/IEC 25022:2016 apmierinātību iedala piecos apakšraksturlielumos – vispārējā apmierinātība, noderīgums, uzticēšanās, patika un ērtums. Apmierinātību var mērīt trīs veidos, gan kā kopējo vispārējo apmierinātību, izmantojot kopējo vispārējās apmierinātības mēru, gan kā specifisku apmierinātības apakšraksturlielumu (noderīgums, uzticēšanās, patika, ērtums), gan kā kopējo apmierinātību, kas iegūta, kombinējot atsevišķus apakšraksturlielumus. Kombinētie apakšraksturlielumi var tikt novērtēt pēc to relatīvās nozīmes un ieguldījuma kopējā apmierinātībā. Lietotāji atbild uz jautājumiem apmierinātības anketā, izvēloties vērtību no skalas, kurai ir, piemēram, divi punkti (piekrītu, nepiekrītu) vai vairāki punkti (sākot no pilnībā piekrītu līdz pilnībā nepiekrītu). No apmierinātības anketām iegūtais kopējais rezultāts bieži tiek mēroģots diapazonā no 0 līdz 100. Dažām anketām ir iepriekšējo rezultātu datu bāze, kas ļauj rezultātus attēlot kā lietotāju procentuālo daļu, kuri iepriekš līdzīgā produktam vai sistēmai ir devuši tādu pašu vai labāku rezultātu.

4.2.2.1. Vispārējā apmierinātība

Vispārējo apmierinātību raksturo mērs “(Sus-1-G) kopējā apmierinātība (*Overall satisfaction*)”, to izmanto 75% aptaujāto uzņēmumu. Šī mēra definīcija tikpat kā sakrīt ar tā nosaukumu – lietotāja kopējā apmierinātība [2]. Tā aprēķināšanai tiek izmantota formula (4.2), un to mēra, veicot aptaujas.

$$X = \sum A_i \quad (4.2)$$

A_i – atbilde uz jautājumu

Apmierinātības kopējie mērījumi ir, piemēram, tīkla veicināšanas vērtējuma (*Net Promoter Score*) un viegluma (*Ease Question*) aptaujas jautājumi.

Kopējās apmierinātības mēra novērtēšana:

1. Kopējā apmierinātība ir svarīga produkta īpašniekam, pasūtītājam, klientam vai izstrādātājam. Ieinteresētā persona šim mēram varētu būt kāda no minētajām.
2. Pieņemot, ka aptaujā tiek izmantota Likerta skala no 1 līdz 5, kur 1 ir pilnībā nepiekrītu, bet 5 ir pilnībā piekrītu, šī mēra mērķis būtu iegūt vidējo novērtējumu, piemēram, lielāku vai vienādu ar 4.
3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Cik lietotāji ir apmierināti ar programmatūru?
 - b. Cik lietotāji nav apmierināti ar programmatūru?
 - c. Kas jāmaina vai jāuzlabo, lai neapmierinātie lietotāju kļūtu apmierināti ar programmatūru?
4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast kopējo apmierinātību un sasniegt kopējo apmierinātību ≥ 4 ,
 - b) novērtēt kopējo apmierinātību un sasniegt kopējo apmierinātību ≥ 4 ,
 - c) kontrolēt kopējo apmierinātību un sasniegt kopējo apmierinātību ≥ 4 ,
 - d) prognozēt kopējo apmierinātību un sasniegt kopējo apmierinātību ≥ 4 .
5. Kopējā apmierinātība standartā tiek aprakstīta kā lietotāja kopējā apmierinātība, kas neko daudz nepaskaidro, taču pats termins apmierinātība jau ir definēts standartā, līdz ar to ir skaidri pateikts, kas ir apmierinātība.
6. Kopējo apmierinātību aprēķina pēc formulas (4.2). *Autoraprāt, šī formula būtu jāpapildina un iegūtie atbilžu rezultāti būtu jādala ar atbilžu skaitu, lai iegūtu vidējo apmierinātības novērtējumu.*
7. Formula satur tikai vienu mainīgo – atbilde uz jautājumu. Standartā ir minēti piemēri aptaujas jautājumiem. Tīkla veicināšanas vērtējuma jautājumi mēra klientu

pieredzi un prognozē darījumu izaugsmi skalā no 1 līdz 10, kur 1 ir maz ticams, bet 10 ir ļoti ticams [19]. Viegluma jautājumi ir par to cik viegls vai grūts lietotājam bija uzdevums skalā no 1 līdz 7, kur 1 ir ļoti grūts, bet 7 ir ļoti viegls [10]. Lai gan ir doti piemēri tam, kādus jautājumus izmantot aptaujā, lai noteiktu kopējo apmierinātību, nav teikts, ka tie ir obligāti izmantojami, kā arī ir iespējams izmantot arī cita veida jautājumus. Līdz ar to aptaujas var atšķirties dažādiem programmatūras novērtējumiem, kas var radīt atšķirīgus rezultātus, kurus nav iespējams salīdzināt. *Lai mērs būtu izmantojams dažādu programatūru salīdzināšanai, būtu jādefinē aptaujas paraugs.*

8. Nav konkrēti definēts, kā tieši veidot aptauju, lai noteiktu kopējo apmierinātību. Tā pat arī nav noteikts, kādas vērtības ir labas un kādas ir sliktas. *Ja apmierinātība tiktu vērtēta skalā no 1 līdz 5, tad varētu pieņemt, ka 1 ir ļoti slikti, bet 5 – ļoti labi.*
9. –
10. Standartā kopējai apmierinātībai nav norādīti papildus kritēriji, bet ir citi vispārējās apmierinātības mēri, kas raksturo, piemēram, apmierinātību ar iespējām, lietotāju uzticēšanos, lietotāju patiku, fizisko ērtumu un citus raksturlielumus.
11. *Apmierinātības aptaujas varētu nodot lietotājiem un apkopot rezultātus izstrādātāju pārstāvis vai produkta īpašnieks, atkarībā no tā, kura būtu ieinteresētā persona. Tā pat arī aptaujas sagatavošana varētu būt šīs pašas personas uzdevums. Mēra aprēķināšanai nepieciešamie dati ir reāli iegūstami, bet tie būs subjektīvi, jo tiek iegūti no lietotāju aptaujām. Datu iegūšanai nepieciešamais laiks un resursi ir atbilstoši iegūstamajiem rezultātiem.*
12. Cilvēciskais faktors
 - a. Vispārīgs mērs.
 - b. *Subjektīvs mērs, jo rezultāti tiek iegūti no lietotāju aptaujām.*
 - c. Izmanto 75% aptaujāto uzņēmumu.
 - d. Šis mērs varētu tikt izmantots kopā ar citiem apmierinātības mēriem, lai izdalītu dažādas apmierinātības šķautnes.
 - e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.
13. SMART kritēriju novērtējumu mēram kopējā apmierinātība skatīt 4.3. tabulā. Standartā aprakstītais mērs teicami atbilsts kritērijam S un M, jo tas ir adresēts uz mērīšanai paredzēto apgabalu, kas ir lietotāju apmierinātība, kā arī ir sniegti vērtīgi ieteikumi aptaujas jautājumu piemēriem. A ir vājš, lai gan cilvēcīgi būtu skaidrs, kā

interpretēt aptaujas jautājumu atbildes, ja tās ir, piemēram, skalā no 1 – 5, standartā nav norādījumi par labajām un sliktajām vērtībām. Kritērijam R un T mērs atbilst labi, tas ir svarīgs mērs un ļauj izprast lietotāju apmierinātību ar programmatūru, lai zinātu, ko nepieciešams uzlabot, kā arī tiek pieņemts, ka mēram nepieciešamajai informācijas un datu iegūšanai tiek atvēlēts laiks, lai rentabli varētu veikt to analīzi.

4.3. tabula

SMART kritēriju novērtējums mēram kopējā apmierinātība

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs kopējā apmierinātība ir izmantojams programmatūras novērtēšanai, tomēr būtu nepieciešams papildināt standartā pieejamo informāciju. Būtu nepieciešams papildināt, piemēram, informāciju par to, kā interpretēt mēra rezultātus, un kādas vērtības ir labas. Šis mērs varētu būt aktuāls un interesējošs visiem programmatūras projektiem, jo tos izmantos lietotāji un ir svarīgi, lai lietotāji, kas izmanto programmatūru, ir apmierināti, un to izmantotu pēc iespējas lielāks skaits lietotāju. Tomēr, lai šī mēra rezultātus salīdzinātu starp dažādām programmatūrām, būtu nepieciešama standartizēta aptauja, kā arī aptaujāto lietotāju kopai būtu jābūt stingri definētai, lai rezultāti būtu līdzvērtīgi interpretējami. SMART kritēriju novērtējums ir ļoti labs, bet arī tajā redzams, ka A kritērijs ir vājš, kas norāda uz to, ka trūkst norādes par rezultātu “labo” un slikto” virzienu.

4.2.4. Lietojumkvalitātes mēri

Standartā ISO/IEC 25022:2016 kopā ir aprakstīti 36 lietojumkvalitātes mēri. Tikai vienu no tiem – peļņa no ieguldījuma – izmanto vairāk nekā 80% aptaujāto uzņēmumu un vēl vienu – kopējā apmierinātība – izmanto 75% aptaujāto uzņēmumu. Pārējie mēri tiek izmantoti retāk. Praktiski paskatoties uz programmatūru un to novērtējot, lietotāju apmierinātība un peļņa no ieguldījuma izsaka ļoti daudz, un tie ir svarīgi un būtiski mēri programmatūras novērtēšanai.

Gan peļņa no ieguldījuma, gan vispārējā apmierinātība ir vispārīgi mēri, tātad tos ir iespējams izmantot visos programmatūras novērtēšanas gadījumos. Taču trūkums ir tāds, ka standartā nav noteikts kā tieši šos mērus interpretēt, līdz ar to nav skaidrs, kā tos salīdzināt starp dažādām programmatūrām. Lai gan peļņa no ieguldījuma ir objektīvs mērs, jo tam nepieciešamie dati tiek iegūti darījumanalīzes ceļā, rezultātus nevar salīdzināt ar citām programmatūrām, jo mērs nenosaka laika periodu kādā iegūta peļņa, kā arī rezultāts tiek izteikts procentos un faktiskā peļņa var ļoti atšķirties dažādos projektos. Vispārējo apmierinātību nevar

uzskatīt par objektīvu mēru, jo to mēra veicot lietotāju aptaujas, līdz ar to rezultāti ir atkarīgi no jautājumiem un lietotāju atbildēm, ko ietekmē daudz un dažādi faktori.

Standartā ir aprakstīts tas, kā ir jārikojas, lai izpildītu standarta prasības. Tas nosaka to, ka ir nepieciešams precīzi definēt to, kā mērs tiek piemērots, piemēram, mērījuma metodes detaļas un izmantotā aptauja. Tā pat ir ļoti svarīgi izmantot vienu un to pašu mērījumu metodi, kad tiek veikta dažādu produktu salīdzināšana. Nodaļā par mēru izvēli ir teikts, ka lai novērtētu lietojumkvalitāti, parasti ir nepieciešams izmērīt vismaz vienu mēru no rezultativitātes, efektivitātes un apmierinātības, un ja tas ir atbilstoši, tad arī no riskbrīvības. Apskatot mēru objektivitāti, standartā ir teikts, ka objektīvi mēri sniedz tiešas norādes par rezultativitāti un efektivitāti, bet subjektīvi mēri var būt tieši saistīti ar apmierinātību.

Tā kā lietojumkvalitātes raksturlielumu relatīvā nozīme ir atkarīga no lietošanas konteksta un mērķiem, kādiem tiek precizēta vai novērtēta lietojumkvalitāte, standartā nav aprakstīts vispārīgs noteikums, kā mērus vajadzētu izvēlēties vai kombinēt. Dažus lietojumkvalitātes mērus, piemēram, laiks, lai pabeigtu uzdevumu, var būt grūti interpretēt atsevišķi. Ir vairāki veidi kā lietojumkvalitātes mēri var tikt salīdzināti, lai tos būtu vieglāk interpretēt, piemēram, atbilstība, etaloni, laikindas, kvalifikācija un apmierinātības normas. Jāuzmanās, veicot lietojumkvalitātes mēru rezultātu vispārināšanu citā kontekstā, kam varētu būt ievērojami atšķirīgi lietotāji, uzdevumi vai vide. Ja lietojumkvalitātes mēri tiek iegūti īsā laika posmā, vērtībās, iespējams, netiek ņemti vērā reti gadījumi, kuriem varētu būt nozīmīga ietekme uz lietojumkvalitāti, piemēram, periodiskas sistēmas kļūdas.

Kopumā standartā aprakstītie mēri ir piemēroti, lai novērtētu programmatūras lietojumkvalitāti, taču tie būtu jāpapildina, lai tie tiktu izmantoti vienādi un rezultātus varētu salīdzināt starp dažādām programmatūrām. Lai gan standartā lietojumkvalitāti apraksta 36 mēri, no kuriem 24 ir vispārīgi mēri, programmatūru varētu novērtēt arī ar mazāku daudzumu mēriem, tādā veidā, koncentrējoties un būtiskāko.

4.3. Programmatūras produkta kvalitātes mēru novērtēšana

Sistēmas/programmatūras produkta kvalitāte ir pakāpe, kādā tā apmierina dažādu ieinteresēto personu noteiktās un netiešās vajadzības, tādā veidā sniedzot vērtību [3]. Šīs noteiktās un netiešās vajadzības ir attēlotas SQuaRE standartu sērijā kvalitātes modelī, kas sistēmas/programmatūras produktu kvalitāti iedala raksturlielos, no kuriem daļa tiek iedalīti arī apakšraksturlielos. Sistēmas/programmatūras produkta ar kvalitāti saistītās izmērāmās īpašības, tiek sauktas par kvantitatīvām īpašībām, un tās var tikt saistīt ar kvalitātes mēriem.

Sistēmas/programmatūras produkta kvalitāti raksturo astoņi raksturlielumi – funkcionālā atbilstība, veiktspējas efektivitāte, saderība, lietojamība, uzticamība, drošība, uzturamība un pārnesamība.

4.3.1. Veiktspējas efektivitāte

Veiktspējas efektivitātes mērus izmanto, lai novērtētu veiktspēju salīdzinājumā ar resursu daudzumu, kas izmantots noteiktos apstākļos [3]. Resursi var ietvert citus programmatūras produktus, sistēmas programmatūras un aparatūras konfigurāciju un materiālus (piemēram, papīru vai datu nesēju). Veiktspējas efektivitātes mērs tiek būtiski ietekmēts un svārstās atkarībā no lietošanas apstākļiem, piemēram, datu apstrādes slodzes, izmantošanas biežuma, saistītajām vietnēm un citiem. Tāpēc veiktspējas efektivitātes mēri var ietvert novērtētās vai aprēķinātās vērtības kļūdu svārstības attiecību ar projektēto vērtību atļauto kļūdas svārstību diapazonu, kā norādīts specifikācijā. Ieteicams uzskaitīt un izpētīt arī tādu faktoru ietekmi kā procesors un atmiņa, ko izmanto citas programmatūras, tīkla noslodze un plānotie fona procesi. Iespējamās svārstības un derīgos diapazonus novērtētajām vai aprēķinātajām vērtībām var noteikt un salīdzināt ar prasību specifikāciju. Ieteicams arī noteikt un definēt uzdevumu, kas būtu piemērots veiktspējas efektivitātes vai jaudas mērījumiem, piemēram, transakcija kā darījumlietojumprogrammas uzdevums, komutācija vai datu pakete, kas nosūtīta kā uzdevums sakaru lietojumprogrammai, notikumu vadība kā uzdevums kontroles lietojumprogrammai un lietotāja izsauktas funkcijas datu izvade kā uzdevums parastai lietotāja lietojumprogrammai. Standarts ISO/IEC 25023:2016 veiktspējas efektivitāti iedala trīs apakšraksturlielumos – laikuzevedība, resursu izmantošana un jauda.

4.3.1.1. Laikuzevedība

Laikuzevedības mēri tiek izmantoti, lai novērtētu, cik lielā mērā produkta vai sistēmas reakcijas un apstrādes laiks un caurlaidspēja, izpildot funkcijas, atbilst prasībām [3]. Laikuzevedību standartā raksturo pieci dažādi mēri, viens no tiem ir “(PTb-1-G) vidējais atbildes laiks (*Mean response time*)”, ko pēc aptaujas rezultātiem, izmanto vairāk nekā 80% aptaujāto uzņēmumu. Vidējais atbildes laiks ir vidējais laiks, kas nepieciešams sistēmai, lai reaģētu uz lietotāja vai sistēmas darbību, to mēra kā vidējo aritmētisko no visiem laikiem, kas sistēmai bijuši nepieciešami, lai reaģētu uz konkrētu lietotāja vai sistēmas darbību. Tā aprēķināšanai tiek izmantota formula (4.3).

$$X = \sum_{i=1}^n (A_i) / n \quad (4.3)$$

A_i – laiks, kas sistēmai nepieciešams, lai reaģētu uz konkrētu lietotāja uzdevumu vai sistēmas uzdevumu i -tajā mērījumā
 n – mērīto atbilžu skaits

Vidējais atbildes laiks mēra novērtēšana:

1. Vidējais atbildes laiks ir svarīgs lietotājam, lai sistēma būtu ērti, patīkami un ātri lietojama. To, lai klients būtu apmierināts, savukārt, cenšas nodrošināt programmatūras pasūtītājs un iekļauj prasību specifikācijā nosacījumus par to, kādam jābūt sistēmas atbildes laikam. Pēc tam, atbilstoši prasību specifikācijai, izstrādātāji izstrādā programmatūru, kas atbilst prasībām, un izmanto serveri, kas spēj apstrādāt prognozēto lietotāju skaitu. Šajā gadījumā ieinteresētās personas būs pasūtītājs un izstrādātājs.
2. Mērķis var būt, piemēram, sistēmas vidējais atbildes laiks mazāks vai vienāds ar 400ms.
3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Kāds ir ilgākais sistēmas atbildes laiks?
 - b. Cik un kādu pieprasījumu atbildes laiks ir ilgāks par 400ms?
 - c. Kā sistēmas atbildes laiku saīsināt līdz 400ms vai mazāk?
4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast sistēmas vidējo atbildes laiku un sasniegt vidējo atbildes laiku zem 400ms,
 - b) novērtēt sistēmas vidējo atbildes laiku un sasniegt vidējo atbildes laiku zem 400ms,
 - c) kontrolēt sistēmas vidējo atbildes laiku un sasniegt vidējo atbildes laiku zem 400ms,
 - d) prognozēt sistēmas vidējo atbildes laiku un sasniegt vidējo atbildes laiku zem 400ms.
5. Vidējais atbildes laiks standartā tiek aprakstīts kā vidējais laiks, kas nepieciešams sistēmai, lai reaģētu uz lietotāja vai sistēmas darbību. Latvijas Zinātņu akadēmijas terminoloģijas komisijas tīmekļa vietnē [7] atbildes laiks tiek definēts kā laika intervāls, kas sākas tad, kad dators saņem lietotāja pieprasījumu, un beidzas ar pieprasījuma izpildi vai ziņojumu, ka pieprasījums nav izpildāms. Standartā pieejamā definīcija nenosaka precīzi to, kad tiek uzsākta atbildes laika mērīšana un

kad tā beidzas. Lai gan pieprasījumu atbildes laiku cilvēki nemēra paši, bet šie dati tiek iegūti no kāda servisa, kas sniedz informāciju par pieprasījumu atbildes laikiem un servera darbību, būtu labi konkrētāk aprakstīt, kad sākas atbildes laiks, un kad tas beidzas. Tādā veidā arī cilvēkiem, kas nav tieši saistīti par programmatūras izstrādi un serveru pārvaldību, būtu skaidrs, kas tiek ietverts atbildes laikā. *Autoraprāt, standartā esošo definīciju varētu papildināt vai pielīdzināt Latvijas Zinātņu akadēmijas terminoloģijas komisijas tīmekļa vietnē pieejamajai definīcijai.*

6. Vidējo atbildes laiku aprēķina pēc formulas (4.2), un tā atbilst tam, kā pieņemts aprēķināt vidējo vērtību.
7. Formula satur divus mainīgos: laiku, kas sistēmai nepieciešams, lai reaģētu uz konkrētu lietotāja uzdevumu vai sistēmas uzdevumu un mērīto atbilžu skaitu. Mērīto atbilžu skaits ir skaidri saprotams un nepārprotams mainīgais, bet standartā nav plašāk aprakstīts, kas ir laiks, kas sistēmai nepieciešams, lai reaģētu uz konkrētu lietotāja vai sistēmas uzdevumu. *To varētu papildināt pēc līdzības ar piektajā punktā aprakstīto mēra definīciju.*
8. Standarts nenosaka to, kā interpretēt mēra rezultātu, vai arī kādas vērtības būtu labas. *Labs atbildes laiks ir līdz 200 ms, [18, 20, 21] detalizētāku atbilžu laiku iespējamo novērtējumu skatīt 4.4. tabulā.*

4.4. tabula

Atbildes laika novērtējums

Atbildes laiks (ms)	Novērtējums
≤ 100 ms	ļoti ātri
≤ 200 ms	ātri
≤ 500 ms	vidēji
> 500 ms	lēni

9. –
10. *Kā papildus kritērijs šim mēram varētu būt pieprasījuma tips. Tādā veidā varētu novērtēt atsevišķi, piemēram, autorizācijas, sadaļu atvēršanās vai bilžu ielādes pieprasījumus.*
11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs varētu būt kāds no izstrādātājiem vai ieinteresētajām personām. Pieprasījumu atbildes laiki ir iegūstami tehniski no servera vai arī no kādas lietotnes, kas nodrošina pieprasījumu atbildes laiku attēlošanu. Formulai (4.3) nepieciešamie dati ir reāli*

iegūstami, bet to iegūšanai varētu tikt izmantotas dažādi tehniski veidi vai lietotnes. Datu iegūšanai nepieciešamais laiks un resursi ir atbilstoši iegūstamajiem rezultātiem.

12. Cilvēciskais faktors

- a. Vispārīgs mērs.
- b. Ļoti ieteicams.
- c. Objektīvs un salīdzināms starp dažādām programmatūrām.
- d. Izmanto vairāk nekā 80% aptaujāto uzņēmumu.
- e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.

13. SMART kritēriju novērtējumu mēram vidējais atbildes laiks skatīt 4.5. tabulā.

4.5. tabula

SMART kritēriju novērtējums mēram vidējais atbildes laiks

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs vidējas atbildes laiks ir piemērots un svarīgs programmatūras novērtēšanai un labi nodēd dažādu programmatūru salīdzināšanai, tomēr būtu nepieciešams papildināt standartā pieejamo informāciju, lai mērs būtu viennozīmīgs un vienādi interpretējams. Galvenokārt būtu jānorāda, kā interpretēt iegūtos rezultātus, kā arī varētu papildināt mēru, lai dažāda tipa pieprasījumi tiktu salīdzināti atsevišķi. SMART kritēriju novērtējums ir labs, bet kritēriji M un A norāda uz trūkumiem mēra definīcijā un tajā, ka nav norādes par rezultātu “labo” un “slikto” virzienu.

Laikuzvedību raksturo arī ir mērs “(PTb-3-G) vidējais darba izpildes ilgums (*Mean turnaround time*)”. Tas ir vidējais laiks, kas nepieciešams, lai pabeigtu darbu vai asinhronu procesu, to mēra kā vidējo aritmētisko no visu darbu izpildei nepieciešamajiem laikiem [3]. Tā aprēķināšanai tiek izmantota formula (4.4).

$$X = \sum_{i=1}^n (Bi - Ai) / n \tag{4.4}$$

A_i – i-tā darba uzsākšanas laiks

B_i – i-tā darba pabeigšanas laiks

n – mērījumu skaits

Vidējais darba izpildes ilgums mēra novērtēšana:

1. Vidējais darba izpildes ilgums ir svarīgs sistēmas mērījums, kas nodrošina to, lai sistēmas lietotāji varētu lietot ātru programmatūru. Gala lietotāju apmierinātība ir svarīga produkta īpašniekam, kā arī programmatūras izstrādātājiem, jo mērķis ir

- radīt kvalitatīvu un ātru programmatūru. Šī mēra ieinteresētā persona varētu būt produkta īpašnieks, programmatūras pasūtītājs jeb klients vai arī izstrādātājs.
2. Mērķis var būt, piemēram, vidējais darba izpildes ilgums mazāks vai vienāds ar 500ms.
 3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Kāds ir ilgākais darba izpildes ilgums?
 - b. Cik un kādu darba izpildes ilgums ir ilgāks par 500ms?
 - c. Kā darba izpildes ilgums samazināt līdz 500ms vai mazāk?
 4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast sistēmas vidējo darba izpildes ilgumu un sasniegt vidējo darba izpildes ilgumu zem 500ms,
 - b) novērtēt sistēmas vidējo darba izpildes ilgumu un sasniegt vidējo darba izpildes ilgumu zem 500ms,
 - c) kontrolēt sistēmas vidējo darba izpildes ilgumu un sasniegt vidējo darba izpildes ilgumu zem 500ms,
 - d) prognozēt sistēmas vidējo darba izpildes ilgumu un sasniegt vidējo darba izpildes ilgumu zem 500ms.
 5. Vidējais darba izpildes ilgums standartā tiek aprakstīts kā vidējais laiks, kas nepieciešams, lai pabeigtu darbu vai asinhronu procesu. Pieminot asinhronu procesu, gribas domāt, ka runa ir par pašas programmatūras izpildes laikiem, kā arī laikuzvedība tiek attiecināta uz pašu programmatūru, bet konkrētā mēra definīcijā, tas nav pieminēts. Standartā pieejamā definīcija nepaskaidro to, ka tiek apskatīti tieši sistēmas jeb programmatūras darbu izpildes ilgumi. Latvijas Zinātņu akadēmijas terminoloģijas komisijas tīmekļa vietnē [6] darba izpildes ilgums tiek definēts kā laiks no darba nodošanas operētājsistēmai līdz rezultātu saņemšanai. *Autoraprāt, standartā esošo definīciju varētu papildināt vai pielīdzināt Latvijas Zinātņu akadēmijas terminoloģijas komisijas tīmekļa vietnē pieejamajai definīcijai.*
 6. Vidējo atbildes laiku aprēķina pēc formulas (4.5), un tā atbilst tam, kā pieņemts aprēķināt vidējo vērtību.
 7. Formula satur trīs mainīgos: darba uzsākšanas laiku, darba pabeigšanas laiku un mērījumu skaitu. Mērīto atbilžu skaits ir skaidri saprotams un nepārprotams mainīgais, tā pat arī darba uzsākšanas un pabeigšanas laiki ir skaidri noteikti mainīgie.

8. Standarts nenosaka to, kā interpretēt mēra rezultātu, un kādas vērtības būtu labas. Iepriekš tika noteikts, ka *labs atbildes laiks ir līdz 200 ms. Darba izpildes ilgums ir viss kopējais laiks no darba uzsākšanas līdz atbildes nodošanai lietotājam, kas ietver arī atbildes laiku, tāpēc darba izpildes ilgums būs ilgāks par atbildes laiku.*
9. –
10. *Kā papildus kritērijs šim mēram varētu tikt norādīts darba jeb uzdevuma tips. Tad varētu atsevišķi izvērtēt dažādu darbu izpildes ilgumus. Vēl varētu noteikt, kā darba izpildes ilgumu ietekmē interneta savienojums un ierīces operētājsistēma un atmiņa.*
11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs varētu būt kāds no izstrādātājiem vai ieinteresētajām personām. Darba izpildes ilgums ir iegūstams tehniski no izstrādātās programmatūras datiem vai arī no kādas lietotnes, kas nodrošina šo informāciju. Formulai (4.4) nepieciešamie dati ir reāli iegūstami, bet to iegūšanai varētu tikt izmantoti dažādi tehniski veidi vai lietotnes. Datu iegūšanai nepieciešamais laiks un resursi ir atbilstoši iegūstamajiem rezultātiem.*
12. Cilvēciskais faktors
- Vispārīgs mērs.
 - Ieteicams.
 - Objektīvs konkrētam produktam.
 - Izmanto 75% aptaujāto uzņēmumu.
 - Mēru ietekmē daudz un dažādi kritēriji, kurus izstrādājot var ņemt vērā, bet nevar uzlabot vai novērst, piemēram, iekārtas atmiņa un operētājsistēma, interneta savienojums un citi, tāpēc noderīgāk būtu mērīt tieši servera vidējo atbildes laiku, jo to ir iespējams uzlabot pašas programmatūras ietvaros.
13. SMART kritēriju novērtējumu mēram vidējais darba izpildes ilgums skatīt 4.6. tabulā.

4.6. tabula

SMART kritēriju novērtējums mēram vidējais darba izpildes ilgums

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs vidējas darba izpildes ilgums ir piemērots programmatūras novērtēšanai, taču to ietekmē dažādi papildus kritēriji, ko nevar uzlabot izstrādājot pašu programmatūru, piemēram, iekārtas atmiņa un operētājsistēma un interneta savienojums. Tā pat arī būtu nepieciešams papildināt standartā pieejamo informāciju, lai mērs būtu viennozīmīgs un vienādi

interpretējams. Galvenokārt būtu jāuzlabo mēra definīcija un jānorāda kā interpretēt iegūtos rezultātus. SMART kritēriju vērtējumi ir vidēji, kritēriji S un T ir labi, bet M un R ir vidēji, kas norāda uz trūkumiem mēra definīcijā, kā arī uzsver to, ka mērs nesniedz kritiskus ieteikumus par saistīto procesu uzlabošanu. Kritērijs A ir vājš, jo nav norādes par rezultātu “labo” un slikto” virzienu.

4.2.2.2. Jauda

Jaudas mēri tiek izmantoti, lai novērtētu, cik lielā mērā produkta vai sistēmas parametra maksimālās robežvērtības atbilst prasībām [3]. Jaudas mērījumus mēra, izmantojot dinamisko analīzi, piemēram, apjomtestēšanu, vai arī veicot sistēmas integrācijas testēšanu vai simulāciju. Maksimālo vērtību un ilgumu sadalījumu var novērtēt dažādos statistiskās analīzes, dinamiskās testēšanas vai operāciju gadījumos. Paredzams, ka maksimālā robeža tiks noteikta kā mērķa vērtība, kas teorētiski var pārsniegt iespējamo reālo vērtību.

Jaudu standartā raksturo četri dažādi mēri, viens no tiem ir “(PCa-2-G) lietotāju piekļuves jauda (*User access capacity*)”, ko pēc aptaujas rezultātiem, izmanto 75% aptaujāto uzņēmumu. Lietotāju piekļuves jauda nosaka, cik lietotāji var vienlaikus piekļūt sistēmai noteiktā laikā, to mēra kā vidējo aritmētisko no visiem novērojumiem, kas veikti par maksimālo lietotāju skaitu, kuri var vienlaicīgi piekļūt sistēmai [3]. Tā aprēķināšanai tiek izmantota formula (4.5).

$$X = \sum_{i=1}^n Ai/n \quad (4.5)$$

A_i – maksimālais lietotāju skaits, kuri var vienlaicīgi piekļūt sistēmai i -tajā novērojumā
 n – novērojumu skaits

Rezultāta vērtības var būt no 0 līdz maksimālajai robežai. Parasti lielāks rezultāts ir labāks.

Lietotāju piekļuves jaudas mēra novērtēšana:

1. Lietotāju piekļuves jauda ir svarīga produkta īpašniekam, lai lietotāji vienmēr varētu piekļūt sistēmai, tā pat arī izstrādātājiem ir svarīgi izstrādāt programmatūru, kurai var vienlaicīgi piekļūt liels lietotāju skaits. Šī mēra ieinteresētās personas var būt produkta īpašnieks, pasūtītājs vai izstrādātājs.
2. Mērķis var būt, piemēram, vienlaikus piekļūt sistēmai var 1000 lietotāju. Pārsvārā sistēmām ir labi apstrādāt pēc iespējas lielāku vienlaicīgo lietotāju skaitu, bet vispirms būtu jānoskaidro, kāds ir aptuvenais lietotāju skaits, kas izmantos konkrēto sistēmu, un cik lietotāji varētu izmantot sistēmu vienlaicīgi.

3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Cik daudz aktīvu lietotāju sistēma var vienlaicīgi apstrādāt?
 - b. Kas būtu jāmaina, lai sistēma spētu vienlaicīgi apstrādāt 1000 lietotāju vai vairāk?
4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast lietotāju piekļuves jaudu un apstrādāt vismaz 1000 vienlaicīgu lietotāju,
 - b) novērtēt lietotāju piekļuves jaudu un apstrādāt vismaz 1000 vienlaicīgu lietotāju,
 - c) kontrolēt lietotāju piekļuves jaudu laiku un apstrādāt vismaz 1000 vienlaicīgu lietotāju,
 - d) prognozēt lietotāju piekļuves jaudu laiku un apstrādāt vismaz 1000 vienlaicīgu lietotāju.
5. Lietotāju piekļuves jauda standartā tiek aprakstīta kā lietotāju skaits, kas vienlaikus var piekļūt sistēmai noteiktā laikā. Šī definīcija ir pilnīgi pietiekama un ir skaidri saprotama un nepārprotama.
6. Lietotāju piekļuves jaudu aprēķina pēc formulas (4.5), kas atbilst tam, kā pieņemts aprēķināt vidējo vērtību, taču mēra nosaukums un apraksts nenorāda, ka tiktu aprēķināta vidējā vērtība. *Autore, šo mēru sākotnēji uztvēra, ka maksimālo lietotāju skaitu, ko sistēma var vienlaicīgi apstrādāt, kaut gan korekti būtu apskatīt lietotāju skaitu, ko jebkurā laikā sistēma spēj apstrādāt. Standartā, savukārt, šo mēru apskata kā vidējo lietotāju skaitu, ko sistēma spēj apstrādāt noteiktā laikā. Līdz ar to, autoraprāt, būtu jāpapildina arī mēra definīcija.*
7. Formula satur divus mainīgos: maksimālais lietotāju skaits, kuri var vienlaicīgi piekļūt sistēmai i-tajā novērojumā un novērojumu skaits. Novērojumu skaits ir skaidri saprotams un nepārprotams mainīgais, tā pat arī ir aprakstīts, kā veikt jaudas mērījumus, piemēram, veicot apjomtestēšanu.
8. Standartā teikts, ka rezultātu vērtības var būt no 0 līdz maksimālajai robežai, un parasti lielāks rezultāts ir labāks, taču var būt arī otrādāk, piemēram, ja sistēmai vienlaicīgi drīkst pieslēgties tikai viens lietotājs. Kā arī loģiski būtu domāt, ka programmatūra ir labāka un kvalitatīvāka, ja tā vienlaicīgi var apstrādāt lielāku skaitu vienlaicīgu lietotāju. Taču jāņem vērā prasību specifikācijā noteiktais, ja nepieciešams apstrādāt nelielu vienlaicīgu lietotāju daudzumu, nav vajadzības izstrādāt programmatūru, kas spēj apstrādāt lielu lietotāju daudzumu. Šādos gadījumos nebūtu jāuzskata, ka programmatūra tāpēc ir mazāk kvalitatīva.
9. –

10. Autoraprāt, papildus kritēriji nav nepieciešami.
11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs varētu būt kāds no testētājiem, jo datu iegūšanai nepieciešams veikt apjomtestēšanu. Formulai (4.5) nepieciešamie dati ir reāli iegūstami. Datu iegūšanai nepieciešamais laiks un resursi ir atbilstoši iegūstamajiem rezultātiem.*
12. Cilvēciskais faktors
 - a. Vispārīgs mērs.
 - b. Ieteicams.
 - c. *Objektīvs un parasti salīdzināms starp dažādām programmatūrām.*
 - d. Izmanto 75% aptaujāto uzņēmumu.
 - e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.
13. SMART kritēriju novērtējumu mēram lietotāju piekļuves jauda skatīt 4.7. tabulā.

4.7. tabula

SMART kritēriju novērtējums mēram lietotāju piekļuves jauda

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs lietotāju piekļuves jauda ir piemērots un svarīgs programmatūras novērtēšanai, taču ne vienmēr būs iespējams veikt objektīvu salīdzināšanu starp dažādām programmatūrām, jo var būt dažāda prasības. Standartā būtu nepieciešams papildināt informāciju par šo mēru, lai mēra nosaukums un definīcija atbilstu formulai, kas aprēķina vidējo lietotāju skaitu, ko sistēma spēj apstrādāt noteiktā laikā. SMART kritēriju novērtējumam ir labs. Kaut gan standarts nosaka virzienu labajām un sliktajām rezultātu vērtībām, rezultātu interpretācijas nosacījumi būtu jāpapildina.

4.3.2. Funkcionālā atbilstība

Funkcionālās atbilstības mērus izmanto, lai novērtētu, cik lielā mērā produkts vai sistēma nodrošināja funkcijas, kas atbilst noteiktajām un netiešajām prasībām, ja tā tiek lietota noteiktos apstākļos [3]. Standarts ISO/IEC 25023:2016 funkcionālo atbilstību iedala trīs apakšraksturlielumos – funkcionālais pilnīgums, funkcionālā pareizība un funkcionālā piemērotība.

4.3.2.1. Funkcionālais pilnīgums

Funkcionālā pilnīguma mērus izmanto, lai novērtētu, cik lielā mērā funkciju kopums aptver visus norādītos uzdevumus un lietotāja mērķus [3]. Funkcionālo pilnīgumu raksturo mērs “(FCp-1-G) funkcionālais pārklājums (*Functional coverage*)”, ko pēc aptaujas rezultātiem, izmanto 75% aptaujāto uzņēmumu. Funkcionālais pārklājums nosaka to, cik liela daļa no paredzētajām funkcijām ir ieviesta, un to mēra kā proporciju starp paredzētajām un trūkstošajām funkcijām. Tā aprēķināšanai tiek izmantota formula (4.6).

$$X = 1 - A/B \quad (4.6)$$

A – trūkstošo funkciju skaits

B – paredzēto funkciju skaits

Funkcijas var tikt definētas prasību specifikācijā, projekta specifikācijā, lietotāja rokasgrāmatā vai visos šajos dokumentos. Trūkstoša funkcija tiek atklāta tad, kad sistēmai vai programmatūra nespēj veikt norādīto funkciju.

Funkcionālā pārklājuma mēra novērtēšana:

1. Funkcionālais pārklājums ir svarīgs gan lietotājam, lai pilnvērtīgi lietotu programmatūru, gan izstrādātājam, jo viņa uzdevums ir izstrādāt programmatūru atbilstoši klienta vēlmēm un prasību specifikācijai, gan klientam, jo viņš pasūta programmatūru un vēlas, lai tajā būtu iespējams veikt visas paredzētās funkcijas. Mēra kontekstā ieinteresētā persona visdrīzāk būs klients vai citos gadījumos izstrādātājs.
2. Mērķim būtu jābūt, ka visas prasību specifikācijā aprakstītās funkcijas ir ieviestas. *Ja izstrādes procesā prasības mainās, tad prasību specifikācija ir jāatjauno.*
3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Cik funkcijas no prasību specifikācijas ir ieviestas?
 - b. Cik funkcijas no prasību specifikācijas nav ieviestas?
 - c. Vai līdz šim neieviestas funkcijas tiks ieviestas?
4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast sistēmas funkcionālo pārklājumu un sasniegt to, ka visas paredzētās funkcijas ir ieviestas,
 - b) novērtēt sistēmas funkcionālo pārklājumu un sasniegt to, ka visas paredzētās funkcijas ir ieviestas,
 - c) kontrolēt sistēmas funkcionālo pārklājumu un sasniegt to, ka visas paredzētās funkcijas ir ieviestas,

- d) prognozēt sistēmas funkcionālo pārklājumu un sasniegt to, ka visas paredzētās funkcijas ir ieviestas.
5. Funkcionālais pārklājums standartā tiek aprakstīts kā jautājums – cik liela daļa no paredzētajām funkcijām ir ieviesta? Šī definīcija ir pilnīgi pietiekama un ļauj nepārprotami saprast, kas tiek aprēķināts. Standartā ir arī papildināts, ka funkcijas var tikt definētas prasību specifikācijā, projekta specifikācijā, lietotāja rokasgrāmatā vai visos šajos dokumentos.
 6. Funkcionālo pārklājumu aprēķina pēc formulas (4.6), un tā atbilst mēra definīcijā aprakstītajam.
 7. Formula satur divus mainīgos: trūkstošo funkciju skaitu un paredzēto funkciju skaitu. Abi mainīgie ir skaidri, saprotami un nepārprotami.
 8. Standarts nenosaka to, kā interpretēt mēra rezultātu, vai arī kādas vērtības būtu labas. *Atbilstoši mērķim, labs rezultāts būtu 1, jeb 100% kas noteiktu, ka visas paredzētās funkcijas ir ieviestas. Jebkurš mazāks rezultāts norādītu, ka kāda funkcija nav ieviesta.*
 9. –
 10. *Autoraprāt, funkcijām varētu piemērot koeficientu, jo maza, viegla neieviesta funkcija nav salīdzināma ar lielu un sarežģītu funkciju, kas nav ieviesta.*
 11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs varētu būt testētājs, citos gadījumos varbūt klients vai kāds no izstrādātājiem. Kā standartā minēts, paredzētās funkcijas ir parakstītas specifikācijās vai citos projekta dokumentos, savukārt trūkstošās funkcijas tiek atklātas tad, kad programmatūra neveic kādu no paredzētajām funkcijām.*
 12. Cilvēciskais faktors
 - a. Vispārīgs mērs.
 - b. Ļoti ieteicams.
 - c. *Objektīvs konkrētam produktam.*
 - d. Izmanto 75% aptaujāto uzņēmumu.
 - e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.
 13. SMART kritēriju novērtējumu mēram funkcionālais pārklājums skatīt 4.8. tabulā.

SMART kritēriju novērtējums mēram funkcionālais pārklājums

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs funkcionālais pārklājums ir piemērots un svarīgs programmatūras novērtēšanai, bet to nevar tieši izmantot dažādu programmatūru salīdzināšanai. Standartā būtu jānorāda, kā interpretēt rezultātus, lai noteiktu kādas vērtības ir labas, tas attēlojas arī SMART kritēriju novērtējumā, kur A kritērijs ir vērtējams kā vājš, bet pārējie kā labi vai teicami.

4.3.3. Uzticamība

Uzticamības mērus izmanto, lai novērtētu, cik lielā mērā sistēma, produkts vai komponente veic noteiktas funkcijas noteiktos apstākļos noteiktā laika periodā [3]. Iekšējie uzticamības mēri tiek izmantoti, lai prognozētu, vai attiecīgais pabeigtais sistēmas/programmatūras produkts izstrādes laikā apmierinās noteiktās uzticamības vajadzības. Ārējie uzticamības kvalitātes rādītāji tiek izmantoti, lai novērtētu atribūtus, kas saistīti ar tās sistēmas darbību, kurā programmatūra ietilpst testēšanas izpildes laikā, lai novērtētu programmatūras uzticamības pakāpi šīs sistēmā darbības laikā. Sistēmas un programmatūru vairumā gadījumu nenosaka. Standarts ISO/IEC 25023:2016 uzticamību iedala četros apakšraksturlielumos – briedums, pieejamība, defektpiecietība un atkopjamība.

4.3.3.1. Briedums

Brieduma mērus izmanto, lai novērtētu, cik lielā mērā sistēma, produkts vai komponente atbilst uzticamības vajadzībām normālā ekspluatācijā [3]. Brieduma jēdzienu var attiecināt arī uz citiem kvalitātes raksturlielumiem, lai norādītu pakāpi, kādā tie normālā ekspluatācijā atbilst nepieciešamajām prasībām. Briedumu raksturo mērs “(R_{Ma-3-G}) atteižu biežums (*Failure rate*)”, ko pēc aptaujas rezultātiem, izmanto 75% aptaujāto uzņēmumu. Atteižu biežums nosaka to, kāds ir vidējais atteižu skaits noteiktā periodā, un to mēra dalot konstatētās kļūdas ar novērojuma ilgumu, tādā veidā iegūstot vidējo kļūdu skaitu vienā laika vienībā. Tā aprēķināšanai tiek izmantota formula (4.7).

$$X = A/B \quad (4.7)$$

A – novērošanas laikā
konstatēto atteižu skaits
B – novērojuma ilgums

Šim mēram izmantotais ilgums testēšanas un operāciju vajadzībām varētu būt atšķirīgs no faktiskā lietojumu vai testēšanas laika. Uzticamības novērtēšanas modelī, šo mēru var izmantot kā ievades informāciju. Šī kvalitātes mēra derīgums ir atkarīgs no testpiemēru adekvātuma vai sistēmas izmantošanas apjoma testēšanas laikā, piem. normālos, izņēmuma un nenormālos gadījumos.

Atteižu biežuma mēra novērtēšana:

1. Zems atteižu biežums ir svarīgs produkta īpašniekam, izstrādātājam, testētājam un gala lietotājam, bet ieinteresētā persona šim mēram varētu būt produkta īpašnieks un testētājs, jo produkta īpašniekam būs svarīgi, lai atteižu biežums ir zems, bet testētājs būs tas, kurš veiks testus, lai veiktu mērījumus.
2. Mērķis varētu būtu, piemēram, lai sistēma darbojas ar mazāk nekā vienu atteici stundā, *pieņemot, ka viena laika vienība būs stunda.*
3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Cik atteices notiek vienas stundas laikā?
 - b. Ko nepieciešams mainīt, lai atteižu skaits būtu mazāks nekā viena atteice vienā stundā?
4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast atteižu biežumu un sasniegt to, ka vienas stundas laikā notiek mazāk par vienu atteici,
 - b) novērtēt atteižu biežumu un sasniegt to, ka vienas stundas laikā notiek mazāk par vienu atteici,,
 - c) kontrolēt atteižu biežumu un sasniegt to, ka vienas stundas laikā notiek mazāk par vienu atteici,
 - d) prognozēt atteižu biežumu un sasniegt to, ka vienas stundas laikā notiek mazāk par vienu atteici.
5. Atteižu biežums standartā tiek aprakstīts kā vidējais atteižu skaits noteiktā periodā. Lai gan definīcija ir skaidra, nav noteikts laika periods, kas ļauj katram šo mēru izmantot citādāk. *Autoraprāt, šī definīcija būtu jāpapildina ar konkrētu laika periodu, tādā veidā šo mēru padarot viennozīmīgu un arī salīdzināmu ar citām programmatūrām.*

6. Atteiču biežumu aprēķina pēc formulas (4.7), un tā atbilst mēra definīcijā aprakstītajam.
7. Formula satur divus mainīgos: novērošanas laikā konstatēto atteiču skaitu un novērojuma ilgumu. Konstatētais atteiču skaits ir konkrēts mainīgais, taču novērojuma ilgums nav precizēts, un tam var tikt piemērotas dažādas mērvienības. *Novērojuma ilgumam būtu jānosaka konkrēta vienība, piemēram, stunda vai diennakts.*
8. Standarts nenosaka to, kā interpretēt mēra rezultātu, vai arī kādas vērtības būtu labas. *Būtu jānosaka ar to, ka tiek noteikta novērojuma ilguma vienība, un atbilstoši tai, varētu definēt labās un sliktās vērtības, piemēram, līdz vienai atteicei diennaktī – ļoti labi, 1 – 10 atteices diennaktī – labi, 10 – 25 atteices diennaktī – vidēji, vairāk par 25 atteicēm diennaktī – slikti.*
9. –
10. *Autoraprāt, šim mēram nav vajadzīgi papildus kritēriji.*
11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs varētu būt testētājs, jo nepieciešamie dati ir iegūstami veicot dažādu testpiemēru izpildi.*
12. Cilvēciskais faktors
 - a. Vispārīgs mērs.
 - b. Ieteicams.
 - c. *Objektīvs un salīdzināms starp dažādām programmatūrām.*
 - d. Izmanto 75% aptaujāto uzņēmumu.
 - e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.
13. SMART kritēriju novērtējumu mēram atteiču biežums skatīt 4.9. tabulā.

4.9. tabula

SMART kritēriju novērtējums mēram atteiču biežums

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs atteiču biežums ir piemērots un svarīgs programmatūras novērtēšanai un labi noder dažādu programmatūru salīdzināšanai. Taču standartā būtu jānorāda ilguma vienības mērvienība, lai rezultātus būtu ērtāk salīdzināt, kā arī būtu jānorāda kā interpretēt rezultātus. SMART kritēriju vērtējums šim mēram ir labs un kritērijs A, kas ir novērtēts viszemāk parāda galvenos trūkumus.

Briedumu raksturo arī mērs “(RMa-4-S) testa aptvērums (*Test coverage*)”, ko pēc aptaujas rezultātiem, izmanto 75% aptaujāto uzņēmumu. Testa aptvērums nosaka to, cik procentuāli no sistēmas vai programmatūras faktiski veiktajām iespējām, darbības scenārijiem vai funkcijām, ir iekļautas saistītajos testa komplektos, un to mēra dalot faktisko ar testkomplektos iekļauto sistēmas vai programmatūras iespēju, darbības scenāriju vai funkciju skaitu [3]. Tā aprēķināšanai tiek izmantota formula (4.8).

$$X = A/B \quad (4.8)$$

A – faktiskais sistēmas vai programmatūras veikto iespēju, darbības scenāriju vai funkciju skaits
B – testkomplektos iekļauto sistēmas vai programmatūras iespēju, darbības scenāriju vai funkciju skaits

Testa aptvēruma mēra novērtēšana:

1. Testa aptvēruma mēra ieinteresētā persona ir programmatūras testētājs, Testētāja uzdevums ir veikt testēšanu, lai atklātu trūkumus pirms programmatūras publicēšanas. Lai to izdarītu ir nepieciešams augsts testa aptvērums.
2. Mērķis varētu būtu, piemēram, sasniegt testa aptvērumu virs 80%.
3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Cik liela daļa programmatūras ir testēta?
 - b. Kuras iespējas, darbības scenāriji un funkcijas nav testētas?
 - c. Kādi testpiemēri vēl ir nepieciešami?
 - d. Cik daudz laika būs nepieciešams testu izpildei?
4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast testa aptvērumu un sasniegt testa aptvērumu virs 80%,
 - b) novērtēt izprast testa aptvērumu un sasniegt testa aptvērumu virs 80%,
 - c) kontrolēt izprast testa aptvērumu un sasniegt testa aptvērumu virs 80%,
 - d) prognozēt izprast testa aptvērumu un sasniegt testa aptvērumu virs 80%.
5. Testa aptvērums standartā tiek aprakstīts kā procenti no sistēmas vai programmatūras faktiski veiktajām iespējām, darbības scenārijiem vai funkcijām, kas ir iekļautas saistītajos testa komplektos. Kopumā šī definīcija ir laba, taču tā nepasaka visu. *Testa aptvērums būtu jāiekļauj arī darbības scenāriju testēšana dažādās platformās, programmatūras koda testēšanas aptvērums un citas lietas, kas nav minētas definīcijā un nav aprakstītas atsevišķi.*

6. Testa aptvērums aprēķina pēc formulas (4.8), taču tās rezultātā netiek iegūti procenti, *tāpēc to būtu jāpapildina ar reizinātāju 100% vai arī jāmaina definīcija un jāapskata rezultāts, kā daļa.*
7. Formula satur divus mainīgos: faktisko un testkomplektos iekļauto sistēmas vai programmatūras iespēju, darbības scenāriju vai funkciju skaitu. Testkomplektos iekļautās iespējas, darbības scenāriji un funkcijas būs visas tās, kam būs izveidoti testpiemēri, bet faktiskās iespējas, darbības scenāriji un funkcijas būs tās, kuras aprakstītas prasību specifikācijā.
8. Standarts nenosaka to, kā interpretēt mēra rezultātu vai arī kādas vērtības būtu labas. *Programmatūru izstrādē sasniegt 100% testa aptvērums ir sarežģīti, kā arī tas ne vienmēr ir nepieciešams, lai programmatūru paradīti labāku. Parasti pieņem, ka labs testa aptvērums ir, piemēram, virs 80%. Standartu varētu papildināt un definēt labās un sliktās vērtības, piemēram, 100% – 90% testa aptvērums – ļoti labi, 90% – 80% testa aptvērums – labi, 80 – 70% testa aptvērums – vidēji, mazāk par 70% testa aptvērums – slikti.*
9. –
10. *Autoraprāt, šim mēram nav vajadzīgi papildus kritēriji.*
11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs būs testētājs, kurš veidos testpiemērus un testkomplektus, veiks testu izpildi un novērtēs testu aptvērums. Autoraprāt, mērs nenosaka to, vai visi gadījumi faktiski ir notestēti, jo formulā tiek ņemtas vērā visas faktiski sistēmā esošās funkcijas, bet nav aprakstīts, kā tiek apskatīti dažādi robežgadījumi, dažādas vides un citi faktori. Tā pat nav skaidrs kā novērtēt daļēji notestētu funkciju.*
12. Mēra aprēķināšanai nepieciešamie dati, lai veiktu aprēķinus ir reāli iegūstami, un to vākšanai nepieciešamais laiks un resursi ir atbilstoši iegūtajiem rezultātiem.
13. Cilvēciskais faktors
 - a. Konkrēts mērs.
 - b. Ieteicams.
 - c. *Objektīvs, ja eksistē kritēriji.*
 - d. Izmanto 75% aptaujāto uzņēmumu.
 - e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.
14. SMART kritēriju novērtējumu mēram testa aptvērums skatīt 4.10. tabulā.

SMART kritēriju novērtējums mēram testa aptvērums

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs testa aptvērums ir piemērots un svarīgs programmatūras novērtēšanai un atspoguļo testēšanas kvalitāti. Taču standartā būtu jāuzlabo mēra definīcija un formula, kā arī būtu nepieciešams norādīt, kā interpretēt mēra rezultātus. SMART kritēriju novērtējums ir labs, zemāk novērtētie kritēriji norāda uz galvenajiem trūkumiem – mēra definīcija un norādījumu trūkums par “labo” un “slikto” virzienu rezultātu interpretēšanā.

4.3.3.2. Pieejamība

Pieejamības mērus izmanto, lai novērtētu, cik lielā mērā sistēma, produkts vai komponents darbojas un ir pieejams, kad tas ir nepieciešams lietošanai [3]. Pieejamību raksturo mērs “(RAv-2-G) vidējais dīkstāves ilgums (*Mean down time*)”, ko pēc aptaujas rezultātiem, izmanto 75% aptaujāto uzņēmumu. Vidējais dīkstāves ilgums nosaka to, cik ilgi sistēma paliek nepieejama, ja rodas atteice, un to mēra dalot kopējo dīkstāves laiku ar novēroto avāriju skaitu. Tā aprēķināšanai tiek izmantota formula (4.9).

$$X = A/B \quad (4.9)$$

A – kopējais dīkstāves laiks

B – novēroto avāriju skaits

Rezultāta vērtības svārstās no 0 līdz bezgalībai un parasti mazāka vērtība ir labāka. Ārēji pieejamību var novērtēt pēc kopējā laika proporcijas, kuras laikā sistēma, produkts vai komponents atrodas strādājošā stāvoklī. Tāpēc pieejamību var raksturot kā brieduma (kas regulē atteižu biežumu), defektpieciētības un atkopjamības (kas nosaka dīkstāves ilgumu pēc katras atteices) kombināciju.

Vidējais dīkstāves ilgums mēra novērtēšana:

1. Vidējais dīkstāves ilgums ir svarīgs mērs, jo ir svarīgi, lai sistēma būtu nepieejama gala lietotājiem pēc iespējas mazāku laiku. Šī mēra ieinteresētā persona varētu būt produkta īpašnieks, programmatūras pasūtītājs jeb klients vai arī izstrādātājs.
2. Mērķis varētu būtu, piemēram, vidējais dīkstāves laiks ne ilgāks par 15 sekundēm.
3. Jautājumi, kas jāatbild, lai sasniegtu izvirzīto mērķi:
 - a. Kāds ir garākais dīkstāves ilgums?
 - b. Kāds ir īsākais dīkstāves ilgums?

- c. Cik bieži dīkstāves ilgums ir ilgāks par 15 sekundēm?
 - d. Ko darīt, lai dīkstāves ilgumu samazinātu?
4. Mērs palīdz veikt šādas funkcijas:
 - a) izprast dīkstāves vidējo ilgumu un sasniegt to, ka vidējais dīkstāves ilgums nav ilgāks par 15 sekundēm,
 - b) novērtēt dīkstāves vidējo ilgumu un sasniegt to, ka vidējais dīkstāves ilgums nav ilgāks par 15 sekundēm,
 - c) kontrolēt dīkstāves vidējo ilgumu un sasniegt to, ka vidējais dīkstāves ilgums nav ilgāks par 15 sekundēm,
 - d) prognozēt dīkstāves vidējo ilgumu un sasniegt to, ka vidējais dīkstāves ilgums nav ilgāks par 15 sekundēm.
 5. Vidējais dīkstāves ilgums tiek aprakstīts kā tas, cik ilgi sistēma paliek nepieejama, kad notiek kļūme, taču pašā aprakstā nav minēts tas, ka tiek apskatīts tieši vidējais dīkstāves ilgums starp visiem dīkstāves ilgumiem.
 6. Vidējo dīkstāves ilgumu aprēķina pēc formulas (4.9), un tā atbilst mēra nosaukumam.
 7. Formula satur divus mainīgos: kopējais dīkstāves ilgums un novēroto avāriju skaits. Novēroto avāriju skaits un kopējais dīkstāves ilgums ir konkrēti mainīgie, *taču kopējam dīkstāves ilgumam būtu nepieciešams norādīt mērvienību, piemēram, stundas vai minūtes.*
 8. Standartā ir minēts tikai tas, ka rezultāti var būt no 0 līdz bezgalībai, kā arī to, ka parasti mazāka vērtība būs labāka.
 9. –
 10. *Autoraprāt, šim mēram nav vajadzīgi papildus kritēriji, taču kādā pārskatā būtu interesanti redzēt īsāko un garāko dīkstāves laiku.*
 11. *Par šim mēram nepieciešamo datu iegūšanu atbildīgs varētu būt kāds no izstrādātājiem vai testētājiem kā arī kāda no ieinteresētajām personām. Dīkstāves ilgums ir iegūstams tehniski no izstrādātās programmatūras datiem vai arī no kādas lietotnes, kas nodrošina šo informāciju. Formulai (4.9) nepieciešamie dati ir reāli iegūstami, bet to iegūšanai varētu tikt izmantotai dažādi tehniski veidi vai lietotnes. Datu iegūšanai nepieciešamais laiks un resursi ir atbilstoši iegūstamajiem rezultātiem.*
 12. Cilvēciskais faktors
 - a. Vispārīgs mērs.

- b. Ieteicams.
- c. *Objektīvs un salīdzināms starp dažādām programmatūrām.*
- d. Izmanto 75% aptaujāto uzņēmumu.
- e. Mēra rezultāta iegūšanai nepieciešamais laiks, izmaksas un resursi ir līdzsvarā ar kvalitāti un iegūstamo rezultātu.

13. SMART kritēriju novērtējumu mēram vidējais dīkstāves ilgums skatīt 4.11. tabulā.

4.11. tabula

SMART kritēriju novērtējums mēram vidējais dīkstāves ilgums

S				M				A				R				T			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

Mērs vidējais dīkstāves ilgums ir piemērots un svarīgs programmatūras novērtēšanai un ir izmantojams dažādu programmatūru salīdzināšanai. Standartā būtu jāprecizē mēra definīcija, lai tā atspoguļotu to, ka mērs norāda tieši vidējo dīkstāves ilgumu, kā arī formulas mainīgajam kopējais dīkstāves laiks būtu jānorāda konkrēta mērvienība, lai rezultātus būtu ērtāk salīdzināt. SMART kritēriji kopumā ir labi. M kritērijs ir vidējs, kas atspoguļo to, ka nepieciešams papildināt mēra definīciju.

4.3.5. Programmatūras produkta kvalitātes mēri

Standartā ISO/IEC 25023:2016 kopā ir aprakstīti 86 sistēmas/programmatūras produkta kvalitātes mēri. Tikai vienu no tiem – vidējais atbildes laiks – izmanto vairāk nekā 80% aptaujāto uzņēmumu un vēl sešus – vidējais atbildes laiks, vidējais darba izpildes ilgums, funkcionālais pārklājums, atteicu biežums, testa aptvērums un vidējais dīkstāves ilgums – izmanto 75% aptaujāto uzņēmumu. Pārējie mēri tiek izmantoti retāk. Visvairāk izmantotie mēri apraksta galvenokārt tādas raksturlielumus kā veiktspējas efektivitāte un uzticamība, kā arī funkcionālā atbilstība. Veiktspējas efektivitāte un uzticamība pēc standartā pieejamās informācijas un autores novērtējuma ir uzskatāmi par uzticamiem un objektīviem raksturlielumiem, un lielākā daļa funkcionālās atbilstības apakšraksturlielumi arī ir uzskatāmi par uzticamiem un objektīviem.

No novērtētajiem mēriem, seši ir vispārīgi, un tikai viens ir konkrēts mērs. Tā pat visi šie mēri ir ieteicami, un divi no tiem ir ļoti ieteicami mēri. Apskatītie mēri ir objektīvi, bet ne visi ir salīdzināmi starp dažādām programmatūrām. Lai šos mērus veiksmīgi varētu pielietot programmatūru novērtēšanā un salīdzināšanā, būtu jāuzlabo mēru definīcijas, lai tās ir skaidras un nepārprotamas, kā arī jāprecizē definīcijas un izmantojamās mērvienības. Tā pat trūkst

norādes par to, kā tieši šos mērus interpretēt, lai rezultātus varētu analizēt un salīdzināt starp dažādām programmatūrām.

Veicot kvalitātes novērtēšanu atbilstoši šim standartam, katram izvēlētajam raksturlielumam vai apakšraksturlielumam jāizmanto visi noteiktie vispārīgie kvalitātes mēri, ja kāds netiek izmantots, tad tas ir jāpamato. Ja tas atbilst konkrētajai situācijai, tad var pielietot kādu no konkrētajiem mēriem. Jā kāds mērs tiek izmantots, vai arī tiek izmantots kāds mērs, kas standartā nav aprakstīts, tad tas ir jāpamato.

Standarta darbības sfēra nosaka, ka standarts nepiešķir mēru vērtību diapazonus vērtētajiem līmeņiem vai atbilstības pakāpēm, jo šīs vērtības tiek definētas, pamatojoties uz sistēmas, produkta vai produkta daļas raksturu, un ir atkarīgi no dažādiem faktoriem, piemēram, programmatūras kategorijas, integritātes līmeņa un lietotāju vajadzībām. Lielākajai daļai kvalitātes mēru tiek izmantota funkcija, kas normalizē rezultāta vērtību diapazonā no 0,0 līdz 1,0. Jo tuvāk 1,0 jo labāk. Daži kvalitātes mēri dod rezultātu, kas ir salīdzināms ar mērķa vērtību, kas jānosaka, kā daļa no prasībām. Atsevišķi kvalitātes mēri, piemēram, vidējais reakcijas laiks var būt grūti interpretējami. Lai tos varētu vieglāk saprast un interpretēt, var veikt salīdzināšanu ar noteiktām prasībām, piemēram, maksimālais pieļaujamais reakcijas laiks ir 0,5 sekundes. Tā pat var veikt salīdzināšanu ar etalonu tam pašam vai līdzīgam produktam vai sistēmai, ko izmanto vienam un tam pašam mērķim, piemēram, jaunās sistēmas vidējais reakcijas laiks, nepārsniedz vecās sistēmas vidējo reakcijas laiku, vai arī laika tendenču salīdzināšanu, piemēram, kā vidējais reakcijas laiks mainās dienas laikā.

Kopumā standartā aprakstītie mēri ir piemēroti, lai novērtētu sistēmas/programmatūras produkta kvalitāti, taču tie būtu jāpapildina, lai to pielietošana būtu vienāda, un iegūtie rezultāti būtu salīdzināmi starp dažādām programmatūrām. Lai gan standartā produkta kvalitāti raksturo 86 mēri, no kuriem 48 ir vispārīgi mēri, produkta kvalitātes raksturlielumus varētu novērtēt arī izmantojot mazāku skaitu kvalitātes mēru, tādā veidā koncertējoties uz svarīgāko.

REZULTĀTI UN SECINĀJUMI

Darba izpētes laikā tika veikta standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 pieejamo mēru analīze. Tās rezultātā tika noskaidrots, ka 24 no 36 lietojumkvalitātes mēriem ir vispārīgi mēri, kas ir vispārēji piemērojami un izmantojami dažādās situācijās. Taču šie mēri nav uzskatāmi par pilnībā objektīviem un tādi, kā tie ir aprakstīti standartos, nav izmantojami dažādu programmatūru salīdzināšanai. Sistēmas/programmatūras produkta kvalitātes mēri ir vairāk objektīvi un savstarpēji salīdzināmi nekā lietojumkvalitātes mēri. Starp 86 produkta kvalitātes mēriem 48 ir vispārīgi pielietojami mēri. Tiem tiek noteikts arī rekomendācijas līmenis, kur 30 mēri ir ļoti ieteicami, 35 – ieteicami un 21 – izmantojams pēc lietotāja ieskatiem. Taču tikai 12 produkta kvalitātes mēri ir vispārīgi, ļoti ieteicami, objektīvi un salīdzināmi starp dažādām programmatūrām.

Aptaujājot uzņēmumus, kas Latvijā ir saistīti ar IT nozari, tika iegūti rezultāti, kas norāda, ka tikai divus mērus no standartiem izmanto vairāk nekā 80% aptaujāto uzņēmumu. Viens no tiem ir lietojumkvalitātes mērs peļņa no ieguldījuma, bet otrs ir produkta kvalitātes mērs vispārējā apmierinātība. Visbiežāk tiek izmantoti mēri, kas raksturo veikspējas efektivitāti un uzticamību. Tā pat no aptaujas rezultātiem izriet tas, ka vienu ceturto daļu no mēriem izmanto mazāk nekā 20% aptaujāto uzņēmumu. Aptaujas rezultāti norāda uz to, ka programmatūras izstrādātāju un testētāju izmantotie kvalitātes mēri dažādos uzņēmumos ir atšķirīgi. Praktiski nozarē pielietotie mēri ne vienmēr ir objektīvi un ticami, kā arī no aptaujas komentāriem var secināt to, ka kvalitātes mēri tiek ņemti vērā izstrādes procesā, bet ne vienmēr tiek atsevišķi mērīti.

Lai atrastu metodoloģiju pašu mēru novērtēšanai, tika veikta dažādu zinātnisko rakstu izpēte par šo tēmu. Rezultātā tika izvēlēta 12 soļu metodoloģija, lai iegūtu derīgus mērus programmatūras novērtēšanai, kas nodrošina praktisku un sistemātisku programmatūras mēru izvēles, noformēšanas un ieviešanas metodi no sākuma līdz beigām. Papildus tika veikta arī mēru novērtēšana pēc SMART kritērijiem.

Programmatūras novērtēšanas mēriem ir jāsniedz informācija, kas nepieciešama programmatūras projektu pārvaldīšanai un programmatūras inženierijas procesu un pakalpojumu uzlabošanai [15]. Kvalitātes mēriem ir jābūt vienkāršiem, viegli definējamiem, skaidriem, derīgiem, sapratīgiem un ar konkrētu mērķi [17]. Veicot mēru novērtēšanu pēc izvēlētās metodoloģijas, tika secināts, ka standartos aprakstītie mēri kopumā ir derīgi programmatūras kvalitātes novērtēšanai, taču liela daļa mēru ir subjektīvi. Standartos aprakstītajiem mēriem ir iespējams gan piemērot ieinteresēto personu, gan izvirzīt mērķus, taču dažādām programmatūrām tie var būt atšķirīgi. Lai standartos aprakstītos mērus varētu

izmantot programmatūru kvalitātes novērtēšanai un salīdzināšanai galvenokārt būtu jāprecizē mēru definīcijas un formulās izmantotie mainīgie, lai tie būtu skaidri un nepārprotami. Tā pat svarīgi būtu norādīt, kā interpretēt iegūtos rezultātus, norādot vērtību “labos” un “sliktos” virzienus. Abos apskatītajos standartos kopā ir aprakstīti 122 mēri, kas raksturo 13 kvalitātes raksturlielumus. Praktiski, lai novērtētu programmatūras kvalitāti, varētu izmantot mazāku skaitu kvalitātes mēru, iegūstot pilnvērtīgu kvalitātes novērtējumu. Tādā veidā tiktu radīta vienota sistēma, pēc kuras varētu novērtēt jebkuru programmatūru, un ja programmatūru kvalitātes novērtēšana tiktu veikta pēc viena parauga, to salīdzināšana būtu ērtāka un vienkāršāka.

Maģistra darba galvenais mērķis bija novērtēt standartos ISO/IEC 25022:2016 un ISO/IEC 25023:2016 aprakstītos mērus, lai noteiktu, vai tie ir praktiski pielietojami programmatūru lietojumkvalitātes un produkta kvalitātes novērtēšanā un salīdzināšanā. Autoraprāt, mērķi ir izdevies sasniegt. Kaut gan padziļinātāk tika novērtēta daļa no mēriem, var secināt, ka kopumā standartos mēri ir aprakstīti labi, taču tos nepieciešams papildināt, lai to izmantošana būtu viennozīmīga, tie būtu objektīvi un rezultātus varētu salīdzināt starp dažādām programmatūrām. Praktiski novērtēt programmatūras kvalitāti varētu arī izmantojot daļu no visiem standartos aprakstītajiem mēriem.

Darba ietvaros tika novērtēta daļa no standartos aprakstītajiem mēriem, tāpēc darbu varētu turpināt, novērtējot pārējos standartos aprakstītos mērus. Rezultātā iegūstot plašāku informāciju par kvalitātes raksturlielumu un to novērtēšanas mēru ticamību un izmantošanu praksē. Autoraprāt, būtu nepieciešams izveidot paraugu tam, kā veikt programmatūras lietojumkvalitātes un produkta kvalitātes novērtēšanu, lai dažādu programmatūru kvalitāti varētu salīdzināt, radot līdzsvaru starp izmaksām, kvalitāti un laiku. Ņemot vērā standartos minēto piebildi par to, ka kvalitātes mēru saraksts nav pabeigts un var tikt pārskatīts standartu nākamajās versijās, iegūtos rezultātus varētu sniegt kā atsauksmes standartu izstrādātājiem.

PATEICĪBAS

Vēlos pateikties Dr. sc. comp. Vinetai Arnicānei un Dr.habil.dat. Jurim Borzovam par līdzdalību zinātniskā raksta “Vai mēs tiešām zinām, kā izmērīt programmatūras kvalitāti?” (“*Do We Really Know How to Measure Software Quality*”) izstrādē, kā par pamudinājumu piedalīties 14. starptautiskajā Baltijas konferencē par datu bāzēm un informācijas sistēmām (*Baltic DB&IS 2020*).

AVOTU UN IZMANTOTĀS LITERATŪRAS SARAKSTS

1. ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. International Organization for Standardization, Geneva, Switzerland (2011)
2. ISO/IEC 25022:2016 Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – Measurement of quality in use. International Organization for Standardization, Geneva, Switzerland (2016)
3. ISO/IEC 25023:2016 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality. International Organization for Standardization, Geneva, Switzerland (2016)
4. V. Arnicanē, J. Borzovs, A. Nesaule-Erina, “Do We Really Know How to Measure Software Quality,” in Proc. 14th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2020). Pieejams: <http://ceur-ws.org/Vol-2620/paper2.pdf>
5. Akadēmiskā terminu datubāze AkadTerm Pieejams: <http://digitalis.lv/term.php> [aplūkots 06.05.2021.]
6. Akadēmiskā terminu datubāze AkadTerm Pieejams: <http://digitalis.lv/term.php?term=turnaround%20time&lang=EN> [aplūkots 14.04.2021.]
7. Akadēmiskā terminu datubāze AkadTerm Pieejams: <http://digitalis.lv/term.php?term=atbildes%20laiks&lang=LV> [aplūkots 14.04.2021.]
8. Dr. L. Rosenberg, T. Hammer un J. Shaw, “Software Metrics and Reliability” Pieejams: <https://www.cs.du.edu/~snarayan/sada/teaching/COMP3705/lecture/p1/smandrelaib.pdf> [aplūkots 23.01.2021.]
9. H. Barkmann, R. Lincke un W. Löwe, “Quantitative Evaluation of Software Quality Metrics in Open-Source Projects” Pieejams: <http://www.arisa.se/files/BLL-09.pdf> [aplūkots 23.01.2021.]
10. J. Sauro “10 THINGS TO KNOW ABOUT THE SINGLE EASE QUESTION (SEQ)” Pieejams: <https://measuringu.com/seq10/> [aplūkots 15.03.2021.]
11. K. Chopra un M. Sachdeva, “Evaluation of Software Metrics for Software Projects” Pieejams: <https://core.ac.uk/download/pdf/322473445.pdf> [aplūkots 23.01.2021.]

12. K. Speights “Return on Investment: What to Expect” Pieejams:
<https://www.fool.com/investing/how-to-invest/stocks/good-return-on-investment/#:~:text=Most%20investors%20would%20view%20an,will%20generate%20significantly%20higher%20returns> [aplūkots 05.03.2021.]
13. K.P. Srinivasan un T. Devi, “Software Metrics Validation Methodologies in Software Engineering” Pieejams:
https://www.researchgate.net/publication/284474204_Software_Metrics_Validation_Methodologies_in_Software_Engineering [aplūkots 23.01.2021.]
14. L. Buglione, A. Abran “Assessment of Measurement Indicators in Software Process Improvement Frameworks” Pieejams:
https://www.academia.edu/16763946/Assessment_of_Measurement_Indicators_in_Software_Process_Improvement_Frameworks [aplūkots 05.03.2021.]
15. L. Westfallm “12 Steps to Useful Software Metrics” Pieejams:
<https://core.ac.uk/download/pdf/21723057.pdf> [aplūkots 23.01.2021.]
16. M. F. Bertoa, A. Vallecillo, “Usability metrics for software components” Pieejams:
https://www.researchgate.net/publication/228584811_Usability_metrics_for_software_components [aplūkots 08.02.2021.]
17. M. Jah, “Software Metrics – Usability and Evaluation of Software Quality” Pieejams:
<https://www.diva-portal.org/smash/get/diva2:215120/fulltext01.pdf> [aplūkots 23.01.2021.]
18. S. Simic “7 Ways to Reduce Server Response Time” Pieejams:
<https://phoenixnap.com/kb/reduce-server-response-time#:~:text=satisfying%20user%20experience.,What%20is%20a%20Good%20Server%20Response%20Time%3F,is%20consistent%20for%20all%20users> [aplūkots 22.04.2021.]
19. “What Is Net Promoter?” Pieejams: <https://www.netpromoter.com/know/> [aplūkots 15.03.2021.]
20. “What is the average server response time?” Pieejams:
<https://www.littledata.io/average/server-response-time> [aplūkots 22.04.2021.]
21. “Why Server Response Time Is Important” Pieejams: <https://www.hrank.com/why-server-response-time-is-important> [aplūkots 22.04.2021.]

PIELIKUMI

Do We Really Know How to Measure Software Quality?

Vineta Arnicane^[0000-0003-3942-9229], Juris Borzovs^[0000-0001-7009-6384],
Anete Nesaule-Erina^[0000-0001-7244-2148]

Faculty of Computing, University of Latvia
Raina blvd. 19, Riga, Latvia
{vineta.arnicane, juris.borzovs}@lu.lv,
anete.nesaule@gmail.com

Abstract. Particular return to McCall’s et al. seminal work has been done in recent international standards ISO/IEC 25022:2016 and ISO/IEC 25023:2016. 122 lower level measures (metrics in McCall’s et al. terminology) were introduced. Authors of standards themselves classify quality measures as Highly Recommended, Recommended, Used at user’s Discretion and Generic, Specific. In this paper, quality measures are additionally classified also from the point of view of objectivity of measurement and of usage in industry.

Keywords: Software engineering · Software quality models · Quality characteristics · Quality metrics · Quality measures.

1 Introduction

We are not going to review software quality models because the authors of [12] already have done it. As James A. McCall, wrote in Wiley Online Library in 2002 [11]:

“All engineering disciplines have some basis in measurement. Software quality measurement frameworks have been developed to provide a measurement approach for software engineering. The frameworks have a common structure. At the highest level are quality factors, that comprise a definition of software quality and represent attributes or characteristics of the software, relate to its overall quality. The second level provides the criteria or software attributes that relate to the factors, and their existence provides the related characteristics of quality. The third level provides the “metrics” or measurements that measure the degree to which those attributes exist.

The use of software quality factors and metrics has become a common practice in the industry, although the applications are not consistent, systematic or typically applied across projects or organizations.” In their seminal paper [10] the authors introduced 11 quality factors, 23 criteria (i.e., sub-factors), and 42 metrics to measure software quality at the U.S. Air Force Electronic Systems Division’s and Rome Air Development Center’s mission to provide standards and technical guidance to software acquisition managers. Two types of metrics

were used. The first type, like a ruler, is a relative quantity measure. The second type is a binary measure which determines the existence (1) or absence (0) of something. The units of the metric were chosen as the ratio of actual occurrences to the possible number of occurrences. So, a total quality of a software product could be characterized as a sum of values of metrics.

For unknown reasons, idea of quantitative evaluation of software was not incorporated into international standard neither in its first version in 1991 [2], nor in its subsequent revisions [3–6]. This view is also shared by [1]: “The metrics in the lower level of the McCall’s, Boehm’s, Dromey’s and FURPS quality models are neither clearly nor completely defined and connected.” (However, to our opinion, it is not true regarding McCall’s lower level definitions.)

Particular return to McCall’s et al. seminal work has been done in recent standards ISO/IEC 25022:2016 [8] and ISO/IEC 25023:2016 [9]. 122 lower level measures (metrics in McCall’s et al. terminology) were introduced. In this paper, quality measures are additionally classified also from the point of view of objectivity of measurement and of usage in industry.

In Section 2 generally explains quality models provided by ISO/IEC 25022:2016 and ISO/IEC 25023:2016 standards. There is an investigation about objectivity of quality measures described in Section 3. The Section 4 provides insight into the capacity of measures to assess subcharacteristics and therefore characteristics of quality models. There is survey about usage of quality measures in the enterprises of Latvia elaborated in Section 5. Section 6 contains results of survey and Section 7 gives insight in conclusions and future work.

2 What do ISO/IEC 25022:2016 and ISO/IEC 25023:2016 offer?

ISO/IEC 25022:2016 and ISO/IEC 25023:2016 are the part of series of International Standards from ISO/IEC 25000 to ISO/IEC 25099 entitled *Systems and software engineering – Systems and software Quality Requirements and Evaluation* with the acronym *SQuaRE*. They give the measures which are supposed as appropriate to assess the extent or degree at what system or product satisfies each of subcharacteristics. There are two quality models defined in the standard ISO/IEC 25010:2011 [7] - *Quality in Use model* and *System/software product quality model*. Each of these models consists of quality characteristics and their subcharacteristics as shown in Fig. 3 and Fig. 4.

Each measure in ISO/IEC 25022:2016 and ISO/IEC 25023:2016 is classified as generic or specific. Generic are generally applicable, can be used in a wide range of situations while Specific are specialised for specific needs. For instance, generic measure is *MRe-1-G Reusability of assets - how many assets in a system can be reusable*. It is measured as ratio between assets which are designed and implemented to be reusable and assets in a system. As example for specific measure can be highlighted *REc-6-S Website visitors converted to customers - the proportion of visitors to a particular web page(s) who become customers*. It is measured using business analytics.

The system / software product quality model's measures are classified according to the recommendation level [9]. The authors of the standard ISO/IEC 25023:2016 advice to use highly recommended measures always. Accordingly recommended measures should be used as quality measures when it is appropriate. At the same time measures with unknown reliability are meant to be used at user's discretion as a reference when developing a new quality measure [9].

3 How objective are measures?

The measure can be applicable and reliable only if its measurements can be obtained objectively. We analysed measures mentioned in ISO/IEC 25022:2016 and ISO/IEC 25023:2016 and obtained assessment is shown in the Fig. 1. For

Ef-1-G	SPI-1-G	FCp-1-G	CIn-2-G	UEp-1-G	SCo-2-G	Md-2-G
Ef-2-S	SCo-1-G	FCr-1-G	CIn-3-S	UEp-2-S	SCo-3-S	Md-3-S
Ef-3-G	REc-1-G	FAP-1-G	UAp-1-G	UEp-3-S	SIn-1-G	MTE-1-G
Ef-4-G	REc-2-G	FAP-2-G	UAp-2-S	UIn-1-S	SIn-2-G	MTE-2-S
Ef-5-G	REc-3-G	PTb-1-G	UAp-3-S	UAc-1-G	SIn-3-S	MTE-3-S
Ey-1-G	REc-4-G	PTb-2-G	ULe-1-G	UAc-2-S	SNo-1-G	PAd-1-G
Ey-2-S	REc-5-S	PTb-3-G	ULe-2-S	RMa-1-G	SAC-1-G	PAd-2-G
Ey-3-S	REc-6-S	PTb-4-G	ULe-3-S	RMa-2-G	SAC-2-S	PAd-3-S
Ey-4-S	REc-7-S	PTb-5-G	ULe-4-S	RMa-3-G	SAu-1-G	PIn-1-G
Ey-5-S	REc-8-G	PRu-1-G	UOp-1-G	RMa-4-S	SAu-2-S	PIn-2-G
Ef-6-S	RHe-1-G	PRu-2-G	UOp-2-G	RAv-1-G	MMo-1-G	PRe-1-G
Sus-1-G	RHe-2-G	PRu-3-G	UOp-3-S	RAv-2-G	MMo-2-S	PRe-2-S
Sus-2-G	RHe-3-G	PRu-4-S	UOp-4-S	RfT-1-G	MRe-1-G	PRe-3-S
Sus-3-G	REn-1-G	PCa-1-G	UOp-5-S	RfT-2-S	MRe-2-S	PRe-4-S
Sus-4-G	COm-1-G	PCa-2-G	UOp-6-S	RfT-3-S	MAN-1-G	
Sus-5-G	CFI-1-S	PCa-3-S	UOp-7-S	RRe-1-G	MAN-2-S	
Sus-6-G	CFI-2-S	CCo-1-G	UOp-8-S	RRe-2-S	MAN-3-S	
STR-1-G	CFI-3-S	Qn-1-G	UOp-9-S	SCo-1-G	Md-1-G	

Legend:

- Subjective (questionnaires, Likert, etc.)
- Objective, but the result depends on the end-user qualification
- Objective for a specific product, but comparisons only relative
- Objective, if industry criteria exist
- Objective, can be used in comparison

Fig. 1. Assessment of possibility to obtain objective measurements for measures.

instance, we assessed as objective measure *RAv-1-G Test function completeness - how completely are test functions and facilities implemented*. Measure is computed as as ratio between system operation time actually provided and system operation time specified in the operation schedule. This measure is objective because both parameters of ratio come from either from statistics or from schedule. The measurements can be used in order to compare two products.

Sus-2-G Satisfaction with features - the satisfaction of the user with specific system features is the example of subjective measure. It is measured using questionnaires. Judgement of people about system's features can greatly depend on their mood, level of tiredness, attitude to system, etc..

Measure *Ef-3-G Errors in task - the number of errors made by the user during a task* is the instance of measures which are measurable objectively but at the same time the result depends heavily on the end-user qualification in the field as well as the level of training to work with the system under test. That also is the reason why the obtained results are hardly applicable to comparison different systems or one system in the different stages of development.

Better situation are with next group of measures which can be measured objectively but the results are specific of product and can be used for comparison in the form of percents. For instance, measure *REc-2-G Time to achieve return on investment - the time taken to achieve the expected return on investment* which can be calculated by business analytics based on careful analysis, though, there is also a good dose of more or less subjective assumptions. Measure *MTe-*

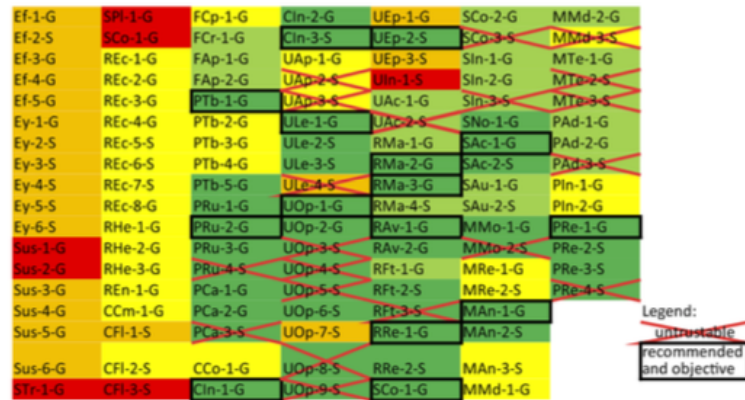


Fig. 2. Coloured quality characteristics for software or software product quality.

1-G Test function completeness - how completely are test functions and facilities implemented which is calculated as ratio between count of test functions implementer as specified and the count of test functions required, is a representative of group of measures which can be measured objectively if there exist industry criteria which help to understand how many test functions should be required. Otherwise this measure becomes very subjective.

4 Can measures given in ISO/IEC 25022:2016 and ISO/IEC 25023:2016 really assess the quality subcharacteristics?

Let us try to understand which of measures are usable from practical point of view - which are objective and reliable. If we put together our findings about objectivity of quality measures with recommendations of standard we would have obtained situation shown in the Fig. 2. We put red crosses on measures that the authors of standard ISO/IEC 25023:2016 considered as measures with unknown reliability. Next step - we put black box around those measures which are objective according to our analysis and are highly recommended by authors of standard.

If to look on Quality in use model's measures of subcharacteristics there are no any objectively measurable measure. That could be expected to happen

(see Fig. 3). Accordingly also quality characteristics cannot be well measured if their subcharacteristics are not such. The situation with Software product

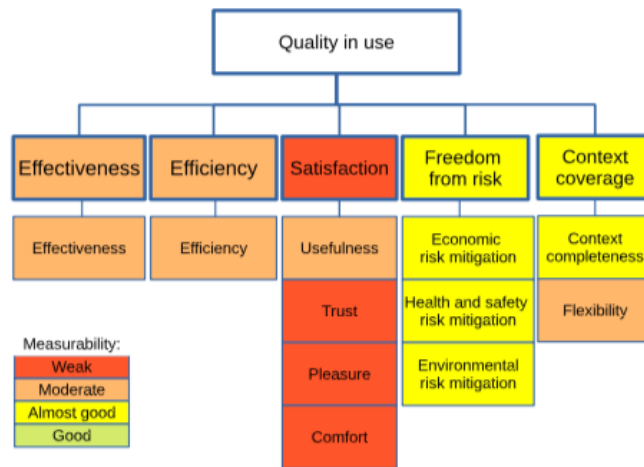


Fig. 3. Measurability of quality characteristics for Quality in use model.

quality model looks much better (see Fig. 4). The most difficult measurable are usability subcharacteristics while reliability, security and performance efficiency subcharacteristics looks measurable and reliable.

5 What measures are used in practise in Latvia?

5.1 Survey Objectives

The University of Latvia conducted a survey on measures used in the Latvia's industry of software development.

The primary objective of the survey was to determine measures of quality characteristics used in Latvia when they carry out software development and maintenance activities.

The aim was to find out which quality characteristics and subcharacteristics are currently used in the IT industry of Latvia and whether all quality measures included in ISO/IEC 25022:2016 and ISO/IEC 25023:2016 standards are well known by quality assurance specialists, testers and developers.

5.2 Survey Description

The survey targeted senior employees involved with quality assurance, testing or development in software development companies or departments of software development in enterprises with another kind of main business.

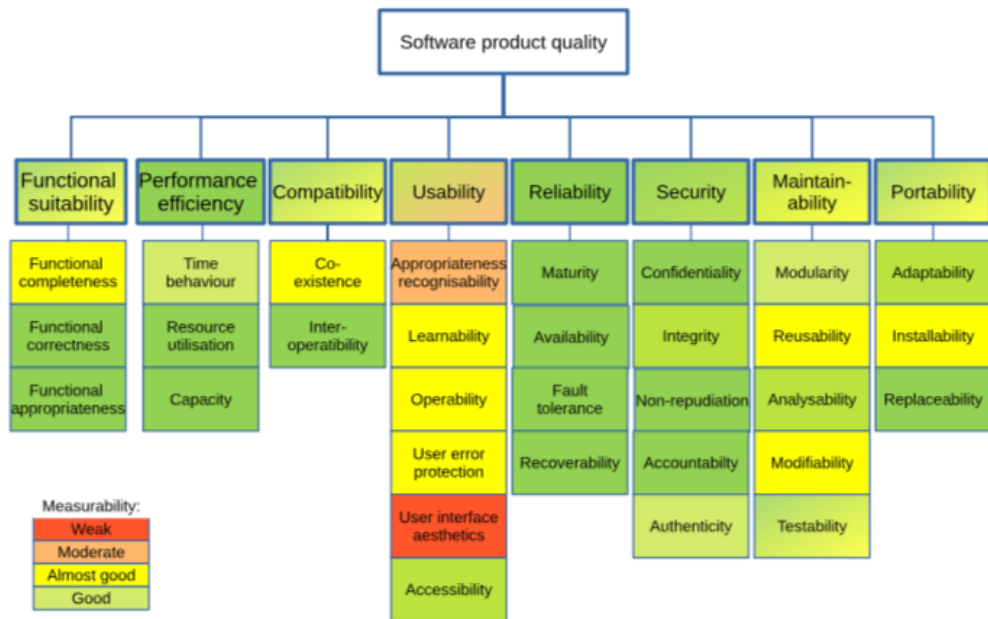


Fig. 4. Measurability of quality characteristics for software or software product quality.

There were three parts in the survey. The first part was an introductory section where the aims of survey and way how to fill it was described. The second part of the survey comprised questions about the company. The third part was the main part of the survey – 122 measures of 42 quality subcharacteristics according to standards ISO/IEC 25022:2016 and ISO/IEC 25023:2016.

The respondents of the survey were invited to complete the survey online. Each measure had predefined answers: 1)Yes; 2)No; 3)Other (*open type answer to give some specific answer*). Respondents were informed that survey terms are taken from standards ISO/IEC 25022:2016 and ISO/IEC 25023:2016. For each measure the definition was given, formulae was explained and the method of measuring was described. Confidentiality and privacy were assured to all respondents and the organization that they represented.

6 Analysis and Summary of Survey Findings

As a result, a total of 17 companies participated in the survey, representing more than 4000 IT employees including almost 800 testers. The survey results show which measures of system and software product quality as well as measure of quality in use are used in the IT industry of Latvia.

Summary of survey's results is shown in the Fig. 5. Measures are color coded in five groups. The red ones are measures that are used rarely - in less than 20% of organizations, while yellow ones are used moderately and green ones frequently or very frequently - even in more than 80% of organizations.

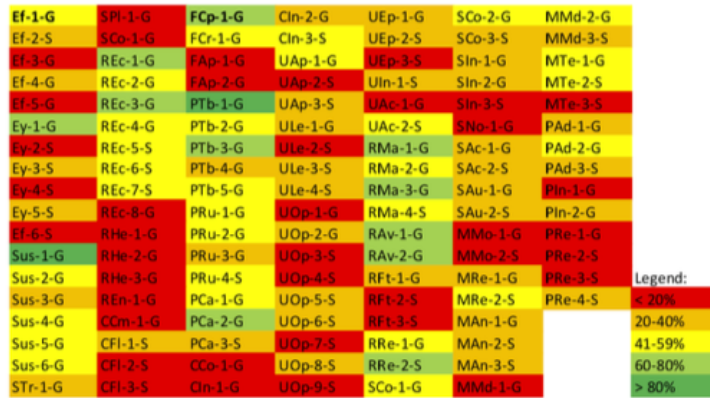


Fig. 5. Results - at what extent measures are used in IT industry of Latvia.

As an example of rarely used measure in surveyed organizations can be mentioned *CIn-1-G Data formats exchangeability - what proportion of the specified data formats is exchangeable with other software or systems*. It is measured as ratio between data formats exchangeable with other software or systems and data formats specified to be exchangeable.

Next group of measures is colored in orange - these measures are used in 20-40% of surveyed organizations. As representative of moderately used measures colored in yellow are used in 41-59% of surveyed organizations. Green color indicates that measure is used frequently. Ones colored in light green are used in 60-89% of surveyed organizations.

The most frequently used measures are colored in dark green. There are only two measures used by more than 80% of surveyed organizations. The one is *PTb-1-G Mean response time* and the other is *Sus-1-G overall satisfaction - the overall satisfaction of the user* measured using questionnaires for users of system or software product.

7 Conclusions and Future Work

The authors of standards ISO/IEC 25022:2016 and ISO/IEC 25023:2016 have done hard work trying to define measures which could allow to assess subcharacteristics and therefore quality characteristics for Software in use and Software product quality models. They divide measures in two sets - generic for usage in wide range of situations while specific for specialised needs. For the Product quality model they give also classification of measures into sets of highly recommended, recommended and measures with unknown reliability.

If measures of both quality models are assessed from the objectivity viewpoint then we see that Quality in use measures all are more or less subjective while the most of Product quality model's measures are objective.

Survey about actual usage of quality measures in IT industry of Latvia shows that only two measures are used in more than 89% of surveyed enterprises. At the

same time one third part of all measures mentioned in standards is used in less than 20% of enterprises. Surprisingly, used measures are not mandatory reliable and objective. This fact urges to continue research in order to understand if it is enough for enterprises with those measures which they are using and, if not, why they do not use other measures.

8 Acknowledgements

The work described in this paper was supported by the project “Innovative information technologies” at the University of Latvia. We are grateful to all respondents of the survey. Any errors or omissions are the faults of the authors.

References

1. Al-Qutaish, Rafa E. Quality Models in Software Engineering Literature: An Analytical and Comparative Study. *Journal of American Science* **6**(3), 166–175 (2010)
2. ISO/IEC 9126:1991 Software engineering – Product quality. ISO/IEC, (1991)
3. ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model. International Organization for Standardization, Geneva, Switzerland (2001)
4. ISO/IEC TR 9126-2:2003. Software Engineering - Product Quality - Part 2: External Metrics. International Organization for Standardization, Geneva, Switzerland (2003)
5. ISO/IEC TR 9126-3: 2003. Software Engineering - Product Quality - Part 3: Internal Metrics. International Organization for Standardization, Geneva, Switzerland (2003)
6. ISO/IEC TR 9126-4: 2004. Software Engineering - Product Quality - Part 4: Quality in Use Metrics. International Organization for Standardization, Geneva, Switzerland (2004)
7. ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. International Organization for Standardization, Geneva, Switzerland (2011)
8. ISO/IEC 25022:2016 Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – Measurement of quality in use. International Organization for Standardization, Geneva, Switzerland (2016)
9. ISO/IEC 25023:2016 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality. International Organization for Standardization, Geneva, Switzerland (2016)
10. McCall, J., Richards, P., Walters, G.: Factors in Software Quality, three volumes, NTIS AD-A049-014, 015, 055 (1977)
11. McCall, James A.: Quality Factors. In: Wiley Online Library (2002), <https://doi.org/10.1002/0471028959.sof265>. Last accessed 26 Feb 2020
12. Miguel, José P., Mauricio, D., Rodríguez, G.: A Review of Software Quality Models for the Evaluation of Software Products. *International Journal of Software Engineering Applications (IJSEA)*, **5**(6), 31–53, (2014)

Maģistra darbs „**Programmatūras kvalitātes mērišana un salīdzināšana, izmantojot ISO/IEC 25022:2016 un ISO/IEC 25023:2016 standartus**” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Darba teksta galīgā versija izgatavota 24.05.2021.

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

Autore: Anete Kristīne Nesaule-Eriņa _____, 24.05.2021.

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **piemērotu/nepiemērotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: prof. Dr.habil.dat. Juris Borzovs _____, 24.05.2021.

Darbs iesniegts **maģistratūras sekretariāta** 24.05.2021.

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: _____.

Recenzents: doc., Dr. dat. Anastasija Ņikiforova.

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

01.06.2021. prot. Nr. _____

Komisijas sekretārs(-e): _____