

Latvijas Lauksaimniecības universitāte
Informācijas Tehnoloģiju fakultāte
Datoru sistēmu katedra

Sandra Sproģe

**INFORMĀCIJAS TEHNOLOĢIJU STUDIJU UN
IZSTRĀDES VIENOTAIS KVALITĀTES MODELIS**

Promocijas darba
KOPSAVILKUMS
doktora grāda ieguvei
Informācijas tehnoloģiju nozarē (Dr.sc.ing.)



IEGULDĪJUMS TAVĀ NĀKOTNĒ



Promocijas darba izstrāde un noformēšana līdzfinansēta no
Eiropas Savienības Sociālā fonda

Sandra Sproģe _____

Paraksts

Jelgava 2014

INFORMĀCIJA

Darba izpildes vieta: Latvijas Lauksaimniecības universitāte, Informācijas tehnoloģiju fakultāte, Datoru sistēmu katedra, Liela iela 2, Jelgava, Latvija.

Eksperimentālā darba izpildes vieta: Latvijas Lauksaimniecības universitāte, Informācijas tehnoloģiju fakultāte, Datoru sistēmu katedra, Liela iela 2, Jelgava, Latvija.

Promocijas darba zinātniskais vadītājs: Latvijas Lauksaimniecības universitātes profesore, Dr.sc.comp. Rudīte Čevere.

Darbs akceptēts LLU Informācijas tehnoloģiju fakultātes Datoru sistēmu katedras paplašinātā akadēmiskā sēdē 2014. gada 28. maijā. Protokols Nr. 5.

Promocijas darbs izstrādāts ar Eiropas Sociālā fonda projekta “Atbalsts LLU doktora studiju īstenošanai” vienošanās Nr. 2009/0180/1DP/1.1.2.1.2/09/IPIA/VIAA/017 atbalstu.

Oficīlie recenzenti

1. Rīgas Tehniskās universitātes profesore, Dr.sc.ing. Larisa Zaiceva;
2. Latvijas Lauksaimniecības universitātes profesors, Dr.sc.ing. Egils Stalidzāns;
3. Vytautas Magnus University, Lietuva, profesors, Dr.habil. Vladislav Fomin.

Promocijas darba aizstāvēšana notiks LLU Informācijas tehnoloģiju nozares promocijas padomes atklātajā sēdē 2014. gada 22. decembrī Jelgavā, Lielā ielā 2, Informācijas tehnoloģiju fakultātes 218. auditorijā plkst. 11.00.

Ar promocijas darbu var iepazīties LLU Fundamentālajā bibliotēkā, Lielā ielā 2, Jelgavā un http://llufb.llu.lv/promoc_darbi.html

Atsauksmes sūtīt promocijas padomes sekretārei – Lielā ielā 2, Jelgavā, LV-3001; tālrunis: 63005621; e-pasts: tatjana.tabunova@llu.lv. Atsauksmes vēlams sūtīt skenētā veidā ar parakstu.

Padomes sekretāre: LLU lektore, Mg.paed. Tatjana Tabunova.

ISBN 978-9984-48-171-5 (online)

SATURS

Promocijas darba aprobācija.....	4
Ievads.....	6
Tēmas aktualitāte	6
Promocijas darba mērķis un uzdevumi	7
Pētījuma metodes	8
Zinātniskais jauninājums.....	8
Pētījuma tēzes	8
Praktiskā vērtība.....	9
Promocijas darba struktūra un apjoms	9
1. Kvalitāte programmatūras izstrādē.....	9
2. Programmatūras produkta kvalitātes modeļi.....	11
3. Kvalitāte augstākajā izglītībā	13
4. Vienotais kvalitātes modelis	18
5. Vienotā kvalitātes modeļa izmantošana	22
Secinājumi	30
Literatūras saraksts	64

PROMOCIJAS DARBA APROBĀCIJA

- Promocijas darbā veikto pētījumu rezultāti ir atspoguļoti šādās publikācijās
1. Cevere, R., **Sproģe, S.** (2013) Introduction of software product and process quality aspects in the study courses of information technologies. In: *Proceedings of the 6th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), p. 267–275.
 2. **Sproģe, S.**, Cevere, R. (2012) Position and role of feedback in the software quality life cycle. *Quality Issues and Insights in the 21st Century*, vol. 1, p. 29–40. (indeksēts EBSCO datubāzē).
 3. Cevere, R., **Sproģe, S.** (2012) Application of single quality model for self-assessment of study programme. In: *Proceedings of the 3rd International Workshop Intelligent Educational Systems and Technology – enhanced Learning*. Riga: RTU (Riga Technical university), p. 54–64.
 4. **Sproģe, S.**, Cevere, R. (2012) Assessment of study programme quality at higher education institution. In: *Proceedings of the 11th International Scientific Conference Engineering for Rural Development*. Jelgava: LLU (Latvia University of Agriculture), p. 663–668. (indeksēts Scopus datubāzē).
 5. **Sproģe, S.**, Cevere, R. (2012) Quality Models in Software Product Development Life Cycle. In: *Proceedings of the 5th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), p. 69–76.
 6. **Sproģe, S.** (2011) Evaluation of study programme external quality. In: *Proceedings of the Annual 17th International Scientific Conference Research for Rural Development*. Jelgava: LLU (Latvia University of Agriculture), vol 1, p. 179–185. (indeksēts Scopus, EBSCO Academic Search Complete datubāzēs).
 7. Cevere, R., **Sproģe S.** (2011) Quality life cycle of the Study Programme and Methodology for Evaluation. In: *Proceedings of the 3rd International Conference: Institutional Strategic Quality Management – ISQM2011, Sibiu, Romania*, p. 65–72, Available at: http://proiecte.aracis.ro/fileadmin/Academis/A0/ISQM_2011/Proceedings_ISQM_2011.pdf (indeksēts EBSCO Education Research Complete datubāzē)
 8. **Sproģe, S.**, Cevere, R. (2011) Software development quality models in engineering education. In: *Proceedings of the 10th International Scientific Conference Engineering for Rural Development*. Jelgava: LLU (Latvia University of Agriculture), p. 536–541. (indeksēts Scopus, EBSCO Academic Search Complete datubāzēs, citēts Scopus).
 9. Čevere, R., **Sproģe, S.** (2011) Methodology for Evaluation of Internal Quality of the Study Programme. EQANIE Conference Vienna 2011. [online] [10.06.2014.]. Available at: <http://www.eqanie.eu/pages/events/past-events/conference-vienna-2011/proceedings.php>

10. Čevere, R., **Sproģe, S.** (2010) Application of Software Quality Models in Evaluation of Study Quality. *Problems of Education in the 21st Century*, vol. 21, p. 37-48. (indeksēts EBSCO Education Research Complete, EBSCO Central & Eastern European Academic Source datubāzēs).
11. **Sproģe, S.**, Čevere, R. (2010) Quality Model of the Curricula of Information Technology Studies. In: *Proceedings of the 4th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), p. 148–157. (indeksēts Web of Science datubāzē, citēts Web of Science).
12. **Sproģe, S.**, Čevere, R., Arhipova I. (2008) Development of the faculty quality model and definition of control processes. *Scientific proceedings of Riga Technical University. Computer science & Information technology and management science*. Riga: RTU, p. 56–62.
13. Čevere, R., **Sproģe, S.**, Arhipova, I. (2008) How to use an experience of the development of ISO 9001 certified quality system in study process improvement. In: *Proceedings of the 3th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), p. 126–136. (indeksēts EBSCO Central & Eastern European Academic Source datubāzē).

Pētījumos iegūtie rezultāti ir prezentēti šādās konferencēs

1. „Introduction of software product and process quality aspects in the study courses of information technologies”. 6th International Scientific Conference “*Applied Information and Communication Technologies AICT’2012*”, April 25–26, 2013, Jelgava, Latvia.
2. „Application of Single Quality Model for Self-assessment of Study Programmes”. *Riga Technical university 3th International workshop on Intelligent Educational Systems and Technology – enhanced Learning INTEL_EDU 2012*, October 10, 2012, Riga, Latvia.
3. „Software development quality models in engineering education”. *11th International Scientific Conference Engineering for Rural Development*, May 24–25, 2012, Jelgava, Latvia.
4. „Quality Models in Software Product Development Life Cycle”. 5th International Scientific Conference “*Applied Information and Communication Technologies AICT’2012*”, April 26–27, 2012, Jelgava, Latvia.
5. „Studiju programmas kvalitātes modelis”. *Apvienotais pasaules latviešu zinātnieku 3. kongress*, 24. –26.10.2011, Rīga, Latvija.
6. „Quality Models in Software Product Development Life Cycle”. *Riga Technical university 52th International scientific conference*, October 12–15, 2011, Riga, Latvia.
7. „Quality life cycle of the Study Programme and Methodology for Evaluation”. *3rd Institutional Strategic Quality Management: From*

- Minimum Requirements to Overall Quality Improvement – ISQM2011*, July 14–16, 2011, Sibiu, Romania.
8. „Software development quality models in engineering education”. *10th International Scientific Conference Engineering for Rural Development*, May 26–27, 2011, Jelgava, Latvia.
 9. „Evaluation of study programme external Quality”. *17th International Scientific Conference Research for Rural Development*, May 18–20, 2011, Jelgava, Latvia.
 10. „Methodology for Evaluation of Internal Quality of the Study Programme”. *EQANIE-Conference: Learning Outcomes and Quality Management in Informatics Education*, February 17–19, 2011, Vienna, Austria.
 11. „Quality Model of the Curricula of Information Technology Studies”. 4th International Scientific Conference “*Applied Information and Communication Technologies AICT’2010*”, April 22–23, 2010, Jelgava, Latvia.
 12. „Development of the faculty quality model and definition of control processes”. *Riga Technical university 50th International scientific conference*, October 12–14, 2009, Riga, Latvia.

IEVADS

Tēmas aktualitāte

Kvalitātes jēdziens ir viens no tiem, kas sabiedrībā tiek ļoti plaši lietots praktiski visās nozarēs. Zināmā mērā var teikt, ka visās dzīves nozarēs ir populāri runāt par kvalitāti. No vienas puses tas ir labi, jo veicina vispārēju uzmanību, atvieglo iespēju veikt noteiktus darbus, piesaistīt finansējumu, kas patiešām dod ieguldījumu attiecīgās nozares attīstībā (Hoyer R. & Hoyer B., 2001). Kvalitāte tiek uzskatīta par vienu no galvenajiem faktoriem nozares konkurētspējas stratēģijā (Li & Meissner, 2008). No otras puses, tas var būt bīstami, jo devalvē pašu kvalitātes jēdzienu un var likties, ka, daudz runājot par kvalitāti, ar to arī tiek panākta tās uzlabošana. Tā ir viena no kļūdām, kuru pieļauj kvalitātes nodrošināšanas sfērā. Cits aspekts, kurš ir saistīts ar kvalitāti, ir tas, ka tā patiešām ir nepieciešamība. Tas ir viens no ceļiem, kā paaugstināt darba efektivitāti un darba rezultātu kvalitāti.

Programinženierijas sfērā kvalitāte vienmēr ir bijusi un būs svarīgs aspekts. Lai gan laika gaitā ir izstrādāti daudzi un dažādi kvalitātes modeļi programmatūras produkta kvalitātes prasību izvirzīšanai un novērtēšanai, kā arī programinženierijā izstrādātas metodes kvalitātes nodrošināšanas īstenošanai, tomēr arvien tiek izstrādāti arī nekvalitatīvi produkti. Daudzi programmatūras izstrādes projekti cieš neveiksmi tāpēc, ka kvalitātes prasības netiek precīzi

definētas. Programmatūras produktam kvalitāti nevar pievienot pēc tā izstrādes, tā ir jāveido jau produkta izstrādes laikā (Bawane & Srikrishna, 2010). Katrā dzīves cikla etapā un procesā veidojas daļa no programmatūras gala produkta kvalitātes, un tās veidošanu ietekmē dažādi faktori.

Nozīmīga loma programmatūras kvalitātes veidošanā ir personālam, kurš piedalās programmatūras izstrādē. Pamati viņu profesionālajai sagatavotībai, zināšanām un kompetencei par programmatūras kvalitāti tiek veidoti jau studiju laikā, apgūstot ar informācijas tehnoloģijām (IT) saistītas studiju programmas. Līdz ar to arī studiju process faktiski ir iekļaujams programmatūras produkta dzīves ciklā. Izglītības jomā studiju programmām ir savi kvalitātes kritēriji, kuri vairāk attiecas uz studiju programmas ārējo kvalitāti, t.i., to kvalitāti, kura izpaužas studiju procesa organizēšanā un studiju norisē. Zinātnisko publikāciju klāstā ir pieejami virkne pētījumu par to kā nodrošināt, mērit un vērtēt kvalitāti augstākās izglītības sfērā (Tam, 2001; Tan & Kek, 2004; Clewes, 2003; Van Der Wende & Westerheijden, 2001). Taču, iepazīstoties ar publicētajiem pētījumiem par programmatūras kvalitātes nodrošināšanu, autorei neizdevās atrast pētījumus, kuros visos dzīves cikla etapos tiktu izmantots vienots skatījums uz kvalitāti. Līdzīgi publikācijās par izglītības kvalitātes vērtēšanu, autorei neizdevās atrast pētījumus, kuros studiju programmas kvalitātes novērtēšanai tiktu izmantoti kvalitātes modeļi, tai skaitā definētas kvalitātes raksturpazīmes. Autore darbā piedāvā veidot vienotu skatījumu uz kvalitāti visa programmatūras dzīves cikla laikā, ietverot tajā arī personāla sagatavošanas un studiju kvalitāti. Ir izvirzīts priekšlikums uz studiju programmu skatīties līdzīgi kā uz programmatūras produktu un to kvalitātes novērtēšanai izmantot vienotu kvalitātes modeli, kas izveidots uz programmatūras produkta hierarhiskā kvalitātes modeļa bāzes. Darbā tiek piedāvāta arī koncepcija, kā informācijas tehnoloģiju pamatstudiju programmas saturā iekļaut programmatūras produkta kvalitāti, lai attīstītu studentu spējas apgūt katra programmatūras izstrādes soļa ietekmi uz gala rezultātu kvalitāti un tādējādi nākotnē radītu priekšnoteikumus kvalitatīvu programmatūras produktu izstrādei.

Promocijas darba mērķis un uzdevumi

Promocijas **darba mērķis** ir piedāvāt risinājumu programmatūras produktu kvalitātes uzlabošanai, pilnveidojot informācijas tehnoloģiju studiju programmu kvalitātes novērtēšanu un programmatūras produkta kvalitātes apguvi.

Definētā mērķa sasniegšanai tika ir izvirzīti šādi **darba uzdevumi**

1. Izpētīt kvalitātes lomu un nozīmi programmatūras produktu dzīves ciklā.
2. Veikt programmatūras produkta kvalitātes modeļu analīzi.
3. Analizēt esošās metodes un paņēmienus studiju kvalitātes nodrošināšanai.
4. Analizēt programmatūras produkta kvalitātes modeļa izmantojamību izglītības sfērā.

5. Definēt paplašinātu programmatūras produkta kvalitātes modeļa dzīves ciklu, ietverot tajā studiju procesu.
6. Izstrādāt vienotu kvalitātes modeli programmatūras izstrādes un studiju kvalitātes novērtēšanai.
7. Veikt vienotā kvalitātes modeļa aprobāciju studiju programmas kvalitātes novērtēšanai.
8. Izstrādāt koncepciju programmatūras kvalitātes modeļa iekļaušanai informācijas tehnoloģiju studiju programmā.

Pētījuma metodes

Literatūras apskats, analīze un sintēze, klasifikācija, profesionālapskates, aptaujas, statistiskā datu apstrāde un analīze, programmēšana.

Zinātniskais jauninājums

1. Izstrādāts vienots kvalitātes modelis, kuru visa dzīves cikla ietvaros var izmantot programmatūras produkta, tā izstrādes un uzturēšanas procesu, kā arī studiju programmas izstrādes un īstenošanas procesa kvalitātes novērtēšanai.
2. Izveidots kvalitātes modelis studiju programmas kvalitātes novērtēšanai:
 - a. vienotais kvalitātes modelis pielāgots studiju programmas raksturošanai;
 - b. izstrādāta metodika modeļa un metriku izmantošanai studiju programmas iekšējās un ārējās kvalitātes novērtēšanai.
3. Izstrādāta uz vienoto kvalitātes modeli balstīta metodika studiju kvalitātes novērtēšanas rezultātu apstrādei un interpretēšanai.
4. Izstrādāta koncepcija programmatūras produkta kvalitātes apguves pilnveidošanai topošo IT speciālistu studiju laikā.

Pētījuma tēzes

1. Programminženierijas un augstākās izglītības jomā kvalitāte nav viennozīmīgs jēdziens. Savstarpēji saistītu procesu un objektu vērtējumā pietrūkst vienota skatījuma uz kvalitāti.
2. Skatoties uz studiju programmu kā uz programmatūras produktu, to kvalitāšu novērtēšanai var izmantot vienotu studiju programmu un programmatūras produktu kvalitātes modeli.
3. Vienotas struktūras kvalitātes modelis dod vienotu skatījumu uz kvalitāti visā programmatūras dzīves ciklā un uz visiem objektiem (produktiem, starpproduktiem un procesiem).
4. Programmatūras produkta kvalitātes modeļa un tajā ietvertās kvalitātes vērtēšanas pieejas ieviešana informāciju tehnoloģiju studiju procesā rada programmatūras produktu kvalitātes uzlabošanas priekšnoteikumus.

Praktiskā vērtība

1. Vienotais kvalitātes modelis var tikt izmantots gan programmatūras produkta izstrādē, gan studiju procesā.
2. Izstrādātais vienotais kvalitātes modelis var tikt izmantots augstākās izglītības iestādē gan studiju programmu satura kvalitātes novērtēšanai, gan programmas realizēšanas kvalitātes novērtēšanai.
3. Izstrādātais lietojums studiju programmas kvalitātes novērtējuma datu apkopošanai, analīzei un prezentēšanai var tikt izmantots studiju programmas kvalitātes pilnveides lēmumu pieņemšanā.

Promocijas darba struktūra un apjoms

Promocijas darbs ir uzrakstīts latviešu valodā, satur anotāciju, ievadu, 5 nodaļas, secinājumus, literatūras sarakstu un 6 pielikumus, tajā skaitā 26 tabulas, 45 attēlus, 4 formulas, kopā 125 lapašpuses. Darbā izmantoti 172 literatūras avoti.

1. KVALITĀTE PROGRAMMATŪRAS IZSTRĀDĒ

Kvalitātes raksturīga pazīme ir tā, ka kvalitāte nav viennozīmīgs jēdziens (Reeves & Bednar, 1994). Ir sastopamas daudzas un dažādas kvalitātes definīcijas, dažādi tās izpratnes un novērtēšanas veidi (Garvin, 1987). Programmatūras izstrādes jomā biežāk sastopamie virzieni, kuros tiek aplūkota kvalitāte, ir:

- atbilstība iepriekš izvirzītajām prasībām (Crosby, 1979);
- piemērotība lietošanai (Juran, 1988);
- pakāpe, līdz kurai tiek apmierinātas lietotāja vajadzības vai cerības (Deming, 1988; Ishikawa, 1985).

Crosbijs atzīmē, ja kvalitātei iepriekš nav izvirzītas prasības, tad gatavo produktu ir grūti novērtēt. Līdz ar to jēdzieni „laba kvalitāte”, „slikta kvalitāte” paliek tikai jēdzienu līmenī, jo katram cilvēkam attiecībā uz tiem ir sava izpratne un prasības. Kvalitātes prasībām ir jābūt definētām tā, lai tās var izmērīt un novērtēt.

Balstoties uz Demingu (Deming, 1986), kvalitāte ir jādefinē kā klienta apmierinātības jēdziens un nevis kā atbilstība izvirzītajām prasībām. Demings uzskata, ka, definējot kvalitāti, ir grūti klienta nākotnes vēlmes saistīt ar nomērāmām produkta raksturpazīmēm. Līdz ar to, lai produktu varētu izstrādāt, lietotājam ir jābūt apmierinātam ar cenu, kas viņam būs jāmaksā. Tas nav viegli, jo līdzko ir sasniegta klienta apmierinātība, tā vēlmes attiecībā uz produktu var mainīties, un līdz ar to mainās arī klienta apmierinātība. Produktam izvirzīto prasību sasniegšana ir neatņemams izstrādes uzņēmuma pienākums. Katram uzņēmuma darbiniekam ir jābūt atbildīgam par sava darba kvalitāti.

Juran viedoklis attiecībā uz kvalitāti ir tāds, ka vārdu „kvalitāte” nevar lietot attiecībā uz lietotāju vajadzību un cerību apmierinātību, jo to ir ļoti grūti sasniegt (Juran, 1988). Tā vietā viņš piedāvā kvalitāti definēt kā „gatavs lietošanai”, kas norāda saistību ar izvirzītajām prasībām un produkta raksturpazīmēm. Juran piedāvāto definīciju principā var interpretēt tāpat kā Crosby piedāvāto – atbilstība iepriekš izvirzītajām prasībām.

Kopumā visi darbā aplūkoti skatījumi uz kvalitātes jēdzienu programminženierijas jomā atspoguļo četrus galvenos principus, kas ir saistīti ar kvalitāti:

1. nepieciešamība nodrošināt kvalitatīvus izstrādes procesus;
2. nepieciešamība definēt kvalitātes jēdzienu katrā konkrētā gadījumā;
3. nepieciešamība visiem dalībniekiem ievērot vienotu skatupunktu uz kvalitāti katrā konkrētā gadījumā;
4. nepieciešamība ievērot kvalitātes raksturpazīmju savstarpējo saistību un nepretrunīgumu.

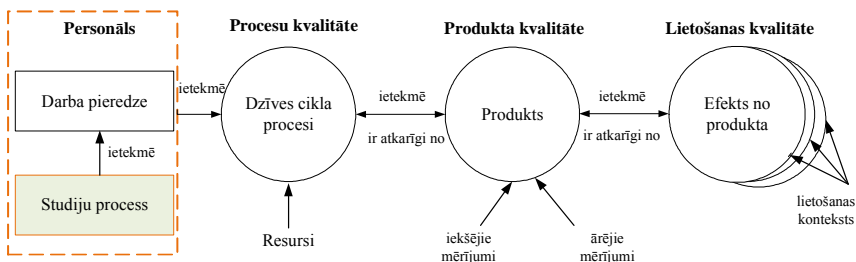
Programmatūras produktu izstrādē par kvalitāti un tās nodrošināšanu ir domāts jau no pašiem programmēšanas pirmsākumiem. Dažādu pieeju un metodiku izveides un apspriešanas gaitā ir izstrādāts zināmā mērā vienots viedoklis, kurš ir dokumentēts programminženierijas jomas starptautiskajos standartos. Šo standartu nolūks ir palīdzēt organizēt programmatūras produkta izstrādi, lai nodrošinātu un pilnveidotu tā kvalitāti.

Darba ietvaros tika pētītas un izmatotas metodikas un pieejas, kas saistītas ar programmatūras kvalitātes nodrošināšanu un kuras iekļautas atbilstošos programminženierijas nozares standartos. Galvenie aplūkoti jautājumi ir programmatūras dzīves cikls un kvalitātes vieta tajā, programmatūras produkta kvalitāte un tās novērtēšana, kā arī mērīšanas process.

Programmatūras produkta kvalitātes prasību definēšana, izmantojot programmatūras kvalitātes modeļus, un šīs kvalitātes novērtēšanai izstrādātās metodes ir apkopotas apjomīgās standartu sērijās:

- ISO/IEC 9126 Programminženierija – Programmatūras produkta kvalitāte;
- ISO/IEC 14598 Programminženierija – Produkta novērtēšana;
- ISO/IEC 25000 Programminženierija – Programmatūras produkta kvalitātes prasības un novērtēšana.

ISO/IEC 9126 standartā ir dots programmatūras produkta kvalitātes dzīves cikls (sk. 1. att.), kurā darba autore piedāvā iekļaut arī informācijas tehnoloģiju studijas, jo tas ir laiks, kad tiek sagatavoti jaunie IT speciālisti. Šo studiju laiku var uzskatīt par vienu no preventīvajiem pasākumiem programmatūras produktu kvalitātes izmaksu samazināšanai.



1. att. **Paplašinātais kvalitātes dzīves cikls**

Zinātnieku veiktie pētījumi parāda – jo vēlāk programmatūras dzīves ciklā kvalitātes problēmas ir atklātas, jo straujāk pieaug kvalitātes izmaksas. Jo ātrāk uzsāk programmatūras produkta kvalitātes kontroles un nodrošināšanas pasākumus, jo mazākas izmaksas būs nepieciešamas (The Case for Application Security...).

Personāla loma programmatūras produkta izstrādē ir atzīmēta arī standartu līmenī. Standartā ISO/IEC 12207-2008 (*Programmatūras dzīves cikla procesi*) ir izdalīts cilvēku resursu vadības process, kura mērķis ir nodrošināt organizāciju ar nepieciešamajiem cilvēkresursiem un lai viņu kompetence būtu atbilstoša biznesa vajadzībām. Kā galvenie uzdevumi šim procesam ir minēti – personāla profesionālo prasmju identificēšana, prasmju attīstīšanas plāni, prasmju iegūšanas nodrošināšana un zināšanu vadība. Iekļaujot programmatūras produkta kvalitātes dzīves ciklā studiju procesu, var panākt un nodrošināt, ka šajā laikā jaunajiem speciālistiem tiek veidota individuālā pieredze par produkta kvalitāti. Studiju procesā, apgūstot kvalitātes raksturpazīmju nozīmi un lomu dažādos programmatūras produkta izstrādes procesos, tiks veikti preventīvie pasākumi kvalitatīvu produktu izstrādei nākotnē.

Izveidojot un izmantojot vienotu kvalitātes modeli visā programmatūras produkta kvalitātes dzīves ciklā, tiktu panākts vienots skatījums uz visiem tā posmiem, starprezultātiem un rezultātiem.

2. PROGRAMMATŪRAS PRODUKTA KVALITĀTES MODEĻI

Kvalitātes modeli var raksturot ar šādu definīciju: ir noteiktas raksturpazīmes un to savstarpējā saistība, kuras veido bāzi, lai specificētu kvalitātes prasības un novērtētu kvalitāti (ISO/IEC 9126-1, 2001).

Vēsturiski ir veidojušies un cits citu papildinājuši daudzi programmatūras kvalitātes modeļi, ar kuru palīdzību ir iespējams noteikt un novērtēt produkta kvalitāti. Darbā ir analizēti zinātniskajās publikācijās biežāk sastopamie modeļi: McCall modelis, Boehm modelis, FURPS modelis, ISO 9126 modelis,

Dromey modelis. Darbā šie kvalitātes modeļi nav salīdzināti savstarpēji, bet gan katrs ar ISO 9126 modeli, kas ir starptautiski atzīts un iestrādāts standartā.

Programmatūras produkta kvalitātes modeļu attīstība laikā dažādās zinātniskās publikācijās ir atrodamas savstarpējas atsauces, kā arī ir veiktas modeļu salīdzināšanas raksturpazīmju līmenī (Al-Qutaish, 2010; Dubey et al., 2012; *Khayami* et al., 2009; Rawashdeh & Matakah, 2006; Ortega et al., 2003). Šī darba autore modeļu salīdzināšanu ir veikusi arī kvalitātes apakšpazīmju līmenī un lielu uzmanību ir veltījusi terminiem, kas izmantoti katrā modelī raksturpazīmju un apakšpazīmju apzīmēšanai. Tā kā kvalitātes modeļa izmantošanā būtiski ir ņemt vērā konkrētās situācijas kontekstu un kvalitātes vērtēšanā ieinteresēto pušu viedokli, tad būtisks ir faktors, kādi termini tiek izmantoti kvalitātes raksturpazīmju apzīmēšanai.

Darbā veiktā modeļu analīze raksturpazīmju līmenī parādīja, ka pamatā visos modeļos par kvalitātes raksturpazīmēm tiek izmantotas tās pašas kvalitātes raksturpazīmes, kuras ir apkopotas ISO 9126 modelī. Ja arī kādā modelī ir dots atšķirīgs nosaukums, tad pēc nozīmes tas atbilst ISO 9126 modeļa kādai raksturpazīmei vai apakšpazīmei.

Ja apskatītajos modeļos kvalitātes raksturpazīmes pēc nosaukumiem sakrīt par 77%, tad piedāvātās apakšpazīmes būtiski atšķiras gan pēc skaita, gan nosaukumiem. Atšķiras arī raksturpazīmju un apakšpazīmju savstarpējā saistība.

Analizējot visos modeļos minētās kvalitātes apakšpazīmes katrai raksturpazīmei, autore ir izdalījusi tās, kuras ir sinonīmi ISO 9126 minētajām, un tās, kuras būtu ņemamas vērā papildus, definējot kvalitātes prasības un pēc tam novērtējot attiecīgo kvalitātes raksturpazīmi. Piecos apskatītajos kvalitātes modeļos kopumā ir piedāvātas 78 unikālas apakšpazīmes kvalitātes raksturpazīmju novērtēšanai. Lielākais apakšpazīmju skaits ir minēts tādām kvalitātes raksturpazīmēm kā *lietojamība* un *funkcionalitāte*. Vienai un tai pašai kvalitātes apakšpazīmei dažādos kvalitātes modeļos tiek piedāvāti dažādi nosaukumi, taču pēc nozīmes tie ir vienādi. Autore vēlas atzīmēt, ka atsevišķos gadījumos piemērotāks nosaukums programmatūras produkta izstrādātājiem un pasūtītājiem palīdzētu gan izvairīt, gan novērtēt attiecīgo kvalitātes rādītāju, piemēram ISO 9126 piedāvā kvalitātes raksturpazīmei *uzturamība* apakšpazīmi *analizējamība*, taču tādi nosaukumi kā *nepretrunīgums*, *kodolīgums*, *lasāmība* un *strukturētība* labāk palīdzētu izprast šo kvalitātes raksturpazīmi.

Jaunā ISO 25010 kvalitātes modeļa izstrāde parāda tendenci, ka esošie kvalitātes modeļi nav pilnīgi un notiek kvalitātes modeļu attīstība. Autore vēlas atzīmēt, ka jaunajā ISO kvalitātes modelī nav piedāvātas pēc būtības jaunas kvalitātes raksturpazīmes, bet gan ISO 9126 modelī esošās raksturpazīmes un apakšpazīmes ir pārstrukturizētas. Tās kvalitātes apakšpazīmes, kas ir piedāvātas kā pilnīgi jaunas, ir jau minētas citos autores analizētajos modeļos, piemēram, apakšpazīme *modularitāte* ir piedāvāta McCall modelī, *integritāte* Boehem modelī, *atkārtota lietojamība* McCall un Boehem modelī, bet *palīdzības iespēja* FURPS modelī.

Līdz ar to autore izvirza priekšlikumu izveidot jaunu kvalitātes modeli, tajā izdalot divus jaunus apakšpazīmju veidus – papildu un neobligātās apakšpazīmes, lai vienu un to pašu raksturpazīmi padarītu konkrētai auditorijai saprotamāku. Modeļa izveide ir apskatīta darba 4. nodaļā. Jauna modeļa izveide ir pamatojama arī ar to, ka programmatūras kvalitātes modeļos izmantoto raksturpazīmju un apakšpazīmju jēdzieni ir pietiekami vispārīgi, un autore tos piedāvā izmantot ne tikai programmatūras kvalitātes novērtēšanā, bet arī studiju programmas novērtēšanā, tādējādi parādot, ka programmatūras produktu kvalitātes modeļus var transformēt uz izglītības sfēru.

3. KVALITĀTE AUGSTĀKAJĀ IZGLĪTĪBĀ

Saistībā ar augstākās izglītības un tās rezultātu kvalitāti eksistē virkne dažādu jautājumu, uz kuriem nav viennozīmīgas atbildes, piemēram, kas ir studiju programmas kvalitāte un kas ir domāts ar jēdzienu „laba kvalitāte”. Vai kvalitatīva ir studiju programma, kura ir akreditēta, vai tā, kura atbilst ieinteresēto pušu vēlmēm un vajadzībām, vai arī tā, ar kuras absolventiem ir apmierināti potenciālie darba devēji? Taču var arī teikt, ka studiju programma ir kvalitatīva, ja studiju process ir sakārtots, vadīts un kontrolēts, ir attiecīgā materiālā bāze un ir sasniegti definētie studiju rezultāti gan katrā studiju kursā, gan visā programmā kopumā.

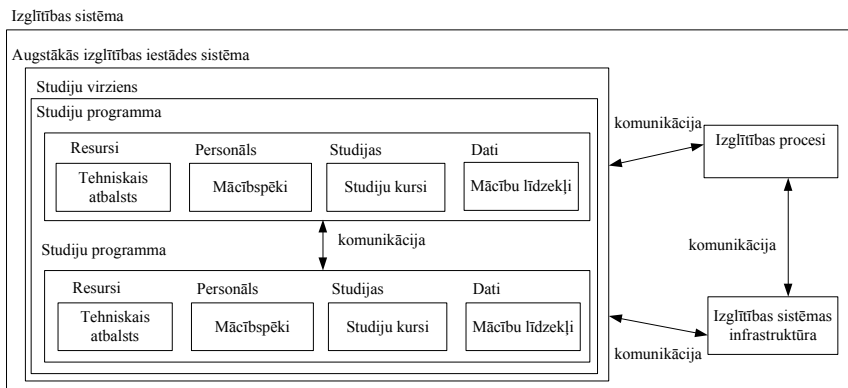
1993. gadā Harvey, Green un Burrows izstrādāja metodoloģiju, kā novērtēt kvalitāti augstākajā izglītībā. Metodoloģijas pamatā ir šādi kvalitātes kritēriji:

- fizisko resursu pietiekamība;
- cilvēku resursu pietiekamība;
- mērķu un uzdevumu skaidrība visiem dalībniekiem;
- studiju kursu un to satura atbilstība studiju programmas mērķiem un uzdevumiem;
- studentu līdzdalība visos līmeņos;
- programmas satura atbilstība piešķiramajam grādam vai kvalifikācijai;
- objektivitāte vērtēšanā;
- konsekvence starp vērtēšanu un studiju kursa uzdevumiem;
- atgriezeniskās saites iegūšana pēc vērtēšanas;
- zināšanu un prasmju iegūšanas pēctecība.

Piedāvātie kritēriji būtiski nav mainījušies arī šodienas piedāvātajās studiju kvalitātes vērtēšana metodoloģijās. Iekļaujoties Eiropas augstākās izglītības telpā, Boloņas deklarācija rekomendē virzīties tieši uz studiju rezultātu un kompetenču mērīšanu, lai novērtētu studiju programmu kvalitāti (Boloņas process..., 2008).

Autore savā darbā neapskata augstākās izglītības kvalitāti kopumā, bet gan studiju programmas kvalitāti, jo tas ir produkts, kuru augstākās izglītības iestādes piedāvā saviem klientiem – topošajiem attiecīgās nozares speciālistiem. Uz studiju programmu nevar skatīties atrauti no pašas augstākās

izglītības iestādes un tur noteiktās izglītības sistēmas, kā arī no visas augstākās izglītības sistēmas valstī. Uz studiju programmu var paskatīties no sistēmiskas pieejas. Pēc analogijas ar programmatūras produktu, arī studiju programma ir elements augstākās izglītības iestādes sistēmā, kura savukārt ir elements augstākās izglītības sistēmā (sk. 2. att.).



2. att. Studiju programma izglītības sistēmas modeli

Studiju programmas kvalitāti ietekmē gan tās elementi – resursi, personāls, studiju process un mācību līdzekļi – gan arī esošā izglītības iestādes un visas valsts izglītības sistēma. Kvalitāti ietekmē, un tā ir arī atkarīga no izglītības sistēmas infrastruktūras un noteiktajiem izglītības procesiem. Gan pašai sistēmai, gan katram tās elementam var piekārtot noteiktu kvalitātes modeli un, izvirzot kvalitātes prasības, ir jādomā, kā tās ietekmē sistēmas elementu kvalitāti un cik daudz tās ir atkarīgas no visas sistēmas kvalitātes.

Kā paraugu izglītības sistēmas kvalitātes prasību klasificēšanai var izmantot programmatūras sistēmas kvalitātes prasību klasificēšanu (ISO/IEC 25030, 2007). Šādā skatījumā studiju programmas kvalitātes prasības ir daļa no studiju programmai piemītošo īpašību prasībām (sk. 3. att.).

Augstākās izglītības institūcijas sistēmas prasības	Studiju virziena prasības	Studiju programmas prasības	Programmai piemītošo īpašību prasības	Funkcionālās prasības		
				Studiju programmas kvalitātes prasības	Lietošanas kvalitātes prasības	
					Ārējās kvalitātes prasības	
					Iekšējās kvalitātes prasības	
			Programmai piešķirto īpašību prasības	Studiju maksas, studiju veida, studiju atbalsta u.c. prasības		
		Studiju programmas izstrādes prasības	Izstrādes procesa prasības			
	Izstrādes organizatoriskās prasības					
	Citas sistēmas prasības	Prasības izglītības sistēmas infrastruktūrai, literatūrai, datiem, saistītiem biznesa procesiem utt.				

3. att. Augstākās izglītības institūcijas sistēmas prasību klasificēšana

Kvalitātes prasības tiek iedalītas trīs grupās: iekšējās, ārējās un lietošanas prasības. Augstākās izglītības institūcijas sistēmas gadījumā ar lietošanas kvalitāti saprot jauno speciālistu iekļaušanos darba tirgū un atbilstību darba devēju vajadzībām, līdz ar to galvenais šo prasību informācijas avots ir darba devēji, kā arī absolventi, kuri strādā atbilstošajā nozarē. Tradicionāli, runājot par augstākās izglītības kvalitāti, tiek lietots arī ārējās kvalitātes novērtēšanas jēdziens, ar to saprotot ārējās institūcijas veiktu novērtēšanu. Šajā promocijas darbā uz studiju programmām tiek attiecināts arī ārējās kvalitātes jēdziens, kas pēc līdzības ar programmatūras produkta ārējo kvalitāti ir nosakāms studiju programmas realizēšanas laikā. Ārējās kvalitātes prasības studiju programmām pamatā var iegūt no pašreizējiem studentiem un absolventiem, kā arī likumdošanas regulām. Nozīmīgi ir arī visi lēmumi un vienošanās, kas saistītas ar Eiropas vienotās izglītība telpas izveidi. Iekšējās kvalitātes prasības attiecīgi veidojas no lietošanas un ārējās kvalitātes prasībām, un tās galvenokārt tiek izmantotas studiju kursu un programmu izstrādes un atjaunināšanas laikā, ko veic izglītības iestādes struktūrvienības.

Studiju programmas kvalitātes novērtēšanai autore piedāvā izmantot šādus variantus:

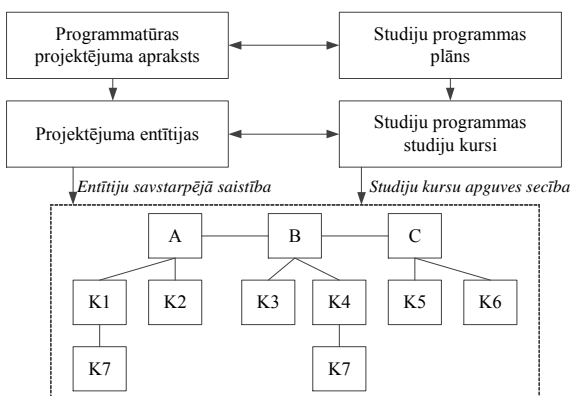
- ārējās kvalitātes novērtējums – studentu un darba devēju aptaujas;
- iekšējās kvalitātes novērtējums – profesionālapskates.

Autore piedāvā studiju programmas kvalitātes novērtēšanai izmantot kvalitātes modeli, kas ir līdzīgs programmatūras produkta kvalitātes modelim. Transformējot programminženierijas jomā izstrādātos kvalitātes modeļus uz izglītības sfēru, var izstrādāt jaunu kvalitātes modeli, kas ir vienots abiem šiem produktiem – programmatūrai un studiju programmai.

Programmatūras produkta un studiju programmas analogija

Programmatūras produkts un studiju programma ir abstrakti produkti, kuru kvalitāti nevar vizuāli novērtēt un šiem abiem produktiem piemērotākā kvalitātes definīcija ir atbilstība iepriekš izvirzītajām prasībām. Līdz ar to ļoti būtisks ir jautājums kādas šīs kvalitātes prasības ir definētas. Izstrādājot gan vienu, gan otru produktu, tiek sagatavoti virkne dokumentu, kuri pēc tam tiek izmantoti arī produkta kvalitātes novērtēšanā. Darbā ir apskatīti abu produktu pamatdokumenti (prasību specifikācija, projektējuma apraksts, programmatūras pirmkods, studiju programmas apraksts, plāns, studiju kursu apraksti) un to savstarpējā līdzība.

Studiju programmas un programmatūras produkta līdzības analīze balstās uz programmatūras projektējuma aprakstam dotajām rekomendācijām. Rekomendāciju shematiskajā attēlojumā ir redzama analogija (sk. 4. att.) ar studiju programmas plānu.



4. att. Programmatūras projektējuma un studiju programmas plāna analogija

Projektējuma apraksta zemākais detalizācijas līmenis ir projektējuma entīcija (attēlā izmantotie apzīmējumi A, B, C, K1, K2...). Projektējuma entīcija ir projektējuma elements (komponents), kas ir strukturāli un funkcionāli atšķirīgs no citiem elementiem, kam tiek piešķirts atsevišķs nosaukums un uz ko atsevišķi atsaucas programmatūras izpildes laikā (IEEE Std 1016). Lai gan entītijas ir dažādas pēc savas dabas, tām piemīt kopējas raksturiezīmes. Katrai projektējuma entīcijai jābūt nosaukumam, nolūkam un funkcijai. Entītiju kopējās raksturiezīmes tiek aprakstītas ar projektējuma entītijas atribūtiem. Projektējuma entītijas atribūts ir projektējuma entītijas raksturiezīme vai īpašība, kurai ir piešķirts nosaukums. Atribūts apraksta būtisku projektējuma informāciju, bet ne informāciju, kas attiecas uz projektēšanas procesu.

Studiju programmas gadījumā līdzīgs elements ir studiju kurss un kursā iekļautās tēmas. Visi vienas programmas studiju kursi (attēlā izmantotie apzīmējumi K1, K2 ...) ir izkārtoti plāna daļās (attēlā izmantotie apzīmējumi A, B un C) un norādīta tā apguves secība. Studiju kursa atribūti apraksta būtisku studiju programmas informāciju, bet ne to kā šī programma tiek realizēta. 1. tabulā ir dota projektējuma entītijas un studiju kursa atribūtu analogija.

1. tabula. **Projektējuma entītijas un studiju kursa atribūti**

Projektējuma entītijas atribūts (LVS 72, 1996)	Studiju kursa atribūts (LLU Studiju kursa nolikums...)
Identificējums – <i>entītijas nosaukums</i>	Studiju kursa nosaukums un kods LLU informācijas sistēmā
Tips – <i>entītijas veida apraksts</i>	Studiju kursa vieta studiju plānā – A, B vai C daļā – un pārbaudes veids
Nolūks – <i>apraksts, kādēļ eksistē entīcija</i>	Studiju kursa mērķis
Funkcija – <i>konstatējums, ko entīcija dara</i>	Studiju kursa satura izklāsts
Pakļautība – <i>visu to entītijū identificējums, kas veido doto entītijū</i>	Kursu sadalījums pa semestriem un studiju darba veidiem
Atkarības – <i>dotās entītijas un citu entītijū attiecību apraksts</i>	Kursa saistība ar citiem kursiem
Saskarne – <i>apraksts tam, kā citas entītijas mijiedarbojas ar doto entītijū</i>	Vēlamās priekšzināšanas studiju kursa apguvei, tā nepieciešamība citu studiju kursu apguvei
Resursi – <i>to projektējuma ārējo elementu apraksts, kurus lieto dotā entīcija</i>	Lekciju un praktisko darbu sadalījums, nepieciešamie tehniskie resursi kursa realizācijai
Apstrāde – <i>to likumu apraksts, kurus lieto entīcija, lai veiktu savas funkcijas</i>	Kursa prasības kredītpunktu ieguvei
Dati – <i>entītijas iekšējo datu elementu apraksts</i>	Nepieciešamās literatūras saraksts kursa satura apguvei. Elektroniskie mācību materiāli

Novērtējot programmatūras produkta iekšējās kvalitātes raksturpazīmes, programmatūras prasības specifikācija un projektējuma apraksts tiek ņemts kā ievades vienums mērījumu veikšanai. Mērījumu laikā tiek novērtēta katra projektējuma entīcija, lai spriestu par visas programmatūras produkta kvalitāti. Tā kā studiju programmas sīkākā vienība ir studiju kurss, tad, novērtējot katru kursu atsevišķi, var tikt iegūts kopējais studiju programmas iekšējās un ārējās kvalitātes novērtējums.

Autore piedāvā studiju procesā izmantot vienotu kvalitātes modeli gan izstrādājot, gan realizējot IT studiju programmu. Tas ir pamatojams ar to, ka IT speciālistiem studiju laikā, apgūstot metodes un paņēmienus, kā radīt programmu, līdztekus vajadzētu aplūkot arī katras šīs darbības ietekmi uz programmatūras produkta kvalitāti. Šādā gadījumā produkta kvalitātes atsevišķie aspekti detalizētāk ir jāapskata visos studijuursos, kuri ir saistīti ar programmatūras izstrādi un ieviešanu. Vienota kvalitātes modeļa izmantošana

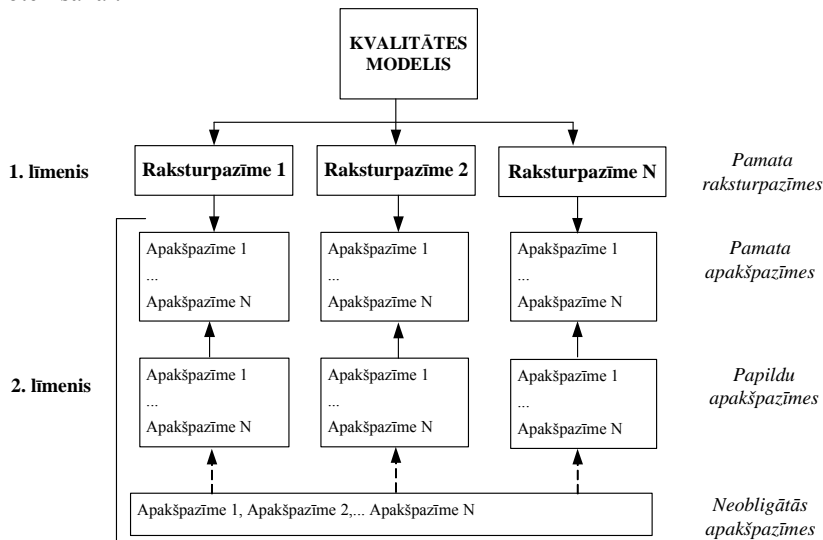
palīdzēs veidot kopēju izpratni gan par pašu kvalitātes jēdzienu, gan kvalitātes prasību izvirzīšanu un novērtēšanu.

4. VIENOTAIS KVALITĀTES MODELIS

Vienotā kvalitātes modeļa (VKM) pamatā ir ņemts programmatūras kvalitātes modelis ISO 9126 un papildināts ar jaunām kvalitātes apakšpazīmēm. Modelim ir divu līmeņu hierarhiska struktūra (sk. 5. att.):

1. līmenis – kvalitātes pamata raksturpazīmes;
2. līmenis – kvalitātes raksturpazīmju apakšpazīmes, kuras savukārt tiek iedalītas:
 - pamata apakšpazīmes;
 - papildu apakšpazīmes;
 - neobligātās apakšpazīmes.

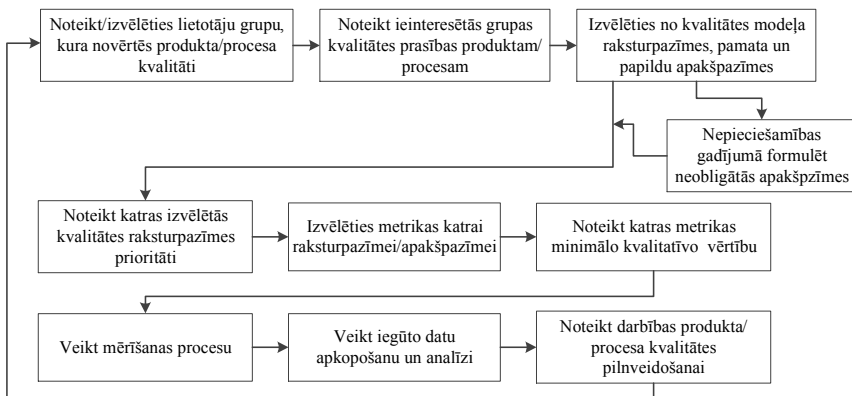
Modelī kvalitātes raksturpazīmju un apakšpazīmju saistība ir 1:N, izņemot neobligātās apakšpazīmes, tām ir saistība N:M. Tas ir skaidrojams ar to, ka neobligātās apakšpazīmes var tikt izvēlētas tādas, kuras var būt raksturojošas vairākām kvalitātes raksturpazīmēm, piemēram, apakšpazīme *lasāmība* var tikt izmantota gan lietojamības, gan uzturamības kvalitātes raksturpazīmes noteikšanai.



5. att. Vienotā kvalitātes modeļa hierarhiskā struktūra

Vienotais kvalitātes modelis katrā konkrētā gadījumā ir jāpielāgo lietošanas kontekstam un kvalitātes vērtēšanā ieinteresēto pušu prasībām. Visos

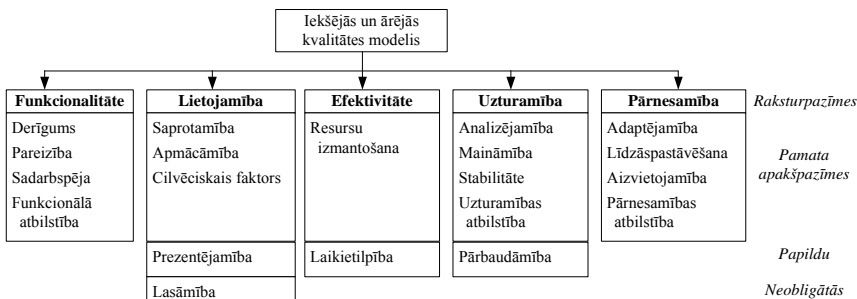
gadījumos pielāgošana jāveic atbilstoši darbību secībai, kas noteikta promocijas darbā izstrādātā vienotā modeļa lietošanas procedūrā (sk. 6. att.).



6. att. Vienotā kvalitātes modeļa izmantošanas procedūra

Vienotais kvalitātes modelis ir veidots ar domu, ka to var izmantot visā programmatūras produkta kvalitātes dzīves ciklā. Tā kā autore šajā ciklā iestrādāja studiju procesu, tad darba izstrādes laikā šis modelis tika aprobēts, novērtējot studiju programmas kvalitāti un parādīts, kā šo modeli ieviest IT studiju programmas saturā. Modeļa aprobācija notika Latvijas Lauksaimniecības universitātē.

Promocijas darbā tiek piedāvāts studiju programmas kvalitātes modelis (sk. 7.att.), kurš ir iegūts no vienotā kvalitātes modeļa, veicot tā pielāgošanu atbilstoši modeļa lietošanas procedūrai. Lai noteiktu modeļa lietošanas kontekstu, ir izmantota 3.2. nodaļā aprakstītā analogija ar programmatūras iekšējās un ārējās kvalitātes jēdzienu. Modelī ir iekļautas 5 kvalitātes raksturpazīmes un 21 apakšpazīme. Par pamata raksturpazīmēm ir izvēlētas: *funkcionalitāte, lietojamība, efektivitāte, uzturamība un pārnesamība*.



7. att. Studiju programmas kvalitātes modelis

Vienotā kvalitātes modeļa raksturpazīmju un apakšpazīmju skaidrojums attiecībā uz studiju programmu:

- **Funkcionalitāte** (Vai studiju programmā ir ietveras visas nepieciešamās zināšanas un prasmes):
 - *derīgums* – ir aktuāla, atbilst studiju virziena mērķiem un darba tirgus prasībām;
 - *pareizība* – ietvertās zināšanas un prasmes ir apgūstamas un atbilst piešķiramās kvalifikācijas/grāda prasībām;
 - *sadarbspēja* – programmā sasniedzamie rezultāti atbalsta piedalīšanos apmaiņas studijās un nodrošina nepieciešamās zināšanas tālākām studijām;
 - *funkcionālā atbilstība* – programma sagatavota atbilstoši normatīvajiem dokumentiem, ir akreditēta.
- **Lietojamība** (Vai programmu ir viegli mācīt un mācīties):
 - *saprotamība* – studiju programmas plāns ir sabalansēts un loģiski strukturēts, izklāstījuma stils atbilst priekšzināšanu līmenim;
 - *apmācāmība* – programmas teorētisko un praktisko zināšanu apguve ir līdzsvarota, mācību materiāli ir definēti un pieejami;
 - *prezentējamība* – mācībspēku oratoru prasmes ir pietiekamas;
 - *lasāmība* – programmas aprakstošie dokumenti ir sagatavoti labā lasāmā valodā, atbilstoši normatīvajiem dokumentiem;
 - *cilvēciskais faktors* – pakāpe, līdz kurai studiju programmas realizāciju var iespaidot konkrēts mācībspēks.
- **Efektivitāte** (Vai studiju programma ir efektīva):
 - *laikietilpība* – programmas apjoma atbilstība apgūšanai atvēlētajam laikam;
 - *resursu izmantošana* – programmas realizācijai ir nepieciešamie resursi.
- **Uzturamība** (Vai studiju programmu ir viegli uzturēt):
 - *analizējamība* – studiju programmas plāns, un tā saturs ir analizējams;
 - *maināmība* – studiju programmu ir iespējams modificēt, nepasliktinot tās apmācāmību;
 - *stabilitāte* – studiju programmas saturu nav nepieciešams atjaunināt/modificēt studiju gada laikā;
 - *pārbaudāmība* – studiju programmas mērķa sasniegšana ir pārbaudāma;
 - *uzturamības atbilstība* – programmai ir izstrādāti nepieciešamie normatīvie dokumenti, studiju programmas kvalitātes novērtējums ir pieejams.
- **Pārnesamība** (Vai studiju programmu ir viegli pielāgot citai auditorijai):
 - *līdzāspastāvēšana* – programma neprasa specifiskas priekšzināšanas;
 - *adaptējamība* – programmas pielāgošana citai valodai vai citai kultūras telpai neprasa izstrādāt to pilnīgi no jauna;

- *aizvietojamība* – pakāpe, līdz kādai programmas rezultātus var iegūt citā programmā;
- *pārnesamības atbilstība* – atbilstība normatīviem dokumentiem, kas regulē studentu apmaiņas programmas.

Koncepcija kvalitātes modeļa iekļaušanai studiju programmas saturā

Programmatūras izstrādes speciālistu sākotnējā sagatavošana notiek studiju laikā, apgūstot ar informācijas tehnoloģijām saistītu studiju programmu. Studiju programmās kā galvenais uzdevums tiek izvirzīts sniegt zināšanas par programmatūras produktu izstrādes rīkiem un paņēmieniem. Paralēli tam tiek aplūkoti izstrādes procesi, to organizēšana, vadība un kvalitātes nodrošināšana. Tādējādi katrā atsevišķā studiju kursā netiek aplūkota apgūstamās rīcības ietekme uz gala produkta kvalitāti. Darba autores pieredze gan informācijas tehnoloģiju sistēmu izstrādē, gan šīs nozares studiju programmu realizācijā, ir ļāvusi izvirzīt hipotēzi, ka būtu nepieciešams jau studiju laikā potenciālajiem informācijas tehnoloģiju sfēras darbiniekiem dot padziļinātas zināšanas par to:

- kā novērtēt gala produkta kvalitāti;
- kādu iespaidu uz produkta kvalitāti rada dažādi izstrādes gaitā izvēlētie risinājumi un pieņemtie lēmumi;
- kādas ir iespējas kvalitāti uzlabot.

Lai to realizētu, ir jāpilnveido informācijas tehnoloģiju studiju programmu saturs. Kā galvenais informācijas avots programmatūras produkta kvalitātes modeļa iekļaušanai studiju kursu satura pilnveidošanai, promocijas darba izstrādāšanas laikā, tika izvēlēta studiju kursu un programmatūras kvalitātes raksturpazīmju savstarpējās ietekmējamības matricas izstrāde, atgriezeniskā saite par programmatūras gala un starpproduktu novērtēšanu IT uzņēmumos un mācībspēku aptauja.

Autore koncepcijā par kvalitātes modeļa iekļaušanu studiju programmas saturā izdala 4 soļus

1. No bakalaura līmeņa informācijas tehnoloģiju studiju programmām izdalīt kursus, kuri ir saistīti ar programmatūras produktu izstrādi.
2. Izdalītos studiju kursus sadalīt četros programmatūras produkta izstrādes etapos – algoritmu izstrāde, datu bāzes izstrāde, kodēšana un testēšana.
3. Izmantojot autores izveidoto uzņēmumu speciālistu un mācībspēku viedokļa apkopojumu par programmatūras produkta kvalitātes raksturpazīmēm studiju procesā, izdalīt tās apakšpazīmes, par kurām attiecīgajos studijuursos būtu jā mācā.
4. Papildina esošos studiju kursu saturu ar attiecīgajām kvalitātes raksturpazīmēm, uzsvāru liekot uz to, ar kādām darbībām kvalitāti var gan palielināt, gan samazināt.

5. VIENOTĀ KVALITĀTES MODEĻA IZMANTOŠANA

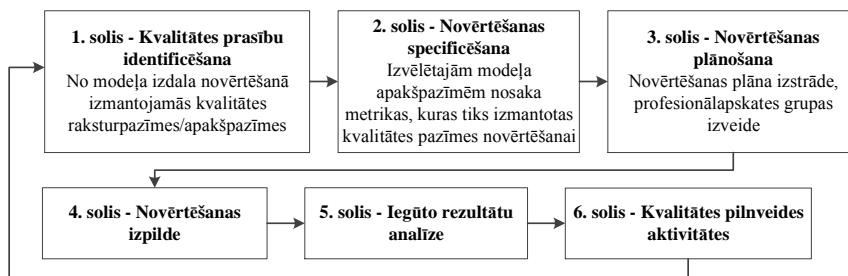
Darbā izstrādātais vienotais kvalitātes modelis tika aprobēts studiju programmu iekšējās un ārējas kvalitātes novērtēšanai.

Studiju programmas iekšējā kvalitāte

Studiju programmas iekšējā kvalitāte ir tajā ietverto studiju kursu saturs. Studiju programmas iekšējās kvalitātes vērtēšanu autore piedāvā organizēt profesionālapaskašu veidā. Apskates dalībnieku grupā būtu jāietver profilējošo katedru mācībspēki un studiju programmas absolventu pārstāvji (esošie maģistranti vai doktoranti). Apskašu darbam var piesaistīt arī nozares speciālistus. Tā kā viena studiju programma satur vairākus desmitus studiju kursu, tad grupu sastāvs var būt mainīgs.

Katra studiju kursa satura novērtēšana ir jāorganizē klātienē profesionālapaskašu veidā. Katra apskate būtu jābalsta uz šādiem dokumentiem – kursu programmas, kursu pieteikuma veidlapas (ja tādas tiek lietotas), studiju kursu esošās programmas (lai novērtētu kursa satura izmaiņas), visas studiju programmas plāns. Iekšējās kvalitātes novērtēšanā līdztekus ir jāizmanto arī augstākās izglītības iestādes bibliotēkas elektroniskais katalogs, lai varētu spriest par studiju kursā norādītās literatūras pieejamību.

Studiju programmas iekšējās kvalitātes novērtēšanai autore piedāvā studiju kursu satura kvalitātes novērtēšanas procedūru, kurā ir izdalīti 6 soļi (sk. 8. att.).



8. att. Studiju programmas iekšējās kvalitātes novērtēšanas procedūra

1. solis – Kvalitātes prasību identificēšana

No kvalitātes modeļa ir jāizdala tās kvalitātes raksturpazīmes un apakšpazīmes, kuras konkrētajā novērtēšanā tiks novērtētas. Atkarībā no tā vai studiju kursa satura novērtēšana notiek pirmo reizi vai atkārtoti, izmantotās kvalitātes raksturpazīmes var būt atšķirīgas. Pirmajā sākotnējā studiju kursa novērtēšanā būtu vēlams izmantot visas raksturpazīmes, jo ir jāiegūst pilns satura kvalitātes novērtējums. Organizējot atkārtotas studiju kursu profesionālapskates, var izdalīt atsevišķas apakšpazīmes, kuras vēlas novērtēt.

2. solis – Novērtēšanas specificēšana

Pēc kvalitātes raksturpazīmju/apakšpazīmju izvēles ir jāizvēlas metrikas, ar kurām tiks novērtēta katra apakšpazīme. Atkarībā no tā, kādas kvalitātes raksturpazīmes ir izvēlētas pirmajā solī, metriku skaits katrā novērtēšanā var atšķirties. Darbā ir dotas studiju kursa iekšējās kvalitātes raksturpazīmju novērtēšanai iespējamās metrikas, skaidrojums uz kādu jautājumu metrika atbild un vienums, kuru izmantot, lai novērtētu kvalitāti.

3. solis – Novērtēšanas plānošana

Pirms uzsākt studiju programmas iekšējās kvalitātes novērtēšanu ir jāveic tās plānošana:

- jāizvēlas kursi, kuri tiks novērtēti. Ja tā ir studiju programmas pirmreizējā novērtēšana, tad būtu jānovērtē visi studiju kursi. Atkārtotas novērtēšanas gadījumā var izvēlēties atsevišķus studiju kursus;
- jāizveido studiju kursu profesionālapaskašu kalendārais plāns;
- jāizveido profesionālapaskašu dalībnieku grupas;
- jāizveido studiju kursa iekšējās kvalitātes novērtēšanas kontroljautājumu saraksts. Atkarībā no tā vai kurss tiek novērtēts pirmo reizi vai atkārtoti, apskates dalībniekiem domāto kontroljautājumu saraksts var būt mainīgs. Darbā ir dots pilns studiju kursa kontroljautājumu saraksts;
- profesionālapskates dalībniekiem jāizsūta studiju kursa apskatei nepieciešamie dokumenti (apskatāmie vienumi) un kursa novērtējuma kontroljautājumu sarakstu;
- jāinformē apskates dalībnieki par klātienes tikšanos un to, kā tiks organizēta studiju kursa novērtēšana.

4. solis – Novērtēšanas izpilde

Katra studiju kursa iekšējās kvalitātes novērtēšanai ir jānotiek, profesionālapskates grupai tiekoties klātienē. Apskates dalībnieki pirms klātienes tikšanās, izmantojot kontroljautājumu sarakstu, novērtē attiecīgā studiju kursa saturu. Kontroljautājumu saraksts apskates dalībniekiem nodrošina vadlīnijas un dod iespēju iegūt vieglāk analizējamu un salīdzināmu informāciju par interesējošām kvalitātes raksturpazīmēm. Profesionālapskates klātienes tikšanās mērķis ir apspriest studiju kursa saturu, dodot ierosinājumus tā pilnveidei.

5. solis – Iegūto rezultātu analīze

Kontroljautājumu saraksta datu elektroniska apstrāde nav nepieciešama, jo studiju kursa satura kvalitātes vērtējums ir kvalitatīvs, to nav nepieciešams izteikt skaitļos. Novērtēšanas būtība ir saprast, kas kursā tiek mācīts, vai tas atbilst studiju programmas izvirzītajiem mērķiem un sasniedzamajiem rezultātiem, kā tas saskaņojas ar citiem studiju programmas kursiem, kā arī ieteikt kursa satura pilnveides priekšlikumus. Profesionālapaskašu dalībnieku aizpildītie kontroljautājumu saraksti studiju kursa profilējošajam mācītbspēkam sniedz papildu informāciju par to, ko nepieciešams pilnveidot studiju kursā.

6. solis – Kvalitātes pilnveides pasākumi

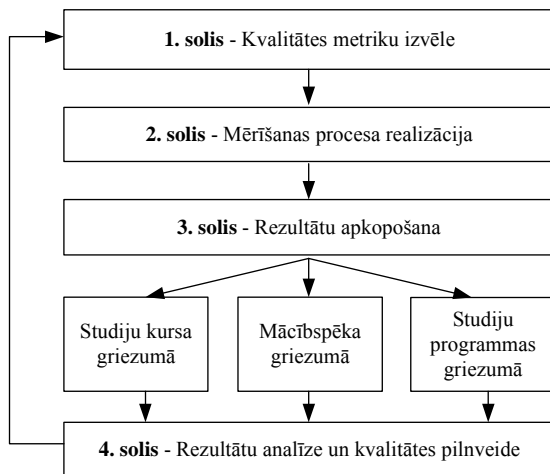
Apskates laikā studiju kursa autors (mācītbspēks) saņem ierosinājumus kursa programmas satura pilnveidei, ja tāda ir nepieciešama. Ja kursa programmā ir jāveic būtiski labojumi, kursa apskate pēc noteikta laika tiek organizēta atkārtoti.

Studiju programmas iekšējās kvalitātes novērtēšana būtu jāveic katram jaunam studiju kursam pēc programmas izstrādes, pirms tā realizācijas uzsākšanas, un atkārtoti visiem studiju programmas kursiem vismaz divas reizes programmas akreditācijas periodā.

Studiju programmas ārējā kvalitāte

Studiju programmas ārējā kvalitāte ir novērtējama pēc tās realizācijas studiju procesā un sasniegtajiem studiju rezultātiem. Studiju programmas ārējās kvalitātes novērtētāji ir studējošie, kuri apgūst attiecīgo programmu.

Studiju programmas ārējās kvalitātes novērtēšanai autore piedāvā procedūru, kurā ir izdalīti 4 soļi (sk. 9. att.).



9. att. Studiju programmas ārējās kvalitātes novērtēšanas procedūra

1. solis – Kvalitātes metriku izvēle

Darbā ir izstrādātas metrikas, kādas var izmantot programmas ārējās kvalitātes novērtēšanai, izmantojot studentu anketēšanu par katru apgūto studiju kursu. Autore ieteikums būtu vienas novērtēšanas laikā neietvert vairāk kā 10 kvalitātes apakšpazīmju, jo katram studentam semestrī ir jānovērtē vidēji 5 – 10 studiju kursi (apgūto studiju kursu skaits semestrī).

2. solis – Mērīšanas process

Kvantitatīvo datu ieguvei ir iespējams izmantot anketēšanas metodi. Studiju kursa novērtējuma anketu vajadzētu veidot kā matricas tipa anketu, un tajā papildus iekļaut vienu atvērta tipa jautājumu. Autore piedāvā izmantot

Likerta skalu ar intervālu no 1 līdz 5: ļoti zems, zems, viduvējs, augsts, ļoti augsts. Speciālas respondentu izlases grupas nav jāveido, jo vienā studiju programmā studē vidēji no 50 līdz 300 studentu un kvalitātes novērtēšanas mērķis ir iegūt katra studiju kursa novērtējumu. Tā kā studiju kursu novērtētāji ir studenti un katrā vērtēšanas reizē var piedalīties no dažiem simtiem līdz vairākiem tūkstošiem studentu (atkarībā no augstskolas lieluma), tad būtu vēlams izmantot augstskolas informācijas sistēmu vai kādu anketēšanas rīku.

3. solis – Rezultātu apkopošana

Studiju programmas ārējās kvalitātes novērtējuma apkopoto datu rezultātiem ir vairākas ieinteresētās puses: studenti, mācībspēki, studiju programmu direktori un universitātes vadība. Līdz ar to studiju kursu novērtēšanas rezultātus autore iesaka apkopot trīs griezumos

1. *Studiju kursa griezumums* – datu apkopojums metriku līmenī.
2. *Mācībspēka griezumums* – datu apkopojums par visiem viena mācībspēka vadītajiem kursiem.
3. *Studiju programmas griezumums* – datu apkopojums kvalitātes modeļa raksturpazīmju, apakšpazīmju un metriku līmenī.

4. solis – Rezultātu analīze un kvalitātes pilnveide

Studiju programmas kvalitātes novērtējuma datu apkopojums 3 griezumos nodrošinās atgriezenisko saiti dažādos līmeņos un, analizējot kvalitātes novērtējuma rezultātus gan pēc katra semestra, gan par ilgāku periodu kopā, mācībspēkiem, studiju programmu direktoriem un augstskolas vadībai atvieglos kvalitātes pilnveides pasākumu izstrādi.

Modeļa aprobācijas laikā, ārējās kvalitātes novērtēšanai no vienotā kvalitātes modeļa tika izvēlētas 3 kvalitātes raksturpazīmes un 9 apakšpazīmes (sk. 3. tabulu).

3. tabula. **Izvēlētās metrikas studiju programmas ārējās kvalitātes novērtēšanai**

Kvalitātes raksturpazīme/apakšpazīme	Metrika
Funkcionalitāte/Derīgums	Vai studiju kursa izklāsts atbilst kursa aprakstam?
Funkcionalitāte/Derīgums	Vai ir iegūtas noderīgas zināšanas, prasmes un kompetence?
Funkcionalitāte/Pareizība	Vai mācībspēks pārzina studiju kursa saturu?
Funkcionalitāte/Sadarbspēja	Vai studiju kurss nedublē citu studiju kursu saturu?
Lietojamība/Saprotamība	Vai studiju kursa tēmas mācībspēks izskaidro saprotami?
Lietojamība/Saprotamība	Vai uzsākot studiju kursu, mācībspēks iepazīstina ar tā apguves prasībām?
Lietojamība/Apmācāmība	Vai mācībspēka izmantotās mācību metodes un paņēmieni sekmē studiju kursa apguvi?
Lietojamība/Prezentējamība	Mācībspēka oratora prasmes?
Lietojamība/Cilvēciskais faktors	Vai mācībspēks konsultācijās ir pieejams?

3. tabulas turpinājums

Kvalitātes raksturpazīme/apakšpazīme	Metrika
Efektivitāte/Laikietilpība	Vai pārbaudes darbi veicina studiju kursa apguvi?
Efektivitāte/Resursu izmantošana	Vai studiju mācību materiāli ir pieejami un vai tie veicina studiju kursa apguvi?

Katras metrikas vērtējuma skalas intervāls ir no 1 līdz 5 (1–ļoti zems, 2–zems, 3–vidējs, 4–augsts, 5–ļoti augsts). Studentiem kursu vērtēšanas procesā ir jānovērtē katrs semestrī apgūtais studiju kurss pēc 3. tabulā dotajām metrikām, kā arī ir iespēja brīvā tekstā pierakstīt ierosinājumus kursa pilnveidei.

Novērtējot un apkopojot datus par visiem vienas studiju programmas studiju kursiem, var iegūt kopējo studiju programmas ārējās kvalitātes novērtējumu pēc izvēlētajām vienotā kvalitātes modeļa raksturpazīmēm, apakšpazīmēm un metrikām. Šādai datu apkopošanai autore piedāvā izmantot vienotu metodi – novērtēšanas laikā iegūto datu vērtību interpretēšanas formu (sk. 4. tabulu).

4. tabula. **Metrikas datu interpretēšanas formas piemērs**

Raksturlielums	Skaidrojums
Metrikas mērķis	Vai studiju kursa izklāsts atbilst kursa aprakstam?
Pielietojuma metode	Novērtējuma anketās saskaita doto negatīvo vērtējumu skaitu (1,2,3) un to salīdzina ar kopējo novērtējumu skaitu
Formula un mainīgo skaidrojums	$X=1 - A/B$ A – negatīvo novērtējumu skaits B – kopējais novērtējumu skaits
Rezultātu interpretācija	$0 \leq X \leq 1$ Jo tuvāk 1, jo labāk
Metrikas skala	Ordināra (1–ļoti zems, 2–zems, 3–vidējs, 4–augsts, 5–ļoti augsts)
Mērījumu tips	A, B – skaits X – skaits

Šāds datu interpretēšanas veids ir transformēts no programmatūras produkta izstrādes sfēras. Programmatūras produkta kvalitātes modelis ISO 9126 standartu līmenī (ISO 9126-2, ISO 9126-3) piedāvā kvalitātes raksturpazīmju metrikas un novērtēšanas laikā iegūto rezultātu interpretācijas veidu.

Studiju programmas kvalitātes raksturpazīmju vērtības aprēķināšanai autore piedāvā izmantot formulu:

$$KRV_i = 1 - \frac{A}{B}, \quad (1)$$

kur KRV_i – kvalitātes raksturpazīmes/apakšpazīmes/metrikas kopējais novērtējums;

- A – kopējais negatīvo novērtējumu skaits;
- B – kopējais novērtējumu skaits.

Ja katrs studiju programmas kurss tiek novērtēts ar skalu no 1 līdz 5, tad par negatīvajiem vērtējumiem tiek pieņemti 1, 2 un 3. Kvalitātes raksturpazīmes, apakšpazīmes vai metrikas kopējais novērtējums vienmēr būs robežās no 0 līdz 1. Jo šis vērtējums ir tuvāk skaitlim 1, jo kvalitāte ir augstāka. Šādā rezultātu interpretēšanas variantā autore piedāvā skatīties novērtējuma virzību. Ja vērtējums ir tuvāk 0.5 vai zem tā, tad ir jādomā par kvalitātes pilnveides pasākumiem pēc attiecīgajām kvalitātes raksturpazīmēm, apakšpazīmēm vai metriķām.

Tā kā darba autore studiju programmas ārējās kvalitātes novērtējuma datus piedāvā apkopot 3 griezumos, tad arī to vizuālais atspoguļojums ir jānodrošina 3 griezumos. Aprobējot vienoto kvalitātes modeli Latvijas Lauksaimniecības universitātes studiju programmu novērtēšanai, tika pieņemts lēmums novērtējuma datus atspoguļot divos veidos:

1. universitātes informācija sistēma – studiju kursa un mācībspēka griezumā;
2. jauns lietojums – studiju programmas greizumā.

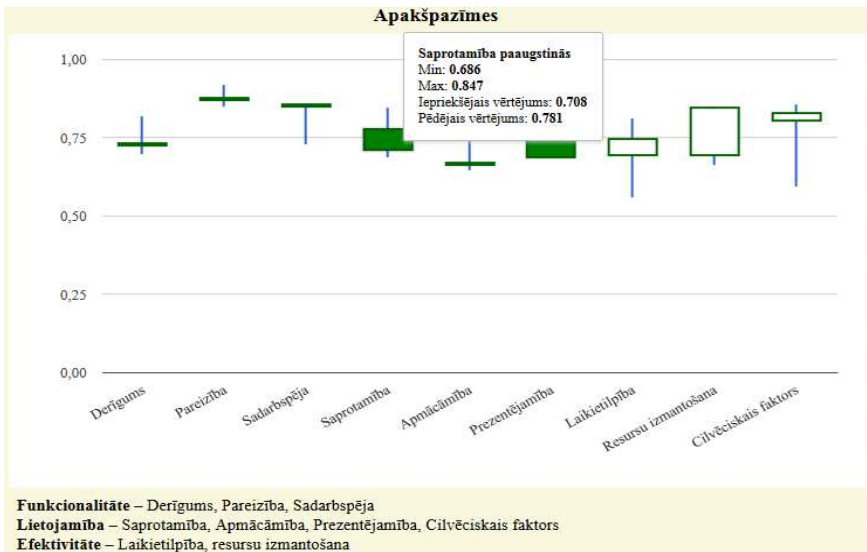
Lēmums ir pamatojams ar to, ka studiju kursu un mācībspēku greizums ir domāts tieši konkrētiem mācībspēkiem un augstskolas vadībai un tie nav paredzēti publiskai apskatei. Šajos griezumos dati ir jāapkopo katras metrikas līmenī, lai tas palīdzētu mācībspēkiem identificēt kursu kvalitātes vājākās vietas un veikt attiecīgus pilnveides pasākumus. Uzlabojot katru kursu atsevišķi, kopumā tiks pilnveidota arī studiju programmas kopējā kvalitāte. Savukārt studiju programmas greizumā datu apkopojums notiek pēc VKM raksturpazīmēm un ir domāts, lai visi augstskolas studenti un mācībspēki varētu iepazīties ar katras augstskolas studiju programmas ārējo kvalitāti. Šim nolūkam darba ietvaros tika izstrādāta jauna lietojumprogrammatūra (SPKV – studiju programmas kvalitātes vērtējums), kuras mērķis ir nodrošināt iespēju katram LLU studentam, mācībspēkam un vadībai iepazīties ar studiju programmas kvalitātes modeli un iegūt datus par universitātes studiju programmu ārējās kvalitātes kopējo novērtējumu.

SPKV lietotājiem piedāvā grafiku veidā apskatīt studiju programmas kvalitātes novērtējumu pēc katras no vienotā kvalitātes modeļa izvēlētās raksturpazīmes un apakšpazīmes. Tā kā kvalitātes modeļa izmantošana studiju programmas novērtēšanai ir jauna pieeja un kvalitātes raksturpazīmju/apakšpazīmju skaidrojumi attiecībā uz studiju programmām ir jauni, tad SPKV ir piedāvāts arī metriķu datu attēlojums.

Datu attēlošanas diagrammās (sk. 10. att.) katrai studiju programmas kvalitātes raksturpazīmei/apakšpazīmei/metriķai ir attēlots:

- novērtēšanas laikā iegūtais minimālais novērtējums;
- novērtēšanas laikā iegūtais maksimālais novērtējums;
- iepriekšējās novērtēšanas rezultāts;

- pēdējās novērtēšanas rezultāts;
- pēdējā un iepriekšējā novērtējuma salīdzinājums – pieaug vai samazinās. Šis salīdzinājums ir redzams arī vizuāli – ja stabiņš diagrammā ir aizkrāsots, tad kvalitātes novērtējums pieaug, ja tukšs, tas samazinās.



10. att. Studiju programmas kvalitātes apakšpazīmju kopsavilkums

Uzsākot kvalitātes pilnveides pasākumus, svarīgi ir saprast vai pilnveide ir jāveic pēc visām kvalitātes raksturpazīmēm. Tāpēc autore, analizējot sešos semestros iegūtos studiju programmu ārējās kvalitātes novērtējuma datus, vēlējās noskaidrot vai pastāv sakarība starp kvalitātes raksturpazīmēm un apakšpazīmēm. Sakarības noteikšanai tika izmantots Spirmena rangu korelācijas (SRK) koeficients, kas mēra sakarības starp rangu skalas mainīgajiem.

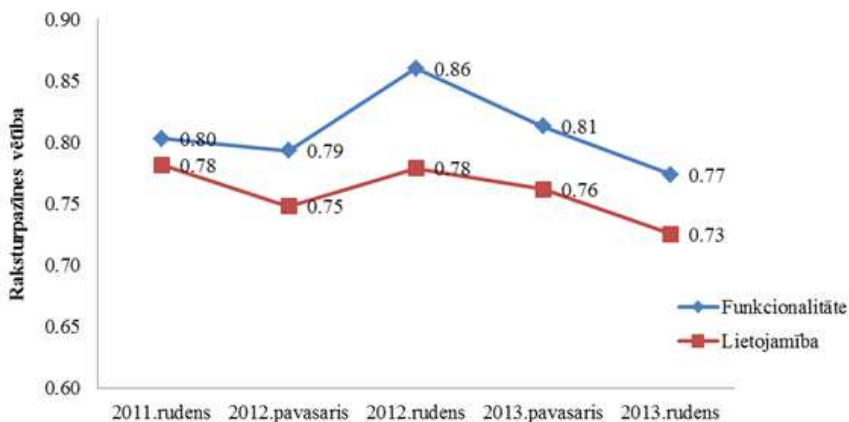
Datu analīze parādīja, ka pastāv cieša un ļoti cieša datu sakarība starp visām trim kvalitātes raksturpazīmēm – *funkcionalitāte*, *lietojamība* un *efektivitāte* (sk. 5. tabula). Tabulā treknrakstā atzīmēts koeficients, kurš norāda uz ļoti ciešu sakarību.

5. tabula. Kvalitātes raksturpazīmju savstarpējās sakarības koeficienti

Raksturpazīme	Funkcionalitāte	Lietojamība	Efektivitāte
Funkcionalitāte	1		
Lietojamība	0.831*	1	
Efektivitāte	0.746*	0.709*	1

* p=0.01, n=94

Ja kvalitātes pilnveides pasākumos tiks palielināta kvalitāte pēc raksturpazīmes *funkcionalitāte*, tad var gaidīt, ka palielināsies arī pēc raksturpazīmes *lietojamība* un otrādi (sk. 11. att.).



11. att. Studiju programmas kvalitātes novērtējums pēc raksturpazīmēm *funkcionalitāte* un *lietojamība* pa studiju semestriem

Apakšpazīmju līmenī tika noteikts vai pastāv sakarības starp vienas kvalitātes raksturpazīmes apakšpazīmēm, gan arī starp vairāku raksturpazīmju apakšpazīmēm. Sakarības ar ticamību 99% ($p=0.01$) pastāv starp visām kvalitātes apakšpazīmēm. Apskatot apakšpazīmju savstarpējo saistību vienas raksturpazīmes ietvaros, tad **ļoti cieša** saistība pastāv starp:

- funkcionalitāte – *pareizība* un *derīgums* (SRK=0.81). Studiju programmas kursu satura atbilstība definētajiem mērķiem un uzdevumiem ietekmē to, vai tiks sasniegti definētie studiju rezultāti;
- lietojamība – *apmācāmība* un *prezentējamība* (SRK=0.835). Studiju programmas kursu realizācijā izmantojamās metodes ietekmē to vai studiju kursu satura izklāsts ir saprotams un ieinteresējošs;
- efektivitāte – *laikietilpība* un *resursu izmantošana* (SRK=0.849). Studiju programmas satura apguvei pieejamie mācību materiāli ietekmē to, vai studiju programmas apguves laiks ir pietiekams.

Kvalitātes uzlabošana un nodrošināšana izglītības iestādē neapšaubāmi nosaka esošo kontroles mehānismu pārveidošanu un jaunu mehānismu ieviešanu. Šī politika būs efektīva tikai tad, ja tai būs kompleksa attīstības programma, kura ietver virkni tādu elementu kā esošās likumdošanas izmaiņšana, jaunu stimulu radīšana un jaunas pieredzes iegūšana. Kvalitātes uzlabošanas politika kā stratēģisku funkciju ietver arī mācībspēku un studentu aktīvu līdzdalību tajās aktivitātēs, kas uzlabo un nodrošina kvalitāti. Studiju process nav tikai katra individuāls darbs, bet gan kolektīva darbs, un studiju kvalitāti var nodrošināt tikai ar kopīgu visa kolektīva darbību. Šajā ziņā studiju

procesam arī ir cieša analogija ar programmatūras izstrādi, jo mūsdienās programmatūras produkts izteikti ir kolektīva kopīga darba rezultāts.

SECINĀJUMI

1. Programmatūras produkta kvalitāte nav viennozīmīgs jēdziens, un tā vienmēr būs atkarīga no skatupunkta no kāda tā tiek aplūkota. Par kvalitāti var spriest un to novērtēt, ja ir izvirzītas konkrētas kvalitātes prasības. Līdz ar to produkta kvalitātes rakstурpazīmes un to savstarpējā saistība ir jāapskata katrā konkrētā situācijā individuāli.
2. Kvalitāte programmatūras izstrādē vienmēr iet kopā ar papildu izmaksām. Lai samazinātu kvalitātes izmaksas nākotnē, ir jāveic dažādi ieguldījumi preventīvos pasākumos. Viens no šādiem pasākumiem ir personāla sagatavošana, kas sākas jau studiju laikā. Jauno informācijas tehnoloģiju speciālistu apmācībā uzsvars ir jāpārliet no tā, kā izstrādāt programmatūras produktu, uz to, kā izstrādāt kvalitatīvu programmatūras produktu.
3. Programmatūras kvalitātes modeļiem ir būtiska nozīme, lai definētu un novērtētu programmatūras produkta kvalitāti, jo modeļi veido vienotu skatījumu uz produkta kvalitāti hierarhiskas kvalitātes rakstурpazīmju struktūras veidā.
4. Kvalitātes modeļu daudzveidība un jaunu modeļu izveide parāda, ka to autori mēģina piedāvāt izstrādātājiem un lietotājiem intuitīvi saprotamākas kvalitātes rakstурpazīmes un apakšpazīmes, mainot tikai nosaukumus, jo pēc nozīmes tās ir vienādas. Piemēram, ISO 9126 modelī ir apakšpazīme *klūdu tolerance*, bet FURPS modelī *klūdu biežums un nopietnība*, savukārt ISO 9126 modelī apakšpazīme *maināmība*, bet Boehm modelī *izaugtspēja* un FURPS modelī *paplašināmība*.
5. Vienotais kvalitātes modelis nodrošina vienotu pieeju kvalitātes prasību izvirzīšanai un novērtēšanai informācijas tehnoloģiju izstrādē un studijās.
6. Studiju programmas kvalitātes novērtēšanā var izmantot vienoto kvalitātes modeli, jo gan programmatūra, gan studiju programma ir abstrakti produkti, kurus nevar fiziski nosvērt, nomērīt vai kā citādi vizuāli novērtēt.
7. Izmantojot vienoto kvalitātes modeli, augstākās izglītības iestādei studiju programmas kvalitāte būtu jāvērtē ne tikai tās akreditācijas laikā, kas ir reizi divos vai sešos gados, bet gan regulāri un, balstoties uz iegūtajiem kvalitātes novērtējumiem, ir jāveic pilnveides pasākumi gan programmas iekšējās, gan ārējās kvalitātes uzlabošanai.
8. Studiju programmas iekšējās kvalitātes novērtēšanā var izmantot profesionālapskates, bet ārējās kvalitātes novērtēšanā ir jāiesaista studenti, jo viņi tiešā veidā ir iesaistīti studiju procesā.
9. Ar ticamību 99% studiju programmas ārējās kvalitātes modelī pastāv korelācija starp rakstурpazīmēm un apakšpazīmēm. Ļoti cieša saistība pastāv starp šādām apakšpazīmēm:

- funkcionalitātes raksturpazīmei – pareizība un derīgums (SRK=0.81). Studiju programmas kursu satura atbilstība definētajiem mērķiem un uzdevumiem ietekmē to, vai tiks sasniegti definētie studiju rezultāti;
 - lietojamības raksturpazīmei – apmācāmība un prezentējamība (SRK=0.835). Studiju programmas kursu realizācijā izmantojamās metodes ietekmē to, vai studiju kursu satura izklāsts ir saprotams un ieinteresējošs;
 - efektivitātes raksturpazīmei – laikietaipība un resursu izmantošana (SRK=0.849). Studiju programmas satura apguvei pieejamie mācību materiāli ietekmē to, vai studiju programmas apguves laiks ir pietiekams;
 - funkcionalitātes un efektivitāte raksturpazīmēm – *derīgums* un *laikietaipība* (SRK=0.84). Studiju programmā definētie studiju rezultāti ietekmē to, vai programmas apguvei nepieciešamais laiks ir pietiekams.
10. Studiju programmu kvalitātes raksturpazīmju savstarpējo saistību var izmantot kvalitātes pilnveides pasākumu organizēšanā. Uzlabojot kvalitāti pēc vienas raksturpazīmes, ar ticamību 99% kvalitātes uzlabosies arī pēc citas.
 11. Programmatūras kvalitātes raksturpazīmju un studiju kursu savstarpējā saistība nodrošinās, ka studenti jau studiju laikā, apgūstot konkrētus studiju kursus, uzzinās, kas un kā var ietekmēt programmatūras kvalitāti. Tādējādi tiks radīti priekšnoteikumi, lai nākotnē izstrādātu kvalitatīvus programmatūras produktus.

Priekšlikumi turpmākajiem pētījumiem

1. Turpmākajos pētījumos iespējams analizēt ieguvumus no profesionālapaskašu ieviešanas studiju programmas iekšējās kvalitātes novērtēšanā.
2. Turpmākajos pētījumos var pārbaudīt un novērtēt, vai studiju programmas ārējās kvalitātes modelī pastāvošo Spirmena ranga korelāciju starp raksturpazīmēm un apakšpazīmēm var izmantot kvalitātes pilnveides pasākumu organizēšanai.
3. Pielāgot vienoto kvalitātes modeli procesu kvalitātes novērtēšanai.
4. Turpmākajos pētījumos būtu nepieciešams analizēt, kādi ir ieguvumi no autores piedāvātās koncepcijas par kvalitātes modeļa iekļaušanu informācijas tehnoloģiju studiju programmas saturā.
5. Piedāvāt izstrādāto vienoto kvalitātes modeli citām augstskolām savu studiju programmu kvalitātes novērtēšanai.

Latvia University of Agriculture
Faculty of Information Technologies
Department of Computer Systems

Sandra Sprōģe

**SINGLE QUALITY MODEL OF INFORMATION
TECHNOLOGY STUDIES AND DEVELOPMENT**

SUMMARY

of the PhD thesis for acquiring the doctor's degree in the field
of Information Technologies (Dr.sc.ing.)



IEGULDĪJUMS TAVĀ NĀKOTNĒ



EIROPAS SAVIENĪBA

Development and design of the PhD thesis is financed by
the European Social Fund

Jelgava 2014

PARTICULARS

Research has been done at: the Department of Computer Systems, Faculty of Information Technologies, Latvia University of Agriculture, Lielā iela 2, Jelgava, Latvia.

Experimental research has been done at: the Department of Computer Systems, Faculty of Information Technologies, Latvia University of Agriculture, Lielā iela 2, Jelgava, Latvia.

Scientific adviser of the PhD thesis: Rudīte Čevere, Dr.sc.comp., professor, Latvia University of Agriculture.

The Thesis has been approved at the enlarged academic meeting of the Department of Computer Systems, Faculty of Information Technologies of Latvia University of Agriculture on 28 May 2014. Minutes No 5.

The PhD thesis was developed with the assistance of the European Social Fund (ESF) project “Support for the Implementation of Doctoral Studies at Latvia University of Agriculture”, Agreement No 2009/0180/1DP/1.1.2.1.2/09/ IPIA/VIAA/017.

Official reviewers

1. **Larisa Zaiceva**, Dr.sc.ing., professor of Riga Technical University.
2. **Egils Stalidzāns**, Dr.sc.ing., professor of Latvia University of Agriculture.
3. **Vladislav Fomin**, Dr.habil., professor, Vytautas Magnus University, Lithuania.

The defence of the PhD thesis in an open session of the Promotion Council in the field of Information Technologies of Latvia University of Agriculture will be held at 11 a.m. on 22, December, 2014, Room 218, Faculty of Information Technologies, Lielā iela 2, Jelgava, Latvia.

The PhD thesis is available for the review at the Research Library of Latvia University of Agriculture, Lielā iela 2, Jelgava and online at http://llufb.llu.lv/promoc_darbi.html.

You are welcome to send your comments, signed and scanned to the secretary of the Promotion Council – Lielā iela 2, Jelgava, LV-3001; phone (+371) 63005621; e-mail: tatjana.tabunova@llu.lv.

Council secretary: Tatjana Tabunova, Mg. paed., lecturer.

TABLE OF CONTENTS

Approbation of the phd thesis	35
Introduction	37
Topicality of the research	37
Aim and tasks of the PhD thesis	38
Research methods	39
Scientific novelty	39
Research theses	39
Practical novelty	40
Structure and volume of the PhD thesis.....	40
1. Quality in software development.....	40
2. Software product quality models	42
3. Quality in higher education	44
4. Single quality model	49
5. Use of single quality model	53
Conclusions	62
Bibliography.....	64

APPROBATION OF THE PHD THESIS

The research results are presented in the following publications:

1. Cevere R., **Sproge S.** (2013). Introduction of Software Product and Process Quality Aspects in the Study Courses of Information Technologies. In: *Proceedings of the 6th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), pp. 267-275.
2. **Sproge S.**, Cevere R. (2012). Position and Role of Feedback in the Software Quality Life Cycle. *Quality Issues and Insights in the 21st Century*, Vol. 1, pp. 29-40, ISSN 2029-9575 (indexed in EBSCO).
3. Cevere R., **Sproge S.** (2012). Application of Single Quality Model for Self-assessment of Study Programme. In: *Proceedings of the 3rd International Workshop "Intelligent Educational Systems and Technology-enhanced Learning"*. Riga: RTU (Riga Technical University), pp. 54-64.
4. **Sproge S.**, Cevere R. (2012). Assessment of Study Programme Quality at Higher Education Institution. In: *Proceedings of the 11th International Scientific Conference "Engineering for Rural Development"*. Jelgava: LLU (Latvia University of Agriculture), pp. 663-668 (indexed in Scopus).
5. **Sproge S.**, Cevere R. (2012). Quality Models in Software Product Development Life Cycle. In: *Proceedings of the 5th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), pp. 69-76, ISBN 978-9984-48-065-7.
6. **Sproge S.** (2011). Evaluation of Study Programme External Quality. In: *Proceedings of the Annual 17th International Scientific Conference "Research for Rural Development"*. Jelgava: LLU (Latvia University of Agriculture). Vol. 1, pp. 179-185, ISSN 1691-4031 (indexed in Scopus, EBSCO Academic Search Complete).
7. Cevere R., **Sproge S.** (2011). Quality Life Cycle of the Study Programme and Methodology for Evaluation. In: *Proceedings of the 3rd International Conference "Institutional Strategic Quality Management - ISQM2011"*, Sibiu, Romania, pp. 65-72, ISBN 978-973-598-913-2. Available at: http://proiecte.aracis.ro/fileadmin/Academis/A0/ISQM_2011/Proceedings_ISQM_2011.pdf (indexed in EBSCO Education Research Complete).
8. **Sproge S.**, Cevere R. (2011). Software Development Quality Models in Engineering Education. In: *Proceedings of the 10th International Scientific Conference "Engineering for Rural Development"*. Jelgava: LLU (Latvia University of Agriculture), pp. 536-541, ISSN 1691-3043 (indexed in EBSCO Academic Search Complete; indexed in Scopus, cited in Scopus).
9. Cevere R., **Sproge S.** (2011). Methodology for Evaluation of Internal Quality of the Study Programme. EQANIE Conference Vienna 2011. Available at: <http://www.eqanie.eu/pages/events/past-events/conference-vienna-2011/proceedings.php>.

10. Čevere R., **Sproģe S.** (2010). Application of Software Quality Models in Evaluation of Study Quality. *Problems of Education in the 21st Century*, Vol. 21, pp. 37-48, ISSN 1822-7864 (indexed in EBSCO Education Research Complete, EBSCO Central & Eastern European Academic Source).
11. **Sproģe S.**, Čevere R. (2010). Quality Model of the Curricula of Information Technology Studies. In: *Proceedings of the 4th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), pp. 148-157, ISBN 978-9984-48-022-0 (indexed in Web of Science, cited in Web of Science).
12. **Sproģe S.**, Čevere R., Arhipova I. (2008). Development of the Faculty Quality Model and Definition of Control Processes. *Scientific Proceedings of Riga Technical University. Computer Science & Information Technology and Management Science*. Riga: RTU, pp. 56-62, ISSN 1407-7493.
13. Čevere R., **Sproģe S.**, Arhipova I. (2008). How to Use an Experience of the Development of ISO 9001 Certified Quality System in Study Process. In: *Proceedings of the 3rd International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), pp. 126-136, ISBN 978-9984-784-68-7 (indexed in EBSCO Central & Eastern European Academic Source).

The research results have been presented at the following conferences:

13. "Introduction of Software Product and Process Quality Aspects in the Study Courses of Information Technologies". The 6th International Scientific Conference "Applied Information and Communication Technologies AICT'2013, April 25-26, 2013, Jelgava, Latvia.
14. "Application of Single Quality Model for Self-assessment of Study Programmes". Riga Technical University, the 3rd International Workshop on Intelligent Educational Systems and Technology – Enhanced Learning INTEL_EDU 2012, October 10, 2012, Riga, Latvia.
15. "Software Development Quality Models in Engineering Education". The 11th International Scientific Conference "Engineering for Rural Development", May 24-25, 2012, Jelgava, Latvia.
16. "Quality Models in Software Product Development Life Cycle". The 5th International Scientific Conference "Applied Information and Communication Technologies AICT'2012, April 26-27, 2012, Jelgava, Latvia.
17. "Quality Model of Study Programme". The joint 3rd World Congress of Latvian Scientists, October 24-26, 2011, Riga, Latvia.
18. "Quality Models in Software Product Development Life Cycle". Riga Technical University, the 52nd International Scientific Conference, October 12-15, 2011, Riga, Latvia.

19. "Quality Life Cycle of the Study Programme and Methodology for Evaluation". The 3rd Institutional Strategic Quality Management: from Minimum Requirements to Overall Quality Improvement - ISQM2011, July 14-16, 2011, Sibiu, Romania.
20. "Software Development Quality Models in Engineering Education". The 10th International Scientific Conference "Engineering for Rural Development", May 26-27, 2011, Jelgava, Latvia.
21. "Evaluation of Study Programme External Quality". The 17th International Scientific Conference "Research for Rural Development", May 18-20, 2011, Jelgava, Latvia.
22. "Methodology for Evaluation of Internal Quality of the Study Programme". EQANIE-Conference: Learning Outcomes and Quality Management in Informatics Education, February 17-19, 2011, Vienna, Austria.
23. "Quality Model of the Curricula of Information Technology Studies". The 4th International Scientific Conference "Applied Information and Communication Technologies AICT'2010", April 22-23, 2010, Jelgava, Latvia.
24. "Development of the Faculty Quality Model and Definition of Control Processes". Riga Technical University, the 50th International Scientific Conference, October 12-14, 2009, Riga, Latvia.

INTRODUCTION

Topicality of the research

The notion of quality is one of the notions being currently in practice and it is used by the society in almost all spheres. Quality is getting attention, to a certain extent, in all areas of human endeavour. On the one hand, it is good as it attracts general attention, facilitates the performance of specific tasks and the raising of funds that really contribute to the development of a particular sphere (Hoyer, 2001). Quality has been regarded as one of the major drivers of competitive strategy in every industry (Li & Meissner, 2008). On the other hand, it may be dangerous since it devalues the notion of quality itself and it may seem that speaking about quality simultaneously improves it. It is one of the mistakes made in the sphere of quality assurance. Real necessity is another aspect related with quality. It is one of the possibilities to increase work efficiency and performance quality.

Quality in software engineering has always been and it will always be an essential aspect. Non-quality products are still being developed regardless that many and various quality models and methods to ensure the quality assurance in software engineering have been developed during the course of time to postulate and assess the quality requirements for a software product. Many software development projects fail due to imprecise definition of quality

requirements. Software product quality cannot be added to a product after its development but rather shall be created during the product development stage (Bawane & Srikrishna, 2010). Part of the quality of a software end product forms in every stage and process of its life cycle, and various factors affect its formation.

The personnel participating in the software quality development plays a significant role in software quality. The core fundamentals for their professional qualification, knowledge and competence on software quality are laid during studies while mastering study programmes related with information technologies. Therefore, the study process should also be included into the software product life cycle. In the sphere of education, study programmes have their own quality criteria, which mainly refer to the external quality of a study programme, i.e. the quality expressed in the organisation of study process and the study process itself. Scientific publications provide a set of research on assurance, measurement, and evaluation of quality in higher education area (Tam, 2001; Tan & Kek, 2004; Clewes, 2003; Van Der Wende & Westerheijden, 2001). However, reviewing literature on research related with the ensuring of software quality, the author failed to find studies applying a single view on quality throughout the entire stages of the life cycle. Similar situation was observed reviewing publications on the valuation of education quality, where the author failed to find studies on the quality valuation of study programmes with the application of quality models including defined quality characteristics and subcharacteristics. The author in her PhD thesis offers to establish a single view on quality throughout the entire life cycle of software encompassing the quality of personnel training and studies. The author has proposed a recommendation to perceive a study programme as software product and to apply a single quality model based on the hierarchic quality model of software product for the evaluation of their quality. The PhD thesis includes the author's developed concept for the inclusion of software product quality in the content of undergraduate programme of information technologies to develop students' abilities to master the impact of every software development step on the quality of end results, and thus, to create preconditions for the future development of quality software products.

Aim and tasks of the PhD thesis

The research aim is to offer a solution for the improvement of software product quality through the improvement of quality assessment of study programmes in information technologies and acquisition of software product quality.

The following tasks have been advanced in the research process to achieve the set aim:

- 1) to study the role and significance of quality in the life cycle of software products;

- 2) to analyse quality models of software products;
- 3) to analyse the present methods and approaches applied to management of the study process quality;
- 4) to analyse the applicability of software product quality model in the sphere of education;
- 5) to define an extended life cycle of software product quality model with the included study process;
- 6) to develop a single quality model for the quality evaluation of software development, and evaluation of software development and study quality;
- 7) to appropate the single quality model for the quality evaluation of study programmes;
- 8) to develop a concept for the inclusion of software quality model in the study programme of information technologies.

Research methods

Literature review, analysis and synthesis, classification, peer review evaluations, questionnaires, processing and analysis of statistical data, programming.

Scientific novelty

1. A single quality model applicable within the framework of the whole life cycle has been developed for the quality evaluation of software product, its development and maintenance processes as well as for the quality evaluation of study programme development and implementation process.
2. A quality model has been developed for the evaluation of a study programme quality:
 - a. the single quality model is adapted to the characterisation of a study programme;
 - b. methods are developed for the application of the model and metrics for the internal and external evaluation of a study programme.
3. A methodology based on the single quality model has been developed for the processing and interpreting of study quality evaluation results.
4. A concept has been developed for the improvement of mastering software product quality during the studies of becoming IT specialists.

Research theses

1. Quality in software engineering and higher education is not an unequivocal notion. A uniform view on quality is lacking in the evaluation of interrelated processes and objects.

2. A single quality model is applicable for the quality evaluation of study programmes and software products perceiving a study programme as software product.
3. The development of a single structure quality model allows obtaining a uniform view on the quality in the entire software life cycle and on all the objects (products, intermediate products, and processes).
4. The introduction of a software product quality model and quality evaluation approaches included into it create preconditions for the improvement of software product quality during the study process of information technologies.

Practical novelty

1. The single quality model may be applied both in the development process of software and study process.
2. The developed single quality model may be applied at higher education institutions for the quality evaluation of study programmes content and the quality evaluation of software implementation.
3. The developed usage for the evaluation data collection, analysis and presentation of study programme quality may be applied for decision-making related with the quality improvement of study programme.

Structure and volume of the PhD thesis

The PhD thesis is written in Latvian containing abstract, introduction, 5 chapters, conclusions, bibliography, and 6 annexes, including 26 tables, 44 figures, and 4 formulae. The total length of the PhD thesis is 125 pages and 172 sources of literature have been used for the research purpose.

1. QUALITY IN SOFTWARE DEVELOPMENT

Quality being an unequivocal term is its characteristic feature (Reeves, Bednar, 1994). Quality has many and various definitions, various conceptions and evaluation types (Garvin, 1987). The most common directions in software development sphere examining quality include:

- conformance to the requirements set previously (Crosby, 1979);
- suitability for use (Juran, 1988);
- degree to which user's needs or expectations are satisfied (Deming, 1988; Ishikawa, 1985).

Crosby emphasises that it is difficult to evaluate the finished product if quality has no previously set specifications. Therefore, the notions "good quality" and "bad quality" remain only phrases since these notions mean different things and requirements to each and every person. It is important to clearly define quality to be able to measure and manage the concept.

One of Deming's (Deming, 1988) strongest points is that quality must be defined in terms of customer satisfaction and not as conformance to specification. Deming considers that the difficulty in defining quality is to translate future needs of the user into measurable product characteristics so that the product can be designed and turned out to give satisfaction at a price that the user will pay. This is not easy, and as soon as a customer feels fairly successful in the endeavour, his needs regarding the product may change and, hence, his satisfaction. The accomplishment of product requirements is an integral task of a product developing company. Every employee shall bear the responsibility for the quality of his work.

Juran's point regarding quality is that one cannot use the word "quality" in terms of satisfying customer expectations or specifications as it is very hard to achieve this (Juran, 1988). Instead he suggests defining quality as "fitness for use" – which indicates references to requirements and products characteristics. Juran's definition could be basically interpreted as Crosby's definition – conformance to specification.

In general, all aspects viewed in the PhD thesis on the notion of quality in software engineering reflect four core principles related with quality:

1. necessity to ensure qualitative development processes;
2. necessity to define the notion of quality in every particular case;
3. necessity for all stakeholders to observe a uniform view on quality in every particular case;
4. necessity to follow the interrelation and consistency of quality characteristics.

The attempts to introduce and to assure quality developing software products have already been made from the early stages of programming. A single view documented in international standards of software engineering has been drafted during the course of development and discussion of different approaches and methods. These standards are aimed at helping organise software product development to assure and improve its quality.

The PhD thesis includes the study and application of methods and approaches related with software quality assurance and incorporated in the respective standards of software engineering. The basic issues studied refer to the software life cycle and the place of quality in it, software product quality and its evaluation, and the process of measuring.

Definition of software product quality requirements based on the software quality model and methods developed for the quality evaluation are summarised in voluminous series of standards:

- ISO/IEC 9126 Software Engineering – Software Product Quality;
- ISO/IEC 14598 Software Engineering – Software Product Evaluation;
- ISO/IEC 25000 Software Engineering – Software Product Requirements and Evaluation.

The ISO/IEC 9126 standard provides the life cycle of software product quality (Figure 1), where the author of the present PhD thesis offers including

also the studies in information technologies since new IT specialists are trained at this time. This study period may be considered as one of the preventive measures for the reduction of software product quality costs.

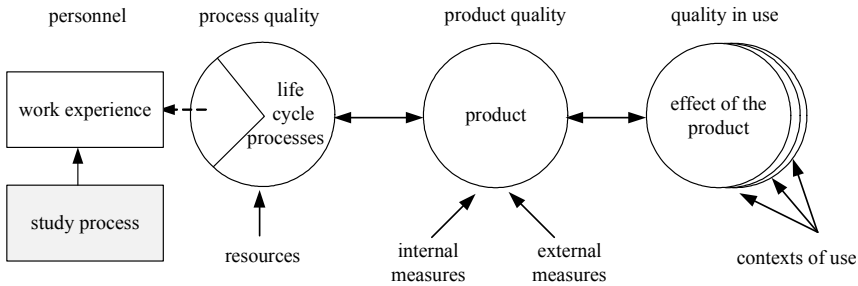


Figure 1. **The extended quality life cycle**

Studies done by researchers outline that the quality costs rapidly increase with later detection of quality problems in the software life cycle. The sooner software product quality control and assurance measures are undertaken, the lower are the costs.

The role of personnel in the development of software product is prescribed also at the level of standards. The ISO/IEC 12207-2008 standard (*Software Life Cycle Processes*) establishes the human resource management process aimed to provide an organisation with the necessary human resources and competence of human resources suitable for business needs. The basic tasks for the mentioned process include – identification of personnel professional skills, skills development plans, assurance of mastering skills, and knowledge management. The inclusion of the study process into the life cycle of software product quality allows the achievement and assurance of individual experience of becoming specialists on the product quality. Preventive measures will be undertaken for the development of quality products in future if the significance and role of quality characteristics at different software product development processes are mastered during studies.

The development and application of a single quality model in the entire software product quality life cycle could enhance the establishment of a uniform view on all its stages, intermediate and final results.

2. SOFTWARE PRODUCT QUALITY MODELS

The following definition characterises the quality model: defined characteristics and their interrelation which create the basis for the specification of quality requirements and quality assurance (ISO/IEC 9126-1, 2001).

Historically, many software quality models for definition and evaluation of product quality have been developed and have improved each other. The

present PhD thesis analyses models most commonly discussed in scientific publications: McCall's model, Boehm's model, the FURPS model, the ISO 9126 model, and Dromey's model. These quality models are not compared mutually but the comparison is done with the ISO 9126 model which is internationally recognised and transposed to the standard.

The development of software product quality models through various scientific publications reveal inter-references and comparison of models at the level of characteristics (Al-Qutaish, 2010; Dubey et al., 2012; *Khayami* et al., 2009; Rawashdeh & Matalkah, 2006; Ortega et al., 2003). The author of the present PhD thesis has compared models also at the level of subcharacteristics and a great deal of attention has been paid to the terms used in each model to denote characteristics and subcharacteristics. Terms used for denoting quality characteristics is a significant factor since the content of individual situation and the opinion of stakeholders involved in quality evaluation are essential for the application of the quality model.

The analysis of models done in the PhD thesis at the level of characteristics reflects that basically the same quality characteristics summarised in the ISO 9126 model are used as quality characteristics in all other models. In case any model uses a different name, still it conforms by the sense to some characteristic or subcharacteristic of the ISO 9126 model.

If the quality characteristics in the analysed models coincide by the names for 77%, then the proposed subcharacteristics essentially differ both by names and number. Interrelation of characteristics and subcharacteristics also differs.

Analysing quality subcharacteristics mentioned in all models for every subcharacteristic, the author has displayed those that are synonyms for the subcharacteristics mentioned in the ISO 9126 model and those that would be additionally considerable for the definition of quality requirements and later evaluation of the respective quality characteristic. In total, five analysed quality models propose 78 unique subcharacteristics for the evaluation of quality characteristics. The largest number of subcharacteristics is mentioned for such quality characteristics as *usability* and *functionality*. Various quality models propose different names for one and the same quality subcharacteristic irrespective of their equal meaning. The author would like to note that a more appropriate name in individual cases could help software product developers and customers propose and evaluate the respective quality parameter, for example, the ISO 9126 model proposes a subcharacteristic *analysability* for the quality characteristic *maintainability*; however, such terms as *consistency*, *conciseness*, *readability*, and *structuredness* would better help understand this quality characteristic.

Development of the new ISO 25010 quality model outlines a tendency that the present quality models are not perfect and the development of quality models is still going on. The author would like to note that the new ISO quality model does not propose new quality characteristics by their essence but the restructured existing characteristics and subcharacteristics. The quality

subcharacteristics proposed as completely new ones are already mentioned in other models analysed by the author, for example, a subcharacteristic *modularity* is proposed in McCall's model, *integrity* – in Boehem's model, *reusability* - in McCall's and Boehem's models, while *supportability* - in the FURPS model.

Therefore, the author proposes to develop a new quality model displaying two new types of subcharacteristics – additional and optional subcharacteristics to make the same characteristic more understandable for a specific audience. The development of a model is described in Chapter 4. The development of a new model characterises with sufficiently general terms of characteristics and subcharacteristics used in software quality models and the author proposes to use them to evaluate also study programmes not only for the evaluation of software quality. Hence, the author demonstrates that software product quality models are transformable to the sphere of education.

3. QUALITY IN HIGHER EDUCATION

Quality of higher education and its results are characterised with a set of different questions having no unequivocal answer, for example, what the quality of study programme is and what the term “good quality” means. Is a quality study programme an accredited study programme or the one which conforms to the stakeholders' interests and needs, or the one which graduates satisfy the potential employers? Though, it may be said that a study programme is qualitative if its study process is co-ordinated, managed, and controlled; it has the necessary material base and it has achieved the defined study outcomes both in every study course and the entire study programme.

In 1993, Harvey, Green and Burrows developed a methodology for evaluation quality in higher education. The methodology is based on the following quality criteria:

- sufficiency of physical resources;
- sufficiency of human resources;
- clarity of aims and tasks for all stakeholders;
- conformance of study courses and their content to the aims and tasks of the study programme;
- participation of students at all levels;
- conformance of the study programme content to the degree or qualification to be awarded;
- objectivity in evaluation;
- consequence between the evaluation and the tasks of a study course;
- obtaining of feedback after evaluation;
- succession of acquisition knowledge and skills.

The proposed criteria have not been essentially changed in the present methodologies for the evaluation of study quality. Joining the European Higher

Education Area, the Bologna Declaration recommends driving directly towards measuring of study outcomes and competences to evaluate the quality of study programmes (Boloņas process..., 2008).

The author of the present PhD thesis does not look at the quality of higher education in general. She instead examines the quality of a study programme as it is a product offered by higher education institutions to their customers – prospective specialists of the respective sector. The study programme cannot be examined in isolation from the higher education institution itself and the education system established there as well as the entire state higher education system. Study programme may be examined through the systemic approach. By analogy with a software product, a study programme also is an element in the higher education system which, in turn, is an element in higher education system (Figure 2).

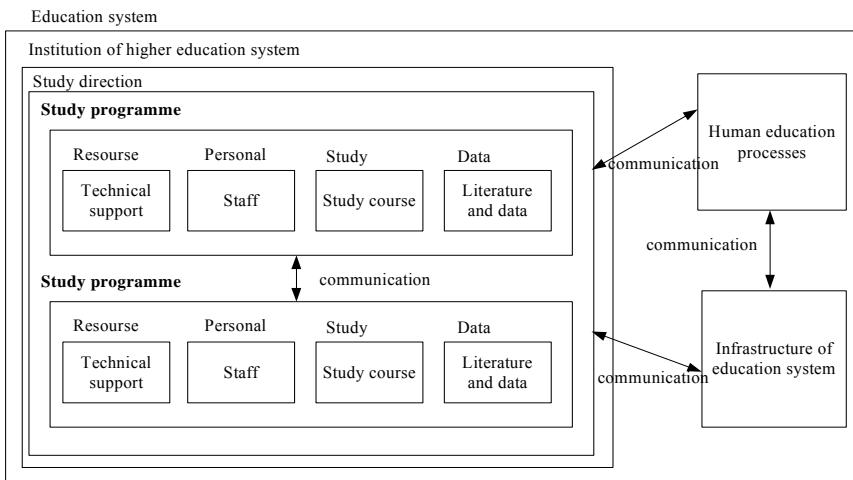


Figure 2. Study programme in the model of education system

Quality of a study programme is impacted both by its elements – resources, personnel, study process, and teaching aids; and the existing system of education institution and the state. Quality depends on and is impacted also by the infrastructure of education system and defined education processes. A certain quality model may be adopted for the system itself and every element of it, and imposing quality requirements one has to think how they impact and depend on the entire system quality.

Classification of software system quality requirements (ISO/IEC 25030, 2007) may be used as example for the classification of education system quality requirements. Viewed in such aspect, study programme quality requirements constitute a part of property requirements characteristic of study programme (Figure 3).

Requirements of higher education institution	Requirements of study direction	Requirements of study programme	Inherent property requirements of study programme	Functional requirements	
				Quality requirements of study programme	Quality in use requirements
					External quality requirements
			Internal quality requirements		
	Assigned property requirements of study	Requirements of study fee, study form, study support etc.			
	Requirements of study programme development	Requirements of the development process			
		Organisational requirements of development			
Other system requirements	Requirements of education system's infrastructure, literature, data, human business processes etc.				

Figure 3. **Classification of higher education institution system requirements**

Quality requirements are divided into internal, external, and use. In case of higher education institution system, quality in use is understood as the integration of the new specialists into the labour market and conformance to the employers' needs, thus, employers and graduates working in the respective sector are the main source of information for the mentioned requirements. Traditionally, the notion of external quality evaluation is also used speaking about the quality of higher education, here understanding the evaluation done by an external institution. The present PhD thesis attributes the notion of external quality also to study programmes. This quality by analogy with the external quality of software product can be defined during the implementation of study programme. External quality requirements for study programmes may be generally obtained from the current students and graduates as well as the legislative regulations. Recently, all decisions and agreements related with the establishment of the single European Higher Education Area have become especially significant. Internal quality requirements develop correspondingly from the requirements of quality in use and external quality, and they are mainly used by the structural units of education institutions during the development and updating of study courses and study programmes.

The author proposes the following variants for the evaluation of study programme quality:

- evaluation of external quality – questionnaires of students and employers;
- evaluation of internal quality – professional peer review evaluations.

The author proposes to use a quality model similar to software product quality model for the quality evaluation of study programme. It is possible to develop a new quality model by transforming quality models developed in software to the sphere of education. This will be a single model for the two products – software and study programme.

Analogy of software product with study programme

Software product and study programme are abstract products the quality of which cannot be evaluated visually and the most appropriate quality definitions for both products is conformity to the previously set requirements. Therefore, the definition of these quality requirements is a very important issue. A set of documents is prepared for the development of both products; later these documents are used also for the evaluation of product quality. The PhD thesis describes basic documents (specification of requirements, description of software design, software source code and description of study programme, curriculum, and descriptions of study courses) of both products and the similarity between the products.

The analysis of analogy between the study programme and software product is based on the recommendations given for the description of software design the schematic depiction of which shows the analogy (Figure 4) with the curriculum plan.

Design entity (symbols A, B, C, K1, K2,... in the figure) is the lowest level of detail in software design description. Design entity is a design element (component) that is structurally and functionally distinct from other elements. Each design entity is separately named and referenced during software implementation (IEEE Std 1016). Despite various natures of entities, they have some common characteristics. Every design entity shall have its name, purpose, and function. Attributes of design entities describe common characteristics of entities. An attribute of design entity is a characteristic or feature of design entity that has a name assigned to it. An attribute describes essential design information; yet, it does not describe information relevant to design process.

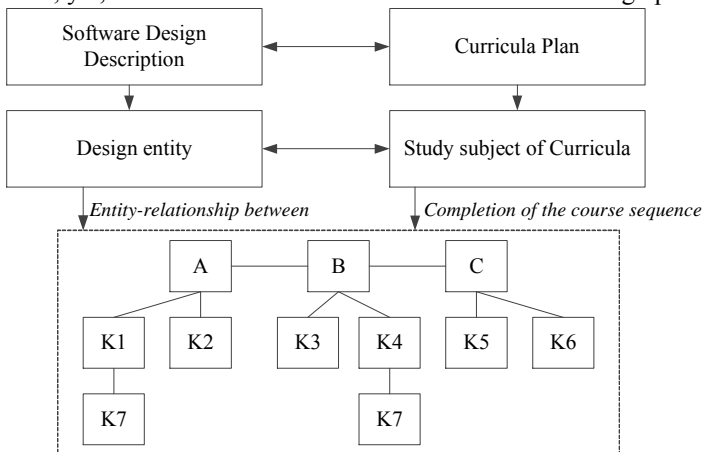


Figure 4. **Analogy between software design and curriculum plan**

In case of study programme, a study course and themes included into the course is an analogous element. All study courses of the same programme (symbols used in the figure K1, K2, ...) are arranged in the parts of plan (symbols used in the figure A, B and C) showing the sequence of their acquisition. Attributes of a study course describe essential features of the study programme; though, they do not describe the way of its implementation in the study process. Table 1 outlines the analogy between design entities and attributes of a study course.

Table 1. Design entities and attributes of study course

Attribute of design entity (LVS 72:1996)	Similar attribute of study course (LLU Studiju kursa nolikums...)
Identification – name of entity	Name of a study course and its code in the information system
Type – description of type of entity	Place of a study course in the study plan – Parts A, B or C, and type of control
Purpose – description of why an entity exists	Aim of a study course
Function – statement of what an entity does	Description of the content of a study course
Subordinates – identification of all those entities that compose the particular entity	Division of a study course into terms and forms of study work
Dependencies – description of relations between the particular entity and other entities	Relations of a course with other courses
Interface – description of interaction of other entities with the particular one	Preliminary knowledge required to acquire the course, necessity of the particular course for other courses
Resources – description of external elements of a design used by the particular entity	Division of lectures and practical work, technical resources necessary for the course
Processing – description of rules used by the particular entity for implementation of its functions	Requirements of a course for obtaining the credit points
Data – description of elements of external data of entity	Bibliography necessary for a course content, electronic materials of lectures

Software requirements specification and design description are taken as input item for measuring when evaluating internal quality characteristics of software product. Every design entity is evaluated during measuring to decide on the quality of entire software product. As a study course is the smallest unit of the study programme then the total evaluation of internal and external quality of the study programme can be obtained by evaluating every course separately.

The author proposes to use a single quality model in the study process both for the development and implementation of the study programme in information technologies. The justification is that specialists in information technologies during their studies should examine the impact of every such activity on the quality of software product parallel to acquiring methods and techniques for the development of software. In this case, individual aspects of software quality should be examined in details in all study courses relating with software development and introduction. The use of single quality model will help establish a common understanding both on the quality notion itself and imposition and evaluation of quality requirements.

4. SINGLE QUALITY MODEL

The single quality model is based on the software quality model ISO 9126 and supplemented with new quality subcharacteristics. The model has a two-level hierarchic structure (Figure 5):

Level 1 – basic quality characteristics;

Level 2 – sub-characteristics of quality characteristics that are divided into:

- basic subcharacteristics;
- additional subcharacteristics;
- optional subcharacteristics.

The type of relation for the quality characteristics and subcharacteristics in the model is 1:N, excluding optional subcharacteristics which have the type of relation n:m. This is explained by the fact that characteristics representative to several quality characteristics may be selected as optional subcharacteristics, for example, a subcharacteristic *readability* may be used both for the determination of usability and maintainability quality characteristics.

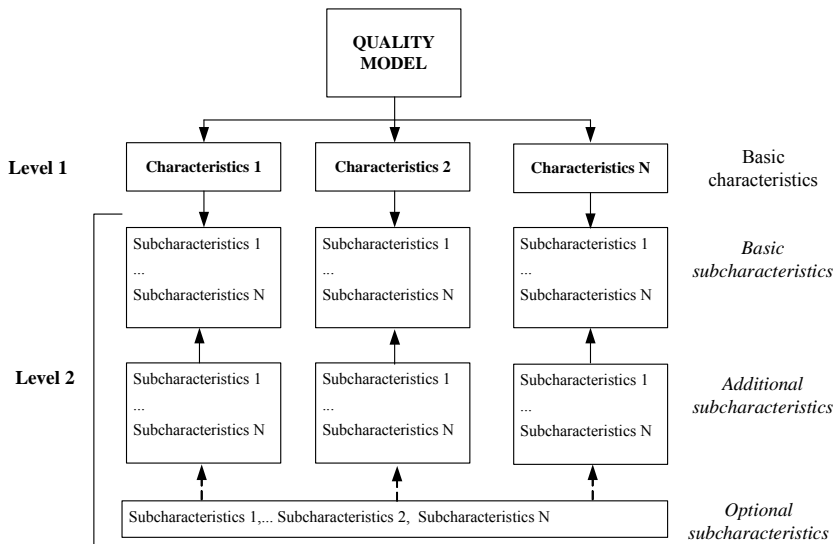


Figure 5. **Hierarchic structure of the single quality model**

The single quality model shall be adjusted to the context of usage and requirements of stakeholders involved in the quality evaluation in each individual case. In all cases, the adjustment shall be done consistent with the sequence of operations prescribed in the application procedure of the single quality model developed in the present PhD thesis (Figure 6).

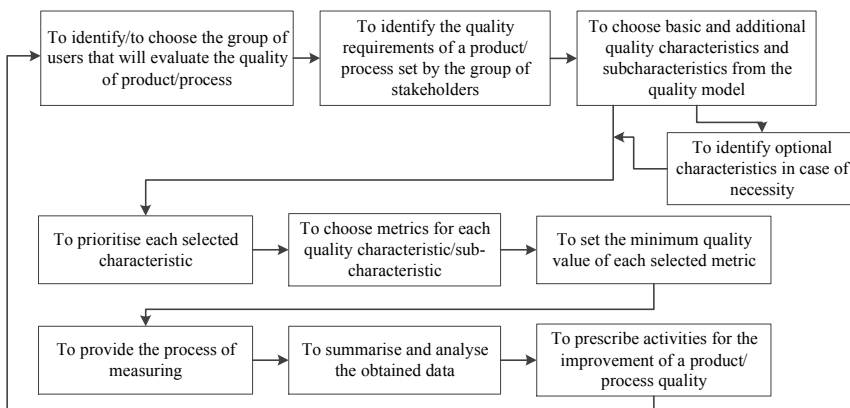


Figure 6. **Procedure for the application of the single quality model**

The single quality model is developed so that it may be used in the entire quality life cycle of software product. As the author included the study process in this cycle, the model was approbated during the PhD thesis development

process by evaluating the quality of study programme and demonstrating how to include the model into the content of study programme in information technologies. The model was approbated at Latvia University of Agriculture.

The author in her PhD thesis proposes a study programme quality model (Figure 7) which is obtained from the single quality model adjusting it to the application procedure of model. The context of model application is determined through the use of analogy with the notion of software internal and external quality described in Chapter 3.2 of the PhD thesis. The model encompasses 5 quality characteristics and 21 subcharacteristics. *Functionality, usability, efficiency, maintainability, and portability* were selected as the basic characteristics.

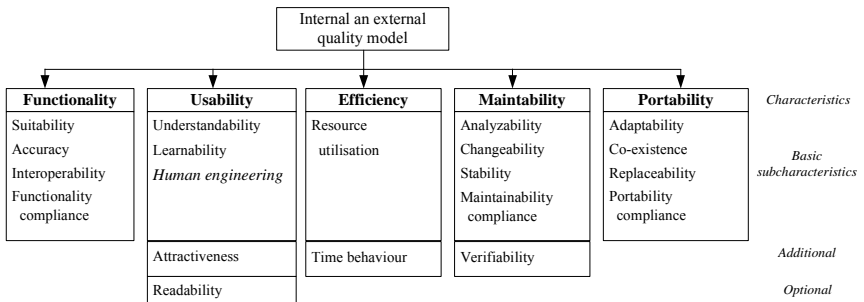


Figure 7. Quality model of a study programme

Explanation of characteristics and subcharacteristics of the single quality model in relation to the study programme:

- **Functionality** (is the necessary knowledge and skills included into the study programme)
 - *suitability* – is topical, conforms to the aims of study direction and labour market requirements;
 - *accuracy* – it is possible to acquire knowledge and skills included into the programme and they conform to the requirements of the qualification/degree to be awarded;
 - *interoperability* – learning outcomes of the programme support participation in exchange studies and ensure the necessary knowledge for further studies;
 - *functionality compliance* – the programme is prepared consistent with the legal and regulatory enactments, it is accredited.
- **Usability** (is the study programme easy to teach and to learn)
 - *understandability* – the curriculum plan is balanced and logically structured, the statement style conform to the level of preliminary knowledge;
 - *learnability* – acquisition of theoretical and practical knowledge of the programme is balanced, teaching aids are defined and available;

- *attractiveness* – oratory skills of the teaching staff are sufficient;
- *readability* – descriptive documents of the programme are prepared in good readable language consistent with the regulatory enactments;
- *human engineering* – degree up to which an individual university lecturer can impact the programme implementation.
- **Efficiency** (is the study programme efficient)
 - *time behaviour* – conformance of the programme volume to the time determined for the programme acquisition;
 - *resource utilisation* – resources necessary for the programme implementation.
- **Maintainability** (is it easy to maintain the study programme)
 - *analysability* – it is possible to analyse the curriculum plan and content;
 - *changeability* – it is possible to modify the study programme – without worsening its teachability;
 - *stability* – it is not necessary to update/modify the content of study programme during the academic year;
 - *verifiability* – it is possible to verify the achievement of study programme aim;
 - *maintainability compliance* – the necessary regulatory documents of the programme are prepared, quality evaluation of the programme is available.
- **Portability** (is it easy to adapt the study programme to another audience)
 - *co-existence* – the programme does not require specific preliminary knowledge;
 - *adaptability* – adjustment of the programme to other language or other cultural space does not require redevelopment of the programme;
 - *replaceability* – degree up to which learning outcomes of the programme may be achieved in another programme;
 - *portability compliance* – conformance to the regulatory documents governing students’ exchange programmes.

Concept for the inclusion of quality model into the content of study programme

Software development specialists are initially trained during studies when they acquire a study programme related with information technologies. Providing knowledge on software product development tools and techniques is the main task advanced by the study programmes. Development processes, their organisation, management, and quality assurance are viewed in parallel. Hence, the impact of action to be acquired on the quality of end product is not examined in every individual study course. The author’s experience both in the development of information technology systems and the implementation of study programmes in the field has allowed advancing the hypothesis that it

would be necessary to provide profound knowledge to future employees of information technology field on:

- evaluation of the quality of end product;
- impact of different solutions chosen and decisions made during the course of development left on the product quality;
- possibilities for quality improvement.

The content of study programme in information technologies shall be improved to implement these changes. The PhD thesis envisaged the selection of the development of mutual vulnerability matrix of quality characteristics between study courses and software, feedback on the evaluation of software end and intermediate products in IT companies, and the questionnaire of the teaching staff as the main source of information for the introduction of software product quality model to improve the content of study courses.

The author in her concept on the inclusion of quality model into the content of study programme identifies four stages.

1. Selection of study courses related with the development of software product from the undergraduate study programme in information technologies.
2. Division of the selected study courses into four stages of software product development – development of algorithms, development, coding, and testing of database.
3. Selection of characteristics to be taught in the respective study courses based on the author's acquired opinion of the corporate specialists and teaching staff on the characteristics of software product quality in the study process.
4. Supplementing the content of the present study courses with the respective quality characteristics emphasising both the activities increasing and decreasing the quality.

5. USE OF SINGLE QUALITY MODEL

The single quality model developed within the scope of the present PhD thesis was approbated for the internal and external quality evaluation of the study programmes.

Internal quality of study programme

Internal quality of the study programme is the content of study courses included into the programme. The author proposes to organise the evaluation of the internal quality of study programme by means of peer review evaluations. The group of peer review evaluation participants should consist of the teaching staff from the governing departments and representatives of the study programme graduates (current master or PhD students). Field specialists may also be involved in peer review evaluations. The group personnel may change as one study programme includes several dozen study courses.

The content evaluation of every study course shall be organised as on-site peer review evaluations. Each such evaluation should base on the following documents – study course programmes, study course application forms (if applicable), programmes of the current study courses (to evaluate the changes in the course content), and the curriculum plan. In parallel, electronic library catalogue of the higher education institution should be used for the evaluation of internal quality to judge on the availability of literature used in references of the study course.

The author proposes to use a study course content quality evaluation procedure consisting of six steps for the internal quality evaluation of study programme (Figure 8).

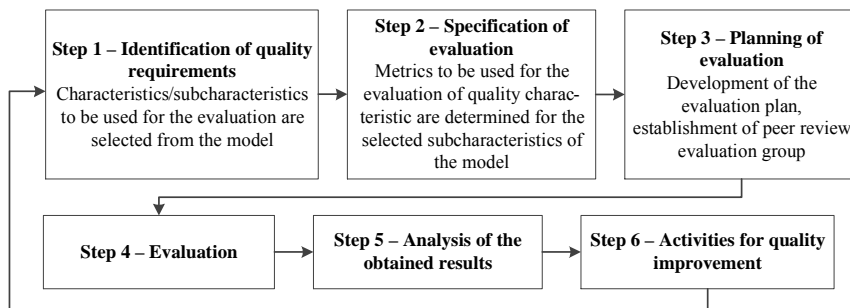


Figure 8. **Procedure for the internal quality evaluation of study programme**

Step 1 – Identification of quality requirements

Those characteristics and subcharacteristics that will be used for the particular evaluation should be selected from the quality model. The characteristics used may differ depending on the evaluation time of the study course – whether it is the first time or a repeated evaluation. It is recommended to use all characteristics in the first or initial evaluation of a study course to obtain a complete quality evaluation of the content. Specific subcharacteristics necessary for the evaluation may be selected when organising repeated peer review evaluations of study courses.

Step 2 – Specification of evaluation

Selection of metrics used for the evaluation of every subcharacteristic follow the selection of quality characteristics/subcharacteristics. The number of metrics in each evaluation may differ depending on the quality characteristics selected in Step 1. The present PhD thesis provides the possible metrics for the evaluation of internal quality characteristics of a study course, description which questions the respective metric answers and the data item to be used for the quality evaluation.

Step 3 – Planning of evaluation

The internal quality evaluation shall be planned prior to the beginning of the study programme evaluation:

- selection of courses for the evaluation. It is recommended to evaluate all study courses if it is the first time evaluation of the study programme, individual study courses may be selected for the repeated evaluation;
- drafting of the time schedule for peer review evaluations of study courses;
- formation of peer review evaluation participant groups;
- drawing up a checklist for internal quality evaluation of a study course. The checklist for peer review evaluation participants may differ depending on the evaluation time of the study course – whether it is the first time or a repeated evaluation. The PhD thesis covers a complete checklist for a study course;
- dispatching of documents to the participants of peer review evaluation necessary for the evaluation of a study course (items to be evaluated) and the checklist for the course evaluation;
- information of the participants of peer review evaluation on the on-site meeting and organisation of the study course evaluation.

Step 4 – Evaluation

The internal quality of every study course shall be evaluated in the form of on-site meetings of the peer review evaluation group. Prior to the on-site meeting, the review participants evaluate the content of the respective study course by means of the checklist. The checklist ensures guidelines for the review participants and possibility to obtain information easier to analyse and compare quality characteristics being of interest. The on-site meeting of the peer review evaluation participants is aimed at discussion of study course content, thus, providing recommendations for its improvement.

Step 5 – Analysis of the obtained results

The checklist does not require electronic data processing since the quality evaluation of the study course content is qualitative and there is no need to express it in figures. The evaluation essence is to understand aspects taught in every study course and their compliance with the aims and learning outcomes set for the study programme, their harmonisation with other courses of the study programme; and to give recommendations for the improvement of the study course content. Checklist filled in by the participants of peer review evaluations provides additional information to a university lecturer of the particular study course on the necessary improvements in the study course content.

Step 6 – Activities for quality improvement

The reviews provide recommendations to the author (university lecturer) of the study course on improvements in the study course content, if necessary. The course review may be repeated after a certain time period if the course content requires essential improvements.

Every new study course after the development of the study programme and prior to its implementation should undergo the evaluation of its internal quality;

all courses of the study programme should undergo a repeated evaluation at least twice within the study programme accreditation period.

External quality of study programme

External quality of study programme can be evaluated after its implementation in the study process and achievement of learning outcomes. Students acquiring the respective study programme are the reviewers of the study programme external quality.

The author proposes to use the procedure consisting of four steps for the external quality evaluation of study programme (Figure 9).

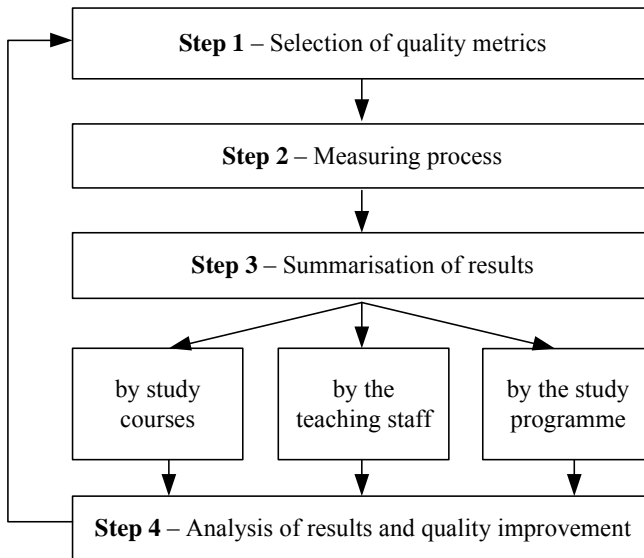


Figure 9. Procedure for the external quality evaluation of study programme

Step 1 – Selection of quality metrics

The PhD thesis provides metrics for the external quality evaluation of study programme by means of questionnaires administered to students on every mastered study course. The author would recommend including not more than 10 quality subcharacteristics in one evaluation period as any student has to evaluate 5-10 study courses per term on average (number of the acquired study courses per term).

Step 2 – Measuring process

The survey method is applicable for obtaining qualitative data. The assessment questionnaire of study course should be made as matrix type questionnaire additionally including one open type question. The author proposes to use a Likert scale with the range from 1 to 5: very low, low,

average, high, very high. There is no need to organise special sample groups of respondents as the number of students per one study programme ranges between 50 and 300 on average, and the quality evaluation is aimed at obtaining the evaluation of every study course. As students are the reviewers of study courses and their number may fluctuate between a few hundred and several thousand per each evaluation time (depending on the size of a higher education institution), it is recommended to use the information system of higher education institution or any other survey tool.

Step 3 – Summarisation of results

Several stakeholders are interested in the results of external quality evaluation of study programme: students, teaching staff, directors of study programmes, and the university management. Therefore, the author offers summarising the results of study course evaluation in three sectional views by:

1. *study course* – collection of data at the level of metrics;
2. *teaching staff* – collection of data on each study course taught by an individual lecturer;
3. *study programme* – collection of data at the level of characteristics, subcharacteristics and metrics of the quality model.

Step 4 – Analysis of results and quality improvement

Results of study programme quality evaluation in three sectional views will ensure the feedback at different levels. The analysis of quality evaluation results both by the evaluation done after each term and aggregated for a longer period will facilitate teaching staff, directors of study programmes, and the management of higher education institution to develop quality improvement measures.

The approbation of model encompassed the selection of three quality characteristics and nine subcharacteristics (Table 3) from the single quality model for the evaluation of external quality.

Table 3. **Metrics selected for the external quality evaluation of study programme**

Quality characteristic/sub-characteristic	Metric
Functionality/ Suitability	Does the delivered study course conform to the course description?
Functionality/ Suitability	Has useful knowledge, skills and competences been acquired?
Functionality / Accuracy	Is a lecturer competent in his/her study course?
Functionality/ Interoperability	Does the study course overlap with the content of another study course?
Usability/ Understandability	Does a lecturer explain themes of the study course understandably?
Usability / Understandability	Does a lecturer introduce students with the requirements for mastering the study course at the beginning of the study course?

Table 3. continued

Quality characteristic/sub-characteristic	Metric
Usability / Learnability	Do teaching methods and approaches used by a lecturer promote mastering of the study course?
Usability / Attractiveness	Oratorical skills of a lecturer
Usability / Human engineering	Are lecturer's tutorials available?
Efficiency/ Time behaviour	Do tests encourage mastering of the study course?
Efficiency/ Resource utilisation	Are study materials available and do they facilitate mastering of the study course?

The evaluation scale of each metric ranges between 1 and 5 (1–very low, 2–low, 3–average, 4–high, 5–very high). Students have to evaluate every study course mastered during the term in each evaluation process consistent with the metrics included into Table 3 as well as they have the opportunity to write freely recommendations for the improvement of each study course.

Evaluating and summarising data on all the study courses of one study programme allows obtaining general evaluation on the external quality of the study programme consistent with the selected characteristics, subcharacteristics, and metrics of the single quality model. The author offers applying a single method for collection of such data and an outline for interpretation of values obtained during evaluation (Table 4).

Table 4. **Outline for data interpretation of a metric**

Characteristic	Interpretation
Purpose of the metrics	Does the content of study course conform to the course description?
Method of application	Count the number of negative estimations (1, 2, 3) given in the questionnaire and compare it with the total number of estimations
Formula and data element computation	$X=1-A/B$ A – number of negative estimations B – total number of estimations
Interpretation of value	$0 \leq X \leq 1$ The closer to 1.0, the better
Metric scale type	Absolute (1-very low, 2-low, 3–average, 4–high, 5-very high)
Measure type	A, B – count X – count/ count

This type of data interpreting is transformed from the field of software product development. The software product quality model at the level of the ISO 9126 standards (ISO 9126-2, ISO 9126-3) proposes metrics of quality characteristics and type for interpreting the results obtained during evaluation.

The author offers applying the following formula for computing the values of study programme quality characteristics:

$$KRV_i = 1 - \frac{A}{B}, \quad (1)$$

where KRV_i – total estimation of quality characteristic /subcharacteristic/ metric;
A – total number of negative estimations;
B – total number of estimations.

If each course of the study programme is evaluated by means of the scale from 1 to 5, then the estimations 1, 2 and 3 are considered as negative ones. Total estimation of a quality characteristic, subcharacteristic or metric will always range between 0 and 1. The closer the estimation is to the figure 1, the higher is the quality. The author proposes to consider the evaluation tendency using such variant for interpreting of the results. If the estimation is closer to 0.5 or below it, one shall think of quality improvement measures according to the respective quality characteristics, subcharacteristics, or metrics.

As the author of the present PhD thesis proposes to summarise the data on external quality evaluation of study programme in three sectional views, their visual representation shall also be provided in three sectional views. A decision to represent the evaluation data in two ways was made during the approbation of the single quality model for the evaluation of study programmes at Latvia University of Agriculture:

1. information system of the University – in the view of a study course and a lecturer;
2. new application – in the view of a study programme.

The decision is based on the fact that the sectional view by study courses and lecturers is directly meant for individual lecturers and the management of higher education institution; it is not meant for public inspection. Data in these sectional views shall be collected at the level of every metric to help university lecturers identify the weak points of course quality and undertake respective improvement measures. The quality of study programme will be improved in general through the improvement of every individual study course. Data summarisation by the sectional view of study programme, in turn, is done consistent with the characteristics of the single quality model and this summary is meant for all the students and lecturers of the university to get acquainted with the external quality of study programme implemented by individual higher education institution. Therefore, new application software (quality evaluation of study programme) aimed at providing a possibility to any student, lecturer and management of Latvia University of Agriculture to get acquainted with the study programme quality model and to obtain data on the total external quality evaluation of the University study programmes was developed within the framework of the PhD thesis.

The users of the study programme quality evaluation software are offered the possibility to view the quality evaluation of a study programme in the form of graphics by each characteristic and subcharacteristic selected from the single quality model. The study programme quality evaluation software also provides representation of metric data since the application of quality model for the evaluation of study programme is a new approach and descriptions of quality characteristics/subcharacteristics in relation to study programmes are new.

Data representation charts (Figure 10) represent the following aspects for every quality characteristic/subcharacteristic/metric of study programme:

- minimum estimation obtained during the evaluation;
- maximum estimation obtained during the evaluation;
- outcome of the previous evaluation;
- outcome of the latest evaluation;
- comparison of the latest and previous evaluation – increase or decrease. This comparison is seen also visually, if the bar in the chart is coloured, the quality evaluation increases, if the bar is blank, the quality evaluation decreases.

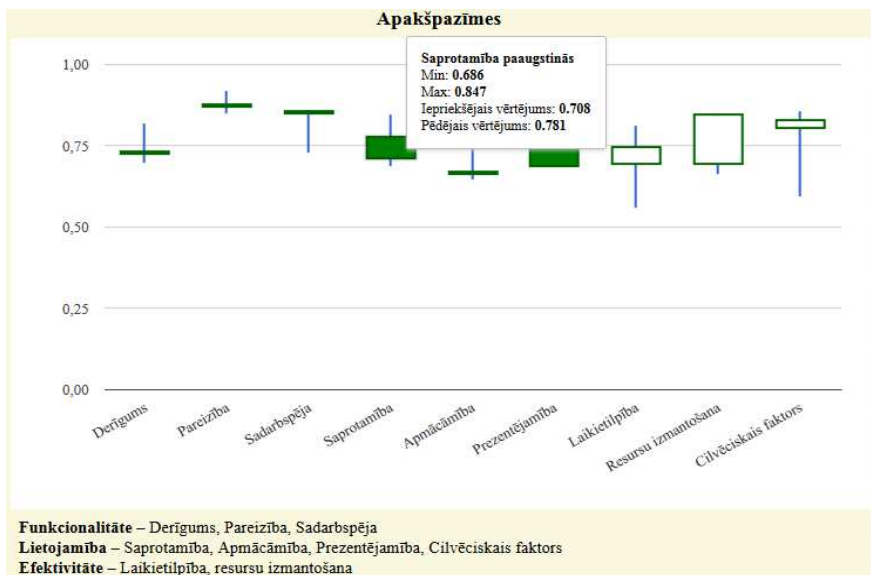


Figure 10. Summary of study programme quality subcharacteristics

Initiating quality improvement measures, it is significant to understand the scope of improvements – shall it be done by all quality characteristics? Therefore, analysing the data obtained on the external quality evaluation of study programmes in six terms, the author wanted to clarify the existence of correlation between quality characteristics and subcharacteristics. The

Spearman rank correlation (SRK) coefficient measuring correlation between variables of rank was used to denote the correlation.

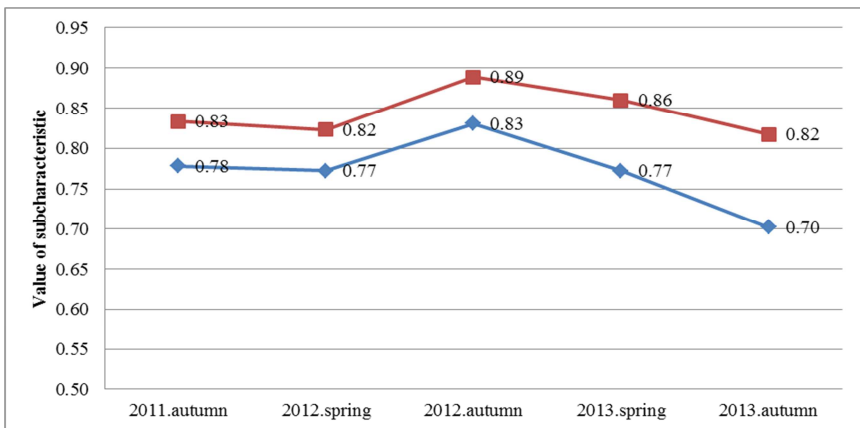
Data analysis showed a close and very close data correlation among all the three quality characteristics – *functionality*, *usability*, and *efficiency* (Table 5). Bold in the table marks coefficients with very close correlation.

Table 5. **Correlation coefficients of quality characteristics**

Characteristic	Functionality	Usability	Efficiency
Functionality	1		
Usability	0.831*	1	
Efficiency	0.746*	0.709*	1

* p=0.01, n=94

If the quality is increased by *functionality* characteristic in quality improvement measures, its improvement by *usability* characteristic may be expected and vice versa (Figure 11).



(Autumn 2011, Spring 2012, Autumn 2012, Spring 2013, Autumn 2013)

Figure 11. **Quality evaluation of study programme according to the characteristics *functionality* and *usability* by study terms**

The existence of correlation among subcharacteristics of one quality characteristics and among subcharacteristics of several characteristics was detected at the level of subcharacteristics. Correlations with the confidence of 99% (p=0.01) exist among all quality subcharacteristics. The correlation of subcharacteristics within the scope of one characteristic demonstrated a **very close** correlation between:

- functionality – *accuracy* and *suitability* (SRK=0.81). Conformance of the study programme course content to the defined aims and tasks impacts the achievement of the defined learning outcomes;
- usability – *learnability* and *attractiveness* (SRK=0.835). Methods used in the implementation of the study programme courses impact understandability of the delivered course and interest in it;
- efficiency – *time behaviour* and *resource engineering* (SRK=0.849). Materials available for the acquisition of the study programme content impact the time sufficiency for the course acquisition.

Quality improvement and assurance in an education institution undeniably determine the introduction of new control mechanisms and transformation of the existing ones. This policy will be efficient only if it encompasses a complex development programme including a set of the following elements: adjustments in the present legislation, creation of new stimuli, and gaining of new experience. The policy of quality improvement as strategic function includes also active participation of the teaching staff and students in the level activities improving and assuring quality. Study process is not only individual work of every and each but it is a team work, and only joint team work may assure quality. In this respect, the study process has also a close analogy with software development since nowadays software product is the outcome of joint team work.

CONCLUSIONS

1. Quality of software product is not an unequivocal notion and it will always depend on the viewpoint it is looked upon. Setting of certain quality requirements allows discussion and evaluation of quality. Therefore, product quality characteristics and their correlation should be viewed individually in every particular situation.
2. Development of quality software is always related with additional costs. Various contributions to preventive measures should be done to reduce quality costs in future. Training of personnel is one of the activities started already during the studies. The emphasis in training of the new specialists in information technologies should be placed from the aspect how to develop software product on the aspect how to develop quality software product.
3. Software quality models are significant for definition and evaluation of software product quality as they establish a single view on product quality – in the form of hierarchic quality characteristics.
4. Diversity of quality models and development of new models demonstrate that their authors try to offer intuitively more understandable quality characteristics and subcharacteristics to the developers and users of

- models. The only changes being in the names, as they are equal by sense, for example, the ISO 9126 model has a subcharacteristic *failure tolerance*, while the FURPS model has *failure frequency and severity*, or the ISO 9126 model uses *changeability*, while the Boehm's model – *augmentability* or the FURPS model – *extensibility*.
5. The single quality model ensures a single approach for the definition and evaluation of quality requirements in the development and studies of information technologies.
 6. The single quality model may be used for the evaluation of a study programme quality as both software and study programme are abstract products that cannot be physically weighted, measured, or somehow differently visually evaluated.
 7. A higher education institution applying the single quality model should evaluate the quality of a study programme not only during its accreditation, i.e. once in 2 or 6 years but regularly and the measures for improvement of internal and external quality should be undertaken basing on the obtained quality estimations.
 8. Peer review evaluations may be used to evaluate the internal quality of study programme, while students should be involved in the external evaluation of study programme as they are directly involved in the study process.
 9. The 99% confidence shows a correlation between characteristics and subcharacteristics in the external quality model of study programme. Very close correlation exists between the following subcharacteristics:
 - functionality characteristic – *accuracy* and *suitability* (SRK=0.81). Conformance of the study programme course content to the defined aims and tasks impacts the achievement of the defined learning outcomes;
 - usability characteristic – *learnability* and *attractiveness* (SRK=0.835). Methods used in the implementation of the study programme courses impact understandability of the delivered course and interest in it;
 - efficiency characteristic – *time behaviour* and *resource engineering* (SRK=0.849). Materials available for the acquisition of the study programme content impact the time sufficiency for the course acquisition;
 - functionality, efficiency characteristic – *suitability* and *time behaviour* (SRK=0.84). Learning outcomes defined in the study programme impact the time sufficiency for the course acquisition.
 10. Interrelation among quality characteristics of study programmes may be applied for undertaking quality improvement measures. Improvement of quality according to one characteristic would also improve other characteristics with the 99% confidence.
 11. The correlation between software quality characteristics and study courses will provide students with knowledge what and how can influence software

quality. The students will master this knowledge during their studies acquiring specific study courses. Therefore, preconditions will be established for the development of quality software products in future.

Proposals for further studies

1. Future studies could be devoted to the analysis of benefits from the introduction of peer review evaluations for the internal quality evaluation of study programme.
2. Further studies could deal with the verification and evaluation of the Spearman rank correlation between characteristics and subcharacteristics existing in the external quality model of study programme to apply it for undertaking quality improvement measures.
3. To adjust the single quality model for the evaluation of process quality.
4. Further studies should analyse benefits from the author's developed concept on the inclusion of quality model into the content of study programme in information technologies.
5. To offer the developed single quality model to other higher education institutions for the quality evaluation of their study programmes.

LITERATŪRAS SARAKSTS

BIBLIOGRAPHY

1. Akadēmiskā terminu datubāze *AkadTerm* [tiešsaiste] [25.11.2013] <http://termini.lza.lv/term.php>.
2. Al-Turki, U., & Duffuaa, S. (2003) Performance measures for academic departments. *International Journal of Educational Management*, 17(7), pp. 330–338.
3. Al-Qutaish, R. (2009) Measuring the Software Product Quality during the Software Development Life-Cycle: An International Organization for Standardization Standards Perspective. *Journal of Computer Science*, 5(5), pp. 392–397.
4. Al-Qutaish, R. (2010) Quality Models in Software Engineering Literature: An Analytical and Comparative Study. *Journal of American Science*, 6(3), pp. 166–175.
5. Anderson, G. (2006) Assuring Quality/Resisting Quality Assurance: Academics' Responses to Quality in Some Australian Universities. *Quality in Higher Education*, 12(2), pp. 161–173.
6. Anthes, G. (2004) *Quality Model Mania* [online] [01.09.2013]. Available at: http://www.computerworld.com/s/article/90797/Model_Maniam.

7. Arhipova, I. & Bāliņa, S. (2006) Statistika ekonomikā un biznesā: risinājumi ar SPSS un MS Excel. Rīga: Datorzinību centrs, 362 lpp.
8. Ariff, M., Sulong, N., Khalifah, Z & Omar, N. (2008) *Designing best practices for teaching and learning using a quality management system framework: the ISO 9001:2000 approaches* [online] [01.03.2014]. Available at: <http://www.aair.org.au/app/webroot/media/pdf/AAIR%20Fora/Forum2002/Sulong.pdf>.
9. Avdjieva, M. & Wilson, M. (2002) Exploring the development of quality in higher education. *Managing Service Quality*, 12(6), pp. 372–383.
10. Azuma, M. (2001) SQuaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality. In *ESCOM (European Software Control and Metrics Conference)*, April 2001, London, pp. 337–346.
11. Badri, M. & Abdulla, M. (2004) Awards of excellence in institutions of higher education: an AHP approach. *International Journal of Educational Management*, 18(4), pp. 224–242.
12. Badri, M., Selim, H., Alshare, K., Grandon, E., Younis, H. & Abdulla, M. (2006) The Baldrige education criteria for performance excellence framework: empirical test and validation. *International Journal of Quality & Reliability Management*, 23(9), pp. 1118–1157.
13. Bagert, D., et al. (1999) Guidelines for Software Engineering Education, Version 1.0, Technical Report CMUISEI-99-TR-032, Software Engineering Institute, Carnegie Mellon University, p. 74.
14. Baldassarre, M., Caivano, D., Pino, F., Piattini, M., Visaggio, G. (2012) Harmonization of ISO/IEC 9001:2000 and CMMI-DEV: from a theoretical comparison to a real case application. *Software Quality Journal*, 20 (2), pp. 309–335.
15. Behkamal, B., Kahani, M., & Akbari, M. K. (2009) Customizing ISO 9126 quality model for evaluation of B2B applications. *Information and Software Technology*, 51(3), pp. 599–609.
16. Bawane, N. & Srikrishna, C. (2010) A Novel Method for Quantitative Assessment of Software Quality. *International Journal of Computer Science and Security*, 3(6), pp. 508–517.
17. Becket, N. & Brookes, M. (2006) Evaluating quality management in university departments. *Quality Assurance in Education*, 14(2), pp. 123–142.
18. Becket, N. & Brookes, M. (2008) Quality Management Practice in Higher Education – What Quality are We Actually Enhancing? *Journal of Hospitality, Leisure, Sport & Tourism Education*, 7(1), pp. 40–54.
19. Berander, P. et al. (2005) Software quality attributes and trade-offs. Sweden, Blekinge Institute of Technology, 2005, p. 100.
20. Biggs, J., Kember, D. & Leung, D. Y. (2001) The revised two-factor study process questionnaire: R-SPQ-2F. *British Journal of Educational Psychology*, 71(1), pp. 133–149.

21. Brochado, A. (2009) Comparing alternative instruments to measure service quality in higher education. *Quality Assurance in Education*, 17(2), pp. 174–190.
22. Brookes, M. & Becket, N. (2007) Quality management in higher education: a review of international issues and practice. *International Journal of Quality Standards*, 1(1), pp. 85–121.
23. Boehm, W., Brown, R. & Lipow, M. (1976) Quantitative Evaluation of Software Quality, International Conference on Software Engineering. *Proceedings of the 2nd international conference on Software engineering*, pp. 592–605.
24. Boehm, B., Brown, J., Kaspar, H., Lipow, M., McLeod, G. & Merritt, M. (1978) *Characteristics of Software Quality*. North Holland.
25. Boehm, B. & Basili, V. (2001) Software Defect Reduction Top 10 List, IEEE Computer, IEEE Computer Society, *Software Management*, pp. 135–137.
26. Boehm, B. & Jain, A. (2006) An initial theory of value-based software engineering. *Value-Based Software Engineering*, Springer, Berlin, Heidelberg, pp. 15–37.
27. *Boloņas process un kvalifikāciju atzīšana*. [tiešsaiste] [05.03.2014.]. Pieejams: www.aic.lv/bolona/Latvija/Atsev_prez/recbol_lv.pdf.
28. Borzovs, J., Viļums, Ē., Čevere, R., Plūme, J. (1997) Ieteikumi programmatūras dokumentācijas komplektam. Rīga, RITI.
29. Borahan, N. & Ziarati, R. (2002) Developing Quality Criteria for Application in the Higher Education Sector in Turkey. *Total Quality Management*, 13(7), pp. 913–926.
30. Cangussu, J., DeCarlo, R. & Mathur, A. (2002) A formal model of the software test process. *IEEE Transactions on Software Engineering*, 28 (8), pp. 782–796.
31. Carroll, J. (1995) The Application of Total Quality Management to Software Development, Information Technology and People. *MCB University Press*, 4, pp 35–47.
32. Cevere, R. & Sproge, S. (2013) Introduction of Software Product and Process Quality Aspects in the Study Courses of Information Technologies. *Proceedings of the 6th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), pp. 267–275.
33. Cheng, Y. & Tam, W. (1997) Multi-models of quality in education. *Quality Assurance in Education*, 5(1), pp. 22–31.
34. Chua, B. & Dyson, L. (2004) *Applying the ISO 9126 model to the evaluation of an elearning system*. Proceedings of ASCILITE, pp. 5–8.
35. Clewes, D. (2003) A student-centred conceptual model of service quality in higher education. *Quality in Higher Education*, 9(1), pp. 69–85.
36. Coates, H. (2005) The Value of Student Engagement for Higher Education Quality Assurance. *Quality in Higher Education*, 11(1), pp. 25–36.

37. Crosby, P. (1979) *Quality is free: the art of making quality certain*. New York: McGraw–Hill.
38. Cruickshank, M. (2003) Total quality management in the higher education sector: a literature review from an international and Australian perspective. *Total Quality Management and Business Excellence*, 14(10), pp. 1159–1167.
39. Curtis, B., Krasner, H. & Iscoe, N. (1988) A field study of the software design process for large systems. *Communications of the ACM*, 31(11), pp. 1268–1287.
40. Cevere, R. & Sproge, S. (2010) Application of Software Quality Models in Evaluation of Study Quality. *Problems of Education in the 21st Century*, 21, pp. 37–48.
41. Cevere, R. & Sproge S. (2011) Methodology for Evaluation of Internal Quality of the Study Programme. EQANIE Conference Vienna 2011. [online] [14.03.2014.]. Available at: <http://www.eqanie.eu/pages/events/past-events/conference-vienna-2011/proceedings.php>.
42. Čevere, R., Sproģe, S. & Arhipova, I. (2008) How to Use an Experience of the Development of ISO 9001 Certified Quality System in Study Process Improvement. In: *Proceedings of the 3rd International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), pp. 126–136.
43. Delors, J. (1996) *Learning: the Treasure within Report to UNESCO of the International Commission on Education for the Twenty-first Century*. Paris: United Nations Educational, Scientific and Cultural Organization.
44. Deming W. (1986) *Out of the crisis: quality, productivity and competitive position*. Massachusetts Inst Technology, p. 373.
45. Deming, W. (2000) *The New Economics for Industry, Government, Education*. The MIT Press; 2nd edition, p. 266.
46. Denning, P. (1992) Educating a New Engineer. *Communications of the ACM*, 35(12), pp 83–97.
47. Douglas, J., Douglas, A., & Barnes, B. (2006) Measuring student satisfaction at a UK university. *Quality Assurance in Education*, 14(3), pp. 251–267.
48. Dromey, R. (1995) A model for software product quality. *IEEE Transactions on Software Engineering*, 21, pp. 146–162.
49. Dubey, S., Ghosh, S. & Rana, A. (2012) Comparison of software Quality models: An analytical approach. *International Journal of Emerging Technology and Advanced Engineering*, 2(2), pp. 111–119.
50. *Education Criteria for Performance Excellence 2011–2012*. [online] [14.11.2013.]. Available at: http://www.nist.gov/baldrige/publications/upload/2011_2012_Education_Criteria.pdf.
51. Eickelmann, N. (2003) An Insider's View of CMM Level 5. *IEEE Software*, 20(4), pp. 9–81.

52. Elbaum, S., Rothermel, G., Kanduri, S. & Malishevsky, A. (2004) Selecting a cost-effective test case prioritization technique. *Software Quality Journal*, 12 (3), pp. 185–210.
53. Esaki, K. (2013) Verification of Requirement Analysis Method for System Based on ISO/IEC 9126 Six Quality Characteristics. In *Trustworthy Computing and Services*, Springer Berlin Heidelberg, pp. 60–68.
54. *EURO-INF Framework standards and accreditation criteria for Informatics programmes. Version II, 2007*, [online] [09.03.2011]. Available at: www.kbs.cs.tu-berlin.de/ecss/docs/euro-inf.pdf.
55. Fam, M., Luo, Y., Wu, G. & Fu, X. (2010) An Improved Analytic Hierarchy Process Model on Software Quality Evaluation. In *Proceedings of the 2nd International Conference Information Science and Engineering (ICISE)*. Hangzhou, China, pp. 1838–1842.
56. Firesmith, D. (2005) Quality Requirements Checklist. *Journal of Object Technology*, 4(9), pp. 31–38.
57. Garvin, A. (1987) *Competing on the eight dimensions of Quality*. Harvard Business Review Nov-Dec., pp 101–109.
58. Ginns, P., & Ellis, R. (2007) Quality in blended learning: Exploring the relationships between on-line and face-to-face teaching and learning. *The Internet and Higher Education*, 10(1), pp. 53–64.
59. Gibbs, W. (1994) Software's Chronic Crisis. *Scientific American*, 271(3), pp. 86–95.
60. Glass, R. (1998) Defining quality intuitively. *IEEE Software*, pp. 103–104.
61. Grendel, T. & Rosenbusch, C. (2010) System accreditation: an innovative approach to assure and develop the quality of study programmes in Germany. *Higher Education Management and Policy*, 22(1), pp. 87–98.
62. *Guide to the Software Engineering Body of Knowledge (SWEBOK) (2004)*. [online] [14.11.2011]. Available at: <http://www.computer.org/portal/web/swebok/htmlformat>.
63. Gvaramadze, I. (2008) From Quality Assurance to Quality Enhancement in the European Higher Education Area. *European Journal of Education*, 43(4), pp. 443–455.
64. Harvey, L., Green, D. & Burrows, A. (1993) Assessing quality in higher education: A transbinary research project. *Assessment & Evaluation in Higher Education*, 18(2), pp. 143–148.
65. Hartleya, R. & Virkusb, S. (2003) Approaches to quality assurance and accreditation of LIS programmes: Experiences from Estonia and United Kingdom. *Education for Information*, 21, pp. 31–48
66. Harvey, L. (2005) A history and critique of quality evaluation in the UK. *Quality Assurance in Education*, 13(4), pp. 263–276.
67. Herbsleb, J., Carleton, A., Rozum, J., Siegel, J. & Zubrow, D. (1994) *Benefits of CMM-based software process improvement: initial results* Technical Report, CMU/SEI-94-TR-13. ESC-TR-94-013, Pittsburgh, p. 64.

68. Houstona, D. & Keatsa, B. (1998) Cost of software quality: a means of promoting software process improvement. *Quality Engineering*, 10(3), pp. 563–573.
69. Hilburn, T. & Towhidnejad, M. (2000) Software Quality: A Curriculum Postscript? In: *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*. New York, pp. 167–171.
70. Hoecht, A. (2006) Quality Assurance in UK Higher Education: Issues of Trust, Control, Professional Autonomy, and Accountability. *Higher Education*, 51(4), pp. 541–563.
71. Hoyer, R. & Hoyer, B. (2001) What is quality. *Quality Progress*, 7, pp. 52–62.
72. Humphrey, W. (1989) *Managing the Software Process*. Addison-Wesley Professional, p. 512.
73. IEEE Std 610.12–1990 IEEE Standard Glossary of Software Engineering Terminology. IEEE Standards Association.
74. IEEE Std 830–1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Standards Association
75. IEEE Std 1074–1997 Standard for Software Life Cycle Processes. IEEE Standards Association.
76. IEEE Std 1016–2009 IEEE Standard for Information Technology – Systems Design – Software Design Descriptions. IEEE Standards Association.
77. Ishikawa, K. (1985) *What is total quality control? The Japanese way*. Englewood Cliffs, NJ: Prentice-Hall, p. 215.
78. ISO 9000 – Quality management. [online] [11.05.2013]. Available at: http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm
79. ISO 8402:1986 International Standard: Quality Vocabulary. International Organisation for Standardization.
80. ISO 9001:2008 International Standard: Quality Management Systems – Requirements. International Organization for Standardization.
81. ISO/IEC 12207:2008 Information technology – Software life cycle processes. International Organization for Standardization.
82. ISO/IEC 14598-2: 2000 Software engineering – Product evaluation – Part 2: Planning and Management. International Organization for Standardization.
83. ISO/IEC 15939:2007 Systems and software engineering – Measurement process. International Organization for Standardization.
84. 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.
85. ISO/IEC 25030:2007 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality Requirements. International Organization for Standardization.

86. ISO/IEC 9126-1:2001 Software Engineering – Product Quality – Part 1: Quality Model. International Organization for Standardization.
87. ISO/IEC 9126-2:2003 Software Engineering – Product Quality – Part 2: External Metrics. International Organization for Standardization.
88. ISO/IEC 9126-3:2003 Software Engineering – Product Quality – Part 3: Internal Metrics. International Organization for Standardization.
89. ISO/IEC 9126-4:2004 Software engineering – Product quality – Part 4: Quality in use metrics. International Organization for Standardization.
90. Jiang, J., Klein, G., Hwang, H., Huang, J. & Hung, S. (2004) An exploration of the relationship between software development process maturity and project performance. *Information & Management*, 41(3), pp. 279–288.
91. Jørgensen, M. (1999) Software quality measurement. *Advances in Engineering Software*, 30, pp. 907–912.
92. Jung, H., Kim, S. & Chung, C. (2004) Measuring software product quality: A survey of ISO/IEC 9126. *Software, IEEE*, 21(5), pp. 88–92.
93. Jung, H. (2007) Validating the external quality subcharacteristics of software products according to ISO/IEC 9126. *Computer Standards & Interfaces*, 29(6), pp. 653–661.
94. Juran J. (1988) *Juran's Quality Control Handbook*. McGraw–Hill.
95. Kannabiran, G., & Sankaran, K. (2011) Determinants of software quality in offshore development – An empirical study of an Indian vendor. *Information and Software Technology*, 53(11), pp. 1199–1208.
96. Karg, L., Grottke, M. & Bechausb, A (2011) A systematic literature review of software quality cost research. *The Journal of Systems and Software*, 84, pp. 415–427.
97. Kember, D., Leung, D. & Kwan, K. (2002) Does the use of student feedback questionnaires improve the overall quality of teaching? *Assessment & Evaluation in Higher Education*, 27(5), pp. 411–425.
98. Khayami R., Towhidi A. & Ziarati K. (2009) *The Analytical Comparison of Qualitative Models of Software Systems*. *World Applied Sciences Journal*, 6(1), pp. 1–6.
99. Kitchenham, B., Pfleeger, S. (1996) Software quality: the elusive target. *IEEE Software*, 13 (1), pp. 12–21.
100. Kitchenham, B., Budgen, D., Brereton, P. & Woodall, P. (2005) An investigation of software engineering curricula. *The Journal of Systems and Software*, 74, pp. 325–335.
101. Kohler, J. (2003) Quality Assurance, Accreditation, and Recognition of Qualifications as Regulatory Mechanisms in the European Higher Education Area. *Higher Education in Europe*, 28(3), pp. 317–330.
102. Krishnan, M., Kriebel, C., Kekre, S & Mukhopadhyay, T. (2000) An empirical analysis of productivity and quality in software products. *Management Science*, 46(6), pp. 745–759.

103. Laitenberger, O. (2001) Cost-effective detection of software defects through perspective-based inspections. *Empirical Software Engineering*, 6(1), pp. 81–84.
104. LR Augstskolu likums. [tiešsaiste] [14.03.2014.]. Pieejams: <http://www.likumi.lv/doc.php?id=37967>.
105. Li, H. & Meissner, J. (2008) *Improving Quality in Business Process Outsourcing through Technology*.
106. Lethbridge, T. (1998) A survey of the relevance of computer science and software engineering education. In: *Proceedings of the 11th International Conference on Software Engineering*. IEEE Computer Society. Press, Silver Spring, MD, pp. 56–66.
107. Lethbridge, T. (2000) What knowledge is important to a software professional. *IEEE Computer*, pp. 44–50.
108. Leveson, N. (1995) *Safeware: System Safety and Computers*, Addison Wesley-Profesional, p. 704.
109. LLU Studiju kursa un prakses dokumentu izstrādāšanas nolikums (2009). [tiešsaiste] [21.06.2014.]. Pieejams: <http://www.llu.lv/getfile.php?id=66441>.
110. LLU Studiju nolikums [tiešsaiste] [21.02.2014.]. Pieejams: <http://www.llu.lv/getfile.php?id=70161>.
111. Losavio, F., Chirinos, L., Lévy, N. & Ramdane-Cherif, A. (2003) Quality characteristics for software architecture. *Journal of Object Technology*, 2(2), pp. 133–150.
112. Luckett, K. (2008) The Introduction of External Quality Assurance in South African Higher Education: An Analysis of Stakeholder Response. *Quality in Higher Education*, 13(2), pp. 97–116.
113. Lundquist, R. (1999) Quality Systems and ISO 9000 in Higher Education. *Assessment & Evaluation in Higher Education*, 22(2), pp. 159–172.
114. LVS 68:1996 Informācijas tehnoloģija. Programminženierija. Programmatūras prasību specifikācijas ceļvedis. – Rīga, Latvijas Nacionālās standartizācijas un metroloģijas centrs, 1996.
115. LVS 72:1996 Informācijas tehnoloģija. Programminženierija. Ieteicamā prakse programmatūras projektējuma aprakstīšanai. – Rīga, Latvijas Nacionālās standartizācijas un metroloģijas centrs, 1996.
116. MacAskill, P., Goho, J., Richard, R., Anderson, K. & Stuhldreier, M. (2008) Creating quality assurance in curriculum: theory and practice at a Canadian community college. *Community College Journal of Research and Practice*, 32, pp. 939–958.
117. Martens, E., & Prosser, M. (1998) What constitutes high quality teaching and learning and how to assure it. *Quality Assurance in Education*, 6(1), pp. 28–36.
118. Matsubara, T. (1994) Does ISO 9000 really help improve software quality? *American Programmer*, 7(2), pp. 38–45.

119. McCall, J., Richards, P. & Walters, G. (1977) *Factors in software quality. Volume i. Concepts and definitions of software quality.* GENERAL ELECTRIC CO SUNNYVALE CA.
120. Ministru Kabineta noteikumi Nr. 668 "Augstskolu, koledžu un studiju virzienu akreditācijas noteikumi" (2012). [tiešsaiste] [04.01.2014.]. Pieejams: <http://www.likumi.lv/doc.php?id=252142>.
121. Mizikaci, F. (2006) A systems approach to program evaluation model for quality in higher education. *Quality Assurance in Education*, 14(1), pp. 37–53.
122. Moore, N. (1996) *Using the Malcolm Baldrige criteria to improve quality in higher education.*
123. Moses, J. (2009) Should we try to measure software quality attributes directly. *Software Quality Journal*, 17(2), pp. 203–213.
124. Motwani, J. & Kumar, A. (1997) The need for implementing total quality management in education. *International Journal of Educational Management*, 11(3), pp. 131–135
125. Noteikumi par Profesiju klasifikatoru, profesijai atbilstošiem pamatuzdevumiem un kvalifikācijas pamatprasībām un Profesiju klasifikatora lietošanas un aktualizēšanas kārtību – Programmēšanas inženieris. [tiešsaiste] [19.06.2013.]. Pieejams: http://www.lm.gov.lv/upload/darba_devejiem/prof_stand_1210.pdf.
126. Ortega, M., Perez, M. & Rojas, T. (2003) Construction of a Systemic Quality Model for Evaluating a Software Product. *Software Quality Journal*, 11, pp. 219–242.
127. Osseo-Asare, A., Longbottom, D. & Murphy, W. (2005) Leadership best practices for sustaining quality in UK higher education from the perspective of the EFQM Excellence Model Source. *Quality Assurance in Education*, 13(2), pp. 148–170.
128. Owalia, M. & Aspinwall, E. (1996) A framework for the dimensions of quality in higher education. *Quality Assurance in Education*, 4(2), pp. 12–20.
129. Paulk, M., Curtis, B., Chrissis, M. & Weber, C. (1993) *Capability Maturity Model for Software, Version 1.1.* Technical Report, CMU/SEI-93-TR-024, Pittsburgh.
130. Paulk, M. (1999). Using the Software CMM with good Judgment. *Institute for Software Research*, p. 12.
131. Paunescu, C. & Fok, W. (2005) Commitment to Quality Education Services through ISO 9000: A Case Study of Romania. *Online Submission*. [online] [15.06.2014]. Available at: <http://files.eric.ed.gov/fulltext/ED491518.pdf>.
132. Padayachee, I., Kotze, P. & van Der Merwe, A. (2010) ISO 9126 external systems quality characteristics, sub-characteristics and domain specific criteria for evaluating e-Learning systems. *The Southern African Computer Lecturers' Association, University of Pretoria, South Africa.*

133. Pfleeger, S. (2009) *Software Engineering: Theory and Practice*, 4th ed., Prentice Hall, p. 792.
134. Pires Da Rosa, M., Saraiva, P. & Diz, H. (2001) The development of an Excellence Model for Portuguese Higher Education Institutions. *Total Quality Management*, 12(7-8), pp. 1010–1017.
135. Pounder, J. (1999) Institutional performance in higher education: is quality a relevant concept? *Quality Assurance in Education*, 7(3), pp. 156–165.
136. Pressman, R. (2010) *Software engineering: a practitioner's* 7th ed. New York: McGraw-Hill Higher Education, p. 895.
137. Ramsden, P. (1991) A performance indicator of teaching quality in higher education: The Course Experience Questionnaire. *Studies in Higher Education*, 16(2), pp. 129-150.
138. Rauhvargers, A. (2004) Kvalitātes definīcijas un metodiskās pieejas izglītībā. [tiešsaiste] [10.6.2009.] Pieejams: http://www.aic.lv/bolona/Latvija/Atsev_prez/91_kvalitate_iev.pdf.
139. Rawashdeh, A. & Matalkah, B. (2006) A new software quality model for evaluating COTS components. *Journal of Computer Science*, 2(4), pp. 373–381.
140. Recommendation of the European Parliament and of the Council of 15 February 2006 on further European cooperation in quality assurance in higher education (2006). *Official Journal of the European Union*. 4(3), pp. 62–64.
141. Reeves, C. & Bednar, D. (1994) Defining Quality: Alternative and Implications. *Academy of Management Review* 19(3), pp. 419–445.
142. Rosa, M. & Amaral, A. (2010) A Self-assessment of Higher Education Institutions from the Perspective of the EFQM Excellence Model. *Higher Education Dynamics*, 1, 20, *Quality Assurance in Higher Education, Part III*, pp. 181–207.
143. Rozemeijer, E. (2007) *Frameworks for IT Management – A Pocket Guide*. Van Haren Publishing, p. 143.
144. Sahney, S., Banwet, D. & Karunes, S. (2004) Conceptualizing total quality management in higher education. *The TQM magazine*, 16(2), pp. 145–159.
145. *Selection and use of the ISO 9000 family of standards* (2009) [online] [21.07.2013]. Available at: http://www.iso.org/iso/iso_9000_selection_and_use-2009.pdf.
146. Skolnik, M. (1989) Quality assurance in higher education as a political process. *Higher Education Management and Policy*, 22(1), pp. 67–86.
147. Sohn, S. & Mok, M. (2008) A strategic analysis for successful open source software utilization based on a structural equation model. *Journal of Systems and Software*, 81(6), pp. 1014–1024.
148. Song, Q., Shepperd, M., Cartwright, M. & Mair, C., (2006) Software defect association mining and defect correction effort prediction. *IEEE Transaction Software Engineering*, 32 (2), pp. 69–82.

149. Spröge, S. & Cevere, R. (2010) Quality model of the curricula of information technology studies. In: *Proceedings of the 4th International Scientific Conference on the Applied Information and Communication Technologies*. Jelgava: LLU (Latvia University of Agriculture), [CD-ROM], pp. 148–157.
150. Standards and Guidelines for Quality Assurance in the European Higher Education Area, (2005), Helsinki: *European Association for Quality Assurance in Higher Education*. [online] [05.11.2010]. Available at: http://www.eqar.eu/uploads/media/050221_ENQA_report_01.pdf.
151. Stelzer, D., Mellis, W. & Herzwurm, G. (1996) Software process improvement via ISO 9000? In *Proceedings of the Twenty-Ninth Annual Hawaii International Conference on System Sciences*, 3-6 January, 1996, Wailea, Hawaii, USA, pp. 703–712.
152. Stelzer, D., Mellis, W. & Herzwurm, G. (1997) A critical look at ISO 9000 for software quality management, *Software Quality Journal*, 6, pp. 65–79.
153. Stewart, N. (1988) Software error costs. *Quality Progress*, 21(11), pp. 48–49.
154. Stevenson, T. & Barnes, F. (2001) Fourteen years of ISO 9000: impact, criticisms, costs, and benefits. *Business Horizons*, 44(3), pp. 45–51.
155. Srikanthan, G. & Dalrymple, J. (2002) Developing a Holistic Model for Quality in Higher Education. *Quality in Higher Education*, 8(3), pp. 216–224.
156. Srikanthan, G. & Dalrymple, J. (2003) Developing Alternative Perspectives for Quality in Higher Education. *International Journal of Educational Management*, 17(3) pp. 126–136.
157. Srikanthan, G. & Dalrymple, J. (2004) A synthesis of a quality management model for education in universities. *International Journal of Educational Management*, 18(4), pp. 266–279.
158. Suryn W., Abran A. & April A. (2003) *ISO/IEC SQuaRE. The second generation of standards for software product quality*. 7th IASTED International Conference on Software Engineering and Applications, November 2003. – USA: Marina del Rey, pp. 807–814.
159. Šmite, D., Dosbergs, D. & Borzovs, J. (2005) Informācijas un komunikācijas tehnoloģijas nozares tiesību un standartu pamati. Rīga, LU Akadēmiskais apgāds, 207 lpp.
160. Tam, M., (2001) Measuring quality and performance in higher education. *Quality in Higher Education*, 7(1), pp. 47–54.
161. Tan, K. & Kek, S. (2004) Service quality in higher education using an enhanced SERVQUAL approach. *Quality in Higher Education*, 10(1), pp. 17–24.
162. Tervonen, I. & Pentti, K. (1998) Towards deeper co-understanding of software quality. *Information and Software Technology*, 39(14–15), pp. 995–1003.

163. *The EFQM Excellence Model* (2009). [online] [01.12.2010]. Available at: <http://www.efqm.org/the-efqm-excellence-model>.
164. *The Case for Application Security*. [online] [24.03.2013]. Available at: http://www.cio.com/documents/webcasts/fortify/application_security/The_Case_For_Application_Security.pdf.
165. Thonhauser, T. & Passmore, D. (2006) ISO 9000 in Education: a comparison between the United States and England. *Research in Comparative & International Education*, 2, pp. 156–172.
166. Van Der Wende, M. & Westerheijden, D. (2001) International aspects of quality assurance with a special focus on European higher education. *Quality in higher education*, 7(3), pp. 233–245.
167. Vaniček, J. (2006) Software quality requirements. *AGRIC. ECON. – CZECH*, 52(4), pp. 177–185.
168. Verbitsky, O. (2012) *Characteristics of Effective Information Systems* [online] [04.04.2013]. Available at: www.visualspec.org/On_System_Design/Characteristics_of_Effective_Information_Systems.
169. Whittaker, J. & Voas, J. (2002) 50 years of software: key principles for quality. *IT Pro*, pp. 28–35.
170. Williams, A., Wiele, A. & Dale, B. (1999) Quality costing: a management review. *Journal of Management Reviews*, 1 (4), pp. 441–460.
171. Yorke, M. (1999) Assuring quality and standards in globalised higher education. *Quality Assurance in Education*, 7(1), pp. 14–24.
172. Zeiss, B., Vega, D., Schieferdecker, I., Neukirchen, H. & Grabowski, J. (2007) Applying the ISO 9126 quality model to test specifications. *Software Engineering*, pp. 231–242.