

TARKVARA ARENDAJA III

EESSÕNA

Eesti kutsekvalifikatsiooni süsteemis määratletakse kutsekvalifikatsiooni nõudeid viiel tasemel. I tase on madalaim ja V tase kõrgeim (vt lisa A – Kutsekvalifikatsiooni süsteemi terminid). Kõik kutsed ei eelda kutsekvalifikatsiooni tasemete fikseerimist I kuni V tasemeni. Iga konkreetse kutse kvalifikatsioonitasemed, sealhulgas vajaduse korral ka haridusnõuded, määrab kindlaks kutsenõukogu.

Tarkvara arendaja III kutsestandardi aluseks on EUCIP, European Certificate of Informatics Professionals (www.eucip.com) sertifitseerimissüsteemi tarkvara arendaja valiktaseme eksami nõuded (EUCIP “Software Developer” elective level profile specification). EUCIP on informaatikaprofessionaalide üleeuroopaline sertifitseerimissüsteem, mille on välja töötanud [CEPIS](http://www.cepis.org), the Council of European Professional Informatics Societies (www.cepis.org).

Kutsestandardi koostas Infotehnoloogia ja Telekommunikatsiooni Kutsenõukogu moodustatud töörühm koosseisus:

Tanel Tammet	TTÜ
Lily Loidap	Eesti Infotehnoloogia Selts
Jaan Oruaas	Eesti Infotehnoloogia Selts
Kaie Piiskop	Riiklik Eksami ja Kvalifikatsioonikeskus
Kristjan Kangur	ADM Group
Urve Mets	BCS Koolitus

2004 a novembris - detsembris viidi läbi kutsestandardi kavandi arvamusküsitlus. Kutsestandardi lõppredaktsiooni koostamisel on töörühm arvestanud arvamusküsitlustel tehtud ettepanekuid ja märkusi.

Käesolev kutsestandard on koostatud esmakordselt

Tarkvara arendaja III kutsestandard on kinnitatud 16. detsembril 2004. a Infotehnoloogia ja Telekommunikatsiooni Kutsenõukogu otsusega nr. 9.

Kutsestandardis määratletud kutsekvalifikatsioon on kantud kutseregistrisse.

1 KASUTUSALA

Kutsestandardite kasutusala on järgmine:

- 1) töötajate kvalifikatsiooninõuete määratlemine
- 2) õppekavade, koolitusprogrammide väljatöötamine
- 3) eksaminõuete väljatöötamine, kutsekvalifikatsiooni tõendamine ja hindamine
- 4) aluse andmine rahvusvaheliste kvalifikatsiooni tõendavate dokumentide võrdlemiseks

2 KUTSESTANDARDIGA SEONDUVAD STATISTILISED KOODID

Eesti Majanduse Tegevusalade Klassifikaatori¹ järgi kuulub tarkvara arendaja töö arvutiteeninduse valdkonda, kood 72.

Ametite Klassifikaatori² järgi kuuluvad tarkvara arendajad 2. pearühma “Tippspetsialistid”, allrühm 213 “Informaatikud”

3 KUTSENIMETUS JA KUTSEKVALIFIKATSIOONI TASE

Eesti keeles:	Tarkvara arendaja III
Inglise keeles:	Software Developer III
Soome keeles:	ATK systemien kehittelijä III

4 KUTSEKIRJELDUS

Programmeerimine:

Etteantud kirjelduste järgi uute ja täiendatud programmide arendus, testimine ja dokumenteerimine vastavalt kokkulepitud standarditele.

Süsteemiarendus:

“Määratletud äri vajadustele vastavate infosüsteemide lahenduste spetsifitseerimine ja arendus.”

Lühikirjeldus

EUCIP tarkvara arendajalt oodatakse märkimisväärse tehnilise rolli täitmist infosüsteemide projekteerimises ja väga suurt efektiivsust keerukate tarkvaramoodulite, mis tavaliselt integreeritakse laiematesse infosüsteemidesse, looja ja hooldajana.

Võimalikud on erinevad spetsialiseerumised rakendustele ja veebiteenustele või süsteemitarkvarale.

Tööülesannete kirjeldus

Määratleb üksikasjalise spetsifikatsiooni ja panustab otseselt keeruka tarkvarasüsteemi tulemuslikku loomisse ja/või uuendamisse, kasutades asjakohaseid standardeid ja töövahendeid.

¹ Statistical classification of economics activities in the European Community (NACE) eestistatud versioon

² International Standard Classification of Occupations (ISCO-88) eestistatud versioon

Kindlustab, et tulemused vastavad nõuetele nii lahenduse tehniliselt kõrge taseme kui ka kokkulepitud funktsionaalsele kirjeldusele vastavuse osas.

On teadlik kasutatavatest standarditest, meetoditest ja töövahenditest, mis on asjakohased antud töökeskkonna jaoks. Teeb kindlaks nende eelised ja puudused ning kasutab neid oskuslikult ja tulemuslikult, et saavutada heal inseneritasemel lahendus, mille puhul on kindlustatud tarvilikud omadused: eesmärgipärasus, töökindlus, tulemuslikkus, turvalisus, ohutus, hooldatavus ja kuluefektiivsus.

Lahendab tehnilisi probleeme tarkvara rakendusprojektides ja teiste tarkvara elutsükli etappide jooksul: uuring, analüüs, kirjeldamine, projekteerimine, programmeerimine, testimine, rakendamine ja tarkvara hooldus.

Kindlustab asjakohase dokumentatsiooni loomise ja hoolduse.

Kasutab andmebaaside haldussüsteeme ja vastavaid analüüsivahendeid, uurimaks andmebaaside jõudlus-statistikat; loob vajadusel aruandeid, mis sisaldavad muuhulgas ettepanekuid täiustusteks ja probleemide lahendamiseks.

Saab aru peamistest võimalustest andmebaaside seadistamisel, suudab pakkuda tuge tarkvara installeerimise, serverite ja rakenduste uuenduste puhul. Kindlustab versioonikontrolli protseduuride järgimise, teeb vajadusel parandusi, hooldab tarnija ja kasutaja juhendeid. Pühendub keerukate ja mittestandardsete olukordade lahendamisele: võtab tehnilise vastutuse tarkvaraarenduse elutsükli teostusfaasis, osaleb uuringutes, analüüsimisel, spetsifitseerimisel, projekteerimisel, programmeerimisel, testimisel, hoolduses, laiendamisel ja migratsioonil. Saavutab heatasemelise toote valmimise.

Kogenud tarkvara-arendaja juhib meeskonda, juhatab tehnilise personali gruppi, omab ekspertteadmisi arhitektuursete lahenduste ning tark- ja riistvara valikuks, hankimiseks ja kasutamiseks. Võtab täieliku vastutuse töö kvaliteedi ja valmimise õigeaegsuse eest ning rakendab kõiki tema käsutusse antud ressursse.

Kui tegemist on veebilahendustega, valib sobivad töövahendid, mallid ja standardid, et luua määratud sisu ja väljanägemisega edumeelne, hea kujundusega ja tehniliselt kõrgetasemeline veebileht. Testib veebilehte ja parandab koodi vead. Aitab vähemkogenud kolleege raskemate probleemide korral. Loob koodistandardid oma organisatsiooni jaoks, arvestades ühenduste läbilaskevõimet ja lehitsejate ühilduvust. Määratleb sobivad serveririistvara ja võrguühendused väikestele ja keskmistele infosüsteemidele.

Hoiab end kursis asjakohase hulga veebilehtede sisuga ja osaleb olulistes aruteludes veebivahendite ja tehnoloogiate arengu osas. Kasutab saadud teadmisi klientide ja kasutajate nõustamisel tulevikutrendide alal, prognoosib vajalikke muudatusi oma organisatsiooni standardite jaoks.

Muudab loogilised määratlused üksikasjalikeks projektlahendusteks, arvestades rakenduskeskkonna tehnilisi ja mittetehnilisi omadusi ning piiranguid.

Interpreteerib objekt - ja andemudeleid asjakohaselt andmebaasi skeemideks etteantud piirangute (andmeühtsus, turvalisus, omandus) tingimustes, loob nõutud andmebaasi-objektid.

Loob ja muudab, testib ja parandab etteantud määratluste järgi suuri ja/või keerukaid mooduleid.

Valmistab ette ja korraldab tarkvara moodulite testimist; teeb kindlaks vead ja tõrgete põhjused, täiendab programmide ja süsteemide konfiguratsiooni täielikult rahuldava tulemuse saavutamiseni.

Panustab integraalsete tarkvaramoodulite määratlemisse nii, et need vastaksid tarkvara testimise etteantud nõuetele ja töötaksid etteantud riistvaraplatvormil

Tarkvara arendaja III kutsekvalifikatsiooni ja EUCIP sertifikaadi saamise eelduseks on eksami sooritamine, mis põhineb kõigil käesoleva kutsestandardi punktis 5 toodud oskustel-teadmistel. Nõutav on oskuste - teadmiste omamine vastavalt infotehnoloogia spetsialist II kutsestandardi (EUCIP baastase) nõuetele ja 3 aastat kutsealase töö kogemust

Vajalikud isikuomadused ja suhtlemisoskused [1]³

(Sotsiaalne kvalifikatsioon)

Programmeerija/süsteemi arendaja roll nõuab esmajärjekorras ratsionaalset vaimset hoiakut, suutelisust mõisteliseks (skemaatiliseks) ja analüütiliseks mõtlemiseks, üksikasjadele pühendumist ning järjekindlalt eesmärgile orienteeritud lähenemisviisi, mis viib tulemuseni läbi paindlikult formuleeritud ja struktureeritud lahenduste.

Teine oluline kogum oskusi on seotud kommunikatsiooni ja tulemusliku suhtlemisega (nii suuliselt kui kirjalikult) töökaaslaste ja klientidega. See hõlmab üldist teadlikkust organisatsiooni funktsioneerimise tasanditest, head meeskonnatöö võimet ja tulemuslikku infohankimist, võimet planeerida, korraldada, teha tehnilisi otsustusi, pakkuda lahendusi ja jätkuprogramme.

Võimalikud koolitusmoodulid sisaldavad koosoleku juhtimist, ajaplaneerimist ja meeskonnatööd.

5 PÕHIOSKUSED JA –TEADMISED (vt lisa B “EUCIP tarkvaraarendaja õppekava moodulid”)

- 5.1 Programmeerimine [4½]
- 5.2 Objektorienteeritud lähenemine süsteemiarendusele [1½]
- 5.3 Tarkvaratehnika põhimõtted [1½]
- 5.4 Tarkvara arendusprotsess [1]
- 5.5 Töö andmebaasidega [1]
- 5.6 CASE- ja IDE-vahendid programmide projekteerimisel ja arendamisel [1]
- 5.7 Arendus koostöökeskkonnas [1]
- 5.8 Operatsioonsüsteemid [4]
- 5.9 Programmeerimiskeeled [4½]
- 5.10 Veebirakenduse projekteerimine ja arendamine [1]
- 5.11 Internetirakenduste loomine [1]
- 5.12 Hajusate ja kriitiliste rakenduste projekteerimine ning juurutamine [1]
- 5.13 Turvaline programmeerimine [1]
- 5.14 Aruannete koostamine [1]
- 5.15 Lahenduste visioneerimine [1]
- 5.16 Lahenduse arhitektuuri määramine [1]
- 5.17 Projekti koordineerimine [1]
- 5.18 Objekt- orienteeritud lähenemine süsteemianalüüsile [1]
- 5.19 Ärilahenduste projekteerimine ja juurutamine toimingute toetamisega [1]
- 5.20 Andmebaaside projekteerimine ja juurutamine [1]

³ Number sulgudes näitab EUCIP punkte

KUTSEKVALIFIKATSIOONI SÜSTEEMI TERMINID

Kutsestandard – dokument, mis määrab kindlaks kutsekvalifikatsioonist tulenevad nõuded teadmistele, oskustele, vilumustele, kogemustele, väärtushinnangutele ja isikuomadustele.

Kutsekvalifikatsioon – antud kutsealal nõutav kompetentsuse tase, mida tunnustatakse kas reguleeritud, ajalooliselt või rahvusvaheliselt kujunenud nõuete alusel.

Kutsekvalifikatsioonisüsteemis on viis taset, kusjuures I tase on madalaim ja V tase kõrgeim.

I tase – töötaja täidab tööülesandeid ühesuguses olukorras, on omandanud kutsealased oskused ja teadmised enamasti kutsealasel väljaõppel, võib vajada juhendamist töö käigus, vastutab oma tööülesannete täitmise eest;

II tase – töötaja täidab tööülesandeid erisuguses olukorras, lisaks enamasti kutsealasel väljaõppel omandatud oskustele ja teadmistele omab vilumust ja kogemust, töötab iseseisvalt, vastutab oma tööülesannete täitmise eest;

III tase – töötaja täidab tööülesandeid erisuguses ja vahelduvas olukorras, lisaks enamasti kutsealasel väljaõppel omandatud oskustele ja teadmistele ning vilumustele ja kogemustele omab meisterlikkust, valmisolekut kutsealaste oskuste ja teadmiste edasiandmiseks, korraldab ressursside jagamist ja teiste tööd ning vastutab selle eest;

IV tase – töötaja täidab analüüsimist ja otsustamist eeldavaid tööülesandeid muutuv olukorras, omab kutsealaseid teadmisi ja oskusi; korraldab ressursside jagamist ja teiste tööd ning vastutab selle eest;

V tase – töötaja täidab teadmiste laiendamist, probleemide lahendamist, teaduslike teooriate ja mõistete rakendamist, olemasolevate teadmiste analüüsimist, süstematiseerimist ja edasiarendamist ning õpetamist eeldavaid tööülesandeid muutuv olukorras, omab laialdasi kutsealaseid teadmisi ja oskusi, korraldab ressursside jagamist ja teiste tööd ning vastutab selle eest.

EUCIP TARKVARAARENDAJA ÕPPEKAVA MOODULID

(Number sulgudes näitab EUCIP punkte, punkte kokku 31)

1. Programmeerimine [4½]

- 1.1 Kasutab erinevaid programmide projekteerimise meetodeid, nagu objekt-orienteeritud (OO) programmeerimine, laskuv meetod, struktuurprogrammeerimine.
- 1.2 Teab, kuidas kasutada abstraktsioonitehnikat probleemide lahendamisel ja projekteerimisel.
- 1.3 Tuleb toime pärandüsteemide eriliste nõuetega programmide loomisel.
- 1.4 Kasutab erinevaid andmestruktuure, näiteks kirje, massiiv ja seotud ahelloend.
- 1.5 Otsustab, kus ja missuguse algoritmi puhul kasutada millist andmestruktuuri.
- 1.6 Kasutab mõnda programmeerimiskeelte põhitüüpi (funktsionaalsed, protseduurilised, objekt-orienteeritud vms) uute algoritmide ja funktsioonide loomisel või olemasolevate programmide muutmisel.
- 1.7 Interpreteerib programmeerimiskeele süntaksit õigesti.
- 1.8 Valib kompileeritava või interpreteeriva programmeerimiskeele vahel.

2. Objektorienteeritud lähenemine süsteemiarendusele [1½]

- 2.1 Väärtustab objektprojekteerimise-keskset lähenemist
- 2.2 Tunnetab objektide kasulikkust ja nendest tulenevat tarkvaramoodulite korduvkasutust
- 2.3 Saab aru objektide ja klasside kasutamisest
- 2.4 Saab aru abstraktsioonist, kapseldusest, polümorfismist ning teadete edastamisest.
- 2.5 Saab aru teostuse ja testimise mudelitest UML-s.
- 2.6 Osaleb UML klasside määramisel.
- 2.7 Kasutab standardseid klasside notatsioone ja konventsioone.
- 2.8 Loob klassidiagramme UML-s.
- 2.9 Määratleb klasside atribuudid, kooslused/seosed, operatsioonid ja meetodid.
- 2.10 Teeb kasutatavaks pärilikkuse ja koonduvuse hierarhiad.
- 2.11 Saab aru püsivate ja mittepüsivate klasside erinevustest.
- 2.12 Saab aru OO projekteerimise ja teiste projekteerimismeetodite erinevustest.

3. Tarkvaratehnika põhimõtted [1½]

- 3.1 Rakendab tarkvaratehnika põhimõtteid: üldistatud programmeerimine vs. detailne programmeerimine - *programming-in-the-small* vs. *programming-in-the-large*.
- 3.2 Peab kinni tarkvaraprotsessi mudelitest: elutsüklid, koskmudel ja variatsioonid, riskipõhine lähenemine, omab algteadmisi evolutsioonilisest mudelist ja prototüüpimisest ning süsteemide klassifikatsioonist.
- 3.3 Arvestab kasutajate vajadustega: inimtegur, kasutatavus, rahvusvahelistumine.
- 3.4 Kasutajaliidese dokumentatsioon, kasutajajuhend.

- 3.5 Projekteerib heatasemelisel, rakendades uuestikasutust, muudetavust, kvaliteetset notatsiooni, projekti hindamist ja valideerimist.
- 3.6 Tarkvara rakendamine: programmeerimise standardid ja protseduurid, modulaarsus, andmete abstraherimine, staatiline analüüs, testimine, testimisvahendid, tõrkekindlus.
- 3.7 Tarkvarasüsteemi hooldus: probleemid hoolduses, hoolduse olemus, hoolduse planeerimine.

4. Tarkvara arendusprotsess [1]

- 4.1 Kirjutab dokumentatsiooni sobivas formaadis, sobivate vahenditega, sh sisedokumentatsiooni.
- 4.2 Kasutab formaalmeetodeid, kasutab tarkvaratehnika (*software engineering*) vahendeid ja keskkondi, teab programmeerimise paradigmat ja protsessi küpsuse rolli.
- 4.3 Teeb kiiret prototüüpimist (*Rapid Prototyping*).
- 4.4 Viib läbi testimise/vastuvõtu/levitamise protseduure
 - a) peamiste kasutajaliidese-komponentide loomiseks
 - b) teiste süsteemi-määramatuste, nagu reaktsiooniaeg (*response time*), skaleeritavus jne., uurimise prototüüpide loomiseks.
- 4.5 Kasutab projekti planeerimise ja jälgimise meetodeid ja tehnikaid. Näited: töö etappide struktuur, kriitilise tee analüüs, konflikti lahendamine.
- 4.6 Korregerib "kurssi" ja mõõdikuid vastavalt muudatuste kontrolli protsessile.
- 4.7 Kasutab asjakohast kodeerimisprotsessi arenduskeskkonnas, mis on mõeldud suurte rööptöötlussüsteemide loomiseks, samuti sardsüsteemide, reaajasüsteemide ja kõrge töökindlusega süsteemide loomiseks.
- 4.8 Viib läbi vastuvõtutesti.
- 4.9 On võimeline määrama etapi tähtaegu.
- 4.10 Testib funktsionaalsust ja süsteemi koormustaluvust.
- 4.11 Kasutab erinevaid tarkvarapakette erinevateks testideks ja veaotsinguks.
- 4.12 Koostab vastuvõtutesti.
- 4.13 Toetab kasutuselevõttu ja üleandmist.
- 4.14 Pakub rakenduste tuge ja tehnilist tuge.

5. Töö andmebaasidega [1]

- 5.1 Andmete impordi ja ekspordi meetodid, s.h. hulgikopeerimismeetodid.
- 5.2 Tulemikogumi juhtimine kursorite ja SQL kasutamisega: lukustava mudeli asjakohane kasutus.
- 5.3 Andmete väljavõte XML formaadis: väljundformaadi ja XML skeemi struktuuride kasutamine.
- 5.4 Andmete käsitlemine, kasutades salvestatud protseduure, toiminguid, triggereid, kasutaja poolt määratud funktsioone ja vaateid.
- 5.5 Programmiobjektide veaotsing ja optimeerimine: objektid sisaldavad salvestatud protseduure, toiminguid, triggereid, kasutaja määratud funktsioone ja vaateid.
- 5.6 Andmete juurdepääsu kontroll, kasutades salvestatud protseduure, toiminguid, triggereid, kasutaja määratud funktsioone ja vaateid.

- 5.7 Määratleb turvalisust objekti tasemel k.a. veeru tasemel, rakendades GRANT, REVOKE, ja DENY.
- 5.8 Teab, kuidas kasutada standardseid andmebaasiliideseid nagu ODBC, JDBC jne.

6. CASE- ja IDE-vahendid programmide projekteerimisel ja arendamisel [1]

- 6.1 Teab, millal ja kuidas kasutada CASE vahendeid: peamised probleemid CASE vahendite kasutamise juures, CASE vahendid erinevatele platvormidele ja keeltele.
- 6.2 Töö enamlevinud IDE vahenditega Windows ja Unix platvormidel.
- 6.3 Lisaprogrammide integreerimine IDE-sse. Näited: Oracle Developer 2000, Rational ROSE, Select, Business Objects.
- 6.4 Arendusprotsessi kohaldamine IDE-s.
- 6.5 Konfigureerimisrakenduse kasutamine.
- 6.6 IDE integreerimine versioonikontrolli süsteemiga. Näide: CVS.

7. Arendus koostöökeskkonnas [1]

- 7.1 Kasutab meeskonnatöö vahendeid koostöökeskkonnas.
- 7.2 Saab aru meeskonnatöö põhiprobleemidest.
- 7.3 Valdab versioonikontrolli, tehnilist dokumentatsiooni ja levitamishahendeid.
- 7.4 Valdab arendust ja testimist.
- 7.5 Kasutab teavitushahendeid: IM, meililist, foorumid.
- 7.6 Edendab koostöökeskkonda.
- 7.7 Rakendab meeskonnatöö protseduure.
- 7.8 On teadlik kehtestatud dokumentatsiooni- ja kodeerimisstandardite olulisusest.
- 7.9 Kasutab arendajate virtuaalsest suhtluskeskkonnast saadavat üksikasjalikku teavet ja veaotsingu viiteid.

8. Operatsioonisüsteemid [4]

- 8.1 Teab enamlevinud operatsioonisüsteemide erinevusi:
 - a) Linux/Unix
 - b) Windows
 - c) MacOS
- 8.2 Installeerib ja täiendab loetletud OS-e.
- 8.3 Saab aru OS-de kontseptuaalsetest probleemidest:
 - a) parallelismi juhtimine, lukustus/ummistus ja nälg.
 - b) ajajaotus
 - c) sisend-väljund operatsioonid ja nende juhtimine
 - d) failisüsteem
 - e) kasutajate ja juurdepääsude administreerimine
- 8.4 Analüüsib võrgu võimalusi.
- 8.5 Konfigureerib võrguliideseid.
- 8.6 Konfigureerib erinevaid võrguprotokolle ja teenuseid (http, SMTP, POP, IMAP, DNS).
- 8.7 Stardib ja peatab erinevaid võrguteenuseid.
- 8.8 Avalikustab võrguressursid (jagatud printerid ja kaustad).
- 8.9 Mõõdab ja jälgib võrgukoormust:

- a) protsessor (mono- ja multiprotsessor)
 - b) võrk
 - c) mälu ja virtuaalmälu
 - d) andmekandjad
 - e) protsessid ja lõimed
 - f) jagatud ressurside kasutus
- 8.10 Häälestab süsteemi vajaliku tootlikkuse saavutamiseks.
- 8.11 Haldab kasutajakontosid ja -grupe ning rakendab vastavat turvapoliitikat.
- 8.12 Rakendab koosvõime-juhendeid (failiformaadid, olemasolevad protokollid, jne.).
- 8.13 Häälestab süsteeme, saavutamaks vajalikku koostöövõimet erinevate OS-de vahel.
- 8.14 Kasutab võimendavaid tehnikaid, näiteks klasterdamine.
- 8.15 Loob klasterduse.
- 8.16 Teostab veaotsingu.
- 8.17 Taastab süsteemi.

9. Programmeerimiskeeled [4½]

- 9.1 Kirjutab protseduurkeeltes tulemuslikku lähtekoodi. Näited: Basic, Pascal, C, Cobol jne.
- 9.2 Kasutab OO programmeerimiskeeli. Näited: C++, Java, Delphi jne.
- 9.3 Kasutab skriptkeeli. Näited: PERL, Python, Ruby jne.
- 9.4 Määratleb märgendkeele põhimõtted.
- 9.5 Kasutab XML-i ning XML-põhise andmebaasi päringuvahendeid. Kasutab XML tehnoloogiat rakendustes, teab XSLT-d ja kuidas seda dokumentide teisendamiseks kasutada.

10. Veebirakenduse projekteerimine ja arendamine [1]

- 10.1 Valib platvormi, mis toetab erinevaid programmeerimiskeeli ja keskkondi
- kas
- a) *Master servletid* ja JSP, mis on J2EE standardi populaarseimad komponendid, kasutuses kriitiliste elementidena e-äri veebilehtede loomisel.
 - b) Loob veebipõhise rakenduse kasutades Java servlette ja Java Server Pages (JSP). Teab ja kasutab API/servlet kontseptsiooni, lisaks tulemuslikku rakenduste arendust Java Server Pages kasutamiseks.
- või
- c) Master COM/COM+/.NET ja ASP.
 - d) Loob veebipõhise rakenduse kasutades ASP või VBA-d .NET keskkonnas..
- 10.2 Teab veebiteenuste kasutamise kontseptsiooni

11. Internetirakenduste loomine [1]

- 11.1 Loob vormimoodulid, mis sisaldavad komponente andmebaasiga suhtlemiseks ja graafilise kasutajaliidese nuppe/ikoone.
- 11.2 Kasutab nii objekte kui koodi mitmekordselt.
- 11.3 Valib asjakohase andmeallika andmeblokkidele.
- 11.4 Kindlustab rakenduse turvalisuse.
- 11.5 Loob ja haldab mitmevormilisi internetirakendusi.

12 Hajusate ja kriitiliste rakenduste projekteerimine ning juurutamine [1]

- 12.1 Valib õige toimingute toetuse tasem.
- 12.2 Planeerib ja projekteerib tootlikkuse, hooldatavuse, laiendatavuse, käideldavuse, skaleeritavuse ja töökindluse aspekte, võttes seejuures arvesse:
 - a) toimingute arvu ajahikus
 - b) ribalaiust
 - c) suutlikust
 - d) tipp- ja keskmise kasutatavuse nõudeid
 - e) reaktsiooniaja nõudeid
 - f) tootlikkuse piire
 - g) serveri protsessid hulkasid
 - h) rööptöötlust
 - i) ootuseid hooldusele
 - j) hooldameeskonna asukohta ja teadmiste taset
 - k) kolmandate osapoolte hoolduslepingute mõju
 - l) tööaega
 - m) käideldavuse taset
 - n) mittetöötamisaja mõju
 - o) partnerite kasvu
 - p) ettevõtete kasvu
 - q) dokumentide hulka
- 12.3 Projekteerimise sidusus olemasolevate rakendustega. Tuletab lahenduse.
- 12.4 Komponentide kauginstallaerimine, võttes arvesse ligipääsuga ja ligipääsuta installatsioone.
- 12.5 Installatsiooni ebaõnnestumise põhjuste otsimine.
- 12.6 Olukorra ja võimaluste hindamine spetsiaalkomponentide kasutamiseks.
- 12.7 Tootlikkuse jälgimine ja optimeerimine: tootlikkuse loendurid jms.
- 12.8 Diagnoosib ja lahendab kasutamisvigu.

13. Turvaline programmeerimine [1]

- 13.1 Rakendab turvalise kodeerimise põhimõtteid.
- 13.2 Minimiseerib, isoleerib ja lihtsustab suurendatud privileegidega koodi kasutamist.
- 13.3 Saab aru peamistest koodi- ja andmestruktuuridest tulenevatest turvariskidest.
- 13.4 Ei usalda programmi väliseid väärtusi (argumente, keskkonna muutujaid jne.).
- 13.5 Väldib kasutamast funktsioone, mis kirjutavad ilma puhvrite pikkust kontrollimata.
- 13.6 Väldib linkimist dünaamiliste teekidega, kasutab staatilist linkimist.
- 13.7 Väldib ajutisi faile avalikes teekides (/tmp).
- 13.8 Tunneb ära nn. race-tingimused.
- 13.9 Projekteerib turvalist infrastruktuuri. Projekteerib hästikasutatavat arhitektuuri, arvestades seejuures turvalisust, tootlikust, hooldatavust, laiendatavust, käideldavust, skaleeritavust, töökindlust.
- 13.10 Jälgib tootlikkuse loendureid ja sündmuste logi.

14. Aruannete koostamine [1]

- 14.1 Administreerib serveri ressursse.
- 14.2 Koostab kvaliteetsed veebiaruanded.
- 14.3 Kasutab kirjade saatmisel malle.
- 14.4 Loob ja muudab peamisi tabelaruandeid.
- 14.5 Aruannete loomisel kasutab XMLi.
- 14.6 Kasutab dünaamilisi andmeid HTML lehtedel.
- 14.7 Tunneb ära aruande peamised osad ja nendevahelised seosed.
- 14.8 Avaldab aruandeid veebis.
- 14.9 Kujundab aruandeid.
- 14.10 Koostab vahearuandeid ja maatriksaruandeid.
- 14.11 Kasutab aruande parameetreid ja kujundab käivitusparameetrite vorme.
- 14.12 Hooldab aruandemalle.
- 14.13 Koostab ja kasutab aruannetes diagramme.
- 14.14 Määrab standardse aruandestiili, saadab olemasolevad aruanded vajalikele saajatele.

15. Lahenduste visioneerimine [1]

- 15.1 Visioneerib ja arendab lahenduse kontseptsiooni.
- 15.2 Analüüsib lahenduse teostatavust. Analüüsib ja täiustab lahenduse projekti. Määrab projekti peamised riskid
- 15.3 Panustab ärivajaduste kogumisse ja analüüsi.
- 15.4 Koostab äri- ja andmevajaduste kontseptuaalse mudeli: *Object Role Modelling* (ORM) ja UML. Hindab kontseptuaalset projekteerimist. Koostab lahenduse loogilise projekti. Koostab loogilise andmemudeli. Hindab pakutud loogilist projekti.
- 15.5 Kirjutab selge spetsifikatsiooni.

16. Lahenduse arhitektuuri määramine [1]

- 16.1 Kogub ja analüüsib kasutaja vajadusi. Kogub ja analüüsib funktsionaalseid vajadusi. Kogub ja analüüsib nõudeid riist- ja tarkvarale ning võrgu infrastruktuurile.
- 16.2 Koostab vajaduste alusel funktsionaalse spetsifikatsiooni: tootlikkus, hooldatavus, laiendatavus, rakendatavus, turvalisus ja juurdepääs.
- 16.3 Koostab funktsionaalsest spetsifikatsioonist tehnilise spetsifikatsiooni: tootlikkus, hooldatavus, laiendatavus, skaleeritavus, käideldavus, rakendatavus, turvalisus ja juurdepääs.
- 16.4 Valib asjakohase tehnoloogia lahenduse teostamiseks.
- 16.5 Loob lahenduse jaoks projekti plaani. Koostab spetsifikatsioonid auditeerimiseks ja sündmuste logimiseks. Koostab projekti plaani rakendamiseks. Koostab projekti plaani hoolduseks. Koostab projekti plaani andmemudelile.
- 16.6 Hindab projekti plaani.

17. Projekti koordineerimine [1]

- 17.1 Koordineerib tarkvara arendusprojekti: planeerimine, kontroll, organiseerimine, konfiguratsioonihaldus, versioonikontroll, kvaliteedi tagamine, mõõdikud.
- 17.2 Standardite kehtestamine arenduse dokumentidele, koodile, koodi ülevaatusel, kasutajaliidesele ja testimisele.

- 17.3 Kehtestab protsessid: arendusdokumentatsiooni ülevaatus, koodi ülevaatus, etappide määramine, probleemide haldus, lähtekoodi haldus, muudatuste haldus, versioonihaldus ja hooldusülesanded.
- 17.4 Kehtestab kvaliteedi ja toimetuleku mõõdikud projekti kontrolliks ja organisatsiooni toimivuse hindamiseks.
- 17.5 Koostab aruanded projekti kulgemise kohta ja selle vastavuse kohta plaanile.

18. Objekt- orienteeritud lähenemine süsteemianalüüsile [1]

- 18.1 Töötab OO analüüsi meeskonnas aktiivse liikmena.
- 18.2 Saab aru, kuidas OO paradigma süsteemi projekteerimisel erineb teistest lähenemistest.
- 18.3 Kasutab peamisi OO analüüsi ja modelleerimise tüüpe ning näitab kuidas nad on omavahel seotud.
- 18.4 Hindab OO lähenemise eeliseid analüüsis (äri ja süsteemid).
- 18.5 Saab aru OO mudelite tüüpide kasutamisest UML-s.
- 18.6 Kasutab UML analüüsi mudelit.
- 18.7 Kasutab ärikeskset modelleerimist.
- 18.8 Kasutab tegevuste modelleerimist.
- 18.9 Koostab kasutusjuhtumeid (*Use Cases*) vajaduste nimistu kogumise ajal.
- 18.10 Saab aru UML dünaamilise modelleerimise tehnikatest (STD-d, jada- ja koostöödiagrammid)).
- 18.11 Saab aru UML projekteerimise ja arhitektuuri modelleerimisest.
- 18.12 Hindab OO elutsüklit ja arenduskeskkonda ärilisest vaatepunktist.

19. Äri lahenduste projekteerimine ja juurutamine toimingute toetamisega [1]

- 19.1 Analüüsib arhitektuurist tulenevaid nõudeid. Analüüsib turvalisusnõudeid.
- 19.2 Analüüsib integratsiooninõudeid. Analüüsib funktsionaalseid nõudeid.
- 19.3 Arendab rakendusi lähtuvalt äri protsessidest.
- 19.4 Rakendab ärireegleid.
- 19.5 Haldab erandeid ja vigu.
- 19.6 Oskab luua ja hallata andmebaasiobjekte.
- 19.7 Oskab esitleda äriandmeid.
- 19.8 Koostab käepärased ja dünaamilised päringud.
- 19.9 Eelistab vahehindamisi.
- 19.10 Haldab äriprotsesside toiminguid.
- 19.11 Ühendab äriteenused ja andmemudelid.
- 19.12 Laiendab ja asendab äriprotsesse.
- 19.13 Kasutab ärirakendusi.
- 19.14 Koostab JSP-lehekülgi klientide ärirakendustele.
- 19.15 Toetub parimatele kogemustele.
- 19.16 Häälestab süsteemi tootlikust.

20. Andmebaaside projekteerimine ja juurutamine [1]

- 20.1 Valib erinevate andmebaasi-arhitektuuride seast sobiliku vastavalt rakenduse nõuetele (relatsiooniline, hierarhiline, maatriks, objekt-orienteeritud).

- 20.2 Kasutab andmete abstraherimist: füüsilisel tasemel, kontseptuaalsel tasemel ja vaate tasemel, objektipõhist loogilist mudelit, kirjepõhist loogilist mudelit ja füüsilist andmemudelit.
- 20.3 Rakendab objektipõhise loogilise mudeli põhimõtet: olemisuhte (*entity-relationship*) mudel, objekt-orienteeritud mudel.
- 20.4 Määrab olemid: olemit kompositsioon ja normaliseerimine.
- 20.5 Projekteerib olemit võtmed: FOREIGN KEY piirangud, PRIMARY KEY piirangud, UNIQUE piirangud.
- 20.6 Projekteerib atribuutide kogumi ühtsuse: CHECK piirangud, andmetüübid ja nullitavus (*nullability*).
- 20.7 Teab, kuidas kasutada andmebaasi projekteerimise vahendeid. Näited: Oracle Designer 2000, ERWin.
- 20.8 Juurutab reaalse andmebaasi.
- 20.9 Loob ja muudab andmebaase: failigrupid, failide paigutus, laienemise strateegia ja mäluvajadused.
- 20.10 Loob ja muudab andmebaasiobjekte: objektid piirangutega, indeksid, salvestatud protseduurid, tabelid, triggerid, kasutajafunktsioonid ja vaated.
- 20.11 Muudab andmebaasiobjekte, toetamaks replikatsiooni ja osalisi vaateid.
- 20.12 Objekti loomise ebaõnnestumise veaotsing.