Tampereen yliopisto

Milla Valio

ITSEARVIOINNIN HYÖDYNTÄMINEN OHJELMOINNIN OPETUKSESSA

Informaatioteknologian ja viestinnän tiedekunta
Kandidaatintyö
Toukokuu 2021

TIIVISTELMÄ

Milla Valio: Itsearvioinnin hyödyntäminen ohjelmoinnin opetuksessa
Kandidaatintyö
Tampereen yliopisto
Tieto- ja sähkötekniikan tutkinto-ohjelma
Toukokuu 2021

Ohjelmoinnin osaajien tarve kasvaa jatkuvasti ja suuremmat opiskelijamäärät ohjelmoinnin kursseilla aiheuttavat haasteita arvioinnin toteuttamiseen. Itsearviointi on keskeinen arvioinnin muoto, jonka käyttäminen on kirjattu sekä perusopetuslakiin että lukiolakiin. Tässä työssä tutkitaan, miten itsearviointia voidaan hyödyntää ohjelmoinnin opetuksessa. Työssä selvitetään, voidaanko itsearviointia integroida ohjelmoinnin opetuksessa käytössä oleviin automatisoituihin arviointityökaluihin niin, että sen käyttäminen parantaa opiskelijoiden oppimistuloksia ja antaa valmiuksia ohjelmistotuotannossa työskentelemiseen.

Kriteeriperusteinen itsearviointiprosessi on sykli, jonka aikana opiskelija tunnistaa oman osaamisensa, vertaa sitä annettuihin kriteereihin ja parantaa osaamistaan heikoiksi havaitsemissaan osissa. Ohjelmoinnin opetuksessa on käytössä lukemattomia arviointikriteereitä, joiden suhteen opiskelijan osaamista voidaan arvioida. Kun pyritään parantamaan oppimistuloksia, on tärkeää tunnistaa niistä eniten haasteita aiheuttavat. Tutkimuksen mukaan opiskelijat kokevat haastavimmiksi suurien kokonaisuuksien hahmottamista vaativat tehtävät.

Itsearviointia hyödynnetään ohjelmistotuotannossa esimerkiksi Scrumin retrospektiivissä ja Review Boardissa. Itsearvioinnin käyttäminen ohjelmoinnin opetuksessakaan ei ole uusi asia. Opetuksessa korostuu tällä hetkellä vahvasti opettajien resursseja säästävien automatisoitujen arviointityökalujen hyödyntäminen arvioinnissa. Monet näistä työkaluista tukevat opiskelijaa oman osaamisensa arvioinnissa, vaikka itsearviointiprosessi ei aina toteudu.

Työssä havaitaan, että itsearviointia käyttämällä on onnistuttu parantamaan oppimistuloksia ohjelmoinnin opetuksessa. Itsearviointi on mahdollista ottaa osaksi jo käytössä olevia arviointimenetelmiä ja sitä voidaan hyödyntää myös silloin, kun tehtävä vaatii suurien kokonaisuuksien hahmottamista. Suurimman hyödyn saavuttamiseksi on tärkeää, että itsearviointia käytetään oikein ja että opiskelijaa tuetaan tarpeeksi itsearvioinnin toteuttamisessa. Huonosti toteutettu itsearviointi voi pahimmillaan heikentää oppimistuloksia. Itsearvioinnin käyttäminen ei myöskään vähennä opettajalta saadun henkilökohtaisen palautteen tärkeyttä.

Avainsanat: Itsearviointi, ohjelmoinnin opetus, oppimisen arviointi, automatisoidut arviointityökalut

SISÄLLYSLUETTELO

1. JOHDANTO

Maailma digitalisoituu tällä hetkellä kovaa vauhtia ja ohjelmoijien tarve kasvaa jatkuvasti. Samalla kun koulutuksella pyritään tuottamaan alalle raudankovia ammattilaisia, pitää ohjelmoinnin perusteita opettaa laajalle joukolle opiskelijoita. Suurien kurssikokojen kohdalla nousee esiin kysymys siitä, miten opiskelijoiden oppimista arvioidaan.

Itsearviointi on yksi keskeisistä arvioinnin muodosta. Sekä Suomen perusopetuslaissa että lukiolaissa todetaan samoin sanoin: ”oppilaan arvioinnilla pyritään ohjaamaan ja kannustamaan opiskelua sekä kehittämään oppilaan edellytyksiä itsearviointiin.” [1, 2] Yliopistolaissa tai ammattikorkeakoululaissa itsearviointia ei mainita [3, 4]. Itsearvioinnin käyttämisellä perusopetuksessa ja toisella asteella on tarkoituksena luoda opiskelijoille tarvittavat taidot oman osaamisensa arviointiin [5]. Jos perusopetuslain ja lukiolain mukaisesti opiskelijalle on kehittynyt kyky itsearviointiin, on tämän taidon hyödyntäminen mahdollista ja järkevää myös seuraavalla oppiasteella.

Tämän työn on tarkoitus tutustuttaa lukija hyötyihin, joita itsearvioinnin hyvällä käyttämisellä voidaan tuoda ohjelmoinnin opetukseen. Työssä etsitään keinoja integroida itsearvioinnin käyttö ohjelmoinnin opetukseen niin, että se tukee opiskelijoita oppimisessa ja antaa valmiuksia työskennellä ohjelmistotuotannossa. Työssä ei pyritä kehittämään uusia valmiita opetusmenetelmiä, vaan keräämään perustietoa ja esittelemään jo olemassa olevia toimivaksi todistettuja tapoja hyödyntää itsearviointia ohjelmoinnin opetuksessa.

Suomen kielen itsearvioinnille on olemassa monta englanninkielistä vastinetta, yleisimmin käytössä ovat *self-evaluation* ja *self-assessment*. *Self-evaluation* viittaa usein arviointiin, jossa opiskelija saa itse vaikuttaa saamaansa arvosanaan. *Self-assessment* viittaa enemmän siihen, että on olemassa kriteeristö, johon opiskelija vertaa omaa osaamistaan. [5]

Tässä työssä itsearvioinnilla tarkoitetaan arvioinnin muotoa, jossa opiskelija vertaa itse ilman opettajalta saatua palautetta suoritustaan annettuun tavoitteeseen tai malliesimerkkiin. Kuten aiemmin todettiin, itsearvioinnilla voidaan tarkoittaa myös opiskelijan omalle työlleen suorittamaa numeerista arviointia, mutta tässä työssä keskitytään itsearvioinnin käyttöön oppimisen välineenä arvosanojen antamisen sijaan.

Työn alussa luvuissa kaksi ja kolme tehdään katsaus itsearvioinnin perusteisiin ja käydään läpi ohjelmoinnin opetuksessa yleisesti käytössä olevia arviointikriteereitä sekä ohjelmoinnin oppimisessa yleisimmin esiin nousevia ongelmia. Tämän jälkeen luvussa neljä tehdään lyhyt katsaus muutamaan ohjelmistotuotannossa käytössä olevaan menetelmään, joissa hyödynnetään itsearviointia. Luvussa viisi tutustutaan laajasti käytössä olevien automaattisten arviointityökalujen tuottamiin mahdollisuuksiin itsearviointiin liittyen. Ennen yhteenvetoa päädytään siihen, millä konkreettisilla keinoilla itsearviointia voidaan sisällyttää automaattisiin arviointityökaluihin ohjelmoinnin opetuksessa.

2. ITSEARVIOINTI ARVIOINTITYÖKALUNA

Palautteen saamisella on keskeinen merkitys oppimisessa [6]. Parhaassa tapauksessa opiskelija saisi jatkuvaa palautetta opettajalta, joka on opettamansa alan ammattilainen. Opettajalla kuitenkin on vain harvoin aikaa keskustella säännöllisesti kaikkien opiskelijoidensa kanssa näiden henkilökohtaisesta oppimisesta. Tutkimus on osoittanut, että opiskelija voi toimia hyödyllisenä palautteen lähteenä itselleen. [6, 7]

Itsearvioinnilla tarkoitetaan arvioinnin muotoa, jossa opiskelija arvioi omaa oppimistaan. Erityisesti siinä keskitytään saavutusten ja oppimistulosten arviointiin. Useassa eri tutkimuksessa on argumentoitu, että korkeakouluopetuksen tarkoitus on kehittää opiskelijan itsearviointitaitoja [8–10].

Eri-ikäisiä opiskelijoita ja eri oppiaineita käsittelevissä tutkimuksissa on löydetty vahvaa näyttöä siitä, että itsearviointi edistää opiskelijan oppimista [11]. Samalla havaittiin, että vaikutukset kasvavat, mikäli opiskelijalle tarjotaan selkeä ohje itsearvioinnin suorittamiseen. Itsearvioinnin vaikutuksiin keskittyvien tutkimusten tulokset kuitenkin vaihtelevat. Toisen useita eri itsearviointiin keskittyviä tutkimuksia tutkineen meta-analyysin mukaan itsearviointi parantaa opiskelijan oppimista ja suoriutumista, mutta myös negatiivisia vaikutuksia voi ilmetä. [12]

Heikosti motivoituneiden opiskelijoiden kohdalla itsearvioinnin käyttäminen ei paranna oppimistuloksia yhtä paljon kuin motivoituneilla opiskelijoilla [7]. Jotta itsearviointi ei vaikuta negatiivisesti oppimistuloksiin, tulee arviointi suunnitella hyvin ja toteuttamisessa miettiä tarkasti itsearvioinnin avulla saatavia hyötyjä ja haittoja. On tärkeää, että opiskelija tunnistaa itsearvioinnin tarjoavan mahdollisuuksia tehdä todellisia parannuksia ja mahdollisesti myös parantaa arvosanaa.

Itsearvioinnin toteuttamiseen on olemassa lukemattomia tapoja ja menetelmiä. Tutkimuksessa on eroteltu kolme erilaista tapaa tuoda itsearviointi osaksi arviointikulttuuria. Nämä tavat ovat strukturoitu itsearviointi, integroitu itsearviointi sekä ohjaava itsearviointi [13]. Seuraavaksi esitellään edellä luetellut itsearviointimenetelmät, jonka jälkeen tutustutaan itsearviointiprosessiin ja sitä hyödyntävään itsearviointimalliin.

Ei ole olemassa yksiselitteistä rajaa, joka erottaa itsearviointimenetelmän muista arviointimenetelmistä. Rajan epämääräisyyden vuoksi suorien itsearviointimenetelmien lisäksi esitellään myös itsearviointiin erittäin läheisesti liittyvät arviointitavat vertaisarviointi ja yhteisarviointi.

2.1 Itsearviointimenetelmät

Opetuksessa yleisimmin käytetty itsearviointitapa on strukturoitu itsearviointi. Siinä opettaja suunnittelee itsearviointilomakkeen, johon opiskelijat vastaavat kirjallisesti. Itsearviointilomake perustuu oppimistavoitteisiin tai vaatimuksiin, jotka on asetettu ennen suorituksen aloittamista. Itsearviointilomaketta voidaan käyttää esimerkiksi yhden kappaleen tai projektin osan suorittamisen jälkeen. [5]

Strukturoidun itsearvioinnin suorittamisen jälkeen sekä opiskelijalla että opettajalla on lista asioista, jotka opiskelija kokee osaavansa ja joita hän ei vielä koe osaavansa. Opettaja näkee listasta asiat, joiden oppiminen on ollut heikompaa ja pystyy kehittämään opetustaan tämän tiedon perusteella. Opiskelijalle strukturoitu itsearviointi tarjoaa listan asioista, joita hän ei vielä osaa. Lista ei kuitenkaan ohjaa tehokkaasti opiskelijaa oppimaan lisää hänelle hankalaksi jääneistä aiheista.

Toinen tapa toteuttaa itsearviointia on integroitu itsearviointi, jossa arviointi sisällytetään osaksi muuta arviointia. Tarkoituksena on integroida edellä käsitellylle strukturoidulle itsearvioinnille tyypillinen itsearviointilomake osaksi muuta arviointitilannetta, joka voi olla esimerkiksi harjoitusprojekti tai koetilanne. Integroidussa itsearvioinnissa pyritään saamaan itsearviointi luontevaksi osaksi arviointia. [5]

Integroidussa itsearvioinnissa toistuvat samat ongelmat kuin strukturoidussa itsearvioinnissa, sillä kyse on käytännössä samasta arvioinnista eri tilanteessa. Myöskään integroitu itsearviointi ei itsessään tue opiskelijaa kehittämään heikkouksiaan. Lisäksi itsearvioinnin hyödyllisyys ja toimivuus voivat vaihdella sen mukaan, millaisiin tilanteisiin itsearviointia sisällytetään. Esimerkiksi kokeen loppuun sijoitetulla itsearviointilomakkeella saadut tulokset eivät välttämättä ole yhtä luotettavia kuin erillisenä täytetyn itsearviointilomakkeen, sillä kokeen täyttämisen jälkeen opiskelija voi olla väsynyt ja mahdollisesti myös kiireessä koeajan ollessa rajattu.

Kolmas tapa toteuttaa itsearviointia on ohjaava itsearviointi. Se on strukturoitua itsearviointia vapaamuotoisempi itsearvioinnin muoto, jossa pyritään saamaan opiskelija reflektoimaan omaa oppimistaan vapaamuotoisemmin. Ohjaava itsearviointi on yleensä improvisoitua ja henkilökohtaista. [5]

Ohjaavan itsearvioinnin ei tarvitse olla monimutkaista. Kun opettaja kysyy tehtävää suorittavalta opiskelijalta, miten työskentely sujuu, hän ohjaa opiskelijaa arvioimaan omaa osaamistaan. Ohjaavan itsearvioinnin hyöty on siinä, että se avaa suoran keskusteluyhteyden opettajan ja opiskelijan välille. Opiskelijan kertoessa itselleen

hankalista asioista, on opettajan helppo reagoida opiskelijan haasteisiin ja auttaa tätä välittömästi oppimisessa. Opettajan on myös helppo tukea opiskelijaa itsearvioinnin suorittamisessa kysymällä tarkentavia kysymyksiä

Kaikissa käsitellyissä itsearviointimenetelmissä on haasteena arvioinnin jatkuvuuden takaaminen. Ongelman ratkaisuksi on kehitetty itsearviointiprosessi, jonka tarkoituksena on varmistaa itsearvioinnin pysyminen jatkuvana osana opetusta ja oppimista.

2.2 Itsearviointiprosessi

Kriteeriperusteinen itsearviointi on prosessi, jonka aikana opiskelija kerää tietoa omasta suorituksestaan tai edistymisestään, vertaa sitä asetettuihin kriteereihin tai tavoitteisiin ja tämän jälkeen muokkaa omaa suoritustaan tai opiskeluaan sen mukaisesti. Hyvin toimivassa itsearviointiprosessissa opiskelija käy syklin säännöllisesti läpi oppimisen aikana kehittäen osaamistaan tai projektiaan jokaisella kerralla eteenpäin.

Toimiva opiskelijoita sitouttava itsearviointi voidaan toteuttaa lukemattomilla erilaisilla tavoilla. Yleisesti itsearviointiin kannustava prosessi koostuu kolmesta osasta; kriteerien ilmaisusta, oman työn vertaamisesta kriteereihin ja havaittujen muutosten tekemisestä. [7, 11]

Ensimmäiseksi opiskelijalle pitää selkeästi ilmaista tehtävän kriteerit. Tämän voi tehdä joko opettaja, opiskelija tai molemmat yhdessä. Jos opiskelija osallistuu itse kriteerien miettimiseen, tutustuu hän samalla tehtävään ja pystyy paremmin tunnistamaan heikon ja hyvän lopputuloksen. Kriteerien määrittämisen lisäksi tulee opiskelijalle opettaa, milloin kriteeri katsotaan täytetyksi.

Toisena osana prosessissa opiskelijat luovat ensimmäisen luonnoksen projektistaan tai pohtivat osaamistaan alussa. Itsearviointi aloitetaan vertaamalla omaa suoritusta annettuihin kriteereihin ja merkitsemällä, mitkä odotukset täyttävät. Seuraavaksi opiskelija poimii kriteereistä ne, jotka eivät vielä täyty ja tekee suunnitelman siitä, miten ne tullaan täyttämään. Tässä kohtaa on tärkeää, että opiskelijalle annetaan myös palautetta itsearvioinnin suorittamisesta ja riittävästi aikaa arvioinnin suorittamiseen.

Kolmantena osana itsearviointiin kannustavassa prosessissa opiskelija käyttää keräämänsä tiedon projektin tai osaamisen parantamiseen. Tämä osa on kriittistä, sillä oman palautteen hyödyntäminen oman työn parantamiseen on tärkein osa itsearviointiprosessia [11]. Myös tässä kohtaa prosessia on tärkeää antaa opiskelijalle

apua ja tukea itsearvioinnin tulosten hyödyntämiseen. Tärkeää on myös muistaa, että tarkoituksena ei ole ottaa itsearviointia osaksi numeraalista arviointia.

Prosessin toisen ja kolmannen osan on tarkoitus toistua jatkuvina projektin oppimisen aikana. Mikäli projektin kriteereissä ilmenee epäselvyyksiä, tarvittaessa voidaan palata myös ensimmäiseen kohtaan eli odotusten ilmaisemiseen. Itsearviointiprosessin aikana opiskelija vertaa useaan kertaan osaamistaan kriteereihin ja saa näin mahdollisuuden huomata konkreettisesti oman osaamisensa paranemisen.

Itsearviointiprosessin toteuttamisen helpottamiseksi on kehitetty useita eri malleja. Nämä mallit pyrkivät ohjaamaan itsearvioinnin tehokkaaseen käyttöön. Itsearvioinnin käyttämistä varten on kehitetty lukuisia eri malleja, joilla pyritään ohjaamaan itsearvioinnin tehokkaaseen käyttöön. Eräs tällainen malli on Helsingin yliopistossa kehitetty DISA-malli [14].

2.3 Esimerkki itsearviointimallista

DISA eli Digital Self-Assessment on Helsingin yliopistossa kehitetty itsearviointiin perustuva arviointimalli, joka on tarkoitettu suurille kursseille. DISA-mallissa opiskelijat antavat itselleen sekä kurssiarvosanan että arvioivat jatkuvasti osaamistaan kurssin aikana. Lopputenttiä ei järjestetä. Opiskelijan itsearviointia verrataan automaattisen työkalun avulla opiskelijan edellisiin suorituksiin huijaamisen estämiseksi.

Itsearviointi on DISA-mallia hyödyntävillä kursseilla mukana koko kurssin ajan. Opiskelijat harjoittelevat itsearvioinnin suorittamista koko kurssin ja he saavat sekä automaattisesti luotua että opettajan antamaa palautetta omista arviointitaidoistaan.

Pilottikursseilla saatujen tulosten mukaan DISA-mallin käyttäminen paransi opiskelijoiden motivaatiota ja sitoutumista opiskeluun, lisäsi uskoa omiin kykyihin ja muutti opiskelun tenttipainotteisesta oppimisesta itseä varten oppimiseen. [15]

DISA-malli seuraa itsearviointiprosessia. Opiskelijalle kerrotaan kurssin aikana mitä hänen odotetaan oppivan kustakin aiheesta. Kurssin kuluessa opiskelija arvioi osaamistaan useaan kertaan ja saa palautetta arvioinnin suorittamisesta.

DISA-mallin käytöstä ohjelmoinnin kursseilla ei ole vielä tehty tutkimuksia. Ohjelmoinnissa painottuvat kuitenkin samanlaiset taidot kuin matematiikassa; looginen ja abstrakti ajattelu, tietyn notaation käyttäminen ja yksityiskohtien tärkeys.

2.4 Vertaisarviointi ja yhteisarviointi

Itsearvioinnin kanssa läheisiä arviointimuotoja ovat vertaisarviointi ja yhteisarviointi (*engl. co-assessment*). Niissä molemmissa hyödynnetään opiskelijan kykyä arvioida omaa työtään tai vastaavaa työtä.

Vertaisarvioinnissa opiskelija arvioi toisen opiskelijan työtä ja antaa siitä palautetta samalla saaden myös omasta työstään palautetta. Kuten itsearviointi, myös vertaisarviointi auttaa opiskelijaa kehittämään arviointitaitoja, vaatii syvällistä ymmärtämistä käsiteltävänä olevasta asiasta ja tukee opiskelijan kykyä arvioida omaa oppimistaan ja edistymistä. Tietojenkäsittelyoppiin ja ohjelmistotekniikkaan keskittyneen tutkimuksen mukaan hyvin toteutettu vertaisarviointi tarjoaa mahdollisuuden maksimoida oppimistuloksen. [16]

Yhteisarvioinnissa opiskelija ja opettaja tapaavat ja keskustelevat tehtävän vaatimuksista. [8] Yhteisarvioinnissa yhdistyvät sekä opiskelijan itselleen antama palaute että opettajan antama palaute. Aiemmin todettiin opettajan antaman palautteen olevan kriittinen osa opiskelijan oppimisprosessia, joten yhteisarvioinnissa saadaan parhaassa tilanteessa hyödynnettyä molempien arviointitapojen parhaat puolet.

3. OHJELMOINNIN ARVIOINTI JA HAASTEET

Ohjelmoinnin aloittamista voidaan verrata uuden vieraan kielen oppimiseen. Käytettävät käsitteet, rakenteet ja syntaksit ovat vieraita. Kielen oppimisen lisäksi ohjelmoinnissa yhdistyy ongelmanratkaisukyky ja looginen ajattelu. Kaiken uuden tiedon hallitsemiseksi opiskelijalta vaaditaan kiinnostusta oppimiseen sekä taitoa hahmottaa suuria kokonaisuuksia.

Tässä luvussa tarkastellaan kriteereitä, joita voidaan arvioida ohjelmoinnin opetuksessa. Lisäksi tehdään katsaus siihen, mitkä asiat opiskelijat kokevat kaikkein haastavimmiksi ohjelmointia opeteltaessa, jotta myöhemmin pystytään etsimään ratkaisuja itsearvioinnin hyödyntämiseen juuri oppimisen kannalta kriittisten asioiden kohdalla.

3.1 Arviointikriteerit

Ohjelmoinnin opetuksessa mahdollisia arvioitavia asioita on lukemattomia. Tampereen yliopiston Informaatioteknologian ja viestinnän tiedekunnan ensimmäinen ohjelmoinnin perusopintojakso on Ohjelmointi 1: Johdatus ohjelmointiin. Opintojakson oppimistavoitteina lukuvuonna 2020–2021 on, että ”opintojakson suoritettuaan opiskelija

- hallitsee ohjelmoinnin peruskäsitteet (muuttujat ja ohjausrakenteet)
- osaa jakaa ohjelman funktioihin
- tuntee joitakin yksinkertaisia tietorakenteita
- osaa muodostaa monimutkaisempia tietorakenteita yksinkertaisemmista rakenteista
- osaa ratkaista pieniä ongelmia ohjelmoimalla
- hallitsee hyvän ohjelmointitavan perusteet
- tuntee olio-ohjelmoinnin peruskäsitteet (luokka ja olio)
- tuntee graafisen käyttöliittymän alkeita” [17]

Oppimistavoitteissa esiintyy sekä syntaksien oikea käyttö että ohjelman rakenteeseen liittyviä tavoitteita. Opiskelijalta vaaditaan ohjelmointikielen perusteiden hallintaa ja hyvän ohjelmointitavan tuntemista.

Yleisesti koodista voidaan arvioida luettavuutta, mukautettavuutta, suoritusnopeutta, turvallisuutta ja yhteensopivuutta. Konkreettisempina arviointikriteereinä voidaan käyttää luokkien, metodien ja muuttujien järkevää nimeämistä, koodin asettelua, kattavia yksikkötestejä, kommenttien määrää ja sisältöä sekä koko ohjelman rakenteen tehokkuutta. Ohjelman ulkopuolisena voidaan arvioida projektinhallintaa, aikataulussa pysymistä ja yhteistyötä. Mahdollisten arviointikohteiden listaa voi jatkaa melkein loputtomasti.

3.2 Suurimmat haasteet oppimisessa

Yli 550 ohjelmointiopiskelijan kyselytutkimuksen mukaan opiskelijat kokevat ohjelmoinnissa suurimmiksi haasteiksi

- hahmottaa, miten suunnitella ohjelma tietyn tehtävän ratkaisemiseksi
- jakaa toiminnallisuudet proseduureihin ja
- löytää ohjelmointivirhe omasta ohjelmasta.

Samassa tutkimuksessa havaittiin, että opiskelijat kokivat yksin opiskelun ja kurssityön parissa työskentelyn luentoja ja harjoitustilaisuuksia hyödyllisemmiksi. [18] Kaikki yllä listatut ongelmat vaativat suurien kokonaisuuksien hahmottamista pieniin yksityiskohtiin keskittymisen sijaan.

Opetusmuodoista hyödyllisimmäksi osoittautuivat tavat, joissa opiskelijat pääsevät itse ratkomaan tehtäviä ja ongelmia. Itse tekeminen on myös havaittu tärkeäksi oppimisen kannalta. Opettajan huolellisesti suunnittelemat materiaalit sekä lähestymistavat ohjaavat opiskelijaa tietämyksen ja taitojen rakentamisessa. [18]

4. OHJELMOINNISSA HYÖDYNNETTÄVIÄ ITSEARVIOINTIMENELMIÄ

Ohjelmistokehityksessä on yleisesti käytössä useita menetelmiä, jotka hyödyntävät itsearviointia. Tässä työssä niistä esitellään Scrumin retrospektiivi ja Review Board. Molemmissa esiteltävissä menetelmissä hyödynnetään itsearvioinnin lisäksi myös vertaisarviointia. Nämä kaksi kulkevat monesti yhdessä ja luovat yhdessä vahvan pohjan ohjelmiston ja kehitysprosessin arviointiin.

4.1 Scrumin retrospektiivi

Scrum on ketterän ohjelmistokehityksen menetelmä, joka on laajasti käytössä ohjelmistotuotannossa. Sen perustana ovat noin kuukauden mittaiset sprintit (*engl. sprint*), joiden aikana tuotetta kehitetään iteratiivisesti. Jokainen sprintti päättyy retrospektiiviin (*engl. Sprint Retrospective Meeting*), joka on enintään kolme tuntia kestävä tapaaminen.

Retrospektiivissä ohjelmistoa kehittämässä ollut scrumtiimi (*engl. Scrum Team*) arvioi Scrum-prosessista vastaava scrummasterin (*engl. Scrum Master*) johdolla, miten edellinen sprintti sujui. Tapaamisessa hyödynnetään sekä oman että koko scrumtiimin toiminnan itsearviointia. Tyypillisiä kysymyksiä ovat mikä meni hyvin edellisessä sprintissä, ja mitä voitaisiin parantaa seuraavaan sprinttiin. Vastaukset kirjoitetaan muistiin ja parannusehdotuksista priorisoidaan tärkeimmät ongelmat ensin korjattavaksi. Retrospektiivin päätarkoituksena on etsiä konkreettisia parannuksia Scrum-prosessiin. [19] Se ei siis keskity itse ohjelmoinnin arviointiin vaan prosessin arvioimiseen ja kehittämiseen.

4.2 Review Board

Review Board on internetpohjainen koodinarviointityökalu, joka on vapaasti saatavilla sekä yksityiseen että kaupalliseen käyttöön. Review Boardin tarkoituksena on toimia alustana koodin itse- ja vertaisarviointiin. [20]

Ohjelmoija lataa kirjoittamansa koodin Review Boardiin. Hän voi halutessaan aloittaa arviointiprosessin arvioimalla itse ensin omaa koodiaan ja tarjoamalla kommentteja lähtöpisteeksi muille arvioijille. Tämän jälkeen muilla projektiin osallistuvilla on mahdollisuus lisätä kommentteja koodiin joko yleisesti tai tiettyyn osaan liittyen. Arvioijat

voivat myös kannattaa muiden tekemiä muutosehdotuksia. Lisäksi arvioija voi merkitä mielestään kriittisimmät kommentit tärkeiksi. [20] Review Boardin käyttäminen mahdollistaa koodin arvioimisen itsearviointiprosessin mukaisesti iteroiden.

Review Boardia voi hyödyntää sekä ennen versionhallintaan tallettamista (*engl. commit*) että tallettamisen jälkeen koodin arvioimiseen. Review Board toimii myös yhdessä monien muiden palveluiden kanssa. Näitä ovat esimerkiksi versionhallintaan käytettävä Git, kommunikointisovellukset Mattermost ja Slack sekä projektinhallintatyökalu Trello. [21]

5. AUTOMAATTISET ARVIOINTITYÖKALUT JA ITSEARVIOINTI

Ohjelmoidessa suoritetaan jatkuvaa itsearviointia. Ohjelmoija kirjoittaa palan koodia ja sen jälkeen kokeilee sen toimivuutta. Nykyaikainen ohjelmointikielen kääntäjä osaa tarjota ohjelmoijalle palautteena tietoa ohjelmassa mahdollisesti olevista virheistä. Ohjelmoijan tehtäväksi jää tulkita virheilmoitukset, arvioida niiden perusteella tarvittavat muutokset ja muokata koodia niiden mukaisesti. Virheilmoitusten lisäksi kääntäjä saattaa antaa ohjelmasta varoituksia, joiden tehtävänä on auttaa ohjelmoijaa korjaamaan suorien virheiden lisäksi myös ohjelman huonosti toteutettuja osia.

Vaikka ohjelmointi onkin jatkuvaa itsearviointia, voidaan ohjelmoinnin opetuksessa käyttää myös työkaluja, jotka auttavat opiskelijaa prosessissa ja tarjoavat tarkempaa palautetta.

5.1 Automaattiset arviointityökalut

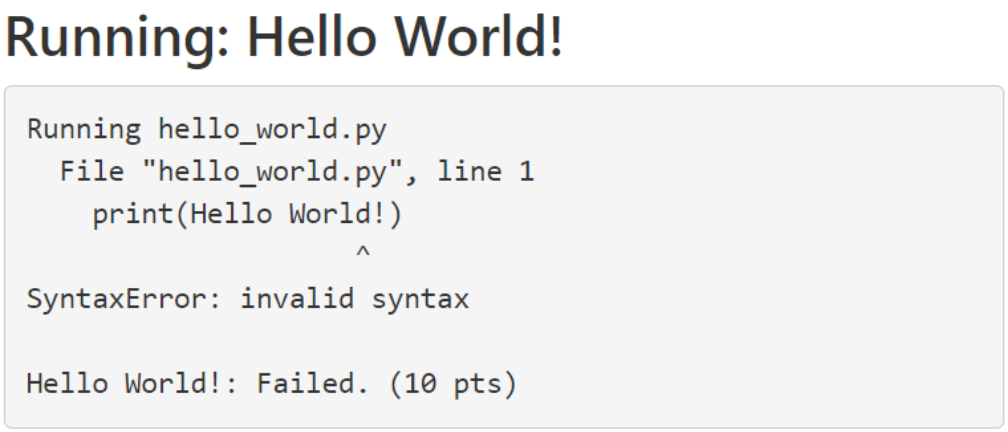
Tietokoneavusteiset arviointityökalut (*engl. computer-assisted assessment*) tarjoavat lukuisia hyötyjä arviointiin. Niiden käyttäminen säästää opettajan resursseja ja tarjoaa opiskelijoille virheetöntä, objektiivista ja puolueetonta palautetta, joka on saatavilla vuorokauden ympäri ja missä tahansa Internetin ulottuvilla. [22, 23] Automatisoituja arviointityökaluja on olemassa lukemattomia.

Kirsti M. Ala-Mutkan kartoituksen mukaan automaattiset arviointityökalut arvioivat kirjallisuudessa raportoidusti sekä dynaamisia että staattisia osia. Nämä sisältävät esimerkiksi ohjelman toiminnallisuudet, tehokkuuden, koodaustyylin ja suunnittelun. [23]

Automaattiset arviointityökalut eivät aina sisällä suoraa itsearviointia, mutta ne antavat opiskelijalle mahdollisuuden saada enemmän palautetta omasta työstään ja vihjeitä oikean ratkaisun löytämiseen. Palautteen perusteella opiskelija voi arvioida omaa suoritustaan ja tehdä siihen tarvittavia muutoksia sekä korjauksia.

Vaikka automaattiset arviointityökalut voivatkin tukea oppimista, ne saattavat myös vaikuttaa negatiivisesti opiskelijan oppimisstrategioihin. Jos automaattinen arviointityökalu on helposti saatavilla, opiskelija saattaa käyttää sitä sen sijaan että testaisi itse ohjelmaansa. [23] Tämän takia opettajan tulee suunnitella tehtävä niin, että opiskelijaa kannustetaan oppimaan ja testaamaan itse omaa ohjelmaansa.

Tampereen yliopistossa käytetään monilla ohjelmointikursseilla Plussa-oppimisympäristöä, jonka ominaisuuksiin kuuluu automaattinen arvioija. Arvioija ajaa opiskelijan ohjelman ja vertaa opiskelijan palauttamaa ohjelmaa opettajan arvioijalle antamaan malliohjelmaan. Plussassa opettajalla on mahdollisuus määrittää tehtävälle maksimipalautusmäärä, joka rajaa opiskelijalla käytössä olevien palautusten määrän. Maksimimäärä kannustaa opiskelijaa oman ohjelman testaamiseen ja käyttämään automaattista arvioijaa vasta viimeisenä tarkastajana palautusta tehdessä.



Kuva 1. *Kuvakaappaus Plussa-oppimisympäristön automaattisen arvioijan palautteesta.*

Mikäli opiskelijan ohjelma ei toimi malliohjelman tavoin, tarjoaa arvioija opiskelijalle palautetta mahdollisista virheistä. Esimerkki automaattisen arvioijan palautteesta on esitetty kuvassa 1, jossa arvioija huomaa ohjelmassa olevan syntaksivirheen ja osoittaa sen sijainnin opiskelijalle.

5.1.1 Mustalaatikkotestaus

Mustalaatikkotestauksessa (*engl. black-box testing*) tarkastellaan, tuottaako ohjelma annetuilla syötteillä halutun tuloksen. Ohjelman sisäistä rakennetta ei tarkastella. Itsearvioinnilla on testauksessa tärkeä rooli, sillä ohjelman kirjoittajan tulee verrata oman ohjelmansa tulosta odotettuun tulokseen. [24]

Mustalaatikkotestauksen avulla voidaan luoda yksinkertaisia ohjelmointiharjoituksia, joissa opiskelijalle annetaan haluttu tulos ja mahdollinen syöte. Näissä tilanteissa tehtävä on tarkasti määritetty ja itsearvioinnin suorittaminen oman ja malliohjelman tulosta vertaamalla on melko yksikäsitteistä. Ohjelmaa voidaan testata myös muilla kuin esimerkkinä annetuilla syötteillä, jolloin opiskelijan tulee myös kehittää itse testausehtoja ja arvioida tarkemmin oman ohjelmansa toimintaa.

Ohjelmoinnin opetuksen lisäksi mustalaatikkotestausta voidaan käyttää myös ohjelmistotuotannossa ohjelman toiminnan arviointiin. Näissä tilanteissa haluttua tulosta ei aina ole yksiselitteisesti määritetty, jolloin oman ohjelman arvioinnin rooli kasvaa.

Kuva 1. *Kuvakaappaus Plussa-oppimisympäristön arvioijan palautteesta*

Kuvassa 2 esitetään yksinkertainen esimerkki, jossa arvioija ajaa opiskelijan palauttaman hello_world.py-tiedoston. Diff-käsky vertaa opettajan syöttämää mallitulostetta opiskelijan ohjelmaan. Arvioija korostaa opiskelijan ohjelman tuottaman virheellisen tuloste korostetaan selkeyden vuoksi. Lopuksi arvioija kertoo, että tehtävä on epäonnistunut.

Mustalaatikkotestauksessa opiskelija saa palautetta ainoastaan ohjelman tulosteen yhdenmukaisuudesta tehtävän kriteerien kanssa. Ohjelmointia opeteltaessa on kuitenkin tärkeää arvioida myös ohjelman sisäistä toimintaa. Tähän tarkoitukseen voidaan käyttää lasilaatikkotestausta.

5.1.2 Lasilaatikkotestaus

Lasilaatikkotestauksessa tarkastellaan mustalaatikkotestauksen tapaan, saadaanko annetulla syötteellä haluttu tulos. Mustalaatikkotestauksesta poiketen lasilaatikkotestauksessa arvioidaan myös ohjelman sisäistä rakennetta. [24]

Mustalaatikkotestausta hyödyntävässä tehtävässä tehtävänannossa kerrotaan haluttu tuloste tietyllä syötteellä. Lasilaatikkotestauksessa voidaan syötteen ja tulosteen lisäksi tehtävän kriteereissä päättää, miten ohjelman sisäinen rakenne tulee toteuttaa. Nämä ohjeet voivat esimerkiksi määrätä, että mitä toistorakennetta tulee käyttää tai että funktion tulee olla rekursiivinen.

Käsitellään yksinkertaista Python-kielistä ohjelmaa, jonka tehtävänä on kysyä käyttäjältä, kuinka monta numeroa tämä haluaa tulostettavan ja tämän jälkeen tulostaa allekkain alkaen numerosta 1 kaikki kokonaisnumerot annettuun numeroon asti. Tehtävä voidaan ratkaista helposti käyttämällä while-rakennetta tai for-silmukkaa.

```
def main():
    syote = input("Kuinka monta numeroa haluat? ")
    vastaus = int(syote)

    for i in range(vastaus):
        print(i+1)


main()
```

Ohjelma 1. *Ohjelma, jossa silmukka on toteutettu for-silmukalla*

Ohjelma 1 näyttää for-rakennetta käyttäen toteutetun ratkaisun. Rivillä 2 kysytään käyttäjältä tämän haluama numeromäärä. Sen jälkeen riveillä 5 ja 6 käydään for-silmukan avulla läpi kaikki numerot aina tähän numeroon asti ja tulostetaan numerot. Toteutetaan sama ohjelma myös for-silmukalla.

```
def main():
    syote = input("Kuinka monta numeroa haluat? ")
    vastaus = int(syote)
    i = 1

    while i <= vastaus:
        print(i)
        i = i + 1


main()
```

Ohjelma 2. *Ohjelma, jossa silmukka on toteutettu while-rakenteella*

Ohjelma 2 tuottaa saman tuloksen kuin ohjelma 1, mutta siinä on käytetty for-silmukan sijaan while-rakennetta rivillä 6 numeroiden läpikäyntiin. Lasilaatikkotestauksen avulla saadaan tarkistettua, että opiskelija hallitsee molemmat tavat.

5.2 Keinoja integroida itsearviointi automaattisiin arviointityökaluihin

Luvussa kaksi todettiin, että itsearvioinnin pitää olla prosessi, joka tukee opiskelijaa oman osaamisen kehittämisessä. On tärkeää, että opiskelija tunnistaa itsearvioinnin tarjoavan mahdollisuuksia tehdä todellisia parannuksia ja mahdollisesti myös parantaa arvosanaa.

Itsearviointi hyödyntää itse tekemällä oppimisen periaatetta. Itsearviointiprosessissa opiskelija on itse vastuussa tehtävän tekemisestä sekä sen parantamisesta. Opettajan roolina on toimia tukena prosessissa.

Arviointityökalu suorittaa osan itsearviointiprosessista opiskelijan puolesta. Tämä voi olla sekä hyvä että huono asia. Automaattinen arviointityökalu auttaa opiskelijaa ongelmien huomaamisessa ja mahdollisesti myös antaa vinkkejä ongelman korjaamisen. Yksinkertaisissa harjoitustehtävissä automaattinen arviointityökalu pystyy tarkistamaan kaikkien pyydettyjen toiminnallisuuksien toiminnan. Kun tehtävän laajuus kasvaa, myös arvioitavien asioiden määrä kasvaa. Näissä tilanteissa automaattinen arviointityökalu ei pysty tarkistamaan kaikkia mahdollisia virheitä ja ongelmia.

Liikaa käytettynä automaattinen arviointityökalu saattaa vähentää opiskelijan itse suorittaman itsearvioinnin määrää. Automaattinen arvioija ei ikinä voi täydellisesti löytää kaikkia ongelmia ohjelmasta. Myöskään opettajan aika ei yleensä riitä antamaan kaikille opiskelijoille säännöllistä ja yksityiskohtaista palautetta tehtävän etenemisestä. Tämän vuoksi on tärkeää, että opiskelija arvioi itse omaa työtään kriittisesti.

Automaattisten arviointityökalujen kylkeen olisi helppo lisätä integroitu strukturoitu itsearviointilomake, johon opiskelijan tulee vastata ennen palautuksen tekemistä. Itsearviointiprosessin mukaisesti opiskelijalle tulee tarjota tehtävän aluksi selkeät kriteerit, joiden mukaan tehtävä tullaan arvioimaan.

Lomakkeessa voidaan pyytää opiskelijaa arvioimaan tehtävänannon kriteerien toteutuminen omassa ohjelmassa esimerkiksi välillä 1–5. Tämän jälkeen opiskelijaa pyydetään valitsemaan muutama kriteeri, joihin tämä haluaa erityisesti keskittyä ennen seuraavaa palautusta. Näistä kriteereistä opiskelija kirjaa ylös muutaman konkreettisen asian, joiden avulla ne saadaan korjattua.

Kun opiskelija suorittaa itsearviointia seuraavan kerran, tulee hänelle näyttää edellisellä kerralla kirjaamansa kommentit ja pyytää kuvailemaan miten näiden asioiden korjaaminen onnistui. Lisäksi opiskelija täyttää itsearviointilomakkeen uudelleen ja

valitsee taas muutaman erityistä huomiota vaativan kriteerin. Tarvittaessa opiskelija voi valita myös saman kriteerin kuin edellisellä kerralla.

Vastaava itsearviointi voitaisiin suorittaa myös vapaamuotoisemmin pyytäen opiskelijaa suoraan nimeämään kehityskohteita ja keinoja korjata ne. Aiemmin esitelty Review Board perustuu tällaiseen avoimeen arviointiin. Koska opiskelija vasta opiskelee uutta asiaa, on strukturoitu arviointilomake kuitenkin todennäköisesti tehokkaampi.

Itsearvioinnin toteuttamista tulee tukea tarjoamalla opiskelijalle helposti saatavilla olevaa apua. Lomakkeen kysymykseen voi liittää linkin, joka vie opiskelijan opintomateriaalissa olevaan lisätietoon. Lisäksi on tärkeää tarjota mahdollisuus avun saamiseen opettajalta. Opettajan on myös mahdollista tarjota tukea ilman, että opiskelija sitä erikseen pyytää. Esimerkiksi jos opettaja huomaa saman kriteerin nousevan monta kertaa peräkkäin opiskelijan korjaamista vaativiin kriteereihin, on opettajan mahdollista tarjota opiskelijalle apua ja kommentteja asian korjaamiseen.

Itsearviointi ei saa olla liian pitkä, jotta opiskelijat jaksavat toteuttaa sen ajatuksella. Jotta itsearviointiprosessi toteutuu, tulee opiskelijaa kannustaa palauttamaan työnsä säännöllisesti, esimerkiksi kerran viikossa. Itsearvioinnin suorittamiseen voi kannustaa esimerkiksi tarjoamalla hyvin täytetystä itsearviointilomakkeesta lisäpisteen.

6. YHTEENVETO

Tässä tutkielmassa selvitettiin keinoja, joilla itsearviointia voidaan integroida ohjelmoinnin opetuksessa käytössä oleviin automaattisiin arviointityökaluihin niin, että itsearviointi tukee opiskelijaa oppimisessa ja parantaa oppimistuloksia. Tämä toteutettiin tutkimalla tieteellisiä julkaisuja, joissa on tutkittu itsearviointia, ohjelmoinnin opetusta, itsearviointia sisältävien menetelmien käyttöä ohjelmistotuotannossa ja automaattisia arviointityökaluja.

Itsearvioinnin oikealla käyttämisellä voidaan parantaa opiskelijoiden oppimistuloksia. On kuitenkin tärkeää, että itsearviointi toteutetaan prosessina, joka on suunniteltu opiskelijan oppimista tukevaksi. Mikäli itsearviointi toteutetaan huonosti, saatetaan aiheuttaa enemmän haittaa kuin hyötyä oppimiselle.

Tutkitun aineiston perusteella tällä hetkellä paljon käytössä oleviin automaattisiin arviointityökaluihin on mahdollista lisätä itsearviointia niin, että opiskelijoiden oppimistulokset paranevat.

Työssä esitettiin valikoiden esimerkki itsearviointimallista ja muutama esimerkki itsearvioinnin hyödyntämisestä ohjelmistotuotannossa. Muita esimerkkejä on olemassa paljon enemmän kuin tässä työssä pystyttiin käsittelemään. Työssä pyrittiin luomaan yleistä mallia, jolla itsearviointia voidaan lisätä jo käytössä olevaan automaattiseen arviointityökaluun. Tällaisia malleja on jo olemassa ja niihin tutustuminen varmasti auttaa uuden mallin kehittämisessä.

LÄHTEET

[1] Perusopetuslaki 628/1998, 1998. Saatavissa: https://www.finlex.fi/fi/laki/ajantasa/1998/19980628.

[2] Lukiolaki 714/2018, 2018. Saatavissa: https://finlex.fi/fi/laki/ajantasa/2018/20180714.

[3] Yliopistolaki 558/2009, 2009. Saatavissa: https://www.finlex.fi/fi/laki/ajantasa/2009/20090558.

[4] Ammattikorkeakoululaki 932/2014, 2014. Saatavissa: https://finlex.fi/fi/laki/ajantasa/2014/20140932.

[5] A. Luostarinen, J.H. Nieminen, E.H. White, P. Nilivaara, I. Peltomaa, N. Ouakrim-Soivio, L. Tuohilampi, Arvioinnin käsikirja, PS-kustannus, Jyväskylä, 2019.

[6] J. Hattie, H. Timperley. The Power of Feedback. REV EDUC RES 2007, Vol. 77 No. 1, s. 81–112.

[7] H. Andrade, A. Valtcheva. Promoting Learning and Achievement Through Self-Assessment. Theory into practice 2009, Vol. 48 No. 1, s. 12–19.

[8] F. Dochy, M. Segers, D. Sluijsmans. The use of self-, peer and co-assessment in higher education: A review. Studies in higher education (Dorchester-on-Thames) 1999, Vol. 24 No. 3, s. 331–350.

[9] D. Boud. The role of self-assessment in student grading. Assessment and evaluation in higher education 1989, Vol. 14 No. 1, s. 20–30.

[10] L.A.J. Stefani. Assessment in Partnership with Learners. Assessment and evaluation in higher education 1998, Vol. 23 No. 4, s. 339–350.

[11] J.A. Ross. The reliability, validity, and utility of self-assessment. Practical assessment, research & evaluation 2006, Vol. 11 No. 10, s. 1–13.

[12] N. Falchikov, D. Boud. Student Self-Assessment in Higher Education: A Meta-Analysis. Review of educational research 1989, Vol. 59 No. 4, s. 395–430.

[13] L. Fan, Implementing Self-Assessment to Develop Reflective Teaching and Learning in Mathematics, World Scientific, Singapore, 2011.

[14] Itsearviointia tenttien sijaan. 2018; Saatavissa (viitattu 29.5.2021): https://www2.helsinki.fi/fi/uutiset/opetus-ja-opiskelu-yliopistossa/itsearviointia-tenttien-sijaan.

[15] J.H. Nieminen, L. Tuohilampi. 'Finally studying for myself' - examining student agency in summative and formative self-assessment models. Assessment and evaluation in higher education 2020, Vol. 45 No. 7, s. 1031–1045.

[16] H. Søndergaard, R.A. Mulder. Collaborative learning through formative peer review: pedagogy, programs and potential. Computer science education 2012, Vol. 22 No. 4, s. 343–367.

[17] COMP.CS.100: Ohjelmointi 1: Johdatus ohjelmointiin, Tampereen yliopisto. Saatavissa (viitattu 29.5.2021): https://www.tuni.fi/opiskelijanopas/opintotiedot/opintojaksot/tut-cu-g-45620?year=2020.

[18] E. Lahtinen, K. Ala-Mutka, H. Järvinen. A study of the difficulties of novice programmers. SIGCSE bulletin 2005, Vol. 37 No. 3, s. 14–18.

[19] K. Schwaber, Agile Project Management with Scrum, Microsoft Press, Sebastopol, 2009.

[20] S. Rawat, A. Sawant, Getting started with review board: analyze and improve your code using the collaborative code review tool, review board, Packt Publishing Ltd, Birmingham, England, 2014.

[21] Review Board Integrations. Saatavissa (viitattu 29.5.2021): https://www.reviewboard.org/integrations/.

[22] N. Kalogeropoulos, I. Tzigounakis, E.A. Pavlatou, A.G. Boudouvis. Computer-based assessment of student performance in programing courses. Computer applications in engineering education 2013, Vol. 21 No. 4, s. 671–683.

[23] K. Ala-Mutka. A Survey of Automated Assessment Approaches for Programming Assignments. Computer science education 2005, Vol. 15 No. 2, s. 83–102.

[24] S. Nidhra. Black Box and White Box Testing Techniques - A Literature Review. International Journal of Embedded Systems and Applications 2012, Vol. 2 No. 2, s. 29–50.