



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

RIITTA HIETARANTA
MOOC-OPPIMISALUSTAN VALINTA JA KÄYTTÖÖNOTTO
TTY:LLÄ

Diplomityö

Tarkastaja: professori Hannu-Matti
Järvinen
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan tiedekunta-
neuvoston kokouksessa 3. kesäkuuta
2015

TIIVISTELMÄ

RIITTA HIETARANTA: MOOC-oppimisalustan valinta ja käyttöönotto TTY:llä

Tampereen teknillinen yliopisto

Diplomityö, 50 sivua, 15 liitesivua

Helmikuu 2017

Tietotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Ohjelmistotuotanto

Tarkastaja: professori Hannu-Matti Järvinen

Avainsanat: MOOC, MOOC-alustavertailu, oppimisalusta, automaattinen ohjelmakoodin arviointi

TTY haluaa kokeilla potentiaalisesti suuria opiskelijamääriä houkuttelevien MOOCien (Massive Open Online Course) järjestämistä. Koska TTY:n nykyinen pääasiallinen oppimisalusta, Moodle, ei kunnolla skaalaudu osanottajamäärältään suurille kursseille, tarvitaan MOOC-käyttöön uusi oppimisalusta. Uuden alustan käyttöönottamista varten selvitettiin ennen tätä diplomityötä Riitta Hietarannan kandidaatintyössä, mitä vaatimuksia valittavan alustan tulisi toteuttaa. Tässä diplomityössä selvitettiin ensin, mitä MOOC-käyttöön sopivia oppimisalustoja on saatavilla. Tämän jälkeen eri oppimisalustojen ominaisuuksia kartoitettiin, ja vertailtiin sitten ominaisuuksia määriteltyihin vaatimuksiin. Parhaiten toiminnalliset vaatimukset sekä aikatauluvaatimukset täyttävä MOOC-alusta, Canvas, valittiin testikäyttöön. Aikatauluvaatimukset muodostuivat kahden testikäyttöä suunnittelevan kurssin, Johdatus ohjelmointiin ja Basics of Information and Knowledge Management, aloitusaikataulusta. Kurssit aikoivat aloittaa alustalla tammikuussa 2015.

Edellä mainitut kurssit olisivat olleet ensimmäiset uuden oppimisalustan käyttäjät. Johdatus ohjelmointiin asetti aikatauluvaatimuksien lisäksi myös vaatimuksen siitä, että alustalla täytyy voida arvostella ohjelmointitehtäviä automaattisesti. Koska alustaa, joka tällaisen ominaisuuden olisi itsessään toteuttanut ei löytynyt, toimintoa ryhdyttiin kehittämään. Ohjelmakoodia arvosteleva työkalu liittyy oppimisalustaan käyttäen Learning Tool Interoperability -spesifikaatiota. Sekä Johdatus ohjelmointiin että Basics of Information and Knowledge Management kurssien suunnitelmat muuttuivat, eivätkä kumpikaan aloitaneet Canvaksen käyttöä alkuperäisten suunnitelmien mukaan. Johdatus ohjelmointiin luopui kokonaan alustan käytöstä, joten automaattiarvostelijan kehitys keskeytettiin. Näiden kahden kurssin sijasta alustan käytön aloittivat kuitenkin tammikuussa 2015 kaksi muuta kurssia, Ohjelmistoarkkitehtuurit ja Software Engineering Methodology.

Oppimisalustan valitsemisen onnistumista arvioitiin sen perusteella, mitä mieltä näiden kahden ensimmäisen käyttäjäkurssin opiskelijat ja henkilökunta olivat alustan käytöstä. Opiskelijoiden mielipiteitä mitattiin järjestämällä anonyymi internetkysely. Henkilökunnan mielipiteitä kysyttiin sähköpostitse. Kyselyistä saatiin suurimmaksi osaksi negatiivista palautetta alustasta, minkä perusteella alustavalinta on epäonnistunut. Suoraa johtopäätöstä valinnan epäonnistumisesta ei kuitenkaan voi tehdä kyselyiden tuloksista, sillä nämä kaksi kurssia eivät olleet osallistuneet alustan vaatimusten määrittelyyn, joten alustaa ei ollut valittu näiden kahden kurssin tarpeita ajatellen. Täten valinnan onnistuminen jäi epäselväksi.

ABSTRACT

RIITTA HIETARANTA: MOOC Platform Selection and Deployment at TUT

Tampere University of Technology

Master of Science Thesis, 50 pages, 15 Appendix pages

February 2017

Master's Degree Programme in Information Technology

Major: Software Engineering

Examiner: Professor Hannu-Matti Järvinen

Keywords: MOOC, MOOC platform comparison, learning management system, automatic program code grading

TUT is willing to experiment on arranging MOOCs (Massive Open Online Course) which potentially attract large masses of students. Because the current primary learning management system (LMS), Moodle, does not properly scale to host courses with large numbers of participants, a new LMS is needed for MOOCs. It was defined in Riitta Hietaranta's bachelor's thesis what requirements the new LMS should implement. First, in this master's thesis, the available MOOC compatible LMSs were searched. Afterwards, the properties of the found LMSs were resolved and then compared against the previously defined requirements. The LMS that best matched both the functional and the schedule requirements, Canvas, was chosen to be deployed and tested. The schedule requirement came from the schedule of the two courses that planned to implement their courses on Canvas, Introduction to Programming and Basics of Information and Knowledge Management. The courses planned to start on January 2015.

These courses were to be the first ones on the LMS. Besides the schedule requirement, Introduction to Programming also set the requirement that the chosen LMS should allow automatic grading of coding assignments. As no LMS implemented this, development for a tool that implemented this was started. The tool would connect to the LMS using Learning Tool Interoperability. Both courses would eventually cancel their original plans of beginning the use of Canvas. On addition to that, Introduction to Programming decided not to use Canvas at all, which resulted in the interruption of the tool development. Instead of these two courses, two other courses, Software Architecture and Software Engineering Methodology started using Canvas on January 2015 instead.

The successfulness of the LMS choice was evaluated based on the opinions of the students and the staff of the two first courses using Canvas. The students' opinions were gathered using an anonymous internet poll. The staff's opinions were asked by email. Most of the feedback from the poll and the emails was negative, which would mean that the choice of the LMS has failed. However, direct conclusions about the failure cannot be conducted, because the two courses did not take part in the requirements evaluation. Hence, the LMS was not chosen with the requirements of these two courses in mind. Therefore, the successfulness of the choice remains unclear.

ALKUSANAT

```
22:29 [REDACTED] Joo tästä asiasta oli [REDACTED]n kanssa kerran hyvä ruokakeskustelu. Pääsin jossain vaiheessa keskustelua sisälle niiden kurssien pitäjien sielunelämään. "Hei miten me voitas tehdä tästä kurssista vieläkin 'parempi'? HEI MINÄ TIEDÄN!! Otetaan käyttöön uusi oppimisalusta, joka 1) on ihan paska, 2) ei ole käytössä millään toisella opintojaksolla, 3) ???, 4) profit!
```

Tämä diplomityö on selvitys siitä, kuinka yhden teekkarin tehtävästä löytää uusi oppimisalusta tuli lähes kolmensadan ihmisen päänvaiva. Legendaarisessa televisiosarjassa, South Parkissa, tontut varastavat öisin kalsareita ja tuntemattoman tuotantovaiheen kautta kääntävät kalsarit hyödyksi. Anonyymina pysyttelevää opiskelijaa lainaten kalsaritonttujen tavoin rohkeat kurssijärjestäjät lähtivät valloittamaan tuntematonta oppimisalustaa. Tästä valloituksesta syntyi vaiheen ??? jälkeen hyöty. Diplomityössä selviää, minkä vaiheiden kautta hyötyyn päästiin, ja mitä hyötyä tästä kokeilusta oli tarkoitus saada.

Kiitän Tampereen teknillisen yliopiston Tietotekniikan laitosta tämän diplomityömahdollisuuden tarjoamisesta. Kiitokset myös työnohjaajalle Hannu-Matti Järviselle ja työssä auttaneelle Essi Isohannille. Eriyisen ison kiitoksen ansaitsevat aviomieheni, appiukkoni ja äitini, jotka ovat pitäneet ihanat pikkuriesat kiireisenä, niin että diplomityön loppuunsaattaminen on ollut mahdollista.

Tampereella, 11.3.2017

Riitta Hietaranta

SISÄLLYSLUETTELO

| | | |
|-------|--|----|
| 1. | JOHDANTO | 1 |
| 2. | VAATIMUSMÄÄRITTELY | 3 |
| 2.1 | Vaatimusmäärittelyn toteutus..... | 3 |
| 2.2 | Vaatimusmäärittelyn tulokset..... | 3 |
| 3. | MOOC-ALUSTOJEN VERTAILU..... | 8 |
| 3.1 | MOOC-alustojen piirteet..... | 8 |
| 3.1.1 | LTI | 9 |
| 3.1.2 | API | 11 |
| 3.2 | Vertailussa käytetyt tiedonhankintamenetelmät..... | 11 |
| 3.3 | Vertailun tulokset | 12 |
| 4. | CANVAS TESTIKÄYTÖSSÄ | 21 |
| 4.1 | Asentaminen..... | 22 |
| 4.2 | Konfigurointi | 22 |
| 4.2.1 | Yhdistyminen yliopiston identiteetinhallintaan | 23 |
| 4.2.2 | Roolien oikeudet | 25 |
| 4.2.3 | Lokalisaatio | 26 |
| 4.3 | Laajentaminen | 26 |
| 4.4 | Ylläpito..... | 27 |
| 5. | CANVAKSEN KÄYTTÖKOKEMUKSET | 28 |
| 5.1 | Canvaksen testikäyttäjät | 28 |
| 5.1.1 | Software Engineering Methodology | 29 |
| 5.1.2 | Ohjelmistoarkkitehtuurit | 29 |
| 5.2 | Kerätyt käyttökokeemukset..... | 29 |
| 5.2.1 | Käyttökokeemukset kyselyn perusteella..... | 29 |
| 5.2.2 | Vastuuhenkilöiden palaute | 32 |
| 5.3 | Testikäytön arviointi | 33 |
| 6. | PYTHON-KOODIN AUTOMAATTIARVOSTELIJA | 34 |
| 6.1 | Käytetyt teknologiat | 35 |
| 6.1.1 | Django | 35 |
| 6.1.2 | Docker | 35 |
| 6.1.3 | Ace Editor | 37 |
| 6.1.4 | Valmiit ohjelmistokomponentit | 37 |
| 6.2 | Arkkitehtuuri | 38 |
| 6.3 | Toteutettu toiminta | 40 |
| 6.4 | Jatkokehitys | 44 |
| 7. | YHTEENVETO | 48 |
| | LÄHTEET..... | 50 |
| | LIITE 1: CANVAKSEN KÄYTTÖKOKEMUSKYSELY | |
| | LIITE 2: KÄYTTÖKOKEMUSKYSELYN TULOKSET | |

LYHENTEET JA MERKINNÄT

| | |
|------|---|
| HTML | Hypertext Markup Language. Verkkosivujen kuvaamiskieli |
| IdP | Identity Provider. Verkkopalvelu, joka autentikoi käyttäjän ja tarjoaa tietoja käyttäjästä palvelua käyttäville tahoille. |
| LMS | Learning Management System. Oppimisalusta |
| LTI | Learning Tool Interoperability. Oppimisalustojen ja oppimistyökalujen yhteentoimivuutta varten kehitetty spesifikaatio. |
| MOOC | Massive Open Online Course eli suurelle osallistujamäärälle suunnattu avoin verkkokurssi. |
| SIS | Student Import System. Yleisnimi oppilastietojärjestelmille |
| TC | Tool Consumer. LTI-spesifikaatioon liittyvä oppimistyökalun käyttäjä. |
| TP | Tool Provider. LTI-spesifikaatioon liittyvä oppimistyökalun tarjoaja |
| TTY | Tampereen teknillinen yliopisto. |

1. JOHDANTO

Opetusala on lähivuosina mullistanut uusi ilmiö, jossa huippuyliopistot jakavat tietämystään internetissä ilmaisilla kursseilla, MOOCeilla. Lyhenne tulee sanoista Massive Open Online Course eli suuri ja avoin verkkokurssi. Avoimuus tarkoittaa sitä, että kuka tahansa voi osallistua kurssille ja suuruus viittaa siihen, että MOOCit parhaimmillaan houkuttelevat tuhansia opiskelijoita osallistumaan. Myös TTY:llä halutaan kokeilla omien MOOCien järjestämistä, ja tässä diplomityössä tehdään pohjatyötä sen mahdollistamiseksi. MOOCin järjestäminen vaatii oppimisalustan, jolle kuka tahansa voi luoda käyttäjätunnuksen ja alustan täytyy sekä tekniseltä suorituskyvyltään että hallintatoiminoiltaan skaalautua tuhansien käyttäjien käyttöön. TTY:n pääasiallinen oppimisalusta, Moodle, täyttää nämä vaatimukset periaatteessa, mutta käytännössä se ei ole toimiva vaihtoehto MOOCeja ajatellen. Moodlen ylläpitäjän, Pasi Häkkisen, mukaan Moodlen avulla suurien opiskelijamäärien hallitseminen on opettajille liian työlästä. Lisäksi TTY:n Moodlesta vapaa käyttäjätunnuksien luominen on estetty, eikä sitä haluta mahdollistaa. MOOCien järjestämistä varten tarvitaan siis uusi oppimisalusta, ja koska MOOC-käyttöön soveltuvia oppimisalustoja on lukuisia erilaisilla ominaisuuksilla varustettuna, on pakko suorittaa valintaprosessi. Tämän diplomityön tutkimuskysymys onkin: Miten valitaan TTY:lle sopiva MOOC-alusta? Lisäksi vastataan kysymyksiin, kuinka suoritettua valinnan onnistumista voidaan arvioida ja miten valittu alusta voidaan sovittaa erikoistuneita ominaisuuksia vaativan kurssin tarpeisiin sopivaksi.

Tätä diplomityötä edeltää Riitta Hietarannan kandidaatintyö, jonka yhteydessä on määriteltä, mitä vaatimuksia uuden oppimisalustan tulee täyttää. Vaatimusten määrittämiseksi haastateltiin niitä ihmisiä, joilla oli suunnitelmassa järjestää MOOC TTY:n nimissä. Nämä vaatimukset määrittävät, mitä tarkoittaa tutkimuskysymyksessä esiintyvä määre ”TTY:lle sopiva”. Diplomityön ensimmäisessä vaiheessa selvitettiin, mitä eri MOOC-alustoja on olemassa ja minkälaisia ominaisuuksia ne toteuttavat. Toisessa vaiheessa eri alustojen toteuttamia ominaisuuksia verrattiin kandidaatintyön määrittämää vaatimuslistaa vastaan ja selvityksen tulokset koostettiin tulostaulukoon. Tulostaulukon perusteella pääteltiin, mikä alusta otetaan TTY:lle testikäyttöön. Kandidaatintyö ja diplomityön kaksi ensimmäistä vaihetta vastaavat tutkimuskysymyksen pääkysymykseen. Kolme seuraavaa vaihetta, asennus, testikäyttö ja kokemustiedon kerääminen testikäytöstä vastaavat kysymykseen valinnan onnistumisen arvioinnista. Kysymykseen erikoistuneista ominaisuuksista vastataan viimeisessä vaiheessa, jossa yhtä suunniteltua MOOC-kurssia varten valmistetaan laajennos valittuun MOOC-alustaan.

MOOC-alustan valintaprosessin ja testikäytön edetessä alkuperäiset MOOC-suunnitelmat muuttuivat. Neljästä vaatimuksesta määritelleestä kurssista yksikään ei lopulta päätenyt käyttämään valittua MOOC-alustaa alkuperäisessä aikataulussa. Sen sijaan kaksi Tietotekniikan laitoksen kurssia Ohjelmistoarkkitehtuurit ja Software Engineering Methodology päätyivät kokeilemaan uutta alustaa. Näiden muuttuneiden suunnitelmien vuoksi edellä mainitun laajennoksen tekeminen jätettiin kesken, mutta jo valmiiksi saadut osat esitellään mahdollisten tulevaisuudessa tehtävien laajennosten takia esimerkin tarjoamiseksi.

Luvussa 2 kuvataan lyhyesti, miten MOOC-alustaan kohdistuvat vaatimukset määritettiin, ja mitä vaatimukset ovat. Luvun 2 asiat on kuvattu laajemmin diplomityötä edeltäneessä kandidaatintyössä. Luvussa 3 on kuvattu MOOC-alustojen vertailuprosessi, vertailun tulokset, oppimisalustan valinta testikäyttöön ja valintaan liittyvät ongelmat. Luvussa 4 kerrotaan lyhyesti vertailun tuloksen perusteella käyttöön otetun oppimisalustan ominaisuuksista sekä asennuksesta ja konfiguroinnista ja näiden vaiheiden ongelmista. Luvussa 5 käsitellään oppimisalustan käytöstä kerättyä käyttäjäpalautetta. Luvussa kuvataan palautteen keräämisessä käytetyt menetelmät, koostetaan palautteessa toistuneita teemoja ja arvioidaan oppimisalustan valinnan onnistumista palautteen pohjalta. Luvussa 6 kerrotaan testikäyttöön valitun MOOC-alustan toiminnan laajentamiseksi tehdystä työstä erikoistuneita ominaisuuksia vaativan Johdatus ohjelmointiin –kurssin tarpeiden täyttämiseksi.

2. VAATIMUSMÄÄRITTELY

Uuden oppimisolustan tarpeen taustalla on TTY:n aloittama MOOC-pilottikurssikokeilu. Kokeiluun ilmoittautui kuusi kurssia. Uuden oppimisolustan valitsemiseksi täytyi ensin selvittää näiltä kuudelta kurssilta sekä yliopiston johtoportaalta, mitä uudelta oppimisolustalta toivottiin. Näiden toiveiden kartoittamiseksi toteutettiin vaatimusmäärittely, jossa kuultiin edellä mainittuja osapuolia. Vaatimusmäärittelyn tuloksena saatiin joukko vaatimuksia, joiden perusteella voitiin määrittää sopivin MOOC-alusta. Vaatimusmäärittely toteutettiin tätä työtä edeltäneen kandidaatintyön yhteydessä. Tässä luvussa käsitellään lyhyesti vaatimusmäärittelyvaihetta sekä esitetään määrittelyssä saadut vaatimukset.

2.1 Vaatimusmäärittelyn toteutus

Vaatimusmäärittely aloitettiin haastattelemalla kesällä 2014 TTY:n MOOC-kokeiluun ilmoittautuneiden pilottikurssien vastuuhenkilöitä sekä TTY:n koulutuspäällikköä, Kirsi Reimania. Haastatteluissa pyrittiin määrittämään, mitä ominaisuuksia haastateltavat halusivat uuden oppimisolustan toteuttavan. Kokeiluun osallistuvia kursseja oli kuusi, mutta haastatteluissa kävi ilmi, että todellisuudessa vain neljällä kurssilla oli vaatimuksia uuden oppimisolustan suhteen. Kahden muun kurssin suunnitelmissa oli käyttää muita oppimisolustoja. Pilottikurssit, joiden vaatimuksia vaatimusmäärittelyssä huomioitiin, olivat Tiedonhallinnan ja logistiikan laitoksen Basics of Information and Knowledge Management, Tietotekniikan laitoksen Johdatus ohjelmointiin, Matematiikan laitoksen matemaattisen mallintamisen kurssi ja Sähkötekniikan laitoksen ”Mitä sähkö on?” Vaatimusmäärittelyn toteutuksesta voi lukea tarkemmin tätä diplomityötä edeltäneestä kandidaatintyöstä ”TTY:n MOOC-pilottiprojektien tarpeiden määrittäminen MOOC-alustan suhteen” [15].

2.2 Vaatimusmäärittelyn tulokset

Seuraavissa taulukoissa listataan vaatimusmäärittelyssä löydetyt vaatimukset. Vaatimukset on lajiteltu kolmeen eri luokkaan sen perusteella, kuinka tärkeää niiden toteutuminen on. Sellaisista ominaisuuksista, joita jokin pilottikurssi tai yliopisto ehdottomasti tarvitsi, seurasi vaatimuksia, jotka on luokiteltu ehdottomat-luokkaan. Ehdottoman vaatimuksen toteutumatta jäämisestä MOOC-alustalla seuraa, että tätä ominaisuutta vaativaa kurssia ei voida järjestää kyseisellä MOOC-alustalla. Sellaiset tarpeet, jotka selvästi olivat merkittäviä, mutta eivät estäisi kurssin toteutumista tai aiheuttaisi suhteetonta haittaa jäädessään toteutumatta, on luokiteltu luokkaan ”tärkeät”. Loput tarpeet on luokiteltu luokkaan ”hyvä olla olemassa”. Luokassa on tarpeita, jotka on haastatteluissa ilmaistu tapaan:

"Olisi kiva, jos...". Näistä johdettujen vaatimusten toteutumatta jättäminen ei tuottaisi ongelmia, mutta toteutuminen toisi alustan käyttäjille lisäarvoa.

Taulukoissa käytetään seuraavia lyhenteitä:

Taulukko 1 Tulostaulukoissa käytetyt lyhenteet

| Lyhenne | Merkitys |
|---------|---------------------------------------|
| OT | Oppimisen tuki |
| TLO | Tiedonhallinnan ja logistiikan laitos |
| TIE | Tietotekniikan laitos |
| MAT | Matematiikan laitos |
| DEE | Sähkötekniikan laitos |

Tärkeysjärjestyksessä luokitellut vaatimukset on esitetty seuraavassa taulukossa.

Taulukko 2 Yliopiston tarpeet

| Yliopiston tarpeet | | |
|---------------------|--|------------|
| Vaatimuksen tärkeys | Vaatus | Lähde |
| Ehdottomat | Henkilökunta voi hakea opiskelijoiden tietoja erilaisilla hakutermeillä (opiskelijanumero, nimi). | TIE |
| | Alusta tukee käyttäjätilin luomista ilman mitään ennakko-vaatimuksia. | DEE, (TLO) |
| | Alusta on mahdollisimman helposti ylläpidettävissä ja mahdollisimman suuri osa ylläpidollisista tehtävistä on käyttäjien itsensä tehtävissä. | OT |
| | Alustan täytyy tarjota keskustelualueita yleiseen keskusteluun ja tehtäväkohtaiseen avunantoon. | TIE |
| | Alustan täytyy tukea ryhmien muodostamista. | TLO, MAT |

| | | |
|---------|---|--------------------|
| | Alustan täytyy tukea automaattisesti tarkistettavissa olevien monivalintatehtävien käyttöä. | TLO, DEE |
| | Alustan täytyy tukea vertaisarvioinnin käyttöä tehtävien arvostelussa. | TLO, TIE |
| | Alustan täytyy tukea ohjelmointitehtävien automaattista arviointia joko natiivisti tai ulkopuolisten arvostelutyökalujen käytön sallimalla. | TIE |
| | Suoritusmerkinnät saa otettua alustalta jossakin käsiteltävissä olevassa formaatissa, esimerkiksi .csv-muodossa. | TIE, TLO |
| | Alustan tulee tarjota näkymä, joka selkeästi ja keskitetysti näyttää opiskelijoiden suorittamat tehtävät sekä niistä ansaitut pisteet ja lisäksi myös tekemättömät tehtävät.. | TIE |
| Tärkeät | Jäsenille täytyy voida määritellä erilaisia oikeuksia. | TIE, DEE |
| | Alusta tukee Shibboleth-kirjautumista. | Kaikki |
| | Alustan käytettävyyden suunnitteluun on panostettu. | Kaikki |
| | Alusta tukee materiaalin osittelua osakokonaisuuksiin. | TIE |
| | Alustan täytyy tarjota keskitetty näkymä kurssin kaikista keskusteluista. | TIE |
| | Alusta tukee usein kysytyt kysymykset –alueen (UKK/FAQ) luomista. | TLO |
| | Alusta tukee PDF:n näyttämistä. | TLO, TIE |
| | Alusta tukee mobiililaitteiden käyttöä. | TLO, TIE, MAT, DEE |

| | | |
|--------------------|--|------------|
| | Alustan täytyy olla mahdollisimman joustavia teknologioita käyttäen toteutettu, jotta erilaisten interaktiivisten tehtävien käyttäminen olisi mahdollista. | TIE, DEE |
| | Alustan puolesta täytyy olla määriteltävissä armonaikaa tehtävien palautukseen, jotta tehtävät on mahdollista palauttaa myöhässä annetun ajan puitteissa. | TIE |
| | Alustan tulee tarjota näkymä, josta selkeästi näkee kurssin osion tekemättömät tehtävät. | TIE |
| Hyvä olla olemassa | Henkilökunta voi tilata muistutuksia ajastetuista tapahtumista. | TLO |
| | Alustan tulisi mahdollistaa ulkopuolisen materiaalin lukeminen alustan sisällä. | TLO |
| | Alusta tukee opiskelijoiden sosiaalista kanssakäymistä monipuolisesti. | TLO |
| | Alusta tarjoaa mahdollisuuden rakentaa kurssille jonkinlainen julkinen etusivu. | TIE |
| | Yliopiston logo ja nimet ovat selvästi esillä alustalla | OT |
| | Alusta tukee materiaalin hakua suoraan versionhallinnan repositoriosta. | TIE |
| | Alustan on hyvä tukea videoiden lataamista ja hakemista useista eri lähteistä, kuten esimerkiksi Youtube ja Vimeo. | Oma huomio |
| | Alustalla täytyy voida luoda keskustelualueita, joiden käyttäjäkuntaa voidaan rajata, ja jotka eivät näy ulkopuolisille. | TLO, MAT |
| | Täytyy olla mahdollista sekä asettaa palautuksen tarkaraja tai jättää se määrittelemättä. | TIE |
| | Alustan tulee tukea tehtävänratkaisuiden anonyymia julkaisua. | TLO |

| | | |
|--|---|-----|
| | Alustan tulee ilmaista selkeästi ero pakollisten ja vapaaehtoisten tehtävien välillä, esimerkiksi listaamalla kaikki pakolliset tehtävät. | TIE |
|--|---|-----|

Pilottikurssien järjestäjien haastatteluista ja vaatimuksista voidaan havaita tyytymättömyys nykyisiä opetustyökaluja, erityisesti Moodlea kohtaan. Esitetyt tarpeet kerrottiin usein projisoituna Moodleen ja sen ominaisuuksiin tai ominaisuuksien puuttumiseen. Myös muissa yhteyksissä, esimerkiksi eri kurssien opetustilaisuuksissa, Moodleen käytön ongelmista on kuultu valitusta. Moodle on TTY:n virallinen opetusala, joten sen ongelmakohdat koskettavat koko yliopistoa. Tässä mielessä voidaan sanoa, että vaikka vaatimusmäärittely tehtiinkin vain neljän kurssin perusteella, ainakin ne kohdat, joissa on toivottu parannuksia Moodleen nähden, saattavat edustaa yliopiston yleistä kantaa. Kuitenkin TTY:llä opetetaan kursseja niin monelta eri alalta ja monella eri tyylillä, että on selvää, että tässä esitetty vaatimuslista ei ole täydellinen.

Vaatimusmäärittelyssä saatujen vaatimusten määrä pilottikurssia kohti vaihteli suuresti sen mukaan, kuinka huolellisesti pilottikurssia oli suunniteltu. Jos pilottikurssin toteutusta oli jo mietitty tarkasti, haastateltavilta sai paljon vaatimuksia, ja ne olivat melko yksityiskohtaisia. Jos taas pilottikurssin suunnittelu oli vasta alkuvaiheilla, vaatimuksia ei ollut paljon, ja ne olivat ympäröiväisiä. Kurssisuunnitelmien tarkentuessa tarkentuivat siis myös ajatukset siitä, mitä ominaisuuksia kurssin opetusallalla täytyy olla.

Mahdollisten jatkoselvitysten, esimerkiksi uuden, laajemman vaatimusmäärittelyn varalta, on hyvä huomioda, että vaatimuksissa ei oteta mitenkään kantaa yleisiin ohjelmitoihin kohdistuviin laatuvaatimuksiin. Esimerkiksi ISO-25010-standardissa määritellään laatuvaatimuksiksi toiminnallinen soveltuvuus, suorituskky, yhteensopivuus, käytettävyys, luotettavuus, turvallisuus, ylläpidettävyys ja siirrettävyys [21]. Käytännössä vaatimusmäärittelyssä kerätyt toiminnalliset vaatimukset ovat etusijalla yleiseen laatuun nähden, koska ne vaikuttavat siihen, onko oppimisalustaa ylipäättään mahdollista käyttää sille suunniteltuun tarkoitukseen. Laatuominaisuuksien toteutuminen vaikuttaa kuitenkin siihen, kuinka mukavaa oppimisalustan käyttäminen on opettajien, opiskelijoiden ja ylläpito henkilökunnan kannalta. Tästä syystä myös laatuominaisuuksien toteutumista oppimisalustoilla tulisi arvioida, jos vastaavanlaisia selvityksiä tehdään vielä toiste.

3. MOOC-ALUSTOJEN VERTAILU

MOOC-alustojen vertailuvaiheessa täytyi aluksi selvittää, mitä MOOC-alustoja on olemassa. Tämä onnistui helposti, koska eri MOOC-alustoilta kursseja listaavia verkkosivuja on monia. Tämän jälkeen voitiin aloittaa eri alustojen ominaisuuksien selvittäminen. Ominaisuuksien selvittämisessä käytettiin tukena vaatimusmäärittelyssä saatua vaatimustalista. Ensimmäisessä aliluvussa on kerrottu, miten MOOC-alustat eroavat perinteisistä oppimisalustoista. Selvitystyössä käytetyt tiedonhankintamenetelmät on kuvattu toisessa aliluvussa. Selvitystyön tuloksena saatiin koottua tulostaulukko, josta käy ilmi kunkin vaatimusmäärittelyn vaatimuksen toteutuminen vertailluilla alustoilla. Tulostaulukko, alustan valinta ja vertailun onnistumisen arviointi ovat viimeisessä aliluvussa.

3.1 MOOC-alustojen piirteet

MOOC-alustat toteuttavat pitkälti samoja ominaisuuksia kuin perinteisetkin oppimisalustat. Opettajat voivat luoda kursseja, ja kursseille voidaan luoda tehtäviä, joita oppilaat pystyvät tekemään. Alusta pitää kirjata opiskelijoiden suorituksista ja käyttäjät voivat kommunikoida keskenään hieman alustasta toiseen vaihdellen esimerkiksi keskustelupalstalla tai yksityisviestein. Perinteisistä oppimisalustoista MOOC-alustat erottuvatkin lähinnä opetustavan ja mahdollisten tehtävätyyppien kannalta. MOOCit eivät voi käytännössä sisältää lähiopetusta, joten on käytettävä muita menetelmiä. MOOC-alustat tukevat usein video-oppitunteihin tai mahdollistavat verkon välityksellä oppituntien seuraamisen liveinä. Tehtävät ovat tyypillisesti automaattisesti arvosteltavissa olevia, esimerkiksi monivalintatehtäviä, tai vertaisten arvosteltavia, koska opetushenkilökunta ei voi millään arvostella tuhansien opiskelijoiden tehtäväpalautuksia.

MOOC-alustoja löytyy sekä verkkopalveluina että itse ylläpidettävänä ohjelmistoina. Verkkopalveluita löytyy huomattavasti enemmän. Verkkopalvelulla tarkoitetaan sitä, että joku ulkopuolinen taho ylläpitää ja hallinnoi alustaa ja tarjoaa sitä kurssien järjestämiseen joidenkin ehtojen mukaisesti. Ehtoja voivat olla esimerkiksi se, että palvelusta täytyy maksaa tai että kurssisisältö on alustan tarjoajan asettamien vaatimusten mukaista. Verkkopalvelu voi olla joko yksittäisille opettajille suunnattu palvelu tai yliopistolle tai muulle organisaatiolle suunnattu. Joidenkin verkkopalveluiden käyttö voi vaatia käyttösopimuksen tekemistä MOOC-alustan tarjoajan kanssa ja toiset ovat kaikille vapaasti käytettävissä, vaatien vain käyttäjätilin luomisen.

Ylläpidettävät MOOC-alustat voidaan asentaa omalle palvelimelle ja niiden ylläpidosta, kuten päivityksistä ja ongelmien korjaamisesta täytyy huolehtia itse. Tällöin palvelimesta

ja ylläpidosta aiheutuvat kustannukset jäävät itselle hoidettaviksi. Lisäksi palvelimen ylläpito ja oppimisalustan päivittäminen täytyy järjestää jotenkin. Myös oppimisalustan käytöstä koituu todennäköisesti jotakin ylläpitotyötä, esimerkiksi käyttäjätunnusten luomista opettajille ja alustan asetusten konfigurointia.

MOOC-alustat ovat opetusmenetelmien kannalta usein melko yksinkertaisia, yleiskäyttöisiä perustoimintoja sisältäviä. Useat alustat mahdollistavat kuitenkin toimintojensa laajentamisen ulkopuolisia oppimistyökaluja ja palveluita hyödyntäen. Käytännössä laajentamismekanismit ovat erilaisia liityntöjä, joilla ulkopuoliset työkalut ja palvelut voivat kommunikoida alustan kanssa. MOOC-alustoilla näitä mekanismeja ovat Learning Tool Interoperability (LTI) ja Application Programming Interface (API). Liitynnät mahdollistavat hyvin erikoistuneiden kurssityökalujen käyttämisen alustalla järjestetyillä kursseilla, vaikka alustalle työkalun toteuttaman toiminnallisuuden tuottaminen ei kannattaisikaan pienten käyttäjämäärien ja alustan monimutkaistumisen vuoksi. Laajennettavuus on TTY:n kannalta merkittävää, koska yliopistolla tarjotaan monialaista ja hyvin erikoistunutta opetusta. Laajentamisen mekanismit LTI ja API esitellään kahdessa seuraavassa aliluvussa.

3.1.1 LTI

LTI on IMS Global Learning Consortiumin kehittämä spesifikaatio, jonka pääasiallinen tarkoitus on tarjota standardi tapa oppimistyökalujen integrointiin [16]. IMS Global Learning Consortium on kansainvälinen, voittoa tavoittelematon jäsenjärjestö, joka pyrkii mahdollistamaan oppimisteknologian kasvun ja vaikutuksen maailmanlaajuisesti sekä tavallisessa opetuksessa että yritysten opetussektoreilla [17]. LTI määrittelee standardin tavan, jolla työkalun käyttäjä (tool consumer, TC), tyypillisesti oppimisalusta, kuten MOOC-alusta, voi kommunikoida alustan ulkopuolisen oppimistyökalun tarjoajan (tool provider, TP) kanssa. LTI mahdollistaa ensinnäkin sen, että opiskelija autentikoituu vain oppimisalustalle eikä erikseen jokaiseen työkaluun, jota jollakin kurssilla käytetään. Tämä tarkoittaa sitä, että kun opiskelija on kirjautunut oppimisalustalle käyttäjätunnuksillaan, LTI:n kautta alustaan linkitetyt ulkopuoliset työkalut saavat alustalta tiedon, että käyttäjä on jo autentikoitu. Täten opiskelijan ei tarvitse luoda uusia käyttäjätunnuksia jokaiseen käyttämäänsä työkaluun. Kirjautuminen mahdollistaa esimerkiksi sen, että LTI-työkalu voi säilyttää käyttäjän etenemistä, esimerkiksi tehtävien tuloksia, pitkäaikaisemmin, kuin mitä selaimen evästeillä voidaan toteuttaa. Kirjautumistiedon lisäksi LTI:n avulla voidaan välittää alustalta työkalulle joitakin perustietoja sekä käyttäjästä että kursista, jonka kautta opiskelija käyttää työkalua. Toisekseen LTI mahdollistaa sen, että ulkopuolinen työkalu voi lähettää oppimisalustalle arvosanoja, jos TC toteuttaa tuloksien lähettämisen sallivan rajapinnan. Rajapinta määritetään toisessa IMS:n spesifikaatiossa, IMS Learning Information Service (LIS). Tämän tyyppinen työkalu voisi olla esimerkiksi molekyylinrakentelutehtäviä tarjoava kemian opetustyökalu, joka palauttaisi käyttäjien

tehtävistä saamat arvosanat oppimisalustalle. LTI:tä tuetaan lukuisilla MOOC-alustoilla, ja myös LTI:tä tukevia oppimistyökaluja on paljon saatavilla.

LTI jakaa vastuun TC:n ja TP:n välillä siten, että TC pitää yllä tietoja käyttäjistä ja arvosanoista, huolehtii käyttäjien autentikoinnista ja aloittaa käyttäjän pyynnöstä TP:n käytön lähettämällä tietyt TC:stä ja TC:n asetuksista riippuvat tunnistetiedot käynnistyksen yhteydessä. Lähetettävistä tunnisteista kerrotaan myöhemmin tässä aliluvussa. TP:n vastuulla on käynnistyskontekstin tunnistaminen, käynnistyksen autentikointi ja oikean materiaalin näyttäminen käyttäjälle käynnistyskontekstin perusteella. Lisäksi TP:n vastuulla on lähettää opiskelijan arvosana oppimisalustalle, jos kyseessä on sellainen arvosteltava tehtävä, jonka arvosanat halutaan sinne välittää. Käynnistyksen autentikoinnilla tarkoitetaan lähetettyjen tunnistetietojen tarkistamista siten, että voidaan todeta, että kyseessä on TP:n sallima TC, eikä jonkun pahantahtoisen tahon yritys käsitellä TP:n materiaaleja. Tarkempaa tietoa tästä autentikoinnista on hieman myöhemmin tässä aliluvussa.

TP ja TC kommunikoiivat keskenään HTTP POST -pyynnöin. LTI määrittelee minkälaista dataa TC:n TP:lle lähettämiin pyyntöihin täytyy ja voi liittää. TP:n TC:lle lähettämien viestien muodon määrittelee LTI:n ohella osittain myös LIS. LTI määrittelee suuren joukon erilaisia parametreja, joilla TC voi lähettää tietoa LTI-työkalun käyttökontekstista TP:lle. TC voi esimerkiksi lähettää TP:lle työkalua käyttävän kurssin nimen, tehtävän nimen, jonka yhteydessä työkalua käynnistetään tai työkalun käynnistäneen käyttäjän yksilöivän tunnisteiden tai hänen nimensä [16]. Arvosanojen palautuksen kannalta tärkein LTI-käynnistyksen parametri on `lis_result_sourcedid`, joka yhdistää yksiselitteisesti työkalun käyttökontekstin, esimerkiksi tietyn kurssin tietyn tehtävän ja työkalun käynnistäneen käyttäjän. Tätä parametria käyttämällä TP voi palauttaa käyttäjän suorittaman tehtävän arvosanan juuri oikean käyttäjän arvosanaksi liittyen juuri oikeaan tehtävään.

LTI:n määrittelemän TP:n ja TC:n välisen viestinvälityksen turvallisuus perustuu OAuth 1.0 -protokollan mukaisiin allekirjoitettuihin viesteihin. Lisäksi viestinvälityksessä on suositeltavaa käyttää salattua HTTPS-yhteyttä salaamattoman HTTP:n sijasta. Viestit allekirjoitetaan TC:n yksilöivän käyttäjäavaimen (consumer key) ja TP:n ja TC:n yhteisen jaetun salaisuuden avulla. On hyvä huomioida, että saman oppimisalustan eri kurssijärjestäjillä voi olla oma käyttäjäavaimensa samaan LTI-työkaluun. LTI-työkalua käyttöönotettaessa sovitaan salasanaa vastaavasta yhteisestä salaisuudesta, joka saatetaan sekä TP:n että TC:n tietoon. LTI-käynnistyksen yhteydessä TC muodostaa salatun viestin salaamalla käyttäjäavaimen käyttäen yhteistä salaisuutta salausavaimena. TC liittää sekä käyttäjäavaimen että salatun viestin HTTP POST -pyynnön parametreihin. Pynnön vastaanottaessaan TP tunnistaa lähettäjän parametreista löytyvän käyttäjäavaimen perusteella, salaa käyttäjäavaimen käyttäen yhteistä salaisuutta ja vertailee salauksen tulosta parametrina saatuun salattuun viestiin. Jos salatut viestit ovat samat, TC on autentikoitu, ja TP sallii pynnön etenemisen eli esimerkiksi jokin tietty parametri määritellyn

käynnistyskontekstin määrittämä materiaali näytetään HTTP-pyyntön lähettämisen laukaiselle opiskelijalle. Jos käynnistyskonteksti on arvosteltava tehtävä, TP voi lähettää arvosanan TC:lle HTTP POST -pyynnöllä, jonka sisältö on XML-muotoinen arvosanan ja käynnistyskontekstin tunnusteen (lis_result_sourcedid) sisältävä dokumentti. Myös tämä viesti turvataan käyttäen OAuth-protokollaa, mutta tässä koko viesti salataan käyttäen yhteistä salaisuutta, jotta kukaan ei voi esimerkiksi korvata alkuperäisen pyynnön arvosanaa paremmalla arvosanalla. [16]

3.1.2 API

Termille API löytyy lukuisia toisistaan hieman eroavia määritelmiä. Yksi riittävän tarkka määrittely löytyy osoitteesta avoinrajapinta.fi: ”Ohjelmointirajapinta (API) määrittelee, miten ohjelmisto tarjoaa tietoja tai palveluita sovelluksille tai muille tietojärjestelmille.” [25] Nykyään APIsta puhuttaessa tarkoitetaan yleisimmin verkkosovelluksien tarjoamia API-rajapintoja. Myös MOOC-alustojen yhteydessä APIt ovat tällaisia. MOOC-alusta-APIt voivat mahdollistaa esimerkiksi kurssin oppilaiden tietojen hakemisen, oppilaiden lisäämisen kurssille tai uuden tehtävän luomisen ohjelmallisesti. API:n paljastamista tiedoista ja palveluista riippuen alustan ulkopuolinen ohjelmisto voi toteuttaa APIa hyödyntäen parhaimmillaan hyvinkin monipuolisia hallinnointi- ja analysointitoimintoja, joita alusta itsessään ei tarjoa.

3.2 Vertailussa käytetyt tiedonhankintamenetelmät

Eri alustojen ominaisuuksia kartoitettiin sekä alustoja havainnoimalla että erilaisista kirjallisista lähteistä. Selvästi paras tiedonhankintamenetelmä oli oman testikurssin luominen alustalle, ja eri ominaisuuksien testaaminen tässä yhteydessä. Monet alustat tarjosivat mahdollisuuden, mutta valitettavasti eivät kaikki. Tämä menetelmä mahdollisti alustan ominaisuuksien tutkimisen omin silmin, mikä luonnollisesti toi paremman käsityksen kuin pelkkä ominaisuuksien kuvauksien lukeminen. Lisäksi tämä menetelmä toi puolueetonta tietoa, toisin kuin alustojen tarjoajien itse kirjoittamat kuvaukset toiminnallisuudesta. Toinen käytetty menetelmä oli erilaisten markkinointimateriaalien, käyttöohjeiden ja vinkkien ja ”usein kysytyt kysymykset” –osioiden lukeminen. Tätä mahdollisuutta käytettiin erityisesti silloin, kun testikurssin tekeminen ei ollut mahdollista, mutta myös havainnoinnin tukena. Joidenkin alustojen osalta vaatimuksien toteutumista tiedusteltiin sähköpostitse, kun julkisista materiaaleista ei löytynyt vastauksia ja kun testikurssit eivät olleet mahdollisia. Lisäksi edX:n ja Canvaksen osalta hyödynnettiin myös alustojen Google Groups –keskustelufoorumeita.

3.3 Vertailun tulokset

Taulukossa 3 on aluksi selitetty muutama myöhemmissä tulostaulukoissa esiintyvä termi. Vertailuun sisälletyt alustat ja niistä tulostaulukoissa käytetyt lyhenteet on listattu taulukossa 4. Lisäksi taulukossa on ilmoitettu, onko alusta saatavilla palveluna, ylläpidettävänä tai sekä että. Varsinaiset vertailun tulokset on esitetty taulukoissa 5, 6 ja 7. Taulukossa 5 ovat kaikista tärkeimmät ehdottomat vaatimukset. Ehdottoman vaatimuksen toteutumatta jäämisestä MOOC-alustalla seuraa, että tätä ominaisuutta vaativaa kurssia ei voida järjestää kyseisellä MOOC-alustalla. Täten toteutumatta jäämisestä seuraa, että kyseistä oppimisalustaa ei voida valita käyttöön, jos halutaan kyseisen vaatimuksen esittäneen kurssin pystyvän käyttämään valittavaa oppimisalustaa. Toteutumatta jääneet ehdottomat vaatimukset on merkitty taulukoon merkinnällä **Ei**. Taulukossa 6 ovat tärkeät vaatimukset ja taulukossa 7 vaatimukset, joiden toteutumisella ei ole niin suurta merkitystä. Taulukot on jäsennellyt siten, että ensimmäisessä sarakkeessa esitetään vaatimus ja lopuissa sarakkeissa määritellään, miten kukin vertailluista alustoista toteuttaa vaatimuksen. Joissakin soluissa esiintyy n/a (not applicable). Tällaisessa tapauksessa vastausta ei ole joko voitu saada, sen saaminen olisi ollut liian vaikeaa saavutettuun hyötyyn nähden tai vaatimus ei ole ollut kyseisen alustan kohdalla aiheellinen. Tulostaulukossa esiintyy muutama erikoinen termi, jotka on selitetty taulukossa 3.

Taulukko 3 Tulostaulukon termit

| | |
|-------------|--|
| Roolit | MOOC-alustan käyttäjäroolit, esimerkiksi opettaja, assistentti ja opiskelija; määrittävät alustalla käyttöoikeudet, esimerkiksi voiko käyttäjä luoda tehtäviä tai avata uusia keskusteluita. |
| Open source | Avoin lähdekoodi; vapaasti saatavilla ja usein myös muokattavissa. |
| Iframe | HTML:n (Hypertext markup language, verkkosivujen kuvaamiskieli) ominaisuus, joka mahdollistaa toisen verkkosivun sisällön upottamisen verkkosivuun. |
| Sovellus | Mobiilialustalle valmistettu versio oppimisalustasta. |

Taulukko 4 Oppimisalustojen lyhenteet

| Oppimisalusta | Lyhenne | Saatavuus |
|---------------|---------|-----------------------|
| Canvas | C | Palvelu/Ylläpidettävä |
| EdX | X | Palvelu/Ylläpidettävä |

| | | |
|--------------|---|---------|
| iversity | i | Palvelu |
| Eliademy | E | Palvelu |
| Udemy | U | Palvelu |
| NovoEd | N | Palvelu |
| OpenLearning | O | Palvelu |
| CourseSites | S | Palvelu |

Taulukko 5 Ehdottomat vaatimukset

| Vaatus | C | X | i | E | U | N | O | S |
|--|--------------------|------------------------|-------|-----------|-------|-------|-----------|-------|
| Henkilökunta voi hakea opiskelijoiden tietoja erilaisilla hakutermeillä (esim. opiskelijanumero, nimi). | Kyllä | Kyllä | n/a | Ei | n/a | Kyllä | Kyllä | Kyllä |
| Alusta tukee käyttäjätilin luomista ilman mitään ennakkovaatimuksia. | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä |
| Alusta on mahdollisimman helposti ylläpidettävissä ja mahdollisimman suuri osa ylläpidollisista tehtävistä on käyttäjien itsensä tehtävissä. | Kyllä | Ei ¹ | n/a | n/a | n/a | n/a | n/a | n/a |
| Alustan täytyy tarjota keskustelualueita yleiseen keskusteluun ja tehtäväkohtaiseen avunantoon. | Kyllä ² | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Ei | Kyllä |

¹ Osa ylläpidollisista tehtävistä hyvin vaikea tehdä, vaikka onkin tehtävissä itse

² Keskusteluita voi itse luoda, nimetä ja sijoitella kuten haluaa

| Vaatus | C | X | i | E | U | N | O | S |
|--|-------|-----------|------------------------|------------------------|-----------|-------|-----------|-----------|
| Alustan täytyy tukea ryhmien muodostamista. | Kyllä | <u>Ei</u> | <u>Ei</u> ¹ | <u>Ei</u> | n/a | Kyllä | Kyllä | Kyllä |
| Alustan täytyy tukea automaattisesti tarkistettavissa olevien monivalintatehtävien käyttöä. | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä |
| Alustan täytyy tukea vertaisarvioinnin käyttöä tehtävien arvostelussa. | Kyllä | Kyllä | Kyllä | <u>Ei</u> | <u>Ei</u> | Kyllä | <u>Ei</u> | Kyllä |
| Alustan täytyy tukea ohjelmointitehtävien automaattista arviointia joko natiivisti tai ulkopuolisten arvostelutyökalujen käytön sallimalla. | Kyllä | Kyllä | Kyllä: API | <u>Ei</u> ² | <u>Ei</u> | Kyllä | <u>Ei</u> | <u>Ei</u> |
| Suoritusmerkinnät saa otettua alustalta jossakin käsiteltävissä olevassa formaatissa, esimerkiksi .csv-muodossa. | Kyllä | Kyllä | <u>Ei</u> | Kyllä | n/a | Kyllä | Kyllä | Kyllä |
| Alustan tulee tarjota näkymä, joka selkeästi ja keskitetysti näyttää opiskelijoiden suorittamat tehtävät sekä niistä ansaitut pisteet ja lisäksi myös tekemättömät tehtävät. | Kyllä | Kyllä | Ei | Kyllä | <u>Ei</u> | Kyllä | <u>Ei</u> | Kyllä |

¹ Mahdollisesti toteutetaan vuonna 2014 lopulla

² Tukee LTI versiota 1.0, Eliademy on ilmoittanut aikovansa toteuttaa tuen LTI 2.0 versiolle [14]

Taulukko 6 Tärkeät vaatimukset

| Vaatus | C | X | i | E | U | N | O | S |
|---|--------------------|--------------------|-------|--------------------|--------------------|-------|-------|--------------------|
| Jäsenille täytyy voida määrittellä erilaisia oikeuksia. | Kyllä ¹ | Kyllä ¹ | n/a | Ei | Kyllä | Ei | Ei | Ei |
| Alusta tukee Shibboleth-kirjautumista. | Kyllä ² | Kyllä ² | Ei | Ei | Ei | Ei | Ei | Ei |
| Alustan käytettävyyden suunnitteluun on panostettu. | Kyllä | Ei | Kyllä | Ei | Kyllä | Ei | Kyllä | Ei |
| Alustan täytyy tarjota keskitetty näkymä kurssin kaikista keskusteluista. | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Ei | Kyllä |
| Alusta tukee FAQ-alueen luomista. | Kyllä ³ | Kyllä | n/a | Kyllä | Ei | Kyllä | Kyllä | Kyllä |
| Alusta tukee materiaalin ositte-lua osakokonaisuuksiin. | Kyllä ⁴ | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä |
| Alusta tukee PDF:n näyttämistä. | Kyllä | Kyllä: iframe | n/a | Ei | Kyllä | n/a | Kyllä | Ei ⁵ |
| Alusta tukee mobiililaitteiden käyttöä. | Kyllä | Ei | Kyllä | Kyllä ⁶ | Kyllä ⁶ | Ei | Ei | Kyllä ⁶ |

¹ Roolit² Open source –versiossa³ Sivuja voi luoda ja nimetä kuten haluaa⁴ Keskusteluita voi itse luoda, nimetä ja sijoitella kuten haluaa⁵ Vain liitteenä⁶ Sovellus

| | | | | | | | | |
|--|---------------|-------|-------|-------|-----------------|-------|-------|-------|
| Alustan täytyy olla mahdollisimman joustavia teknologioita käyttäen toteutettu, jotta erilaisien interaktiivisten tehtävien käyttäminen olisi mahdollista. | Kyllä: LTI | Kyllä | n/a | Ei | Ei | n/a | Ei | Ei |
| Alustan puolesta täytyy olla määriteltävissä armonaikaa tehtävien palautukseen, jotta tehtävät on mahdollista palauttaa myöhässä annetun ajan puitteissa. | Ei | Kyllä | n/a | Ei | Ei ¹ | Kyllä | Kyllä | Ei |
| Alustan tulee tarjota näkymä, josta selkeästi näkee kurssin osion tekemättömät tehtävät. | Kyllä | Kyllä | Kyllä | Kyllä | Ei | Kyllä | Ei | Kyllä |

Taulukko 7 ”Hyvä olla olemassa” -vaatimukset

| Vaatus | C | X | i | E | U | N | O | S |
|--|--------------------|--------------------|-------|-------|-------|-------|-------|-------|
| Henkilökunta voi tilata muistutuksia ajastetuista tapahtumista. | Ei | Ei | n/a | Ei | Ei | n/a | Ei | Kyllä |
| Alusta tukee Haka-tunnuksien käyttöä. | Kyllä ² | Kyllä ² | Ei | Ei | Ei | Ei | Ei | Ei |
| Alusta tarjoaa mahdollisen rakentaa kurssille jonkinlainen julkinen etusivu. | Ei | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä | Kyllä |
| Yliopiston logo ja nimet ovat selvästi esillä alustalla. | Kyllä | Ei | Kyllä | Kyllä | Ei | Kyllä | Kyllä | Kyllä |

¹ Tehtäville ei voi määrittellä deadlinea

² Open source -versiossa

| Vaatus | C | X | i | E | U | N | O | S |
|--|-------|-------|-------|------------------|-------|-------|-------|-------|
| Alusta tukee opiskelijoiden kanssakäymistä monipuolisesti. | Kyllä | Ei | Ei | Ei | Kyllä | Ei | Kyllä | Ei |
| Alustan tulisi mahdollistaa ulkopuolisen materiaalin lukeminen alustan sisällä. | Kyllä | Kyllä | n/a | Ei | Kyllä | n/a | Ei | Kyllä |
| Alusta voi tukea materiaalin hakua suoraan repositoriosta. | Ei | Ei | n/a | Ei | Ei | n/a | Kyllä | Ei |
| Alustan on hyvä tukea videoiden lataamista ja hakemista useista eri lähteistä, kuten esimerkiksi Youtube ja Vimeo. | Kyllä | Kyllä | Kyllä | Kyllä: iframe | Ei | Kyllä | Kyllä | Ei |
| Alustalla täytyy voida luoda keskustelualueita, joiden käyttäjäkuntaa voidaan rajata, ja jotka eivät näy ulkopuolisille. | Kyllä | Ei | n/a | Ei | Ei | n/a | Kyllä | Kyllä |
| Täytyy olla mahdollista sekä asettaa palautuksen takaraja tai jättää se määrittelemättä. | Kyllä | Kyllä | n/a | Kyllä | Ei | Kyllä | Ei | Kyllä |
| Alustan tulee tukea tehtävänratkaisuiden anonyymia julkaisua. | Ei | Ei | n/a | Ei | Ei | n/a | Ei | Ei |
| Alustan tulee ilmaista selkeästi ero pakollisten ja vapaaehtoisten tehtävien välillä, esimerkiksi listamalla kaikki pakolliset tehtävät. | Ei | Ei | Ei | Ei | Ei | Ei | Ei | Ei |

Testikäyttöön valittiin Canvas ja sen ylläpidettävä versio. Vertailussa sekä Canvas että NovoEd täyttivät kaikki ehdottomat vaatimukset. NovoEd täytti vähemmän kahden muun

luokan vaatimuksia kuin Canvas. Kaksi merkittävää puutetta NovoEdissa näiden vaatimusten kannalta ovat käyttäjien erilaisten oikeuksien määrittäminen ja tuki Shibbolethin käyttöön. Se, että NovoEdissa ei voi määrittää erilaisia oikeuksia käyttäjille tarkoittaa sitä, että esimerkiksi kurssiassistentteille ei voi määrittellä vastuuhenkilöä suppeampia oikeuksia. Shibboleth-tuen puuttuminen taas johtaa siihen, että opiskelijat joutuvat muistamaan yhden ylimääräisen käyttäjätunnukset yliopiston järjestelmiin. Kuten myöhemmin käyttäjäkokemuksista kävi ilmi, tämä koettiin Canvaksen testikäytössä negatiiviseksi asiaksi, koska intranet-kirjautumista ei saatu Shibboleth-tuesta huolimatta lukuvuoden 2014-2015 testikäyttöön mukaan.

Canvas ei välttämättä ole kaikista vertailluista alustoista paras vaihtoehto, koska palveluina tarjottavista alustoista monia asia jäi epäselväksi, ja jotakin muuta vaatimusjoukkoa käyttämällä jokin muu alusta olisi voinut pärjätä paremmin. Aikataulusyistä Canvas oli kuitenkin paras vaihtoehto MOOC-pilotteja varten. Ensimmäisten MOOC-alustan käyttäjien oli tarkoitus aloittaa uuden alustan käyttö vuoden 2015 alusta. Jos TTY olisi alkanut tarpeen tullen kilpailuttaa oppimisalustojen tarjoajien kanssa tehtäviä sopimuksia tai vain aloittanut sopimusneuvottelut jonkun tarjoajan kanssa sopimushinnan ollessa riittävän pieni, MOOC-alusta ei olisi millään ehtinyt tarpeeksi ajoissa käyttöön ensimmäisten testikäyttäjien aikatauluun nähden. Vertailun tulos on valmistunut lokakuun 2014 alussa ja alusta olisi pitänyt saada käyttöön viimeistään joulukuun 2014 alussa, jotta testikäyttäjät olisivat ehtineet luoda kurssinsa uudelle alustalle. Tämä olisi jättänyt vain hieman yli puolitoista kuukautta sopimuksen luomiseen TTY:n ja oppimisalustan tarjoajan välillä. Edellä mainitusta aikataulurajoitteesta johtuen käyttöön täytyi myös valita nimenomaan itse ylläpidettävä versio Canvakselta lisensoidun Canvaksen sijasta.

Vertailluista alustoista mikään ei suoraan toteuttanut Johdatus ohjelmointiin -kurssin erityisvaatimuksia ohjelmointitehtävien automaattitarkistuksesta. EdX oli edistynein tämän vaatimuksen suhteen, sillä alustan yhteyteen on kehitetty ohjelmisto nimeltään XServer. Kyseinen ohjelmisto hoitaa ohjelmointitehtävien arvioinnin varsinaisen alustan puolesta. XServer ei kuitenkaan ole sellaisenaan käyttöönottovalmis, vaan ohjelmistoon olisi täytynyt ohjelmoida melko paljonkin toimintoja lisää, ennen kuin sitä olisi voitu käyttää. Lisäksi XServer olisi toiminut vain edX:n kanssa. Edellä mainituista syistä valittiin XServerin sijaan monen eri alustan tukeman LTI:n hyödyntäminen ja automaattiarvioinnin suorittavan työkalun ohjelmoiminen. Valmiita LTI:tä tukevia ohjelmointitehtäviä arvioivia työkaluja ei löytynyt. Canvaksen kanssa myös API:n hyödyntäminen automaattiarvioinnin tekevän työkalun kanssa olisi mahdollista, mutta myös tämä vaihtoehto olisi aiheuttanut työkalun alustariippuvuuden. LTI-työkalun ohjelmoimisen hyötynä muihin vaihtoehtoihin nähden on se, että tuotoksena syntyy MOOC-alustasta riippumaton automaattiarvostelija. Arvostelijaa voisi käyttää minkä tahansa LTI:tä tukevan alustan yhteydessä tai työkalun toteutuksesta riippuen aivan itsenäisenä työkaluna. Tällaista LTI:tä

hyödyntävää arvostelijaa ryhdyttiin toteuttamaan tämän diplomityön yhteydessä Tietotekniikan laitoksella. Toteutus jäi kesken, koska Johdatus ohjelmointiin luopui kokonaan Canvaksen käytön aloittamisesta. Tästä kerrotaan tarkemmin luvussa 5. Työkalun toteutusta käsitellään luvussa 6.

Myös saksalainen Hasso Plattner Institute (HPI) on tehnyt vertailua eri MOOC-alustojen välillä, ja myös HPI päätyi valitsemaan saman Canvaksen avoimen lähdekoodin version kuin TTY:kin. HPI:n lähtökohta vertailuun oli hyvin samanlainen, kuin TTY:llä. He halusivat koettaa MOOCien järjestämistä MOOCeja järjestävien huippuyliopistojen innoittamana. Toisin kuin tässä diplomityössä, HPI teki vertailua vain neljän avoimen lähdekoodin oppimisalustan välillä, sillä he halusivat ehdottomasti valita alustan, jota pystyi muokkaamaan ja kehittämään omien tarpeiden mukaan. HPI:n alustalle asettamia vaatimuksia olivat muun muassa: [24]

- Intuitiivinen ja tehokas työkalu sisällön editointiin
- Mahdollisuus järjestellä materiaali helposti seurattavaksi kokonaisuudeksi, mutta myös mahdollisuus valita tiettyjä materiaaleja jättäen osa väliin
- Sisällön esitys hypertekstiä hyödyntäen eli opiskellessa on mahdollisuus poiketa yhdestä asiasta toiseen liittyvään asiaan helposti
- Käyttäjystävälliset ja interaktiiviset arviointityökalut
- Keskustelufoorumi, joka mahdollistaa henkilökunnalta opiskelijoiden aktivoinnin ja sallii opiskelijoiden kysellä sisällöstä.

HPI valitsi Canvaksen sen modernin käyttöliittymän, hyvän tehtävien ja kokeiden hallinnan, keskustelufoorumin ja käyttäjystävällisen kurssirakenteen hallinnan vuoksi [24]. HPI:n päätyminen Canvaksen käyttöön on merkittävää TTY:n kannalta, koska valintaan on päädytty samanlaisista lähtökohdista, vaikkakin erilaisten vaatimusten perusteella. On myös hyvin merkittävää, että HPI on muokannut Canvasta omiin tarpeisiinsa sopivaksi hyvin pienellä vaivalla. HPI raportoi, että neljä täysipäiväisesti työskentelevää opiskelijaa muutti viidessä viikossa kokonaan Canvaksen käyttöliittymän ja navigoinnin, lisäsi uuden sisältötyypin oppimisyksiköihin ja muutti navigoinnin sopivaksi kuuden viikon kurssiaikatauluun [24]. Tämä osoittaa, että TTY:lläkin voitaisiin helposti muuttaa Canvasta, jos joitakin muutoksia haluttaisiin tehdä. HPI on myös järjestänyt MOOCeja muokkaamallaan Canvaksella, joten todistettavasti se on mahdollista.

Vertailun voidaan nähdä onnistuneen, koska vertailussa saatiin selvitettyä laajasti asetettujen vaatimusten toteutumista eri MOOC-alustoilla. Vertailussa oli suuri otos kaikista MOOC-alustoiksi tulkittavissa olevista oppimisalustoista. Vertailun tuloksena saatiin myös valittua testikäyttöön sellainen alusta, joka toteutti kaikki ehdottomat vaatimukset ja myös hyvin paljon muiden vaatimusluokkien vaatimuksia tiukoista aikatauluvaatimuksista huolimatta. Jos vastaavanlaisia MOOC-alustaselvityksiä tehdään myöhemmin, on hyvä huomioida seuraavat asiat. Ensinnäkin, monet MOOC-alustat ovat ohjelmistoina

vielä hyvin tuoreita ja niiden ominaisuuksia kehitetään jatkuvasti. Täten nyt saatu tulos-
taulukko ei välttämättä ole paikkansapitävä kovin pitkään alustojen ominaisuuksien
muuttuessa ja lisääntyessä. Toiseksi, tämän vertailun valmistumisen jälkeen Euroopan
unionin tukema ECO (E-learning, Communication and Open-Data) on julkaissut MOOC-
alustojen ominaisuuksien tarkistuslistan dokumentissa D 2.1. Analysis of existing MOOC
platforms and services [26]. Tarkistuslista toimii hyvänä tukena MOOC-alustojen toimin-
nallisuuden hahmottamisessa, ja siitä olisi ollut tämänkin vertailun yhteydessä merkittä-
vää hyötyä kokonaisuuksien hahmottamisen helpottamiseksi. Vaikkakin hyödyllinen, tar-
kistuslista on kuitenkin niin yleisluontoinen, ettei pelkästään sen perusteella olisi pystytty
valitsemaan sopivaa alustaa esimerkiksi tämän MOOC-kokeilun pilottikursseja varten.
Tarkistuslistan tueksi tarvitaan myös konkreettisia toiminnallisia vaatimuksia.

4. CANVAS TESTIKÄYTÖSSÄ

Edellä kuvatun vertailun perusteella uudelle MOOC-alustalle suunniteltujen kurssien testikäyttöön valittiin Canvas ja sen ylläpidettävä avoimen lähdekoodin versio. Canvas asennettiin TTY:n palvelimelle yhteistyössä TTY:n tietohallinnon Moodlesta vastaavan Pasi Häkkisen kanssa. Häkkinen vastaa myös Canvaksen ylläpidosta.

Canvas on Yhdysvaltalaisen Instructuren ylläpitämä MOOC-alusta. Avoimen lähdekoodin version lisäksi alustaa tarjotaan Canvas Networkissa ja lisensoituna palveluna. Canvas Network on verkkopalvelu, jossa kuka tahansa voi järjestää kaikille avoimen MOOCin, kunhan se vastaa kurseille asetettuja laatuvaatimuksia. Lisensoituna palveluna Canvas tarjotaan Canvaksen ja palvelun ostavan tahon osapuolen tekemän sopimuksen mukaisesti, esimerkiksi alustaa ostajan brändin mukaiseksi muokaten tai tiettyä asiakaspalvelutasoa tarjoten.

Canvas mahdollistaa sekä kaikille avoimet sekä suljetut kurssit, joihin opiskelijat täytyy kutsua erikseen sähköpostitse. Alustalla materiaali voidaan jaotella moduuleiksi, esimerkiksi viikoittain, ja moduuleihin voi sijoittaa monipuolisesti eri materiaalia. Materiaali voi olla esimerkiksi alustalle luotuja sisältösivuja tai tehtäviä, tiedostoja, linkkejä ulkopuolisiin internetsivuihin tai keskustelujä. Tehtäviä on saatavilla eri tyyppisiä, mukaan lukien vertaisarvioitavia. Tavallisten tehtävien lisäksi alustalla voi järjestää kokeita, joiden tehtävätyypit ovat vielä lukuisimmat kuin tavallisten tehtävien. Voidaan sanoa, että muihin vertailtuihin alustoihin nähden Canvaksen opetus- ja sisällön järjestämistoiminnot ovat monipuoliset.

Canvas mahdollistaa omien kurssiensa kopioimisen ja vanhojen Canvas-kurssien sisältöjen tuomisen uudelle kurssille. Tämä mekanismi mahdollistaa materiaalin uudelleenkäytön. Canvas tukee myös Moodlen kurssien sisältöjen tuomista Canvas-kurssille, joten TTY:n opettajat voivat hyödyntää Moodleen luomiaan kursseja Canvakselle siirtyessään.

TTY:n käyttöön valittu avoimen lähdekoodin versio Canvakselta ei sisällä MOOC-opiskelijoille tärkeää kurssikatalogia. Yleisesti kurssikatalogissa listataan alustalla tällä hetkellä tarjotut ja tulossa olevat kurssit sekä tarjotaan hakutoimintoja esimerkiksi eri alojen kurssien löytämiseksi. Jotta TTY:n Canvaksessa voisi tarjota ulkopuolisille MOOC-kursseja, pitäisi saatavilla olevien kurssien listaamista ja etsimistä varten tarjota jokin menetelmä.

Seuraavissa aliluvuissa raportoidaan Canvaksen käyttöönottoon, muokkaamiseen ja ylläpitoon liittyviä olennaisia asioita.

4.1 Asentaminen

Canvaksen avoimen lähdekoodin versiota ylläpidetään versionhallintapalvelu Githubissa repositoriossa nimeltä instructure/canvas-lms [18]. Palvelussa ovat saatavilla sekä lähdekoodi että viralliset asennusohjeet, ongelmanratkaisuohteita ja alustan tunnettujen virheiden lista. Canvaksen asennusohjeessa suositellaan käyttämään palvelimena Ubuntu-palvelimen pitkäaikaisesti tuettua versiota 12.04 [19]. Ubuntu-palvelimen käyttäminen ei kuitenkaan ollut järkevää, koska yliopistolla on yleisesti käytössä Red Hat –palvelimet, ja asentaminen kuitenkin oli mahdollista myös Red Hat –palvelimelle virallisia asennusohjeita mukaillen. Canvas asennettiin aluksi virtuaalipalvelimelle TTY:n Tieto- ja sähkötekniikan tiedekunnan Lintula-ympäristöön. Tällä palvelimella Canvas toimi verkko-osoitteessa mooc-dev.pit.cs.tut.fi. Canvas-asennuksen voisi ohjeiden mukaan tarpeen tullen jakaa myös useammalle palvelimelle [19], mutta TTY:n asennus on kokonaisuudessaan yhdellä palvelimella. Tietokannaksi asennettiin ohjeiden suosittelema PostgreSQL, ja palvelinohjelmistona toimii Apache.

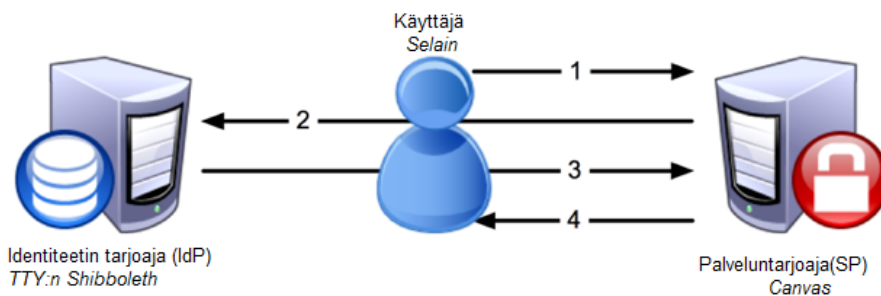
Hyvin nopeasti Canvaksen asennuksen jälkeen kävi ilmi, että alustan käyttämä Lintulan palvelin ei tulisi suoriutumaan oikeassa kurssikäytössä hitautensa ja vähäisen muistikapasiteetin takia. Lintulan palvelin oli myös avoimesta verkosta eristetty, joten Canvasta olisi siten tarvinnut aina käyttää TTY:n sisäverkosta käsin. Lisäksi palvelimelle annettu verkko-osoite oli aivan liian monimutkainen ulkoa muistettavaksi. Canvas-kokeilun tukemiseksi TTY:n tietohallinto järjesti oppimisalustalle uuden, tehokkaamman palvelimen, joka oli avoin myös ulkoiselle. Palvelin toimii verkko-osoitteessa mooc.tut.fi, joka on huomattavasti helpompi muistaa ja joka myös sopii hyvin mahdolliseen tulevaisuuden MOOC-käyttöön. Ensimmäinen Canvasta kokeileva kurssi ehti aloittaa alustan käytön, ennen kuin uusi palvelin ehdittiin ottaa käyttöön. Täten kurssi aloitti käytön vanhalla palvelimella ja siirtyi kesken kurssin uudelle palvelimelle. Uudelle palvelimelle siirryttäessä kurssin käyttäjätilit ja muu data, jota vanhan palvelimen tietokantaan oli ehtinyt kertyä, siirrettiin uuden palvelimen tietokantaan. Koska tämä onnistui ilman ongelmia, voidaan päätellä, että tarvittaessa ongelmien ilmetessä myös tietokantojen palauttaminen varmuuskopioista onnistuisi yhtä lailla ongelmitta.

4.2 Konfigurointi

Asennuksen jälkeen Canvas on käyttökelpoinen oppimisalusta. TTY:n käyttöä varten täytyi kuitenkin tehdä vielä joitakin lisätöitä. Ensimmäisessä aliluvussa kuvataan, mitä vielä tarvitsisi tehdä, jotta TTY:n intranet-tunnukset saataisiin Canvaksella käyttöön. Toisessa aliluvussa kuvataan, mitä on tehtävissä, jotta alusta sopisi paremmin suomalaiseen käyttöön. Viimeisessä aliluvussa kuvataan, miten Canvaksen valmiita roolierittelyitä on muutettu TTY:lle sopiviksi.

4.2.1 Yhdistyminen yliopiston identiteetin hallintaan

Ottaen huomioon, että TTY:llä on muutenkin erittäin laajasti käytössä intranet-tunnuksilla kirjautuminen, on järkevää, että myös uudelle oppimisalustalle voi kirjautua käyttäen samoja intranet-tunnuksia. Vain yksien tunnuksien käyttö vähentää muistettavien tunnuksien määrää, ja siten myös ylläpitotyötä unohtuneiden tunnuksien ja salasanojen palauttamiseksi. Canvas tukee kolmea eri kirjautumisjärjestelmää, jotka mahdollistavat Canvasta käyttävän instituution omien käyttäjätunnuksien käytön myös Canvaksessa. Alustan tukemia järjestelmiä ovat SAML, LDAP ja CAS. Näistä TTY:n tukemia ovat SAML-pohjainen Shibboleth ja LDAP, joista Shibbolethia käytetään aina ensisijaisesti. Tästä syystä myös Canvaksen käyttö intranet-tunnuksilla aiotaan toteuttaa hyödyntäen Shibbolethia LDAP:n sijaan. Shibbolethin toimintaperiaate on esitetty yksinkertaistettuna kuvassa 1 [27].



Kuva 1 Shibbolethin toimintaperiaate

Kuvassa identiteetin tarjoaja (identity provider, IdP) on Shibbolethin identiteetin hallinnan ydin. Sen vastuulla on autentikoida käyttäjä ja välittää eteenpäin tietoja käyttäjistä [27]. Palveluntarjoaja (service provider, SP) voi olla mikä tahansa verkkopalvelu, joka hyödyntää IdP:n autentikointia ja sen ylläpitämiä käyttäjätietoja sisäänrakennetun käyttäjätilien hallinnan sijasta tai sen ohella. Tämän diplomityön yhteydessä käyttäjät ovat TTY:n opiskelijoita ja opettajia, SP on Canvas ja IdP on TTY:n Shibbolethin IdP. Kuvan numeroidut nuolet esittävät tavanomaisen Shibbolethin käyttötapausten. Numerot viittaavat seuraaviin tapahtumiin: [27]

1. SP havaitsee käyttäjän pyrkivän käyttämään pääsyoikeuksilla rajattua resurssia.
2. SP muodostaa autentikointipyynnön, lähettää pyynnön käyttäjän IdP:lle ja ohjaa käyttäjän selaimen IdP:n kirjautumisnäkyymään.
3. IdP autentikoi käyttäjän, lähettää autentikointivastauksen SP:lle ja ohjaa käyttäjän selaimen takaisin SP:n näkyymään.
4. SP tarkastaa IdP:n vastauksen ja lähettää pyynnön eteenpäin resurssille, joka palauttaa alun perin pyydetyn sisällön käyttäjälle.

TTY:tä ja Canvasta esimerkkinä käyttäen kohta 1 voi olla esimerkiksi tilanne, jossa opiskelija aloittaa Canvaksella kurssin tehtävän tekemisen. Kohdan 2 seurauksena opiskelija

näkee kuvan 2 mukaisen näkymän selaimessaan. Kun opiskelija kirjoittaa kenttiin käyttäjätunnuksensa ja salasansa, ja painaa ”Login”, tapahtuu kohta 3. Vihdoin kohdan 4 jälkeen opiskelijalle näytetään tehtävä, jota hän alun perin yritti aloittaa.

Kuva 2 TTY:n IdP

Intranet-tunnuksien käyttöönotto Canvaksella ei kuitenkaan onnistunut samanaikaisesti alustan käyttöönoton kanssa. Ensiksikin, Canvaksen SAML-autentikointi ja TTY:n Shibboleth eivät toimi saumattomasti yhteen. Pasi Häkkisen mukaan Canvaksen SAML-autentikoinnin konfiguraatitiedosto ei ole yhteensopivassa muodossa TTY:n Shibbolethin IdP:n konfiguraatitiedoston kanssa. Tästä seuraa, että Canvaksen konfigurointi IdP:lle vaatii manuaalista konfigurointia, kun tavallisesti se voitaisiin tehdä automaattisesti. Kuitenkin suurin ongelma intranet-tunnuksien käyttöönotossa alustalla on se, että Canvas olettaa, että näillä erillisillä kirjautumisjärjestelmillä kirjautuvilla käyttäjillä on jo olemassa käyttäjätili alustalla. Tämä tarkoittaa sitä, että kun opiskelija kirjautuu ensimmäistä kertaa Canvaksen intranet-tunnuksillaan, Canvas ei luo opiskelijalle uutta käyttäjätiliä IdP:ltä saamiensa opiskelijan tietojen perusteella, vaan tili täytyy luoda jollakin menetelmällä etukäteen.

Vaihtoehtoja käyttäjätilien luomiseen on monia. Teknisesti kaikista yksinkertaisin vaihtoehto olisi luoda uudet tunnukset alustalle Canvaksen tarjoamaa SIS Import –toimintoa käyttäen aina, kun uusille opiskelijoille luodaan tunnuksia. SIS (Student Information System) on yleisnimitys oppilastietojärjestelmille [28], joten SIS Import viittaa oppilastietojen tuomiseen oppilastietojärjestelmästä Canvaksen. Tällä menetelmällä voidaan lisätä suuri määrä opiskelijoita yhdellä kertaa. Tässä menetelmässä ongelmaksi muodostuu kuitenkin se, että vaikka suurin osa uusista opiskelijoiden tunnuksista luodaan syksyllä ja keväällä, kansainvälisille vaihto-opiskelijoille tunnuksia saatetaan luoda koska tahansa. Yksittäisten opiskelijoiden lisääminen olisi liian työlästä tehtävän suorittavalle ylläpito-henkilökunnalle. Toinen vaihtoehto olisi Canvaksen muokkaaminen kooditasolla siten, että Canvas loisi uudet käyttäjätilit automaattisesti ensimmäisellä kerralla, kun käyttäjä kirjautuu käyttäen intranet-tunnuksia. Tämä vaihtoehto on melko vaivalloinen ja myös hyvin riskialtis Canvaksen päivittämisen suhteen, jos tehtyjä muutoksia ei suostuta sisällyttämään viralliseen Instructuren ylläpitämään Canvas-versioon. Viralliseen versioon tehdyt päivitykset voivat helposti rikkoa itse kehitellyn kirjautumissysteemin, joten tästä vaihtoehdosta voi aiheutua hyvinkin paljon ylläpitotyötä koodin korjailemisen muodossa.

Kolmas pohdittu ja kannattavin vaihtoehto on erillisen tunnuksia luovan työkalun tekeminen. Tässä vaihtoehdossa toimintaperiaate on se, että Canvaksesta ohjataan ensimmäistä kertaa intranet-tunnuksilla kirjautuvat käyttäjät siirtymään kirjautumistyökaluun. Ohjaus tapahtuisi kirjautumisnäkyvän yhteydessä näkyvällä linkillä, joka kysyy, onko tämä ensimmäinen kerta, kun käytät Canvasta intranet-tunnuksilla. Kirjautumistyökalu toimii tässä tapauksessa SP:n asemassa. Se tekee kuvan 1 kohdan 2 eli lähettää käyttäjän kirjautumaan TTY:n IdP:lle. Käyttäjän kirjautumisen jälkeen IdP välittää työkalulle Canvaksen käyttäjätilin luomiseen tarvitsemat tiedot, kuten opiskelijanumeron ja sähköpostiosoitteen. Opiskelijanumero sijoitetaan Canvaksen käyttämään SIS ID –tunnisteeseen. Tiedot saatuaan työkalu käyttää Canvaksen APIa luomaan uuden käyttäjätilin Canvakselle. Onnistuneen luomisen jälkeen käyttäjälle kerrotaan, että hän voi nyt kirjautua Canvaksen käyttäen TTY:n intranet-tunnuksia ja annetaan linkki Canvaksen kirjautumissivulle. Tämän jälkeen Canvaksen kirjautuminen tapahtuu aivan kuvan 1 mukaisesti. Tämä menetelmä vaatii muutoksia vain Canvaksen kirjautumisikkunan näkymätiedostoon, joten viralliseen versioon tulevat muutokset eivät aiheuta ainakaan suuria ongelmia. Vain kirjautumisikkunaan kohdistuvat muutokset täytyy huomioida Canvaksen päivityksiä asentaessa.

4.2.2 Roolien oikeudet

Canvaksen käyttäjätileillä on rooleja kahdella eri tasolla: kurssitasolla ja koko Canvaksen tasolla. Käyttäjällä on jokin rooli jokaisella kurssilla, jonka jäsen hän on ja sen lisäksi hänellä on jokin rooli koko Canvaksen tasolla. Kurssitason rooli voi vaihdella kurssittain. Esimerkiksi jollakin kurssilla käyttäjä voi olla opiskelija ja toisella kurssiassistentti. Canvaksen laajuisesti käyttäjä taas voi olla Canvaksen oletusasetuksilla vain pääkäyttäjä tai peruskäyttäjä. Rooleille ollaan sekä Canvaksen että kurssien tasolla määritelty joukko oikeuksia, jotka voidaan joko sallia tai kieltää roolin omaavilta käyttäjiltä. Ylläpitotyön helpottamiseksi Canvaksen luotiin ylimääräinen koko Canvaksen laajuinen rooli, opettaja. Opettajalle annettiin oikeus luoda ja poistaa kurseja ja lisäksi laajat oikeudet kursien sisältöjen muokkaamiseen. Ilman tätä roolia ylläpitohenkilökunta olisi joutunut luomaan jokaisen kurssin opettajien puolesta.

Canvaksen laajuisen opettajaroolin luomisesta seurasi se, että toinen laajan opettajaroolin omaava pystyi luomaan toiselle opettajalle käyttäjätilin samanlaisilla oikeuksilla. Tästä johtuen aliluvussa 5.1.2 tarkemmin esiteltävä Ohjelmistoarkkitehtuurien kurssi pääsi aloittamaan Canvaksen käytön ilman kunnan ohjeistusta Canvaksen käyttöön. Vaikka mahdollisuus, että opettajat voivat lisätä uusia opettajia vähentääkin ylläpidon taakkaa, seuraa tästä helposti laaja ongelma, kun kurssien järjestäjät eivät saa ohjeistusta hyviin käytänteisiin alustan käytössä ylläpitohenkilökunnalta. Ylläpitohenkilökunnan sijaan alustan käytön opettajina ovat tällöin toiset opettajat, jotka eivät ehkä tiedä kaikista helpoimpia ja tehokkaimpia tapoja alustan käyttöön.

4.2.3 Lokalisaatio

Canvaksessa palvelun lokalisaatio on toteutettu mielenkiintoisella tavalla. Palvelu päättää palveluun valitun käyttökielen pohjalta lokalisaatioon liittyviä asioita, kuten 12 tai 24 tunnin kellon käytön. Täten lokalisaation konfiguroimiseksi ei joudu tekemään muuta, kuin kielen ja aikavyöhykkeen valinnan. Canvaksesta ei kuitenkaan ole olemassa suomenkielistä käännöstä, joten kieleksi täytyy valita brittienglanti, jotta saadaan lähimmäs oikeat lokalisaatioasetukset käyttöön. Alustan yhdysvaltalaislähtöisyys näkyy kuitenkin varsinkin verkkopalvelun kalenterissa lokalisaation näennäisestä huomioimisesta huolimatta. Kalenterissa viikko alkaa sunnuntaista ja kellonajat näytetään käyttäen am- ja pm-merkintöjä, vaikka kalenterimerkintöjä tehdessä verkkopalvelu huoliikin 24 tuntisen kellon lukemat. Erityisesti sunnuntaialkuinen viikko aiheuttaa todennäköisesti väärinkäsityksiä, kun jonkun tapahtuman katsotaan olevan esimerkiksi viikon toisena päivänä katsomatta tapahtuman päivämäärää. Canvaksen Cody Cutrerin mukaan näiden lokalisaatio-ongelmien korjaamiseksi työskennellään jatkuvasti [4].

Vaikka Canvaksesta ei olekaan suomenkielistä käännöstä, sen tekeminen olisi kuitenkin kohtalaisella vaivalla mahdollista. Alusta käyttää eri kieliversioiden hallintaan käännös-palvelua nimeltä Transifex. Palveluun kirjataan ohjelmiston sisältämät tekstit, esimerkiksi valikoissa esiintyvät sanat ja ohjetekstit. Palveluun perustettavan kielitiimin jäsenet voivat kääntää tekstejä, ja sekä vertaiskäyttäjät että kielitiimin päällikkö voivat arvioida käännöksiä tai ehdottaa toisia käännöksiä. Suomenkielisen käännöksen tekeminen vaatisi siis vain sitoutuneen käännöstiimin, joka huolehtisi ensiksi Canvaksen tämän hetkisten tekstien kääntämisestä ja sen jälkeen käännöksen ylläpitämisestä, kun ohjelmistoon ja siten sen teksteihin tulee muutoksia.

4.3 Laajentaminen

Canvas tukee kahta eri menetelmää lisäominaisuuksien saamiseksi alustan sisäänrakennetun toiminnan lisäksi. Canvas tarjoaa sekä erittäin monipuolisen API:n että LTI-tuen. Nämä laajentamismenetelmät on esitelty yleisellä tasolla jo aliluvuissa 3.1.1 ja 3.1.2.

Canvaksen API avaa mahdollisuuden käsitellä mm. kursseja, käyttäjiä, tehtäviä, ryhmiä ja moduuleja [20]. API on niin laaja, että se antaa käytännössä rajattomat mahdollisuudet alustan toiminnallisuuden laajentamiselle. API:n käyttö tapahtuu salatun HTTPS-yhteyden yli ja tuettuja metodeja ovat GET, POST, PUT ja DELETE. Toisin sanoen alustalta voi sekä hakea tietoja (GET), lisätä uusia tietoja (POST), muokata olemassa olevia tietoja (PUT) ja poistaa asioita (DELETE). API:n käyttö vaatii alustaa ajavalta palvelimelta välimuistiohjelmiston nimeltään REDIS. Sitä ei ole tällä hetkellä asennettu, mutta tarvittaessa se on helposti lisättävissä.

Canvas tarjoaa kaksi eri periaatetta LTI-työkalujen hyödyntämiselle. Työkalun voi linkittää kurssin navigointiin tai kurssisisältöön, jolloin opiskelijat pääsevät käyttämään LTI-työkalun palveluita, mutta tuloksia ei voida palauttaa alustalle. Toinen hyödyntämisperiaate on LTI-työkalujen käyttäminen tehtävissä siten, että Canvakselle kirjautuu suoritusmerkintä opiskelijan tekemästä tehtävästä. Tällöin Canvaksessa valitaan tehtävälle arvoselutavaksi ”External tool”, joka viittaa ulkoiseen LTI-työkaluun. Tällöin opiskelijan aloittaessa tehtävän tekemisen opiskelijan selaimen joko avataan uusi, työkalun sisältävä ikkuna tai työkalunäkymä avataan Canvaksen käyttäen iframea. Kun opiskelija on saanut tehtävän valmiiksi, tehtävästä saadut pisteet palautuvat Canvaksen nähtäväksi.

LTI-työkalut täytyy erikseen konfiguroida käyttöä varten. Tämä voidaan tehdä joko kursikohtaisesti tai ylläpidon tekemänä koko Canvaksen laajuisesti, jolloin kaikki Canvaskurssit voivat käyttää työkalua. Työkalun tietoihin Canvaksessa täytyy konfiguroida muun muassa käyttäjäavain ja jaettu salaisuus, työkalun URL-osoite ja mitä tietoja alusta jakaa työkalulle käyttäjistä. Canvas tarjoaa neljä eri vaihtoehtoa käyttäjätietojen jakamiselle: ei mitään, vain e-mail, vain nimi tai julkinen, jolloin tietoina lähetetään muun muassa nimi, e-mail ja SIS ID.

4.4 Ylläpito

Canvaksen ylläpidon kannalta yksi merkittävimmistä seikoista on oppimisalustan päivittäminen. Canvaksen lähdekoodi päivittyy hyvin usein Githubin repositorion stable-haaraan. Tämä tarkoittaa sitä, että Canvakselta tulee usein versioita, jotka on määritelty varkaasti toimiviksi eli tuotantokäyttöön sopiviksi, verrattain vähän virheitä sisältäviksi. Esimerkiksi 31.1.2016 päivityksiä on tullut kolme, kuusi ja 19 päivää sitten, mikä on hyvin tiheä tahti uusien stable-versioiden luomisessa [19]. Ei kuitenkaan ole mielekasta asentaa päivityksiä niin usein, kun niitä olisi saatavilla, koska päivityksistä aiheutuu käyttökatkoja. Ylläpitotehtäviin tulee Canvaksessa myös onnistuneen intranet-tunnusten integroinnin jälkeen liittymään opettajien tunnusten luominen.

5. CANVAKSEN KÄYTTÖKOKEMUKSET

Aliluvussa 2.1 on listattu kurssit, jotka osallistuvat TTY:n MOOC-pilottikokeiluun. Kyseisten kurssien vastuuhenkilöiden haastatteluissa kävi ilmi, että vain kahdella kurssilla, Tietotekniikan laitoksen Johdatus ohjelmointiin ja Tiedonhallinnan ja logistiikan laitoksen Basics of Information and Knowledge Management, oli selkeä suunnitelma alkaa käyttää uutta valittua MOOC-alustaa. Kummallakin kurssilla oli tarkoitus järjestää tammikuussa 2015 alkava opintojakson toteutuskerta käyttäen Canvasta kurssin oppimisalustana. Siten näiden kahden kurssin oli tarkoitus olla TTY:n Canvaksen ensimmäisiä testikäyttäjää. Kumpikaan näistä kursseista ei kuitenkaan loppujen lopuksi aloittanut alustan käyttöä aikatauluongelmien takia.

Tiedonhallinnan ja logistiikan laitos otti Canvaksen koekäyttöön aiottua myöhemmin, syyslukukaudella 2015. Tietotekniikan laitoksen suunnitelmat Johdatus ohjelmointiin kurssin suhteen ovat sen sijaan muuttuneet niin, että Canvasta ei aiota ainakaan näillä näkymin ottaa ollenkaan käyttöön. Kurssilla aiotaan sen sijaan hyödyntää Aalto-yliopiston kehittämää varta vasten ohjelmoinnin opetukseen suunniteltua A+-alustaa.

Alkuperäisistä suunnitelmista poiketen alustan käytön aloittivat sen sijaan kaksi muuta tietotekniikan laitoksen kurssia; Software Engineering Methodology ja Ohjelmistoarkkitehtuurit. Kummallakin kurssilla syynä Canvaksen käyttöönotolle oli Tietotekniikan laitoksella aiemmin käytössä olleen oppimisjärjestelmän, Idlen, ylläpidon loppuminen ja siten käytöstä poistuminen. Kumpikin kurssi oli aiemmin käyttänyt Idleä, joten nyt kurssit tarvitsivat sille korvaajan. Kurssien vastuuhenkilöiden vaihtoehtona oli joko kokeilla Canvasta tai opetella käyttämään Moodlea. Tietotekniikan laitoksen opettajien keskuudessa näyttää vallitsevan vahva mielipide siitä, että Moodle on opettajalle vaikeakäyttöinen alusta. Tämän käsityksen vuoksi näiden kahden kurssin vastuuhenkilöt päättivät mieluummin kokeilla Canvasta.

Tässä luvussa kerrotaan näiden kahden ensimmäisen kurssin kokemuksista testikäytön ajalta. Käyttäjien kokemuksia voidaan pitää tärkeimpänä mittarina alustan valinnan onnistumisen arvioinnissa. Aliluvussa 5.1 kerrotaan lyhyesti näistä kahdesta alustaa ensimmäisenä testanneesta kurssista. Aliluvussa 5.2 käsitellään testikäytöstä saatua palautetta.

5.1 Canvaksen testikäyttäjät

Kahdessa seuraavassa aliluvussa kuvataan lyhyesti Software Engineering Methodologyn ja Ohjelmistoarkkitehtuurien kursseja ja se, miten kurssit hyödynsivät Canvasta opetuksessaan.

5.1.1 Software Engineering Methodology

Software Engineering Methodology on kurssi, jolla opetetaan ohjelmistotuotannon alan perustietämystä, esimerkiksi tällä hetkellä pinnalla olevien ketterien ohjelmointimenetelmien toimintatapoja. Kurssin voidaan sanoa olevan yksi Tietotekniikan laitoksen massakursseista eli sille osallistuu vuosittain paljon opiskelijoita. Kurssi koostuu luennoista, viikkoharjoituksista, harjoitustyöstä ja tentistä.

Canvasta käytettiin kurssilla viikkoharjoitusilmoittautumiseen, harjoitustyöryhmien muodostamiseen, harjoitustyön tehtävien palauttamiseen ja harjoitustyöassistentin kanssa tapaamisen varaamiseen. Canvaksen ohella kurssilla oli käytössä myös perinteinen kotisivu, jossa hoidettiin muun muassa kurssin tietojen, kuten luentoajat ja suoritusvaatimukset, jakaminen ja luentokalvojen jakelu.

5.1.2 Ohjelmistoarkkitehtuurit

Ohjelmistoarkkitehtuurit on kurssi, jossa opetetaan ohjelmistokokonaisuuksien rakenteen suunnittelemista. Myös tämä kurssi voidaan luokitella yhdeksi laitoksen massakursseista. Kurssi koostuu luennoista, pakollisista vierailuluennoista, viikkoharjoituksista, harjoitustyöstä ja tentistä. Viikkoharjoitukset koostuvat kotona ennen harjoituskertaa tehtävistä esitehtävistä sekä paikan päällä harjoituksissa suoritettavista tehtävistä.

Canvasta käytettiin kurssilla viikkoharjoitusilmoittautumiseen, harjoitusryhmiin ilmoittautumiseen, viikkoharjoitusten esitehtävien palauttamiseen ja vierailuluentojen läsnäolojen merkkäamiseen. Lisäksi kurssilla oli käytössä erilliset kotisivut, joiden kautta hoidettiin tiedottaminen ja kurssimateriaalien jakelu.

5.2 Kerätyt käyttökokemukset

Seuraavassa kahdessa aliluvussa kuvataan, minkälaisia mielipiteitä ja kokemuksia eri käyttäjiltä saatiin kerättyä testikäytön pohjalta. Ensimmäisessä aliluvussa käsitellään internet-kyselyä ja toisessa sähköpostitse saatua palautetta.

5.2.1 Käyttökokemukset kyselyn perusteella

Canvaksen käyttäjien käyttökokemuksia kartoitettiin käyttäjille jaellulla kyselyllä. Kysely toteutettiin Googlen tuottamalla Google Forms –palvelulla [11]. Palvelussa voi luoda monipuolisesti erilaisia kyselylomakkeita, joihin pääsee vastaamaan yksinkertaisesti kyselyn URL-osoitteen tuntemalla, ilman erityisiä kyselykutsujen lähettämisiä. Kyselystä tiedotettiin kummankin kurssin Canvaksen kurssisivuilla ”announcement”- eli tiedotustoimintoa käyttäen. Uuden tiedotteen luominen muun muassa aiheuttaa sähköposti-ilmoituksen lähettämisen kaikille opiskelijoille, jotka eivät ole poistaneet ilmoitusta käytöstä.

Tiedote jää myös näkyvään kurssin ”Announcements” –osioon ja etusivulle, kun etusivun on valittu näyttävän kurssin viimeisimmät aktiviteetit. Täten URL-osoitteen eli kyselylinkin jakelu tämän kanavan kautta oli järkevä valinta, koska sen voitiin olettaa saavuttavan joko sähköpostin kautta tai Canvas-kursseilla pysyvästi näkyvänä kaikki kursseiden opiskelijat. Kyselystä toteutettiin sekä suomen- että englanninkielinen versio Software Engineering Methodologyn englanninkielisyyden takia. Kyselyyn sai vastata anonymisti maaliskuun 2015 alusta toukokuun 2015 alkuun. Google Formsin toteutustavan takia kyselyyn olisi halutessaan voinut vastata useamman kerran. Vain yhteen vastauskerran salliminen olisi vaatinut käyttäjältä Googlen käyttäjätilillä kirjautumista, mitä ei voitu vaatia, koska kaikilla kurssin opiskelijoilla ei voitu olettaa olevan kyseistä käyttäjätiliä. Lisäksi kyselyyn olisi voinut vastata kuka tahansa, joka olisi saanut kyselylinkin tietoonsa. Saaduissa vastauksissa ei kuitenkaan näy viitteitä siitä, että kukaan olisi käynyt vastaamassa moneen otteeseen tai että kukaan kurssien ulkopuolinen henkilö olisi vastannut kyselyyn. Kyselyistä otetut kuvankaappaukset löytyvät liitteestä 1 ja kyselyihin saadut vastaukset löytyvät liitteestä 2.

Saatujen vastauksien perusteella käy ilmi, että kyselyssä on ollut virhe, ja kysymys ”Millä kurssilla käytät Canvasta?” ei ole ollut pakollinen. Onneksi lähes kaikki vastaajat ovat kuitenkin tähän kysymykseen vastanneet. Toinen ongelma kyselyn kanssa on se, että väärinkäsityksestä johtuen myös osa kurssien henkilökunnasta on vastannut tähän kyselyyn, vaikka kysely oli opiskelijoille suunnattu. Kurssihenkilökunnan vastaukset on kuitenkin melko helppo tunnistaa vastausten joukosta vastausten sisällön perusteella.

Palautteista voidaan tulkita muutamia toistuvia teemoja:

- Canvaksen käytettävyys on huono.
- Canvas on hidas.
- Canvas on keskeneräinen.
- TTY:llä on käytössä liian monta oppimisalustaa tai portaalia.
- Kurssien tieto (aikataulut, tehtävät, materiaalit jne.) jakautuu liian moneen paikkaan.
- Canvas on modernimman oloinen, kuin Moodle tai Idle.
- Canvas lähettää sähköposti-ilmoituksia kurssilla tapahtuneista muutoksista.

Canvaksen käytettävyyden huonoudesta tuli selvästi eniten palautetta. Osasta palautetta käy selvästi ilmi, että huono käytettävyys liitettiin siihen, että viikkoharjoituksiin ilmoittautuminen tapahtui epäloogisen navigointipolun kautta. Viikkoharjoituksiin ilmoittautuminen tapahtui valikon ”People” ja sen jälkeen ”Groups” eli ryhmät kautta. Tämä ei luonnollisesti ole intuitiivista, kun tavoitteena on johonkin ilmoittautuminen. Canvaksen ryhmien tarkoitus on se, että Canvas-kurssille osallistuvat opiskelijat voidaan jakaa pieniin ryhmiin, jotka voivat opiskella yhdessä esimerkiksi keskustellen Canvaksen ryhmäalu-

eella ja palauttaa yhteistyönä tehtyjä tehtäviä. Kurssin henkilökunta voi luoda tietyn määrän ryhmiä valmiiksi, joihin opiskelijat voivat sen jälkeen itse ilmoittautua. Tätä mahdollisuutta hyödynnettiin kursseilla viikkoharjoitusilmoittautumisten mahdollistamiseen. Kursseilla luotiin ryhmiä, jotka nimettiin esimerkiksi ”Viikkoharjoitusryhmä 1 ma klo10”. Ryhmän jäseneksi ilmoittautuminen tarkoitti kyseiseen viikkoharjoitukseen ilmoittautumista. Ryhmien muodostus -toimintoa käytettiin siis aivan toisin, kuin Canvaksen kehittäjät olivat sen suunnitelleet. Siten tässä tapauksessa ei voida sanoa, että kyse olisi Canvaksen huonosta käytettävyydestä. Osassa kyselyvastauksista kuitenkin puhutaan vain yleisesti alustan huonosta käytettävyydestä tai epäintuitiivisuudesta. Näistä vastauksista ei voi päätellä, liittyykö mielipide viikkoharjoitusilmoittautumiseen vai onko vastaaja muuten sitä mieltä. Toisaalta käytettävyyttä on myös keuhuttu, esimerkiksi vastaaja numero 16 vastaa Canvaksen positiivisista puolista näin: ” [--] Jotkut asiat on hyvin jaoteltu ja pidetty simppeleinä. lisäksi navigointi-menu helpottaa paljon asioimista.” Kaiken kaikkiaan, palautteen perusteella ei voi päätellä Canvaksen käytettävyydestä juuri mitään, sillä lähes kaikki palaute, kun jätetään ryhmäilmoittautumisiin liittyvä palaute huomiotta, on liian ympärilyöreää johtopäätösten tekemiseen.

Myös Canvaksen hitaudesta tuli jonkin verran palautetta. Osassa vastauksista ei ole aivan selvää tarkoitetaanko hitaudella navigoinnin hitautta vai sivujen latauksen ja toimintojen tekemisen hitautta. Kurssihenkilökunnasta Samuel Lahtinen on suullisesti raportoinut Canvaksen palvelimen vastauksien hitaudesta, joten on selvää, että myös opiskelijat ovat varmasti kohdanneet tämän tyyppistä hitautta. On epäily, että Canvaksen hitaus saattaisi olla selaimesta riippuvaista. Tilannetta on hyvä seurata, ja tarvittaessa yrittää palvelimen kapasiteetin parantamista hitauden vähentämiseksi. Myös Canvas ohjelmistona kannattaa pitää ajan tasalla, sillä jos syynä on oppimisalustaohjelmiston hitaus, sen parantamiseksi tehdään varmasti päivityksiä.

Canvaksen keskeneräisyydestä tuli palautetta ”Vapaa sana” -kysymykseen. Vastaaja numero 5 puhuu järjestelmän varhaisversiosta ja vastaaja numero 7 sanoo aiheesta näin: ”Seuraavalla kerralla, kun tällaisia järjestelmiä tehdään, hoitakaa testaaminen jotain muuta kautta kuin tunkemalla ne pakolliseksi kursseille.” Vaikuttaisi siltä, että opiskelijoiden keskuudessa on luultu, että kyseessä on hiljattain TTY:llä kehitetty alusta eikä alustan taustoja tai sen käytön syitä kursseilla ole tunnettu. Mielipiteitä keskeneräisyyden tunnusta ei kuitenkaan voi jättää huomiotta, sillä hyvin suunnitellun ohjelmiston ei pitäisi tuntua keskeneräiseltä, vaikka sitä luultaisiinkin tuoreeksi ohjelmistoksi.

Liian monesta oppimisalustasta tai portaalista kertoo vastaaja numero 20: ”Canvaksessa ei varsinaisesti ole vikaa. Suurin ongelma tulee siinä, että TTY:llä on käytössä noin 200 eri portaalaa, joista eri kurssien asiat tulee kaivaa.” Osittain tähän liittyen vastaaja numero 18 valittelee yksittäisen kurssin tiedon jakautumisesta eri paikkoihin: ” [--] typerää käyt-

tää kolmea eri nettisivua (POP, kotisivut, Canvas) eri kurssin asioihin.” Vastaajien mielestä siis oppimisalustoja ja portaalreja on ylipäänsä liikaa, mutta erityisen hankalaksi opiskelijan kannalta kyselyssä koetaan, että samaan kurssiin liittyvät asiat eivät keskity yhdelle alustalle tai portaalille.

Positiivista palautetta tuli vain vähän. Positiivisena puolena mainittiin esimerkiksi moderni ulkoasu verrattuna Moodleen ja Idleen. Sen lisäksi Canvaksen lähettämät sähköposti-ilmoitukset saivat kiitosta esimerkiksi vastaajalta numero 9: ” Ei tarvitse päivittäin kytätä, tuleeko jotain uutta materiaalia. Palvelua voi käydä katsomassa silloin, kun sähköpostiin tulee ilmoitus uudesta sisällöstä.” Myös kalenteritapahtumien muutoksista lähtevää sähköpostia kiiteltiin.

Ainakin Software Engineering Methodology ehti aloittaa Canvaksen käytön ennen kuin uusi mooc.tut.fi-palvelin otettiin käyttöön. Vaikka tämä osoitteen muuttuminen aiheutti-kin ylimääräistä hämminkiä kurssin opiskelijoille ja henkilökunnalle, asiasta ei kuitenkaan mainita missään palautteissa. Palvelimen vaihto onnistui sujuvasti, koska vanhan palvelimen tietokannasta tehty kopio saatiin siirrettyä uudelle palvelimelle ongelmitta.

5.2.2 Vastuuhenkilöiden palaute

Vaikkakin osa kurssien henkilökunnasta vastasi myös opiskelijoiden kyselyyn, kurssien vastuuhenkilöiltä pyydettiin kuitenkin vielä erikseen sähköpostitse yhteenveto Canvaksen käytöstä. Yhteenvedossa pyydettiin kertomaan päällimmäisenä mieleen jääneitä asioita ja mikä Canvaksessa oli hyvää tai huonoa.

Ohjelmistoarkkitehtuurin kurssilta positiivista palautetta saatiin tehtävien hallinnasta, suoritustietojen viemisestä alustalta ja opiskelijan kannalta niiden tarkastelusta alustalla. Lisäksi opiskelijoiden kanssa kommunikointi suorituksista Canvaksen kautta koettiin toimivaksi. Toisaalta negatiivista palautetta tuli alustan hitaudesta esimerkiksi viikkoharjoitusmerkintöjä kirjatessa ja ryhmien käytön ja sähköposti-ilmoitusten lähetysten asetusten säätämisen kankeudesta. Myös amerikkalaistyyllisistä ajan ja päivän ilmaisutavoista sekä viikon alkamisesta sunnuntaista tuli negatiivista palautetta. Ongelmana koettiin myös, että ryhmien jäsenet eivät saaneet toistensa sähköpostiosoitteita näkymään. Suurilla MOOC-kursseilla on ymmärrettävää, että sähköpostiosoitteita ei tule näyttää toisille osallistujille, mutta lähiopetuksen yhteydessä tällaisen pitäisi tuki olla mahdollista. Myös virheitä, joissa suoritusmerkintöjä katosi eikä kalenteri päivittynyt oikein, oli havaittu.

Software Engineering Methodologyn puolesta positiivista palautetta annettiin ryhmätöiden palautuksen onnistumisesta, mutta toisaalta harmiteltiin sitä, että ryhmien palauttamia tehtäviä pystyi lataamaan vain yhden tai kaikki, ilman mahdollisuutta valita jokin tietty osajoukko. Myös ryhmille palaveriaikojen luomisen ja ajan varaamisen helppoutta

kiiteltiin. Canvas mahdollistaa julkisten sivujen luomisen kurssille, ja tätä ominaisuutta pidettiin hyvänä, vaikka sitä ei kyseessä olevalla toteutuskerralla käytettykään.

5.3 Testikäytön arviointi

Negatiivista palautetta tuli paljon verrattuna positiivisen palautteen määrään. Tästä voidaan päätellä, että alustan valinta ei ole onnistunut ainakaan Software Engineering Methodology ja Ohjelmistoarkkitehtuureja ajatellen. Onnistumisen arvioinnissa on kuitenkin otettava huomioon, että nämä kaksi kurssia eivät osallistuneet vaatimusmäärittelyyn sopivan alustan valitsemiseksi. Kurssien toteutustapa huomioiden, jos kurseilta olisi kerätty vaatimukset, Canvas tuskin olisi toteuttanut niitä, tai se olisi toteuttanut ne jonkin LTI:tä tai APIa hyödyntävän työkalun avulla. Suuri osa negatiivisesta palautteesta selittyy Canvaksen ominaisuuksien epätarkoituksenmukaisesta käytöstä, eli ominaisuuksia on hyödynnetty käyttötarkoituksiin, joihin ne eivät kunnolla sovellu. Tästä on seurannut, että ominaisuuksien käyttö on tuntunut epäintuitiiviselta ja kankealta.

Sekä Software Engineering Methodologyn että Ohjelmistoarkkitehtuurin tietojen jakautumista moneen eri paikkaan olisi voinut helpottaa käyttämällä Canvasta myös tiedottamiseen ja kurssimateriaalien jakamiseen erillisten kotisivujen sijasta. Canvas olisi mahdollistanut nämä asiat helposti. POP-oppilastietojärjestelmän käyttö kurssi- ja tentti-ilmoittautumiseen sekä virallisten oppimistavoitteiden ja kurssiaikataulujen julkaisemiseen tulee jatkossakin olemaan pakollista, sillä Canvas ei näihin tarpeisiin pysty järkevästi vastaamaan. Seuraavalla toteutuskerralla tämän testikäytön jälkeen Software Engineering Methodology oli hyödyntänyt Canvaksen julkisia sivuja informaation jakautumisen vähentämiseksi.

Huomioon ottaen sen, että alustan käyttäjät ilmaantuivat yllättäen eikä heidän tarpeitaan ollut huomioita alustan valinnassa, testikäyttö sujui yllättävän hyvin. Suurimmaksi ongelmaksi näytti muodostuvan ryhmäilmoittautumisen löytämisen vaikeus. Tämä on kuitenkin pieni ongelma siihen nähden, että Canvaksen käyttö onnistui kuitenkin jotenkin, esimerkiksi ilman palvelun kaatuilua.

6. PYTHON-KOODIN AUTOMAATTIARVOSTELIJA

Tässä luvussa kerrotaan Johdatus ohjelmointiin –kurssia varten suunnitellun automaattisen arvioinnin työkalun suunnitteluratkaisuihin ja toteutetuista ja suunnitelluista ominaisuuksista. Automaattisen arvioinnin työkalulla viitataan tässä yhteydessä ohjelmaan, joka kykenee arvioimaan opiskelijoiden ohjelmakoodia ennalta määriteltyjen kriteereiden valossa itsenäisesti ilman kurssihenkilökunnan puuttumista arviointiin. Automaattisen arvioinnin käyttö Johdatus ohjelmointiin –kurssilla on välttämätöntä kurssihenkilökunnan tekemän arvioinnin sijaan opiskelijoiden kurssin aikana tuottaman ohjelmakoodin määrän takia. Uutta automaattisen arvioinnin työkalua tarvittiin, koska kurssin oli tarkoitus siirtyä käyttämään Canvasta, ja Canvas ei itsessään tarjonnut sopivaa arviointityökalua. Automaattiarvostelijaa kehitettiin Tietotekniikan laitoksella tämän diplomityön yhteydessä. Automaattiarvostelija yhdistyy Canvukseen käyttäen aliluvussa 3.1.1 esiteltyä LTI-protokollaa. Käytännössä automaattiarvostelija on Canvaksesta riippumaton, itsenäinen verkkopalvelu, joka voisi hyödyntää mitä tahansa LTI:tä tukevaa oppimisalustaa, kun oppimisalusta vain pystyy lähettämään LTI-kutsussa arvostelijan tarvitsemat LTI-spesifikaatioissa määritellyt tiedot. Automaattiarvostelija näyttää opiskelijalle ohjelmointitehtävän, joka opiskelijan tulisi suorittaa. Arvostelijaan on sisällytetty mahdollisuus Python-koodin kirjoittamiseen ja ajamiseen verkossa, mikä poistaa tarpeen erillisten ohjelmointieditorin ja ajoympäristön asentamiselle. Kun opiskelija on todennut kirjoittamansa koodin täyttävän tehtävänannon vaatimukset, opiskelija palauttaa työnsä arvosteltavaksi ja saa palautteen vastauksensa suoriutumisesta arvostelijan automaattisista testeistä reaaliaikaisesti. Tehtäville voidaan määrittää maksimirytykskerrat, jotta opiskelijat testaavat ohjelmansa toimintaa ennen tarkastukseen jättämistä, ja aukioloaika, jota ennen ja jonka jälkeen tehtävä ei ole saatavilla.

Kirsti Ala-Mutka on tehnyt jaottelun eri tekijöistä, joita tietokoneavusteisella arvioinnilla (computer aided assessment, CAA) voidaan arvioida sekä dynaamisessa eli oppilasohjelman ajona vaativassa arvioinnissa tai staattisessa arvioinnissa, joka voidaan suorittaa oppilaskoodia tarkastelemalla [3]. Tässä diplomityössä suunniteltu automaattiarvostelija on hyvin yksinkertainen ja se testaa Ala-Mutkan luettelemista tekijöistä vain dynaamista arviointia vaativaa toiminnallisuutta ja tehokkuutta ja staattista arviointia vaativaa erikoisominaisuudet-tekijää. Toiminnallisuuden testaaminen tarkistaa, että ohjelma toimii annettujen vaatimusten mukaisesti, tehokkuuden testaamisessa mitataan esimerkiksi ohjelman suoritukseen kulunutta aikaa ja erikoisominaisuudet voivat olla esimerkiksi tietyn sanan käyttäminen ohjelmakoodissa [3]. Tehokkuutta voidaan testata tässä arvostelijassa siten, että ohjelman suoritukseen asetetaan tavoiteaika, jossa ohjelman täytyy valmistua ennen ohjelman suorituksen keskeyttämistä, jolloin tehtävästä annetaan hylätty arvosana.

Tämä ominaisuus myös lopettaa mahdolliset ikuiset silmukat. Erikoisominaisuuksien testaaminen tässä arvostelijassa toteutetaan hakemalla ennalta määritellyjä sanoja oppilaskoodista, jolloin voidaan esimerkiksi tarkastaa, onko oppilas käyttänyt ohjelmassa for-silmukkaa silloin, kun tehtävän annossa on niin käsketty tekemään.

Automaattiarvostelijan kehitys on keskeytetty, koska Johdatus ohjelmointiin –kurssilla päätettiin ryhtyä käyttämään A+-oppimisalustaa Canvaksen sijasta. Automaattiarvostelija löytyy Githubin repositoriosta [namiskuukkel/django-grader](#) [13].

6.1 Käytetyt teknologiat

Tässä ohjelmistossa käytettyjä teknologioita ovat Django, Docker, Ace Editor ja LTI. LTI on esitelty jo aliluvussa 3.1.1, mutta muut teknologiat esitellään seuraavissa aliluvuissa.

6.1.1 Django

Tätä ohjelmistoa tehtiin käyttäen Python-kielistä sovelluskehystä Djangoa [5]. Django on websovelluskehitykseen erikoistunut, hyvin monipuolinen sovelluskehys. Django valittiin käyttöön, koska Tietotekniikan laitoksella kehitettiin samaan aikaan myös toista ohjelmointitehtävien palautukseen liittyvää ohjelmistoa käyttäen samaa sovelluskehystä. Täten Djangoa valinnalla pyrittiin helpottamaan tulevaa ylläpitotyötä. Automaattiarvostelija on myöskin teknisesti hyvin yksinkertainen ohjelmisto, jolla ei ole esimerkiksi mitään tiukkoja reaaliaikavaatimuksia, joten tämän takia käytettävän ohjelmointikielen tai sovelluskehityksen valintaan ei tarvinnut kiinnittää erityistä huomiota. Django on myös melko suosittu sovelluskehys, ja sitä käyttävät muun muassa merkittävät palvelut, kuten NASA, Instagram ja Pinterest [6].

6.1.2 Docker

Ala-Mutkan mukaan oppilaan kirjoittaman ohjelman ajaminen on riskialtis tehtävä [3]. Hänen mukaansa ohjelmassa voi olla virheitä tai jopa pahantahtoisia ominaisuuksia, jotka saavat ohjelman yrittämään toimintoja, jotka voisivat aiheuttaa vahinkoa ajoympäristönsään. Tästä syystä Ala-Mutkan mukaan dynaamisen arvioinnin olennainen edellytys on turvattu ajoympäristön eli hiekkalaatikon tarjoaminen oppilaskoodin ajamiselle. Tässä automaattiarvostelijassa Docker [7] toteuttaa kyseisen hiekkalaatikon. Docker on virtuaalikoneen tapaisesti toimiva ohjelmisto, jota voidaan käyttää suljetun ohjelmakoodin ajoympäristön luomiseen oppilaskoodin ajamiseksi.

Erotuksena virtuaalikoneeseen Docker on paljon nopeampi, koska sen toteutusperiaate on erilainen. Toisin kuin virtuaalikoneet, jotka sisältävät kokonaisen käyttöjärjestelmän ja ovat vahvasti toisistaan eristettyjä, Docker-kontit eli virtuaalikonetta vastaavat suoritus-

yksiköt sisältävät vain käyttötilanteessa tarvittut komponentit, ja eri kontit jakavat isäntäkoneen käyttöjärjestelmän ytimen [8]. Docker käynnistyy hetkessä, kun virtuaalikoneen käynnistys kestää käyttöjärjestelmän käynnistyksen verran aikaa. Dockerin toiminta perustuu imageihin ja kontteihin (container). Imaget ja kontit voi ymmärtää olio-ohjelmoinnin luokka-analogian kautta. Luokat kuvaavat joukon ominaisuuksia ja oliot ovat instansseja, jotka toteuttavat luokan ominaisuudet. Docker imaget kuvaavat joukon ohjelmistoja ja Docker-kontit ovat instansseja, joissa nämä ohjelmistot käytännössä toimivat. Imagessa voidaan esimerkiksi kertoa, että imagen pohjalta tehty kontti ajaa Ubuntu-käyttöjärjestelmää ja sen päällä pyörivää Pythonia ja tietokantaa. Docker mahdollistaa Dockerfile-kuvaustiedostojen käytön, joissa voi sanella käytettävät ohjelmistot ja muita kontin parametreja, esimerkiksi kontin kanssa jaettavat tiedostot. Kuvaustiedostojen käyttö tekee imagejen hallinnasta ja käytöstä erittäin helppoa, sillä kaikki muutokset voi tehdä kuvaustiedostoon ja imagen voi luoda tiedostosta automaattisesti yhdellä komentorivikomennolla.

Dockerin käyttö automaattiarvostelijassa perustuu toiseen hyvin samantapaiseen projektiin, ja sen saamiin turvallisuusarvioihin. Projekti on nimeltään Codecube, ja se on saatavilla Githubin repositoriossa [hmarr/codecube](https://github.com/hmarr/codecube) [23]. Codecuben tarkoitus on tarjota käyttäjilleen ohjelmointiympäristö verkossa, jolloin käyttäjien ei tarvitse asentaa itselleen kaikkia ohjelmakoodin ajamiseen tarvittavia ajoympäristöjä. Codecube käyttää käyttäjiensä koodin ajamisen hiekkalaatikkona Dockeria. Valitettavasti Codecuben kehitys näyttää pysähtyneen vuonna 2013.

Docker toimii siten, että kontin sisältä käsin ei pysty näkemään isäntäjärjestelmästä mitään muuta, kuin erikseen kontin käynnistyksen yhteydessä määritellyt tiedostot. On kuitenkin aina mahdollista, että tietoturva voi pettää. Jos kontista pääsisi murtautumaan ulos, kontin käyttäjä eli automaattiarvostelijan tapauksessa opiskelija voisi päästä tekemään toimenpiteitä Dockeria ajavalla palvelimella tai tarkistelemaan tiedostoja, joiden pitäisi olla kontin käyttäjän ulottumattomissa. Tällaisia kontista murtautumisia ei kuitenkaan ole raportoitu uudemmissa versioissa Dockeria, joten voidaan luottaa, että ulos murtautuminen ei ole mahdollista. Tästä syystä Dockerin turvallisuuden uskotaan olevan oppilaskoodin ajamiseen riittävää. Automaattiarvostelijan tietoturvaa parantaa se, että kontin käyttöaika on rajattu, koska arvostelija hävittää kontin tietyn ajan jälkeen mahdollisten oppilaskoodin ikuisten silmukoiden tappamiseksi. Täten Dockerista murtautumiseen on vain rajattu aika. Lisäksi kontti käynnistetään ajamaan Python-tulkkiä, joten oppilaskoodin täytyy ensin kyetä sammuttamaan tulkin suoritus. Tulkin lopetus taas voidaan estää kieltämällä tulkin lopettavien komentojen, kuten `quit()` ja `exit()`, käyttö oppilaskoodissa Django lomakkeiden validointimekanismilla, joka kuvataan tarkemmin aliluvussa 6.1.4. Tulkin lopettavan tai muuten kielletyn määritellyn sanan löytyminen oppilaskoodista ai-

heuttaa lokimerkinnän syntymisen, eikä oppilaskoodia tämän jälkeen ajeta. Yksi turvallisuutta lisäävä tekijä, jota ei vielä ole automaattiarvostelijaan toteutettu, mutta joka tulisi lisätä, on Python-tulkin ajaminen ilman pääkäyttäjän oikeuksia.

6.1.3 Ace Editor

Ace Editor on monipuolinen Javascript-kielellä toteutettu usealle ohjelmointikielelle sopiva tekstieditori [1]. Editori voidaan ottaa käyttöön millä tahansa verkkosivulla sisällyttämällä sivun lähdekoodiin tietty HTML:n script-elementti, ja editori toimii millä tahansa Javascriptiä tukevalla selaimella, kunhan selaimen käyttäjä on sallinut Javascriptin käytön. Ace Editoria on käytetty erittäin laajasti verkossa eri palveluissa ohjelmakoodin kirjoittamiseen tai esittämiseen. Editoria käyttävät esimerkiksi Github, Codecademy ja Khan Academy [2], joista jälkimmäiset ovat merkittäviä ohjelmoinnin verkko-opetuksen tarjoajia. Ace Editorin käytön yleisyydestä johtuen se valittiin myös automaattiarvostelijan käyttöön mahdollistamaan opiskelijoille ohjelmakoodin kirjoitus verkossa. Editorina Ace Editor mahdollistaa monia asennettavista ohjelmointieditoreista tuttuja ominaisuuksia, kuten ohjelmointikielen avainsanojen värikoodaus, automaattinen sisennys ja esimerkiksi silmukoiden tai funktioiden sulkeminen pois näkyvistä tilan säästämiseksi.

6.1.4 Valmiit ohjelmistokomponentit

Edellä kuvattujen teknologioiden lisäksi automaattiarvostelija hyödyntää myös joitakin pienempiä ohjelmistokomponentteja toteutuksessaan.

LTI:n käyttöön Django-ympäristössä on otettu mallia Universitat Obert de Catalunya projektista, joka löytyy Githubin repositoriosta `hdn8/django-uocLTI` [9]. Automaattiarvostelijan LTI-kansiosta löytyvä `views.py` ja `__init.py__` on muokattu kyseisen projektin vastaavista tiedostoista ja `utils.py` ja `tests.py` ovat sellaisenaan projektista lainattuja. `__init.py__` sisältää LTI:n käyttäjäavaimen ja jaetun salaisuuden, joten tätä tiedostoa ei oikeassa käyttötilanteessa tule julkaista.

Django-uocLTI:n mallia noudattaen automaattiarvostelija käyttää ohjelmistokirjastoa nimeltään `ims-lti-py`, joka löytyy Githubin repositoriosta `tophatmonocle/ims_lti_py`. Kyseinen kirjasto helpottaa LTI:n käyttöä Python-kielellä ohjelmoitaessa. Tämä komponentti käyttää edelleen ohjelmistokirjastoja `lxml` ja `python-oauth2`, joista ensimmäinen helpottaa XML:n ja HTML:n käsittelyä ja toinen LTI:n OAuth-kirjautumista.

Ace Editorin käytön helpottamiseksi automaattiarvostelija käyttää myös komponenttia nimeltään `Django-ace`, joka on saatavilla Githubin repositoriassa `bradleyayers/django-ace`. Komponentti mahdollistaa sen, että Ace Editorin kanssa saadaan helposti käyttöön Django lomakkeiden (form) validointi. Django tekee kaikille lomakkeille validoinnin automaattisesti aina, kun sovelluksen käyttäjä lähettää jonkin lomakkeen. Validoinnissa

tarkastetaan ensinnäkin, että tiettyyn lomakekenttään annettu arvo on kentälle määriteltyyn tyyppiin nähden sopiva, esimerkiksi sähköpostiosoitteentään on syötetty sähköpostiosoite. Toiseksi, validoinnissa tarkistetaan lomakkeen kentille erikseen ohjelmoijan määrittelemät säännöt. Esimerkiksi sähköpostiosoitteen vahvistuskentälle voidaan määrittellä, että kentän arvon täytyy olla sama kuin ensimmäiseen sähköpostiosoitteentään syötetty arvo. Kun Ace Editorin esittämiseen käytetään Django-acea, editori toimii Django-lomakkeen kaltaisesti, ja editoriin kirjoitettuun koodiin voidaan soveltaa Django-validointimekanismia. Tämän automaattiarvostelijan yhteydessä Ace Editorin validointisäännöksi merkitään aliluvussa 6.1.2 kuvattujen Python-tulkin lopettavien sanojen löytyminen. Jos kiellettyjä sanoja löytyy, validointi epäonnistuu, mikä voidaan todeta automaattiarvostelijan koodissa `form.is_valid()` -funktioilla.

6.2 Arkkitehtuuri

Django ja LTI yhdessä sanelevat hyvin pitkälti tämän ohjelmiston arkkitehtuurin. Django toteutustapa ohjaa käyttämään Model-View-Controller-suunnittelumallin (MVC), kaltaista Django Model-Template-View-mallia (MTV). Seuraava esittely MVC-mallista ja mallin toteutumisesta Djangoissa on hyvin yksinkertaistettua. Varsinkin Django osalta yksinkertaistetaan paljon, koska kuten edellä on todettu, Django on hyvin monipuolinen sovelluskehys, ja se mahdollistaa paljonkin joustoa ohjelmoijan tarpeiden mukaan. MVC-mallin kuvaus taas on pidetty hyvin yksinkertaisena, koska eri lähteet määrittelevät mallin hyvinkin eri tavoin. MVC-suunnittelumallissa ohjelmiston vastuut on jaettu kolmeen eri osaan: malliin, näkymään ja kontrolleriin. Malli sisältää sekä käyttäjälle näytetyt tiedot sekä logiikan, jolla tietoja muutetaan käyttäjän toimien tuloksena, näkymä vastaa tiedon esittämisestä käyttäjälle ja kontrolleri käsittelee käyttäjän tekemät toiminnot [22]. Django MTV-mallissa oikeastaan vain malliosuus vastaa pitkälti perinteisen MVC-mallin vastuunjakoja. Django näkymässä valitaan, mitä tietoa näytetään ja kuinka se näytetään. Täten MVC-näkymän vastuut jakautuvat Djangoissa näkymille ja templateille. Django näkymä vastaa tiedon hakemisesta mallilta ja oikeiden templatejen valinnasta. Template taas vastaa tiedon esittämistavan kuvaamisesta esimerkiksi HTML-sivuna. Kontrollerin toiminnoista Djangoissa on vastuussa itse sovelluskehys. Se esimerkiksi ohjaa ohjelman ajon käyttäjän syöttämän URL-osoitteen perusteella oikealle näkymäfunktiolle, joka vuorostaan valitsee oikeat tiedot näytettäväksi käyttäjälle. [10] MTV-mallista seuraa, että sovelluksen toiminnallisuus jaetaan tyypillisesti malli- (`models.py`), template- (`.html`) ja näkymätiedostoihin (`views.py`).

LTI:n käyttö määrittelee oman osansa arkkitehtuurista. LTI tuo sovellukseen logiikkaa, joka tarkistaa tietyt asiat LTI-käynnistyksen oikeellisuuden kannalta, ja logiikkaa, joka lähettää opiskelijan arvosanoja takaisin oppimisalustalle. Kun käyttäjä avaa LTI-työkalua käyttävän tehtävän Canvasissa, Canvas lähettää työkalulle HTTPS-pyyntö, jossa on

iso joukko LTI-spesifikaation määrittelemiä tietoja POST-parametreina. Canvaksen lähettämistä parametreista on esimerkki kuvassa 3. LTI:n käyttö määrittää ohjelman arkkitehtuurissa sen, että ohjelman käytön täytyy alkaa LTI-kutsulla. Djangoa käytettäessä tämä tarkoittaa sitä, että verkkosovelluksen käyttö täytyy aloittaa tietyistä URL-osoitteista, jolloin tietty näkymäfunktio tekee käytön aloitukseen tarvittavat toimenpiteet. Nämä toimenpiteet ovat automaattiarvostelijan tapauksessa tarkistuksia LTI:n lähettämien parametrien oikeellisuudesta tai läsnäolosta pyynnössä. Lähetetyistä parametreista täytyy tarkistaa muun muassa se, onko parametreissa välitetty `oauth_signature` eli OAuth-kirjautumiseen liittyvä allekirjoitus oikea niin ikään parametrina välitettyyn käyttäjäavaimen ja käyttäjäavaimen liittyvään yhteiseen salaisuuteen nähden. Canvaksen lähettämistä LTI-parametreista tarkistetaan myös käyttäjän sähköpostin läsnäolo, sillä automaattiarvioija käyttää käyttäjän sähköpostiosoitetta käyttäjän yksilöivänä tunnisteena eli käyttäjätunnukseksi. Lisäksi automaattiarvostelija käyttää LTI-parametreista LTI:n käynnistyskontekstin yksilöivää tunnistetta (`context_id`), Canvas-kurssin nimeä (`context_title`), tehtävän nimeä (`resource_link_title`) ja arvosanan palautusosoitetta (`lis_outcome_service_url`). Kurssin ja tehtävän nimet tarvitaan vain oikeiden tehtävien näyttämiseen ja tiedon esittämiseen. Käynnistyskontekstin tunniste ja arvosanan palautusosoite taas ovat olennainen osa LTI:n toimintalogiikkaa, ja niitä tarvitaan, jotta automaattiarvostelija voi palauttaa arvosanoja Canvukseen. Arvosanan takaisinlähetyt tuo arkkitehtuuriin funktion, joka koostaa tehtävän palautukseen liittyvistä tiedoista LTI:n mukaisen XML:n takaisin Canvakselle lähetettäväksi.

| Variable | Value |
|--|--|
| context_id | "dce15f5e1dd897af7842757e0162cd71c7a65424" |
| context_label | "Tadaal" |
| context_title | "Tadaal" |
| custom_canvas_assignment_points_possible | "2" |
| custom_canvas_assignment_title | "Tehtävä 2" |
| custom_canvas_enrollment_state | "active" |
| ext_ims_lis_basic_outcome_url | "https://mooc.tut.fi/api/lti/v1/tools/3/ext_grade_passback" |
| ext_outcome_data_values_accepted | "url,text" |
| ext_roles | "urn:lti:instrole:ims/lis/Administrator,urn:lti:instrole:ims/lis/Instructor,urn:lti:instrole:ims/lis/Student,urn:lti:role:ims/lis/Instru" |
| launch_presentation_document_target | "iframe" |
| launch_presentation_locale | "en-GB" |
| launch_presentation_return_url | "/courses/1/assignments" |
| lis_outcome_service_url | "https://mooc.tut.fi/api/lti/v1/tools/3/grade_passback" |
| lis_person_contact_email_primary | "riitta.hietaranta@tut.fi" |
| lis_result_sourcedid | "3-1-23-1-ef29d99669e957f2421ccc87be058e2673418e9f" |
| lti_message_type | "basic-lti-launch-request" |
| lti_version | "LTI-1p0" |
| oauth_callback | "about:blank" |
| oauth_consumer_key | "test" |
| oauth_nonce | "lieQyEUu2sxgyZIV9HqPqwE0AjYB1WoAWnfiOPZ6KU" |
| oauth_signature | "EJfzNybuHgk7mHVCmGGMHSQNKvs=" |
| oauth_signature_method | "HMAC-SHA1" |
| oauth_timestamp | "1434387395" |
| oauth_version | "1.0" |
| resource_link_id | "1a2fa2e0a39d72e292e17e19eb65affeb0dbe31d" |
| resource_link_title | "Tehtävä 2" |
| roles | "Learner,Instructor,urn:lti:instrole:ims/lis/Administrator" |
| tool_consumer_info_product_family_code | "canvas" |
| tool_consumer_info_version | "cloud" |
| tool_consumer_instance_contact_email | "canvas@listmail.tut.fi" |
| tool_consumer_instance_guid | "db5feb7d22b812b6e333b6103ddd75e383a56029.mooc.tut.fi" |
| tool_consumer_instance_name | "TUT Canvas" |
| user_id | "7b2b4ca3746efd7068b99bbfedd83089a83bd5" |
| user_image | "https://secure.gravatar.com/avatar/2aeb1361332f9ec66be9c60da6b438f7?s=50&d=https%3A%2F%2Fmooc.tut.fi%2Fimages%2Fmessages%2Favatar-50.png" |

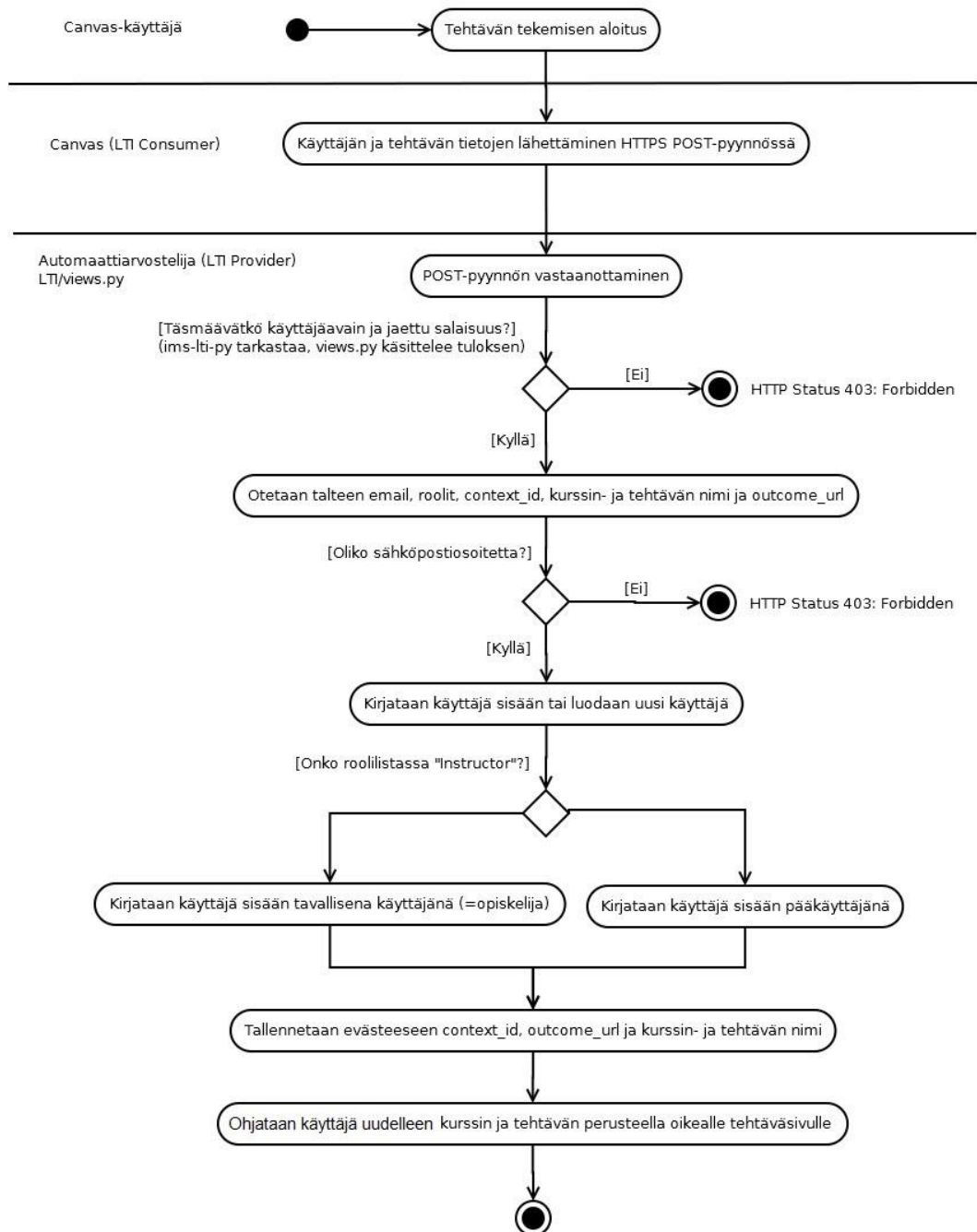
Kuva 3 Esimerkki Canvaksen LTI-käynnistyksen parametreista

6.3 Toteutettu toiminta

Tässä aliluvussa kuvataan automaattiarvostelijan toimintaa tärkeimpien ominaisuuksien osalta. Esimerkiksi tietokantatoimenpiteitä ja virhetarkistuksia jätetään käsittelemättä epäolennaisina kokonaiskuvan hahmottamisen kannalta.

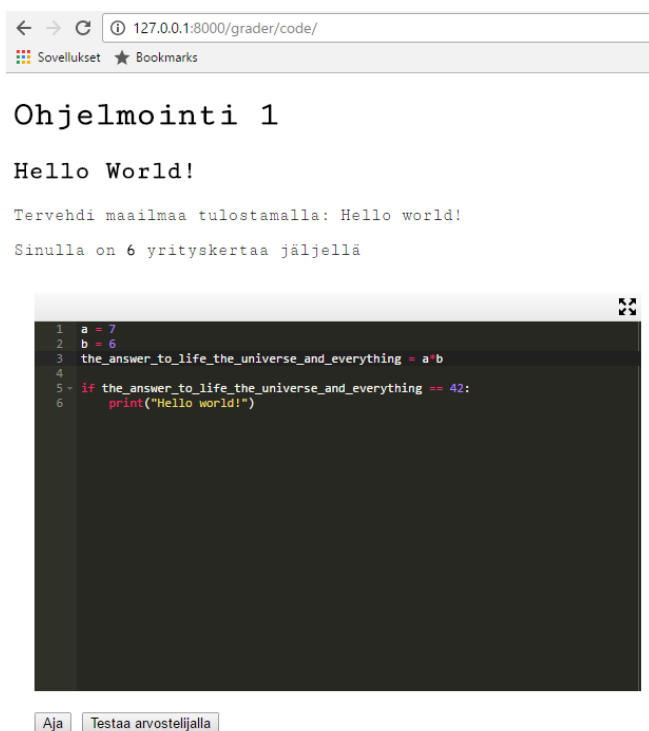
Automaattiarvostelijan käyttö alkaa aina Canvaksesta. Kuvassa 4 on esitetty aktiviteetti-kaavio, joka ilmaisee, kuinka automaattiarvostelijan käynnistyminen etenee Canvakselta automaattiarvostelijaan. Kyseessä on LTI-käynnistyksen, joten kuvassa näkyy paljon edellisessä aliluvussa esiteltyä LTI:n toimintaa ja termistöä. Vaakaviivoilla on eroteltu prosessin eri toimijat. Ylin toimija, Canvas-käyttäjät, käynnistää toimenpiteen omalla selaimellaan. Kuvan esittämä toiminta on käyttäjälle näkymätöntä, kunnes toiminta etenee johonkin loppupisteistä, jotka on esitetty mustavalkoisilla ympyröillä. Kaksi oikealla olevaa päätepistettä ovat virhetilanteita, joihin ei normaalitilanteessa pitäisi päätyä, mutta joiden jälkeen automaattiarvostelijan käyttöä ei voi jatkaa ennen vikatilanteen korjaamista. Kun päädytään alimpaan loppupisteeseen, automaattiarvostelijan varsinainen ajo- ja arvostelutoiminta pääsee alkamaan. Kuvassa esiintyy kysymys instuctorista eli opettajasta LTI:n roolilistassa. Kun opettaja on rooleissa läsnä, automaattiarvostelija kirjaa käyttäjän sisään pääkäyttäjänä. Pääkäyttäjällä on oikeus tarkastella alustalla olevia kursseja ja tehtäviä ja

luoda uusia kursseja ja tehtäviä tietokantaan. Kurssia, tehtäviä ja oppilaskansioita varten tarvittavat kansiot ja tiedostot joudutaan luomaan käsin arvostelijaa ajavalle palvelimelle. Näiden kansioiden sijainti ilmoitetaan kursseja ja tehtäviä luotaessa.



Kuva 4 Automaattiarvostelijan käynnistyminen

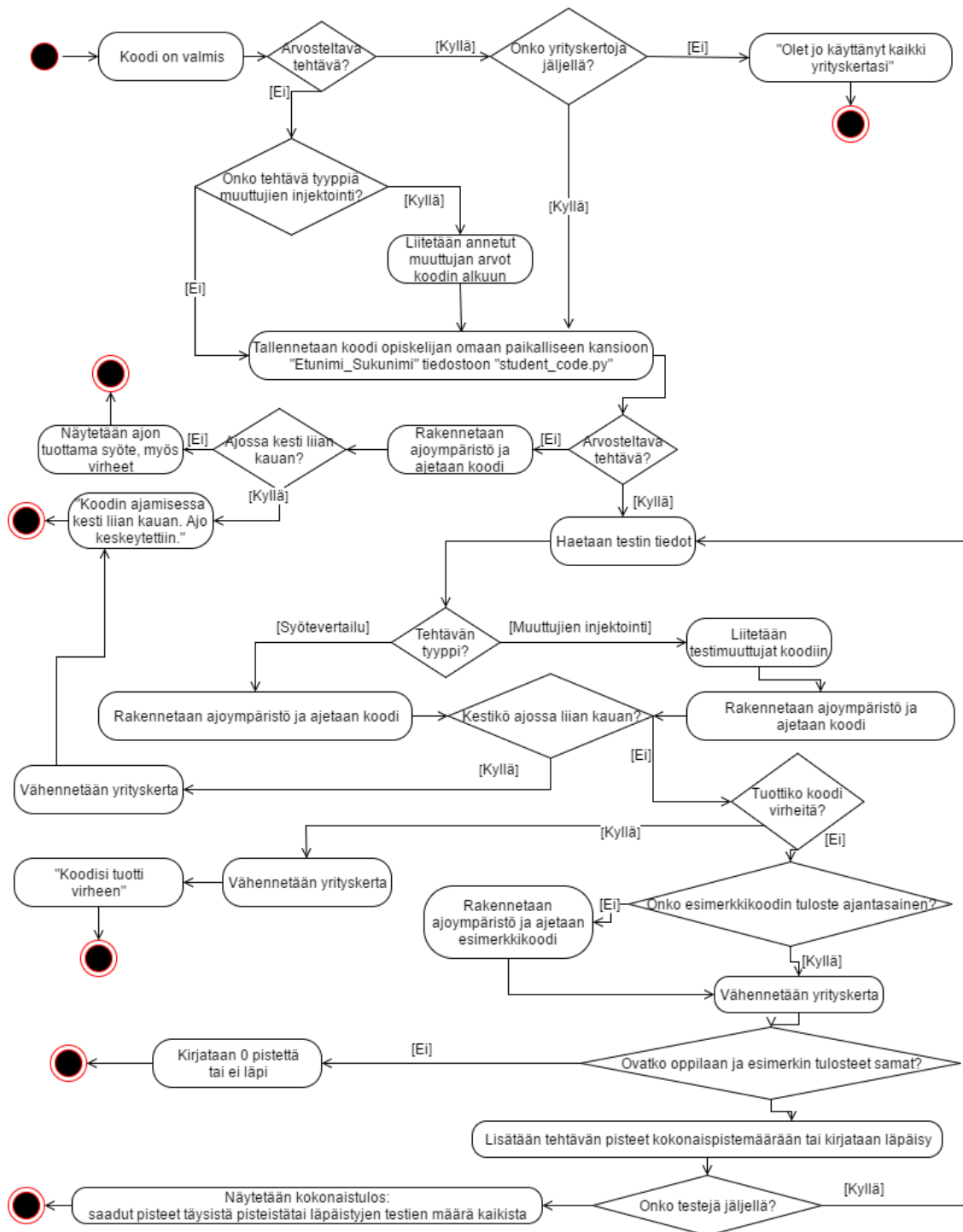
Kun automaattiarvostelija on käynnistynyt, arvostelija näyttää LTI-käynnistyksen parametrien määrittelemän tehtävän. Tällainen näkymä on esitetty kuvassa 5.



Kuva 5 Esimerkki koodausnäkyvästä

Ensimmäinen ja yksinkertaisin toteutettu tehtävätyyppi on opiskelijan koodin tuottaman tulosteen ja esimerkkikoodin tuottaman tulosteen vertailu. Kuvan 5 tehtävä on tämän tyyppinen. Toinen arvostelijaan toteutettu tehtävätyyppi on muuttujien injektointi. Tässä tehtävässä testataan opiskelijan koodin toimivuutta erilaisilla muuttujien arvoilla. Tehtävänannossa määritellään esimerkiksi, että tehtävässä tulee olemaan kokonaislukutyypiset muuttujat *a* ja *b*. Opiskelijan tulee jättää muuttujat määrittelemättä koodissaan, sillä tehtävää tarkastaessa, arvostelija injektoi kyseiset muuttujat testiarvoilla opiskelijan koodiin. Tällaisen tehtävän yhteydessä näkymässä on varsinaisen koodi-ikkunan yläpuolella toinen pieni ikkuna, johon opiskelija voi määritellä omia muuttujiaan testiajoja varten. Kolmas aloitettu, mutta ei loppuun asti ehtinyt tehtävätyyppi on avainsanojen etsiminen koodista. Tehtävänannossa voidaan esimerkiksi vaatia, että ohjelmassa käytetään `for`-silmuksia, jolloin yksi arvostelijan ajamista testeistä voi tällöin katsoa, löytyykö ”`for`”-avainsanaa koodista.

Kun opiskelija on saanut koodinsa valmiiksi, hän voi valita joko koodin ajamisen, jolloin automaattiarvostelija vain ajaa ohjelman tai arvostelun, jolloin arvostelija tekee kaikki kyseiselle tehtävälle määritellyt testit ja antaa ohjelmalle arvosanan. Tavallisen ajon ja arvostelun tarkempi eteneminen kahden toteutetun tehtävätyypin osalta on havainnollistettu kuvassa 6. Sekä ajossa että arvostelussa automaattiarvostelija tekee Dockerin osalta seuraavia toimenpiteitä ajon ja arvostelun yhteydessä: Ensimmäiseksi kopioidaan



Kuva 6 Automaattiarvostelijan ajon- ja arvostelun aikaiset tehtävät

Dockerfile opiskelijan kansioon, tai tarkastuksen ohessa tarvittaessa myös esimerkikoodin kansioon, kun esimerkikoodista ei ole ajankohtaista tulostetta. Toiseksi Dockerfilen pohjalta tehdään Docker-image. Lopulta imagesta luodaan kontti. Oppilaskoodin sisältämä tiedosto jaetaan kontin kanssa, ja käynnistyttyään kontti ajaa koodin. Dockerin tuottama tuloste eli opiskelijakoodin tuloste ohjataan onnistuneessa ajossa result.txt-tiedos-

toon, ja koodin tuottaessa virheitä error.txt-tiedostoon. Periaatteessa uuden imagen luomisen jokaisen ajon yhteydessä ei pitäisi olla tarpeellista, koska imagesta tehty kontti tuhoutuu ajon loputtua. Jossakin vaiheessa ohjelman kehitystä näytti kuitenkin siltä, että uudessa kontissa saattoi näkyä samasta imagesta aiemmin tehtyyn konttiin lisättyjä tiedostoja. Tästä syystä uusi levykuva luodaan varmuuden vuoksi, jotta automaattiarvostelijaa käyttävä opiskelija ei vahingossakaan voisi nähdä toisen opiskelijan kirjoittamaa koodia. Tämän varotoimenpiteen tarpeellisuus tulee tarkistaa, jos ohjelmaa vielä kehitetään. Uuden levykuvan luomisenkin kanssa koko kontin käynnistys ja koodin ajo on kuitenkin niin nopeaa, että käytännössä tästä varotoimenpiteestä ei ole haittaa.

Koodin tavallisen ajamisen jälkeen, opiskelija voi muuttaa koodiaan ja ajaa sen uudestaan rajoittamattomasti tai lähettää tehtävän arvosteluun. Arvostelun jälkeen opiskelija voi valita yrittävänsä uudestaan tai lähettävänsä arvostelun tuloksen Canvakselle. Arvosanan palautusta ei kuitenkaan vielä ole onnistuneesti toteutettu.

6.4 Jatkokehitys

Nyt toteutetun automaattiarvostelijan toiminnot ovat melko suppeat, ja ohjelmiston laatuakin voisi parantaa monin eri tavoin niin visuaalisuuden ja käytettävyyden kuin arkkitehtuurin ja hyvien koodauskäytänteiden, kuten riittävän kommentoinnin, kannalta. Tässä aliluvussa esitetään ideoita, joilla automaattiarvostelijaa voisi jatkossa laajentaa ja parantaa.

Tärkein automaattiarvostelijaa laajentava jatkokehitysidea on Gitlabin yhdistäminen arvostelijan toimintaan. Gitlab on versionhallintapalvelu, joka on otettu käyttöön joillakin Tietotekniikan laitoksen kursseilla oppilaskoodien tallettamiseksi. Automaattiarvostelijaan Gitlab yhdistyisi siten, että arvostelija tallentaisi oppilaskoodit Gitlabissa oppilaiden henkilökohtaisiin repositorioihin Gitlabin API:n kautta paikallisen tallennuksen ohella. Paikallinen tallennus tarvitaan mahdollisten tietoverkko- tai ohjelmisto-ongelmien varalta, jotta oppilaskoodi ei virhetilanteessa katoaisi lopullisesti. Gitlabiin tallentamalla kaikki omat koodit ovat opiskelijalle helposti saatavilla. Jos Johdatus ohjelmointiin ja muut ohjelmoinnin kurssit käyttävät Gitlabia oppilaskoodien tallentamiseen samalla tavalla opiskelijoiden omiin repositorioihin, opiskelijalle alkaa kurssi kurssilta kertyä koodiportfolioa hänen tekemistään tehtävistä ja harjoitustöistä. Tästä portfolioista opiskelija voi seurata omaa kehittymistään ohjelmoijana, ja hän voi myös halutessaan käyttää sitä työnhaun apuna.

Gitlabia voidaan käyttää oppilaskoodin tallentamisen lisäksi myös automaattiarvostelijan tehtävän kuvauksien, esimerkivastausten ja testitapausten tallentamiseen. Automaattiarvostelija voi hakea tehtävämateriaalit versionhallinnasta Gitlabin API:a käyttäen. Tällöin

kaikki kurssin tehtävämateriaalit ovat kätevästi keskitetyksi saatavissa ja käsiteltävissä Gitlabissa.

Tärkein automaattiarvostelijan käytettävyyden kannalta laatua parantava jatkokehitys-idea on graafisen käyttöliittymän tekeminen opettajaa varten. Graafisella käyttöliittymällä luotaisiin ja muokattaisiin tehtäviä ja testitapauksia. Nyt tehtävät ja testitapaukset täytyy tehdä jollakin tekstieditorilla automaattiarvostelijan ymmärtämään muotoon. Kuvassa 7 on esitetty yhden tehtävän kuvaileva tiedosto `description.py`. Kaikki yhtäsuuruusmerkkien vasemmalla puolella olevat tekstit täytyy kirjoittaa täsmälleen oikein tai automaattiarvostelija toimii väärin, antamatta järkevää virheilmoitusta. Myös esimerkiksi muuttujien ”scale” ja ”type” arvojen täytyy olla täsmälleen oikein kirjoitettu. Tämä tapa on erittäin virhealtis kirjoitusvirheille ja ihmisen on työstä tuottaa tällaisia tiedostoja. Graafinen käyttöliittymä poistaisi vääränmuotoisista tehtävä- ja testikuvauksista aiheutuvat virheet, koska automaattiarvostelija loisi kuvaustiedostot ohjelmallisesti käyttäjän graafiseen käyttöliittymään syöttämien tietojen perusteella. Jos tiedostojen luominen ja muokkaaminen siirretään automaattiarvostelijan vastuulle, on kyseenalaista, kannattaako tiedostoja enää silloin tallentaa Gitlabiin. Gitlabissa tiedostoja voitaisiin muokata helposti, mikä taas voi johtaa siihen, että tiedostot päätyvät väärään formaattiin ihmisen tekemän muutoksen takia. Gitlabiin tallentaminen toki takaa tiedostoille versionhallinnan tuomat hyödyt, kuten tehtyjen muutosten tallentumisen, muutosten peruuttamisen mahdollisuuden ja varmuuskopioinnin automaattisuuden.

Automaattiarvostelijan laadukkuutta automaattisen arvostelun työkaluna voisi parantaa lisäämällä valikoimaan uuden tyyppisiä testejä. Eräs lisättäväksi suunniteltu testitapaus on tehtävätyyppi, jossa opiskelija tekee funktion annettujen ohjeiden mukaisesti. Opettajan luoma ohjelma kutsuu opiskelijan funktiota, ja vertailee sitten ohjelman ajon tulostetta esimerkivastauksen vastaavan funktion käytön tuottamaan tulosteeseen. Tällaisella testillä voitaisiin opettaa ensinnäkin funktioiden tekemistä, mutta myös työelämässä tarpeellista jonkun toisen tekemän koodin kanssa yhteen toimivan ohjelmayksikön luomista. Myös toisten opiskelijoiden ja oman koodin arvioiminen suhteessa esimerkivastaukseen voisi olla lisättävä tehtävätyyppi. Näiden lisäksi automaattiarvostelijassa voitaisiin ottaa käyttöön jokin lukuisista valmiista ohjelmakoodin tyyliä tarkastavista työkaluista. Arvostelijaan voisi myös lisätä mahdollisuuden useiden kooditiedostojen lataamiseen alustalle, jolloin uutena tehtävätyyppinä saataisiin laajempien ohjelmakokonaisuuksien arvostelu. Tällöin kooditiedostot kirjoitettaisiin jollakin suurempien kokonaisuuksien hallintaan paremmin soveltuvalla ohjelmointiympäristöllä arvostelijaan integroidun Ace Editorin sijasta.

Uusien tehtävätyyppien tekemisen lisäksi automaattiarvostelijaa voitaisiin kehittää edelleen myös tekemällä siitä tutkimuksen apuväline. Aalto-yliopistossa on tehty tutkimusta

```

1  #!/usr/bin/python
2  # -*- coding: UTF-8 -*-
3
4  tests = ['test1']
5  success = "Jee, hyvin meni!"
6  #scale = "pass"
7  #passes_required = 17
8  scale= "numeric"
9  total_points = 2
10 points_required = 2
11 #example_code = "example.py" Not used for now! Always name the example file like this
12
13 test1 = { "type": "compare_output",
14           "name": "Testi 1",
15           "description": "Tulostetaan stringi",
16           "timeout": 15,
17           "test_results": ["Jokin meni pieleen", "expected"],
18           "test_result_limits": [5, 0],
19           "points": 2,
20           }
21
22 '''test2 = { "type": "find_property",
23             "name": "Testi 2",
24             "description": "Onko käytetty testissä määriteltyjä mekanismeja",
25             "find": ["moi", "print"],
26             "timeout": 15,
27             "test_results": ["Jokin meni pieleen", "expected"],
28             "test_result_limits": [5, 0],
29             "points": 3,
30             }'''

```

Kuva 7 Testin kuvaustiedosto

siitä, miten ohjelmointia opitaan seuraamalla ohjelmointitehtävien tekemistä. Juha Helminen et al. tutkimuksessa ”How Do Students Solve Parsons Programming Problems? — An Analysis of Interaction Traces” ohjelmoinnin opiskelijat ratkaisivat ohjelmointitehtäviä, joissa valmiista koodipätkistä järjestelemällä kootaan toimiva ohjelma. Opiskelijoiden vuorovaikutus ohjelmointiympäristön kanssa nauhoitettiin, jotta opiskelijoiden oppimista voitaisiin tutkia [12]. Tämän automaattiarvostelijan käyttämä Ace Editor mahdollistaisi samanlaisen vuorovaikutuksen nauhoituksen. Ace Editorin API:n kautta voidaan seurata editorin tekstin muutoksia, joten vaikka jokaisen napin painalluksen voisi tallentaa. Monipuolista APIa voitaisiin varmasti hyödyntää myös uusien tehtävätyyppien kehittämisessä.

Automaattiarvostelijasta voitaisiin tehdä myös yleiseen levitykseen sopiva lisäämällä Pythonin ohelle muita ohjelmointikieliä. Myös LTI-riippuvuus voitaisiin poistaa yleiskäyttöisyyden lisäämiseksi. Tällöin arvostelijaa käytettäisiin ilman yhdistymistä oppimisolustoihin. Tämä toisaalta vaatisi myös lokalisoinnin käyttöönoton, jotta käyttökieli

voisi olla jokin muu kuin suomi. Lokalisointia ei nyt ole huomioitu ohjelmaa tehtäessä, mutta Django mahdollistaa sen melko helposti.

Kaiken kaikkiaan automaattiarvostelijassa olisi paljon potentiaalia, jos sen kehitystä jatkettaisiin. Jo valmistunut osuus ei ole missään nimessä ideaalisesti toteutettu, mutta ainakin osa toteutusperiaatteista ovat hyviä, kuten Dockerin, Ace Editorin ja LTI:n käyttö. Näitä toteutusperiaatteita ja kehitysideoita hyödyntäen voitaisiin saada aikaan monipuolinen ja hyvä automaattiarvostelija.

7. YHTEENVETO

Tämän diplomityön tutkimuskysymys on: ”Miten valitaan TTY:lle sopiva MOOC-alusta? Kuinka suoritettujen valinnan onnistumista voidaan arvioida? Miten valittu alusta voidaan sovittaa erikoistuneita ominaisuuksia vaativan kurssin tarpeisiin sopivaksi?” Pääkysymykseen sopivan alustan valinnasta on vastattu luvuissa 2 ja 3. Vastaus on, että ensin määritellään mitä tarkoittaa TTY:lle sopiva ja tämän jälkeen selvitetään, mitä eri vaihtoehtoja on olemassa ja kuinka sopivuuden kriteeristö toteutuu eri vaihtoehtoisissa. Kriteeristön täyttymisen perusteella voitiin valita TTY:lle sopivin MOOC-alusta. Luvussa 2 on kerrottu tätä diplomityötä edeltäneen kandidaatintyön vaatimusmäärittelyn tuloksista, jotka muodostavat sopivuuden kriteeristön. Luvussa 3 on kuvattu kriteeristöä vastaan vertailu. Pelkkää kriteeristöä vastaan vertailtuna testikäyttöön MOOC-alustan valinta testikäyttöä varten ei ollut itsestään selvä, koska vertailuista alustoista sekä valittu Canvas että NovoEd pärjäsivät lähes yhtä hyvin. Tiukka käyttöönottoaikataulu huomioiden Canvasin valinta oli kuitenkin selvä.

Tutkimuskysymyksen toisen osan vastaus valinnan onnistumisesta on moniosainen. Toisaalta valinnan voidaan nähdä onnistuneen jo siinä mielessä, että ylipäänsä löydettiin MOOC-alusta, joka täytti tärkeimmät kriteerit ja niin monia muiden tärkeysluokkien kriteereistä. Valinnan onnistumisen puolesta puhuu, että myös Hasso Plattner Institute päätyi samaan valintaan samantapaisista lähtökohdista, ja he ovat onnistuneesti käyttäneet Canvasia MOOC-käytössä. Valinnan onnistumiseen vaikuttavat myös luvussa 4 mainitut tekijät. Vaikkakin Canvasin periaatteissa tuki TTY:n intranet-tunnuksien käyttöä, käytännössä tunnuksien käyttöönotto alustalla ei ollutkaan helppoa. Myös lokalisaation ongelmat tuottivat käyttäjille päänvaivaa, kuten käyttäjäkokemuksia kerätessä kävi ilmi. Intranet-tunnus- ja lokalisaatio-ongelmien kannalta MOOC-alustan valinta ei onnistunut.

Tärkein valinnan onnistumiseen vaikuttava tekijä on kuitenkin alustan käyttäjien tyytyväisyys. Tätä käsiteltiin luvussa 5. Luvussa kuvataan testikäyttöön asennettujen Canvasin ensimmäisten käyttäjien kokemuksia alustasta kyselytutkimuksen perusteella. Lisäksi kokemuksia kerättiin alustaa käyttäneiden kurssien henkilökunnalta erikseen sähköpostitse. Testikäyttövaiheeseen siirryttäessä alkuperäiset suunnitelmat olivat muuttuneet siten, että vaatimusmäärittelyssä haastatelluista kurseista ne kaksi, Johdatus ohjelmointiin ja Basics of Information and Knowledge Management, jotka aikoivat aloittaa valittavan alustan käytön tammikuussa 2015, eivät aloittaneetkaan alustalla. Sen sijaan alustan käytön aloittivat ensimmäisenä kurssit Ohjelmistoarkkitehtuurit ja Software Engineering Methodology. Näiltä kurseilta kerätyt käyttäjäkokemukset olivat suurelta osin negatiivisyyteisiä. Canvasin valinnan ei voida täten nähdä olleen onnistunut. Onnistumisen arvi-

oimisessa tulee kuitenkin huomioida, että Ohjelmistoarkkitehtuureilta ja Software Engineering Methodologylta ei kerätty vaatimuksia vaatimusmäärittelyssä. Kurssien toteutustapa huomioiden, jos kursseilta olisi kerätty vaatimukset, Canvas tuskin olisi toteuttanut niitä tai se olisi toteuttanut ne jonkin LTI:tä tai APIa hyödyntävän työkalun avulla. Suuri osa negatiivisesta palautteesta selittyy Canvaksen ominaisuuksien epätarkoituksenmukaisesta käytöstä eli kun ominaisuuksia on hyödynnetty käyttötarkoituksiin, joihin ne eivät kunnolla sovellu. Tästä on seurannut, että ominaisuuksien käyttö on tuntunut epäintuitiiviselta ja kankealta. Valinnan onnistumista ei siis voida käyttäjäkokemusten osalta yksiselitteisesti arvioida, sillä valinta kyllä täyttää sille asetetut kriteerit, mutta alustan käyttäjät ovat olleet tyytymättömiä, vaikkakin syynä on osittain se, että käyttäjien tarpeita ei ole huomioitu valinnassa. Kokonaisuudessaan voidaan sanoa, että selvää vastausta valinnan onnistumiseen ei saatu. Valinta on osittain onnistunut ja osittain epäonnistunut. Valinnan onnistumista voitane arvioida vasta, kun alustan käyttökokemuksia saadaan niiltä kursseilta, joiden tarpeita ajatellen alusta on valittu.

Viimeiseen tutkimuskysymyksen osaan ”Miten valittu alusta voidaan sovittaa erikoistuneita ominaisuuksia vaativan kurssin tarpeisiin sopivaksi?” vastattiin luvussa 6. Luvussa kuvattiin Johdatus ohjelmointiin -kurssia varten toteutetun LTI-työkalun arkkitehtuuria, komponentteja ja ominaisuuksia. Vaikkakin työkalun toteutus jäi kesken kurssin muuttuneiden suunnitelmien takia, työkalun toteutusperiaatteita on silti hyödyllistä kuvata siltä varalta, että Canvakselle tai jollekin muulle oppimisalustalle halutaan joskus toteuttaa vastaavanlaista projektia.

LÄHTEET

- [1] Ace, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://ace.c9.io/>
- [2] Projects Using Ace, Ace, verkkosivu. Saatavissa (viitattu 29.06.2015): <http://ace.c9.io/#nav=production>
- [3] K. M. Ala-Mutka, A survey of automated assessment approaches for programming assignments. *Computer Science Education*, 15(2):83-102, June 2005.
- [4] AM/PM time and the First day of the week, Canvas LMS Users Google Groups – foorumi, verkkosivu. Saatavissa (viitattu 28.06.2016): <https://groups.google.com/forum/#!topic/canvas-lms-users/1C7JyZ2r1ys>
- [5] Django, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://www.djangoproject.com/>
- [6] Sites Using Django, Django Project, Django, verkkosivu. Saatavissa (viitattu 11.06.2015): <https://www.djangoproject.com/start/overview/>
- [7] Docker, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://www.docker.com/>
- [8] What is Docker?, Docker, verkkosivu. Saatavissa (viitattu 29.06.2015): <https://www.docker.com/what-docker>
- [9] N. Finney, Django-uocLTI-repositorio, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://github.com/hdn8/django-uocLTI>
- [10] N. George, The Django Book: The Model-View-Controller Design Pattern, verkkosivu. Saatavissa (viitattu 30.06.2015): <http://djangobook.com/model-view-controller-design-pattern/>
- [11] Google Forms, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://www.google.com/forms/about/>
- [12] J. Helminen et al., How Do Students Solve Parsons Programming Problems? — An Analysis of Interaction Traces, International Computing Education Research Workshop, Auckland, New Zealand, September 09 - 11, 2012, ACM, New York, NY, pp. 119-126. Saatavissa: <http://dl.acm.org/citation.cfm?doid=2361276.2361300>
- [13] R. Hietaranta, Django-Grader-repositorio, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://github.com/namiskuukkel/django-grader/tree/master/grader>

- [14] R. Hietaranta, Ideas forum: Allow the use of an external grading tool, verkkosivu. Saatavissa (viitattu 28.04.2015): <http://helpdesk.eliademy.com/forums/193876-ideas-forum/suggestions/6106051-allow-the-use-of-an-external-grading-tool>
- [15] R. Hietaranta, TTY:n MOOC-pilottiprojektien tarpeiden määrittäminen MOOC-alustan suhteen, kandidaatintyö.
- [16] What is learning tools interoperability?, IMS Global Learning Consortium, verkkosivu. Saatavissa (viitattu 10.04.2015): <http://developers.imsglobal.org/>
- [17] About IMS Global Learning Consortium, IMS Global Learning Consortium.. Saatavissa (viitattu 10.04.2015): <http://www.imsglobal.org/background.html>
- [18] Canvas LMS –repositorio, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://github.com/instructure/canvas-lms/>
- [19] Production Start, Instructure, verkkosivu. Saatavissa (viitattu 23.06.2015): <https://github.com/instructure/canvas-lms/wiki/Production-Start>
- [20] Canvas LMS - REST API and Extensions Documentation, Instructure. Saatavissa (viitattu 22.10.2016): <https://canvas.instructure.com/doc/api/index.html>
- [21] ISO/IEC 25010:2011(en) (Esikatselu): 4.2 Product Quality Model, ISO, verkkosivu. Saatavissa (viitattu 17.06.2015): <https://www.iso.org/obp/ui/#iso:std:35733:en>
- [22] A. Leff, J. T. Rayfield, Web-Application Development Using the ModelNiewlController Design Pattern, Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International, Seattle, Washington, USA, September 4- 7, 2001, IEEE, pp. 118 – 127.
- [23] R. Marr, Codecube-repositorio, verkkosivu. Saatavissa (viitattu 11.03.2017): <https://github.com/hmarr/codecube>
- [24] C. Meinel, M. Totschnig, C. Willems, openHPI: Evolution of a MOOC platform from LMS to SOA, Proceedings of the 5th International Conference on Computer Supported Education (CSEDU), tammikuu, 2013. Saatavissa: https://hpi.de/fileadmin/user_upload/fachgebiete/meinel/papers/Web-University/2013_Meinel_CSEDU.pdf
- [25] Avoimen rajapinnan määritelmä, Open API -työryhmä, 11.10.2014. Saatavissa (viitattu 18.10.2016): <http://avoinrajapinta.fi/>
- [26] S. Ortega et al., D 2.1. Analysis of existing MOOC platforms and services, E-learning, Communication and Open-data, 31.07.2014. Saatavissa: <http://ecolearning.eu/wp->

content/uploads/2014/07/D2.1.Analysis-of-existing-MOOC-platforms-and-services-_v1.2.pdf

[27] Basic Interaction, Shibboleth, verkkosivu. Saatavissa (viitattu 11.12.2016): <https://wiki.shibboleth.net/confluence/display/CONCEPT/Home>

[28] Student information system, Wikipedia, päivitetty 09.06.2015, verkkosivu. Saatavissa (viitattu 16.06.2015): https://en.wikipedia.org/wiki/Student_information_system

LIITE 1: CANVAKSEN KÄYTTÖKOKEMUSKYSELY

Tässä liitteessä ovat kuvankaappaukset kummastakin eri versiosta Canvasta koskevaa kyselyä. Kuvassa Kuva 8 on suomenkielinen kysely, joka jaeltiin sekä kursseilla Software Engineering Methodology että Ohjelmistoarkkitehtuurit. Kuvassa Kuva 9 on sama kysely englanniksi. Koska tämä kysely jaeltiin vain kurssilla Software Engineering Methodology, kyselystä oli poistettu ensimmäinen kysymys ”Millä kurssilla käytät Canvasta?”.



Kysely Canvaksesta

Tämä kysely kartoittaa käyttäjäkokemuksia koskien TTY:n uutta oppimisalustaa Canvasta. Kyselyn vastauksia tullaan hyödyntämään Canvasta käsittelevässä diplomityössä sekä opetuksen kehittämisessä Canvaksella. Kyselystä vastaa Riitta Hietaranta (riitta.hietaranta@tut.fi).

Millä kursseilla käytät Canvasta?

- Software Engineering Methodology
- Ohjelmistoarkkitehtuurit
- Osallistun molemmille kursseille

Jos valitsit "Osallistun molemmille kursseille", määrittele tarvittaessa seuraavissa kysymyksissä kumpaa kurssia vastauksesi koskee.

Onko sinulla ollut ongelmia Canvaksen kanssa? Kuvaile minkälaisia.

Ongelmia voivat olla esimerkiksi kurssijärjestelyihin liittyvät asiat, käytettävyysongelmat tai hitaus.

Onko Canvaksessa joitakin hyviä puolia? Mitä ne ovat?

Voit esimerkiksi verrata Canvasta Moodleen tai Idleen.

Onko Canvaksen käyttöön annettu riittävästi opastusta?

- On
- Ei
- En osaa sanoa

Onko jotakin mihin olisit tarvinnut lisää opastusta?

Vapaa sana

Jos vielä jäi jotakin mielenpäälle Canvaksen liittyen, siitä voit kirjoittaa tähän.

Kuva 8 Canvas-kysely suomeksi



Questionnaire about Canvas

This questionnaire collects user experiences about TUT's new learning platform Canvas. The answers will be used for a Master's Thesis about Canvas and for improving the teaching on Canvas.

The person responsible for this questionnaire is Riitta Hietaranta (riitta.hietaranta@tut.fi)

Have you experienced any problems with Canvas? Please describe the problems.

Problems could be for example related to how the course is organized on Canvas, usability issues or slowness.

Are there any good properties on Canvas? Please describe them.

You can for example compare Canvas with Moodle or some other learning platform you've used.

Has there been enough guidance on how to use Canvas?

- Yes
 No
 I don't know

Is there something you would have needed more guidance with?

Is there anything else on your mind you'd like to share about Canvas?

LIITE 2: KÄYTTÖKOKEMUSKYSELYN TULOKSET

Tilansäästämiseksi seuraavissa taulukoissa käytetään kursseista lyhenteitä SEM Software Engineering Methodology -kurssista ja OHAR Ohjelmistoarkkitehtuureista. Vastaajat on numeroitu, jotta saman ihmisen vastauksia eri kysymyksiin on helpompi tarkastella. Vastaa-
taajanumeroiden perään on merkitty H, kun vastauksen epäillään olevan henkilökunnan jäsenen tekemä.

| # | Millä kurs- silla käytät Canvasta? | Onko sinulla ollut ongelmia Canvaksen kanssa? Kuvaile minkälaisia. |
|-----|--|---|
| 1 | SEM | Toimintoja etsittiin välillä porukalla, mutta taisivat sitten löytyä. |
| 2 H | OHAR | Välillä joutuu odotteleen vähän liikaa (5-10s), se on kärsimättömälle ihmiselle liikaa. Henkilökunta ei oo osannu pistää kaikkee aina avautuun ajoissa. |
| 3 | OHAR | Kyseessä on käytettävyysongelma tai pikemminkin se, että tiedot vaikeasti saatavilla/käytettävyys sekava. Käymieni keskustelujen pohjalta, myös kurssin vetäjien/tiedon hallinnoijien toimenpiteet ovat epäloogiset (esim. tehtävän lisäys paikasta x ja tämän julkaisu/aukaisu paikasta y). |
| 4 | OHAR | Hidas ja paikoin sekava. |
| 5 | OHAR | Kaikki ilmottautumiset (viikkoharjoituksiin, harjoitustyöryhmään) oli hyvin 'mielenkiintoisen' polun takana, jota en olisi koskaan löytänyt ellei sitä oltaisi erikseen luennolla kerrottu. Lisäksi alusta ei ilmeisesti tue minkäänlaista tapaamisajankohdan valintaa kun piti välipalautus varata doodlen kautta. Ja lopuksi myös harjoitustehtävien läpikäynti on ollut ainakin ensimmäisissä harjoituksissa vaikeaa alustan jäykkyyden takia. |
| 6 | OHAR | Ensimmäisenä silmään iskevä käytettävyysongelma on se, että käyttäjärühmät löytyvät People-välilehden kautta. Kurssillaolijaa kiinnostaa huomattavasti vähemmän lista kurssilla olevista ihmisistä kuin käyttäjärühmät, joten on kummallista että käyttäjärühmät on tuollaisen mutkan takana. |

| | | |
|---------|------|--|
| | | Myös se että ajanvaraus palautetilaisuuksiin toimisi yhtä sujuvasti suoraan Canvasin kautta kuin se toimi Idlessä olisi suuri plussa. |
| 7 | OHAR | <p>Käytettävyys aivan perseestä. En tajua miten tällaista voi käyttää tai edes harkita moodlen / idlen korvaamiseen. Myös kurssi tulee saamaan tästä palautetta, koska ollaan pakotettu opiskelijat käyttämään tällaista keskeneräistä järjestelmää.</p> <p>Viikkoharjoituksiin ilmoittautuminen ei todellakaan ole loogista ja sattumalta löysin paikan jostain People kohdan alta. Myös viikkoharjoitusten (ja muu palautus) toimi huonosti ja oli kankeaa. Muita ominaisuuksia ei juuri tullut käytettyä. Muutenkin pyrin pysymään kaukana järjestelmästä. En edes tajua miten joku kuvittelee tämän olevan jotenkin parempi järjestelmä, mitä tällähetkellä on käytössä. Myös idle on mielestäni ihan turha joten ihmettelen, miksi tällainen järjestelmä oli pakko tehdä.</p> |
| 8 | OHAR | <p>- Palautuksessa ei voi antaa sekä teksti (text entry), että tiedosto (file upload) palautusta. Sääli, sillä usein arkkitehtuurit kursseilla on pariin otteeseen tarvittu esim. piirrosta ykköstehtävään ja pientä kirjoitelmaa kakkostehtävään.</p> <p>- Canvasta pitäis voida käyttää intratunnuksen avulla. Jotenkin ylimääräisen salasanan keksiminen palvelua varten tuntui tarpeettomalta.</p> |
| 9 | SEM | Palvelusta on vähän vaikea löytää asioita. Linkit eivät ole selkeästi esillä tai sellaisessa paikassa, josta ensimmäisenä tulee katsottua. En myöskään koe, että tämä olisi tuonut kalenteria lukuunottamatta mitään lisäarvoa tai uusia/parempia ominaisuuksia verrattuna vaikkapa Moodleen. |
| 10 H | SEM | <p>Opiskelijalista ei saa tulostettua (en ainakaan on keksinyt miten). Joissakin pikkujutuissa ei ollut raja-arvojen tarkistuksia (sai syöttää negatiivisia lukuarvoja ;-)).</p> <p>Typerä amerikkalainen (EN/ISO-standardin vastainen) kalenteri; maanantain pitäisi olla viikon ensimmäinen päivä, ja kellonajat 01-23/24. En ole keksinyt mistä nuo saisi vaihdettua eurooppalaiseksi/suomalaiseksi.</p> |

| | | |
|---------|----------|---|
| | | Ht-tiedostojen palautuksessa ei voi rajata latausta vain assari-X:lle tuleviin töihin (voi ladata kaikki tai yhden). |
| 11 | SEM | Käytettävyydessä moodlea epäselvempi. |
| 12 | SEM | <p>Kurssin alussa oli hieman ongelmia, koska viikkoharjoitusryhmiin ilmoittautuminen ja harjoitustyöryhmän muodostaminen eivät olleet täysin selviä.</p> <p>Jotta ryhmiin pääsi, piti ensin valita People ja vasta sitten Groups.</p> <p>Harjoitustyöryhmän muodostamisessa hämmensi hieman se, että ryhmiä oli jo luotu valmiiksi aika monta. Yleensä opiskelijat ovat tehneet itse uuden ryhmän.</p> <p>Canvasin oletusohjeistus ryhmistä ei oikein soveltunut tälle kurs- sille. Course groups -sivulla Canvas ohjeistaa ryhmien käyttöä seuraavasti: " Groups are a good place to collaborate on projects or to work out schedules for things like study sessions. Every group gets a calendar, a wiki, discussions, and a little bit of space to store files. Groups can collaborate on documents, or even schedule web conferences. It's really like a mini-course where you can work with a smaller number of students on a more focused project."</p> |
| 13 | Molemmat | <p>-Ryhmäläisten sposteja ei saanut mistään (kumpikin kurssi)</p> <p>-Spostispämmi toimii vähän hassusti (joskus tulee yllättäen iso läjä sposteja pienestä asiasta (toisen assarin ajoista taisi tulla 3 vai 4 sposti) ja joskus ei tule odotetuista asioista (kuten uusi viestiketju ryhmän omalla keskustelupalstalla).</p> |
| 14 H | OHAR | <p>"Aakkosjärjestys" hirveetä kakkaa, suoritusten kirjaaminen tuskaa. Speed Graded hitaampi ku ei speed versio, varsinkin jos tarvii vaan pisteyttää. Miksi näyttää myös ne opiskelijat, jotka eivät palautta- neet mitään? (tähän joku optio)</p> <p>Kalenteri & aika pitäis saada järkeväksi, ameriikan mallinen on ihan sieltä.</p> <p>Välillä vähän hämärän peitossa, onko joku juttu tallentunut vai ei.</p> |

| | | |
|----|------|---|
| | | <p>Grades-kohdassa kun avaa sen pikanäkymän, josta näkee mitä palautettu & vaihtaa pistemäärän, se ei näy päänäkymässä.</p> <p>Sivujen odottelu nyppii välillä, varsinkin tehtävien muokkauksen yhteydessä.</p> <p>Ryhmäkohtaisesti asioiden järkkäily sujuvammaksi ja mahdollisuus saada helposti ulos esim. ryhmän sähköpostiosoitteet.</p> <p>Mistä näkee ryhmän tekemät palautukset? (muuten kun katso ensin peoplen kautta ryhmän jäsenet sitten gradesin kautta jonkun ryhmäläisen kautta palautus)</p> |
| 15 | | Järjestelmästä ei löydy yhtikäs mitään. Aivan järkyttävän huono käyttöliittymä. |
| 16 | OHAR | Canvas on välillä vähän hidas ja aluksi oli vaikea löytää etsittävät asiat/vastuukset. Esim.ryhmän muodostaminen oli mielestäni vähä oudossa paikassa. |
| 17 | SEM | <p>Opiskelijat kyselivät, näkeekö mistään mitä aikavaroituksia he ovat tehneet (esim. ht-välinäytöt eli tapaamisajat). Eivät ole ainakaan vielä löytäneet mistään Canvaksesta tuota tietoa.</p> <p>Palautetut dokumentit kyllä kuulemma näkyvät.</p> |
| 18 | SEM | Aivan kamala ja epäintuitiivinen käyttöliittymä, vaikea löytää asioita (esim. omia ryhmiä), hirvittävä määrä turhaa sähköpostia, type-rää käyttää kolmea eri nettisivua (POP, kotisivut, Canvas) eri kurssin asioihin. En ymmärrä miksi tietotekniikan laitos sooloilee kurssisisältöjensä esittämisen kanssa kun kaikilla muilla kursseilla riittää seurata yhtä paikkaa (POP tai Moodle riippuen kurssista) ja voi olla varma että saa kaiken ja ajantasaisen tiedon kurssista. Opiskelijan kannalta aivan kamalaa. |
| 19 | OHAR | Koko homma on kovin epäloogisesti järjestetty/asioita on hankala löytää ja sivut tuntuvat lataavan pikkasen hitaasti. Korostetut ilmoitukset näyttävät kirjoituskentiltä ja se jos joku häiritse pitkään. |

| | | |
|----|----------|---|
| 20 | OHAR | People -> Groups ei tuntunut kovin loogiselta paikalta harjoitukseen ilmoittautumiselle. Tehtävien palautukseen olisi kiva saada mahdollisuus lisätä liitetiedostoja (tavallisen tekstin lisäksi). |
| 21 | Molemmat | |
| 22 | OHAR | Joskus näyttää filut kahteen kertaan listassa. Valikot hyvin sekavia. Groups people alivalikon alla? Miksi overdue tehtävät näytetään ensin? Ei minua kiinnosta enää ne. Tehtävien palautuksessa tekstiä on hyvin vaikea jäsentää annetuilla työkaluilla. |
| 22 | Molemmat | Canvasin käyttöliittymä on epäjohdonmukainen ja välillä hidas. Asioita on vaikeampi löytää kuin esimerkiksi IDLEstä. Uusi lisäjärjestelmä vanhojen rinnalle on mielestäni rasittava. Canvas ei toimi kaikilla selaimilla niin hyvin, kuin sen pitäisi. Ryhmät ovat sekavia ja tehtävien palauttaminen on harmillisen hankalaa; oikean palautusalueen löytäminen vaatii n. 5-10 eri näkymän käyttöä. |
| 23 | Molemmat | <ul style="list-style-type: none"> - toimii liian hitaasti - OHARissa samoja asioita joutui palauttelemaan useampaan paikkaan - monesti tuntuu vaikealta löytää haluttuja asioita |

| | |
|----|--|
| # | Have you experienced any problems with Canvas? Please describe the problems. |
| 24 | No |
| 25 | |
| 26 | <ul style="list-style-type: none"> - The interface sometimes hides things, such as the selection of times for the review sessions (I had to access this part through the link in an email). - There are too many sections but not enough content to be displayed. Perhaps having a dashboard would be better. - I had already unreserved a time slot for a review session but, when I checked my calendar, the notification for that event was still there. |

| # | Millä kursilla käytät Canvasta? | Onko Canvaksen käyttöön annettu riittävästi opastusta? | Onko jotakin mihin olisit tarvinnut lisää opastusta? |
|----|---------------------------------|--|---|
| 1 | SEM | En osaa sanoa | |
| 2 | H OHAR | On | |
| 3 | OHAR | Ei | Oikeastaan kaikkeen mutta eipä kurssin vetäjäkään ollut ko. alustaa ennen käyttänyt. Eipä tietty opiskelija näkymä ainakaan kauhean mullistavan vaikea ole eli ei ihan hirveästi sellaista mitä voisi oppia... |
| 4 | OHAR | On | |
| 5 | OHAR | On | Ylipäättään se että tämänlaisen alustan käyttö vaatii teekkareiden opastamista toimiakseen on iso miinus. Tämänkaltainen palvelu pitäisi suunnitella niin intuitiiviseksi, että sen käytön voisi aloittaa kirjautumalla sisään ilman erillistä opastusta. |
| 6 | OHAR | On | |
| 7 | OHAR | En osaa sanoa | Tällaisen "ohjelman" käyttöön ei pitäisi tarvita mitään ohjeistusta. |
| 8 | OHAR | En osaa sanoa | - |
| 9 | SEM | Ei | |
| 10 | H SEM | En osaa sanoa | Yleisrakenne, eli mikä on se rakenne jonka tekijät ovat kuvitelleet. Nyt tuollainen rakenne tehtiin kurssille |

| | | | |
|---------|----------|---------------|--|
| | | | näppituntumalla kokeilemalla (mutta toimii silti). Mitähän kaikkia toimintoja tuolla oikein onkaan...? |
| 11 | SEM | On | |
| 12 | SEM | On | Eipä oikeastaan. Kurssin alussa tosiaan ryhmät hämmensivät, mutta selvisin tutkimalla Canvasia tarkemmin. |
| 13 | Molemmat | En osaa sanoa | Ryhmäilmoittautuminen oli sellaisessa paikassa mistä en älynnyt sitä hakea. |
| 14 H | OHAR | On | Ilmeisesti kalenterin kautta jotenkin tapahtumia luomalla olisi saanut ryhmäilmoittautumisen, nyt tuli käytettyä lopulta Doodlea, kun se vaati vähemmän ylläpitoa itseltä. |
| 15 | | Ei | Naurettavaan käyttöliittymään. |
| 16 | OHAR | Ei | - |
| 17 | SEM | | |
| 18 | SEM | Ei | |
| 19 | OHAR | En osaa sanoa | Lähinnä se, että mistä löytää harkkaryhmäilmoittautumiset. |
| 20 | OHAR | On | |
| 21 | Molemmat | Ei | |
| 22 | OHAR | On | |
| 22 | Molemmat | Ei | Normaalien tehtävien kuten kurssipalautusten ja muiden tekemiseksi voisi tehdä jonkin YouTube-tutorialin, josta tulisi ilmi Canvasin pääpiirteet. Tämä olisi pitänyt tehdä heti Canvasin julkaisun yhteydessä. |

| | | | |
|----|----------|---------------|--|
| 23 | Molemmat | En osaa sanoa | |
|----|----------|---------------|--|

| # | Has there been enough guidance on how to use Canvas? | Is there something you would have needed more guidance with? |
|----|--|--|
| 24 | Yes | |
| 25 | Yes | |
| 26 | No | If there was a place to see the notifications that I received in my email. |

| # | Millä kurssilla käytät Canvasa? | Onko Canvaksessa joitakin hyviä puolia? Mitä ne ovat? |
|-----|---------------------------------|--|
| 1 | SEM | Moodlea ei ole paljoa käyttänyt, mutta Canvaksen ensivaikutelma on parempi. |
| 2 H | OHAR | Modernimman näkönen, tehtävät ja niihin liittyvä palautepuoli toimii ihan mukavasti. |
| 3 | OHAR | En osaa sanoa IDLE:stä (käyttänyt tasan yhdessä kurssissa) mutta Moodleen verrattuna en ole vielä löytänyt mitään erikoisen hyvää |
| 4 | OHAR | No eipä juuri. Samanoloinen systeemi. Idleen verrattuna tietty melkein mikä vaan on parempi, idle nyt vaan on vanhenunut. Moodle2 on ihan jees. |
| 5 | OHAR | Verrattuna esimerkiksi idleen, canvas on turhan sekava ja epälooginen kokonaisuus. Itselle ei tule mieleen mitään erityisen hyvää, sillä lähinnä |

| | | |
|---------|----------|--|
| | | alusta vaikuttaa pahasti keskeneräiseltä työversioltä, joka kaipaa vielä paljon työtä ja muutamaa hyvää UX-suunnittelijaa. |
| 6 | OHAR | Mielekkäämpi ja modernimpi ulkoasu. |
| 7 | OHAR | Moodle on hyvä, idle menettelee vaikka ei siihenkään mielellään koske, mutta Canvukseen en koskisi pitkällä tikullakaan ellei ole pakko. |
| 8 | OHAR | - Se, että canvas osaa automaattisesti nostaa esille "tulevat, menneet ja ohitetut" tehtävät on aika jees. - ... kuten myös kommentti kenttä palautukselle. Jos nyt vielä tehtävänannotjen kirjoittajat muistuttaisivat ihmisiä että tuohon voi naputella vaikka kurssipalautetta tehtävänannosta. |
| 9 | SEM | Ei tarvitse päivittäin kytätä, tuleeko jotain uutta materiaalia. Palvelua voi käydä katsomassa silloin, kun sähköpostiin tulee ilmoitus uudesta sisällöstä. Kalenteriominaisuus on tosin plussaa. |
| 10 H | SEM | Viikkoharjoituksiin (vh) ilmoittautuminen onnistui People-toiminnolla tekemällä sille ryhmän. Ryhmiin jako onnistuu; opiskelijat voivat itse ilmoittautua tietyn kokoi- siin ht-ryhmiin ja vh-ryhmiin. Kalenteriin saa tehtyä ht-välinäyttöaikoja helposti yhdellä toiminnolla, esim. 11 h samana päivänä 30 min paloina. On tuo selkeämpi (yksinkertaisempi) kuin Moodle. Tiedostojen (dokumenttiversiot) palautus onnistuu opiskelijoilta hel- posti. |
| 11 | SEM | |
| 12 | SEM | Canvas on lähettänyt mukavasti sähköposti-ilmoituksia aina, kun kalen- teritapahtumat ovat muuttuneet. Oli myös hienoa, kun varatun sprint 1:n review-tapaamisen sai näkymään automaattisesti omassa Google-kalen- terissa kaikilla laitteilla. |
| 13 | Molemmat | |

| | | |
|---------|----------|--|
| 14 H | OHAR | Opiskelijoiden kanssa palautuksista kommunikointi on ihan sujuvaa. Pystyi tekeen ryhmäilmoittautumiset ja juttujen palauttamiset. Alkutus- kan jälkeen kuvienkin lisääminen ihan ok tehtävänantoihin. |
| 15 | | Ei. |
| 16 | OHAR | Canvasissa löytyy hyviäkin puolia. Jotkut asiat on hyvin jaoteltu ja pi- detty simppeleinä. lisäksi navigointi-menu helpottaa paljon asioimista. |
| 17 | SEM | |
| 18 | SEM | En ole huomannut mitään hyvää verrattuna Moodleen. |
| 19 | OHAR | Ainakin siitä näkyy lähestyvät dedikset ihan suht hyvin ja palautuskin näyttäisi toimivat. Tykkään silti idlestä ja moodlesta enemmän. |
| 20 | OHAR | Harjoituspisteiden seuraaminen on mukavan helppoa. Tehtävien palautus huomattavasti mukavampaa kuin esimerkiksi sähköpostilla. |
| 21 | Molemmat | |
| 22 | OHAR | |
| 22 | Molemmat | Käyttöliittymä yrittää joiltain osin olla selkeämpi esim. deadlinejen osalta, kuin Moodle tai IDLE. |
| 23 | Molemmat | |

| | | |
|---|---|-------------------|
| # | Millä kurs- silla käytät Canvasta? | Vapaa sana |
| 1 | SEM | |

| | | |
|----|--------|--|
| 2 | H OHAR | Sähköpostit ja muut yhteystiedot saataville, niin ei tarte käyttää Canvasta kommunikaatioon. ryhmäilmoittautumiset jne. jonnekin muualle ku people-osan alle. |
| 3 | OHAR | Summa summarum: käyttöliittymä selkeämmäksi ja/tai loogisemmaksi ettei tarvi palloilla kymmentä minuutia, että löytää haluamansa. |
| 4 | OHAR | |
| 5 | OHAR | Olisiko mitenkään mahdollista jättää näitten alustojen varhaisversioiden testaaminen suljettuihin testiryhmiin, eikä ottaa niitä testattavaksi varsinaisille kursseille jossa ne lähinnä sotkevat kaikkien aikataulua ja aiheuttavat ylimääräistä työtä? |
| 6 | OHAR | Ihan hyvä pohja järjestelmäksi mutta kaipaa vielä lisäominaisuuksia. Toivottavasti Canvasiin jatkokehitetään oikeita asioita tulevaisuudessa. |
| 7 | OHAR | Seuraavalla kerralla, kun tällaisia järjestelmiä tehdään, hoitakaa testaaminen jotain muuta kautta kuin tunkemalla ne pakolliseksi kursseille. |
| 8 | OHAR | |
| 9 | SEM | Tarvitseeko sitä nyt sataa eri palvelua käyttää? Se Moodle on ihan hyvä, eikä se ole edes ollut käytössä vielä kauaa. Ärsyttää, kun joutuu joka kurssilla kaivamaan asiat eri paikasta Mennään sillä Moodlella, se on parempi. |
| 10 | H SEM | No kyllä tuota Canvas:n käyttöä jatketaan ensi vuonnakin. |
| 11 | SEM | |
| 12 | SEM | Ryhmiin käyttöliittymä näyttää olevan suunnattu paremmin massiivisille vekkokursseille, joilla opiskelijat luovat itse omia vapaamuotoisia opiskeluryhmiä. Käyttöliittymä tuntui sopivan huonommin kursseille, jossa on valmiita viikkoharjoitusryhmiä ja jossa luodaan oma harjoitustyöryhmä. |

| | | |
|----|----------|--|
| 13 | Molemmat | |
| 14 | H OHAR | import-export-ominaisuudet jos toimivat hyvin Excel-Canvas-Excel, niin sen kanssa monet näistä jutuista varmaan hoituisivat kivasti. |
| 15 | | Kauhea. |
| 16 | OHAR | - |
| 17 | SEM | |
| 18 | SEM | Ehdoton ei uusille rinnakkaisille järjestelmille joita nykyiselläänkin on jo liikaa. |
| 19 | OHAR | Ihan hyvä alku, mutta kaipaa paljon parannusta käytettävyyden kannalta. |
| 20 | OHAR | Canvaksessa ei varsinaisesti ole vikaa. Suurin ongelma tulee siinä, että TTY:llä on käytössä noin 200 eri portaalia, joista eri kurssien asiat tulee kaivaa. Jos portaalien käyttöä saataisiin yhdistettyä, olisi tämänkin käyttö paljon mukavampaa. Nykyisellään tämä on ehkä portaalien parhaasta päästä. |
| 21 | Molemmat | |
| 22 | OHAR | Ei ole järin käyttäjäystävällinen. |
| 22 | Molemmat | Olisi kiva, jos olisi yksi alusta, mihin tehtävät palautetaan. Moodlea käytetään laajasti muillakin kursseilla, IDLE on ollut ohjelmistotekniikassa erittäin laajalti käytössä. Canvas saa minulta kurssiarvosanan 3, koska sen näkymissä on selkeitä vaikeuksia ja epäjohtonmukaisuutta ja sen käyttöön ei ole olemassa kovin selkeää opastusta. Eri kurssien deadlinejen ja tehtävien palauttamisen näyttämisen pitäis olla helpompaa. |
| 23 | Molemmat | |

Englanninkielisessä versiossa ”Vapaa sana” –kysymystä vastaavaan kysymykseen ”Is there anything else on your mind you'd like to share about Canvas?” ei ollut vastannut kukaan.