



Oamk Journal

Oulun ammattikorkeakoulun julkaisu

Tämä on alkuperäisen julkaisun rinnakkaistallenne. Rinnakkaistallenne saattaa erota alkuperäisestä sivutukseltaan ja painoasultaan.

This is an electronic reprint of the original publication. This version may differ from the original in pagination and typographic detail.

Käytä viittauksessa alkuperäistä lähdettä/Please cite the original version:

Ojala, P. 2022. Ohjelmiston jatkuvaa käyttöönottoa pilotoitiin veikkauskisasovelluksella. Oamk Journal 159/2022. <http://urn.fi/urn:nbn:fi-fe2022091959511>

Ohjelmiston jatkuvaa käyttöönottoa pilotoitiin veikkauskisasovelluksella

19.10.2022 - Ojala Pekka

Ohjelmistokehityksen automatisoituun julkaisuun tähtäävä prosessi koostuu ohjelmiston jatkuvasta integroinnista, jatkuvasta jakelusta ja jatkuvasta käyttöönotosta. Automaatio tehostaa tuotekehitystä ja parantaa ohjelmistoa kehittävän yrityksen kilpailukykyä markkinoilla. Jatkovaa käyttöönottoa pilotoitiin kesällä 2022 naisten jalkapallon EM-turnausta varten tehdyllä veikkauskisasovelluksella. Sovellus julkaistiin Netlify-palvelussa, joka automatisoi käyttöönoton tehden uusien ohjelmistoversioiden julkaisusta nopeaa ja suoraviivaista.

Eräs ketterän ohjelmistokehityksen periaatteista on ”julkaise varhain ja julkaise usein”. Ohjelmointi aloitetaan hyvin varhaisessa vaiheessa tuotekehitystä, ja ohjelmakoodi muuttuu nopeasti, jopa useita kertoja saman päivän aikana. Koska koodi muuttuu koko ajan ja koska ohjelmoijia on useimmiten enemmän kuin yksi, tarvitaan versionhallintaratkaisu koodimuutosten hallintaan.

Paikkaa, jossa koodi versionhallintajärjestelmässä on, kutsutaan repositorioksi. Sieltä koodiin tehdyt muutokset pitää vielä saada julkaisuun joko testaus- tai tuotantoympäristöön.

Jatkuva integrointi

Jatkuva integrointi (Continuous Integration, CI) tarkoittaa niitä käytänteitä ja toimintatapoja, joilla ohjelmiston koodimuutosten integrointi repositorioon on automatisoitu työkaluihin tukeutuen. Automatisoinnin ohella CI-työkalujen tehtävä on varmistaa uuden tai muuttuneen koodin oikeellisuus. Prosessi perustuu testaukseen ja lähdekoodien yhdistämiseen (kuvio 1). [1]

Jatkuva integrointi (Continuous Integration, CI)

Muutos ohjelmakoodissa
käynnistää automaattisesti
käännöksen

Käännös (Build)

Jos käännös menee läpi,
suoritetaan automatisoidut
testit

Testaus (Test)

Jos testit menevät läpi, eri
kehittäjien koodit yhdistetään

Yhdistäminen
(Merge)

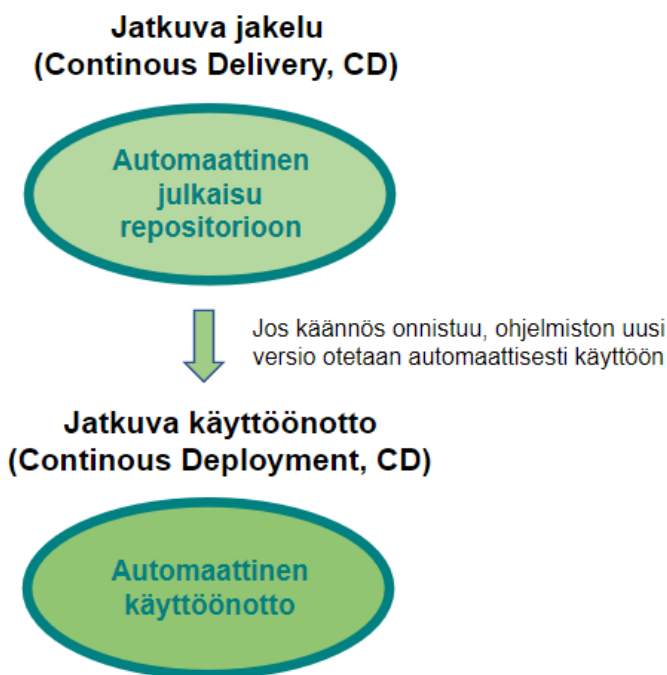
KUVIO 1. Jatkuva integrointi.

Jos ohjelmistokehityksessä ei käytetä jatkuvaa integrointia, kehittäjien on kommunikoiduttava ja koordinoitava keskenään versionhallintaan viettäviä koodimuutoksista. Tämä on ongelmallista, koska manuaalinen koordinaatio vaikeutuu koodimäärän ja kehittäjämäärän kasvaessa. [1]

Jatkuva jakelu ja jatkuva käyttöönotto

Versionhallinta repositorioiden on ohjelmiston kehittäjiä varten. Ohjelmiston varsinaiset käyttäjät eivät useinkaan ole tietoisia versionhallintaratkaisista. Voi jopa olla, että he eivät tiedä versionhallinnasta mitään, eikä heidän tarvitsekaan tietää. Ohjelmiston käyttäjien intressit ovat testaus- tai tuotantoympäristössä olevassa varsinaisessa ohjelmistossa. Ilman automaatiota ohjelmiston kehittäjien on erikseen huolehdittava siitä, että repositoriin kohdistuneet koodimuutokset päätyvät testaus- tai tuotantoympäristöön, kuten esimerkiksi web-palvelimelle, mikäli kyseessä on selaimella käytettävä web-sovellus.

Jatkuva jakelu (Continuous Delivery, CD) tarkoittaa sitä, että repositorioon kohdistuvien koodimuutosten perusteella ohjelmistosta käännetään automaattisesti uusi versio, joka onnistuneen käynnöksen seurauksena voidaan viedä joko testaus- tai tuotantoympäristöön. Kun koodiin kohdistuneet muutokset päätyvät testaus- tai tuotantoympäristöön, ne näkyvät myös ohjelmiston varsinaisille käyttäjille versiopäivitysten muodossa. Jatkuva käyttöönotto (Continuous Deployment, CD) vie automaation astetta pidemmälle, sillä onnistuneen käynnöksen seurauksena koodimuutokset viedään testaus- tai tuotantoympäristöön automaattisesti (kuvio 2). [2]

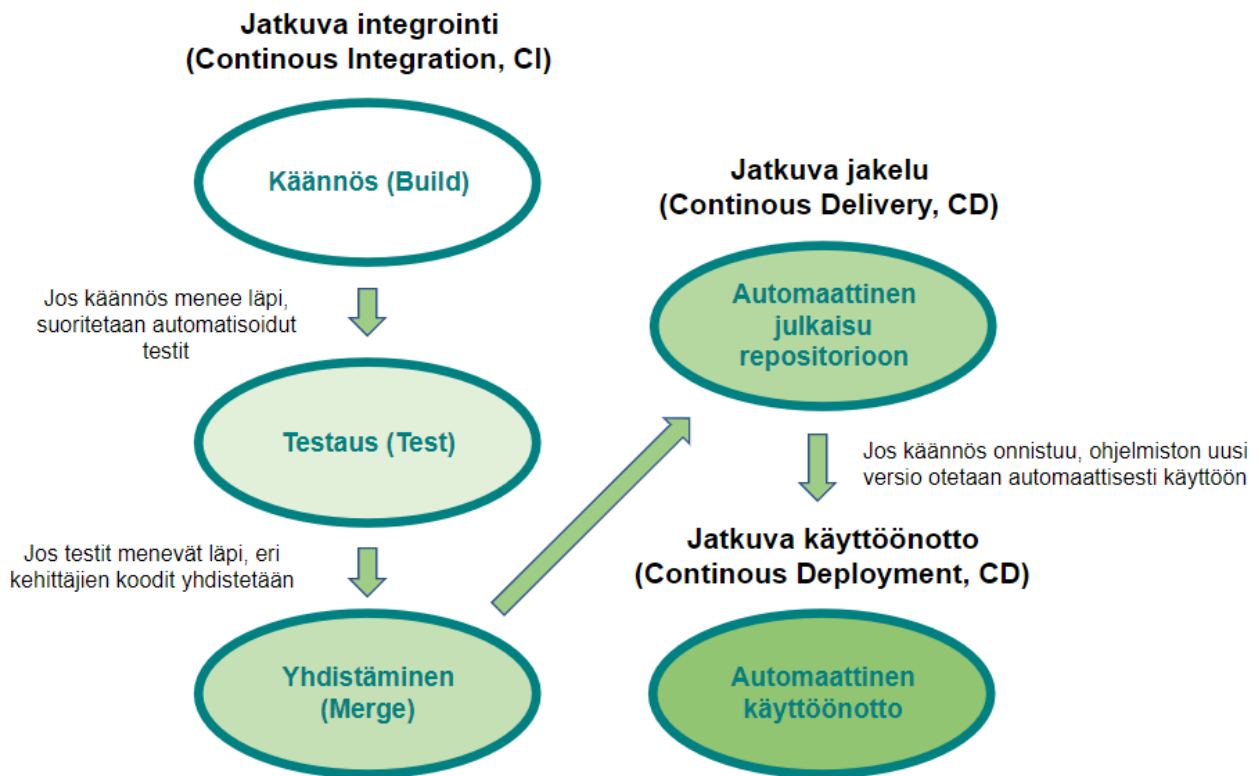


KUVIO 2. Jatkuva jakelu ja jatkuva käyttöönotto.

CI/CD-putki

CI/CD-putki (pipeline) tarkoittaa automatisoitua prosessia, joka sisältää jatkuvan integroinnin, jatkuvan jakelun ja jatkuvan käyttöönoton. Putki määrittelee mitä missäkin vaiheessa tapahtuu. Määrittely tehdään CI/CD-työkalulla. CI/CD-putken (kuvio 3) tyypillisiä vaiheita ovat lähdekoodin muutos repositoriossa, käynnös, testaus ja käyttöönotto. Kun repositorioon kohdistuu lähdekoodin muutos, CI/CD-työkalu käynnistää automaattisesti putkeen määritellyt toiminnot. Lähdekoodin muutosta seuraa tyypillisesti käynnös. Mikäli se menee läpi, suoritetaan

automatisoidut testit. Jos testit menevät läpi, ohjelmistosta otetaan käyttöön uusi versio. [3]



KUVIO 3. CI/CD-putki.

Toimiva CI/CD-putki tehostaa tuotekehitystä ja mahdollistaa ohjelmistoa kehittäville yritykselle paremman kilpailuaseman markkinoilla. Myös ohjelmiston ylläpidettävyys paranee ja tuen järjestäminen helpottuu. Hyvin rakennettu CI/CD-putki on automatisoitu, nopea, tietoturvallinen ja johdonmukainen. Se on lisäksi tiukasti integroitu versionhallintaan ja ongelmatilanteissa se kykenee osoittamaan ongelmakohdan putkessa. Tällaisen putken rakentamisen ja ylläpidon edellytys on ammattitaitoinen kehittäjätiimi. [4]

Jalkapallon EM-turnauksen veikkaussovellus

Naisten jalkapallon EM-turnaus järjestettiin Englannissa kesällä 2022. Turnausta varten kehitettiin sovellus veikkauskisan toteuttamiseksi. Jalkapallon arvoturnauksen veikkauskisa on aiemmin järjestetty kaksi kertaa käyttämällä Googlen työkaluja. Nyt haluttiin kehittää sovellus itse helpottamaan tulosten ja tulostaulun ylläpitoa. Toinen motivaattori oli ohjelmointi. Toisinaan on mukava

ohjelmoida vapaa-ajalla jotain sellaista, jolla ei ole suoraa kytkentää opettajan työhön.

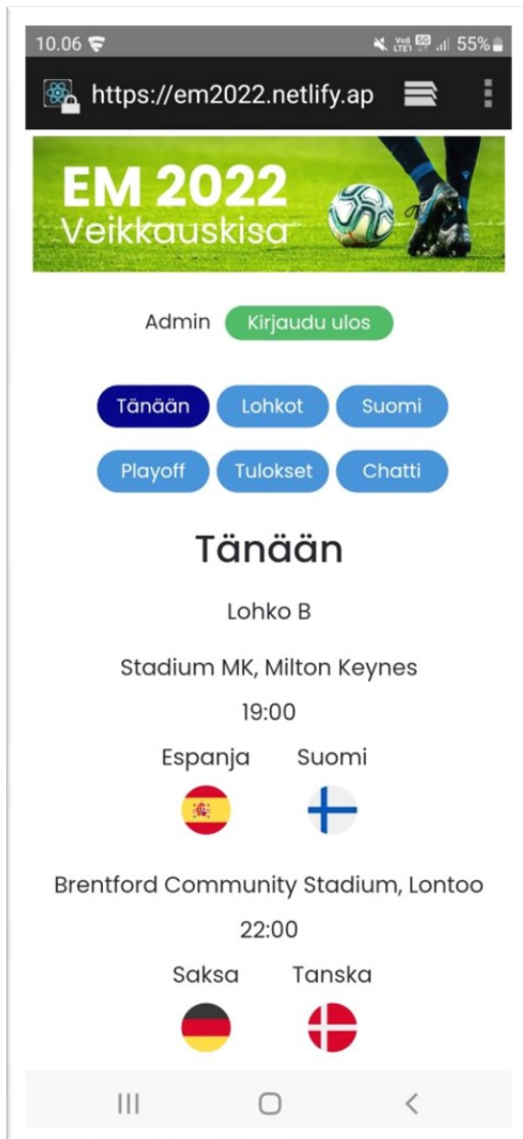
Veikkauskisasta informoitiin etukäteen tietojenkäsittelyn ja liiketalouden henkilöstöä, jotta osallistujamäärä oli selvillä ennen sovelluksen kehittämisen aloittamista. Kehittäminen päätettiin aloittaa, jos kisaan ilmoittautuu vähintään 5 osallistujaa. Lopulta 10 henkilöä ilmoittautui mukaan.

Sovelluksen kehittäminen käynnistyi ennen vapaajaksoa, mutta suurin osa kehittämisestä tapahtui vapaajakson kahden ensimmäisen viikon aikana. Sovelluksen julkaisusta ilmoitettiin osallistujille juhannuksen jälkeisenä maanantaina 27.6. Turnaus alkoi keskiviikkona 6.7., eli osallistujille jäi reilu viikko aikaa veikkauksien tekoon. Kaikki veikkaukset piti tehdä ennen turnauksen avauspeliä (kuva 1).



KUVA 1. Suomi-huumaa Milton Keynesissä (kuva: Sanna Rönkkö).

Sovellus koostuu rekisteröitymis- ja kirjautumissivuista, etusivusta, alkulohkoveikkauksesta, Suomi-veikkauksesta, Playoff-veikkauksesta, tulostaulusta ja chatista. Sovellus on kehitetty Reactilla mobiiliystävälliseksi Bootstrapiin perustuen. Sovelluksen tietokanta on Firebase Realtime Database. Kuvassa 2 on sovelluksen kirjautumisen jälkeinen etusivu, joka näyttää kuluvan päivän ottelut.



KUVA 2. Sovelluksen etusivu.

Alkulohkoveikkauksessa veikattiin jokaisen alkulohko-ottelun tulos 1X2-menetelmällä. Alkulohkoja turnauksessa oli neljä ja jokaisessa alkulohkossa oli neljä maata. Alkulohkojen otteluita turnauksessa oli yhteensä 24. Kuvassa 3 on alkulohkoveikkauksen näkymän alkuosa ennen turnauksen alkamista ja alkulohkovaiheen päättymisen jälkeen.

Tänään Lohkot Suomi

Playoff Tulokset Chatti

Alkulohkoveikkaus

Valitse pelin voittaja tai tasapeli. Saat pisteen jokaisesta oikeasta tuloksesta.

Tee kaikki tämän sivun veikkaukset ennen turnauksen ensimmäistä peliä. Siihen saakka voit muuttaa aiemmin tekemiäsi valintoja.

Pelattu: 0/24 Pisteet: 0/24

Lohko A

06.07.2022 klo 22:00

Englanti Tasapeli Itävalta

Tänään Lohkot Suomi

Playoff Tulokset Chatti

Alkulohkoveikkaus

Turnaus on alkanut. Et voi enää muuttaa veikkaustasi.

Pelattu: 24/24 Pisteet: 16/24

Lohko A

06.07.2022 klo 22:00

Englanti Tasapeli Itävalta

Oikein, 1 piste

KUVA 3. Alkulohkoveikkauksen näkymän alkuosa ennen turnauksen alkamista ja alkulohkovaiheen päättymisen jälkeen.

Suomi-veikkauksessa veikattiin Suomen alkulohko-otteluiden maalimäärät, Suomen sijoitus alkulohkossa sekä Suomen menestyminen turnauksessa. Kuva 4 on Suomi-veikkauksen näkymän alkuosa ennen turnauksen alkamista ja Suomen alkulohko-otteluiden päättymisen jälkeen.

Tänään Lohkot Suomi

Playoff Tulokset Chatti

Suomi-veikkaus

Erilaisia Helmarit-veikkauksia. **Tee ne kaikki ennen turnauksen ensimmäistä peliä.** Siihen saakka voit muuttaa aiemmin tekemiäsi valintoja.

MAALILUKUVEIKKAUS

Veikkaa joukkueiden tekemien maalien määrä pelissä. Saat 2 pistettä jokaisesta oikeaan osuneesta maalimäärästä.

Pelattu: 0/3 Pisteet: 0/12

08.07.2022 klo 19:00

Espanja - Suomi

	0	1	2	3	4	5+
	0	1	2	3	4	5+

Tänään Lohkot Suomi

Playoff Tulokset Chatti

Suomi-veikkaus


Turnaus on alkanut. Et voi enää muuttaa veikkaustasi.

MAALILUKUVEIKKAUS

Pelattu: 3/3 Pisteet: 4/12

08.07.2022 klo 19:00

Espanja - Suomi

	0	1	2	3	4	5+
	0	1	2	3	4	5+

Väärin

KUVA 4. Suomi-veikkauksen näkymän alkuosa ennen turnauksen alkamista ja Suomen alkulohko-otteluiden päättymisen jälkeen.

Playoff-veikkauksessa veikattiin turnauksen eri vaiheista jatkoon pääsevät joukkueet siten, että veikkauksen ensimmäisessä vaiheessa alkulohkoista valittiin 8 puolivälieriin selviytyvää joukkuetta. Sen jälkeen veikkaus jatkui valitsemalla neljä välieriin selviytyvää joukkuetta. Kolmannessa vaiheessa valittiin kaksi finalistia ja Playoff-veikkauksen päätti mestarin valinta kahdesta finalistista. Kuvassa 5 näkyy Playoff-veikkauksen näkymän alkuosa ennen turnauksen alkamista ja Playoff-vaiheen päättymisen jälkeen.

Tänään Lohkot Suomi

Playoff Tulokset Chatti

Playoff-veikkaus

Tämän sivun veikkauksessa on **neljä vaihetta. Tee ne kaikki ennen turnauksen ensimmäistä peliä.** Siihen saakka voit muuttaa aiemmin tekemiäsi valintoja.

ALKULOHKOT (1/4)

Pisteet: 0/8

Lohko A

Englanti

Itävalta

Norja

Pohjois-Irl

Tänään Lohkot Suomi

Playoff Tulokset Chatti

Playoff-veikkaus

Turnaus on alkanut. Et voi enää muuttaa veikkaustasi.

ALKULOHKOT (1/4)

Pisteet: 6/8

Lohko A

Englanti

Itävalta

Norja

Pohjois-Irl

Osittain oikein, 1 piste

KUVA 5. Playoff-veikkauksen näkymän alkuosa ennen turnauksen alkamista ja Playoff-vaiheen päättymisen jälkeen.

Tulostaulu sisälsi osallistujat lajiteltuna kertyneen pistemäärän perusteella. Tulostaulussa osallistujasta näkyi pelkästään nimimerkki siihen saakka, kun kyseisen osallistujan mestariveikkaus meni pieleen.

Liiketalouden osaston naiset ottivat kisassa kaksoisvoiton. Kisan voitti ylivoimaisesti **Teija Harju**, joka osallistui veikkauskisaan nimimerkillä Hillahullu. Toisesta sijasta käytiin kova kamppailu ja lopulta sen vei nimiinsä **Sanna Rönkkö** nimimerkillä paminkello. Kuvassa 6 näkyy tulostaulun näkymän alkuosa (kaksi parasta) turnauksen päättymisen jälkeen.

Tulostaulu			
Henkilö nimimerkin takana ja hänen mestariveikkaus (lippu) tulevat näkyviin siinä vaiheessa kun mestariksi veikattu joukkue ei selviydy jatkoon, tai jos mestariveikkaus onnistui, viimeistään sitten kun turnaus on päättynyt.			
Nimimerkki	Nimi	Pts	Mestari
Hillahullu	Teija Harju	58	
paminkello	Sanna Rönkkö	40	

KUVA 6. Tulostaulu-näkymän alkuosa turnauksen päättymisen jälkeen.

Esimerkki jatkuvasta käyttöönotosta

Jotta edellä kuvatun kaltaisen sovelluksen voi julkaista verkossa, tarvitaan palvelu, jossa julkaisun voi tehdä. Julkaisuun oli aluksi kaksi vaihtoehtoa: Microsoft Azure ja Heroku. Ensin päädyttiin kokeilemaan Azurea, joka oli ennestään tuttu työn puolesta. Parin tunnin jälkeen tultiin siihen johtopäätökseen, että Azure on aivan liian järeä palvelu yksittäisen sovelluksen julkaisuun. Sen jälkeen tutustuttiin Herokun dokumentaatioon, joka ei innostanut.

Lopulta päädyttiin kartoittamaan muita palveluita ja löydettiin [Netlify-palvelu](#). Sen dokumentaation ja tutoriaalien perusteella muodostui käsitys helposta ja suoraviivaisesta käyttöönotosta. Palveluun rekisteröidyttiin ja aloitettiin sovelluksen ensimmäinen käyttöönotto. Palvelu tarjoaa useita eri tapoja sovelluksen käyttöönottoon.

Käyttöönotto tehtiin kuvassa 7 esitetyn prosessin mukaan tuomalla koodit palveluun Git-repositoriosta. Tämä edellyttää niin kutsutun kolmannen osapuolen oikeuksien antamista Netlifylle GitHubin repositorioon. Ensimmäisessä vaiheessa valitaan Git-versionhallinnan toimittaja, kuten esimerkiksi GitHub-palvelu. Sen jälkeen GitHub-palvelusta valitaan repositorio, jonka koodit halutaan ottaa käyttöön. Lopuksi palvelussa määritellään sivuston ja sen kääntämisen asetukset, sekä tehdään ensimmäinen käyttöönotto.

Import an existing project from a Git repository

From zero to hero, three easy steps to get your site on Netlify.

1. Connect to Git provider

2. Pick a repository

3. Site settings, and deploy!

KUVA 7. Sovelluksen käyttöönoton vaiheet Netlify-palvelussa, kun käyttöönotto tehdään Git-repositoriosta.

Käyttöönotto oli kokonaisuutena helppoa. Ainoa pieni ongelma alussa oli se, että sovelluksen kehittämisessä käytetty paikallisen ympäristön Node.js-versio oli eri kuin palvelun oletusversio. Netlifyn foorumilta löytyi kuitenkin nopeasti ratkaisu ongelmaan, jonka jälkeen palvelu määriteltiin käyttämään sitä Node.js-versiota, jota sovelluksen kehittämisessä oli käytetty.

Jatkuva käyttöönotto Netlifyn palvelulla toimi hyvin. Kun GitHubin repositorioon vietiin koodimuutoksia ensimmäisen käynnöksen jälkeen, palvelu tunnisti koodimuutokset automaattisesti ja käynnisti välittömästi käynnöksen, jonka edistymistä pystyttiin seuraamaan Netlifyn palvelussa.

Käynnökset tapahtuivat nopeasti alle minuutissa, pois lukien ensimmäinen käynnös. Onnistuneen käynnöksen jälkeen sovelluksen uusi versio oli saman tien käyttäjien saatavilla eli tässä tapauksessa tuotantoympäristössä. Kuvissa 8 ja 9 näkyy viimeisimmän käynnöslokin alkuosa ja loppuosa Netlify-palvelussa.

Deploy log

```
1      8:41:53 AM: Build ready to start
2      8:41:55 AM: build-image version: 0e2f4c52031ab562db66aec633308326e3b108d0 (focal)
3      8:41:55 AM: build-image tag: focal
4      8:41:55 AM: buildbot version: 0d1cf41e4e345a224c39ab2a783aa4af5c76f6e8
5      8:41:55 AM: Fetching cached dependencies
6      8:41:55 AM: Starting to download cache of 184.1MB
7      8:41:57 AM: Finished downloading cache in 1.692721384s
8      8:41:57 AM: Starting to extract cache
9      8:42:03 AM: Finished extracting cache in 5.617610828s
10     8:42:03 AM: Finished fetching cache in 7.39144205s
11     8:42:03 AM: Starting to prepare the repo for build
12     8:42:03 AM: Preparing Git Reference refs/heads/master
13     8:42:04 AM: Parsing package.json dependencies
14     8:42:05 AM: Starting build script
15     8:42:05 AM: Installing dependencies
```

KUVA 8. Sovelluksen viimeisimmän käännöslokien alkuosa Netlify-palvelussa.

Deploy log

```
98 8:42:35 AM: Creating deploy upload records
99 8:42:35 AM: 6 new files to upload
100 8:42:35 AM: 0 new functions to upload
101 8:42:36 AM: Site deploy was successfully initiated
102 8:42:36 AM:
103 8:42:36 AM: (Deploy site completed in 659ms)
104 8:42:36 AM:
105 8:42:36 AM: _____
106 8:42:36 AM: Netlify Build Complete
107 8:42:36 AM: _____
108 8:42:36 AM:
109 8:42:36 AM: (Netlify Build completed in 19.2s)
110 8:42:36 AM: Starting post processing
111 8:42:36 AM: Post processing - HTML
112 8:42:36 AM: Caching artifacts
113 8:42:36 AM: Started saving node modules
114 8:42:36 AM: Finished saving node modules
115 8:42:36 AM: Started saving build plugins
116 8:42:36 AM: Finished saving build plugins
117 8:42:36 AM: Started saving pip cache
118 8:42:36 AM: Finished saving pip cache
119 8:42:36 AM: Started saving emacs cask dependencies
120 8:42:36 AM: Finished saving emacs cask dependencies
121 8:42:36 AM: Started saving maven dependencies
122 8:42:36 AM: Finished saving maven dependencies
123 8:42:36 AM: Started saving boot dependencies
124 8:42:36 AM: Finished saving boot dependencies
125 8:42:36 AM: Started saving rust rustup cache
126 8:42:36 AM: Finished saving rust rustup cache
127 8:42:36 AM: Started saving go dependencies
128 8:42:36 AM: Finished saving go dependencies
129 8:42:36 AM: Build script success
130 8:42:36 AM: Post processing - header rules
131 8:42:36 AM: Post processing - redirect rules
132 8:42:36 AM: Post processing done
133 8:42:39 AM: Site is live ✨
134 8:42:40 AM: Uploading Cache of size 154.7MB
135 8:42:41 AM: Finished processing build request in 45.741087939s
```

KUVA 9. Sovelluksen viimeisimmän käännöslokin loppuosa Netlify-palvelussa.

Positiivinen automaatiokokemus

Paikallinen kehitysympäristö yhdessä versionhallinnan kanssa on kätevä, mutta rajoitteinen ratkaisu. Jotta sovelluksen voisi julkaista verkossa, tarvitaan palvelu. Veikkauskisasovelluksen julkaisuun onnistuttiin löytämään palvelu, jolla sovelluksen jatkuvan käyttöönoton pystyi toteuttamaan vaivattomasti ja nopeasti.

Ohjelmistokehityksessä automatisoinnilla on tärkeä rooli. Veikkauskisasovelluksen jatkuvan käyttöönoton pilotti tarjosi positiivisen automaatiokokemuksen, jota on mahdollista hyödyntää myös opettajan työssä monin tavoin. Esimerkkeinä opetus ja hankkeet, joilla voi olla tarve hankkeissa kehitettävien sovelluksien julkaisuille. Myös sovelluksen kehittämisessä onnistuttiin, sillä veikkauskisa saatiin vietyä läpi ongelmitta.

Sovelluksen kehittäminen täysin opetuksen ulkopuolella toi hauskaa vaihtelua opettajan työhön. Yhteisöllisyyden näkökulmasta veikkauskisa tarjosi osallistujille mahdollisuuden olla tekemisissä kollegoiden kanssa työn ulkopuolella.

Pekka Ojala

lehtori

Oulun ammattikorkeakoulu, Informaatiotekniikan yksikkö, tietojenkäsittely

Lähteet

[1] Rehkopf, M. 2022. What is continuous integration? Atlassian. Hakupäivä 2.9.2022. <https://www.atlassian.com/continuous-delivery/continuous-integration>

[2] Pittet, S. 2022. Continuous integration vs. delivery vs. deployment. Atlassian. Hakupäivä 2.9.2022. <https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>

[3] Red Hat. 2022. What is a CI/CD pipeline? Hakupäivä 2.9.2022. https://www.redhat.com/en/topics/devops/what-is-ci-cd?sc_cid=7013a000002pwNIAAI&gclid=CjwKCAjwu5yYBhAjEiwAKXk_eKoPaWSnJw

[Ff4b1YPDqnMpjK_qCh28uqDI5CRpW15H9vphD1c7gBsBoCtzcQAvD_BwE&gclid=a
w.ds](https://www.techtarget.com/searchsoftwarequality/CI-CD-pipelines-explained-Everything-you-need-to-know)

[4] Bigelow, S. J. 2022. CI/CD pipelines explained: Everything you need to know. Tech Accelerator. Hakupäivä 2.9.2022.

<https://www.techtarget.com/searchsoftwarequality/CI-CD-pipelines-explained-Everything-you-need-to-know>

METATIEDOT

Tyyppi: Artikkel

Julkaisija: Oulun ammattikorkeakoulu

Julkaisunumero: 159/2022

Julkaisuvuosi: 2022

Tekijätiedot: Ojala Pekka

Oikeudet: CC BY-SA 4.0

Kieli: suomi

Pysyvä osoite: <http://urn.fi/urn:nbn:fi-fe2022091959511>

Tiivistelmä: Jatkuva integrointi, jatkuva jakelu ja jatkuva käyttöönotto muodostavat ohjelmistokehityksen CI/CD-putken (Continuous Integration, Continuous Delivery, Continuous Deployment), joka automatisoi ohjelmiston uusien versioiden käyttöönoton. Jatkuvaa käyttöönottoa pilotoitiin kesän 2022 aikana tukeutumalla Netlify-palveluun. Esimerkkisovelluksena käytettiin naisten jalkapallon EM-turnausta varten kehitettyä veikkauskisasovellusta. Veikkauskisaan osallistui 10 henkilöä tietojenkäsittelyn ja liiketalouden koulutusohjelmista. Netlify-palvelu osoittautui toimivaksi ratkaisuksi, sillä uusien versioiden käyttöönotto palvelun avulla oli suoraviivaista ja nopeaa. Palvelu tunnisti GitHub-palvelun repositorioon viedyt koodimuutokset käynnistäen automaattisesti käännöksen, joka tapahtui nopeasti. Onnistuneen käännöksen jälkeen uusi versio näkyi heti sovelluksen käyttäjille.