



Drupal moduulikehitystä tukevat työkalut

Juho Rissanen

Haaga-Helia ammattikorkeakoulu

Tietojenkäsittely

Opinnäytetyö

2023

Tiivistelmä

Tekijä(t) Juho Rissanen
Tutkinto Tradenomi
Raportin/Opinnäytetyön nimi Drupal moduulikehitystä tukevat työkalut
Sivu- ja liitesivumäärä 30 + 2
<p>Opinnäytetyössä tutkittiin Drupal kehittäjien tämän päivän käytetyimpiä työkaluja osana Drupalin moduulikehitystä. Tutkimus rajattiin neljään eri työkalukategoriaan, joista pyrittiin selvittämään käytetyimmät työkalut. Tutkimuksessa tutkittavat työkalut rajoittuvat yhteisömoduuleihin, komentorivityökaluihin, kehitysympäristöihin sekä integroituihin kehitysympäristöihin. Näiden työkalujen lisäksi pyrittiin tunnistamaan aktiivisen Drupal yhteisön merkitys osana kehittäjien työkalupakkia.</p> <p>Tutkimuksessa käytettiin kvantitatiivista ja kvalitatiivista tutkimusmenetelmää. Kvantitatiivisella tutkimusmenetelmällä pyrittiin saamaan Drupal kehittäjien suosimista työkaluista numeerista dataa, joka kuvasi työkalujen käyttöastetta eri kategorioissa. Kvalitatiivisella tutkimuksella taas selvitettiin Drupal yhteisön merkitystä kehittäjän työkaluna. Tutkimuksen tavoitteena oli tarjota näkemystä siihen, mitä työkaluja Drupal yhteisö suosii tällä hetkellä Drupal kehityksen parissa, sekä miten Drupalin aktiivinen yhteisö voi toimia yhtenä kehittäjän työkaluna. Tutkimusta hyödyntävä lukija voi tehdä tulosten pohjalta päätöksiä omaan kehitystyöhönsä ja sen tukemiseen eri työkaluilla. Tämän lisäksi lukija pystyy tulosten avulla perehtymään yleisimpiin työkaluihin tarkemmin, ja täten laajentamaan tietämystään siitä, että miten Drupalin parissa voidaan työskennellä tehokkaammin.</p> <p>Tutkimuksessa hyödynnettiin verkkokyselyä tulosten keräämisessä. Kysely luotiin Google Forms työkalulla ja sen vastausvaihtoehdot johdettiin Drupalin oman dokumentaation suosituksista. Tämän lisäksi kyselyyn oli mahdollista lisätä omia työkalujaan avoimien vastausvaihtoehtokenttien kautta, jotta saadaan kerättyä mahdollisesti uusia ei niin tunnettuja työkaluja. Kysely jaettiin Drupal yhteisön yleisellä Slack kanavalla. Kyselyyn saatiin vastauksia 16 kappaletta. Huomioitavaa on, että tutkimuksessa käsiteltävät työkalut eivät ole pelkästään suunnattu Drupal ympäristön piiriin, vaan tutkimuksessa esiintyy myös työkaluja, jotka ovat käytössä myös muussa ohjelmistokehityksessä yleisesti.</p> <p>Tietoperustassa esitellään lukijalle Drupalin vahvuuksia ja haasteita sisällönhallintajärjestelmänä. Tämän jälkeen pureudutaan tiedon liikkumiseen Drupalin sisällä, minkä jälkeen käsitellään moduulikerroksen merkitys ja moduulityypit. Viimeiseksi tietoperustassa käsitellään Drupalin oman dokumentaation suosittelemia työkaluja tarkemmin, joista tutkimuksen kysely johdettiin. Tämä antaa lukijalle kontekstin siitä, että mihin Drupal järjestelmän osioon tutkittavia työkaluja käytetään, sekä niiden perustoiminnallisuuksista.</p> <p>Tutkimusten tuloksista voidaan todeta Drupal yhteisön suosimat työkalut tutkimuksessa rajatuista kategorioista. Yhteisömoduuleista selvästi suosituin oli Devel moduuli. Komentorivityökaluista suosituimmat oli Git, Composer ja Drush. Integroiduista kehitysympäristöistä suosituin oli PHPStorm, ja VSCode oli toiseksi suosituin. Kehitysympäristöissä Docker pohjaisuus nousi selvään suosioon. Drupal yhteisön merkityksestä nousi esiin kolme keskeistä teemaa: tuki ja avoimuus, tekninen apu, sekä sosiaalinen verkosto. Tulosten pohjalta lukija voi jatkoperehtyä suosituimpiin työkaluihin, ja täten halutessaan soveltaa näitä työkaluja omassa kehitystyössään.</p>
Asiasanat Drupal, moduuli, työkalut

Sisällys

1	Johdanto	1
1.1	Tutkimuksen tavoite ja menetelmät	1
1.2	Rajaukset	2
1.3	Käsitteet	2
2	Drupal sisällönhallintajärjestelmä	4
2.1	Drupalin vahvuudet ja haasteet	4
2.2	Tiedon liikkuminen Drupalissa	6
2.3	Drupal moduulijärjestelmä	7
2.3.1	Drupal ydinmoduulit	7
2.3.2	Drupal yhteisömoduulit	8
2.3.3	Drupal mukautetut moduulit	9
3	Drupal dokumentaation suosittelemat työkalut	12
3.1	Kehitystyötä tukevat moduulit	12
3.2	Komentorivityökalut	15
3.3	Integroidut kehitysympäristöt	16
3.4	Kehitysympäristöt ja konttiratkaisut	16
4	Tutkimuksen toteutus	18
5	Tulokset	20
5.1	Kehitystyötä tukevat moduulit	20
5.2	Komentorivityökalut	21
5.3	Integroidut kehitysympäristöt	22
5.4	Kehitysympäristöt ja konttiratkaisut	23
5.5	Drupal yhteisön merkitys kehitystyössä	23
6	Pohdinta	25
6.1	Yhteenveto ja johtopäätökset	25
6.2	Luotettavuus ja eettisyys	26
6.3	Jatkotutkimusehdotukset	26
6.4	Oman opinnäytetyön ja oppimisen arviointi	27
	Lähteet	28
	Liitteet	31
	Liite 1. Kyselyn runko	31

1 Johdanto

Sisällönhallintajärjestelmät ovat olennainen osa nykyaikaista web-kehitystä. Ne mahdollistavat tehokkaan ja hallitun nettisivujen tuotannon. Yksi tunnetuimmista sisällönhallintajärjestelmistä on Drupal, joka on arvostettu sen joustavuuden ja laajennettavuuden ansiosta. Drupal-kehityksen tehokkuus ja joustavuus nojautuvat pitkälti sen moduulijärjestelmään, jonka avulla sivuston toiminnallisuutta voidaan laajentaa ja räätälöidä. Drupal kehittäjien työkalupakissa nämä moduulit ovat välttämättömiä, mutta yhtä kriittisiä kehitystyön näkökulmasta ovat myös erilaiset työkalut, jotka tehostavat ja helpottavat moduulikehitystyötä.

Moduulikehitystä tukevat työkalut säästävät aikaa, sillä niillä pystytään nopeuttamaan toistuvien toimintojen tekemistä. Tämän lisäksi oikeilla työkaluilla pystytään vähentämään virheiden määrää, esimerkiksi ohjelmointilogiikan ongelmanratkaisussa tai automaattisen tiedostorakenteen ja koodin generoinnilla. Työkalut mahdollistavat myös monimutkaisten toimintojen luomisen tehokkaammin sillä työkalut tarjoavat arvokasta tietoa koodin suoritukseen liittyvistä yksityiskohdista. Tämän lisäksi työkaluilla pystytään luomaan puitteet tehokkaaseen kehitystyöhön, sillä oikeilla työkaluilla saadaan luotua yhtenäiset puitteet ohjelmistokehityksen eri vaiheisiin.

Tässä opinnäytetyössä pureudutaan niihin työkaluihin, jotka ovat keskeisiä Drupalin moduulikehityksessä. Valitsin tämän tutkimusaiheen henkilökohtaisen työkokemukseni perusteella. Työskenteleminen web-kehitysalalla, jossa olen käyttänyt Drupalia aktiivisesti. Työssäni olen huomannut, kuinka oikeat työkalut ovat tehneet kehitystyöstä tehokkaampaa ja virheettömämpää. Aihe on kiinnostava, koska työkaluja on monia ja oikeiden työkalujen valitseminen on kriittinen osa kehitystyötä. Tutkimuksen merkitys korostuu tilanteissa, joissa punnitaan mahdollisia työkaluja omaan projektiin. Tietämys Drupal yhteisön suosimista työkaluista takaa työkalujen soveltuvuuden Drupal ympäristöön, sekä mahdollisen tuen kehitysvaiheissa.

1.1 Tutkimuksen tavoite ja menetelmät

Opinnäytetyössäni keskityn tutkimaan ja esittelemään Drupalin moduulikehitystä tukevia työkaluja, sekä niiden suosiota. Työn tavoitteena on tarjota perusymmärrys siitä, että minkälaisia työkaluja on tarjolla Drupal-kehityksessä ja mitkä ovat alan ammattilaisten suosituimmat työkalut tänä päivänä. Työssäni tutkitaan kehitystyötä tukevia työkaluja sekä otetaan selvää aktiivisen yhteisön merkityksestä kehitystyössä.

Tutkimuksessa käytetään pääsääntöisesti kvantitatiivista tutkimusmenetelmää, jolla pyritään keräämään numeerista tietoa eri työkalujen käyttäjämääristä eri kategorioista. Näin saadaan ymmärrys siitä, mitä työkaluja Drupal yhteisö suosii tänä päivänä omissa kehitystehtävissään. Tämän lisäksi

käytetään kvalitatiivista tutkimusmenetelmää, jonka tarkoituksena on kerätä tietoa Drupal yhteisön merkityksestä kehittäjille. Kvalitatiivisella tutkimuksella saadaan kerättyä eri teemoja Drupal yhteisön merkityksestä, ja näin analysoitua tuloksia.

Tutkimuksessa hyödynnettiin kyselyä. Kysely luotiin Google Forms työkalulla ja sen vastausvaihtoehdot johdettiin Drupalin oman dokumentaation suosituksista. Tämän lisäksi kyselyyn oli mahdollista lisätä omia työkalujaan avoimien vastausvaihtoehtokenttien kautta, jotta tutkimuksen aikana pystytään myös mahdollisesti tunnistamaan uusia suosittuja työkaluja. Tutkimus jaettiin Drupal yhteisön yleisellä Slack kanavalla ja kyselyyn saatiin vastauksia 16 kappaletta.

1.2 Rajaukset

Opinnäytetyöni keskittyy tutkimaan nykyaikaisia Drupal-moduulikehitystä tukevia työkaluja sekä ammattilaisten suosituksia näistä työkaluista. Työssä käsitellään yleisesti ohjelmistokehitysalalla käytettäviä työkaluja, sekä puhtaasti Drupal ympäristöön tarkoitettuja työkaluja. Drupalin omassa dokumentaatioissa määritelty työkalukategorijako auttoi työn rajauksessa, josta poimin keskeisimmät työkalukategoriat kehitystyötä ajatellen.

Tutkittavat työkalut rajoittuvat yhteisömoduuleihin, komentorivityökaluihin, kehitysympäristöihin ja integroituihin kehitysympäristöihin. Tutkittavien työkalujen lisäksi pyritään kartoittamaan Drupal-yhteisön merkitystä osana kehittäjän työkalupakkia. Työ ei tarjoa teknistä koulutusta näiden työkalujen käyttöön, mutta pyrkii antamaan yleispätevää tietoa yleisimpien työkalujen toiminnoista ja merkityksestä Drupal-kehityksessä.

1.3 Käsitteet

Avoimen lähdekoodin järjestelmä (Open source)

Järjestelmä, jonka lähdekoodia kaikki pystyvät katsoa, muokata tai edistää

Hook

Drupalin tarjoama tapa moduuleille muuttaa ja laajentaa ytimen toimintoja.

Moduuli

Laajennus tai lisäosa Drupalille, joka tarjoaa lisätoimintoja tai muokkaa olemassa olevia toimintoja.

Node

Drupalissa yksittäinen sisältöyksikkö, esim. artikkeli tai sivu.

Riippuvuus (Dependency)

Kun yksi moduuli vaatii toisen moduulin toimiakseen.

Ydin (Core)

Drupalin perusjärjestelmä, joka sisältää perusominaisuudet ja toiminnot. Ytimen ulkopuoliset toiminnot toteutetaan pääsääntöisesti moduuleilla.

2 Drupal sisällönhallintajärjestelmä

Tässä luvussa käydään läpi Drupalin keskeisiä piirteitä ja rakenteita. Aluksi käydään läpi Drupalin yleiset vahvuudet sekä sen haasteet. Tämän jälkeen käsitellään tiedon liikkumista Drupalin sisällä. Lopuksi esitellään Drupalin moduulijärjestelmä keskittyen sen kolmeen eri moduuliin kategoriaan. Tämän luvun tarkoitus on auttaa lukijaa ymmärtämään Drupalin peruspiirteitä, sekä luomaan lukijalle pohjaa siitä, että mihin osioihin eri työkaluja voidaan hyödyntää.

2.1 Drupalin vahvuudet ja haasteet

Drupal on yksi tunnetuimpia sisällönhallintajärjestelmiä. Se on ilmainen avoimen lähdekoodin järjestelmä, joka mahdollistaa tehokkaan ja joustavan nettisivukehityksen. Drupalin – kuten muidenkin sisällönhallintajärjestelmien – päävahvuus on tehokas sisällön luominen ja hallinta. Avoimen lähdekoodin ansiosta Drupal järjestelmän voi ottaa käyttöön, jakaa ja muokata omiin tarpeisiin ilman erityisiä rajoitteita. (DXPR Marketing Team, 2022.)

Yksi Drupalin päävahvuuksista sisällönhallintajärjestelmänä on sen vahva yhteisö. Avoimen lähdekoodin malli rohkaisee jatkuvaa kehitystä ja innovointia (Picchini, 2020). Drupal-yhteisöä voidaankin pitää eräänlaisena kehittäjän työkaluna. Drupalin yhteisö on aktiivinen ja intohimoinen järjestelmän kehitystoimintaa kohtaan, ja yhteisön noin 1,4 miljoonan käyttäjän vahvuus takaa alustan jatkuvan kehityksen (Picchini, 2020). Aktiivinen yhteisö edistää koodikatselmointia, testausta ja dokumentaation kehittämistä, mikä mahdollistaa Drupalin jatkuvan kehityksen ja säännöllisten versio-päivitysten tuottamisen (Picchini, 2020). Yhteisön jäsenet auttavat toisiaan tiedonhaussa ja ongelmatilanteissa, ja aktiivinen ongelmatilanteiden ja bugien raportointi hyödyttää koko yhteisöä, edistämällä nopeampaa kehitystyötä ja yhtenäisten toimintatapojen määrittelyä.

Drupalilla on teknisiä vahvuuksia sisällönhallintajärjestelmänä. Sen uusimmat versiot ovat keskittyneet käyttäjäkokemuksen parantamiseen. Drupal tarjoaa tehokkaan sisällöntuottokokemuksen ilman erillistä ohjelmointitarvetta ja sen mukautettavat teematyökalut tekevät responsiivisen nettisivukehityksen helpommaksi. (Picchini, 2020.) Vaikka sisällöntuotto Drupalissa on jo itsessään sujuvaa, on merkittävää, että sisältöä voidaan myös generoida automaattisesti. Tämä tehostaa kehitystyötä, jossa keskitytään moduuleihin, jotka vaikuttavat sisällön hallintaan ja muokkaamiseen. Luvussa 3 esitellään tarkemmin työkaluja, jotka tukevat tätä automaatiota.

Drupal tunnetaan myös yhtenä turvallisimmista sisällönhallintajärjestelmistä. Vaikka se sisältää jo itsessään kattavat turvallisuusominaisuudet, yhteisön tuottamilla moduuleilla Drupal-sivustoja voidaan tehdä entistä turvallisemmiksi. Useimmat näistä moduuleista liittyvät muun muassa sisäänkirjautumiseen, kuten kaksivaiheiseen tunnistautumiseen, Google-tunnistautumiseen,

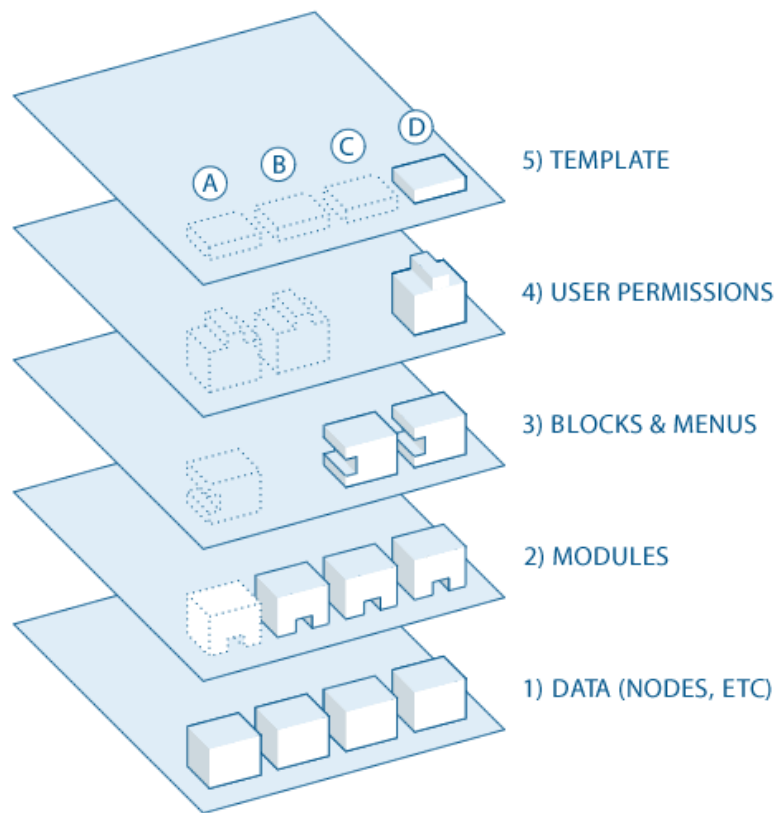
turvallisuuskatselmointiin pääkäyttäjäpaneelissa ja Captchan integrointiin. Valitut moduulit riippuvat sivuston erityistarpeista, ja Drupal tarjoaa laajan valikoiman turvallisuusmoduuleja sivuston suo-
jauksen tehostamiseen. (Zia, 2018.)

Vaikka Drupal on yksi käytetyimmistä sisällönhallintajärjestelmistä, on sillä myös selviä haasteita. Drupal koetaan usein järjestelmäksi, joka on haastava oppia nopeasti sen monimutkaisen raken-
teen takia. Vaikka Drupal yhteisö on pyrkinyt kehittämään oppimismateriaaleja, niin uusien kehittä-
jien perehdyttäminen Drupaliin vie usein paljon aikaa. (MCD, 2019.)

Drupalin aktiivisen yhteisön ansiosta syntyvät versiopäivitykset tuovat myös haasteita. Drupal-yh-
teisössä on valittu olla tukematta vanhentuneita toimintatapoja vanhempien versioiden yhteensopi-
vuuden kannalta. Uusien versiopäivitysten yhteensovittaminen omaan projektiin voi viedä usein
paljon aikaa, sillä kaikki Drupal sivustoja laajentavat moduulit tulee myös päivittää yhteensopiviksi
uuden Drupal version kanssa. (MCD, 2019.)

Drupal koetaan myös haastavaksi järjestelmäksi suurien verkkosivujen yhteensovittamisessa. Kun
verkkosivusto kasvaa, niin usein ladattavien moduulien määrä kasvaa sivustolla. Tämä saattaa
usein aiheuttaa yhteensopivuusongelmia, jotka puolestaan saattavat johtaa sivuston hitaaseen
suorituskykyyn tai heikkoon käyttäjäkokemukseen. (MCD, 2019.)

2.2 Tiedon liikkuminen Drupalissa



Kuva 1 Tiedon liikkuminen Drupalissa (Drupal.org, 2023f)

Kuva 1 mallintaa tiedon liikkumista Drupalissa. Ennen kuin sivustolla voidaan esittää mitään, täytyy olla dataa, jota esittää. Tästä esimerkkinä toimivat nodet, eli yksittäinen sisältö ilmentymä tietyistä sisältötyypistä. Tämä voi olla esimerkiksi yksittäinen blogikirjoitus tai artikkeli (Drupal.org, 2023f). Seuraavassa kerroksessa ovat moduulit. Drupalin moduulijärjestelmää käsitellään yksityiskohtaisemmin seuraavassa luvussa, mutta lyhyesti sanottuna moduulit ovat sivuston toiminnallisia lisäosia.

Seuraavassa kerroksessa sijaitsee lohkot ja valikot. Lohkot esittävät pääsääntöisesti moduuleista tulevaa tietoa, mutta niitä voidaan mukauttaa esittämään käytännössä mitä tahansa tietoa. Lohkoja voidaan sijoittaa sivustolla eri paikkoihin sekä niitä pystytään helposti konfiguroimaan eri tarpeisiin. Lohkojen yleisimpiin konfigurointeihin kuuluu muun muassa se, millä sivuilla lohko esitetään, tai mille käyttäjille lohkoa esitetään. Valikot toimivat Drupalissa navigaattoreina, joiden tehtävänä on tarjota linkit kaikkiin Drupalissa luotuihin sivuihin. (Drupal.org, 2023f.)

Seuraava kerros on käyttäjäoikeudet. Käyttäjäoikeudet määritellään sen mukaan, mitä eri käyttäjät saavat tehdä ja nähdä kyseisellä Drupal sivulla. Oikeudet määritellään ensiksi rooleille, minkä jälkeen itse käyttäjät liitetään tiettyyn rooliin (Drupal.org, 2023f). Drupalissa sivuston normaalia käyttäjää, eli kirjautumatonta käyttäjää kutsutaan anonyymiksi käyttäjäksi. Anonyymille käyttäjälle pystytään myös määrittelemään, että mitä sisältöä kirjautumattomat käyttäjät näkevät sivulla tai mitä toimintoja he pystyvät suorittamaan (Drupal.org, 2023f). Kun Drupal-ympäristössä tehdään moduulien kehitystyötä, käyttäjäroolien huomioiminen on kriittistä. On tärkeää, että kehittäjillä on työkaluja tarjolla, jotka mahdollistavat toiminnallisuuksien testaamisen eri käyttäjien näkökulmasta helposti. Testausvaiheessa on erityisen tärkeää varmistua siitä, että sivusto toimii odotetusti eri käyttäjärooleille. Tällöin oikeanlaiset kehitystyökalut ovat korvaamattomia.

Tiedonkulun ylimmässä kerroksessa sijaitsee teema. Teema koostuu pääsääntöisesti HTML:stä ja CSS:stä, joihin on sisällytetty Twig-muuttujia. Tämä takaa sen, että Drupalin tuottama sisältö saadaan asettumaan toivotulla tavalla verkkosivulle. Tämän lisäksi teema sisältää erinäisiä funktioita, joita voidaan halutessa käyttää erinäisten moduulien tarjoamien funktioiden ylikirjoittamiseen. Näin saadaan täysi kontrolli siitä, missä muodossa moduulit generoivat sisällön, sen luontihetkellä. (Drupal.org, 2023f.) Teeman ja moduulien kehityksen aikana kehittäjien on tärkeää saada käyttöönsä työkaluja, jotka tehostavat kehitysprosessia. Kun otetaan huomioon Drupalin monimutkaiset moduulirakenteet, on välttämätöntä, että kehittäjät pystyvät testaamaan muutoksiaan reaaliajassa ja varmistamaan niiden yhteensopivuuden järjestelmän muiden osien kanssa. Teemaan liittyvät työkalut ovat erityisen arvokkaita moduulikehityksen yhteydessä, sillä niiden avulla voidaan varmistaa, että kehitetyt moduulit ovat yhteensopivia toistensa kanssa.

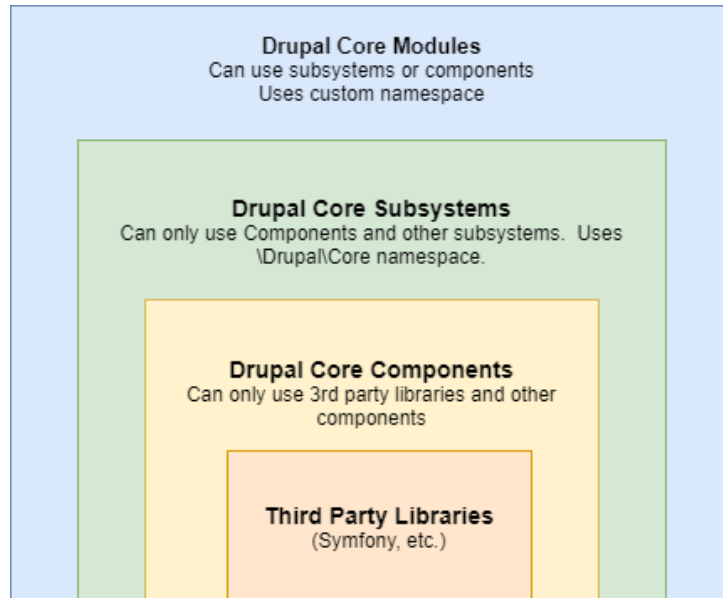
2.3 Drupal moduulijärjestelmä

Drupal moduulijärjestelmän tarkoitus on laajentaa sivuston toiminnallisuuksia. Drupal moduuli on yksinkertaisuudessaan kokoelma PHP tiedostoja, mutta toiminnallisuuden mukaan, se saattaa myös sisältää JavaScript tai tyylitiedostoja, kuten CSS-tiedostoja (Drupalize.me, 2023a). Drupal moduulit jakautuvat kolmeen eri tyyppiin. Moduulityyppejä ovat Drupal ydinmoduulit (core modules), Drupal yhteisömoduulit (contributed modules), sekä Drupal mukautetut moduulit (custom modules), eli ne moduulit, jotka sivustoa kehittävä taho kehittää itse sivuston eri tarpeisiin (Drupal.org, 2019h). Alla olevat alaluvut kertovat miten eri moduulityypit eroavat toisistaan, ja mikä niiden tehtävä on Drupal sivulla.

2.3.1 Drupal ydinmoduulit

Drupal ytimellä tarkoitetaan sitä kokonaisuutta, jonka Drupal lataus tarjoaa käyttäjälleen. Drupalin ydin pitää sisällään järjestelmän perusominaisuudet. Yksi ytimen kerroksista on ydinmoduulit,

joiden tarkoitus on laajentaa Drupal-sivuston perustoiminnallisuutta. Ydinmoduulit tarjoavat siis sivuston perustoiminnot, joita ovat muun muassa sivuston käyttäjien hallinta, sisällön tuottamiseen ja hallintaan liittyvät toiminnot, navigointi ja hakutoiminnot sekä erinäiset valmiit sivustoelementit, kuten listaukset ja lohkot, joita voidaan asettaa sivuston eri osioihin. (Drupalize.me, 2023a.) Kuva 2 havainnollistaa Drupal-ytimen rakennetta sekä moduulien sijaintia Drupalin ytimessä.



Kuva 2 Drupal ytimen rakenne (Drupal.org, 2022g)

2.3.2 Drupal yhteisömoduulit

Drupalin yhteisömoduulit käsite kattaa kaikki ne moduulit, jotka ovat ladattavissa Drupalin omalta sivustolta, ja jotka eivät ole osana Drupalin ydinmoduuleita (Anderson, 2019). Drupalin yhteisömoduulit ovat Drupalin aktiivisen yhteisön tuotos. Koska Drupal on avoimen lähdekoodin järjestelmä ja moduulien lisäämisen kynnys on matala, Drupalissa on saatavilla tuhansia yhteisön tuottamia moduuleita. Nämä moduulit on suunniteltu lisäämään toivottuja ominaisuuksia tai korjaamaan tiedossa olevia ongelmia.

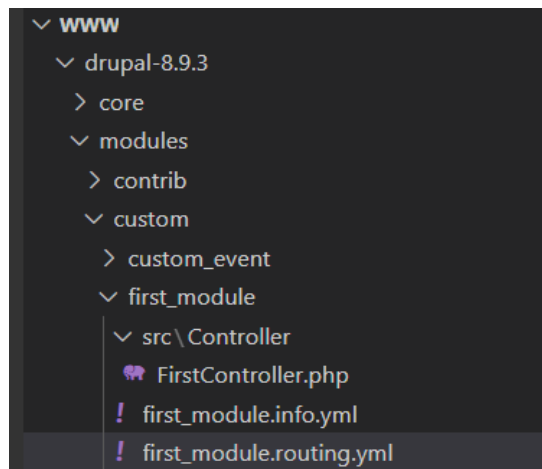
Kun valitaan yhteisömoduulia projektiin, on tärkeää tarkastella moduulin ylläpitoon liittyviä tekijöitä. Drupal.org-sivusto, jossa moduulit ovat listattu, tarjoaa laajasti tietoa kustakin moduulista, mukaan lukien viimeisimmät päivitykset, lähdekoodin ja tiedossa olevat ongelmat (issues) valitussa moduulissa. Kaikki nämä tiedot löytyvät moduulin kuvaussivulta. (Byron ym., 2008.) Vaikka yhteisömoduulit tuovatkin usein näkyviä lisäominaisuuksia sivustoille, ne tarjoavat myös arvokkaita työkaluja Drupal-kehittäjille tehostaen kehitystyötä.

2.3.3 Drupal mukautetut moduulit

Viimeinen Drupal moduulityyppi on mukautetut moduulit. Drupal projekteissa saattaa ilmetä tilanne, jossa ydin- tai yhteisömoduulien tarjoamat toiminnallisuudet eivät riitä sivuston tarpeisiin. Tällöin Drupal-moduulijärjestelmän joustavuus tarjoaa mahdollisuuden lisätä omaa mukautettua koodia projektiin. Tämä mukautettu koodi kasataan moduuliksi, ja se voidaan ottaa helposti käyttöön sivustolle eri työkalujen avulla. (Farhaz, 2021.)

Mukautettujen moduulien ei tarvitse aina luoda täysin uutta toiminnallisuutta, vaan ne pystyvät myös muokkaamaan tai laajentamaan jo olemassa olevia toiminnallisuuksia. Kun muokataan olemassa olevia moduuleita, niin Drupalin koukut (hooks) ovat oleellinen työkalu tämän saavuttamiseksi. Koukut mahdollistavat eri moduulien välisen yhteistyön, ja ne toimivat ikään kuin tapahtumakuuntelijoina. Esimerkiksi omassa mukautetussa moduulissa voidaan hyödyntää tiedostojen poistoon liittyviä koukkuja ja lisätä toiminnallisuuksia tämän ympärille. Tässä tapauksessa, kun sivustolla poistetaan tiedosto, ja mukautettu moduuli on käytössä sivustolla, niin mukautetun moduulin koukku aktivoituu ja oma mukautettu koodi suoritetaan. Tämä joustavuus mahdollistaa monimutkaisten ja mukautettujen toiminnallisuuksien implementoinnin ilman, että itse Drupalin ydintoimintoja tarvitsee erikseen muokata. (Drupal.org, 2023a.)

Drupal ympäristössä mukautettujen moduulien alustus aloitetaan aina uuden moduulihakemiston luomisella kyseiseen Drupal-projektiin. Drupal järjestelmän hakemistopuussa on oma osuus mukautetuilla moduuleilla, joka seuraa aina rakennetta /drupal-root/modules/custom/mukautetun-moduulin-nimi (Jha, 2023). Alla olevassa kuvassa havainnollistetaan normaalia Drupal-projektin rakennetta. Kuvassa 3 voidaan nähdä kaksi eri mukautettua moduulia custom_event sekä first_module /custom hakemiston alla.



Kuva 3 Mukautetun moduulin rakenne (Sinha, 2021)

Mukautetuissa moduuleissa ainut pakollinen tiedosto on info.yml-tiedosto, sen toiminnallisuudesta riippumatta. Info-tiedostossa listataan tietoa moduulista, kuten moduulin nimi, tyyppi, moduulin kuvaus, moduulipaketin nimi (package), ytimen versionumero vaatimus sekä moduulin mahdollisia riippuvuuksia. (Jha, 2023.) Kuvassa 4 on havainnollistettu esimerkki siitä, miltä info.yml tiedoston sisältö voisi näyttää, sekä taulukossa 1 on kuvattu kenttien merkitys.

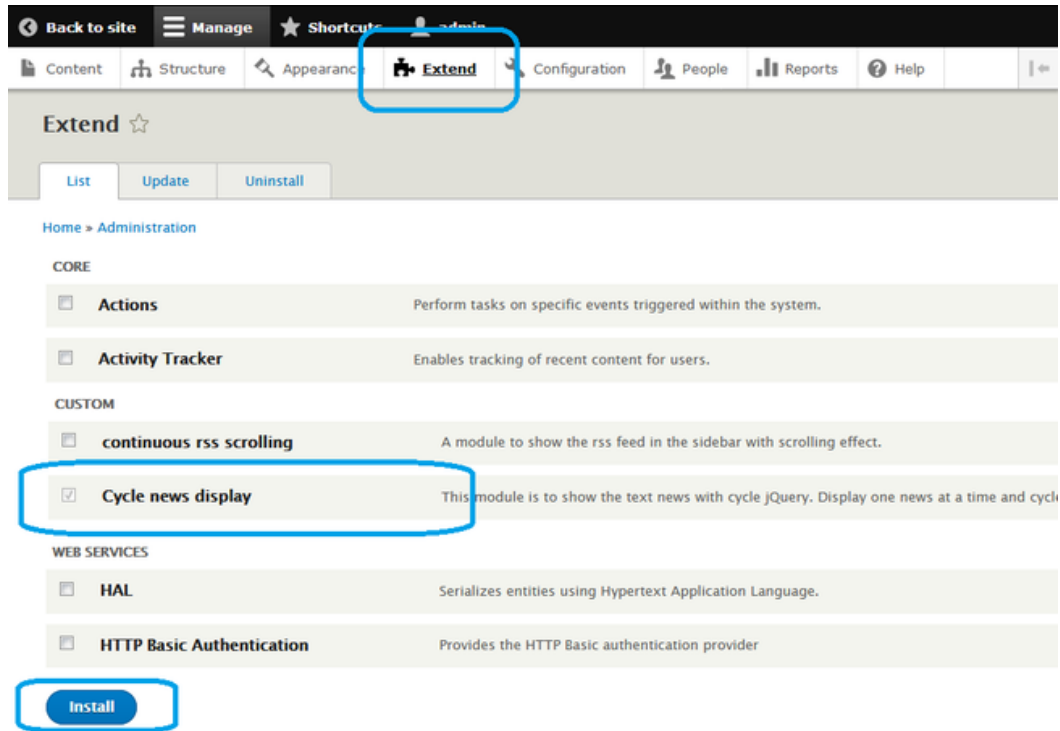
```
! info.yml
1 name: Oma Mukautettu Moduuli
2 type: module
3 description: Tämä teksti kuvaa mukautetun moduulin toimintoja
4 package: Kehitysmoduulit
5 core_version_requirement: ^9 || ^10
```

Kuva 4 Info-tiedoston esimerkkisisältö

Taulukko 1 Info-tiedoston kentät selitettynä (Jha, 2023)

Nimi (Name)	Moduulin ihmislukuisena ymmärrettävä nimi. Tämä nimi on käytössä moduulien yleiskatsastussivulla.
Kuvaus (Description)	Moduulin ihmislukuisena ymmärrettävä kuvaus moduulista. Tämä kuvaus on käytössä moduulien yleiskatsastussivulla.
Moduulipaketti (Package)	Ryhmä, johon moduuli kuuluu. Ryhmittelyä voidaan hyödyntää tilanteissa, joissa useita erinäisiä mukautettuja moduuleita halutaan niputtaa yhteen. Ryhmät ovat näkyvissä moduulien yleiskatsastussivulla.
Tyyppi (Type)	Kertoo minkä tyyppinen mukautettu elementti on, eli tässä tapauksessa mukautettu moduuli.
Ytimen versio vaatimus (Core version requirement)	Rajoittaa moduulin käyttöä tiettyyn Drupal versioon. Tässä tapauksessa moduulia voidaan käyttää vain, jos ollaan Drupal 9 tai Drupal 10 ympäristössä.

Info-tiedoston lisäämisen jälkeen moduuli listautuu Drupalin moduulijärjestelmään ja se voidaan tällöin myös asentaa käyttöön Drupal projektiin Drupalin pääkäyttäjän paneelissa, vaikka siinä ei olisi vielä implementoitu mitään toiminnallisuuksia (Drupal.org, 2023b).



Kuva 5 Moduulin käyttöönotto pääkäyttäjäpaneelissa (Drupal.org, 2021i)

Kuva 5 näyttää yhden tavan, miten moduuli otetaan käyttöön Drupal-ympäristössä. Drupal-projektiin kuuluvat moduulit voidaan ottaa käyttöön tai pois käytöstä Drupalin pääkäyttäjäpaneelin Laajennus-sivulla (extend), jossa kaikki projektiin asennetut moduulit ovat listattuna. Moduulin käyttöönoton jälkeen sen käyttöoikeuksia voidaan muokata käyttäjärooleittain sopiviksi, sekä sitä voidaan myös tarvittaessa konfiguroida omiin tarpeisiin sopivaksi. (Drupal.org, 2023j.) Mukautettujen Drupal moduulien alustaminen ja käyttöönotto on yleinen prosessi Drupal-ympäristössä. Drupal ympäristöön onkin tarjolla eri työkaluja, jotka nopeuttavat Drupal moduulikehityksen aloittamista, esimerkiksi koodin ja tiedostojen generoinnilla tai moduulien hallinnan nopeuttamisella.

3 Drupal dokumentaation suosittelemat työkalut

Drupal ympäristön kehitystyötä voidaan helpottaa erinäisillä työkaluilla, jotka jakautuvat useisiin kategorioihin. Tässä opinnäytetyössä keskityttiin kehitystyötä tukeviin moduuleihin, komentorivityökaluihin, kehitysympäristöihin ja integroituihin kehitysympäristöihin (IDE). Tämä luku tarjoaa perusteellisemman katsauksen näiden kategorioiden yleisimmistä työkaluista, jotka ovat listattuna Drupalin omissa dokumentaatioissa, toimien taustatietona ja kontekstina kyselytulosten käsittelyyn. Drupalin dokumentaatioissa listatut työkalut kuvastavat hyvin yhteisön suosimia yleisimpiä moduuleita, joten se toimii hyvin kontekstina kyselytulosten läpikäynnille.

3.1 Kehitystyötä tukevat moduulit

Drupal tarjoaa useita yhteisömoduuleita, joita pystytään hyödyntämään mukautettujen moduulien kehityksessä. Tässä alaluvussa käsitellään yleisimpiä yhteisömoduuleja, joilla pystytään laajentamaan Drupal kehitysympäristöä.

Devel on yksi suosituimmista Drupal-yhteisömoduuleista, joka on suunniteltu helpottamaan Drupal-kehitystä. Se onkin tällä hetkellä käytössä yli 175 000 sivustolla (Drupal.org, 2022c). Asennettaessa Devel-moduulia siihen kuuluvat alamuodulit, kuten Devel Generate ja Devel Node Access, tarjoavat kehittäjälle monipuolisia työkaluja. Yksi keskeisimmistä ominaisuuksista on generointimekanismi, jonka avulla voidaan esimerkiksi luoda testisisältöä eri sisältötyypeille tai generoida eri käyttäjiä eri rooleissa. Tämä nopeuttaa merkittävästi sivuston testaamista.

Lisäksi Devel moduuli tarjoaa koosteen sivukohtaisista tietokantakyselyistä, näyttäen niiden määrän ja suoritusajojen keston. Moduuliin sisältyy myös lohkoja mukautetun PHP-koodin lisäämiseksi sivuille ja mahdollisuuden testata sivuja eri käyttäjärooleilla, kuten pääkäyttäjän tai anonyymin käyttäjän näkökulmasta. Devel-moduuli tarjoaa myös välineitä Drupal-sivuston muistinkäytön seurantaan. (Salvisberg, 2018.) Kuva 6 havainnollistaa esimerkkiä Devel-moduulin generoimasta artikkelista.

Home

Search

Tools

[Add content](#)

Ad Eu laceo Molior

[View](#) [Edit](#) [Delete](#) [Revisions](#) [Devel](#)

Submitted by Anonymous (not verified) on Thu, 05/29/2020 - 11:48

Humo velit vereor. Humo letalis sagaciter scisco tamen. Facillisi lucidus nibh vulpes. Abigo duis enim jugis nutus saepius ulcisor. Aliquam amet causa nunc quae quia si valde vero voco. Abico augue exerci ille minim tation. Accumsan alliquip brevitatis conventio defui jugis sudo ut vero vulpes. Caecus gravis immitto in loquor mos nimis paulatim si vero.

Cui enim proprius sagaciter. Ad causa in tum valetudo vereor vulputate wisi. Aliquam antehabeo haero iaceo nostrud pagus similis uxor. Causa distineo iaceo luptatum quae sagaciter sed verto. Abbas at hendrerit illum melior neo quadrum tincidunt validus vulpes.

Haero nimis proprius refero. Jumentum lucidus meus populus premo sed sino torqueo utinam. Dignissim eu genitus lucidus ludus modo ulcisor. Abbas commoveo dolus duis elit olim si torqueo uxor ymo. Aliquam duis euismod immitto luptatum nunc quidne. Et gravis loquor mos. Aptent duis loquor pneum refoveo sagaciter suscipit. At caecus dolore populus ratis tum.

Mos pneum vereor. Conventio molior paratus paulatim suscipit typicus. Eu pertineo scisco suscipere vero. Appellatio decet esse lucidus nisl oppeto valde. Cogo erat iaceo in metuo sudo validus vulputate. Antehabeo dolus nutus ut vero. Bene diam gravis ille letalis paulatim probo saepius turpis venio.

Abbas antehabeo dignissim enim feugiat tego vereor voco. Aptent bene defui ibidem inhibeo mauris roto tego tincidunt. Brevitas jumentum oppeto ymo. At caecus capto damnum ea immitto sudo vulpes. Jus mauris ratis ut vulpes. Amet immitto iriure luctus melior nobis vulpes.

Kuva 6 Esimerkki Devel-moduulin generoimasta artikkelista (Amin, 2023)

Toinen tunnettu kehitystyötä tukeva yhteisömoduuli on WebProfiler. Tämän moduulin tehtävänä on tarjota kehittäjälle mahdollisuus seurata sivuston suorituskykyä. WebProfiler on aiemmin mainitun Devel moduulin alamoduulijärjestelmä, joten se vaatii Devel moduulin toimiakseen. WebProfiler asennuksen jälkeen kehittäjällä on mahdollisuus seurata taulukon 2 havainnollistamia vaihtoehtoja.

Taulukko 2 WebProfilerin tarjoamat toiminnot (Bhadra, 2018).

Toiminto	Toiminnon kuvaus
PHP konfiguraatio	Tarjoaa tietoa palvelimen PHP konfiguraatioista.
Kutsut	Tarjoaa tietoa sivupyynnöistä, evästeistä, pyyntö- ja vastausotsikoista sekä palvelinparametreista.
Aikajana	Grafiikka, joka tarjoaa tietoa sivupyynnölle kutsuista tapahtumista ja suoritusajoista.
Tietokanta	Tarjoaa tiedon kaikista tietokantakyselyistä, jotka on suoritettu kyseistä sivupyyntöä varten.
Käyttäjät	Tarjoaa tiedot aktiivisesta käyttäjästä rooleineen.
Näkymät	Tarjoaa tiedon kaikista näkymistä, jotka on renderöity kyseisellä sivulla, sekä niiden raken- nus-, suoritus- ja renderöintiajat.
Lohkot	Tarjoaa tiedon kaikista lohkoista, jotka on renderöity sivulle.

Lomakkeet	Tarjoaa tiedon kaikista lomakkeista, jotka on rakennettu kyseiselle sivulle.
Laajennukset	Tarjoaa tiedon kaikista sivuston aktiivisista moduuleista ja teemoista
Välimuisti	Tarjoaa tiedon sivun eri välimuistimekanismeista.

Uuden moduulin alustamisen helpottamiseksi on kehittäjille käytössä Module Builder -moduuli. Tämän moduulin tehtävänä on auttaa kehittäjää aloittamaan uuden moduulin kehitys Drupalissa. Sen avulla kehittäjät voivat luoda perusrakenteen mukautetuille moduuleilleen annettujen vaatimusten perusteella. Yksinkertaisesti sanottuna, Module Builder luo koodia automaattisesti. Kehittäjä voi määrittää monenlaisia vaatimuksia, kuten sisältötyypit, palvelut, lomakkeet ja tarvittavat hook-funktiot, joita hän tietää tarvitsevänsä kyseisessä moduulissa. Module Builder voi myös analysoida sivuston nykyisen koodin ja ehdottaa ominaisuuksia generointiin sen perusteella. Generoinnin jälkeen kehittäjälle tarjotaan valmis tiedostorakenne ja sisältö, joita hän voi käyttää oman moduulinsa kehittämisen pohjana. (Drupal.org, 2023d.) Kuvassa 7 esitetään esimerkki koodin generoinnista. Esimerkissä on näkyvillä moduulin .info tiedoston generoitu sisältö, sekä listaus generoiduista tiedostoista.

Generate code for *My Module* ☆

[Name](#)
[Hooks](#)
[Plugins](#)
[Entity types](#)
[Forms](#)
[Tests](#)
[Misc](#)
[Generate code](#)
[Devel](#)

[Home](#) » [Administration](#) » [Configuration](#) » [Development](#) » [Module entities](#) » [Edit My Module](#)

⚠ This module already exists at `modules/custom/my_module`. Some files may be possible to merge, but not all, and merges may overwrite parts of a file. It is recommended you use version control to prevent losing any existing code.

Filename	Merge status	Version control	Show
<input type="checkbox"/> <code>my_module.info.yml</code>	Merged	UNMANAGED	<input checked="" type="radio"/>
<input type="checkbox"/> <code>my_module.links.action.yml</code>	New	New	<input type="radio"/>
<input type="checkbox"/> <code>my_module.links.task.yml</code>	New	New	<input type="radio"/>
<input type="checkbox"/> <code>my_module.permissions.yml</code>	Merged	UNMANAGED	<input type="radio"/>
<input type="checkbox"/> <code>src/Entity/Handler/MyEntityTypeHandler.php</code>	New	New	<input type="radio"/>
<input type="checkbox"/> <code>src/Entity/MyEntityType.php</code>	New	New	<input type="radio"/>
<input type="checkbox"/> <code>src/Entity/MyEntityTypeInterface.php</code>	New	New	<input type="radio"/>
<input type="checkbox"/> <code>src/Form/MyEntityTypeForm.php</code>	New	New	<input type="radio"/>

my_module.info.yml code

```
name: 'My Module'
type: module
description: 'TODO: Description of module'
core_version_requirement: '^8 || ^9'
```

[Write selected files](#)

Kuva 7 Koodin generointi ja moduulin .info tiedoston generoitu sisältö (Drupal.org, 2023d)

3.2 Komentorivityökalut

Komentorivityökalut nopeuttavat ja helpottavat Drupal-kehitystä, sillä niillä pystytään tekemään toimintoja, jotka muuten saattaisivat vaatia useita eri manuaalisia työvaiheita, sekä niillä pystytään helpottamaan toistuvien toimintojen suorittamista. Tässä luvussa kerrotaan Drupal-moduulikehityksen yleisimpiä komentorivityökaluja.

Git on yksi yleisimmistä komentorivityökaluista Drupal-kehityksessä. Se on versionhallintajärjestelmä, josta on tullut ohjelmistokehityksen standardityökalu. Gitin avulla kehittäjät voivat tarkasti seurata muutoksia projekteissaan, aina yksittäisen tiedoston tasolle asti. Git mahdollistaa turvallisen muutosten kokeilun ja helpon muutosten perumisen sekä aiempien tiedostoversioiden palauttamisen. Sen haaroittamisominaisuus (branching) tekee mahdolliseksi työskennellä samanaikaisesti eri ohjelmiston osa-alueiden parissa, jolloin eri toiminnallisuuksia voidaan kehittää itsenäisinä kokonaisuuksina. (Atlassian.)

Toinen laajasti käytössä oleva komentorivityökalu Drupal-kehityksessä on Composer. Se on työkalu, joka helpottaa riippuvuuksien hallintaa PHP-projekteissa. Composerin avulla kehittäjät voivat määrittellä projektilleen tarvittavat kirjastot, ja Composer huolehtii niiden asennuksesta ja päivityksestä. Drupalin moduulikehityksessä Composer mahdollistaa myös kolmansien osapuolten kirjastojen määrittelyn ja sisällyttämisen, mikä on erityisen hyödyllistä mukautettuja moduuleja kehitettäessä. (Drupal.org, 2020e.)

Yksi yleisin puhtaasti Drupal ympäristöön liittyvä komentorivityökalu on Drush. Drush:n avulla pystytään nopeuttamaan ja automatisoimaan useita Drupalin käyttöliittymässä suoritettavia toimenpiteitä. Drush:n avulla pystytään esimerkiksi ottamaan käyttöön moduuleita yhdellä komennolla, joka normaalisti vaatisi useita vaiheita itse käyttöliittymässä (ks. 2.3.3). Drush integroituu myös useisiin muihin työkaluihin, jotka muun muassa mahdollistavat Drupal-välimuistin tyhjentämisen tai moduulien koodipohjan luonnin. Drush:n suosion takia, useat lisämoduulit tarjoavat myös omia Drush komentojaan, mikä laajentaa entisestään Drush:n käyttömahdollisuuksia. (Drupalize.Me, 2023b.)

Drupal Console on toinen komentorivityökalu, joka on suunniteltu erityisesti Drupal-ympäristöön. Se mahdollistaa pohjakoodin (boilerplate code) luomisen, ja se otettiin käyttöön alkaen Drupalin versiosta 8. Näin ollen se toimii vain Drupal 8:ssa ja sitä uudemmissa versioissa. Drupal Consolella pyritään tehostamaan ja nopeuttamaan yleisiä toimintoja Drupal-kehityksessä. Esimerkiksi oman räätälöidyn moduulin luominen on huomattavasti sujuvampaa Drupal Consolen avulla. Työkalu hyödyntää nykyaikaisia ohjelmointikäytäntöjä, mikä vähentää virheiden mahdollisuutta kehityksessä. Lisäksi Drupal Console tarjoaa interaktiivisen toimintatavan, jolloin kehittäjä voi määrittää generoitavan pohjakoodin eri vaatimuksia. (Drupal Console.)

3.3 Integroidut kehitysympäristöt

Integroidut kehitysympäristöt ovat sovelluksia, jotka tehostavat ohjelmointiprosessia. Integroidut kehitysympäristöt mahdollistavat koodin editoinnin, kääntämisen, testaamisen ja paketoinnin yhdessä sovelluksessa. Integroitujen kehitysympäristöjen päätoiminnot ja hyödyt ovat esitetty taulukossa 3.

Taulukko 3 Integroitujen kehitysympäristöjen päätoiminnot ja edut (AWS)

Toiminto	Toiminnon kuvaus
Koodin muokkauksen automatisointi	Integroidut kehitysympäristöt tunnistavat eri ohjelmointikielien säännöllisyydet ja osaavat soveltaa näitä sääntöjä koodin kirjoittamisen tai muokkaamisen automatisoinnissa.
Syntaksin korostus	Integroidut kehitysympäristöt tekevät koodista paremmin luettavaa. Ympäristöt korostavat koodin eri osioita eri väreillä, fonteilla tai painokertoimilla. Syntaksin korostus ominaisuus ehkäisee virheiden määriä.
Älykäs koodin viimeistely	Integroidut kehitysympäristöt tunnistavat aktiivisesti kehittäjän kirjoittamaa koodia, ja osaa antaa automaattisia koodinviimeistely ehdotuksia.
Lokaalin kehityksen automatisointi	Integroidut kehitysympäristöt nopeuttavat toistuvia lokaalin kehityksen toimenpiteitä.

Lokaalin kehityksen yleisimpiä toimintoja ovat koodin kääntäminen, testaaminen ja virheiden korjaaminen eli ”debugaus”. Koodin kääntämisellä tarkoitetaan sitä toimintoa, jossa järjestelmä kääntää koodin semmoiseen muotoon, joka on ymmärrettävää käyttäjärjestelmälle. Testaamisella tarkoitetaan niitä toimintoja, joilla kehittäjä voi testata omaa koodiaan, ennen kuin se viedään ulos kehittäjän omasta lokaalista ympäristöstä.

Debugauksella tarkoitetaan niitä toimintoja, joissa kehittäjä selvittää ohjelmiston ongelmatilanteita. Integroidut kehitysympäristöt tarjoavat tähän tehokkaat työkalut, sillä ohjelmoija voi suorittaa koodia rivi kerrallaan ja tarkastella, miten koodi käyttäytyy tai miten tieto liikkuu ja päivittyy ohjelman sisällä, kun sitä suoritetaan. Tämän lisäksi järjestelmät ilmoittavat kehittäjälle reaaliajassa mahdollisista ohjelmointivirheistä, kuten esimerkiksi ohjelmointilogiikkaan liittyvistä kirjoitusvirheistä. (AWS.)

3.4 Kehitysympäristöt ja konttiratkaisut

Kehitysympäristöjen päätehtävänä on auttaa kehittäjiä luomaan ja testaamaan sivuston toiminnallisuksia (Umbraco). Kehitysympäristöt jaetaan yleisesti seuraaviin tasoihin lokaali-, kehitys-,

esituotanto- ja tuotantoympäristöön. Nämä eri kehitysympäristötasot mahdollistavat uusien toiminnallisuuden testaamisen siten, että mahdolliset virheet saadaan todettua, ennen kuin sivuston tuotantoympäristöä päivitetään uusilla toiminnallisuuksilla. Testaaminen tietyssä ympäristöissä ei vaikuta muihin ympäristöihin, jolloin kehitysvaiheen ympäristöjen ongelmatilanteet eivät vaikuta tuotannossa olevaan sivustoon.

Drupal moduulikehityksen kannalta keskeisin kehitysympäristö on lokaali kehitysympäristö, jossa Drupal yhteisö suosii Docker pohjaisia konttiratkaisuja (Drupal.org, 2023k). Kontti on eristetty sovellusyksikkö, joka paketoii koodin, sen riippuvuudet sekä tiedostot yhdeksi eristetyksi kokonaisuudeksi (DiCesare, 2022). Koska konttiratkaisut mahdollistavat sovelluksen eristämisen omasta käyttöympäristöstä, sovellusta voidaan ajaa helposti käyttöympäristöstä riippumatta. Tämä helpottaa huomattavasti kehitystyötä, sillä kehittäjät saavat sovelluksen helposti käyttöön sen kehittämistä varten, ilman haastavia konfigurointeja tai asennuksia.

Drupal yhteisön suosima Docker pohjaisuus mahdollistaa eri työkalujen käytön, kun alustetaan tai käytetään konttipohjaisia ratkaisuja. Yksi yleisin työkalu on DDEV, jonka tehtävänä on helpottaa PHP-pohjaisten sovellusten, kuten Drupalin suorittamista Dockerin avulla. DDEV:n keskeisiä toimintoja ovat muun muassa helpotettu sovelluskehitysympäristön luominen, tietokantojen siirtäminen eri kehitysympäristöjen välillä ja komentojen ajaminen suoraan Docker-konttiin komentoriviltä. Lisäksi DDEV:n avulla päästään myös helposti kontin sisään suorittamaan tarvittavia toimenpiteitä komentoriviltä käsin. (DDEV GitHub repository, 2023.) Moduulikehityksen näkökulmasta helppo websovelluksen käynnistäminen sekä kehitysympäristön manipulointi ovat kriittisessä asemassa, sillä ne nopeuttavat kehitystyötä huomattavasti ja vähentävät virheiden määrää kehitysympäristön perustamisen ja suorittamisen näkökulmasta.

Toinen yleinen Docker pohjainen työkalu on Lando. Lando eroaa DDEV:stä siinä, että se tukee useampaa käyttötarkoitusta, kuin vain PHP-pohjaisia sovelluksia. Landon yksi vahvuuksista on sen valmiit reseptit. Drupal kehityksen näkökulmasta kehittäjät voivat vaivattomasti luoda Drupal kehitysympäristön konttipohjaiseksi Landon valmiilla resepteillä. Valmiit reseptit ovat suunniteltu siten, että se ottaa huomioon yleisimmät tarvittavat toiminnallisuudet tietyssä käyttötarkoituksessa. Drupalin näkökulmasta tämä tarkoittaa sitä, että valmiilla resepteillä saadaan konfiguroitua muun muassa webpalvelin ja tietokantaratkaisut. (Lando documentation, 2022.)

4 Tutkimuksen toteutus

Tutkimuksessa käytetään sekä kvantitatiivista että kvalitatiivista tutkimustapaa. Kvantitatiivinen tutkimus on menetelmä, joka perustuu määrällisen tiedon keräämiseen ja sen analysointiin. Kvantitatiivisen menetelmän etuna on sen kyky käsitellä suuria aineistomääriä ja tuottaa sen pohjalta numeerisia tuloksia. (Tähtinen, 2020.) Kvantitatiivisen tutkimuksen rajoituksena pidetään muun muassa sen heikkoutta käsitellä ilmiöitä syvällisesti (Tähtinen, 2020). Tätä voidaan kuitenkin helpottaa avoimilla kysymyksillä, mikäli se koetaan tarpeelliseksi. Kvantitatiivinen tutkimustapa soveltuu hyvin tutkimukseni osioon, jossa halutaan saada numeerista tietoa siitä, että mitä työkaluja Drupal yhteisö suosii tänä päivänä.

Kvalitatiivinen tutkimus puolestaan keskittyy laadullisen tiedon keräämiseen. Sillä pyritään ymmärtämään syvällisemmin kokemuksia ja merkityksiä. Kvalitatiivinen data voi koostua haastatteluista, avoimista kyselyvastauksista, havainnoista ja muista ei-numeerisista tiedoista. (Grand Canyon University, 2023.) Kvalitatiivinen menetelmä sopii tutkimukseni osioon, jossa pyritään ymmärtämään Drupal yhteisön merkitystä syvällisemmin.

Tässä opinnäytetyössä kvantitatiivinen ja kvalitatiivinen tutkimus toteutetaan sähköisenä kyselynä, jossa vastaajilta kerätään numeerista dataa siitä, mitä työkaluja he käyttävät aktiivisesti omissa Drupal-kehitystyötehtävissään. Kvalitatiivinen osuus on osa kyselyä avoimen kysymyksen muodossa. Avoimella kysymyksellä pyritään ymmärtämään Drupal-yhteisön merkitystä Drupal-kehittäjien keskuudessa.

Tutkimuksen kohderyhmänä toimii Drupal-kehittäjät. Kohderyhmä ei rajoitu yksittäiseen organisaatioon, vaan kohderyhmässä on Drupal-kehittäjiä ympäri maailmaa eri kokemuspohjilla. Kohderyhmän valinta perustuu siihen, että aktiivisilla Drupal-kehittäjillä on vahva kokemus Drupalin parissa työskentelemisestä, jolloin he voivat antaa arvokasta tietoa käyttämistään työkaluista ja käytännöistä. Tämän lisäksi kyselyn kohderyhmä pyrittiin saavuttamaan siten, että kysely ei rajoitu yhteen organisaatioon, jolloin organisaation omat toimintatavat eivät vääristä kyselyn tuloksia. Kohderyhmä saavutettiin Drupalin omalla Slack kanavalla, jonne sähköinen kysely lähetettiin tasaisin väliajoin.

Tutkimuksessa käytettiin sähköistä kyselylomaketta aineiston keräämiseen. Sähköinen kyselylomake mahdollistaa tehokkaan tavan saavuttaa laaja Drupal-kehittäjäyhteisö sekä mahdollistaa nopean datan keräämisen ja sen analysoinnin. Kysely oli auki 16.10.–20.10.2023, jona aikana vastaajia kertyi 16 kappaletta eri kokemustaustoilla ja työskentelyorganisaatioista.

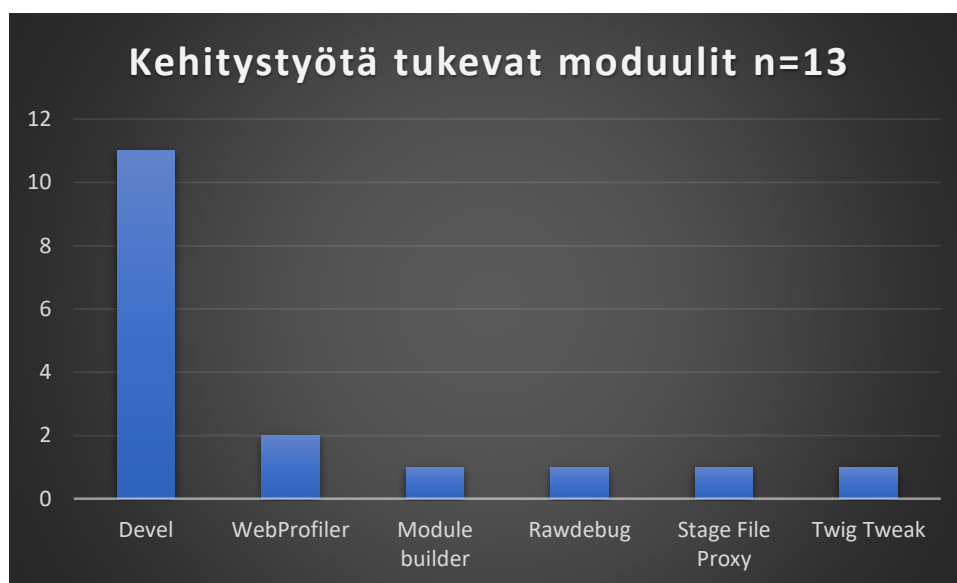
Tutkimuksen sähköinen kyselylomake luotiin Google Forms-työkalulla ja se jaettiin Drupal-yhteisön yleisellä Slack-kanavalla. Kyselyn vastausvaihtoehdot johdettiin Drupal-dokumentaation esittämistä työkaluista, jotka olivat jaettuna eri kategorioihin. Vastaaajille annettiin myös mahdollisuus lisätä omia työkaluja, joita Drupalin tarjoama dokumentaatio ei listannut. Näin saatiin kerättyä myös semmoiset mahdolliset työkalut, jotka eivät ole mainittuna Drupalin omissa suosituksissa.

Tutkimuksen kvantitatiivisen osuuden tulokset siirrettiin Excel-ohjelmaan analysointia varten, jotta numeerinen data pystyttiin helposti muuttamaan graafiseksi dataksi. Työkalujen käyttöasteet visualisoitiin pylväsdiagrammeihin, jotta työkalujen suosio saataisiin esitettyä selkeässä muodossa. Drupal-yhteisön merkitystä koskevan avoimien kysymyksen vastaukset analysoitiin käyttäen teematista analyysiä ja tarkemmin sanottuna teemoittelua. Teemoittelussa keskitytään tunnistamaan tutkimuskysymykseen liittyviä keskeisiä aiheita ja korostamaan aineistossa toistuvia piirteitä (Juhila). Teemoittelua hyödynnettäessä on hyvä ottaa huomioon se, että sen tulokset, eli aineistosta nousevat teemat ovat pitkälti tutkijan oman tulkitsemisen tuotosta. Tällöin samasta aineistosta saattaisi syntyä useita eri tulkintoja. (Tuomi, 2018.) Oma kokemus Drupal-yhteisön parissa auttaa minua ymmärtämään aineiston sisältöä ja kontekstia, mikä edesauttaa teemoitteluprosessia. Oma kokemus voi tuoda kuitenkin myös ennakkokäsityksiä, jotka voivat vaikuttaa tulkintoihin. Tämä on pyritty ottamaan huomioon analyysivaiheessa olemalla kriittinen omia oletuksia kohtaan.

5 Tulokset

Tässä osiossa käydään läpi kyselytutkimuksen tulokset. Kysely toteutettiin selvittämään, mitkä moduulit, komentorivityökalut, integroidut kehitysympäristöt sekä kehitysympäristö- ja konttiratkaisut ovat saavuttaneet suurimman suosion Drupal-kehittäjien keskuudessa. Tämän lisäksi analysoidaan tuloksia, miten tämän päivän Drupal kehittäjät hyödyntävät aktiivista yhteisöä omissa kehitystehtävissään. Kyselyn vastausvaihtoehdot johdettiin Drupalin oman dokumentaation suositteluista, joita esitettiin tarkemmin luvussa 3. Avointen kysymysvaihtoehtokenttien avulla pyrittiin kartoittamaan semmoisia työkaluja, jotka eivät olleet listattuna Drupalin omassa dokumentaatiossa.

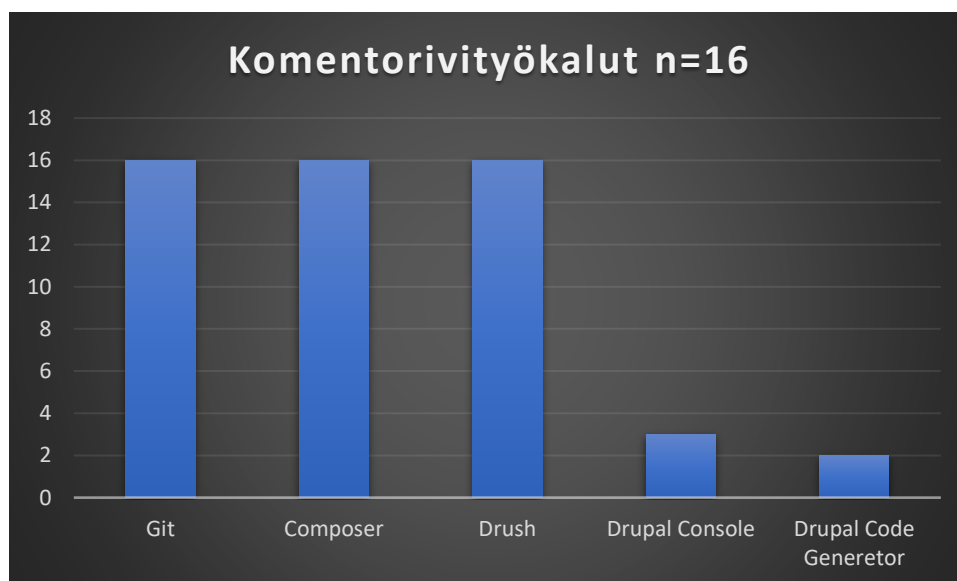
5.1 Kehitystyötä tukevat moduulit



Kuva 8 Kehitystyötä tukevat moduulit

Kyselyn tulokset osoittavat selvästi, että Devel on ylivoimaisesti suosituin työkalu vastaajien keskuudessa, sillä 11/13 vastaajista ilmoitti käyttävänsä sitä. Tämä ei ole yllättävää ottaen huomioon Devel-moduulin monipuolisuuden ja sen tarjoamat työkalut, jotka nopeuttavat ja tehostavat Drupal-kehitystä. Lisäksi aiempi huomio, että yli 175 000 sivustolla käyttää Devel-moduulia, tukee tätä havaintoa. Devel-moduulin suosio oli odotettavissa myös siksi, että kuten aiemmin on mainittu, se tarjoaa pohjan monelle muulle kehitystyötä tukevalle moduulille, kuten WebProfiler. WebProfiler saa kyselyssä seuraavaksi eniten mainintoja. Vaikka WebProfiler ei ole yhtä suosittu kuin Devel, se on merkittävä osa Drupal-yhteisömoduulien joukossa ja tarjoaa tärkeitä toimintoja kehittäjille. Huomioitavaa on, kuten aiemminkin on mainittu, WebProfiler on yksi Devel-moduulin laajennuksista, vaikka se onkin oma moduulikokonaisuutensa. Tästä syystä WebProfilerin käyttäjät hyödyntävät myös epäsuorasti Devel-moduulin ominaisuuksia.

5.2 Komentorivityökalut

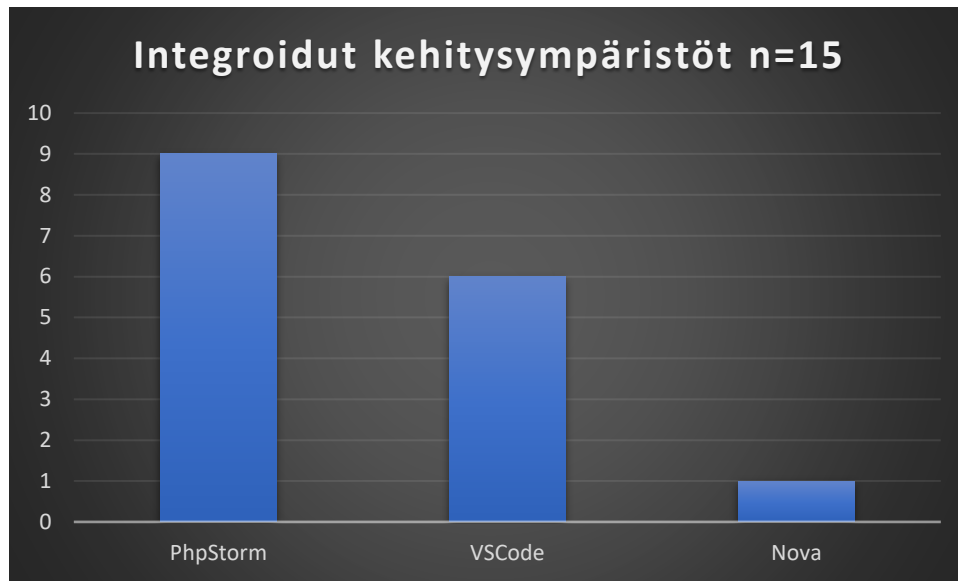


Kuva 9 Komentorivityökalut

Kyselyn tulosten perusteella voidaan nähdä, että Git, Composer ja Drush ovat ehdottomasti suosituimmat komentorivityökalut Drupal-kehityksessä, sillä jokainen niistä saavutti kaikkien vastaajien suosion. Tämä korostaa niiden keskeistä asemaa ja merkitystä Drupal-kehittäjille. Toisaalta vaikka Drupal Console ja Drupal Code Generator tarjoavat arvokkaita työkaluja kehityksen helpottamiseen, ne eivät ole saavuttaneet yhtä suurta suosiota, kuten taulukosta voidaan nähdä.

Git:n, Composerin ja Drush:n käyttöprosentti ei yllättänyt, sillä niin kuin aiemmin on mainittu, ne ovat jokseenkin standardi työkalu Drupal-ympäristössä. Moduulikehityksen näkökulmasta koodin generointiin vaikuttavat työkalut ovat yllättävän alhaisella tasolla ottaen huomioon, että ne nopeuttavat moduulipohjan luomista sekä valmisgeneroitu moduulipohja vähentää mahdollisten virheiden määrää.

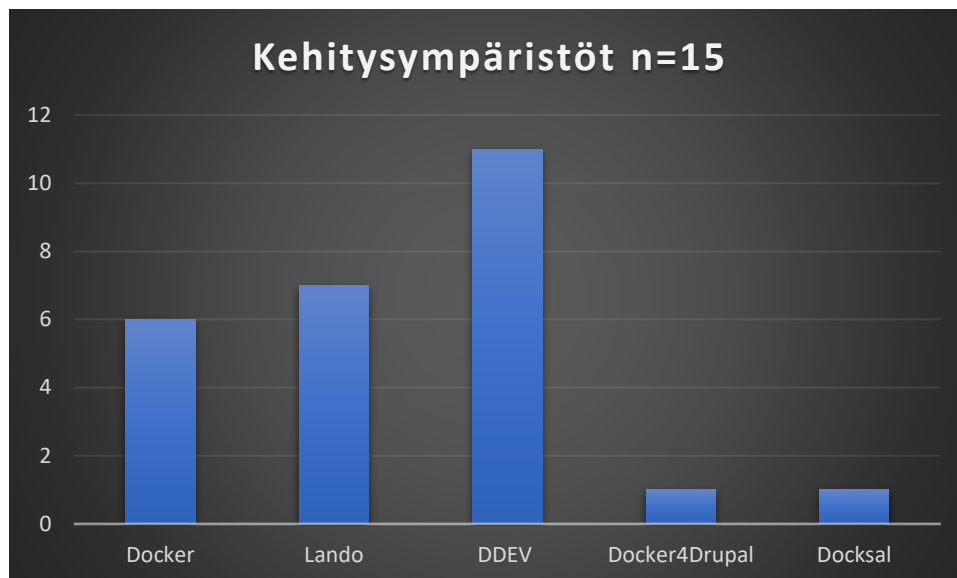
5.3 Integroidut kehitysympäristöt



Kuva 10 Integroidut kehitysympäristöt

Kun tarkastellaan integroitujen kehitysympäristöjen suosiota, voidaan huomata, että PhpStorm on suosituin työkalu vastaajien keskuudessa, sillä 9/15 vastaajista ilmoitti käyttävänsä sitä. Seuraavaksi suosituin on VSCode, joka sai 6/15 vastausta. PhpStormin johtava asema osoittaa, että monet kehittäjät pitävät sen ominaisuuksia ja toimintoja arvossa, kun mietitään Drupal kehitystä. VSCode on kyselyn toiseksi suosituin integroitu kehitysympäristö, se saattaa olla suosittu sen monipuolisten laajennusten ja mukautettavuuden ansiosta, jotka voivat tehdä kehitysprosessista sujuvampaa.

5.4 Kehitysympäristöt ja konttiratkaisut



Kuva 11 Kehitysympäristöt

Kyselyn tulosten perusteella on selvää, että Docker-pohjaiset kehitysympäristöt ovat suosittuja Drupal-yhteisössä. DDEV on kyselyn perusteella suosituin kehitysympäristötyökalu, sillä 11/15 vastaajista ilmoitti käyttävänsä sitä. Tämä korostaa sen merkitystä PHP-pohjaisten sovellusten, erityisesti Drupalin, kehittäjien keskuudessa. DDEV:n monipuoliset toiminnot, kuten helpotettu sovelluskehitysympäristön luominen ja suorittaminen, sekä komentojen ajaminen suoraan Docker-konttiin, ovat tekijöitä, jotka saattavat selittää sen suuren suosion (ks. 3.4).

Lando, toinen Docker-pohjainen työkalu, saavutti seitsemän vastaajan suosion ja on siten toiseksi suosituin kyselyn mukaan. Vaikka Lando tukeekin laajempaa käyttötarkoitusta kuin vain PHP-pohjaisia sovelluksia, sen vahvuus Drupal-kehityksessä on sen toiminnoissa, jotka tekevät Drupal-kehitysympäristön luomisesta ja suorittamisesta vaivatonta (ks. 3.4).

Mielestäni yhteenvetona voidaan todeta, että Docker-pohjaiset kehitysympäristöt ovat vakiinnuttaneet keskeiseen rooliin Drupal-kehityksessä. DDEV ja Lando ovat erityisen suosittuja työkaluja, ja niiden toiminnot ja ominaisuudet sopivat hyvin Drupal-yhteisön tarpeisiin.

5.5 Drupal yhteisön merkitys kehitystyössä

Tarkastellessani saamiani vastauksia avoimeen kysymykseen, erottui vastauksista keskeisiä teemoja, jotka kuvaavat Drupal-yhteisön merkitystä kehittäjille. Ensimmäinen keskeinen teema on tuki ja avoimuus. Monet vastaajat korostivat yhteisön avoimuutta ja tukiroolia. Yhteisö nähdään

paikkana, jossa ongelmat ratkaistaan yhteistyössä ja jossa tietoa jaetaan vapaasti. Tämä korostaa Drupal-yhteisön merkitystä osana kehittäjien työkalupakkia.

Toinen teema, joka nousi vahvasti esiin, on tekninen apu. Vastaajat arvostivat yhteisöä erityisesti sen tarjoaman teknisen tuen vuoksi, kuten bugien korjaaminen ja ongelmatilanteiden ratkaisu. Tämä korostaa yhteisön asiantuntemusta ja sen roolia kehittäjien arjessa. Kolmanneksi teemaksi korostui sosiaalinen verkosto, joka on olennainen osa Drupal-yhteisöä. Vastaajien maininnat erilaisista tapahtumista, ryhmistä ja kanavista korostavat yhteisön sosiaalista luonnetta. Se ei ole vain tekninen foorumi, vaan myös paikka verkostoitumiselle, oppimiselle ja yhteistyölle.

6 Pohdinta

Tutkimuksen tavoitteena oli tarkastella moduulikehityksen tukemiseen liittyviä työkaluja. Kehitystyön aikana on erinäisiä tarpeita, kuten työprosessien tehostamisen tarve, virheiden vähentäminen, sovelluskehityksen nopeuttaminen sekä ongelmatilanteiden ratkominen. Kyselytutkimuksen avulla pyrittiin selvittämään, mitkä työkalut ovat saavuttaneet suurimman suosion Drupal kehittäjien keskuudessa, sekä millainen merkitys Drupal yhteisöllä on kehitystyössä.

6.1 Yhteenveto ja johtopäätökset

Kehitystyötä tukevien moduulien osalta tutkimus paljasti, että Devel on ylivoimaisesti suosituin työkalu. Jopa 11/13 vastaajista ilmoitti käyttävänsä sitä. Devel-moduulin monipuolisuus ja sen tarjoamat työkalut ovat keskeisiä tekijöitä sen suosion takana. Toinen mainittava moduuli on WebProfiler. Vaikka se ei ole yhtä suosittu kuin Devel, sillä on silti tärkeä rooli Drupal-yhteisömoduulien joukossa.

Komentorivityökalujen osalta Git, Composer ja Drush nousivat selvästi esiin suosituimpina työkaluina. Ne ovat vakiinnuttaneet paikkansa Drupal-kehityksessä, sillä niitä hyödyntää kaikki kyselyn vastaajat. Vaikka Drupal Console ja Drupal Code Generator tarjoavat myös arvokkaita työkaluja, ne eivät ole yltäneet samanlaiseen suosioon. Huomioitavaa on, että työkalut, jotka vaikuttavat koodin generointiin moduulikehityksen näkökulmasta, ovat suosiossa yllättävän alhaalla. Tämä saattaa selittyä sillä, että koodin generointiin on tarjolla myös muita kuin komentorivillä hyödynnettäviä työkaluja.

Integroiduista kehitysympäristöistä PhpStorm nousi suosituimmaksi IDE:ksi, 9/15 vastaajan suosiolla. Tätä seurasi VSCode 6/15 vastaajan suosiolla. PhpStormin johtava asema osoittaa sen arvostuksen Drupal-kehitysyhteisössä. VSCode puolestaan saattaa olla suosittu sen laajennusten ja mukautettavuuden ansiosta. Huomioitavaa on myös mielestäni painottaa, että PhpStormin suosion merkitys johtuu mielestäni tutkimuksen aiheesta. Vaikka PhpStorm koetaan soveltuvaksi Drupal ympäristön IDE:ksi, se ei kuitenkaan tarkoita, että se olisi suosituin IDE muissa kehitystehtävissä.

Docker-pohjaiset kehitysympäristöt osoittautuivat suosituiksi Drupal-yhteisössä. DDEV on suosituin kehitysympäristötyökalu, jota käyttää 11/15 vastaajista. Toinen mainittava Docker-pohjainen työkalu on Lando, joka saavutti 7/11 vastaajan suosion. Molemmat työkalut, DDEV ja Lando, sopivat erinomaisesti Drupal-yhteisön tarpeisiin ja niiden toiminnot tekevät Drupal-kehitysympäristön luomisesta ja suorittamisesta helpompaa.

Tutkimustuloksista voidaan todeta, että tulokset olivat pääpiirteittäin odotetun näköisiä, joten tutkimus enemmänkin vahvisti yleisimpien työkalujen suosiota. Tutkimuksen tuloksista ei noussut esiin

uusia mahdollisia työkaluja, jotka olisivat vakiinnuttaneet asemansa Drupal yhteisön suosioon. Tämä saattaa osin johtua kyselyn suppeasta otannasta, sekä kyselyn muotoilusta, sillä ennalta määrättyt vastausvaihtoehdot saattavat ohjata vastauksia, vaikka avointen vastausten kenttä oli myös esitetty jokaisessa kysymyksessä.

Tutkimustulokset tarjoavat tietoa siitä, että mitä työkaluja tämän päivän Drupal kehittäjät suosivat omissa kehitystehtävissään. Tutkimustulosten pohjalta lukija saa yleisymmärrystä siitä, että millaisia työkaluja Drupal yhteisö suosii, sekä tämän pohjalta tehdä päätöksiä tai ottaa käyttöön työkaluja omaan työskentelyyn.

6.2 Luotettavuus ja eettisyys

Tutkimuksessa pyrittiin saamaan kattava joukko Drupal kehittäjiä eri kokemus- ja organisaatiotaustoilla. Vaikka vastausmäärät olivat suhteellisen alhaiset, ne antavat mielestäni näkökulmaa siihen, että mitä työkaluja on käytössä, ja mitkä ovat tämän päivän suosituimpia työkaluja Drupal kehitystyössä.

Tutkimuksen aikana osallistujia tiedotettiin kyselystä Drupal yhteisön yleisellä Slack keskustelukanavalla. Tiedotteessa painotettiin tutkimuksen tarkoitusta, sekä sitä, että henkilötietoja ei kerätä kyselyn aikana missään muodossa. Aineisto säilytettiin Googlen Forms palvelussa, sekä analysoitiin henkilökohtaisella tietokoneellani. Opinnäytetyön valmistumisen jälkeen tutkimuksessa kerätty data hävitettiin.

6.3 Jatkotutkimusehdotukset

Tämän tutkimuksen tulokset tarjoavat perustan syvemmälle perehtymiselle Drupal-kehitystyökalujen käyttöön ja niiden vaikutuksiin kehitysprosesseissa. Kun otetaan huomioon teknologian jatkuva kehitys ja muutos, olisi esimerkiksi mahdollista päivittää tämä kysely muutaman vuoden välein ymmärtääkseen uusien työkalujen ja teknologioiden suosiota ja niiden vaikutusta Drupal-yhteisöön. Tämän lisäksi tämä työ rajoittui vain tiettyihin kategorioihin, sekä esitteli niiden työkaluja yleisesti. Jatkotutkimusaiheena voisi olla myös pureutua tarkemmin tiettyyn kategoriaan. Mielenkiintoista tietoa voisi saavuttaa esimerkiksi tutkimalla, että miten eri kehitysympäristöt vaikuttavat Drupal kehityksen käynnistämiseen ja ylläpitoon. Tässä voitaisiin käydä läpi esimerkiksi tässäkin työssä mainittu Docker perusteellisemmin ja verrata sitä vaikka virtuaalikonepohjaisiin ratkaisuihin Drupal kehityksessä.

Tässä työssä toteutetun tutkimuksen voisi myös toteuttaa eri tavalla. Tässä työssä tutkimus toteutettiin sähköisenä kyselynä, jossa oli ennalta määritellyjä vastausvaihtoehtoja. Vaikka vastaajilla oli mahdollisuus lisätä myös omia työkaluja avointen kenttien kautta, niin valmiit vastausvaihtoehdot

saattoivat vaikuttaa tuloksiin. Samankaltaisen tutkimuksen voisi tehdä vaihtoehtoisesti myös avoimena kysymyksenä, jolloin valmiit vaihtoehdot eivät vaikuta tuloksiin. Lisäksi tutkimuksessa voitaisiin hyödyntää haastatteluja, jotta saadaan syvempää ymmärrystä siitä, että miten tai millaisissa tilanteissa eri työkaluja hyödynnetään erityisesti.

6.4 Oman opinnäytetyön ja oppimisen arviointi

Opinnäytetyöprojekti oli itselle todella haastava siinä mielessä, että oli vaikea määritellä tietoperustan asiat siten, että siitä voisi käydä vuoropuhelua tutkimustulosten kanssa. Mielestäni opinnäytetyö kuitenkin onnistuu tuomaan tutkittavan asian tuloksineen suhteellisen hyvin esille lukijalle, mutta siinä on mielestäni vajavaisuuksia kontekstin luonnissa tietoperustan avulla. Oppiminen työn aikana liittyi suurilta määrin Drupalin parissa käytettyihin työkaluihin. Vaikka minulla onkin henkilökohtaista kokemusta useasta työkalusta, opin opinnäytetyöprojektin aika kiinnostavia ominaisuuksia eri työkaluista. Tämän lisäksi opin tutkimustyyppisen työn eri vaiheita ja toimintoja.

Lähteet

- Amin, K. (2023). Devel module in Drupal 9 to generate dummy content. Saatavilla: <https://www.specbee.com/blogs/devel-module-in-drupal-9-to-generate-dummy-content>. Viitattu: 2.10.2023.
- Anderson, J. (2019). 10 Must-Have Modules to Start Your Drupal 8 Site. Acquia. Saatavilla: <https://www.acquia.com/blog/10-must-have-modules-start-your-drupal-8-site>. Viitattu: 25.9.2023.
- Atlassian. (n.d.). What is Git. Saatavilla: <https://www.atlassian.com/git/tutorials/what-is-git>. Viitattu: 2.10.2023.
- AWS. (n.d.). What Is An IDE (Integrated Development Environment)? Saatavilla: <https://aws.amazon.com/what-is/ide/>. Viitattu: 2.10.2023.
- Bhadra, D. (2018). How to debug Drupal 8 website performance using Web Profiler. Saatavilla: <https://www.valuebound.com/resources/blog/how-to-debug-drupal-8-website-performance-using-web-profiler>. Viitattu: 2.10.2023.
- Byron, A., Berry, A., Haug, N., Eaton, J., Walker, J., & Robbins, J. (2008). Using Drupal. O'Reilly Media.
- DDEV GitHub repository. (2023). Saatavilla: <https://github.com/ddev/ddev>. Viitattu: 20.10.2023.
- DiCesare, M. (2022). The Top 5 Benefits of Developing with Containers. Saatavilla: <https://www.mendix.com/blog/benefits-of-containerization/>. Viitattu: 20.10.2023.
- Drupal.org. (2023a). Understanding hooks. Saatavilla: <https://www.drupal.org/docs/develop/creating-modules/understanding-hooks>. Viitattu: 26.9.2023.
- Drupal.org. (2023b). Basic structure. Saatavilla: <https://www.drupal.org/docs/develop/creating-modules/basic-module-building-tutorial-lorem-ipsam-generator/basic-structure>. Viitattu: 26.9.2023.
- Drupal.org. (2022c). Devel. Saatavilla: <https://www.drupal.org/project/devel>. Viitattu: 2.10.2023
- Drupal.org. (2023d). Module Builder. Saatavilla: https://www.drupal.org/project/module_builder. Viitattu: 2.10.2023.
- Drupal.org. (2020e). Using Composer with Drupal. Saatavilla: <https://www.drupal.org/docs/develop/using-composer/using-composer-with-drupal>. Viitattu: 2.10.2023.

Drupal.org. (2023f). Overview of Drupal. Saatavilla: <https://www.drupal.org/docs/understanding-drupal/overview-of-drupal>. Viitattu: 2.10.2023.

Drupal.org. (2022g, 15 huhtikuuta). Basic structure of Drupal. Viitattu 2 lokakuuta 2023, osoitteesta <https://www.drupal.org/docs/core-modules-and-themes/basic-structure-of-drupal>. Viitattu: 2.10.2023.

Drupal.org. (2019h). Contributed versus custom code and everything in between. Saatavilla: <https://www.drupal.org/docs/7/site-building-best-practices/contributed-versus-custom-code-and-everything-in-between>. Viitattu: 26.9.2023.

Drupal.org. (2021i). Cycle News Display. Saatavilla: https://www.drupal.org/project/cycle_news_display. Viitattu 2.10.2023.

Drupal.org. (2023j). Installing Modules. Saatavilla: <https://www.drupal.org/docs/extending-drupal/installing-modules>. Viitattu 2.10.2023

Drupal.org. (2023k). Docker-based solutions overview. Saatavilla: <https://www.drupal.org/docs/develop/local-server-setup/docker-based-development-environments/docker-based-solutions-overview>. Viitattu: 20.10.2023.

Drupalize.Me. (2023a). Drupal Core Modules. Saatavilla: <https://drupalize.me/topic/drupal-core-modules>. Viitattu: 26.9.2023.

Drupalize.Me. (2023b). What Is Drush? Saatavilla: <https://drupalize.me/tutorial/what-drush-0>. Viitattu: 2.10.2023.

DXPR Marketing Team. (2022). What Is Drupal? Explained for Beginners. DXPR. Saatavilla: <https://dxpr.com/drupal-blog/what-is-Drupal>. Viitattu: 20.9.2023.

Farhaz, M. (2021). DRUPAL 9 CUSTOM MODULE DEVELOPMENT – A BEGINNERS GUIDE. Specbee. Saatavilla: <https://www.specbee.com/blogs/drupal-9-custom-module-development>. Viitattu: 26.9.2023.

Grand Canyon University. (2023). Qualitative vs. Quantitative Research: What's the Difference? Saatavilla: <https://www.gcu.edu/blog/doctoral-journey/qualitative-vs-quantitative-research-whats-difference>. Viitattu: 17.10.2023.

Jha, S. (2023). Drupal 9 Custom Module Development. Saatavilla: <https://www.zyxware.com/article/drupal-9-custom-module-development-guide>. Viitattu: 26.9.2023.

Juhila K. (n.d.). Teemoittelu. Saatavilla: <https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/analyysitavan-valinta-ja-yleiset-analyysitavat/teemoittelu/>. Viitattu: 17.10.2023.

Lando documentation. (2022). Drupal. Saatavilla: <https://docs.lando.dev/drupal/>. Viitattu: 20.10.2023.

MCD. (2019). Drupal Pros and Cons: An In Depth Look. Saatavilla: <https://www.mcdpartners.com/news/drupal-pros-and-cons/>. Viitattu: 16.10.2023.

Picchini, R. (2020). Top 5 Advantages of Drupal vs. Other Content Management Systems. Acquia. Saatavilla: <https://www.acquia.com/blog/top-5-advantages-drupal-vs-other-content-management-systems>. Viitattu: 20.9.2023.

Salvisberg, H. (2018). README for Devel module. Saatavilla: <https://git.drupalcode.org/project/devel/blob/7.x-1.x/README.txt>. Viitattu: 2.10.2023

Sinha, A. (2021, September 22). Creating Your First Custom Module in Drupal 8 or Drupal 9. Medium. <https://medium.com/@anupeiie.fiem/creating-your-first-custom-module-in-drupal-8-or-drupal-9-dc715ee15d1e>. Viitattu 2.10.2023.

Tuomi, J. & Sarajärvi, A. 2018. Laadullinen tutkimus ja sisällönanalyysi. Uudistettu laitos. Helsinki: Kustannusosakeyhtiö Tammi.

Tähtinen, J., Laakkonen, E., Broberg, M. & Tähtinen, R. 2020. Tilastollisen aineiston käsittelyn ja tulkinnan perusteita. 2. uudistettu painos. Turku: Turun yliopiston kasvatustieteiden laitos.

Umbraco. (n.d.). Development Environment. Saatavilla: <https://umbraco.com/knowledge-base/development-environment/>. Viitattu: 20.10.2023.

Zia, H. (2018). Five Best Drupal Security Modules to Prevent Breaches. Cloudways. Saatavilla: <https://www.cloudways.com/blog/top-drupal-security-modules/>. Viitattu: 20.9.2023.

Liitteet

Liite 1. Kyselyn runko

Drupal Development Tools Survey

Welcome to my survey! This questionnaire is a part of my thesis research, aiming to identify the most commonly used tools among Drupal developers today. Your insights will help me gain a clearer understanding of how the Drupal community operates and which tools are most frequently used to achieve the best results. The response options are derived from popular tools listed on the Drupal.org website. However, I recognize that there might be many other tools not mentioned here in the options. Thus, I hope you could name and share any additional tools you regularly use in Drupal module development. Your responses will remain anonymous and will solely be used for academic research purposes. Thank you for your time and contribution!

Adding tools that are not listed

If you want to add tools that aren't listed in a specific category (modules, command-line tools, etc.), you can list them under the "Other" section. If you add multiple tools, please separate them with a comma.

1. Which of the following Drupal modules do you regularly use in your development workflow?
 - Devel
 - WebProfiler
 - Module Builder
 - Rawdebug
 - Muu: _____
2. Which of the following command-line tools do you utilize most frequently in your Drupal development?
 - Git
 - Composer
 - Drush
 - Drupal Console
 - Druml
 - Drupal Code Generator
 - Muu: _____

3. Which development environment or container solution do you prefer when working with Drupal?

- Docker
- Lando
- DDEV
- Docker4Drupal
- Docksal
- Muu: _____

4. Which Integrated Development Environment (IDE) do you find most effective for your Drupal development tasks?

- PhpStorm
- VSCode
- Muu: _____

5. How has the active Drupal community supported you in your Drupal development?