



Antti Akonniemi toimii ohjelmistokehittäjänä ja toimitusjohtajana Kisko Labs Oy:ssä. Antti on myös yksi Suomen Ruby Brigaden (www.rubybrigade.fi) perustajia.

Haluatko vaihtaa leiriä? Ota yhteyttä ja liity taisteluun saamattomuutta vastaan!

Kun aika on oikeasti rahaa – enterprise-kehityksen tulevaisuus

Ruby on Rails on alun perin 37Signals -yhtiön David Heinemeier Hanssonin kehittämä web-ohjelmointikehys, jonka pohjalla on japanilaisen Yukihiko Matsumoton kehittämä Ruby-ohjelmointikieli. Rails kehitettiin auttamaan 37Signalin web-kehitystä. Se vastasi kysymyksiin siitä, kuinka toteuttaa web-sovelluksia ja -palveluita mahdollisimman helposti, nopeasti, elegantisti ja hauskasti.

Railsin avulla on mahdollista tehdä helposti ylläpidettävää koodia. Uudet yksikkötestikehykset mahdollistavat sujuvan testien kirjoituksen. Ei enää tekosyitä sille, että testien kirjoittaminen olisi tylsää, hankalaa tai inhottavaa. Täysin yksikkötestattua koodia on ilo ylläpitää ja muokata. Näin ylläpito on myös halpaa. Uusina tuulina mainittakoon tuki ja työkalut Behaviour Driven Developmentille (BDD) sekä Story Driven Developmentille. Ylläpitoa helpottaa myös Railsin korkea abstraktiotaso. Haluatko vaihtaa tietokannan MySQL:stä Oracleen – no problem! Platformista toiseen – ei hätää! Toisin kuin monessa muussa kehityksessä, Railsissa abstrahointi toimii.

Toisin sanoen, Rails muuttaa ohjelmistokehityksen taloutta visiolla. Sovitut käyttäytymismallit ja tietynlainen jääräpäisyys nopeuttavat ohjelmistokehitystä sekä laskevat kuluja.

Bisnes mielessä

Kehittäminen Railsilla on nopeaa. Ensimmäinen prototyyppi on valmis minuuteissa tai viimeistään tunneissa. Osallistuin äskettäin Java-projektiin, jossa olin hämmentynyt projektin starttauksen kestäessä viikkoja. Senkään jälkeen projektin tuotosta ei mielestäni voinut kutsua prototyyppiä. Onhan se XML-tiedostojen konffaus varmasti suurta hupia, mutta riittääkö hupia todella viikoiksi?

Liiketoimintakin lienee kiinnostunut nopeasta kehityksestä. Enkä puhu nyt ison sinisen konsulttitalon ikuisuusprojekteista, vaan sellaisista, joissa asiakas saa mahdollisimman nopeasti arvoa tuottavan sovelluksen käyttöönsä pienellä investoin-

nilla. Ajattele, jos liiketoiminnan ideoima sovellus olisi käytössä jo kuukauden päästä. Ei pelkästään säästöä kehityskuluissa vaan sovellus ulkona tuottamassa rahaa huomattavasti nopeammin. Liian hyvää ollakseen totta? Kokeile itse.

Investointi on muuten hauska sana. Se odottaa jotain vastineeksi. Kulu taas ei. Kuinka moni ohjelmistoprojekti on yritykselleen kulu sen sijaan, että sitä voisi kutsua investoinniksi? Ja tuleeko kehittäjien piitata tästä? Ei varmasti kaikkien, mutta jokaisen, joka kehtaa kutsua itseään ammattilaiseksi, olisi hyvä uhrata asialle ajatus. Miksi? Ymmärrän, että nämä ovat monelle tylsiä ja ehkä jopa turhia asioita pohdittavaksi. Väitän silti, että kannattaa miettiä, miten leipä pöydällesi saapuu. Kaikki ei pyöri mahdollisimman nopean algoritmisi ympärillä, vähiten nykyisessä finanssikriisissä. Samaa hengenvetoon voisin haastaa teidät yrittäjiksi, mutta jätetään se toiseen kertaan.

DRY - Don't Repeat Yourself

Railsin yksi tärkeimmistä ideologioista on DRY. Älä toista itseäsi. Se pätee omaan koodiisi ja sitä kautta myös muihin projekteihin. Ihmiset julkaisevat moduleitaan toisten käytettäväksi ja muokattavaksi. Paljon tärkeitä ohjelmistojen osia löytyy valmiina käytettäväksi ja vieläpä abstrahoituna sille tasolle, että se ei rajoita ohjelmaasi. Tarjolla on työkaluja tarpeeseen kuin tarpeeseen.

Nuorella nuoren ongelmat

Ennen kuin alan kuulostaa Railsin provikapalkkaiselta myyjältä, on hyvä käsitellä Railsin puutteita. Rails ei ole valmis. Sovelluskehys on vielä nuori ja on paljon tehtävää, ennen kuin voidaan puhua kypsästä työkalusta. Useimmiten Railsia koskeva kritiikki koskee skaalautuvuutta. Suurimpana syynä tähän pidetään Twitter.comia, jonka palvelu tökki pahasti noin vuosi sitten. Tekijäkaarti kaatoi syyt Railsin päälle. Osittain syy oli Railsin ja osittain Twitterin arkkitehtuurin. Tämä tapah-tuma kuitenkin herätti Railsin kehitystiimin panostamaan suorituskykyyn. Työ jatkuu edelleen, mutta kokemuksen ansiosta merkittäviä parannuksia on jo tehty.

Mainittu Twitter on todennäköisesti käytetyin Railsilla tehty sovellus. Toinen esimerkki korkean kuorman sovelluksesta on Friends for Sale Facebook -applikaatio. 300 miljoonaa sivulatausta kuukaudessa. Kuinka monta sivua teidän Java-portaalinne jakoikaan?

Kuten muillakaan kielillä, suurimmat ongelmat skaalautuvuuden kanssa eivät johdu kielivalinnasta vaan toteutuksesta. Rails ei ole yksin alkuaikojen skaalautuvuusongelmiensa kanssa. Myös Javalla oli nuoruudessaan vastaavia ongelmia, jotka nyt on ratkaistu. Rails ei myöskään ole kaikkea kaikille. Tärkeintä onkin valita paras työkalu käsillä olevaan työhön.

Meressä on paljon kalaa

Entä muut vaihtoehdot, kuten Python ja Django? Django on suunniteltu enemmän web-sivustojen ja -sivujen rakentamiseen, kun taas Rails pikemminkin web-palveluja ja -ohjelmistoja varten. Python ja Django ovat varmasti parempia vaihtoehtoja kuin Java tai PHP, mutta väitän, että Rubylla ohjelmointi on silti tehokkaampaa. En ole Python-asiantuntija, mutta en voi ymmärtää, kuinka pääsisin samaan tai nopeampaan kehitystahtiin käyttäen Djangoa/Pythonia, kuin minkä nyt saavutan Railsin ja Rubyn avulla. Ehkä tämä johtuu siitä, että päähäni sattuu, kun luen Python-koodia. Kieli ei vain aukea minulle. Onneksi tämän artikkelin maksimipituus estää minua kertomasta, miltä Java-koodin kirjoittaminen tai lukeminen tuntuu. Jos haluat, voin kuvailla fiilistä joskus naamatusten.

Mistä koodaajia?

Ammatikseen Railsia käyttäviä on vielä harvassa, saati sitten kilpailevia Django-/PHP-ohjelmistokehityksiä käyttäviä. Sen sijaan harrastajia, kiinnostuneita ja kohta työttömiä Java-koodaajia on paljon. Fiksujen ja innostuneiden kaverien kouluttaminen Rails-osaajiksi on helppoa, koska kieli on niin looginen. Täysin oppineeksi ehtiminen kestää hiukan kauemmin. Rails-osaajien lukumäärä kasvaa koko ajan ja isotkin organisaatiot ovat lähteneet leikkiin mukaan. Muun muassa Nokia, IBM ja Futurice tuottavat Suomessa Rails-koodia. Töitä siis pitäisi piisata.

Rubyn suosio on kovassa nousussa (kuva 1), mutta mistään mainstream-kielestä ei vielä puhuta. Ruby on viimeisimmän (lokakuu 2008) TIOBE Programming Community Indexin mukaan kymmeneksi suosituin kieli. Top5: 1. Java, 2. C, 3. C++, 4. (Visual) Basic, 5. PHP.

Kuoleva Java

Vaikka puhun tässä Railsista, suuri kiitos Javan kuolemasta menee PHP:lle. PHP muutti ohjelmistokehitystä oikeaan suuntaan todistaessaan, että ohjelmistokehitysprojektin budjetiksi ei tarvita miljoonia euroja vaan pienemmälläkin investoinnilla selvittää. PHP todisti, että ei tarvitse olla insinööri tuottaakseen web-palveluja. Ja kyllähän

Javaakin ymmärtää, vaikkei olisi penaaalin terävin kynä. Kokonaisuuden hanskaaminen vaatii kuitenkin hieman enemmän.

Java on ns. yleiskieli ja olemassa olevien Java-ohjelmistojen määrä on huikea. Java-ohjelmoijia on todella paljon. Yleiskielisyys ja kooderilauma eivät kuitenkaan mielestäni oikeuta Javan valintaan ohjelmistoprojektin toteuttamiseen. Se, että organisaatiossa käytetään muissa järjestelmissä Javaa, saattaa jo olla parempi syy. Tosin nykyään Railsinkin saa toimimaan Java-alustalla käyttäen jRubya tai jopa .NETin päällä IronRubylla.

Leikki sikseen

Mutta olkoon provosointi. Henkilökohtaisesti käytän Rubya siksi, että se on vilpittömästi ensimmäinen ohjelmointikieli, jonka ymmärrän. Railsia siksi, että saan sillä asioita aikaiseksi. Paljon. Ohjelmointikieliä on monia, mutta sen sijaan, että tekisi kaiken kuten aina ennenkin, tulisi miettiä myös sitä, mitä kielivalinta merkitsee projektille, liiketoiminnalle ja organisaatiolle. Hyvä kysymys on myös se, mitä ja millä työkalulla haluat tulevaisuudessa tehdä töitä. Töitä... Heh. Kun työskentelen Rails-projektin parissa, se ei tunnu työltä. Tekisin samaa hommaa ilman palkkiotakin. Aina-kin melkein.

Rails lyhyesti

- Pohjautuu MVC-arkkitehtuuriin
- Tarkoitettu lähinnä WWW-palvelujen rakentamiseen
- Mm. sisäänrakennettu AJAX- ja XML-tuki
- Tietokantatuki mm. MySQL-, PostgreSQL-, SQLite-, Oracle- ja DB2-kannoille

Ruby on Railsin asennukseen löytyy helppo ohjeistus osoitteesta:
www.rubyonrails.org/down

Hyviä kirjoja Railsin opetteluun:
Agile Web Development with Rails (Thomas, Hansson, Breedt ja Clark)
The Rails Way (Fernandez)
Sekä kotimaisiin voimin tehty Beginning Ruby on Rails E-Commerce: From Novice to Professional (Hellsten ja Laine)

Kuva 1.

