

Case:
MEK
LaatuVerkko

eficode

Miksi Ruby ja Rails?

Tero Parviainen, EfiCode Oy

- * Tero Parviainen, edustamassa EfiCodea jossa olen töissä johtavana arkkitehtinä
- * Tulin puhumaan parista mielenkiintoisesta teknologiasta, eli Rubysta ja Railsistä.
- * Erityisesti yhdestä isommasta keissistä jossa ollaan niitä käytetty tänä vuonna.
- * Eli rakensimme viime keväänä suurehkon webbisoftan Matkailun Edistämiskeskuselle

Henkilökunnan työtyytyväisyys

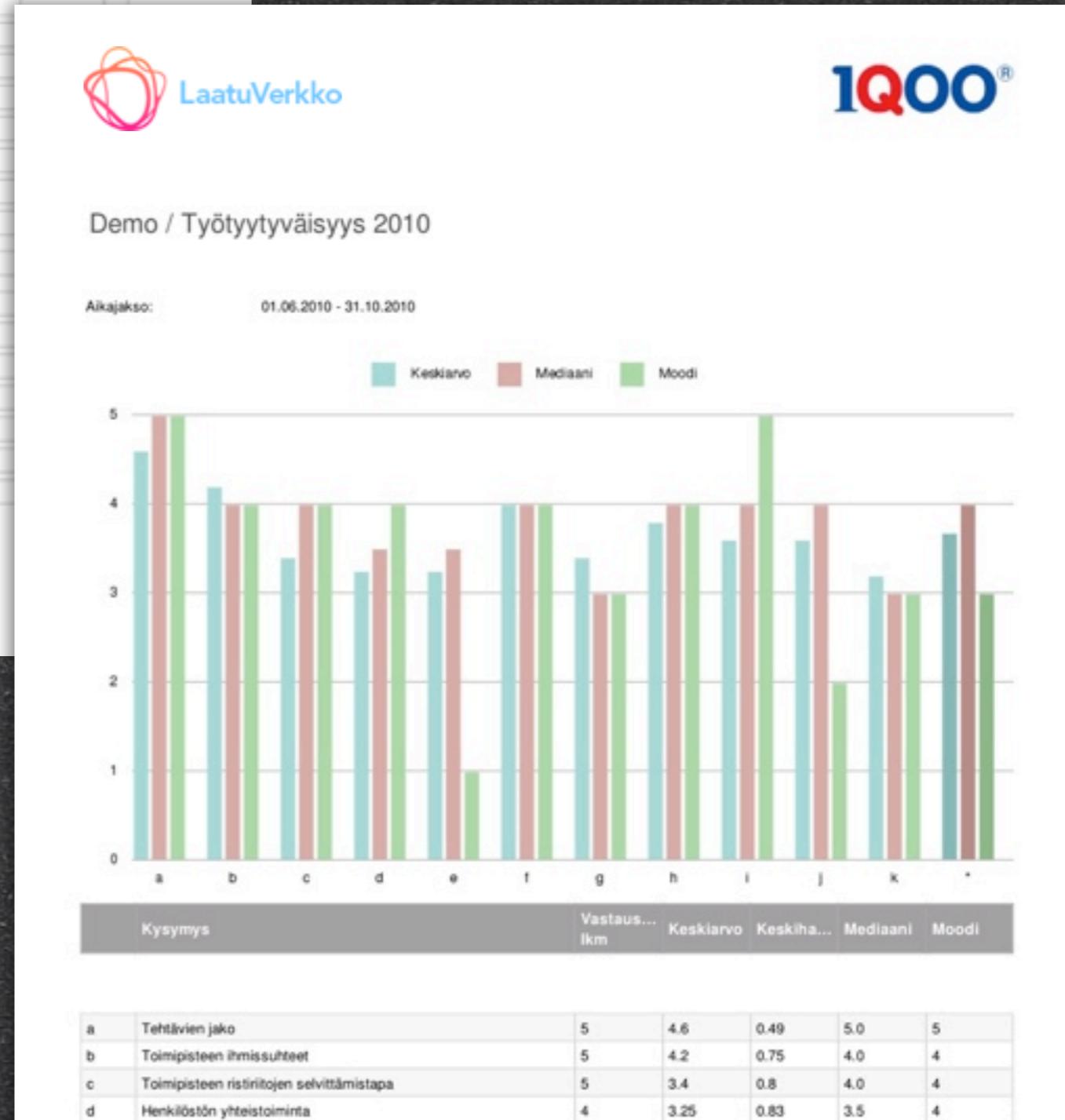
Pyydämme Sinua vastaamaan alla oleviin kysymyksiin asteikolla

 = erittäin hyvä,  = erittäin huono, ? = ei mielipidettä tai kokemusta

						?
Tehtävien jako	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Toimipisteen ihmissuhteet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Toimipisteen ristiriitojen selvittämistapa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Henkilöstön yhteistoiminta	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Oma motivaatio	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Työstä saamani arvostus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mahdollisuus kehittää omaa työtehtävää	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tiedotustoiminta yrityksessä	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Koulutustoiminta	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Työstä saamani korvaus ja etujen muoto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Perehdyttäminen työhön	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ylpeys työtäni kohtaan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Itsennäisyys ja toimintavapaus työtehtävissä	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aikomus vaihtaa työtä tai työpaikkaa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Esimieheni suhtautuminen minuun	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Esimieheni antama palautte	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Esimieheni antama tuki tarvittaessa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Esimieheni ihmissuhdetaidot	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Lupausten pitäminen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Yhteistoiminta muiden henkilöiden kanssa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kiitos vastauksistasi!

**Palautteen
keräys ja
raportointi**



- * Kyseessä on LaatuVerkko-niminen systeemi, joka on suomalaisille matkailualan yrityksille suunnattu webbisofta
- * Yritykset voivat kerätä palautetta asiakkailtaan ja henkilöstöltään, sähköisesti tai paperilomakkeilla.
- * Saavat kerätystä datasta monenlaisia näkymiä ja raportteja, joilla voivat seurata omaa kehittymistään ja verrata omia tuloksiaan kilpailijoihinsa.
- * Mukana parisataa matkailualan yritystä eri toimialoilta, eli hotelleja, hiihtokeskuksia, ravintoloita, kylpylöitä jne.
- * Melko iso projektti, tehtiin n. 4–5 hengen tiimillä puolisen vuotta, eli kokoluokaltaan n. 500 henkilötyöpäivän projekti.

Laki julkisista
hankinnoista
30.3.2007/348

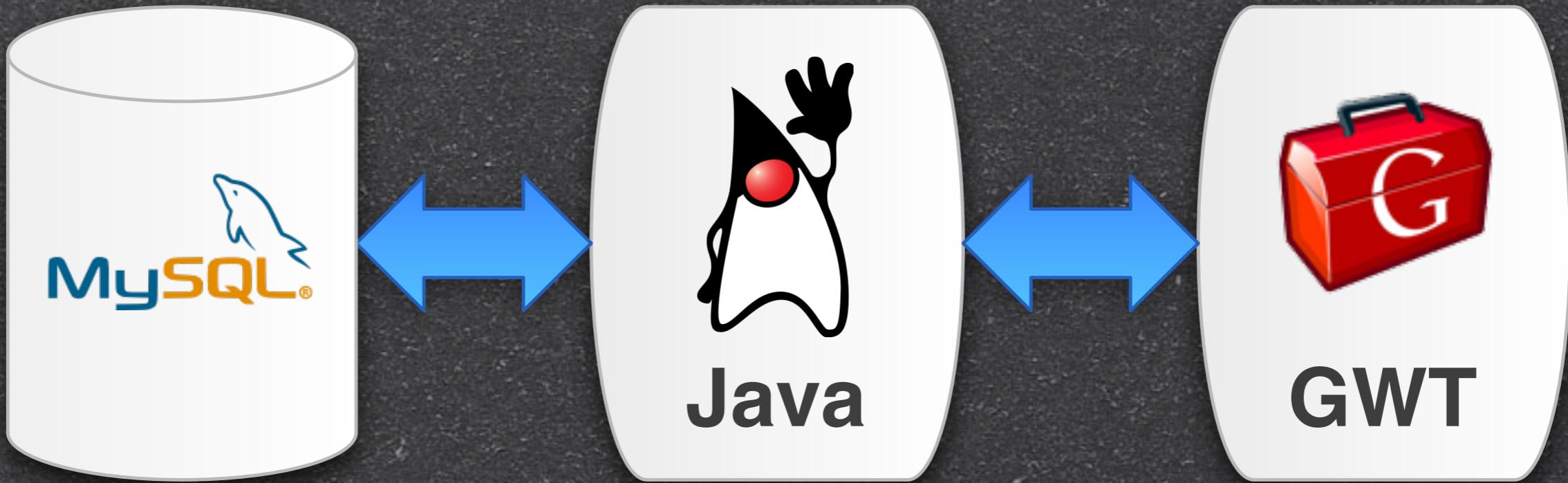


Janne Hellsten / <http://www.flickr.com/photos/50813633@N00/3853790862/>

* Asiakkaana oli siis Matkailun Edistämiskeskus, joka on Työ- ja elinkeinoministeriön alainen virasto, eli julkishallintoa.

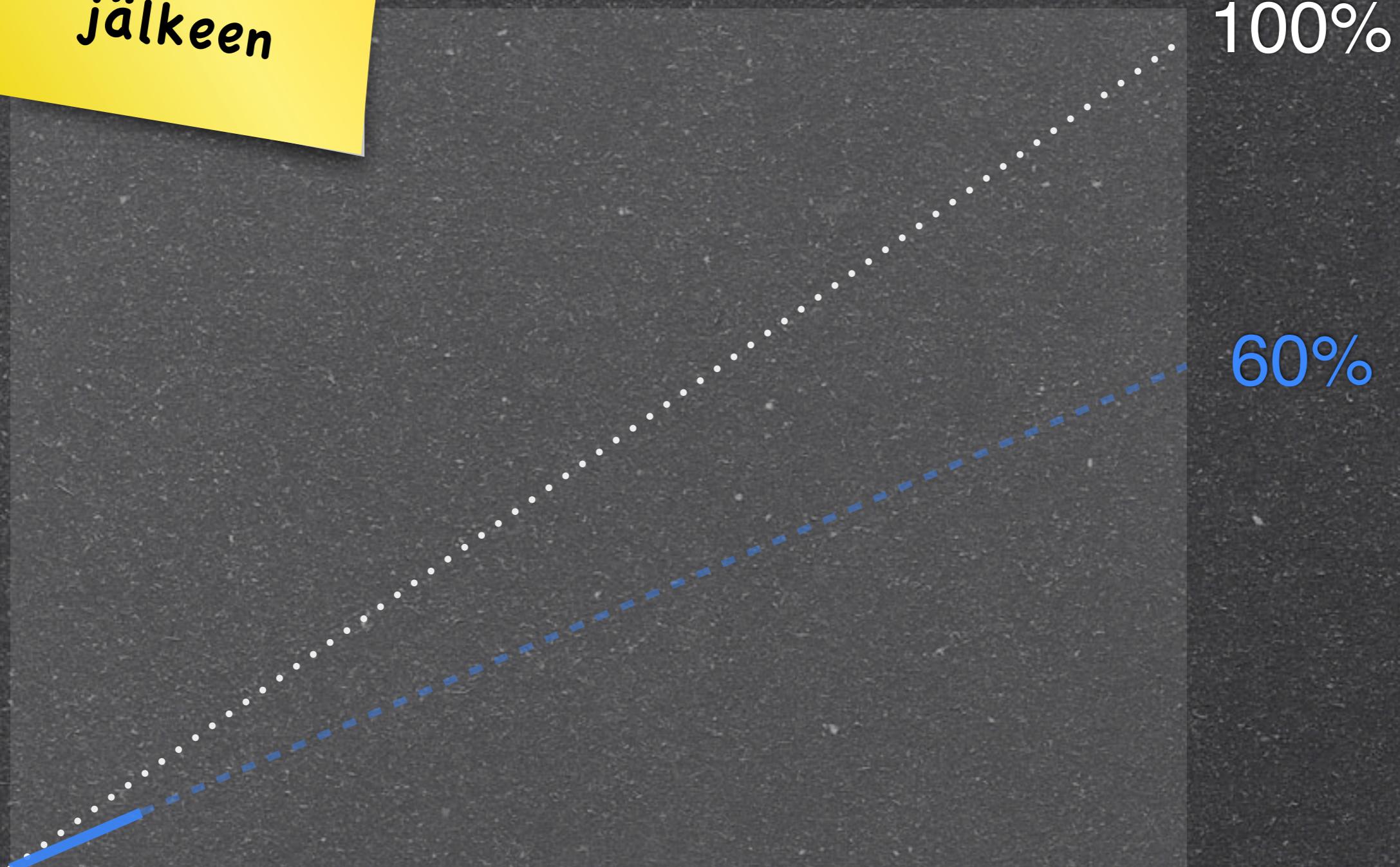
* Eli julkisesti kilpailutettu projektti. Kilpailutus, jonka EfiCode siis voitti, tehtiin syksyllä 2009. Hommat aloitettiin tämän vuoden alussa.

Arkkitehtuuri 1.0



- * Järjestelmää lähdettiin aluksi tekemään Java-alustalla.
- * Kaavailin systeemille hienon kolmikerrosarkkitehtuurin, jossa oli kannan lisäksi Java-serveri, ja käyttäjien selaimissa pyörivä käyttöliittymäsofта.
- * Tämä arkkitehtuuri oli palvellut meitä aiemmin erittäin hyvin useissa projekteissa, joissa oli tehty aika vaativiakin juttuja, kuten reaalialkapäivityksiä ja rikkaita interaktioita.

*Ennuste 1.
sprintin
jälkeen*



5 kuukautta

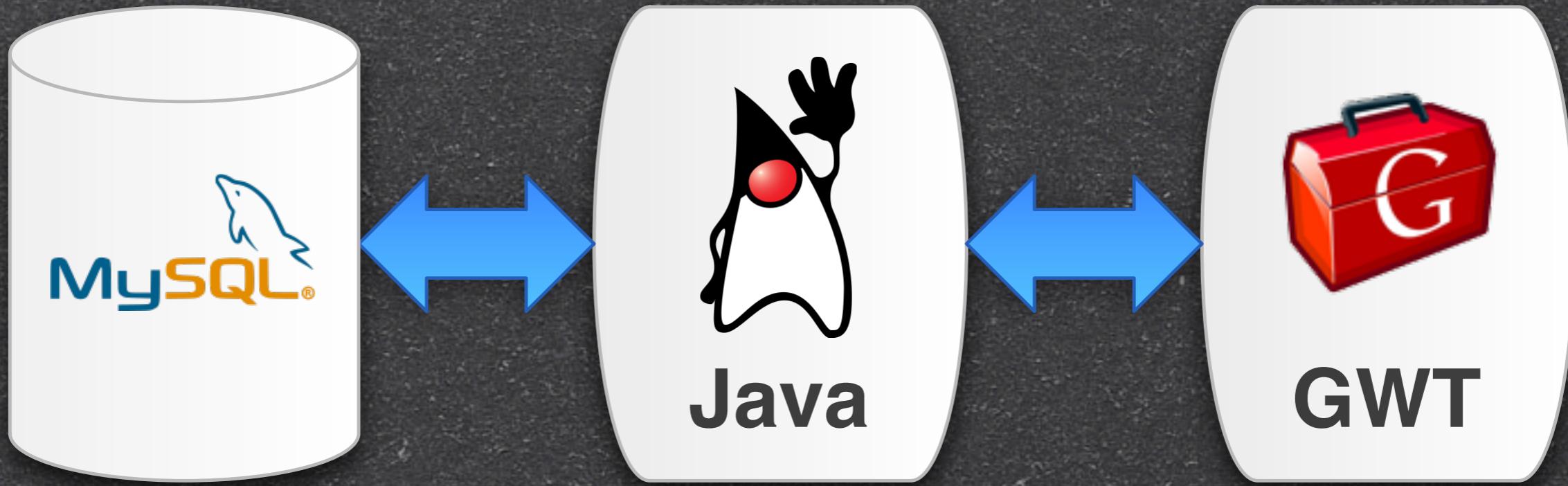
- * Hyvin aikaisin selvisi että homma ei mennyt ihan niin kuin Strömsössä.
- * Oltiin jo jääty aikataulusta, ja projektio näytti aika synkältä – olimme sovitussa aikataulussa pääsemässä vain 60% valmiustasoon.
- * Sitäkin tärkeämpi seikka: Kehitystiimin yleinen mielipide oli se, että kehitys oli karmea suojuksua ja yksinkertaisetkin asiat olivat turhan hankalia. Edistymisen tunnetta ei ollut.



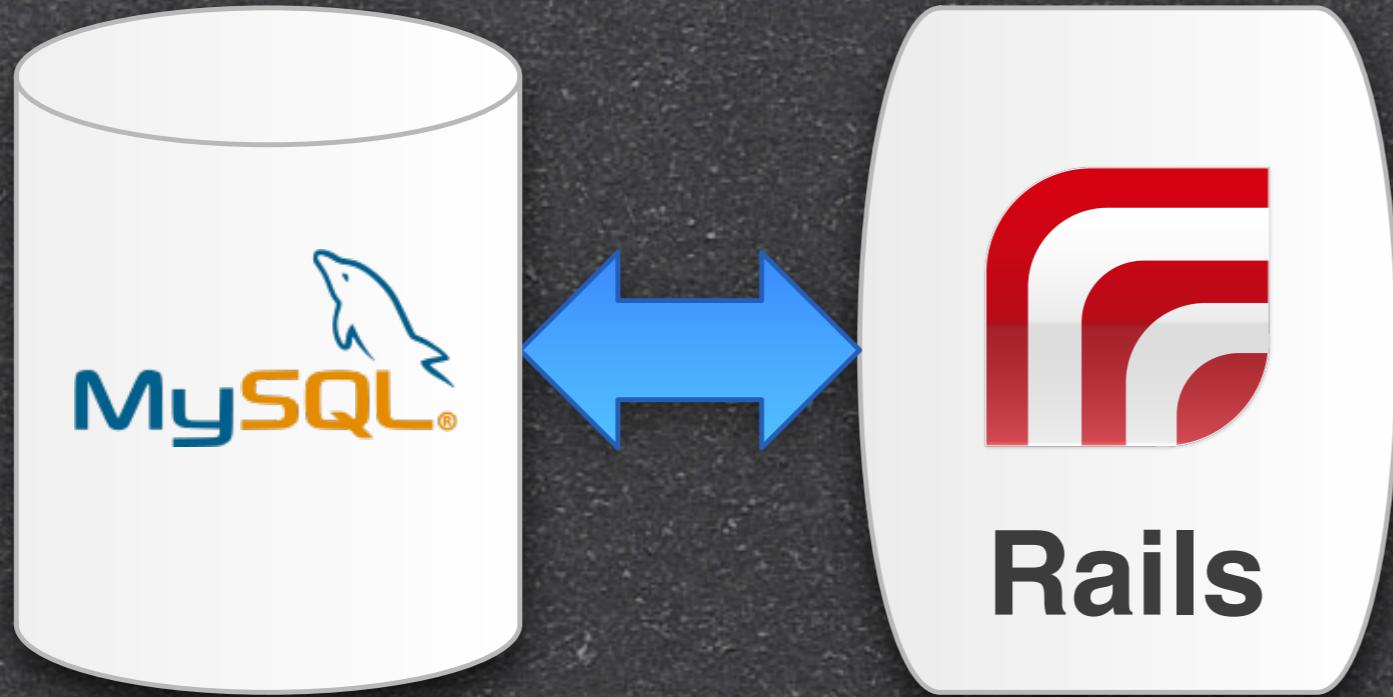
John Lambert Pearson / <http://www.flickr.com/photos/orphanjones/434806424/>

- * Oli siis tehtävä tarkka uudelleenarvointi toteutusalustan suhteen, vielä kun se oli mahdollista.
- * Oliko teknologia valinta väärä, pitäisikö alusta vaihtaa?

Arkkitehtuuri 1.0



- * Tässä vaiheessa oli melko selvää, että valittu alusta ei palvellut tarkoitusta.
- * LaatuVerkossa kyse on hyvin pitkälti datan syöttöstä ja näyttämisestä. Siinä mielessä siis melko tyypillinen businessofta.
- * Täytyy saada tehokkaasti dataa kantaan, ja takaisin sieltä ulos moninaisissa muodoissa. Rikkailta interaktioilla ja "reaaliaikaisuudella" on vähemmän merkitystä.



* Asiaa harkittiin tiimin kesken, ja myös asiakkaan kanssa. Harkinnan jälkeen päätimme ottaa takataskusta toisen ydinteknologioistamme, eli Railsin.

* Arkkitehturia yksinkertaistettiin tiputtamalla pois selainpään käyttöliittymäsofta, ja siirtymällä perinteisempään sivupohjaiseen käyttöliittymään. Rikkaita interaktioita tehtäisiin tarvittaessa Javascriptillä.

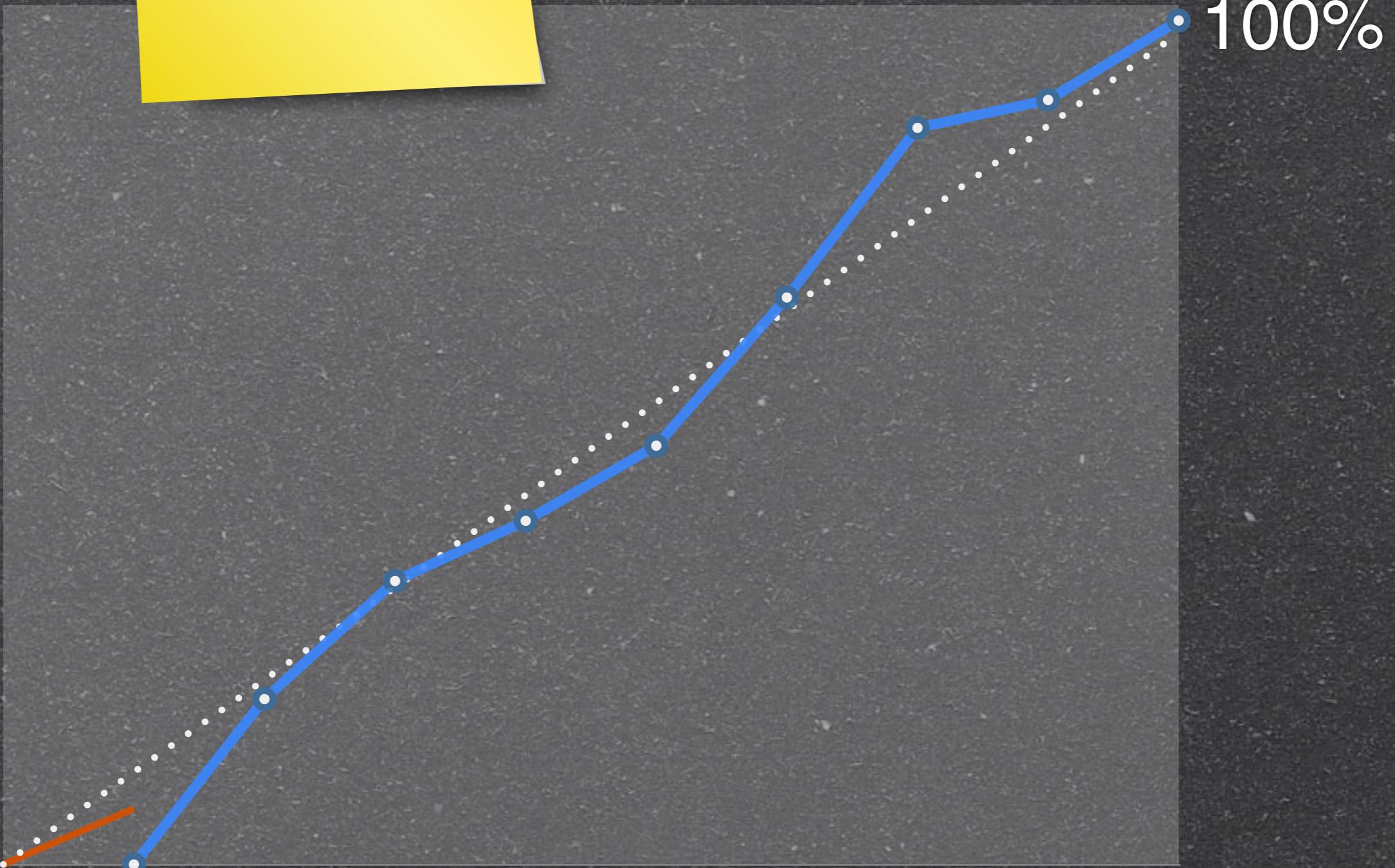
**Chuck Norris can
finish an infinite
loop in 1.3
seconds.**



Dru Bloomfield / <http://www.flickr.com/photos/athomeinscottsdale/4526833657/>

- * Tämän jälkeen homma alkoi toimimaan.
- * Valitsimme Railsin koska se on todellinen täsmäase tämän tyypisiin sovelluksiin: Se tekee erittäin helpoksi ja kevyeksi tällaisten data-vetoisten businessoftien teon, koska se on suunniteltu nimenomaan sitä varten.
- * Miksi Railsia ei sitten valittu heti aluksi? Syitä on monia, ehkä tärkein niistä oli, että LaatuVerkosta vuonna 2004 tehdy aikaisempi versio oli tehty Java-alustalla, ja ajattelimme että siitä voisi käyttää osia uudelleen. Tämä kuitenkin osallistui liian pieneksi edukksi ollakseen merkityksellinen. Lisäksi vanhan järjestelmän koodi oli sellaista että sen oli parempi antaa olla. :)

Lopullinen tulos



5 kuukautta

* Teknologiavaihdos siis toimi. Projektin viettiin päätökseen menestyksekkäästi, ja softa julkaistiin kesäkuussa.

* Tänä syksynä on tehty myös pienempi jatkokehitysprojekti joka meni erittäin mallikkaasti. Kehitys on ollut siis kestävä.



Miksi
Rails?

Milos Milosevic / <http://www.flickr.com/photos/21496790@N06/5065834411/>

- * Mitkä sitten ovat sellaisia tekijöitä jotka tekevät Railsista hyvän kehitysalustan?
 - * Tietysti yhtenä tärkeänä syynä on aikaisemmin mainittu keveys, ja sopivuus juuri tämän tyypisiin softiin.
 - * Lisäksi on kaksi muuta mielestäni tärkeää syytä, joista puhun tänään.
-
- * Mikään mielenkiintoinen softa ei kuitenkaan ole niin yksinkertainen, että pyöritellään vain dataa kantaan ja takaisin.



PDF:t



Excelit



Graafien
generointi



Kuvan-
käsittely



Moni-
kielisyys



Auth



Mailit



Palvelinten
hallinta

Oheistauhka

* Business-softiin kuuluu aina kaikennäköistä toiminnallisuutta, joihin tarvitaan **kirjastoja**.

* LaatuVerkossa ei teknisesti tehdä mitään erityisen tajunnanrääytävää, mutta tarvitsemme silti monenlaisia apukirjastoja mm. PDF:ien ja Excel-dokumenttien generointiin, graafien generointiin, kuvien manipulointiin, monikielitukeen, käyttöoikeuksien hallintaan ja tarkistuksiin, mailien lähetykseen ja palvelinten hallintaan.

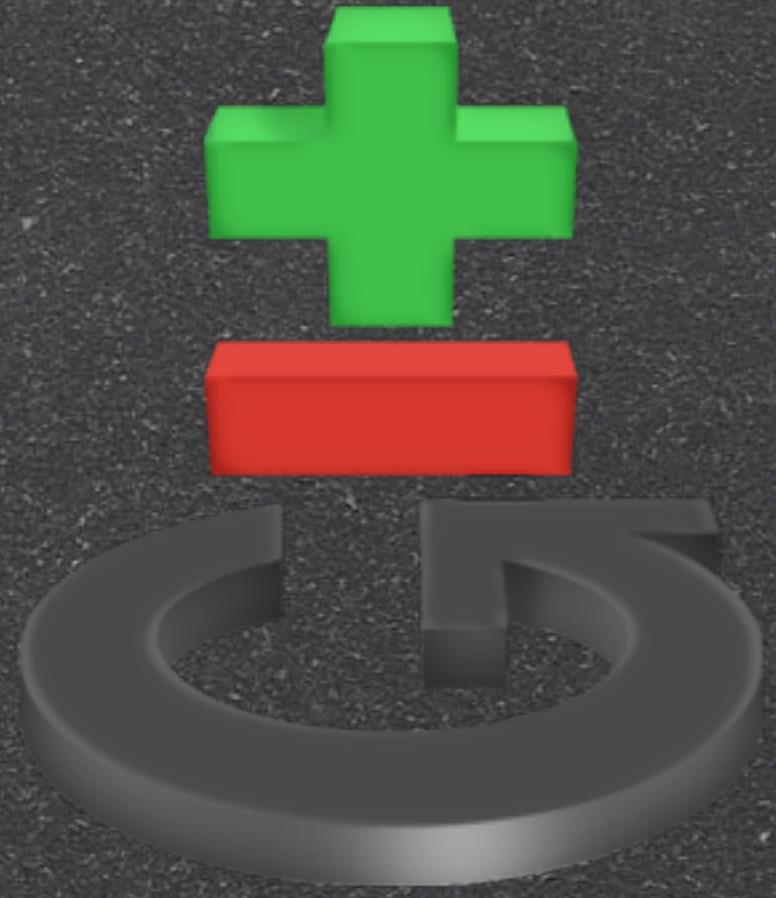


Yhteisö



William Warby / <http://www.flickr.com/photos/wwarby/1584818205/>

- * Kun softaa tehdään avoimen lähdekoodin alustalla, niin tällaiset tarpeet edellyttävät että alustalla on elävä open source -yhteisö joka rakentelee kirjastoja erilaisiin käyttötarkoituksiin.
- * Rubyä ja Railsia ympäröivä open source -yhteisö on erittäin laaja ja aktiivinen.



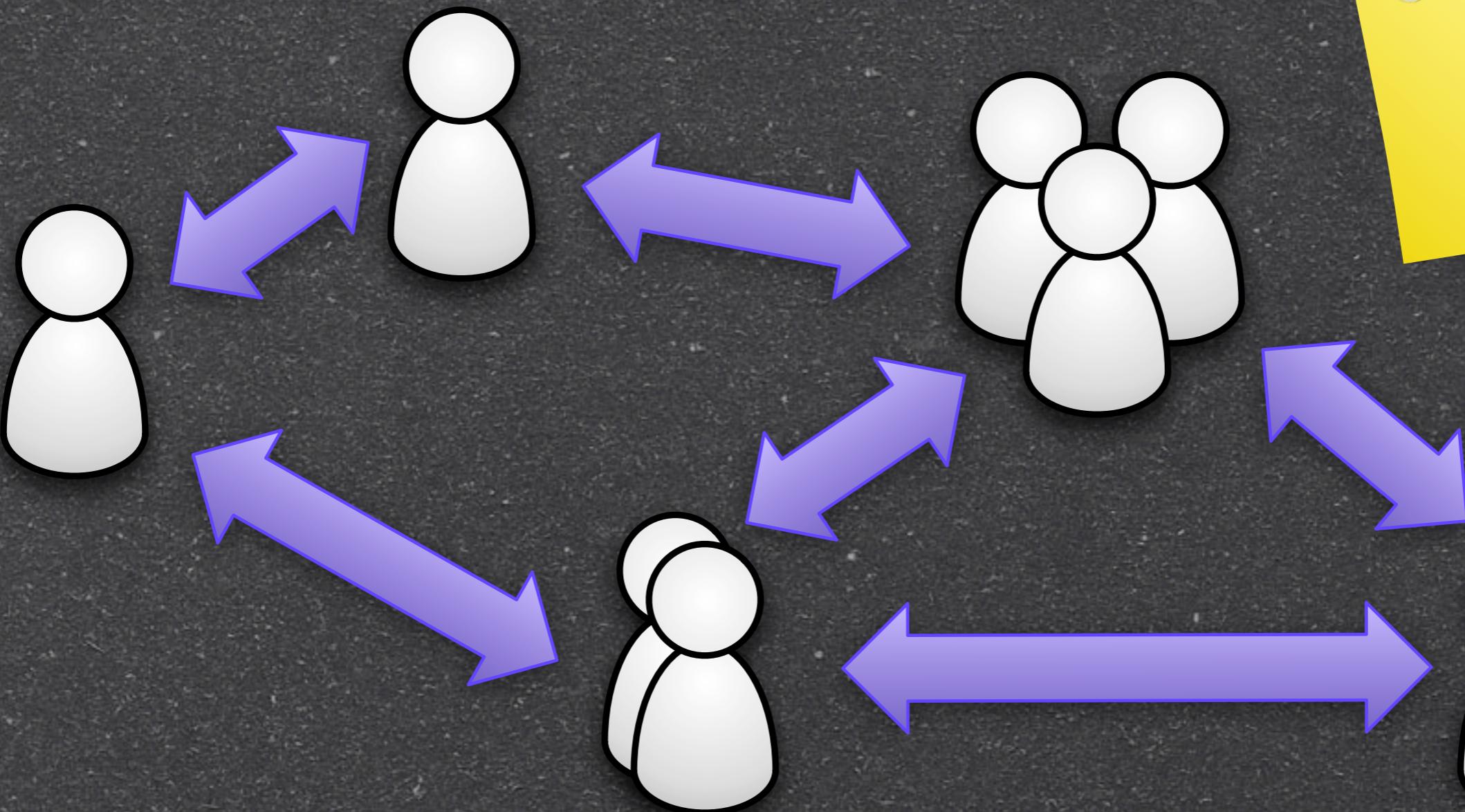
Git



GitHub

"Social
coding"

- * Rubystä ja open sourcesta ei voi puhua mainitsematta Git-versionhallintajärjestelmää ja sen päälle rakennettua GitHub-palvelua.
- * Kysy kuinka monelle tuttuja, ja alusta tarvittaessa:
- ** Git on versionhallintajärjestelmä, tarkemmin ottaen hajautettu sellainen. Linus Torvalds pani alulle joitakin vuosia sitten, koska oli sitä mieltä että kaikki olemassa olevat vaihtoehdot on ihan paskoja, ja tarvitsi kernelin kehitystä varten jotain parempaa.
- ** GitHub on vähän myöhemmin kehitetty kaupallinen palvelu, joka tarjoaa Git-hostausta. Ilmaiseksi open source -kehittäjille.



"Social
coding"

“...possibly the biggest advance in software development technology in ten years.” - Joel Spolsky

* Gitin ja GitHubin merkitystä viime vuosien open source -kehitykselle ei voi vähätellä.

* Laskee avoimen lähdekoodin projektien pystytyksen riman todella alas. Kun haluat laittaa jotain koodia muiden saataville, voit tehdä sen parissa minuutissa. Ei hyväksyntäprosesseja, ei byrokratiaa.

* GitHub tuo lisäksi mukaan sosiaalisen kerroksen, tekee omalla pienellä tavalla samaa kuin sosiaalisen median murros suuremmassa mittakaavassa. Voit seurata muita hakkereita, kommentoida sitä mitä ne tekevät, yms.

Ruby
GitHubissa

104 239 repositoriota

23 123 käyttäjää

<http://www.r-chart.com/2010/08/github-stats-on-programming-languages.html>

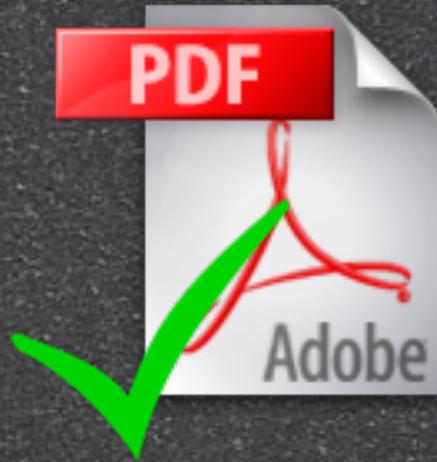
* Ruby-yhteisö siirtyi Gitin sankoin joukoihin joitakin vuosia sitten. Tällä hetkellä ylivoimainen valtaosa avoimesta Ruby-koodista on GitHubissa.

* GitHubissa on tällä hetkellä yli sata tuhatta avointa Ruby-projektiä, joita on ollut tekemässä lähes 25 000 käyttäjää.

* Suuri osa näistä on toki pienimuotoista puuhastelua, mutta joukossa on myös paljon kypsiä ja vakavasti otettavia softia.

* Puuhasteluprojektejakaan ei pidä toki aliarvioida. Kun avointa koodia on näin paljon löydetävissä, käytettävissä ja muokattavissa, löytyy useimpiin ongelmiin ratkaisu. Ongelman täytyy olla aika eksoottinen, jos siihen ei kukaan ole tähän mennessä tehnyt jonkinlaista avointa ratkaisua.

Oheistauhka



PDF:t



Excelit



Graafien
generointi



Kuvan-
käsittely



Moni-
kielisyys



Auth



Mailit



Palvelinten
hallinta

- * Meillä ei siis ollut mitään ongelmia löytää avoimia kirjastoja kaikkiin näihin tarpeisiin.
- * Useimmissa tapauksissa ratkaisuja oli jopa niin monia, että valintoja oli vaikea tehdä.
- * Se, että on valinnanvara, on toki kuitenkin enemmän hyvä kuin huono asia.

* Vaikka kirjastoja olisi kuinka paljon, ei se kuitenkaan tarkoita että niistä voisi tosta vaan rakentaa toimivan softan kuin palapelinä. Tästä päästään Rubyn toiseen selkeään etuun.



Germán Meyer / <http://www.flickr.com/photos/kainet/133456680/>

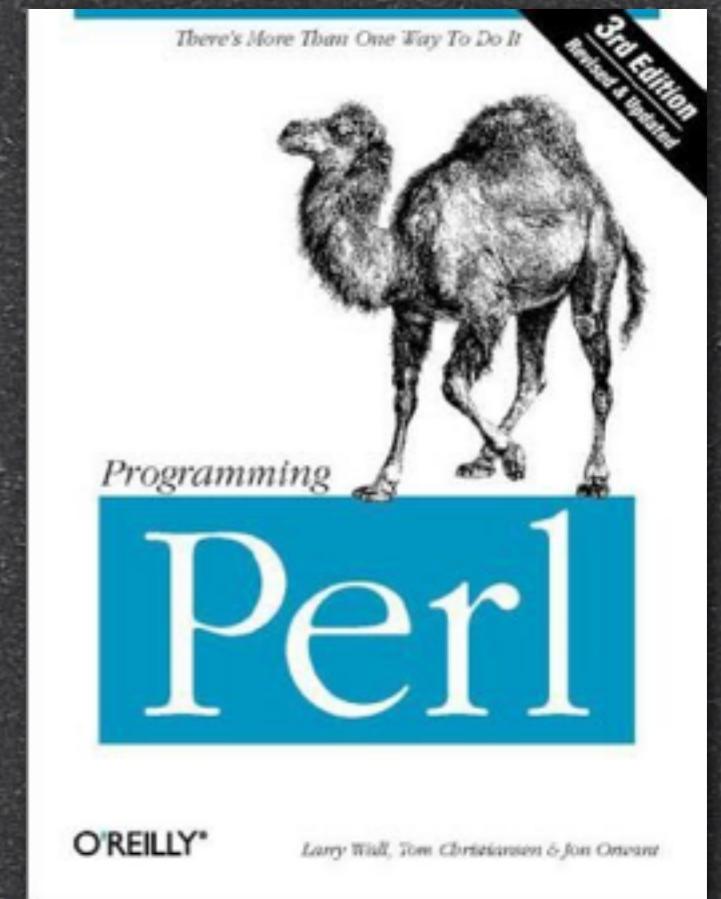
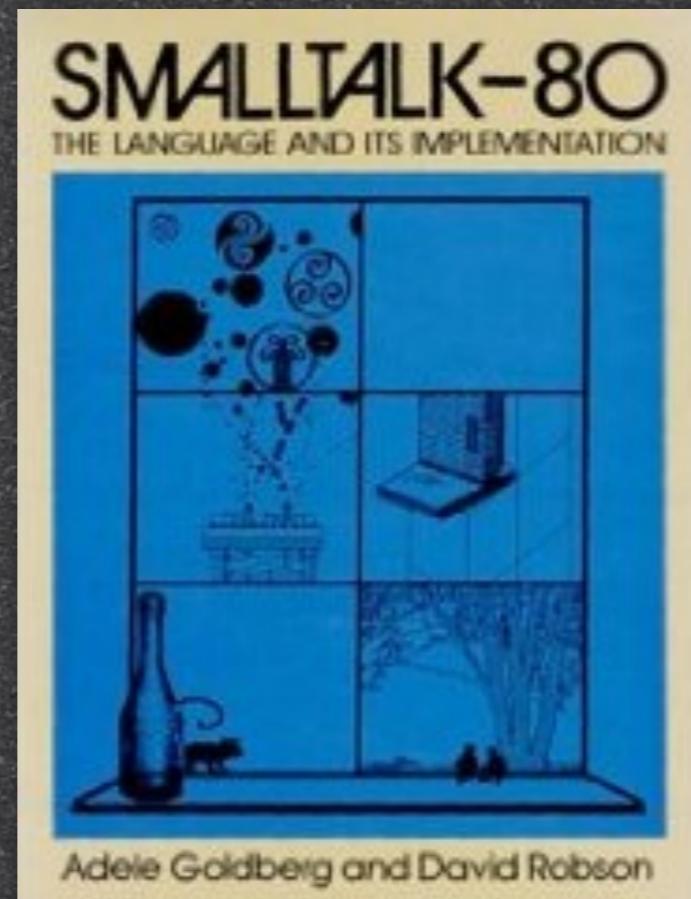
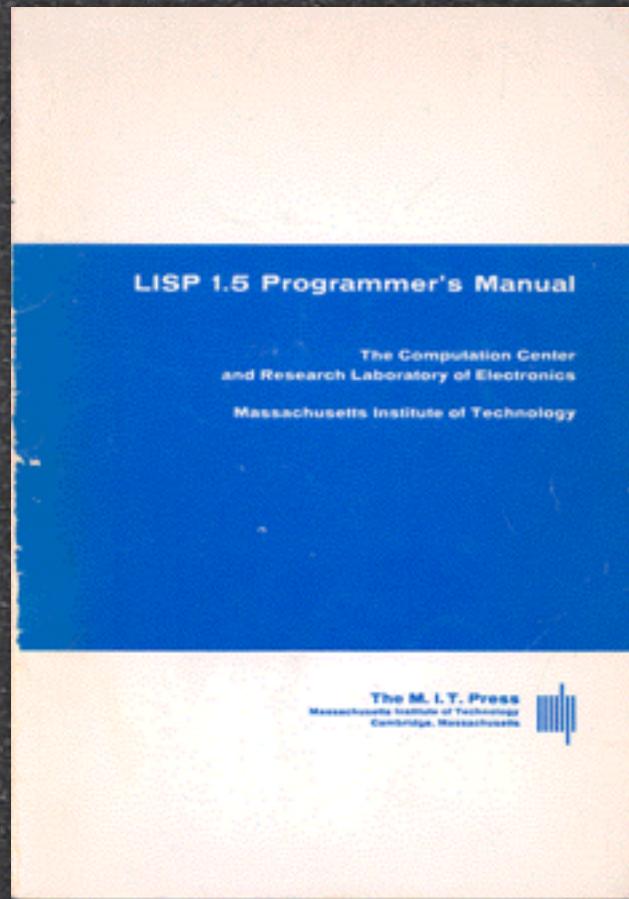
Mukaut-taminen

- * Jotta hyvä softa saadaan rakennettua, se edellyttää että kaikki tämä saatavilla oleva teknologia on **mukautettavissa** omiin tarpeisiin.
- * Aina on sellaisia juttuja, jotka ovat juuri sinun softallesi erityisiä. Usein ne ovat juuri niitä jotka tekevät siitä softasta tekemisen arvoisen.
- * Väitän että Ruby on juuri tällaiseen mukauttamiseen erityisen hyvä alusta. Avaan tätä vähän katsomalla sitä, mistä Ruby on saanut alkunsa.

juuret

idealismi

käytännöllisyys



Lisp
1958

Smalltalk
1980

Perl
1987

- * Rubyllä on kaksi selkeää historiallista vaikuttajaa.
- * Toisaalta on vahvat vaikutteet Lispistä ja Smalltalkista, jotka korostavat sitä että asiat on tehty oikein, kaikki on johdonmukaista, yhdenmukaista, yksinkertaista ja eleganttia, ja jokainen ohjelma on kuin kukkanen.
- * Mutta sitten toisena merkittävänen vaikutteena täysin toiselta laidalta tulee Perl, jossa on tietynlainen "rapatessa roiskuu" -meininki, ajatus siitä että eihän tällä nyt mitään kauneuskilpailuja voiteta, mutta mitä sitten, se toimii.
- * Lisp ja Smalltalk ovat erittäin arvostettuja teknologioita. Perl ei niinkään, mutta sen sijaan sillä on rakennettu moninkertaisesti enemmän ihan oikeaa businessoftaa.



Ruby

(1 000 000 annosta)

Lisp pohjaksi

- * Lambda, sulkeumat
- * Metaohjelointi
- * Dynaaminen tyypitys

Anna kypsyä parikymmentä vuotta.

Smalltalkin oliomalli

- * "Kaikki on olioita"
- * Duck-tyypitys
- * Introspektio
- * Metodikutsu == viesti

Juuri ennen tarjoilua:

Perlın ominaisuudet

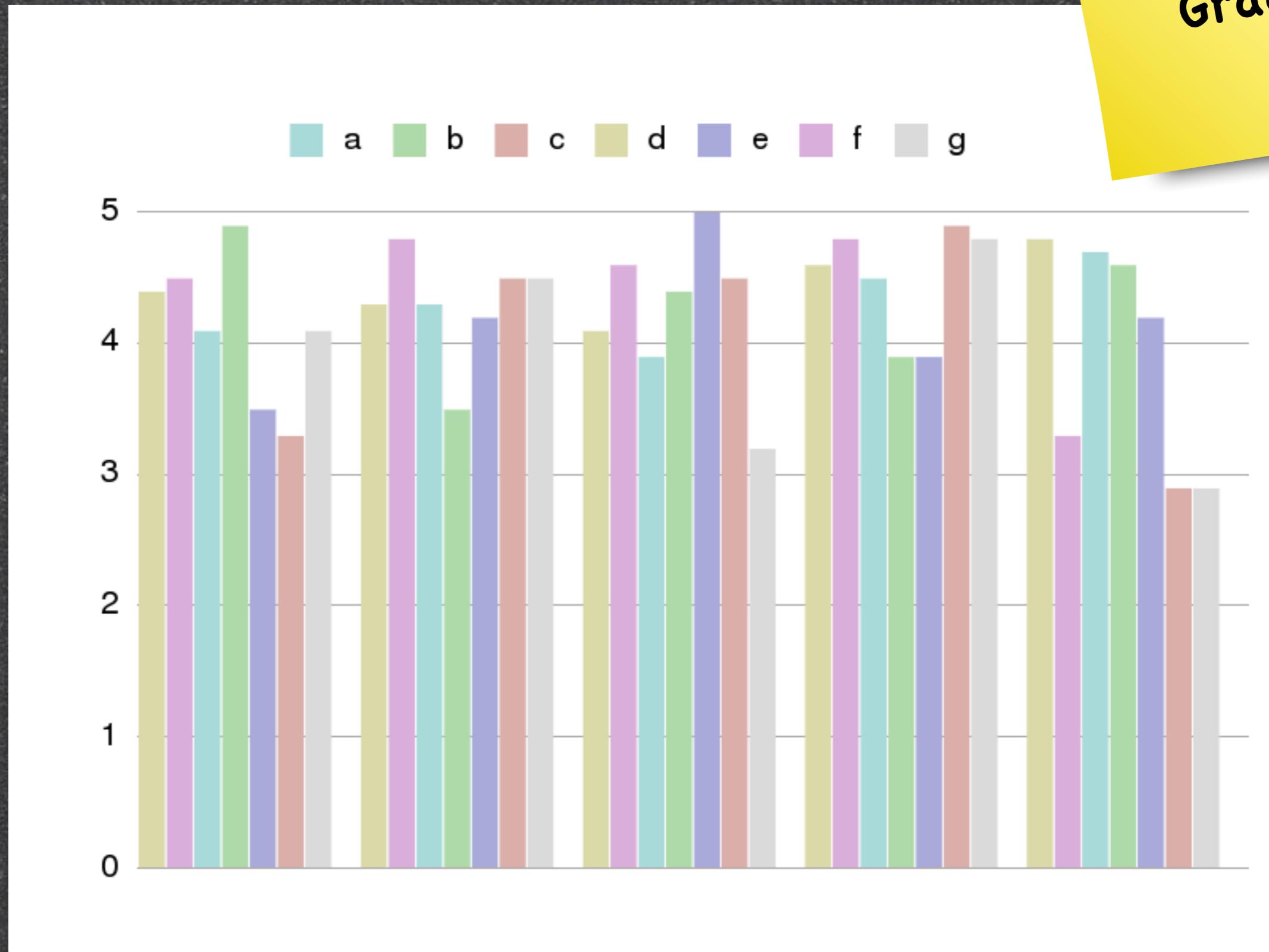
- * Unix-integraatio
- * Merkkijonojen käsittely
- * Regex-literaalit

Sekoita hyvällä maulla ja nauti.

[cipher] / <http://www.flickr.com/photos/h4ck/2981763309/>

- * Tällä reseptillä syntyy melko mielenkiintoinen soppa, jossa toisaalta yhdistyy tietty tekemisen ja aikaansaamisen meininki, ja toisaalta selkärankaa antaa näiden vanhojen arvostettujen kielten johdonmukaisuus ja eleganssi.
- * Syntvä kieli on kuin muovailuvahaa, joka venyy kaikenlaisiin tarkoituksiin, mutta homma ei silti leviä käsiin kun taustalla on ohjelmostimalli joka on oikeasti loppuun asti mietitty.
- * Tämä näkyy monilla tavoin ihan päivittäisessä projektityöskentelyssä.

Graafit

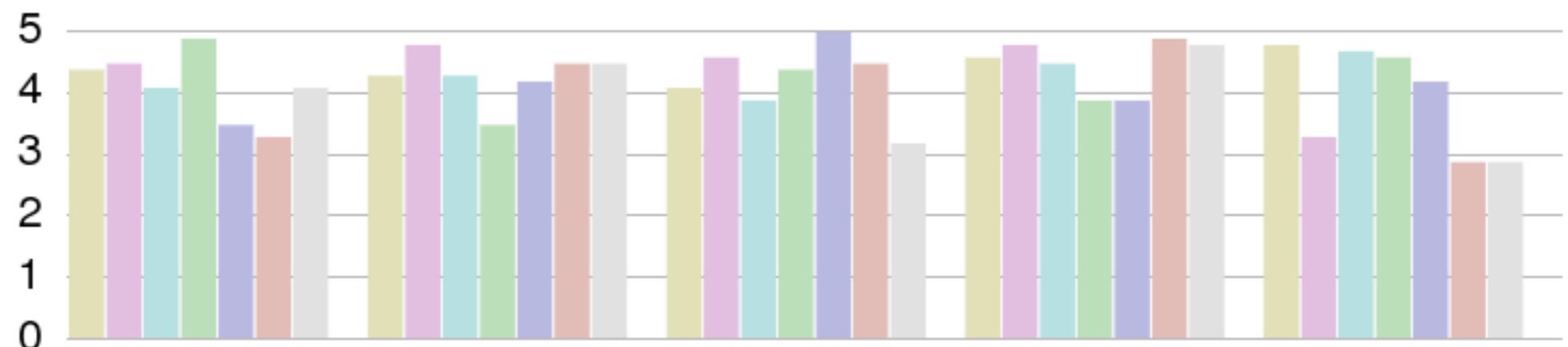


- * Hyvä esimerkki tästä: LaatuVerkossa käytettiin avoimen lähdekoodin kirjastoa graafien generoimiseen.
- * Kirjasto on laadukas, ja täytti suurimman osan niistä tarpeista jotka meillä oli.
- * Vastaan tuli kuitenkin joitain pienempiä ongelmia.



Fail.

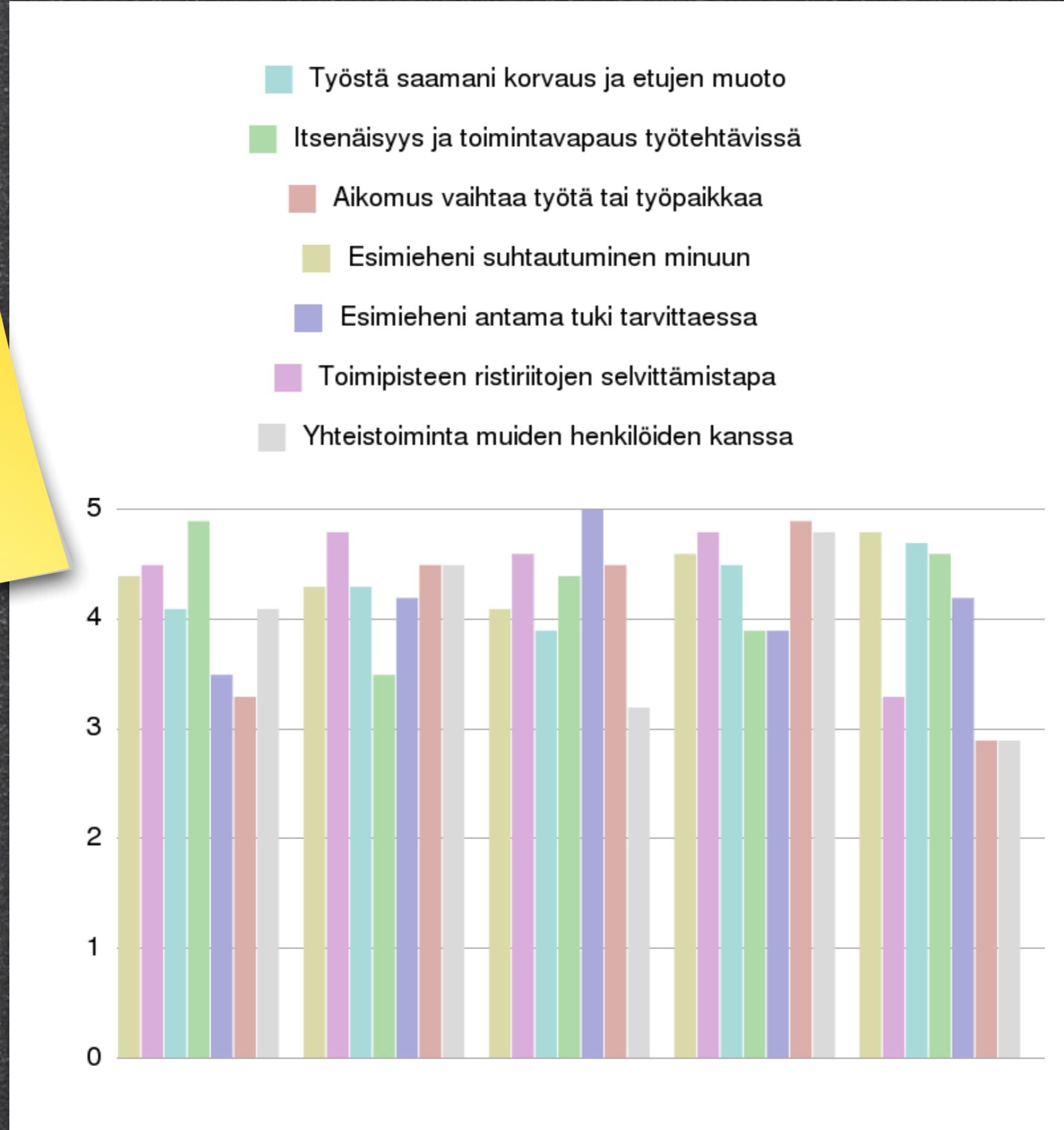
- Työstä saamani korvaus ja etujen muoto
- Itsenäisyys ja toimintavapaus työtehtävissä
- Aikomus vaihtaa työtä tai työpaikkaa
- Esimieheni suhtautuminen minuun
- Esimieheni antama tuki tarvittaessa
- Toimipisteen ristiriitojen selvittämistapa
- Yhteistoiminta muiden henkilöiden kanssa



* Kirjaston tekijä on olettanut että graafien tulisi olla ennalta määritetyn korkuisia riippumatta siitä mitä dataa niissä on. Meidän tapauksessamme graafin selitteessä saattoi olla pitkiä tekstejä, joiden takia itse graafi rutistui pieneen tilaan.

* Tämä on aivan tyypillinen esimerkki siitä, minkälaisia tilanteita tulee vastaan kun valmiita kirjastoja käytetään softaa rakennettaessa.

Viiden rivin korjaus



* Tässä tapauksessa saimme ongelman ratkaistua muutaman rivin muutoksilla, mihin ei todellakaan mennyt kauaa. Esimerkiksi Javan kanssa emme todennäköisesti olisi päässeet näin helpolla.

* Tässä osoittautuu se, miten helppoa Rubyn kanssa on saada asiat toimimaan halutulla tavalla.

Kustomointi



Philip Pessar / <http://www.flickr.com/photos/southbeachcars/3598700737/>

- * Kaikessa tässä on kyse siitä, että voidaan ottaa avoimen lähdekoodin teknologiaa, ja sitten muovata siitä sellainen softaa joka tarvitaan.
- * Se ei siis riitä, että kirjastoja on. Niistä täytyy voida kustomoida se oma tuote.
- * Avointa lähdekoodia on siis paljon, ja ihmiset muovaavat sitä vapaasti omiin tarkoituksiinsa sopivasti. Herää tietyysti kysymys mitä tämä tarkoittaa lisensioinnin ja tekijänoikeuksien kannalta?

MIT
-lisenssi

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

"Tee mitä
lystäät..."

...kunhan et
haasta minua
oikeuteen."

- * Ruby-maailmassa kysymys on ratkaistu valitsemalla sellainen lisenssi joka mahdollistaa tämän kaiken.
- * Ylivoimaisesti yleisimmin käytetty lisenssi on MIT.
- * Kyseessä on hyvin yksinkertainen ja selkeä lisenssi. Se on tässä kokonaisuudessaan. Huomatkaa että sen voi jopa lukea ja ymmärtää.
- * Mahdollistaa sen, että niitä aiemmin kuvattuja teknologian etuja ei blokata. Koodi on muovattavissa myös silloin kun tehdään kaupallista softaa.

Katedraalit ja basaarit



Miemo Penttinen / <http://www.flickr.com/photos/miemo/349424719/>

- * Rubyssä siis kaiken kaikkiaan kiteytyy Raymondin legendaarinen basaarivertaus: Suuri joukko ihmisiä pyörittelee avointa koodia, muuttaa ja laajentaa toistensa koodia, eikä mikään johtaja tai muu taho ohjaa tätä kaikkea ylhäältäpäin.
- * Lisäksi teknologia on sellaista että se tukee tai jopa vahvistaa tämäntyyppistä kehitystä.
- * Raymond käytti alkuperäisessä basaarivertauksessa esimerkkinä Linuxin kehitystä. Ei ole varmaan sattumaa, että sieltä kotoisin oleva Git otettiin niin vahasti käyttöön nimenomaan Ruby-yhteisössä. Ruby-yhteisö toimii pitkälti samalla tavalla kuin kernel-yhteisö.



@teropa

twitter

* LaatuVerkossa, kuten monissa muissa Eficodella ja muualla tehdyissä Ruby-projekteissa on osoitettu, että aivan kuten Linuxilla, myös Ruby-alustalla voidaan tehdä bisnestä.

* Ne samat seikat jotka tekevät näistä vahvoja avoimia alustoja, edistävät myös laadukkaan kaupallisen softan rakentamista. Koodia on paljon, ja se on pääosin laadukasta. Yhteisö on vahva. Kaikki on muokattavissa itselle sopivaksi.