

Miten avoimen lähdekoodin kehitys ja ketterät menetelmät kohtaavat?

Jarl-Erik Malmström

Referaatti
Helsingin Yliopisto
Tietojenkäsittelytieteen laitos

Helsinki, 1. helmikuuta 2013

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Jarl-Erik Malmström			
Työn nimi — Arbetets titel — Title			
Miten avoimen lähdekoodin kehitys ja ketterät menetelmät kohtaavat?			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Referaatti	1. helmikuuta 2013	5	
Tiivistelmä — Referat — Abstract			
Tiivistelmä.			
Avainsanat — Nyckelord — Keywords			
agile, ketterä, open source, avoin			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Sisältö

1	Johdanto	3
2	Avoimen lähdekoodin ketterät menetelmät	3
2.1	yksilöt ja vuorovaikutus	3
2.2	Toimiva ohjelmisto	4
2.3	Asiakasyhteistyö	4
2.4	Muutoksiin sopeutuminen	4
3	Mitä ketterät menetelmät voisivat oppia avoimista projek- teista?	4
3.1	Tallettajan rooli	4
3.2	Tulokset julki	5
4	Lähteet	5

1 Johdanto

Avoimen lähdekoodin (open source software) kehittäjäyhteisö on tyypillisesti maantieteellisesti hajaantunut. Yhteisön jäsenten etäisyys vaikeuttaa jäsenten välistä kommunikaatiota. Verrattuna perinteiseen työyhteisöön, minkä jäsenet työskentelevät yleensä samassa tilassa.

Ketterät ohjelmistotuotantomenetelmät (agile methods) eivät vaikuta sopivilta hierarkiseen ja hajaantuneeseen organisaatorakenteeseen. Mutta avoimen lähdekoodin projektit toteuttavat usein ketterien menetelmien periaatteita. Projektit ovat valmiita muutoksille, vastaavat jatkuvaan palautteeseen toimittamalla uusia toiminnallisuuksia, arvostavat yhteistyötä ja kohtaavat haasteita.

Linux käyttöjärjestelmän levinneisyys ja ohjelmistoalan muutos kohti ketteriä menetelmiä ovat osoittaneet, että avoimilla projekteilla ja ketterillä menetelmillä molemmilla on etunsa. kumpikaan ei ole tarjonnut yksiselitteistä ratkaisua ohjelmistotuotannon ongelmiin. Referoidussa artikkelissa[1] Hugo Corbucci ja Alfredo Goldman pyrkimys yhdistämään molempien menetelmien edut.

2 Avoimen lähdekoodin ketterät menetelmät

Miten ketterät menetelmät toteutuvat avoimen lähdekoodin projekteissa?

Hugo Corbucci ja Alfredo Goldman esittävät, että avoimen lähdekoodin projekteissa toteutetaan ketteryyttä ja monien avoimen kehityksen ominaisuuksia voidaan suoraan liittää ketterään julistukseen (agile manifesto).

He pitävät ketteränä sellaisia ohjelmistosuunnittelun metodeja, mitkä seuraavat ketterän julistuksen periaatteita: yksilöt ja vuorovaikutus yli prosessien ja työkalujen, toimiva ohjelmisto yli kattavan dokumentaation, asiakasyhteistyö yli sopimusneuvottelun ja muutoksiin sopeutuminen yli suunnitelman seuraamisen.

2.1 yksilöt ja vuorovaikutus

Kirjoittajat väittävät avoimen lähdekoodin projektien olevan lähes poikkeuksetta ketteriä. He kirjoittavat että useissa tutkimuksissa on havaittu avoimen lähdekoodin ohjelmistokehityksessä käytettävän kohtuullisesti työkaluja yhteisön kommunikaation ylläpitämiseksi. Versionhallinta, projektin kotisivu ja postituslistat olivat pääasialliset yhteydenpitovälineet käyttäjien ja kehittäjien keskuudessa. Yhteisön yksilöt ja sen vuorovaikutus kiinnostavan asian ympärillä ovat koko avoimen kehityksen ydin. Yhteistoiminta liittyy useimmiten lähdekoodin ympärille ja dokumentaation laadintaan. Yhteistoiminta vie prosesseja eteenpäin ja muokkaa yhä paremmin yhteisön tarpeisiin sopiviksi.

2.2 Toimiva ohjelmisto

Hieman yli puolet avoimen lähdekoodin projekteista päivittävät ja tarkistavat dokumentaatiota usein ja kolmannes ylläpitää dokumentteja kertoakseen kuinka ohjelmiston osat toimivat ja kuinka se on organisoitu. Kirjoittajat näkevät dokumentoinnin tukevan projekteja eikä olevan projektien pääasiallinen tarkoitus.

2.3 Asiakasyhteistyö

Asiakassopimukset ovat harvoin ongelma avoimissa projekteissa. Asiakas palkkaa kehittäjän tai organisaation kehittämään toiminnallisuuden. Liiketoimintamalli ei takaa yhteistyötä asiakkaan kanssa, mutta yhteistoiminta on kuitenkin koko avoimen kehityksen perusta. Asiakas on yhteistyössä toiveidensa mukaan.

2.4 Muutoksiin sopeutuminen

Avoimen lähdekoodin projekteilla on yleensä virstanpylväitä (milestone) ja julkaisuja (releases). Monilla projekteilla on vain lyhyen ajan suunnitelmia. Jos pitkän ajan suunnitelmia on, ne ovat tavoitteita ilman painetta niiden toteuttamiseen. Hugo Corbucci ja Alfredo Goldman kirjoittavat sopeutumisen olevan välttämätöntä kilpailuhenkisessä ympäristössä. Projektien on vastattava käyttäjien muuttuviin tarpeisiin.

3 Mitä ketterät menetelmät voisivat oppia avoimista projekteista?

3.1 Tallettajan rooli

Monilla avoimen lähdekoodin projekteilla on pieni määrä henkilöitä, joilla on oikeus lisätä koodia runkohaaraan (trunk branch). Runkohaara muodostaa uuden julkaistavan ohjelmiston. Yhteisö luottaa tallettajien (committer) koodiarviointiin. Ketterässä ohjelmistokehityksessä kaikilla kehittäjillä on oikeus lisätä koodia runkohaaraan.

Hugo Corbucci ja Alfredo Goldman esittävät valvontaa tuotantokoodin yksinkertaisuuden takaamiseksi. Useimmilla ketterillä menetelmillä on kehittäjätiimeissä johtaja tai valmentaja, joka on yleensä muita jäseniä kokeilempi. Valvontatehtävä voisi olla tiimin johtajalla ellei se aiheuta viivästystä tuotannossa tai häiritse hänen muita tehtäviään. Toinen vaihtoehto olisi vuorotella valvontavastuuta osalla kehittäjiä.

3.2 Tulokset julki

Kirjoittajien mielestä suljetun lähdekoodin ohjelmistot voisivat hyötyä julkisista vianseurantajärjestelmistä (bug tracking) ja testituloksista. Ohjelmistojen tuottajien olisi hyväksyttävä siten koodia julkaistavaksi. Julkiset työkalut kannustaisivat käyttäjiä osallistumaan tuotantoprosessiin.

Ketterässä kehityksessä vianseuranta ja testit ovat tärkeitä työkaluja. Ketterät menetelmät eivät sano, että asiakkaan tulisi käyttää näitä työkaluja. Kuitenkin useimmissa menetelmissä asiakasta pitäisi pitää osana kehitystyötä.

4 Lähteet

- [1] Corbucci, Hugo ja Alfredo Goldman: *Open Source and Agile: Two worlds that should have a closer interaction*. Teoksessa *ESELAW - V Experimental Software Engineering Latin American Workshop*, Salvador, Brazil, 2008.