



Tarmo Toikkanen (PsM) toimii tutkijana ja ohjelmistokehittäjänä Taideteollisen korkeakoulun Medialaboratoriossa, Oppimisympäristöjen tutkimusryhmässä
Lisätietoja:
<http://tarmo.fi>
<http://lemill.org>
<http://legroup.uiah.fi>

XP2006: Ketterät menetelmät hajautetuissa kehitysympäristöissä

XP – eXtreme Programming. Vuosituhannen alussa se oli käytännössä ainoa laajalti tunnettu radikaali ketterä menetelmä ohjelmistojen tekoon. Nykyisin menetelmiä on tusinoittain, mutta XP:n mukaan nimetty vuosittainen konferenssi pitää vanhasta nimestään kiinni. Tänä vuonna XP2006 toi alan gurut Ouluun.

Hajautetut projektit

Merkittävä teema XP2006:ssa olivat jaetut ja hajautetut projektit ja niiden hallinta. Itsekin pidin tästä teemasta esityksen, jonka tässä referoin.

Kun projektiin kuuluvat tiimit eivät sijaitse samassa paikassa, puhutaan jaetusta (distributed) projektista. Tyypillisessä skenaariossa yksi tai useampia tiimejä on sijoitettu halpatuotantomaihin pääarkkitehtien pysyessä yhtiön kotimaassa.

Pääongelmia ovat:

projektin tilan näkyvyys heikkenee, projektin aika- ja kustannusarviot huononevat, integrointi ja kokonaisuuden hallinta vaikeutuvat, ja viestintäkanavat heikkenevät.

Kustannusten alentaminen ajaa yhä useamman yrityksen käyttämään ulkomaisia tai alihankintatiimejä, joten jaetun projektin haasteet ovat varsin yleisiä.

Etenkin pienemmissä projekteissa ja PK-yrityksissä yleisiä ovat myös hajautetut (dispersed) projektit, joissa tiimin jäsenet voivat sijaita eri puolilla maailmaa. Usein yksi tiimin jäsen työskentelee Suomessa toisessa kaupungissa muiden ollessa PK-seudulla, tai tiimiin on hankittu ulkomainen erikoisasiantuntija joka pysyy omassa maassaan. Hajautetun projektin haasteet ovat samat kuin jaetunkin, lisättynä yhdellä vakavalla ongelmalla; tiimidynamiikka kärsii.

Ketterien menetelmien pääpiirteet, verrattuna perinteisiin "raskaisiin" menetelmiin, auttavat yllä mainituissa ongelmissa oheisen kuvan mukaan.

Ketterien menetelmien pääpiirteet ja ongelmien ratkaisut

Tiheät toimitukset:

Kun uusi versio julkaistaan 2-4 viikon välein, projektin näkyvyys ei voi heiketä merkittävästi.

Todellisuus dokumenttien sijaan:

Koska työssä käsitellään itse toimivaa ohjelmaa eikä vain speksejä, sekä projektin näkyvyys että arvioiden pitävyys paranevat.

TDD (test-driven development):

Automaattiset testit auttavat laadun valvonnan, asiakkaan tilaamien ominaisuuksien toiminnan varmentamisessa sekä tuotteen integroinnissa.

Toimintatapojen arviointi ja parannus:

Joka julkaisun yhteydessä työn tekijöiden itse tekemä toimintatapojen arviointi ja parannus nostavat arvioiden laatua ja parantavat näkyvyyttä, sekä tietysti nostavat tuottavuutta.

Tiheät arviot:

Joka julkaisun yhteydessä tehty arvio asiakkaan kanssa parantaa projektin näkyvyyttä ja vähentää kokonaisriskiä.

Jatkuva integrointi:

Poistaa integrointiongelmat käytännössä täysin.

Toisaalta ketterät menetelmät itsekin kärsivät hajautuksesta, pääosin näin; eri paikoissa sijaitsevat tiimit sekä asiakkaan edustaja vaikeuttavat pariohjelmointia, asteittaista määrityksen tarkentamista ja asiakasedustajan täyttää hyödyntämistä. Eri paikoissa sijaitsevat tiimien jäsenet vaikeuttavat pariohjelmointia, heikentävät luottamusta, vaikeuttavat flow'n muodostumista ja hidastavat tiimin itseorganisoitumista.

Yhteenvetona voisi todeta, että ketterien menetelmien peruspiirteet itse asiassa korjaavat monia jaetun ja hajautetun kehityksen vakavimpia ongelmia, eli ne tältä osin soveltuvat raskaita BDUF-menetelmiä paremmin hajautettuihin skenaarioihin. Toisaalta taas ketterät menetelmät luottavat raskaita enemmän ihmisten väliseen kommunikaatioon, jota taas hajautus vaikeuttaa huomattavasti. Monia viestintäongelmia voidaan kuitenkin korjata nykyisillä internetiä hyödyntävillä viestintävälineillä.

Viestinnän tehostaminen hajautetussa kehityksessä

Kuinka sitten parantaa viestintää, kun tiimin jäsenet sijaitsevat pitkin maailmaa? Tässä oma esimerkkini: toimin projektin "scrum masterina", eli eräänlaisena fasilitaattorina. Tehtäväni on siis pitää projekti rullaamassa ja poistaa ongelmat kehittäjien tieltä.

Projekti on 3-vuotinen EU-rahoitteinen R&D-projekti, jossa kehitetään EU-maiden opettajille oppimateriaalin jakelu-, tuotanto- ja hakupalvelua. Kehittämämme LeMill-palvelinohjelmisto on vapaata ja avointa koodia. Kehittäjät sijaitsevat neljässä maassa eivätkä ole ennen tavanneet toisiaan.

Ensimmäinen ja tärkein vaihe on saada kaikki kehittäjät yhteen paikkaan: kasvotusten tapaaminen ja ajan viettäminen on olennaisen tärkeää, eikä sitä mikään videokonferenssi voi korvata. Optimaalinen alussa yhdessä vietettävä aika olisi 2-3 kuukautta, mutta jo viikko on parempi kuin ei mitään. Yhdessäolon aikana tutustutaan, opetellaan toimimaan yhdessä ja koetaan jotain mielenkiintoista (kuten saunominen Suomessa). Tavoitteena on saada luotua jotain yhteistä tiimin jäsenten välille.

Etävaiheen alettua keskeisessä asemassa on "information radiator" eli tiedonvälittäjä. Ilman hajautusta tämä tyypillisesti olisi yhteisen työtilan seinät, mutta hajautetussa tilanteessa tiedonvälittäjä on netissä. Me päädyimme Trac-nimiseen avoimeen ohjelmistoon, joka yhdistää wikin, versionhallinnan (Subversion) ja tikettijärjestelmän yhteen kokonaisuuteen. Tähän järjestelmään tallennamme kaikki suunnitteludokumentit, user storyt, tehtävät, virheraportit, kokousten pöytäkirjat, aivan kaiken. Ja kaikki on kaikkien luettavissa,

ei mitään hierarkkisia käyttörajoituksia. Vaarallista hajautuksessa on, että joku putoaa kärryiltä, joten kaiken tiedon on oltava netissä saatavilla – käytäväkeskusteluihin ei voi enää luottaa.

Eräs Tracin vahvuus on sen yksinkertaisuus ja joustavuus – wikiä voi käyttää monella eri tavalla. Olemme joutuneet itse opettelemaan toimivat käytännöt ja ne on tallennettu wikiin ohjeiksi. Tämä on koettu selvästi paremmaksi vaihtoehdoksi kuin jäykkä ja kontrolloitu projekinhallintajärjestelmä, sillä tämäkin projekti on vajaan vuoden aikana muuttunut, kasvanut ja kehittynyt niin paljon, että tiukempi järjestelmä olisi moneen kertaan rajoittanut, miten asioita voidaan tehdä.

Pelkkä pull-tekniikka ei kuitenkaan riitä yhteydenpitoon – kaikkien kehittäjien postilaatikkoon kolisee jatkuvasti sähköposteja Trac-järjestelmästä: aina kun joku lähettää lähdekoodia versionhallintaan tai mihin tahansa tikettiin tehdään muutoksia, kerrotaan tästä kehittäjien postituslistalla. Toista postituslistaa käytetään kaikkien asiomaisten väliseen kommunikointiin.

Lisäksi kehittäjät aina töissä ollessaan liittyvät yhteiselle IRC-kanavalle, jossa he voivat selvittää kulloinkin ajankohtaisia ongelmia. Kanavalla on myös botti, joka kertoo Tracissa tapahtuneista muutoksista. Muutokset ovat saatavilla myös RSS-feedinä.

Työn organisoinnin suhteen suuria ongelmia oli alussa siinä, että yksikään kehittäjä ei tee pelkäänsä tätä projektia. Hyväksi ratkaisuksi havaitsimme viikon mittaiset keskitetyt kehityssprintit, jolloin kaikki kehittäjät tekevät vain tätä projektia. Toinen toimiva ratkaisu on wikiin tehty viikkokalenteri, johon kehittäjät merkitsevät, milloin ovat töissä projektin parissa.

Työn suunnittelua tehdään postituslistan lisäksi myös audio- ja videokonferensseissa. Skype ja FlashMeeting ovat projektissa yleisimmin käytetyt välineitä. Ja tietysti niitä pakollisia kasvotusten tapaamisiakin on, mutta vain muutamia vuodessa.

Kaikki käyttämämme ohjelmistot ovat vapaita tai ainakin ilmaisia ohjelmistoja. Niiden avulla monet hajautetun työskentelyn ongelmat on ainakin lievennetty niin, että toistaiseksi (10 kuukautta projektin alkamisesta) olemme aikataulussa. Tuottavuus on kuitenkin ehdottomasti alle sen, mitä yhtenäisellä tiimillä voitaisiin saada, eli hajautetusta tiimistä ei ainakaan näillä lääkkeillä saa samaa tehoa irti kuin motivoituneesta, samassa tilassa toisiaan tukevasta tiimistä.

Seuraavalla sivulla XP lyhenteitä ja terminologiaa.

XP lyhenteitä ja terminologiaa (T. Toikkanen)

BDUF: Big Documentation Up-Front, eli dokumentit edellä etenevä projekti. Projektille tyypillistä on kattava dokumentointi ja koodin ilmaantuminen vasta verrattain myöhään projektissa. Vertaa TDD.

FlashMeeting: Tutkimustyöhön tarkoitettu videokonferenssi-palvelu, joka vaatii ainoastaan Flashia tukevan selaimen, kaiuttimet, mikrofonin ja valinnaisesti web-kameran. www.flashmeeting.com. Monia vastaavia palveluita on olemassa vaihtelevin hinnoitteluin.

IRC: Suomessa kehitetty jutustelualusta Internet Relay Chat, jota etenkin ohjelmistokehityspiireissä suositaan. IRCin kanaville voidaan luoda botteja, jotka valvovat kanavaa, poistavat spämmääjät, antavat lisätietoa kanavaan liittyvistä aiheista, jne.

RSS: Uutisvirtojen välittämiseen käytetty xml-formaatti. Modernit selaimet osaavat käsitellä rss-feedejä suoraan (esim. Firefoxin "Live Bookmarks").

Scrum master: SCRUM-projektinhallintamenetelmään kuuluva rooli, joka on sekä hallinnon että kehittäjien ulkopuolella. Scrum Masterin tehtäviin kuuluu organisatoristen esteiden poistaminen kehitystyön tieltä sekä kehittäjien tukeminen SCRUMin sääntöjen noudattamisessa. <http://scrumalliance.org>

Skype: Suosittu ilmainen internet-puhelujärjestelmä, jolla voi soittaa myös tavallisiin puhelimiin. <http://www.skype.com>

TDD: Test-Driven Development, eli testit edellä etenevä projekti. Tyypillisesti TDD-projektissa vaatimusmäärittely kuvataan automaattisesti suoritettavina testeinä, joita lähdetään toteuttamaan. Minkä tahansa ominaisuuden toteutus aloitetaan tekeillä yksikkötestit, ja kun testit menevät läpi, on ominaisuus valmis. Dokumentaatiota syntyy verrattain vähän, ja tyypillisesti vasta projektin lopuksi – aiemmin tuotettu dokumentaatio kun vanhenee välittömästi.

Tikettijärjestelmä: Järjestelmä, johon kaikki tehtävät kirjataan, arvioidaan, tarkennetaan, annetaan tehtäväksi ja kuitataan tehdyiksi. Muita termejä: issue management, trouble ticket system.

User story: Joissain ketterissä menetelmissä käytetty vaatimusten dokumentointitapa. User storyssa kuvaillaan asiakkaan näkökulmasta, asiakkaan ymmärtämällä kielellä, miten tietty toiminto tapahtuu. Asiakkaan on täten mahdollista ymmärtää ja varmistaa, että ohjelma todella toimii niin.

Wiki: Yhteisöllinen sivusto, jossa kaikki jäsenet voivat tasavertaisesti ja helposti muokata kaikkia sivuja. Toisin kuin useimmat muut internetin viestintävälineet, Wiki kannustaa yhteisymmärryksen muodostamiseen.

Teksti: Pekka Abrahamsson
Outi Salo
Kuva: XP2006-tapahtuman
kuva-arkisto

*Pekka Abrahamsson
tutkimusprofessori,
XP2006 -ohjelmatoimikunnan
päällikkö
VTT*

*Outi Salo
tutkija,
XP2006 local chair
VTT*

Ketteriin menetelmiin keskittyvä XP2006-konferenssi oli menestys!

Ketterät (engl. agile) menetelmät ovat olleet ohjelmistoteollisuuden saatavilla 1990-luvun lopulta lähtien. Kirjallisuutta aihepiiristä on ilmestynyt ripeämmällä tahdilla kuin mistään muusta ohjelmistotalan innovaatiosta pitkään aikaan. Suomessakin on ketteryyteen liittyvistä tutkimustuloksista ja yritysten saavutuksista uutisoitu viime aikoina.

Ketteriin menetelmiin on ladattu paljon odotusarvoa, kuten ohjelmistotuotannon kustannusten aleneminen, ohjelmistovirheiden määrän väheneminen ja tuotantosyklin nopeutuminen sekä nopea reagointi asiakkaalta tuleviin muutostarpeisiin. Perustana on ajattelutavan muutos, jossa painopiste on eri sidosryhmien kommunikaatiossa ja yhteistyössä sekä muutosherkkydessä.

Oulussa järjestettiin kesäkuussa 2006 ketteriin menetelmiin ja niiden soveltamiseen keskittyvä konferenssi XP2006, jossa suomalainen ohjelmistoteollisuus oli kattavasti läsnä. Noin 300 konferenssiin osallistujasta joka toinen oli suomalainen.

Osallistujia saapui 25 maasta ja 85 % heistä oli teollisuuden edustajia. Yrityksiä paikalla oli yli 100. Tämä sinänsä on jo poikkeuksellista, koska konferenssi keskittyy menetelmiin ja prosesseihin, eikä niinkään työkaluihin tai kehitysteknologiaan.