

APPELSIINI

www.appelsiini.fi

APPELSIINI

Kehittämisen menetelmät

Miksi Extreme Programming?

2.5.2007

Ryhmä: Marko Keinänen, Jari Kauppi, Jarmo
Levo, Jan Partanen ja Jouni Tikkanen

- Tutkimusongelmaksi ryhmätyössämme valittiin ketterät menetelmät, joista kysymykseksi tarkentui "Miksi käyttää XP:tä perinteisten kehitysmenetelmien sijaan?"
- Lähdimme liikkeelle siitä, että halusimme selvittää
 - mistä XP:ssä on kyse?
 - mitkä on sen hyödyt?
 - miten se vaikuttaa kuluihin?
 - mikä on asiakkaan rooli ja työpanos?
 - miten laatu asiat on otettu XP:ssä huomioon?
 - miten projektin johtaminen eroaa perinteisistä menetelmistä?
 - muuttuvatko vaatimukset projektin aikana?

- Selvityksen aihealue on melko laaja ja jo alkuvaiheessa havaitsimme, että tutkimusongelma ja sen piiristä valitut teesit on rajattava heti alkuvaiheessa tarkasti. Työkaluina hyödynnettiin henkilökohtaisia tapaamisia, Fle3-tiedonrakentelu ympäristöä ja sähköpostia.
- Valittavissa olevat teesit olivat:
 1. Suunnittelu
 2. Osatoimituksia
 3. Yhteinen metafora
 4. Yksinkertainen ratkaisu
 5. Testaus
 6. Ennakoiva korjaaminen
 7. Pariohjelmointi
 8. Yhteinen omistajuus
 9. Jatkuva integraatio
 10. 40-tuntinen työviikko
 11. Saatavilla oleva asiakas
 12. Koodausstandardi

- Ryhmän jäsenistä Jarmolla on omakohtaista kokemusta Extreme Programmingista ja hän pystyikin nopeasti sekä keskustelussa että viesteinä tiedonrakentelussa tuomaan esiin XP:n pääperiaatteita. Myös kirjallisista lähteistä saimme esiin paljon erilaisia näkökulmia siihen mistä XP:ssä on kyse. Osa esille nousseista periaatteista, esimerkiksi pariohjelmointi tai yhteinen omistajuus, herätti paljon keskustelua ryhmässä puolesta ja vastaan.
- Epäilijät joutuivat myöntämään, että XP:stä löytyi myös helposti hyväksyttäviä piirteitä, esimerkiksi käyttäjätarinoiden hyödyllisyys ymmärrettiin. Samoin joustavuuden ja nopeuden katsottiin kuuluva XP:n hyviin puoliin melko ilmiselvinä. Alkuperäisessä aiheen valintaan liittyneessä keskustelussa oli puhuttu jonkin verran myös laadusta ja meitä askarrutti XP:n ja laadun suhde.

- Teesiäänestyksessä voittajiksi selvisivät aihealueet 1, 7 ja 11:
 1. Suunnittelu
 2. pariohjelmointi ja
 3. saatavilla oleva asiakas.
- Tutkimusongelmaa avattiin kolmen iteraatiokierroksen avulla, joissa kussakin aiheeseen syvennettiin ennalta tehtyjen selvitysten perusteella ja joihin muut ryhmät esittivät omat kriittisen arvionsa.

- Tämän teesin osalta keskustelu keskittyi pääosin käyttäjätarinoihin. Niitä arvioitiin hyvin tutkimusongelmaa vasten: "mitkä ovat xp:n hyödyt."
- Selkeästi nähtiin, että käyttäjätarinoiden kautta tehty ohjelmiston määrittely ja toteutus tuo hyötyä muuttuvien vaatimusten ympäristössä. Myöskin tilanteessa, jossa tilaajalla ei aluksi ole selkeää käsitystä siitä mitä projektin pitäisi tuottaa, käyttäjätarinat auttavat hyvin luomaan tämän käsityksen ja kartoittamaan reaali maailman toimintoja.

- Käyttäjätarinoiden vahvuutena pidettiin myös ymmärrettävää kieltä: käyttäjätarinan kertoja voi omin sanoin kuvata toteutettavia vaatimuksia (ilman formaalia määrittelykieltä). Lisäksi tilaajan kertomien käyttäjätarinoiden katsottiin olevan sitä alkulähteiltä saatavaa "oikeaa tietoa", jota kehittäjät nimenomaan tarvitsevat oikean tuloksen aikaansaamiseksi.
- Tällä kierroksella löytyi täsmennystä siihen, millainen hyvä käyttäjätarina on (Mike Cohnin mukaan):
 - riippumaton
 - sisältö on aina täsmennettävissä
 - hankkijalle tai käyttäjälle arvokas
 - ennustettavissa
 - pieni

- Pariohjelmointi herätti hiukan vähemmän keskustelua kuin käyttäjätarinat. Historiatietona saatiin selville, että pariohjelmointia on tehty jo 70-luvulla ja sitä on jo silloin pidetty laadullisesti hyvänä tapana tuottaa koodia. Tähän ei välttämättä kaikkialla ole ollut mahdollisuutta, koska joka paikassa ei ole ollut jokaiselle koodarille tarjolla omaa päätettä, vaan koodausta on tehty vuoronperään.

- Tietona (viite: Vesa Oinonen) löydettiin yhteenveto pariohjelmoinnin hyödyistä:
- Kaksi ohjelmoijaa havaitsee virheet helpommin kuin yksi
- Ihminen tulee helposti sokeaksi omalle työlleen - toinen henkilö huomaa virheitä tehokkaammin
- Jatkuva koodin katselmointi
- Ratkaisut täytyy perustella parille, joten ratkaisuita täytyy miettiä tarkemmin
- Ihmisten luottamus ratkaisuun varmempi kun se on tehty pareittain [4]
- Heikompi koodiriviä/henkilö tehokkuus, mutta koodin refaktorointiin ja bugien korjaukseen kuluu myöhemmin vähemmän aikaa
- Tehokkainta kun parin henkilöt saman tasoisia, mutta hallitsee eri osa-alueita
- Informaation epäformaali tiedottaminen tehostuu kun pareja vaihdetaan
- Ohjelmistoa suunnitellessa mahdollisuus brainstorming-tekniikan käyttöön

- Keskusteluissa saatavilla oleva asiakas nähtiin tärkeänä, mutta asiakkaan ei tarvitse olla "päivystäjä" vaan riittää, kun hän on saatavilla riittävän lyhyellä, ehkä joidenkin tuntien, varoitusajalla.
- Saatavilla olevan asiakkaan tulee olla oman alueensa expertti. Lisäksi hänellä tulee olla myös päätösvaltaa, jotta hän voi ottaa kantaa toteutukseen liiketoiminnallisesta näkökulmasta. Jonkinlaisena riskinä nähtiin se, että jos saatavilla on liian monta asiakkaan edustajaa, on hanketta vaikeampi hallita.
- Iteraatiokierroksessa etsitäänkin vastausta siihen, "mitkä ovat hyödyt saatavilla olevasta asiakkaasta?" ja kuka on se "saatavilla oleva asiakas".

- On oletettavaa, että asiakas sitoutuu paremmin ohjelman kehitykseen, jos asiakkaalta edellytetään myös oman resurssin sitomista käynnissä olevaan kehitystyöhön.
- Ilman asiakkaan panostusta asiakkaalle toteutettavia tietojärjestelmähankkeita on hyvin vaikea viedä onnistuneesti lävitse, sillä yleensä vain asiakas tietää omat toimintatapansa ja tarpeensa. Eri projekteissa on tietysti eroja painotuksilla oma työ - ostettu työ, mutta oman työn osuus ei voi olla nolla tai projektin epäonnistuminen on hyvin todennäköistä.

- Tyypillisesti ulkopuolisia resursseja halutaan käyttää kolmesta eri syystä
 1. Oma osaaminen puuttuu eikä sitä ole järkevää hankkia
 2. Omat resurssit halutaan kohdistaa omaan ydinliiketoimintaan
 3. Kehityksen nopeuttamiseksi tarvitaan lisää resursseja

- Riippumatta siitä, mikä on ulkopuolisen resurssin hankinnan syy, edellytetään XP:ssä aina asiakkaan saatavilla olemista. Haasteena on kuinka paljon projektiin joudutaan varaamaan asiakkaan avainresursseja, joiden työaika halutaan aina säästää asiakkaan ydinliiketoimintaan. Muussa tapauksessa toimittajalta edellytetään asiakkaan substanssiosaamista, mutta kuinka monelta sitä aidosti löytyy?
- Jos tarvittava asiakkaan liiketoiminnan tuntemus on olemassa, keskitytään silloin kenties ns. bulkkisovelluksen tekemiseen – ei räätälöityjen ohjelmistojen tuottamiseen. Tämä ei kuitenkaan tarkoita etteikö XP:tä voisi menetelmänä hyödyntää, sillä kaikissa tapauksessa sovelluksella pitäisi olla tilaaja, joka on siis asiakkaan roolissa huolimatta siitä, että kyseessä onkin yrityksen "sisäinen asiakas".

- Asiakkaan saatavuudella tavoitellaan erityisesti kolmen sovelluksen kehityksen kannalta olennaisen työvaiheen nopeuttamista ja laadun varmistamista:
 1. Käyttötapausten kuvaaminen tai kuvausten auditointi
 2. Sovelluksen demojen testaus ja kommentointi
 3. Valmiin sovellusosan testaus ja hyväksyntä

- Näiden lisäksi asiakasta tarvitaan vastaamaan esille nouseviin kysymyksiin, asettaman tarvittavat prioriteetit tehtävien ja/tai vaatimusten kesken sekä määrittelemään muut vaatimukset. Ilman asiakkaan läsnäoloa koko menetelmä romuttuu kokoelmaksi hyviä yksittäisiä ideoita tai käytäntöjä, mutta kokonaisuus ei toimi halutulla tavalla ja menetelmä on verrattavissa perinteisiin menetelmiin.
- Osittain asiakkaan saatavuus menee myös ristiin käyttäjätarinoiden muodostamisen kanssa, sillä hyväkin kuvaus voi jättää aukkoja, joiden selvittämiseen tarvitaan asiakkaan läsnäoloa. Ilman asiakkaan läsnäoloa tällaisten aukkojen selvittäminen saattaa vaatia hyvinkin pitkiä aikoja tai pahimmillaan se johtaa siihen, että toteutuksessa tehdään olettamuksia, jotka osoittautuvat virheellisiksi vasta valmiin sovelluksen testausvaiheessa tai sen tuotannossa ollessa!

Jouni:

- Vaikka en haluaisi suoraan myöntää, sai Jarmo minua vakuutettua ensialkuun vaikeilta tultuneista periaatteista liittyen pariohjelmointiin tai yhteiseen koodin omistajuuteen. Käyttäjätarinat tunsin tietyssä mielessä jo entuudestaan, mutta XP:n tapa käyttää niitä määrittelydokumentteina ja asiakasohjauksen välineinä oli paitsi uusi, myös hiukan outo. Sen toimivuus riippuu mielestäni paljon projektista ja henkilökemioista mutta kieltämättä kokeilemisen arvoinen juttu.

- Lähiopetus 3x4h = 12h
- Työjärjestyskeskustelut sähköpostitse 3x0.5h = 1.5h
- Tiedonhaku 11h
- Pohdintaa tiedonhaun pohjalta 5h
- Teesien kirjoittaminen 7h
- Viestien kirjoittamista ja kysymyksenasettelua 14h
- Kriittistä arviointia muille ryhmille 3h
- Väliraportin 1 kirjoittaminen 3h
- Kaikki yhteensä: 56,5 tuntia

Jarmo:

- Lähtökohtana XP:n tutkimiselle oli uskoakseni tilanne, jossa itselläni oli vähän kokemusta XP:stä ja muilla niukemmin. Tämä johti ainakin aluksi siihen, että muut - erityisesti Jonkka ja Jan - haastoivat minua XP:n perusteeseistä. Jo lähitunneilla asioista nousi hyvää ja ansiokasta keskustelua.
- Huomasin, että rooli "XP:n puolustajana" vaati runsaasti lisäperehtymistä itse asiaan. Sainpahan vihdoin luettua aiemmin hankkimani kirjan (käyttäjätarinoista) aiheeseen liittyen. Huomasin, kuinka hyvä tapa oppia on, kun joku kyseenalaistaa puolustamaasi asiaa ja joutuu vasta-argumentoimaan asiaa siten, että saa sanoman perille vastapuolelle. Voipi sitä joutua itsekin muuttamaan mielipiteitään matkalla.

- 12 h lähiopetustunnit
- 15 h tiedonhaku ja User Stories-kirjan luku
- 13 h ajatusten kokoaminen aiheen tiimoilta omien kokemusten ja luetun materiaalin perusteella
- 13 h viestien laadinta ja vastaukset
- 1 h ryhmän välinen kommunikaatio sähköpostilla
- 3 h muiden ryhmien tuotoksiin perehtyminen
- 1 h muiden ryhmien tuotosten kommentointi
- Yhteensä: 58 h

Marko:

- Oma kokemus aihealueesta ennalta oli varsin vaatimaton ja sen vuoksi aiheesta oli alkuun vaikea saada otetta. Tämä näkyi mielestäni koko ryhmän toiminnassa, sillä tehtävän fokuointi tuotti alkuun vaikeuksia.
- Kuten Jari omassa kommentissaan mainitsi, oli "punaisen langan etsiminen" ryhmien viestiketjuista ajoittain varsin työlästä. Vaikka Fle3 tuokin jonkin verran lisäarvoa tiedon jakamiseen, niin on siinä myöskin varsin merkittäviä puutteita. Tämän vuoksi kaipasin erityisen suuresti mahdollisuutta muokata ja jäsenellä tietoja omaa käyttötarkoitustani varten. Nyt ainoaksi järkeväksi tavaksi jäi tiedon kopioiminen omaan tiedostoon, josta sen saattoi jäsenellä haluamaansa muotoon. Tämän "kankeuden" vuoksi osa kurssin aiheista käsiteltiin meilinvaihtona, jolloin tiedon jäsentely ja arkistointi omaan käyttöön oli yksinkertaisempaa.

- 12 h lähiopetustunnit (3x4h)
- 2 h ryhmän välinen kommunikaatio sähköpostilla
- 12 h tiedonhaku ja tietoihin perehtyminen (WWW ja Fle3 -keskustelut)
- 15 h pohdinnat ja niiden yhteenvedo tiedon jäsentelyä varten
- 4 h muiden ryhmien tuotoksiin perehtyminen
- 1 h muiden ryhmien tuotosten kommentointi
- 4 h väliraportin 3 kirjoittaminen
- 4 h loppuraportin valmistelu ja viestin vaihto aiheesta ryhmän jäsenten kanssa
- 5 h loppuraportin ja esityksen laadinta
- Yhteensä 59 tuntia

Jari:

- Aihe alue oli mielenkiintoinen, ja ryhmässä oli hyvin erilaista osaamista tältä alueelta. Se oli minusta hyvä asia ryhmälle, mutta välillä minusta hajanaisuus jopa haittasi, sillä keskustelun fokusoituminen oli välillä kateissa. Olisiko meidän pitänyt "ryhmäytyä" vähän enemmän ennen tiedonrakentamista?
- Aikaa meni paljon siihen, että etsi ryhmien viestiketjuista punaista lankaa. Olisiko voitu käyttää aikaa lähiopetuksessa siihen kuinka viestejä kannattaa linkittää toisiinsa, tai kuinka "muistilapuista" saa jäsenneltyä kokonaisuuksia. Voisiko oppimisympäristössä lajitella, siirrellä ja ryhmitellä viestejä?

- Lähiopetus 2x4h = 8h
- Ketterien menetelmien luento 3 h
- Työjärjestyskeskustelut sähköpostitse ~ 1.5h
- Tiedonhaku ja aihealueen kirjallisuuteen tutustumista ~15h
- Viestien ja väliraporttien lukemista ja pohdintaa Flessä ~ 20h
- Kriittistä arviointia muille ryhmille ~ 3h
- Yhteensä noin 51 h

Jan:

- Alun liikkeelle lähtö oli melko tahmaista, kukaan ei oikein tuntuvat saavan selkeää otetta siihen, miten asiassa tulisi edetä. Vasta toisen lähiopetuskerran jälkeen tiedonrakentelu lähti kunnolla käyntiin ja se punainen lankakin alkoi löytymään. Keskustelun rajaaminen vain valittuihin teeseihin osoittautui hyväksi ratkaisuksi, asiasta oli niin paljon tietoa tarjolla, ettei kaikkeen olisi voinutkaan syvällisesti perehtyä.
- Jarmo osoittautui selkeästi ryhmästämme aiheesta kaikkein kiinnostuneimmaksi, ja onnistui löytämään aiheesta hyvinkin syvällistä tietoa. Pitkälti häneen vakuuttavien argumenttiansa pohjalta minäkin aloin lämpenemään XP:n mahdollisuuksille, vaikka se tilaajaorganisaation näkökulmasta syökin organisaation parhaita reursseja toimitusprojekteihin osallistumiseen (olen työssä "tilaavassa organisaatiossa").

- Lähiopetus 12 h
- Tiedonhaku ja tietoihin perehtyminen 12 h
- Muiden ryhmien tuotoksiin perehtyminen 3 h
- Ryhmätyön koordinointiin liittyvät työjärjestyskeskustelut yms. 2 h
- Viestien laadinta ja vastaukset, väliraportti 20 h
- Yhteensä 49 h.

Kysymyksiä?

