

KEHITTÄMISEN MENETELMÄT – MIKSI EXTREME PROGRAMMING?

Versio	1.0
Opettaja/tarkastaja	Eija Kalliala
Ryhmä	Marko Keinänen, Jari Kauppi, Jarmo Levo, Jan Partanen ja Jouni Tikkanen
Päiväys	2.5.2007

**Miksi käyttää XP:tä perinteisten
kehitysmenetelmien sijaan?**

SISÄLTÖ

1	JOHDANTO	3
2	ENSIMMÄINEN ITERAATIOKIERROS	4
2.1	TUTKIMUSONGELMA.....	4
2.2	TIEDON TULVAA.....	5
2.3	EXTREME PROGRAMMING TEESIT	5
2.4	VALITUT TEESIT.....	6
3	TOINEN ITERAATIOKIERROS	6
3.1	SUUNNITTELU / KÄYTTÄJÄTARINAT.....	7
3.2	PARIOHJELMOINTI	7
3.3	SAATAVILLA OLEVA ASIAKAS.....	8
3.4	VIRIKKEEKSI KOLMANNELLE JA VIIMEISELLE ITERAATIOKIERROKSELLE	9
4	KOLMAS ITERAATIOKIERROS	9
4.1	SAATAVILLA OLEVA ASIAKAS.....	9
4.2	KYSYMYKSET JA POHDINNAT	10
5	YHTEENVETO JA AJANKÄYTTÖ	12
5.1	JOUNI.....	12
5.2	JARMO.....	13
5.3	MARKO.....	14
5.4	JARI.....	15
5.5	JAN.....	16

1 JOHDANTO

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

Näistä voittajiksi selvisivät aihealueet 1, 7 ja 11, eli suunnittelu, pariohjelmointi ja saatavilla oleva asiakas. Tutkimusongelmaa avattiin kolmen iteraatiokierroksen avulla, joissa kussakin aiheeseen syvennettiin ennalta tehtyjen selvitysten perusteella ja joihin muut ryhmät esittivät omat kriittisen arvionsa.

2 ENSIMMÄINEN ITERAATIOKIERROS

2.1 TUTKIMUSONGELMA

Kysymys oli, miksi Extreme programming?

Kun valitsimme aiheetta, nousivat sanat "ketterät menetelmät" muita useammin esiin ja kun ryhmässä joillakin oli myös omaa kokemusta sekä ohjelmistoprojekteista että osin myös Extreme Programming-menetelmästä, oli valinta aika helppo. Lopulta rajasimme aiheetta hiukan tarkemmin kun luovuimme ketteristä menetelmistä yleensä ja keskityimme vain Extreme Programming -menetelmään.

Vaikka aihe löytyikin nopeasti, oli tutkimusongelman tarkemmassa hahmottamisessa alkuun vielä tekemistä. Heti alusta alkaen XP:stä tuntui löytyvän materiaalia paljon ja helposti. Se ei ole ihme, onhan kyse sentään muotitermistä, mutta tiedon etsiminen ja sen määrä vei fokusta pois kysymyksen asettelusta.

Tutkimusongelma täsmennettiin seuraavaan muotoon:

Haluaisin selvittää,

- mistä XP:ssä on kyse
- mitkä on sen hyödyt?
- miten se vaikuttaa kuluihin?
- mikä on asiakkaan rooli ja työpanos?
- miten laatuasiat on otettu XP:ssä huomioon?
- miten projektin johtaminen eroaa perinteisistä menetelmistä?

- muuttuvatko vaatimukset projektin aikana?

2.2 TIEDON TULVAA

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.

2.3 EXTREME PROGRAMMING TEESIT

XP:n teeseiksi on esitetty seuraavia kahtatoista kohtaa:

- * The Planning Process
- * Small Releases
- * Metaphor
- * Simple Design
- * Testing
- * Refactoring
- * Pair Programming
- * Collective Ownership
- * Continuous Integration
- * 40-Hour Week
- * On-Site Customer
- * Coding Standard

Nämä ovat vain otsikot, tarkemmat pitkät selitykset löytyvät viestistä otsikon "XP:n teesejä ja niiden ruodintaa" alta.

Teeseistä käytiin keskustelua ja saatiin tietoa myös muilta ryhmiltä ja esimerkiksi käyttäjätarinoista oli saatu hyviä kokemuksia. Käyttäjätarinat olikin yksi eniten kommentoiduista aiheista ja kuten Jarmo eräässä kohtaa kirjoittikin, käyttäjätarinoiden funktio ei lopu vain toiminnallisen ympäristön kuvaamiseen vaan niiden pohjalta suunnitellaan mm. hyväksymistestit.

Näkökulmia alkoi kerääntyä ja samassa tahdissa kasvoi myös käsiteltävänä olevan tiedon määrä. Kun keksimme tämän 12-kohtaisen XP:n teesilistan, alkoi käsiteltävänä olevan tiedon määrä ylittää kohtuuden rajat ja päätimme rajata käsittelyn vain muutama näistä.

2.4 VALITUT TEESIT

Valitsimme jatkokäsittelyyn näin aluksi neljä teesiä. Valinnassa ei sinänsä ole yllätyksiä, mukana on esimerkiksi ryhmän jäseniä alusta alkaen keskusteluttanut pariohjelmointi. Sen lisäksi valituiksi tulivat käyttäjätarinat - tosin yllättävän pienellä äänisaaliilla - yksinkertainen ratkaisu sekä saatavilla oleva asiakas.

3 TOINEN ITERAATIOKIERROS

Ensimmäisen kierroksen tiedonrakentelun aikana havaitsimme, että valitsemastamme aiheesta löytyi tietoa niin paljon, että meidän oli rankasti rajattava jatkotyö tietyille alueille. Käytimme valinnassa löytämäämme XP:n teesilistaa, ja äänestimme keskenämme siitä 3 meitä kiinnostavinta teesiä.

Valituiksi syvennettäviksi tutkimusongelman kohteiksi tulivat 27.3.2007 teesit:

1: suunnittelu / käyttäjätarinat

7: pariohjelmointi

11: saatavilla oleva asiakas

3.1 SUUNNITTELU / KÄYTTÄJÄTARINAT

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). Se on:

- riippumaton
- sisältö on aina täsmennettävissä
- hankkijalle tai käyttäjälle arvokas
- ennustettavissa
- pieni

3.2 PARIOHJELMOINTI

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 on 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 samantasoisia, mutta hallitsee eri osa-alueita
- Informaation epäformaali tiedottaminen tehostuu kun pareja vaihdetaan
- Ohjelmistoa suunnitellessa mahdollisuus brainstorming-tekniikan käyttöön

Nämä löydetyt hyödyt ja vastaukset peilaantuvat hyvin alkuperäisen tutkimusongelman kysymystenasetteluun: "Mitkä ovat XP:n hyödyt" ja "Miten laatuasiat on otettu XP:ssä huomioon"

3.3 SAATAVILLA OLEVA ASIAKAS

Tämän teesin käsittely herätti kaikkien eniten pohdintaa valitsemistamme teeseistä. Ehkä osin siksi, että tämän teesin suhteen ryhmällä ei ollut aivan samaa vakuuttuneisuutta "oikeasta" vastauksesta.

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, jos saatavilla on liian monta asiakkaan edustajaa. Tällöin hanketta on vaikeampi pitää "hanskassa".

Kun tätä teesiä peilaa tutkimusongelman kysymystenasetteluun, voisi vetää teesin käsittelystä yhteen, että tässä ei aivan voida vastata kysymykseen: "Mitkä ovat hyödyt saatavilla olevasta asiakkaasta?". Vastaus riippuu siitä, kuka on se "saatavilla olevasta asiakas".

3.4 VIRIKKEEKSI KOLMANNELLE JA VIIMEISELLE ITERAATIOKIERROKSELLE

Olisiko teesin "saatavilla oleva asiakas" syventävä käsittely viimeisen kierroksen ykköskohteena? Se kuitenkin jäi tällä iteraatiokierroksella käsitellyistä teeseistä kaikkein avoimimmaksi.

4 KOLMAS ITERAATIOKIERROS

Toisella iteraatiokierroksella valittiin syvennettäviksi tutkimusongelman kohteiksi teesit:

- 1: suunnittelu / käyttäjätarinat
- 7: pariohjelmointi
- 11: saatavilla oleva asiakas

Näistä kolmas, eli saatavilla oleva asiakas keräsi eniten ääniä saaden 15 annetusta äänestä viisi. Koska saatavilla oleva asiakas herätti myös laajaa keskustelua, pohdintaa ja kommentointia, päätettiin kolmannella iteraatiokierroksella keskittyä vain ja ainoastaan saatavilla olevaan asiakkaaseen, johon ei pohdintojen pohjalta voitu vakuuttua "oikeasta" vastauksesta.

4.1 SAATAVILLA OLEVA ASIAKAS

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 olevasta asiakas".

4.2 KYSYMYKSET JA POHDINNAT

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.

Tyypillisestihän 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 työaikaa 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!

Asiakkaan läsnäolon avulla voidaan myös osaltaan varmistua siitä, että sovelluskehitykseen liittyvä tieto siirtyy – tai pysyy – asiakkaalla itsellään, eikä tieto jää ainoastaan yrityksen ulkopuolisten konsulttien/ohjelmoijien haltuun. Asiakkaalta voi olla läsnä myös erilaisissa rooleissa olevia edustajia, joiden vastualueet poikkeavat toisistaan. Läsnäoloon ei riitä että paikalla on joku asiakkaan edusta, vaan edustajan tulee aina olla kyseisen (liiketoiminnan) osa-alueen syvällinen asiantuntija, joka pystyy vastaamaan vaikeisiinkin kysymyksiin ja jolla on myös valtuuksia päättää niistä. Liian

usean asiantuntijan käyttö taas saattaa aiheuttaa turhia haasteita projektin hallintaan.

5 YHTEENVETO JA AJANKÄYTTÖ

Valittuihin aihe-alueisiin, suunnittelu/käyttäjätarinat, pariohjelmointi ja saatavilla oleva asiakas päästiin pureutumaan erittäin hyvin. Aihe-alueiden rajausta voidaan siten pitää onnistuneena valintana. Omakohtaisia oppimiskommentteja kirjasimme seuraavasti:

5.1 Jouni

Vaikka en haluaisi suoraan myöntää, sai Jarmo minua vakuutettua ensialkuun vaikeilta tultuneista periaatteista liittyen pariohjelmointiin tai yhteiseen koodin omistajuuteen. Käyttäjätarinat tunsin tietysti 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.

Ehkä parhaiten minulle meni läpi teesi "saatavilla oleva asiakas", mistä eniten kirjoittelinkin ja mistä olin myös kiinnostunut.

Oppimisprosessi oli varsin nykivää, alkuun löytyi paljon tietoa ja innoissamme sitä keräsimme kunnes tietovuori alkoi ylittää käsityskykyämme. Tässä vaiheessa innostus oli kovaa ja tietoa kerättiin huimaa tahtia. Sitten tuli pysähtyneisyyden vaihe kun mietimme miten jatkamme ja teesiäänestyksen jälkeen tilanne on ollut taas normaali. Pääsimme keskittymään ydinaiheisiin (valittuihin kolmeen teesiin) ja ruotimaan niistä lähtien paitsi alkuperäisiä tutkimusongelmia myös teesien keskinäisiä suhteita.

Vaikka en ehkä viimeistä vastausta kysymykseen "miksi Extreme Programming" saanutkaan, sain ainakin paljon eväitä siihen mistä on kyse ja millaisissa tilanteissa kannattaa harkita tällaista uudempaa ja radikaalilta kuulostavaa menetelmää.

Käytetty aika:

- 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

5.2 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.

Alustavasti puhuimme aiheen laajan kentän rajaamisesta kolmeen osa-alueeseen, jotka saivat eniten kannatusta. Äänestyksen jälkeen asia lähti taas liikkeelle. Suurin into lienee vähän laantunut ryhmällämme. Keskustelimme oikeastaan enemmän lähituntien yhteydessä asioista kuin Fle:n välityksellä.

Ajatusten jäsentäminen kirjoittamisen kautta kahlitsee mielestäni jonkin verran asian eteenpäin vientiä, koska se hidastaa itse oppimisprosessia.

Spontaanisuus katoaa ja ns. "villit ideat" tai oivallukset jäävät helposti kontrolloivan minän varjoon.

Itse asia kiinnostaa minua yhä enemmän. Tuo ketterien menetelmien luentokin oli hyvä, tosin sehän ei käsitellyt projektinhallintaa XP:n vaan Scrummin kannalta. Jatkan varmasti asiaan paneutumista.

Ajankäyttöni:

- 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

5.3 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 fokusointi 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 käsiteltiinkin meilinvaihtona, jolloin tiedon jäsentely ja arkistointi omaan käyttöön oli yksinkertaisempaa.

Tämän kaltaisen uuden aihealueen opiskelussa -ja etenkin aiheesta tehtävien esitysten valmistelussa- sähköisillä työvälineillä välitetty viestintä saattaa hidastaa ja lisätä käytettyä työmäärää merkittävästi kahdesta syystä

- 1) Yksinkertaisen asian käsittely venyy helposti useiden kymmenien viestien ketjun mittaiseksi. Puhelinsoitolla tai "face-to-face" asia saataisiin käsitellyksi minuutissa
- 2) Viestiketjujen vasteajat voivat olla tunteja tai jopa päiviä. Siinä ajassa alkuperäinen ajatus saattaa jo kadota.

Itse XP-ohjelmoinnin menetelmiin kurssi on tuonut huomattavasti uutta tietämystä ja perustelujen käsittely on ollut erittäin mielenkiintoista. Valitsemamme osa-alueet, teesit, osoittautuivat tärkeiksi ja siten myös fokuointi onnistuneeksi.

Käyttämäni aika:

- 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 yhteenveto 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

5.4 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 fokuoituminen 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 jäseneltyä kokonaisuuksia. Voisiko oppimisympäristössä lajitella, siirrellä ja ryhmitellä viestejä?

Kolmannen iteraation keskustelu jäi vähäiseksi. Tämä johtui varmaan osittain siitä, että meillä oli alusta alkaen jo paljon tietoa aiheesta. Näin ei enää löytynyt paljoa uutta, mutta olisimmeko voineet pohtia vielä enemmän XP:n hyötyjä verrattuna perinteisiin kehittämisen menetelmiin.? Nyt keskityttiin aikalailta XP:n sääntöihin ja miten niitä työelämässä käytetään.

XP-ohjelmointiin olen nyt törmännyt kurssin aikana myös muita kirjoja ja lehtiä lukiessa. Ehkä en vaan ole aikaisemmin kiinnittänyt XP:hen huomiota, kun en työskentele päivittäin ohjelmoinnin parissa. Nyt aihealue on tullut tutuksi ja sitä voi hyödyntää tarvittaessa.

Ajankäyttö:

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

5.5 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 argumenttiensa pohjalta minäkin aloin lämpenemään XP:n mahdollisuuksille, vaikka se tilaajaorganisaation näkökulmasta syökin organisaation parhaita reursseja toimitusprojekteihin osallistumiseen (olen työssä "tilaavassa organisaatiossa").

Teeseistä "saatavilla oleva asiakas" oli minulle ehdottomasti kaikkein kiinnostavin. Keskusteluissamme minulle avautui (mielestäni) aika hyvin se näkökulma ja odotukset, mitä ohjelmistojä toimittavalla organisaatiolla on asiakkaasta.

Tehdyn työn perusteella sain perustietoa XP:stä, ja voin mahdollisesti hyödyntää tietoja työssäni jo syksyn eräessä ohjelmistotoimituksessa (organisaatiomme on tehnyt tarjouspyynnön yhden ohjelmiston toimituksesta). Jos tarjoajalla on kehittämistyössä tarjolla menetelmän XP, niin ainakin tiedän, mistä on kyse.

Entäpä tämä oppimisympäristö FLE3? Uusi tapa tämä oli minulle, ja jossain määrin aika haasteellinen. Yksinkertaisen asian läpi käyntiin voi mennä aikaa vaikka kuinka. Toisaalta omat vastaukset tulee harkitummin tehtyä. Kaipaen itse kuitenkin enemmän sitä välitöntä keskustelua ja palautetta, keskusteluosapuolen ensikommentteja ja reaktioita. Tällainen ympäristö toki puolustaa paikkaansa silloin, kun ryhmän jäsenet ovat maantieteellisesti hajallaan. Tähän ympäristöön liitettynä web-kameraa käyttäen pidetyt chatit voisivat olla hyvä yhdistelmä.

Käytetty aika:

- 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.