

10 Kohti ketterää ohjelmistokehitystä

- Perinteinen ohjelmistokehitys perustuu vesiputousmalliin, jossa tavoitteena on ensisijaisesti projektin vieminen läpi tietyssä ajassa.
- Sovelluksen määrittelytyö tehdään yhteistyössä tilaajan kanssa. Määrittelytyön tulokset dokumentoidaan eli kirjataan tarkasti ylös paperille. Asiakkaan hyväksynnän jälkeen määrittely lyödään lukkoon ja aloitetaan sovelluksen teknisen toteutuksen suunnittelu. Suunnittelutyön tulokset dokumentoidaan ja lyödään lukkoon. Sovellus toteutetaan ja toiminnallisuus testataan. Valmis!
- Hyvältä kuulostaa, mikä vikana? Baskerville ym. (2003) kuvaavat erityisesti Internetiin liittyvää ohjelmistokehitystä seuraavasti:
 - Sovellusten toteuttamiseen käytössä oleva aika on pienempi kuin koskaan aikaisemmin
 - Sovelluksille asetettavat vaatimukset muuttuvat nopeassa tahdissa
 - Sovelluksen laadun kriteerit poikkeavat perinteisestä ohjelmistokehityksestä
 - Tuotteen saaminen ajoissa (ensimmäisenä) markkinoille on kriittistä
- Internetiin liittyvä ohjelmistokehitys poikkeaa perinteisestä ohjelmistokehityksestä.

7307010 Hypermedian ohjelmointi (kevät 2005)

132

Perinteinen ohjelmistokehitys: vesiputousmalli

- Ohjelmistokehitys perustuu perinteisesti vesiputousmalliin. Vesiputousmallin keskeiset vaiheet ovat esitutkimus, määrittely, suunnittelu, toteutus ja testaus.
- Vesiputousmallissa vaiheet viedään läpi peräkkäin yhden kerran. Sovelluksen haluttu toiminnallisuus siis kiinnitetään ennen teknisen toteutuksen suunnittelun aloittamista. Asiakas on mukana pääasiassa esitutkimuksessa ja määrittelyssä.
- Vesiputousmalli soveltuu käyttöön sellaisissa projekteissa, joissa
 - toteutetaan järeitä järjestelmiä,
 - kehitystyö kestää pitkään (jopa vuosia),
 - järjestelmän käyttöaika on pitkä (vuosia),
 - käyttäjien tarpeet ovat tiedossa ja
 - sovellusalue on vakaa.
- TTY:llä kehitetty HYTT-laaturjärjestelmä on (pelkistetysti) vesiputousmallin mukainen elinkaarimalli, jota voidaan käyttää myös esimerkiksi astettaisen kehittämisen mallin ja EVO-mallin mukaisessa kehitystyössä.

7307010 Hypermedian ohjelmointi (kevät 2005)

133

Internet-nopeudella tapahtuva ohjelmistokehitys (1/2)

- Baskerville ym. (2003) ovat listanneet joukon Internet-nopeudella tapahtuvassa ohjelmistokehityksessä yleisesti käytössä olevia toimintatapoja:
 - Rinnakkainen kehitystyö: versioita kehitetään rinnan eri työvaiheissa (suunnittelu, toteutus, laadunvarmistus)
 - Versioiden tiivis julkaiseminen: työlääksi osoittautuneet ominaisuudet tuleviin versioihin, ...
 - Työvälineiden käyttäminen: Integrated Development Environment (IDE), sovelluskehitykset, luokka- ja funktiokirjastot, ...
 - Tiivis yhteistyö asiakkaan kanssa: yksityiskohtaisen, kerralla tapahtuvan vaatimusmäärittelyn sijaan asiakas otetaan mukaan sovelluksen kehitystyöhön
 - Vakaa sovellusarkkitehtuuri: arkkitehtuurin vakaus antaa enemmän tilaa sovelluksen ominaisuuksien muokkaamiseen ja dokumentaation supistamiseen
 - Komponenttien uudelleenkäyttö: uudelleenkäytettävien komponenttien toteuttaminen ja etenkin hyödyntäminen
- Pääosa luetelluista toimintatavoista pätee myös perinteisessä ohjelmistokehityksessä

7307010 Hypermedian ohjelmointi (kevät 2005)

134

Internet-nopeudella tapahtuva ohjelmistokehitys (2/2)

- Osa yleisesti käytössä olevista toimintatavoista vaatii alustukseksi lyhyen moraalisaarnan:
 - Toimintavat ovat *yleisesti käytössä*, niiden soveltamista ei tässä *erikseen suosittella*
 - Opettele säännöt kunnolla ennen niiden rikkomista.
- Ja ne loput toimintavat:
 - Ylläpidon sivuuttaminen: sovellusten elinkaaren lyhyys voi johtaa siihen, että sovellusten toteuttamistyöhön haasteita aiheuttavan ylläpidettävyyden huomioimisen sijaan ylläpidettävyyteen tähtäävät toimenpiteet unohdetaan osittain tai kokonaan
 - Toimintavan päivittäinen räätälöinti: kehitysprosessin säätäminen tilanteen mukaan, esimerkiksi työvaiheiden ohittaminen (jopa sovelluksen laadun heikkenemisen kustannuksella), ...
- Internet-nopeus edellyttää uudenlaista toimintatapaa, kun sovelluskehitystä halutaan nopeudesta huolimatta tehdä hallitusti. Varteenotettavan vaihtoehdon tarjoavat ketterän ohjelmistokehityksen (Agile Software Development) menetelmät

7307010 Hypermedian ohjelmointi (kevät 2005)

135

Ketterä ohjelmistokehitys: taustaa

- Ketterät ohjelmistokehityksen menetelmät (Agile Software Development) pyrkivät vastaamaan kevyempien ja liukasliikkeisempien ohjelmistokehitysprosessien tarpeeseen, joita on erityisesti Internet-sovellusten ja mobiilisovellusten kehittäjillä
 - agile [ˈædʒaɪl] a, ketterä, notkealiikkeinen, ripeäliikkeinen, terävä-älyinen (kielikone.fi)
 - "the quality of being agile; readiness for motion; nimbleness, activity, dexterity in motion" (Oxford English Dictionary)
- Ketterien ohjelmistokehityksen menetelmien soveltajista koostuva Agile Alliance (www.agilealliance.org) on julkaissut ketterän ohjelmistokehityksen julistuksen, joka sisältää seuraavat neljä kohtaa (agilemanifesto.org/):
 - **Individuals and interactions** over processes and tools
 - **Working software** over comprehensive documentation
 - **Customer collaboration** over contract negotiation
 - **Responding to change** over following a plan

Ketterä ohjelmistokehitys

- Abrahamsson ym. (2002) esittää ketterän ohjelmistokehityksen määritelmän, jonka mukaan seuraavat piirteet ovat tyypillisiä ketterille menetelmille:
 - **Inkrementaalinen kehitystyö:** inkrementaalinen ~vaiheittain kasvava, pieniä muutoksia, uusien versioiden tiivis julkaiseminen
 - **Yhteistyö:** tiivis kommunikaatio asiakkaan ja kehittäjien välillä sekä kehittäjillä keskenään koko kehitystyön ajan
 - **Suoraviivaisuus:** ketterät menetelmät itsessään ovat yksinkertaisia, helposti opittavia ja muokattavia sekä hyvin ohjeistettuja
 - **Mukautuvuus:** muutosten tekeminen tulee olla mahdollista missä kehitystyön vaiheessa tahansa
- Ketterät menetelmät eivät ole täysin uusi keksintö. Tärkein uusi tekijä on ihmisten huomioiminen tärkeimpänä yksittäisenä projektin onnistumiseen vaikuttavana tekijänä. Myös toteutustyön tuottavuus ja ohjattavuus on nostettu entistä keskeisempään asemaan.
- Esimerkkejä menetelmistä: Extreme Programming, Scrum, Crystal family of methodologies, RUP, ASD ja DSDM. Agile Modeling tukee ketteriä menetelmiä.

Extreme Programming (XP)

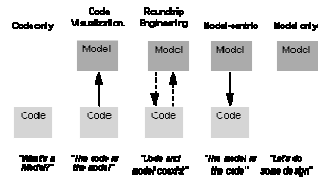
- Kent Beckin 1990-luvun lopussa kehittämä Extreme Programming (XP) on käytetyin ketterän ohjelmistokehityksen menetelmä.
- XP-ohjelmoinnin keskeiset arvot ovat:
 - Kommunikointi: asiakas ja tekijät tuntevat toisensa ja kommunikoivat jatkuvasti
 - Yksinkertaisuus: sovellus toteuttamisessa käytetään yksinkertaisimpia mahdollisia ratkaisuja
 - Palaute: sovellusalueen asiantuntija ohjaa kehitystyötä oikeaan suuntaan ja tekniset asiantuntijat informoivat tätä rajoitteista ja mahdollisuuksista
 - Rohkeus: yksinkertainen suunnittelu, koodin uudelleenmuokkaaminen (refactoring), uusien ominaisuuksien vaatiminen
- XP-ohjelmointia voidaan hyödyntää projekteissa, joissa toteuttajaryhmät ovat pieniä tai keskikokoisia, käytössä on yhtenäinen tila (tai tiivis kommunikaatio on muuten mahdollista), osallistujat hyväksyvät toimintatavan ja käytössä oleva tekniikka mahdollistaa toteutuksen jatkuvan muuttamisen tehokkaasti.
- Tekijöiden osaaminen on ratkaiseva tekijä onnistumisessa.

XP-ohjelmoinnin toimintatavat

- Abrahamssonin ym. (2002) mukaan XP-ohjelmoinnille tyypillisiä toimintatapoja ovat:
 - Suunnittelupeli (Planning Game): toteuttaja ja asiakas, käyttäjätarinat
 - Versioiden tiivis julkaiseminen: ~kuukausittain
 - Metaforat: yhteinen kieli asiakkaan ja toteuttajien välille
 - Yksinkertaisuus
 - Testaaminen: yksikkötestit & toiminnalliset testit määritellään ennen toteutusta
 - Koodin uudelleenmuokkaus (refactoring): toistuvien osien poistaminen, ...
 - Ohjelmointi pareittain: ehkä tutuin XP-piirre
 - Koodin yhteisomistus: kaikki voivat muokata kaikkea koko ajan
 - Jatkuva integrointi
 - 40-tuntinen työviikko: intensiivinen työ vaatii riittävästi lepoa
 - Yhdenmukaiset ohjelmointikäytännöt, osallistuva asiakas, avoin työtila ja tarkat, yhteisesti sovitut säännöt

Haaste: mallintaminen

- Toisen lähestymistavan ohjelmistojen toteutustyön tehostamiseen esittelevät mallintamiseen perustuvat menetelmät. Ketterät menetelmät ja mallintaminen eivät välttämättä ole vaihtoehtoisia keskenään
- Mallintamisessa monimutkaisen reaali maailman ilmiön perusteella tuotetaan malli, joka kuvaa ilmiötä sovelluksen näkökulmasta riittävällä tarkkuudella. Mallintamisessa äärimmäisenä tavoitteena on se, että kaikki koodi tuotetaan korkeamman tason mallin perusteella. Yleisin väline mallintamiseen on Unified Modeling Language (UML).



Kuva: Daniels, J./Brown, A.

- Mallintamiseen perustuvia lähestymistapoja ovat esimerkiksi UML-kieleen perustuva Model-Driven Architecture (<http://www.omg.org/mda/>) sekä Domain-Specific Modeling

7307010 Hypermedian ohjelmointi (kevät 2005)

140

Poiminta: ketterämpi rajapintadokumentaatio ja PHP

- XP-ohjelmoinnissa pyritään viimeiseen asti välttämään kaikkea ylimääräistä toimintaa, kuten esimerkiksi rajapintadokumenttien kirjoittamista. XP projekteissa tiivis kommunikaatio korvaa suuren osan kirjallisesta dokumetaatiosta. Useissa tilanteissa rajapintadokumentaatio on kuitenkin tarpeellinen.
- Ketterässä mallintamisessa (Agile Modelling) korostetaan mm. sitä, että sovelluksen dokumentoinnissa on pidettävä huolta siitä, ettei informaatiota kopioida useisiin paikkoihin. Eräs esimerkki olemassa olevan tiedon hyödyntämisestä on rajapintadokumentaation automatisoitu tuottaminen kooditiedostoihin sijoitettujen määrämuotoisten kommenttien perusteella.
- JavaDoc on Java-kielestä tuttu väline rajapintadokumenttien tuottamiseen. PHP-kielelle löytyy vastaava toteutus, PHPDoctor (phpdoctor.sourceforge.net/). Myös muita vaihtoehtoja on olemassa. Esimerkki määrämuotoisesta funktion esittelystä:

```
/**
 * Palauttaa kieliversioidun merkkijonon tunnisteiden perusteella.
 * Tieto käyttäjän kielestä luetaan istunnosta.
 * @param stringid Merkkijonon tunniste
 * @return Merkkijono käyttäjän kielellä
 */
function select_text( $stringid ) {
```

7307010 Hypermedian ohjelmointi (kevät 2005)

141

Poiminta: asetustiedostot vs. koodin tuottaminen

- Sovellusten toteutustyötä voidaan tietyissä tapauksissa tehostaa koodin ohjelmallisella tuottamisella korkeamman tason kuvauksen perusteella. Tämä on keskeinen periaate mallintamiseen perustuvissa menetelmissä.
- Mallien esittämiseen voidaan käyttää esimerkiksi XML-kieltä. XML-muotoisia kuvauksia voidaan hyödyntää esimerkiksi seuraavilla tavoilla:
 1. Sovellus lukee XML-kuvauksen ja toimii sen mukaisesti
 2. Yleisestä XML-kuvauksesta tuotetaan sovelluskohtainen esitysmuoto, jota sovellus osaa lukea
 3. XML-kuvauksen perusteella tuotetaan dynaamisesta toiminnallisuudesta vastaava ohjelmakoodi
- Koodin tuottaminen korkeamman tason kuvauksesta ei useinkaan ole ketterän ohjelmistokehityksen tarkoittama yksinkertainen mahdollinen ratkaisu. Tietyissä käyttötarkoituksissa toimintaperiaatteen avulla voidaan kuitenkin automatisoida sovelluksen toteuttamiseen liittyviä rutiineja ja lisätä esimerkiksi sovelluksen ajonaikaista suorituskykyä.

7307010 Hypermedian ohjelmointi (kevät 2005)

142

Lopuksi

- Tässä esityksessä ohitettiin täysin erilaiset prototyyppien menetelmät, joita käsitellään kattavasti esimerkiksi kirjassa Preece, J. *Human-Computer Interaction*, Addison-Wesley, Essex, UK. 1999. Suositeltavaa lukemista on myös Haikala, I., Märijärvi, J. *Ohjelmistotuotanto*, Satku, Helsinki, Suomi. 2002.
- Ketterä ohjelmistokehitys saattaa kuulostaa mielenkiintoiselta erityisesti siksi, että usein tylsäksi koettu dokumentointi on perinteistä ohjelmistokehitystä pienemmässä roolissa. Dokumentoinnin keveys on kuitenkin väärä peruste ketterien menetelmien valinnalle.
- Ketterässä ohjelmistokehityksessä korostuu osaaminen. Tekijätimin on hallittava käytössä olevan ketterän menetelmän lisäksi yleiset ohjelmistojen suunnitteluun ja toteuttamiseen liittyvät teknologiat ja menetelmät, jotta toimintatapa osataan yksinkertaistaa oikealla tavalla
- Ketterät menetelmät asettavat haasteen myös esimerkiksi sopimusten tekemiselle. Kun työn laajuutta ei voida määrittellä toteutettavien ominaisuuksien ja toimintojen listalla, muuttuu määrittäminen hankalammaksi. Asiakas suostuu harvoin maksamaan sovelluksesta kulutettujen tuntien mukaan. Luottamus asiakkaan ja toteuttajan välillä korostuu.

7307010 Hypermedian ohjelmointi (kevät 2005)

143