

# **OpenUP –ohjelmistokehitysprosessi**

Sami Männistö

Helsinki 14.11.2008

Seminaari

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

## HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET

Tiedekunta/Osasto <b>Matemaattis-luonnontieteellinen</b>		Laitos – Institution <b>Tietojenkäsittelytieteen laitos</b>	
Tekijä – Författare <b>Sami Männistö</b>			
Työn nimi – Arbets titel <b>OpenUp ohjelmistokehitysprosessi</b>			
Oppiaine – Läroämne <b>Tietojenkäsittelytiede</b>			
Työn laji – Arbets art <b>Seminaari</b>		Aika – Datum <b>14.11.2008</b>	Sivumäärä – Sidoantal <b>8</b>
Tiivistelmä – Referat  <p>OpenUp on iteratiivinen ja inkrementaalinen ohjelmistokehitysprosessi, joka on syntynyt avoimen ohjelmistokehityksen tuloksena Eclipse Process Framework projektissa.</p> <p>OpenUP on myös ketterä menetelmä. Samalla kun sen juuret ovat Unified Process:issa ja RUP:issa, niin OpenUP korostaa ketterien menetelmien filosofian mukaan ohjelmistokehityksen kiinteää yhteistyötä vaativaa luonnetta.</p>			
Avainsanat – Nyckelord <b>OpenUP, RUP, UP, ketterät menetelmät, menetelmät, prosessit</b>			
Säilytyspaikka – Förvaringställe			
Muuta tietoja – Övriga uppgifter			

**Sisältö**

<b>1. Johdanto .....</b>	<b>1</b>
<b>2. OpenUP Historiaa.....</b>	<b>1</b>
<b>3. SPEM .....</b>	<b>2</b>
<b>4. EPF ja EPFC .....</b>	<b>2</b>
<b>3. OpenUP menetelmä .....</b>	<b>3</b>
3.1. OpenUP kerrokset.....	3
3.1.1. Mikroinkrementit.....	4
3.1.2. Iteraation elinkaari.....	5
3.1.3. Projektin elinkaari .....	6
3.2. OpenUP perusperiaatteet.....	8
<b>Lähteet .....</b>	<b>9</b>

## 1. Johdanto

OpenUP on iteratiivinen ja inkrementaalinen ohjelmistokehitysprosessi, joka on syntynyt avoimen ohjelmistokehityksen tuloksena Eclipse Process Framework projektissa. OpenUP on ketterä menetelmä, jonka juuret ovat Unified Process menetelmissä. OpenUP on muokattavissa ja laajennettavissa käyttämällä Eclipse Process Framework Composer työkalua.

## 2. OpenUP Historiaa

OpenUP prosessin juuret ovat IBM:n RUP:issa (Rational Unified Process). Joukko RUP:in kehittäjiä alkoi miettiä miten voitaisiin kehittää kevyempi malli RUP:ista [Kro07]. Lähtökohtana oli kehittää ketterä lähestymistapa RUP:iin, kuitenkin samalla hyödyntäen muiden ketterien menetelmien, kuten XP ja Scrum, parhaita käytäntöjä. Työ aloitettiin IBM:ssä, mutta hyvin pian todettiin, että tarvitaan laajempi joukko kehittämään sitä. Aloitettu työ lahjoitettiin 2006 Eclipse säätiölle osaksi Eclipse Process Frameworkia (EPF). Lahjoitus sisälsi OpenUP:n pohjalla olevia RUP:in osia ja teknologiaa.

OpenUP:n kehitystyötä jatkettiin osana Eclipse Process Framework (EPF) projektia. Aktiivisesti kehitystyössä oli mukana useita ihmisiä yli 20sta eri yhtiöstä [Gus08]. Kehittämisessä mukana olevilla oli taustaa mm. seuraavien menetelmien kehittämisestä: Agile RUP, Dynamic System Development Method (DSDM), Agile Model Driven Development (AMDD) ja Scrum. Kehittämisessä ei lähdetty tyhjältä pöydältä, vaan siinä pyrittiin löytämään yhteisiä tärkeitä asioita eri menetelmistä. Tuloksena päädyttiin karsimaan ja harmonisoimaan eri menetelmien käytäntöjä.

Tuloksena syyskuussa 2007 julkaistiin OpenUP 1.0. Jonka tavoitteena ei ole olla lopputuote, vaan ensimmäinen julkaisu kehityspolulla, joka jatkuu avoimen ohjelmistokehityksen proses-

sina EPF projektin alla [Kro07]. Tämän hetkinen versio OpenUP 1.5.0.1. julkaistiin 27.10.2008 [EPF08].

### 3. SPEM

SPEM (Software Process Engineering Meta-model) prosessimetamalli määrittelee formaalin kielen, jolla voi kuvata kehitysprosesseja. Malli on hyvin yleinen ja sillä voi kuvata minkä tahansa kehitysprosessin.

SPEM specificaatio julkaistiin marraskuussa 2007. SPEM määrittelee elementit, joilla kuvataan prosessi, samoin kuin elementit, joilla hallitaan ja jäsennetään tätä tietoa. Peruselementit, joilla tietoa hallitaan ovat: Method Library, Method Plug-in, Method Package, Process Package, Method Configuration [EPF08].

### 4. EPF ja EPFC

Eclipse Process Framework (EPF) on avoimen ohjelmistokehityksen projekti, jonka tavoitteena on kustomoitava ohjelmistokehitysprosessien mallinnuskehys, jossa mukana on malli-prosessi ja työkaluja. Sen on tarkoitus sopia usean erilaisen prosessin ja erilaisten kehitystyö-  
lien kanssa [EPF08].

Projektin tavoitteet

- o tarjota laajennettava kehys ja mallityökalut ohjelmistokehitysprosessien mallintamiseen. menetelmien ja prosessien luomiseen, menetelmäkirjastojen hallintaan ja muokkaamiseen sekä prosessien julkaisuun. (EPFC)
- o Tarjota laajennettava esimerkkiprosessi, joka sisältää kehityksen ja hallinnan näkökulman, tukee iteratiivista ja inkrementaalista kehittämistä. (OpenUP)

EPF projektin tuloksena on syntynyt Eclipse Process Framework Composer (EPFC) ja OpenUP.

EPFC työkalu prosessien mallintamiseen on rakennettu Eclipse alustalle. EPFC hyödyntää SPEM 2.0 prosessimetamallia prosessien mallintamisessa. EPFC:ää voi käyttää erilaisia projektityyppejä ja kehitysmalleja käyttävien prosessien mallintamiseen.

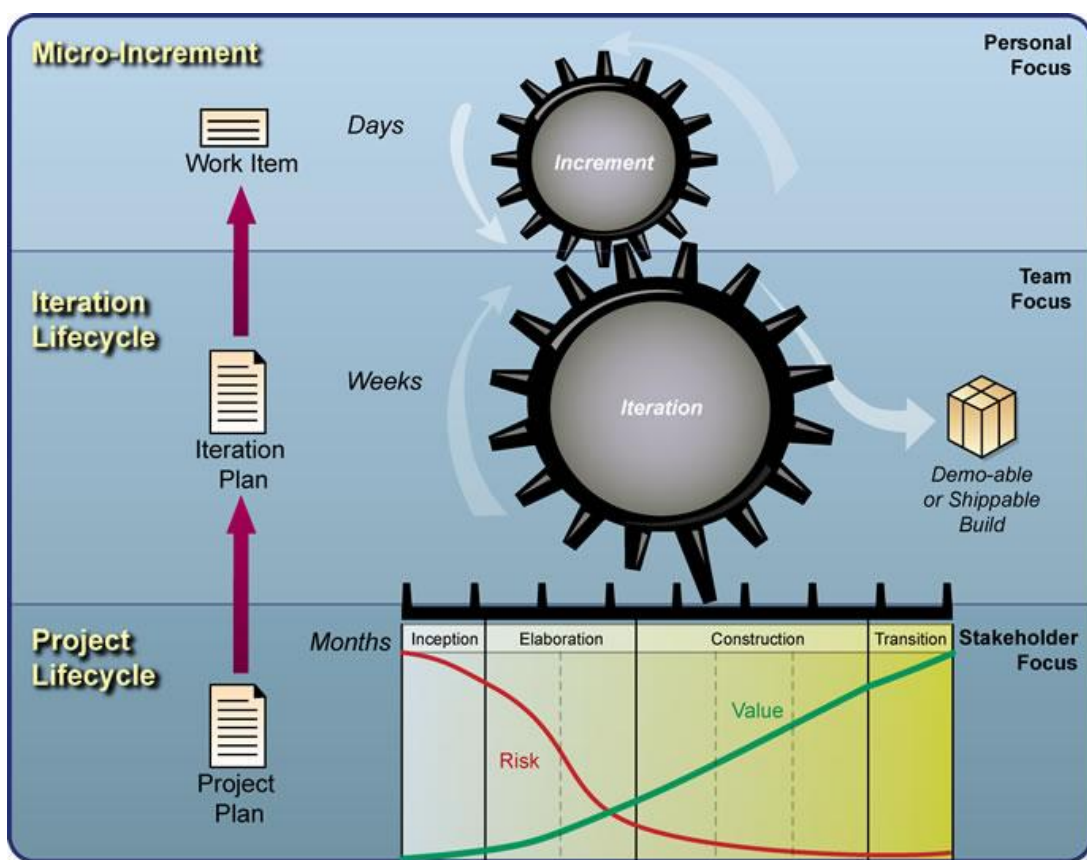
### **3. OpenUP menetelmä**

OpenUP on erittäin kevyt ohjelmistokehitys prosessi, jonka perusta on Unified Process (UP) ohjelmistokehitysprosessissa. OpenUP on samanaikaisesti minimaalinen, kokonainen ja laajennettava prosessi [Vel08]. OpenUP on minimaalinen, mutta riittävä, koska se sisältää vain keskeiset asiat ohjelmistokehitysprosessista. Kokonainen prosessi, koska se pystyy kuvaamaan koko prosessin järjestelmän rakentamiseksi. Laajennettava, koska sitä voidaan pitää perusprosessina, jota täydennetään tai laajennetaan tarpeen mukaan. [EPF08]

OpenUP on myös ketterä menetelmä. Samalla kun sen juuret ovat Unified Process:issa ja RUP:issa, niin OpenUP korostaa ketterien menetelmien filosofian mukaan ohjelmistokehityksen kiinteää yhteistyötä vaativaa luonnetta [Kro07].

#### **3.1. OpenUP kerrokset**

OpenUP yhdistää iteratiivisen ja inkrementaalisen lähestymistavan prosessin ohjattuun elinkaareen. OpenUP voidaan jakaa kolmeen eri kerrokseen. Nämä kerrokset ovat mikroinkrementit, iteraation elinkaari ja projektin elinkaari. Tasot on esitetty kuvassa 1.



Kuva1. OpenUP kerrokset [OpenUP].

### 3.1.1. Mikroinkrementit

Henkilötasolla OpenUP projekti on organisoitu mikroinkrementteina. Mikroinkrementti kuvaa pientä yksikköä työtä, joka tuottaa selkeän mitattavan edistymisaskelen projektiin. Tämä edistyminen on yleensä mitattavissa tunteina tai muutamina päivinä. Prosessin mukaan järjestelmää toteutetaan intensiivisessä yhteistyössä sitoutuneen ja itseohjautuvan tiimin voimin [OpenUP]. Mikroinkrementin käsite auttaa yksittäistä tiimin jäsentä osittamaan oman työnsä siten, että jokainen tuottaa mitattavaa arvoa tiimin osana. Mikroinkrementit tarjoavat erittäin nopean palautteen, joka ohjaa joustavaa päätöksentekoa jokaisen iteraation sisällä [Kru07].

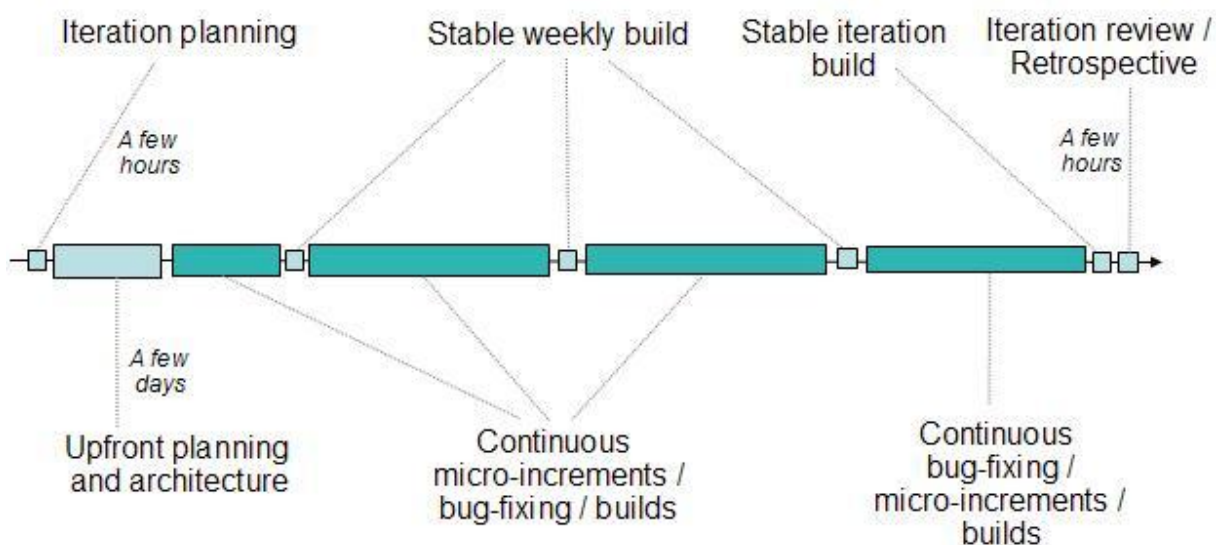
Mikroinkrementin pitää olla hyvin määritelty ja jokaisen mikroinkrementin päivittäistä etenemistä pitää pystyä seuraamaan. Mikroinkrementit tarkennetaan ja seurataan käyttäen työkohteita. Järjestelmä kehittyy mikroinkrementteina, useiden työkohteiden suorittamisen kautta. Mikroinkrementtien etenemistä seurataan tiimin kesken tapaamisissa tai yhteistyötyökalujen kautta. Avoimuus etenemisen seurannassa tiimin kesken tuo projektiin läpinäkyvyyttä ja tehostaa edelleen tiimin yhteistyötä.

### **3.1.2. Iteraation elinkaari**

OpenUP:n mukaan projekti jaetaan iteraatioihin. Iteraatio on suunniteltu, aikaikkunaan sidottu intervalli, joka tyypillisesti mitataan viikoissa. Iteraatioiden tarkoituksena on saada tiimi toimittamaan suunnitellulla tavalla inkrementaalista lisäarvoa projektin intressitahoille. Iteraatio-suunnitelma määrittää mitä toimitetaan iteraatiossa. Tulos on yleensä demottava tai toimitusvalmis käännöspaketti. Itseohjautuva OpenUP tiimi organisoii iteraation siten, että iteraation tavoitteet tulevat täytetyksi. Samalla tiimi sitoutuu toimittamaan iteraation päätteeksi iteraation tuotoksen. Iteraation tehtävät saadaan työkohteiden listasta. OpenUP määrittää iteraation elinkaaren, joka ohjaa miten mikroinkrementit edistävät järjestelmää inkrementaalisesti kohti iteraation tavoitetta [OpenUP].

Iteraation suunnittelu, arviointi ja etenemisen seuranta tehdään työkohteiden kautta. Iteraatio-suunnitelma laaditaan valitsemalla ensisijaiset työkohteet. Ketterät arviointitekniikat auttavat arvioimaan kuinka monta työkohdetta voidaan turvallisesti mahduttaa aikarajoitettuun iteraatioon [Kro07]. Kuvassa 2. on esitetty iteraation elinkaarimalli.





Kuva 2. Iteraation elinkaarimalli [OpenUP]

### 3.1.3. Projektin elinkaari

OpenUP jäsentää projektin elinkaaren neljään vaiheeseen: Aloitus-, tarkennus-, rakennus- ja siirtymävaihe. Projektin elinkaarimalli tarjoaa projektin intressitahoille ja tiimin jäsenille näkyvyyttä ja päätöspisteitä läpi koko projektin. Elinkaarimalli tuo mahdollisuuden tehokkaaseen valvontaan ja tilaisuuksia tehdä ns. jatketaan/ei jatketa päätöksiä. Projektisuunnitelma määrittää projektin elinkaarimallin [OpenUP].

Projektin elinkaarimalli tarjoaa intressitahoille kokonaiskuvan ja läpinäkyvyyttä projektiin. Se tarjoaa myös ohjausmekanismin, jonka avulla voidaan kontrolloida mm. projektin rahoitusta, laajuutta, riskejä ja sen tuottamaa lisäarvoa.

OpenUP organisoii iteraatiot vaiheisiin. Jokaisella vaiheella on erityinen tarkoitus ja virstanpylväsehto.

**Aloituvaihe.** Määritä laajuus ja tavoitteet projektille.

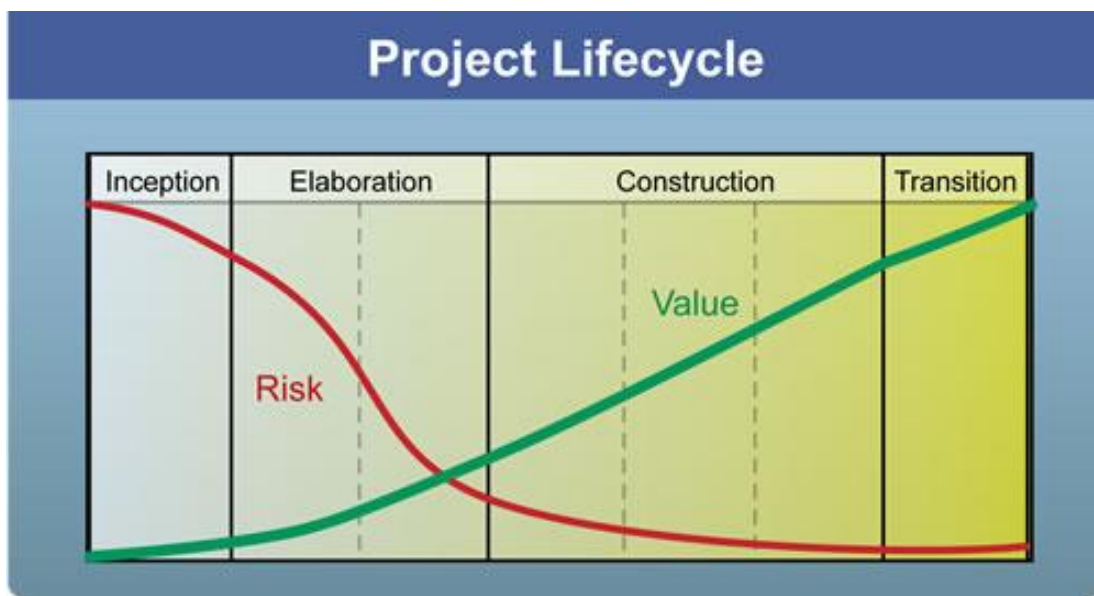
**Tarkennusvaihe.** Muodosta ymmärrys vaatimuksista ja toteuta suorittava arkkitehtuuri avainskenaarioita ja laatuvaatimuksia varten.

**Rakennusvaihe.** Toteuta järjestelmän toiminnallisuus.

**Siirtymävaihe.** Julkaise järjestelmä loppukäyttäjille.

Tämä projektin elinkaarimalli erottaa OpenUP:n monista muista ketteristä menetelmistä. Elinkaarimalli kiinnittää huomion projektin laajuuteen ja ratkaisuihin jo ennen laajempaa toteutusta, projektin alkupäässä aloitus- ja tarkennusvaiheessa. Tarkennusvaiheen lopussa on yleensä käytetty 20-25 % budjetista, mutta 30-40 % aikataulusta [Kro07].

Projektin elinkaarimalli tuo seurantaan ennen kaikkea kahteen intressitahoille tärkeimpään asiaan, riskien hallintaan ja arvon tuottamiseen intressitahoille. Vaiheistuksella pyritään laskemaan riskiä samalla kun seurataan intressitahojen saamaa hyötyä. Kuvassa 3. on esitetty riskien lasku aikaisessa vaiheessa projektin elinkaarta.



Kuva3. Projektin vaiheet [OpenUP]

### 3.2. OpenUP peruseriaatteet

OpenUP:n perustuu neljän peruseriaatteen varaan [BPS07]. Nämä kaikki neljä periaatetta ovat suoraan johdettavissa Ketterän manifestin, (Agile Manifesto), julistamasta ketterän kehityksen perusmääritelmästä [AGILE08]. Taulukossa 1. on rinnakkain esitetty OpenUP:n peruseriaatteet ja ketterän manifestin perusmääritelmän ydinkohdat.

<b>OpenUP:n neljä peruseriaatetta</b>	<b>Ketterä manifesti</b>
<b>Tee yhteistyötä</b> saavuttaaksesi yhteisen linjan ja ymmärryksen	<b>Yksilöitä ja vuorovaikutusta</b> enemmän kuin prosesseja ja työkaluja
<b>Keskity</b> arkkitehtuuriin aikaisin minimoidaksesi riskin ja organisoidaksesi kehitystyön	<b>Toimivaa sovellusta</b> enemmän kuin kokonaisvaltaista dokumentaatiota
<b>Tasapainota</b> vaatimukset ja rajoitteet tärkeysjärjestykseen, jotta maksimoidaan intressitahojen hyöty	<b>Asiakasyhteistyötä</b> enemmän kuin sopimusneuvotteluita
<b>Kehity</b> hankkimalla jatkuvasti palautetta ja parantamalla	<b>Muutokseen reagoimista</b> enemmän kuin suunnitelman noudattamista.

Taulukko 1. OpenUP ja Ketterä manifesti [Gus08].

## Lähteet

- [AGILE08] Manifesto for Agile Software Development <http://agilemanifesto.org/>  
[13.11.2008]
- [BPS07] Borg, A., Patel, M., Sandahl, K., Extending the OpenUP/Basic Requirements Discipline to Specify Capacity Requirements. *Requirements Engineering Conference, 2007. 15th IEEE International Volume , Issue , 15-19 Oct. 2007*, sivut 328 – 333.
- [EPF08] Eclipse Process Framework Project (EPF). <http://www.eclipse.org/epf/>  
[13.11.2008]
- [Gus08] Gustafsson, B., OpenUP – The Best of Two Worlds. *Methods & Tools Spring 2008*, sivut 21-32.
- [Kro07] Kroll, P., OpenUP in Nutshell. *The Rational Edge*.  
[<http://www.ibm.com/developerworks/rational/library/sep07/kroll/>]
- [OpenUP] OpenUP 1.5, publish created 21.08.2008.
- [Vel08] Velzen van, T. Skillful and maneuverable: OpenUP and the Eclipse Way. *The Rational Edge*.  
[<http://www.ibm.com/developerworks/rational/library/edge/08/jul08/vanVelzen/>]