

Ohjelmistoprosessimalleja

Seuraavissa kuvauksissa oletetaan, että projektissa ei ole tavoitelähtöisen kälisuunnittelun osaamista. Lopuksi palataan kysymykseen, mitä tapahtuu, jos käyttötapausselvitykset ja tavoitelähtöinen kälisuunnittelu lisätään prosessimalliin.

Alkuosan lähde:

Boehm B. W.,
A Spiral Model of Software Development and Enhancement.
IEEE Computer, Vol. 21, No. 5, May 1988, s. 61-72.
<http://www.sce.carleton.ca/faculty/ajila/4106-5006/Spiral.Model.Boehm.pdf>

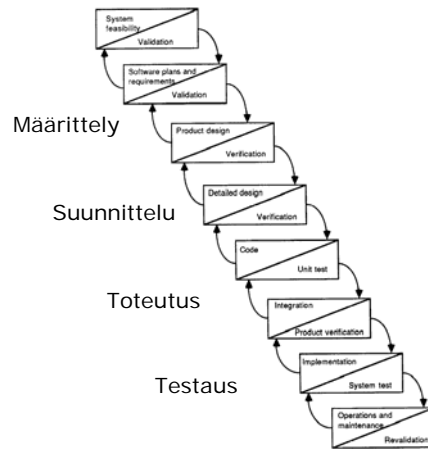
Copyright © 2004 / Sari A. Laakso

Prosessimallit "Koodaa ja korjaa" -malli

- Toistuvat vaiheet
 - Koodin kirjoittaminen
 - Ongelmien korjaaminen koodista
- Raportoituja ongelmia
 - Huonosti jäsennetyn koodin muuttaminen kallista
 - Korjauskierrosten tuloksena syntyvää huonosti jäsentynyttä koodia alkaa olla muutaman syklin jälkeen vaikeaa ja aikaavievää muuttaa
 - Tarvitaan *suunnitteluvaihe* ennen koodausta
 - Käyttäjiltä tulee lisää muutostarpeita koodiin
 - Koodattu toiminnallisuus ei vastaa käyttäjien tarpeita, minkä seurauksena tehtyä työtä joudutaan heittämään pois tai sitä joudutaan ainakin muuttamaan paljon
 - Tarvitaan *vaatimusanalyysivaihe* ennen suunnitteluvaihetta

Copyright © 2004 / Sari A. Laakso

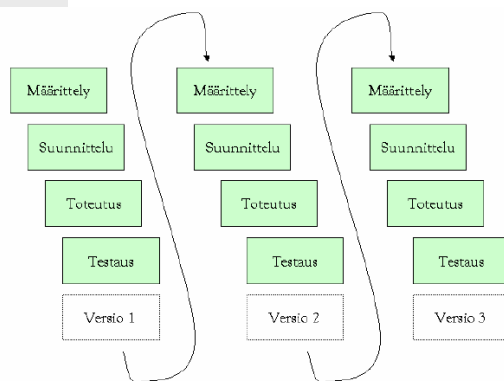
Prosessimallit Vesiputousmalli



Copyright © 2004 / Sari A. Laakso

- Erityisesti interaktiivisten järjestelmien kehittämisessä ongelma: yksityiskohtaiset dokumentit vaiheiden lopputiloina.
- Suunnitteluvaiheessa:
 - Perusteellinen dokumentointi myös huonosti ymmärretyistä kohdista, esim.
 - yksityiskohtaisesti kuvatut epämääräiset käsitteelliset ratkaisut (esim. laatikkokaavioita, navigointikarttoja) tai
 - huonosti ymmärrettyjä toteutusratkaisuja, joista laaditaan yksityiskohtaiset toteutussuunnitelmat.
 - => Käyttökelpotonta koodia, speksit ja koodi uusiksi myöh.
 - Mitä pitäisi tehdä? ↗
- Määrittelyvaiheessa:
 - Vaikea tuottaa kattavaa vaatimusjoukkoa.
 - Tulos näkyy vasta lopuksi!

Prosessimallit Evoluutiomalli



(Haikala, 1998)

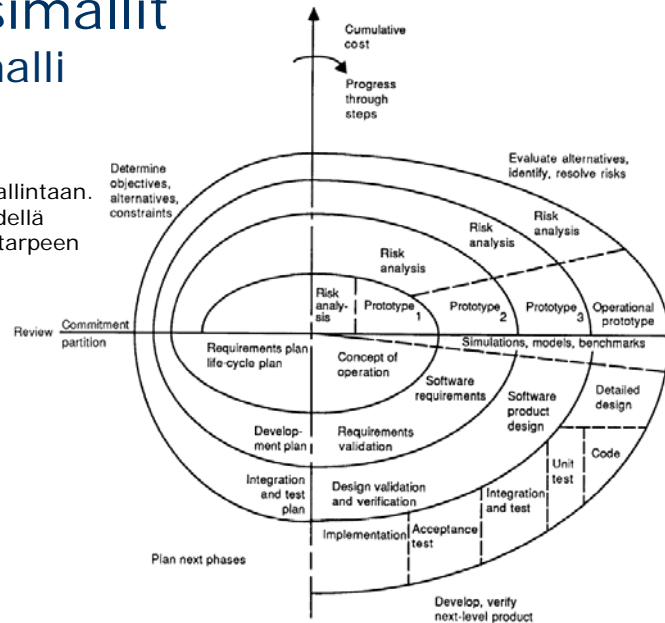
Copyright © 2004 / Sari A. Laakso

- Ohjelmistotuotetta laaditaan inkrementaalisesti: aluksi tehdään osa toiminnallisuutta, jonka päälle seuraavassa vaiheessa uutta jne.
- Lähteessä mallin sanotaan sopivan tilanteisiin, joissa käyttäjät kokevat: *"I can't tell you what I want, but I'll know when I see it."* – Missä ongelma oikeasti on?
- Raportoituja ongelmia mm.
 - Pitkän tähtäimen arkkitehtuurisuunnittelu puuttuu. Uusien lisäysten seurauksena koodista alkaa tulla jäsentymätöntä.
 - Koodatun version ongelmien korjaamisesta seuraa uudelleenkirjoitusta, kuten vesiputousmallissakin (mutta vähemmän ja paloittain).

Prosessimallit Spiraalimalli

(Boehm, 1988)

Painottuu riskienhallintaan.
Hyödyntää mm. edellä
kuvattuja malleja tarpeen
mukaan.



Copyright © 2004 / Sari A. Laakso

Spiraalimalli Vaihe 1

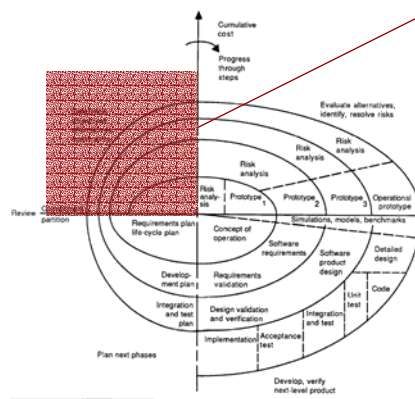


Figure 2. Spiral model of the software process.

Copyright © 2004 / Sari A. Laakso

- Jokainen spiraalin kierros alkaa määrittelemällä...
 - **tavoitteet** (esim. kaksinkertaistaa tuottavuus 5 vuodessa),
 - **ratkaisuvaihtoehdot** tavoitteen saavuttamiseksi (esim. valmiin ohjelman ostaminen, ohjelmiston kehittäminen itse, työkäytäntöjen muutos, ohjelman vaihtoehtoinen design A, design B),
 - vaihtoehtoihin liittyvät **rajoitteet** (esim. kustannukset 10 000 \$ per hlö, aikataulu).

Spiraalimalli Vaihe 2

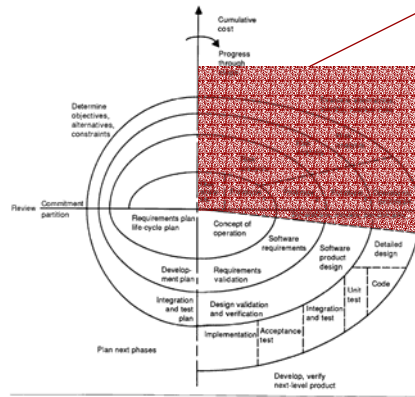


Figure 2. Spiral model of the software process.

Copyright © 2004 / Sari A. Laakso

- Ratkaisuvaihtoehtojen arviointi
 - Riskien paikantaminen
 - Panostaminen epävarmoihin, *riskialttiisiin* kohtiin:
 - Kokematon kehittäjä keskittyy jo ennestään hyvin tuntemiensa kohtien määrittelyyn (huono tulos)
 - Kokenut kehittäjä osaa keskittyä **selvittämään epävarmoja kohtia** mahdollisimman pitkälle
- Epävarmat kohdat saadaan ratkaistua *laitimalla prototyyppejä, simuloimalla, mallintamalla* jne.
- Jäljellä olevat riskit määräävät, mitä seuraavassa vaiheessa (ja seuraavilla kierroksilla) tehdään, ks. seuraavan sivun taulukko.

Spiraalimalli Jatkotoimenpiteet riskien ohjaamana

Käli- ja suorituskykyriskit dominoivat	Budjetti- ja aikatauluriskit dominoivat
Spiraalimalli painottuu kohti evoluutiomallia .	Jos käli- ja suorituskykyriskit on jo saatu hallintaan esim. aiempien prototyyppien avulla, spiraalimalli muuttuu kohti vesiputousmallia .
Suurimmista riskikohdista tehdään yksityiskohtaisia prototyyppejä ; konkreettisilla kokeiluilla saadaan selville, miten kannattaa jatkaa. (Määrittelyyn ja speksaukseen minimaalisesti resursseja.)	Jokaista määrittelyä seuraa validointivaihe ja seuraavan syklin suunnitteluvaihe.
Spiraalimalliin merkityt speksien laatimisvaiheet tiedostetaan, mutta näitä vaiheita ei suoriteta.	Prototyyppien laatiminen, simulointi, mallinnus ym. riskinvähentäminen tiedostetaan, mutta näitä vaiheita ei suoriteta (ei tarpeen).

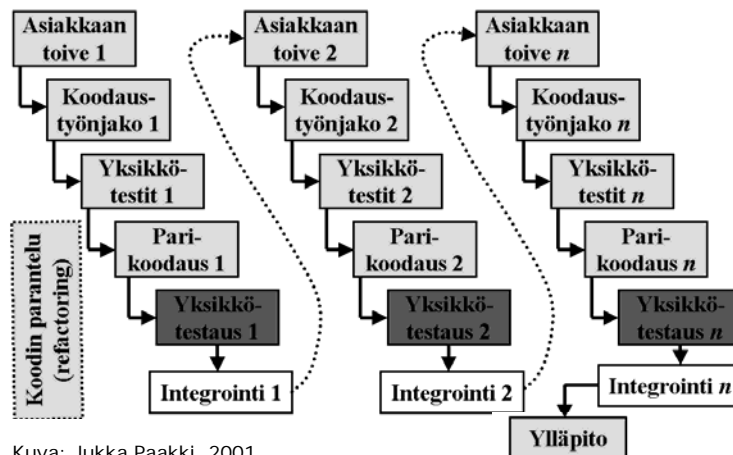
Copyright © 2004 / Sari A. Laakso

Vielä yksi prosessimalli: Extreme Programming (XP)

- XP:n lähtöoletus: Koska asiakkailta tulee aina paljon muutospyyntöjä projektin kuluessa, projektissa on varauduttava jatkuvaan vaatimusten muuttumiseen.
- Ratkaisu: Nopeatempoiset koodausyykliä, jotka tuottavat kokonaisia testattuja ohjelman versioita.
- Parin viikon pituiset koodausyykliä:
 - Asiakas kirjoittaa käyttäjätarinoita, jotka ovat sekä
 - toteutuksen lähtökohtana toimivia vaatimuksia että
 - hyväksymistestauksen testitapauksia.
 - Ohjelmoijat pilkkovat käyttäjätarinan ohjelmointitehtäviksi, jakavat työt ja kirjoittavat testit; koodaavat ja testaavat.
 - Koodattu ja testattu versio annetaan asiakkaalle, jolta koodaajat saavat palautetta.

Copyright © 2004 / Sari A. Laakso

Extreme Programming (XP) Vaiheet



XP:n käyttäjätarinat (user stories) Esimerkkejä 1/2

- User opens "My Companies" window and sees all the companies that he is assigned to. Then user clicks a company and sees projects of his that are for that company. This would replace the "My Projects" window.
- When a user creates a project, he selects a project "type" or template that creates a default set of action items for the project.
- User sets an action item's priority.
- Only people who can see a project can see a project's action items.

<http://c2.com/cgi/wiki?AtsUserStories>

Copyright © 2004 / Sari A. Laakso

XP:n käyttäjätarinat (user stories) Esimerkkejä 2/2

- A user marks a project as being owned by his group. (Question: can a user ever belong to more than one group?)
- Provide a box that will display the application name and version number for ATS, as well as tech. support contact info.
- A user wants access to the system, so he finds an ATS system administrator, who enters in the user's first name, last name, middle initial, email address, username (unique), and phone number. Users may be marked as "deleted." Users may also be marked as administrators.

<http://c2.com/cgi/wiki?AtsUserStories>

Copyright © 2004 / Sari A. Laakso

Extreme Programming (XP) Kälistuunnittelun lisäämisen vaikutus?

- Mitkä ovat XP:n keskeisimmät ongelmat?
- Toimisiko XP, jos
 - korvaisimme asiakkaiden kirjoittamat käyttäjätarinat käyttöliittymäsuunnittelijoiden laatimilla tavoitepohjaisilla **käyttötapauksilla** ja
 - käyttöliittymäsuunnittelija laatisi toteutettavalle käyttötapaukselle **kälistratkaisun** ennen kuin koodaajat jakaisivat käyttötapauksen osiin, laatisivat testit ja koodaisivat ko. toiminnallisuuden?

Copyright © 2004 / Sari A. Laakso

