Asiakas: Kjell Lemström Ohjaaja: Anita Kasari

pics98

Projektisuunnitelma

Samuli Kaski
Timo Koskinen
Jani Leinonen
Mika Pennanen

Helsinki 29. toukokuuta 1998

Ohjelmistotuotantoprojekti, kesä 1998
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

1 Tausta (MP)

Projektin asiakkaita ovat Kjell Lemström ja prof. Esko Ukkonen. Kesän `pics98`-projektissa parannellaan PICSearch v.2 -ohjelmistoa havaittujen virheiden ja parannusehdotusten pohjalta. PICSearch on kuvanhaku- ja kuvausalgoritmien kehittäjille suunnattu ohjelmisto, jolla voidaan hakea kuvia tietokannasta niiden sisällön perusteella.
PICSearch v.2 -ohjelmistoon lisätyn verkkotuen ansiosta voi tietokanta, johon kyselyt kohdistetaan, sijaita missäpäin Internet-verkkoa tahansa.

Vastaavia projekteja WWW:ssä:
1. IBM, Query by Image Content(QBIC)
`http://wwwqbic.almaden.ibm.com/`
2. MARS
`http://jadzia.ifp.uiuc.edu:8000/`
3. Virage
`http://www.virage.com/`

2 Tavoitteet ja rajaukset (SK)

Projektin ensisijaisena tavoitteena on parantaa nykyistä PICSearch -ohjelmistoa lisäämällä siihen uusia ominaisuuksia ja korjata siinä olevia implementointipuutteita. Syksyn 1997 ohjelmistotuotantoryhmältä jäi joitakin seikkoja toteuttamatta tai niitä ei toteutettu loppun asti. Näihin ja käytössä ilmenneisiin puutteisiin pyrkii tämä kesän 1998 ohjelmistotuotantoryhmä `pics98` puuttumaan.

Asiakas Kjell Lemström osallistuu ryhmäkokouksiin niin usein kuin vain mahdollista. Palautettavien dokumenttien tarkastamisen ja järjestettävien demojen seuraamisen ohella asiakas on myös ilmaissut halukkuutensa saada projektiryhmän sisällä välitettävät sähköpostit mikäli ne vain sisältävät projektiin ja sen läpivientiin liittyvää tietoa.

Projektin alkuvaiheessa on toimintasuunnitelmaan sisällytetty kaksi demotilaisuutta, mutta mikäli aika riittää ja demonstroitavaa on, projektiryhmä on valmis järjestämään asiakkaalle lisädemotilaisuuksia, jotta projektin aikaansaannoksia voitaisiin konkreettisesti arvioida ennen projektin päättymistä.

Projektin loppuvaiheessa projektin aikana tuotetut dokumentit ja ohjelmistopaketti luovutetaan asiakkaalle arvioitavaksi. Mikäli vielä tässä vaiheessa puutteita ilmenee, tekee projektiryhmä asianmukaiset korjaukset. Asiakkaan aktiivinen rooli projektin alkuvaiheista asti johtanee siihen, että projektin loppuvaiheessa asiakkaalle luovutettava työ täyttää asiakkaan toiveet ja ettei suurempiin korjaustoimenpiteisiin ole enää aihetta ryhtyä.

Projekiryhmässä käytetään dokumentointikielenä englantia. Poikkeuksena tähän ovat kuitenkin ryhmän sisäiset dokumentit, jotka kirjoitetaan suomeksi. Sisäisiä dokumentteja ovat projektisuunnitelma, vaatimusmäärittely, suunnitteludokumentti, kokouspäiväkirjat ja loppuraportti.

3 Ympäristö (TK)

PicSearch-ohjelmiston suunnitteluideologian keskeisenä osana on ollut ohjelmiston siirrettävyys ja sopeutuminen erilaisiin käyttöympäristöihin. `pics98`-projekti noudattaa tätä periaatetta. Koska projektin läpivientiin on varattu rajallisesti aikaa, määritellään kuitenkin tarkemmin se ympäristö, jossa ohjelmisto on tarkoitus toteuttaa.

3.1 Laite- ja ohjelmistoympäristö

Ohjelmisto tarvitsee täydellisesti toimiakseen TCP/IP-yhteyden, joten laitteiston on oltava kytkettynä verkkoon.

Ohjelmisto toteutetaan ensisijaisesti UNIX (Linux/Solaris) -käyttöjärjestelmälle. Lisävaatimuksina ovat Tcl/Tk-kirjasto ja graafinen XWindows-ikkunointijärjestelmä. Jos käytettävissä on tarvittavat työkalut ja tarpeeksi aikaa, tehdään toteutus myös Windows 95/NT -käyttöjärjestelmille.

Toteutus tehdään C++ ja Tcl/Tk -ohjelmointikielillä.

3.2 Arkkitehtuuri

Teknisenä yleisarkkitehtuurina käytetään asiakas-palvelin -mallia. Käyttäjän työasemassa toimiva asiakasohjelma voi olla yhteydessä joko samassa työasemassa tai verkkoyhteyden takana sijaitsevaan palvelinohjelmaan.

Ohjelmisto on suunnattu ensijaisesti asiantuntijakäyttöön, mutta käyttöliittymä pyritään suunnittelemaan sellaiseksi, että eritoten asiakasohjelman käyttö on mahdollisimman suoraviivaista ja auttaa näin käyttäjää nopeasti omaksumaan ohjelman toimintatavan. Toteutuksessa noudatetaan yleisesti tunnettuja käyttöliittymästandardeja, ja vältetään näistä poikkeavia erikoisratkaisuja. Näitä suunnitteluperiaatteita noudatetaan sekä ohjelmiston asiakas- että palvelinosuuden toteutuksessa.

4 Projektin organisaatio (TK)

`pics98`-projektiryhmän kotisivu on osoitteessa
`http://www.cs.helsinki.fi/group/pics98/`.

Projektiryhmän jäsenet ovat

Kjell Lemström (KL), asiakas
Email: klemstro@cs.helsinki.fi

Anita Kasari (AK), ohjaaja
Email: ahalmine@cs.helsinki.fi

Lilli Nenonen (LN), projektin vastuuhenkilö
Email: lilli.nenonen@cs.helsinki.fi

Samuli Kaski (SK)
Email: samuli.kaski@helsinki.fi

Timo Koskinen (TK)
Email: timo.t.koskinen@helsinki.fi

Jani Leinonen (JL), 1. projektipäällikkö
Email: jani.leinonen@cs.helsinki.fi

Mika Pennanen (MP), 2. projektipäällikkö
Email: mika.pennanen@helsinki.fi

Projektipäällikkyys on jaettu siten, että Jani Leinonen toimii projektipäällikkönä projektin alusta sen puoliväliin asti, jonka jälkeen tehtävä siirtyy Mika Pennaselle.

5 Toimintasuunnitelma (JL)

`pics98`-projektin kokonaiskesto on noin kolme ja puoli kuukautta. Projekti alkoi 18.5.1998 ja päättyy 31.8.1998.

5.1 Projektin aikataulu

Projektisuunnittelma 21.5. - 1.6.

- Ryhmän sisäinen tarkastus 28.5.
- Luovutus projektivastaavan, asiakkaan ja ohjaajan tarkistettavaksi 29.5.
- Projektivastaavan, asiakkaan ja ohjaajan palaute 1.6.

Määrittelydokumentti 28.5. - 2.7.

- Ykkösversion luovutus asiakkaalle ja ohjaajalle (ja projektivastaavalle) 12.6.
- Asiakkaan ja ohjaajan (ja projektivastaavan) palaute 15.6. (klo 16)
- Vaatimusmäärittelydokumentin korjaus 18.6. - 2.7.
- Katselmointi 9.7.

Suunnitteludokumentti 22.6. - 30.7.

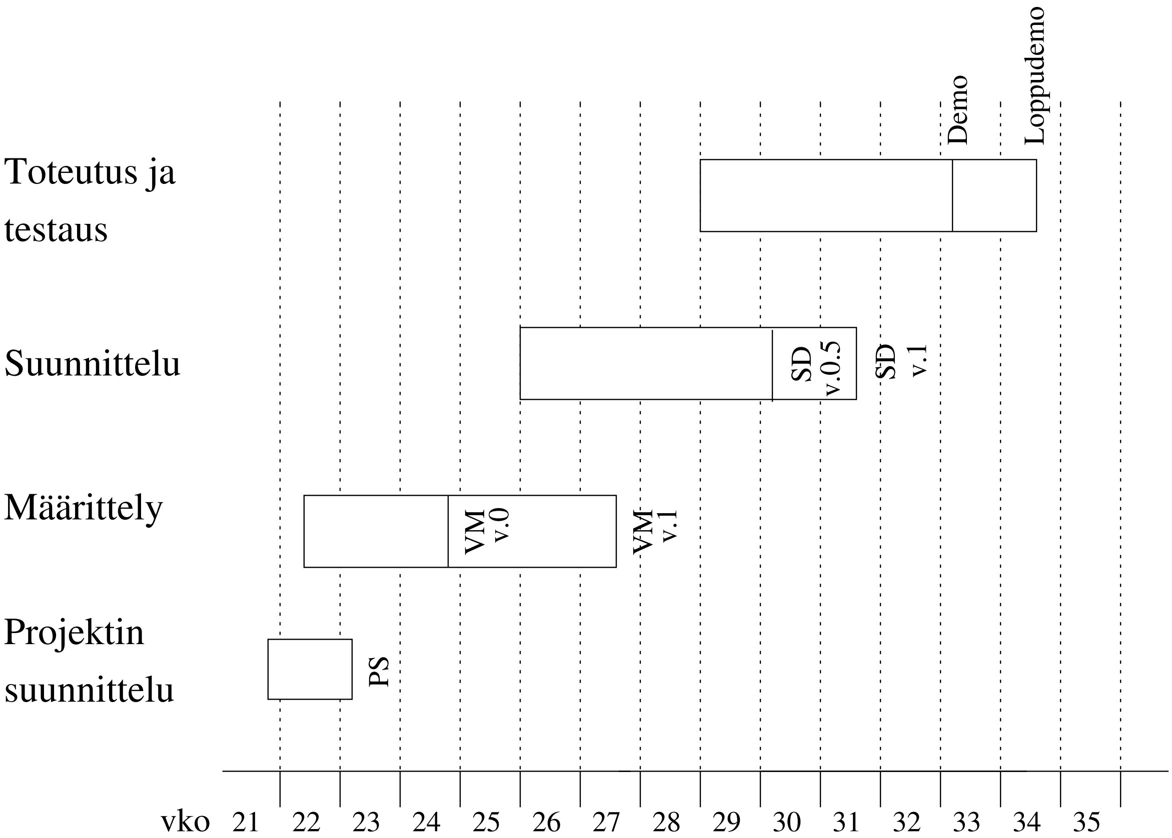
- Ylinollaversion (versio 0.5) luovutus asiakkaalle 20.7.
- Asiakkaan palaute 23.7
- Suunnitteludokumentin korjaus 23.7. - 30.7.

Toteutus ja testaus 13.7. - 20.8.

- Testaus 6.8.-20.8.
- 1. demo 10.8.
- Loppudemo 20.8.

Dokumentointia (viimeistely) 10.8.-31.8.

- Ylläpito- ja käyttöohje
- Toteutusdokumentti
- Loppuraportti



Kuva 1: Gantt-kaavio projektin kulusta

6 Menetelmät ja standardit (SK)

Projektijäsenet kunnioittavat edellisen ryhmän aikaansaannoksia ja mikäli jonkun tietyn menetelmän tai ohjelmiston käyttäminen on tämän tavoitteen edellytyksenä, toimivat ryhmän jäsenet parhaalla näkemällään tavalla.

6.1 Dokumentointi

Ohjelmistotuotantoprojektin aikana tuotettavien dokumenttien kirjoittaminen pyritään teettämään tasaisesti ryhmän neljän opiskelijajäsenen kesken. Mikäli dokumentointivastaavaa ei työmäärän puitteissa tarvita, toimii dokumenttien hallitsijana projektipäällikkö. Jos kuitenkin dokumenttien hallintaan vaadittavat työmäärät ovat merkittäviä, on projektipäällikön dokumentointivelvotteita luonnollisesti vähennettävä.

Ryhmä käyttää CVS -versionhallintaohjelmistoa, jolla luotiin projektin dokumenteille ja lähdekoodeille oma hakemistopuunsa. CVS -versionhallintaohjelmisto mahdollistaa kaikkien ryhmäjäsenten samanaikaisen työskentelyn projektin dokumenttien ja lähdekoodien parissa.

Dokumenttien hallintavastuu kuuluu kaikille ryhmän jäsenille. Projektipäällikön rooli on kuitenkin viime kädessä toimia siltana asiakkaaseen päin. Vaikka työmäärät pyritään jakamaan tasaisesti ryhmän jäsenten kesken, on projektipäällikön tehtävänä huolehtia siitä että ryhmän jäsenet kasaavat dokumentit ja toimittavat ne ohjaajalle, valvojalle ja asiakkaalle.

Virallisten yleiseen käyttöön tarkoitettujen dokumenttien lisäksi pyritään ryhmän sisäiset dokumentit jättämään tuleville ryhmille ja ulkopuolisille tutkittavaksi. Tähän tarkoitukseen sopii parhaiten ryhmälle muodostettu WWW-sivu, jonka ylläpitoon ei olla erikseen määrätty ketään ryhmän jäsenistä.

6.2 Dokumentointivälineet

Dokumentointiin käytetään LaTeX-ladontaohjelmistoa sen joustavuuden ja käyttöjärjestelmäriippumattomuuden vuoksi. Päätökseen vaikuttivat myös ohjaajan ja asiakkaan omat hyvät kokemukset LaTeX:sta.

Tyylipohjana ryhmän kaikissa dokumenteissa toimii ohjaajan ryhmän käyttöön toimittama pohja, joka kuitenkin elää projektin edetessä kun kaikki ryhmäjäsenet tulevat tutuiksi LaTeXin kanssa. Tämä projektisuunnitelma on kuitenkin esimerkkinä siitä, millaiselta loppulliset dokumentit tulevat näyttämään.

Edellinen ryhmä on käyttänyt dokumentoinnissaan HTML-kieltä ja DOC++ -työkalua, jolla lähdekoodista saadaan automaattisesti generoitua rajapinnoista ja kommenteista muodostuva dokumentti. Kesän ryhmä tulee toteutusvaiheessa käyttämään samaa työkalua.

6.3 Ohjelmointi

Ohjelmoinnin, kuten muunkin kierrätettävän materiaalin, yhteydessä pyritään noudattamaan edellisien ryhmien käyttämiä tapoja ja ohjeita. Edellisen ryhmän noudattamat ohjelmointiohjeet löytyvät osoitteesta
`http://www.cs.Helsinki.FI/research/picsearch/koodausohjeet.html`,
joita siis tämäkin ryhmä pyrkii noudattamaan. Tässä vaiheessa on kuitenkin jo ilmennyt että aikaisempien ryhmien käyttämä kahden merkin sisennys on liian epäselvää luettavaksi ja ylläpidettäväksi, joten ryhmä päätyi neljän merkin sisennykseen. Vanhoja lähdekoodeja ei lähdetä uudelleensisentämään. Vain uutta luo-

taessa pyritään noudattamaan neljän merkin sisennystä.

6.4 Tuotteenhallinta

Ryhmän kaikki tuotokset pyritään pitämään CVS-versionhallintaohjelmiston tarkan valvonnan piirissä. Ensisijaisena tarkoituksena on kuitenkin, että vähintään dokumentit ja lähdekoodit pidetään CVS-puussa. Kaikilta ryhmän jäseniltä edellytetään CVS-ohjelmiston käytön osaamista ja ohjelmiston asettamien sääntöjen noudattamista.

7 Riskienhallinta (MP)

Riskien hallinnan pohjaksi on otettu B.W. Boehmin [1] artikkeli, joka on hyvä johdanto ohjelmistoprojektien riskienhallintaan. Boehmin mukaan riskienhallinta on kuusivaiheinen prosessi, jonka vaiheet ovat 1) riskien tunnistaminen 2) riskianalyysi 3) riskien priorisointi 4) riskienhallinnan suunnittelu 5) riskien poistaminen sekä 6) riskien seuranta. Tässä projektissa riskienhallinta toteutetaan kevennettynä versiona, jonka päävaiheet ovat projektin suunnitteluvaiheessa tehtävä riskien tunnistaminen ja arviointi sekä koko projektin ajan jatkuva riskien seuranta.

7.1 Riskien tunnistaminen ja arviointi

Boehm luettelee artikkelissaan ohjelmistoprojektien riskeistä "kymmenen kärjessä" -listan:
1. Henkilöstölliset riskit
2. Epärealistiset aikataulut ja budjetit
3. Väärän toiminnallisuuden ja ominaisuuksien toteuttaminen
4. Väärän käyttöliittymän toteuttaminen
5. "Kultaaminen"
6. Jatkuva vaatimusmäärittelyn muuttaminen
7. Ulkopuolisten komponenttien heikkoudet
8. Puutteet alihankinnassa
9. Reaaliaikavaatimukset
10. Tietojenkäsittelyopilliset rajoitukset

Riskit on seuraavassa järjestetty arvioidun vaikutuksen mukaan, vaikutus on laskettu arvioimalla riskin toteutumisen todennäköisyys asteikolla 0–1 sekä sen vakavuus asteikolla 0–10 ja kertomalla luvut keskenään.

1. Epärealistiset aikataulut ja budjetit: 0.7 x 8 = 5.6
Ryhmän ja projektipäällikön kokemus ohjelmistoprojektin läpiviemisestä on vaatimaton, joten virhearviointien todennäköisyys on suuri. Riskiin varaudutaan priorisoimalla hyvin ominaisuudet vaatimus- ja toiminnallisessa määrittelyssä, mikä mahdollistaa ei-tärkeiden komponenttien toteutuksesta luopumisen riskin toteutuessa.

Mahdollinen itsenäinen työskentely saattaa hajaannuttaa ryhmää ja aiheuttaa epätietoisuutta projektin etenemisestä. Riskiin varaudutaan järjestämällä yhteisiä kokoontumisia säännöllisesti sekä informoimalla muuttuneista asioista ja tilanteista.

2. Jatkuva vaatimusmäärittelyn muuttaminen: 0.6 x 7 = 4.2
Tyypillisesti vaatimusmäärittely elää projektin aikana. Alkuvaiheessa ideoita syntyy runsaasti ja ne kirjataan mielellään vaatimuksiksi toteutettavalle ohjelmistolle. Jotta tämä sinänsä positiivinen ajatusten ja ideoiden kypsyminen ei pääsisi vaikuttamaan aikatauluihin tai hämärtämään projektin tavoitetta, pyritään tunnistamaan heti alussa ohjelmiston varsinainen ydintoiminnallisuus, jonka päälle voidaan joustavasti tuoda uutta toiminnallisuutta. Ohjelmistoa parantavien ominaisuuksien priorisoiminen on tärkeää.

3. Henkilöstölliset riskit: 0.7 x 5 = 3.5
Projektissa käytetään samoja toteutustyökaluja kuin aikaisemmissa kahdessa projektiryhmässä. Tcl/Tk on suurimmalle osalle ryhmästä täysin tuntematon, samaten C++-koodausrutiini puuttuu monelta. Koska projektissa käytetään työkaluja, jotka ovat suurimmalle osalle ryhmäläisistä uusia, on varsin todennäköistä, että jossakin vaiheessa uusien asioiden opiskelu tulee hidastamaan projektin etenemistä. Riskiin varaudutaan tunnistamalla opiskelutarpeet ajoissa ja vaikutusta voidaan lieventää pitämällä vaatimusmäärittely riittävän suppeana ja selkeänä välttämättömien ominaisuuksien osalta. Mahdollisuuksien mukaan asioiden selvittämiseen käytetään asiantuntevia ulkopuolisia tahoja (esim. edellisten ryhmien jäseniä).

4. Kultaaminen: 0.5 x 5 = 2.5
Vähemmän tärkeiden toimintojen puolinainen toteuttaminen tulee mahdolliseksi, mikäli ei riittävän tarkasti priorisoida vaatimusmäärittelyssä tärkeitä ja ei-tärkeitä ominaisuuksia. Laaditaan yhdessä asiakkaan kanssa suunnitelma siitä, mitkä toiminnot on oltava valmiina ennen kuin lisätoiminnallisuutta voidaan alkaa toteuttaa.

5. Ulkopuolisten komponenttien heikkoudet: 0.4 x 3 = 1.2
Ohjelmiston kehitysympäristö ja -työkalut voivat aiheuttaa ongelmia projektin edis-

tymisessä. Tcl/Tk ja LaTeX ovat useimmille tuntemattomia, joten niiden osalta arvioiminen on hankalaa. Mahdollisen C++-kääntäjän myöhäisempi saaminen Windows95 -ympäristöön saattaa kuluttaa aikaa tutustumiseen.

6. Väärän toiminnallisuuden ja ominaisuuksien toteuttaminen: 0.2 x 5 = 1.0
Koska projektin pohjana on valmis ohjelmisto ja projektin tarkoituksena ei ole tehdä suuria toiminnallisia muutoksia, riskin toteutuminen ei ole kovinkaan todennäköistä. Myös jatkuva yhteistyö asiakkaan kanssa on avainasemassa projektin suunnan pitämisessä oikeana.

7. Tietojenkäsittelyopilliset rajoitukset: 0.2 x 5 = 1.0
Kehitettävä ohjelmisto ei todennäköisesti ole algoritmisesti vaativa. Mikäli riski kuitenkin toteutuu, pyritään kyseessä olevan ominaisuuden tärkeys arvioimaan ja arvion perusteella joko ohjaamaan lisää resursseja tehtävään tai jättämään ominaisuus pois ohjelmistosta. Toimintojen ennalta priorisointi on oleellista, jotta tärkeyden arviointi onnistuu tarvittaessa nopeasti.

8. Väärän käyttöliittymän toteuttaminen: 0.1 x 8 = 0.8
Syntyvän ohjelmiston käyttökelpoisuuden kannalta on erittäin tärkeää, että käyttöliittymä on hyvä. Tämän vuoksi käyttöliittymäriskiä tullaan seuraamaan jatkuvasti siten, että pyritään järjestämään mahdollisimman todenmukaista käytettävyystestausta ryhmän sisällä ja mahdollisuuksien mukaan ulkopuolisella avulla. Riski on kuitenkin pieni, sillä käyttöliittymämuutokset tehdään jo olemassaolevaan käyttöliittymään.

9. Reaaliaikavaatimukset: 0.1 x 2 = 0.2
Reaaliaikasuorituskyky ei sinäänsä ole ohjelmiston pääominaisuus. Edellisessä NetPICSearch-projektissa otettiin jo huomioon mm. tietoliikenteen osalta verkkoviive, jota pyrittiin minimoimaan. `pics98`-projektin tavoitteena on ohjelmiston paranteleminen.

10. Puutteet alihankinnassa: 0
Projektissa ei käytetä alihankintaa, joten tämä kohta jää riskienhallinnan ulkopuolelle.

Näiden lähinnä projektisuunnittelun ja työskentelymenetelmien parantamisen keinoin hallittavien riskien lisäksi projektissa on luonnollisesti vahingoista tms. odottamattomista tapahtumista johtuvia riskejä, kuten sairastumiset, tietojen katoamiset laite- tai ohjelmistovikojen vuoksi jne. Näistä henkilöstöön kohdistuvat riskit voidaan hallita pitkälle samoin keinoin kuin monet ylläolevistakin riskeistä, eli priorisoimalla tehtävät riittävän hyvin. Projektin jäsenet käyttävät hyväkseen versionhallintajärjestelmä CVS:ää estääkseen tiedostojen päivitysongelmat. Tietojen menetykset hallitaan yhteisesti sovitulla varmuuskopioinnilla. Vikaantuneiden laitteiden korvaamisesta uusilla on vastuussa tietojenkäsittelytieteen laitos.

7.2 Riskienseuranta

Boehm esittää artikkelissaan riskien seurantamenetelmäksi projektin 10 vakavimman säännöllistä arvioimista. Tämä voidaan tehdä listaamalla ko. riskit tärkeysjärjestyksessä säännöllisin väliajoin ja pitämällä kirjaa siitä, kuinka monta kertaa kukin riski on listalla esiintynyt ja millä sijalla se on edellisellä kerralla ollut. Projektin aikana listaa seuraamalla voidaan havaita pitkään listalla viipyvät ja nopeasti nousevat riskit, mikä helpottaa huomattavasti riskienhallintaa. Kesäprojektin lyhyen toteutuksen (n. 3kk.) takia, projektipäällikkö hoitaa pääosin riskien seurantaa. Jokainen projektin jäsen on kuitenkin vastuussa projektin riskien hallinnasta. Jäsenet seuraavat myös itsenäisesti projektin etenemistä ja ilmoittaa mahdollisista projektiin liittyvistä riskeistä projektipäällikölle.

Projektin riskikartoitukset ja -arvioinnit suoritetaan ainakin projektin puolivälissä. Havaittaessa projektin etenemistä vaikeuttavia tekijöitä ryhmä suorittaa projektipäällikön johdolla tarvittavat riskianalyysit.

Viitteet

[1] Boehm B. W. "Software Risk Management: Principles and Practices". IEEE Software, January 1991, pp. 32-41.