

Ohjeita projektityön tekemiseksi

Copyright (c) 1994-2006, Joensuun yliopisto, Tietojenkäsittelytieteen ja tilastotieteen laitos

0) 18.09.2006 Markku Tukiainen, alustava

1) 16.10.2006 Markku Tukiainen, päivitetty dokumenttiluetteloa vastaavaksi

1. Johdanto

Projektityön tarkoituksena on antaa konkreettinen käsitys kokonaisen ohjelmistohankkeen läpiviemisestä ympäristössä, jossa pyritään laadukkaaseen lopputuotteeseen noudattamalla vastuuntuntoisen ohjelmistotuotannon periaatteita.

Projektityönä tehdään jokin todellinen ohjelmointiprojekti, jonka tuloksen on tarkoitus olla käytettävissä sellaisenaan, tai lähes sellaisenaan, tuotantokäytössä. Siten projektityössä joudutaan toimimaan ”todellisessa” ympäristössä vastakohtana normaalien harjoitustöiden ”leikkiympäristölle”.

Projektityöt tehdään ryhmitöinä. Tämä edellyttää, että jokaisen osallistujan on voitava olla koko syyslukukauden (periodit 1 ja 2) ajan käytettävissä. Projektityöt vaativat lisäksi ajoittain runsastakin työpanosta, joten kurssin lisäksi ei samalle ajalle kannata ottaa muita runsaasti työtä vaativia opintojaksoja tai muita hankkeita.

Tämä dokumentti sisältää joukon projektityön tekemisessä tarvittavia ohjeita. Teksti perustuu kurssin aiempien vastuuhenkilöiden Jorma Sajaniemen, Pirkko Voutilaisen, Raimo Raskin ja Ramin Miraftabin ansiokkaaseen pohjatekstiin, jota on sitten muokattu uusien työprosessien mukaiseksi. Alkuperäinen Jorma Sajaniemen teksti pohjaa osittain Elaine Kantin artikkeliin A Semester Course in Software Engineering (Software Engineering Notes 6,4 (August 1981) 52-76).

2. Aikataulu

Projektityössä laaditaan asiakkaan käyttöön tuleva ohjelmisto ja siihen liittyvä joukko dokumentteja. Dokumentit ja niihin liittyvät tarkastuspöytäkirjat säilytetään ryhmäkohtaisissa hakemistoissa, joista työn ohjaaja käy hakemassa tarvitsemansa tiedot arvostelua varten.

Projektiin liittyy joukko vaiheita, tarkempi kuvaus vaiheista, niihin liittyvistä aktiviteeteista, rooleista ja artefakteista löytyy UPEDU-dokumentista /UPEDU, 2006/. Artefakteista useat ovat dokumentteja, joita tarkastelee ja katselmoi joukko projektin osallisia. Tästä syystä valmistuneesta dokumentista on syytä lähettää ilmoitus tarvittaville henkilöille (yleensä asiakas, kurssin vastuopettaja, projektin tukiohjaaja). Ilmoitus on lähetettävä sähköpostina ja siinä on mainittava myös dokumentin tiedostonimi. Mikäli dokumenttiin liittyy paperiliitteitä, niin ne on jätettävä samassa yhteydessä samoille henkilöille.

2.1 Projektiin liittyvät iteraatiot ja niiden päättymispäivämäärät

Inception	22/09/2006
Elaboration 1	06/10/2006
Elaboration 2	20/10/2006
Construction 1	03/11/2006
Construction 2	10/11/2006
Construction 3	24/11/2006
Transition 1	08/12/2006
Transition 2	15/12/2006

Seuraavassa dokumenttiluettelossa numeroilla merkityt dokumentit ovat arvosteltavia dokumentteja, asiakaskommunikaation toteutumisen vuoksi dokumenteista lihavoidut on jätettävä kokonaisuudessaan paperiversiona asiakkaalle.

Vaiheiheet, vaihetulokset (suluissa englanniksi) ja niihin liittyvät dokumentit

*** inception 22.9. (objectives)**

- Visio

- Riskilista
- Iteraationsuunnitelmat elaboration 1 ja 2 varten

- 1. Sanasto

- Käyttötapausmäärittelyt (tärkeimmät toimijat ja käyttötapaukset määritelty ja tapahtumienkulut niille hahmoteltu)
- Järjestelmän throw-away prototyyppi
- Projektiloki
- CVS-varasto toiminnassa, muutospyyntöpohja valmiina

*** elaboration 1 6.10. (architecture definition/sketch)**

- 2. Vaatimusmäärittely, Täydentävät määritykset

- Käyttötapausmäärittelyt (siltä osin kuin järkevää)
- Käyttöliittymän prototyyppi
- 3. Ohjelmistonkehityssuunnitelma (projektisuunnitelma)

*** elaboration 2 20.10. (architecture & design)**

- 4. Ohjelmiston arkkitehtuuri (realisaatiomäärittely arkkitehtuurisesti merkittävälle käyttötapausk-

sille valmis)

- 5. *Suunnittelumalli* (melko täydellinen, sisältää arkkitehtuurin, joitain yksityiskohtia päätetään vielä construction-vaiheessa)
- Käyttötapausmäärittelyt (noin 80%:sesti valmis)
- Konfiguraationhallintasuunnitelma
- Iteraatio-suunnitelma (construction 1,2,3)
- 6. *Implementaatiomalli* (hyvin alustava, tärkeimmät komponentit identifioitu ja protoiltu)

*** construction 1 3.11. (alpha 1)**

- 7. *Testaussuunnitelma*
- Implementaatiomalli (siltä osin kuin järkevää)
- Käyttötapausten realisaatiomäärittely lopuille käyttötapausten (siltä osin kuin järkevää)
- 8. *Käyttöohje* (siltä osin kuin järkevää)
- alpha 1
 - testauksessa syntyvät dokumentit

*** construction 2 10.11. (alpha 2)**

- Implementaatiomalli (siltä osin kuin järkevää)
- Käyttötapausten realisaatiomäärittely lopuille käyttötapausten (siltä osin kuin järkevää)
- Käyttöohje (siltä osin kuin järkevää)
- alpha 2
 - testauksessa syntyvät dokumentit

*** construction 3 24.11. (initial operational capability / beta)**

- Käyttötapausten realisaatiomäärittely
- Implementaatiomalli
- käyttöohje (siltä osin kuin järkevää)
- Iteraatio-suunnitelma (transition 1,2)
- beta
 - testauksessa syntyvät dokumentit

*** transition 1 8.12. (release candidate)**

- käyttöohje
- release candidate
- testauksessa syntyvät dokumentit

*** transition 2 15.12. (product release)**

- kaikki valmiina
- Projektin loppuraportti (post mortem)
- Opintojakson loppuarviointi

Asiakkaalle tehdään ohjelmiston ensiasennus viimeistään perjantaina 8.12.2006. Laadittavan ohjelmiston toiminta tarkastetaan hyväksymistestauksessa, jonka johtaa kurssin luennoitsija ja jonka sisältöä on kuvattu myöhemmin tässä dokumentissa. Hyväksymistestauksen järjestämisestä sovitaan erikseen. Se on kuitenkin pidettävä **viimeistään** perjantaina 15.12.2006.

3. Arvostelu

Projektityön kokonaisarvosana riippuu toteutuksen (koodin) laadusta ja harjoitustyön tuloksesta (asiakkaan kannalta), dokumenttien laadusta ja lisäksi työskentelyn kurinalaisuudesta.

Itse järjestelmä arvostellaan pisteillä 0..90, (toteutuksen kannalta 0..40, asiakkaan kannalta 0..50). Kukin arvosteltava dokumentti (yhteensä siis 8 kpl) arvostellaan asteikolla 0..10., samoin kuin työskentely arvostellaan asteikolla 0..30. Dokumentit arvostellaan viimeisinä versioinaan.

Pisteiden maksimiarvot määräytyvät siis seuraavasti:

Osa	Max p
Koodi	40
Asiakas	50
Dokumentit	80
Työskentely	30

Kurssin kokonaisarvosana riippuu arvostelujen summasta seuraavasti:

pisteet	arvosana
100-119	1
120-139	2
140-159	3
160-179	4
180-200	5

4. Ryhmän organisointi

Ryhmien jäsenistä sovitaan töiden jakamisen yhteydessä.

Kukin ryhmä voi organisoida haluamallaan tavalla. Ryhmän on kuitenkin nimettävä projektipäällikkö, muutosten hallinnan vastuhenkilö ja versiohallinnan vastuhenkilö. Ryhmän ei ole pakko nimetä muita organisaatioon liittyviä rooleja joskin kokemus on opettanut, että selkeä työnjako ja vastuiden määrittely on tarpeen työskentelyn onnistumiselle. Rooleja voidaan siirtää ryhmän sisällä henkilöltä toiselle työskentelyn aikana.

Yksittäisille tehtäville on syytä aina nimetä vastuhenkilö (vaikka muutkin osallistuisivat tehtävän tekemiseen), jonka on huolehdittava siitä, että työ tulee ajallaan valmiiksi.

HUOM: Kaikki projektin tehtävät määritellään dotProjektin avulla. Ajankäytön kirjaus suoritetaan näiden ennalta määriteltyjen tehtävien perusteella.

Projektiin osallistuvat myös ohjaajat ja asiakas. Lisäksi kullakin ryhmällä on partneriryhmä, jonka dokumentteihin ryhmä perehtyy. Eri osapuolien tehtävät ovat seuraavat.

Asiakas:

- ¥ määrittelytyöhön osallistuminen
- ¥ vaatimusmäärittelyn, toiminnallisen määrittelyn ja käyttöohjeen hyväksyminen
- ¥ ryhmän esittämiin kysymyksiin vastaaminen
- ¥ mahdollisesti tarvittavien erikoisresurssien (laitteet, ohjelmistot) järjestäminen

Projektipäällikkö:

- ¥ dokumenttien syntymisestä vastaaminen
- ¥ dokumentteihin mahdollisesti liittyvän paperimateriaalin toimittaminen ohjaajille sekä partneriryhmän jäsenille
- ¥ dokumenttien toimittaminen asiakkaalle paperiversioiden
- ¥ yhteydenpito ohjaajaan
- ¥ yhteydenpito asiakkaaseen
- ¥ projektin etenemisestä huolehtiminen

Ryhmän jäsenet (myös projektipäällikkö):

- ¥ dokumenttien ja ohjelmien tekeminen ja testaaminen
- ¥ ajankäytön kirjaaminen ja tiedottaminen
- ¥ yhteisiin kokouksiin osallistuminen
- ¥ partneriryhmän dokumentteihin perehtyminen ja niiden arvioiminen yhteiskokouksissa

Ohjaaja:

- ¥ ryhmän työskentelyn avustaminen
- ¥ tarvittavien resurssien järjestäminen
- ¥ yhteydenpito asiakkaaseen
- ¥ merkittävien muutosten hyväksyminen
- ¥ odottamattomien tilanteiden hoitaminen

Ohjaajat yhdessä:

- ¥ tehdyn työn ja syntyneiden tulosten arvostelu

Ryhmän on pidettävä asiakkaaseen jatkuvaa yhteyttä. Kohdassa "Aikataulu" on kuvattu, mitkä dokumentit on toimitettava asiakkaalle. Lisäksi asiakkaalle on toimitettava järjestelmästä testiversioita sitä mukaa kuin niitä valmistuu. Testiversioiden mukana on oltava lista niistä piirteistä, jotka eivät vielä toimi (jotta asiakas ei turhaan raportoi niistä ja jotta hänelle ei

syntyisi mielikuvaa, että järjestelmä on täynnä virheitä). Testiversioita on syytä toimittaa myös ohjaajalle - jopa selvästi tiheämmin kuin asiakkaalle.

5. Ajankäytön seuranta

Jokaisen on pidettävä kirjaa henkilökohtaisesta ajankäytöstään. Kirjanpidosta on selvittävä minkä tyyppisiin tehtäviin aikaa on kulunut, minä päivinä ja kuinka paljon. Lisäksi on hyvä, jos mukana on henkilökohtaisia arvioita ongelmakohdista ja töiden mielekkyydestä.

Tehtävien suunnittelu ja ajankäytön seuranta suoritetaan dotProject-ohjelmalla.

6. Versiohallinta ja muutosten hallinta

Dokumentit ja ohjelmakoodi on asetettava CVS-versiohallinnan piiriin.

Tuotedokumenttien ja ohjelmakoodin versiotunnuksen ensimmäistä numeroa on vaihdettava silloin, kun vaatimusmäärittelyssä, suunnittelumallissa tai käyttöohjeessa tapahtuu olennainen (käyttäjälle näkyvä) muutos. Versiotunnuksen ensimmäisen numeron vaihtaminen on versiohallinnan vastuuhenkilön tehtävä.

Työskentelydokumenttien (ohjelmistonkehityssuunnitelma, testausraportti, iteraatiosuunnitelmat) versiotunnuksen ensimmäistä numeroa ei vaihdeta koskaan.

Käytettävä versiohallintamenetelmä on kuvattava projektisuunnitelmassa.

Muutosten hallinta on hoidettava liitteessä 2 kuvatulla tavalla kirjaamalla puutteet mantisin avulla. Kustakin kirjatusta puutteesta muodostetaan erillinen tehtävä.

7. Dokumenttien kirjoittamisesta

Projektityöhön sisältyy paljon kirjoittamista. Kirjoitustyyli ei suoranaisesti vaikuta harjoitustyön arvosteluun, mutta hyvin kirjoitettu dokumentti selittää asiat usein paremmin kuin huonosti kirjoitettu.

Dokumenttien pituus ei ole sama asia kuin laatu. Jos dokumentti on selvästi pitempi kuin mitä suosituksessa on mainittu ja pituus johtuu yksityiskohtien runsaudesta, niin asia on aivan hyvä. Jos pituus sen sijaan johtuu huonosta rakenteesta, toistosta ja merkityksettömistä kuvauksista "järjestelmän erinomaisesta modulaarisuudesta ja poikkeuksellisesta laajennettavuudesta", niin arvostelu tulee olemaan rankkaa.

Kullakin dokumentilla tulee olla yksi toimittaja, joka on vastuussa dokumentin valmistumisesta ajallaan sekä dokumentin johdonmukaisuudesta ja ulkoasusta. Dokumentin kirjoittamiseen osallistuu yleensä useita kirjoittajia. Toimittajan velvollisuus on tällöin tarkastaa kaikkien kirjoittamat osuudet ja yhdenmukaistaa ne. Jokaiselle dokumentille (opintojakson loppuarviointia lukuunottamatta) on lisäksi järjestettävä tarkastus liitteessä 1 kuvatulla tavalla.

Kunkin dokumentin loppuun on merkittävä selkeästi dokumentin toimittaja sekä eri kirjoittajat ja heidän osuutensa dokumentista. Lisäksi jokaisesta dokumentista on ilmentävä työn nimi, dokumentin nimi ja versio sekä päivämäärä.

Eri dokumenteille on olemassa erilaisia sisällysluettelon esimerkkejä, mutta niitä ei ole syytä noudattaa orjallisesti. Sisällysluettelo onkin yleensä syytä muodostaa kulloisenkin tarpeen mukaiseksi.

Dokumentit on tehtävä HTML-muotoon. HTML:n on oltava myös lähdekoodisena luettavaa.

Mikäli dokumenteissa tarvitaan kuvia, ei niiden piirtämisen kannata tuhata paljoa aikaa. Kuvat voi piirtää aivan hyvin käsin, mikäli se on ajankäytön kannalta tarkoituksenmukaista.

Sisällysluetteloon ei tarvitse laittaa sivunumeroita, koska niiden ylläpitäminen on hidasta.

Dokumentteihin ja ohjelmiin laitetaan Copyright-merkintä tämän dokumentin alussa olevan mallin mukaisesti.

8. Throw-away proto

Ennen varsinaisen työn aloittamista tehdään valitulla toteutusvälineellä järjestelmästä yksinkertainen demoluonteinen prototyyppi. Prototyypin ei tarvitse toimia kuin muutamilta osiltaan eikä sitä arvostella. Prototyypin tekemisen tarkoituksena on perehdyttää tekijät käytettävään välineeseen ja sen mahdollisuuksiin (jotta esimerkiksi vaatimusmäärittelyssä ei tulisi luvattua liian monimutkaisia asioita). Lisäksi prototyyppiä on syytä esitellä asiakkaalle, jotta hänelle syntyy konkreettinen mielikuva järjestelmän toimintatavoista.

Prototyypin ominaisuudet ja sen tekemisen aikana tehdyt tärkeimmät havainnot kirjataan projektilokiin.

Jatkokehitysprojekteissa ei rakenneta erillistä prototyyppiä, vaan perehdytään vanhan järjestelmän toimintaan.

9. Vaatimusmäärittely ja täydentävät määrittelyt

Vaatimusmäärittelyssä kuvataan ne yleiset ominaisuudet, jotka laadittavalla järjestelmällä halutaan olevan. Tietosisältö ja toiminnot kuvataan vain yleisellä tasolla välttämättä näyttöihin liittyvää terminologiaa.

Dokumentissa kuvataan:

- ¥ järjestelmän yleiset tavoitteet
- ¥ järjestelmän kohdealueen erikoistermit
- ¥ järjestelmän käsittelemät tiedot
- ¥ järjestelmään liittyvät toiminnot ja näytöt yms. mahdollisimman yksityiskohtaisesti (käyttäjän kannalta ajateltuina)
- ¥ järjestelmän tulostamat raportit mahdollisimman yksityiskohtaisesti
- ¥ järjestelmään sisältyvät muut ohjelmat mahdollisimman yksityiskohtaisesti (syötteet, tulokset, käsittelysäännöt)

Toiminnalliset vaatimukset dokumentoidaan käyttötapauskaavion avulla. Kustakin käyttötapauksesta laaditaan myös tekstimuotoinen esitys.

Toiminnallisten vaatimusten lisäksi kuvataan järjestelmän toimintaympäristölle asetettavat vaatimukset ja järjestelmän yleiset ominaisuudet. Vaatimusmäärittelyssä on kuvattava seuraavat asiat:

A Toiminnallisuus

- ¥ tietosisältö ja toimintovaatimukset (käyttäjän kannalta)
- ¥ yhteensopivuus (muihin järjestelmiin nähden)
- ¥ sovellusalueen standardien noudattaminen
- ¥ turvallisuus (luvatonta käyttöä vastaan)
- ¥ toimintaympäristö (laitteisto, muut ohjelmistot)

B Luotettavuus

- ¥ kypsyys (virhetiheydellä mitattuna)
- ¥ vikasietoisuus (laitteisto/ohjelmistovirheisiin)
- ¥ toipumiskyky (laitteisto/ohjelmistovirheistä)

C Käytettävyys

- ¥ ymmärrettävyys (käsitteistön “luonnollisuus”)
- ¥ opittavuus
- ¥ operoitavuus (käytön vaatima työpanos)

D Tehokkuus

- ¥ ajankäyttö (vasteaikoineen)
- ¥ muiden resurssien käyttö

E Ylläpidettävyys

- ¥ analysoitavuus (vian etsinnässä)
- ¥ muunneltavuus (muutosten yhteydessä)
- ¥ stabiilisuus (muutosten aiheuttamien yllättävien sivuvaikutusten suhteen)
- ¥ testattavuus

F Siirrettävyys

- ¥ liitettävyys (toisiin määriteltyihin ohjelmistoihin)
- ¥ asennettavuus (työmääränä)
- ¥ siirrettävyyteen liittyvien standardien noudattaminen
- ¥ korvaavuus (mahdollisuus käyttää toisen määritellyn ohjelmiston tilalla)

Vaatimukset on ilmaistava mitattavissa tai todennettavissa olevina määrittäjinä. Jokainen vaatimus on ilmaistava yksityiskohtaisesti ja tarkasti. Jos jonkin seikan suhteen ei ole vaatimuksia, niin tämä on ilmaistava selkeästi.

Vaatimusmäärittelyn ohjeitus on 4-8 sivua.

10. Projektisuunnitelma

Projektisuunnitelma antaa perustan varsinaisen työn hallinnalle ja valvonnalle. Suunnitelmassa kuvataan projektin tehtävät, tarvittavat resurssit, aikataulu sekä laadunvarmistamisen ja edistymisen seurantamenetelmät.

Projektisuunnitelmassa työ jaetaan tehtäviin, jotka kuvataan dotProject-ohjelmalla.

Projektisuunnitelmassa on kuvattava myös laadunvarmistus ja versiohallinta (joita on käsitelty muualla tässä ohjeessa).

Harjoitustyön puitteissa ei välttämättä ole mahdollista tehdä haluttua järjestelmää kokonaisuudessaan. Tämän takia projektisuunnitelmassa on esitettävä neljä eri toteutusvaihtoehtoa:

- ¥ Ydin, jossa on vain ne piirteet, jotka ehdottomasti tarvitaan, jotta järjestelmällä voitaisiin tehdä ylipäätään mitään
- ¥ Kohtuuton, mutta käyttökelpoinen järjestelmä (joka tullaan toteuttamaan harjoitustyönä)
- ¥ Järjestelmä, joka tehtäisiin mikäli aikaa olisi käytettävissä riittävästi
- ¥ Erinomainen järjestelmä, jossa on kaikki erikoispiirteet, joita ylipäätään voisi toivoa

Edellisistä ainoastaan toinen vaihtoehto kuvataan yksityiskohtaisesti järjestelmän kehityksen aikana.

Projektsuunnitelman ohjepituus on 8-15 sivua.

11. Suunnittelumalli

12. Implementaatiomalli

Implementaatiomallissa kuvataan järjestelmän yleisrakenne, sen pääosat ja näiden väliset yhteydet. Pääosista kuvataan niiden toiminnallinen luonne ja annetaan hahmotelma teknisestä toteutuksesta.

Implementaatiomallin tulee olla riittävän yksityiskohtainen, olennaista on, että vaatimusmäärittely, arkkitehtuurin kuvaus, suunnittelumalli ja implementaatiomalli kuvaavat yhdessä ohjelmien toiminnan sekä ulkoisesti että sisäisesti. Näiden dokumenttien avulla pitää kenen tahansa pätevän atk-ammattilaisen pystyä tuottamaan käyttöohje, testaussuunnitelma ja itse ohjelmisto.

Suunnittelumallissa kuvataan suunnittelussa ja ohjelmoinnissa noudatettavat periaatteet ja menetelmät. Näihin kuuluvat myös ohjelmointityyliin liittyvät kysymykset ja tiedostojen, funktioiden yms. nimentäperiaatteet.

Moduuli- ja ohjelmatason kuvauksia joudutaan käytännössä tekemään työn edetessä,.

Implementaatiomallin ja suunnittelumallin ohjepituus on 25-40 sivua.

13. Käyttöohje

Käyttöohje on tarkoitettu peruskäyttäjälle, jonka tulee sen avulla pystyä käyttämään järjestelmään kuuluvia ohjelmia sekä ymmärtämään järjestelmän ja käyttäjän varsinaisten työtehtävien välinen yhteys.

Käyttöohjeen lukija ei välttämättä hallitse atk-tekniikkaa. Siten kuvauksessa on vältettävä atk-erikoistermien käyttöä ja asioita on mahdollisuuksien mukaan selvennettävä esimerkeillä.

Käyttöohjetta on voitava lukea ilman mitään muuta materiaalia. Siten käyttöohjeessa ei saa viitata esimerkiksi toiminnalliseen määrittelyyn, vaan mahdollisesti tarvittavat osat on kopioitava käyttöohjeeseen.

Käyttöohje voidaan rakentaa periaatteessa kahdella tavalla: kädestä pitäen johdattelevaksi alkeisoppikirjaksi tai kaikki yksityiskohdat tarkasti luettelevaksi käsikirjaksi. Projektityössä joudutaan tyytymään pelkästään jälkimmäiseen muotoon.

Käyttöohjeessa tulee olla hyvät apuneuvot tietojen hakemiseksi. Esimerkiksi sisällysluettelon tulee olla kuvaava. Se voi matkia järjestelmän päävalikkoja tai olla käyttäjän työnkulun mukainen. Lisäksi olisi hyvä, jos käyttöohjeen lopussa olisi aakkosjärjestyksessä oleva hakusanaluettelo, jonka avulla käyttäjä löytää kaikki ne sivut, joilla hakusana esiintyy.

Virheilmoituksista kuvataan (aakkos)järjestyksessä olevana luettelona itse virheilmoitustekstin lisäksi ne toimenpiteet, jotka järjestelmä on ilmoituksen antamisen lisäksi tehnyt. Ilmoituksen syistä kerrotaan paitsi välitön syy (joka normaalisti selviää jo itse ilmoitustekstistä) myös tavanomaiset välilliset syyt, joista johtuen virhetilanteeseen on päädytty. Lisäksi kerrotaan tavat, joilla virhetilanteesta voidaan toipua.

Käyttöohjeessa on varottava passiivin käyttöä. Ilmaisuihin "näytölle kirjoitetaan uudet tiedot" ei anna lukijalle minkäänlaista vihjetä siitä, tekeekö kyseisen työn tietokone vai käyttäjä itse.

Käyttöohje voidaan (asiakkaan toivomuksesta) laatia esim. Word-dokumentiksi. Yleensä käyttöohje on myös online-ohjeena järjestelmässä.

Käyttöohjeen ohjepituus on 20-35 sivua.

14. Testaussuunnitelma

Testauksen tarkoituksena on etsiä virheitä ohjelmasta tai moduulista. Testaussuunnitelman tarkoitus on osoittaa, mitkä asiat on tarkoitus testata ja miten, sekä antaa kriteerit testauksen lopettamiselle. Testaussuunnitelmaa tarvitaan yhtä hyvin testattaessa jotakin osaa ensimmäistä kertaa kuin testattaessa osan uutta versiota.

Testauksen suorittamisessa on kaksi mahdollisuutta. Joissakin tapauksissa on mahdollista luoda erillinen testausohjelmisto ohjelman tai moduulin testaamiseksi. Tällaisen ohjelman tapauksessa testiaineisto on helppo pitää yhdessä paikassa ja tuloksista syntyvää tiedostoa voidaan helposti vertailla toivottuun tulokseen.

Toinen mahdollisuus on suorittaa testaus käsin, jolloin testitapauksista on pidettävä kirjaa.

Testaustavasta riippumatta voidaan testauksessa soveltaa erilaisia tekniikoita, joita käsitellään Ohjelmistotuotannon kurssilla. Projektityön yhteydessä testauksessa on sovellettava mahdollisimman monia eri tekniikoita.

Testaussuunnitelmassa esitetään testattavat asiat luettelona, josta käy ilmi tarvittavien toimenpiteiden lisäksi syy, miksi kyseistä seikkaa testataan (esim. viite toteutussuunnitelmaan). Lisäksi suunnitelmassa on kuvattava hyväksymisperusteet silloin kun ne eivät ole ilmeisiä.

Testaussuunnitelmassa on oltava erilliset osat moduulitestausta ja integrointitestausta varten.

Testaussuunnitelmassa on myös kuvattava menettelyt testaamisen suorittamista, virheiden raportointia ja niiden korjaamista varten.

Testaussuunnitelman ohjepituus on 15-25 sivua.

15. Testausraportti

Testausraporttiin kirjataan mitä (yleensä testaussuunnitelman mukaisia kohtia) on testattu, millä versiolla, koska ja kuka testauksen on tehnyt ja mikä on ollut testauksen tulos kussakin yksittäisessä kohdassa. Testausraportista on käytävä selkeästi ilmi se, miksi testaaminen on voitu lopettaa.

Testausraportin tulee kuvata testauksen eteneminen ja siihen tulee sisältyä ohjelmien ja moduulien eri versioiden kaikki testaukset.

Testausraportille ei voi antaa ohjepituutta, koska sen pituus riippuu testauskertojen määrästä. Yhden testauskerran tulosten esittämisen ohjepituus on 2 sivua: yksi sivu testauskohteille ja toinen havainnoille.

16. Projektin loppuraportti

Projektin loppuraportissa kuvataan projektin kulku ja sen tärkeimmät tunnusluvut. Erityisesti on syytä vertailla projektin toteutumista suhteessa projektisuunnitelmaan (projektisuunnitelman

kaikkien kohtien osalta).

Loppuraportissa tulee käsitellä seuraavia asioita:

- ¥ lyhyt kuvaus projektin etenemisestä
- ¥ kohdatut ongelmat
- ¥ lopputuloksen kuvaus
- ¥ tuloksen käyttökelpoisuus asiakkaalle
- ¥ aikataulussa pysyminen ja siihen liittyneet ongelmat; yhteenveto myöhästymisistä
- ¥ suunnitellut ja toteutuneet työmäärät tehtävittäin ja henkilöittäin; yhteenveto työmäärien alituksista ja ylityksistä
- ¥ ohjelmiston koko (ohjelmariveinä ja toimintopisteinä)
- ¥ kirjoitetut ohjelmarivimäärät henkilöittäin (ja syyt sille, että eivät jakaudu likimain tasaisesti)

17. Opintojakson loppuarviointi

Opintojakson loppuarvioinnilla on merkitystä paitsi kirjoittajille itselleen (“mitä voisin tästä oppia”) myös kurssin jatkosuunnittelulle.

Loppuarvioinnissa tulee käsitellä seuraavia asioita:

Suunnitteluprosessi: miten hyvin suunnittelu onnistui ja mitkä olivat pahimpia ongelmia, mitkä asiat tekisit seuraavalla kerralla samoin / eri tavalla

- ¥ Testaus: oliko testaussuunnitelma onnistunut, oliko testaus oletettua vaikeampaa
- ¥ Hallinto: millainen organisaatio ryhmällä oli, oliko se onnistunut ratkaisu, miten kurssin yleinen ohjaus toimi
- ¥ Ajankäyttö: yhteenveto (tuntien tarkkuudella) omasta ajankäytöstä jaoteltuna eri tehtäviin käytettyyn aikaan, arvio ajankäytön järkevyydestä, olisiko johonkin pitänyt käyttää enemmän aikaa
- ¥ Laitteistot: miten laitteistot toimivat, olivatko laitteistot ja ohjelmistot asianmukaisia
- ¥ Seuraava kurssi: ohjeita ja vihjeitä seuraavan kurssin pitäjälle ja osallistujille

Opintojakson loppuarvioinnille ei ole ohjepituutta.

Jokainen osallistuja kirjoittaa itsenäisesti oman loppuarviointinsa ja lähettää sen sähköpostina luennoijalle. Loppuarviointeja ei laiteta ryhmän dokumenttihakemistoon eikä niitä jaeta muille.

18. Hyväksymistestaus

Hyväksymistestauksessa asiakas varmistuu siitä, että valmis järjestelmä on vaatimusmäärittelyn ja toiminnallisen määrittelyn mukainen. Hyväksymistestaus tehdään kurssin luennoitsijan johdolla ja ohjaajien, asiakkaan sekä tekijöiden läsnäollessa. Hyväksymistestauksessa käydään läpi seuraavanlaisia asioita:

- ¥ tekijät demonstroivat järjestelmän toimintaakäydään läpi vaatimusmäärittely kohta kohdalta ja selvitetään vaatimusten toteutuminen valmiissa järjestelmässä
- ¥ käydään läpi järjestelmän dokumentit
- ¥ käydään läpi muutostenhallinnan tila
- ¥ käydään läpi versiohallinnan tila

Koska asiakkaalle on toimitettu testiversioita jo järjestelmän laatimisen aikana, on asiakas voinut varmistua järjestelmän virheettömyydestä jo ennen hyväksymistestausta.

Liite 1: Tarkastusmenettely

Tarkastusmenettelyn tarkoitus on etsiä puutteita ohjelmista ja dokumenteista mahdollisimman aikaisessa vaiheessa, jolloin puutteiden korjaaminen on vielä halvinta.

Tarkastusmenettely kohdistetaan kaikkiin muihin dokumentteihin paitsi Opintojakson loppuarviointiin. Lisäksi ohjelmakoodi tarkastetaan pareittain. Ohessa on esimerkinomaiset tarkastuslistat kullekin tarkastettavalle dokumentille erikseen ja lisäksi yksi tarkastuslista ohjelmamoduulien tarkastamista varten. Jokaisen tarkastajan tulee aina lisätä tarkastuslistaan kaksi uutta kohtaa.

Tarkastustilaisuuden tulokset kirjataan erityiselle pöytäkirjalomakkeelle, joka tallennetaan projektihakemistoon.

Huom: Löydetyt puutteet kirjataan mantisilla sopivina tehtäväkokonaisuuksina!

Dokumenttien tarkastus

Jokaisella tarkastuksella on valvoja, joka ei saa olla kukaan tarkastettavan työn tekijöistä (paitsi jos ryhmän kaikki jäsenet ovat osallistuneet työhön).

Tarkastukseen osallistuu koko ryhmä. Tarkastettava materiaali on jaettava osallistujille vähintään kaksi päivää ennen tarkastustilaisuutta. Jokainen käy läpi tarkastettavan materiaalin tarkastuslistan avulla ja kokoa huomautuksensa erityiselle tarkastuskertomuslomakkeelle. Tarkastukseen ja/tai tarkastustilaisuuteen voi osallistua myös yksi tai useampi partneriryhmän jäsen.

Tarkastustilaisuudessa valvoja varmistaa, että kaikki ovat täyttäneet tarkastuskertomuslomakkeensa. Jos valmistelu on puutteellista, niin tarkastustilaisuus on siirrettävä myöhempään ajankohtaan. Tämän jälkeen käydään tarkastettava materiaali läpi kohta kohdalta ja jokainen esittää huomautuksensa asianomaisen kohdan tullessa vastaan. Koko ajan valvoja pitää huolen siitä, että virheitä vain etsitään ja paikannetaan mutta niitä ei ryhdytä korjaamaan.

Lopuksi valvoja päättää tarvitaanko uusintatarkastus vai ei.

Ohjelmamoduulien tarkastus

Dokumenttien ryhmäkohtaisen tarkastuksen lisäksi suoritetaan ohjelmamoduulien tarkastus pareittain. Tarkastukseen osallistuu moduulin tekijä ja kumppanina joku toinen saman ryhmän jäsen. Tarkastustilaisuudessa moduulin tekijä selittää moduulin yksityiskohdat kumppanille. Kumppani tekee kysymyksiä moduulista ja tekijä antaa vastaukset. Tekijä ja kumppani tarkastavat yhdessä moduulin tarkastuslistan avulla. Tekijä korjaa havaitut puutteet, jonka jälkeen kumppani vielä tarkastaa, että vaaditut korjaukset on tehty.

Tarkastuskertomus

Ryhmä: _____

Tarkastuksen kohde: _____

Päiväys: _____.____.____

Tarkastaja: _____

Ajankäyttö: __ h __. __. ____

__ h __. __. ____

__ h __. __. ____

Virheiden yms. lkm:

__ kpl suuria (voisi aiheuttaa järjestelmän hylkäämisen)

__ kpl keskikokoisia (voisi aiheuttaa vikaraportin)

__ kpl pieniä

Tarkastustilaisuuden pöytäkirja

Ryhmä: _____

Tarkastuksen kohde: _____

Kohteen koko riveinä: ____

Päiväys: __. __. ____

Valvoja: _____

Tekijät: _____

Tarkastajat, heidän ajankäyttönsä ja tarkastuslistan uudet kohdat:

Nimi: _____ h

Nimi: _____ h

Nimi: _____ h

Nimi: _____ h

Virheiden yms. lkm:

__ kpl suuria

__ kpl keskikokoisia

__ kpl pieniä

Uusintarkastus:

__ tarvitaan

__ ei tarvita

Tarkastuslista vaatimusmäärittelyä varten

1. Onko dokumentti ulkoasultaan muodollisesti kunnossa:

¥ Otsikkotiedot

¥ Versiohallinnan vaatimat tiedot

- ¥ Toimittaja ja kirjoittajat osuukseineen
 - ¥ Onko dokumentin eri kohdat helposti viitattavissa muista dokumenteista
2. Ovatko kaikki tarvittavat asiat mukana:
 - ¥ Ovatko kaikki työohjeessa mainitut kohdat mukana
 - ¥ Ovatko kaikki asiakkaan esilletuomat vaatimukset mukana
 3. Vastaako dokumentti aiempia dokumentteja:
 - ¥ Dokumentin vertaaminen mahdollisesti jo hyväksyttyihin dokumentteihin
 - ¥ Ovatko viittaukset muihin dokumentteihin kunnossa
 4. Ovatko kohdat sisäisesti kunnossa:
 - ¥ Onko käyttöympäristö kuvattu käyttöjärjestelmän versionumeroa myöten
 - ¥ Ovatko kaikki vaatimukset selvästi yksilöityjä
 - ¥ Voidaanko jokaisesta vaatimuksesta selvittää yksikäsitteisesti onko se saavutettu vai ei
 - ¥ Ovatko vaatimukset saavutettavissa käytössä olevilla resursseilla
 5. Ovatko kohtien keskinäiset suhteet kunnossa:
 - ¥ Ovatko tehokkuusvaatimukset saavutettavissa muut vaatimukset huomioonottaen
 - ¥ Ovatko käytettävyyksivaatimukset saavutettavissa yhteensopivuusvaatimusten täytyessä

Tarkastuslista projektisuunnitelmaa varten

1. Onko dokumentti ulkoasultaan muodollisesti kunnossa:
 - ¥ Otsikkotiedot
 - ¥ Versiohallinnan vaatimat tiedot
 - ¥ Toimittaja ja kirjoittajat osuukseineen
 - ¥ Onko dokumentin eri kohdat helposti viitattavissa muista dokumenteista
2. Ovatko kaikki tarvittavat asiat mukana:
 - ¥ Projektin yleiskatsaus ja viitteet muihin dokumentteihin
 - ¥ Työn jako tehtäviin
 - ¥ Arviot tehtävien tuotosten koosta; mahdollinen koodin uudelleenkäyttö on huomioitava
 - ¥ Arviot tehtäviin tarvittavista resursseista (henkilötyö, muu) ja resurssien organisointi
 - ¥ Käytettävä kehitystyön malli ja sen vaiheet, etapit, dokumentit, versiohallinnan piiriin kuuluminen
 - ¥ Vaiheiden suunnitellut alkamis- ja päättymisajat, etappienpäivämäärät
 - ¥ Käytettävät standardit, menetelmät ja apuvälineet
 - ¥ Versiohallinta (tarkoitus, eri henkilöiden oikeudet, ...)
 - ¥ Luettelo projektiin kuuluvista muista työskentelysuunnitelmista (testaussuunnitelma, ...)
 - ¥ Käytettävät työn ja tuloksen mittausmenetelmät
3. Vastaako dokumentti aiempia dokumentteja:
 - ¥ Dokumentin vertaaminen jo hyväksyttyihin dokumentteihin (vaatimusmäärittely “miksi mitään tehdään”, ...)
 - ¥ Ovatko viittaukset muihin dokumentteihin kunnossa
4. Ovatko kohdat sisäisesti kunnossa:
 - ¥ Onko työn jako tehtäviin onnistunut (vesiputousmallin jaottelu on liian karkea)
 - ¥ Ovatko arviot tehtävien tuotosten koosta sopivia
 - ¥ Onko koodin uudelleenkäytössä huomioitu kaikki saatavilla oleva valmis koodi
 - ¥ Vastaavatko tuotettavat dokumentit normaalia käytäntöä
 - ¥ Muodostavatko tuotettavat dokumentit sopivan kokonaisuuden

- ☒ Onko versiohallinnan piiriin sijoitettu kaikki tarpeellinen
- ☒ Ovatko etapit sopivasti valittu
- ☒ Vastaavatko valitut standardit, menetelmät ja apuvälineet normaalia käytäntöä
- ☒ Vastaavatko valitut mittausten menetelmät normaalia käytäntöä

5. Ovatko kohtien keskinäiset suhteet kunnossa:

- ☒ Ovatko resurssiarviot sopivia tehtävien tuotosten kokoihin nähden
- ☒ Onko eri tehtäviin sijoitetuilla henkilöillä tarvittavat tiedot ja taidot tai pitäisikö projektiin lisätä projektihenkilöstön koulutusta
- ☒ Ovatko etappien ajoitukset sopusoinnussa tehtävien ajoitusten kanssa
- ☒ Vastaavatko valitut mittausten menetelmät ja suunnitellut tarkastukset projektin laajuutta ja vaikeusastetta

Tarkastuslista suunnittelumallia varten

1. Onko dokumentti ulkoasultaan muodollisesti kunnossa:

- ☒ Otsikkotiedot
- ☒ Versiohallinnan vaatimat tiedot
- ☒ Toimittaja ja kirjoittajat osuiksineen
- ☒ Onko dokumentin eri kohdat helposti viitattavissa muista dokumenteista

2. Ovatko kaikki tarvittavat asiat mukana:

- ☒ Onko järjestelmän yleisrakenne ja olennaiset pääkomponentit selitetty ennen yksityiskohtiin menemistä
- ☒ Ovatko ohjelmointityyliin liittyvät ohjeet mukana

3. Vastaako dokumentti aiempia dokumentteja:

- ☒ Dokumentin vertaaminen jo hyväksyttyihin dokumentteihin
- ☒ Ovatko viittaukset muihin dokumentteihin kunnossa

4. Ovatko kohdat sisäisesti kunnossa:

- ☒ Riittävätkö esitetyt asiat jatkotyöskentelylle (jos tästä jatkaisivatkin aivan uudet henkilöt)
- ☒ Onko jokainen komponentti riittävän täydellinen

5. Ovatko kohtien keskinäiset suhteet kunnossa:

- ☒ Ovatko ohjelmiston eri osien väliset yhteydet kunnossa
- ☒ Ovatko ohjelmiston osat mahdollisimman riippumattomia toisistaan

Tarkastuslista käyttöohjetta varten

1. Onko dokumentti ulkoasultaan muodollisesti kunnossa:

- ☒ Otsikkotiedot
- ☒ Versiohallinnan vaatimat tiedot
- ☒ Toimittaja ja kirjoittajat osuiksineen
- ☒ Onko dokumentin eri kohdat helposti viitattavissa muista dokumenteista

2. Ovatko kaikki tarvittavat asiat mukana:

- ☒ Onko jokainen näyttö, raportti, massa-ajo ja muu toiminto kuvattu
- ☒ Onko käyttäjän mahdollisuudet ohjata järjestelmän toimintaa (asetusnäytöin, parametrein tms.) kuvattu
- ☒ Onko dokumentissa hakemista helpottavia apuneuvoja kuten sisällysluettelo ja hakusanaluettelo
- ☒ Onko virheilmoistusluettelo mukana

3. Vastaako dokumentti aiempia dokumentteja:

☒ Dokumentin vertaaminen jo hyväksyttyihin dokumentteihin

☒ Ovatko viittaukset muihin dokumentteihin kunnossa (jos niitä on ylipäättään pakko laittaa)

4. Ovatko kohdat sisäisesti kunnossa:

☒ Onko kieli käyttäjälle ymmärrettävää

☒ Onko passiivin käyttöä vältetty

☒ Onko hankalia asioita valaistu esimerkeillä

☒ Ovatko esimerkit luontevia ja auttavatko ne asian ymmärtämisessä

☒ Onko virhetilanteista toipuminen kuvattu käyttäjää helpottavasti

☒ Onko virheilmoitusten selostuksissa kuvattu kaikki työhöjeen mainitsemat seikat

5. Ovatko kohtien keskinäiset suhteet kunnossa:

☒ Onko sisällysluettelo kunnossa

☒ Onko hakusanaluettelo kunnossa

☒ Ovatko tekstissä olevat viittaukset dokumentin toisiin kohtiin kunnossa

Tarkastuslista testaus suunnitelmaa varten

1. Onko dokumentti ulkoasultaan muodollisesti kunnossa:

☒ Otsikkotiedot

☒ Versiohallinnan vaatimat tiedot

☒ Toimittaja ja kirjoittajat osuukseineen

☒ Onko dokumentin eri kohdat helposti viitattavissa muista dokumenteista

2. Ovatko kaikki tarvittavat asiat mukana:

☒ Onko testattavat asiat kuvattu

☒ Jos järjestelmässä on useita samantyyppisiä näyttöjä (raportteja tms.), niin onko niitä varten olemassa yleinen testauspohja, josta saadaan helposti kunkin yksittäisen näytön testauslomake

☒ Onko testauksen lopettamiskriteerit kuvattu

☒ Onko testauksen käytännön järjestelyt kuvattu

☒ Onko testauksen tulosten kirjaaminen, virheiden raportointi ja niiden korjaaminen kuvattu

☒ Onko mahdollisesti tarvittavat apuohjelmat kuvattu

☒ Onko työnjako eri henkilöiden kesken kuvattu

☒ Onko moduulitestaus ja integrointitestaus kuvattu erikseen

☒ Onko käytettävät testausperiaatteet ("jokainen ohjelmariivi suoritetaan vähintään kerran" tms.) kuvattu

3. Vastaako dokumentti aiempia dokumentteja:

☒ Dokumentin vertaaminen jo hyväksyttyihin dokumentteihin

☒ Ovatko viittaukset muihin dokumentteihin kunnossa

4. Ovatko kohdat sisäisesti kunnossa:

☒ Onko lukijalle selvää, milloin käsitellään ohjelmoinnin yhteydessä tehtävää testausta ja milloin erillisenä toimintana suoritettavaa testausta

☒ Onko jokaiselle testitapaukselle kuvattu odotettu tulos

☒ Onko testaaminen käytännössä mahdollista toteuttaa suunnitelman mukaisesti

5. Ovatko kohtien keskinäiset suhteet kunnossa:

☒ Onko lukijalle selvää se, mitä asioita pitää tehdä yhdessä tai peräkkäin tietyssä järjestyksessä testaamisen onnistumiseksi

☒ Muodostavatko moduulitestaus ja integrointitestaus järkevän kokonaisuuden

Tarkastuslista testausraporttia varten

1. Onko dokumentti ulkoasultaan muodollisesti kunnossa:
 - ☒ Otsikkotiedot
 - ☒ Versiohallinnan vaatimat tiedot
 - ☒ Toimittaja ja kirjoittajat osuuksineen
 - ☒ Onko dokumentin eri kohdat helposti viitattavissa muista dokumenteista
2. Ovatko kaikki tarvittavat asiat mukana:
 - ☒ Onko tehty yhteenveto testauksen etenemisestä
 - ☒ Ovatko yksittäiset testausraportit mukana (liitteenä)
3. Vastaako dokumentti aiempia dokumentteja:
 - ☒ Dokumentin vertaaminen jo hyväksyttyihin dokumentteihin
 - ☒ Ovatko viittaukset muihin dokumentteihin kunnossa
4. Ovatko kohdat sisäisesti kunnossa:
 - ☒ Käykö dokumentista selvästi ilmi se, millä perusteilla testaus on lopetettu
5. Ovatko kohtien keskinäiset suhteet kunnossa:
 - ☒ Vastaako yhteenveto yksittäisiä testausraportteja

Tarkastuslista projektin loppuraporttia varten

1. Onko dokumentti ulkoasultaan muodollisesti kunnossa:
 - ☒ Otsikkotiedot
 - ☒ Versiohallinnan vaatimat tiedot
 - ☒ Toimittaja ja kirjoittajat osuuksineen
 - ☒ Onko dokumentin eri kohdat helposti viitattavissa muista dokumenteista
2. Ovatko kaikki tarvittavat asiat mukana:
 - ☒ Ovatko kaikki työohjeessa mainitut kohdat mukana
3. Vastaako dokumentti aiempia dokumentteja:
 - ☒ Dokumentin vertaaminen jo hyväksyttyihin dokumentteihin
 - ☒ Ovatko viittaukset muihin dokumentteihin kunnossa
4. Ovatko kohdat sisäisesti kunnossa:
 - ☒ Onko asiakkaan kanssa keskusteltu lopputuotteen hyödyllisyydestä asiakkaalle
 - ☒ Onko ongelmat tuotu esille rehellisesti
 - ☒ Onko suunnitelmaan nähden syntyneet poikkeamat selitetty
5. Ovatko kohtien keskinäiset suhteet kunnossa:
 - ☒ Selviävätkö mahdolliset henkilökohtaiset työpanospoikkeamat muista kohdista tai onko ne selitetty

Tarkastuslista ohjelmamoduulia varten

1. Onko ohjelmamoduuli ulkoasultaan muodollisesti kunnossa:
 - ☒ Otsikkotiedot
 - ☒ Versiohallinnan vaatimat tiedot
 - ☒ Kirjoittajat osuuksineen
2. Ovatko kaikki tarvittavat asiat mukana:
 - ☒ Tekeekö moduuli kaiken sen, mitä määrittäksessä kuvataan

3. Vastaako dokumentti aiempia dokumentteja:

☒ Moduulin vertaaminen jo hyväksyttyihin dokumentteihin

☒ Ovatko viittaukset muihin dokumentteihin kunnossa

4. Ovatko kohdat sisäisesti kunnossa:

☒ Onko funktioiden yleistiedot kommentoitu asianmukaisesti

☒ Onko tietorakenteet ja muuttujat kommentoitu

☒ Auttavatko ohjelman toimintaa kuvaavat kommentit lukijaa ymmärtämään ohjelmaa vai toistavatko ne vain itse ohjelmakoodia

☒ Onko muuttujat nimetty mahdollisten ohjeiden mukaisesti

5. Ovatko kohtien keskinäiset suhteet kunnossa:

☒ Onko globaaleja muuttujia käytetty vain perustelluista syistä

☒ Muodostavatko funktiot siistin rakennelman

Liite 2: Muutosten hallinta

Muutosten hallinnan tavoitteena on pitää järjestelmän tilanne tiukasti hallinnassa, estää harkitsemattomien muutosten tekeminen ja varmistua siitä, että yhtään muutosta ei tehdä ilman kaikkien osapuolien yhteistä sopimusta.

Muutosten hallintaa varten ryhmä valitsee keskuudestaan henkilön, joka toimii muutosten hallinnan vastuuhenkilönä.

Muutostarvetta syntyy tarkastusmenettelyn tuloksena (dokumentit, ohjelmakoodi), asiakaspalautteena sekä palautteena ohjaajilta. Muutosehdotukset kirjataan muutoslomakkeelle, joka toimitetaan (sähköpostina tai muulla tavoin) muutosten hallinnan vastuuhenkilölle.

Vastuuhenkilö valmistelee päätösehdotuksen ja tuo sen ryhmän päätettäväksi.

Mikäli kyseessä on merkittävä muutos, vastuuhenkilön on oltava yhteydessä ohjaajaan ennen muutoksesta päättämistä.

Huom: Jos muutospyynnön sisältämä puute päätetään korjata, vastuuhenkilö kirjaa puutteen mantisilla ja siitä määritellään erillinen tehtävä projektiin.

Vastuuhenkilö huolehtii siitä, että muutospyynnön tila vastaa muutospyynnön todellista käsittelyvaihetta.

Muutospyyntö

*****Ilmoittaja täyttää

Ryhmä: _____

Päiväys: _____.____._____

Ilmoituksen tekijä: _____

Ohjelmiston versio: _____

Haluttu muutos:

Kyseessä on ☐ määrittelyvirhe
☐ toteutusvirhe
☐ ehdottomasti tarvittava määritysmuutos
☐ muu määritysmuutos

*****Vastuuhenkilö täyttää

Tunniste: _____

Saapunut: _____.____._____

Lisäselitys: _____

Ehdotus ryhmälle:

Toimenpide ☐ käyttäjän ospastus
☐ yhdistäminen toiseen muutospyyntöön
☐ tehdään muutos
☐ ei aiheuta toimenpiteitä

Puutteen aiheutti toiminto

☐ Vaatimusten määrittely
☐ Projektin suunnittelu
☐ Suunnittelu
☐ Toteutus
☐ Testaus ja kelpuutus
☐ Käyttöönotto
☐ Ylläpito
☐ Kokoonpanon hallinta
☐ Dokumenttien hallinta
☐ Projektin ohjaus
☐ Koulutus

Puute havaittiin toiminnossa

☐ Vaatimusten määrittely
☐ Projektin suunnittelu
☐ Suunnittelu
☐ Toteutus
☐ Testaus ja kelpuutus

- ☐ Käyttöönotto
- ☐ Ylläpito
- ☐ Kokoonpanon hallinta
- ☐ Dokumenttien hallinta
- ☐ Projektin ohjaus
- ☐ Koulutus

Komponentti, jossa puute oli _____

Työmääräarvio: _____ henkilötyötuntia

Dokumentit ja moduulit, joita muutos koskee:

Ryhmän päätös:

- Toimenpide
- ☐ käyttäjän opastus
 - ☐ yhdistäminen toiseen muutospyyntöön
 - ☐ tehdään muutos
 - ☐ ei aiheuta toimenpiteitä