

Projektisuunnitelma

DHT – Distributed Hash Table

Helsinki 26.5.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Marko Rähkä
Risto Saarelma
Antti Salonen
Tuomas Toivonen
Tomi Tukiainen
Simo Viitanen

Asiakas

Jussi Lindgren

Johtoryhmä

Juha Taina
Turjo Tuohiniemi

Kotisivu

<http://www.cs.helsinki.fi/group/dht/>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.9	5.2.2004	Työstöversio
1.0	9.2.2004	Tarkastettava versio
1.1	12.2.2004	Tarkastuksen perusteella korjattu versio
1.2	26.5.2004	Poistettu yhteystiedot arkistokappaleesta

Sisältö

1 Johdanto	1
1.1 Terminologiaa	1
1.2 Projektin tavoite	1
1.3 Dokumentin rakenne	1
2 Organisaatio	2
2.1 Henkilöstö	2
2.2 Vastualueet	2
3 Työn yleiskuvaus	4
4 Kokoarvio	5
4.1 Funktiopistelaskelma DHT-ohjelmistolle	5
5 Työvaiheet	7
5.1 Vaatimusten määrittely	7
5.2 Tekninen suunnittelu	7
5.3 Tuotteen toteuttaminen	7
5.4 Testaus	8
5.5 Viimeistely	8
5.6 Tuotteen toimittaminen asiakkaalle	8
6 Tuotokset	9
6.1 Dokumentit	9
6.1.1 Projektisuunnitelma	9
6.1.2 Määrittelydokumentti	9
6.1.3 Suunnitteludokumentti	9
6.1.4 Testaussuunnitelma	9
6.1.5 Tekninen dokumentti	10
6.1.6 Testausraportti	10
6.1.7 Loppuraportti	10
6.2 Ohjelmisto	10
6.2.1 DHT-kirjasto	10

	ii
6.2.2 Testiohjelmat	10
7 Aikataulu	11
8 Työskentelytavat	13
8.1 Ohjelmointi	13
8.2 Dokumentointi	13
8.2.1 Työtuntilistat ja edistymisraportit	13
8.2.2 Päiväkirjat	14
8.3 Versionhallinta	14
8.4 Kokoukset	15
8.4.1 Suunnittelu	15
8.4.2 Seuranta	15
8.4.3 Tarkastus	15
8.4.4 Ohjelmakoodin tarkastus	15
8.5 Kommunikointi	16
9 Riskianalyysi	17
9.1 Kademia-algoritmin toteuttamisessa on ongelmia	17
9.2 GNUnet-alustassa on puutteita	17
9.3 Työntekijöitä poistuu kesken projektin	17
9.4 Asiakas muuttaa vaatimuksia	18
9.5 Aikataulusta myöhästytään	18
9.6 Suunnitelman toteuttamisessa ilmenee ongelmia	18
10 Laadunvalvonta	19
10.1 Dokumentaatio	19
10.2 Ohjelmakoodi	19
10.2.1 Ohjelmakoodin testaus	20

1 Johdanto

Projekti toteutetaan Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojekti-kurssin harjoitustyönä keväällä 2004. Tässä dokumentissa kuvataan, miten projektiryhmä (kts. organisaatio) suunnittelee toteuttavansa tämän projektin. Dokumentti on tarkoitettu pääasiassa ryhmän itsensä, asiakkaan, sekä ohjelmistotuotantoprojekti-kurssin henkilöstön luettavaksi.

1.1 Terminologiaa

DHT	Distributed Hash Table, hajautettu hajautustaulu
GNUnet	avoin sovelluskehys turvallisille vertaisverkkoratkaisuille
Kademlia	eräs kirjallisuudessa kuvattu DHT-toteutus
Wiki	kollektiivisesti ylläpidettävä WWW-sivusto, joka sisältää työkalut itsensä muokkaamiseen

1.2 Projektin tavoite

Vertaisverkkoihin (Peer-to-Peer, P2P) perustuvat tiedonhallintamenetelmät ovat viime aikoina saaneet osakseen paljon huomiota sekä tutkijapiireissä että tavallisten käyttäjien keskuudessa. Tämän projektin tavoitteena on toteuttaa vertaisverkkoon perustuva hajautettu hajautustaulu ja sen testaamiseen tarvittava interaktiivinen komponentti, jonka toiminnallisuus määritellään määrittelyvaiheessa.

DHT-toteutuksen algoritmien osuus tehdään Kademlia-järjestelmän kuvauksen¹ mukaisesti ja vertaisverkon perustana käytetään GNUnetiä².

1.3 Dokumentin rakenne

Dokumentin loppuosa on rakenteeltaan seuraavanlainen. Luvussa 2 esitellään projektin osapuolet ja heidän roolinsa. Luvuissa 3 ja 4 kuvataan tarkemmin toteutettavaa järjestelmää. Luvuissa 5 ja 6 esitellään projektin työvaiheet ja niihin liittyvät tuotokset. Luvussa 7 on projektin alustava aikataulu. Luvussa 8 käsitellään projektin toteuttamisessa käytettäviä työskentelytapoja. Luvussa 9 käydään läpi projektin toteuttamiseen liittyviä riskejä. Luvussa 10 käsitellään lopuksi laadunvalvontaan liittyviä asioita.

¹Petar Maymounkov, David Mazières: Kademlia: A Peer-to-peer Information System Based on the XOR Metric <http://kademlia.scs.cs.nyu.edu>

²Free Software Foundation: GNUnet <http://www.ovmj.org/GNUnet/>

2 Organisaatio

Tämä luvussa käydään läpi projektin organisaatio. Kerrotaan sekä projektiin osallistuvat henkilöt, että heidän vastuualueensa projektissa.

2.1 Henkilöstö

Projektiin osallistuvat seuraavat henkilöt³:

Nimi	Rooli	Sähköpostiosoite	Puhelinnumero
Hallintoryhmä			
Jussi Lindgren	asiakas	–	
Marianne Korpela	ohjaaja	–	
Juha Taina	vastuuhenkilö	–	
Projektiryhmä			
Tomi Tukiainen	projektipäällikkö	–	–
Antti Salonen	asiakasyhteyshenkilö	–	–
Tuomas Toivonen	testausvastaava	–	–
Marko Räihä	WWW- ja ympäristövastaava	–	–
Risto Saarelma	dokumenttivastaava	–	–
Simo Viitanen	yleismies	–	–

Tärkeitä osoitteita:

Projektiryhmän sähköpostilista:	ohtuk04-dht-list@cs.helsinki.fi
Ryhmän WWW-sivut:	http://www.cs.helsinki.fi/group/dht
Ryhmän jäsenten omat sivut:	http://db.cs.helsinki.fi/tkt_dht/usermod/wiki.pl?Projektiryhmä

2.2 Vastuualueet

Vastuualueet määritellään seuraavasti:

Asiakas vastaa pääasiassa tuotteen määrittämisestä. Ryhmä voi kääntyä asiakkaan puoleen joissain GNUnet-alustaan liittyvissä teknisissä kysymyksissä. Tämän lisäksi asiakas osallistuu joidenkin työvaiheiden hyväksymiseen liittyviin tarkastuksiin.

Ohjaajan tehtävänä on ohjata ryhmää oikeaan suuntaan, esimerkiksi neuvomalla ongelmatilanteissa ja auttamalla ryhmää pääsemään alkuun. Ohjaaja ei varsinaisesti osallistu ryhmän työskentelyyn.

Vastuuhenkilö voi auttaa projektin hallinnon ja etenemisen kannalta epäselvissä tapauksissa.

Projektipäällikön vastuulla on projektin yleinen valvonta ja aikataulussa pysymisestä huolehtiminen.

³Projektiryhmän jäsenten yhteystiedot poistettu dokumentin arkistokappaleesta.

Asiakasyhteyshenkilö hoitaa kommunikoinnin projektiryhmän ja asiakkaan välillä. Hän hoitaa mahdolliset asiakastapaamisten ulkopuolella tapahtuvat tarkennukset ja vastaavat kysymykset molempiin suuntiin.

Testausvastaavan vastuulla on testausmenetelmien ja testitapausten määrittäminen. Hänellä on vastuu myös testauksen suorittamisesta.

WWW-vastaavan vastuulla on WWW:ssä julkaistava materiaali. Ryhmä käyttää WWW-tekniikkana Wiki-sivustoa, joten WWW-vastaavan työksi jää pitää huoli Wikin käytöstä.

Ympäristövastaavan vastuulla on ryhmän käyttämän ympäristön rakentaminen ja ylläpitäminen. Ympäristöön sisältyy CVS-palvelin ja Wiki-sivustojärjestelmä.

Dokumenttivastaavan vastuulla on huolehtia, että oikeat dokumentit ovat valmiita oikeaan aikaan. Dokumenttivastaava jakaa dokumentointitehtävät ryhmän kesken ja huolehtii, että dokumentti tulee tehtyä.

Yleismies huolehtii muista projektiin liittyvistä pikkuasioista, kuten ryhmän sisäisestä kommunikoinnista ja tuntikirjanpidon raportoinnista vastuuhenkilölle.

Projektipäällikön ja yleismiehen tehtävät vaihtuvat kun suunnitteluvaihe on päättynyt.

3 Työn yleiskuvaus

Tuotettava DHT-ohjelmisto toteutetaan GNUnet-järjestelmän osaksi siten, että DHT-ohjelmistosta tulee yksi GNUnetin moduuli. GNUnet on yleinen vertaisverkkojen toiminnallisuuden tarjoava sovelluskehys, jonka toiminnallisuutta DHT-moduuli laajentaa. DHT-moduuli toteutetaan noudattaen Kademia-järjestelmän teknistä kuvausta (kts. johdanto).

DHT-verkossa voi olla hajautettuna useita hajautustauluja, siten että kukin solmu voi sisältää useaan eri hajautustauluun talletettuja alkioita. DHT-moduulin tulee tarjota vähintään seuraava API, siten että jokainen operaatio ottaa parametrina kohteena olevan hajautustaulun tunnistein.

- Join (liittää verkon solmun johonkin tauluun)
- Leave (erottaa verkon solmun taulusta)
- Insert (lisää tauluun avain-arvo -parin)
- Fetch (hakee taulusta arvon)

GNUnet asettaa vaatimukset laitteistolle ja käyttöjärjestelmälle. Tavoitteena on, ettei DHT-moduuli tiukenna näitä vaatimuksia.

Lisäksi toteutetaan yksinkertainen testisovellus, joka toimii samalla esimerkksiovelluksena DHT-moduulin mahdollisia tulevia käyttäjiä varten. Testisovellus on laajuudeltaan sellainen, että sen avulla voidaan kattavasti suorittaa DHT-moduulin testaus testausvaiheessa. Testisovellukseksi on alustavasti ehdotettu jonkinlaista keskusteluohjelmaa.

4 Kokoarvio

Arvioitaessa ohjelmointityön suuruutta koodiriveinä on huomioitu seuraavat osa-alueet:

- Eri kutsutyypin käsittely. Tämä osa-alue sisältää valtaosan logiikasta, eikä sen tarkka jäsentely ole ennen määrittely- ja suunniteluvaiheita selvä. Keskeisiä asioita ovat muun muassa:
 - kytkeytyminen GNUnettiin
 - FIND_NODE- ja FIND_VALUE-algoritmien toteutus (Kademlian ehkä keskeisimmät algoritmit)
 - kokonaislaajuus erittäin karkeasti arvioituna luokkaa 3000 riviä koodia
- tietorakenne k-jonoille, todennäköisesti yksinkertainen taulukkopohjainen toteutus (id:n lisäys, haku, poisto) - n. 200 riviä koodia
- solmuun sijoitetun datan talletus, vaatimuksena todennäköisesti vain muistiin tallettaminen, eli esim. yksinkertainen hajautustaulu tai hakupuu - n. 200 riviä koodia
- testisovellus, todennäköisesti yksinkertainen keskusteluohjelmisto - n. 500 riviä koodia

Koodin arvioitu kokonaismäärä on siis 4000 koodirivin kokoluokkaa. Koodirivillä tarkoitetaan tässä yhtä C-kielen lausetta.

4.1 Funktiopistelaskelma DHT-ohjelmistolle

Tässä esitetty funktiopistelaskelma on tehty noudattaen "Software Engineering - A Practitioners Approach-kirjan sivujen 87-91 ohjetta. Koska monimutkaisuuden säätökertoimien valinta on subjektiivinen toimenpide, kannattaa tätä laskelmaa pitää lähinnä suuntaa antavana. Lisäksi tämäntyyppinen funktiopistelaskelma soveltuu paremmin käyttöliittymä-painotteisiin ohjelmistoihin, jollainen DHT ei ole.

		Paino	Selitys painolle	Pisteet
Käyttäjän syötteitä (kpl)	4	3	Yksinkertainen	12
Tulosteita käyttäjälle	2	5	Keskinkertainen	10
Käyttäjän kyselyitä	3	3	Yksinkertainen	9
Tiedostoja	2	7	Yksinkertainen	14
Ulkoisia liittymiä	1	10	Monimutkainen	10

Yhteensä 55 pistettä

Monimutkaisuuden säätökertoimet, asteikolla 1-5:

- Require reliable backup and recovery? 1
- Are data communications required? 5

- Are there distributed processing functions? 5
- Is performance critical? 3
- Will the system run in existing, heavily utilized environment? 2
- Does the system require on-line data entry? 5
- Does the on-line data entry require input transaction to be built over multiple screens or operations? 5
- Are the master files updated on-line? 5
- Are inputs, outputs, files or inquiries complex? 2
- Is the internal processing complex? 5
- Is the code designed to be reusable? 3
- Are conversion and installation included in the design? 0
- Is the system designed for multiple installations in different organizations? 0
- Is the application designed to facilitate change and ease of use by the user? 5

Yhteensä: 46

Funktiopisteet: 61

5 Työvaiheet

Projekti toteutetaan useassa työvaiheessa vesiputousmallin mukaan. Periaatteessa seuraavaan työvaiheeseen siirryttäessä edellisten vaiheiden olisi oltava valmiina, käytännössä seuraava vaihe voidaan tarvittaessa aloittaa hieman etuajassa edellisen vaiheen ollessa vielä käynnissä.

Projektin työvaiheet ovat seuraavat:

5.1 Vaatimusten määrittely

Määrittelyvaiheessa laaditaan yksityiskohtainen kuvaus toteutettavan järjestelmän ominaisuuksista ja tarjoamista palveluista. Määrittelydokumentti näytetään asiakkaalle, joka voi vaatia määritelmään muutoksia.

Määrittelyvaihe päättyy kun projektiryhmä ja asiakas ovat tyytyväisiä määritelmään ja määritelmän kuvaava määrittelydokumentti läpäisee tarkastuksen.

5.2 Tekninen suunnittelu

Tässä vaiheessa laaditaan yksityiskohtainen suunnitelma määritelmädokumentissa kuvailun järjestelmän toteuttamiseksi. Suunnitteluvaiheessa määritellään myös menettelytavat järjestelmän testaamiseksi. Vaiheen aikana voidaan kirjoittaa prototyypiohjelmaa joilla kokeillaan suunniteltujen ratkaisujen toimivuutta. Prototyypiohjelmat on tarkoitettu kertakäyttöisiksi eikä niitä liitetä varsinaiseen tuotekoodiin. Englanninkielisen teknisen dokumentin kirjoittaminen aloitetaan tämän vaiheen aikana.

Vaiheen tuloksena syntyvät suunnitteludokumentti ja testaussuunnitelma. Koska tuote on kirjasto eikä sovellusohjelma, myös suunnitteluvaiheen dokumentit näytetään asiakkaalle. Asiakas voi halutessaan ehdottaa muutoksia toteutussuunnitelmaan.

Suunnitteluvaihe päättyy, kun suunnitteludokumentti ja testaussuunnitelma ovat valmiit ja läpäisevät tarkastuksen. Teknisen dokumentin ei ole pakko olla valmis suunnitteluvaiheen loppuun mennessä.

5.3 Tuotteen toteuttaminen

Toteutetaan järjestelmä suunnitteludokumentin mukaan. Yksikkötestaus on kiinteä osa toteutusprosessia. Toteutettuihin moduuleihin kirjoitetaan heti alussa yksikkötestit, joiden avulla moduulin toimivuus voidaan määrittää kaikissa toteutusvaiheissa. Järjestelmään toteutetaan regressiotestauskehys, jonka avulla voidaan testata aiheuttaako jokin tehty muutos virheitä muutetun moduulin sijasta jossakin toisessa moduulissa.

Toteutusvaihe päättyy kun suunnitelmadokumentin kuvailema järjestelmä on toteutettu kokonaisuudessaan ja järjestelmän kaikki tarkastettava ohjelmakoodi läpäisee tarkastuksen.

5.4 Testaus

Testausvaiheessa tutkitaan järjestelmän käytännön toimintaa ja suorituskykyä testausdokumentissa kuvailuilla järjestelmätesteillä. DHT-algoritmin tapauksessa tähän liittyy jonkinlaisen simuloitun verkkoympäristön rakentaminen. Testausvaiheen tuotoksena syntyy testien tulokset esittävä testausraportti.

Testausvaihe päättyy kun koko testaussuunnitelma on käyty läpi ja kaikki tuotteesta testauksessa löytyneet virheet on korjattu.

5.5 Viimeistely

Viimeistelyvaiheessa tuotteeseen tehdään GNUnet-projektiin liittämisen mahdollisesti vaatimia kosmeettisia muutoksia. Tässä vaiheessa laaditaan myös projektin kulkua kuvaava loppuraportti.

Viimeistelyvaihe päättyy kun projekti on valmis liitettäväksi GNUnet-projektiin ja loppuraportti sekä suunnitteluvaiheessa aloitettu tekninen dokumentti ovat valmiita. Teknisen dokumentin on oltava valmis viimeistelyvaiheen loppuun mennessä.

5.6 Tuotteen toimittaminen asiakkaalle

Projektissa tuotetut lähdekoodit ja dokumentit lähetetään pakattuna sähköpostin liitetiedostona asiakkaalle. Lähdekoodit ovat hakemistopuussa jonka asiakas liittää GNUnetin CVS:ään.

Tämä on projektin viimeinen työvaihe ja päättyy, kun asiakas hyväksyy toimitetun tuotteen.

6 Tuotokset

Tässä luvussa kuvataan kaikki projektin tuotokset.

6.1 Dokumentit

Seuraavassa kuvataan projektin tuottamat dokumentit. Dokumentit kirjoitetaan suomeksi ellei toisin määritetä.

6.1.1 Projektisuunnitelma

Projektisuunnitelma sisältää kuvauksen projektin yleisistä toimintatavoista ja alustavan aikataulun. Projektisuunnitelma tarkastetaan asiakkaan ollessa läsnä. Projektisuunnitelmaa käytetään ohjaamaan ryhmän työskentelyä koko projektin ajan.

6.1.2 Määrittelydokumentti

Määrittelydokumentti sisältää yksityiskohtaisen kuvauksen siitä, mitä palveluja toteutettavan järjestelmän täytyy tarjota (toiminnalliset vaatimukset). Määrittelydokumentissa pyritään myös dokumentoimaan sellaisia vaatimuksia, mm. suorituskyvyn suhteen, jotka eivät ole alunperin asiakkaan erikseen ilmoittamia (laadulliset vaatimukset). Määrittelydokumentti tarkastetaan asiakkaan ollessa läsnä. Määrittelydokumenttia käytetään suunnitteluvaiheen työn pohjana.

6.1.3 Suunnitteludokumentti

Suunnitteludokumentti sisältää yksityiskohtaisen suunnitelman määrittelydokumentissa kuvaillun toiminnallisuuden toteuttamiseksi. Suunnitteludokumentti kuvaa toteutettavan järjestelmän niin yksityiskohtaisesti, että järjestelmän ohjelmoiminen on sen perusteella suoraviivaista. Suunnitteludokumentti luettelee toteutettavat ohjelmamoduulit, moduulien väliset yhteydet, moduulien sisältämät algoritmit ja moduulien toteuttamat rajapinnat, käytetyt tietorakenteet, käytetyt kirjastot ja rajapinnat sekä käytännöt symbolien nimeämiseen ja ohjelmakoodin sientämiseen. Koska toteutettava järjestelmä sisältää pääasiassa algoritmia eikä sovellusohjelmaa, myös suunnitteludokumentti esitetään asiakkaalle, jolloin asiakkaan on mahdollista esittää siihen muutoksia. Suunnitteludokumenttia käytetään toteutusvaiheessa.

6.1.4 Testaussuunnitelma

Testaussuunnitelma määrittelee menettelytavat järjestelmän testaamiseksi. Testaussuunnitelmaa käytetään toteutus- ja testausvaiheessa.

6.1.5 Tekninen dokumentti

Tekninen dokumentti sisältää projektin ulkopuoliselle ohjelmoijalle tarkoitetun kuvauksen järjestelmästä. Koska järjestelmä aiotaan integroida kansainväliseen GNUnet-projektiin, kirjoitetaan dokumentti englanniksi. Tekninen dokumentti on tarkoitettu tuotettua ohjelmakoodia hyödyntävien osapuolien apuvälineeksi.

6.1.6 Testausraportti

Testausraportti sisältää tiedot suoritetusta testauksesta. Testausraporttia voidaan käyttää mm. tuotteen laadun arviointiin. Testausraportti on tarkoitettu projektiryhmän, asiakkaan ja kurssin henkilöstön käytettäväksi.

6.1.7 Loppuraportti

Loppuraportti kuvaa projektin kulkua ja kertoo mahdollisesti ilmenneistä ongelmista. Raportti kertoo lisäksi suunnitelmiin ja aikatauluun tehdyistä muutoksista sekä näiden syistä. Loppuraportti on tarkoitettu ryhmän jäsenten ja ohjelmistotuotantoprojekti-kurssin henkilöstön käyttöön.

6.2 Ohjelmisto

Ohjelmistoon sisältyvät ohjelmakoodit varsinaisesta DHT-kirjastosta sekä joistakin kirjastoa testaavista ja kirjaston ominaisuuksia esittelevistä ylimääräisistä ohjelmista.

6.2.1 DHT-kirjasto

DHT-kirjaston ohjelmakoodi on C-kielinen, suunnitteludokumentin mukainen toteutus määrittelydokumentissa kuvatusta järjestelmästä. Toteutettuihin moduuleihin sisältyy yksikkötestit, joiden avulla moduulien toimivuus voidaan määrittää.

6.2.2 Testiohjelmat

Keskeisin testiohjelma on esimerkkisovellus, joka hyödyntää DHT-kirjaston ominaisuuksia ja toimii esimerkkinä DHT-kirjaston käyttämisestä. Lisäksi testiohjelmiin kuuluu jonkinlainen regressiotestauskehys, joka ajaa automaattisesti DHT-moduulien yksikkötestit. Regressiotestejä käytetään paljastamaan yhden moduulin muuttamisen aiheuttamat mahdolliset toimintavirheet toisissa moduuleissa.

7 Aikataulu

Tässä luvussa kuvataan projektin aikataulu ja se, miten projekti jakaantuu pienempiin osiin. Osien välisiä suhteita on selvennetty GANTT-kaavion avulla. Aikataulua laadittaessa on otettu huomioon tuotosten eräpäivät, demotilaisuus ja pääsiäisloma.

Aikataulu on laadittu siten, että kunkin työvaiheen päättymispäiväksi on merkitty se päivä, jolloin työvaiheen tuotos on valmiina tarkastettavaksi.

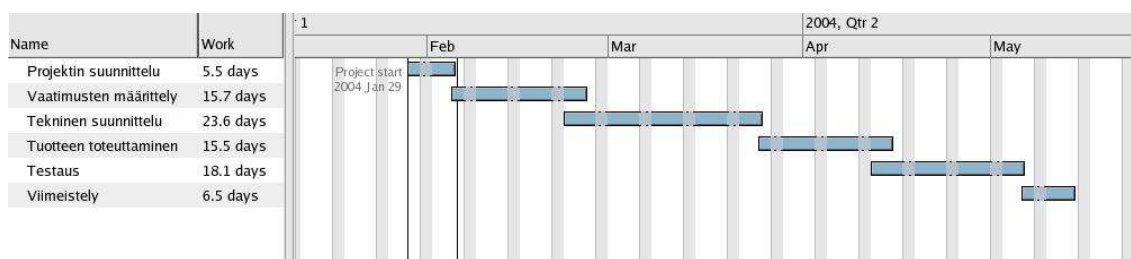
Projekti jakautuu työvaiheisiin seuraavan taulukon mukaisesti:

Työvaihe	Kausi	Tuotoksen tarkastuspäivä
Projektin suunnittelu	22.1. – 5.2.2004	9.2.2004
Tuotteen määrittely	5.2. – 26.2.2004	1.3.2004 (Tarkastetaan määrittelydokumentti)
Tuotteen suunnittelu	26.2. – 25.3.2004	29.3.2004 (Tarkastetaan suunnitteludokumentti)
Tuotteen toteuttaminen	25.3. – 15.4.2004	19.4.2004 (Tarkastetaan vaikeat kohdat ohjelmakoodista)
Tuotteen testaaminen	15.4. – 6.5.2004	
Viimeistely	6.5. – 14.5.2004	

Työvaiheiden tuotokset ovat seuraavat:

Työvaihe	Työvaiheen tuotos
Projektin suunnittelu	projektisuunnitelma
Tuotteen määrittely	määrittely- ja osa testausdokumenttia
Tuotteen suunnittelu	suunnittelu-, testausdokumentti ja käyttöohje
Tuotteen toteuttaminen	ohjelmakoodi
Tuotteen testaaminen	testausraportti
Viimeistely	viimeistellyt dokumentit ja loppuraportti

Seuraavassa kuvassa on projektin työvaiheiden aikataulua kuvaava GANTT-kaavio. Käytännössä työvaiheiden päällekkäisyys saattaa olla suurempi kuin kuvassa esitetty, esimerkiksi testauksen ja toteutuksen osalta.



Seuraavassa projektin suunniteltuun aikatauluun merkittävästi vaikuttavat asiat:

- pääsiäisloma
- projektipäällikön ja yleismiehen roolit vaihtuvat suunnitteludokumentin tarkastamisen jälkeen 29.3.2004
- demotilaisuus 7.5.2004, viikko 19
- projektin suunniteltu päättymispäivä 14.5.2004

- projektin viimeinen mahdollinen päättymisajankohta 30.5.2004, viikko 22

Projektin katsotaan päättyneen onnistuneesti, kun kaikki projektiin liittyvät tuotokset on ryhmän enemmistön päätöksellä ja kurssin henkilöiden puolesta todettu valmiiksi ennen projektin viimeistä mahdollista päättymisajankohtaa. Mikäli tuotokset eivät ole valmiita viimeiseen mahdolliseen päättymisajankohtaan mennessä, päättyy projekti kokonaan tai osittain epäonnistuneesti.

8 Työskentelytavat

8.1 Ohjelmointi

Ohjelmointikielenä käytetään C-kieltä. Ohjelmakoodi ja sen kommentointi tehdään englanniksi kohderyhmän kansainvälisyyden vuoksi. Projektin päätyttyä tämän materiaalin sekä englanninkielisen teknisen dokumentaation julkaisu GNUnetin CVS-repositorion välityksellä on asiakkaan vastuulla. Ohjelmoinnissa noudatetaan GNUnetin käyttämää tyyliä. GNUnet käyttää ohjelmakoodin kommentoinnin selaustyökaluna Doxygeniä⁴, joten kommentointi pitää tehdä sen mukaisesti⁵.

Ohjelmakooditiedostot ovat Unix-tiedostoja, joten jos joku ohjelmoija muokkaa koodia jossain muussa ympäristössä (esim. OS X tai Windows), niin pitää huomioida että rivien vaihtomerkit ja tiedostojen muoto ovat Unix-konvention mukaiset ennen kuin siirtää tiedostoja CVS-repositorioon.

8.2 Dokumentointi

Projektiin liittyvät dokumentaatiot kirjoitetaan suomeksi Wiki-järjestelmään ja konvertoidaan automaattisesti PDF- ja PS-muotoihin. Materiaali julkaistaan Wikissä⁶ Dokumenttilinkin alla. Julkaisumuodot ovat Wikin kautta muokattava HTML ja tästä materiaalista automaattisesti konvertointi LaTeX:n kautta PDF- sekä PS-tiedostomuotoihin. Sihteerivuoro kiertää niiden kesken, joilla on käytössä kannettavat tietokoneet.

8.2.1 Työtuntilistat ja edistymisraportit

Ryhmä raportoi projektin edistymisestä kurssin vastuuhenkilöille. Tämän vuoksi ryhmän jäsenet pitävät työtuntikirjanpitoa, joista koostetaan edistymisraportit. Työtuntilistat tehdään määrämuotoiseen tekstitiedostoon, jonka jokainen sijoittaa CVS-repositoriossa projekti/työtunnit/etunimisukunimi.txt -tiedostoon. Tiedosto alkaa opiskelijan nimellä (ensimmäinen rivi). Tämän jälkeen tiedostossa on tietueita, joissa on neljä kenttää erotettuna tabilla tai spacella. Yksi tietue aina omalle riville.

Tietueen kentät ovat:

- päivämäärä: (muoto pp.kk.vvvv),
- projektin vaihe:
 - AT - aineistoon tutustuminen,
 - YP - ylläpito,
 - PS - projektisuunnitelma,

⁴Doxygen: <http://www.doxygen.org/>

⁵<http://www.stack.nl/~dimitri/doxygen/docblocks.html>

⁶http://db.cs.helsinki.fi/~tkt_dht/

- VA - vaatimusanalyysi,
 - SU - suunnittelu,
 - KO - koodaus,
 - LR - loppuraportti,
- työtunnit: (nn tunnin tarkkuudella tai nn.nn puolen tunnin tarkkuudella) ja
 - työkuvaus: muutaman sanan kuvaus työstä.

Tiedostossa olevat #-merkillä alkavat rivit tulkitaan kommentteiksi, eikä niitä sisällytetä edistymisraporttiin. Yleismies tekee työtuntikirjanpidosta yhteenvetoraporttin, johon hän lisää tiedot raporttiin mennessä alkaneista työvaiheista, valmistuneista dokumenteista ja muista ryhmän toimintaan vaikuttaneista asioista. Hän lähetettää raportin sähköpostilla osoitteeseen ohtu@cs.helsinki.fi otsikolla "DHT-ryhmän edistymisraportti".

Raportoinnin eräpäivät ovat:

Kausi	Raportin eräpäivä
19.01.2004 - 01.02.2004	02.02.2004
02.02.2004 - 22.02.2004	23.02.2004
23.02.2004 - 07.03.2004	08.03.2004
08.03.2004 - 21.03.2004	22.03.2004
22.03.2004 - 04.04.2004	05.04.2004
05.04.2004 - 25.04.2004	26.04.2004
26.04.2004 - 09.05.2004	10.05.2004

8.2.2 Päiväkirjat

Jokainen pitää vapaamuotoista päiväkirjaa, johon kirjaa kokemuksia ja mielipiteitä ohjelmistotuotantoprojektikurssista. Päiväkirjat ovat wikissä vapaasti muiden luettavissa ja linkki niihin sijoitetaan jokaisen omalle sivulle. Projektissa mukana olijoiden omat sivut ovat wikissä Projektiryhmä-linkin alla. Päiväkirjojen päivitys noudattaa samoja eräpäiviä kuin tunti- ja edistymisraportointi.

8.3 Versionhallinta

Versionhallintaan käytetään CVS-versionhallintaohjelmaa. CVS-repositorio sijaitsee ryhmän kotihakemistossa TKTL:n palvelimilla hakemistossa /home/group/dht/cvsroot. CVS-repositoriossa on GNUnet:stä asiakkaan tarjoama versio. Asiakas järjesti dhtstart-tägin GNUnet:n CVS-repositorioon, mikä sieltä exportattiin. Projektin oma ohjelmakoodi tulee CVS-repositoriossa hakemistoon GNUnet/src/applications/dht.

CVS-repositoriota voi selata WWW-selaimella cvsweb-työkalun avulla, johon linkki löytyy Wiki-työkalusta.

8.4 Kokoukset

Kokoukset tapahtuvat ryhmän tapaamisajankohtina, jotka ovat maanantaina 16:15 - 18:00 TKTL:n luokassa B476 ja torstaina 16:15 - 18:00 Aleksandrian luokassa 329. Paikalla ovat aina vähintään ryhmän jäsenet sekä ohjaaja. Tämän lisäksi paikalla voi olla tarpeen mukaan myös asiakas sekä kurssin vastuuhenkilö. Sihteeri tekee kokouksista pöytäkirjat ja julkaisee ne Wikissä.

8.4.1 Suunnittelu

Suurin osa kokouksista on suunnittelukokouksia. Niissä valmistellaan työn alla olevaa vaihetta, vaihdetaan ajatuksia ja ideoita sekä jaetaan työvaiheeseen liittyvät tehtävät ryhmän kesken. Ennen tarkastuskokousta olevassa suunnittelukokouksessa lukitaan tarkastuksen kohteeksi tuleva dokumentti muutoksilta.

8.4.2 Seuranta

Osassa kokouksia pidetään alussa erillinen seurantaosuus, johon käytetään korkeintaan puoli tuntia aikaa. Sen aikana käydään läpi työvaiheen ja koko projektin edistymistä, arvioidaan aikataulun pitämistä ja käydään läpi mahdolliset toteutuneet riskit. Seurantaa varten projektipäällikkö tuottaa esityslistan, josta kirjataan erillinen pöytäkirja Wikiin. Projektipäällikön on julkaistava esityslista ja lähetettävä se ryhmän sähköpostilistalle viimeistään päivää aikaisemmin. Projektipäällikkö kirjaa seurannassa mahdollisesti todetut muutokset projektisuunnitelmaan. Seurantaa tehdään joka viidennen suunnittelukokouksen yhteydessä. Ensimmäinen kokouksen seurantaosuus pidetään 12.2.2004.

8.4.3 Tarkastus

Tarkastuskokouksessa käydään läpi valmistunut dokumentti. Jokainen kokoukseen osallistuva saa dokumentin luettavaksi vähintään kaksi päivää aikaisemmin. Lisäksi jokainen on tutustunut siihen ennen kokoukseen tuloa. Dokumenttia ei kokouksen aikana korjata vaan korjaustarpeessa olevat asiat kirjataan ylös ja korjaus tehdään myöhemmin. Dokumenttivastaava julkaisee korjatun version Wikissä sekä lähettää sen kaikille sähköpostin liitetiedostona.

8.4.4 Ohjelmakoodin tarkastus

Tuotettu koodi tarkastetaan hankalimmilta osioilta. Koodin tarkastus kokouksista sovitaan erikseen toteutusvaiheessa.

8.5 Kommunikointi

Ryhmän keskinäinen pääkommunikointikanava on kaksi kertaa viikossa tapahtuvat tapaamiset. Tapaamisten lisäksi on käytössä UseMod Wiki-ympäristö⁷. Täältä löytyy dokumentit, kokospöytäkirjat, linkkejä aiheeseen, ohjeita sekä keskustelupalsta.

Käytössä on myös ryhmäsähköpostiosoite ohtuk04-dht-list@cs.helsinki.fi. Lisäksi voimme tarpeen tullen ottaa käyttöön IRC-kanavan reaaliaikaiseen viestintään. Yhteydenotot puhelimitse on tarkoitettu viimeiseksi hätävaraksi kiireellisten asioiden varalta, joita ei voida muulla keinoin tarpeeksi nopeasti hoitaa.

⁷http://db.cs.helsinki.fi/tkt_dht/

9 Riskianalyysi

DHT-projektin keskeiset riskit liittyvät toteutettavaksi valittuun Kademia-algoritmiin ja GNUnet-alustaan. Lisäksi useat yleiset riskit koskevat tätäkin projektia.

9.1 Kademia-algoritmin toteuttamisessa on ongelmia

Algoritmi on verrattain uusi eikä sen käytännön suorituskyvystä ole kovin paljoa tietoa. Algoritmin käyttäytymisessä saattaa ilmetä odottamattomia ongelmia.

Mikäli ongelmia ilmenee, eivätkä ne johdu virheistä algoritmin toteutuksessa, on mietittävä mikä algoritmin ominaisuus tarkkaan ottaen aiheuttaa ongelman. Kademia-algoritmissa on joitakin empiirisesti määritettyjä säätöparametreja. Ilmeneviä ongelmia kannattaa yrittää ratkaista muuttamalla näiden parametrien arvoja ennen kuin harkitaan algoritmin itsensä muuttamista. Jos katsotaan tarpeelliseksi muuttaa algoritmia itseään, keskustellaan tästä ensin asiakkaan kanssa.

Todennäköisyys: Mahdollinen

Algoritmi ei ole kovin monimutkainen, mutta hajautettujen järjestelmien käyttäytymistä on vaikea ennakoida.

9.2 GNUnet-alustassa on puutteita

GNUnet-alustaa kehitetään aktiivisesti. Alustassa saattaa ilmetä sekä toiminnallisia puutteita että ohjelmointivirheitä. Löydetyistä ohjelmointivirheistä tiedotetaan GNUnetin kehittäjille. Koska GNUnetin lähdekoodi on avointa, virheitä voi korjata itsekin mikäli katsoo tämän tarpeelliseksi ja tuntee GNUnetin toiminnan riittävän hyvin. Määrittelyvaiheessa selvitetään, mitä palveluita DHT-moduuli tarvitsee GNUnetiltä. Mikäli GNUnet ei toteuta lainkaan jotain tarvittua toimintoa, joka kuitenkin kuuluisi perusalustaan eikä DHT-moduuliin, otetaan yhteyttä GNUnetin kehittäjiin ja keskustellaan GNUnet-API:n mahdollisesta laajentamisesta heidän kanssaan.

Todennäköisyys: Pieni

GNUnetin ohjelmakoodin laatua on vaikea arvioida tutustumatta siihen tarkemmin. DHT-moduuli ei tarvitse kovin monimutkaisia palveluita, joten on todennäköistä että GNUnet tarjoaa riittävät palvelut sen toteuttamiseen.

9.3 Työntekijöitä poistuu kesken projektin

Jos joku ryhmäläisistä jättää projektin kesken, jaetaan työtehtävät jäljellä olevien kesken. Projekti on luultavasti riittävän suppea toteutettavaksi vajaallakin ryhmällä.

Todennäköisyys: Pieni

Koska kyseessä on opiskeluprojekti, ei tässä tapauksessa tarvitse olla huolissaan paremman työpaikan perässä poistuvista työntekijöistä. Keskeyttämistä vaativia henkilökohtaisia esteitä saattaa tietenkin edelleen ilmetä.

9.4 Asiakas muuttaa vaatimuksia

Koska projektia toteutetaan vesiputousmallilla, on palattava suunnittelu- tai määrittelyvaiheeseen mikäli vaatimukset muuttuvat kesken projektin. Mikäli näin tapahtuu, pyritään hyödyntämään olemassaolevaa materiaalia niin paljon kuin mahdollista, jotta projekti saataisiin edelleen toteutettua aikataulussa.

Todennäköisyys: Pieni

Toteutettavassa järjestelmässä ei ole käyttöliittymäkomponentteja tai monimutkaista rajapintaa, joista asiakkaalla voisi olla paljon muutosehdotuksia. Asiakas voi esittää järjestelmän suorituskykyä koskevia lisävaatimuksia kesken projektin, mutta tässä ei projektin määritelmää luultavasti tarvitse muuttaa, vaan suunnitteluvaiheeseen palaaminen riittää.

9.5 Aikataulusta myöhästyään

Mikäli asetetusta aikataulusta on myöhästytty, laaditaan seuraavassa ryhmätapaamisessa uusi aikataulu. Jos projektin valmistuminen alkaa näyttää myöhästymisten takia epävarmalta, pyritään karsimaan toteutettavia ominaisuuksia, mikäli tämä on mahdollista.

Todennäköisyys: Mahdollinen

Jos projekti on hyvin suunniteltu, ei aikataulussa pysyminen ole luultavasti ongelma. Riskiskenaarioiden toteutumiseen liittyy kuitenkin yleensä työmäärän lisääntyminen, ja usean riskin toteutuminen saattaa käytännössä johtaa myöhästymiseen aikataulusta.

9.6 Suunnitelman toteuttamisessa ilmenee ongelmia

Suunnitteluvaiheessa ei pystytä ottamaan huomioon kaikkia käytännön seikkoja. Huomioimattomat seikat saattavat jäädä pelkiksi toteutuksen yksityiskohdiksi, mutta jokin myöhemmin ilmenevä ongelma saattaa myös estää toteuttamasta osaa suunnitelmasta. Tällöin suunnitelman tämä osa on kirjoitettava uusiksi niin, että ilmennyt ongelma otetaan huomioon.

Todennäköisyys: Mahdollinen

Ennakoimattomia ongelmia ilmenee melko varmasti. Toteutettavan projektin yksinkertaisuuden vuoksi ne eivät kuitenkaan välttämättä vaadi suunnitelman muuttamista.

10 Laadunvalvonta

10.1 Dokumentaatio

Dokumentaatiossa pyritään

- selkeyteen
- virheettömyyteen
- oleellisuuteen.

Dokumentaation laatua seurataan tarkastuksilla ja erityisesti projektiryhmän sisäisen, tehokkaan *peer review* -prosessin mahdollistavalla Wiki-ympäristöllä.

10.2 Ohjelmakoodi

Ohjelmiston toteutuksessa pyritään ensisijaisesti

- virheettömyyteen
- yksinkertaisuuteen ja selkeyteen
- ylläpidettävyyteen
- vastaavuuteen vaatimusmäärittelyn kanssa.

Ohjelmistokoodin laatua käytännössä määrittäviä tekijöitä ovat

- selkeä integroituvuus jo toteutettuun GNUnet-ympäristöön:
 - ohjelmakoodi on GNUnet-koodistandardien mukaista
 - tarjottu ohjelmointirajapinta (API) on helppo omaksua aiemman GNUnet-tietämyksen valossa
 - on käytetty GNUnetin tarjoamia rutiineja esimerkiksi muistinhallintaan ja perustietorakenteisiin
- *free software* -henkisyys (jotta toteutettu ohjelmakoodi voidaan palauttaa osaksi *upstreamia* on sen seurattava ns. vapaille ohjelmistoille tyypillisiä konventioita).

10.2.1 Ohjelmakoodin testaus

Testaus pyritään mahdollisuuksien mukaan automatisoimaan käyttämällä esimerkiksi xUnit-tyyppisiä kehyksiä testitapausten toteuttamiseen. Tavoitteena on myös toteuttaa Extreme Programming -oppien mukaisesti ensin testitapaukset ja vasta sen jälkeen tapauksilla testattava koodi.

Testausprosessille laaditaan soveltuvat mittarit, joista esimerkkinä testitapausten läpäisyprosentti. Tavoitteena on, että määrittelyvaiheessa pyritään jakamaan toteutettava ohjelmisto selkeisiin toiminnallisyksiköihin. Näin voidaan toteutusvaiheessa seurata toteutuksen edistymistä eli tietty yksikkö on valmis, kun se läpäisee siihen liittyvät testitapaukset.