Turun yliopisto
University of Turku

Isojen ohjelmistojen vaatimusten ja ratkaisujen vanheneminen

Case: Aineistonhallintajärjestelmä HANSABASE

Joon Boucht

Pro gradu -tutkielma
Turun Yliopisto
Tulevaisuuden teknologioiden laitos
Tietojenkäsittelytiede
Kesäkuu 2018

TURUN YLIOPISTO
Luonnontieteiden ja tekniikan tiedekunta
Tulevaisuuden teknologioiden laitos
Tietojenkäsittelytiede

Boucht, Joon Henrik: Isojen ohjelmistojen vaatimusten ja ratkaisujen vanheneminen
Case: Aineistonhallintajärjestelmä HANSABASE

Pro-gradu -tutkielma, 99 s.
Pro-gradu -tutkielman ohjaajat: FT Ville Leppänen ja FT Jukka Teuhola.
Kesäkuu 2018

Tässä tutkielmassa tutkin isojen ohjelmistojen vanhenemiseen johtavia syitä, sekä keinoja vanhenemisen hidastamiseksi, tai peräti estämiseksi. Mallitapauksena käytin Hansaprintin aineistonhallintajärjestelmää HANSABASE Mediapankkia, joka taustoitettiin sekä käsitteiden että järjestelmän toiminnallisuuden ja toimintaympäristön osalta. Tieteellistä pohjaa aiheeseen haettiin D. L. Parnaksen ’Software Aging’ -teoriasta, sekä ketterän kehityksen työkaluista.

HANSABASE -järjestelmän vanhenemista sekä työkaluja järjestelmän vanhenemisen hidastamiseksi tarkastellaan kolmen näkökulman kautta: liiketoimintaympäristön, ’Software Aging’ -teorian ja 2000-luvun alun kehityssuunnitelman.

Tarkastelussa havaitaan, että ’Software Aging’ -teorian mukaisten työkalujen käyttö ja menetelmien noudattaminen, niin ohjelmiston kehityksessä, kuin sen huollossa ja korjauksessakin on ensiarvoisen tärkeää ohjelmiston vanhenemisen hidastamisen kannalta, mutta sen lisäksi vanhenemiseen vaikuttivat monet muutkin tekijät. Vaikuttavia tekijöitä on niin sisäisiä kuin ulkoisiakin. Vanhenemiseen vaikuttavat mm. ’Omistajan valinnat’, ’Tekniikasta vastanneiden valinnat’, ’Emojärjestelmän kehitys ja valinnat’, ’Teknisen henkilökunnan osaaminen’, Asiakkaiden valinnat’, ’Kilpailijat’ ja ’Maailman muuttuminen’. Lisäksi havaitaan selvästi, että ohjelmistojen vanhenemista ei voi estää, mutta vanhenemista voi hidastaa.

HANSABASE -järjestelmän vanhenemiseen vaikutti moni tekijä, ei pelkästään ohjelmakoodin vanheneminen. Suurimmassa osassa olivat ulkoiset tekijät, tärkeimmän asiakkaan tekemät valinnat ja palvelusta luopuminen, mutta myös yrityksen johdon päätökset resurssoinnista ja painotuksista. Sidosryhmillä ja ympäristöllä on siis iso merkitys ohjelmistojen vanhenemisessa!

Ohjelmistojen vanhenemiseen vaikuttavien tekijöiden laaja-alaisuutta kuvaamaan esitän vielä lopuksi ohjelmistojen vanhenemisen sipulimallin sekä esitän että ’Software Aging’-teoria voisi laajentaa poikkitieteelliseksi ja että aihepiirin tutkimusta pitäisi jatkaa.

Avainsanat: ohjelmistojen vanheneminen, software aging, Parnas, ketterät menetelmät, agile, vanhenemisen sipulimalli

Esisanat - taustaa tutkielmalleni ja sen aiheelle.

Kirjoitin vuosituhannen vaihteessa tietojenkäsittelytieteen tutkielmaa aiheena ”Aineistonhallintajärjestelmät. Case: HANSABASE Mediapankki”. Tarkoituksena oli pohtia jo silloin pitkään käytössä olleen järjestelmän kehittämistä tutkien muita kilpailevia järjestelmiä ja ratkaisuja. Pääsin työssäni tuolloin jo melko pitkälle, mutta sitten elämä ja sen monta käännettä tulivat väliin, ja tutkielmani jäi kesken. Missään vaiheessa tarkoituksena ei ollut kokonaan jättää kirjoittamista, mutta aikaa kului. Kului yli kymmenen vuotta, mutta vihdoin päätin, että olisi aika saattaa työ valmiiksi. Otin tuolloin yhteyttä nykyisiin ohjaajiini Ville Leppäseen ja Jukka Teuholaan, joiden kanssa ryhdyimme miettimään että, miten saisimme työn hoidettua kunnialla loppuun. Löytyisikö joku malli, jolla yli 10 vuotta sitten kerättyä materiaalia ja kirjoitettua tekstiä saisi hyödynnettyä mahdollisimman hyvin, niin että aiheesta saisi ajankohtaisen?

Varsin pian, Ville Leppäsen loistavasta ideasta, päädyimme muuttamaan tutkielman tieteellisen näkökulman nykyään kovin ajankohtaiseen, mutta vielä aika vähän tutkittuun teemaan: ”Ohjelmistojen vanheneminen - Software Aging”. HANSABASE järjestelmä oli tuolloin vanhenemassa, ja syitä vanhenemiseen olisi hyvä pohtia. Tällä ajatuksella ja tuolta pohjalta voisin parhaiten hyödyntää aiemmin tekemääni työtä, tuoda sitä tähän päivään. Saisin mahdollisuuden rakentaa tutkielmastani mielenkiintoisen, relevantin ja mahdollisimman ajankohtaisen kokonaisuuden.

Kiitos tästä uudesta mahdollisuudesta hyvät herrat!

Lisäksi suurkiitos rakkaille tukijoukoilleni kotona erittäin pitkästä pinnasta tämän (lähes) ikuisuusprojektini kanssa.

Sisällysluettelo

Esisanat - taustaa tutkielmalleni ja sen aiheelle. ii

1 Johdanto 1

2 ’Software Aging’ - ohjelmistojen ja järjestelmien vanheneminen 4
2.1 Mitä on ohjelmistojen vanheneminen - ’Software Aging’? 5
2.1.1 Vastaväitteitä 5
2.1.2 Ohjelmistot vanhenevat! 5
2.2 Mikä aiheuttaa ohjelmistojen vanhenemisen 6
2.2.1 Tekemättömät muutokset 6
2.2.2 Epäpätevää leikkaushoitoa 6
2.2.3 Kaikki ei ole sitä miltä näyttää 7
2.3 Vanhenemisen hinta 7
2.3.1 Kyvyttömyys pysyä vauhdissa 7
2.3.2 Heikentynyt suorituskyky 7
2.3.3 Vähentynyt luotettavuus 8
2.4 Vanhenemisen vaikutusten ennaltaehkäisy 8
2.4.1 Onnistumiseen tähtäävä suunnittelu – Design for Success 8
2.4.2 Parempaa kirjanpitoa, eli dokumentoinnin merkitys 10
2.4.3 Ulkopuoliset mielipiteet, eli ohjelmistokatselmukset 10
2.4.4 Miksi ohjelmistojen vanhenemista ei voi estää? 11
2.5 Ohjelmistojen geriatriaa 11
2.5.1 Heikkenemisen pysäyttäminen 11
2.5.2 Dokumentointi jälkikäteen 12
2.5.3 Modularisointi jälkikäteen 12
2.5.4 Amputaatio 12
2.5.5 Laajan ohjelmakoodin leikkaus ja uudelleenjärjestely 12
2.6 Yhteenveto oikeasta toimintamallista 13
2.6.1 Asenteita on muutettava 13
2.6.2 Dokumentoi ja katselmoi 13
2.6.3 Vanheneminen on ennakoitava 14
2.7 Ketterät menetelmät teknisen velan hoidossa 14
2.7.1 Teknisen velan luokittelu 14
2.7.2 Erilaisia tekniikoita Agilessa teknisen velan syntymisen estämiseksi 17
2.7.3 Erilaisia tekniikoita Ketterässä kehityksessä syntyneen teknisen velan hoitamiseen 19
2.8 Vielä lopuksi 21

3 Case: Aineistonhallinta ja aineistonhallintajärjestelmät vuosituhannen taitteessa 22
3.1 Mitä on digitaalinen aineistohallinta 23
3.2 Jako tuotteisiin ja palveluihin 24
3.3 Jako käyttöryhmän mukaan 25
3.4 Jako toiminnallisuuden mukaan 26
3.4.1 Aineistojen luettelointi 26
3.4.2 Aineistojen arkistointi ja varastointi 27
3.5 Jako teknisen toteutustavan mukaan 27
3.6 Käyttökohteita 28
3.6.1 Kuva-arkisto 29
3.6.2 Dokumentti-arkistot 30
3.6.3 Brändin, eli tuotemerkin hallinta 30
3.6.4 Monimediajulkaiseminen 30
3.6.5 Web-to-print –sovellukset 31
3.6.6 Painotöiden valmistus 31
3.7 Aineistonhallintaan liittyviä keskeisiä käsitteitä vuosituhannen taitteessa 32
3.8 Tiedon tallennusmuodot 36
3.9 Aineistonhallintaan liittyviä suosituksia 37
3.10 Tekninen toteutus 40
3.10.1 Dokumenttien varastointi, tietokantatekniikat 40
3.10.2 Tekstitietokannat vs. relaatiotietokannat 41

3.10.3 Natiivi XML –tietokannat vs. relaatiotietokannat 42
3.10.4 Rajapinnat ja integrointi muihin järjestelmiin 43
3.11 Aineistonhallintajärjestelmien komponentit 44
3.11.1 Dokumenttien tallentaminen järjestelmään 44
3.11.2 Dokumenttien luokittelu, metadata 45
3.11.3 Dokumenttien etsiminen ja käyttöönotto 46
3.11.4 Käyttöoikeuksien hallinta 46
3.11.5 Dokumenttien versiohallinta ja seuranta 46

4 Case: Hansaprintin aineistonhallintaratkaisu vuonna 2002: HANSABASE Mediapankki 47
4.1 Hansaprintin esittely 47
4.2 Aineistonhallinnan merkitys Hansaprintille 48
4.3 Hansaprintin aineistonhallintaratkaisu: HANSABASE Mediapankki 49
4.4 Teknologia 50
4.4.1 Arkkitehtuuri 50
4.4.2 Laitealusta 51
4.4.3 Tietokanta ja hakukieli 51
4.5 Ominaisuudet 52
4.5.1 Rakenne 52
4.5.2 Tiedostomuodot 52
4.5.3 Käyttäjäoikeudet 52
4.5.4 Tietokantahaut 53
4.5.5 Materiaalin syöttö 54
4.5.6 Materiaalin valinta ja käyttöönotto 55
4.5.7 Käyttötuki 55
4.5.8 Järjestelmän ylläpito 55
4.5.9 Raportointi 56
4.5.10 Tietoturva 56
4.5.11 Lisäominaisuudet 56
4.6 Emojärjestelmä: Grafimedian MediaVu (os. Maria) 58
4.6.1 Esittely 58
4.6.2 Historiaa 58
4.6.3 Ajatuksia HANSABASE Mediapankin kehityksestä vuonna 2002 60

5 Näkökulmat tilanteeseen 66
5.1 Mitä tapahtui HANSABASELLE, näkökulma liiketoimintaympäristön kannalta 66
5.1.1 HANSABASE-asiakkaita ja erilaisia palvelun käyttötapoja 67
5.1.2 Mitä tapahtui suurimmalle asiakkaalle 72
5.1.3 HANSABASEn kehitys emojärjestelmän kannalta 75
5.2 Miten HANSABASE vanheni, näkökulma ’Software Agingin’ kannalta 75
5.2.1 Tekemättömät muutokset 76
5.2.2 Epäpätevää leikkaushoitoa 76
5.2.3 Kaikki ei ole sitä miltä näyttää 77
5.2.4 Vanhenemisen hinta - Kyvyttömyys pysyä vauhdissa 77
5.2.5 Vanhenemisen hinta - Heikentynyt suorituskyky 77
5.2.6 Vanhenemisen hinta - Vähentynyt luotettavuus 78
5.2.7 Vanhenemisen vaikutusten ennaltaehkäisy HANSABASE -järjestelmän kannalta 78
5.2.8 Onnistumiseen tähtäävä suunnittelu – Design for Success 78
5.2.9 Parempaa kirjanpitoa, eli dokumentoinnin merkitys 79
5.2.10 Ulkopuolisten mielipiteet, eli ohjelmistokatselmukset 80
5.2.11 Miksi ohjelmistojen vanhenemista ei voi estää? 80
5.2.12 Ohjelmistojen geriatriaa 80
5.2.13 Heikkenemisen pysäyttäminen 80
5.2.14 Dokumentointi jälkikäteen 81
5.2.15 Modularisointi jälkikäteen 81
5.2.16 Amputaatio 81
5.2.17 Laajan ohjelmakoodin leikkaus ja uudelleenjärjestely 81
5.2.18 HANSABASEn vanhenemisen kannalta merkittävimmät kohdat 82
5.3 HANSABASElle 2000-luvun alussa tehdyn kehityssuunnitelman vaikutus 82

6 Pohdintaa 87
6.1 Taustaa 87
6.2 HANSABASEn erilaisen käytön merkitys vanhenemiselle 88

6.3 ’Software Agingin’ parhaat työkalut HANSABASEn vanhenemista vastaan....88
6.4 HANSABASEn kehityssuunnitelman parhaat työkalut vanhenemista vastaan....90
6.5 Yhteenveto vanhenemiseen vaikuttaneista syistä....91

7 Loppupäätelmät....94
7.1 Vanhenemisen sipulimalli....94
7.2 Loppusanat....96

1 Johdanto

Tutkielmassani tutkin ohjelmistojen vanhenemiseen johtavia syitä, sekä mitä voitaisiin tehdä vanhenemisen hidastamiseksi, tai peräti estämiseksi. Mallitapauksena käytin Hansaprintin aineistonhallintajärjestelmää HANSABASE Mediapankkia, joka taustoitettiin sekä käsitteiden että järjestelmän toiminnallisuuden ja toimintaympäristön osalta.

Hansaprintin ja HANSABASEn lyhyt esittely

HANSAPRINT Oy[1] oli vuonna 2001 Suomen suurin painotalo, joka tarjosi asiakkailleen myös jatkuvasti monipuolistuvia digitaalisia lisäpalveluja. Hansaprintin liikevaihto vuonna 2000 oli 880 Mmk. Tuolloin arvioitiin, että vuonna 2001 Hansaprint ylittäisi yhden miljardin Suomen markan rajan.

Kuva 1.1. Hansaprint Oy:n ja HANSABASE Mediapankin logot noin vuodelta 2000

HANSABASE Mediapankki oli yrityskohtainen digitaalisen media-aineiston hallintajärjestelmä, joka helpotti ja nopeutti digitaalisen median kanssa työskentelyä. Palvelu oli tarkoitettu ensisijaisesti digitaalisen kuva-aineiston hallinta- ja jakelujärjestelmäksi sekä valmiiden dokumenttimuotoisten aineistojen ja multimedian tallentamiseen keskitettyyn tietokantaan. Hansaprint kehitti HANSABASEn lisäpalveluksi painoasiakkailleen.

Mikä on tutkielman haaste

Tutkielmani perusoletuksena on, että HANSABASE -järjestelmälle tapahtui vuosien kuluessa jotain, jota voidaan kutsua vanhenemiseksi ja työn tarkoitus on sitä vasten tarkastella vanhenemista ilmiönä pureutuen sen syihin.

Edellä kuvatulta pohjalta olen hahmotellut tutkielman haasteita. Pohjana oli ajatus, että haasteiden tulisi olla sellaisia, että mikäli niihin löydettäisiin tutkielmassa vastaus, olisi silloin päästy oikeasti hyödylliseen lopputulokseen.

[1] http://www.hansaprint.fi

Haasteet joihin päädyin ovat seuraavat:

- H1: Mistä ohjelmistojen vanheneminen johtuu ja mikä siihen johtaa?
- H2: Mitä voisi tehdä ohjelmistojen vanhenemisen hidastamiseksi?
- H3: Voiko ohjelmistojen vanhenemisen estää kokonaan?

Asiaa lähdettiin tutkimaan ajatuksella, että ohjelmistojen vanheneminen johtuisi erityisesti ohjelmalle tehdyistä ja tekemättömistä ohjelman suunnittelijoiden ja ohjelmoijien teoista. Hypoteesina se, että ’Ohjelmat vanhenevat ajan kuluessa väistämättä’.

Miten asiaa on tässä tutkittu

Aihetta on tutkielmassani tutkittu ensin läpikäymällä aihepiiriin liittyvää tutkimusta ja julkaisuja (luku 2). Artikkeleita aiheesta ’Software Aging’ löytyy jonkin verran. Osa näistä on alueen taustaa läpikäyviä ja perusteita määritteleviä, osa enemmänkin ”War Story”- tyyppisiä kertomuksia ohjelmistojen vanhenemisesta, ja osassa syvennytään aihepiiriin matemaattisten mallien kautta. Tämän tutkielman näkökulmaan sopivat parhaiten ensimmäisen tyypin artikkelit. ”War Story”-tyyppiset, sekä aihepiirin matemaattinen tausta jätetään tutkielman ulkopuolelle.

Tutkielman luvussa ’2. ’Software Aging’ - ohjelmistojen ja järjestelmien vanheneminen”, käydään läpi ohjelmistojen vanhenemisen teoreettista taustaa teorian alkutaipaleelta alkaen, mutta myös modernimmalla näkökulmalla ketterän kehityksen näkökulmasta. Mikä aiheuttaa ohjelmien ja ohjelmistojen vanhenemista, ja mitä voidaan tehdä sen ehkäisemiseksi tai hillitsemiseksi.

Tutkielman luvussa ’3. Case: Aineistonhallinta ja aineistonhallintajärjestelmät vuosituhannen taitteessa’ luodaan perustaa tutkielman Case -osiolle, eli esitellään käsitteet aineistonhallinta ja aineistonhallintajärjestelmät, minkä tyyppisiä aineistonhallintajärjestelmiä tuolloin oli, minkälaisia käsitteitä ja toimintoja niihin liittyi.

Tutkielman luvussa ’4. Case: Hansaprintin aineistonhallintaratkaisu vuonna 2002: HANSABASE Mediapankki’ esitellään HANSABASE -järjestelmä tuolloin, sekä ajatuksia sen kehittämisestä vuonna 2002. Sekä luvut 3 että 4 perustuvat tilanteeseen tuolloin myös käsitteiden tulkinnan osalta.

Tutkielman luvussa ’5. Näkökulmat tilanteeseen’ käydään läpi HANSABASE-ohjelmiston vanhenemista kolmen näkökulman kautta. Ensin mietitään syitä HANSABASE-palvelun

vanhenemiselle, kun asiaa tarkastellaan vallinneen liiketoimintaympäristön näkökulmasta. Analyysi perustuu pitkälti omalle käsitykselleni tapahtuneesta. Toiseksi analysoidaan tapahtunutta käyttäen näkökulmana tutkielman luvussa 2 esille tuotua ’Software Aging’ -teoriaa. Ja kolmanneksi käydään läpi mitä ohjelmiston vanhenemiseen olisi vaikuttanut 2000-luvun alussa tehdyn HANSABASE-kehityssuunnitelman toteuttaminen aikanaan.

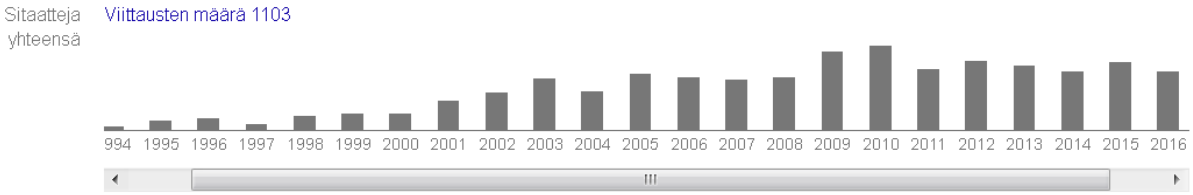
Luvussa ’6. Pohdintaa’ kolmen näkökulman kautta vanhenemista ilmiönä. Millaisia syitä löydettiin, ja mitkä HANSABASEn tapauksessa olivat syistä tärkeimmät, ja mitkä työkalut tehokkaimmat vanhenemisen hidastamiseen. Miten haasteisiin H1-H3 on saatu vastauksia, ja mitä vastauksia. Oliko tehty hypoteesi oikea?

Luvussa ’7. Loppupäätelmät’ esitetään uusi sipulimalli, jolla pystytään kuvaamaan ohjelmistojen vanhenemiseen vaikuttavien tekijöiden laaja-alaisuutta. Mietitään ’Software Aging’ -teorian kattavuutta ja laajentamisen mahdollisuuksia ja tarvetta, sekä tehdään ehdotus aiheesta jatkotutkimukselle.

2 ’Software Aging’ - ohjelmistojen ja järjestelmien vanheneminen

Tutkielmani teoriaosuus, ”Software Aging, ohjelmistojen vanheneminen” perustuu pitkälti D.L Parnasin artikkeliin ”Software Aging” (Parnas 1994), joka julkaistiin vuonna 1994 konferenssijulkaisussa ”Proceedings of the 16th international conference on Software engineering”.

Artikkeliin on viitattu sen julkaisun jälkeen jo yli 1000 kertaa, joten sitä voidaan pitää erittäin tärkeänä lähteenä tässä aihepiirissä, ja jopa koko ’Software Aging’ -keskustelun pohjana.



			Viittauksia					Viittauksia
Vuosi	**Viittausta**	**Kumula-tiivinen**	**Vuodesta**		**Vuosi**	**Viittausta**	**Kumula-tiivinen**	**Vuodesta**
1993	4	4	1089		**2006**	58	375	772
1994	3	7	1085		**2007**	54	429	714
1995	10	17	1082		**2008**	57	486	660
1996	13	30	1072		**2009**	84	570	603
1997	7	37	1059		**2010**	91	661	519
1998	15	52	1052		**2011**	65	726	428
1999	17	69	1037		**2012**	75	801	363
2000	17	86	1020		**2013**	70	871	288
2001	32	118	1003		**2014**	64	935	218
2002	40	158	971		**2015**	73	1008	154
2003	56	214	931		**2016**	64	1072	81
2004	42	256	875		**2017**	17	1089	17
2005	61	317	833					

Kuva 2.1: Parnasin artikkeli ohjelmistojen vanhenemisesta vanhenemistutkimuksen lähteenä

Myös toisia luokitteluja löytyy, mutta pääsääntöisesti ne perustuvat Parnasin luokitteluun. Näitä toisia luokitteluja ei ole esitelty tässä tutkielmassa.

Seuraavassa kappaleessa lähteenä on pääosin Parnasin edellä mainittu artikkeli, joten sitä ei ole erikseen joka kohtaan merkitty.

2.1 Mitä on ohjelmistojen vanheneminen - ’Software Aging’?

Käsitteen jo 1990-luvun puolivälissä tieteelliseen keskusteluun tuonut D. L. Parnas on pukenut ajatuksen seuraavaan muotoon: ”Ohjelmat, niin kuin ihmisetkin, vanhenevat. Vanhenemista ei voi estää, mutta voimme ymmärtää mikä sen aiheuttaa, suorittaa toimenpiteitä sen vaikutusten rajoittamiseksi, väliaikaisesti kumota osan sen aiheuttamasta vahingosta, valmistautua päivään jona ohjelma ei enää voi toimia … meidän on luovuttava innosta saada valmiiksi ohjelman ensimmäinen versio, ja keskityttävä tuotteidemme pitkäaikaiseen terveyteen.” (Parnas 1994.)

2.1.1 Vastaväitteitä

Käsite tai edes koko vanhenemisprosessin olemassaolo ei ole kiistaton. Sitä vastaan on esitetty monenlaisia vastaväitteitä. On muun muassa todettu, että ohjelmat eivät rappeudu, sillä ohjelmat ovat matemaattisia kokonaisuuksia, eikä matematiikka rappeudu. On myös väitetty, että jos teoreema oli virheetön 200 vuotta sitten, on se sitä vielä huomennakin, siis se ei vanhene. Toisin sanoen, jos ohjelma on oikein tänään, on se sitä vielä sadan vuodenkin päästä, ja päinvastoin, jos ohjelma on väärin sadan vuoden päästä, on sen täytynyt olla väärin jo silloin kun se kirjoitettiin. (Parnas 1994.)

Kaikki edellä esitetty pitää periaatteessa paikkansa, mutta on kuitenkin asioita, jotka ovat unohtuneet tällaisia perusteita esittäviltä.

2.1.2 Ohjelmistot vanhenevat!

Ohjelmistojen vanheneminen on yhä tärkeämpi ilmiö ohjelmistojen taloudellisen merkityksen jatkuvasti kasvaessa. Monelle korkean teknologian yritykselle ohjelmistot ovat yrityksen pääomaa. On luonnollista, että uusien ohjelmistotuotteiden kirjoittajat ja omistajat ajavat omaa asiaansa, eivätkä kannata vanhenevien ohjelmistojen elvyttämistä, eivätkä osaa niitä muutenkaan arvostaa.

On myös esitetty (Parnas 1994), että nykyään asiat olisivat toisin kuin aiemmin, eli että vanhenemista ei enää tapahtuisi. Siis jos ohjelmistot olisivat alkujaankin suunniteltu käyttäen tämän päivän ohjelmointikieliä ja tekniikoita, ei tässä tilanteessa oltaisi. He siis väittävät, että nyt valmistettavat ohjelmat eivät tulisi koskaan vanhenemaan. Väite loukkaa vanhoja ohjelmoinnin mestareita. Ikään kuin aiemmin ei olisi osattu ohjelmoida.

2.2 Mikä aiheuttaa ohjelmistojen vanhenemisen

Parnasin mukaan, on olemassa kahden tyyppistä, toisistaan selvästi eroavaa ohjelmistojen vanhenemista (Parnas 1994). Ensimmäiseksi sellaista, joka aiheutuu siitä, että tuotteen omistaja ei ole tehnyt ohjelmistoon tarpeellisia muutoksia, tai kehittänyt sitä riittävästi. Ohjelmistoon kohdistuvien ulkoisten vaatimusten muuttumista ei ole huomioitu, eikä ohjelmistoa ole pidetty ajan tasalla. Toiseksi sellaista, joka aiheutuu nimenomaan ohjelmistoon tehdyistä muutoksista. Näiden kahden yhteisvaikutus voi johtaa ohjelmistotuotteen arvon nopeaan putoamiseen. (Parnas 1994).

2.2.1 Tekemättömät muutokset

Kun ohjelmistoa ei päivitetä riittävän usein, johtaa se ajan myötä siihen, että käyttäjät tulevat tyytymättömiksi ja vaihtavat uuteen tuotteeseen. Erinomainen 1960-luvulla kehitetty ohjelmisto voisi toimia tänäänkin täydellisesti, mutta kukaan ei käyttäisi sitä! Ohjelmisto on vanhentunut, vaikka kukaan ei ole siihen koskenutkaan. Itse asiassa se on vanhentunut, **koska** kukaan ei ole sitä päivittänyt. (Parnas 1994.)

Maailmankuvan muuttuessa käyttäjien käsitykset ohjelmistoihin liittyvistä tarpeista muuttuvat, mikä luo ohjelmistojen kehittämistyöhön ja ohjelmistoille sinänsä muutospaineita. Tämä on aivan luonnollista kehitystä, eikä siis ohjelmiston aiempien versioiden toiminnallinen vika.

2.2.2 Epäpätevää leikkaushoitoa

Ohjelmistoa on siis päivitettävä, jotta se ei vanhenisi. Mutta myös ohjelmiston muuttaminen voi aiheuttaa vanhenemista. Syypäitä tähän ovat muutoksia tekevät ihmiset, jotka eivät ymmärrä ohjelmistoa riittävän hyvin. Tällöin ohjelman rakenne heikkenee. Kun tällaisia ammattitaidottomia muutoksia tehdään riittävän monta, **ei kukaan** enää ymmärrä ohjelmistoa. Ohjelmiston alkuperäiset suunnittelijat eivät **enää** ymmärrä muutettua ohjelmaa, eivätkä muutoksien tekijät **vieläkään** ymmärrä ohjelmaa. Eli jäljellä ei ole enää ketään, joka ymmärtäisi sitä.

Ylläpidosta tulee koko ajan kalliimpaa. Uudet muutokset vievät aina vain kauemmin ja aiheuttavat uusia virheitä. Ja koska ylläpitäjien mielestä heillä ei ole aikaa dokumenttien ylläpitoon, tulee ohjelman dokumentaatiosta aina vain epäjohdonmukaisempaa ja epätarkempaa, joka edelleen hankaloittaa tulevien muutosten tekemistä.

2.2.3 Kaikki ei ole sitä miltä näyttää

Ohjelmissa olevat muistin allokointi- ja vapautusongelmat yhdistetään usein ohjelmiston vanhenemiseen, mutta kyse ei ole samasta asiasta. Jos ohjelma ei hoida muistinkäsittelyä kunnolla, voi muisti täyttyä ja työtiedostot paisua, aiheuttaen ohjelmiston toiminnan heikentymistä. Tämän kuitenkin usein johtuu ohjelman suunnitteluviasta, joten se voi ”iskeä” ohjelmaan milloin tahansa ohjelman elinkaaren aikana. Mainittu ongelma on kuitenkin usein paljon helpompi korjata kuin tutkielman aiheena oleva ”ohjelmiston vanheneminen”. Järjestelmän muisti ja levyjärjestelmä voidaan puhdistaa, paremmat rutiinit voivat nopeuttaa siivouksen alkamista, jonka jälkeen ohjelmistoa voidaan pitää jopa täysin ”parantuneena”.

2.3 Vanhenemisen hinta

Ohjelmistojen vanhenemisoireet muistuttavat paljon ihmisen vastaavia vanhenemisoireita:

1. Kyvyttömyyttä pysyä vauhdissa: ohjelmistojen omistajat huomaavat, että pysyminen markkinoiden vauhdissa on yhä vaikeampaa ja menettävät samalla asiakkaita uudemmille tuotteille.
2. Heikentyvää suorituskykyä: ohjelmistojen tilan ja ajankäytön tehokkuus vähenevät asteittain ohjelman rapistuvan rakenteen vuoksi.
3. Vähenevä luotettavuutta: vanhentuvan ohjelmiston suoritusympäristön muutoksista aiheutuvat ongelmat luotettavuudelle.

Ja kaikki nämä muutokset aiheuttavat omistajalle todellisia kustannuksia.

2.3.1 Kyvyttömyys pysyä vauhdissa

Ohjelman vanhetessa sen koko väkisinkin kasvaa. Koodin lihominen johtuu siitä, että helpoin tapa lisätä ohjelmistoon ominaisuuksia on lisätä siihen uutta koodia. Koon kasvaessa muutosten tekeminen tulee hankalammaksi, sillä hallittavaa koodia on koko ajan enemmän ja silloin oikeiden muutettavien rutiinien löytäminen on hankalampaa ja muutoksia pitää tehdä yhä useampaan paikkaan. Lopputuloksena on, että muutoksia ei tehdä riittävän nopeassa tahdissa. Saadakseen uusia ominaisuuksia asiakkaat vaihtavat ”nuorempaan” ohjelmistoon.

2.3.2 Heikentynyt suorituskyky

Ohjelmakoon kasvaessa sen tarvitseman muistin määrän kasvaa. Saadakseen riittävän suorituskyvyn on asiakkaiden päivitettävä koneitaan. Suoritus heikkenee johtuen ohjelman surkeasta rakenteesta, joka on seurausta pitkäaikaisesta ad-hoc -ylläpidosta. ”Nuorempi” tuote on nopeampi ja tarvitsee vähemmän muistia, koska se on jo alun perin suunniteltu tukemaan uusia ominaisuuksia.

2.3.3 Vähentynyt luotettavuus

Ohjelmaa ylläpidettäessä syntyy lisää virheitä. Jo ohjelmoinnin alkuaikoina todettiin monessa projektissa (Parnas 1994), että usein kun ohjelmiston virhe- tai häiriömäärää yritetään vähentää, onkin häiriömäärä kasvanut. Keskimääräisesti ottaen kutakin korjattua virhettä kohden syntyy useampi kuin yksi uusi. Usein vaihtoehtoina on joko projektin hylkääminen kokonaan tai virheiden korjaamisen lopettaminen. Parnas kertoo kuulleensa eräästä kaupallisesta tuotteesta, jolle oli kertynyt yli 2000 korjaamattoman tunnetun virheen lista. (Parnas 1994.)

2.4 Vanhenemisen vaikutusten ennaltaehkäisy

Ohjelmia suunnitellessa tulisi ajatella paljon pidemmälle kuin vain ensimmäisen version julkaisuun. Tulisi suunnitella ennakoiden ohjelman vanheneminen. Kokemattomille ohjelmoijille tulee onnistumisen tunne jo ensimmäisen onnistuneen käännöksen tai demonstraation jälkeen. Kokeneet ohjelmoijat kuitenkin ymmärtävät kyseessä olevan vasta pidemmän prosessin alku.

Vastuulliset ja asiantuntevat organisaatiot ymmärtävät, että ohjelman ensimmäisen onnistuneen ajokerran ja ensimmäisen julkaisuun pääsevän version välillä tarvitaan enemmän työtä, kuin on tarvittu ensimmäisen onnistuneen suorituksen aikaansaamiseen. Laajamittainen ohjelmiston testaaminen ja tiukat katselmukset sen kehityksen aikana ovat todella tarpeellisia.

Koska ohjelmistojen vanheneminen on niin suuri ongelma, on erittäin tärkeää löytää keinoja vanhenemisen hidastamiseen ja sen vaikutusten rajaamiseen. Onneksi tilannetta voidaan helpottaa monellakin tavalla. Seuraavassa käsitellään kolmea metodia: 1. onnistumiseen tähtäävää suunnittelua, 2. parempaa kirjanpitoa, eli dokumentaation merkitystä sekä 3. toisten mielipiteiden kuuntelua, eli ohjelmistokatselmusten merkitystä.

2.4.1 Onnistumiseen tähtäävä suunnittelu – Design for Success

Jo ohjelmiston suunnitteluvaiheessa on huomioitava, että ohjelmisto tulee väistämättä muuttumaan tulevaisuudessa.

Muuttumisen hallintaa auttamaan on aikojen kuluessa syntynyt tai kehitetty useita ohjelmointitekniikoita. Tällaisia ovat mm. informaation piilottaminen (information hiding), käsitteellistäminen (abstraction), vastuiden erittely (separation of concerns), datan piilottaminen (data hiding) ja olioperusteisuus (object-orientation).

Suunnittele huomioiden tulevat muutostarpeet

Tämän ensimmäisen periaatteen soveltaminen alkaa sillä, että yritetään hahmottaa tuotteen elinkaaren aikana todennäköisesti tapahtuvat muutokset.

Koska varsinaisia muutoksia ei pysty tarkasti ennustamaan, on niitä mietittävä muutosten luonteen kautta. Tällaisia muutostyyppejä ovat käyttöliittymään tulevat muutokset, toimintaympäristön muutokset (esim. mahdollinen siirtyminen uuteen ikkunointijärjestelmään), tiedon esittämistavan muutokset sekä ohjelmiston mahdollinen vienti uuteen käyttöjärjestelmään.

Koska on mahdotonta tehdä kaikkia muutoksia yhtä helpoiksi, on tärkeää arvioida eri muutostyyppien todennäköisyydet sekä suunnitella ohjelmisto niin, että kaikkein todennäköisimmin tulevat muutokset rajataan mahdollisimman pieneen osaan koodia. Näin muutosten aiheuttama koodaustarve pienenee ja tulee hallitummaksi.

Miksi muutosten huomioiminen suunnittelussa unohdetaan

Vaikka edellä mainittu yksinkertainen metodi muutosten hallinnalle on laajalti hyväksytty vaikuttaa siltä, että sitä noudatetaan ohjelmistojen suunnittelussa harvoin. Onkin oleellista miettiä mistä tämä mahtaisi johtua.

Ensinnäkin ohjelmoinnin oppikirjoissa metodi saatetaan mainita, mutta niissäkään ei varsinaisesti opeteta eri muutostyyppien todennäköisyyksien arviointiprosessia. Parnaksen mukaan asiaa käsitellään kuitenkin vain pinnallisesti, puhutaan ohjelman yksityiskohtien piilottamisesta ja käsitteellistämisestä, mutta eri muutosluokkien todennäköisyyksien arviointia ei varsinaisesti opeteta. (Parnas 1994.)

Toiseksi syynä voi olla tekijöiden kiire tai kärsimättömyys. Ohjelmoijat eivät arvioi muutoksia etukäteen, sillä he ovat liian kärsimättömiä. He vain pyrkivät turhan innokkaasti saamaan ensimmäisen version ohjelmasta toimimaan tai sitten pääsemään asetettuun tavoiteaikatauluun. Johdon päällimmäisenä tavoitteena on pelkästään seuraava deadline, jolloin muutosten hallinta jää liian pienelle huomiolle.

Kun aikataulu- tai muiden ulkoisten paineiden takia tehdään tarkoituksellinen päätös osittaisen ratkaisun toteuttamisesta, tai ongelmatilanteen vain osittaisesta hoitamisesta, syntyy niin sanottua teknistä velkaa (technical debt) (Holvitie ym 2016). Teknisestä velasta ja sen hoidosta enemmän kappaleessa 2.7.

Kolmantena mahdollisuutena tulee metodin ”luonnottomuus”: esimerkiksi tarkalla ”informaation piilottamisen periaatteella” tehdyt suunnitelmat ovat varsin erilaisia kuin luonnolliset ohjelmoijan intuitioon perustuvat suunnitelmat. Jopa silloin, kun ohjelmoijaa erikseen pyydetään noudattamaan periaatetta, ei asia ole hänelle selvä, eikä sitä silloin noudateta, kuten pitäisi.

Neljäs syy voi olla vääränlaisen koodin jäljittely: toistaiseksi löytyy vain vähän hyviä esimerkkejä sovelluksista, jotka on toteutettu noudattaen näitä suunnitteluperiaatteita. Ja koska suunnittelijoilla on taipumus jäljitellä näkemiään ratkaisumalleja, ei uusissa ohjelmissakaan metodia noudateta.

Viidentenä saattaa olla kyseessä käsitteiden sekoittaminen. Ohjelmoijat ovat usein taipuvaisia sekoittamaan suunnitteluperiaatteet ja ohjelmointikielet toisiinsa. Mahdollisesti he ajattelevat, että jos käytössä ei ole objektiorientoitunutta ohjelmointikieltä, silloin ei voisi noudattaa objektiorientoituneita suunnittelumalleja. Tai vastaavasti he kuvittelevat, että pelkkä objektiorientoituneen ohjelmointikielen käyttäminen tekisi ohjelmoinnista oikeanlaista. Kumpikaan edellä mainituista ei pidä paikkaansa. (Parnas 1994).

Kuudentena syynä voi olla ammattitaidottomuus. Edelleenkin monelta ammattilaiselta puuttuu riittävä ohjelmistosuunnittelun koulutus, jolloin tällainen tärkeä asia jää helposti kokonaan huomiotta.

Yhteenvetona edellä olevasta voidaankin Parnasin tavoin todeta, että muutoksen huomioonottava suunnittelu on menestykseen tähtäävää suunnittelua!
(Parnas 1994)

2.4.2 Parempaa kirjanpitoa, eli dokumentoinnin merkitys

Vaikka ohjelma olisi hyvin suunniteltu, on se usein dokumentoitu huonosti tai voi olla jopa kokonaan dokumentoimatta. Jos dokumentaatiota löytyy, on se usein huonosti järjestettyä, epäjohdonmukaista, epätäydellistä tai sellaisen henkilön kirjoittamaa, joka ei oikeastaan tunne järjestelmää. Tästä johtuen ylläpitäjät saattavat jättää dokumentaation kokonaan huomiotta. Vielä pahempaa on se, että esimiehet jättävät dokumentoinnin usein hyvin pienelle huomiolle, sillä sen tekeminen ei nopeuta ohjelmiston ensimmäistä julkaisua.

2.4.3 Ulkopuoliset mielipiteet, eli ohjelmistokatselmukset

Tekniikan aloilla, kuten lääketieteessäkin, toisien ammattilaisten tekemien katselmusten ja tarkastusten tarvetta ei kyseenalaista kukaan. Esim. talon, laivan tai lentokoneen

suunnitteluun liittyy aina tietty määrä suunnitteludokumentteja, jotka käyvät läpi muiden spesialistien huolellisen tarkastuksen.

Näin ei perinteisesti ole ohjelmistoteollisuudessa, sillä 1. monella ohjelmoijalla ei ole lainkaan ohjelmoinnin ammattikoulutusta ja ohjelmoinnin tutkinnoissa ei painoteta kurinalaisuutta. 2. Laatutarkastajien löytäminen on hankalaa, sillä projekteilla ei ole varoja ulkopuolisten asiantuntijoiden palkkaamiseen. 3. Aikataulupaineista johtuen suunnittelijat ajattelevat, että asianmukaisiin katselmuksiin ei ole aikaa. 4. Moni ohjelmoija ei voi sietää ajatusta, että hänen työnsä tulisi jonkun toisen tarkastamaksi katselmuksissa.

Jokainen suunnitelma pitäisi tulla katselmoiduksi ja hyväksytyksi ajoissa sellaisen henkilön toimesta, jonka vastuualueeseen kuuluu tuotteiden pitkän aikavälin tulevaisuus (Parnas 1994).

Viime vuosina ketterät menetelmät (Agile) ovat vallanneet alaa ohjelmistojen kehitysmenetelminä, ja niissä katselmukset on sovellettu käytäntöön.

2.4.4 Miksi ohjelmistojen vanhenemista ei voi estää?

Kykymme suunnitella huomioiden tulevat muutokset riippuu kyvystämme ennustaa tulevaisuutta, mutta kykenemme ennustamaan tulevaisuutta vain likimääräisesti ja epätäydellisesti. Vuosien mittaan tullaan joka tapauksessa tekemään alkuperäisiä olettamuksia rikkovia muutoksia, dokumentaatio ei koskaan tule olemaan täydellinen, eivätkä edes katselmoijat millään huomaa kaikkia virheitä.

Kaikki ehkäisevät toimenpiteet ovat ehdottoman kannattavia, mutta ohjelmien vanhenemisen ongelmaa ne eivät poista.

2.5 Ohjelmistojen geriatriaa

Ennaltaehkäisy on tietenkin paras lääke, kun halutaan hidastaa ohjelmien vanhenemista, mutta toteutettiin ennaltaehkäisy miten hyvin tahansa, joudumme joka tapauksessa tekemisiin vanhenneiden ohjelmien kanssa. Tässä kappaleessa käydään läpi useampia keinoja, joilla jo vanhentunutta ohjelmaa voidaan hoitaa. (Parnas 1994.)

2.5.1 Heikkenemisen pysäyttäminen

Kun havaitaan, että ohjelmaan on pitkäaikaisen huolimattoman ylläpidon vuoksi päässyt kertymään vanhenemisen oireita, pitäisi aivan ensimmäiseksi panostaa vanhenemisen

hidastamiseen. Myös ohjelman huollossa pitäisi noudattaa aiemmin mainittuja vanhenemisen hidastamiseen tähtääviä menetelmiä. Tämä ei kuitenkaan ole helppoa, sillä vanhojen virheiden siivoaminen on aina hankalampaa kuin asiaan tarttuminen jo alusta alkaen.

2.5.2 Dokumentointi jälkikäteen

Dokumentoinnin tason parantaminen on usein merkittävä tekijä ohjelman vanhenemisen hidastamisessa, ja usein jopa nuorentaa ohjelmaa. Vanhenneen ohjelman dokumentaatio on syytä kirjoittaa uudestaan, sillä ajan kuluessa muutosten dokumentointi on joko laiminlyöty kokonaan tai se on erillisissä muistioissa, eikä sitä ole integroitu ohjelmiston aiempaan dokumentaatioon. Jos ohjelma on arvokas, olisi tärkeä suunnitella, toteuttaa ja tarkastaa sen dokumentaatio kokonaan uudestaan. Dokumentointi ei usein ole ohjelmoijien mielestä mielekästä työtä, mutta sen hyödyt ovat ilmeiset. Ylläpidon helpottumisen lisäksi ohjelman laatu usein kehittyy dokumentoinnin myötä. Systemaattisen koodin läpikäynnin yhteydessä siinä havaitaan piirteitä (virheitä, tuplakoodia, lähes samanlaisia funktioita ja muita kehittämiskohteita), jotka kannattaa korjata.

2.5.3 Modularisointi jälkikäteen

Ohjelmaa huollettaessa kannattaa sen modularisointia kehittää. Saman tehtäväalueen funktioiden kerääminen yhteen on sekä järkevää, että tulevan huollon kannalta edullista. Ohjelman rakennetta muutetaan niin, että jokainen moduuli pitää sisällään sellaisia suunnittelupäätöksiä, jotka todennäköisesti tulevat muuttumaan. Kokenut konsultti tai osaava ohjelmoija näkee usein helposti, miten vanha koodi on kirjoitettu, ja osaa hahmottaa miten koodi kannattaa kerätä uusiksi moduuleiksi.

2.5.4 Amputaatio

Osaa ohjelman koodista on ajan kuluessa muutettu niin moneen kertaan ja niin suunnittelemattomasti, että kyseinen osa ei ole säilyttämisen arvoinen. Tällainen osa voidaan korvata täysin uudella osiolla, joka toteuttaa tarvittavan asian tehokkaammin, ehkä täysin omalla tavallaan. Koodin poistaminen ei ole helppoa, sillä usein ohjelmoijat eivät halua hyväksyä jonkin osan työstä muuttuneen arvottomaksi. Tässäkin auttaa, jos arvion tekee ulkopuolinen taho, jolla ei ole tunnesiteitä koodiin. Kun uusi osa koodataan, tehdään se tällä kertaa noudattaen vanhenemisen ehkäisemiseen tähtääviä periaatteita.

2.5.5 Laajan ohjelmakoodin leikkaus ja uudelleenjärjestely

Suurien ja tärkeiden ohjelmistojen tai ohjelmistoperheiden päästessä rappeutumaan on oleellista tehdä perustavaa laatua oleva koodin uudelleenjärjestely. Eri ohjelmaversiot

käydään läpi ja niiden erot ja erojen syyt kirjataan. Jos erovaisuuksista löytyy kokonaisuus, joka voidaan paketoida ja sopia siihen yhteneväinen rajapinta niin että kaikki pystyvät sitä käyttämään, on mahdollista vähentää koodimäärää radikaalisti. Kokonaisuus voidaan mahdollisesti tiivistää yhteen ohjelmaan, joka käyttää tätä moduulia eri käyttötarkoitusten mukaan. Tarkoitus on siis identifioida ja poistaa ylimääräisiä komponentteja ja turhia riippuvuussuhteita sekä yhdistää toiminnallisuuksia ja vähentää siten kompleksisuutta. Kun tämä kaikki toteutetaan noudattaen vanhenemisen ehkäisemiseen tähtääviä periaatteita sekä dokumentoidaan hyvin, helpotetaan samalla radikaalisti myös tulevaa ylläpitoa ja vanhenemisen ehkäisyä.

2.6 Yhteenveto oikeasta toimintamallista

Jos ohjelmiston vanheneminen halutaan estää tai ainakin hidastaa sitä, pitää asia ottaa vakavasti, eli vanheneminen pitää tunnustaa ongelmaksi ja panostaa sen ehkäisyn suunnitteluun jo ohjelmistoprojektin alusta lähtien.

2.6.1 Asenteita on muutettava

Heti ensimmäiseksi on ymmärrettävä, että ohjelman saaminen pyörimään ensimmäisen kerran ei voi olla ainoa merkittävä asia. Nykyhetken aikataulupaineiden ei saa antaa vaikuttaa lopputulokseen, sillä muuten meillä käsissämme rikkinäinen ohjelma ja samalla huonosti voiva yhtiö. Ohjelmistoteollisuuden pitää luoda standardeja vanhenemisen hallintaan tähtääville rakenteille, menetelmille ja dokumentaatiolle sekä varmistaa, että ”laatusinettiä” ei saa, jos standardeja ei noudateta.

2.6.2 Dokumentoi ja katselmoi

Ohjelmasuunnitelmat tulee tarkkaan dokumentoida ja katselmoida ennen ohjelmoinnin alkamista. Valmiit ohjelmat tulee dokumentoida ja katselmoida. Myös kaikki ohjelmiin tehtävät muutokset tulee dokumentoida ja katselmoida. Tarkka mahdollisten tulevien muutosten analyysi on syytä olla osa kaikkien ohjelmien kehitysprosessia. Vähänkään isommissa organisaatioissa pitäisi olla vastuuhenkilö, tai jopa osasto erityisesti tarkastamaan ohjelmasuunnitelmat muutosten hallinnan osalta. Heillä pitäisi olla mahdollisuus estää tulevan ohjelmistohallinnan kustannuksella pelkkiin pikavoittoihin tähtäävät muutokset.

Jos ohjelmaa ei ole dokumentoitu, se ei ole valmis. Jälkikäteen tehtävä ohjelman dokumentointi on aina kalliimpaa, lopputulos on huonompi, eikä siitä ole ollut muutosten hallinnan kannalta lainkaan hyötyä.

2.6.3 Vanheneminen on ennakoitava

Muilla teknologiasektoreilla tuotteen vanheneminen on tunnettu asia ja se huomioidaan jo tuotteen suunnittelu- ja markkinointisuunnitelmissa. Takuuaika on hyvin rajallinen, varaosiakin saa vain ennalta suunnitellun ajan. Sama pitäisi toteuttaa myös ohjelmistotekniikassa. Jos ohjelmiston elinkaari ja samalla sen vanheneminen suunniteltaisiin etukäteen ja siihen valmistauduttaisiin asiaankuuluvalla tavalla, olisi uuden version suunnitteluun ja toteutukseenkin tarvittavat varat olemassa silloin kun se on ajankohtaista, eikä kestämättömiä yllätyksiä tulisi eteen.

2.7 Ketterät menetelmät teknisen velan hoidossa

Jos ohjelmiston kehitysprosessissa otetaan teknistä velkaa, on sen hoidosta huolehdittava. Nykyisin suosituissa ketterissä (Agile) kehitysmenetelmissä pyritään juuri tähän tavoitteeseen. Ajatuksena on, että ohjelmistoa kehitetään pienin laadukkain askelin, jolloin jokainen osio testataan hyvin ennen kuin se luokitellaan valmiiksi. Iteraatiokierrokset ovatkin hyvin lyhyitä, vain noin 1-4 viikkoa. Esimerkiksi Agile Scrum -menetelmä perustuu alusta alkaen backlogiin, sen käsittelyyn, askelten hyötyjen arviointiin ja priorisointiin. Backlogiin kirjataan kaikki toiminnallisuudet, jotka ohjelmistoon halutaan kehittää, arvioidaan niiden tarpeellisuus ja hyödyt sekä priorisoidaan ne.

Backlog on myös työkalu teknisen velan hallintaan (Technical debt). Sen avulla huolehditaan, että esim. aikataulukiireiden vuoksi ohitetut/oikaistut kohdat huomioidaan tulevaisuudessa, eikä niitä unohdeta. Myös mikäli prosessissa havaitaan tarpeellisia muutoksia, jotka eivät sovi sen hetkiseen kuvaukseen, kirjataan ne backlogiin sekä arvioidaan niiden hyödyt ja merkitys.

2.7.1 Teknisen velan luokittelu

Nita Andansare on web-artikkelissaan ”Managing Technical Debt with Agile” kuvannut teknisen velan hoitoa ketterin menetelmin (Andansare 2014).

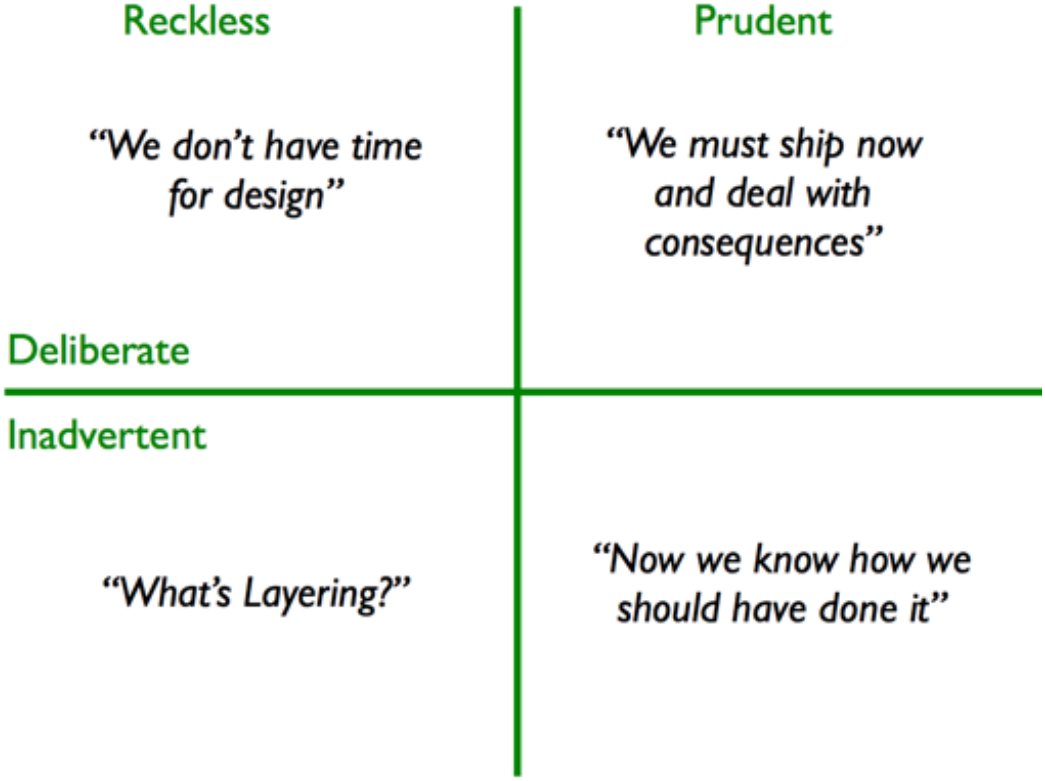
Teknisen velan tyypit Andansaren mukaan:

1. Tahaton velka
2. Tahallinen velka
   a. Lyhytaikainen, joka on sallittua vain, jos ollaan sitouduttu aikatauluttamaan, priorisoimaan ja hoitamaan velka pikaisesti.

b. Pitkäaikainen, joka on sallittua vain, jos on sitouduttu tukemaan projektia velan ratkaisemisessa.

Lyhytaikaisen velan ottaminen on reaktiivinen ja taktinen toimenpide, kun taas pitkäaikainen velka on proaktiivinen ja strateginen toimenpide. Seuraus tästä on se, että lyhytaikainen pitäisi maksaa pois nopeasti, mielellään jo seuraavan ohjelmakehityskierroksen aikana, kun taas pitkäaikaista velkaan voidaan suvaita pidemmän aikaa, muutaman vuoden tai pidempäänkin.

Kuvassa 2.2. luokitellaan Martin Fowlerin nelikentän avulla syitä projekteissa syntyvään tekniseen velkaan (Technical Debt Quadrant) (Fowler 2009).



Kuva 2.2. Martin Fowler, Technical Debt Grid Quadrant (Fowler 2009).

Kuvan 2.2. nelikenttää voidaan yrityksissä käyttää sen päättämiseen, että

- miten erilaisiin teknisen velan syihin suhtaudutaan
- millä taktiikalla velasta aiheutuviin tilanteisiin ja ongelmiin varaudutaan
- miten havaitut asiat hoidetaan eteenpäin.

Millaiset projektit päätyvät nelikentän eri lohkoihin? Seuraavassa on käyty läpi kukin neljännes ja kuvattu niihin luokiteltavien projektien ominaisuuksia (Andansare 2014).

1. Harkittu ja Tahallinen (Prudent and Deliberate)

Tähän päädytään silloin, kun päätös ottaa teknistä velkaa on harkittu ja tahallinen. Tähän neljännekseen päätyvät tilanteet, joissa tuote välittömästi julkaisemalla saavutetaan houkutteleva hyöty. Esimerkiksi vastataan bisnekseen kohdistuvaan uhkaan tai halutaan hyödyntää tilaisuus, jolla olisi positiivinen vaikutus taloudelliseen lopputulokseen. Tilanteita, joissa tähän voitaisiin oikeutetusti päätyä voivat olla

- Uuteen markkinaan tuleminen – voi olla hyväksyttävää tuoda tuote markkinoille ensimmäiseksi huolimatta sen matalammasta laadusta ja ominaisuuksien useista väliaikaista korjauksista.
- Säätelyn tai lain vaatimukset – projekteissa, joissa määräysten mukaisuus on erityisen tärkeää, saattaa myöhästymisellä olla vakavia taloudellisia tai operatiivisia vaikutuksia bisnekselle.
- Sesonkiajan mahdollisuus – kaupan alalla joulusta on tullut tärkein osa vuotista sykliä, joten sillä, että jos ei onnistu julkaisemaan joulumarkkinoille, voi olla merkittävä negatiivinen vaikutus yrityksen kokonaissuoritukselle.

2. Holtiton ja Tahallinen (Reckless and Deliberate)

Tähän päädytään silloin, kuin päätös ottaa teknistä velkaa on holtiton mutta tahallinen. Tähän neljännekseen päätyminen heijastelee huonoa johtamista; johtamista, jossa määräaikoihin pääsemiseksi oiotaan/oikaistaan, mutta tavoitteena on ennemminkin havaittujen toiminnallisten tarpeiden ratkaiseminen, kuin taustalla oleva selkeä bisnesmahdollisuus. Kyseessä on projekteille hyvin tavallinen syy ottaa teknistä velkaa.

Esimerkkejä tällaisesta tilanteesta:

- Projektin päättämistä kiirehditään, koska tulossa olisi paljon uusia projekteja.
- Projektin päättämistä vauhditetaan, koska kukaan ei ole luonut asiakkaaseen kunnollista kommunikaatiosuhdetta. Heille ei siis ole pystytty kertomaan mikä vaikutus projektin lopputuloksen laadulle on sillä, että sen päättämistä kiirehditään.
- Projektissa tehdään oikaisuja/oikaistaan, koska ryhmää kannustetaan julkaisemaan projekti määrättyyn ajankohtaan mennessä.

3. Holtiton ja Tahaton (Reckless and Inadvertent)

Tällaista tapahtuu taitamattomille ohjelmoijille, jotka ovat eivät ymmärrä koodin pätkän poistamisen tai lisäämisen vaikutuksia, ja aiheuttavat siitä syystä valtavasti teknistä velkaa! Tässä neljänneksessä olevan teknisen velan hoitaminen edellyttää huolellista prosessien ja työkalujen käyttöä, hyödyntäen pariohjelmointia, koodikatselmuksia, jatkuvaa integrointia, automatisoitua testausta, jne.

4. Harkittu ja Tahaton (Prudent and Inadvertent)

Tilanne on hyvin luonnollinen. Tiimit haluavat kehittyä ja parantaa tekemäänsä kokemuksen ja relevantin osaamisen hankkimisen jälkeen.

Yritykset suhtautuvat teknisen velan ottamiseen hyvin erilaisin tavoin, aivan kuten myös taloudellisen velankin ottamiseen. Osa yrityksistä ei halua ottaa velkaa lainkaan, osa taas kokee velan hyvänä työkaluna ja haluavat ymmärtää miten sitä voi hyödyntää järkevästi. Osa firmoista seuraa teknistä velkaa tiimin nopeuden muutoksen kautta. Kun tiiminopeus alkaa pudota teknisen velan kerääntymisestä johtuen, keskittyy tiimi erityisesti teknisen velan korvaamiseen, kunnes tiiminopeus on taas palautunut.

2.7.2 Erilaisia tekniikoita Agilessa teknisen velan syntymisen estämiseksi

Andansare on listannut artikkelissaan erilaisia keinoja ja menetelmiä teknisen velan syntymisen estämiseksi. Tällaisia keinoja tai menetelmiä ovat ”Vedetään tähän raja”, Vahva ”Valmiin määritelmä”, TDD, Test-drive Development, eli ”Testiajo”-kehitys ja Yhteistyöhön perustuva kehitys joista seuraavassa enemmän. (Andansare 2014.)

1. ”Vedetään tähän raja”

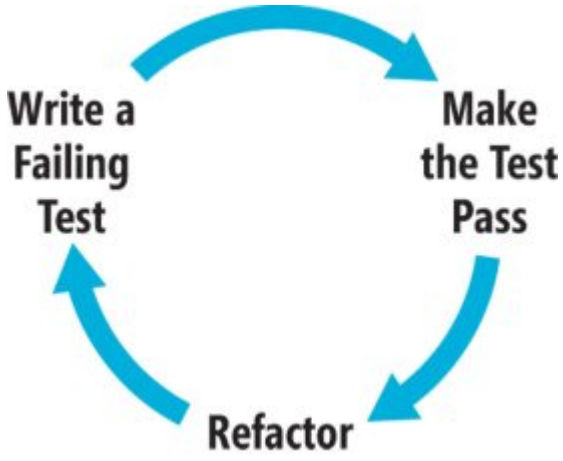
   Tehdään kulttuurimuutos
   - ”Tästä päivästä eteenpäin teemme toisin.”
   - ”Emme enää hyväksy teknistä velkaa.”
   - ”Ainoat sallitut perusteet velalle ovat taktiset tai strategiset bisnesperusteet”

2. Vahva ”Valmiin määritelmä” (Definition of Done)

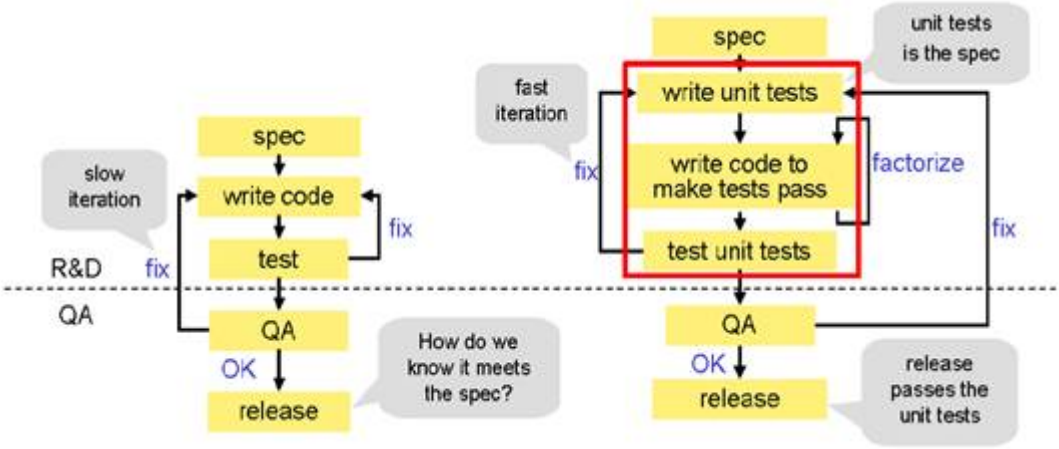
   Agile kehitykseen liittyviä menetelmiä ja työkaluja, joihin sitoutumista on syytä harkita
   - Koodikatselmukset (Code reviews)
   - Staattiset analyysit ohjelmatestauksessa (Static analysis)
   - Dynaamiset puhdistukset (Dynamic purification)
   - Kompleksisuus analyysit (Complexity analysis)
   - Kirjalliset osiotestit (Written unit tests)
   - Nolla tiedossa olevaa virhettä (0 known bugs)
   - Testien läpimeno (Test cases pass)
   - Automatisoidut testit (Test cases automated)

3. TDD, Test-drive Development, eli ”Testiajo”-kehitys

TDD menetelmä: Kirjoita testi, epäonnistu testissä, kirjoita lisää ohjelmakoodia, jotta läpäiset testin. Muuta ohjelmarakenteita tarpeen mukaan. Kehitä korkeampilaatuisempaa virtaviivaistettua ohjelmakoodia. Toista tätä prosessia tarpeen mukaan kaikkiin kehitettäviin ominaisuuksiin. Kuvassa 2.3. TDD menetelmä yksinkertaistettuna Agile Testing Frameworkin mukaan. Kuvassa 2.4. Andansare on kuvannut TDD kehitystä havainnollisen kaavion avulla.



Kuva 2.3 Test Drive Development menetelmän periaate (ATF 2018)



Kuva 2.4 Test Drive Development kaaviomuodossa (Andansare 2014).

4. Yhteistyöhön perustuva kehitys (Collaborative Designs)
   - Valkotaulusessiot ohjelmiston suunnittelusta
   - Ryhmän kollektiivisen kokemuksen hyödyntäminen
   - Joukkoviisauden hyödyntäminen (Wisdom of the crowd)

- Käyttäjäsuunnittelustudio (User design studio)
- Vähennetään riippuvuutta aihepiirin erityisasiantuntijoihin (Less reliance on subject-matter experts)

2.7.3 Erilaisia tekniikoita Ketterässä kehityksessä syntyneen teknisen velan hoitamiseen

Ansansare on artikkelissaan (Andansare 2014) kuvannut tekniikoita Ketterässä kehityksessä syntyneen teknisen velan hoitamiseen

1. Debt Backlog
   - Kirjataan syntynyt velka erilliseen Debt Backlog taulukkoon (Kuva 2.5).
   - Käytetään, kun velka on taloudellisessa mielessä oikeutettua
   - Velka nimetään ja sitä seurataan sen mukaan
   - Jokaiseen velkaan liittyvä tehtävät listataan erikseen ja niihin liitetään arvio työmäärästä
   - Seurataan, miten hyvin aikataulunkorjaukseen sitoudutaan.
   - Työkaluja: tuotteen backlog kirjaukset, erillinen Excel taulu seurantaan, erillinen nimike vikatietokannassa, muistilaput jne.
   - Seurataan velkaa jatkuvasti toiminnan ohessa

Debt Backlog - My Natural Healer iPhone App

Debt	Business Justification	Sprint Incurred	Estimate	Work Items	Commitment	Note
Variable names preceded with "email" even though email is only part of their use	Confusion factor is not currently there, but will be in the future.	sprint 2	1 week	Change variable names to more meaningful names.	sprint 3	DONE
Duplicate code for displays in Remedy Controller and Favorites view	Meet early May release schedule, willing to accept cost of double maintenance for now.	sprints 3 - 4	3 weeks	New file to contain common methods for remedy display, change existing code to conform	Release 1.1 - expected out in July 2010	Servicing this debt may push the SEARCH function out to a later release.
The same method handles both designating and undesignating a favorite	Too risky to do right before the customer milestone demo	sprint 6	2 weeks	Bust out undesignate into separate method, add logic to parts of code to know which method to invoke	Sprint 7, possibly sprint 8	

Kuva 2.5 Esimerkki Debt Backlogista (Three Beacons 2011)

2. Velan hoitamiseen sitoutuminen
   - Otetaan teknisenvelan tehtävät mukaan tuleviin sprintteihin
   - Hoidetaan velka niin nopeasti kuin on järkevää
   - Linjassa sen kanssa, mihin sitouduttiin aikanaan, kun velan ottaminen todettiin oikeutetuksi.
   - Lyhennä velkaasi vauhdissa! Pidä se niin lähellä nollaa kuin mahdollista.

3. Keksityt tarinat (Makeup Stories) – hyvä työkalu teknisen velan seuraamiseen
   - Käyttäjätarinat (User Story), ovat Agile menetelmän osa, jolla toimintoja määriteltäessä pyritään siirtämään fokus vaatimusten kirjoittamisesta niistä puhumiseen. (https://www.mountaingoatsoftware.com/agile/user-stories)
   - ”Keksitty tarina” on menettely, jolla muut kuin käyttäjätarinat saadaan kirjattu backlogiin. Keksitty tarina liittyy ohjelmakoodissa olevaa ongelmaa, pahoittelee ja lupaa korjata sen!
   - Käytetään kun esimerkiksi aina kun teknistä velkaa on syntynyt, tai kun mistä tahansa syystä on tingitty laadusta. Listassa käytetään merkinnälle poikkeavaa väriä.
   - Merkintä lisätään tuotteen Backlogiin. Esimerkki merkinnästä kuvassa 2.6
   - Priorisoidaan tarina seuraavaan sprinttiin (mikäli mahdollista)

Who	Description	Estimate (Hours)	Remaining (Hours)	Notes
	User login			
Chris	Login window	8	4	
	Hide password	2	2	
John	Verify login id is correct	16	0	i/f with Security system
John	Verify password is correct	16	12	i/f with Security system
	Alert and log if incorrect	8	8	
Chris	Block after 3 incorrect tries	24	0	Log occurrence to "watch file"
	Launch main window if correct	4	4	
	Refactor and improve Connection Timeout			Technical debt item
Mona	Framework services API change	8	2	
Mark	Adjust code to use system timers	40	20	
	Display main window			
Bala, Tim	Query DB for video connections	32	0	Bala to do DB arch
Tim	Establish connections	16	16	
	Display sub-tiles according to user config	64	64	
	Research competition			
	Identify main competitors	40	40	
Paul	Gap analysis	120	80	
	Document and share market differentiators	80	80	
Amy	Brief marketing and sales teams	16	0	
	Total hours	494	332	
	Total relative points from product backlog	43	43	

Kuva 2.6. Esimerkki keksitystä tarinasta backlogissa (Three Beacons 2011)

4. Keksittyjen tarinoinen mukaanotto (Makeup Stories Adopt) – hyvä tapa varmistaa, että keksityt tarinat käsitellään.
   - Mukaan ottamisen toimintamalli
   - Jos käyttäjätarinalla on velkavaikutus, otetaan velkarivi osaksi käyttäjätarinan työstämistä.
   - Lisätään keksityn tarinan estimaatti käyttäjätarinaan

5. Velalle hoidolle dedikoitu sprintti.
   - Samantyyppinen kuin ”virheiden korjaus sprintti”
   - Toteuta vain, jos todella on tarpeen – kun projektin aikana syntyvä velka kasvaa huomattavan suureksi, tiimi on huolestunut velan kasvusta, tulevat ominaisuudet perustuvat koodiin, jolla on rasitteena velkaa.
   - Tai yhdistetään virheenkorjaus ja velan hoito sprintti
   - Määräajoin hoidetaan projektin aikana kertynyt velka pois

2.8 Vielä lopuksi

Kuten tässä luvussa olemme voineet todeta, ohjelmistot vanhenevat, eikä sitä voi täysin välttää. Vanhenemiselle on monia syitä, mutta onneksi vanhenemiseen voidaan myöskin varautua ja sitä voidaan hidastaa noudattamalla alusta alkaen hyvän suunnittelun ja dokumentoinnin periaatteita. Käytettävät ohjelmointimenetelmät valitaan niin, että kokonaisuus pysyy mahdollisimman hyvin tiimin hallussa jo niin toteuttamisvaiheessa, kuin myös tuotteen elinkaaren varrella erilaisista ympäristössä tapahtuvista muutoksista huolimatta. Tämä edellyttää niin sitoutumista, kuin panostustakin, joten vanhenemiseen varautumisesta aiheutuu kustannuksia, jotka on huomioitava jo budjetointivaiheessa.

On siis tärkeää, että vanhenemiseen varaudutaan ja ymmärretään, että ohjelmien tekeminen kunnolla vaatii aikaa ja rahaa!

3 Case: Aineistonhallinta ja aineistonhallintajärjestelmät vuosituhannen taitteessa

Tässä luvussa esitellään käsitteet aineistonhallinta ja aineistonhallintajärjestelmät, niin kuin ne ymmärrettiin vuonna 2002. Se mitä nykyään (vuonna 2018) luetaan kuuluvaksi aineistonhallinnan alueeseen, on varmasti tästä laajentunut, ehkä joidenkin käsitteiden osalta myös hieman muuttunut, mutta luvussa kuvataan nimenomaan käsitteiden tilannetta vuonna 2002. Nämä määrittelyt toimivat pohjana myöhemmin luvussa viisi esitettävälle HANSABASE -järjestelmän vanhenemisen analyysille.

Siirtyminen täysin digitaalisiin prosesseihin kaikkialla yritysmaailmassa on tuonut mukanaan suuret määrät digitaalista aineistoa. Suuret määrät digitaalista aineistoa luovat laajat mahdollisuudet sen monipuoliselle uudelleen käytölle.

Yritykset ovat keränneet digitaalisista aineistoistaan valtavia tietokantoja, mutta eivät pysty hyödyntämään niitä sellaisenaan niin hyvin kuin voisi olla mahdollista. Liiketoiminnallisista syistä yritysten on saatava aineistonsa paremmin hallintaan. Yhä kovemmat tehokkuus- ja tuottavuusvaatimukset, nopeasti kehittyvät kansainvälisen kaupan mahdollisuudet, cross-mediatuotanto -trendi julkaisuteollisuudessa, ja nopea teknologinen kehitys ovat kaikki luoneet painetta tähän suuntaan. Vaikka muutos onkin suuri ja vaikea, ovat myös sen tuloksiin kohdistuvat odotukset valtavat.

Tärkeänä lähteenä luvulle on toiminut VTT:n mediatekniikan asiantuntijapalvelun GT – raportti ”Digitaalinen aineistonhallinta”, nro 2, toukokuu 2001 (Nurmi ym. 2001).

Aivan aluksi tämän luvun kappaleessa 3.1. on kuvattu mitä digitaalisella aineistonhallinnalla tarkoitetaan. Sen jälkeen kappaleissa 3.2 - 3.5 on esitelty erilaisia tapoja luokitella markkinoilla olevia aineistonhallintajärjestelmiä. Perusteina kappaleessa 3.2 on käytetty jakoa tuotteisiin ja palveluihin, kappaleessa 3.3 jakoa järjestelmien koon mukaan, ja kappaleessa 3.4. jakoa toiminnallisuuden mukaan luettelointijärjestelmiin sekä arkistointi- ja varastointijärjestelmiin. Kappaleessa 3.5 jakoperusteena on käytetty järjestelmän teknistä toteutustapaa, jolloin järjestelmät on jaettu karkeasti itsenäisiin ja asiakas-palvelin-sovelluksiin. Kappaleessa 3.6. on käyty läpi aineistohallintajärjestelmien erilaisia käyttökohteita.

Aineistonhallinnan keskeisiä käsitteitä on selitetty kappaleessa 3.7 sekä tiedon tallennusmuotoja kappaleessa 3.8. ja kappaleessa 3.9. esitellään aineistonhallintaan liittyviä suosituksia.

Kappale 3.10 on varattu järjestelmien teknisen toteutuksen kuvaamiseen ja erilaisten ratkaisumallien vertailuun. Kappaleessa vertaillaan dokumenttien varastointimenetelmiä, tietokantaratkaisuja sekä ratkaisujen rajapintoja ja integrointitekniikoita. Kappaleessa 3.11 esitellään aineistohallintajärjestelmien komponentteja, tallentamista, luokittelua, materiaalin etsimistä ja käyttöönottoa sekä käyttöoikeuksien hallintaa, dokumenttien versionhallintaa ja dokumenttien jakamista.

3.1 Mitä on digitaalinen aineistohallinta

Digitaalinen aineistonhallinta on yleistermi digitaalisessa muodossa olevien tietojen hallintamenetelmille. Aineistonhallinta (Digital Asset Management, DAM; Media Asset Management, MAM) tarkoittaa kaiken uudelleenkäytettävän digitaalisen tiedon – kuvien, äänen, videon, tekstien sekä valmiiden sivujen ja dokumenttien – organisointia nopeaa etsintää, noutamista ja uudelleenkäyttöä varten.

Edellä olevissa lyhenteissä esiintyvä englanninkielen käsite ”asset” tarkoittaa myös varoja, omaisuutta, arvokasta lisää. Se pitää siis sisällään myös aineiston arvon ymmärtämisen. Digitaalisen aineiston määrän kasvaessa, kasvaa samalla yritysten omaisuus, jos asia ymmärretään niissä oikein.

Materiaalien digitaalinen hallinnointi poistaa monta aikaa vievää askelta. Jo pelkästään aineistojen perinteisen postitse tai kuriirilla lähettämisen tarpeen pois jääminen tehostaa yhteistyötä eri tahojen kanssa huomattavasti. Tehokkaat hakutoiminnot puolestaan tekevät materiaalin etsimisestä vaivatonta ja sujuvaa. Aineistojärjestelmistä hyötyvät niin markkinoinnin, mainonnan ja viestinnänkin ammattilaiset, kuin sellaiset verkottumaan pyrkivät yrityksetkin, joilla on toimipisteitä ja yhteistyökumppaneita laajalla alueella.

Aineistonhallinnan etuja voidaankin listata esim. seuraavasti:

- Yhteistyökumppanit ja alihankkijat voidaan liittää järjestelmään.
- Ryhmä- ja yhteistyö on mahdollista yritysten kesken.
- Aineistoilla on yksi säilytyspaikka, jossa check-in/check-out sekä version hallinta.
- Nopeutettu valmistusprosessi, koska aineistoa voidaan työstää jatkuvasti.
- Aineistojen etsiskelyyn ja lähettämiseen ei kulu aikaa.
- Keskitetyn hallinnan sekä sovittujen laatukriteerien ansiosta aineiston käyttäjä voi luottaa aineiston laatuun. Käyttäjä tietää mitä saa.

Aineistonhallintajärjestelmien käyttökohteet, ominaisuudet, kohdeorganisaatiot ja toteutustavat vaihtelevat huomattavasti. Aineistonhallintajärjestelmä voi olla valmis kaupallinen tuote,

jonkin tietokannan hallintaohjelmiston päälle palapelinä eri toimittajien osista koottu kokonaisuus, mittatilaustyönä tehty multimediatietokanta tai esimerkiksi organisaation ulkopuolelta hankittu palvelu. Kevyitä aineistonhallintajärjestelmiä toimitetaan myös muiden ohjelmistojen mukana ja useissa ohjelmistoissa on toimintoja aineistonhallintaan.

Järjestelmien ominaisuudet vaihtelevat huomattavasti, ja useimmat kaupalliset tuotteet on suunnattu määrätyille käyttöalueille. Osa järjestelmistä tarjoaa ominaisuuksia jatkuvan datavirran käsittelyyn, ja ne ovat suunnattuja animaatiota, videota ja ääntä tuottaville organisaatioille. Pääasiassa painotuotteita tuottaville organisaatioille tarkoitetut järjestelmät taas tarjoavat usein integrointimahdollisuuden suosittujen sivunvalmistusohjelmistojen kanssa. Uusmediayrityksille profiloidut järjestelmät taas integroituvat usein web-sivustonhallintajärjestelmiin.

Ennen kuin aineistonhallintajärjestelmä voidaan ottaa käyttöön, pitää määritellä organisaation tai prosessin työnkulku. Koska työnkulut vaihtelevat oleellisesti yrityksestä toiseen, on erittäin tärkeää, että aineistonhallintajärjestelmä on räätälöitävissä erilaisiin tarpeisiin. Järjestelmän on tuettava olemassa olevaa työnkulkua siltä osin kuin nykyinen työnkulku halutaan säilyttää.

Aineistonhallintajärjestelmät kehittävät aineistojen luokittelua ja järjestämistä sekä helpottavat aineistojen löytämistä. Esimerkiksi kuvien kohdalla tämä tarkoittaa sitä, että niitä voidaan etsiä järjestelmästä avainsanoilla sekä muiden piirteiden, kuten koon ja resoluution perusteella. Järjestelmiin, niiden käyttöön ja täysipainoiseen hyödyntämiseen liittyy huomattava joukko muitakin ominaisuuksia, joista enemmän seuraavassa. (Nurmi ym. 2001.)

3.2 Jako tuotteisiin ja palveluihin

Karkein jaottelu järjestelmille on niiden jako tuotteisiin ja palveluihin. Aineistonhallintajärjestelmä voidaan hankkia organisaation sisälle tuotteena, johon kuuluu ohjelmisto ja mahdollisesti myös laitteisto. Tuote voi olla ohjelmistolisenssi, jota käytetään olemassa olevalla laitteistolla, tai esimerkiksi konsulttiyrityksen kokoama kokonaisjärjestelmä.

Ohjelmistolisenssiin perustuva järjestelmä on täysin omassa hallinnassa ja integroitavissa muihin järjestelmiin. Ylläpitoon täytyy kuitenkin varata resursseja sekä varautua hankkimaan lisää laitteistoa tai ohjelmistoa tarpeen mukaan. Isot painokelpoiset kuvat ovat kooltaan kymmeniä megatavuja, valmiit sivut ja liikkuvaa kuvaa ja ääntä sisältävät multimediatiedostot

jopa gigatavuja. Järjestelmän tallennuskapasiteettivaatimukset ovat siten huomattavat, ja tällaiseen käyttöön soveltuvat varmennetut tallennusjärjestelmät ovat erittäin kalliita.

Mikäli organisaatiossa ei ole halua, mahdollisuuksia tai resursseja hankkia ja ylläpitää aineiston hallintaa itse, voidaan se ostaa myös palveluna. Sovelluspalvelun tarjoaja (Application Service Provider, ASP) hoitaa tällöin aineistonhallintajärjestelmän ohjelmiston ja laitteiston hankinnan ja ylläpidon.

Palvelujärjestelmä voi perustua johonkin kaupalliseen aineistonhallintatuotteeseen tai se voi olla jonkin tietokannanhallintajärjestelmän päälle tuotettu järjestelmä. Yleisen aineistohallintapalvelun lisäksi palveluntarjoaja voi erikoistua tarjoamaan määrätyn alan lisäarvopalveluja, kuten töiden välittämistä painoyrityksille tai tiedonsiirtopalveluja.

Ongelmaksi palvelujärjestelmissä voi tulla se, että järjestelmä on tietoliikenneyhteyden päässä, eikä suoraan organisaation hallinnassa. Tietoliikennekatkot tai -ongelmat voivat hankaloittaa järjestelmän käyttöä tai tehdä sen jopa mahdottomaksi. Vuosituhannen alussa oli vielä sellainen tilanne, että suuria aineistomääriä käsitellessä tietoliikennekustannukset saattoivat olla huomattavat, suhteellisesti paljon suuremmat kuin nykyään. Materiaalin käyttö web-sivuston sisältönä saattoi olla ongelmallista, ellei web-palvelin sijainnut samassa paikassa aineistonhallinnan kanssa. Palvelujärjestelmät ovat kuitenkin organisaatiolle vaivattomampia, sillä järjestelmän ylläpitoon tai laitteistoon ei tarvitse varata kalliita resursseja. (Nurmi ym. 2001.)

3.3 Jako käyttöryhmän mukaan

Aineistonhallintajärjestelmät on käyttöalueensa lisäksi yleensä suunnattu tietyntyyppiselle asiakasorganisaatiolle. Käyttöryhmän mukaan järjestelmät voidaan jakaa karkeasti yksittäiskäyttäjä-, työryhmä- ja yritysjärjestelmiin.

Yksinkertaisimmat järjestelmät on tarkoitettu yksittäiselle käyttäjälle tai enintään muutaman hengen ryhmälle. Ne ovat yleensä itsenäisiä ohjelmistoja, eli niitä käytetään yhdessä työasemassa ja tiedot tallennetaan samalle koneelle.

Seuraavaan kategoriaan kuuluvat työryhmille, osastoille tai pienille yrityksille tarkoitetut järjestelmät. Näissä on yleensä enemmän toiminnallisuutta kuin yksittäiskäyttäjän järjestelmissä. Työryhmäjärjestelmät käyttävät useimmiten asiakas–palvelin –arkkitehtuuria. Tiedot tallennetaan palvelinkoneelle, josta niitä käytetään erityyppisillä asiakasohjelmilla.

Yritysversioita käytetään suurissa organisaatioissa mahdollisesti koko organisaation laajuisesti. Niissä ei useinkaan käytetä valmiita kaupallisia tuotteita, vaan järjestelmät on yleensä räätälöity ja integroitu muihin käytössä oleviin järjestelmiin. Suuret asiakkaat vaikuttavat myös kaupallisten tuotteiden tarjontaan, sillä he voivat sanella toimittajilleen vaatimukset yhteensopivuudesta omien järjestelmiensä kanssa. Nämä ominaisuudet siirtyvät sitten toimittajien kaupallisiin tuotteisiin. (Nurmi ym. 2001.)

3.4 Jako toiminnallisuuden mukaan

3.4.1 Aineistojen luettelointi

Yksinkertaisimmat järjestelmät ovat lähinnä vain aineistoluetteloijia. Ne on tarkoitettu luokittelemaan ja järjestämään tallennettu aineisto helppoa hakemista ja uudelleenkäyttöä varten. Niitä käytetään luomaan luetteloita tiedostojärjestelmään ja mahdollisiin ulkoisiin tietovarastoihin tallennetuista aineistoista. Erilaisille siirrettäville medioille tallennettu aineisto voidaan myös yleensä liittää luetteloon. Siirrettäville medioille tallennetusta aineistosta tallennetaan järjestelmään yleensä pieni esikatseluversio selattavaksi. Lähes kaikki luetteloijat mahdollistavat luettelon käyttämisen web-selaimella ja ovat jossain määrin ohjattavissa komentojonoilla (scripting). Luetteloijissa voi olla kehittyneempiä ominaisuuksia, kuten yksinkertainen työnkulun ohjaus ja ODBC –yhteys (Open Database Connectivity) tietokantoihin. ODBC on avoin tietokantaliitäntärajapinta, jonka avulla voidaan käyttää standardilla tavalla mitä tahansa tietolähdettä, johon on saatavissa ODBC –ohjain. (Nurmi ym. 2001.)

Tutkimuslaitos GISTICS[2] Inc. on vuodesta 1987 toiminut tutkimuslaitos, joka oli jo vuonna 1991 käsitellyt julkaisuissaan myös luettelointijärjestelmien ongelmia, joita ovat muun muassa suorituskyvyn heikentyminen noin 50 000 aineistoelementin jälkeen, käyttöjärjestelmän tiedostojärjestelmää käyttävän tallennuksen turvattomuus ja versionhallinnan puuttuminen. Myös työnkulun ohjauksen rajoittuneisuus ja useampien hajautettujen arkistojen ylläpidon ongelmat rajoittavat luettelointiohjelmistojen käyttömahdollisuuksia. (Nurmi ym. 2001.)

[2] Gistics: Information services for the Industrial Internet (https://gistics.com/).
”Native to Silicon Valley, GISTICS began in 1987 making the world a better place to live, play, work, and learn with technology. Our mission brings us to the work of humanizing the opportunities and challenges of the Internet of Things.GISTICS constitutes a new generation of information services for companies and individuals in pursuit of new opportunities of the Industrial Internet. We provide simple, effective, and stress-tested programs for innovation and growth. As performance partners to industry leaders and entrepreneurs, we bring a deep expertise and set of distinguishing capabilities to each client engagement.” (Gistics 2017)

3.4.2 Aineistojen arkistointi ja varastointi

Kehittyneempiä aineistonhallintajärjestelmiä voisi luonnehtia aineistoarkistoiksi tai aineistovarastoiksi. Niissä on tyypillisesti kaikki luetteloijien perusominaisuudet täydennettyinä mukautettavalla rakenteisella työnkulun ohjauksella sekä kehittyneellä versionhallintatuella. Jotkin tuotteet tarjoavat myös ohjelmointirajapinnan, mikä mahdollistaa niiden muokkaamisen tarpeiden mukaan. (Nurmi ym. 2001.)

Useimmat aineistoarkistot ovat kaksi- tai kolmikerroksisia asiakas–palvelin-järjestelmiä. Kaksikerroksisissa järjestelmissä on loogisesti erillinen asiakas- ja palvelinkerros, jotka voivat tosin sijaita samassa koneessa. Mahdollinen kolmas kerros on erillinen tallennuskerros eli käytännössä tietovarastona toimiva tietokantajärjestelmä. Aineistoarkistot käyttävät tietovarastoinaan usein kaupallisia relaatiotietokantatuotteita, kuten Oraclea, Informixia tai IBM DB2:ta, tai ainakin tarjoavat mahdollisuuden käyttää näihin tallennettua aineistoa.

Raskaimmat järjestelmät tarjoavat kokonaisia työnkulkuratkaisuja tai toisiinsa tiukasti integroitavia järjestelmiä, joilla voidaan kokonaisjärjestelmänä hoitaa sekä aineistonhallinta, että julkaisujen tekeminen eri medioille. Osa tarjottavista tuotteista on työkalupaketteja, jotka eivät ole valmiita järjestelmiä, mutta joiden avulla voidaan toteuttaa tiedettyyn tarkoitukseen räätälöity aineistonhallintasovellus.

3.5 Jako teknisen toteutustavan mukaan

Teknisesti aineistonhallintajärjestelmät voidaan jaotella karkeasti itsenäisiin ja asiakas-palvelin-sovelluksiin. Itsenäiset sovellukset, joita ajetaan yleensä yksittäisessä työasemassa, ovat useimmiten toiminnallisuudeltaan lähinnä tiedostoluetteloijia. Toiminnallisuudeltaan edistyneemmät järjestelmät ovat lähes poikkeuksetta asiakas–palvelin –mallin mukaisia. Näihin kuuluu palvelinosa, jota käytetään erilaisilla asiakasohjelmilla. Lisäksi järjestelmiin saattaa kuulua erillinen tallennuskerros, joka kuitenkin on asiakkaalle näkymätön. Tässä ns. kolmikerrosarkkitehtuurissa toiminnallisuus eli ns. middleware –kerros sijoittuu selain- ja tallennus –kerroksen väliin. Kerrosarkkitehtuurin etu on se, että muutokset yhdessä kerroksessa eivät aiheuta muutosten tarvetta muissa kerroksissa. Lisäksi middleware –kerroksen toiminnallisuutta voidaan hajauttaa eri tietokoneille kuormituksen mukaan. Esimerkiksi raskaat operaatiot voidaan tarpeen mukaan ohjata omalle koneelle, jolloin järjestelmän suorituskyky ei kärsi. Ulkoistetut aineistonhallintapalvelut noudattavat asiakas–palvelin- tai kolmikerrosmallia. (Nurmi ym. 2001.)

Asiakasohjelma voi olla alustakohtainen sovellus, Java-sovellus, web-selaimessa käytettävä Java –sovelma tai pelkkä web-selain. Alustariippumattomat ratkaisut asiakasohjelmissa ovat yleistymässä, koska alustariippuvaiset asiakasohjelmistot rajoittavat käyttäjäkuntaa ja niiden kehittäminen ja ylläpitäminen useille alustoille samanaikaisesti on kallista.

Useimmat aineistonhallintajärjestelmät ovat perinteisesti toimineet Apple Macintosh –ympäristössä tai niihin on ainakin ollut saatavissa Macintosh –asiakasohjelma, sillä graafiseen suunnitteluun ja painotöiden valmistukseen on tunnetusti käytetty Macintosh –laitteita. Itsenäisiä järjestelmiä on saatavissa Windows-, Unix- ja Macintosh-ympäristöön. Samoilla alustoilla toimivat useimmat palvelinkomponentitkin.

Osa asiakas–palvelin-järjestelmistä tarjoaa mahdollisuuden aineiston julkaisuun tai käyttämiseen web-sivustoissa. Tällaiset järjestelmät käyttävät hyväkseen kolmannen osapuolen web-palvelin-ohjelmistoa.

Materiaalin käyttöön on kaksi päätapaa. Aineistot tai aineistosta generoidut, kyseiseen tarkoitukseen sopivat versiot, voidaan saada järjestelmästä ulos yksittäin, ja varsinainen julkaisu generoidaan näistä erikseen, tai julkaisu määritetään ja generoidaan suoraan aineiston hallintajärjestelmästä. Jälkimmäinen tapa vaatii kokonaisjärjestelmää, jossa aineiston hallintajärjestelmä on integroitu muihin tuotantojärjestelmiin.

Jotkut aineistonhallintajärjestelmät mahdollistavat sopivan version generoinnin järjestelmään syötetystä aineistoelementistä sitä tarvittaessa. Näin toimivat järjestelmät ovat lähes poikkeuksetta aineistovarastoja. Osa järjestelmistä, lähinnä aineistoluetteloijat, antavat ulos saman aineiston, joka järjestelmään on syötetty. Sopivan version generointi edellyttää varsin mutkikasta ohjelmistoa ja sitä, että järjestelmä tuntee käytetyn tallennusmuodon. Generointi saattaa olla myös hidasta, joskin tähän vaikuttaa luonnollisesti käytetyn laitteiston nopeus. Syötetyn version käyttöönotto on varsin nopeaa, mutta mikäli käyttötarkoitus vaatii aineiston muokkaamista, vaatii se manuaalista lisätyötä. (Nurmi ym. 2001.)

3.6 Käyttökohteita

Digitaalisessa muodossa olevaa tietoa tuotetaan yhä enemmän, jolloin tarve aineistonhallintajärjestelmille kasvaa. Aihe on varsin uusi, eikä vakiintuneita toimintamalleja ole ehtinyt syntyä.

Prepress tai laajemmin premedia –yritykset käyttävät aineistonhallintajärjestelmiä asiakkaittensa aineistojen hallintaan. Aluksi järjestelmiä käytettiin lähinnä digitaalisessa

muodossa olevien kuvien arkistoimiseen, mutta yhä enenevässä määrin niihin halutaan tallentaa myös muuta materiaalia. Etenkin web –sivustot tuottavat huomattavia määriä digitaalista materiaalia, jota voidaan ja halutaan hyödyntää muutenkin. (Nurmi ym. 2001.)

Koska aineistojärjestelmiin voidaan tallentaa monessa vaiheessa olevaa materiaalia, se on avuksi jo materiaalin suunnitteluprosessin aikana. Kommentointikierros tai välivedosten tarkastaminen käy ripeästi, kun kaikki kommentoivat aineistoa samassa paikassa. Aineistojärjestelmä voi myös olla yrityksen ”brand book”, tai sen kautta voi jakaa markkinointi-, esite- ja lehdistömateriaalin. Uusinta on sähköisen kaupankäynnin ominaisuuksien integrointi aineistojenhallintajärjestelmiin. Tämä edellyttää yleensä myös tekijänoikeuksien hallintaa digitaalisessa ympäristössä (Digital Rights Management, DRM).

Eräs tärkeä käyttäjäryhmä ovat suuret, globaalisti toimivat yritykset. Tällöin kyse on tuotemerkin eli brändin hallinnasta. Tyypillistä on, että yrityksissä aineistohallinnan projekteista vastaavat markkinointi- tai tietotekniikkaosastot. Vasta myöhemmässä vaiheessa, projektin edetessä mukaan otetaan mainostoimistoja, reproja ja kirjapainoja. Reprojen ja kirjapainojen haasteena on päästä keskusteluihin markkinointi- ja tietotekniikkaosastojen henkilöiden kanssa oikeaan aikaan. Monella graafisen alan yrityksillä ei ole kovin selvää käsitystä markkinointi- ja tietotekniikkaosastojen toiminnasta. Tässä on varmasti eräs selitys sille, miksi digitaalisesta aineistojen hallinnasta ei ole tullut nykyistä suurempaa liiketoiminta-aluetta graafisen alan yrityksille. (Nurmi ym. 2001.)

3.6.1 Kuva-arkisto

Aineistonhallintajärjestelmää voidaan käyttää kuvien arkistointiin. Tällöin kuviin on liitettävä luokitusta helpottavaa metatietoa. Sama kuva voi tavallisesti kuulua useampaan luokkaan. Kuvia voidaan lisätä järjestelmään suoraan aineiston tuottamiseen käytettävistä sovelluksista erillisellä asiakasohjelmalla, verkkoyhteyden yli web-selaimella tai järjestelmän valvoman toimintokansion (hot folder) avulla.

Mikäli tallennettuun aineistoon voidaan luoda erilaisia näkymiä, helpottuu aineiston etsiminen ja selaaminen huomattavasti. Kun näkymästä suodatetaan pois ei-toivotut aineistoelementit, on halutun tyyppisen tai aiheisen aineiston löytäminen helppoa ja nopeaa.

Suomessa oli vuosituhannen alussa kymmeniä erilaisia kuva-arkistoja, joista tosin vain muutama toimi kaupallisesti. Mikäli kuva-arkistosta on tarkoitus myydä kuvia, oleellisia ovat kuvaajien tunnettuus ja yhteydet eri asiakkaisiin. (Nurmi ym. 2001.)

3.6.2 Dokumentti-arkistot

Koska lähes kaikissa vaiheissa yritysten prosesseja tuotetaan kasvavia määriä erimuotoisia dokumentteja, on niiden hallinta yritysten tehokkaalle toiminnalle keskeinen kysymys. On tärkeää löytää nopeasti määrättyjen dokumenttien viimeisimmät versiot, joskus myös vanhemmat versiot samoista dokumenteista. Tärkeää on myös huolehtia siitä, että dokumentit ovat vain hyväksytyn käyttäjäryhmän luettavissa, ja ehkäpä vielä tarkemmin määritellyn ryhmän editoitavissa. Määrättyihin projekteihin liittyvistä dokumenteista kiinnostuneiden on myös saatava tieto uusista tiedoista mahdollisimman tehokkaasti. Keskitetyn järjestelmän avulla vältetään lisäksi sähköpostijärjestelmien tukkeutuminen suurista liitetiedostoista. Lisäksi järjestelmään voidaan rakentaa erilaisia hyväksymiskäytäntöjä, palaute- ja kommentointimahdollisuuksia sekä monipuoliset hakuominaisuudet. Keskitetty järjestelmä myös edesauttaa mahdollisten toimintasääntöjen ja laatujärjestelmän ohjeiden noudattamista.

3.6.3 Brändin, eli tuotemerkin hallinta

Brändin hallinta tai graafinen ohjeistus tarjoaa erinomaisen käyttökohteen aineistonhallintajärjestelmille. Järjestelmän avulla markkinointiviestinnästä vastaava osasto voi huolehtia siitä, että organisaation graafinen ilme on yhdenmukainen ja kaikilla organisaation osilla on käytettävissä samat ajantasaiset logot, tuotekuvat ja muu tuotemateriaali. Tällaisessa käytössä aineisto arkistoidaan järjestelmään ja sitä voidaan käyttää mistä tahansa organisaation sisältä, mutta aineiston lisääminen ja muokkaaminen on tarkasti rajattu. (Nurmi ym. 2001.)

Tietokantaan voidaan antaa oikeuksia esimerkiksi organisaation käyttämälle mainostoimistolle, joka tuottaa uutta materiaalia sinne suoraan. Vastaavasti voidaan antaa painotalolle lupa käydä hakemassa aineisto suoraan brändi-tietokannasta. Lisäksi on mahdollista luoda digitaalisia työnkulkuja digitaalisine vedoksineen, tarkistuksineen ja hyväksymisineen tietokannan käyttäjien välille.

3.6.4 Monimediajulkaiseminen

Aineistonhallintajärjestelmiä voidaan käyttää avuksi myös monimediajulkaisemisessa. Tällöin kaikki julkaisujen eri versioiden aineistoelementit syötetään järjestelmään ja niistä generoidaan haluttu julkaisu. Järjestelmä pitää kirjaa aineistoelementtien välisistä suhteista ja siitä, mitkä elementit kuuluvat julkaisuun. Julkaisusta voidaan generoida erilaisia versioita eri medioille ja päätelaitteille. Julkaisu voidaan hakea helposti järjestelmästä uudelleen, ja se

voidaan päivittää aineistoelementtien uusimmilla versioilla. Tällöin esimerkiksi yrityksen tuoteluetteloon saadaan aina uusimmat tuotekuvat ja tekstit. (Nurmi ym. 2001.)

Kohdemedian ominaisuudet, mahdollisuudet ja rajoitukset määräävät minkä tasoisia materiaaleja medialle voidaan tuoda, esim. median mukaan eri resoluutioiset kuvat, tai vaikkapa kuvan sijasta vastaava videoleike. Julkaisujen tuottaminen voidaan automatisoida, vaikka niin, että www-sivujen kautta saatava tuoteluettelo on aina ajan tasalla (sivut ovat dynaamiset) tai niin, että uusi versio generoidaan järjestelmästä joka yö.

3.6.5 Web-to-print –sovellukset

Web-to-print –sovelluksissa käyttäjä voi luoda web-yhteyden yli painovalmiin PDF –tiedoston käyttäen apuna valmiita taittomalleja ja aineistopankin tietoja. Tiedosto voidaan tulostaa esim. digitaalipainokoneella. Sovelluksia voivat olla esim. käyntikortit, esitteet ja postikortit.

Aineistonhallintajärjestelmällä voidaan myös hallita painotöiden varastoa. Varastossa olevat painotyöt ovat katseltavissa järjestelmässä olevan sähköisen- esim. PDF –muodossa olevan näyttöversion kautta. Hallintajärjestelmässä oleviin painotöihin liitetään minimikappalemäärän hälytysraja, jonka alituttua asiasta ilmoitetaan vastuuhenkilölle, joka tekee päätöksen siitä, että tilataanko samaa painotyötä lisää, tuotetaanko uusi korvaava versio, tai poistuuko painotyö sen loputtua kokonaan varastosta. Järjestelmän kautta voidaan myös suoraan tehdä painotyötilauksia.

3.6.6 Painotöiden valmistus

Aineistonhallintajärjestelmiä ovat perinteisesti käyttäneet painotöitä valmistavat premedia- ja prepress-yritykset digitaalisina arkistokaappeina tai ainakin arkistokaapin sisällysluettelona. Painotöiden valmistuksessa aineistonhallintajärjestelmä voi toimia työnkulkuratkaisuna, minkä avulla hoidetaan digitaalisesti koko painotyön valmisteluprosessi. Kunkin työvaiheen suorittaja saa tällöin työn omaa vaihettaan varten samasta järjestelmästä. Järjestelmä voi myös olla integroitavissa muihin työnkulkujärjestelmiin. Integroitu työnkulkujärjestelmä tarjoaa selviä etuja – turhan työn määrä vähenee ja prosessissa tehdään vähemmän virheitä. (Nurmi ym. 2001.)

Tuolloin monet graafisen alan yritykset tarjoisivat aineistojen hallinnan palveluita maksutta asiakkailleen. Aineistonhallinnalla ymmärrettiin lähinnä erilaisten valmiiden töiden tallentamista muistivälineelle odottamaan mahdollista myöhempää käyttöä.

Toimintamallia oltiin kuitenkin muuttamassa siten, että aineistonhallinnasta oli tulossa liiketoimintaa, josta peritäisiin käytön mukainen maksu. Aineistonhallinta oli muuttumassa asiakkaiden tarpeista lähteväksi palveluksi.

3.7 Aineistonhallintaan liittyviä keskeisiä käsitteitä vuosituhannen taitteessa

Asiakirja (Record):

Vrt. dokumentti. Asiakirjan ja dokumentin ero liittyy niiden lainvoimaisuuteen. Kun dokumenttina voidaan pitää lähes mitä tahansa dataa sisältävää säiliötä, voidaan asiakirjaa sen sijaan pitää hyvinkin virallisena dokumenttina. Asiakirja on siis aina dokumentti, mutta läheskään kaikki dokumentit eivät ole asiakirjoja (Virtanen 2001.)

Asiakirjaan on tallennettu informaatiota missä tahansa fyysisessä muodossa tai mediassa, jotka organisaatio on luonut tai vastaanottanut ja jotka ovat todisteena liiketoimintaan liittyvistä päätöksistä ja transaktioista (Cisco etc 1999).

Asiakirjahallinto (Record Management - RM) ja
Arkistointijärjestelmä (Record Management System - RMS):

Asiakirjahallinto tai arkiston hallinta liittyy asiakirjojen, ei niinkään dokumenttien, hallintaan. Tätä tarkoitusta varten on olemassa omat ohjelmansa, jotka keskittyvät nimenomaan lainvoimaisten dokumenttien eli asiakirjojen arkistointiin ja hallintaan. Usein asiakirjojen hallintaan liitetään läheisesti myös erilaiset skannaus- tai kuvantamistekniikat, koska hyvin monet sopimukset liikkuvat vielä paperina eri osapuolten välillä ja näiden saattaminen sähköiseen muotoon edellyttää pääsääntöisesti skannausta (Virtanen 2001.)

Järjestelmää, joka pyrkii vastaamaan asiakirjahallinnon haasteisiin, kutsutaan Arkistointijärjestelmäksi.

ASP –ratkaisut (Application Service Provider):

ASP –ratkaisut ovat palveluja, joissa palvelun tarjoaja vuokraa sovellusta tai/ja palvelintilaa asiakkaan käyttöön internetin yli. ASP –palvelun käyttäjä ei joudu investoimaan erikseen palvelimiin, eikä niiden ylläpitohenkilökuntaan, ohjelmistolisensseihin, eikä jossain tapauksessa edes palvelua pyörittävään henkilökuntaan, vaan nämä asiat hoidetaan palvelun tarjoajan toimesta. Yleensä tällainen palvelu hinnoitellaan esim. kuukausimaksupohjaisesti, jolloin asiakas pystyy budjetoimaan palvelun käytön aiheuttamat kulut huomattavasti tarkemmin.

Digitaalinen Aineistonhallinta, (Digital Asset Management, DAM; Media Asset Management, MAM):

Digitaalinen Aineistonhallinta on yleistermi digitaalisessa muodossa olevien tietojen hallintamenetelmille. Aineistonhallinta tarkoittaa kaiken uudelleenkäytettävän digitaalisen tiedon – kuvien, äänen, videon, tekstien sekä valmiiden sivujen ja dokumenttien – organisointia nopeaa etsintää, noutamista ja uudelleenkäyttöä varten. (Nurmi ym. 2001.)

Selvyyden vuoksi erotettakoon erikoistuneet aineistonhallintajärjestelmät dokumenttien- ja asiakirjojenhallintajärjestelmistä (DMS, EDMS, RM, RMS). Tässä tutkielmassa keskitytään erityisesti aineistonhallintajärjestelmiin.

Dokumentti (Document):

Sanaa dokumentti käytetään melko runsaasti arkikielessä ja yleensä sillä viitataan jollakin toimisto-ohjelmalla luotuun tiedostoon tai vaikkapa käsin kirjoitettuun paperiin. Virallisemmin dokumentilla tarkoitetaan eräänlaista säiliötä, joka sisältää informaatiota, tai vaikka vain dataa ja se vaatii jonkinlaista hallintaa tai johtamista (Virtanen 2001).

Hot folder – toimintokansio:

Järjestelmän valvoma toimintokansio. Yleisesti työnkulkujärjestelmien käytössä oleva ohjaustekniikka. Kun järjestelmä huomaa tiedoston ilmestyneen toimintokansioon tutkii se saapuneen materiaalin, tekee johtopäätökset sekä käynnistää prosessin seuraavan vaiheen.

Kolmikerrosarkkitehtuuri:

Ohjelmistoarkkitehtuuri, jossa ohjelman eri toiminnalliset tasot on erotettu omiksi ohjelmakerroksiksiin; näitä kerroksia ovat käyttöliittymä-, välitaso- ja tallennuskerros. Kerrosarkkitehtuurin etu on se, että muutokset yhdessä kerroksessa eivät aiheuta muutosten tarvetta muissa kerroksissa. Lisäksi välitaso –kerroksen toiminnallisuutta voidaan hajauttaa eri tietokoneille kuormituksen mukaan.

Metadata:

Yksinkertaisimmillaan metadatan voidaan sanoa olevan rakenteista dataa datasta. Metadata on kuvaavaa informaatiota konkreettisesta tai sähköisestä objektista tai resurssista. Metadata on terminä suhteellisen uusi, mutta sen taustalla oleva konsepti on ollut käytössä niin kauan kuin informaatiokokoelmia on järjestelty. Kirjaston

kirjakorttiluettelot (kortistot) edustavat vakiintunutta ja tunnettua metadatan tyyppiä, joka on palvellut kokoelmien hallinnan ja lähteiden hakemisen työkaluina vuosisatoja. Metadata voidaan tuottaa joko käsin tai se voidaan saada automaattisesti ohjelmalla. Esimerkiksi yksittäisen dokumentin luonti- ja tallennuspäivämäärät, otsikko, aihe, koko tai esim. sen juokseva numero ovat metadataa.

Monimediajulkaiseminen (Cross-media julkaiseminen):

Monimediajulkaisemisella tarkoitetaan samasta lähdeaineistosta erilaisille medioille niiden omilla ehdoilla tuotettavia julkaisuja. Ideana on erottaa ulkoasu sisällöstä. Näitä medioita voivat olla perinteisen printin lisäksi esim. WWW, WAP, DTV jne. Monimediajulkaisemisella saavutetaan huomattavia säästöjä, kun sisältöä ei tarvitse tuottaa erikseen, omina työketjuinaan jokaiselle medialle. (Nurmi ym. 2001.)

Natiivi asiakasohjelma:

Natiivi asiakasohjelma on perinteisellä ohjelmoinnilla esim. Visual Basic tai C++ kehitysympäristöllä toteutettu ns. perinteinen ohjelma, jolla ollaan yhteydessä rajapintojen avulla tietokantoihin ja palveluihin.

Natiivitiedosto:

Natiivitiedosto on esim. taitto-ohjelman oma tallennusmuoto, joka ei ole yleensä ole standardia, muiden ohjelmien tuntemaa muotoa. Ohjelman natiivitiedoston tiedostomuodosta voi tosin kehittyä alan de facto standardi -tiedostomuoto, jonka käsittelyä muutkin ohjelma hallitsevat ainakin määrättyyn tasoon asti. Esimerkkinä tässä mainittakoon vaikka Adobe PhotoShop kuvankäsittelyohjelman oma kuvamuoto (.psd).

ODBC tietokantaliitäntä –rajapinta:

ODBC (Open DataBase Connectivity) on avoin tietokantaliitäntä –rajapinta, jonka avulla voidaan käyttää standardilla tavalla mitä tahansa tietolähdettä, johon on saatavissa ODBC –ohjain (ODBC -driver). ODBC on alun perin Microsoftin kehittämä. ODBC luo välitason (middle layer), jonka avulla ohjelman tekemät tietokantakyselyt kääntyvät tietokantajärjestelmän tuntemaan muotoon. Kyselyjen tulokset välittyvät takaisin ohjelman tuntemassa muodossa.

JDBC tietokantaliitäntä –rajapinta:

JDBC (Java Database Connectivity) on Java -pohjainen tietokantaliitäntä –rajapinta, jonka avulla Java -sovellukset voivat suorittaa SQL-lauseita. JDBC on samankaltainen kuin

ODBC, mutta suunniteltu erityisesti Java ohjelmille, kun taas ODBC on kieliriippumaton rajapinta. JDBC:n on kehittänyt JavaSoft, Sun Microsystemsin tytäryhtiö.

Sisällönhallinta (Content Management - CM):

Sisällönhallintaan erikoistunut Content-wire.com -internetsivusto (2001) määrittelee sisällönhallinnan seuraavasti: ”Yksinkertaisimmillaan sisällönhallinta on kokoelma sääntöjä ja prosesseja, joilla myötävaikutetaan sisällön luomiseen, ohjataan yhteistyötä sekä hallitaan itse sisältöä. Se voi sisältää dokumentinhallintaa läpi koko yleisen asiankäsittelyn, keskitettyyn sisältökantaan pohjautuvan median hallintaa sekä yleisiä työkaluja sisällön luomista ja julkaisua varten. Laajimmillaan se on yhdistelmä sovellustyökaluja ja liiketoimintaprosesseja, joiden avulla yritykset voivat tehokkaasti hallita ja jakaa suuria määriä hajanaista informaatiota erilaisiin medioihin niille parhaiten sopivilla tavoilla”. (Virtanen 2001.)

Sisällönhallinta voidaan lyhyesti määritellä kyvyksi käsitellä sisältöä koko sen elinkaaren ajan ja samalla ylläpitää sisältöä erillään sen esittämisestä (Turocy 2001, 2).

Työmääräin (Job Tickets):

Työmääräimet ovat dokumentteja, jotka sisältävät painotyön tekijälle tietoja asiakkaasta, työn suoritustavasta ja jälkikäsittelystä.

Työnkulun Ohjausjärjestelmä (Workflow System):

Työnkulku (workflow) on määritelty sarja tehtäviä, joiden tavoitteena on tuottaa määritelty lopputulos. Erikoistunut työnkulun ohjausjärjestelmä (workflow system) antaa mahdollisuuden määritellä erilaisia työnkulkuja erityyppisille töille. Esimerkiksi julkaisuympäristössä dokumentti voidaan automaattisesti reitittää kirjoittajalta toimittajalle, toimittajalta oikolukijalle, ja lopulta oikolukijalta tuotantoon. Kaikissa työnkulun vaiheissa vaiheen toteuttaja (henkilö, ryhmä tai ohjelma) on vastuussa määrätystä tehtävästä. Kun tehtävä on suoritettu työnkulun ohjausjärjestelmä tiedottaa siitä seuraavan vaiheen suorittajalle sekä toimittaa vaiheessa tarvittavan datan. (Nurmi ym. 2001.)

Välitaso –kerros (Middleware –kerros):

Kts. Kolmikerrosarkkitehtuuri:Kolmikerrosarkkitehtuurin keskimmäinen kerros, eli ns. Välitaso –kerros, sisältää yleensä ohjelmiston varsinaisen bisneslogiikan.

3.8 Tiedon tallennusmuodot

Aineistonhallintajärjestelmissä tietoa voidaan tallentaa joko binääriobjekteina suoraan tietokantaan tai tiedostoina levyjärjestelmään, jolloin tietokantaan tulee ainoastaan linkki varsinaiseen tiedostoon.

Järjestelmällä hallittavien tiedostojen pitäisi voida olla mitä tahansa digitaalista muotoa, kuitenkin eräissä järjestelmissä on rajoitettu ns. tuettujen tiedostomuotojen valikoimaa. Järjestelmään voidaan rakentaa erilaisiin tiedostomuotoihin liittyen niiden hallintaa helpottavia ominaisuuksia. Esimerkiksi tietyntyyppisistä tiedostoista voidaan automaattisesti generoida ns. näyttöversioita, tai niiden sisältöjä voidaan analysoida hakutoimintoja helpottamaan.

Mikäli aineistonhallintajärjestelmää käytetään julkaisujen tuottamiseen, pitää järjestelmän pystyä luetteloimaan ja arkistoimaan myös kokonaiset julkaisut. Julkaisun tallennusmuodon on hyvä olla rakenteinen ja mahdollistaa viitteet aineistoelementteihin tai elementtien indeksointi suoraan julkaisuihin. (Nurmi ym. 2001.)

Joka tapauksessa järjestelmään tallennetaan aina tiedostoon liittyvää tietoa, ns. metadataa, tietoa tiedosta. Toisissa järjestelmissä metadataa on vain rajoitetusti, toisissa hyvinkin runsaasti. Aineistojen uudelleen käytön ja löytymisen kannalta metadatan arvo on kuitenkin äärettömän suuri; sitä paremmin sopiva tiedosto löytyy, mitä tarkemmin se on metadatassaan kuvattu.

Joissakin tiedostomuodoissa voi olla jo tiedostorakenteeseen tallennettuna metatietoa tiedoston sisällöstä, kuten luontiaika, paikka, tekijä, laitteisto sekä tekijänoikeustiedot. Metatietoa sisältäviä tiedostomuotoja ovat mm. EPS-, GIF- ja FlashPix -kuvatiedostot, erilaiset äänitiedostot sekä QuickTime –videot. Adobe on kehittämässä koko tuotekirjossaan käyttöön tulevaa XMP –metadatatekniikkaa ja lehtikuvien hallintaan määritelty ns. IPTC saatetieto, jota pystytään hyödyntämään mm. PhotoShop -ohjelmistossa.

Aineistonhallintajärjestelmän tulisikin tukea suoraan mahdollisimman suurta joukkoa erilaisia tiedostomuotoja ja pystyä hyödyntämään automaattisesti niissä valmiina olevaa metatietoa aineiston järjestämisessä ja luokittelussa. Automaattisesti saatavan tiedon lisäksi yritysten kannattaa ehdottomasti panostaa kuvaavan tiedon lisäämiseen omissa aineistonhallintajärjestelmissään.

Tekijänoikeustietojen merkitys digitaalisessa aineistossa kasvaa jatkuvasti, joten nykyisissä järjestelmissä kiinnitetäänkin paljon huomiota tekijänoikeustietojen sisällyttämiseen järjestelmän lisäksi myös itse tiedostoihin, joko upotettuna metadatana, tai erilaisina digitaalisina vesileimoina.

3.9 Aineistonhallintaan liittyviä suosituksia

Digitaalisesta aineistosta tallennettu metatieto määrää aineiston arvon, sillä ilman käyttökelpoista metatietoa aineistojen löytäminen on vaikeaa. Aineistojen hallinnan alueelle on kehitetty XML –pohjaisia sanastoja, jotka mahdollistava sekä aineistojen, että aineistoja koskevan metatietojen vaihtamisen eri järjestelmien välillä. Tarkoituksena on edistää digitaalisten aineistojen uudelleen käyttämistä. Tämä kehitystyö on kuitenkin vasta alussa. Pisimmällä vuosituhanteen alussa oltiin Yhdysvalloissa, missä sikäläisen yhteistyöorganisaation Graphic Communication Associationin alaisuudessa toimiva Idealliance julkisti huhtikuussa 2001 ensimmäisen version Prism -suosituksesta (Publishing Requirements for Industry Standard Metadata). (Nurmi ym. 2001.)

Digitaalisille kuvatiedostoille on puolestaan syntynyt niin ikään XML –pohjainen sanasto DIG35. Sillä on merkitystä myös aineistojen hallinnassa. Onhan suuri osa tallennettavasta aineistosta edelleenkin kuvia.

IPTC –kentät (IPTC IIM Datasets):

IPTC:n (International Press and Telecommunication Council) on määritellyt useita sanomalehtityössä noudatettavia standardeja. Information Interchange Model (IIM) Record 2 pitää sisällään DataSet käsitteen toimituksellisen tiedon sisällyttämiseen alun perin JPEG muotoisiin kuvatiedostoihin. Adobe on ottanut kentät käyttöön Photoshop -ohjelmistossaan ns. ”File Info” -kentissä. (IPTC 2001.)

DIG35:

Digital Imaging Group julkisti vuoden 2000 syksyllä DIG35 metatietosuosituksen. Sen avulla kuvaajat, kuvien käyttäjät voivat liittää kuvaan erilaisia metatietoja. Ennen kuin tämä on mahdollista, pitää kuvien käsittelyohjelmistojen valmistajien implementoida DIG35 omiin sovelluksiinsa. Tämän jälkeen DIG35- suositusten mukaisia tietoja voidaan siirtää sovellusohjelmien välillä. DIG35 mahdollistaa mm. sen, että valokuvaaja voi liittää otok-seensa tietoja mm. kameran asetuksista, kuvausajankohdasta, sijainnista (GPS -paikan-nuksen avulla) sekä tekijänoikeuksista. Kuvan käyttäjä voi puolestaan kirjoittaa metatietoihin vaikkapa kuvatekstin tai muuta kuvaa koskevaa tietoa, kuten luokitus- ja tunnistetietoa sekä tietoa kuvan aiheesta. (Nurmi ym. 2001, DIG35 2000.)

Sovellusten kehittäjille Kodak on julkistanut DIG35 –suositusta tukevan Kodakin Picture Metadata Toolkitin. DIG35 ei rajoitu mihinkään kuvaformaattiin, mutta Digital Imaging Group on tehnyt yhteistyötä saadakseen määrityksensä vuosituhannen alussan valmistumisvaiheessa olleeseen uuteen JPEG2000 –kuvantallennusmuotoon. Kansainvälinen standardisoimisjärjestö, joka vastaa JPEG2000 –suosituksen kehittämisestä, on sisällyttänyt DIG35 –määrityksen mukaiset kentät JPEG2000 –formaatin kakkososaan (Part 2). Vastaavasti yhteistyötä on tehty ISO:n MPEG-7 –ryhmän kanssa.

DIG35 on XML –pohjainen ja siitä on olemassa sekä XML DTD, että XML Schema. Digital Imaging Groupissa on edustettuna mm. Adobe Systems Inc., Agfa-Gevaert N.V., Canon Inc., Eastman Kodak Company, Fuji Photo Film Co., Ltd., HP ja Microsoft Corporation. (Nurmi ym. 2001.)

PRISM (Publishing Requirements for Industry Standard Metadata):

PRISM eli Publishing Requirements for Industry Standard Metadata –suositus määrittelee XML –pohjaisen metatietosanaston digitaalisille aineistoille. Sen sovellusalueita ovat aikakauslehtiaineistot, kirjat, luettelot, kuvat ja uutisaineistot. Kyse on tällaisten aineistojen julkaisemiseen ja uudelleenkäyttöön liittyvän tiedon hallinnasta. PRISM –suosituksessa on määritelty aineiston metatieto sekä digitaalisen aineiston siirtämisen vaatima paketointi. Tällaista paketointia tarvitaan erityisesti kuva- ja uutisaineistojen välityksessä, mutta myös digitaalisen sisällön jälleenmyynnissä. Tarkoituksena on synnyttää PRISM –sanastoa tukevia ohjelmistoja siten, että digitaalisten aineistojen ohella aineistoa koskevat metatietoja voidaan vaihtaa näiden tietojärjestelmien välillä.

Ensimmäisiä PRISM –suosituksen soveltajia vuosituhannen alussa oli aikakauslehtien kustantaja, mediajätti Time-Warnerin Time, Inc, joka aineistojen esittämisen, arkistoinnin ja jakelun perusta on PRISM –suosituksessa. Tavoitteena on tehostaa sisällön hallintaa talon sisällä. Lisäksi PRISM –suosituksen mukaiset aineistot mahdollistavat Timen edustajan mukaan uusien liiketoimintamallien kehittämisen. Kyse on siitä, että sisäistä tuotantoa tehostetaan ohjaamalla sitä ihmisten asemasta aineistoon liitetyllä metatiedolla, jota tietokoneet osaavat tulkita.

PRISM:ssä hyödynnetään Dublin Corea (kirjastolähtöinen metatietosanasto julkaisuille), NITF:iä (uutisaineistojen merkintään tehty suositus) ja ICE:ia (aineistojen jälleenmyyntiin kehitetty sanasto ja yhteyskäytäntö). PRISM noudattaa myös W3C:n RDF –suositusta, joka on suunniteltu metatiedon esittämiseen ja jakeluun. (Nurmi ym. 2001, PRISM 2001.)

JDF (Job Definition Format):

XML –pohjainen JDF –sanasto (Job Definition Format) on ensisijaisesti tehty perinteistä paino-prosessia varten. Sen avulla erilaiset graafisen alan tuotantojärjestelmät saadaan keskustelemaan toistensa kanssa sekä ylemmän tason tuotannonohjausjärjestelmien kanssa. Aluksi JDF:n kohdealue oli painotuotanto, mutta mm. Adoben ja Agfan edustajat visioivat vuosituhannen alussa JDF:n laajentuvan myös aineistojen hallinnan puolelle, mutta tiettävästi mitään konkreettisia suunnitelmia ei silloin ollut.

JDF –suosituksen ensimmäinen versio hyväksyttiin keväällä 2001 ja ensimmäisten JDF –pohjaisten järjestelmien odotettiin vuonna 2002 tulevan markkinoille vielä kuluvan vuoden aikana. JDF:n menestystä on tällä hetkellä vaikea ennakoida, mutta alan merkittävien yritysten, kuten Adobe, Agfa, Heidelberg, MAN Roland sekä suuri joukko pienempiä laite- ja järjestelmätoimittajia, mukana oleminen antaa sille hyvät edellytykset. JDF:n kehitystyötä tekee CIP4 –organisaatio (Cooperation for the integration of Process in Prepress, Press and Postpress). (JDF 2001, Nurmi ym. 2001.)

XMP (Adobe) (The eXtensible Metadata Platform):

Seuraava XMP:n kuvaus on peräisin Adoben tuotekuvauksista sekä White Paperseista ADOBE 2001 ja ADOBE WP 2018.

Metadatan merkitys on kasvussa kaikentyyppisessä julkaisemisessa. Metadataa sisältävät dokumentit voivat huomattavasti parantaa hallittujen tietovarastojen käyttökelpoisuutta tuotannon ohjelmistojen välisissä työnkuluissa. XMP (The eXtensible Metadata Platform) on Adobe -sovellusten ja työnkulkuun osallistuvien muiden sovellusten yhteinen XML kehys (framework), joka standardoi dokumenttien metadatan luomisen, käsittelyn ja vaihdon julkaisutyönkuluissa. (XMP 2001.)

XMP koostuu kehyksestä, skeemasta, XMP pakettiteknologiasta, ja XMP ohjelmistokehitysympäristöstä (SDK), joka on saatavissa käyttöön vapaalla open-source lisenssillä. XMP perustuu W3C:n avoimeen metadata-standardiin, joka tunnetaan nimellä Resource Description Framework (RDF). (RDF 2001.)

XMP upottaa metadatan natiivitiedostojen sisälle. Koska metadata on liitetty tiedoston sisälle, säilyttää tiedosto tiedon ympäristöstään, vaikka se kuljetettaisiin ulos alkuperäisestä ympäristöstään. Upotettu metadata voi pitää sisällään minkä tahansa XML skeeman, mikäli se on kuvattu RDF syntaksin mukaisesti. Laajennettava, upotettu metadata

natiivitiedostoissa tuo merkittävästi uusia mahdollisuuksia uudelleen hyödyntämiseen, arkistointiin, ja automaatioon julkaisujärjestelmien työnkuluissa.

Koska XMP on saatavissa vapaalla open-source lisenssillä, voidaan se integroida mihin tahansa järjestelmään tai ohjelmistoon. Adobe on integroinut XMP:n Adobe® Acrobat® 5.0, Adobe InDesign® 2.0, ja Adobe Illustrator® 10. ohjelmistoihin. Muihin Adobe –tuotteisiin XMP tulee myöhemmin.

XMP kehystä ovat ilmoittaneet tukevansa sellaiset yhtiöt kuin Documentum, IBM, Kodak, KPMG, North Plains Systems, ja monet muut.

Tärkeimmät edut:

- Koska XMP upottaa metadatan natiivitiedostojen sisälle, säilyttävät tiedostot tiedon alkuperäisestä ympäristöstään, vaikka se välitettäisiin tietokannan tai julkaisujärjestelmän ulkopuolelle.
- Koska XMP on laajennettava, voi käyttäjä sisällyttää minkä tahansa XML skeeman XMP paketin sisälle, mikäli se on kuvattu RDF syntaksin mukaisesti
- XMP perustuu W3C standardeihin, jonka ansiosta integrointi on varsin selkeää. Metadata kehyksen standardointi mahdollistaa metadatan välittämisen monien erilaisten järjestelmien ja ohjelmistojen välillä.
- XMP on saatavissa vapaalla open-source lisenssillä. Käyttäjät ja integraattorit pääsevät hyödyntämään lähdekoodia ohjelmistokehitysympäristön (SDK) avulla.
- Acrobat 5.0, InDesign 2.0, Illustrator 10, ja lopulta kaikki muutkin Adobe sovellukset tulevat tukemaan XMP:tä. Tämä mahdollistaa helpon metadatan välityksen Adobe sovellusten välillä.
- XMP on integroitavissa mihin tahansa järjestelmään tai ohjelmistoon. Koska kehys on täysin laajennettavissa, voidaan se laajentaa sisältämään minkä tahansa XML skeeman.

3.10 Tekninen toteutus

3.10.1 Dokumenttien varastointi, tietokantatekniikat

Useat aineistonhallintajärjestelmät perustuvat tunnettuun tietokannanhallintajärjestelmään tai osaavat käyttää sellaiseen tallennettua aineistoa. Yleisimpiä aineistonhallintajärjestelmien käyttämiä kaupallisia relaatiotietokantoja olivat vuonna 2002 Oracle ja Microsoft SQL Server. Kaupallista tietokantaa käyttävissä järjestelmissä varsinainen aineistonhallintajärjestelmä on lähinnä käyttöliittymän tarjoava ohjelmisto varsinaisen tietokannan päällä. Kaupallisen

tietokantatuotteen käyttö ratkaisee monia aineistonhallintajärjestelmän toteutuksen käytännön ongelmia. Sillä on kuitenkin myös varjopuolia, joista tärkein on hinta. Tietokannanhallintajärjestelmät, samoin niiden asiakaslisenssit, ovat kalliita. (Nurmi ym. 2001.)

Mikäli aineistonhallintajärjestelmällä halutaan käsitellä kokonaisia julkaisuja tai sivuja, tietokantaratkaisun tuleekin pystyä hallitsemaan monimutkaisia olioita sekä niiden suhteita. Tällaiseen käyttöön perinteiset relaatiotietokannanhallintajärjestelmät eivät ole parhaita mahdollisia ratkaisuja. Oliotietokannat sopisivat monimutkaisen tiedon tallennukseen ja olioiden välisten suhteiden mallinnukseen, mutta useimmat kaupalliset toteutukset eivät vuonna 2002 olleet luotettavuudeltaan relaatiotietokantajärjestelmien tasolla.

Kolmas lähestymistapa on käyttää tekstimuotoisen datan tallentamiseen tekstitietokantaa tai strukturoituja rakenteita tukevia natiivi XML –tietokantoja. Näissä ratkaisuissa päätietokantaa käytetään lähinnä metadatan tallentamiseen, tallennetut tiedostot löytyvät tietokantaan tallennettujen linkkien kautta levyjärjestelmästä. Yleensä näissä tapauksissa järjestelmän hallinnollista dataa kuitenkin säilytetään tekstitietokannan rinnalla olevassa relaatiotietokannassa, joka voi olla myös jokin kevyempi järjestelmä kuten vuonna 2002 suomalaisen MySQL AB:n Open Source tietokanta MySQL (MySQL 2001) tai Huges Technologiesin MiniSQL (MiniSQL 2001).

3.10.2 Tekstitietokannat vs. relaatiotietokannat

Tekstitietokantojen etu on niiden kyvyssä indeksoida tehokkaasti tekstitiedostojen sisältöjä, jonka ansiosta tietokantamoottorille tehdyt haut ovat erityisen nopeita. Tekstitietokantoihin voidaan myös liittää erityisiä kieliominaisuuksia, kuten esim. suomen kielen taivutusten hallinta tai erilaisia tesaurus- ja sanakirjatoiminteita. Kyselyjä voidaan suorittaa myös luonnollisen kielen muodossa.

Sanojen taivutusten hallinta tarkoittaa sitä, että kun järjestelmästä haetaan sanan taivutetulla muodolla löytää hakukone myös sanan muiden taivutusten esiintymät. Tesaurus– ja sanakirja ominaisuudet tarkoittavat, että haettaessa jollain sanalla löytää järjestelmä myös dokumentit joissa esiintyy samaa tarkoittava, esim. vene -> pursi, tai vene -> boat, båt. Joissakin ratkaisuissa voidaan myös liikkua käsitetasolta ylös ja alas, esim. jos leopardi ei löydy haetaan muita kissapetoja, jos niitä ei löydy haetaan petoeläimiä.

Tekstitietokantojen hakuominaisuudet ovat erityisen käyttökelpoisia laajojen strukturoimattomien tietomassojen hallinnassa. Järjestelmä sisältävät yleensä tuen suurelle määrällä tekstitiedostoformaatteja. Tiedostot pystytään siirtämään natiivitiedostomuodosta

tekstitietokantaan ja takaisin. Eräs tunnetuimmista ja laajimmin käytössä olevista tekstitietokannoista on BRS/Search, jonka vuonna 2002 omisti Open Text Corporation (BRS 2002).

Tekstitietokannat ovat yleensä rakenteeltaan litteitä ja siksi niitä ei yleensä kannata käyttää relaatioita sisältävien tietokantarakenteiden tallentamiseen. Vastaavasti relaatiotietokannat ovat lisänneet ominaisuuksiinsa tekstipohjaisen aineiston indeksointi- ja hakumahdollisuuksia. Mm. Oracle sisältää uusimmissa versioissa tällaisia mahdollisuuksia. Lisätietoa mm. Oracle Text Application Developers Guide (Oracle 2018).

3.10.3 Natiivi XML –tietokannat vs. relaatiotietokannat

Metadatan merkityksen kasvaessa XML -muotoinen data tulee yhä tavallisemmaksi aineistonhallintajärjestelmissä. Tämä herättääkin ajatuksen tiedon tallentamisesta XML –muodossa natiivi XML –tietokantaan ja XML –työkalujen käytöstä tietojen hakemisessa ja muokkauksessa. Natiivi XML –tietokantoja olikin tarjolla vuonna 2002 jo useita mm. Software AG:n Tamino (Tamino 2002) ja Ixiasoft Inc.:n TextML Server (Ixiasoft 2002). (Dyck 2001.)

Samalla perinteiset relaatiotietokantavalmistajat olivat tuolloin päättäneet vastata haasteeseen kehittämällä omien tietokantajärjestelmiensä kykyä hallita natiivia XML –dataa. Erityisesti Oracle Corporation oli tietokantansa Oracle9i release 2 versiossa kehittänyt XML –ominaisuuksiaan voimakkaasti ja kilpaili niissä natiivi XML –tietokantojen kanssa. Myös muut toimijat kuten esim. IBM ja Microsoft olivat kehittämässä vastaavia ominaisuuksia omiin relaatiotietokantaratkaisuihinsa.

Toinen asia sitten on kannattaako XML –muodossa olevaa tietoa kuitenkaan tallentaa XML –tietokantaan. Varsinkin silloisissa natiivi XML -tietokannoissa oli vielä paljon puutteita mm. niiden hallinta-, ohjelmointi-, rajapintaominaisuuksissa verrattuna isojen relaatiotietokantojen vastaaviin ominaisuuksiin. Lisäksi XML –hakukielien standardeissa oli vielä selviä puutteita. Kehitelty XPath-kyselysyntaksi ei tukenut ryhmittely-, järjestys-, eikä summaustekniikoita, ja paljon monipuolisempi XQuery-kyselykieli oli vielä luonnosvaiheessa. (Dyck 2001.)

Tuolloin oli selvää, että tulevina vuosina kaikkien tietokantatuotteiden tuli pystyä nopeasti verifioimaan, tallentamaan ja palauttamaan dataa XML –muodossa. Aika tulisi näyttämään saavuttaisivatko natiivi XML –tietokannat perustietokantaominaisuuksissaan perinteiset relaatiotietokannat ennen kuin perinteiset relaatiotietokannat ovat saaneet toimintoihinsa XML –ominaisuudet, vaiko päinvastoin. Tuolloin eWEEK julkaisun artikkelin kirjoittajan Timothy Dyckin käsitys asiasta oli, että historiaan ja hänen kokemuksiinsa perustuen

perinteiset relaatiotietokannat voittavat, ja että relaatiotietokantakoneet ovat oikea paikka sekä XML, että ei-XML datalle. (Dyck 2001.)

Natiivi XML –tietokantojen tärkein etu on niiden vapaan muodon dokumenttiorientoitunut tietokantakone (storage engine). XML –dokumenttien muotoa ei tarvitse kuvata ennen niiden tallentamista tietokantaan. Tähän perustuen natiivit XML –tietokannat pystyvät käsittelemään hyvin osittain strukturoitua dataa (semistructured data). Dyckin mukaan natiivi XML –tietokannat olivat oikea työkalu organisaatioille, joiden sovellukset ovat suuntautuneet kokonaisten dokumenttien tallentamiseen. Tällaisia dokumentteja voivat olla esimerkiksi manuaalit, esitteet ja web-sivut. (Dyck 2001.)

3.10.4 Rajapinnat ja integrointi muihin järjestelmiin

Parhaimman hyödyn aineistonhallintajärjestelmästä ja sen sisältämästä arvokkaasta materiaalista saa integroimalla järjestelmä kiinteästi yrityksen prosesseihin ja käytössä oleviin toisiin ohjelmiin ja järjestelmiin. Järjestelmä tulisi saada osaksi yrityksen jokapäiväistä toimintaa, sovittuja toimintatapoja ja laatujärjestelmän ohjeita.

Aiemmin digitaaliset aineistojen hallintajärjestelmät olivat tyypillisesti arkistoja, kuten kuva-pankkeja, jonne säilöttiin aineistot tulevaa käyttöä varten. Vuonna 2002 tarjolla oli jo laajempia, digitaalista työnkulkua tukevia järjestelmiä. Näihin järjestelmiin voitiin integroida aineistojen käsittelyyn tarkoitettuja ohjelmistoja. Integraation taso oli oleellinen. Löyhässä integraatiossa käyttäjä joutuu antamaan käsiteltävää aineistoa koskevan metatiedon manuaalisesti. Käyttäjän kannalta parempi ratkaisu on se, että aineistojen hallintajärjestelmä ”seuraa” käyttäjän toimenpiteitä sovellusohjelmassa siten, että esimerkiksi dokumentin ja sen muodostavien komponenttien suhteet rekisteröityvät automaattisesti aineistojen hallintajärjestelmään. Myös sovellusohjelman ns. header-tiedot sekä muut hallinnolliset tiedot, kuten tiedoston nimi, tyyppi ja luontiaika pitää saada aineistojen hallintajärjestelmään talteen mahdollisimman automaattisesti. Jos järjestelmään tallennettuihin aineistoihin ei liitetä metadataa, ei niiden löytyminen seuraavan kerran tarvittaessa ole lainkaan varma, tai se on lähes mahdotonta. Toisin sanoen järjestelmän käyttökelpoisuus on suoraan verrannollinen sen metadatan tasoon. Käyttäjän kannalta paras tilanne olisi se, että hän voisi käyttää sovellusohjelmasta suoraan aineistojen hallintajärjestelmää ilman että koko ajan pitää tehdä check-out/check-in operaatioita aineistolle. (Nurmi ym. 2001.)

Elektroniseen julkaisemiseen painottuvat järjestelmät ovat usein integroitavissa web-sivuston hallintajärjestelmään tai aineisto voi olla käytettävissä suoraan web-sivuston sisällöksi.

Painotöiden valmistukseen tarkoitetut järjestelmät integroidaan usein muiden tukijärjestelmien kanssa. Tällöin on tavallista, että aineistonhallintajärjestelmä tukee yrityksessä käytössä olevaa OPI –järjestelmää, jonka avulla aineistosta voidaan käyttää ns. kevyempiä taittoversioita ja liittää raskaat originaalikuvat vasta tulostettaessa.

Teknisesti järjestelmien integrointi tapahtuu yleensä ohjelmistojen omia ohjelmointirajapintoja käyttäen (API). Tämä saattaa olla hyvinkin hankalaa, sillä kaikissa, varsinkaan vanhemmissa järjestelmissä ei tuolloin ollut standardeja rajapintoja tai sitten rajapintoja ei ollut lainkaan. Uudemmissa järjestelmissä integrointitarpeet oli jo huomioitu varsin hyvin.

Tuolloin arvioitiin, että XML –standardin kehittyessä sitä tultaisiin hyödyntämään yhä enemmän rajapinnoissa. Oli kehitetty toimialakohtaisia XML –sanastoja, joita tukemalla voitaisiin integroitua helpommin kuin täysin omia rajapintoja käyttäen. Etäällä toisistaan sijaitsevien järjestelmien välille voitiin järjestää kommunikointia hyödyntäen esim. HTTP -protokollaa. XML-RPC –rajapinnan (XML Remote Procedure Call) (XML-RPC 2002) avulla välitetään funktiokutsuja käyttäen HTTP –protokollaa tiedonsiirtoon ja XML –muotoa tiedon koodaamisen. Protokolla on suunniteltu mahdollisimman yksinkertaiseksi mahdollistaen kuitenkin monimutkaisten data struktuurien välittämisen, käsittelyn ja palauttamisen.

Relaatiotietokannoilla oli jo standardirajapintoja (ODBC/JDBC), joiden avulla sovelluksen ei tarvinnut tietää toisesta puolesta sen enempää kuin, että kommunikoinnin yhteinen kieli oli määritelty.

3.11 Aineistonhallintajärjestelmien komponentit

3.11.1 Dokumenttien tallentaminen järjestelmään

Järjestelmän käytön helppous riippuu pitkälle siitä, millaista aineistoa organisaatio käsittelee ja millaiset syöttöominaisuudet järjestelmä tarjoaa. Uuden aineiston lisäämisen tulisi olla mahdollisimman helppoa. Aineistonhallinnan olisi hyvä integroitua organisaation muihin ohjelmistoihin siten, että uusi tai muokattu aineisto voidaan tallentaa järjestelmään suoraan esimerkiksi käytettävän ohjelmiston ”Tallenna nimellä” -toiminnolla. Aineisto voidaan myös ”tulostaa” aineiston hallintajärjestelmään käyttäen tarkoitusta varten suunniteltua kirjoitinohjainta. Tulostuksesta ei kuitenkaan saada alkuperäisen tiedoston rakennetta, kuten mahdollisia kerroksia eikä mahdollista tiedostoon liitettyä metatietoa. (Nurmi ym. 2001.)

Mahdollisen suoran sovellustuen lisäksi järjestelmän on hyvä tukea sovellusriippumattomia automaattisia aineiston syöttötapoja. Eräs mahdollisuus on käyttää toimintokansioita. Kun kansioon tallennetaan, siirretään tai kopioidaan tiedosto, järjestelmä käsittelee sen automaattisesti ja liittää tiedostoon kansion asetusten mukaiset metatiedot. (Nurmi ym. 2001.)

Dokumenttien tallentamisen yhteyteen voidaan myös liittää monenlaisia lisätoiminteita, mm. tallennettavan aineiston analysointia, metadatan ennakkoerottelu tai erilaisten esikatselumuotojen luonti aineistosta.

3.11.2 Dokumenttien luokittelu, metadata

Jotta aineistoa voidaan etsiä aiheen tai sisällön perusteella, on sen oltava luokiteltu erilaisiin kategorioihin.

Käytettäessä manuaalisesti ylläpidettävää hakemistorakennetta jokainen hakemisto on yksi kategoria, joka voi edelleen sisältää alikategorioita. Aineisto luokitellaan sijoittamalla tiedostot sopiviin hakemistoihin. Ennemmin tai myöhemmin aineiston määrän kasvaessa tällainen järjestelmä käy kuitenkin riittämättömäksi. Kategorioiden määrä kasvaa hallitsemattomaksi joko leveys- tai syvyyssuunnassa, tai jokaiseen kategoriaan tulee liian paljon aineistoa. Tietyn aineistoelementin löytäminen suuresta aineistomäärästä tulee hankalaksi, koska pelkässä hakemistorakenteessa ei ole mitään oleellista lisätietoa helpottamaan tietyn aineistoelementin tunnistamista ja löytämistä. (Nurmi ym. 2001.)

Hakemistohierarkian ongelmana on myös se, ettei aineistoa voida sijoittaa kuin tiettyyn hakemistoon tekemättä elementistä kopiota. Tietty aineisto voi kuulua aivan luonnollisesti useampaan kategoriaan, jotka eivät ole samassa hierarkian haarassa. Aineistonhallintajärjestelmän tulisikin siis tarjota useita näkymiä samaan aineistoon eri käyttötarkoituksia ja -tilanteita varten ja mahdollistaa tietyn aineiston sijoittaminen useampaan kategoriaan.

Käytännössä aineistonhallintajärjestelmissä aineiston luokittelu tapahtuu usein liittämällä aineistoelementtiin avainsanoja. Joissain ratkaisuissa aineistoelementti liitetään puumaisen luokkahierarkian soluun. Luokittelu ei siis ole riippuvainen aineiston tallennusjärjestelyistä, joskin tallennusjärjestelyt, kuten hakemistorakenne, voidaan ottaa huomioon luokittelussa. (Nurmi ym. 2001.)

3.11.3 Dokumenttien etsiminen ja käyttöönotto

Tekstimuotoisen tiedon hakeminen aineistonhallintajärjestelmästä on suhteellisen helppoa. Kuvien, äänen ja videon hakeminen on puolestaan ongelmallisempaa. Useimmiten näitä mediatyyppejä haettaessa käytetään aineistoon liitettyä metatietoa. (Nurmi ym. 2001.)

Useissa järjestelmissä kuvista tehdään luetteloa varten pienet sormenpääkuvat (thumbnail) tai kuvakkeet ja aineistoa haetaan sormenpääkuvien luetteloa selaamalla. Luettelon selaaminen on nopea tapa löytää tietty kuva. Tämä ratkaisu toimii kuville, koska varsin pieneenkin esikatselukuvaan saadaan riittävästi tietoa kuvan tunnistamiseksi.

Video- ja äänitiedostoille kuvake-esitys ei ole hyvä ratkaisu. Yksi ruutu (frame) videosta ei useimmiten ole riittävän kuvaava otos videon sisällöstä sen tunnistamiseksi. Varsinkin videon ensimmäisen ruudun käyttäminen on vaikeaa, koska se on usein joko tyhjä tai sisältää vain tekijänoikeustietoja, jotka ovat sormenpääkoossa liian epäselviä luettavaksi. Äänitiedostolle on hyvin vaikea keksiä mielekästä kuvaketta, joka kuvaisi sen sisältöä. Useimmat järjestelmät näyttävät ääni- ja videotiedostoille vain järjestelmän kyseiselle tiedosto tyypille käyttämän normaalin kuvakkeen. (Nurmi ym. 2001.)

3.11.4 Käyttöoikeuksien hallinta

Käyttäjien oikeuksien määrittely ja hallinta on keskeinen osa useamman käyttäjän järjestelmän ylläpitoa. Tyypillisesti käyttäjienhallinta tarkoittaa aineiston tai metatiedon selaamiseen, lisäämiseen tai muokkaamiseen oikeutettujen käyttäjien määrittelyä.

3.11.5 Dokumenttien versiohallinta ja seuranta

Monen käyttäjän järjestelmissä versionhallinnalla pyritään estämään usean henkilön samanaikaiset muutokset samaan aineistoelementtiin. Versionhallinta saattaa myös pitää kirjaa aineiston versioista ja mahdollistaa aikaisempien versioiden käytön tai pitää kirjaa elementtien välisistä suhteista kuten siitä, perustuuko elementti johonkin toiseen aineistoelementtiin tai onko kyseessä vaihtoehtoinen versio. (Nurmi ym. 2001.)

4 Case: Hansaprintin aineistonhallintaratkaisu vuonna 2002: HANSABASE Mediapankki

Tässä luvussa esitellään HANSABASE-sovellus ja Hansaprint Oy vuosien 2001-2002 tilanteen mukaan. Nämä määrittelyt toimivat pohjana myöhemmin luvussa viisi esitettävälle HANSABASE -järjestelmän vanhenemisen analyysille.

Kuva 4.1. HANSABASE esitteitä eri vuosilta sekä käyttöohjeiden kansia.

4.1 Hansaprintin esittely

HANSAPRINT Oy[3] oli vuonna 2001 Suomen suurin painotalo, joka tarjosi asiakkailleen myös jatkuvasti monipuolistuvia digitaalisia lisäpalveluja. Hansaprintin liikevaihto vuonna 2000 oli 880 Mmk. Tuolloin arvioitiin, että vuonna 2001 Hansaprint tulisi ylittämään yhden miljardin Suomen markan rajan.

Hansaprintillä oli tuolloin viisi tehdasta; kolme tuotantoyksikköä Turussa, yksi Salossa ja yksi Vantaalla. Hansaprintin omistivat tuolloin TS-Yhtymä (60%) ja Helsinki Media (40%).

[3] http://www.hansaprint.fi

Hansaprintin tuoteryhmät olivat lehdet, luettelotuotteet, markkinointiesitteet, tekniset tuotemanuaalit, suoramarkkinointituotteet sekä ns. Print+ -palvelut, joihin kuuluvat sivunvalmistuksen lisäksi HANSABASE Mediapankki sekä HANSANET –palvelu, joka on aineistonsiirron hallintajärjestelmä. HANSABINDilla tarkoitetaan stiftauksen, Inkjet –osoitteistuksen, logistiikka- ja postituspalveluiden lisäksi monipuolisia selektiivisiä sidonta- ja liitteistysmahdollisuuksia. Myös näillä osa-alueilla Hansaprint oli uranuurtaja pohjoismaissa. Print+ -palveluihin kuuluivat myös graafinen suunnittelu ja puhelinluetteloiden tietokantapalvelut.

4.2 Aineistonhallinnan merkitys Hansaprintille

Aineistonhallinnalla oli tuolloin merkitystä Hansaprintille kolmellakin eri tavalla. Ensinnäkin suurena painotalona sillä oli luonnollisesti isot tarpeensa tuotannollisen digitaalisen aineistonsa hallintaan, toiseksi Hansaprintillä oli tarve oman brändinsä ja graafisen ilmeensä hallintaan ja kolmanneksi Hansaprint oli Print+ -palvelunaan kehittänyt asiakkailleen aineistonhallinnan palveluja. Hansaprintin saavutettua useiden tärkeiden painoasiakkaidensa luottamuksen aineistonhallintapalvelujen tarjoajana oli Print+ -palvelusta kehittynyt strategisesti erittäin merkittävää osa liiketoimintaa, ei kuitenkaan vielä liikevaihdollisesti.

Vuonna 2000 kartoitin Hansaprintille aineistonhallinnan merkitystä, ja tein silloin mm. seuraavia tärkeitä huomioita:

- Ainoa tapa, jolla graafisella alalla voitiin menestyä jatkossa, oli se, että yrityksellä oli kaikki Asset Managementiin liittyvät asiat kunnossa.
- Aineistonhallintaratkaisuilla saatiin merkittävimmät asiakkaat sitoutettua käyttämään samaa yritystä.
- Kaikkien taltioitujen tiedostojen joustava käyttö oli asiakkaalle avainasia.
- Työkohtaisen aineiston vastaanotto ja hallinta olivat erityisen tärkeitä.
- Pelkkien kuvapankkijärjestelmien tarjoaminsessa ei sinänsä ollut mitään uutta, sillä niitä oli saatavissa tuolloinkin, vaikka suoraan kaupasta hankittavissa. Järjestelmän pystyttäminen ei välttämättä vaatinut suuria investointeja, eikä taitoja (valmiita tuotteita mediapankin pystyttämiselle löytyi jo useitakin).
- Uusi ja mielenkiintoinen tapa lähestyä aineiston hallintaa oli käsitellä sitä projektiluonteisesti, jolloin tiettyä projektia varten luotiin oma www-alueensa (vrt. Hansaprintin tuolloinen Hanska-palvelu -> www.hanska.fi).
- Järjestelmillä tuli voida hallita kuva-aineistojen lisäksi myös muita tiedostomuotoja kuten ääni- ja videotiedostoja.
- Lähes kaikki järjestelmät toimivat webin kautta ja kaikissa oli ostoskorominaisuus.
- Useimmissa oli mahdollisuus räätälöidä käyttöliittymää.

- Jos kehitys jatkuisi entiseen tapaan, jokainen itseään kunnioittava painotalo tarjoaisi kuvapankkipalveluja, joten HANSABASE Mediapankkiin pitäisi kehittää joitain muista oleellisesti poikkeavia ominaisuuksia.
  (Boucht 2000)

Kuva 4.2. HANSABASE -esitteiden kansia vuosilta 1998, 2001 ja 2004 (Hansaprint HB 1998, 2001, 2004).

4.3 Hansaprintin aineistonhallintaratkaisu: HANSABASE Mediapankki

HANSABASE Mediapankki oli yrityskohtainen digitaalisen media-aineiston hallintajärjestelmä, joka helpotti ja nopeutti digitaalisen median kanssa työskentelyä.

Palvelu oli tarkoitettu ensisijaisesti digitaalisen kuva-aineiston hallinta- ja jakelujärjestelmäksi, ja lisäksi valmiiden dokumenttimuotoisten aineistojen ja multimedian tallentamiseen keskitettyyn tietokantaan.

HANSABASE Mediapankki tarjosi ratkaisun useissa eri paikoissa (mainostoimistot, reprot, yrityksen eri toimipisteet) sijaitsevan kuva- ja media-aineiston hallintaan, arkistointiin ja jakeluun. Aineiston hallinta, haku ja jakelu tapahtuivat World-Wide Web- selaimella yleisen Internet- verkon kautta, joten tietokannan omistajalla oli mahdollisuus antaa HANSABASE Mediapankki- järjestelmän käyttöoikeudet vaikkapa mainostoimistolleen, joka nouti järjestelmästä haluamansa kuvat esimerkiksi esitteen taittosuunnittelua varten.

Hansaprintin HANSABASE Mediapankki –ratkaisu perustui Grafimedian Maria/MediaVu aineistonhallintaratkaisuun, mutta oli Hansaprintin omista tarpeista johtuen kehittynyt omaan

suuntaansa. Hansaprint oli tehnyt omaa kehitystyötään pitkäjänteisesti, jolloin lähtökohtana oli palvelua kehitettäessä aina ollut tuottaa asiakkaalle ominaisia ratkaisuja, mikä tässä ei tarkoita pelkästään asiakkaan graafisen ilmeen toteuttamista palveluun. Huolimatta siitä, että palvelut pitivät sisällään standardiominaisuuksia, oli ne sovitettu yhdessä asiakkaan kanssa mahdollisimman hyvin heidän tarpeisiinsa ja työnkulkuihinsa. Tämän lisäksi asiakkaille oli toteutettu vain heidän omassa käytössään olevia erikoispalveluja. Kokemuksemme mukaan vain räätälöinnillä päästiin asiakkaan kannalta parhaimpaan tulokseen.

HANSABASE Mediapankki -palvelu oli ensimmäisiä suomalaisia internetin yli toimineita ASP –palveluja. Mediapankin palvelimet olivat aina sijainneet Hansaprintin tiloissa, ja olleet Hansaprintin hallinnassa, huollossa ja ylläpidossa, ja asiakkaat olivat ikään kuin vuokranneet palvelua internetin ylitse käyttöönsä. Hansaprint takasi sopimuksissaan asiakkailleen riittävän palvelutason. HANSABASE Mediapankilla oli vuonna 2002 n. 8000 käyttäjää, joista melkoinen osa oli ulkomailla.

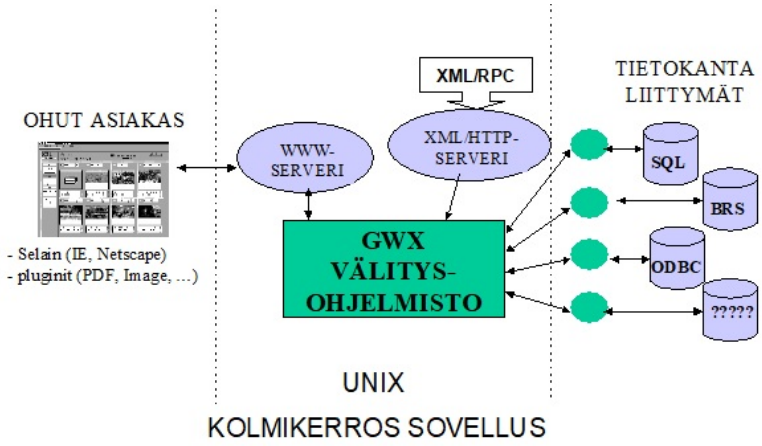
4.4 Teknologia

Tässä kappaleessa kuvataan HANSABASE Mediapankin teknistä ratkaisua sellaisena kuin se oli vuonna 2002. Kappaletta kirjoitettaessa on lähteinä käytetty seuraavia dokumentteja: Leinonen Heikki, Maria/MediaVu –tuotekuvaus (Leinonen 2000), Leinonen Heikki, MediaVu – Materiaalinhallintajärjestelmä (Leinonen 2001), Boucht Joon, HANSABASE Mediapankki: palvelukuvaus, v.1.3 (Boucht 2001).

4.4.1 Arkkitehtuuri

HANSABASE Mediapankki -ratkaisu perustuu Grafimedian MediaVu/Maria alustaan.

Ratkaisu rakennettiin Grafimedian kehittämän GWX –tuotearkkitehtuurin (General Web eXtension) päälle. GWX oli sovellusalusta, jonka avulla voitiin rakentaa sovelluksia hajautettuun verkkoympäristöön useille eri laitealustoille ja tietokantaresursseille. GWX käytti luokkaperustaista kehityskonseptia, jonka avulla voitiin hajauttaa ja jakaa toimintoja suoritettavaksi verkossa eri paikkoihin. Lisäksi GWX mahdollisti dynaamisten HTML-sivujen muodostamisen sekä tietokantakyselyiden tekemisen normaalin HTML-kielen seassa. (Leinonen 2000)



Kuva 4.3. Marian/MediaVU:n kolmikerros rakenne

4.4.2 Laitealusta

Järjestelmän palvelimena toimi vuonna 2002 Sun Solaris -palvelin. Työasemaksi kelpasi mikä tahansa laite, jossa toimi standardi selainohjelma (minimivaatimuksena oli Internet Explorer tai Netscape Navigator v.3.x, suositus v.4.x tai uudempi). Järjestelmän käyttöliittymä oli käyttöjärjestelmäriippumaton, joten käyttö oli mahdollista sekä Windows-, Macintosh- että UNIX-työasemalta.

4.4.3 Tietokanta ja hakukieli

Varsinaisena mediatietokantana toimi Open Textin BRS/Search täystekstitietokantaan, mutta lisäksi järjestelmään oli liitetty SQL Server- ja MiniSQL-tietokantoja erilaisten toimintojen hoitoon.

Täystekstitietokanta indeksoi kaiken Mariaan syötettävän tiedon, joka takasi aineistohauille erittäin lyhyen vasteajan materiaalin määrästä riippumatta. Myös tietokannan indeksi oli eräissä käyttöliittymissä selattavissa käyttöliittymän kautta. Indeksiin voitiin tehdä suoraan hakuja sanan alku-, keski- ja loppuosan perusteella. Järjestelmä sisälsi lisäksi suomen kielioppi- ja tavutusrakenteen tuntevan ohjelma-moduulin, joka helpotti huomattavasti hakujen tekemistä suomen kielen ongelmallisen taivutussäännöstön kanssa. Tällöin esim. hakusana "yö" löytää kortin, jonka tekstissä lukee "öinen". (Lingsoft Oy:n Twols/Finstems).

4.5 Ominaisuudet

Tässä kappaleessa kuvataan HANSABASE Mediabankin ominaisuuksia sellaisina kuin ne olivat vuonna 2001.

4.5.1 Rakenne

Marian perusideana oli tekstitietokantaan tallennettava materiaalikortti, jossa sanallisesti kuvatiin arkistoitu materiaali sekä teknisesti että sisällöllisesti. Kortteja ja materiaaleja voitiin linkittää toisiinsa monin eri tavoin. Korteista voitiin muodostaa sarjoja tai paketteja, erityyppisiä materiaaleja voitiin liittää toisiinsa, esim. tekstidokumentti ja siihen liittyvä kuva tai vaikkapa kaikki yhteen mainoskampanjaan liittyvät materiaalit (kuvat, tekstit, taittopohjat, radiomainokset ja videoklipit).

4.5.2 Tiedostomuodot

HANSABASE Mediapankki oli täysin avoin järjestelmä minkä tahansa digitaalisessa muodossa olevan materiaalin hallintaan ja arkistointiin. Järjestelmä pystyi useista tiedostoformaateista generoimaan automaattisesti selailuun tarkoitetun sormenpääkuvan sekä sitä hieman suuremman näyttökuvan. Näin tuettuja tiedostomuotoja olivat mm. kaikki tunnetuimmat kuva-formaatit ja PDF –tiedostot.

Harvinaisemmista formaateista (joista ei lisäoptiollakaan synny selailukuvaa automaattisesti) voitiin kuitenkin aina syöttää selailukuvat järjestelmään käsin.

Ns. taittotiedostojen osalta järjestelmään generoitiin manuaalisesti kaikki taittotyöhön liittyvät tiedostot sisältävä tiedostopakkaus, ja generoitiin näyttökuvaksi taiton, esim. esitteen PDF –versio.

4.5.3 Käyttäjäoikeudet

Järjestelmän käyttäjät ryhmiteltiin käyttäjäryhmiin, joille myönnettiin työnkuvaa vastaavat käyttöoikeudet kuhunkin tietokantaan. Tarpeen mukaan tietokannoista voitiin muodostaa erilaisia yhteenliittymiä ja "näkymiä", jolloin käyttäjän ei tarvinnut tietää, mihin tietokantaan hänen etsimänsä materiaali kuului. Lisäksi oli mahdollista liittää käyttäjäryhmäkohtaisia oikeuksia esim. tiettyihin rajoitettuihin kategorioihin.

4.5.4 Tietokantahaut

Tietokannan hakukieli sisälsi kaikki normaalit Boolean –operaattorit (AND, OR, NOT, XOR) sekä tekstin rakenteeseen liittyvät ehdot NEAR (lähekkäin), ADJ (lähekkäin ja annetussa järjestyksessä), SAME (samassa kappaleessa) ja WITH (samassa lauseessa). Suomen kielen tuki toi lisää hakumahdollisuuksia, mm. hakusanan perusmuodon tai taivutetun muodon sekä yhdyssanan alku- tai loppuosan perusteella.

HANSABASE Mediapankki -ratkaisuissa oli käyttöliittymät pyritty toteuttamaan aina käyttäjille sovitetusti, siten, että kaikki käyttäjäryhmälle turhat toiminteet oli piilotettu, eikä erillisen hakukielen käyttö ollut välttämätöntä.

HANSABASE Mediapankista aineistoja voitiin etsiä usealla eri tavalla. Yksinkertaisin hakutapa oli materiaalin selailu. Se voitiin tehdä käyttäjäliittymästä riippuen kymmenen (tai yhdeksän) sormenpään, 150 (tai 200) sormenpään tai 150 (tai 200) tekstirivin muodossa. Tämä oli tietenkin aivan liian hidasta satojen tai tuhansien dokumenttien ollessa kyseessä.

Toinen hakutapa oli vapaatekstihaun käyttö. Kaikki dokumenttien infokorteilla oleva tieto oli indeksoitu tehokkaasti tekstitietokantaan, jonka ansiosta järjestelmä pystyi löytämään sekunneissa kaikki dokumentit, joista löytyi annettu yhdistelmä hakusanoja. Myös jokerimerkinkäyttö oli mahdollista. Näinkin saattoi haluttujen dokumenttien löytäminen olla hankaa, sillä käyttäjän tuli pystyä arvaamaan mitä hakusanoja korteille on syötetty.

Kolmas hakutapa oli hakukortin käyttö, jolla haku voitiin kohdistaa haluttuun kenttään infokortilla. Infokortin numeerisiin kenttiin (päivämäärät, tiedostokoot jne.) voitiin kohdistaa myös numeerisia vertailuja. Tällöin saatiin esim. etsittyä materiaalit, jotka oli tallennettu ennen tiettyä päivämäärää.

Neljäs hakutapa perustui ennalta määriteltyihin kategorioihin, eli aineistoluokkiin. Luokkahierarkia voi olla monitasoinen (jopa kuuden tason kategorioita oli käytössä). Kukin dokumentti voi kuulua niin moneen eri kategoriaan kuin oli tarpeen. Käytettäessä kategoriahakua hakija käytti järjestelmävalvojien ennalta määrittelemiä hakusanoja, joten todennäköisyys sopivan hakusanan käyttämiselle ja haluttujen dokumenttien löytämiselle kasvoi.

Viides ja monipuolisin hakutapa oli yhdistää aiemmin tässä lueteltuja hakutapoja käyttäjän haluamalla tavalla.

Hakutuloksen muodon voi käyttäjä valita käyttöliittymään toteutetuista vaihtoehdoista. Uusimpana mahdollisuutena oli järjestää hakutulos myös ennalta määrättyjen tekijöiden tai kenttien mukaan.

4.5.5 Materiaalin syöttö

Materiaalia voidaan syöttää järjestelmään kahdella eri tavalla, joko yksittäin käyttöliittymän kautta tai massasiirtoina taustalla hyödyntäen ns. Autoload –ominaisuutta. Yksittäissyötössä täytetään aluksi uuden, tyhjän kortin tiedot ja liitetään tämän jälkeen korttiin työaseman levyllä oleva materiaalitiedosto.

Autoload -tapauksessa arkistoitavat materiaalit kopioitiin sovittuun palvelimen lataushakemistoon, josta tausta-ajo siirsi ne automaattisesti järjestelmään. Tällöin kunkin materiaalin kortti täytettiin jälkikäteen. Määrättyjen kenttien arvot oli mahdollista välittää tietokantaan tässä yhteydessä tiedostoon liitettävällä metadata-tiedostolla.

Asiakkaan kuva-aineistoon tallennettu IPTC (International Press and Telecommunication Council) - standardin mukainen saatetieto voitiin automaattisesti konvertoida HANSABASE –infokortille. IPTC –muotoa käytettiin yleisesti mm. lehtikuvissa ja sitä pystyttiin hyödyntämään esim. Photoshop –ohjelmassa.

Materiaalin uudelleenkäytön helpottamiseksi järjestelmä generoi automaattisesti aineiston sisään syötön yhteydessä kuvamateriaalista sovitut kuvamuodot. Nämä muodot olivat asiakkaittain valittavissa. Seuraavassa on lueteltu yleisimmät Mediapankki –toteutuksissa käytössä olleet muodot:

1. Sormenpää –näyttökuva, 120x120 pikseliä.
   Tätä tiedostoa käytetään sormenpääselailussa, ja se on käytettävissä myös suoraan esim. asiakkaan www-sivuilla.
2. Ns. iso sormenpää tai web-muotoinen kuva, 512x512 pikseliä.
   Tätä tiedostoa käytetiin infokortilla, ja se oli käytettävissä myös suoraan esim. asiakkaan www-sivuilla tai PowerPoint –esityksissä.
3. Ns. layout-, leiska- tai asemointikuvan, 72 DPI.
   Tämä tiedosto oli tarkoitettu taitotyössä käytettäväksi (, myös ns. OPI -muoto oli mahdollinen). Kuvamuodossa oli säilytetty originaalin senttimetrikoko, mutta käytettyjä värejä sekä resoluutiota oli pudotettu voimakkaasti.

Näiden lisäksi järjestelmä tallensi alkuperäisen kuvan sellaisenaan tietokantaan (eräissä tapauksissa pakaten).

4.5.6 Materiaalin valinta ja käyttöönotto

Aineistoja voitiin ottaa käyttöön kahdella eri tavalla. Ensimmäinen tapa oli ladata työasemaan halutun tiedoston haluttu muoto suoraan tiedoston infokortilla olevaa nappulaa painamalla.

Toinen, monipuolisempi tapa oli käyttää järjestelmän tilausmekanismia. Selailulistasta poimittiin rastittamalla etsityt materiaalit ns. koriin. Koriin voitiin poimia materiaaleja useasta eri hausta. Poimintojen jälkeen valittiin ns. tilaus-näyttö, jolta valittiin haluttu toimenpide käyttäjän profiilin mukaisista vaihtoehdoista. Profiilista riippuen korin sisältö voitiin tilata esim. CD-rommille poltettuna, ftp-siirtona, sähköpostin liitteenä jne.

4.5.7 Käyttötuki

HANSABASE Mediapankki –palveluun kuului olennaisena osana HelpDesk –palvelu. Palvelu oli käyttäjien käytössä puhelimitse Suomen toimistoaikoina (arkisin 8-17), tai sähköpostitse ympärivuorokauden. Palvelukeskuksessa asiakkaiden kysymyksiin oli vastaamassa noin kymmenen spesialistia.

Lisäksi järjestelmä sisälsi online-käyttöohjeen, jota ylläpidetiin järjestelmän kehittyessä. Käyttöohjeesta oli myös mahdollista tehdä painettu versio erikseen näin sovittaessa. Järjestelmän käyttöönottoon liittyen järjestettiin asiakkaan osoittamalle kohderyhmälle käyttäjäkoulutusta.

4.5.8 Järjestelmän ylläpito

Järjestelmän ylläpitäjälle järjestelmä tarjosi runsaasti toimintoja. Automaattipoistojen avulla voitiin määritellä tietyin aikavälein käynnistyvät poistoajot, jotka annettujen kriteerien perusteella siivosivat vanhentuneet kannasta yön aikana. Materiaalien siirto tietokannasta toiseen voitiin tehdä myös käsin käyttöliittymän kautta. Lähes kaikki järjestelmän ylläpitoon tarvittavat toimenpiteet voitiin tehdä käyttöliittymän eli selaimen kautta, mm. käyttäjien, käyttäjäryhmien sekä tietokantojen luominen, poistaminen ja muokkaus.

4.5.9 Raportointi

Järjestelmän perusversio sisälsi käytön kuukausittaisen perusraportoinnin. Raportti kertoi järjestelmän käytöstä esim. seuraavia asioita:

1. raportointikuukauden käyttökerrat.
2. käyttäjätunnuskohtaisesti tehdyt tiedostojen lataukset ja tilaukset.
3. kaikki tehdyt tilaukset aikajärjestyksessä.
4. käyttäjätunnusten määrän (kumuloituna).
5. tilaukset ja aineiston lataamiset kuukausitasolla koko käytön ajalta.

Perusraportointia oli myös mahdollista kehittää asiakkaan haluamaan suuntaan.

Lisäksi järjestelmä kertoi kunkin dokumentin osalta sen eri muotojen lataamiskerrat niitä erikseen kysyttäessä.

4.5.10 Tietoturva

Tietoturvasta HANSABASE Mediapankissa oli huolehdittu neljällä tasolla: tietokannan käyttäjätunnuksilla, salasanoilla ja käyttöoikeuksilla, käyttöjärjestelmän käyttäjätunnuksilla ja salasanoilla, tiedonsiirron reitityksillä (vain HTTP ja mahdollisesti FTP sallittuja) sekä SSL -turvatulla www-palvelinliikenteellä. Oli myös mahdollista ottaa käyttöön ns. PGP-suojaus tiedonsiirrossa.

4.5.11 Lisäominaisuudet

Seuraavat ominaisuudet olivat järjestelmän optioita, joiden toteutuksesta neuvoteltiin asiakkaan kanssa aina erikseen. Ominaisuuksien toteuttaminen vaati aina tarkempia määrittelyjä.

Kuvien julkaisu –toiminne

Toiminne mahdollisti Mediapankissa arkistoituina olevien materiaalien julkaisemisen, eli tällä toiminteella luotiin kannan ulkopuolelle näkyviä linkkejä haluttuihin tiedostoihin. Tämä mahdollisti HANSABASE Mediapankissa olevan materiaalin käytön esimerkiksi suoraan yrityksen www –sivuston kautta.

Palvelua pystyttiin jatkojalostamaan tuottamaan automaattisesti ennalta sovitun mukaisia HTML –sivuja sovittuun paikkaan (esim. asiakkaan internet-sivuilla olevat Press-sivut).

PressiCD –toiminne

Toiminne mahdollisti PressiCD aineiston ylläpitämisen Mediapankissa sekä PressiCD tilausten tekemisen suoraan palvelusta.

Ominaisuus sisälsi tavan hallita CD:lle tuleva aineisto ja CD:n indeksisivulle tulevat tekstit sekä kansien ja indeksin painatusmahdollisuuden, CD:n etiketin tulostamismahdollisuuden ja mahdollisuuden tehdä CD –tilaus suoraan palvelusta.

Työnkulun hallinta

Työnkulun hallinta oli tarkoitettu esimerkiksi painotuotteiden (markkinointiaineistot, esitteet jne.) suunnittelun ja valmistusprosessin hallintaan. Tuotantoprosessit (työvaiheketjut ja resurssit) määriteltiin työkohtaisesti. Työvaiheketju saattoi sisältää myös järjestelmän ulkopuolisia työvaiheita (esim. käännöstyö alihankkijalla). Järjestelmä huolehti materiaaleista, työvaiheiden oikeasta suoritusjärjestyksestä ja aikatauluvalvonnasta.

Yhteydet ulkopuolisiin järjestelmiin (XML –rajapinta)

Asiakkaan omien järjestelmien ja HANSABASE Mediapankki –järjestelmän välinen kommunikaatio voitiin rakentaa HANSABASE Mediapankkiin toteutetun XML –rajapinnan avulla. Esim. digitaalisen aineiston tuonti ja vienti Mediapankin ja asiakkaan järjestelmän välillä pystyttiin toteuttamaan XML –rajapintaa hyödyntäen.

4.6 Emojärjestelmä: Grafimedian MediaVu (os. Maria)

Seuraavassa esitellään HANSABASE-järjestelmän emojärjestelmä MediaVu sellaisena kuin se oli vuonna 2002.

4.6.1 Esittely

MediaVu eli Maria oli Grafimedian aineistojenhallinta- ja arkistointijärjestelmä. MediaVu oli käytössä monenlaisissa arkistointisovelluksissa. Lehtitaloista se oli käytössä mm. Helsingin Sanomissa, jossa kuvien lisäksi arkistoitiin lehden sivut ja juttujen leikkeet PDF –muodossa. Sitä käytettiin myös museoissa, esimerkiksi Koulumuseossa ja Helsingin kaupunginmuseossa. Muita käyttäjiä olivat mm. YLE, MTV3, Ruutunelonen, Suomen Keltaiset Sivut, Credonet ja Publicis Marché (entinen Keskon MK-Mainos). (Nurmi 2001)

4.6.2 Historiaa

Turun Sanomat (allekirjoittanut projektin projektipäällikkönä) oli aikanaan vuosia mukana kehittämässä Suomen Keltaisten Sivujen (SKS), Nokian ja Grafimedian kanssa puhelinluettelon Keltaisten Sivujen internet-palvelua, Interaktiiviset Keltaiset Sivut. Osittain tästä projektista poiki Grafimedialle Maria –järjestelmän www-kehitysalusta mariaw3.

Grafimedian vielä kehitettyä kuvapankki- ja aineistonhallintakonseptiaan eteenpäin, lähti Hansaprint siltä pohjalta tuottamaan aineistonhallintapalvelua omaan ja asiakkaidensa käyttöön. Ensimmäisestä versiosta lähtien Hansaprintin oma henkilökunta oli tehnyt kaikki HANSABASE Mediapankki -palvelun sovitukset ja räätälöinnit sekä liitännät erilaisiin tuotantojärjestelmiin. Alkuperäisen järjestelmän käyttöliittymä, joka oli Grafimedian kehittämänä varsin tehokas ja monipuolinen, luokiteltiin Hansaprintin testikäyttäjien kanssa ns. insinöörikäyttöliittymäksi, koska se osoittautui varsin vaikeaksi ns. tavallisen aineistoja tarvitsevan käyttäjän käyttöön. Näiden kokemusten pohjalta Hansaprint lähti jo projektin alussa kehittämään omien käyttäjiensä kanssa varsinaisesti ”peruskäyttäjälle” räätälöityä käyttöliittymää.

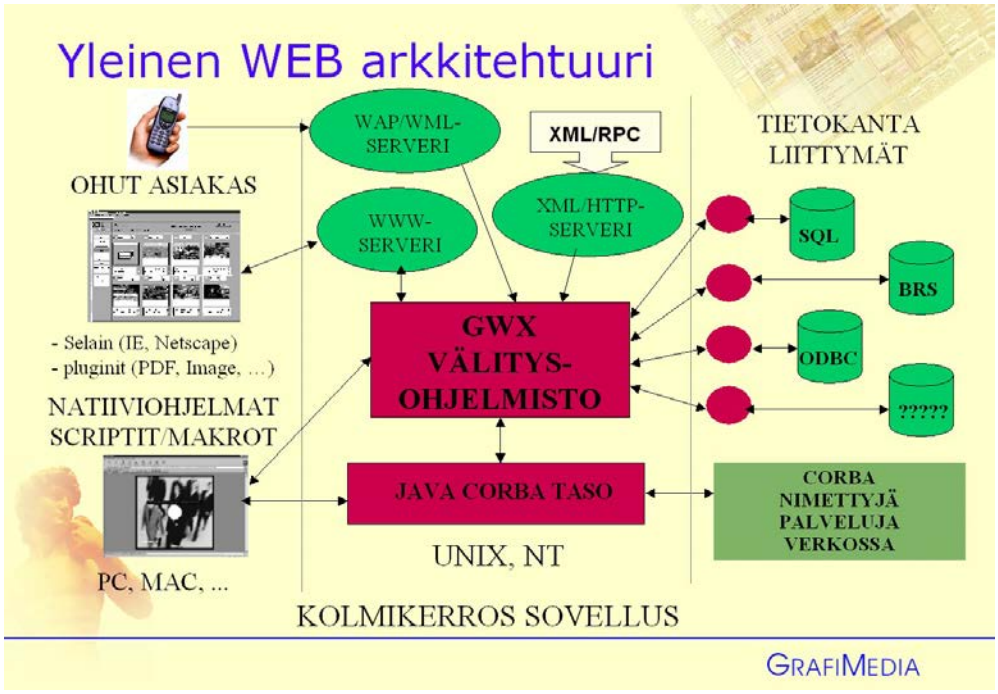
HANSABASE Mediapankki sai ensimmäisiksi asiakkaikseen mm. Finnet Yhtiöt, TeamWAREn ja Nokia Yhtymän. Hansaprint toteutti vaativille asiakkailleen järjestelmään näiden toivomia lisäyksiä ja räätälöintejä, mikä johti kehitystä omaan HANSABASE -suuntaansa. Toisaalla Grafimedia kehitti Maria/MediaVu tuotetta muiden asiakkaidensa kanssa.

Luonnollisesti tiedot järjestelmän kehityksestä pidettiin ajan tasalla, mutta varsinaisia järjestelmien sovituksia ei tuolloin matkan varrella erikseen tehty. Kun sitten Grafimedian järjestelmäarkkitehdit olivat tahollaan päässeet niin pitkälle seuraavan sukupolven työkalussa, että Hansaprint alkoi olla ainoita vanhan (mariaw3) ympäristön käyttäjiä, oli Hansaprintinkin vihdoin tehtävä päivitys ympäristöönsä. Varsinkin kun samalla tulivat esiin Y2K -vaatimukset. Toteutettu päivitys oli perusteellisesti muuttuneen kehitysympäristön takia varsin työläs ja vei ennalta oletettua huomattavasti pidemmän ajan, varsinkin kun samaan aikaan vei aikansa uusien asiakkaiden palvelujen ja ns. vanhojen asiakkaiden aina kiireellisten uusien tarpeiden toteuttaminen. Kuitenkin päivitys vihdoin saatiin tehtyä.

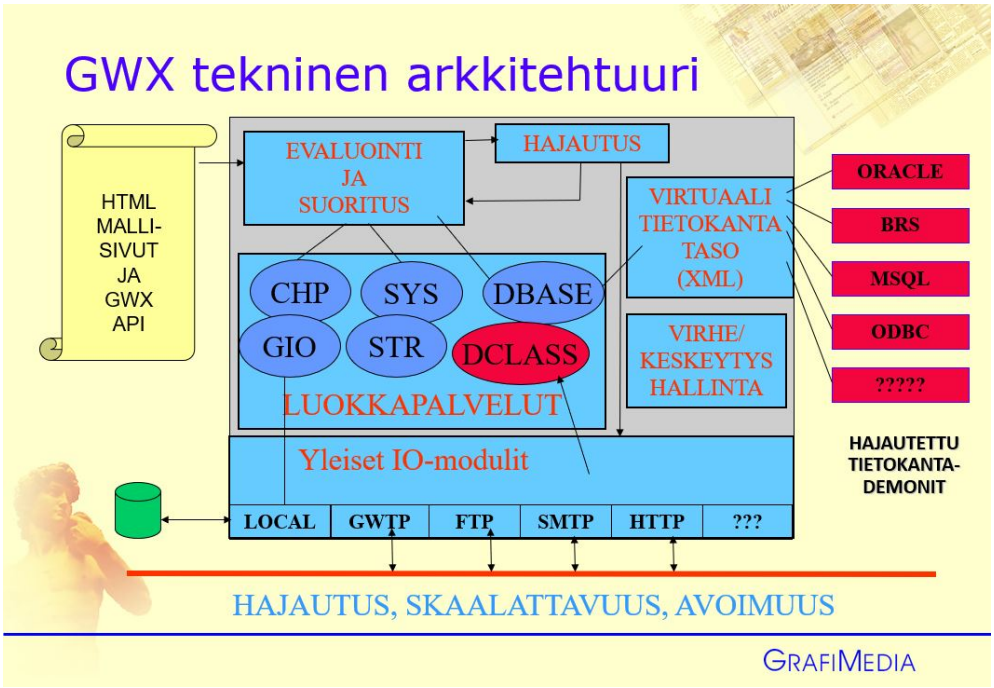
Tämän päivityksen jälkeen Hansaprint oli edelleen kehittänyt voimakkaasti HANSABASE Mediapankki palvelua tekemällä laajennuksia omaa ympäristöönsä omiin ja asiakkaidensa tarpeisiin. Tämä kehitys oli onnistunutta ja moni hieno uusia ominaisuus näki päivän valon.

Samaan aikaan Grafimedia oli jatkanut omaa vankkaa kehitystään niin järjestelmän ominaisuuksien kuin teknologiankin kanssa.

Seuraavassa kaksi kuvaa MediaVU ratkaisun rakenteesta vuodelta 2002. Kuva 4.4. kuvaa Yleisen WEB arkkitehtuurin kolmikerrosrakenteen. Ylin sovellustaso on käyttöliittymätaso, keskimmäinen ns. välittäjätaso, ja kolmas ns. palvelu- ja tietokantataso. Kuva 4.5. esittelee Maria/MediaVun GWX arkkitehtuurin periaatteet.



Kuva 4.4 Maria/MediaVu Yleinen WEB Arkkitehtuuri (Leinonen 2001)



Kuva 4.5 Maria/MediaVu GWX tekninen arkkitehtuuri (Leinonen 2001)

4.6.3 Ajatuksia HANSABASE Mediapankin kehityksestä vuonna 2002

Vuonna 2002 oltiin siinä tilanteessa, että piti vakavasti miettiä HANSABASE-järjestelmän kehittämistä. Seuraavassa on lyhyesti käyty läpi tuolloin kerättyjä ajatuksia.

Ensin mietittiin syitä, miksi uutta versiota HANSABASE Mediapankkista oikein tarvittiinkaan. Seuraavat kolme tärkeintä syytä kirjattiin:

1. Asiakkaiden kasvavat tarpeet
2. Grafimedian jo tehdyn kehityksen hyödyntäminen
3. Nykyisen ylläpidon raskaus, resurssien vähyys -> keskittyminen oleelliseen

Jotain olisi siis tehtävä.

Sitten harkittiin monesta näkökulmasta, miten järjestelmän uusiminen ominaisuuksien kannalta pitäisi tehdä. Erilaisista vaihtoehdoista listattiin seuraavat kolme:

1. kokonaan uusi HANSABASE Mediapankki –ratkaisu perustuen johonkin muuhun järjestelmään.
2. kerralla uusi HANSABASE Mediapankki –ratkaisu perustuen tulevaan MediaVuhun ja nykyiseen HANSABASE Mediapankkiin
3. asteittain uusittu HANSABASE Mediapankki, tuoden palveluun mukaan uusia MediaVu ominaisuuksia pikku hiljaa ja tarpeen mukaan

Näistä tuolloin päädyttiin kolmanteen vaihtoehtoon, eli pikku hiljaa uusittuun Mediapankkiin.

Lisäksi kirjattiin muita teknisiä kehitystarpeita, kun HANSABASE -palvelun toimintaa haluttiin laajentaa asiakasyritysten, käyttäjien ja tietokantoihin tallennetun materiaalin osalta. Tällaisia olivat:

1. Uusi levyjärjestelmä
2. Uusi varmistettu palvelin
3. Hajautus palvelimille -> kuormituksen jako / load balancing.
4. Tietoliikenneyhteyksien kehittäminen

Näiden teknisten kehityskohteiden osalta kuitenkin päädyttiin vielä odottamaan sopivampaa investointiaikaa.

MediaVu 2002 uudet ominaisuudet vs. HANSABASE Mediapankki

Koska HANSABASE Mediapankki -palvelu perustui MediaVun työkaluihin ja ympäristöön, ja esittelin HANSABASE Mediapankin juuri edellä, ei MediaVun ominaisuuksia esitellä tässä kohtaa sen tarkemmin, vaan listataan eroja sen ja HANSABASE Mediapankki –ratkaisun välillä vuonna 2002 sekä tuolloin tehty arvio ominaisuuden hyödyllisyydestä. Tavoitteena tuolloin oli kartoittaa miten HANSABASE-järjestelmää tulisi kehittää, ja miten ja mitkä MediaVu ominaisuudet olisi hyvä ottaa käyttöön HANSABASE Mediapankissa. Tähän on otettu kantaa alla olevassa listauksessa.

MediaVu:n ominaisuudet, jotka poikkesivat HANSABASE –ratkaisusta

Ero 1	**Aineistosarjat ja paketit**
Luokka:	Rakenne
Ominaisuus:	Kortteja ja materiaaleja voidaan linkittää toisiinsa monin eri tavoin. Korteista voidaan muodostaa sarjoja tai paketteja, eri tyyppisiä materiaaleja voidaan liittää toisiinsa, esim. tekstidokumentti ja siihen liittyvä kuva tai vaikkapa kaikki yhteen mainoskampanjaan liittyvät materiaalit (kuvat, tekstit, taittopohjat, radiomainokset ja videoklipit).
HANSABASE:	Tämä on HANSABASE Mediapankissa toteutettu pakkaamalla samaan kokonaisuuteen kuuluvat aineistot yhteen esim. ZIP tai SIT -pakkaukseen.
Arvio tarpeesta 2002:	Sarjat ja paketit kannattanee ottaa käyttöön myös HANSABASE Mediapankissa
Ero 2	**Sekä Web-pohjaiset, että natiivit käyttöliittymät**
Luokka:	Käyttöliittymä: Erilaiset ratkaisut
Ominaisuus:	MediaVu -ympäristöön on toteutettu sekä web-pohjaisia käyttöliittymiä, että erillisiä käyttöliittymiä natiiveina asiakasohjelmina
HANSABASE:	HANSABASE Mediapankin käyttö on mahdollista vain web-pohjaisella käyttöliittymällä.
Arvio tarpeesta 2002:	Tarvetta natiiviin asiakasohjelmaan ei ole tiedossa.
Ero 3	**MediaVu:n ns. insinöörikäyttöliittymä**
Luokka:	Käyttöliittymä: Erilaiset ratkaisut
Ominaisuus:	MediaVu standardi web-käyttöliittymä on erittäin monipuolinen, ns. insinöörikäyttöliittymä,
HANSABASE:	HANSABASE Mediapankin käyttöliittymä on optimoitu ns. peruskäyttäjälle.
Arvio tarpeesta 2002:	Näitä ominaisuuksia voisi olla HANSABASE Mediapankin ns. Advanced Users -käyttöliittymässä

Ero 4	**Ylläpitäjän ja arkistonhoitajan työkalut**
Luokka:	Järjestelmän ylläpito
Ominaisuus:	Marian ylläpitäjälle ja arkistonhoitajalle järjestelmä tarjoaa runsaasti toimintoja
HANSABASE:	On HANSABASE Mediapankissa, mutta eri tekniikalla
Arvio tarpeesta 2002:	Mietitään kannattaako tekniikat yhdistää vaiko pysyä omassa. Lisäksi mietitään admin-työkalun tekemistä asiakkaiden admin-käyttäjille.

Ero 5	**Materiaalien siirto manuaalisesti tai tausta-ajona**
Luokka:	Järjestelmän ylläpito
Ominaisuus:	Materiaalien siirto tietokannasta toiseen voidaan tehdä joko käsin käyttöliittymän kautta tai automaattisesti tausta-ajoina annetuilla kriteereillä.
HANSABASE:	Tämä mahdollista joissain kannoissa erikseen määrättyyn toiseen tietokantaan
Arvio tarpeesta 2002:	Kannattaa tutustua tekniikkaan ja harkita yleisemmän tavan käyttöönottoa

Ero 6	**Ylläpitotoimet selaimen kautta**
Luokka:	Järjestelmän ylläpito
Ominaisuus:	Lähes kaikki järjestelmän ylläpitoon tarvittavat toimenpiteet voidaan tehdä käyttöliittymän eli selaimen kautta, mm. korttien oletusarvojen muokkaus.
HANSABASE:	Myös HANSABASE Mediapankissa mahdollista muuten paitsi oletusarvojen osalta
Arvio tarpeesta 2002:	Kenttien oletusarvojen käyttöönottoa ja niiden editointimahdollisuuksien tuomista ylläpitäjän käyttöliittymään kannatta harkita

Ero 7	**Käyttäjien hallinta MediaVun tavalla**
Luokka:	Järjestelmän ylläpito
Ominaisuus:	Käyttäjien hallinta on toteutettu kummassakin ratkaisussa omalla tavallaan
HANSABASE:	toteutettu HANSABASE Mediapankissa omalla erittäin toimivalla tavallaan
Arvio tarpeesta 2002:	Selvitetään yhdistämistarve ja/tai mahdollisuudet.

Ero 8	**Eri palveluosien hajautus useille palvelimille**
Luokka:	Tekniikka: Arkkitehtuuri
Ominaisuus:	Eri palveluosien hajautus useille palvelimille
HANSABASE:	Meillä tänä päivänä www-palvelu pyörii samassa palvelimessa kuin BRS -tietokanta.
Arvio tarpeesta 2002:	Hajautus kannattanee lisätehon aikaansaamiseksi.

Ero 9	**Järjestelmän sovitus useammille alustoille (UNIX ja NT)**
Luokka:	Tekniikka: Laitealusta
Ominaisuus:	Järjestelmän palvelimena toimii joko UNIX- tai NT-pohjainen palvelin
HANSABASE:	Meillä tänä päivänä käytössä UNIX palvelimet.
Arvio tarpeesta 2002:	Kannattanee harkita hajautusta mietittäessä.

Ero 10	**Laajempi tuki tuetun median tiedostomuodoille**
Luokka:	Tekniikka: Tiedostomuodot
Ominaisuus:	Perusjärjestelmä tukee kaikkia tunnetuimpia kuvaformaatteja (mm. JPG, TIFF, EPS, BMP, GIF, Targa) sekä MPEG-1 -videoformaattia.
HANSABASE:	Myös erittäin laajasti HANSABASE Mediapankissa, paitsi MPEG-1 sormenpäiden osalta.
Arvio tarpeesta 2002:	Selvitetään tekniikka ja mietitään käyttöönottoa.

Ero 11	**Web-to-print ominaisuus**
Luokka:	Tekniikka: Lisäominaisuudet
Ominaisuus:	Mariaan on myös saatavissa erillinen PDF -mallinnukseen pohjautuva web-to-print –moduuli, joka soveltuu esim. käyntikorttien ja tuote-esitteiden selainpohjaiseen muokkaukseen.
HANSABASE:	Vastaavaa teknologiaa jo HANSABASE Mediapankissa sekä TS:n Verkkopalveluiden tuotteissa
Arvio tarpeesta 2002:	Kannattanee tutustua kuitenkin toteutukseen tarkemmin

MediaVu:n ominaisuudet, jotka puuttuivat HANSABASE Mediapankki –ratkaisusta

Ero 12	**Kortit ja materiaalit tyypin mukaan eri tietokantoihin**
Luokka:	Rakenne
Ominaisuus:	Kortit ja niihin liittyvät materiaalit jaetaan tyypin ja käyttötarkoituksen mukaan eri tietokantoihin.
HANSABASE:	Materiaalityyppeihin perustuvaa jakoa ei ole toteutettu HANSABASE Mediapankissa.
Arvio tarpeesta 2002:	Ei suoraa tarvetta tähän. Arvioida myöhemmin.

Ero 13	**Materiaalin versionhallinta**
Luokka:	Rakenne
Ominaisuus:	Mariaan kuuluu osana myös materiaalien versionhallinta, mikä mahdollistaa usean eri version tallentamisen samasta materiaalista. Eri versioita voidaan kuitenkin aina käsitellä myös yksittäisinä dokumentteina.
HANSABASE:	Ei vielä käytössä.
Arvio tarpeesta 2002:	Otetaan käyttöön.

Ero 14	**Näkymät yli usean tietokannan**
Luokka:	Rakenne
Ominaisuus:	Tarpeen mukaan tietokannoista voi muodostaa erilaisia yhteenliittymiä ja "näkymiä", jolloin käyttäjän ei tarvitse tietää, mihin tietokantaan hänen etsimänsä materiaali kuuluu.
HANSABASE:	Ei käytössä.
Arvio tarpeesta 2002:	Ei suoraa tarvetta tähän. Arvioida myöhemmin.

Ero 15	**Käyttöliittymän muokattavuus**
Luokka:	Käyttöliittymä: Muokattavuus
Ominaisuus:	Käyttäjän omassa profiilissa voidaan määritellä mm. käyttöliittymän kieli ja omat pikavalintanapit
HANSABASE:	Ei käytössä.
Arvio tarpeesta 2002:	Otetaan käyttöön profiileihin liittyen.

Ero 16	**Tiedot korteilta kuvan metadataksi**
Luokka:	Käyttöliittymä: Materiaalin syöttö
Ominaisuus:	Maria-materiaalikortilla olevat kenttätiedot voidaan siirtää haluttaessa myös itse kuvatiedostoon (esim. JPG-muotoiseen kuvaan).
HANSABASE:	Ei käytössä
Arvio tarpeesta 2002:	Mielenkiintoinen. Selvitetään käyttömahdollisuus

Ero 17	**Hakutoiminteeseen mahdollisuus kohdentaa edellisiin hakuihin.**
Luokka:	Materiaalin etsintä
Ominaisuus:	Maria kertoo haun jälkeen, kuinka monesta kortista ja kuinka monta kertaa käyttäjän antama hakusana esiintyy hakutuloksessa. Hakutulokset kerätään tuloslistaan, jossa vanhat haut ovat näkyvissä koko istunnon ajan. Tehtyihin hakuihin voidaan myöhemmin viitata uusia hakuja muodostettaessa ja käyttää niitä näin tarkentavien hakujen pohjana.
HANSABASE:	Ei käytössä
Arvio tarpeesta 2002:	Voisi olla myös HANSABASE Mediapankissa. Advanced Users -käyttöliittymässä

Ero 18	**Tekstimuotoisen aineiston selailu ja hakutulosten havainnointi.**
Luokka:	Materiaalin selailu
Ominaisuus:	Tekstimuotoista aineistoa selataan tekstilistalla, jossa näytetään tekstin otsikot ja osa leipätekstistä. Selailuikkunassa käyttäjän antamat hakusanat näytetään tekstin seassa korostettuna.
HANSABASE:	HANSABASE Mediapankissa ei tekstimuotoista aineistoa ole erikseen huomioitu.
Arvio tarpeesta 2002:	Harkitaan myöhemmin.

Ero 19	**Materiaalien keräily koriin, korin lataaminen ja lähetys toiselle käyttäjälle.**
Luokka:	Materiaalin valinta ja käyttöönotto
Ominaisuus:	Keräilykorissa oleva materiaali voidaan lähettää toiseen yrityksen sisällä tai talon ulkopuolella olevaan järjestelmään halutun henkilön tai ohjelmiston käytettäväksi (esim. toimitus- tai taittojärjestelmän kori tai autorepro-ohjelma).
HANSABASE:	Ei käytössä vielä.
Arvio tarpeesta 2002:	Tämä tullaan todennäköisesti toteuttamaan HANSABASE Mediapankin omalla tekniikalla.

Ero 20	**Aineistokorttien tulostus ja aineistojen välitys sähköpostilla eteenpäin.**
Luokka:	Materiaalin valinta ja käyttöönotto
Ominaisuus:	Materiaalin kortti voidaan tulostaa joko kuvan kanssa tai ilman, ja materiaali voidaan lähettää sähköpostilla eteenpäin.
HANSABASE:	Ei käytössä.
Arvio tarpeesta 2002:	Kannattanee toteuttaa.

Ero 21	**Materiaalien käyttöönottoon liittyvä käyttöliittymän ja toimintojen profiilikohtainen konfigurointi.**
Luokka:	Materiaalin valinta ja käyttöönotto
Ominaisuus:	Kaikki materiaalin käyttöönottoon liittyvät valinnat ja vaihtoehdot ovat profiloitavissa käyttäjäryhmän tarpeiden mukaan, esim. työasemaan kopioitavan kuvan koko ja tiedostomuoto. Käyttäjä voi myös itse valita, mitkä useimmin tarvittavista toimenpiteistä ovat näkyvissä ns. pikanappeina, jolloin esim. kuvan siirto työasemalle voidaan tehdä suoraan yhden napin painalluksella ilman koritoimintoa.
HANSABASE:	Ei käytössä.
Arvio tarpeesta 2002:	Otetaan käyttöön profiileihin liittyen.

Ero 22	**Tietokannan indeksin selailumahdollisuus**
Luokka:	Tekniikka: Tietokanta
Ominaisuus:	Myös tietokannan indeksi on selattavissa käyttöliittymän kautta.
HANSABASE:	Ei käytössä.
Arvio tarpeesta 2002:	Mietitään ominaisuuden toteuttamista HANSABASE Mediapankin ns. Advanced Users –käyttöliittymässä.

Ero 23	**Thesaurus-tyyppinen asia- ja synonymisanasto käyttöön**
Luokka:	Tekniikka: Tietokanta
Ominaisuus:	Tietokantaan saa lisäoptiona myös thesaurus –tyyppisen asia- ja synonyymisanaston, jota voidaan käyttää apuna korttien täytössä ja hakuja muodostettaessa.
HANSABASE:	Ei käytössä.
Arvio tarpeesta 2002:	Kannattaa tutustua tekniikkaan ja harkita käyttöönottoa.

Ero 24	**Järjestelmän hajauttaminen**
Luokka:	Tekniikka: Arkkitehtuuri
Ominaisuus:	GWX mahdollistaa järjestelmän hajauttamisen
HANSABASE:	Ei käytössä.
Arvio tarpeesta 2002:	Otettaneen mm. tietoturvasyistä käyttöön uudessa versiossa.

Ero 25	**Tekstimuotoisen tiedon tallentaminen RTF -muodossa kantaan**
Luokka:	Tekniikka: Tiedostomuodot
Ominaisuus:	Tekstit syötetään järjestelmään RTF –muodossa, jolloin tekstiin määritellyt tyylit voidaan automaattisesti konvertoida vastaaviksi materiaalikortin kentiksi. RTF –muotoinen teksti ladataan konvertoituna suoraan tekstitietokantaan, jolloin teksti on aina haettavissa sekä kortilla olevan liitetiedon, että itse tekstin perusteella.
HANSABASE:	Erillisiä tekstiominaisuuksia ei HANSABASE Mediapankissa ole käytössä.
Arvio tarpeesta 2002:	Harkitaan tarvetta.

Ero 26	**Audio- ja videomateriaaleille automaattinen sormenpää.**
Luokka:	Tekniikka: Lisäominaisuudet
Ominaisuus:	Audio- ja videomateriaaleille Maria tarjoaa oman materiaalikortin sekä yleisimmät audioformaatit (WAV, AIFF jne.) tunnistavan konversio-ohjelman, jolla järjestelmä luo automaattisesti "audiosormenpään" alkuperäisestä äänitiedostosta.
HANSABASE:	Audiosormenpäitä HANSABASE Mediapankki ei osaa tehdä.
Arvio tarpeesta 2002:	Selvitetään tekniikka ja mietitään käyttöönottoa.

Ero 27	**Print-on-demand ominaisuus.**
Luokka:	Tekniikka: Lisäominaisuudet
Ominaisuus:	Mariaan on myös saatavissa erillinen print-on-demand -moduuli, joka mahdollistaa esim. esitemateriaalien tuotannon ja varaston seurannan
HANSABASE:	Varaston seurantaa HansaBind ympäristössä
Arvio tarpeesta 2002:	Voisi olla hyödyllinen ominaisuus. HansaBind ratkaisun liitäntää HANSABASE Mediapankkiin varastonseurannan osalta kannattaa harkita.

Ero 28	**MediaFlow työnkulku.**
Luokka:	Tekniikka: Lisäominaisuudet
Ominaisuus:	Uusi MediaFlow[4] työnkulku monipuolisine mahdollisuuksineen. Sisältäen töiden ja tehtävien seurannan, tilat, varaukset ja suorituksien kirjaukset, aikatietojen osalta deadlinet, tiedon ajankäytöstä ja varoitukset sekä kehittyneen tuotantotilastoinnin.
HANSABASE:	Ei käytössä
Arvio tarpeesta 2002:	Ehdottomasti tärkeä ominaisuus

Tehtyjen arvioiden jälkeen, budjetillisista ja liiketaloudellisista syistä, Hansaprint päätyi valitsemaan maltillisen etenemistavan. HANSABASE Mediapankkia kehitettiin tuossa vaiheessa tuoden palveluun mukaan uusia MediaVu ominaisuuksia vain pikku hiljaa ja tarpeen mukaan.

[4] Grafimedia oli tuolloin kehittänyt Hansaprintin alkuperäisen idean ja demototeutuksen pohjalta (Seppo Haapakoski/Hanska) uutta MediaFlow -työnkulkusovellusta. MediaFlow -sovelluksen pilottikäyttö Turun Sanomien sivunvalmistuksessa oli lähtenyt käyntiin syksyllä 2001. Tavoitteena oli saada liitettyä samat työnkulkuominaisuudet myös HANSABASE Mediapankkiin. Ensivaiheessa ominaisuuksia hyödynnettäisiin mm. Nokia Issue Management, jossa työnkulut ja hyväksyntäkierrokset olivat varsin monimutkaisia.

5 Näkökulmat tilanteeseen

Tässä luvussa käydään läpi HANSABASE-ohjelmiston vanhenemista kolmen näkökulman kautta. Kappaleessa 5.1. mietitään syitä HANSABASE-palvelun vanhenemiselle, kun asiaa tarkastellaan vallinneen liiketoimintaympäristön näkökulmasta. Analyysi perustuu pitkälti omalle käsitykselleni tapahtuneesta. Kappaleessa 5.2. analysoidaan tapahtunutta käyttäen näkökulmana tutkielman luvussa kaksi esille tuotua Parnasin ’Software Aging’-teoriaa. Kappaleessa 5.3. arvioidaan mitä olisi vaikuttanut ohjelmiston vanhenemiseen 2000-luvun alussa tehdyn HANSABASE-kehityssuunnitelman toteuttaminen aikanaan.

5.1 Mitä tapahtui HANSABASELLE, näkökulma liiketoimintaympäristön kannalta

Kappaleessa 5.1. mietitään syitä HANSABASE-palvelun vanhenemiselle, kun asiaa tarkastellaan vallinneen liiketoimintaympäristön kautta. Analyysi perustuu pitkälti omalle käsitykselleni tapahtuneesta.

HANSABASE-palvelun historiaa

Tämän tutkielman luvun neljä alussa kerrotaan HANSABASE-järjestelmän merkityksestä Hansaprintille. Hansaprint oli kehittänyt painoasiakkailleen tarjottavia sähköisiä lisäarvopalveluja, joita kutsuttiin kokonaisuutena Print+ -palveluiksi. HANSABASE Mediapankin lisäksi Print+ -palveluihin kuuluivat mm. HANSANET -aineistonsiirtopalvelu ja HANSABIND -palvelu liitteistyksen hallintaan. Hansaprintissä tarvittiin HANSABASEa lisäarvopalveluroolinsa lisäksi tuotannollisten painoaineistojen hallintaan sekä oman brändinsä ja graafisen ilmeensä ylläpitoon ja ohjaukseen.

Alun perin HANSABASE oli WWW-ympäristöön toteutettu kuvapankki, joka kehittyi ajan myötä täysiveriseksi aineistonhallintajärjestelmäksi. Se oli ensimmäisiä webin hyötysovelluksia ja perustui Grafimedian Maria-tuotteeseen.

Toimin Hansaprintissä HANSABASE-projektin projektipäällikkönä aivan projektin alkuajoista lähtien. Ensimmäisiä HANSABASE -asiakkaita oli Nokia Mobile Phones, joka käytti palvelua aluksi puhtaasti kuvapankina laitekuviensa hallintaan. Hansaprint painoi noihin aikoihin suurimman osan Nokia Mobile Phonesin puhelinten käyttöohjeista. Myöhemmin Nokia käytti HANSABASEa moneen muuhunkin tarkoitukseen. Järjestelmän merkitystä kuvaa se, että esimerkiksi Nokialle niinkin merkittävä asia, kuin Nokian Online Brandbook toteutettiin myös HANSABASEn pohjalle. Lisäksi esimerkiksi ”Nokia Oyj:n historia” -kirjasarjan kirjoittaja professori Martti Häikiö käytti HANSABASEa kirjoittamisen apuna teoksiin tulevien kuvien hallinnassa, keräilyssä ja luokittelussa.

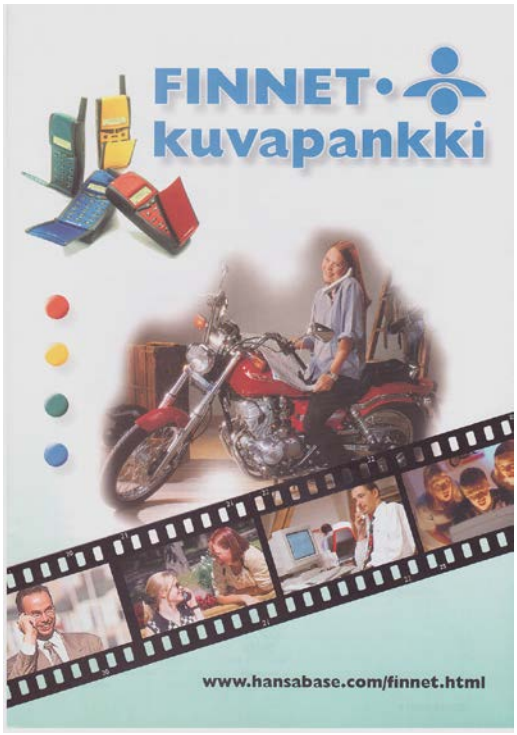
Nokian rinnalle järjestelmä sai uusia asiakkaita Hansaprintin muista painoasiakkaista. Sähköiset Print+ -palvelut olivat varsin merkittävä kilpailutekijä Hansaprintille, kolmanneksi tukijalaksikin kutsuttu. Muut tukijalat olivat perinteistä painotyötä edustavat aikakauslehdet ja erilaiset manuaalit. Aiemmin isossa roolissa ollut puhelinluetteloliiketoiminta oli jo kukoistuksensa ajoista huomattavasti kuihtunut.

5.1.1 HANSABASE-asiakkaita ja erilaisia palvelun käyttötapoja

Muista HANSABASE-palvelun asiakkaista mainittakoon esimerkiksi: Expert-ketju, Finnet Yhtiöt, Raisio Yhtymä, Porin Jazz, TeamWARE Group, Tietokuva Oy/Turun Sanomat, Turun kaupunki, WSOY:n kirjakerhot sekä Hansaprint itse.

Expert-ketju oli tuohon aikaan Suomen, Euroopan ja koko maailman suurin kodinkonealan ketju. Suomessa Expert käytti HANSABASEa graafisen ilmeensä sekä tuotekuva-materiaalien hallintaan. Järjestelmään tallennettiin logot, imagokuvat, tuotekuvat, ilmoituspohjat ja muut mainonnan peruselementit. Myös uudet televisiomainosfilmit olivat nähtävillä HANSABASEssa. Seuraava asiakkaan antama suositus on peräisin tuon aikaisesta HANSABASE -esitteestä: ”Yksi mediapankin merkittävimmistä hyödyistä on se, että saamme yhä enemmän sukunäköisyyttä valtakunnalliseen ja alueelliseen Expert-mainontaan.” (Hansaprint HB 2001).
Tallennetuista korkearesoluutioisista tuotekuvista järjestelmä generoi ennalta sovitut versiot esim. WWW-käyttöön. Tämä oli toki kaikkien kuvapankkien perusominaisuuksia. Expert-ketjua varten toteutukseen lisättiin ominaisuus, jolla viikkokampanjatuotteista saatiin automaattisesti tulostettua A4-kokoiset kampanjahintalaput. Tulostus tapahtui joko liikkeissä tai palvelukeskuksissa. Aineistopankin käyttö oli pitkälti sidoksissa painosopimukseen, ja se päättyi, kun monen vuoden sopimus aikanaan loppui.

Finnet yhtiöt oli peräti 45 yksityisen alueellisen puhelinyhtiön ryhmittymä, jonka yhtenäistä graafista ilmettä hallittiin HANSABASE-järjestelmällä. Kantaan talletettiin valtakunnallisten kampanjoiden materiaalit alueelliseen käyttöön sekä fiilis- ja tuotekuvamateriaalia, joka kävi kaikkiin tilanteisiin (Hansaprint FK 1998). Hansaprint painoi tuohon aikaan kaikki Suomen puhelinluettelot ja painosopimukseen lisättiin HANSABASE-palvelu Print+ -palveluna. Käyttö jatkui Finnet -ryhmittymän voimassaollessa painosopimusten alaisena.



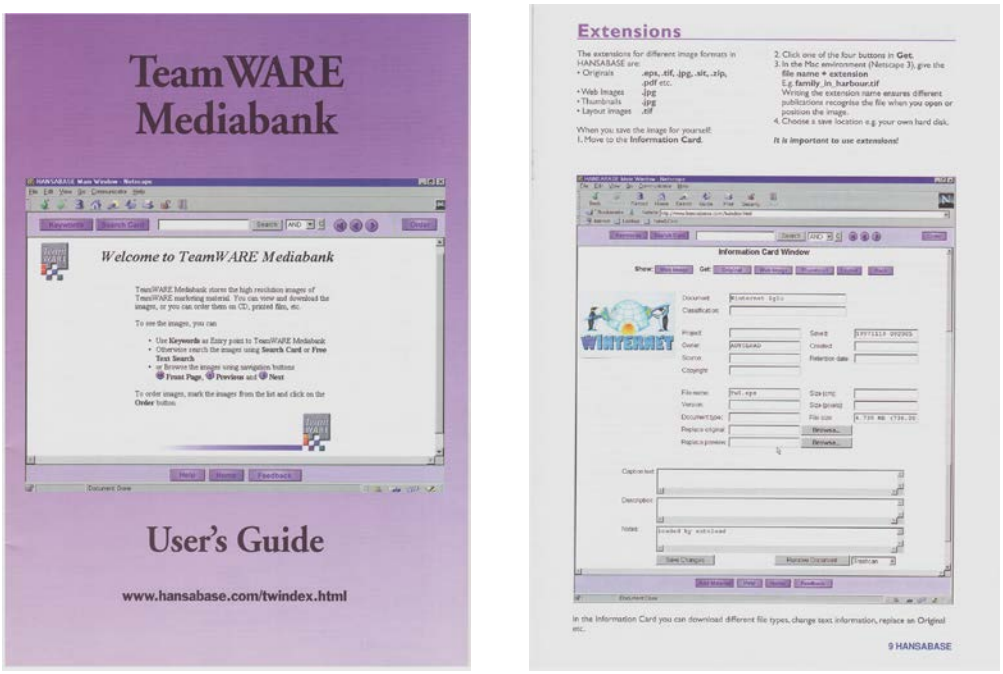
Kuva 5.1 Finnet kuvapankin käyttöohjeen kansi sekä ote käyttöohjeesta (Hansaprint FK 1998)

Porin Jazz käytti HANSABASEa useana vuotena tulevan kesän festivaalin ja artistien mainontaan. Lehdistölle jaettiin käyttöoikeudet, joilla he pääsivät lataamaan kulloisenkin graafisen ilmeen mukaisia materiaaleja sekä kuvia uusista artisteista sitä mukaa, kun sopimuksia julkaistiin. Tapahtuman aikana Porin Jazzin kuvaajat latasivat kantaan kuvia päivän tapahtumista ja artisteista. Lehdistö pääsi hakemaan sieltä viimeisimpiä kuvia artikkeleihinsa ja raportteihinsa. Kantaan kertyi kuva- ja materiaaliarkistoa vuosien kuluessa. Hansaprint teki sopimuksen Porin Jazzin kanssa painotöistä ja HANSABASE-palvelusta, ja sai vaihtokaupassa lippupaketteja yms. Painosopimuksen loputtua palvelu jäi edelleen Porin Jazzin käyttöön useaksi vuodeksi, tällöin rahallista korvausta vastaan.

Raisio Yhtymä käytti HANSABASEa laajan yhtymänsä kuvapankkina niin WWW- kuin painotuotekäytössä sekä graafisen ilmeensä hallintaan. Pankista löytyivät kaikki mallimateriaalit, dokumenttipohjat, värikartat, banneriohjeet jne. Myös seuraava asiakkaan antama suositus on peräisin tuon aikaisesta HANSABASE -esitteestä: ”HANSABASEn käyttöönoton myötä harppasimme riippukansioarkistoinnista nykyaikaan” (Hansaprint HB 2001). Raisiolle tuotettiin myös PressiCD:itä tuotemateriaalista lähetettäväksi asiakkaille ja lehdistölle.

TeamWARE Group oli tuohon aikaan yksi maailman johtavista työryhmäohjelmistojen toimittajista. TeamWARE käytti TeamWARE Mediapankkia painotuotteidensa ja WWW-sivujensa kuvapankkina. Kuvia pääsivät hakemaan mainos- ja viestintätoimistot sekä

tuotepäälliköt ympäri maailman. Yrityksen yhtenäinen myynnin ja markkinoinnin tuki oli Euroopan lisäksi saatavilla samalla kellon lyömällä myös Aasiassa ja Tyynenmerenalueella. (Hansaprint HB 1998, Hansaprint TW 1998). Palvelua käytettiin myös apuna TeamWAREn vuosittaisen asiakastapahtuman järjestelyissä. TeamWAREn HANSABASE-sopimus ja palvelun käyttö loppuivat, kun painosopimukset päättyivät.



Kuva 5.2. TeamWare Mediabank -palvelun käyttöohjeen kansi sekä ote käyttöohjeesta (Hansaprint TW 1998)

Tietokuva Oy/Turun Sanomat (TS) julkaisi päivän Turun Sanomassa olleita kuvia kuvapankissaan. Yleisöllä oli mahdollisuus tilata kuvista paperiversioita järjestelmään toteutetulla toiminnolla. Palvelu toteutettiin lukijoiden tarpeeseen, sillä monet lehden kuvissa esiintyneet ihmiset tilasivat kuvia toimitukselta, ja kuvatilausten manuaalinen hallinta oli erittäin työlästä toimitukselle. Palveluun kertyi pitkän aikaa TS:n kuvamateriaalia, ja kuvia oli mahdollista selailla ja hakea järjestelmän WWW-käyttöliittymän helppokäyttöisyyden myötä. Oleellista oli, että kuvien metadatoihin kirjattiin oikeat ja riittävän kattavat avainsanat esiintymispäivämäärän lisäksi.

Myös **Turun Kaupunki** käytti HANSABASEa kuvamateriaalinsa hallintaan. Kaupunki oli ilman muuta yksi suurimmista HANSABASE-asiakkaista. WWW-sivujen kehitystä varten HANSABASE integroitiin Fujitsun Turulle toimittamaan julkaisujärjestelmään. Tämän mahdollistamiseksi Hansaprint rakensi järjestelmään rajapinnan, jonka kautta Fujitsun julkaisuohjelma pystyi pyytämään halutun kuvan halutussa koossa, jolloin WWW-sivuja tuotettaessa HANSABASE-järjestelmästä sai haettua halutusta kuvasta halutun kokoisen

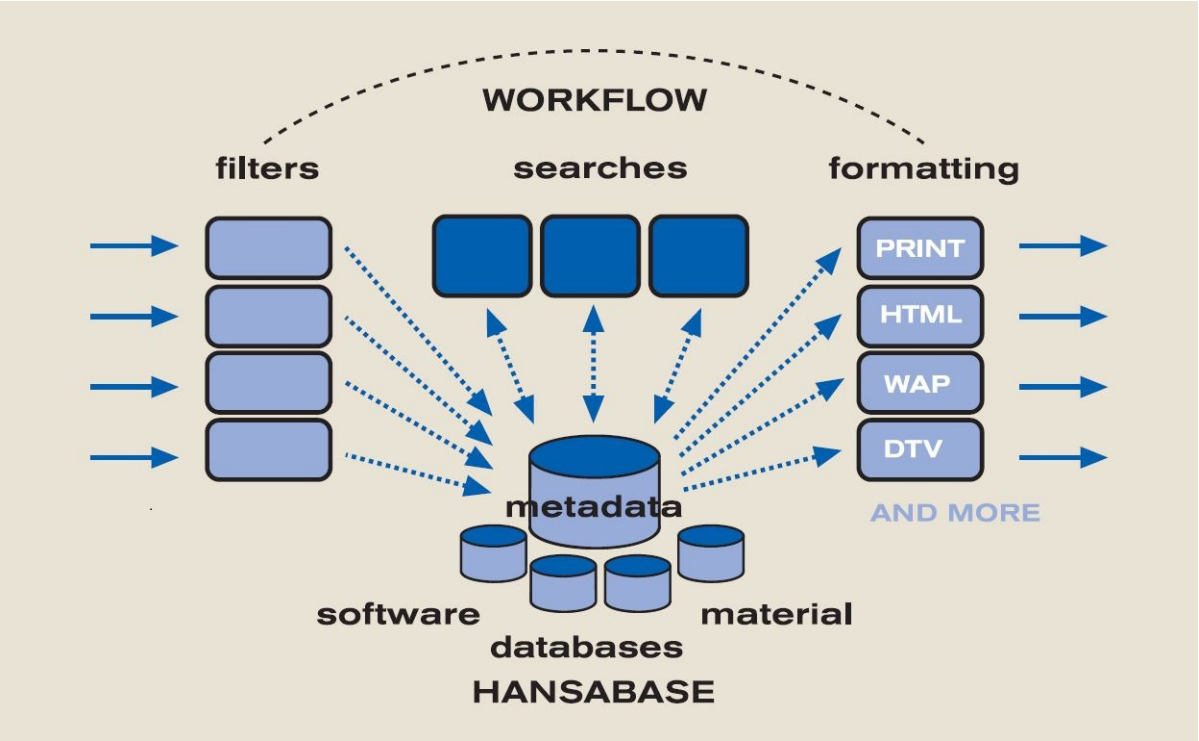
version. Turun kaupungille toteutettiin myös lehdistötiedotepankki, joka generoi tapatuma- ja tiedotekohtaisesti lehdistölle materiaalisivun, joka aktivoitui vasta määriteltynä julkaisupäivänä. Lehdistötiedotteen julkaisun aikaan pääsi tapahtuman materiaaleihin tutustumaan ja myös lataamaan niitä Turun Kaupungin sivuilla olleesta lehdistötiedotteet osiosta. Myös Turun Kaupungin museoiden taideteosten ja esineistön kuvien hallintaan kehitettiin oma käyttöliittymä. Palvelu oli tärkeä myös museoiden materiaalin luettelointia ajatellen.

WSOY:n kirjakerhot käyttivät HANSABASEa moninaisten kirjakerhojensa kuvamateriaalien hallintaan. Kirjat kuvattiin uusina, eri kuvakulmista, jonka jälkeen ne talletettiin kuvapankkiin, josta halutut versiot olivat saatavissa tarvittaessa kirjakerhon painotuotteisiin ja kuluttajille suunnattuihin nettipalveluihin. Myös seuraava asiakkaan antama suositus on peräisin tuon aikaisesta HANSABASE -esitteestä: ”HANSABASE tehostaa kirjakerholehtien ja muun markkinointiviestinnän tuotantoa. Oman henkilökunnan lisäksi mediapankkia käyttävät freelancer -suhteessa olevat taittajat ja mainonnan suunnittelijat.” (Hansaprint HB 2001). Palvelu tehosti tuotannon työnkulkua huomattavasti. WSOY:n kirjakerhoilla palvelun käyttö jatkui painosopimusten jatkumisen ajan.

Myös **Hansaprint** itse käytti HANSABASE-järjestelmää useampaankin eri tarkoitukseen. Luonnollisesti kuva- ja materiaalipankkina, mutta myös graafisen ohjeistuksensa hallintaan. Kantaan tallennettiin lisäksi Hansaprintin asiakaslehden sekä Hansalainen henkilökuntalehden pdf-versiot. Lisäksi erilliseen historiakantaan kerättiin erilaista Hansaprintin ja TS:n historiamateriaalia.

Eräs mielenkiitoinen käyttökohde HANSABASElle oli virallisten asiakirjojen versionhallinta-järjestelmä, jolla hallinnoitiin mm. Hansaprintin paino- ja muita sopimuksia. Järjestelmä lähetti dokumentille asetetulle vastuuhenkilölle hälytyksen määräaikana ennen sopimuksen vanhenemista, jolloin vastuuhenkilö osasi aloittaa sopimuksen uusintaan tai sen päättämiseen liittyvät asianmukaiset toimenpiteet. Lisäksi järjestelmässä säilyivät sopimusten vanhat versiot, jolloin niihin pystyi palaamaan tarkistusmielessä tarpeen mukaan.

HANSABASE -palvelun rooli kehittyi vuosien kuluessa pelkästä painotöiden materiaalin hallintapalvelusta ja kuvapankista putkeksi erilaisiin medioihin. Katso kuva 5.3.



Kuva 5.3. HANSABASE -palvelun looginen rakenne. Kuva HANSABASE -esitteestä vuodelta 2004 (Hansaprint HB 2004)

Kuvassa 5.3. esitetään HANSABASE -palvelun looginen rakenne. Vasemmalta oikealle lukien nähdään aineistojen kulku putkessa HANSABASE -järjestelmän läpi. Ensin aineistot syötetään järjestelmään, jolloin ne käyvät läpi erilaiset suodattimet riippuen materiaalin tyypistä sekä asiakaskannassa määritellyistä ominaisuuksista. Järjestelmä luo näyttökuvat ja erottaa metadatan aineistoista. Nämä talletetaan järjestelmän tietokantoihin. Materiaalia saa haettua kannoista suorittamalla erilaisia hakua. Vastauksena saadaan metadataa, tai tiedostolistoja. HANSABASE järjestelmästä saadaan noudettua materiaaleja eri medioihin noudattaen mediaspesifejä aineistomäärityksiä, esim. skaalaamaan aineistoa median vaatimaan kokoon ja muotoon. Tämän kokonaisuuden läpi johtaa työnkulku (workflows), jolla materiaaleille saadaan määriteltyä halutunlaisia, esim. aineistotyyppi- ja asiakaskantakohtaisia työnkulkuja.

Kun Hansaprint kehitti HANSABASE-asiakkailleen kullekin sopivaa versiota palvelusta, johti se ohjelmakoodin pirstaloitumiseen, hallinnan vaikeutumiseen ja samalla ohjelmiston vanhenemiseen. Ohjelmisto oli alun perin suunniteltu kuvapankiksi, ja laajeni luonnollisesti kaiken digitaalisen aineiston hallintaan, mutta myös esimerkkisi brand bookiksi brandin hallintaan tai vaikka pressipankiksi tiedotteiden ja niihin liittyvien materiaalien jakeluun. Myös tällainen järjestelmän laajentaminen selkeästi eri suuntiin luo koodiin sirpaleisuutta, joka vanhensi ohjelmistoa.

Matkan varrella vanhenemiseen ja pirstaloitumiseen pyrittiin vastaamaan ajoittaisilla ja valitettavasti vain osittaisilla koodin siivouksilla. Järjestelmän pohja, eli emojärjestelmä päivitettiin uudempaan versioon kertaalleen, mutta vähäisistä resursseista johtuen työtä ei tehty riittävän kattavasti. Myös henkilökunnan ammattitaidossa olisi aivan varmasti ollut kehitettävää.

5.1.2 Mitä tapahtui suurimmalle asiakkaalle

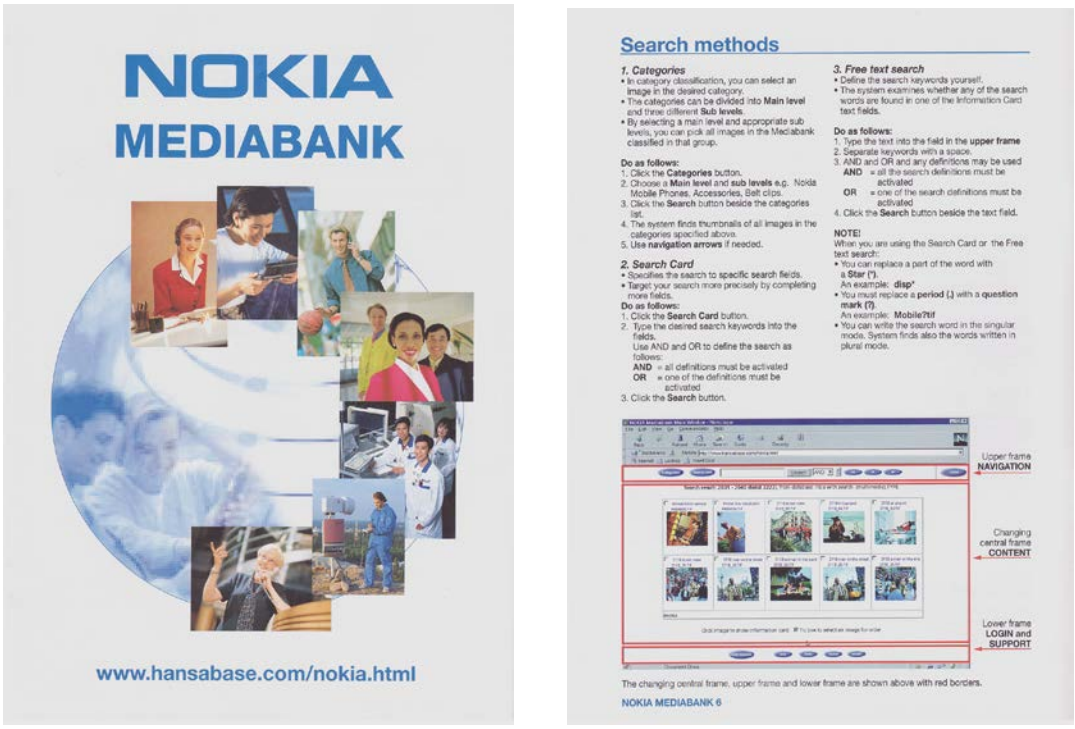
Kun ohjelmiston kehitys aloitettiin, HANSABASE rakennettiin Grafimedian Maria-tuotteen päälle. Tavoitteena oli saada aikaan WWW-pohjainen digitaalisen aineiston hallintajärjestelmä Hansaprintin asiakkaiden sähköisten materiaalien hallintaan painotöiden tuottamisen tueksi. Siis sekä helpottamaan painoprosessia että asiakkaalle omaan käyttöön kuvien, logojen yms. hallintaan.

Matkan varrella asiakkaita tuli lisää, samalla uusilta asiakkailta tuli spesifejä tarpeita, joihin pyrittiin vastaamaan. Ja valitettavasti ohjelmakoodi samalla pirstoutui, vaikka sitä yritettiin hallita perinteisin rakenteisiin perustuvin menetelmin.

Resurssien puutteessa oli helpompaa kopioida koodi uudelle asiakkaalle vanhasta ympäristöstä ja työstää siihen uusia ominaisuuksia, joita ei viety vanhoille asiakkaille. Koodista siis pikku hiljaa muodostui useita eri versiota.

Hansaprintin suurin asiakas oli tuohon aikaan Nokia Mobile Phones (NMP). Hansaprint painoi silloin lähes kaikkien Nokian matkapuhelinmallien manuaalit. Kehitimme HANSABASEn pohjalta Nokia Mobile Phonesin tuolloin Suomessa sijainneelle Markkinointiosastolle Nokia Mediabank-ratkaisun.

Ajan kuluessa toki myös Nokia kasvoi ja kehittyi. Yhä useammat osastot ottivat palvelun käyttöön, ja sen merkitys kasvoi. Se mitä kaikkea Nokian sisällä tapahtui tuona aikana ei ole tietenkään tarkkaan tiedossamme, mutta joitakin vaiheita näkyi myös meille palveluntoimittajille. Ratkaisevaa HANSABASEn kannalta oli, että Nokia otti organisaatiossaan 1990-luvun lopussa laajasti käyttöön Documentum-dokumentinhallintaohjelmiston. Isossa organisaatiossa oli valtavasti dokumentteja niin kaupallisella, juridisella kuin tuotepuolellakin, ja niiden keskitetty turvallinen hallinta oli yritykselle ensiarvoisen tärkeää.



Kuva 5.4. Nokia Mediabank -palvelun käyttöohjeen kansi sekä ote käyttöohjeesta (Hansaprint NMB 1998)

Hansaprint osallistui tuohon aikaan Nokian manuaalien tuotannossa käytettyjen järjestelmien osalta Nokian auditointeihin, jolloin mm. tietoturvaan liittyviin asioihin kiinnitettiin suurta huomiota. Auditointien tuloksena myös HANSABASE-järjestelmään tuotiin uusia tietoturvallisuutta lisääviä piirteitä. Läpäisimme auditoinnit ja osoitimme täyttävämme vaaditut tiukat kriteerit palvelujen toimittamiselle Nokialle.

Nokian toiminnan laajennuttua ympäri maailman, heille tuli tarve pystyä käyttämään materiaaleja maailman laajuisesti. Tämä johti siihen, että Nokialla aloitettiin hajautetun aineistonhallinnan suunnittelu. Ajatuksena oli, että palvelimia olisi esim. viidessä kuudessa pisteessä eri puolilla maailmaa, joista järjestelmä noutaisi fyysisen tiedoston sen mukaan mistä se kulloinkin olisi tehokkainta. Sisällöt replikoituisivat tarpeen mukaan. Heillä oli jo käytössään mm. WWW-sivujensa hallinnassa Akamain järjestelmä, joka käytti tiedostojen välitykseen ja replikointiin CDN-metodeja (content delivery network). Nykyään CDN-palveluja käyttävät lähes kaikki kansainväliset sivustot, ja niitä saa myös useilta webbihotelleilta lisäpalveluna.

Nokia teki 2003-2004 evaluointia Nokia Mediabank -palvelun kehityksestä. Kesällä 2004 teimme Nokialle ehdotuksen Nokia Mediabankin HANSABASE -pohjaisesta kehityksestä, jossa huomioitiin Nokian esittämät tarpeet ja osallistuimme tarjouksellamme kilpailutukseen. Pärjäsimme tässä vaiheessa hyvin ja olimme kahden parhaan joukossa, toisen jatkoon

päässeen ollessa Documentum. Toisella kierroksella Nokia vertaili tarkkaan HANSABASE-ratkaisun ja Documentum-ratkaisun mahdollisuuksia ja vahvuuksia.

Prosessin päätteeksi, syystä tai toisesta, Nokia päätyi ratkaisuun, jossa materiaalien hallinta siirtyi osaksi Documentum-kokonaisuutta. Documentumissa ei tuolloin ollut valmiina kehittynyttä kuvien tai muiden digitaalisten materiaalien hallintaa, sillä se oli nimenomaan asiakirjojen hallintaan tarkoitettu järjestelmä. Tästä syystä he joutuivat kehittämään koko digitaalisen aineistonhallinnan kaikkine ominaisuuksineen lähes puhtaalta pöydältä.

Vuonna 2004 Nokia päätti irtisanoa HANSABASE-palvelun. Johtuen edellä mainituista puutteista Documentumin digitaalisen aineistonhallinnan ominaisuuksissa, sen kehittämisen viedessä aikaa (ja rahaa) ilmeisesti ennakoitua enemmän, saimme palvella heitä vielä parin vuoden ajan HANSABASE-pohjaisella Nokia Mediabank järjestelmällä. Viimein vuoden 2006 loppupuolella ominaisuus valmistui Documentumiin. Tässä vaiheessa autoimme vielä Nokia Mediabank -materiaalien ja -metadatojen sekä käyttäjätietojen siirrossa. Palvelut jatkuivat pitkälle vuoden 2007 puolelle. Tarjouduimme tuolloin vielä hoitamaan erilaisia rooleja Documentum Nokia Mediabankissa, mutta emme valitettavasti päässeet niistä sopimukseen. Tiedossani ei ole, miten Documentumin aineistonhallinnan kanssa lopulta kävi, tai miten kauan se oli käytössä Nokialla.

Noihin samoihin aikoihin Nokia vaihtoi mainostoimistonsa brittiläiseen (luopuen suomalaisesta) ja siirsi oman markkinointiosastonsa toimintoja Englantiin. Jälkeenpäin miettiessäni tätä kehityskulkua olen päätynyt siihen käsitykseen, että on mahdollista, että joku uusi päällikkö Nokian organisaatiossa, mahdollisesti sijoituspaikaltaan englantilainen, halusi rakentaa oman toimintaympäristönsä ja keräsi kaikki prosessiin liittyvät asiat ja tekijät lähelle itseään. Mikäli Nokia olisi päätynyt kehittämään kansamme Nokia Mediabank-ratkaisua Hansaprintin HANSABASE-järjestelmän pohjalta, olisi kokonaisuuden (ja kustannusten) hallinta pysynyt paremmin heidän omissa käsissään, ja Hansaprint olisi todennäköisesti saanut jatkaa palvelun tarjoamista pidempäänkin.

Nokian päädyttyä luopumaan Nokia Mediabank-palvelusta, HANSABASE menetti suuren asiakkaan, mikä mitä todennäköisemmin (muista asiakkaista riippumatta) vähensi Hansaprintin halua panostaa palveluun ja johti samalla koko järjestelmän kehityksen hidastumiseen. Tämä taas johti asiakkaiden kannalta HANSABASEn kiinnostavuuden laskuun ja ajan myötä asiakaskunnan pienenemiseen.

5.1.3 HANSABASEn kehitys emojärjestelmän kannalta

Grafimedian Maria-järjestelmä toimi siis pohjana HANSABASE Mediabank-järjestelmän kehitykselle. Marian kehittyessä ja saadessa uusia asiakkaita (mm. Lehtikuva ja Helsingin Sanomat) otimme uusia ominaisuuksia myös HANSABASE-ympäristöön. Niitä ei varsinaisesti viety kaikkiin asiakasversioihin, vaan ne jäivät ikään kuin pankkiin odottamaan tarvitsijaa. Pikkuhiljaa ominaisuuksia aktivoitiin, mutta ei järjestelmällisesti tai kattavasti. Osittain tähän saattoi liittyä aikanaan Hansaprintin ja Grafimedian välillä tehdyn sopimuksen muoto. Hansaprint oli valinnut mallin, jossa järjestelmästä ei maksettu kattavaa ja tietysti korkeampaa ylläpitomaksua, joka olisi sisältänyt kaikki uudet ominaisuudet ja versiot, vaan matalampaa, joka edellytti aina uusia ominaisuuksia mukaan otettaessa erillistä kauppaa, ja mahdollista uutta käyttöönottoprojektia.

Grafimedia kuitenkin kehitti omaa ympäristöään edelleen ja rakensi Mariasta kokonaan uuden, MediaVu-ratkaisun asiakkaidensa käyttöön. Toteutus oli modernimpi, objektiorientoituneempi ratkaisu. Hansaprint lykkäsi uuden MediaVu-pohjaisen ratkaisun käyttöönottoa, sillä odotettavissa oleva transitiotyö oli valtava. Lopulta transitio uuteen ohjelmaympäristöön tehtiin, vanhat roskat siivottiin pitkälti samalla, muttei ehkä riittävän kattavasti. Oltiin taas lähes samanlaisessa tilanteessa kuin aiemminkin, vanha koodi oli sekametelisoppaa, uusi modernimpaa ja siistimpää, mutta koska uutuuksia ei otettu kaikille käyttöön, alkoi koodin pirstaloituminen uudelleen.

Nokia Mediabank-palvelun jatkon neuvottelujen aikana Hansaprint suunnitteli vielä Grafimedian aineistonhallinnan uusimman MediaXi -version käyttöönottoa, mutta prosessi keskeytyi, kun Nokia päätyi muihin ratkaisuihin. Tämä päätös saattoi olla juuri se lopullinen niitti HANSABASEn alamäelle ja lopulta kuihtumiselle.

HANSABASEn viimeiset vaiheet eivät ole tiedossani, koska siirryin muihin tehtäviin ja lopulta kokonaan pois talosta.

5.2 Miten HANSABASE vanheni, näkökulma ’Software Agingin’ kannalta

Kappaleessa 5.2. Analysoidaan tapahtunutta käyttäen näkökulmana tutkielman luvussa 2 esille tuotua Parnasin ’Software Aging’ -teoriaa kohta kohdalta.

HANSABASE-palvelu rakennettiin pohjautuen Grafimedian Mariaan, kuitenkin ajatuksena loppuasiakkaan käyttökokemuksen helpottaminen. Alla oleva järjestelmä pohjautui vahvasti Grafimedian toteuttamiin rakenteisiin, logiikkaan ja tietomalleihin. Kun Grafimedia (GM) lisäsi

ominaisuuksia järjestelmäänsä, eivät ne automaattisesti siirtyneet HANSABASEen, vaan aina käytiin erilliset neuvottelut, joissa sovittiin millä tavalla, ja mihin hintaan ominaisuuksia saatiin käyttöön. Hansaprint ei ollut valmis maksamaan jokaisesta päivityksestä erikseen, ja siksikin järjestelmä jäi auttamattomasti jälkeen emostaan. Toisaalta uusia asiakkaita tuotiin järjestelmään monistamalla vanhoja rakenteita, ne kevyesti uudelle asiakkaalle sovittamalla. Myös uusia ominaisuuksia toteutettiin asiakkaittain, sitomatta niitä kokonaisuuteen riittävän vahvasti. Ominaisuudet jäivät Hansaprintin omaisuudeksi, eikä niitä vastaavasti viety Grafimedian suuntaan.

Seuraavissa kappaleissa 5.2.1 – 5.2.6 mietitään HANSABASE-järjestelmän vanhenemista kunkin luvussa kaksi esitetyn vanhenemisen syyn kautta. Havaintoni on, että varsin kattavasti ovat vanhenemisen syyt osuneet järjestelmässä kohdalle.

5.2.1 Tekemättömät muutokset

’Tekemättömät muutokset’ (kappale 2.2.1) viittaa siihen, että ohjelmistoon ei tehdä riittävästi muutoksia, jotta se vastaisi käyttäjien toiveita. Ajan kuluessa myös aivan varmasti käyttäjien maailmankuva ja käsitykset ohjelmistoihin liittyvistä tarpeista muuttuivat. Nämä molemmat asiat auttamattomasti tapahtuivat HANSABASE-kehityksessä, vaikka asiakkaita kyllä kuunneltiin, ja muutoksiakin haluttiin tehdä.

Haasteellista HANSABASE-ympäristössä oli se, että vaikka sen peruskäyttö olikin kuva- ja aineistopankkipohjaista, oli asiakkailla heti alun perin erilaisia tarpeita, joita pyrittiin toteuttamaan määrättyihin rajoihin asti. Ominaisuuksien toteuttaminen on arvokasta, joten resurssien rajallisuudenkaan vuoksi siihen ei riittävän hyvin kyetty, jonka osa asiakkaista varmasti koki puutteena.

5.2.2 Epäpätevää leikkaushoitoa

’Epäpätevää leikkaushoitoa’ (kappale 2.2.2) viittaa siihen, että ohjelmistoon tekevät muutoksia ihmiset, joilla ei ole riittävän syvää osaamista sen tekemiseen. Tämäkin voidaan ajatella tapahtuneen HANSABASEssa, sillä alkuperäiset rakenteet, toimintaluupit ja taustalla olevat skriptit olivat Grafimedian (GM) tekemiä ja niitä muokattiin Hansaprintin toimesta. Toki GM myös koulutti Hansaprintin henkilökuntaa, mutta ehkä ei riittävän syvällisesti. Henkilökunta myös vaihtui, eikä osaaminen siirtynyt riittävän hyvin uusille henkilöille. Lisäksi henkilökunta oli osin itseoppinutta ja ilman varsinaista ohjelmointikoulutusta. Korjauksia tehtiin osin kotikutoisesti, joka ilman muuta johti koodin degeneroitumiseen. Pirstoutumista tapahtui, ja se kyllä se toki havaittiin, mutta sitä vastaan ei mielestäni taisteltu riittävästi. Ehkä ei ymmärretty riittävän hyvin pirstoutumisen aiheuttamaa uhkaa.

5.2.3 Kaikki ei ole sitä miltä näyttää

’Kaikki ei ole sitä miltä näyttää’ (kappale 2.2.3) viittaa siihen, ohjelmistossa voi olla ollut jo alun perin suunnitteluvika, joka tulee esille paljon myöhemmin, ja voi seurauksiltaan vaikuttaa ohjelmiston vanhenemisen oireilta. Jos tällainen suunnitteluvika havaitaan ja tunnistetaan, voi sen korjaaminen olla paljon helpompaa kuin vanhenemisen. Tällaisia vikoja lienee HANSABASEssakin, esimerkiksi GM:n Mariassa ollutkin, mutta niiden korjaamisesta tai korjaamattomuudesta minulla ei ole tietoa. Ovat siis korjaantuneet korjaamisen jälkeen seuraavan versiopäivityksen yhteydessä. Toisaalta tämä on HANSABASElle ollut ongelma, sillä kaikki korjauksia ei ole otettu välittömästi tai ehkä ollenkaan käyttöön, kuten kappaleen 5.2 alussa olen asiaa kuvannut.

5.2.4 Vanhenemisen hinta - Kyvyttömyys pysyä vauhdissa

Ohjelmistojen vanhenemisesta puhuttaessa ”Kyvyttömyys pysyä vauhdissa” (kappale 2.3.1) viittaa siihen, että koska helpoin tapa lisätä ominaisuuksia on lisätä koodia, hidastuu ohjelman ylläpito helposti. Koodin lisääminen johtaa koodimassan kasvuun ja ohjelmiston ylläpidon hankaloitumiseen. Käyttäjien kannalta muutoksia ei tehdä riittävän nopealla tahdilla, joka johtaa asiakkaiden kyllästymiseen ja tarpeeseen hakea muita ratkaisuja.

HANSABASEn tapauksessa koodia varmasti lisättiin ilman varsinaista siivousta, mutta se ei ehkä käyttäjäkunnan osalta kuitenkaan johtanut käytön vähenemiseen, sillä HANSABASE oli suurimmalle osalle asiakkaista Hansaprintin Print+ -palvelu, jota käytettiin painopalvelun lisänä.

5.2.5 Vanhenemisen hinta - Heikentynyt suorituskyky

Ohjelmistojen vanhenemisesta puhuttaessa ’Heikentynyt suorituskyky’ (kappale 2.3.2) viittaa siihen, että ohjelmakoodin ad-hoc lisääminen johtaa isompaan suorituskykytarpeeseen asiakkailla ja lopulta jopa käyttölaitteistojen päivitystarpeeseen. Kilpaileva nuorempi tuote on nopeampi ja tarvitsee vähemmän suorituskykyä ja on siksi kiinnostavampi.

HANSABASEn tapauksessa suorituskykyvaatimukset toki kasvoivat koodin lisääntyessä, joka saattoi ilmetä esim. selainversiovaatimusten tiukkenemisena, mikä taas mahdollisesti hankaloitti asiakkailla palvelujen käyttöönottoa tai järjestelmän käytön jatkamista. Suorituskyvyn heikkenemine ei kuitenkaan ollut merkittävässä roolissa HANSABASE palvelun vanhenemisessa.

5.2.6 Vanhenemisen hinta - Vähentynyt luotettavuus

Ohjelmistojen vanhenemisesta puhuttaessa ’Vähentynyt luotettavuus’ (kappale 2.3.3) viittaa siihen käytännön havaintoon, että usein yhden virheen korjaaminen johtaa useampaan uuteen, eli ohjelmaan ylläpidettäessä syntyy lisää virheitä. Tämä taas voi helposti ajan myötä johtaa koko projektin hylkäämiseen.

HANSABASEn tapauksessa näin kävi sikäli, että jossain vaiheessa todettiin, että oli välttämätöntä tehdä päätös lähteä päivittämään HANSABASE Maria-pohjaisesta MediaVu pohjaiseen järjestelmään, ja kirjoittaa kaikki toiminnot uuden mallin mukaisiksi ja samalla siivota kertynyt ”plakki” pois.

5.2.7 Vanhenemisen vaikutusten ennaltaehkäisy HANSABASE -järjestelmän kannalta

Seuraavissa kappaleissa 5.2.8–5.2.11 mietitään luvussa 2 esiteltyjä vanhenemisen ehkäisykeinoja HANSABASE-järjestelmän vanhenemisen ennalta ehkäisemisen kannalta. Mitä esitettyjä keinoja käytettiin käytännössä ja miten esitetyt keinot olisivat käsitykseni mukaan auttaneet HANSABASEn kohdalla.

5.2.8 Onnistumiseen tähtäävä suunnittelu – Design for Success

Kappaleessa 2.4.1 käytiin läpi onnistumiseen tähtäävän suunnittelun merkitystä vanhenemisen ennaltaehkäisyssä sekä mietittiin syitä miksi se saattaa unohtua tai jäädä kokonaan toteuttamatta. Kappaleessa lueteltuja tekniikoita ei HANSABASE-järjestelmän vanhimmassa Maria-pohjaisessa versiossa pahemmin käytetty, mutta MediaVu -pohjaisessa jo paremmin.

Tästä syystä otan tässä erityisesti kantaa syihin, joista johtuen esitettyjä tekniikoita ei käytetty.

SYY ASIAN HOITAMATTOMUUTEEN	MITEN HANSABASESSA
Muutostyyppejä ei tunnettu, eikä niiden todennäköisyyksiä osattu arvioida, koska niiden todennäköisyyksien arviointia ei ollut opetettu.	Ilman muuta näin HANSABASEssa. Ohjelman kehityksen yhteydessä ei osattu varautua tuleviin muutoksiin riittävän hyvin. Asiasta olisi kyllä ollut paljonkin hyötyä muutosten hallinnan kannalta
Tekijöiden kiire ja kärsimättömyys. Johdon päällimmäisenä tavoitteena seuraava deadline. Tietoinen päätös osittaisen	Tämäkin luonnehdinta sopii HANSABASEn kehitykseen hyvin. Tekijöillä oli jatkuva kiire seuraavaan toteutukseen, samoin johdolla

ratkaisun toteuttamisesta.	tarve edetä seuraavaan asiakkaaseen. Syntyi teknistä velkaa. Paremmalla ja rauhallisemmalla lähestymisellä olisi ilman muuta päästy parempaan lopputulokseen.
Metodin luonnottomuus, eli ohjelmoija ei koe metodia omakseen, eikä hahmota mallin kokonaisuutta	Ohjelmoijilla ei ollut osaamista eikä halua metodien käyttöön ohjelmoinnissa. Esim. olioperusteiset ratkaisut eivät olleet kehittäjille tuttuja, joten siihen suuntaan ei lähdetty. Lisäkoulutus olisi ilman muuta auttanut.
Vääränlaisen koodin jäljittely	Kaikki uudet HANSABASE-asiakkaiden toteutukset olivat edellisen toteutuksen kopioita, jolloin vääränlainen koodi monistui.
Käsitteiden sekoittaminen, eli suunnittelumalli ei ole sama kuin ohjelmointikieli.	Ohjelmoinnissa ei osattu lähteä kehitty-neempään toimintamalliin, vaan noudatettiin ohjelmointikielen rajoittuneita perusmalleja. Osaaminen korkeampi taso olisi auttanut.
Ammattitaidottomuus	Osa tekijöistä oli taustaltaan, varsinkin alkuaikoina, varsin vajaalla ohjelmointi-koulutuksella varustettuja. Ohjelmoijien lisäkoulutus tai erilaiset rekrytoinnit olisivat auttaneet. Myöhemmin palkattiin uusia osaavia ohjelmoijia, mutta silloin juna oli pitkälti jo mennyt.

5.2.9 Parempaa kirjanpitoa, eli dokumentoinnin merkitys

’Parempaa kirjanpitoa’ (kappale 2.4.2) viittaa laadukkaan dokumentaation kiistämättömään merkitykseen ohjelmien kehitykselle ja ylläpidolle. Hyvä dokumentaatio auttaa niin järjestelmän kehittäjiä, jatkokehittäjiä kuin sen huollosta vastaavia vaativan tehtävänsä hoidossa. Tai siis auttaa, kun sellainen on olemassa. Kyseessä on aikaa ja kustannuksia vievä työvaihe, tai itseasiassa jatkuva tehtävä. HANSABASE-järjestelmän dokumentaation ylläpitoa ei seurattu erikseen, vaan tekijät dokumentoivat omaa käyttötarkoitustaan varten omalla tavallaan, tai sitten eivät dokumentoineet lainkaan. Tämä johti yhä hankalampaan ylläpitoon, erityisesti tekijöiden vaihtuessa. Ja samalla aiheutti järjestelmän vanhenemista. Emojärjestelmän dokumentaatio oli lähtökohtaisesti parempi ja se tuki ylläpitoa, mutta kun HANSABASE kasvoi omalle suunnalleen, oli siitä vain osittainen hyöty.

5.2.10 Ulkopuolisten mielipiteet, eli ohjelmistokatselmukset

’Ulkopuolisten mielipiteillä’ (kappale 2.4.3) viitataan tässä erikseen järjestettäviin ohjelmistokatselmuksiin. Niiden avulla on mahdollista ulkopuolisen tahon toimesta verifioida kehityksen oikea suunta ja sovittujen työtapojen ja mekanismien noudattaminen. Tällaisia katselmuksia ei HANSABASE-järjestelmässä erikseen pidetty. Toki haasteellisista uusista ominaisuuksista tai korjauksista käytiin projektiryhmässä keskusteluja, mutta varsinaista ulkoista auditointia ei koskaan tehty. Jos tällainen menettely olisi ollut käytössä, olisi ohjelmoijien menettelyt varmasti paremmin seuranneet sovittuja tapoja, ja ohjelmakehityksessä säilynyt paremmin ryhti. Ohjelmiston elinkaaren loppuvaiheessa prosessiin otettiin ketterän kehityksen työkaluja, jolloin katselmuksiakin tehtiin, mutta pienessä mittakaavassa. Tällöin kuitenkin oltiin jo ohjelmiston elinkaaren siinä vaiheessa, että katselmuksista oli vain hyvin rajallinen hyöty.

5.2.11 Miksi ohjelmistojen vanhenemista ei voi estää?

Kappaleessa 2.4.4 on todettu, että vaikka ohjelmiston kehittäjät tekisivät kaikkensa, vanhenee ohjelmisto joka tapauksessa. Tämä pitää paikkansa, sillä ohjelmiin tehdään kaikesta huolimatta jossain vaiheessa muutoksia, jotka rikkovat sovittuja sääntöjä. Lisäksi ulkoisilla tekijöillä on suuri rooli ohjelmistojen vanhenemisessa. HANSABASE ei ollut tässäkään mielessä mitenkään poikkeus. Ehkäpä HANSABASEn tapauksessa, palvelun ollessa Hansaprintin Print+ -lisäpalvelu, oli ulkoisilla tekijöillä normaaliakin isompi rooli.

5.2.12 Ohjelmistojen geriatriaa

Kappaleessa 2.5 ja sen alaosioissa käytiin läpi työkaluja vanhenneiden ohjelmistojen hoitoon. Ohjelmistot vanhenevat joka tapauksessa ajan myötä huolimatta siitä, miten hyvin vanheneminen otetaan huomioon ohjelmistojen kehityksessä ja huollossa. Kappaleissa 5.2.13 – 5.2.17 tutkitaan, miten näitä työkaluja on käytetty HANSABASEssa, onko niiden käyttäminen auttanut, ja jos työkaluja ei ole käytetty, olisiko mahdollisesti kuitenkin kannattanut.

5.2.13 Heikkenemisen pysäyttäminen

’Heikkenemisen pysäyttämisellä’ (kappale 2.5.1) viitataan siihen, että vanhenemisen oireita havaittaessa pitäisi heti tarttua korjaaviin toimenpiteisiin heikkenemisen pysäyttämiseksi. Myös ohjelman huollossa pitäisi noudattaa ennaltaehkäiseviä toimenpiteitä. Tätä ei järjestelmällisesti noudatettu HANSABASE-kehityksessä, joten vanhenemisen oireet valitettavasti kumuloituivat.

5.2.14 Dokumentointi jälkikäteen

’Dokumentointi jälkikäteen’ (kappale 2.5.2.) viittaa siihen, että vaikka dokumentointi olisi tehty aikanaan, mutta sitä ei olisi tunnollisesti ylläpidetty, tai sitä ei olisi tehty lainkaan, kannattaa dokumentointi kuitenkin jälkikäteenkin tehdä tai sen tasoa parantaa. Prosessissa havaitaan piirteitä, jotka kannattaa korjata tai toteuttaa eri tavalla. HANSABASEn dokumentointia ei kehitetty projektin aikana merkittävästi, eikä edellä mainittuja etuja näin saavutettu. Jos tämä olisi tehty ajallaan, olisi siitä aivan varmasti ollut hyötyä.

5.2.15 Modularisointi jälkikäteen

’Modularisoinnilla jälkikäteen’ (kappale 2.5.3) viitataan siihen, että ohjelman kehityksen ja huollon aikana kannattaa kerätä saman tehtäväalueen toiminnallisuuksia yhteen ja näin selkeyttää koodia, joka myös helpottaa mahdollisia tulevia isompien muutosten toteutuksia. Tätä mallia noudatettiin HANSABASEssa jonkin verran, ja koodi parani matkan varrella, mutta ei riittävästi. Ilman muuta HANSABASEssa olisi kannattanut tehdä jälkikäteen modularisointia järjestelmällisemmin.

5.2.16 Amputaatio

’Amputaatiolla’ (kappale 2.5.4) viitataan matkan varrella moneen kertaan ja suunnittelemattomasti muutetun koodinpätkän poistamiseen ohjelmakoodista korvaten sen uudella ja tehokkaalla moduulilla. Tätäkin mallia noudatettiin HANSABASEssa jonkin verran, ja näin matkan varrella saatiinkin koodia siivottua, mutta ei riittävästi. Tehokas työkalu, jolla vanhan saa osittain uudistettua, ja vältetään paikatun paikkaaminen.

5.2.17 Laajan ohjelmakoodin leikkaus ja uudelleenjärjestely

’Laajan ohjelmakoodin leikkauksella ja uudelleenjärjestelyllä’ (kappale 2.5.5) viitataan tilanteessa, jossa laajan ohjelmiston koodi on päässyt pahasti rappeutumaan, tehtävään suursiivoukseen, jossa käydään läpi koodin eri versiot, kirjataan erot ja yhtäläisyydet, ja mahdollisuuksien mukaan yhdistellään koodien samanlaiset osuudet. HANSABASEssa rinnakkaisia versioita oli paljon ja niiden ylläpito oli lopuksi todella hankalaa. Tätä toimintamallia olisi ilman muuta kannattanut noudattaa HANSABASEssa voimakkaasti. Jonkin verran tätä tehtiin, ja koodikin selkeytyi, mutta ei riittävästi. Kuitenkaan silloinkaan ei noudatettu riittävän hyvin muita vanhenemiseen ehkäisemiseen tähtääviä periaatteita. Tämä on tehokas työkalu, jolla oikein toteutettuna vanhan saa uudistettua, mutta hallittuja menetelmiä on silloin noudatettava.

5.2.18 HANSABASEn vanhenemisen kannalta merkittävimmät kohdat

Edellä esitellyistä toisessa luvussa esitellyn Parnasin ’Software aging’-teorian mukaisista vanhenemisen syistä sekä vanhenemisen hidastamiseen esitetyistä työkalusta olen tähän kerännyt mielestäni merkittävimmät. Valinnat on perusteltu kuudennessa luvussa kappaleessa 6.3.

i. Heikkenemisen pysäyttäminen (5.2.13, 2.5.1)
ii. Laajan ohjelmakoodin leikkaus ja uudelleen järjestely (5.2.17, 2.5.5)
    Modularisointi jälkikäteen (5.2.15, 2.5.3)
iii. Parempaa kirjanpitoa, eli dokumentoinnin merkitys (5.2.9, 2.4.2)
    Dokumentointi jälkikäteen (5.2.14, 2.5.2, 2.5.3)
iv. Ulkopuoliset katselmukset (5.2.10, 2.4.3)

5.3 HANSABASElle 2000-luvun alussa tehdyn kehityssuunnitelman vaikutus

Kappaleessa 5.3. käydään vielä läpi, miten ohjelmiston vanhenemiseen olisi vaikuttanut 2000-luvun alussa tehdyn HANSABASE-kehityssuunnitelman toteuttaminen.

Vuonna 2002 oltiin siinä tilanteessa, että piti vakavasti miettiä HANSABASE-järjestelmän kehittämistä. Noita ajatuksia ja mahdollisia uusia ominaisuuksia on käyty tarkemmin läpi kappaleessa 4.6.3. Tässä kappaleessa sen sijaan on tarkoitus pohtia sitä, miten ko. ominaisuudet olisivat tuolloin auttaneet HANSABASEn vanhenemista vastaan.

Kappaleessa mietittiin monesta näkökulmasta, miten järjestelmän uusiminen ominaisuuksien kannalta pitäisi tehdä. Tärkeimpiä vaihtoehtoja listattiin seuraavat kolme:

4. kokonaan uusi HB perustuen johonkin muuhun järjestelmään.
5. kerralla uusi HB perustuen tulevaan MediaVuhun ja nykyiseen HB ratkaisuun
6. pikku hiljaa uusittu HB, tuoden palveluun mukaan uusia ominaisuuksia pikku hiljaa

Näistä tuolloin päädyttiin kolmanteen vaihtoehtoon, eli pikku hiljaa uusittuun Mediapankkiin.

Oma mielipiteeni on, että tehdyllä linjan valinnalla valitettavasti edesautettiin HANSABASEn vanhentumista, sillä paljon korjattavaa oli jo kertynyt, eikä hitaalla kehityksellä korjattu vanhoja riittävän nopeasti.

Lisäksi kappaleessa listattiin tuolloin havaittuja järjestelmän teknisiä kehitystarpeita: levyjärjestelmä uusimista, palvelimen uusimista paremmin varmistetuksi, järjestelmän hajautusta ja kuormituksen jakoa sekä järjestelmän tietoliikenneyhteyksien kehittämistä.

Kun näihinkään ei tartuttu riittävän määrätietoisesti, ei loikkaa nuorempaan järjestelmään syntynyt. Kaikki olisivat olleet tärkeitä.

Kappaleessa 4.6.3. käytiin läpi tarkemmin myös MediaVun uusia ominaisuuksia ja niiden hyödyllisyyttä HANSABASElle. Tässä kappaleessa tarkoitus on pohtia sitä, miten ko. ominaisuudet olisivat tuolloin auttaneet HANSABASEn vanhenemista vastaan.

MediaVu:n ominaisuudet, jotka poikkesivat vuonna 2002 HANSABASE –ratkaisusta

Ero 1 **Aineistosarjat ja paketit**
Arvio tarpeesta 2002: Sarjat ja paketit kannattanee ottaa käyttöön myös HANSABASE Mediapankissa
Arvio nyt: **Ei** varsinaisesti olisi ominaisuutena vaikuttanut vanhenemista vastaan.

Ero 2 **Sekä Web-pohjaiset että natiivit käyttöliittymät**
Arvio tarpeesta 2002: Tarvetta natiiviin asiakasohjelmaan ei ole tiedossa.
Arvio nyt: **EI.** En näe, että tuolla olisi ollut käytännön hyötyä vanhenemista vastaan. Muutenkin suuntana silloin oli WWW-pohjaiset ratkaisut.

Ero 3 **MediaVu:n ns. insinöörikäyttöliittymä**
Arvio tarpeesta 2002: Näitä ominaisuuksia voisi olla HANSABASE Mediapankin ns. Advanced Users -käyttöliittymässä
Arvio nyt: **KYLLÄ**. Monipuolisempi käyttöliittymä ammattikäyttäjille olisi auttanut monipuolisemmassa käytössä, ja samalla laajentanut käyttäjäpohjaa.

Ero 4 **Ylläpitäjän ja arkistonhoitajan työkalut**
Arvio tarpeesta 2002: Mietitään kannattaako tekniikat yhdistää vaiko pysyä omassa. Lisäksi mietitään admin-työkalun tekemistä asiakkaiden admin-käyttäjille.
Arvio nyt: **KYLLÄ**. Monipuolisempi käyttöliittymä asiakkaan admin käyttäjille olisi auttanut monipuolisemmassa käytössä, ja samalla laajentanut käyttäjäpohjaa.

Ero 5 **Materiaalien siirto manuaalisesti tai tausta-ajona**
Arvio tarpeesta 2002: Kannattaa tutustua tekniikkaan ja harkita yleisemmän tavan käyttöönottoa
Arvio nyt: **EI.** En näe, että tuolla olisi ollut käytännön hyötyä vanhenemista vastaan.

Ero 6 **Ylläpitotoimet selaimen kautta**
Arvio tarpeesta 2002: Kenttien oletusarvojen käyttöönottoa ja niiden editointimahdollisuuksien tuomista ylläpitäjän käyttöliittymään kannatta harkita
Arvio nyt: **KYLLÄ.** Ylläpitotoimien helpottaminen olisi edesauttanut käyttäjätyytyväisyyttä. Ylläpitäjien mielipiteellä on suuri merkitys.

Ero 7 **Käyttäjien hallinta MediaVun tavalla**
Arvio tarpeesta 2002: Selvitetään yhdistämistarve ja/tai mahdollisuudet.
Arvio nyt: **KYLLÄ.** Ylläpitotoimien toteuttaminen samalla tavalla kuin emojärjestelmässä olisi helpottanut järjestelmän ylläpitoa tulevaisuudessa. Käyttöliittymän käyttäjäystävällisyys toki olisi pitänyt miettiä läpi. (Vertaa ”insinöörikäyttöliittymä”)

Ero 8 **Eri palveluosien hajautus useille palvelimille**
Arvio tarpeesta 2002: Hajautus kannattanee lisätehon aikaansaamiseksi.
Arvio nyt: **KYLLÄ.** Ilman muuta tärkeä ominaisuus, jolla auttaa palvelun skaalautuvuudessa ja sovituksessa erilaisiin tarkoituksiin erilaissa käyttöympäristöissä. Järjestelmä olisi pystynyt kasvamaan käyttäjien tarpeiden mukaan, ja siten hidastanut sen vanhenemista.

Ero 9 **Järjestelmän sovitus useammille alustoille (UNIX ja NT)**
Arvio tarpeesta 2002: Kannattanee harkita hajautusta mietittäessä.
Arvio nyt: **KYLLÄ.** Alustan joustavuus olisi kasvanut, jolloin HANSABASE olisi sopinut useampiin tarpeisiin. Näin ollen mahdollisuus olisi hidastanut ohjelmiston vanhenemista.

Ero 10 **Laajempi tuki tuetun median tiedostomuodoille**
Arvio tarpeesta 2002: Selvitetään tekniikka ja mietitään käyttöönottoa.
Arvio nyt: **KYLLÄ.** Tiedostomuotojen tuen laajentaminen on oleellista mediapankin käyttökelpoisuuden kannalta. Kaikenlaisen digitaalisen materiaalin käsittely pitää olla mahdollista. Mikäli tiedostomuotojen tuessa ei pysytä ajan tasalla, ohjelmisto vanhenee samassa suhteessa.

Ero 11 **Web-to-print ominaisuus**
Arvio tarpeesta 2002: Kannattanee tutustua kuitenkin toteutukseen tarkemmin
Arvio nyt: **EI.** Ominaisuus oli lähes samanlaisena TS:n/Hansaprintin toteuttaman jo olemassa.

MediaVu:n ominaisuudet, jotka puuttuivat vuonna 2002 HANSABASE Mediapankki –ratkaisusta

Ero 12 **Tietokortit ja materiaalit tyypin mukaan eri tietokantoihin**
Arvio tarpeesta 2002: Ei suoraa tarvetta tähän. Arvioida myöhemmin.
Arvio nyt: **EI.** Suoraa hyötyä en nytkään näe, mutta aivan varmasti Grafimedian kanssa olisi ollut hyvä keskustella eduista. Eräs etu olisi tietenkin se, että emojärjestelmän kanssa samanlaisuus säilytettäisiin mahdollisimman pitkälle.

Ero 13 **Materiaalin versionhallinta**
Arvio tarpeesta 2002: Otetaan käyttöön.
Arvio nyt: **Ei.** Materiaalien versionhallinta on jossain määrin aineistonhallintajärjestelmissä perusominaisuutena, mutta HANSABASEn osalta sillä ei ollut niin isoa merkitystä järjestelmän vanhenemisen kannalta.

Ero 14 **Näkymät yli usean tietokannan**
Arvio tarpeesta 2002: Ei suoraa tarvetta tähän. Arvioida myöhemmin.
Arvio nyt: **EI.** Suoraa hyötyä en nytkään näe, mutta aivan varmasti Grafimedian kanssa olisi ollut hyvä keskustella eduista. Eräs etu olisi tietenkin se, että emojärjestelmän kanssa samanlaisuus säilytettäisiin mahdollisimman pitkälle.

Ero 15 **Käyttöliittymän muokattavuus**
Arvio tarpeesta 2002: Otetaan käyttöön profiileihin liittyen.
Arvio nyt: **KYLLÄ.** Perustuu siihen, että käyttäjillä on erilaisia tarpeita ja käyttäjien vaikutelma vanhentumisesta perustuu pitkälti omaan käyttökokemukseen. Näin ollen mahdollisuus säätää käyttöliittymää niin että, omat työt sujuvat joustavammin, on erityisen tärkeää.

Ero 16 **Tiedot korteilta kuvan metadataksi**
Arvio tarpeesta 2002: Mielenkiintoinen. Selvitetään käyttömahdollisuus
Arvio nyt: **KYLLÄ.** Kuvissa oleva metadata seuraa materiaalin mukana järjestelmästä riippumatta ja on luettavissa esimerkiksi kuvankäsittelyohjelmassa. Mielestäni tämä ominaisuus olisi tuonut järjestelmän erityisratkaisusta yleistyökalujen joukkoon, pienellä mutta merkittävällä ominaisuudella.

Ero 17 **Hakutoiminteeseen mahdollisuus kohdentaa edellisiin hakuihin.**
Arvio tarpeesta 2002: Voisi olla myös HANSABASE Mediapankissa. Advanced Users -käyttöliittymässä
Arvio nyt: **KYLLÄ.** Tehokäyttäjän tarvitsema ominaisuus. Ei sovi peruskäyttäjän intuitiiviseen käyttöliittymään, mutta on tehokas lisämahdollisuus järjestelmän käytettävyyttä (vanhenemista) arvioivien tehokäyttäjien käyttöön.

Ero 18 **Tekstimuotoisen aineiston selailu ja hakutulosten havainnointi.**
Arvio tarpeesta 2002: Harkitaan myöhemmin.
Arvio nyt: **EI.** Ominaisuudella ei niin suurta merkitystä monimuotoisen digitaalisen aineiston hallinnassa. Enemmän toimitusjärjestelmien aputyökalu tai tekstien hallinnassa.

Ero 19 **Materiaalien keräily koriin, korin lataaminen ja lähetys toiselle käyttäjälle.**

Arvio tarpeesta 2002: Tämä tullaan todennäköisesti toteuttamaan HANSABASE Mediapankin omalla tekniikalla.

Arvio nyt: **KYLLÄ JA EI.** Tämä toteutettiin HANSABASEn omilla työkaluilla. Ominaisuus on todella näppärä ja siitä todellakin oli hyötyä palvelun kiinnostavuuden kannalta. Ominaisuuden toteuttaminen MediaVU:n työkaluin tosin tukisi sitä, että emojärjestelmän kanssa oltaisiin mahdollisimman pitkälle samanlaisia.

Ero 20 **Aineistokorttien tulostus ja aineistojen välitys sähköpostilla eteenpäin.**

Arvio tarpeesta 2002: Kannattanee toteuttaa.

Arvio nyt: **KYLLÄ JA EI.** Pienehköjä ominaisuuksia, mutta kannattaa toteuttaa, koska auttavat järjestelmän monipuolisessa käytössä, joka edesauttaa asiakkaan positiivista asennetta järjestelmän käytön jatkamiseen.

Ero 21 **Materiaalien käyttöönottoon liittyvä käyttöliittymän ja toimintojen profiilikohtainen konfigurointi.**

Arvio tarpeesta 2002: Otetaan käyttöön profiileihin liittyen.

Arvio nyt: **KYLLÄ.** Profiilit ja niihin liittyen käyttöliittymän konfigurointi mahdollistaa käyttöliittymän sovittamisen mahdollisimman hyvin käyttäjän omiin tarpeisiin. Erityisesti tehokäyttäjillä saadaan näin työtekoa joustavammaksi, joka taas tukee tärkeän käyttäjäryhmän positiivista suhtautumista järjestelmän käytön jatkamiseen.

Ero 22 **Tietokannan indeksin selailumahdollisuus**

Arvio tarpeesta 2002: Mietitään ominaisuuden toteuttamista HANSABASE Mediapankin ns. Advanced Users –käyttöliittymässä.

Arvio nyt: **KYLLÄ.** Tietokannan indeksin selailumahdollisuus on niin syvällinen ominaisuus, että se sopii parhaiten kovimpien tehokäyttäjien käyttöön. Heille siitä tosin voisi olla paljonkin iloa, joka taas voisi tukea tärkeän käyttäjäryhmän positiivista suhtautumista järjestelmän käytön jatkamiseen. Prioriteettimielessä tämä ei ole kovin tärkeä ominaisuus.

Ero 23 **Thesaurus-tyyppinen asia- ja synonymisanasto käyttöön**

Arvio tarpeesta 2002: Kannattaa tutustua tekniikkaan ja harkita käyttöönottoa.

Arvio nyt: **KYLLÄ**. Hieno ominaisuus kannan hakuominaisuuksien parantamiseksi. Mielestäni olisi toteutuessaan nostanut palvelun imagoa.

Ero 24 **Järjestelmän hajauttaminen**

Arvio tarpeesta 2002: Otettaneen mm. tietoturvasyistä käyttöön uudessa versiossa.

Arvio nyt: **KYLLÄ**. Tämä olisi ollut tärkeä ominaisuus esim. Nokian tarpeisiin, mutta muutenkin tietoturvan ja käytön katkeamattoman jatkumisen kannalta. Ominaisuus joka olisi tuonut HANSABASElle jatkovuosia.

Ero 25 **Tekstimuotoisen tiedon tallentaminen RTF -muodossa kantaan**

Arvio nyt: **EI**. HANSABASEssa puhtaan tekstimuotoisen datan säilyttäminen ei ole oleellista, vaan kyseessä on enemmän toimitusjärjestelmän ominaisuus. Eräs etu olisi tietenkin se, että emojärjestelmän kanssa samanlaisuus säilytettäisiin mahdollisimman pitkälle.

Ero 26 **Audio- ja videomateriaaleille automaattinen sormenpää.**

Arvio tarpeesta 2002: Selvitetään tekniikka ja mietitään käyttöönottoa.

Arvio nyt: **KYLLÄ.** Niiden puuttuminen HANSABASEsta oli ilman muuta puute. Puutteen korjaus on oleellista vanhenemien ehkäisylle tai ohjelman virkistämiselle.

Ero 27 **Print-on-demand ominaisuus.**

Arvio tarpeesta 2002: Voisi olla hyödyllinen ominaisuus. HansaBind ratkaisun liitäntää HANSABASE Mediapankkiin varastonseurannan osalta kannattaa harkita.

Arvio nyt: **KYLLÄ.** Painotalon aineistonhallintajärjestelmälle hyvinkin sopiva moduuli. Kätevä siis Print+ -palveluksi, joka tuottaa painotyötilauksia. Ei ehkä kovin merkittävä ominaisuus vanhenemisen estoa ajatellen.

Ero 28	**MediaFlow työnkulku.**
Arvio tarpeesta 2002:	Ehdottomasti tärkeä ominaisuus
Arvio nyt:	**KYLLÄ.** Oleellinen ominaisuus painotalon aineistonhallintajärjestelmälle. Integroisi HANSABASEn vielä paremmin Hansaprintin tuotantojärjestelmiin. Taustalla Seppo Haapakosken Hanska, joten konsepti oli alun perin lähtöisin Hansaprintistä.

Huomiona listalla olevista ominaisuuksista totean, että tarjolla oli useita erityisesti pääkäyttäjien ja ylläpitäjien työtä helpottavia ominaisuuksia. Pääkäyttäjä ja ylläpitäjä ovat erittäin merkittävässä roolissa järjestelmän toimivuutta ja hyödyllisyyttä arvioitaessa, joten ominaisuudet, jotka kuuluvat edellä mainittuun ryhmään, ovat varsin järkeviä toteuttaa.

Näistä ominaisuuksista osa toteutettiin ajan myötä HANSABASEen, mutta ei kaikkia. Kyseessä oli pääosin lista ohjelmiston lisäominaisuuksista, jotka toki olisivat nuorentaneet palvelua monipuolisuudellaan, mutta niiden toteuttaminen ei suoranaisesti tukenut ohjelmiston rakenteellista nuorentamista tai korjannut ohjelmoijien aiempia virheitä.

6 Pohdintaa

6.1 Taustaa

Tutkin tutkielmassani ohjelmistojen vanhenemiseen johtavia syitä, sekä mitä voitaisiin tehdä vanhenemisen hidastamiseksi tai peräti estämiseksi. Mallitapauksena käytettiin Hansaprintin aineistonhallintajärjestelmän HANSABASE Mediapankin vanhenemista. Mallitapaus taustoitettiin sekä käsitteiden että järjestelmän toiminnallisuuden ja toimintaympäristön osalta.

Tavoitteena oli etsiä vastauksia seuraaviin asetettuihin haasteisiin:

- H1: Mistä ohjelmistojen vanheneminen johtuu ja mikä siihen johtaa?
- H2: Mitä voisi tehdä ohjelmistojen vanhenemisen hidastamiseksi?
- H3: Voiko ohjelmistojen vanhenemisen estää kokonaan?

Hypoteesina kirjoitettaessa oli ajatus siitä, että ajanhammas söisi järjestelmän. Varmasti ajanhammaskin järjestelmää nakersi, mutta oliko se kuitenkaan tärkein vanhenemiseen vaikuttava tekijä. Mitkä muut kuin sisäiset voimat vaikuttivat ohjelmiston vanhenemiseen?

Tutkielmassa lähestyttiin isojen järjestelmien vanhenemista tarkastellen asiaa peilaten Parnasin ’Software Aging’ -teoriaan sekä ’Ketterän kehityksen’ -mallin (Agile) ohjelmistojen vanhenemista vastaan kehitettyihin menetelmiin.

Tutkielman luvussa ’5. Näkökulmat tilanteeseen’ käytiin läpi HANSABASE-ohjelmiston vanhenemista kolmen näkökulman kautta. Ensin mietittiin vanhenemisen syitä tarkastellen tilannetta vallinneen liiketoimintaympäristön näkökulmasta. Analyysi perustui pitkälti omalle käsitykselleni tapahtuneesta. Toiseksi analysoitiin tapahtunutta käyttäen näkökulmana tutkielman luvussa 2 esille tuotua ’Software Aging’ -teoriaa. Miten eri kohdat Parnasin listasta osuivat HANSABASE kehitykseen ja miten niiden noudattamisesta olisi ollut hyötyä? Kolmanneksi mietittiin mitä ohjelmiston vanhenemiseen olisi vaikuttanut 2000-luvun alussa tehdyn HANSABASE-kehityssuunnitelman toteuttaminen aikanaan. Kohta kohdalta kävin läpi mikä ominaisuuksista olisi mielestäni auttanut järjestelmän vanhenemista vastaan enemmän, mikä vähemmän.

Lopputuloksia tarkastelen ensin liiketoimintaympäristön kannalta. Mikä merkitys oli HANSABASEn lisäarvopalvelun luonteella? Entä mitä vaikuttivat yksittäisiin asiakastarpeisiin kehitetyt lisäominaisuudet? Tukivatko ne palvelun pitkäikäisyyttä, vai johtivatko nopeampaan vanhenemiseen?

Seuraavaksi lähestyn vanhenemiskysymystä Parnasin ’Software Aging’ -teorian kautta nostaen esille kohdat, joita noudattaen olisi vanhenemista mielestäni saanut eniten hidastettua. Sitten tarkastelen mitkä kehityskohdat 2000-luvun alussa tehdyn HANSABASE-kehityssuunnitelman työlistasta olisivat mielestäni eniten auttaneet järjestelmän vanhenemista vastaan.

Vielä yhteenvetona esitän tarkastelun pohjalta muodostuneen käsityksen järjestelmän vanhenemiseen vaikuttaneista syistä, joita oli paljon enemmän kuin pelkkä ajan hampaan vaikutus ohjelmakoodiin.

6.2 HANSABASEn erilaisen käytön merkitys vanhenemiselle

HANSABASE Mediapankkia käytettiin eri asiakkaiden tapauksissa kuvapankkina tai jopa mediapankkina, mutta myös erittäin moneen erityiskäyttötarkoitukseen. Johtiko tämä siihen, että HANSABASE nähtiin helpommin vanhenevana järjestelmänä? Entä vanhensiko se jo sinänsä järjestelmää?

Asiakkaiden tarpeiden toteuttamisen yhteydessä erityisominaisuuksia ei viety automaattisesti kaikkien käyttöön. Tämä johti selkeästi järjestelmän ylläpidon hankaloitumiseen, kun koodi silppuuntui. Toki tätä yritettiin hallita, mutta ei riittävän voimakkaasti. Erityisominaisuudet siis vanhensivat palvelua.

Näkemykseni mukaan myös se, että asiakkailla oli hyvin erilaisia käsityksiä HANSABASEn roolista, johti monen kohdalla siihen, että HANSABASE koettiin rajatuksi työkaluksi eikä nähty mahdollisuutta ottaa palvelua laajempaan aineistonhallintakäyttöön. Näin ollen myös käsitetasolla HANSABASE vanheni.

6.3 ’Software Agingin’ parhaat työkalut HANSABASEn vanhenemista vastaan

Tässä kappaleessa nostan esille Parnasin ’Software Aging’ -teorian työkalut ja menetelmät, joita noudattaen olisi järjestelmän vanhenemista mielestäni saatut eniten hidastettua.

Heti aluksi on todettava, että jos Parnasin ohjelmistojen vanhenemisteoria olisi ollut Hansaprintissä käytössä, kun HANSABASE -prosessiin lähdettiin, ja sitä olisi noudatettu alusta alkaen sekä koko ohjelmiston elinkaaren aikana, olisi lopputulos ollut ohjelmakoodin kannalta merkittävästi parempi, mikä osaltaan olisi auttanut palvelun pysymisessä validina.

Nostan tässä esiin muutaman kohdan, jotka tuntuvat osuvimmilta kuvaamaan HANSABASEn tilannetta sekä työkaluina tehokkaimmilta tilanteen parantamisen kannalta.

i. Heikkenemisen pysäyttäminen (5.2.13, 2.5.1)

Kun järjestelmässä havaitaan laadullista heikkenemistä, kannattaa tarttua välittömästi korjaaviin toimenpiteisiin: selvittää mistä on kysymys, suunnitella tarvittavat korjaukset ja toteuttaa ne. Tämä vie toki aikaa, mutta säästää tulevilta isommilta hankaluuksilta. Vanhenemisen kannalta on helpompi tehdä korjaukset heti, kuin vasta sen jälkeen, kun väärin tai tehottomasti toimivan koodin päälle on ohjelmoitu kerros tai useampi lisäkoodia. HANSABASE oli siinä tilanteessa, että ohjelmakoodi oli heikentynyt jo pidemmän aikaa. Heikkenemisen pysäyttäminen oli siis erittäin tärkeää!

ii. Laajan ohjelmakoodin leikkaus ja uudelleen järjestely (5.2.17, 2.5.5)
Modularisointi jälkikäteen (5.2.15, 2.5.3)

Erityisesti siinä tilanteessa, kun ohjelmaa huolletaan tai siihen kehitetään uusia ominaisuuksia, kannattaa kerätä saman tehtäväalueen toimintoja yhteen ja näin selkeyttää ohjelmakoodia. Kun koodi on päässyt monen korjauskerran aikana pahasti rappeutumaan, kannattaa käydä se tarkkaan läpi ja kerätä havaitut saman tyyppiset osiot omiin moduuleihinsa. Tämän tyyppinen leikkaushoito ja modularisointi olisi ollut HANSABASE -ohjelmakoodille erityisen virkistävää.

iii. Parempaa kirjanpitoa, eli dokumentoinnin merkitys (5.2.9, 2.4.2)
Dokumentointi jälkikäteen (5.2.14, 2.5.2, 2.5.3)

Vaikka dokumentointi saattaa tuntua raskaalta ja tuottamattomalta työltä, on se ensiarvoisen tärkeää, varsinkin isojen ja toivottavasti pitkäikäisten ohjelmistojen hallittavuuden kehittämisessä. Muista myös valvoa, että dokumentaatio tehdään sääntöjen mukaisesti. Kun muutostarpeita tulee, tai pitää korjata virheitä, on hyvästä dokumentaatiosta mittaamatonta hyötyä. Kun havaitaan, että dokumentaatiota ei ole, se on jäänyt jälkeen tai toteutettu huolimattomasti, on syytä heti tarttua tehtävään. Samalla ohjelmakoodi tulee mietittyä läpi ja mahdolliset anomaliat ja kertyneet turhat koodit saadaan korjattua. HANSABASEn osalta tästä olisi ollut todella suuri hyöty.

iv. Ulkopuoliset katselmukset (5.2.10, 2.4.3)

Ulkopuolisilla katselmuksilla saadaan varmistettua ja valvottua, että sovittuja käytäntöjä noudatetaan sekä tarkistettua tehty suunnittelu- ja ohjelmointityö ennen kuin päädytään tilanteeseen, johon ei haluttaisi joutua. Olisi oleellisen tärkeää, että yrityksen ohjelmistokehitysprosessiin kuuluisi katselmointi ja hyväksyntä vaiheet, jotka suorittaisivat erityisesti tuotteen pitkän aikavälin tulevaisuudesta vastaava henkilö.

Kuten kappaleen alussa totesin, kaikki Parnasin teoriassa esitellyt asiat, niiden ymmärtäminen, sekä esiteltyjen työkalujen käyttö edesauttavat vanhenemisen hidastamisessa ja myös mahdollisesti jo tapahtuneen vanhenemisen korjaamisessa. Valitsin edellä luetellut kohdat siksi, että koin, että erityisesti ne olisivat tukeneet HANSABASEn kehitystä paremmin vanhenemista kestäväksi ohjelmistoksi.

6.4 HANSABASEn kehityssuunnitelman parhaat työkalut vanhenemista vastaan

Kappaleessa 5.3. pohdittiin 2000-luvun alussa tehdyssä HANSABASE -kehityssuunnitelmassa listattujen ominaisuuksien vaikutusta HANSABASEn vanhenemista vastaan. Seuraavassa poimin noista ominaisuuksista mielestäni kolme tärkeintä. Monen muunkin näistä olisin voinut listalle kelpuuttaa, mutta nämä koin tarpeellisimmiksi. Lisäksi olen järjestänyt ominaisuudet mieleiseeni tärkeysjärjestykseen.

Ero 21- Materiaalien käyttöönottoon liittyvä käyttöliittymän ja toimintojen profiilikohtainen konfigurointi. Kaikki materiaalin käyttöönottoon liittyvät valinnat ja vaihtoehdot olisivat profiloitavissa käyttäjäryhmän tarpeiden mukaan, esim. työasemaan kopioitavan kuvan koko ja tiedostomuoto. Käyttäjä voi myös itse valita, mitkä useimmin tarvittavista toimenpiteistä ovat näkyvissä ns. pikanappeina, jolloin esim. kuvan siirto työasemalle voidaan tehdä suoraan yhden napin painalluksella ilman koritoimintoa. Profiilit ja niihin liittyen käyttöliittymän konfigurointi mahdollistaa käyttöliittymän sovittamisen mahdollisimman hyvin käyttäjän omiin tarpeisiin. Erityisesti tehokäyttäjillä saadaan näin työtekoa joustavammaksi, joka taas tukee tärkeän käyttäjäryhmän positiivista suhtautumista järjestelmän käytön jatkamiseen.

Ero 15 - Käyttöliittymän muokattavuus. Käyttäjän omassa profiilissa voidaan määritellä mm. käyttöliittymän kieli ja omat pikavalintanapit. Ominaisuuden merkittävyys perustuu siihen, että käyttäjillä on erilaisia tarpeita ja käyttäjien vaikutelma vanhentumisesta perustuu pitkälti

omaan käyttökokemukseen. Näin ollen mahdollisuus säätää käyttöliittymää niin, että omat työt sujuvat joustavammin, on erityisen tärkeää.

Muokattavuus monella tasolla tuo käyttöliittymässä usein tehtävien toimenpiteiden tekemiseen tehokkuutta ja käyttäjille lisää käytön mukavuutta. Käyttäjät, ja erityisesti tehokäyttäjät ja kantojen ylläpitäjät ovat tärkeitä mielipidevaikuttajia, kun ollaan tilanteessa, jossa mietitään mahdollisuutta vaihtaa järjestelmä toiseen järjestelmään. Heihin kannattaa siis panostaa.

Ero 24 - Järjestelmän hajauttaminen. GWX järjestelmässä oli mahdollista hajauttaa järjestelmän toimia useille palvelimille. Tämä olisi ollut tärkeä ominaisuus esim. Nokian tarpeisiin, mutta muutenkin tietoturvan ja käytön katkeamattoman jatkumisen kannalta. Ominaisuus joka olisi tuonut HANSABASElle jatkovuosia.

Hyödyllisyydestään huolimatta nämä olivat vain palvelun ominaisuuksia, jotka eivät ratkaisisi ohjelmiston syvempiä ongelmia. Näiden käyttöönotto pitäisi toteuttaa noudattaen Parnasin teorian menettelyjä ja työkaluja.

6.5 Yhteenveto vanhenemiseen vaikuttaneista syistä

Tarkastelun pohjalta syntyneen käsityksen mukaan niin HANSABASEn vanhenemiseen, kuin muidenkin isojen ohjelmistojen, vaikuttaa moni muukin tekijä kuin pelkästään ohjelmakoodin kuriton tai kurinalainen toteuttaminen, dokumentointi ja ohjelmiston asiallinen hoito. Vaikuttavia tekijöitä on niin sisäisiä kuin ulkoisia.

Ohjelmiston vanhenemiseen ovat vaikuttaneet ytimessä olevan ohjelmoinnin menetelmien noudattamisen lisäksi muun muassa seuraavat asiat:

1. Omistajan valinnat, joita ovat
    - Johdon painotukset, joita saattavat ohjata jotkin täysin muut kuin ohjelmiston vaatimukset. Esimerkiksi tuotannollisten paineiden siirtyminen johonkin täysin toiseen tuotealueeseen.
    - Projektille allokoidut resurssit, jotka voivat vaihdella taloudelliseen tilanteeseen liittyen, mutta myös muista syistä, esim. painopisteen siirtyessä tuotteista toisiin. IT-firmoissa johdossa mukana on IT-johtoa, joten näkökulma on toinen kuin esim. painotalolla, jonka liiketoiminnan painopiste on painotuotteiden tuottaminen, jossa IT-palvelut ovat vain työkaluja tai korkeintaan lisäarvopalveluja.
    - Muutokset asiakunnassa

2. Tekniikasta vastanneiden valinnat
    - Teknologiavalinnat, eli millaiset palvelimet, levyjärjestelmät, internet-yhteydet jne. on hankittu yrityksen ja palvelun käyttöön
    - Menetelmävalinnat, eli mitä työkaluja käytetään, milloin käytetään uusia työkaluja menetelmiä, milloin pitäydytään vanhassa. Otetaanko esim. ketterän kehityksen menetelmät prosessiin mukaan, tunnetaanko Parnasin teoria ja noudatetaanko sen antamia ohjeita kehityksessä.

3. Emojärjestelmän kehitys ja valinnat
    - HANSABASEn tilanteessa järjestelmä perustui ulkopuolisen emojärjestelmän sovittamiseen ja muokkaamiseen haluttuun tarkoitukseen. Kun emojärjestelmälle tehtiin kehitystoimenpiteitä, piti tehdä päätös muutosten tuomisesta HANSABASEen tai tuomisen lykkäämisestä, tai niistä luopumisesta. Emojärjestelmän kehitys siis ohjasi HANSABASEn kehitystä, mutta viiveellä. Myös Hansaprintin IT -johdon päätökset ohjasivat emojärjestelmän ominaisuuksien käyttöönottoa.

4. Teknisen henkilökunnan osaaminen
    - Luonnollisestikin teknisen henkilökunnan osaamistaso on yksi merkittävistä tekijöistä. Jos tietoa vanhenemisen viivästämiseen ja korjaamiseen käytettävistä menetelmistä ei ole, ei niitä myöskään voi käyttää. Tai, jos ohjelmointitekniikoiden osaaminen ei ole riittävän korkeaa tasoa, ei lopputuloskaan ole häävi.

5. Asiakkaiden valinnat
    - Asiakkaiden tarpeet ovat kaiken lähtökohta, eli jos ne eivät kohtaa, tai kasvavat ohi järjestelmältä tarjolla olevien ominaisuuksien, koetaan järjestelmän olevan vanhentunut.
    - Asiakkaiden omat sisäpoliittiset valinnat saattavat johtaa esim. painopisteen siirtymiseen pois järjestelmän suunnalta. Esimerkiksi arveluni Nokialla tapahtuneista muutoksista ja niiden syistä kappaleessa 5.1.2.
    - Asiakkaiden teknologialinjaukset. Esimerkiksi kansainvälisen yhtiön tietohallinnossa tehdään linjanvetoja, joihin palvelun konsepti ei enää istu, tai siitä tulee oma erillinen saarekkeensa. Tällöin syntyy tilanne, jota ei kovin pitkään asiakkaan hallinnossa suvaita. Ohjelmisto siis vanhenee asiakkaan silmissä, vaikka mitään ei varsinaisesti tapahdukaan itse ohjelmiston kannalta.

6. Kilpailijat
    - Perinteisten kilpailijoiden rinnalle tulee uusia kilpailijoita rinnakkaisilta tai täysin muilta ratkaisualueilta. Tällöin toiselta ratkaisualueelta tulevan kilpailijan muut ominaisuudet saattavat muuttua vahvuuksiksi, joiden kanssa kilpaileminen on haastavaa. Esimerkiksi aineistonhallinnan sisällyttäminen puhtaan dokumentinhallinnan alueelta tulevaan tuotteeseen.

7. ”Maailman muuttuminen”
    - Käsitteistön muuttuminen, laajeneminen. Ajan kuluessa markkinoilla ja teknologioissa tapahtuu, syntyy kokonaan uusia alueita, vanhat laajenevat yhteen suuntaan ja suppenevat toiseen suuntaan. Se mitä ennen käsitettiin termin merkitsevän, ei välttämättä enää tänään katakaan termillä tarkoitettua aluetta. Esimerkiksi kattava ja monipuolinen aineistohallintajärjestelmä 2000-luvun alusta, ei enää ehkä olisikaan sitä 2010-luvun lopussa.

7 Loppupäätelmät

Ohjelmistojen vanhenemiseen vaikuttavien tekijöiden laaja-alaisuutta kuvaamaan esitän vielä lopuksi vanhenemisen sipulimallin sekä mietin pitäisikö ’Software Aging’-teoriaa laajentaa poikkitieteelliseksi.

7.1 Vanhenemisen sipulimalli

Ohjelmistojen vanhenemiseen vaikuttavien tekijöiden kokonaisuus voidaan kuvata havainnollisena sipulimaisena rakenteena, jonka ytimessä on idea ohjelmistosta tai palvelusta. Tuon kovan ytimen ympäriltä löytyvät muut ohjelmiston tai palvelun vanhenemiseen vaikuttavat kerrokset.

Kuva 7.1. Vanhenemisen sipulimalli. Ohjelmiston tai palvelun vanhenemiseen vaikuttavat kerrokset (Joon Boucht, 2018). Onion-diagrammin mallipohja Free Industry Powerpoint Templates[5] -palvelusta.

i. ’Ohjelmisto- tai palveluidea’, ydin

Ohjelmiston tai palvelun kehittäminen lähtee liike- tai toimintaideasta, joka evaluoidaan tarkkaan monelta näkökulmalta, jonka jälkeen tehdään päätökset mahdolliseen

[5] http://www.free-powerpoint-templates-design.com/

ohjelmaprojektiin lähtemisestä. Idea on koko kuvion syvimpänä ytimenä. Mikäli se ei kestä evaluointia, ei muilla kerroksilla ole merkitystä.

ii. ’Ohjelman toteutus ja ylläpito’ -kerros

Seuraava kerros pitää sisällään ohjelmiston tai palvelun varsinaisen suunnittelun ja toteutuksen. Tässä kerroksessa on oleellista, että tunnetaan ja noudatetaan Parnasin ”Software aging”- teorian suosituksia, vältetään kuvattuja sudenkuoppia, ja otetaan käyttöön ketterän kehityksen työkalut vanhenemisen hallintaan. Näitä samoja ohjeita on syytä noudattaa myös kaikessa huolto- ja korjaustyössä. Muussa tapauksessa saatetaan ohjelmisto altistaa sitä vanhentaville tekijöille liian helposti heti alusta alkaen.

iii. ’Tekniset valinnat’ ja ’Valinnaiset ominaisuudet’ - kerros

Kolmas kerros pitää sisällään yrityksen IT-johdon tekemät tekniset valinnat sekä emojärjestelmän ominaisuuksien valinnat. Tekniset valinnat sisältävät sekä laitteistoihin ja yhteyksiin että ohjelmointitekniikoihin ja -alustoihin liittyvät valinnat.

Valinnaiset ominaisuudet viittaavat ohjelmiston emojärjestelmässä tehtyihin valintoihin ja linjanvetoihin sekä itse ohjelmistossa tehtyihin valintoihin mukaan otettavista emojärjestelmän ominaisuuksista tai piirteistä. Ulkoiset tekijät vaikuttavat siis jo tälläkin tasolla vanhenemiseen. Jos esimerkiksi emojärjestelmää päätetään kehittää ja siirtää se vaikka uudelle alustalle, eikä tätä muutosta kuitenkaan haluta tehdä ytimessä olevalle ohjelmistolle, voi ohjelma vanheta nopeastikin. Parnaksen periaatteita on noudatettava myös tällä tasolla.

iv. ’Omistajan valinnat’ ja ’Asiakkaan valinnat’ - kerros

Omistajan sekä asiakkaiden tekemät valinnat ja päätökset ovat myös erittäin suuressa roolissa ohjelmiston vanhenemisen kannalta. Omistajan valinnoista esimerkiksi johdon painotukset ja niiden muuttuminen, vaikka toisen bisnes-alueen hyväksi, tai projektille allokoitujen resurssien määrän muuttuminen voivat joko edes auttaa ohjelmiston pysymistä elinvoimaisena tai vastaavasti vanhentaa sitä radikaalisti.

Myös asiakkaiden tekemät valinnat ja linjaukset ovat merkittäviä: asiakkaan tarpeet ja palveluntuottajan mahdollisuudet niiden täyttämiseen määräävät vanhenemisen suunnan. Myös asiakkaan tekemät sisäpoliittiset valinnat, esimerkiksi ”suosi kotimaista” tai

”keskitetään ostetun palvelun toimijat Englantiin” voivat välillisesti vaikuttaa palvelun tai ohjelmiston vanhenemiseen. Samoin asiakkaan teknologialinjaukset kuten ”elämme täysin Macintosh ympäristössä” tai ”käyttämämme palvelut replikoituvat maailmalaajuisesti” saattavat tukea tai heikentää palvelutarjoajan ohjelmiston tulevaisuutta, eli nuorentaa tai vanhentaa välillisesti palveluntarjoajan ohjelmistoja.

v. ’Kilpailijat’ ja ’Maailman muuttuminen’ - kerros

Maailma muuttuu. Käsitteiden merkitykset muuttuvat ja niiden kattamat alat laajenevat. Kun arvioidaan palvelun tai ohjelmiston ajankohtaisuutta, verrattaan sen toimintoja ja ominaisuuksia siihen, mitä käsitteen piiriin kuuluu tarkasteluhetkellä. Teknologiat vanhenevat, vanhat menettävät merkitystään ja uusia syntyy. Se, mikä tänään on uutta ja hienoa, on mahdollisesti jo huomenna vanhentunutta. Ohjelma, joka on täydellisesti täyttänyt asiakkaidensa tarpeet aikanaan, voi myöhemmin olla toiminnoiltaan hyvinkin vajaa, vaikka sinänsä olisi toteutukseltaan aivan yhtä hyvä, kuin silloin kun ohjelmisto julkaistiin.

Maailman muuttuessa myös bisnesalueiden rajat hämärtyvät. Tuotteisiin otetaan ominaisuuksia rinnakkaisilta sektoreilta. Näin ilmaantuu uusia kilpailijoita, joilla on omat vahvuutensa. Tämä kaikki vanhentaa ohjelmistoa asiakaskunnan silmissä.

Tässä esitellyistä kerroksista Parnaksen teorialla katetaan vain tasot 2 ja 3. Tasot 4 ja 5 ovat järjestelmälle ulkoisia tekijöitä.

7.2 Loppusanat

HANSABASE -järjestelmän vanhenemiseen vaikutti moni tekijä, ei pelkästään ohjelmakoodin vanheneminen. Suurimmassa osassa olivat ulkoiset tekijät, tärkeimmän asiakkaan tekemät valinnat ja palvelusta luopuminen, mutta myös yrityksen johdon päätökset resurssoinnista ja painotuksista. Sidosryhmillä ja ympäristöllä on siis iso merkitys ohjelmistojen vanhenemisessa!

Mutta koska sipulin ydin ja sen oikeanlainen toteuttaminen ovat ensiarvoisen tärkeitä, pidän Parnasin ’Software Aging’ -teoriaan tutustumista sekä antoisana että suositeltavana. Se valmistelee suurien kokonaisuuksien suunnittelijaa koodin väistämättömään vanhenemiseen tarjoten työkaluja vanhenemisen hidastamiseen sekä jo vanhentumisesta kärsivän koodin virkistämiseen.

Tutkielmassa tehtyjen havaintojen perusteella näen ’Software Aging’ -teorian merkityksen suurena. Olisi erittäin suositeltavaa, että sen opiskelu kuuluisi opinto-ohjelmaan kaikissa ohjelmointia opettavissa koulutuslaitoksissa.

Edellä esitetty herättää myös kysymyksen, että pitäisikö ’Software Aging’ -teoriaa laajentaa tuomalla siihen mukaan poikkitieteellisesti myös esim. talous- ja käyttäytymistieteellistä näkökulmaa? Nykyinen alan kirjallisuus tuntuu olevan kovin vajaa muiden huomioiden osalta.

Tutkielman kirjoittajan toiveena on, että tässä tutkielmassa saavutettuja tutkimustuloksia syvennettäisiin tulevaisuudessa uusilla tutkimuksilla. Sipulimallia ei ole pureskeltu vielä loppuun asti ja esimerkiksi yhteyksiä Lean Startup ajatteluun voisi kartoittaa. Mutta ne ovatkin sitten uuden tarinan aiheita.

Lähteet

(ADOBE 2001) Adobe product description, XMP (The eXtensible Metadata Platform) http://www.adobe.com/products/xmp/main.html Nykyinen: https://www.adobe.com/products/xmp.html

(Adobe WP 2001) Adobe White Papers 2018, https://www.adobe.com/products/xmp/white-papers.html

(Agile Alliance 2017) Agile Alliance (2017), Agile Glossary, https://www.agilealliance.org/agile101/agile-glossary/

(Andansare 2014) Andansare, Nita (2014, November), ”Managing Technical Debt with Agile”, http://www.agilerecord.com/managing-technical-debt-agile/, Agile Record

(ATF 2018) Agile Testing Framework (31.1.2018) https://www.agiletestingframework.com/atf/testing/test-driven-development-tdd/

(Boucht 2000) Boucht, Joon, 2000, Asset Management in Drupa 2000, DAM_in_drupa.v.0.8.doc

(Boucht 2001) Boucht, Joon, HANSABASE Mediapankki: palvelukuvaus, v.1.3, HB_palvelukuvaus.v.1.3.doc, (31.5.2001)

(BRS 2002) Open Text: BRS/Search, http://www.opentext.com/dataware/brs_search.html

(Cisco etc 1999) Cisco, Susan L. - Strong, Karen V. (1999) The value added information chain. The information management journal, Vol 33, No 1, 4-15.

(DIG35 2000) DIG35 (2000), http://www.i3a.org/i_dig35.html Nykyinen: DIG35 Specification Version 1.1: www.bgbm.fu-berlin.de/TDWG/acc/Documents/DIG35-v1.1WD-010416.pdf

(Dyck 2001) Dyck, Timothy, XML Database Doubts, eWEEK, December 3, 2001 http://www.eweek.com / (http://www.eweek.com/article/0,3658,s%253D709%2526a%253D19372,00.asp)

(Fowler 2009) Fowler, Martin (14.10.2009, reposted 19.11.2014) https://martinfowler.com/bliki/TechnicalDebtQuadrant.html

(Gistics 2017) https://gistics.com/ ja https://gistics.com/why-gistics/, (15.10.2017)

(Hansaprint FK 1998) Hansaprint Oy, Finnet kuvapankin käyttöohje, 1998

(Hansaprint HB 1998) Hansaprint Oy, HANSABASE -esite, 1998

(Hansaprint HB 2001) Hansaprint Oy, HANSABASE -esite, 2001

(Hansaprint HB 2004) Hansaprint Oy, HANSABASE -esite, 2004

(Hansaprint TW 1998) Hansaprint Oy, TeamWARE Mediabank käyttöohje, 1998

(Hansaprint NMB 1998) Hansaprint Oy, Nokia Mediabank käyttöohje, 1998

(Holvitie ym 2016) Holvitie J., Licorish S.A., Martini A. & Leppänen V (2016) Co-Existence of the ‘Technical Debt’ and ‘Software Legacy’ Concepts”, Johannes Holvitie, Sherlock A. Licorish, Antonio Martini, and Ville Leppänen

(IPTC 2001) IPTC org (2001), http://www.iptc.org/site/standards.html#other Nykyinen: https://iptc.org/standards/photo-metadata/iptc-standard/

(Ixiasoft 2002) Ixiasoft Inc.: TextML Server, http://www.ixiasoft.com/products/textmlserver/index.asp

(JDF 2001) CIP4 organisaation määrittelemä Job Definition Format, http://www.cip4.org/ Nykyinen: https://www.cip4.org/what-is-jdf.html

(MiniSQL 2001) Hughes Technologies Pty Ltd.: MiniSQL, http://www.hughes.com.au/products/msql/

(MySQL 2001) MySQL AB: MySQL Database Server, http://www.mysql.com/

(Leinonen 2000) Leinonen Heikki, Maria/MediaVu –tuotekuvaus, v1.7, mediavu17_white_fin_2P.doc

(Leinonen 2001) Leinonen Heikki, MediaVu - Materiaalinhallintajärjestelmä, PowerPoint -esitys, MariaPitkaFinnGraf.ppt, FinnGraf messut 2001

(Nurmi ym. 2001) Nurmi, Olli, Antikainen Hannele, Siltanen Olli (2001, toukokuu). Digitaalinen aineistonhallinta, GT –raportti, nro 2, toukokuu 2001

(Mountaingoat 2018) https://www.mountaingoatsoftware.com/agile/user-stories, (22.4.2018)

(Oracle Text 2002) Oracle Text (formerly interMedia Text), Oracle Technology Network http://otn.oracle.com/products/text/content.html

(Oracle 2018) Oracle Text Application Developers Guide, https://docs.oracle.com/cd/B19306_01/text.102/b14217/title.htm

(Parnas 1994) Parnas, D. L. (1994, May). Software aging. In Proceedings of the 16th international conference on Software engineering (pp. 279-287). IEEE Computer Society Press

(PRISM 2001) Publishing Requirements for Industry Standard Metadata, http://www.prismstandard.org/ Nykyinen: https://www.idealliance.org/prism-metadata

(RDF 2001) W3 Resource Description Framework, http://www.w3.org/RDF Nykyinen: https://www.adobe.com/products/xmp.html

(Tamino 2002) Software AG: Tamino XML Server, http://www.softwareag.com/tamino/

(Three Beacons 2011) http://www.threebeacons.com/

(Virtanen 2001) Virtanen Jani, Dokumentinhallinnan ratkaisut - case: Hansaprint oy, Pro Gradu-tutkielma, TuKKK, 2001

(XMP 2001) XMP Software Development Kit, http://partners.adobe.com/asn/developer/xmp/main.html Nykyinen: https://www.adobe.com/products/xmp.html

(XML-RPC 2002) http://www.xmlrpc.com/ (5.6.2018)