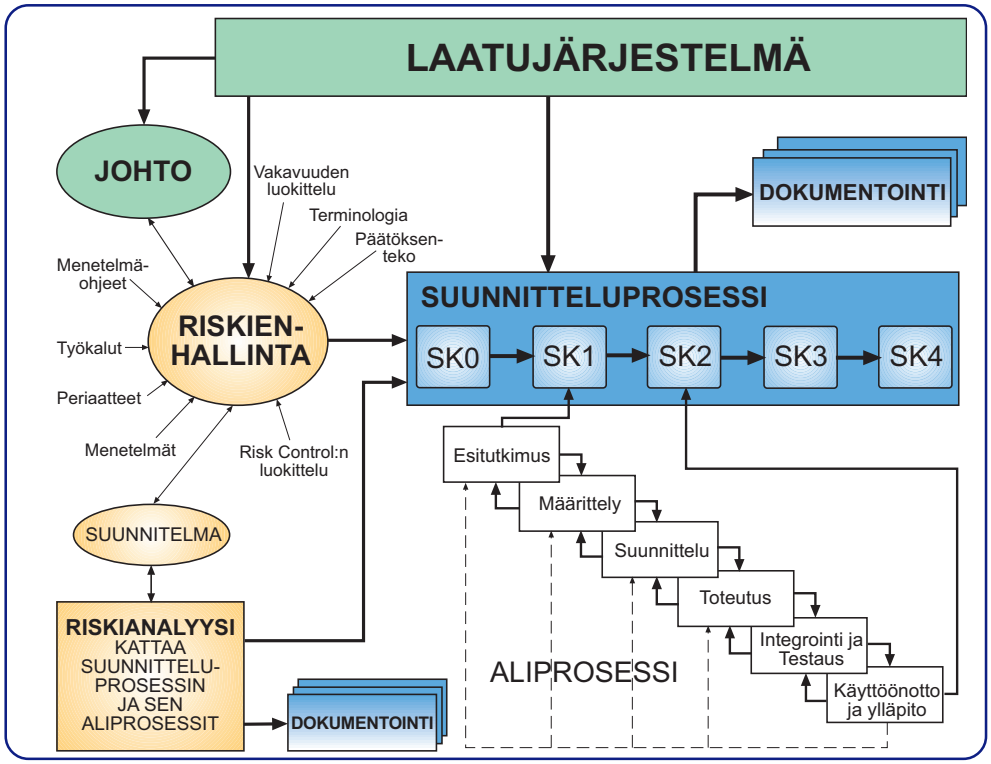


Ilpo Pöyhönen, Kaarle Kylmälä, Hannu Harju,
Pia Kemppainen-Kajola, Kalle Kuhakoski,
Greig Spankie & Olli Ventä

Vaatimukset ohjelmistoa sisältäville lääkintälaitteille

Hallinta ja menetelmät vaatimustenmukaisuuden osoittamiseksi

VTT TIEDOTTEITA – MEDDELANDEN – RESEARCH NOTES 2150

Vaatimukset ohjelmistoa sisältäville lääkintälaitteille

Hallinta ja menetelmät vaatimusten-mukaisuuden osoittamiseksi

Ilpo Pöyhönen, Kaarle Kylmälä, Hannu Harju,
Pia Kemppainen-Kajola, Kalle Kuhakoski,
Greig Spankie & Olli Ventä

VTT Tuotteet ja tuotanto

ISBN 951–38–6060–4 (nid.)
ISSN 1235–0605 (nid.)

ISBN 951–38–6061–2 (URL: http://www.inf.vtt.fi/pdf/)
ISSN 1455–0865 (URL: http://www.inf.vtt.fi/pdf/)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

VTT, Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Tuotteet ja tuotanto, Tekniikankatu 1, PL 1306, 33101 TAMPERE
puh. vaihde (03) 316 3111, faksi (03) 316 3282, (03) 316 3499, (03) 316 3493

VTT Industriella system, Tekniikankatu 1, PB 1306, 33101 TAMMERFORS
tel. växel (03) 316 3111, fax (03) 316 3282, (03) 316 3499, (03) 316 3493

VTT Industrial systems, Tekniikankatu 1, P.O.Box 1306, FIN–33101 TAMPERE, Finland phone internat.
+ 358 3 316 3111, fax + 358 3 316 3282, + 358 3 316 3499, + 358 3 316 3493

VTT Tuotteet ja tuotanto, Tekniikantie 12, PL 1301, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 6752

VTT Industriella system, Teknikvägen 12, PB 1301, 02044 VTT
tel. växel (09) 4561, fax (09) 456 6752

VTT Industrial system, Tekniikantie 12, P.O.Box 1301, FIN–02044 VTT, Finland phone internat. + 358 9
4561, fax + 358 9 456 6752

Otamedia Oy, Espoo 2002

Pöyhönen, Ilpo, Kylmälä, Kaarle, Harju, Hannu, Kemppainen-Kajola, Pia, Kuhakoski, Kalle, Spankie, Greig & Ventä, Olli. Vaatimukset ohjelmistoa sisältäville lääkintälaitteille. Hallinta ja menetelmät vaatimustenmukaisuuden osoittamiseksi [Requirements for medical device software]. Espoo 2002. VTT Tiedotteita – Research Notes 2150. 135 s. + liitt. 40 s.

Avainsanat medical device software, medical devices, medical systems, risk management, risk analysis, manufacturing process assessment

Tiivistelmä

Terveydenhuollossa käytettävien laitteiden ja järjestelmien suorituskyky ja monimutkaisuus lisääntyvät vuosi vuodelta. Osaltaan muutokset johtuvat teknologisista muutoksista, mutta myös potilaan hoitomenetelmät kehittyvät ja lisäävät muutospaineita laitteiden suorituskyvylle ja turvallisuudelle. Hyvin usein myös hoito- ja tutkimusmenetelmien tukena käytettävät järjestelmät sisältävät tietokoneen tai palvelimen, käyttöjärjestelmän ja laajan joukon erilaisia sovellusohjelmia ja ajureita.

Näiden teknologisten ja itse hoitomenetelmissä tapahtuneiden muutosten seurauksena ohjelmistojen osuus lääkintälaitteissa ja lääkintälaitejärjestelmissä on kasvamassa. Lääkintälaitteet ja järjestelmät on ennen markkinoille saattamista hyväksytettävä eri markkina-alueilla voimaan saatettujen säädösten mukaisesti. Mikäli markkinoille saattamisen aikana huomataan, että tuote ei täytä sille asetettuja standardien tai säädösten vaatimuksia, niin tästä voi aiheutua huomattavia lisäkustannuksia yritykselle johtuen uudesta tuotekehitysjaksoista sekä uusista varmennustesteistä. Viivästyksillä, mahdollisista jälkitoimenpiteillä tai jopa markkinoilta poisvetämisellä on myös kielteinen vaikutus ostajan mielikuvaan yrityksen ja sen tuotteiden luotettavuudesta ja laadusta.

Koska ohjelmistopohjaisten laitteiden ja järjestelmien luotettavuuden, turvallisuuden ja suorituskyvyn, s.o. vaatimustenmukaisuuden osoittaminen on itse valmiista tuotteessa vaikeaa, tässä julkaisussa on erityisesti käsitelty arviointia tuotantoprosessissa.

Tässä julkaisussa yhdistetään EU:n ja FDA:n asettamat vaatimukset lääkintälaitteiden ohjelmistoille sekä esitetään vaatimustenmukaisuuden osoittamismalli. Julkaisussa käsitellään laajasti myös riskienhallintaprosessia ja -menetelmiä, joilla pystytään riskien tunnistamisen ja arvioimisen ohella myös tehostamaan kehitysprosessia ja varmistamaan tuotteen markkinoillesaattamisen onnistuminen. Julkaisu on tarkoitettu lääkintälaitteiden valmistajille ohjeistamaan ohjelmistotuotannon eri osa-alueita tavoitteena yhdistää tuoteriskien hallinta ja riskianalyysit kiinteäksi osaksi suunnittelua. Tällöin suunnitteluprosessi tuottaa turvallisempia ja luotettavampia tuotteita lyhentäen tuotekehitysjaksoja. Lisäksi valmistaja kykenee osoittamaan viranomaisille standardien ja direktiivien vaatimuksenmukaisuuden tehokkaammin.

Pöyhönen, Ilpo, Kylmälä, Kaarle, Harju, Hannu, Kemppainen-Kajola, Pia, Kuhakoski, Kalle, Spankie, Greig & Ventä, Olli. Vaatimukset ohjelmistoa sisältäville lääkintälaitteille. Hallinta ja menetelmät vaatimustenmukaisuuden osoittamiseksi [Requirements for medical device software]. Espoo 2002. VTT Tiedotteita – Research Notes 2150. 135 p. + app. 40 p.

Keywords medical device software, medical devices, medical systems, risk management, risk analysis, manufacturing process assessment

Abstract

The performance and complexity of devices and systems used in health care increase year by year. Changes are partly result from technological changes but also methods of treatment are developing and adding pressure for changes of performance and safety of devices. Regularly, the systems used to support methods of treatment and examination include a computer or a client, an operating system and a large amount of different kind of application programs and drivers.

Because of the changes in technology and methods of treatment, the portion of software in medical devices and systems is increasing. Before access to the market, medical devices and systems have to be accepted against promulgation of the regulations in different market areas. If during the access to the market, it is detected that the product does not fulfil the issued requirements of standards and regulations, some remarkable additional cost for the company might be due to the new manufacturing processes and new assurance tests.

Because the demonstration of reliability, safety and performance of the product itself is very difficult, this publication concerns assessment of manufacturing processes.

In this publication, EU and FDA requirements for medical device software are integrated and a model for demonstration of compliance with the requirements is introduced. In addition, risk management processes and methods are largely concerned. With these processes and methods one is able, in addition to identify and estimate risks, also improve the development processes and assure the succeed access to the market.

The publication is intended to the manufacturers of medical devices to guide different sectors of software engineering aim to integrate product risk management and risk analysis to the compact part of design. In this way the design process will produce more reliable and safer products at a shorter product development time. In addition, the manufacturer is able to effectively demonstrate the compliance with the requirements of standards and directives for the authorities.

Alkusanat

Tämä tutkimushanke toteutettiin kolmen lääkintälaitteita valmistavan yrityksen, Instrumentarium Imaging Oy:n, Orion Soredex Oy:n ja Philips Medical Systems Finland Oy:n sekä VTT Tuotteet ja tuotannon yhteistyönä. Näiden yritysten lisäksi rahoituksesta vastasi Teknologian kehittämiskeskus (Tekes). Varsinaisen tutkimustyön hoiti pääasiallisesti VTT Tuotteet ja tuotanto. Tutkimusprojektiin kuuluivat myös kaikkien kolmen lääkintälaiteyrityksen erilliset, myös Tekesin rahoittamat tuotekehitysprojektit, joissa kehitettiin yrityksen validointimallia ja testattiin tässä projektissa laadittu vaatimustenmukaisuuden osoittamismalli.

Kiitämme kaikkia osallistuneita tahoja ja henkilöitä arvokkaasta panoksesta.

Kirjoittajat

Sisällysluettelo

Liitteet

Symboliluettelo

CEN	European Committee for Standardization
CDRH	FDA/Center for Devices and Radiological Health
CFR	Code of Federal Regulations
CMM	Capability Maturity Model
COTS	Commercial of The Self
DHR	Design History Record, tuotantohistoria
ECRI	Emergency Care Research Institute
EN	European Norm, euronormi
EU	European Union Euroopan yhteisö
FDA	U.S. Food and Drug Administration
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects, and Criticality Analysis
GHTF	Global Harmonisation Task Force
GMP	Good Manufacturing Practices
IEC	International Electrotechnical Commission
IEEE	The Institute of Electrical and Electronics Engineers, Inc.
ISO	International Organization for Standardization
LOC	Vakavuustaso (Level Of Concern)
MDD	Medical Devices Directive. Terveydenhuollon laitteita ja tarvikkeita koskeva direktiivi 93/42/ETY:1993.

MRA	Mutual Recognition Agreements
NB	Notified Body, ilmoitettu laitos
NIST	National Institute of Standards and Technology
PEMS	Programmable Electrical Medical System, ohjelmoitava sähkökäyttöinen lääkintälaitejärjestelmä
PHA	Preliminary Hazard Analyses
RAMS	Reliability, Availability, Maintainability, Safety
RMF	Risk Management File. Riskinhallintakansio
RMS	Risk Mangement Summary. Riskinhallintaselostus
SEI	Software Engineering Institute
SFS	Suomen Standardisoimisliitto r.y.
SRHA	Software Requirements Hazard Analysis. Ohjelmistovaatimusten vaara-analyysi
SRS	Software Requirement Specification. Ohjelmiston vaatimusspesifikaatio
UML	Unified Modelling Language
V&V	Verifiointi ja validointi

1. Johdanto

Lääkintälaitteiden tehtävä terveydenhuollossa on sekä potilaan diagnostisoimisessa että hoidossa merkittävä. Ohjelmistojen osuus lääkintälaitteissa on koko ajan kasvamassa, mikä edellyttää viranomaisilta säädöksiä ohjelmistojen riittävän turvallisuuden ja suorituskyvyn toteamiseksi. Tehtävä on kuitenkin vaikeaa ja johtaa helposti joko liian monimutkaisiin tai liian yleispäteviin ohjeisiin ja määräyksiin. Käytännössä vaatimustenmukaisuuden osoittaminen voikin johtaa tuotteen huomattavaan viivästymiseen markkinoilta etenkin kun viranomaisia on useampia. Vallalla on kaksi merkittävää järjestelmää: Euroopan yhteisön (EU) säännökset ja USA:n terveysministeriön (FDA) ohjeet.

Euroopan yhteisön säädökset sisältävät huomattavia vaatimuksia tuotteen ohjelmiston vaatimustenmukaisuuden osoittamiseksi. Tämä tapahtuu tehokkaiden validointimenetelmien avulla, jotka kohdistuvat suunnitteluprosessiin ja sen työkaluihin sekä tuotteeseen liittyvien riskien hallintaan. Myös USA modernisoi lainsäädäntöään sisällyttäen siihen suunnitteluun liittyviä vaatimuksia, mm. ohjelmiston validoinnin. Lisäksi EU:n ja eräiden muiden tärkeiden talousalueiden välillä on solmittu sopimuksia, joiden mukaan osapuolet tunnustavat toistensa tekemät vaatimustenmukaisuuden arviointitulokset uusien säädösten pohjalta tehtyinä tietyssä laajuudessa ja tietylle tuoteryhmälle. Yhtenä tuotealueena ovat lääkintälaitteet.

EU:n direktiivi MDD (1993) edellyttää turvallisuuteen ja suorituskykyyn liittyvien ohjelmistojen validoimista. Validointivaatimukset koskevat sekä lääkintälaitteita ohjaavia ohjelmistoja että itsenäisiä lääketieteellisiä analyysiohjelmistoja. Suunnittelussa ja valmistuksessa käytettävien työkalujen ja ohjelmistojen vaikuttaessa tuotteen vaatimustenmukaisuuteen asetetaan niille vastaavat vaatimukset. Ohjelmisto voi olla myös valmis ostettava ohjelmistokomponentti, joka liitetään sovelluskokonaisuuteen. USA:ssa pakollisia ovat FDA:n vaatimukset, jotka perustuvat säädöskokoelman (CFR) säännöksiin.

Ohjelmiston validointi voi olla monimutkaista ja vaativaa. Mikään yksittäinen validointidokumentti ei voi ilmaista kaikkea vaatimustenmukaisuuden osoittamiseksi tarvittavaa tietoa kaikissa erilaisissa olosuhteissa. Myöskään ei ole olemassa mitään yksikäsitteistä ohjetta tai standardia, joka voisi tukea ohjelmistotuotantoa tai validointia. Laitevalmistajilla on useita vaihtoehtoisia, päällekkäisiä ja rinnakkaisia menetelmämahdollisuuksia vaatimustenmukaisuuden osoittamiseksi. Alan standardit ja ohjeet eivät yksilöi menetelmiä, vaan jättävät menetelmävalinnan valmistajan vastuulle.

Eräänä ratkaisuna esitetään tuotekehitysmalli, jossa yhdistetään EU:n ja FDA:n asettamat vaatimukset lääkintälaitteiden ohjelmistoille sekä esitetään vaatimustenmukaisuuden osoittamismalli, jossa yhdistetään kummankin osapuolen vaatimukset sekä pyritään tehokkaaseen ohjelmiston turvallisuuden ja suorituskyvyn validointiin.

Arviointilaitoksen tekemät ohjelmistoarvioinnit suunnitellaan tapauskohtaisesti. Arviointitapoja on useita. Ne riippuvat sekä yksittäisen ohjelmiston valmiusasteesta ja kriittisyydestä että yrityksen ohjelmistotuotannosta ja -menetelmistä laadunvarmistuksineen. Pääasiallisena tavoitteena kaikissa lähestymistavoissa on arvioida lääkintälaitteiden ohjelmistojen vaatimustenmukaisuus yleisesti hyväksyttyihin alan standardeihin nähden.

EU:n alueella arviointilaitos (ns. ilmoitettu laitos) arvioi yrityksen tuotekehityksen menettelytavat ja tuotekohtaiset validointidokumentit ja myönteisessä tapauksessa antaa tästä todistuksen. Tämä todistus on osoitus siitä, että yritys on ohjelmistotuotannossaan käyttänyt riittävän tehokkaita riskienhallintamenetelmiä ja sillä on toimiva verifiointi- ja validointikäytäntö. Suomessa VTT Automaatio Terveydenhuollon tuotetekniikka on virallinen ilmoitettu laitos (NB) 0537, jonka valvovana viranomaisena toimii Suomessa Lääkelaitos.

Suunnittelun kustannusvähennykset voivat olla huomattavat aikaistettaessa vaatimustenmukaisuuden osoittamista loppuvaiheen testauksesta alkuvaiheen määrittelyyn ja suunnitteluun. Testausmäärät ja -ajat vähentyvät ja tarkentuvat varhaisessa elinkaarivaiheessa aloitetulla verifioinnilla ja validoinnilla.

Ohjelmistovalidointi on kriittistä toimintaa, jossa voidaan virheitä vähentämällä ja korjaavilla toimenpiteillä lisätä lääkintälaitteen käyttökelpoisuutta ja käyttövarmuutta sekä potilaisiin ja käyttäjiin kohdistuvaa turvallisuutta. Lisäksi validoinnilla lisätään valmistajien oikeusturvaa.

Tietoturva tulee tulevaisuudessa olemaan merkittävä tekijä ohjelmistojen validoinnissa. Tietoturva on niiden keinojen muodostama kokonaisuus, joiden avulla tietoriskejä pyritään minimoimaan. Tietoturvaan kuuluvia keinoja ovat mm. tietojen turvaaminen menetelmineen ja välineineen, tietojen turvaamiseen osoitetut resurssit sekä käytettävän välineistön tietoturvallisuuteen liittyvät ominaisuudet.

Suunniteltavan laitteen tai ohjelmiston tavoiteparametrit on aina asetettava ja näiden keskinäinen painotus määriteltävä. Käyttökelpoisia parametreja ovat *luotettavuus* (R), *saatavuus* (A), *huollettavuus* ja *ylläpidettävyys* (M), *turvallisuus* (S) ja *tietoturva* (S).

Julkaisun tarkoituksena on tehostaa valmistajan ohjelmiston tuotekehityksen elinkaaren eri vaiheiden aikana tapahtuvaa dokumentointia siten, että valmistaja kykenisi tuotetulla dokumentaatiolla osoittamaan valvovalle viranomaiselle ohjelmistonsa täyttävän sille asetetut vaatimukset Euroopan yhteismarkkina-alueella sekä Pohjois-Amerikan markkina-alueella.

Tästä syystä hankkeessa on tietoisesti lähdetty rakentamaan yhtä menettelytapaa, joka kelpaisi sekä FDA:lle että EU-alueen ilmoitetulle laitoksellekin. Ratkaisua tukee viime aikoina enenevässä määrin havaittu FDA-vaatimusten lähestyminen kansainvälisiin harmonisoituihin standardeihin. Lisäksi FDA sallii käytettävän vaihtoehtoista tapaa osoittaa ohjelmiston täyttävän sille asetetut vaatimukset.

2. Viranomaisvaatimukset

EU:n, Pohjois- Amerikan (USA ja Kanada), Japanin ja Australian markkina-alueet ovat perustaneet yhteistyöelimen, joka pyrkii yhtenäistämään lääkintälaitteiden turvallisuuteen, tehokkuuteen/suorituskykyyn ja laatuun liittyviä vaatimuksia. Tarkoituksena on edistää teknologista innovaatiota ja kansainvälistä kauppaa. Vaikka selviä merkkejä viranomaisvaatimusten harmonisoitumisesta on olemassa, eivät ne ole vielä harmonisoituneet täysin kaikilla markkina-alueilla. Markkinoille saattaminen tapahtuu vielä eri markkina-alueilla niiden omien sääntöjen ja menetelmien mukaan. Tässä luvussa kuvataan ohjelmistoa sisältävän lääkinnällisen laitteen tai ohjelmistotuotteen ohjelmistoille EU:n ja FDA:n asetettavat vaatimukset ja luodaan katsaus tarvittaviin dokumentteihin. Lisäksi verrataan EU:n ja FDA:n vaatimuksia ja tuodaan esille merkittävät erot sekä annetaan esimerkkejä yleisimmistä hakemuksissa esiintyneistä ongelmista.

Terveydenhuollon tuotteiden globaali viranomaisvaatimusten harmonisointi GHTF:n piirissä etenee koko säädöskentässä kattaen tuotekohtaiset turvallisuus- ja toimintavaatimukset, viranomaisten valvontamenettelyt, arviointilaitosten suorittamaan laadunvalvontaan liittyvät toimenpiteet sekä arviointitoimenpiteiden toteutuksen. Tuotekehityksen kannalta kiinnostavia julkaisuja ovat tuotteeseen liittyvä Essential Principles of Safety and Performance of Medical Devices on a Global Basis (GHTF 1999) sekä arviointitoimintaan liittyvät Design Control Guidance for Medical Device Manufacturers (GHTF 1999) ja Process Validation Guidance for Medical Device Manufacturers (GHTF 1999).

Yllämainitut vaatimukset voidaan kiteyttää siten, että terveydenhuollon tuotteita valmistavan yrityksen ohjelmistotuotannolle sekä ohjelmistoa sisältävälle tuotteelle johdettavat vaatimukset tulevat ensisijaisesti:

- asiakkaiden ja markkinoiden tarpeista
- viranomaisvaatimuksista
- laatujärjestelmävaatimuksista
- valmistajan omista vaatimuksista.

Valmistaja seuraa näiden tarpeiden kehittymistä varmistaakseen oman osaamisensa. Esimerkiksi markkinatutkimukset ja standardointityöskentelyyn osallistuminen antaa tietoa tulevista muutoksista. Tässä julkaisussa painotetaan tärkeimpänä eri markkina-alueiden viranomaisvaatimuksia, jotka kohdistuvat tuotteen turvallisuuteen, suorituskykyyn sekä suunnittelun jäljitettävyyteen.

Vaatimukset kohdistetaan prosessin kaikkiin niihin vaiheisiin, joilla lopullinen tuote luodaan. Tästä seuraa, että valmistajan on määriteltävä ohjelmistotuotanto yhdeksi toimintojensa erityisprosessiksi, jonka kyvykkyys tuottaa laadukasta ja turvallista ohjelmistoa arvioidaan säännöllisesti.

Ohjelmistotuotannon esitutkimusvaiheessa toisaalta asiakkaan vaatimukset ja tarpeet ja toisaalta teknologinen valmius ja kustannustekijät kootaan yhteen. Vaiheen tärkeimpänä tavoitteena on muodostaa selkeä käsitys asiakkaan tarpeista ja vastata kysymykseen, kannattaako tuotetta edes ruveta tekemään. Esitutkimusvaihe on tärkeä, koska siinä arvioidaan asiakkaan esittämiä vaatimuksia ja toiveita, jotka sitten muuttuvat käyttäjävaatimusten dokumentiksi ja tästä taas edelleen tuotteen järjestelmävaatimuksiksi.

Valmistaja asettaa ohjelmistolleen vaatimuksia, jotka kohdistuvat usein luotettavuuteen, toistettavuuteen ja kustannustehokkuuteen. Näitä vaatimuksia voidaan ohjailla erilaisilla menetelmäohjeistuksilla ja tyylioppailla. Jo pelkästään näiden vaatimusten tiedostamisella valmistaja parantaa toimintansa laatua ja luo puitteet ohjelmistotuotannolleen.

Oleellista on, että ohjelmistotuotanto on selkeästi vaiheistettu ja jokaisen elinkaaren vaiheen tulos- ja lähtötietovaatimukset on dokumentoitu riittävän selkeästi yrityksen laatudokumenteissa. Ohjelmistotuotannon vaihejakomalleja käsitellään useammissa eri standardeissa, esim. IEEE 1074 (1997), ISO/IEC 12207 (1995).

Markkina-alueet vaikuttavat välittömästi ohjelmiston hyväksyntäprosesseihin, joita käsitellään myöhemmin tässä luvussa. Lisäksi on huomioitava, että markkina-alueet voivat vaikuttaa myös tuotteen teknisiin ominaisuuksiin, esimerkiksi seuraavilla tavoilla:

- Vallitseva käytäntö (kliininen praktiikka) eri markkina-alueilla voi vaikuttaa tuotteen teknisiin ominaisuuksiin tai käyttöliittymäsuunnitteluun.

- Markkina-alueella voi olla alkeellinen tietoliikenteen infrastruktuuri, mikä voi aiheuttaa ongelmia mm. telelääketieteen sovelluksissa.

- Eri kieliversiot, joihin vaikuttavat myös vallitseva käytäntö. Tiimissä oltava markkina-alueen asiantuntija, joka hallitsee myös terminologian.

- Kieliversio-ongelman voi myös aiheuttaa järjestelmän monikielisyys, jossa käyttöjärjestelmä, osa ajureista ja sovellusohjelma voivat olla konfiguroidut eri kielille.

- Terveydenhuollon rakenne (porrastettu terveydenhuolto, terveydenhuollon palveluketjut, yksityinen tai julkinen terveydenhuolto jne.).

Laatujärjestelmät asettavat ohjelmistotuotannolle vaatimuksia, jotka ohjaavat organisaatiota, projektinjohtoa, katselmuskäytäntöjä, dokumentointia sekä ohjelmistotuotannon vaiheistusta. Käytännössä tästä seuraa, että valmistajan on tuotettava ohjelmistoa jonkinlaisen vaihejakomallin mukaisesti. Vastaavaa vaatimusta edellyttää myös EU-alue (MDD 1993, SFS-EN 60601-1-4 1999) sekä USA-markkinat (FDA 1997, FDA 1999a).

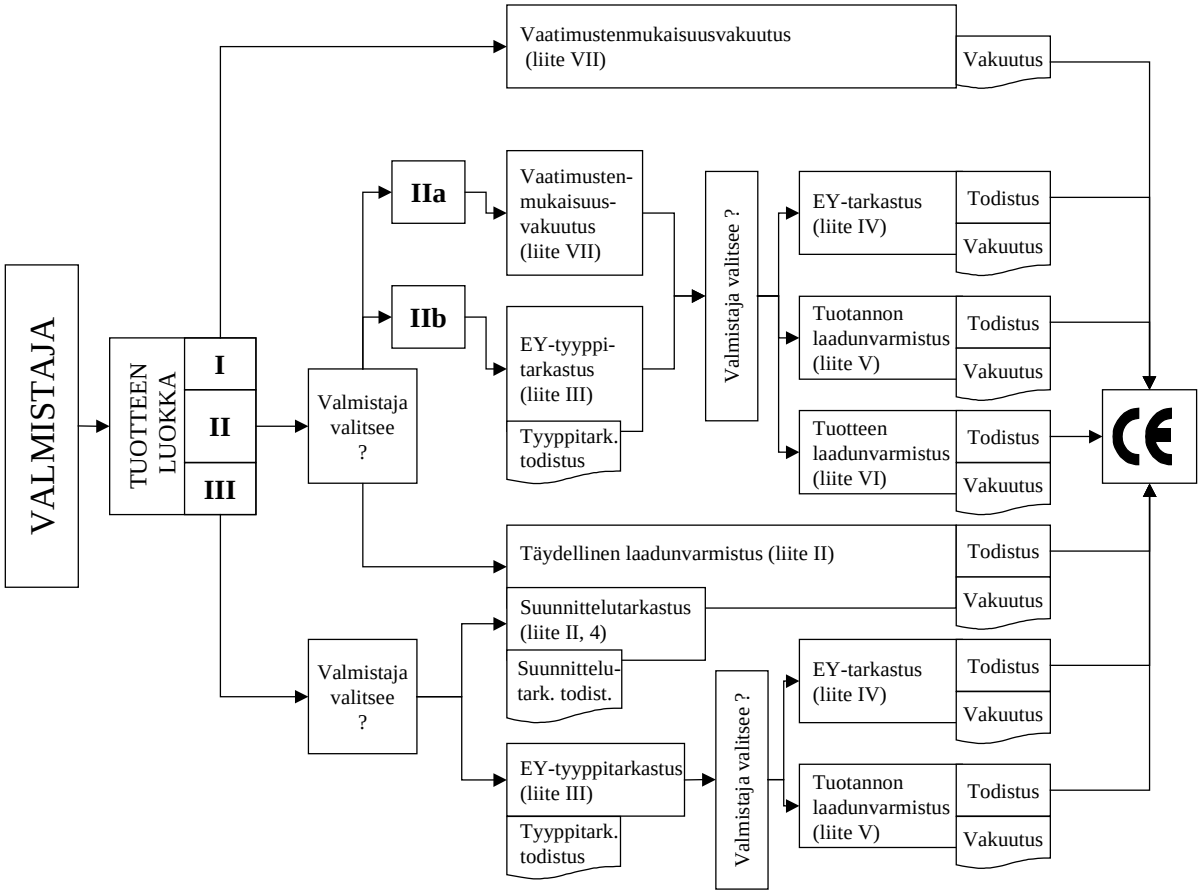
Ohjelmiston viranomaisvaatimukset kohdistuvat pääasiallisesti niihin asioihin, joilla voi olla vaikutusta potilaan, käyttäjän tai ympäristön turvallisuuden tai laitteen suorituskyvyn heikkenemiseen. Käytännössä tämä aiheuttaa sen, että valmistajalla on suunnittelutiimissä viranomaisasiantuntijan lisäksi asiantuntijoita useilta eri aloilta ja lisäksi vielä sovelluskohtainen asiantuntija, joka tuntee myös eri teknologiat ja ymmärtää sovelluksen aiheuttamat riskit ja teknologiavaatimukset.

Viranomaisvaatimukset ovat yksi merkittävä tekijä tuotteen markkinoille saattamiseksi, ja näin ollen se on asetettava yhdeksi olennaiseksi vaatimukseksi ohjelmiston suunnittelussa.

2.1 Euroopan yhteismarkkina-alue

Ohjelmiston sisältävän lääkinnällisen laitteen tai potilaan hoitoon tai tilan tarkkailuun käytettävän ohjelmiston markkinoille saattaminen tapahtuu EU-alueella direktiivin 93/42/EEC (MDD 1993) mukaisesti. EU:hun kuuluvat maat ovat siirtäneet sen kansallisiksi laikseen. Suomessa direktiivin sisältö on laissa L 1505 (1994) ”Laki terveydenhuollon laitteista ja tarvikkeista” ja sen asetuksessa 1506/94. Lain noudattamista Suomessa valvoo sosiaali- ja terveysministeriön alainen Lääkelaitos, jolle kuuluu markkinavalvonta ja vaaratilannejärjestelmän Suomen osuuden hoitaminen.

Direktiivissä annetaan olennaiset vaatimukset tuotteen turvallisuudelle, suorituskyvylle ja käytettävyydelle sekä säädetään kyseisten tuotteiden suunnittelu- ja valmistusprosesseja tuotteen vaatimuksenmukaisuuden varmistamiseksi. Siksi valmistajan on kyettävä osoittamaan, että ohjelmistotuotanto ja sen avulla tuotettu ohjelmisto täyttää direktiivin olennaiset vaatimukset.



Kuva 1. Direktiivin reitit tuotteen markkinoille saattamiseksi.

Korkeamman riskiluokan omaavissa tuotteissa tämän vaatimuksenmukaisuuden arvioinnin suorittaa ilmoitettu laitos. Arvioinnin laajuus riippuu tuoteluokasta, johon tuote kuuluu. Valmistajalla on myös mahdollisuus valita vaihtoehtoisia reittejä kuvan 1 mukaisesti, joilla tuote voidaan saattaa markkinoille. Arviointi kohdistuu tuotteen suunnittelumenetelmiin, valmistukseen ja itse tuotteeseen.

Tuotteen tulee täyttää direktiivin (MDD 1993) liitteen 1 olennaiset vaatimukset, ja direktiivin artiklan 5 mukaan jäsenvaltioiden on pidettävä vaatimustenmukaisina tuotteita, jotka vastaavat yhdenmukaistettujen standardien vaatimuksia. Yhdenmukaistetut standardit löytyvät Euroopan yhteisöjen virallisesta lehdestä.

Ohjelmiston osalta tämä yhdenmukaistetun standardin käyttö tarkoittaa sitä, että kun valmistaja soveltaa ohjelmistotuotannolleen ja ohjelmistolle standardin SFS-EN 60601-1-4: 1999 vaatimuksia, täyttää ohjelmisto myös sille asetetut direktiivin olennaiset vaatimukset. Standardi edellyttää tiettyjen menettelytapojen noudattamista, koska pass/fail-testit eivät sovellu valmiin ohjelmiston testaamiseen. Standardin lähestymistapa on kertoa vaatimus, ja käyttäjä itse määrittelee, miten tämä kyseinen vaatimus saavutetaan.

Menettelytapa noudattaa yleisiä laadunohjauksen periaatteita (SFS-EN ISO 9000 -sarja).

Valmiin ohjelmiston vaatimustenmukaisuutta ilman tuotekehitysprosessin aikana syntynyttä tuotedokumentaatiota on käytännössä mahdotonta arvioida joko ohjelmiston koon, laajuuden (useissa erillisissä järjestelmissä) tai monimutkaisuuden takia. Tästä johtuen ohjelmiston vaatimustenmukaisuuden arviointi kohdistuukin ohjelmistoa suunnittelevan ja valmistavan ohjelmistotuotantoprosessin ja sen tuottamien dokumenttien arviointiin. Arviointi suoritetaan prosessikuvausten, menetelmäohjeiden ja erilaisten raporttien pohjalta, joten suunnittelun dokumentaatio on avainasemassa vaatimustenmukaisuuden osoittamisessa.

Ohjelmoitavalla tekniikalla toteutetun lääkintälaitteen riskinhallinta helpottaa havaitsemaan olennaiset monimutkaisuudet ohjelmistoteknologiassa ja varmistaa piilevien vaarojen varhaisen tunnistamisen. Vaarojen varhainen tunnistus on välttämätöntä, jos halutaan osoittaa, että riittävä turvallisuustaso on saavutettu projektin eri vaiheissa.

Standardin asettamien vaatimusten täyttymisessä korostuu henkilöstön pätevyys, jolloin vaatimukset voidaan pitää olennaisissa elementeissä. Tämä edellyttää henkilöstöltä ohjelmiston laadunvarmistustekniikoiden ja vaaran arviointitekniikoiden hyvää tuntemusta.

Standardin SFS-EN 60601-1-4 vaatimuksenmukaisuuden osoittaminen perustuu tuote- tai projektikohtaiseen riskinhallintakansioon (RMF) ja sen osana olevaan riskinhallinnan selosteeseen (RMS). Näillä valmistaja voi osoittaa ohjelmistotuotteensa täyttävän sille asetetut vaatimukset. Osa standardin vaatimuksista sijoittuu luontevasti yleisiin laatutiedostoihin, jolloin tuotekohtainen riskienhallintakansio saadaan sisällöltään pienemmäksi ja sen tuottamiseen käytettyä aikaa vähennettyä.

Standardin vaatimukset kohdistuvat taulukon 1 esittämiin asioihin.

Taulukko 1. EU-vaatimukset ohjelmistolle (EN 60601-1-4 1999).

Standardin EN 60601-1-4 vaatimus	Tarkoitus
Mukana seuraavat asiakirjat, 6.8	Kuvaa laitteen käyttöä, toimintaa ja rakennetta siten, että laitetta voidaan käyttää sekä huoltaa tarkoituksenmukaisella tavalla.
Dokumentointi, 52.201	Kaikki tuotteeseen liittyvä dokumentaatio, jota käytetään tämän standardin vaatimusten mukaisuuden osoittamiseen
Riskinhallinta suunnitelma, 52.202	Tuotekohtainen riskienhallinta suunnitelma: laajuus, kuinka, mitä, katselmukset jne.
Tuotekehityksen elinkaari, 52.203	Tuotteelle sovellettu tuotekehityksen elinkaari ja vaiheet
Riskinhallinta prosessi, 52.204	Sisältää analyysin ja riskinvalvontatoimenpiteet
Henkilöstön pätevyys, 52.205	Osoittaa henkilöstön pätevyys esim. koulutusrekisterin avulla
Vaatimus spesifikaatio, 52.206	Järjestelmän vaatimusspesifikaatio sisältäen myös kaikki alijärjestelmät
Arkkitehtuuri, 52.207	Kuvaa järjestelmän arkkitehtuurin sisältäen moduulit ja niiden rajapinnat
Suunnittelu ja toteutus, 52.208	Järjestelmän suunnittelu sisältäen myös alijärjestelmät ja testispesifikaatiot
Verifiointi, 52.209	Suunnitelman mukaan toteutettu järjestelmän verifiointi erityisesti turvallisuusvaatimusten osalta
Validointi, 52.210	Suunnitelman mukaan toteutettu järjestelmän validointi aiotussa käyttötarkoituksessa ja olosuhteissa sisältäen erityisesti turvallisuusvaatimukset
Muutokset, 52.211	Suunnittelumuutosten dokumentointi
Arviointi, 52.212	Arviointi, että järjestelmä on suunniteltu tämän standardin vaatimusten mukaisesti

Liitteessä B on luonnosehdotus tarkastuslistaksi, jolla voidaan osoittaa tuotteen täyttävän standardin SFS-EN 60601-1-4 vaatimukset. Liitteessä on myös opastavaa tietoa standardin edellyttämän RMF:n laatimiseksi ja tarkastuslistan täyttämiseksi.

2.2 Pohjois-Amerikan markkina-alue

Tuotteen markkinoille saattaminen Pohjois-Amerikan markkina-alueella (tässä tapauksessa USA) poikkeaa hieman Euroopassa vallitsevasta käytännöstä. Markkinoille saattaminen tapahtuu lakikokoelman CFR 21 (FDA 1998a) mukaan. Lakia valvovana viranomaisena toimii FDA. Markkinoillesaattamisprosessi voidaan suorittaa Premarket Approval, Premarket Notification 510(k), Investigational Device Exemptions tai Humanitarian Device Exemptions -menettelytavan mukaan (FDA 1995). Valittavaan hyväksyntäprosessiin vaikuttaa FDA:n tuoteluokka ja se, onko laitteessa käytetty tutkimus- tai hoitomenetelmä jo markkinoilla käytössä. Tuoteluokka määritellään lähteessä (FDA 1999b).

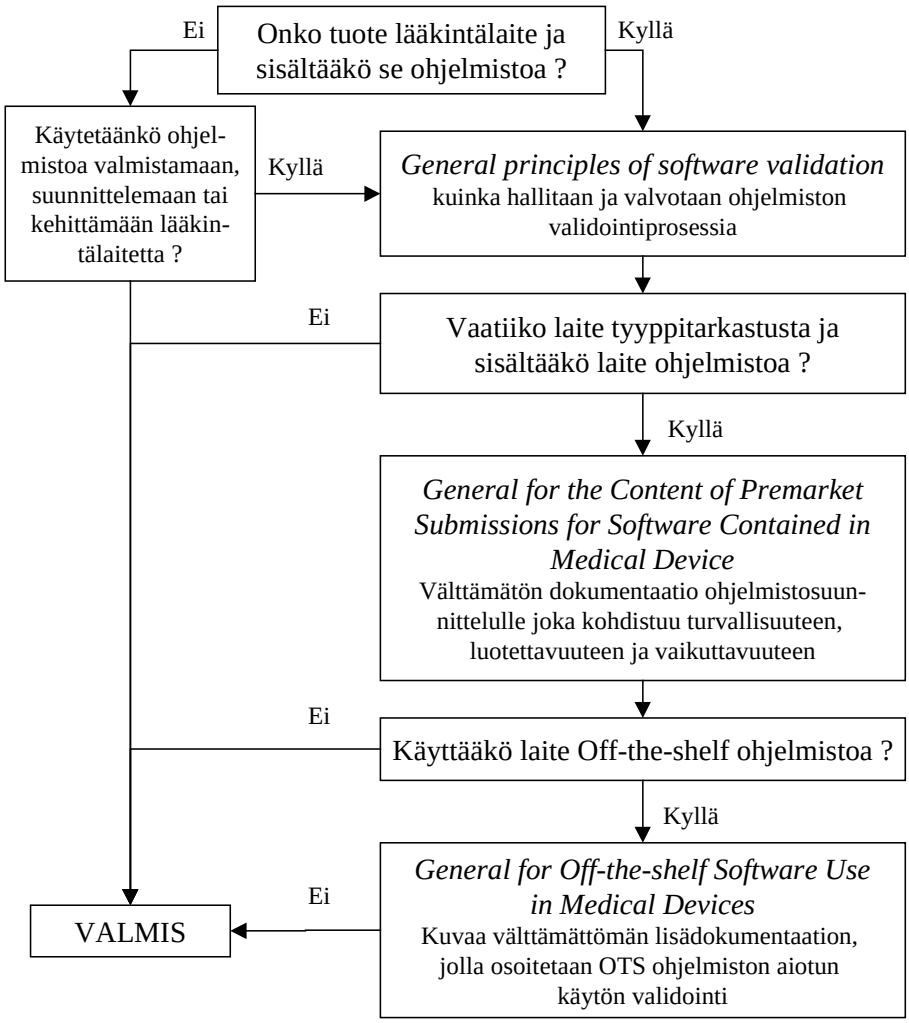
Julkaisussa keskitytään tarkastelemaan Premarket Notification 510(k) -hakemuksen (FDA 1995) avulla tapahtuvaa markkinoille saattamista. Tämä on yleisin suomalaisten valmistajien käyttämä menettelytapaa.

Vaatimusten sisältö on lähes vastaava standardin SFS-EN 60601-1-4 kanssa. Vaatimusten yhdenmukaisuuteen palataan myöhemmin.

Hakemuksen keskeisimpiä asioita on vakavuustason Level of Concern (LOC) määrittäminen. LOC:n pohjalta hakemuksen mukaan liitettävä tuotedokumentaatio laaditaan. Siksi dokumentoinnin markkinointilupahakemuksessa tulisi olla johdonmukainen laitteen tarkoitetun käytön, ohjelmiston LOC:n markkinointilupahakemuksen tyypin kanssa.

Lääkintälaitteen ohjelmiston LOC vaihtelee teknologian, yhteiskunnallisten arvojen ja ohjelmiston ominaisuuksien muuttuessa. LOC:n määrittäminen pitää perustua valmistajan omiin selkeisiin kriteereihin, jos FDA ei ole ennalta määritellyt tiettyjä tasoja ja niiden tarkastusprosesseja. Koska FDA:n määrittelyt ohittavat valmistajan perustelemat tasot, onkin suositeltavaa selvittää ne mahdollisimman varhain FDA:lta ennen hakemuksen toimittamista.

Premarket Notification 510(k) alkaa hakemuksen lähettämisellä FDA:lle, jonka liitteeksi laitetaan kaikki tuotteeseen liittyvä tekninen dokumentaatio, mittaus- ja testidata ja ohjelmiston osalta validointi- ja verifiointi-informaatio sekä testidata, joka tukee väitteitä suorituskyvystä ja turvallisuusominaisuuksista. Hakemus on saatavissa internetissä (FDA 1999b). FDA on julkaissut kaikille hakemustyypeille soveltuvan ohjeellisen dokumentin (FDA 1998b), jota sovelletaan ohjelmistoa sisältäville lääkintälaitteille. Opasdokumentit on tarkoitettu FDA:n tutkijoille ja henkilöille, jotka käyttävät ohjelmistoa lääkintälaitteen suunnitteluun, kehitykseen tai valmistukseen. Opasdokumentin suhde muihin FDA:n dokumentteihin on kuvan 2 mukainen.



Kuva 2. Opasdokumentin suhde muihin dokumentteihin.

FDA:n opasdokumentti edellyttää, että ohjelmistosta selvitetään hakemuksen yhteydessä taulukossa 2 mainitut seikat:

Taulukko 2. FDA- vaatimukset ohjelmistolle (FDA 1998b).

Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices	Tarkoitus
Level of Concern, 3.1	Vakavuuden taso
Software Description, 3.2	Ohjelmiston kuvaus
Device Features Controlled by Software, 3.2.1	Ohjelmistolla ohjatut laitteen toiminnot
Operational Enviroment, 3.2.2	Käyttöympäristö
Device Hazard Analysis, 3.3	Vaara-analyysi
Software Requirements Specification (SRS), 3.4	Ohjelmiston vaatimusspesifikaatio
Architecture Design Chart, 3.5	Arkkitehtuuri kaavio suunnittelusta
Design Specification, 3.6	Suunnittelu spesifikaatio
Traceability Analysis, 3.7	Jäljitettävyys tuotteen dokumentointiin ja elinkaaren aikana tehtyihin toimintoihin.
Developmet, 3.8	Kehitys
Validation, Verification and Testing, 3.9	Validointi, verifiointi ja testaus
Revision Level History, 3.10	Versionhallinta
Unresolved Anomalies (Bugs), 3.11	Selvittämättömät ongelmat (bugit)
Release Version Number, 3.12	Myyntiin vapautettu ohjelmaversio

2.3 Vaatimusten yhdenmukaisuus

FDA on päivittänyt lääkintälaitteiden ohjelmistoille tarkoitetun ohjedokumentin (FDA 1998b) yhdenmukaiseksi standardien IEC 60601-1-4, SFS-EN ISO 9001, SFS-EN ISO 9000-3 ja laatujärjestelmäsäädösten CFR 21 (FDA 1998a: Part 820) kanssa. Maailmanlaajuisessa terveydenhuollon sektorissa tapahtuva menettelytapojen harmonisointi osaltaan myös lähentää Euroopan ja USA:n vaatimuksia toisiinsa. Tästä syystä voidaan jo olettaa, että standardin SFS-EN 60601-1-4 ehdottama menettely ohjelmiston vaatimuksenmukaisuuden osoittamiseksi kelpaa myös FDA:lle.

FDA:n tutkijat Herrmann & Zier painottavat artikkelissaan (1996: uusittu 1999), että laitevalmistajien on oleellista tunnistaa muutamia FDA:n ohjetyöskentelyssään suosimia periaatteita. Ensiksi, FDA ei vaadi minkään tietyn standardin noudattamista, vaikka se aktiivisesti ottaakin osaa kansallisten ja kansainvälisten yhdenmukaisten standardien laatimiseen. Toiseksi, on laitevalmistajien vastuulla valita ja noudattaa mitä hyvänsä kansallista tai kansainvälistä standardia, kunhan valinta perusteellaan. Kolmanneksi, standardit ovat työkaluja, joilla perustellaan tuotteen yhteensopivuus lääkintälaitteiden määräysten, FDA:n periaatteiden ja ohjedokumenttien kanssa.

Edellä mainittu artikkeli käsittelee varsinaisesti standardin SFS-EN 60601-1-4 yhdenmukaisuutta FDA:n arviointiohjeen (FDA 1991) vaatimusten kanssa. Lisäksi erityisenä tarkastelun kohteena ovat standardin elinkaaren kehitystoimenpiteiden, riskienhallinnan toimenpiteiden ja tuotedokumentoinnin vastaavuus markkinoille pääsyohjeen (FDA 1998b) määräysten kanssa.

Artikkelin yhteenvetona mainitaan, että ”There is a strong correlation between FDA's software guidance document (FDA 1991) and SFS-EN 60601-1-4 (1996). Almost all of the premarket submission requirements identified in the software guidance document are addressed by the standard or its normative references. Therefore, SFS-EN 60601-1-4 can be a useful tool for demonstrating compliance with U.S. medical device software regulations, including those covered by the FDA guidance document”. Kaarisulkujen sisältö lisätty selvyyden vuoksi.

FDA on korvannut Herrmann & Zierin artikkelissaan mainitseman opasdokumentin vuonna 1998 uudella opasdokumentilla Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices (FDA 1998b), joka on päivitetty yhteensopivaksi SFS-EN 60601-1-4, ISO 9001 (SFS-EN ISO 9001) ja 9000-3 (SFS-EN 9000-3) kanssa.

FDA:n opasdokumentin (FDA 1998b) ja SFS-EN 60601-1-4 asettamia vaatimuksia on verrattu keskenään ja todettu niiden vastaavan toisiaan taulukon 3 mukaisesti:

Taulukko 3. FDA vs. EU-vaatimukset.

Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices	SFS-EN 60601-1-4
3.1 Level of Concern	52.204.3.2.2-3 Severity Level
3.2 Software Description	52.204 Riskienhallintaprosessi
3.2.1 Device Features Controlled by Software	52.204.3.1.5-6 Riskienhallintaprosessi
3.2.2 Operational Enviroment	52.208.2 Suunnittelu ja implementointi
3.3 Device Hazard Analysis	52.204 Riskienhallintaprosessi
3.4 Software Requirements Specification	52.206 Vaatimus spesifikaatio 52.207 Arkkitehtuuri 52.208 Suunnittelu ja implementointi
3.5 Architecture Design Chart	52.207 Arkkitehtuuri
3.6 Design Specification	52.208 Suunnittelu ja implementointi
3.7 Traceability Analysis	52.201 Dokumentointi 52.210 Validointi 52.211 Muutokset 52.212 Arviointi
3.8 Developmet	52.203 Tuotekehityksen elinkaari
3.9 Validation, Verification and Testing	52.209 Verifiointi 52.210 Validointi 52.208.1 Suunnittelu ja implementointi
3.10 Revision Level History	52.201 Dokumentointi 52.211 Muutokset
3.11 Unresolved Anomalies (Bugs)	6.8 Mukana seuraavat asiakirjat
3.12 Release Version Number	52.201 Dokumentointi
	52.205 Henkilöstön pätevyys
CFR 820.22 Quality audit [1]	52.212 Arviointi

[1] Sisäisen auditoinnin tulokset eivät ole FDA-arvioijan käytössä.

Edellä mainittujen kahden dokumenttien kuvaamien vaatimusten välillä voidaan havaita seuraavanlaisia eroja:

- LOC on tavallaan sovellusalueperustainen vakavuuden määrittely, jonka perusteella laaditaan hakemuksen mukaan liitettävä dokumentaatio. LOC voi vaihdella vähäisestä [Minor] merkittävään [Major]. Tuotteen ollessa vakavuustasolla ”vähäinen” on hakemuksen mukaan liitettävä dokumentaatio huomattavasti suppeampi, kuin vakavuustasolla ”merkittävä” oleva tuotedokumentaatio. Vastaavaa määritystä SFS-EN 60601-1-4 ei tunne, mutta toisaalta hieman vastaavaan menettelyyn päästään riskienhallintasuunnitelman (52.202) ja riskienhallintaprosessin (52.204) kautta. Mikäli riskienhallintaprosessi osoittaa, että tuotteen aiheuttama riski on erittäin pieni tai merkityksetön, voidaan riskienhallintaprosessin edellyttämä dokumentaatio jättää hieman suppeammaksi. Tämä on kuitenkin aina tapauskohtaisesti arvioitava.

- Standardi SFS-EN 60601-1-4 arvioi tuotteen aiheuttaman riskin vahingon vakavuuden ja esiintymistodennäköisyyden tulona. FDA:n opasdokumentin mukaan ohjelmistoviat ovat luonteeltaan systemaattisia, ja sen tähden niiden esiintymistodennäköisyyttä ei voida määritellä käyttämällä perinteisiä staattisia metodeja. Riskin arvioinnin ohjelmistotuotteelle tulisi perustua vian aiheuttaman vaaran vakavuuteen olettaen, että vika esiintyy. Tätä lähestymistapaa voidaan FDA:n mukaan käyttää myös määrittelemään sen vahingon vakavuus, joka voi seurata jokaista vaara-analyysissä tunnistettua vaaraa.

Yllämainitun perusteella voidaan todeta, että valmistaja voi määritellä ohjelmistotuotannonprosessinsa siten, että se täyttää molempien markkina-alueiden vaatimukset.

Suunnittelussa ja valmistuksessa käytettävien työkalujen ja ohjelmistojen vaikuttaessa tuotteen vaatimustenmukaisuuteen tulee niiden soveltuvuus käyttötarkoitukseensa kelpuuttaa sekä EU:n että FDA:n kuvaamassa järjestelmässä.

2.4 Yleisimmin havaitut puutteet arvioinneissa

Ohjelmiston vaatimustenmukaisuuden arviointi perustuu suunnittelu- ja tuotedokumentaation arviointiin. Alla olevassa tekstissä on vertailtu standardin SFS-EN 60601-1-4 ja 510(k)-hakemuksen yleisimpiä puutteita. Standardin SFS-EN 60601-1-4 puutteiden havainnointi perustuu käytännön työssä saatuun kokemukseen.

Puutteet, joita SFS-EN 60601-1-4 arvioinnissa on havaittu:

- ohjelmiston riskianalyysin systemaattisuuden puuttuminen
- riskienhallinnan tehokkuuden arviointi ja riskienvähentämiskeinojen toteutuksen verifiointi puutteellista
- vaatimusten ja dokumenttien jäljitettävyys tai puutteellista
- muutoshistoria ja muutosten hyväksyttäminen puutteellista
- suunnitelmat (riskienhallinta, verifiointi ja validointi) puutteellisia
- spesifikaatiot ja riskien kuvaus spesifikaatioissa puutteellista
- verifioinnin, testauksen ja validoinnin hyväksyntärajat puutteellisia
- jäännösriskien siirtyminen riskianalyysistä laitteen mukana seuraaviin dokumentteihin puutteellista
- riittämättömät työkalujen validointiraportit, joilla työkalu hyväksytään osaksi ohjelmistotuotannon työkaluja
- COTS-komponenttien kuvaus ja validointi aiottuun käyttötarkoitukseen puutteellista.

Vastaavasti yleisimmät 510(k)-hakemuksen puutteet ovat:

- ohjelmistoon liittyvää dokumentaatiota ei ole tuotettu
- riittämätön LOC:n arviointi
- riittämätön vaara-analyysi
- järjestelmällisen dokumentoinnin puute
- riittämätön testaus
- ei osoitusta, että laite on tosiasiassa tietokone-ohjattu
- testituloksiin liittyvää dokumentaatiota ei ole tuotettu
- riittämättömästi tai vajavaisesti kuvatut tuotekehitysprosessit
- ei aiempaa kelpuutusta ohjelmistolle tai kaupalliselle off-the-shelf-ohjelmistolle (COTS)
- testiohjelmiston dokumentointi riittämätön hakemuksen laitteelle
- ei konfiguroinnin hallintasuunnitelmaa tai proseduureja
- riittämätön yhteenveto ja raportit.

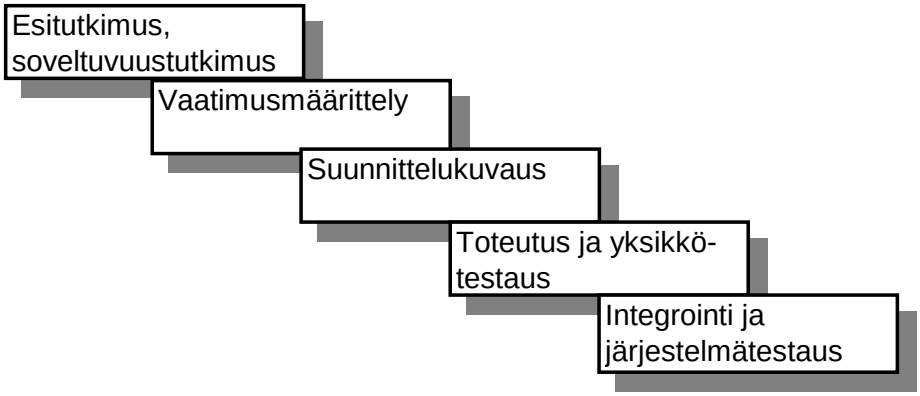
3. Ohjelmistotuotanto

Ohjelmistotuotannon elinkaarimalli muodostuu perättäisistä vaiheista, jotka kuvaavat, minkä tehtävien kautta ohjelmisto valmistetaan. Prosessi alkaa ohjelmistolta vaadittavien ominaisuuksien tunnistamisesta ja dokumentoinnista ja päättyy kehitetyn ohjelmiston kelpuutukseen kaikkien vaadittujen ominaisuuksien suhteen. Vaihejako määrittää tarkastuspisteet, joissa varmistetaan, että ohjelmistotuotantoprosessi voi edetä seuraavaan vaiheeseen.

Luvussa tarkastellaan aluksi vesiputousmallia, joka on usein viranomaisvaatimusten perustana. Seuraavaksi tarkastellaan käytännön ohjelmistotuotantoprosesseja, joita ovat ohjanneet yhä integroituneempien kehitysympäristöjen ja komponenttiteknologioiden kehittyminen ja työasemien ja -ympäristöjen verkottuminen. Lopuksi kuvataan ohjelmistokehityksen laatutavoitteet, erityisesti tarkastellaan viranomaisattribuuttien (turvallisuus, käytettävyys ja suorituskyky) muuntumista laatukriteereiksi. Laatukriteerien täyttymisen osoittamiseksi on kehitetty suuri joukko mittareita, joita myös luvussa esitellään.

3.1 Elinkaarimallien vaiheet

MDD ja FDA vaativat, että ohjelmistotuotantoprosessi on jaettu vaiheisiin, mutta eivät kerro miten. Toisaalta ne edellyttävät tuotantoprosessilta dokumentteja, jotka on nimetty ja joiden sisältö implikoi tietynlaisen vaihejaon olemassaoloa. Vesiputousmalli, jossa tuotanto etenee hyvin määritellyissä peräkkäisissä vaiheissa vaatimusten määrittelystä testaukseen, sopii periaatteessa lääkintälaitteen tuotantoprosessin kuvaukseksi. Vesiputousmallista on käytössä useita variaatioita, jotka poikkeavat toisistaan vaiheiden määrän ja laajuuden suhteen. Kuva 3 esittää erään sulautetun ohjelmiston tuotannon elinkaarta, jossa ohjelmiston vaatimukset tulevat pääosin laitteen toiminnallisista vaatimuksista, ja käyttöönotto ja koulutus kuuluvat koko laitteen käyttöottoon ja koulutukseen loppukäyttäjälle läpinäkymättömästi.



Kuva 3. Yleinen vesiputousmalli ohjelmistotuotannon elinkaarelle.

Vesiputousmalli edustaa ennen kaikkea hallinnollista näkökulmaa ohjelmistotuotantoon, vaiheet etenevät aikataulussa suunnitelluilla resursseilla ja tuottavat edellisessä vaiheessa määritellyt tulokset, jotka vaiheiden päättyessä verifioidaan. Vesiputousmallissa oletetaan edellisen vaiheen määritykset oikeiksi. Uusia tuotteita kehitettäessä tai yritykselle uusia ohjelmistotekniikoita sovellettaessa tämä voi helposti aiheuttaa projektien epäonnistumisia.

Vesiputousmallin eri vaiheiden tuottamat keskeiset tulokset ovat seuraavat:

- Elinkaarimallin mukainen ohjelmiston toteuttamissuunnitelma, ohjelmistotuotannon erityispiirteet huomioonottava projektisuunnitelma, joka määrittää ohjelmistokonseptin ja eri vaihetuotteet ohjelmiston laajuuden ja ulkoisten vaatimuksien mukaisesti.
- Vaatimusmääritysdokumentti, joka sisältää paitsi keskeiset ohjelmiston toiminnalliset vaatimukset ja rajoitukset, myös ei-toiminnalliset vaatimukset ja rajoitukset sekä ulkoisten rajapintojen kuvaukset. Vaatimusten tulee olla niin selkeästi ja yksikäsitteisesti kuvatut, että niiden toteutuminen voidaan objektiivisesti verifioida ja validoida ennalta määritellyillä menetelmillä, kuten katselmusten, testien ja esimerkiksi järjestelmän korkeiden kuormitustilanteiden simulointien avulla.
- Suunnitteludokumentit, jossa kuvataan ohjelmiston toiminnalliset kokonaisuudet ja niiden väliset rajapinnat. Alustava suunnittelukuvaus laajennetaan käsittämään tarpeelliset yksityiskohtaiset suunnittelukuvaukset.
- Ohjelmistokoodi dokumentaatioineen sekä yksikkötestauksen tulokset.
- Integrointi- ja hyväksymistestausraportit.
- Käyttöönotto-, käyttö- ja koulutusdokumentaatio.
- Viittaukset muihin kehitettäviin suunnitelmiin.

Koko ohjelmiston elinkaaren aikana kehitetään ja toteutetaan ohjelmistotuotannon rinnalla seuraavia suunnitelmia:

- Verifiointi- ja validointisuunnitelma, jossa määritellään toimenpiteet, joilla varmistetaan edeltävän vaiheen määritysten toteutuminen vaiheen tuottamien artefaktien perusteella (verifiointi), ja toimenpiteet, joilla varmistetaan, että valmis ohjelmisto toimii, kuten vaatimusmäärittely edellyttää (validointi).
- Konfiguraation hallintasuunnitelma, joka määrittää konfiguraation hallinnan alaiset artefaktit, ja menetelmät, joilla niiden eheys turvataan läpi ohjelmistotuotannon elinkaaren. Näitä voivat olla muun muassa keskeiset dokumentit, ohjelmakoodi, tallennusmediat ja laitteisto- ja käyttöjärjestelmäympäristö.

- Ohjelmiston riskienhallintasuunnitelma, kun ohjelmistoon liittyy vaaratekijöitä. Vaaratekijät tulee tunnistaa jo konseptivaiheessa, jotta niistä tulevat vaatimukset saadaan mukaan ohjelmiston vaatimusmäärittelyyn ja ohjelmistotuotannon elinkaaren eri vaiheiden toteutukseen.
- Ohjelmiston laadunvarmistussuunnitelma, jossa kuvataan toimet, joilla varmistetaan, että havaitut puutteet ja virheet korjataan. Samoin tunnistetaan poikkeamiset standardeista, muista ohjeista ja suunnitelmista.

Laadunvarmistussuunnitelma on eräänlainen sateenvarjo, jonka avulla varmistetaan ohjelmiston elinkaaren tuottamien tulosten, verifioinnissa ja validoinnissa sekä konfiguraation hallinnassa käytettyjen menetelmien ja työkalujen riittävyys ja täydellisyys aiottuun käyttötarkoitukseen siten, että vaatimustenmukaisuus toteutuu.

Esimerkiksi ISO 9000 -standardin katselmuskäytännöt asettavat suuret vaatimukset laadunvalvonnan ammattitaidolle, jotta vaatimus vaihetuotteiden täydellisyydelle, oikeellisuudelle ja ristiriidattomuudelle voidaan niiden perusteella hyväksyä verifioiduiksi.

3.2 Käytännön ohjelmistotuotantoprosessit

Nykyisten ohjelmistotuotantoprosessien kehittymistä ovat ohjanneet yhä integroituneempien kehitysympäristöjen ja komponenttiteknologioiden kehittyminen ja työasemien ja -ympäristöjen verkottuminen. Niitä kuvaavat elinkaarimallit perustuvat tiimipohjaiseen kehitystyöhön ja ohjelmistokomponenttien käyttöön, ja vaihejako perustuu toimivien ohjelmistoversioiden tuottamiseen.

Iteratiivisella ja inkrementaalisella kehityksellä pyritään pienentämään ohjelmistotuotantoprosessin riskejä toteuttamalla aluksi ohjelmiston keskeisimmät ominaisuudet kevyemmillä todentamis- ja kelpoistamisaktiviteeteilla. Ohjelmistotuotantoprosessissa keskitytään ennen kaikkea toimivien ohjelmaversioiden tuottamiseen mahdollisimman varhain, ja näin pyritään varmistamaan valitun arkkitehtuurin sekä muiden keskeisten oletusten ja valintojen toimivuus.

Nämä iteroinnit ja kussakin vaiheessa noudatettu mallin mukainen dokumentaatio voidaan kuitenkin koota tuotantoversiovaiheessa yhdeksi, esimerkiksi perinteisen vesiputous-elinkaarimallin mukaiseksi dokumentaatioksi, jos tuotteeseen liittyy viranomaisvaatimuksia. Myös ohjelmistotuotannon vaiheiden dokumentointistandardit sopivat parhaiten vesiputousmalliin. Inkrementaalisessa ohjelmistotuotantoprosessissa jokainen iteraatiokierros tuottaa vastaavat osat standardidokumentaatioon.

Tuotantoversion dokumentointi perinteisen vaihejaon tavalla soveltuvien standardien mukaisesti helpottaa turvallisuussuunnitelman mukaisen riskienhallintadokumentin tuottamista ja hyväksymistä, jos ohjelmistotuotteeseen sisältyy viranomaisvaatimuksia. Muutoin joudutaan helposti tuottamaan ylimääräisiä perusteluja ja selvityksiä vaatimustenmukaisuudesta.

Standardienmukaisuuden toisena etuna on se, että valmiita mittaus- ja dokumentointijärjestelmiä on kaupallisesti saatavilla, eikä niitä tarvitse itse kehittää. Lisäksi niitä on liitetty integroituihin kehitysympäristöihin, mikä mahdollistaa hyvin pitkälle dokumentoinnin automatisoinnin.

3.3 Ohjelmistotuotantoprosessin työkaluja

Seuraavassa tarkastellaan joitain markkinoilla olevia työmenetelmiä ja niiden ominaisuuksia.

Yleisimmät kaupalliset ohjelmistotuotantoprosessit eivät sovellu sellaisenaan turvallisuusvaatimuksia sisältävien ohjelmistojen tuotantoon, koska pääpaino on markkinoille tuonnissa ja projektiriskeissä. Työtapana yleinen tiimipohjainen inkrementaalinen kehitystyö, jossa tiimi itse vastaa myös laadunvalvonnasta, ei vastaa riippumattoman verifioinnin ja validoinnin vaatimuksia. Ohjelmistoprosessien koulutus- ja muu aineisto perustuu myös vastaaviin yritysten ohjelmistotyökalujen käyttöön. Ohjelmistotyökalujen versiokehitys on myös nopeaa, ja uusien versioiden myötä vanhoille ei helposti saa virheenkorjauksia, eikä työkalujen laatu ehdi aina stabiloitua. Tuki ohjelmistotuotannolle esimerkiksi ISO 9000:n tai FDA:n vaatimusten mukaisesti on yleensä kannanottopaperien tasoa. Näissä kuvataan lyhyesti, miten tuotteita voidaan käyttää näiden erikoisvaatimusten mukaisiin ohjelmistoprojekteihin.

Toisen ryhmän muodostavat toimittajat, jotka valmistavat esimerkiksi testaus- tai ohjelmistoprosessien mittaustyökaluja sekä eri standardien mukaisia dokumentointi- paketteja ja niiden käyttöön liittyvää ohjeistusta, koulutusta ja konsultointipalveluja, joilla yritykset voivat hankkia esimerkiksi ISO-9000:n tai CMM-tasojen 2 tai 3 sertifioinnin.

Kolmas ryhmä on erittäin vaativiin sovelluksiin sertifioituja reaaliaikakäyttöjärjestelmiä valmistavat yritykset. Näihin liittyy myös vastaavien sertifioitujen sovellusten kehitystuki ohjelmistokehitystyökaluineen ja prosessi- ja dokumentointiohjeineen.

Kaupallisten menetelmien hyödyntäminen vaatii aina paljon koulutusta ja sitoutumista yrityksissä sekä mahdollisesti myös suuria muutoksia aikaisempiin prosesseihin.

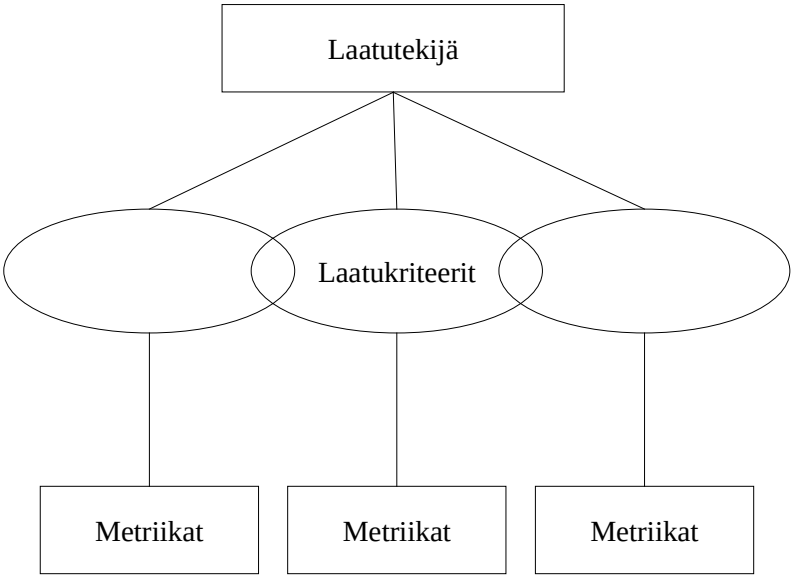
3.4 Ohjelmiston luotettavuus laatuhierarkiassa

Ohjelmistotuotteen laatu on mitattavissa, jos valmistaja valitsee laatukriteerit, niille mitat ja mitoille laadun rajat. Laatukriteerejä ovat mm. käytettävyys, toimintavarmuus ja turvallisuus. Turvallisuus ja siihen läheisesti liittyvä käytettävyys eli palvelujen saatavuus, vaikuttavuus sekä suorituskyky ovat keskeisimmät viranomaistaholta tulevat laatukriteerit. Valitettavasti sekä turvallisuus ja käytettävyys ovat suoraan huonosti mitattavissa.

Turvallisuudella tarkoitetaan hyväksyttävää arviota riskistä, joka määritellään toiminnan epäonnistumisen vakavuuden ja todennäköisyyden tulona. Jos vakavan epäonnistumisen todennäköisyys on riittävän pieni, toiminnon suorittaminen voidaan hyväksyä. Ohjelmisto voi johtaa järjestelmän vaaraan kahdella tavalla: joko ohjelmiston ulostulo tai ajastusvirheet johtavat järjestelmän vaaralliseen tilaan, tai ohjelmisto ei havaitse tai käsittele niitä laitteistovikoja, joihin sen kuuluisi reagoida.

3.4.1 Laatuhierarkia

Yhteinen piirre kaikille laadunmittausjärjestelmille on kolmiportainen hierarkia, jonka ylimmällä hallinnollisella tasolla on laadunohjaus laatutekijöillä, seuraavalla tasolla kriteerit, joilla vastataan hallinnon vaatimuksiin ja alimmalla tasolla mitat (metriikat), joilla mitataan laadun riittävyyttä (kuva 4).



Kuva 4. Laatuhierarkia (McCall 1994).

Taulukko 4. Luotettavuus on laatuominaisuus, jolla perusteellaan luottamusta järjestelmän tarjoamiin palveluihin. Luotettavuus koostuu kuudesta attribuutista: toimintavarmuus, käytettävyys/saatavuus, ylläpidettävyys, turvallisuus, luottamuksellisuus ja tiedon eheys.

Attribuutti	Tarkenne
Toimintavarmuus	Palvelun jatkuvuus, ts. ohjelmiston kyky säilyttää toiminnan taso määritellyissä olosuhteissa ennalta määritellyn ajan.
Käytettävyys	Palvelun käyttövalmius, johon kuuluu helppous käyttää ja oppia käyttämään ohjelmistoa, mm. vaadittavien syötteiden
Ylläpidettävyys	Soveltuvuus muutoksiin, korjauksiin ja uudelleen käyttöön sekä helppous havaita ja korjata ohjelmistovirheitä.
Turvallisuus	Kriittisiltä seurauksilta välttyminen.
Luottamuksellisuus	Tiedot ja palvelut ovat vain niihin oikeutettujen saatavissa eikä niitä paljasteta sivullisille.
Tiedon eheys	Eri lähteissä sijaitsevat tiedot ovat yhdenmukaisia, esimerkiksi ajan tasalla; yksittäinen tieto, esimerkiksi potilaan syntymäaika, on sama kaikissa tietokannoissa.

Kolmiportaisella rakenteella pyritään aikaistamaan laadunhallintaa ohjelmiston elinkaaressa siten, että valitut toimenpiteet vaikuttavat myönteisesti myöhemmissä elinkaaren toiminnoissa. Aikaistamisella on myös merkitystä ohjelmisto-vaatimusten muodostamiselle sekä painotukselle sopivan laatuisen tuotteen toteuttamiseksi. Vaatimukset kohdistuvat sekä toimintoihin, aikataulutukseen että budjetointiin.

3.4.2 Laatuattribuutit: luotettavuus

Tässä julkaisussa laatutekijäksi valitaan luotettavuus (dependability), mikä koostuu taulukon 4 esittämistä attribuuteista.

Toimintavarmuus on järjestelmän tuottaman palvelun jatkuvuutta, ts. ohjelmiston kyky säilyttää toiminnan taso tietyissä olosuhteissa ennalta määritellyn ajan. Käyttäjä odottaa järjestelmän toimivan tietyn ajan. Toimintahäiriöt voivat aiheutua mistä ohjelmiston kehitysvaiheessa tehdystä virheestä tahansa. Siksi luotettavuuden varmistaminen kuuluu kaikkiin ohjelmistotuotannon vaiheisiin.

Usein etenkin puhekielessä toimintavarmuudesta käytetään termiä luotettavuus. Sillä ymmärretään myös käsitteitä tarkkuus, ristiriidattomuus, jämeryys tai kykyä toimia epänormaaleissa olosuhteissa. Luotettavuudelle on kehitetty useita mittaustapoja (mm. MTTF, vikatiheys) ja ennustemalleja (mm. luotettavuuden kasvumallit).

Turvallisuudella voi olla kytkentöjä muihin luotettavuusattribuutteihin: esimerkiksi laitteen käytettävyyden pettäminen hoitotilanteessa voi johtaa potilaan vaaraan (säästävän leikkauksen eli laparoskopian aikana menetetään videokuva), tai tiedon eheyden murtumisen seurauksena voi olla väärä hoitotoimenpide (tajuttoman potilaan lääkeaineallergia ei näy tietokannassa). Turvallisuusattribuutti ei kuitenkaan ole korvattavissa näillä muilla attribuuteilla, koska esimerkiksi toimintavarma tuote ei välttämättä ole turvallinen. Turvallinen tuote on vikaantuessaankin vaaraton. Tämä merkitsee sitä, että turvallisuusattribuuttiin liittyvät myös ne toimenpiteet, joilla turvallisuudesta varmistutaan vikojen yhteydessä.

Ylläpidettävyys on määritelty (McCall 1994) helppoutena, jolla ohjelmisto voidaan ymmärtää, soveltaa tai muuttaa. Se ei ole riippumaton muista attribuuteista: Ylläpidettävyys on verrattavissa joustavuuteen ja sovellettavuuteen, jos sovellusympäristön muutokset vaikuttavat ohjelmiston määrittelyyn. Ylläpidettävyys on verrattavissa todennettavuuteen ja korjattavuuteen, koska ongelmat on havaittava ja korjattava.

Tietoturva merkitsee informaation keräämisen, tallettamisen ja siirtämisen turvaamista. Se on purettu saatavuudeksi (turvataan järjestelmän tuottamien palvelujen ja tietojen saatavuus), luottamuksellisuudeksi (varmistutaan, että palvelut ja tiedot ovat vain niihin oikeutettujen saatavissa, eikä niitä paljasteta tai muulla tavoin saateta sivullisten käyttöön) ja eheydeksi (varmistetaan tietojen muuttumattomuus ja havaitaan tietojen muuttuminen). Liitteessä A on lyhyt kuvaus FDA:n ja EU:n näkökulmasta tietoturvaan.

3.4.3 Laatukriteerit

Ohjelmiston laatutavoitteiden saavuttamista tarkkaillaan kaikissa ohjelmistotuotannon elinkaarivaiheissa, mistä syystä tavoitteet muunnetaan sovelluskohtaisille vaiheille ominaisiksi kriteereiksi. Johdon, asiakkaan tai käyttäjän toiveita ja määräyksiä laadusta (laatuattribuutteja) arvioidaan kriteereillä siten, että tietty laatuattribuutti voi sisältää tietyn joukon kriteereitä ja kriteeri voi kohdistuu yhteen tai useampaan attribuuttiin. Taulukko 5 esittää McCallin (1994) kriteerit attribuuteittain.

Taulukko 5. Laatukriteerit McCallin (1994) mukaan.

Laatuattribuutit	**Kriteerit, joista laatuattribuutti koostuu**
Oikeellisuus	Jäljitettävyys, yhtenäisyys ja täydellisyys
Toimintavarmuus	Virhesietoisuus, yhtenäisyys, tarkkuus ja yksinkertaisuus
Käytettävyys	Toimintakelpoisuus, koulutus, kommunikatiivisuus, input/output -määrä ja input/output -suhde
Eheys	Pääsyn hallinta ja pääsyn tarkistus
Suorituskyky	Suorituksen tehokkuus ja taltioinnin tehokkuus
Ylläpidettävyys	Yhtenäisyys, yksinkertaisuus, suppeus, modulaarisuus ja itsestäänselittävyys
Mukautuvuus	Modulaarisuus, yleistävyys ja itsestäänselittävyys
Testattavuus	Yksinkertaisuus, modulaarisuus ja itsestäänselittävyys
Siirrettävyys	Modulaarisuus, itsestäänselittävyys, alustariippuvuus ja ohjelmistoriippuvuus
Uudelleen-käytettävyys	Yleistävyys, modulaarisuus, alustariippuvuus ja ohjelmisto-riippuvuus ja itsestäänselittävyys
Vuorovaikutteisuus	Modulaarisuus, ominaisuuksien yhteisyys ja tiedon yhteisyys

McCallin esittämät laatuattribuutit ovat kaikenkattavia. Ne saattavat tukea toisiaan tai olla ristiriidassa keskenään.. Esimerkiksi jos ohjelmistotuote on luotettava, se on yleensä myös testattava ja käytettävä. Joustavuuden liiallinen kasvattaminen saattaa kuitenkin olla ristiriidassa luotettavuuden kanssa ja erityisesti se lisää kustannuksia ja alentaa tehokkuutta.

Laatuattribuutin suhdetta käyttäjän tai asiakkaan tarpeisiin tai järjestelmäominaisuuksiin kuvataan taulukossa 6 (McCall 1994). Tavoitteena on löytää ohjelmistotuotteelle sopivat laatuvaatimukset tarkastelemalla tuotteen ominaisuuksia: sovellustyyppi, odotettu elinikä, käyttöriski, suorituskykyvaatimukset jne.

Taulukko 6. Laatuattribuuttien suhde järjestelmäominaisuuksiin. 1 Henkilöturvallisuus, 2 Realiaikasovellus, 3 Kriittinen liiketoimintasovellus, 4 Tiedonkäsittely, 5 Vuorovaikutus muiden järjestelmien kanssa, 6 Vaarallisten materiaalien käsittely, 7 Pitkä elinkaari, 8 Jatkuvasti muuttuvat säännöt, 9 Erityiset käyttäjän kvalifioinnit, 10 Jatkuva käyttö, 11 On-linekäyttö. (McCall 1994)

Laatuattribuutit	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Oikeellisuus	X	X	X			X					X
Toimintavarmuus	X	X	X			X				X	X
Käytettävyys									X		X
Eheys			X	X							
Suorituskyky		X									X
Ylläpidettävyys							X			X	X
Mukautuvuus							X	X	X		
Testattavuus	X						X				
Siirrettävyys							X				
Uudelleenkäytettävyys							X				
Vuorovaikutteisuus					X						
Turvallisuus	X					X					
Tuettavuus							X	X	X	X	

3.4.4 Laatumitat

Jokaiselle laatuattribuutille on tarjolla useita mittareita. Mittaaminen kohdistetaan prosessiin, tuotteeseen tai resursseihin. Tässä kohdassa esitellään yleisimmät laatumetriikat ja seuraavassa luotettavuusmetriikat.

Mitat on validoitava. Fenton (1995) muistuttaa mittaamisen perusperiaatteista:

- Mittaamisella täytyy olla selkeät tavoitteet. Täytyy tietää mitä aikoo mitata. Jos attribuutti on kohteen ominaisuus niin kuin pituus on henkilön ominaisuus, mittaamista ei voi aloittaa ennen kuin on tunnistanut sekä kohteen että ominaisuuden.

- Mitan täytyy säilyttää kokeelliset relaatiot. Jos henkilö A on pidempi kuin henkilö B, kaikki käyvät mitat antavat saman tuloksen olipa asteikko mikä tahansa. Tämä yleensä epäonnistuu ohjelmistolaadun mittaehdokkailta.

- Matemaattisen kuvauksen määritteleminen ei vielä tee mittausta. Mittaa käytetään ohjelmistojen yhteydessä huolimattomasti. Melkein mikä tahansa ohjelmasta laskettu numero nimetään mitaksi.

Fenton määrittelee mittaamisen prosessiksi, jossa attribuuttiin liitetään numero siten, että empiiriset suhteet säilyvät.

Fentonin mukaan ohjelmistoattribuuteilta puuttuu yleensä hyvä empiirinen malli. Mittojen toimivuutta yritetään arvioida vertaamalla niiden ennustamia tuloksia todellisiin tai tekemällä korrelaatioanalyysi. Prosessissa, tuotteessa ja ympäristössä on monia tekijöitä, joita ei voi kokeessa vakioida, joten ainoaksi keinoksi jää nähdä mittaluvut satunnaislukuina, otoksena ja tehdä aineistolle tilastollinen analyysi. Otoksen koko on yksi muuttujista. Empiirinen relaatio jää stokastiseksi, mikä ei ole aina tarkoituksenmukaista.

Mitoilla on käyttöarvoa vasta, kun a) ne voidaan validoida, b) ne voidaan helposti laskea, ja lasketaan myös, c) on olemassa vertailuasteikko (benchmarking), joka kertoo, mitä lukuarvo tarkoittaa eli onko tulos hyvä vai huono, d) huonon tai hyvän suorituskyvyn syyt tunnetaan ja niihin voidaan puuttua. Puhutaankin kokonaisen mittausohjelman pystyttämisestä.

Mittatyökaluja on saatavilla verkosta. Vaikeutena ei ole mittojen puute vaan niiden valinta. Osa mitoista on epäinformatiivisia yksinään, osa ei täytä mitan määritelmää Fentonin esittämällä tavalla. Osalla mitoista on käänteinen riippuvuussuhde: prosessin optimoiminen yhden mitan suhteen tuottaa huonoja tuloksia toisen suhteen.

Mills (1988) luokittelee mitat kolmella tavalla:

1. Tuote- ja prosessimitat
2. Objektiiviset ja subjektiiviset mitat, joiden arvo ei riipu mittaajasta.
3. Yksinkertaiset ja johdetut mitat, joista johdetut mitat ovat yksinkertaisten mittojen funktioita (esim. ohjelmarivien lkm/työtunnit).

Taulukossa 7 on lueteltu tuotemittoja, jotka ovat joko laajalti tunnettuja ja käytettyjä tai sisältävät mielenkiintoisen näkökulman.

Prosessimitoille on yhteistä se, että niiden käyttö tähtää alkavan tai käynnissä olevan projektin laadun tai kustannusten ennustamiseen. Ne eivät välttämättä sovi projektin

toteutuksen laadun mittaamiseen. Kustannuksia ennustavista malleista on tunnetuin Boehmin (1984) COCOMO, joka yhdistää viisitoista kustannuksien syntymiseen liittyvää muuttujaa ja ohjelmiston koon malliksi, jonka kustannusennusteen tarkkuus on 20 % 70 %:ssa tapauksista.

Ohjelmiston perusmitta on ohjelmarivien lukumäärä. Sitä on käytetty mm. luotettavuuden ennustamiseen, vaikka kokeellinen relaatio luotettavuuden ja lukumäärän välillä on epämääräinen. Intuitiivisesti on selvää, että ohjelmakoon kasvaessa virheiden tekemisen mahdollisuus kasvaa. Erilaiset kompleksisuusmitat ovat myös suosittuja. Suunnittelun laadun mittana on käytetty mm. ohjelman sisältämien moduulien tai rakenneosien välisten kytkentöjen määrää. Näillä on merkitystä virheen etenemisessä ajon aikana ja koodiin tehtyjen muutosten vaikutuksen leviämisessä. Ongelmana on jälleen se, että on erittäin vaikea kuvata täsmällisesti, kuinka kytkentöjen määrä vaikuttaa laatuun. Ohjelman käyttötarkoituksella on varmasti vaikutusta, mikä ei näy mitassa.

Ohjelmistoprosessin tuottavuudelle on olemassa ylivoimaisesti suurin joukko mittareita, joista työaika, kustannukset työtuntia tai ohjelmariviä kohti ovat tavallisimmat. Osa tuottavuuden mitoista antaa tietoa myös luotettavuudesta, sillä esimerkiksi lähdekoodin rakenne, muutosten tekemisen helppous, uudelleenkäytettävyys ovat sekä tuottavuuden että luotettavuuden tekijöitä.

Taulukko 7. Tunnetuimpia mittoja.

1. **Ohjelmiston koon mitat**	
a) ohjelmarivien lukumäärä (lines of code)	
b) FP (function points)	”Weighed sum of the number of inputs, outputs, inquiries and master files”
c) bang	”Total functionality of the software system delivered to the user”
2. **Kompleksisuusmitat**	
a) syklomaattinen kompleksisuus v(G)	Ohjausvuon graafista laskettu mitta ohjelmiston rakenteelle
b) solmut (knots)	Jos ohjelmalistaukseen piirretään hyppykäskyjen vaikutus suorituksen siirtymiseen, solmut on sama kuin leikkaavien piirrosviivojen lukumäärä.
c) tietovuo (information flow)	Tietovuo on ohjelmamoduulin sisäänmenojen ja ulostulojen lukumäärän ja moduulin pituuden funktio.
3. **Halsteadin tuotemitat**	
a) ohjelmiston sanasto (vocabulary) n	Operaattorien ja operandien lukumäärä, kun kukin lasketaan vain kerran.
b) ohjelman pituus (length) N	Operaattorien ja operandien lukumäärä, kun kaikki esiintymät lasketaan.
c) ohjelman laajuus (volume)	V = N log n
4. **Laatumittoja**	
a) vikasisältö: suunnittelumuutosten lkm, koodikatselmuksissa löytyneiden virheiden lkm, testauksen aikana löydettyjen virheiden lkm, koodimuutosten lkm	
b) MTTF (mean time to failure)	MTTF-ennuste voidaan johtaa esimerkiksi Musan (1980, 1987) mallista tai testaustuloksista.

Elinkaaren vaiheisiin on omat mittansa, tästä määrittelyvaiheen esimerkki: NASA Software Assurance Technology Center jakaa verkossa työkalua nimeltä ARM. Se on ohjelma, joka etsii englanninkielisestä vaatimusmäärittelystä avainsanoja ja -rakenteita (Wilson et al. 1996). Kirjoittajat toteavat, että lukumäärät viestivät vaatimusmäärittelyn

laatua, koska määrittelyn on oltava selkeästi kirjoitettu, hyvin jäsennelty ja kattava, ja kaikkia näitä ominaisuuksia voi mitata tarkkailemalla dokumentin kieltä. Vaatimusmäärittelyä on helppo parantaa yksinkertaisilla menetelmillä kuten esimerkiksi kirjoitusohjeilla, jotta tulos on selkeä. Dokumentin ei tarvitse olla kiinnostava, mutta sen täytyy olla yksikäsitteinen. Vaatimusmäärittelyn vaatimukset tulee esittää sellaisessa muodossa, että niitä voidaan mitata selkeillä ja toistettavilla mittareilla.

Mitoista ei ole hetkellistä hyötyä, sillä yksi mittaustapahtuma ei vielä vaikuta mihinkään. Mittoja voi käyttää laadunosoituksessa hyväksi vasta, kun tuotteesta tai prosessista on kerätty aineistoa, jolle voidaan tehdä analyysi: mittaukset on tehty vakioidussa ympäristössä, mittaajan tai ympäristömuutosten vaikutukset on kompensoitu, luonnollisen vaihtelun amplitudi tunnetaan. Jos voidaan osoittaa, että laatu on mittajärjestelmän mielessä, tilastollisesti merkitsevästi kohonnut vaaditulle tasolle ja pysyy siellä, mittoja voidaan käyttää hyödyksi myös lääkintälaitteen ohjelmistotuotannon laadunarvioinnissa. Koska laatujärjestelmät eivät sisällä valmista kehystä mittajärjestelmälle eivätkä MDD tai FDA ota kantaa mittoihin, niiden käyttäminen kuuluu vaihtoehtoisiin menetelmiin, jotka joudutaan perustelemaan arvioijalle erikseen. Tiedossamme ei ole ennakkotapausta, josta voisi ottaa mallia.

Vaikka mitan validoiminen voi olla vaikeaa tai työlästä, se joudutaan tekemään vain kerran, ja sen jälkeen mitta tuo prosessiin lisäarvoa vähentämällä vaatimustenmukaisuuden osoittamiseen kuluvaa työmäärää.

3.4.5 Luotettavuusmitat

Luotettavuusmitoilla tarkoitetaan todennäköisyysarviointiin pohjautuvia mittalukuja ohjelmiston virhetoiminnasta. Toinen tapa arvioida luotettavuutta perustuu laatumittoihin, joita esiteltiin edellisessä kohdassa. Kummatkin mitat ovat olleet jo vuosikymmeniä lupaavia lähestymistapoja arvioida ohjelmiston luotettavuutta, mutta varsinaista läpimurtoa ei ole tapahtunut. Kuitenkin kyse on aina mittaamisesta jollakin esittämisasteella ja mittojen käyttäminen yhtenäistää ja täsmentää mittausprosessia. Usein riittää pelkkä karkea arvio esimerkiksi jonkun tietyn riskinvähennyksen suuruudesta. Arvio voi perustua teknisen ratkaisun pätevyyteen tai kokemusperäiseen tietoon, jolle ei tarvitse etsiä täsmällistä todennäköisyysarvioita.

Laatumittoja ja niillä kuvattavia laatukriteereitä on kehitetty lukuisia, ja monet niistä ovat keskenään ristiriitaisia. Yrityksen voi olla vaikea perustella ohjelmiston luotettavuutta niiden pohjalta.

Myöskään luotettavuusmittojen käyttäminen ei ole ongelmatonta. Ne laadittiin pääasiassa 1970-luvulla laitteistoille tarkoitetuista luotettavuuden arviointimalleista. Senkin jäl-

keen on kehitystä tapahtunut, mutta mitään varsinaista läpimurtoa ei todennäköisyyspohjaisessa arvioinnissa ole tapahtunut. Laskentamallin pitää olla oikea sekä kerätyn luotettavuustiedon että ympäristöolettamusten suhteen, muuten laskennan tuloksena olevat ennusteet eivät ole tarkkoja. Tarkkoja malleja ja tuloksia tarvitaan silloin, kun vaaditaan korkeaa luotettavuutta.

Mallit ovat kuitenkin hyödyllisiä alemmilla vaatimustasoilla. “Älä luota ohjelmistoon” on eräissä lähteissä (mm. Leveson 1995) käytetty motto. Käytetään riskinvähennykseen ensin tukevia menetelmiä, esimerkiksi laitteistopohjaista vikasietoisuutta, ja kun vaadittava luotettavuus ohjelmistolle on saatu alas, käytetään luotettavuuden arvioimiseen laatumittoja tai yksinkertaisia luotettavuusmittoja.

Luotettavuusmitat pyrkivät mittaamaan ja ennustamaan virhetoiminnan todennäköisyyttä tietyllä aikavälillä (MTTF). Nämä mitat yleensä laaditaan kehitettäessä sovellukselle sopivia luotettavuusmalleja (mm. Musa et al. 1987). Niiden tuottamia ennusteita ja arvioita voidaan siis pitää myös mittoina.

Useimmat mallit perustuvat samoihin olettamuksiin, mm. Musan malli:

- Testisyötteet ovat satunnaisotoksia syöteympäristöstä.
- Kaikki ohjelmiston virhetoiminnat ovat havaittavia.
- Vikaantumisvälit ovat keskenään riippumattomia.
- Vikavälit ovat ajallisesti eksponentiaalisesti jakautuneita.

Näiden olettamusten pohjalta Musan malliin saadaan virheiden kumulatiivinen kokonaismäärä:

$$d(t) = D(1 - e^{-bct}), \tag{1}$$

missä D on virheiden kokonaismäärä,

b, c ovat vakioita, jotka määrätään käyttökokemustietojen perusteella vastaavalle ohjelmistolle, ja

$d(t)$ on kumulatiivinen ajassa t havaittujen virheiden kokonaismäärä (2).

$$MTTF(t) = \frac{e^{bct}}{cD}. \tag{2}$$

Vakioiden b, c ja D määrääminen ei ole yksinkertaista, mutta mallin hyödynnettävyyden kannalta merkittävää.

Kattavuusmitat ilmoittavat mikä osuus ohjelmakokonaisuudesta on suoritettu joko testaamalla tai todellisessa suoritusympäristössä. Mitä korkeampi kattavuus, sitä todennäköisemmin jäljellä ei ole alkuperäisiä ohjelmistovirheitä. Kattavuusmitat saadaan parhaiten sijoittamalla ohjelman haarautumiin laskureita. Kattavuusmittoja ovat silloin seuraavat (Kitchenham 1990, Woodward et al. 1979):

- Käskykattavuus, jossa suoritetut käskyt laskettiin.
- Haarakattavuus, jossa suoritettujen käskyjen määrä laskettiin.
- Ehtokattavuus, jossa ’tosi’ ja ’epätosi’ -arvot laskettiin kaikista ohjelman loogisista ilmauksista.
- Polkukattavuus, joka ilmaisee eri suoritepolkujen määrän. Polkukattavuus on täsmällisin kattavuusmitoista, sillä ohjelman kaikkien mahdollisten kokonaispolkujen määrä on yleensä niin iso, ettei täydellistä polkukattavuutta voida odottaa saatavan suoritettua.
- Datakattavuus, jossa mitataan ohjelman dataelementtien käyttömäärää.

Näitä kattavuusmittoja hyödynnetään parhaiten ohjelmiston kehittämisen aikana, sillä niiden käyttäminen edellyttää ohjelmakoodin tuntemusta.

3.4.6 Esiin tulleita laatuongelmia

FDA ylläpitää tietokantaa lääkintälaitteista löytyneistä vioista, jotka ovat tulleet ilmi joko järjestelmätesteissä, asennuksen aikana tai potilaskäytössä. Wallace ja Kuhn (2000) kävivät läpi vuosina 1983–1997 esiin tulleet viat, joiden alkusyy oli ohjelmistossa (383 kappaletta). Vian ilmitulotavat jaettiin kolmeentoista luokkaan (taulukko 8). Vian todennäköisin syy ilmoitettiin kolmessatoista luokassa. Ylivoimaisesti suurin osa raportoiduista vioista johtui ohjelmiston loogisesta virheestä (43 %) tai siitä, että ohjelma laski tuloksen väärin (24 %) (taulukko 9).

Valitettavasti tekijät eivät kohdistaneet vikoja laatuattribuuteille. Sen sijaan he analysoivat, millä toimenpiteillä viat olisi voitu joko ehkäistä tai havaita ennen laitteen luovutusta asiakkaalle. Yhteenveto analysoinnista on esitetty taulukossa 10.

Taulukko 8. Vikojen ilmitulotavat lääkintälaitteissa (Wallace & Kuhn 2000).

Vian ilmitulotapa	Osuus vioista	Vian ilmitulotapa	Osuus vioista
Väärä siirto tai liike	22 %	Virheellinen vaste	3 %
Data: virheellinen tieto tai tiedon katoaminen	1 %	Laatuvirhe: tuote ei vastannut laatuvaatimuksia	1 %
Näyttövirhe	8 %	Puutteellinen palvelu	10 %
Funktio	29 %	Järjestelmävirhe	1 %
Yleinen	0 %	Ajastusvirhe	1 %
Sisäänmeno	4 %	Ohjeistusvirhe	1 %
Ulostulo	19 %		

Taulukko 9. Vian todennäköisimmät syyt lääkintälaitteissa (Wallace & Kuhn 2000).

Vikaluokka	Osuus vioista	Vikaluokka	Osuus vioista
Laskenta	24 %	Logiikka	43 %
Muutoksen vaikutus	6 %	Toiminnon puuttuminen	3 %
Tuotteenhallinta	1 %	Muu virhe	3 %
Data	5 %	Laadunvarmistus	3 %
Vikasietoisuus	1 %	Virheellinen tai puuttuva vaatimus	4 %
Alustus	2 %	Ajastus	3 %
Liityntävirhe	2 %		

Taulukko 10. Lääkintälaitteiden vikojen ennaltaehkäiseminen (Wallace & Kuhn 2000).

Vikatyyppi	Ehkäiseminen	Havaitseminen
Väärinkoodatut vakiot	Koodin lukeminen	Lukeminen, yksikkötestit
Puuttuva suunnittelun ja toteutuksen välinen verifiointi	Jäljitettävyysanalyysi, muutosten hallinta	Muutosten tarkistaminen, regressiotestit
Väärä lähdekoodi (wrong master program)	Tuotteenhallinnan välineiden käyttö	Tuotteenhallinnasta vastaava tarkistaa version
Virheellinen input-data	Arvoalueiden määrittely, toimenpiteet poikkeustapauksissa	Input-määrittelyn verifiointi, testit
Ylikuormitus kaataa ohjelman tai laitteen	Vikasietoisuuden rakentaminen, kuormitusrajojen määrittely	Kuormitustestit, epänormaalien input-arvojen syöttäminen
Alustusarvojen tallettaminen epäonnistuu	Käyttöönoton ja jatkuvan käytön ehtojen määrittely	Koodin lukeminen, kuormitustestit
Ohjelmiston laiteliittymä tai yhteydet muihin ohjelmistoihin eivät toimi	Vaatimusmäärittely, interface-ehtojen jäljittäminen	Katselmukset, järjestelmätestit
Ohjauksen logiikkavirheet tai -puutteet	Jäljitettävyysanalyysi, suunnittelun ja toteutuksen vertaaminen	Koodin lukeminen, testit
Osa järjestelmäfunktioista puuttuu	Vaatimusmäärittely, jäljitettävyysanalyysi	Jäljitettävyysanalyysi, testit
Kirjoitusvirhe koodissa aiheuttaa laitteen yhteensopimattomuuden	Koodin lukeminen	Algoritmien läpikäynti, testaaminen
Testaussuunnitelmaa ei ollut tai sitä ei noudatettu	Projektin johtamisen tarkistaminen	Projektin tilanteen seuranta
Vaatimusmäärittely ei kattanut erikoistilanteita	Mallinnus, formaalit menetelmät, jäljitettävyys	Vaatimuskatselmukset, järjestelmätestit
Epäsynkronisuus	Simulointi, suunnittelun katselmointi	Ajastuksen analysointi, testit

4. Lääkintälaitteiden ohjelmistot

Lääkintälaitteessa on tyypillisesti erikoislaitetoimintoja, joita toteutetaan lisääntyvässä määrin nk. laiteläheisellä ohjelmistolla. Muun ohjelmiston alustana on yhä useammin nykyaikainen PC, jossa on käyttöjärjestelmä ja sitä vastaavat muut ohjelmat, ehkä paljonkin laskentaa sisältävä sovellusohjelmisto, mahdollisesti paikallinen tai lähiverkon kautta käyttöön saatava tietokanta, tietoliikenneohjelmistoa sekä käyttöliittymä. Laiteläheinen ohjelmisto ja suuri osa sovellusohjelmistoa on usein itse tehtyä tai alihankittua, muissa ohjelmistoissa turvaudutaan yhä useammin kaupallisiin järjestelmiin ja komponentteihin. Tässä luvussa kuvataan valmiskomponenttien (COTS, commercial-of-the-self) ja valmisohjelmistojen riskejä, riskinhallintaa ja viranomaisvaatimuksia sekä kuvataan PC -laitteiden ja -käyttöjärjestelmien konfigurointia lääkintälaitekäyttöön.

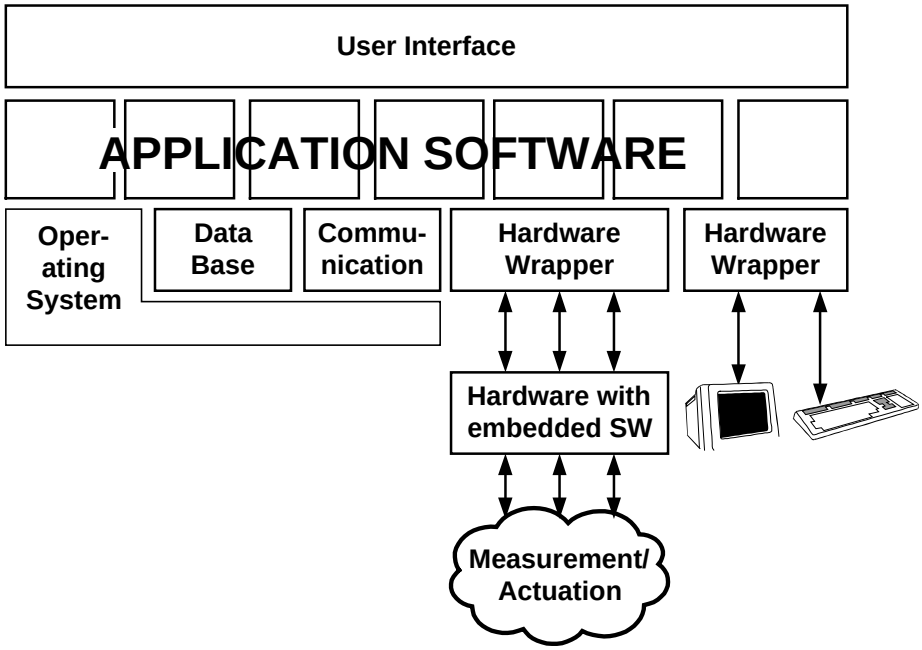
4.1 Lääkintälaitteen tyypillinen arkkitehtuuri

Lääkintälaite on tyypillinen nykyaikainen nk. älykäs laite, jonka toimintoja toteutetaan sekä ohjelmistolla, elektroniikalla että mekaniikalla tai niiden kombinaatioilla. On myös tavallista, että lääkintälaite on ohjelmistoltaan, elektroniikaltaan ja mekaniikaltaan varsin monimutkainen, niin suunnittelun kuin valmistuksen osalta suurta ammattitaitoa ja osaamista vaativa nk. high tech -tuote. Useimmissa valmistajayrityksissä siten laitteen suunnittelu ja implementointi jakaantuu myös organisatorisesti ja projektirakenteellisesti eri teknologia-alueiden kesken.

Lääkintälaitteella yleensä mitataan suureita potilaasta. Mittaus on joko jatkuvaluonteista, ja mitata voidaan joko yhtä (esimerkiksi verenpaine) tai useampaa suuretta (monikanavainen MKG, anestesialaitteet) samanaikaisesti. Mittaustulosta käytetään tyypillisesti joko potilaan tutkimukseen tai tilan seurantaan. Lääketieteelliset kuvauslaitteet puolestaan toimivat suurimmaksi osaksi kuva eli mittaus kerrallaan. Turvallisimmissa mittauksissa kohteena oleva suure voidaan mitata kehoa häiritsemättä tai rasittamatta (mitataan esim. hermosignaalien aiheuttamia sähkökenttiä kehon ulkopuolelta). Moni mittaus on kuitenkin mahdollista vain joko laitteen itse kehoon kohdistaman, esimerkiksi röntgensäteilyn avulla tai muun järjestelyn, esimerkiksi varjoaineen injektoinnin, avulla. On kuitenkin ilmeistä, että ohjelmistoa sisältäviä lääkintälaitteita jo lähitulevaisuudessa käytetään mitä moninaisimpiin tarkoituksiin, ja ne toimivat siten yleisesti sekä mittaus- ja/tai toimilaitteina potilaaseen nähden. Lääkintälaite sisältää yhä enemmän tutkimus- ja hoitotilannetta tukevia toimintoja, lähtien esimerkiksi hälytysrajojen asettamisista ja hälytysten hyväksikäytöstä monipuolisiin tietokantaoperaatioihin.

Laiteläheinen mittaus- ja toimilaiteläheinen ohjaus toteutetaan, monista eri syistä, suurelta osin elektroniikkakorteille sulautetuilla ohjelmistoilla. Nk. korkeamman tason oh-

jelmistologiikka toteutetaan sovellusohjelmamoduuleissa, ja niitä varten lääkintälaitteissa on nykyisin oma tietokone, tavallisimmin PC. Sulautetut ohjelmat kykenevät yleensä vain binääriseen tiedonsiirtoon, joten väliin on järkevää asettaa nk. hardware wrapper –moduuleita, linkiksi binäärimuuttujien ja sovellusohjelmien käyttäjäläheisten muuttujien ja tietorakenteiden välille (kuva 5). Näytön ja näppäimistön lisäksi lääkintälaitteessa voi olla erikoisnäyttöjä, valintakytkimiä, varoitusvaloja jne.



Kuva 5. Mittaus- ja ohjaus toteutetaan elektroniikkakorteille sulautetuilla ohjelmistoilla.

Lääkintälaitteen tietokoneessa on hyvin usein käyttöjärjestelmä, monet toiminnot käyttävät tietokantaa, ja yhä useammin laitteeseen liittyy erilaisia etätoimintoja (esimerkiksi laitteesta otetaan selainyhteyksiä sairaalan potilastietokantaan, tai laitteeseen otetaan muualta yhteyttä). Käyttöliittymään liittyvät toiminnot tavataan toteuttaa arkkitehtuurisesti omissa ohjelmistomoduuleissa, sekoittamatta niihin sovelluslogiikoita.

Lääkintälaite on vielä usein itsenäinen stand alone -laite, jolla on yksi käyttäjä. Uusimmissa tuotteissa toimintoja voi olla myös hajautettu, eli laitteilla on useita käyttäjiä omine käyttöliittymineen, tai yksi käyttäjä voi hallita useaa mittausasemaa.

4.2 Lääkintälaitteiden ohjelmistotuotannon tyypillisiä piirteitä

Kuten varsin monella laitetoimittaja-alalla, ohjelmistotuotannon merkitys on noussut suuriin mittasuhteisiin varsin äskettäin. Yritysten tuotekehityksen osaaminen ja sisäinen arvostus korostaa vielä varsin paljon itse sovellusalan osaamista, so. röntgenkuvantamista ja sen fysiikkaa, lääketiedettä, magnetismia, mittaustekniikkaa jne. Ohjelmistoammattilaiset joutuvat näistä ydinalueista hieman sivuun ja siten ns. apulaisen rooliin. Monen tuotekehittäjän ura ohjelmistoammattilaiseksi on kulkenut esimerkiksi elektroniikkasuunnittelijan toimen kautta, jolloin ohjelmistotekniikan peruskoulutus on jäänyt vähäiseksi, ja niin kutsutusti työ on opettanut.

Monien arvostettujen alojen (lääketiede, fysiikka, elektroniikka) peruskoulutukseen on jo pitkään kuulunut esimerkiksi ohjelmointikielen kurssi, ja moni on esimerkiksi opiskeluaikana tehnyt jonkun harjoitustyön yliopiston keskustietokoneella. Siten käsitys ohjelmistotekniikasta on peräisin ajalta, jolloin tietokoneita oli hyvin vähän, ohjelmia eri tarkoituksiin oli vähän, ja nykytarpeisiin nähden varsin pienet ja yksinkertaiset ohjelmat voitiin perustellusti kirjoittaa lähtien puhtaalta pöydältä. Hyvin pieni osa laitteiden ominaisuuksista oli toteutettu ohjelmistoilla. Ohjelman kirjoittamista ja muuttamista pidettiin ehkä yksinkertaisena ja helppona, ja sen on katsottu edelleen helpottuneen entistä tehokkaampien ja käyttäjäystävällisempien tietokoneiden ja ohjelmistotyökalujen yleistyttyä. Tämä kaikki on osittain johtanut illuusioon, että ohjelmistokehitys ja ohjelmiston laadunhallinta on helppoa ja yksinkertaista toimintaa, eikä siihen esimerkiksi yrityksen johdon tule kiinnittää niin paljon huomiota.

Kritiikistä huolimatta on todettava, että suomalaiset lääketieteen laitteet ovat varsin kilpailukykyisiä, laadukkaita ja menestyneitä, ja yrityksissä on myös erittäin asiantuntevia ja aikaansaavia ohjelmistoammattilaisia, joskin ohjelmistokehityksen osalta organisaatiot ovat vielä pieniä. Alalla on totuttu toimimaan laatutietoisesti mekaniikan ja elektroniikan osalta, ja sama on heijastumassa myös ohjelmistoon.

Ohjelmistotuotantoprosessista on tunnistettavissa, ja useimmiten dokumentoitunakin elinkaari- ja vaihejakomalli. Mallina on yleensä vesiputousmalli, tuoteversioittain etenevine inkrementteineen. Yritykset ovat siirtymässä tai jo siirtyneet oliopohjaisuuteen, ja kehitysprosessi on muuntumassa vastaavasti. Mallinnustekniikat on poimittu UML:stä tai käytetään muuten vastaavia. Yleisin ohjelmointikieli on C++, ja työkaluna Microsoftin Visual Studio. Case-työkaluista kotimainen Prosa on monelle tuttu; kiinnostusta kalliimpiin ja kattavampiin työkaluihin on. C:tä ja jonkin verran ehkä Fortrania on myös toteutuskielenä. Useimmiten ohjelmiston kehityslaitteistona on PC tai Unix-työasema. Laiteläheiseen ohjelmointiin käytetään joko C:tä tai assembleria sekä vastaavia toimittaja- tai prosessorikohtaisia työkaluja.

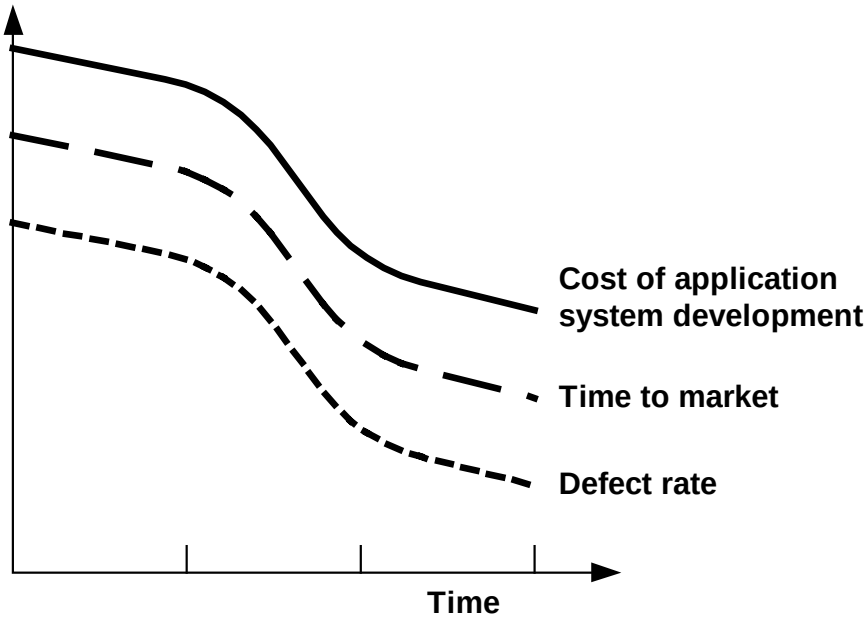
COTSien käyttö on varsin tavallista tietoliikenteen, tietokantaoperaatioiden ja käyttöliittymän toteutuksessa. Signaali- ja kuvankäsittelyynkin on käytettävissä muualta saatavia moduuleja. Toisaalta yrityksissä on tietty tendenssi olla ohjelmiston suhteen omavarainen, eli mieluummin toteutetaan itse tai alihankitaan. Alihankinnan yhteydessä on tavallista vaatia lähdekoodi ja mahdollinen muu dokumentointi alihankkijalta. Ohjelmistoalihankinta sinänsä on alalla tavallista.

Ohjelmiston verifiointi ja validointi, viranomaisiakin varten on varsin uusi asia, johon on paneuduttu, mutta suunnittelijat tiedostavat osaamisessa ja kokemuksessa olevan puutteita. V&V:tä varten on usein eri ihmiset tai eri organisaatio. V&V ja sen vaatimaa dokumentointia ei useimmiten pystytä käyttämään hyödyksi varsinaisen tuotekehityksen aikana, vaan se tehdään hyötyjen saamisen kannalta liian myöhään ja pääasiassa, koska viranomaiset ovat alkaneet vaatia sitä. Ohjelmiston ja järjestelmän testaus on sinänsä laajamittaista ja vie paljon resursseja ja aikaa. Vaatimustietokannat tai vastaavat ovat tavallisia, mutta linkkejä ohjelmistokehityksen vaiheisiin ei juuri ole, eivätkä hieman vanhemmat työkalut sellaista järkevästi tuekaan.

Lääkintälaiteala kansainvälistyy ja yritysten omistuksissa tapahtuu kiihtyvällä vauhdilla muutoksia. Toimialajärjestelyillä usein haetaan kustannussäästöjä ja tuotesynergioita. Koska ohjelmistot ovat yhä isompia ja monimutkaisempia ja ohjelmistotuotannon liiketoimintaprosesseihin ei ole ehkä kiinnitetty tarpeeksi huomiota, ohjelmistotuotantoon ja sen johtamiseen asetetaan erittäin suuria haasteita, mihin ei ehkä olla varauduttu.

4.3 Siirtyminen tehostettuun uudelleenkäyttöön

Myös lääkintälaitteita valmistavien yritysten menestyminen riippuu enenevässä määrin siitä, kuinka hyvin ne onnistuvat tehostamaan ohjelmistotuotantoprosessejaan. Yksinkertaisesti voidaan sanoa, että toiminnan on oltava sekä nopeaa (menestyvät yritykset saavat tuotteensa ja palvelunsa markkinoille oikeaan aikaan), laadukasta (ei virheitä tuotteissa tai palveluissa, tuotteet ja palvelut odotusten mukaisia) että pienin kustannuksin tapahtuvaa (tuotteiden ja palvelujen tuottaminen sekä ylläpito). Näihin tavoitteisiin päästään parantamalla ohjelmistotuotantoa yleensä, jolloin keskimäärin päästään ehkä 5–10 %:n vuosittaiseen tehokkuuden kasvuun. Mutta ennen kaikkea tavoitteisiin päästään siirtymällä tietoisesti systemaattiseen uudelleenkäyttöön, jolloin on voitu todeta, että nopeus (time-to-market) on noussut 2–5-kertaiseksi, virheet ovat vähentyneet puoleen tai kymmenenteen osaan ja ylläpitokustannukset ovat samoin vähentyneet puoleen tai kymmenenteen osaan (kuva 5). Tällaiset selkeät parannukset ovat olleet mahdollisia vain, kun nk. ohjelmistotuotannon uudelleenkäytön aste on noussut näissä yrityksissä 40–95 %:iin.



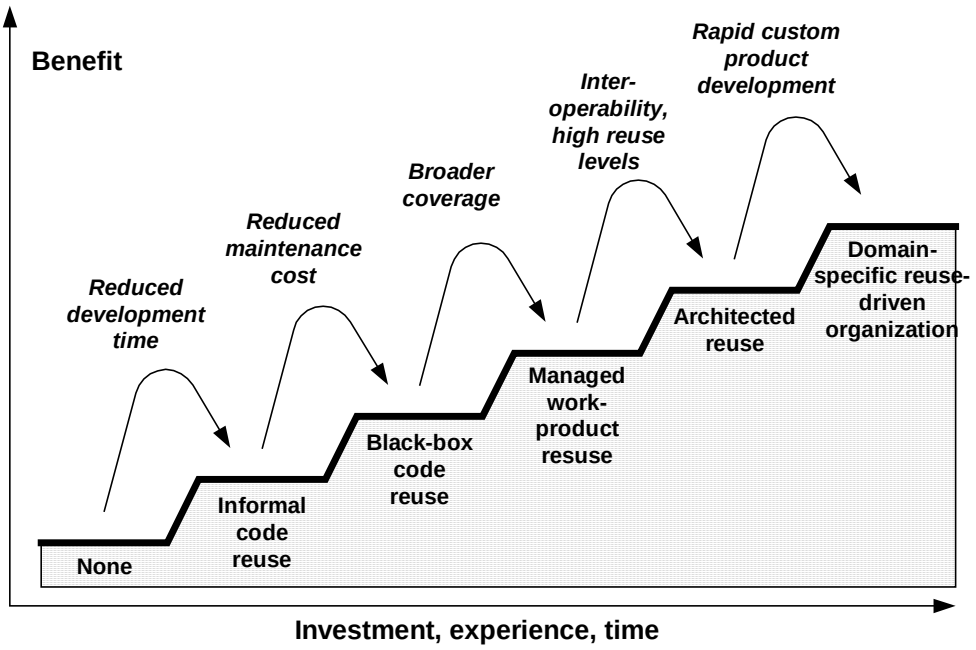
Kuva 6. Systemaattisella uudelleenkäytöllä nopeus (time-to-market) kasvaa, virheet ja ylläpitokustannukset vähentyvät.

Komponenttipohjaisella ohjelmistokehityksellä (component based software development, CBSD) tarkoitetaan sekä suunnittelutekniikoita että -työkaluja, joiden avulla sovelluksia kootaan uudelleenkäytettävistä valmisohjelmista tai ohjelmistokomponenteista laajamittaisen ohjelmoinnin sijasta. Komponenteista kootaan (component assembly) sovelluksia, joissa komponentit kommunikoivat rajapintojensa kautta. Komponenttipohjaista ohjelmistokehitystä on kahta päätyyppiä: 1) tuotteen tai sovelluksen kokoaminen markkinoilta hankittavista ohjelmistokomponenteista (Commercial Off-The-Self, COTS), 2) ohjelmistokomponenttien tuottaminen ja kehitysprosessin muuntaminen komponentteihin perustuvaksi, mikä voidaan rinnastaa yrityksen sisäiseksi (tai alihankinta-) tuotteistamiseksi ohjelmistotuotannossa.

Ohjelmistokomponentit liittyvät erottamattomasti tapoihin osittaa isoja ohjelmia. Kuvausta tai suunnitelmaa, joka esittää ohjelmiston jaon pienempiin osiin ja niiden välisiin erilaisiin riippuvuuksiin ja vuorovaikutuksiin, sanotaan ohjelmistoarkkitehtuuriksi. Vaikka ohjelmistoja tuotetaan mitä erilaisimpiin tarkoituksiin, samanlaiset ominaisuus- tai komponenttitarpeet toistuvat, mistä seuraa samanlaisia kontekstitarpeita komponenteille. Menestyvässä uudelleenkäytössä yritykset kykenevät sekä löytämään tuotteistaan ja palveluistaan erilaisia invariansseja että tuottamaan tehokkaat mekanismit markkinoiden usein vaatimiin erilaistuksiin.

Komponentointiin perustuvaa uudelleenkäyttöä ei ole syytä rajoittaa vain ohjelmakoodiin. Se on erittäin tehokasta myös vaatimusmäärittelyssä, analyysimallinnuksessa, toteutussuunnittelussa, implementoinnissa sekä testauksessa ja muussa laadunhallinnassa. 50-vuotinen ohjelmistotekniikan historia osoittaa, että tehokas uudelleenkäyttö on aluksi vaikeaa. Yksittäinen suunnittelija voi tehostaa työtään 10–20 % oman uudelleenkäytön järjestämisellä. Yksittäisessä projektissa muiden työn hyödyntämisellä saavutetaan ehkä toiset 10–20 %:n tehokkuuden lisäys. Vasta joukko samanlaisia projekteja tai vaikkapa saman tuoteperheen eri tuoteversioiden kehittäminen voi tuoda uudelleenkäytön tehot esiin.

Siirtyminen tehokkaaseen ohjelmiston uudelleenkäyttöön ei voi tapahtua hetkessä, se ei tapahdu vahingossa tai itsestään eikä ilman kustannuksia. Se vaatii uusia taitoja alkaen halukkuudesta käyttää muiden koodia, jatkuen ohjelma- ja suunnittelutietokannan hallinnalla (jotka vähitellen jalostuvat sovellusarkkitehtuureihin ja niiden vaatimiin komponentteihin), ja saavuttaen viimein erilaisia komponentointia hyödyntäviä systemaattisia toimintamuotoja. Koska muutos on perusteellinen, se on suunniteltava ja johdettava hyvin – ja etenemään askeleittain (kuva 7).



Kuva 7. Askeleittain etenevä siirtyminen tehokkaaseen ohjelmiston uudelleenkäyttöön.

Seuraavat periaatteet kannattaa muistaa:

- varmista riittävän pitkäikäinen johdon sitoutuminen ja resursointi
- suunnittele ja toteuta siten kuin tarvitaan: järjestelmä- ja ohjelmistoarkkitehtuuri, ohjelmistokehitysprosessit sekä organisaation osaaminen kohti systemaattista uudelleenkäyttöä; aloita pienellä pilotilla ja laajenna vähitellen. Aloita arkkitehtuureista, jotka laajentuvat joustavasti.
- siirry vähitellen organisaatioon, jossa komponenttien tuotanto on erillään sovellusprojekteista
- tee uudelleenkäytön tuloksia todelliseen käyttöön ja tarpeeseen
- pidä kehittyvää komponenttien kokoelmaa ja arkkitehtuureja tuoteportfoliona, jolla on taloudellinen arvo. Siten uudelleenkäyttö myös kohdistuu sellaisiin sovelluksiin ja niiden osiin, joissa tuotot ovat suuret.
- korosta alusta alkaen myös muutosta organisaatiokulttuurissa. Hyödynnä muutosagentteja ja menestystarinoita.
- Panosta infrastruktuuriin, koulutukseen ja ammattitaitoon kehittämällä niitä jatkuvasti
- mittaa ja optimoi edistymistä vakuuttaaksesi itsesi ja muut.

4.3.1 Sisäinen tuotteistaminen

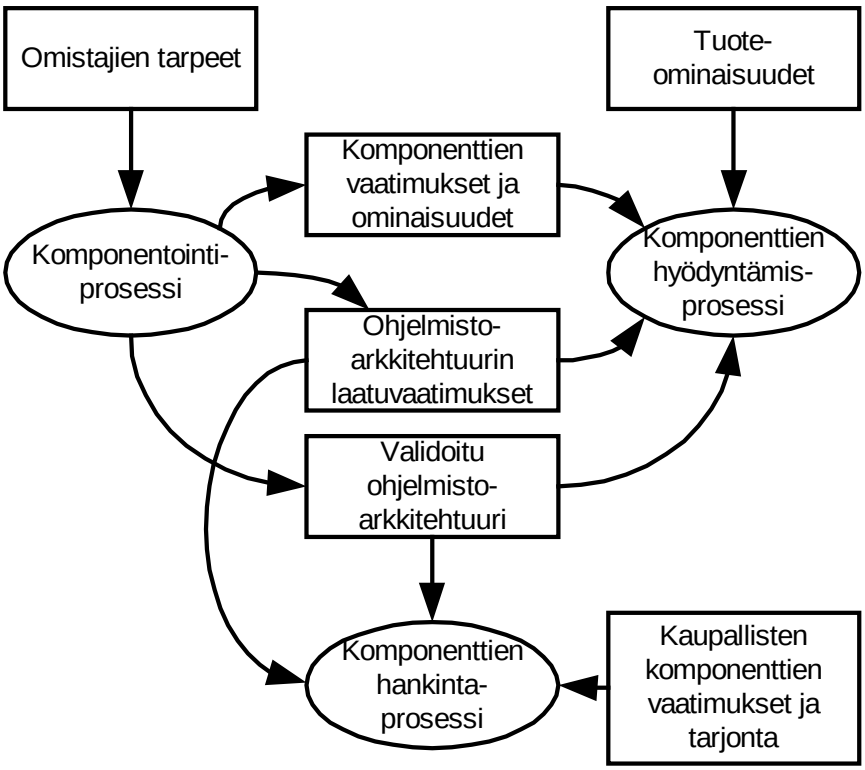
Ohjelmistokomponentit liittyvät erottamattomasti tapoihin osittaa isoja ohjelmia. Kuvausta tai suunnitelmaa, joka esittää ohjelmiston jaon pienempiin osiin ja niiden välisiin erilaisiin riippuvuuksiin ja vuorovaikutuksiin, sanotaan siis ohjelmistoarkkitehtuuriksi. On tärkeää, että iso ohjelma ositetaan järkeviin kokonaisuuksiin niin, että osat riippuvat vain vähän ja selkeällä tavalla toisistaan, mikä edellyttää nk. hyvää arkkitehtuurisuunnittelua. Vaikka ohjelmistoja tuotetaan mitä erilaisimpiin tarkoituksiin, samanlaiset ominaisuus- tai komponenttitarpeet toistuvat ja, mikä on yhtä tärkeää, myös samanlaiset arkkitehtuurisuunnittelutarpeet toistuvat, mistä seuraa samanlaisia kontekstitarpeita komponenteille. Tällaista hyödyllistä samanlaisuutta voi ohjelmistotuotannossa edistää kahdella päätavalla :

1) Tuotetaan ohjelmistoja vain samalle sovellusalueelle, jolloin päästään hyödyntämään tarpeiden ja vaatimusten samanlaisuutta (johtaa esimerkiksi nk. ohjelmistoperheisiin tai perusjärjestelmiin, joista konfiguroidaan tai muuten sovitetaan yksittäisiä toimituksia).

2) Toistetaan samankaltaisten arkkitehtuuriongelmien ratkaisuja samanlaisilla rakenteisilla ratkaisuilla (johtaa nykyisin tärkeiden nk. architectural design patternien käyttöön).

Abstraktia ohjelmistoluurankoa, jolla määritellään millaisia komponentteja ko. luurankoon liittyvissä ohjelmistoissa on ja mitkä ovat niiden suhteet, sanotaan frame-workiksi (viitekehyksiksi). Valitsemalla komponentit ja viimeistelemällä muutenkin rakennetta saadaan (so. instantioidaan) ohjelmistoarkkitehtuuri. Siten nk. domain specific frameworkeillä on tärkeä sija kehitettäessä yrityksen tietotekniikkastrategiaa. Muutamilla aloilla (domain) framework-tason standardointi on ehtinyt jo pitkälle, mikä on edellytys komponenttimarkkinoiden kehittymiselle.

Ohjelmistoarkkitehtuurit tai viitekehykset ja komponentit ja ovat siten lähes erottamattomat. Mitä spesifisemmistä komponenteista on kyse, sitä ainutkertaisempaan kehykseen se sopii ja päinvastoin. Edelleen, taitavasti ja kokeneesti suunnitellut arkkitehtuurit taipuvat tehokkaalla tavalla monenlaisiin käyttäjävaatimuksiin menettämättä komponenttiensa uudelleenkäyttöastetta paljoakaan.



Kuva 8. Komponenttien kehittämis-, hankinta- ja hyödyntämisprosessien vuorovaikutus.

Kun sovellusarkkitehtuurit ovat yritys- tai yritysryhmäkohtaisia, ohjelmistotuotannon on mahdollista muuttua tuotantolinjamaiseksi. Yritys tai yritysverkko määrittelee hyödyntävänsä arkkitehtuurin tai mukautuu markkinasegmentillään vallitsevaan arkkitehtuuriin. Ohjelmistotuotannossa sekä uudelleenkäyttö tehostuu että luodaan mekanismit ohjelmistotuotteen tehokkaaseen erilaistamiseen tai asiakaskohtaistamiseen. Ohjelmistotuotelinjan kehittäminen asettaa organisaatioille ja henkilöstölle erilaisia vaatimuksia riippuen siitä, missä kehitysvaiheessa tuotelinja on ja millaisille markkinoille tuotteet on suunnattu (Kuva 8).

4.3.2 Kaupallisten ohjelmistojen käyttö

Systemaattinen ja ammattitaitoinen uudelleenkäyttö luo kestäviä puitteita myös ostajayrityksen ehdoilla tapahtuvaan valmisohjelmien (COTS) hyödyntämiseen sekä hyvin organisoituun ohjelmistoalihankintaan. Arkkitehtuurisuunnittelun esiin nostama komponentti voidaan joko tehdä itse, ostaa markkinoilta tai alihankkia. Tietyillä yleisillä alueilla (käyttöliittymät, tietokannat, tietoliikenne, signaalinkäsittely, numeerinen laskenta jne.), joiden vaatimaa arkkitehtuurikontekstia voidaan pitää nk. yleisesti tunnettuna tai arkkitehtuurisidonnaisuus on löyhää, on toisaalta ollut olemassa menestyvääkin liiketoimintaa. Edelleen pitänee paikkansa, että tällä tavalla olemassa oleva komponenttitarjonta vaikuttaa soveltajayritysten ohjelmistosuunnitteluun.

Toisinaan yritykset niin sanotusti tekevät ohjelmistoarkkitehtuureistaan avoimia, jolloin periaatteessa kaikki voivat tehdä niiden kanssa yhteensopivia komponentteja ja osajärjestelmiä. Tässä mielessä avointen viitekehysten kehittyminen ja yleistyminen on avainasemassa ohjelmistokomponenttimarkkinoiden kehittymiselle. Ääriesimerkkinä voidaan pitää Microsoftin käyttöjärjestelmiä, jotka ovat myös avoimia (hyvin laajoja eikä niin alaspesifisiäkään) arkkitehtuureja, ja jotka siten ovat tasoittaneet tietä laajalle sovellusohjelmistomarkkinalle. (Ohjelmistoarkkitehtuurin hallitsijana Microsoft on toki saanut kritiikkiäkin toimintatavoistaan!)

4.3.3 Yrityksen oman vanhan ohjelmiston hyödyntäminen

Varsin tavallinen tilanne alalla pitkään toimineissa yrityksissä on, että olemassa olevaa ohjelmistoa on periaatteessa runsaasti, mutta vanhentuneeseen käyttöympäristöön, vanhentuneilla platformeilla, vanhentuneilla ohjelmointikielillä jne. Valitettavan usein tällaiset ohjelmistot ovat huonosti dokumentoituja, jolloin niiden käyttökelpoisuutta voi olla vaikeaa tai työlästä arvioida. Hieman vastaavia tilanteita on runsaasti myös nk. toimialarationalisointien yhteydessä, jolloin on yhdistettävänä useita ohjelmistovarantoja, tuotteita ja tuoteperheitä, organisaatiokulttuureja jne.

On olemassa ainakin kaksi periaatteellista lähestymistapaa pidentää vanhan ohjelmiston elinikää: re-engineering yleisesti sekä laajemmin ajateltuna erilaiset legacy systemsien hallinnan periaatteet.

1) Re-engineering tarkoittaa arvioitavana olevan, yleensä ohjelmakoodin ja ääritapauksissa binäärikoodin, abstraktiotason nostamista niin korkealle, että ohjelman rooli uudessa ympäristössä voidaan järkevästi ja luotettavasti ottaa huomioon. Toisin sanoen joudutaan tulemaan ohjelman suunnitteluvaiheita taaksepäin. Tehtävään on olemassa automaattisia työkalujakin, jotka kuitenkin ovat yleensä kalliita, ja ne toimivat usein liian kaavamaisesti – esimerkiksi tuottavat hyvin suuren määrän oliodiagrammeja, että ne eivät todellisuudessa nosta abstraktiotasoa; tulostavat sen vain uuteen, ehkä vaikeaselkoisempaankin formaattiin. Manuaalinen re-engineering on siten suositeltava, mutta työläs tapa, jos koodimassaa on paljon.

2) Legacy systems: Tavallinen lähtötilanne on nykyään, että organisaatiossa on käytössä vanha, samaa liiketoiminta-aluetta palveleva ohjelma, jota syystä tai toisesta harkitaan uusittavaksi. Uusimispaineita aiheuttavat ennen muuta ohjelmiston käyttöympäristössä tapahtuvat ja tapahtuneet muutokset, jolloin vanha ohjelma – so. malli todellisuudesta – ei enää vastaa muuttunutta tilannetta. Todellisuus vanhaa ohjelmaa spesifioitaessa ja suunnitellessa oli eri tai se nähtiin eri tavalla kuin nykytodellisuus. Ohjelmistolta vaadittavat ominaisuudet muuttuvat. Erilaiset kapasiteetti- ym. rajoitukset aiheuttivat aikaisemmin kohtuullisina pidettäviä suunnittelukompromisseja, mutta nyttemmin näiden kompromissien takia vanha ohjelma näyttää tulleen suorituskykynsä äärirajoille. Yksittäisen ohjelman muu tietotekniikkaympäristö muuttuu, ja vanhalta ohjelmalta vaadittaisiin ehkä nyt aiempaa joustavampaa yhteiskäyttövalmiutta muiden ohjelmien kanssa.

Ohjelma voi vanhentua myös sisäisesti, eli se on toteutettu vanhalla ohjelmointikielellä, sitä voidaan ajaa vain vanhoissa tietokoneissa tai käyttöjärjestelmissä tms. Muuttunut käyttöprofiili voi myös laukaista ohjelmassa piileviä vikoja, joihin ei aiemmin törmätty, aikana jolloin korjaaminen olisi ollut vielä helppoa. Ohjelma on myös saattanut läpikäydä useampia ympäristöporttausmuutoksia ja ohjelman selkeys on kärsinyt. Toisinaan muutospaineen aiheuttaa myös se nähtävissä oleva hyöty- ja mahdollisuuksien potentiaali, joka uusitulla ohjelmistolla voisi olla.

Vanha, sellaisenaan toimiva ohjelma voi olla inhimillisestä näkökulmasta niin laaja ja monimutkainen, että kaikkia riippuvuuksia ei tunneta. Vanhat ohjelmistosuunnittelumenetelmät (strukturoitu ohjelmointi, SA ym.) eivät sittenkään tukeneet kovin modulaarista ohjelmistosuunnittelua, joten muutos yhdessä paikkaa ohjelmaa heijastui helposti ohjelman käyttäytymisen odottamattomina muutoksina muualla. Vanha ohjelma saattaa olla myös vanhaan käyttötarkoitukseensa tiukasti optimoitu, jolloin koodi on vaikeam-

min tulkittavaa ja ymmärrettävää ja siten myös vaikeammin muutettavaa. Optimin karkaaminen olisi ehkä vähäisempi haitta.

Vanhan ohjelman dokumentointi on usein puutteellinen. Toisinaan dokumentointia ei ole tehty alkujaankaan, tai se on yksinkertaisesti hävinnyt. Pitkään elinkaareen mahtuu myös muutoksia, uusintaversioita, joita ei ehkä ole dokumentoitu kunnolla, tai kunnollista versionhallintaa ei ole ylläpidetty. Vanhassa ohjelmassa saattaa olla osuuksia, jotka on ostettu alihankintana, ja alihankkijaa ei enää ole, tai vaikka tällainen ohjelmanosa olisi suunniteltu omassa organisaatiossa, avainhenkilöt ovat vaihtuneet, ja edellytykset ohjelman parantamiseen ovat huonot. Vanha ohjelma saattaa olla myös niin kiinteästä operatiivisessa käytössä, että muutosten kokeiluihin ei katsota olevan tuotantomenetysten pelossa varaa.

Vanhan järjestelmän käyttöiän pidentäminen on ehkä epämuodikas strateginen päätös. Suoraviivaisia, mutta ei riskittömiä menettelyjä ovat erilaiset nk. porttaukset, eli ohjelma muunnetaan uudelle ohjelmointikielelle, ohjelma työstetään ajettavaksi uudessa käyttöjärjestelmässä jne. Arkisin ja tavallisin käyttöiän pidentämistekniikka on yksinkertaisesti toteuttaa tarvittavat muutokset, mitä myös versiohallinnaksi tai ohjelmistoylläpidoksi kutsutaan. Yksi käyttöiän pidentämistekniikkatyyppi on nk. wrapper, joka on ohjelmapalanen vanhan ohjelman ja muun ohjelmiston välissä. Vanhaan ohjelmaan päin wrapper toteuttaa kaikki tai tarvittavat input- ja output- rajapinnat, ulospäin wrapper toteuttaa uuden konseptin tai infrastruktuurin mukaiset interfacet (esim. COM, CORBA). Wrapper siis tulkitsee interaktiot muun ohjelmistoympäristön kanssa vanhalle ohjelmalle ymmärrettävään muotoon. Jossakin vaiheessa wrapper-strategia saattaa tulla työläämmäksi kuin uuden ohjelman tekeminen.

Maailmalla on myös muutamia menestystarinoita vanhojen ohjelmien restauroinnista, jolloin ohjelmasta puretaan kaikki puutteet, viat, vanhat suunnittelukompromissit ja muut vanhentuneet sidonnaisuudet, toteutetaan tarvittavat laajennukset ja uudistukset jne. Historiallisten taideteosten tapaan vanhat ohjelmat koetaan silloin erittäin arvokkaiksi ja säilyttämisen arvoisiksi. Taiteen restauroinnit vaativat yleensä 3–5-kertaisen työpanoksen teoksen alkuperäiseen tuottamiseen verrattuna, ja sama suhdeluokka pätee ehdottomasti myös nykyajan taideteoksille eli hienoille, isoille tietokoneohjelmille, mutta toisinaan se on ehdottomasti kannattanut.

Hallittu, perusteltu, strukturoitu päätöksenteko on avainasemassa näiden lähestymistapojen hyödyntämisessä. Eli on osattava objektiivisesti arvioida vanhan järjestelmän puutteet, on kyettävä arvioimaan vanhan järjestelmän ylläpito- ja muutoskustannukset (myös välilliset kustannukset) jos käyttöikää pidennetään, on arvioitava uuden järjestelmän kehittämisen kustannukset ja riskit. On kiinnitettävä myös huomiota siihen, että ohjelmistokehitysprosessit ovat korkeatasoisia ja ammattitaidolla hoidettuja.

4.4 COTS-ohjelmistot PC-ympäristössä

COTS-käsitteelle (commercial-off-the-shelf) annetut määritelmät vaihtelevat jonkin verran. Yleisesti ottaen termillä tarkoitetaan vapaasti myynnissä olevaa tuotetta, jonka ostaja voi ottaa osaksi omaa järjestelmäänsä. ”Kaupallinen” määrittää tuotetta kahdella tavalla:

1) Valmistaja pyrkii tekemään tuotteestaan mahdollisimman yleiskäyttöisen, toisin sanoen saamaan samalla tuotantokustannuksella mahdollisimman paljon asiakkaita.
2) Tuotanto ei ole valtiollisen elimen välittömässä hallinnassa (oleellinen kysymys esimerkiksi sotilaallisissa sovelluksissa).

Automaation ollessa kyseessä termillä voidaan tarkoittaa niin elektroniikkaa kuin ohjelmistoa, joista nyt keskitytään jälkimmäiseen. Tavallisesti ostajalla ei ole minkäänlaista hallintaa COTSin valmistajan ohjelmistontuotantoprosessiin eikä oikeutta saada yksityiskohtaisia dokumentteja COTSin tuotannon eri vaiheista. Nämä ovat vaatimustenmukaisuuden arvioinnin kannalta COTSeille ominaisia piirteitä, ja seuraavassa analyysissä COTSilla tarkoitetaan tällaisia ohjelmia, riippumatta siitä, ovatko ne ilmaisjakelua, sharewarea vai kaupallisia ohjelmia.

Näkökulman mukaan myös sellaiset tietokonejärjestelmien perusohjelmat, kuten käyttöjärjestelmät, laiteajurit, kääntäjät, linkkerit yms. lasketaan COTSeiksi. Esimerkiksi FDA:n kanta on, että käyttöjärjestelmät tulkitaan COTSeiksi (ks. FDA 1999a: kohta III A). Tässä tutkimuksessa joudutaan resurssien rajallisuuden vuoksi jättämään nämä systeemiohjelmistot analyysin ulkopuolelle ja keskittymään seuraaviin COTSeihin, joiden katsotaan edustavan lääkintälaitealalla tyypillisiä käyttökohteita:

- graafinen käyttöliittymä
- tietokantarajapinta
- yksinkertainen peruskuvankäsittely.

Yleiskäyttöisten PC-koneiden ja niiden käyttöjärjestelmien yleistyessä erilaisissa laiteohjaustehtävissä on COTS-ohjelmiin kohdistuva kiinnostus kasvanut. Esimerkiksi tietokantasovellusten ja graafisten käyttöliittymien ohjelmoiminen alusta asti itse on monen erikoisalan ohjelmistotuottajan kannalta epätaloudellista, kun kaupan hyllyltä löytyy useita valmiita ratkaisuja, jotka vaativat parhaimmillaan vain hieman konfigurointia toimiakseen. Laitevalmistajan näkökulmasta COTSeista onkin hyötyä silloin, kun ohjelmistosta pystytään löytämään suunnitteluvaiheessa sopiva lohko, joka voidaan toteuttaa COTS-ohjelmalla. COTSien käytössä on paljon yhteisiä piirteitä kuin alihankin-

nassa yleensä, tarjouskilpailusta toimitusten jatkuvuuden varmistamiseen, joskin ohjelmistolla ja vaikkapa mekaanisella komponentilla on eroja, jotka joudutaan huomioimaan. Seuraavassa on lueteltu erityisesti COTSien käyttöön liittyviä riskejä. On muistettava, että näiden lisäksi COTSeilla on samoja ongelmia kuin itse tuotetuilla ohjelmilla. Tämän jälkeen esitetään joitain kirjallisuudessa esitettyjä keinoja riskien minimoimiseksi. Lopuksi esitetään hyvin suppea katsaus FDA:n alustaviin COTS-vaatimuksiin.

4.4.1 COTS-riskit

Hankintavaihe

Järjestelmän suunnittelun alkuvaiheissa on COTSille määritelty joukko ominaisuuksia, jotka sen on täytettävä. Tästä vaatimusmäärittelystä ilmenevät esimerkiksi COTSin syöttö- ja ulostulorajapinnoille sekä toiminnallisuudelle asetetut ehdot. Tässä osiossa keskitymme erityisesti niihin vaatimuksiin, jotka kohdistuvat tiettyyn suorituskelpoiseen ohjelmaan määrättynä ajanhetkenä. Tähän vaiheeseen liittyvät riskit syntyvät siitä mahdollisuudesta, että COTS-ohjelma ei hankintahetkellä täytä sille asetettuja perusvaatimuksia.

Hankittavan COTS-ohjelman perusteellinen testaaminen ei useinkaan ole käytännössä mahdollista ohjelmien monimutkaisuuden vuoksi. Kuten yllä mainittiin, on ulkopuolisen ohjelmistotuottajan prosessista usein mahdoton saada tietoa ja testiraporttienkin saaminen voi olla hyvin vaikeaa: tämä riippunee pitkälti siitä, miten tärkeäksi COTSin valmistaja kyseisen ostajan kokee. Siinäkin tapauksessa, että testiraportteja saadaan, ei ostaja voi enää vaikuttaa testattaviin asioihin eikä voi varmistua esimerkiksi yksikkötestitulosten luotettavuudesta. Näin ollen on hyvinkin mahdollista, että ohjelmassa olevat puutteet ja virheet jäävät havaitsematta.

Ei-halutut toiminnot voivat tavanomaisten ohjelmointivirheiden lisäksi johtua valmistajan tahallisesti asettamista salaovista. Salaovi aktivoidaan sen tekijän tuntemalla herätteellä, jolloin tapahtuu käyttäjän kannalta jotain odottamatonta ja usein haitallista. Tuhoisien salaovien asettaminen ei kuitenkaan ole kaupallisen ohjelmistotuottajan edun mukaista, joten tällaisten todennäköisyys ei vaikuta kovin suurelta. Koska COTS on ajettava ohjelma, sen mukana voi tulla yksi tai useampia tietokoneviruksia. Näiltä voidaan kuitenkin suojautua melko tehokkaasti perinteisin virustorjuntakeinoin.

Näin ollen hankittavan COTS-ohjelman virheellisten toimintojen todennäköisimmät syyt ovat ohjelmointivirheissä. Kuten jo aiemmin todettiin, ohjelman täydellinen testaaminen on ostajalle mahdotonta. Sitä se on myös valmistajalle. Niinpä useimmat valmistajat pyrkivät ohjaamaan testausresurssejaan mahdollisimman tehokkaalla tavalla. Tämä voidaan toteuttaa esimerkiksi käyttöprofiilien avulla. Tällöin testejä pyritään ti-

lastollisessa mielessä painottamaan käyttäjien oletettua käyttötapaa vastaavaksi. Mikään ei kuitenkaan takaa sitä, että COTSin valmistajan laatimat profiilit vastaisivat tietyn ostajan tapausta. Esimerkiksi lääkintälaitesovelluksen tapa käyttää signaalinkäsittelypakettia voi poiketa oleellisesti ohjelman laatijan ”kohdeyleisön” tavasta. Niinpä kyseisen ostajan kannalta oleelliset toiminnot ja sekvenssit ovat saattaneet jäädä hyvin vähäiselle testaamiselle, jolloin virheitä voi helpommin esiintyä.

Toisaalta, jos sama COTS-ohjelma on ollut todistettavasti pitkään menestyksellisesti käytössä vastaavissa olosuhteissa kuin mihin sitä ollaan hankkimassa, sen luotettavuudelle voidaan katsoa kertyneen kokeellista näyttöä, jollaista on vaikea itse tuotetulle ohjelmalle nopeasti saada ja jonka merkitystä ei tule ylenkatsoa.

Systeemi-integraation ongelmat

COTSin hankintahetken tulisi tavallisesti ajoittua varsin aikaiseen vaiheeseen ohjelmistotuotannossa, sillä ohjelman käytön oppimiselle, testaamiselle ja integroimiselle muuhun järjestelmään on varattava riittävästi aikaa. Niinpä hankintahetkellä ei useinkaan voida määrätä COTSin dynaamisia ominaisuuksia kovinkaan tarkasti, sillä muu systeemi ei ole riittävän valmis. On siis olemassa riski, että vaikka ohjelma hankintahetkellä täyttäisikin kaikki staattiset vaatimukset, integraatiovaiheessa ohjelma saattaa osoittautua epätarkoituksenmukaiseksi. Erityisesti aika-kriittisissä sovelluksissa tämä voi olla ongelma, varsinkin jos COTSin sisäistä rakennetta tunnetaan niin huonosti, että suoritusaika-arvioita on vaikea tehdä (kasvaako lineaarisesti, neliöllisesti jne.)

Ostaja ei voi myöskään olla varma siitä, että kaikki COTSin tarvitsemat resurssit ja riippuvuudet tunnetaan. Esimerkiksi Windows-ympäristössä tämä voi johtaa vakaviin ongelmiin seuraavasti: oletetaan, että COTS käyttää jotain DLL-tiedostoa (dynamic linking library), jota ostaja ei tunne. Ohjelmisto toimii aluksi oikein. Uuden Windows-version, huoltopäivityksen tai jonkun muun ohjelman asennuksen yhteydessä DLL muuttuu. Tämän seurauksena myös COTSin toiminta muuttuu käyttäjälle yllättävällä tavalla.

Mikäli ohjelmaan integroidaan useita eri COTS-ohjelmia, on hyvin todennäköistä, että näillä kaikilla on omat tietorakenteensa. Rakenteiden kirjavuus voi hankaloittaa suunnittelua ja toteutusta. Yksinkertaisena esimerkkinä voitaisiin ajatella kahta matriisien käsittelyyn tarkoitettua ohjelmaa. Ensimmäisen ohjelma aloittaa rivien ja sarakkeiden numeroimisen nollasta, toinen puolestaan ykkösestä. Ei ole mahdotonta kuvitella tilannetta, jossa ohjelmoija indeksoi väärin, koska sotkee kutsuttavat ohjelmat.

Käyttöönotto ja käyttö

Koska COTSin valmistaja pyrkii saavuttamaan tuotteellaan mahdollisimman suuren asiakaskunnan, on siinä todennäköisesti ostajan haluamien ominaisuuksien lisäksi myös ylimääräisiä ominaisuuksia. Viime kädessä tämä johtaa omaa ohjelmaa suurempaan konfigurointitarpeeseen. Tällöin väärän konfiguroinnin mahdollisuus kasvaa. Väärästä konfiguroinnista voi seurata ohjelman virheellinen tai puutteellinen toiminta.

Väärä konfigurointi voi tapahtua jo tuotantovaiheessa, asennettaessa järjestelmää asiakkaalle tai käytön aikana. Asennuksen jälkeen riskin muodostavat uusien ohjelmien asennukset, jotka saattavat automaattisesti muuttaa konfigurointia tai saattavat toimiakseen vaatia käyttäjältä muutoksia. Käyttäjä ei välttämättä ymmärrä tekemiensä muutosten liittyvän mitenkään alkuperäiseen ohjelmaan. Tällaisia ristiriitoja saattaisi syntyä esimerkiksi tietokannan asetusten kanssa.

Ylläpito

Ylläpidon tehtävänä on säilyttää koko laitteen palvelutaso ajan kuluessa sekä mahdollisesti toteuttaa päivitysluonteisia parannuksia. Omia ohjelmia voidaan jatkuvasti kehittää. Tyypillisesti tämä tapahtuu siten, että kerätään tietoja vikaantumisista, ohjelmakoodista etsitään virheelliset kohdat, tarvittava korjaus suunnitellaan, dokumentoidaan ja toteutetaan ja korjausversio asennetaan hallitusti asiakkaan järjestelmään. Kaikki tämä vaatii selvää käsitystä oman ohjelman toiminnasta, ohjelmiston on oltava organisaation älyllisessä hallinnassa. COTSien tapauksessa näin ei ole. Vaikka ilmiselviä vikoja havaittaisiin, niitä ei yleensä pystytä korjaamaan. Yleensä korjauksissa joudutaan tyytymään kiertoteiden etsintään ja vikatilanteiden estämiseen, työskentelemällä siis COTSia ympäröivän oman koodin kanssa. Tämä todennäköisesti heikentää laitteen palvelutasoa. Vaikka COTSin valmistaja myöntyisi tarvittaviin muutoksiin, kestävät nämä todennäköisesti kauemmin kuin itse tehtäessä.

Samantyyppinen ongelma on edessä silloin, kun asiakkaan käyttötarve muuttuu ja asiakas esittää voimistuvia toivomuksia tietyn toiminnallisuuden lisäämiseksi. Mikäli toimintojen toteuttaminen vaatii muutoksia COTSiin, on tilanne jälleen hankala. Toisinaan tämä ongelma voidaan ratkaista hankkimalla uusi versio COTSista tai kilpailijan tuote. Tällöin kuitenkin joudutaan testaamaan ja varmistumaan koko COTSin luotettavuudesta uudelleen.

On myös tarkoin harkittava COTSin päivittämiseen liittyviä kysymyksiä. COTSin valmistaja saattaa väittää uuden ja vanhan version olevan täysin yhteensopivia vanhojen toimintojen osalta. Kokemus on kuitenkin osoittanut, että näin ei aina käytännössä ole.

4.4.2 Keinoja riskien pienentämiseksi ja haittojen minimoimiseksi

Suunnitellun ja dokumentoidun valintamenettelyn käyttö

Kun halutut toiminnot sisältäviä COTSeja on markkinoilla useita, voidaan riskejä pienentää kiinnittämällä valinnassa huomio oikeisiin asioihin. Muiden käyttäjien haastattelut, tuoteinformaation läpikäynti, oikeiden kysymysten esittäminen ja dokumenttien vaatiminen ym. keinot auttavat muodostettaessa kuvaa COTS-ohjelman sisältämistä riskeistä. Web-sivulla http://www.sei.cmu.edu/str/descriptions/cbsd.html ”Component-Based Software Development / COTS Integration”, on annettu muutamia esimerkkejä valintakriteereitä sisältävistä standardeista.

N-versio-ohjelmointi

Yksi tapa vähentää ohjelmistosuunnittelun ja ohjelmoinnin aiheuttamien vikatilanteiden lukumäärää on N-versio ohjelmointi. Tällä tarkoitetaan sitä, että ohjelman kriittisiksi todetuista lohkoista tehdään N (N= 2 tai 3 tyypillisesti) versiota mahdollisimman riippumattomasti. Lisäksi ohjelmassa on oltava tarvittavat toiminnot eri lohkojen ajamiseen ja niitten tulosten vertaamiseen. Mikäli kaikki lohkot antavat saman tuloksen, voidaan jatkaa suoraan. Muutoin tilanteen mukaan suoritus voidaan lopettaa tai käytetään äänestysprosessia todennäköisimmän tuloksen löytämiseksi. COTSien tapauksessa tätä voitaisiin soveltaa hankkimalla eri valmistajilta COTS-ohjelmat samaan tarkoitukseen. Riippumattomuudelle asetetut vaatimukset toteutunevat, jos tuottajilla ei ole yhteyksiä.

Menetelmän haittoja ovat “ylimääräisten” COTSien lisenssimaksut, äänestysprosessin toteuttamisesta aiheutuvat kustannukset sekä useiden lohkojen ajamisen hidastava vaikutus. Usean COTSin käyttämisestä samassa ohjelmassa aiheutuvia riskejä on jo käsitelty edellä. Lisäksi voidaan aina kysyä, toimiiko äänestysprosessi luotettavasti. On myös esitetty, että tietyntyyppiset virheet esiintyvät toisistaan riippuen, jolloin menetelmä ei takaa kaikkien virhetyyppien suhteen luotettavaa toimintaa.

Redundanssi

Menetelmän perusajatuksena on, että ohjelmiston kriittinen osa voidaan jakaa luotettavaan, mutta yksinkertaiseen ytimeen ja toisaalta epäluotettavaan, mutta tehokkaaseen ja monipuolisempaan ulkokerrokseen. Ulkokerros ohjaa laitteen toimintaa ytimen tarkkaillessa systeemin tilaa jatkuvasti. Mikäli järjestelmän tila alkaa ajautua pois siitä tilajoukosta, jonka ydin kykenee hallitsemaan, ydin siirtää ohjauksen itselleen ja resetoi ulomman kuoren. Kun systeemi on saatettu turvalliseen perustilaan, palautetaan laitteen ohjaus taas tehokkaammalle ulkokuorelle. COTS-tapauksessa tämä voisi tarkoittaa sitä, että ostaja toteuttaa ostettavasta toiminnallisuudesta yksinkertaistetun version itse ja asettaa COTS-ohjelman ulkokuoreksi.

Yksi menetelmän ongelma on, että jostain täytyy löytyä luotettava ydin, mikä ei aina ole helppo tehtävä. Toiseksi käyttökelpoisuutta rajoittaa se, että järjestelmän tilan mittaamisen on oltava helppoa, nopeaa ja luotettavaa, sekä myös se, että sallittu tilajoukko pitää pystyä selvästi rajaamaan. Jaon ytimen ja ulkokuoren välillä on istuttava sovellukseen. Ytimen ja ulkokuoren versionhallinta saattaa myös aiheuttaa omia haasteitaan järjestelmän kehittyessä.

Code wrappers

Muistuttaa toiminnaltaan redundanssia, mutta nyt luotettava osa on ikään kuin “ohut kuori” epäluotettavan (COTS) ohjelman ympärillä. Wrapperi tarkkailee COTSin syötteitä ja ulostuloja. Mikäli nämä eivät ole sallitulla alueella, voidaan käyttäjää hälytyttää ja/tai lopettaa ohjelman suoritus. Tällaisista menetelmistä voi olla hyötyä erityisesti silloin, kun järjestelmän jatkuva ohjauksen säilyminen ei ole kriittistä, vaan suoritus voidaan lopettaa välittömästi ongelmatilanteessa (lentokoneen ohjaus vs. röntgenkuvaus.)

Wrappereiden käyttökelpoisuutta rajoittaa se, että niitä on mielekästä tehdä vain melko yksinkertaisten ja paikallisten ongelmien tunnistamista varten.

Hiekkalaatikko (resurssienkäytön ajonaikainen tarkkailu)

Ohjelmallinen kehikko, joka vahtii ohjelman käyttämiä resursseja, esimerkiksi muistia, prosessoriaikaa ja levytilaa. Tarkoitettu yleensä tilanteisiin, joissa halutaan suojata tietokoneen tiedot silloin, kun tietokoneessa ajetaan vieraita ohjelmia. Soveltamalla saattaisi olla hyötyä COTSienkin tapaukseen.

4.4.3 FDA:n COTS-vaatimusten lyhyt esittely

Seuraavat tiedot perustuvat luonnokseen (FDA 1999a), joten niitä ei tule pitää FDA:n lopullisina vaatimuksina, tällä hetkellä dokumentti ei sido FDA:ta eikä yleisöä. Toisaalta dokumentissa kuitenkin todetaan, että se käsittelee avainasioita, joita FDA:n tarkastajien tulisi laitevalmistajien anomuksista etsiä.

Dokumentissa seuraavat kaksi käsitettä ovat oleellisia:

1) Minimaalinen vaara (minimal hazard): Kun vioittuminen, väärä toiminta tai OTS-ohjelmiston väärinkäyttö ei aiheuta minkäänlaista mahdollisuutta potilaan vakavalle vammautumiselle, OTS-ohjelmiston sanotaan aiheuttavan minimaalisen vaaran.

2) Merkittävä vaara (significant hazard): Kun vioittuminen, väärä toiminta tai OTS-ohjelmiston väärinkäyttö aiheuttaa todennäköisesti kuoleman tai vakavan vamman potilaalle, OTS-ohjelmiston sanotaan aiheuttavan merkittävän vaaran.

Kaikille lääkintälaitteen sisältämille COTS-komponenteille on täytettävä perusvaatimustiedot (basic requirements.) Tarkan tunniste-, toiminta- ja resurssimäärittelyn lisäksi tässä vaiheessa vaaditaan dokumentteja, joilla luvan anoja osoittaa varmistuneensa COTSin oikeanlaisesta toiminnasta, esim. testiraportit. Tämän jälkeen laitevalmistajan on suoritettava vaara-analyysi (hazard analysis). Mikäli laite aiheuttaa pahimmillaankin vain minimaalisen vaaran, asettaa tämä ylärajan COTSin vaarallisuudelle, ja tämä riittää tähän kohtaan. Muutoin joudutaan listaamaan laitteen vaarat, arvioimaan näiden vakavuus ja mahdolliset aiheuttajat. Mikäli tämän jälkeen voidaan todeta, että laitteen aiheuttama vaara on minimaalinen, voidaan prosessi lopettaa. Muutoin on jatkettava vaaran pienentämisvaiheeseen. Mikäli laitteeseen jää minimaalista suurempaa vaaraa, arvioidaan sitä suhteessa saavutettavaan hyötyyn.

4.5 Suositellut käytännöt palvelimien tietoturvan varmentamisessa

4.5.1 Käyttöönoton suunnittelu

Tietokoneiden käyttöönotosta tehdään ohjeistus, jota noudatetaan säännönmukaisesti. Ohjeistus käsittää myös tietoturvakysymykset, eli lukujen 4.5.2 ja 4.5.3 suositellut käytännöt.

Yhdenmukaisella konfiguroinnilla ongelmat voidaan havaita nopeasti järjestelmän normaalista poikkeavana toimintana, joten myös turvallisuuteen liittyvät ongelmat huomataan hyvissä ajoin.

4.5.2 Järjestelmän konfigurointi

Järjestelmän konfiguroinnissa on otettava huomioon seuraavia asioita (CERT 1999):

- ohjelmistojen pitäminen ajan tasalla
- vain tarpeellisten käyttöjärjestelmä- ja verkkopalvelujen tarjoaminen
- käyttäjien todentaminen
- laitteiden ja tiedostojen käyttöoikeuksien asettaminen
- järjestelmän käytön ja verkkoyhteyksien tapahtumien tallentaminen
- varmuuskopioinnin ja tiedonpalautusten suunnittelu
- viruksilta ja vastaavilta suojautuminen
- turvallisesta etähallinnasta huolehtiminen.

Valmistajat julkaisevat ilmitulleiden ongelmien perusteella korjauksia ja päivityksiä käyttöjärjestelmiinsä ja oheislaitteiden laiteohjaimiin. Windows NT:n osalta näytönohjaimen laiteajurit ovat erityisen tärkeitä koko käyttöjärjestelmän vakaudelle, koska ne toimivat muun käyttöjärjestelmän kanssa samalla suojaustasolla.

Eri organisaatiot julkaisevat tiedonantoja havaituista turvallisuusriskeistä ja niiden korjaustoimenpiteistä sitä mukaa kun niitä on saatettu julkiseen tietoon. Seuraavat, lähinnä Yhdysvaltojen liittohallinnon rahoittamat organisaatiot, ovat suositeltavia lähteitä. Ohjelmistojen valmistajat ovat hitaampia tuomaan omien tuotteittensa ongelmia julkisuuteen.

- Computer Emergency Response Team (CERT), www.cert.org.
- Computer Incident Advisory Capability (CIAC), www.ciac.org.
- Computer Operations, Audit, and Security Technology (COAST) www.cs.purdue.edu/coast/coast.html.
- Computer Security Technology Center (CSTC) ciac.llnl.gov/cstc/CSTCHome.html

Nämä myös pitävät yllä turvallisuusongelmiin keskittyviä postituslistoja, joihin liitämällä voi varmistaa tiedon saannin esiintyneistä turvallisuusongelmista ilman aktiivista suorantaa. Työasemavalmistajat tarjoavat myös ylläpitosopimuksia, mutta halvempienkin työasemien käyttöjärjestelmien ja laiteohjainten virhekorjauksista ja versionpäivityksistä saa yleensä valmistajan www- tai BBS-palvelimilta.

Turvallisuusongelmien määrä on tietyn perustason jälkeen verrannollinen järjestelmän tarjoamiin palveluihin. Ne tulee suhteuttaa koneen käyttörooliin, järjestelmätoimittajat toimittavat käyttöjärjestelmänsä alun perin mahdollisimman yleiskäyttöiseksi konfiguroituna.

Käyttäjien sisään kirjoittautumisella paitsi varmistetaan, että vain asiaankuuluvat henkilöt pääsevät järjestelmään, myös huolehditaan siitä, että käyttäjillä on vain tehtäviensä suorittamiseen tarvittavat oikeudet. Tällä ehkäistään asiattomasta ja asiantuntemattomasta käytöstä mahdollisesti aiheutuvat ongelmat. Tätä varten on myös selvitettävä eri käyttäjäryhmien roolit ja näihin kuuluvien tehtävien vaatimat käyttöoikeudet.

Arkaluontoiset tiedot ja järjestelmän turvallisuuteen vaikuttavat muut tiedot, kuten loki- ja konfiguraatiotiedostot ja hakemistot suojataan käyttäjäryhmien roolien mukaisesti. Lisäksi on syytä asettaa käyttöoikeudet vastaavasti järjestelmän hallinnointi- ja konfigurointiohjelmille ja -palveluille.

Järjestelmän lokitiedostoihin kerätään kaikki tarpeellinen tieto järjestelmän toiminnasta ja käytöstä. Merkittävien tapahtumien suodatukseen, automaattiseen käsittelyyn ja tiedoksi saattamiseen esimerkiksi sähköpostin kautta on olemassa myös valmiiksi konfigu-

roituja ohjelmistoja eri käyttöjärjestelmille. Kirjautuvien tapahtumien seuranta on kuitenkin ohjeistettava.

Säännöllinen varmuuskopiointi kuuluu itsestään selvänä järjestelmän turvallisuuteen. Mutta tiedonpalautuksen onnistumisen säännöllinen varmentaminen unohtuu monessa tapauksessa, ja tämä on myös syytä ohjeistaa.

Virustorjuntaohjelmistojen säännönmukainen käyttö auttaa havaitsemaan virukset, troijalaiset ynnä muut turvallisuusongelmat. Lisäksi voidaan ohjeistaa mm. mistä ja miltä välineiltä ohjelmistoja saa asentaa.

Laajoissa verkoissa tietoturvan ylläpito on käytännössä mahdotonta jo kustannussyistä ilman turvallista etähallintaa. Ohjelmistotoimittajina voidaan hyvin suositella suomalaisia alan yrityksiä.

4.5.3 Järjestelmän eheyden ylläpito

Tämä käytäntö varmennetaan ennen kaikkea huolehtimalla siitä, että vain asiaankuuluvilla henkilöillä on fyysinen pääsy koneiden luokse. Esimerkiksi NT- palvelimen tietoturva saadaan helposti murrettua, jos kone voidaan käynnistää asennuslevykkeellä.

4.6 Työasemien tietoturva

Vaikka edellä käsiteltiinkin palvelimien tietoturvan parantamista, niin samat käytännöt soveltuvat myös työasemille. Käyttäjien tietoturva tietoisuuden parantaminen kuuluu edellisten lisäksi työasemien tieturvan parantamisen suositeltuihin käytäntöihin. CERTin julkaisusarjat tarjoavat hyvin yksityiskohtaiset soveltamisohjeet tietoturvallisuuden parantamiseen Windows NT- ja UNIX-järjestelmissä, sekä työasema- että palvelinkäytössä, ja niistä voidaan koota kuhunkin laite- ja toimintaympäristöön sopiva käyttöönottosuunnitelma, jossa turvallisuuskysymykset on otettu huomioon. Windows 95/98 eivät tue mitään turvaominaisuuksia, ja ne sopivat laitealustoiksi vain asiakassovelluksille, joiden tietoturva hoidetaan palvelinkoneilla ja -sovelluksilla.

Eri käyttäjäprofiileihin liittyvien oikeuksien valinnassa on syytä pitää mielessä, ettei tehtävien suorittamista hankaloiteta tarpeettomasti, koska tämä taas vaikuttaa hyvin negatiivisesti käyttäjien tietoturva-ajatteluun.

5. Analyysit ja testit

Luvussa keskitytään ohjelmiston verifioinnin ja validoinnin tärkeimpiin analysointi- ja testaustekniikoihin. V&V:ssä ohjelmiston toteutusta verrataan vaatimuksiin, joihin on kuvattu sekä viranomaisen, yrityksen että projektin näkökulmat. Vaatimukset on kirjattu dokumenteiksi, joiden muoto ja sisältö on määritelty yrityksen prosessikäsikirjassa. Tehtävänä on osoittaa, että vaatimuksiin on vastattu oikein ja täydellisesti ja lisäksi että vaatimus ja sen toteuttaminen ovat jäljitettävissä ohjelmistossa ja sen dokumenteissa. Osoittamisesta vastaa laiteprojektin projektipäällikkö.

EU:n ja FDA:n vaatimukset kohdistuvat turvallisuuteen ja suorituskykyyn ja hoidon saatavuuteen. Koska sekä suorituskyvyn että hoidon saatavuuden alkuperäinen tavoite on myös turvallisuudessa, tässä luvussa keskitytään selkeyden vuoksi vain turvallisuuteen, vaikka esitettävät menetelmät ja tekniikat soveltuisivatkin yleisemmin kaikille luotettavuusattribuuteille.

Osoittamistapoja ovat riskianalyysin menetelmät ja tekniikat, staattiset analyysit ja dynaamiset testit. Luvussa esitellään riskianalyysin tekniikat vika- ja vaikutusanalyysi sekä vikapuuanalyysi. Yleisimmät staattiset analyysit ja dynaamiset testit esitellään myös lyhyesti. Dynaamista testausta käytetään kohdistetusti moduulitestauksissa ja yksittäisten vaatimusten verifioinnissa. Ohjelman toiminta varmistetaan ajamalla sitä tietyillä syötteillä, jotka määräytyvät vaatimuksista. Dynaamista testausta voi käyttää myös luotettavuuden kvantitatiiviseen arviointiin, kun numeerisena aineistona käytetään tilastollisen testauksen tuottamia vikahavaintoja. Testitapaukset johdetaan silloin ohjelmiston käyttöprofiilista.

5.1 Turvallisuuden verifiointi ja validointi

Eri lähteissä termit verifiointi ja validointi määritellään hieman eri tavoin. Erot johtuvat pikemminkin määrittelyn vaikeudesta kuin sisällöllisistä eroista. Laprie (1998) esittää termit seuraavasti:

- Validoinnilla viitataan kaikkiin menetelmiin ja tekniikoihin, joilla saavutetaan luottamus järjestelmän kykyyn suoriutua sille asetetuista sovitussa spesifikaatiossa esitetyistä tehtävistä.
- Verifiointi on prosessi, jolla selvitetään, täyttyvätkö tietyt joko yleiset tai järjestelmälle spesifistiset ominaisuudet tai ehdot, jotka on suoraan johdettu järjestelmän spesifikaatiosta.

Validointi nähdään koko järjestelmän kelpoisuuden arviointina lopullisessa käyttöympäristössään ja -tarkoituksessaan. Verifioinnilla varmistetaan yhden työvaiheen tuloksen vastaavan sille etukäteen asetettuja vaatimuksia. Myös vaatimusten jäljittäminen ja vertaaminen edellisen työvaiheen tuloksiin on verifiointia.

Sekä EU että FDA määrittävät verifioinnin ja validoinnin turvallisuuden todentamisena ja kelpoistuksena, siksi myös tässä luvussa kavennetaan verifioinnin ja validoinnin merkitys vain turvallisuuteen. Suorituskyky ja hoidon saatavuus ovat myös turvallisuuteen liittyviä laatuattribuutteja, jotka turvallisuuden verifioinnissa ja validoinnissa otetaan huomioon.

Luotettavuusvaatimusten validoinnin mukaan suoritettujen järjestelmätasojen vaara-analyysien täydellisyyden ja ristiriidattomuuden tarkistamista sekä toiminnallisten vaatimusten luotettavuusosuuksien kattavuuden tarkistamista koko kehitysprosessin aikana. Jälkimmäinen edellyttää sellaisen luotettavuuslinjan määrittelyä, johon kattavuutta verrataan.

Luotettavuusvaatimusten validointi koostuu seuraavista analyysimenetelmistä, joiden suorittamiseen yleensä tarvitaan erillistä analyysitekniikkaa:

- alustava vaara-analyysi
- vaara-analyysi
- yhteisvika-analyysi
- kvantitatiivinen luotettavuusarviointi.

Vaarojen arviointi tulisi aina aloittaa alustavalla vaara-analyysilla. Se on järjestelmätason kvalitatiivinen menetelmä, jolla tunnistetaan kriittiset ongelmat ja määritellään järjestelmän luotettavuusvaatimukset. Tarkastelu voi kohdistua järjestelmiin, prosesseihin, toimintoihin tai käyttötapauksiin. Analyysi on nopea ja kustannustehokas perustuen erilaisiin läpikäyntityylisiin tarkasteluihin ja tarkistuslistoihin. Tuloksena luokitellaan alustavasti järjestelmän ohjelmistosta tai laitteistosta aiheutuvat riskit ja päätetään lisätarkastelujen kohdistamisesta mm. parantamalla luotettavuussuunnittelua. Menetelmä on myös mahdollista tehdä poikkeamatarkastelun (HAZOP) tai vika- ja vaikutusanalyysin (FMEA) tyylisenä tarkasteluna. Kuvassa 9 on yksi esimerkki alustavasta vaara-analyysikaaviosta.

Tarkastelu perustuu alustaviin tietoihin ja siten analyysi vaatiikin aina seuranta-analyysin tietojen tarkentuessa.

Preliminary Hazard Analysis

Example PHA Worksheet

Area: Meeting Date:

Document number: Team Members:

Hazard: Potential Accident	Cause	Major Effects	Accident Severity Category	Corrective/Preventive Measures Suggested

Kuva 9. Esimerkkikaavio alustavasta vaara-analyysista.

Ohjelmiston vaara-analyysit ovat joukko peräkkäisiä ja iteroivia validointitoimenpiteitä *järjestelmän allokointitasoilla (toiminto, osatoiminto, osajärjestelmä, komponentti jne.).* Jokaisella tasolla tavoitteena on tunnistaa potentiaalisia kyseisen tason luotettavuusvaatimuksiin vaikuttavia virhetoimintoja ja vaaroja sekä niihin johtavia syitä. Kaikissa vaara-analyyseissä tarkastellaan virhetoimintojen merkittävyyttä luotettavuustavoitteiden kannalta sekä määrätään virhetoiminnon tai vaaran kriittisyys. Kaikki vaara-analyysit suositellaan tehtävän perinteisillä riskianalyysin tekniikoilla.

Vaatimusmäärittelyssä voi olla kahdenlaisia virheitä: 1) vaatimukset ovat virheellisiä, 2) vaatimukset ovat oikeita, mutta virheellisenä toteutettuja. SFS-EN 60601-1-4 lähteekin nimenomaan alkuperäisistä tarpeista, joihin validointeja tulisi verrata nimenomaan mahdollisten vaatimusvirheiden läsnä ollessa. Jos lopputuotteessa on puutteita tai yhtäpitämättömyyksiä vaatimusten kanssa, verifioinneilla kyetään tällaiset virheet tunnistamaan. Verifiointia tukevat täsmällisesti ja ylläpidettävästi käytetyt jäljitettävyysmenetelmät. Vaatimusmäärittelyn verifioinneilla tulisi myös tunnistaa määrittelystä puuttuvat toimintotarpeet sekä teknisesti kehitystyötä että ylläpitoa ajatellen.

Ohjelmistovaatimusten vaara-analyysi (Software Requirements Hazard Analysis, SRHA) perustuu alustavan vaara-analyysin ja mahdollisen esisuunnittelun tuloksiin, ja menetelmä on niiden jatkoa tunnistettaessa kriittisiä ohjelmiston spesifikaatiovirheitä. Kriittisyyden määrää riski, joka liittyy spesifikaatiovirheisiin. SRHA:ssa tarkastellaan järjestelmätason vaatimuksia, rajapintadokumentteja ja ohjelmiston vaatimusmäärittelyä.

SRHA soveltuu hyvin suunnittelun varhaiseen ohjaukseen, jossa

- tunnistetaan kriittiset ohjelmistovaatimukset
- varmistetaan, että kriittiset vaatimukset ovat oikeita ja täydellisiä
- suositellaan kriittisten kohtien parantamista tai testitapauksia.

SRHA:ssa analysoidaan tai katselmoidaan järjestelmä-, osasysteemi- ja liitäntävaatimus-spesifikaatioita ja muita systeemidokumentteja tarkoituksena selvittää, että 1) kriittiset vaatimukset on oikein allokoitu ohjelmistolle, 2) kriittiset kohteet alustavan vaara-analyysin tuloksista on tunnistettu ja 3) kriittisten vaatimusten jäljitettävyys systeemispesifikaatiosta detaljitason ohjelmistovaatimusten spesifikaatioon on olemassa. Lisäksi tarkastelukohteina ovat toimintakaaviot, tietovuokaaviot, ajastuskaaviot ja muu ohjelmistodokumentaatio.

SRHA:n verifiointi- ja validointitekniikka vaatii systemaattisen ja oikeamuotoisen vaatimusspesifikaation. Sama vaade tulee myös suunnitteluvaiheelta, sillä yhtäpitävyys näiden vaiheiden välillä on edellytys virheettömälle arkkitehtuuri- ja detaljitason suunnittelulle. Jos vaatimusmäärittelyssä on virheitä, myöhemmissä kehitysvaiheissa voi olla virheitä, joita ei verifioimisella tai validoimisella kyetä tunnistamaan. Verifioitu vaatimusmäärittely onkin lähtökohta muulle verifioinnille ja validoinneille.

Järjestelmän arkkitehtuuri voi koostua redundanttisista osista, yhteisistä tai samantyyppisistä komponenteista tai samasta ohjelmistosta. Eri näkemyksiä arkkitehtuurille on lukuisia (mm. looginen näkymä, suoritusnäkymä, kehittämisnäkymä, ohjelmisto/kovonäkymä jne.). Kriittisen yhteisvirheen eli yksittäisvian vaikutuksesta aiheutuvan usean komponentin kriittisen vikaantumisen mahdollisuus kasvaa näissä tapauksissa. Vikoja, jotka vahingoittavat varmistuksia tai riippumattomia olettamuksia järjestelmän toiminnasta, kutsutaan yhteisvioiksi.

Yhteisvika-analyysille ei ole olemassa erillistä suoritustekniikkaa, vaan mm. kaikki perustekniikat, kuten FMEA ja FTA, soveltuvat tietynlaiseen yhteisvikojen tarkasteluun. Yhteisvika-analyysi voidaan rinnastaa myös muihin vaara-analyysimenetelmiin siksi, että yhteisvirheet ja -viat ovat erityisen tärkeitä juuri ohjelmistoille, joiden kahdentaminen ei vähennä mahdollisten virhetoimintojen esiintymistä.

Menetelmällä tarkastellaan kohteen yhteisvikoja eri tavoin riippuen yhteisvian vaikutusalueesta. Sisäiset tapahtumavirrat katkaistaan erottamalla yhteisvirheisiin alttiit systeemiosuudet toistaan. Systeemin osat sekä kehitetään että ylläpidetään toisistaan riippumattomasti. Erityisen tärkeää on tunnistaa systeemin osien väliset liitynnät ja vuorovaikutukset.

Verifioinnin ja validoinnin kohteena ovat vaiheiden tuotedokumentit. Siten onnistuakseen ne tarvitsevat mahdollisimman täydellisen ja virheettömän suunnitteludokumentaation ja lähdekielisen koodin. V&V tulee aina suunnitella etukäteen ja suorituksessa noudattaa suunnitelmia ja tarkistaa niissä esitettyjen kriteerien oikeellisuus. Suunnitelmat ovat olennainen osa kehitystyövaiheita ja ne tulisikin tehdä rinnan niiden kanssa.

Oikeellisuuden verifioimisessa tarkastetaan eri prosessivaiheissa vaiheille asetettujen spesifikaatioiden virheetön toteuttaminen tuotosdokumentaatiossa. Oikeellisuuden verifioiminen täydentää luotettavuusvaatimusten validointia: edellinen tarkistaa vaiheelle asetettujen spesifikaatioiden toteutumisen, jälkimmäinen alkuperäisten tavoitteiden toteutumisen jokaisessa vaiheessa.

Oikeellisuuden verifioiminen koostuu seuraavista toimenpiteistä :

1. Staattiset analyysit, kuten tarkistukset, läpikäynnit ja katselmukset, joita voidaan soveltaa kaikissa kehitysprosessin vaiheissa jokaiselle vaihedokumentille (spesifikaatiot ja suunnitteluaineisto, verifiointisuunnitelmat jne.).
2. Testit, jotka sisältävät useita tekniikoita, kuten toiminnalliset, rakenteelliset, raja-arvotestit jne. Testitekniikat kattavat erityisesti ne järjestelmäkäyttäytymisen dynaamiset ominaisuudet, joita on vaikea tunnistaa staattisin analyysein tai formaalein menetelmin.
3. Formaalit menetelmät ja oikeaksitodistamiset, joista edelliset hyödyntävät selkeiden, täsmällisten ja yksiselitteisten vaatimusten, olettamusten, spesifikaatioiden ja sunnitteluratkaisuiden kuvaamista. Yhdessä jälkimmäisten toimien kanssa ne pyrkivät osoittamaan toteutuksen vastaavuuden sille asetettuihin ominaisuuksiin.
4. Käyttäytymisanalyysit perustuvat joko spesifikaatiosta, suunnittelusta tai toteutuksesta matemaattisesti johdettuun kohteen käyttäytymismalliin. Mallien avulla todennetaan yleisiä ominaisuuksia, kuten täydellisyyttä ja yhtenäisyyttä, sekä tiettyjä vaatimuksista johdettuja ominaisuuksia, kuten tietyn tapahtumasekvenssin olemassaoloa. Käyttäytymismallit ovat päätöstaulukoita, Petriverkkoja, tilakoneita jne.
5. Jäljitettävyysanalyysit koostuvat ristiviittauksista ja matriiseista. Ne palvelevat kolmea päämäärää:
   - jäljittämistä vaiheelle asetuista vaatimuksista toteutukselle
   - jäljittämistä toteutuksesta vaatimuksille
   - jäljittämistä vaatimuksista testitapauksiin ja todennustoimiin.

Staattisin analyysein voidaan valtaosa virheistä selvittää heti niiden syntymishetkellä. Toimet ovat helppokäyttöisiä eivätkä kovin kalliita, niiden tehokkuus on hyvä, mutta ne edellyttävät kuitenkin tarkkaa paneutumista toimiin.

Riippumatta testivaiheesta (moduuli-, integrointi- tai järjestelmätestaus) eri testitekniikoita hyödynnetään rinnan. Tilastolliset todennäköisyystestit täydentävät deterministisiä testejä silloin, kun testattavien tapausten määrä on erittäin suuri. Tilastoilla kohdennetaan determinististä testaamista.

Luotettavuuden validointimenetelmät koostuvat seuraavista toimenpiteistä, joista kaksi ensimmäistä kuuluvat attribuuttiin tietoturva ja kolmas on yleinen soveltuen attribuutteihin:

1. Tunkeutumisanalyysi, jolla järjestelmällisesti tunnistetaan virheitä, jotka voivat johtaa ei-toivottuihin tilanteisiin.
2. Salapolkuanalyysi, jolla etsitään tietoturvan luottamuksellisuusominaisuuden heikkouksia tai dokumentoimattomia tahattomasti tai tahallisesti piilotettuja sisäisiä tai ulkoisia takaportteja ohjelmistojärjestelmän tietoihin.
3. Eksperimentaalinen arviointi, jossa tarkastellaan järjestelmän kriittisimpien osien luotettavaa käyttäytymistä. Arviointi perustuu todellisessa käyttöympäristössä kerättyyn luotettavuustietoon. Tekniikoita ovat virheen syöttäminen ja tilastollinen merkittävyyden arviointi.

Prosessilaadun arvioinnin tekniikoita ovat tarkistukset, katselmukset, auditioinnit jne. Niillä pyritään selvittämään, että projekti on asianmukaisesti suunniteltu ja ohjattu. Arvioimiseksi on kehitetty laatujärjestelmiä, mm. Capability Maturity Model ja ISO 9000, jotka osaltaan keskittyvät virheitä ennaltaehkäisevään hallintaan.

Tässä luvussa käsitellään ensisijaisesti ohjelmiston luotettavuusvaatimusten validointimenetelmiä ja tekniikoita, sekä oikeellisuuden verifiointitekniikoista staattisia analysointeja ja dynaamisia testejä.

5.2 Luotettavuusvaatimusten validointitekniikat

Luotettavuusvaatimusten[1] analyysitekniikkoja hyödynnetään laitteen ja ohjelmiston sekä valmistusprosessin aikaisissa riskinvähennykseen tähtäävinä toimenpiteinä että arvioitaessa valmiin tuotteen riskittömyyttä. Kaikissa niissä tapauksissa, joissa tuotteelle asetetaan luotettavuustavoitteet jossakin projektin varhaisessa vaiheessa, luotettavuuden

[1] Tässä luotettavuudella tarkoitetaan attribuutteja toimintavarmuus, käyttövarmuus, ylläpitovarmuus ja turvallisuus.

analysointi kohdistuu näistä tavoitteista muodostettuihin luotettavuusvaatimuksiin. Ne voivat olla joko erillisiä luotettavuutta parantavia tai riskiä vähentäviä toimintoja tai ei-toiminnallisia vaatimuksia liittyen kyseisen laitteen tai ohjelmiston toimintoon.

Luotettavuusvaatimusten määräytyminen laitteelle ja sen ohjelmistolle on iteratiivinen prosessi. Ne voivat syntyä missä tahansa elinkaaren aikana, mutta tavoitteena on varhaiset elinkaarivaiheet, laitteen toiminnallinen vaatimusmäärittely tai arkkitehtuurivaihe, joiden dokumenttien pohjalta kyetään tekemään alustavaa riskianalyysia. Vaatimukset kirjataan aina määrittelyvaiheen dokumentteihin ja huolehditaan analyysein ja testein niiden sekä määrittelyn että toteuttamisen oikeellisuudesta, täsmällisyydestä ja ristiriidattomuudesta.

Tässä luvussa tarkastellaan kolmea keskeisintä ensi sijassa laitoksille, järjestelmille ja laitteille alun perin tarkoitettua turvallisuusanalyysin tekniikkaa, joita on sovellettu muihinkin luotettavuusattribuutteihin kuin turvallisuuteen. Menetelmien, jotka ovat vioittumistapa-, vaikutus- ja kriittisyysanalyysi, vikapuuanalyysi ja poikkeamatarkastelu, on todettu soveltuvan myös ohjelmistoille (Leveson 1995, Ippolito & Wallace 1995).

5.2.1 Vika- ja vaikutusanalyysi

Vika- ja vaikutusanalyysi (FMEA) sekä sen johdannainen vika-, vaikutus- ja kriittisyysanalyysi (FMECA) ovat vaara-analyysitekniikoita, jotka erityisen hyvin soveltuvat mekaanisten ja elektronisten järjestelmien arviointiin, mutta joita on myös sovellettu ohjelmistojen sekä prosessien että tuotteen arvioimiseen.

FMEA on kvalitatiivinen analyysitekniikka, jolla selvitetään kohteen eri osien vioittumistavat sekä määritellään niiden vaikutukset ja kriittisyydet kohteen muihin osiin sekä kohteelta vaadittuun toimintaan. Tekniikat osoittavat, miten kohde käyttäytyy erilaisissa vikatilanteissa, miten se suoriutuu tehtävästään sekä minkälaisia parannuksia tulisi tehdä yksittäisvikojen paljastamiseksi ja vähentämiseksi. Tekniikka myös osoittaa ovatko ongelmien ratkaisemiseksi määritellyt suojaukset ja varmistukset asianmukaiset.

FMECA on kvantitatiivinen analyysitekniikka siinä mielessä, että sillä priorisoidaan vikatapahtumat riskien poistamis- ja vähentämisprosessien käynnistämiseksi. FMECA:lla kyetään jäljittämään riskienhallinnan toimenpiteitä.

FMEA ja FMECA kuuluvat perustekniikoihin, sen alhaalta-ylös-lähestymistapa voidaan liittää tehokkaasti moneen muuhun menetelmään, erityisesti ohjelmiston tai järjestelmän vikapuuanalyysiin. Tunnistusmenetelmistä usein poikkeamatarkastelu, HAZOP, on vaihtoehtoinen menetelmä. HAZOP-tekniikkaa käytetään yleisesti teollisuusprosessien

häiriöiden tarkasteluun, kun taas FMEA:t ovat yleisiä komponenttitasolla. Kuva esittää esimerkkiformaatit FMEA:sta ja FMECA:sta. Taulukoiden sarakkeet ovat sovelluskohtaisia, ne tulisi valita asianomaisen vaara-analyysin tavoitteiden mukaisiksi. Teknistä suoritustapaa on selostettu standardissa IEC 60812 (1985)[2] sekä mm. viitteessä (Palady 1998).

Tuotteen tai suunnittelun kompleksisuus ja suunnittelutiedon saatavuus määrittävät ensi sijassa FMECA:n suoritustarkkuuden. Suoritustapoja on yleensä kaksi: fyysinen, ja toiminnallinen. Fyysistä lähestymistapaa käytetään silloin, kun kohteesta on saatavilla riittävästi yksityiskohtaista tietoa, toiminnallista, kun systeemi on niin laaja tai kompleksinen, että yksityiskohtainen fyysinen käsittely vie runsaasti aikaa. Myös ohjelmistoille kummatkin lähestymistavat ovat olleet käytössä, mutta toiminnallinen tapa on suosituin.

Ohjelmistoille tyypillisiä toiminnallisia vioittumistapoja:

- toiminta ennenaikaista
- toiminnan epäonnistuminen määrätyssä ajassa
- toiminta epäsäännöllistä
- toiminnan päättymisen epäonnistuminen
- toiminnan menettäminen tai epäonnistuminen (mm. käytön aikana)
- toiminnan osittainen heikentyminen.

[2] Standardia IEC 60812 ollaan uusimassa.

Hardware Failure Mode and Effect Analysis

Example FMEA worksheet

System: _____ Detailed part: _____

Description:

Failure mode	Causes	Effect levels			Detection	Protection	Recommendations Remarks
		Local	Next higher	End			

Process Failure Mode, Effects and Criticality Analysis

Example FMECA worksheet

Company: System: Teem members:

Project: Subsystem:

Meeting date: Document:

Ref.	Process	Failure mode	Effect on	Potential effect	Potential cause	Existing controls	Existing conditions				Recomm. action	Conditions after action			
							Sev	Occ	Det	APN		Sev	Occ	Det	APN

Detection

Severity

Action Priority Number Occurance

Kuva 10. Esimerkkejä FMECA:n tarkasteluraportin formaatista. Vioittumistapa on kaikkien FMEA-tyylisten tarkasteluiden keskeisin kohta. Vioittumistapaan johtavat syyt, sen vaikutukset, havaintokeinot ja estotavat tunnistetaan. Määrätään vian kriittisyysaste sekä suositellaan korjaavia toimia tai jatkoanalyyseja.

Vioittumistavan vaikutusten tarkastelu voidaan ulottaa toimintoon tai laitteeseen itseensä, käyttöliittymään, muihin kun tarkasteltavan kohteen toimintoihin, ylemmälle tasolle (järjestelmä) tai niiden tehtäviin ja ympäristöön. Vaikutusten kriittisyys voidaan mitata kvalitatiivisesti tai kvantitatiivisesti. Määrittämisessä tulisi ottaa huomioon vaikutuksen suuruuden lisäksi aina esiintymistiheys ja kohdistaa se erikseen toimintoon, käyttöliittymään jne. Lisäksi tulisi ottaa huomioon vioittumistavan tai sen seuraustapahtuman havaintokyvykkyys ja suojaavat toimenpiteet.

Havainnointitavan tulisi mielellään olla sellainen, mistä vioittumistapahtuman esiintyminen voidaan käytännössä havaita. Niitä vioittumistapoja tai virhetapahtumia, joille ei analysoinnissa löydy havaitsemiskeinoja, tulisi tuloksissa selkeästi korostaa.

Tulosten arviointi voidaan suorittaa usealla tasolla kriittisyydestä, laajuudesta, kompleksisuudesta ja projektoinnista riippuen. Kriittisimpiin ja yleensä kaikkiin korostettuihin tuloksiin arviointi tulisi aina tehdä ja viedä tulokset parantaviksi toimenpiteiksi esimerkiksi joko uudelleen suunnittelulla tai käyttötoimenpitein. Suositukset näistä annetaan joko analysoinnin tuloksina tai erillisen arviointityöryhmän tuloksina.

5.2.2 Vikapuuanalyysi

Bell Telephone Laboratories kehitti vuonna 1961 vikapuuanalyysin (Fault Tree Analysis) Yhdysvaltojen ilmavoimille toimintavarmuuden ja turvallisuuden tarkasteluun. Vikapuuanalyysilla etsitään tiettyihin vaaratapahtumiin, joita kutsutaan huipputapahtumiksi, johtavia syitä. Analyysilla ei tunnisteta vaaroja, vaan huipputapahtumien täytyy olla tiedossa ennen tarkasteluun ryhtymistä. Vikapuuanalyysin tavoitteena voi olla tarkasteltavan kohteen

- luotettavuusmallin laadinta
- heikkojen kohtien tai merkittävien vikayhdistelmien tunnistaminen
- valvontojen ja varmennusten riittävä arviointi
- luotettavuuden kvantitatiivinen määrittäminen.

Vikapuu on looginen kaavio, joka esittää kohteen kriittisen vikaantumisen eli huipputapahtuman riippuvuuden kohteen osien vioista tai ulkoisista tapahtumista. Analyysissa lähdetään liikkeelle tarkasteltavan kohteen vioittumistapahtumasta selvittämällä mistä tapahtumasta tai tapahtumakombinaatiosta se voi aiheutua. Edetään kohti yksinkertaisia, toisistaan riippumattomia perustapahtumia, joiden esiintymisestä on saatavissa kokemusperäistä tietoa. Löydetyt syyt: viat ja virheet, joko korjataan tai esitetään sellaisia ehkäiseviä, suojaavia tai muita toimenpiteitä, joilla tapahtumaketjut voidaan pysäyttää. Viat ja virheet voivat aiheutua ohjelmoinnista, suunnittelusta, inhimillisistä tekijöistä tai laitteistosta.

Vikapuuanalyysi on alun perin tarkoitettu järjestelmä- ja laitteistoanalyyseihin kvantitatiivisen todennäköisyysarvioinnin tekemiseksi, mutta sitä on myös menestyksellä sovellettu ohjelmistoihin (Software Fault Tree Analysis, SFTA, Leveson 1995, Ippolito & Wallace 1995). SFTA soveltuu kaikkiin ohjelmistoprosessin vaiheisiin vaatimusmäärittelystä koodausvaiheeseen.

Vikapuun teknistä suorittamista ohjeistaa standardi IEC 61025 (1990) yleisesti. Vikapuuanalyysi on perustekniikka, jonka ylhäältä-alas-lähestymistapa voidaan liittää tehokkaasti moneen muuhun menetelmään. SFTA voidaan liittää myös järjestelmän tai lait-

teiston (elektroniikan) vikapuuhun. Tällöin laitteiston ja ohjelmiston vikapuut yhdistetään koko järjestelmän analysoimiseksi. Tämä on merkittävää siksi, että monet vaarat voivat aiheutua ohjelmistovirheen, laitteistovian tai inhimillisen virheen yhdistelmästä.

Koska vikapuu kuvaa tapahtumat loogisesti, kaikki logiikkaa noudattavat tapahtumasarjat voidaan kuvata. Myös FMECA voidaan muuntaa vikapuiksi ja arvioida kvantitatiivisesti tietyn tapahtuman luotettavuus. FMECA:lla usein etsitään kriittiset tapahtumat, joita sitten tarkemmin tutkitaan vikapuulla. Myös päinvastaista menettelyä käytetään. Tällöin FTA:lla on ensin tunnistettu lyhyet minimikatkosjoukot, joita sitten on tarkemmin tutkittu FMEA:lla.

5.2.3 Poikkeamatarkastelu

Poikkeamatarkastelu (Hazard and Operability Study, HAZOP) on alun perin tarkoitettu kemianlaitoksille riskien tunnistusmenetelmäksi, jota erityisesti käytetään toimintaparametrien muutoksista tai puutteellisesta suunnittelusta johtuvien häiriöiden ja niiden aiheuttamien onnettomuusriskien kartoittamiseen (Kletz 1986). Tekniikka kelpaa myös pienempien vikaantumisten syiden hakuun, ja siten sillä voidaan ainakin periaatteessa kartoittaa ohjelmistosta ohjattavalle kohteelle aiheutuvia ongelmia.

HAZOP suoritetaan käyttämällä avainsanoja (ei, enemmän, takaisinpäin jne.). Näistä avainsanoista päätellään, minkälaisia kriittisiä seurauksia voi sattua tai minkälaisia käytön aikaisia ongelmia voi esiintyä. Myös syitä tarkastellaan, mutta vain lähinnä kelpoistettaessa määrätyn avainsanan merkittävyyttä. Suositellaan systeemiä tai prosessia parantavia ratkaisuja. Menetelmä on hyvin suosittu prosessiteollisuudessa tehokkaana työkaluna prosessin riskien kartoituksessa ja parantamisessa.

HAZOP on tavallaan muunnelma FMECA:sta ja vastaavasti sovellettavissa ohjelmistotuotannon kaikille vaiheille. Poikkeamatarkastelu on yleensä aina ryhmätyötä, mitä FMECA ei välttämättä ole. Ryhmää vetää HAZOP-tekniikkaan perehtynyt asiantuntija. Istuntoihin osallistuu tarkasteltavan kohteen tuntevia asiantuntijoita.

5.3 Ohjelmiston oikeellisuuden verifiointitekniikat

Yleisesti ottaen ohjelmistoja testataan, koska halutaan

- löytää vikoja, esimerkiksi lähdekoodin debuggaus
- osoittaa yksittäisen lopputuoteyksilön vaatimuksenmukaisuus
- osoittaa työvaiheen tuloksen tai koko järjestelmän määrittelyn mukainen toiminta, virheettömyys.

Ensimmäinen motiivi on vallitseva keskeneräisten artefaktien yhteydessä; ohjelman kirjoittaja pyrkii löytämään koodistaan virheet, jotta ne voidaan korjata ennen moduulin varsinaista hyväksyntätestausta. Seuraava motiivi ei ole puhtaan ohjelmistotuotteen yhteydessä yleensä kovin oleellinen, koska ohjelmisto voidaan jäljentää luotettavasti, jäljennöksen virheettömyys tarkistaa nopeasti ja helposti tarvittaessa bitin tarkkuudella. Tässä luvussa keskitytäänkin viimeiseen näkökulmaan, jossa testausta tehdään osana verifiointia ja validointia. Samat tekniikat ovat silti usein käyttökelpoisia niin vianetsintä- kuin verifiointitestaukseenkin.

Testit voidaan jakaa dynaamisiin ja staattisiin. Dynaaminen testaus edellyttää tutkittavan kohteen suorittamista eli esimerkiksi ohjelmalohkon ajamista. Ohjelmistoalalla testauksella tarkoitetaan tavallisesti nimenomaan dynaamista testausta. Järjestelmässä, jossa samanaikaisesti kehitetään ohjelmistoa ja rautaa (hardware), voidaan ohjelman dynaaminen testaus aloittaa simulaattorilla, mikäli kehitettävä rauta ei ole valmista. Staattinen testaus tutkii kohteen ominaisuuksia suorittamatta/käyttämättä sitä. Usein staattista testausta kutsutaan analyysiksi.

Testausmenetelmät voidaan luokitella myös testaajan testauksen kohteesta tietämän informaation määrän perusteella. Black box -testaamisesta puhutaan silloin, kun testaaja ei tiedä kohteen sisäisestä rakenteesta mitään ja testaus tapahtuu pelkästään vertaamalla kohteen “ulkoista” toimintaa vaatimuksiin (esim. funktion toiminnan vertaaminen vaatimus-määrittelyyn). White box -testaaminen puolestaan edellyttää testaajalta testattavan kohteen sisäisen rakenteen tuntemusta. Tällöin testejä suunniteltaessa ja toteutettaessa kiinnitetään merkittävästi huomiota kohteen rakenteeseen.

Mitä laajemmasta, monimutkaisemmasta ja turvallisuuskriittisemmästä ohjelmistosta on kysymys, sitä oleellisemmaksi nousee kysymys siitä, mikä on oikea taho vastaamaan testauksesta. Tässä yhteydessä ei ole mahdollisuutta puuttua tähän kysymykseen tarkemmin, mutta yleisesti todetaan, että testaajan riippumattomuuden testattavan kohteen luomisesta tulee kasvaa vaatimusten koventuessa.

Tässä yhteydessä ei myöskään ole mahdollisuutta perehtyä testausta helpottaviin ja automatisoiviin CASE-työkaluihin. Todetaan, että näitä on markkinoilla useita ja että eriasteisesti automatisoitu testaaminen on monissa yrityksissä ollut rutiininomaista jo lukuisia vuosia.

5.3.1 FDA:n käsityksiä testauksesta

Esitettävät käsitykset perustuvat FDA:n luonnokseen (FDA 1997), joka kuvaa FDA:n yleisiä vaatimuksia ohjelmiston validoinnille. Eri tyyppisille sovellusalueille on olemassa tarkempia julkaisuja, mutta ainakin tämä julkaisu keskittyy enimmäkseen tarvittavan

dokumentaation ja prosessinaikaisen riskienhallinnan määrittelemiseen, eikä niinkään testaamisen kohteisiin tai yksityiskohtaisiin testausmenetelmiin.

FDA toteaa, että jokaisen ohjelmistoprojektin yhteydessä vastuullisen tahon tulee määrittää ja perustella

1) tarvittavan validointityön määrä
2) käytettävien validointitekniikoiden spesifinen yhdistelmä.

Lisäksi FDA korostaa, että tavallisesti pelkkä testaus ei voi täysin verifioida ohjelmiston täydellisyyttä ja virheettömyyttä. Testauksen lisäksi tarvitaan muita verifiointitekniikoita sekä strukturoitu ja dokumentoitu ohjelmistotuotantoprosessi kattavan validoinnin varmistamiseksi. Huomattakoon, että tässä FDA tarkoittanee testeillä nimenomaan dynaamisia testejä. Muut verifiointimenetelmät ovat tällöin erilaisia staattisia analyysejä.

5.3.2 Tekniikoiden soveltuvuus ja tehokkuus

Kirjallisuudessa on esitetty lukuisia testausmenetelmiä, joista tässä yhteydessä on mahdollista esitellä vain osa. Monet eri nimellä kulkevat testit ovat saman teeman muunnelmia, joten esityksen yleisyyttä ei juurikaan menetetä näiden poisjättämisellä.

Menetelmien luokittelu voidaan tehdä eri näkökulmista, esim. white box vs. black box, dynaamiset vs. staattiset jne. Nämä eivät ohjelmistoa tuottavan organisaation näkökulmasta kuitenkaan ole kaikkein hyödyllisimpiä luokituksia. Nyt on lähdetty siitä, että vaikka yritysten ohjelmistoprosessit vaihtelevat paljon, tiettyjä samoja työvaiheita (aktiviteetteja) esiintyy kaikissa. Testaustekniikat on ryhmitelty työvaiheen mukaan (taulukko 11). Kyseiset kehitysvaiheet ovat Storeyn (1996), mutta eivät eroa merkittävästi yleisen vesiputousmallin vaiheista Haikala 1998). Taulukon 11 sarakkeessa 7 kuvataan standardin turvallisuuden eheystasosuositus minimitasolle (suositus pätee tarkastellulle tasolle ja siitä kriittisimmille tasoille). Samaa tekniikkaa (esim. monia staattisia analyyseja) voidaan käyttää useissa eri työvaiheissa, kun taas jotkut tekniikat ovat ymmärrettävästi spesifisempiä. Jaottelua voidaan perustella käyttökelpoisuudella, sillä taulukosta voidaan nähdä suoraan yhden työvaiheen tuloksen verifioimiseen tarjolla olevat keinot. Alempana on annettu kullekin testaustekniikalle lyhyt kuvaus. Tarkempia kuvauksia löytyy alan kirjallisuudesta, ks. esimerkiksi (IEC 61508 2000, Beizer 1989).

Taulukko 11. Staattisten analyysien ja testaustekniikoiden käyttö elinkaaren vaiheissa ja suositus turvallisuuden eheystasolle viitteen (IEC 61508 2000) mukaan. 1 Vaatimusmäärittely; 2 Arkkitehtuurisuunnittelu; 3 Suunnittelu; 4 Toteutus; 5 Järjestelmäintegrointi; 6 Verifiointi & Validointi; 7 Minimi turvallisuuden eheystaso TET 1... TET 4.

Tekniikka	1	2	3	4	5	6	7
Läpikäynti (Walktrough)	X	X	X	X	X		TET 1
Suunnittelukatselmus	X	X	X	X	X		TET 1
Tarkistuslista	X	X	X	X			
Formaalitodistus		X					TET 4
Ohjausvuoanalyysi			X				TET 2
Tietovuoanalyysi			X				TET 2
Symbolisuoritus	X		X				TET 3
Oikopolkuanalyysi					X		
Raja-arvoanalyysi				X	X		TET 3
Virheen arvaus				X	X		
Prosessin simulointi				X		X	
Ajoitus- ja muistitestit					X		
Suorituskyvyn testaus					X		TET 3
Rasitustestaus					X		

Wallace & Kuhn (2000) analysoivat lääkintälaitteiden ohjelmistovikoja (ks. kohta 3.4.6), jotka on saatu tarkastelemalla FDA:n tietokantoja. Viat, joita oli yhteensä 342, luokiteltiin ohjelmiston virhetyypin mukaan. Lisäksi he tarkastelivat ja suosittelivat menetelmiä, joilla estetään virheiden syntyminen ja havaitaan virheen olemassa olo. Menetelmäsuositukset kohdistuvat ensisijassa staattisiin analyyseihin ja testeihin sekä kehitysprosessin tekniikoihin.

Virhetoimintojen kohteet on luokiteltu laskentaan (mm. algoritmit), muutosten vaikutukseen, tuotteen hallintaan, dataan, virhesietoisuuteen, alustukseen, liittymiin ja logiikkaan sekä erittelemättömiin, kuten suorituskykyyn, tuloihin ja lähtöihin sekä kirjoitusvirheisiin. Lisäksi virhetoiminnat luokitettiin kohteisiin laadunvarmistus, vaatimusmäärittely ja ajastus. Yleisimmät vikaluokat ovat tutkimuksen mukaan logiikka- ja laskentavirheet, joita oli 67 % kokonaismäärästä.

Laskentavirheisiin kuuluu useita eri tyyppisiä virhelähteitä: raja-arvot, alueet, siirtymät matemaattisesta ilmaisusta toteutukseen jne. Viitteen suosittamat staattiset virheiden estämiseen tarkoitetut menetelmät ovat koodin läpikäynnit sekä määrittely-, suunnittelu-

ja koodikatselmukset. Muista analyyseista suositeltiin jäljitettävyysanalyysia, muutosten vaikutusanalyysia ja kriittisen polun analyysia.

Koodin läpikäynnillä varmistutaan mm. taulukoitujen tietojen ja koodin välisistä suhteista, alustuksen virheettömyydestä ensimmäisellä suorituskerralla, ohjelmiston liittymistä ulkoisiin laitteisiin ja ohjelmistoihin, ohjauslogiikan täydellisyydestä ja virheettömyydestä sekä synkronoinnista kahden prosessin välillä. Koodin läpikäyntiä ja tarkistusta suositetaan myös virheiden etsimiseen seuraavien virhetoimintojen osalta: väärin koodattu datataulukko, määrittelymuutosten verifiointi, alustusvika, liityntäviat, epätäydellinen tai virheellinen liittyminen.

5.3.3 Staattiset analyysit

Läpikäynti (Walkthrough)

Artefaktin laatija johdattaa arviointiryhmän artefaktin sisällön läpi vaihe vaiheelta, jolloin ryhmän jäsenet voivat kysyä ja kommentoida sisältöä tekniikan, tyylin, virheiden yms. osalta. Tämä toteutetaan johdettuina tilaisuuksina, joissa puheenjohtaja huolehtii siitä, että yhteen kohtaan ei juututa liian pitkäksi ajaksi. Tavoitteena on vain paikallistaa virheitä ja epäselvyyksiä, havaittujen vikojen korjauksesta huolehtii artefaktin laatija. Ryhmän jäsenillä on oltava riittävästi erikoisosaamista kyseiseltä alalta. Organisaation arvoasetelmat eivät saisi haitata kommunikointia. Näin ollen esimerkiksi koodiläpikäynnissä arviointiryhmä koostuu muista ohjelmoijista, mielellään sellaisista, jotka eivät ole tuottamassa koodia kyseiseen projektiin. Tilaisuuksien vaatimat resurssit, valmistelu mukaan lukien, on huomioitava projektien suunnitelmassa.

Suunnittelukatselmus (Desk checking, design review)

Verifioija simuloi ohjelman toimintaa ilman ohjelman suoritusta. Menetelmän avulla voidaan löytää virheitä erityisesti ohjelman logiikassa (Scavo 1994). Poikkeaa läpikäynnistä siten, että artefaktin laatija ei johdattele verifioijia järjestelmän läpi. Virheiden löytämiseen saattaa olla paremmat edellytykset, mutta vastaavasti verifiointiin tarvitaan enemmän resursseja.

Faganin tarkistukset on systemaattinen auditiointimenetelmä virheiden ja puutteiden löytämiseksi laadunvarmistusdokumenteista. Fagan esittelee koko ohjelmistotarkistuksen käsitteistön, tarkistuslistoja ja aikansa luotettavuustietoa.

Tarkistuslistat

Joukko kysymyksiä, joiden avulla voidaan kriittisesti tarkastella järjestelmän eri ominaisuuksia. Tarkistuslistojen avulla voidaan varmistaa eri näkökulmien mukaan ottami-

nen arvioinnissa. Kysymykset ovat yleensä yleisluontoisia, joten niitä voidaan käyttää useille erilaisille järjestelmille.

Formaalit todistukset

Järjestelmän suunnittelun tai toteutuksen jonkin osa-alueen virheettömyys todistetaan formaalisti. Tämä tarkoittaa matemaattisen esitysmuodon käyttöä. Tällöin voidaan todistuksissa käyttää esimerkiksi ensimmäisen asteen predikaattilogiikkaa. Erityistä hyötyä tästä on silloin, mikäli järjestelmän vaatimusmäärittelyssä on käytetty formaaleja menetelmiä. Mainittakoon, että formaalit menetelmät vaativat huomattavasti aikaa ja osaamista, joten niitä käytetään tavallisesti vain hyvin kriittisiin kohteisiin.

Ohjausvuo-analyysi

Ohjelman rakennetta tutkimalla pyritään löytämään ohjelmalohkoja, joihin ei koskaan päästä käsiksi, umpikujia, päättymättömiä silmukoita sekä muita rakenteellisia puutteita. Tyypillisesti analysoitava ohjelma esitetään suunnatun graafin avulla.

Tietovuo-analyysi

Diagrammin avulla esitetään ohjelman läpi kulkeva tietovirta. Diagrammissa on selvästi merkittynä kukin tieto sekä tietoja käsittelevät prosessit. Tällaisen yleiskuvan avulla voidaan arvioida prosessien soveltuvuutta sekä arvioida suunnitellun ja alun perin vaaditun tietovirran välistä suhdetta.

Symbolinen suorittaminen

Formaali menetelmä, jolla on paljon yhteistä formaalien todistusten kanssa. Sen sijaan, että ohjelmaa suoritettaisiin oikeilla syötteillä, käytetään symbolisia muuttujia. Ohjelman toiminnot, esimerkiksi muuttujien alustukset ja laskuoperaatiot kohdistetaan symbolisiin muuttujiin, jolloin suorituksen tuloksena syntyviä kaavoja voidaan verrata asiakkaan odotuksiin (speksin verifiointi) tai vaatimusmäärittelyn perusteella ilman ohjelmaa laskettuihin (suunnittelun/koodin verifiointi). Koska jokainen polku ohjelman läpi täytyy laskea kattavassa analyysissä, on menetelmä käytännössä hyvin työläs.

Oikopolkuanalyysi

Oikopolut ovat järjestelmään tahattomasti suunniteltuja tiloja, jotka voivat johtaa järjestelmän yllättävään, ei-haluttuun toimintaan tietyissä olosuhteissa. Laitteistotasolla kyseessä voivat olla todelliset fyysiset sähköä johtavat “polut”, mutta oikopolku voi myös olla seurausta esimerkiksi ohjelmiston ajoitusten epäsäännöllisyydestä. Oikopolut johtuvat yleensä järjestelmän monimutkaisuudesta, jolloin suunnittelija ei pysty hallitsemaan kaikkia järjestelmään kohdistuvia vaatimuksia. Erityisen riskin muodostavat ohjelmistoon myöhemmin tehtävät korjaukset. Oikopolkuanalyysi pyrkii löytämään kyseisiä vikoja raudan ja ohjelmiston topologiaa tutkimalla. Ohjelman ohjausvuosta

etsitään loogisia oikopolkuja vertaamalla rakenteita tunnettuihin vaarallisiin ratkaisuihin. Ainakin rautapuolella on olemassa lukuisia, varsin laajasti käytettyjä työkaluohjelmistoja mahdollisten oikopolkujen automaattiseen tunnistamiseen.

5.3.4 Dynaamiset testit

Dynaaminen testaaminen voidaan jakaa seuraaviin päämenetelmiin: funktionaalinen, rakenteellinen ja tilastollinen. Funktionaalisessa testauksessa järjestelmän kaikki vaatimusmäärittelyssä kuvatut toiminnot testataan. Toimintojen toteutuksesta ei olla kiinnostuneita, ainoastaan vaatimusten mukaisesta toiminnasta. Rakenteellinen testaus hyödyntää järjestelmän toteutuksen tuntemusta. Tutkittavina voivat olla ohjelman rutiinit ja suorituspolut. Tilastollisen testauksen ero edellä mainittuihin on siinä, että järjestelmälle annettavat syötteet eivät ole deterministisesti valittuja, vaan ne arvotaan satunnaisesti valitusta jakaumasta. Satunnaisuuden avulla vältetään urautuneen testauskäytännön tiedostamaton yksipuolisuus. Erikoistapauksena voidaan mainita käyttäjäprofiilien mukainen tilastollinen testaaminen, jossa syötteiden valintaa ohjaa todellisen tai oletetun käyttäjän syöteprofiili. Seuraavassa esitettäviä tekniikoita voidaan käyttää eri päämenetelmien yhteydessä.

Raja-arvoanalyysi

Järjestelmää tutkitaan syötteillä, jotka ovat määritellyn toiminta-alueen raja-arvoja, sallitun syöteavaruuden reunoilla. Myös rajojen ulkopuolisia arvoja kokeillaan, sekä juuri rajojen sisäpuolella olevia arvoja.

Virheen arvaus

Kokeneilla testaajilla on usein tuntumaa siitä, missä vikoja saattaa piillä. Virheiden arvauksessa testaaja soveltaa kokemustaan uuteen tuotteeseen. Mahdolliset testitapaukset analysoidaan ja hyväksytyt lisätään osaksi testisuunnitelmaa.

Prosessin simulointi

Erityisesti järjestelmissä, joissa laitteiston ja sitä ohjaavan ohjelmiston kehitys tapahtuu rinnakkain, voidaan testausta tehostaa huomattavasti simulaattorilla. Kehitettävän laitteen toiminnot hoitaa tietokoneessa ajettava simulaattori, jonka toiminnan runkona on malli todellisen laitteen toiminnasta. Yksityiskohtaiset laitetason ilmiöt, erityisesti ajoitukseen liittyvät, ovat usein käytännössä mahdottomia mallintaa todellista laitetta vastaaviksi, mutta tavallisten toimintojen oikeellisuudesta saadaan usein simulaattoreilla hyvä tuntuma, ja virheitä voidaan korjata varsinaisen laitteen valmistumista odoteltaessa. Simulaattorilla voidaan myös turvallisesti testata poikkeustilanteita, jotka ovat todellisessa järjestelmässä vaarallisia, esimerkiksi ydinvoimalaitoksen hätäalasajo. Yleen-

sä varsinaista verifiointia ei kuitenkaan voida simulaattorilla tehdä, mainittujen puutteiden vuoksi.

Ajoitus- ja muistitestit

Järjestelmän toiminnoille asetettujen ajoitus- ja muistirajoitusten täyttyminen on tutkittava. Ajoitustekijät ovat tärkeitä erityisesti reaaliaikajärjestelmissä. Kyseiset ominaisuudet on yleensä mielekästä testata vasta hyvin lähellä valmista tuotetta olevalla järjestelmällä. Työkaluina voidaan käyttää erilaisia muistinhallinnan ja prosessinsuorituksen seurannan ohjelmistoja.

Suorituskyvyn testaus

Järjestelmän suorituskyvylle on yleensä vaatimusmäärittelyssä annettu joukko ehtoja. Tyypillisiä ovat esimerkiksi aikarajoitukset, eli tietyn toiminnan toteuttamiseen saa kulua enintään annettu aikamäärä. Suorituskyvyn mittaukset ovat tyypillisiä esimerkkejä funktionaalisesta black box -testauksesta.

Rasitustestaus

Järjestelmän toimivuuden tutkimista poikkeuksellisen kovissa kuormitusolosuhteissa kutsutaan stressitestaamiseksi. Kuormituksen laatu riippuu ohjelman tyypistä. Kuormitusta voidaan aiheuttaa ohjelman syötteillä tai tietokoneessa samanaikaisesti ajettavilla muilla ohjelmilla. Käytettävän stressikuormituksen kovuus riippuu normaalitilanteiden kuormittavuudesta. Usein ollaan kiinnostuneita siitä, suoriutuuko ohjelma laajasta tai monimutkaisesta tehtävästä siedettävässä ajassa. Myös tietojen säilymistä on tarkkailtava.

5.3.5 Analyysimenetelmien valinta

Analyysimenetelmien tulee olla tunnustettuja ja tieteellisesti päteviä menetelmiä. Lisäksi niiden pitää soveltua analysoitavaan järjestelmään ja niiden käyttäjillä on oltava tarpeellinen menetelmäkoulutus. Analyysin on oltava jäljitettävissä, toistettavissa ja verifioitavissa, ja se tuottaa tulokset muodossa, joka auttaa riskin luonteen ymmärtämisessä ja valvonnassa.

Analyysimenetelmän valintaan vaikuttavat lukuisat tekijät. Siksi valitseminen onkin yksilöllinen prosessi, mihin ei ole mahdollista suosittaa yleistä kaikille sovellusaloille sopivaa ehdotusta. Sopivien menetelmien valitseminen on sovellusalan asiantuntijoiden tehtävä. Suositeltavinta on valita menetelmät varhaisessa vaiheessa tarkastelun kohteena olevassa kehitysprosessia. Usein koko analysointi vaatii useasta asiantuntijasta koostuvan ryhmän, jossa yhdistyy

- vahva käytännön kokemus ja kyky vetää projekteja

- sovellusalueen tuntemus
- ohjelmistotekniikan tuntemus
- analyysitekniikoiden tuntemus.

Analyysimenetelmän valintaa voidaan kuitenkin helpottaa seuraavilla kriteereillä, jotka ovat standardista IEC 60300-3-1 (2001). Kriteerien yhteydessä mainitaan standardin IEC 60300-3-1[3] suositukset tässä julkaisussa käsitellyille menetelmille: vikapuuanalyysi sekä vika- ja vaikutusanalyysi. Jos kumpikaan analyyseista ei ole sovelias, suositetaan muita menetelmiä kriteerin yhteydessä. Kriteerit ovat:

a) Järjestelmän monimutkaisuus. Mm. redundanssin ja diversiteetin (erilaisuuden) lisääminen järjestelmään lisää myös perusteellisen analysoinnin tarvetta. Soveltuu sekä vikapuuanalyysille että vika- ja vaikutusanalyysille.

b) Järjestelmän uutuus. Kokonaan uuden järjestelmän suunnitteleminen voi edellyttää perusteellista analysointia. Sekä vikapuuanalyysi että vika- ja vaikutusanalyysi sopivat.

c) Kvalitatiivinen/kvantitatiivinen analyysi. Mahdollisesti välttämättä tarvitaan kvantitatiivista analyysia. Sekä vikapuuanalyysi että vika- ja vaikutusanalyysi sopivat.

d) Yksinkertaiset/moninkertaiset viat. Seuraukset voivat aiheutua sellaisista vikojen yhdistelmistä, joita ei voida ehkäistä. Vikapuuanalyysi sopii, vika- ja vaikutusanalyysi ei.

e) Aika-/jonoriippuva käyttäytyminen. Järjestelmä voi olla aikariippuva, esimerkiksi vikaantumisen jälkeen se noudattaa hallitun suorituskyvyn alentumisen periaatetta. Tapahtumien järjestys voi vaikuttaa järjestelmän vikaantumiseen. Esimerkiksi tapahtuma A aiheuttaa tapahtuman B, ei päinvastoin. Kumpikaan tarkastelluista analyyseista ei sovellu kriteerin täyttämiseen. Sopivia ovat Bayesin luotettavuusanalyysi, tapahtumapuuanalyysi, Markovin analyysi ja Petriverkkoanalyysi.

f) Riippuvat tapahtumat. Vikaantumis- ja korjausominaisuudet voivat olla riippuvia järjestelmän tilasta. Kumpikaan tarkastelluista analyyseista ei sovellu. Kriteerin yhteydessä tulisi valita jokin seuraavista analyyseista: tapahtumapuu, Markov ja Petriverkko.

g) Vaikutusten/syiden tunnistaminen. Vaikutusten tunnistaminen on usein suoraviivaista ainakin ylimmillä tasoilla. Syiden tunnistaminen vaatii harkintaa ja on virhe-

[3] Standardiluonnos IEC 60300-3-1 Ed.2 ohjeistaa hyvin lyhyesti edellä mainittujen analyysien lisäksi seuraavia menetelmiä: Vikataajuuden ennustaminen, Markovin analyysi, Petriverkkoanalyysi, HAZOP, inhimillisten tekijöiden analyysi, kuorimitus- ja lujuusanalysi, totuustaulukko sekä Bayesin luotettavuusanalyysi.

altista. Vika- ja vaikutusanalyysi lukeutuu ensisijassa vaikutusanalyyseihin. Siinä syiden tunnistaminen ei ole etusijalla. Vikapuuanalyysi on vain syiden tunnistamista varten, vaikeuksia silla ei tarkastella.

h) Luotettavuusvaatimusten allokointi. Suunnittelu voi vaatia luotettavuusvaatimusten allokointia suunnitteluosille. Vikapuu soveltuu erityisen hyvin, vika- ja vaikutusanalyysi vain varauksin tämän kriteerin täyttämiseen.

i) Taitavuus. Jotkut analyysimenetelmät vaativat perehtyneisyyttä ja kokemusta. Sellainen on jossakin määrin vikapuuanalyysi, vika- ja vaikutusanalyysi on nopeasti omaksuttavissa.

j) Yleisyys ja hyväksyttävyys. Sekä vikapuuanalyysi että vika- ja vaikutusanalyysi ovat yleisiä ja hyväksyttäviä menetelmiä lääkintälaitteiden standardienkin mukaan.

k) Tukivälineiden tarve. Vikapuuanalyysin tulosten kvantitatiivista laskentaa ja graafista piirtämistä varten suositellaan erityistyökaluja. Vika- ja vaikutusanalyysille riittää normaali taulukointiohjelma.

l) Uskottavuustarkistukset. Vikapuuanalyysin sekä kvalitatiivinen että kvantitatiivinen suorittaminen vaatii kelpoistamista sillä se on hyvin virheherkkä. Myös vika- ja vaikutusanalyysin tulokset vaativat tarkistusta.

m) Työkalujen saatavuus. Sekä vikapuuanalyysille että vika- ja vaikutusanalyyseille on olemassa runsaasti kaupallisia työkaluja.

n) Standardointi ja ohjeistus. Sekä vikapuuanalyysille että vika- ja vaikutusanalyyseille on olemassa runsaasti ohjeita. Edellinen on standardoitu IEC 61025:ssa ja jälkimmäinen IEC 60812:ssa.

Analyysin epäonnistumiseen vaikuttavia tekijöitä voivat olla

- kokemusperäisen tiedon puuttuminen
- systemaattisen työtavan puuttuminen
- väärä rajaus
- vikamekanismin virheellinen mallintaminen (alku-, väli- ja lopputapahtumat)
- analyysin väärä aloitustaso.

Vikamekanismit ohjelmistopohjaisissa järjestelmissä ovat usein laajoja ja monimutkaisia, ja silloin mallintamisessa tarvitaan alan kokemusta ja vastaavien järjestelmien vika-, vaikutus-, havaitsemis- ja riskikontrollitietoja. Systematiikka on dokumentoitavan osoittamistavan kulmakiviä. Yhdessä standardinmukaisuuden kanssa systemaattisella

analysoinnilla saadaan kattava vikamalli aikaan. Rajaamisessa on kriittisen alueen (järjestelmä, ympäristö, työvälineet jne.) oltava ainakin mukana, mutta myös liian laajan tarkastelualan valinta tekee analysoimisesta helposti virhealttiin.

Ohjelmistokehityksen vaiheita analysoitaessa oikean lähtötason valinta riippuu monesta seikasta. Tärkeimmät valintaan vaikuttavat tekijät ovat dokumentaation laajuus, yksityiskohtaisuus ja täsmällisyys sekä tunnistettavien seikkojen tunnistamisen helppous. Esimerkiksi jos valitaan ohjelmiston lähdekielinen koodi lähtötasoksi, saattaa seurausvaikutusten tunnistaminen olla hyvin työlästä. Vaikutuksia kuvaava tapahtumaketju kasvaa helposti liian suureksi. Jos taas ollaan liian ylhäällä, esimerkiksi järjestelmätasolla, tulee vikapuu liian laajaksi analysointia varten.

6. Riskienhallintaprosessi

Tässä luvussa kuvataan lääkintälaitteen ohjelmistoa sisältävän tuotteen kehittämistä tukeva riskienhallintaprosessi, joka viedään läpi tuotekohtaisen riskienhallintasuunnitelman mukaisesti. Se etenee systemaattisesti tuotteen kehittämisvaiheiden myötä vaarojen tunnistamisesta riskien suuruuden ja merkityksen arvioinnin kautta suunnittelun lähtötietoihin vaikuttavien riskien vähimmäistämiseen ja tuotannon jälkeisiin käyttö-, ylläpito- ja muutosvaiheisiin. Tarvittavien toimien raportointi osaksi tuotteen riskienhallintakansiota kuvataan.

6.1 Johdanto

Riskienhallinta on laaja käsite. Sitä tarvitaan, koska riskeihin liittyy aina epävarmuustekijöitä. Riskienhallinnalla tarkoitetaan päätöksentekoprosessia, jossa päätetään hyväksyttävän riskin tasoista ja menetelmistä, joilla asetetut tasot saavutetaan. Lisäksi riskienhallintaa on asetettujen tasojen noudattamisen valvonta.

Riskienhallintaprosessi on kokonaisvaltainen suunnittelun tukiprosessi. Se liittää yhteen monia eri elementtejä vaarojen alustavasta tunnistamisesta riskien siedettävyyden arviointiin ja mahdollisten riskiä pienentävien ratkaisujen tunnistamiseen sekä tarkoituksenmukaisten valvonta- ja parannustoimenpiteiden valintaan, toteuttamiseen ja seurantaan. Prosessi vaiheistetaan koko tuotteen elinkaarelle siten, että kunkin elinkaaren vaiheen jälkeen on arvioitava, miten vaiheen aikana tehdyt ratkaisut ja toimenpiteet vaikuttavat laitteen turvallisuuteen. Riskienhallintaprosessin vaiheistukseen sisältyvät seuraavat toimenpiteet:

1. ***Riskianalyysi*** on jäsennelty prosessi, joka tunnistaa tuotteen käyttötarkoituksen ja vaarat ja arvioi vaaroista johtuvien riskien suuruuden sekä sen, mikä on haitallisten seurausten todennäköisyys ja laajuus (ks. kohta 6.4).

2. ***Riskien merkityksen arvioinnissa*** päätetään riskienhallinnan suunnitelmassa (ks. kohta 6.5) esitetyillä kriteereillä jokaiselle tunnistetulle vaaralle arvioidun riskin suuruuden perusteella mahdollisista riskien vähentämistoimenpiteistä.

3. ***Riskin valvonta*** on jäsennelty prosessi tarvittavan riskin vähentämistoimenpiteiden ohjaukseen ja valvontaan siten, että jokaiselle vaaralle jäännösriski on hyväksyttävä (ks. kohta 6.6).

4. ***Tuotannon jälkeisen tiedon käsittely*** on järjestelmällinen menettelytapa, jossa katselmoidaan lääkintälaitteesta tai vastaavasta laitteesta saatu riskeihin liittyvä informaatio (ks. kohta 6.7).

Tehokkaaseen yrityksen riskienhallintaan kuuluvat myös yritysjohdon määrittelemät riskienhallinnan periaatteet, jotka ohjaavat yrityksen linjaa hallita riskejä. Periaatteet näkyvät yrityksen kaikissa toiminnoissa (yritystoiminnan riskit, teknologiariskit, tuoteriskit, projektinhallinnan riskit) selkeinä eri osastojen välisinä yhtenäisinä toimintaohjeina, työkaluina sekä menettelytapoina.

Terveydenhuollon tuotteita koskevassa direktiivissä (MDD 1993: liite G) on riskienhallinnalle tärkeä maininta, jonka mukaan laitteen ja tarvikkeen käyttöön mahdollisesti liittyvien riskien hyväksyttävyydestä päätettäessä voidaan ottaa huomioon potilaan saama hyöty. Edelleen mainitaan, että valitessaan turvallisuuden varmistamiseksi soveltuvia ratkaisuja valmistajan tulee käyttää seuraavia periaatteita:

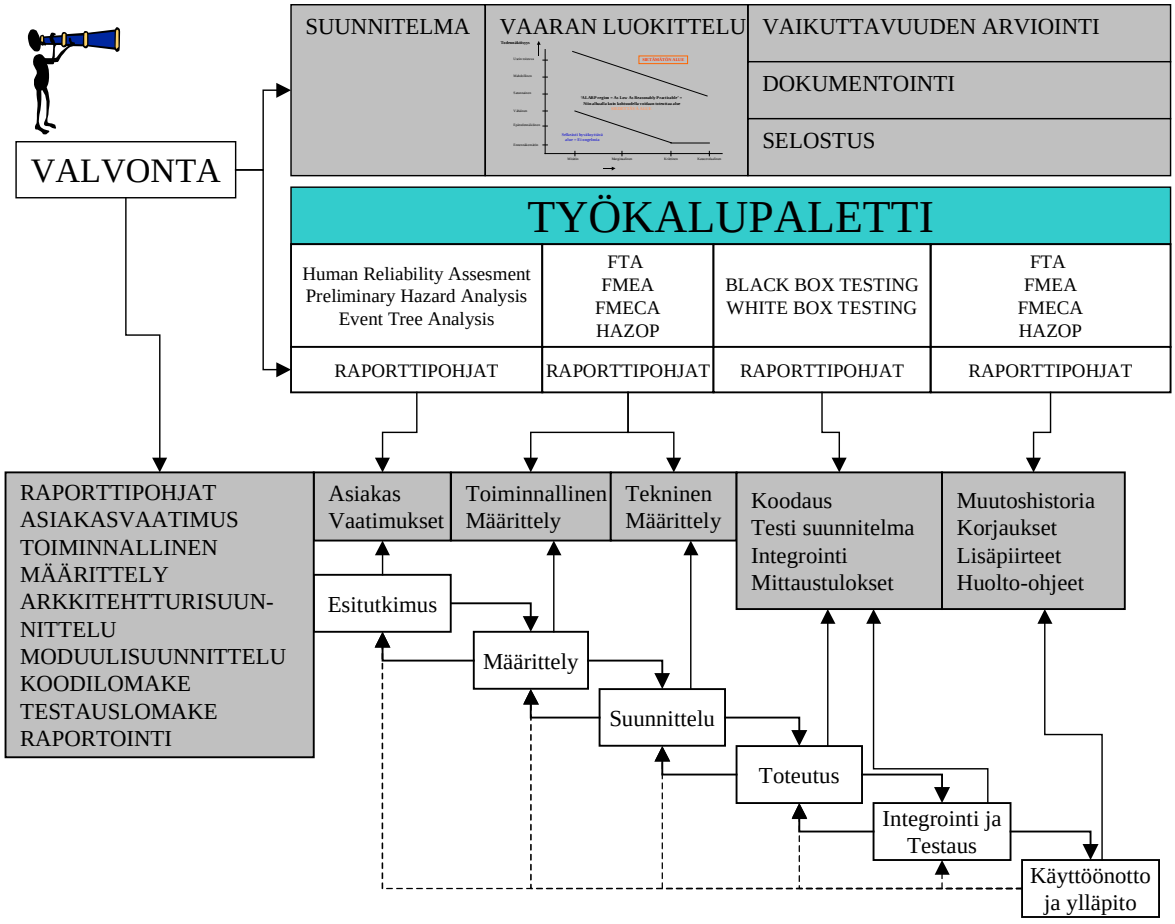
- poistettava tai vähimmäistettävä riskit
- toteutettava tarvittaessa asianmukaiset suojelutoimenpiteet, mm. riskien hälytysjärjestelmä, jos riskit eivät ole poistettavissa
- tiedotettava käyttäjille jäljellä olevista riskeistä, jotka johtuvat toteutettujen suojelutoimenpiteiden riittämättömyydestä.

Lisäksi direktiivi vaatii, että kaikista ei-toivotuista sivuvaikutuksista aiheutuvien riskien on oltava hyväksyttäviä verrattuna terveydenhuollon tuotteen suunniteltuun suorituskykyyn.

Riskienhallintaprosessi on myös yrityksen toimintaa kehittävä prosessi, koska sillä saadaan seuraavia toimintaetuja:

- tunnistetaan häiriöt jo ennen vahingon syntymistä
- vähennetään häiriöitä
- hyödynnetään palautetietoja parannusehdotuksissa
- kehitetään työvälineitä
- kyetään hallitsemaan menetelmäkuvauksia
- tarkennetaan vaihedokumenttien sisältöä
- löydetään mahdolliset organisaattoriset ongelmat
- parannetaan jäljitettävyyttä dokumenttien välillä.

Kuva 11 esittää joitakin riskienhallintaan kuuluvista elementeistä.



Kuva 11. Esimerkki riskienhallinnan prosessikuvauksesta, jossa on valmistajan tarvitsemia työvälineitä riskin analysoimiseksi, valvomiseksi ja tulosten dokumentoimiseksi.

Riskienhallintaprosessissa noudatetaan määriteltyjä menetelmäohjeita mm. vakavuuden luokittelulle, tuotekohtaiselle suunnittelulle, riskin vaikuttavuuden arvioinnille sekä riskianalyysin suorittamiselle. Projektin alussa prosessi tukee menetelmäohjeilla alustavien riskienhallinta- verifiointi- ja validointisuunnitelmien määrittelyä.

Standardi SFS-EN 60601-1-4 edellyttää, että riskienhallintaprosessin menetelmäohjeita noudatetaan läpi koko tuotekehityksen, joka sisältää riskianalyysin ja riskin valvonnan. Prosessin päämääränä on hallita riskiä siten, että se on jatkuvasti suurinta siedettävää riskiä pienempi ja niin alhaalla kuin kohtuudella voidaan toteuttaa.

6.2 Terminologia ja käsitteet

Riskienhallinnan termit ja käsitteet vaihtelevat aiheuttaen joskus sekaannusta, joten taulukkoon 12 on koottu riskienhallinnan oleellisimmat termit standardeista SFS-EN 60601-1-4 (1996, 1999), SFS-EN1050 (1997), SFS-EN 1441 (1998) ja ISO/FDIS 14971 (2000).

Taulukko 12. Tässä julkaisussa noudatettavat riskienhallinnan termit.

Termi	Lähde	Selitys
VAHINKO *Harm*	SFS-EN1441	Fyysinen vamma, terveyshaitta tai omaisuusvahinko
VAARA *Hazard*	ISO/FDIS 14971	Vahingon aiheuttaja
VAARATILANNE *Hazardous situation*	ISO/FDIS 14971	Olosuhteet, joissa ihmiset, omaisuus tai ympäristö ovat alttiina vaaralle
SUURIN SIEDETTÄVÄ RISKI *Maximum Tolerable Risk*	IEC601-1-4	Suurin mahdollinen riskin arvo, joka voidaan sallia
RISKI *Risk*	ISO/FDIS 14971	Vahingon esiintymistodennäköisyyden ja vakavuuden yhdistelmä
JÄÄNNÖSRISKI *Residual risk*	ISO/FDIS 14971	Turvallisuustoimenpiteiden toteuttamisen jälkeen jäljelle jäävä riski.
TURVALLISUUS *Safety*	ISO/FDIS 14971 IEC601-1-4	Tila, jossa vahingon riski on hyväksyttävällä tasolla.
RISKIANALYYSI *Risk anlysis*	ISO/FDIS 14971	Saatavilla olevan tiedon käyttö vaarojen tunnistamiseksi ja riskin suuruuden arvioimiseksi.
VAARAN TUNNISTUS *Hazard Identification*		Vaaran olemassaolon tunnistaminen ja sen karakterisointi.
RISKIN VALVONTA *Risk Control*	ISO/FDIS 14971	Prosessi, jolla tehdään päätökset ja toimenpiteet riskin pienentämiseksi määritellylle tasolle tai pitämiseksi riski määritellyllä tasolla.
RISKIN ARVIOINTI *Risk assesment*	ISO/FDIS 14971	Riskianalyysin ja riskin merkityksen arvioinnin kokonaisprosessi.
RISKIN MERKITYKSEN ARVIOINTI *Risk Evaluation*	ISO/FDIS 14971	Riskianalyysiin perustuva päätös siitä, onko hyväksytty riski saavutettu perustuen yhteiskunnan asettamiin sen hetkisiin arvioihin.
RISKIN SUURUUDEN ARVIOINTI *Risk Estimation*	SFS-EN 1050	Vahingon vakavuuden ja esiintymistodennäköisyyden tulo.
RISKIENHALLINTA *Risk Management*	ISO/FDIS 14971	Yritysjohdon systemaattisesti soveltama menettelytapa, joka sisältää menettelytavat ja käytännöt tehtävien riskin analysoimiseksi, merkityksen arvioimiseksi ja hallitsemiseksi.
TURVALLISUUS-TOIMENPIDE Safety measure	SFS-EN 1050	Toimenpide, joka poistaa vaaran tai pienentää riskiä
VAKAVUUS *Severity*	ISO/FDIS 14971	Mitta-asteikko mahdollisen vaaran seurauksille

6.3 Luettelo vaaratekijöistä riskienhallintakansioon

Standardiluonnoksen ISO/DIS 14971 mukaan valmistajan tulee koota ja ylläpitää luetteloa lääkintälaitteen tunnetuista ja kohtuudella ennalta nähdyistä normaali- ja vikatilanteisiin liittyvistä vaaroista. Luetteloa on ylläpidettävä riskienhallintakansiossa. Soveltuvan esimerkin luettelon sisällöstä ja formaatista saa lääkintälaitteen riskianalyysistandardista SFS-EN 1441. Siitä on esimerkki liitteessä C.

Terveydenhuollossa hoidon tai tutkimuksen aikana potilas joudutaan tarkoituksella altistamaan vaaroille. Vaarojen tulee kuitenkin aina olla oikeassa suhteessa saavutettuun hyötyyn. Riskienhallinnalla poistetaan käyttäjään, huoltohenkilöön, muihin henkilöihin ja ympäristöön kohdistuvat tarpeettomat vaarat.

Toteutuneessa vaaratilanteessa vaaran poistaminen tai riskin pienentäminen on yleensä mahdotonta tai ainakin vaikeaa ja työlästä. Tulisikin jo kehitysprosessissa selvittää mahdollisten vaarojen syntymekanismit siten, että riskienhallinnan keinoilla kyetään pureutumaan vaaratekijöiden alkutapahtumiin. Tällä menettelyllä suunnitteluprosessi tehostuu niin aikataulullisesti kuin kustannuksellisestikin.

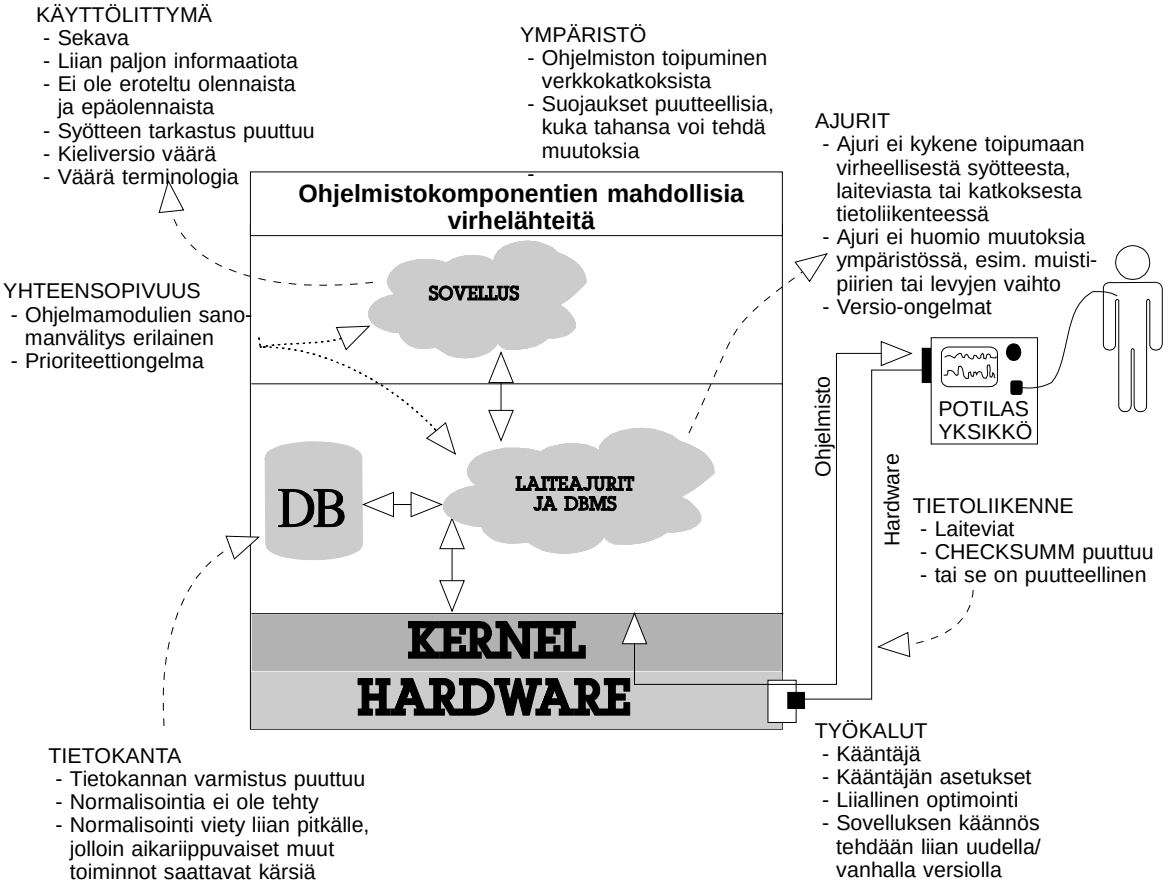
Tarkastellaan aluksi ohjelmistopohjaisen järjestelmän tai laitteen vaaratekijöitä ja sitten turvallisen suorittamisen oletusarvoja ja väärään käyttöä sekä niiltä suojautumista. Kohdassa 6.3.3 tähdennetään vaaratekijöiden ottamista huomioon jo suunnittelun alkuvaiheissa.

6.3.1 Järjestelmän vaaratekijät

Monimutkaisen terveydenhuollon tuotteen toimintahäiriöt voivat aiheutua useasta samanaikaisesta viasta. Lääkintälaitteiden perusstandardi SFS-EN 60601-1 edellyttää laitteen toimivan turvallisesti vielä yhden vian tapauksessa. Siksi järjestelmä tulisikin suunnitella vikasietoiseksi ja turvallisesti vikaantuvaksi. Järjestelmän tulisi suoriutua tärkeistä toiminnoistaan teknologia-, oletusarvo- ja käyttövirheiden esiintyessä tai vaihtaa tarvittaessa suorittaminen turvalliseen tilaan. Turvalliseen tilaan johtaminen merkitsee usein järjestelmän toiminnan pysäyttämistä, mitä hoitotilanteesta riippuen ei aina voida pitää oikeana ratkaisuna. Sekä vikasietoisuus että turvallisesti vikaantuminen edellyttävät tapauksesta riippuen tilojen valvontaa ja/tai viestintää järjestelmän käyttäjälle (ks. jäännösriskien ilmoitusvelvollisuudesta kohdassa 6.4.2).

Standardi SFS-EN 60601-1-4 edellyttää valmistajan tunnistavan järjestelmän vaarat ja alkutapahtumat, jotka voivat olla peräisin mm. seuraavista järjestelmän asioista (ks. kuva 12):

- järjestelmän ohjelmisto- ja laitteistokomponenttien yhteensopivuus
- käyttöliittymä, komentokielen varoitukset ja virheviestit
- käyttöliittymä- ja käyttöohjetekstin kielenkäännöksen täsmällisyys
- tiedon suojaus inhimillisiltä, tahallisilta ja tahattomilta syiltä
- riski/hyötykriteerit
- kolmannen osapuolen ohjelmistot.



Kuva 12. Ohjelmiston virhetoimintaan johtavia syitä.

Mitä aikaisemmassa vaiheessa vaaratekijät tunnistetaan, sitä paremmin vaarojen valvontatoimenpiteet saadaan määritettyä vaatimuksiksi ohjelmiston vaatimusspesifikaatioon (ks. esimerkkejä taulukossa 13) Vaaroja kartoitetaan esimerkiksi alustavalla vaaraanalyysilla, vikapuuanalyysilla tai vika- ja vaikutusanalyysilla (ks. luku 5).

Taulukko 13. Riskianalyysissa havaitun riskin muuttuminen vaatimukseksi.

ALKUTAPAHTUMA, VAARA, RISKI	**VAATIMUS ohjelmiston vaatimusmäärittelyssä**
Ohjelman vasteaika liian hidas	Keskusmuistia 128 MB
Kuvassa virheellinen löydös	Algoritmi kuvan tarkastamiseksi
Virheellinen syöte	Syötteen tarkastus
Tietoliikenneverkon katkos	Verkon monitorointi, ilmoitus käyttäjälle katkoksesta
Käyttäjä aktivoi toiminnon väärään aikaan	Toiminnon aktivointi kielletty tässä vaiheessa
Ohjelma harhautuu väärään ohjelmamoduliin	Tarkastus ohjelmamodulissa, estetään pääsy

Ohjelmiston kriittisiä toimintoja ovat mm. suuri CPU-kuormitus, dynaaminen muistin allokointi, ajastukset, sanomanvälitys ja samanaikaisten prosessien käyttäytyminen.

Ohjelmistopohjaisten järjestelmien käyttökokemuksia ja tapahtuneita virheitä ei yleensä ole kerätty tai dokumentoitu kattavasti. FDA ylläpitää kuitenkin tietokantaa kaikista ilmoitetuista lääkintälaitevioista ohjelmistovirheet mukaan lukien. Kohdassa 3.4.6 kuvataan yhteenveto luokitelluista ohjelmistovirheistä ilmitulotapoineen sekä suosituksineen havainto- ja estokeinoiksi.

Leveson & Turnerin (1992) raportti kuvaa tietokoneohjatun sädehoitolaitteen käyttöön liittyviä ohjelmistoperäisiä onnettomuuksia. Heidän mukaansa mikään onnettomuus ei johdu pelkästään tietokonevirheistä vaan hyvin monista kokonaisjärjestelmään liittyvistä tekijöistä. 1980-luvun puolivälissä tapahtuneet onnettomuudet muuttivat FDA:n ohjeistusta. Aikaisemmin ei vaadittu ilmoituksia laitteiden käytön aikaisista häiriöistä, vaan ainoastaan valmistajan tuli raportoida esiin tulleista vioista. Niinpä FDA ei ensimmäisissä onnettomuuksissa kyennyt päättelemään häiriön varsinaista yhteyttä ohjelmistovirheeseen. Mutta myös monet muut tekijät olisivat säästäneet onnettomuuksilta.

Sädehoitolaitteen onnettomuuksissa näitä tekijöitä olivat käyttöohjeiden puutteellisuus ja järjestelmän käytön huono johto, puutteellinen laadunvarmistus ja yliluottamus ohjelmaa kohtaan. Muita tekijöitä olivat huonot ohjelmointi- ja suunnittelukäytännöt ja epärealistiset riskiarvot, joiden perusteella ohjelmistoon luotettiin liikaa.

Hyvät käyttöohjeet olisivat estäneet onnettomuudet. Sädehoitolaitteen hoitaja sai tietyn toimintahäiriöilmoituksen, jonka ohjeet kuvasivat joko säteilyn ali- tai yliannostukseksi. Vastaava ilmoitus oli tullut usein ennenkin, joissakin laitteissa jopa päivittäin, joten hoitaja toimi kuten oli aina toiminut.

Onnettomuuksien arviointi paljasti yleensäkin yliluottamuksen tietokoneohjelmiin, jolloin muita turvallisuutta lisääviä käytäntöjä laiminlyötiin. Eräässä tapauksessa kameravalvonta oli epäkunnossa eikä hoitaja nähnyt potilasta erillisessä sädehoitohuoneessa.

Potilas osasi epäillä jostakin syystä sädehoitolaitteen toimintaa, mutta ei ehtinyt poistumaan sädehoitopöydältä ennen yliannostusta.

Vastaavassa muun yrityksen valmistamassa sädehoitolaitteessa oli mekaaniset suojaukset eikä laite aiheuttanut onnettomuuksia. Kyseisen onnettomuuslaitteen valmistanut yritys oli kuitenkin poistanut tuotteestaan mekaaniset varmistukset ja lukitukset ja korvannut ne ohjelmistopohjaisilla suojauksilla. Ensimmäisten onnettomuuksien jälkeen virhelähteeksi tulkittiin mitta-anturi, mutta ohjelmistoa ei osattu epäillä. Tulkintaa perusteltiin ensinnäkin sillä, että ohjelmointivirheet oli poistettu laajoilla testauksilla, toiseksi, että ohjelma ei vanhene, ja kolmanneksi, että tietokoneen toimintahäiriöt johtuvat komponenttivioista.

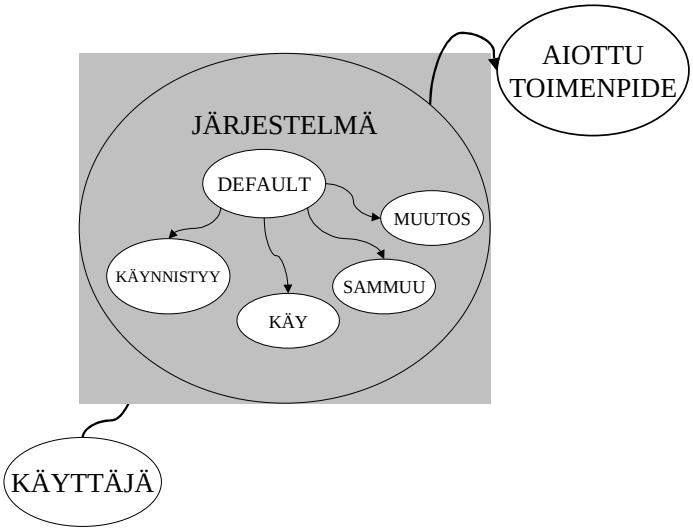
Epäilyt olisivat johtaneet ohjelmistoon, jos olisi tiedetty laitteen sisältävän erään aikaisemman laitemallin ohjelmistomoduulin. Moduulissa oli jo todettu sama ohjelmistovirhe, joka oli myöhemmässä mallissa aiheuttanut onnettomuudet, mutta tuotteen laatupäällikkö oli saanut nimenomaan tiedon, ettei tuotteessa ole kyseisen mallin ohjelmistomoduuleita.

Leveson ja Turner (1992) toteavat raportissaan, että pelkästään testein ja analyysein ei olisi kyetty tunnistamaan varsinaista ohjelmavirhettä kyseisessä sädehoitolaitteessa. Onnettomuudet tapahtuivat 1980-luvun puolivälissä ja niiden vaikutukset näkyvät lisääntyneissä riskienhallinnan ja laadunvarmistuksen toimintatavoissa kaikkialla ohjelmistoja hyödyntävissä teollisuusaloissa.

6.3.2 Oletusarvot ja väärä käyttö

On tärkeää arvioida järjestelmän teknologioiden soveltuvuus käyttötarkoitukseensa. Suunnittelussa kiinnitetään aina huomiota laitteen suorituskykyyn ja toimintaan, mutta tärkeitä ovat myös järjestelmän käynnistys, suorituksen tarkkailu ja sammutus. Tavoitteena on järjestelmän käynnistyminen turvalliseen ja käytön kannalta mahdollisimman valmiiseen tilaan.

Käyttötarkoitukseen kuuluvia ominaisuuksia pyritään huomioimaan oletusarvoilla, joiden määrittely monimutkaisessa lääkintälaitteessa on vaativaa. Turvallisuuden kannalta tärkeitä oletusarvoja ovat laitteen toimintatila, johon laite käynnistetään (ks. kuva 13), hälytysten esto/sallinta ja raja-arvot sekä potilastyypin valinta, jos sitä ei käynnistystoimenpiteen alussa jo ole kysytty.



Kuva 13. Järjestelmän on oltava nopeasti valmiustilassa.

Oletusarvojen määrittelyyn osallistuvat suunnittelijan lisäksi myös sovellusalueen asiantuntija, kliinikko ja käyttäjä, koska suunnittelija ei välttämättä tunne kaikkia sovellusalueen ja aiotun toimenpiteen asettamia vaatimuksia.

Vaaratekijät voivat aiheutua myös käyttäjän virheellisestä toimenpiteestä. Toimenpide saattaa aiheuttaa useita perättäisiä virhetoimintoja, joiden seurauksena on varsinainen vaaratilanne. Ketjureaktion monimutkaisuuden vuoksi käyttäjä ei useinkaan huomaa aiheuttamaansa virheellistä toimintoa. Siksi järjestelmä tulisikin suunnitella suojaamaan inhimillisten tekijöiden alullepanemista riskeistä.

Myös hyvin määritellyillä toimintatavoilla, oikealla koulutuksella ja käyttöönohjauksella, valvonnalla sekä luontaisella turvallisuussuunnittelulla kyetään pienentämään virheellisestä käytöstä johtuvia vaaratilanteita. Luontaisessa turvallisuussuunnittelussa tulisikin ottaa huomioon virheellisen käytön mahdollisuus.

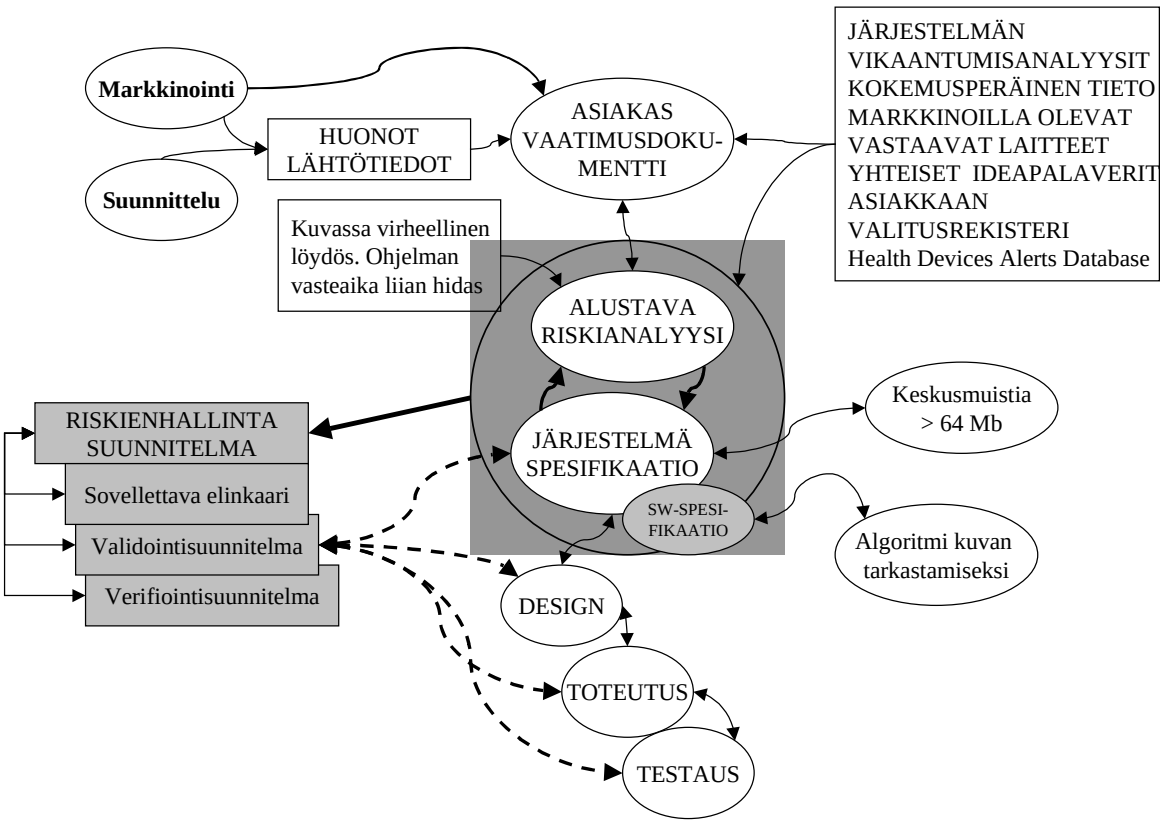
Alkutapahtumia ovat myös hoitotilanteen nopeat muutostapahtumat. Esimerkiksi hoitotapahtuma voi edellyttää välittömästi tiettyä muutosta, mutta laitteen muuttaminen tai muuttamista varten tarvittava alustaminen kestää liian kauan.

6.3.3 Vaaratekijät ajoissa huomioon

Tuotteen suunnitteluprojektissa on usein kyse pelkästään olemassa olevan tuotteen muuttamisesta tai uuden suunnittelusta vanhan olemassa olevan tuotteen teknologian päälle. Näissä tapauksissa valmistajalla on hyvä käsitys tuotteen vaaroista. Lisäksi hän

mahdollisesti tuntee markkinoiden vastaavien tuotteiden vaarat esimerkiksi ECRI:n *Health Devices Alerts Database* -tietokannasta. ECRI (http://www.ecri.org) on ylläpitänyt tietokantaa vuodesta 1977, ja siihen on kuvattu yli 850 000 tapahtumaa.

Vaaraluetteloa ylläpidetään riskienhallintakansiossa. Se voi olla tarkistuslista tai toimenpideohje, joka on sellaisessa muodossa, että se on helposti ylläpidettävissä ja toistettavissa sekä saatavilla suunnittelua mahdollisimman hyvin tukevassa muodossa.



Kuva 14. Eräät suunnittelun vaatimat lähtötiedot hyödyntävät myös markkinointia. Mahdollinen yhteinen menetelmäohje tukee kumpaakin toimintaa .

Olemassa olevan tiedon hyödynnettävyys edellyttää yhteisten menetelmäohjeiden kirjoittamista markkinoinnille ja suunnittelulle. Menetelmäohjeet tukevat toimintaa raporttipohjilla ja tarvittaessa hyväksyntä- ja tarkastuskriteereillä. Menetelmäohjeiden tulee tukea markkinointia siten, että ohjeita noudattamalla tuotetaan suunnittelutiimin ymmärtämä asiakasvaatimusdokumentti. Dokumentin tulee olla niin yksityiskohtainen, että suunnittelutiimi kykenee tuottamaan siitä alustavalla vaara-analyysilla järjestelmän määrittelyn (ks. kuva 14). Alustavia vaara-analyyseja ja järjestelmän määrittelyitä tehdään iteroiden siten, että vielä projektin edetessä menetelmäohjetta tarkennetaan.

Suunnittelun alkuvaihe hyötyy menetelmäohjeista ainakin seuraavasti:

- Suunnitteluprosessi saa määrämuotoiset dokumentit markkinoinnilta.
- Määrämuotoisten dokumenttien analysointi on laadukasta.
- Menetelmäohjeilla kyetään ongelmatilanteet havaitsemaan ajoissa, esimerkiksi markkinointi on voinut antaa puutteellisia lähtötietoja.
- Suunnitteluprosessiin saadaan toistettavuutta.
- Aikataululliset riskit kyetään arvioimaan paremmin.
- Valmiit tarkastuslistat vaaroista, syistä ja avainsanojen käytöstä.

6.4 Riskianalyysi

Valmistajan on tunnistettava lääkintälaitteen vaarat ja vaaralliset tapahtumat kaikissa kohtuullisesti ennalta nähtävissä olosuhteissa ml. vikatilanteet ja väärä käyttö (SFS-EN 60601-1-4 1996, 1999). Vaara-analyysissa tunnistetaan soveltuvin osin potilaan, käyttäjän, huoltajan, sivullisen ja ympäristön turvallisuutta vaarantavat tapahtumat. Vaaroihin johtavien tapahtumien syyt määritetään. Niihin sisältyvät soveltuvin osin laitteisto-, ohjelmisto- ja integrointiviat sekä ympäristö- ja inhimilliset tekijät, ml. ergonomiset rajoitteet. Riskianalyysi kattaa siis laitteen, ympäristön ja ihmiset.

Vaarojen tunnistamisen lisäksi riskianalyysiin kuuluvat vaarallisista tapahtumista johtuvien mahdollisten seurausten määrittämiset. Lisäksi määritetään näiden tapahtumien todennäköisyydet. Tietyn tapahtuman todennäköisyys voidaan ilmaista määrällisesti tai laadullisesti.

Vaarojen tunnistamisessa tulee käyttää tuotteen kehityselinkaaren vaiheeseen sopivia menetelmiä. SFS-EN 60601-1-4 määrää myös menetelmävalinnat kirjattavaksi riskienhallintakansioon. Dokumenttiin kirjataan myös menetelmillä saadut tulokset sekä vakavuustasojen luokitustapa, jota tarkastellaan kohdassa 6.4.1.

Riskienhallinnan yhteenvetoon kirjataan riskianalyysien tuloksista kaikki tunnistetut vaarat, niihin johtavat alkutapahtumat ja riskin suuruus. Myös riskienhallintakansion vaaraluetteloa pitää päivittää.

6.4.1 Riskin luokittelu

Riskin suuruus *R* ilmaistaan yleensä vaarallisen tapahtuman *i* vakavuuden *S* ja vaarallisen tapahtuman todennäköisyyden tai –taajuuden F_I tulona:

$$R = S \times F_i . \tag{3}$$

Riskienhallinta sisältää vaaroihin liittyvien riskien luokittelumenettelyt (taulukko 14). Niillä tuetaan päätöksentekoa arvioitaessa tuotekohtaisen riskin pienentämistä ja hyväksyttävyyttä. Luokittelussa riskin suuruus voidaan ilmaista joko määrällisesti tai laadullisesti, mutta yleensä riskin kummatkin elementit, *S* ja *F*, ilmaistaan laadullisesti alla olevan taulukon tapaisesti.

Taulukko 14. Riskitaso.

SEVERITY (VAKAVUUS) **LIKELIHOOD** **TODENNÄKÖISYYS**	Neglible Mitätön	Marginal Marginaalinen	Critical Kriittinen	Catatrophic Katastrofaalinen
Frequent / Usein toistuva				
Probable / Mahdollinen				
Occasional / Satunnainen				
Remote / Vähäinen				
Improbable / Epätodennäköinen				
Incredible / Ennen näkemätön				

Riskin vakavuustasot voidaan kuvata esimerkiksi seuraavasti:

Katastrofaalinen	Mahdollisesti useita kuolemantapauksia tai vakavia vammautumisia
Kriittinen	Mahdollisesti yksi kuolema tai vakava vamma
Marginaalinen	Mahdollinen vammautuminen
Mitätön	Pieni vammautumisen mahdollisuus

Riskin toisen elementin eli todennäköisyyden määrittäminen saattaa yleisestikin olla vaikeaa. Erityisen vaikeaa se on systemaattisten virheiden kohdalla, joiden ohjeistaminenkin on standardiluonnoksessa SFS-EN 60601-1-4 vasta valmisteilla. Lääkintälaitteiden riskienhallinnan standardinluonnos ISO 14971 opastaa liitteessä E hyödyntämään asiaan kuuluvia historiatietoja tai asiantuntija-arvioita. Lisäksi viite neuvoo arvioimaan todennäköisyyttä luotettavuusanalyyseilla tai simulointitekniikoilla.

Ohjelmistojen käyttökokemusten keräyksessä ja analyysissa on varsin paljon puutteita. Ohjeet ovat harvinaisia. FDA ylläpitämästä ja julkaisemasta tietokannasta on apua vir-

heiden esiintymisanalyysiin ja johtopäätösten tekoon (ks. kohta 3.4.6). Luotettavuusanalyysienkin kanssa täytyy kuitenkin olla varovainen. Jos järjestelmästä tehty luotettavuustarkastelu on tuottanut hyvin pieniä virhetodennäköisyyksiä, voidaan helposti jättää tunnistamatta ohjelman osuus tapahtuneissa onnettomuuksissa (Leveson & Turner 1992).

Käyttökokemukset muista kuin juuri tarkastelun alla olevista järjestelmistä eivät tietenkään auta suoraan tämän järjestelmän luotettavuusanalyyseissa. Niistä saatava apu perustuu siihen, että käyttökokemuksista voi löytyä sellaisia virhetekijöitä, jotka ovat mahdollisia virhetekijöitä joissakin tarkasteltavaa järjestelmää koskevissa tilanteissa. Toisaalta virheetöntä toimintaa osoittavat käyttökokemukset voivat Leveson & Turnerin (1992) mielestä olla harhaan johtavia, koska kyse on yleensä räätälintyönä kuhunkin sovellukseen tehtävistä tuotteista.

6.4.2 ALARP -periaate

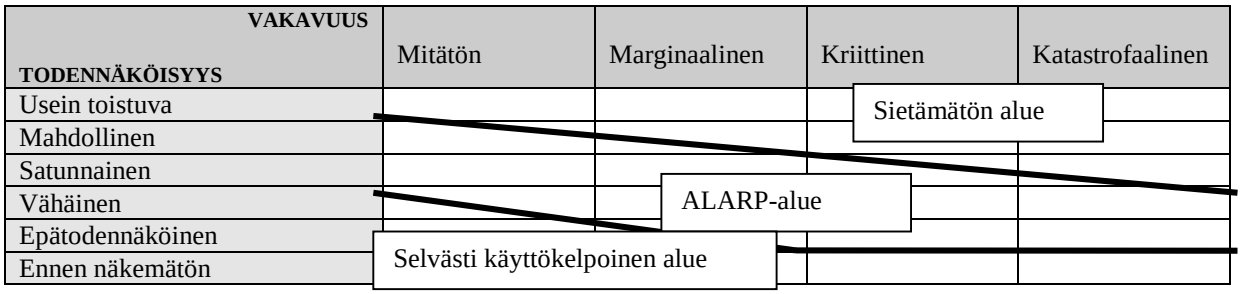
ALARP-periaatetta selostetaan standardeissa SFS-EN 60601-1-4, IEC 61508 ja IEC 14971. ALARP (As Low As Reasonably Possible) -alueella riskit on vähennetty alhaisimmalle kohtuudella toteutettavissa olevalle tasolle, jossa riski on vielä siedettävä. Erityisesti ALARP-alueella joudutaan pohtimaan jäännösriskin siedettävyyttä, sillä alue sijaitsee sietämättömän riskialueen ja selvästi käyttökelpoisen riskialueen välissä, missä toimenpiteisiin ryhtyminen on helppoa (ks. taulukko 15).

Sietämättömällä alueella riski on aina niin vakava, että sitä on alennettava pienentämällä vakavuutta ja/tai vaaran todennäköisyyttä.

Riski on selvästi käyttökelpoisella alueella, kun vakavuus ja/tai vaaran todennäköisyys on niin pieni, että riski on mitätön verrattuna muiden vaarojen hyväksyttyihin riskeihin. Tällä alueella ei tarvita riskinvähennyksen toimenpiteitä. On huomattava, että standardi SFS-EN 60601-1-4 ei määrittele hyväksyttävää riskiä, vaan jättää määrittelyn erityisstandardeille. Hyväksyttävä riski on usein pääteltävä tapauskohtaisesti. Tukea päätöksentekoon saa yleisstandardin SFS-EN 60601-1 yksittäisvian periaatteesta sekä jo käytössä olevien lääkintälaitteiden käyttökokemustiedoista.

ALARP-alueella riskejä voidaan pitää siedettävänä vain, kun hyödyt ovat merkittäviä ja riskinvähentäminen tästä edelleen on joko epäkäytännöllistä tai kallista verrattuna riskinvähennyksen suuruuteen. Siten riskiä ALARP-alueella ei voida pitää hyväksyttävänä vain siksi, että potilaan hoitoennuste parantuu.

Taulukko 15. Riskialueet.



6.4.3 Tuotteen riskianalyysi

Standardissa SFS-EN 1441 on ohjeita valmistajalle siitä, kuinka varmentaa, todentaa ja dokumentoida terveydenhuollon tuotteiden vaatimusten täyttyminen.

Standardi SFS-EN 60601-1-4 edellyttää, että ohjelmistolle tehdään riskianalyysi, mutta ei määrittele sen suoritustapaa vaan viittaa EN 1441:een. Siten riskianalyysi suoritetaan standardin SFS-EN 1441 pohjalta (ks. liite E).

6.4.4 Ohjelmiston riskianalyysi

Turvallisuuteen liittyvän ohjelmiston riskianalyysin suoritus ja siitä seuraavat vaatimukset tarvittaville menetelmille ohjelmiston elinkaaren eri vaiheissa on ehkä epämääräisimmin määritelty yleisissä standardeissa. Militääristandardit ovat pyrkineet määrittelemään asian tarkemmin, mutta käyttötapauksien vuoksi ne asettavat vaatimukset sen mukaan, että ohjelmistoon liittyvät vaarat ovat korkeita (FDA:n termi), tai sietämättömiä "normaalilla ohjelmistotuotantoprosessilla", ja asettavat epäkäytännöllisen tiukat vaatimukset pienempiä riskejä sisältävän ohjelmiston tuotannolle.

Standardi IEC 61508 pyrkii yleiseksi standardiksi. Se perustuu ajatukseen, että ohjelmiston toiminnon täytyy täyttää tietty turvallisuuden eheystaso, jos tehtävänä on turvallisuuden ylläpito, tai riskin pienentäminen. Eheystaso määrittää ohjelmistotuotannon elinkaaren vaiheissa tarvittavat tehtävät ja menetelmät sekä mm. testausten täydellisyysvaatimukset, joilla saavutetaan tarvittava riskinvähennys.

Ohjelmistokomponentti voidaan myös suoraan toteuttaa käyttötarkoituksensa perusteella valitulle turvallisuuden eheystasolle. Tällaisia ovat esimerkiksi korkean luotettavuuden turvallisuusjärjestelmien toteuttamiseen tarkoitetut reaaliaikaiset käyttöjärjestelmät, joita on sertifioitu IEC 60601-4 ja IEC 61508 -standardien mukaisesti. Tulevai-

suudessa valmiiksi sertifioidut komponentit tulevat varmaan helpottamaan korkeaa turvallisuutta edellyttävää ohjelmistotuotantoa.

Brittiläisen APES (Assuring Programmable Electronic Systems) -projektin tavoitteena on ohjeistaa IEC 61508 -standardin soveltamista siten, että standardia voidaan yksikäsitteisesti noudattaa eri teollisuusaloilla. Yksinkertaistettuna standardi tuo mukaan ohjelmistotuotantoprosessiin seuraavat ohjelmiston turvallisuuden elinkaaren vaiheet:

1. Vaarojen tunnistaminen koetelluilla riskianalyysimenetelmillä, kuten HAZOPilla.
2. Vaarojen seurausten arviointi.
3. Todennäköisyysluokkien määrittäminen. Ne edustavat vaarojen todennäköisyyksiä kehittyä onnettomuuksiksi.
4. Riskiluokkien määrittäminen. Seurausten ja todennäköisyyksien mukaan riskit jaetaan sietämättömiin, epätoivottaviin, jotka voidaan hyväksyä vain, jos riskin vähentäminen ei missään suhteessa vastaa tuloksia (kohtuuttomat kustannukset, ja merkittävien saavutettavien ominaisuuksien menetys jne., mikä tulee perustella hyvin), hyväksyttäviin, joissa riskin vähentämisestä saavutetut edut eivät vastaa kustannuksia, ja merkityksettömiin.
5. Turvallisuusvaatimusten määrittäminen järjestelmälle. Riskejä voidaan vähentää joko rajoittamalla seurauksia tai pienentämällä todennäköisyyksiä.
6. Jäännösriskit kohdistetaan ohjelmistolle tai sen toiminnoille, jos osat voidaan osittaa riippumattomiksi, kun ne voivat olla joko alkutekijöinä vaaran toteutumisessa tai vastuussa turvatoiminnoista. Tämä on iteratiivinen prosessi, joka jatkuu, kunnes riskit ovat ALARP-alueella tai jos sietämättömien riskien jäljelle jäämisen vuoksi hankkeesta täytyy luopua.
7. Ohjelmiston tai sen turvallisuuteen liittyvien toimintojen kuuluminen johonkin riskiluokkaan määrittää ohjelmistolta tai sen toiminnolta vaadittavan turvallisuuden eheystason.
8. Turvallisuuden eheystaso määrittää ohjelmiston määrittelylle, suunnittelulle, toteutukselle, verifioinnille ja validoinnille tulevat lisävaatimukset, joiden toteutumista varmistavien toimenpiteiden tuottamat tietueet lisätään ohjelmiston riskinhallintatiedostoon.

Edellä kuvattu vaiheistus käsittelee IEC 61508:lle tyypillisiä käsitteitä kuten mm. turvallisuuden eheystasoja ja niiden avulla tapahtuvaa riskinvähennystä. Turvallisuuden eheystasomenettely on suunnittelua tukeva tapa hoitaa riskinvähentyminen. Esimerkiksi jos lääkintälaitteen riskiin sisältyvä vaarallisen tapahtuman taajuus on esim. 10^{-1}/a ja siedettävän riskin taajuus esim. 10^{-4}/a, on eri riskin vähennyskeinojen epäonnistumisto-

dennäköisyyksien tulolla päästävä tähän tai sen alle. Tuloilla laskeminen edellyttää riippumattomuutta. Standardi IEC 61508 asettaa tiukat vaatimukset riippumattomuudelle turvallisuuden eheyttä näin kohdennettaessa. Menettely on hyvin suosittu eri teollisuusaloilla. Valitettavasti tämän julkaisun puitteisiin ei ole mahdollista sisällyttää laajempaa kuvausta IEC 61508:n menettelytavasta.

6.4.5 Riskianalyysin raportointi

Riskianalyysin suoritus raportoidaan, jolloin raportti tukee jatkuvaa riskienhallintaprosessia, ja sitä voidaan käyttää tarvittaessa osoitusdokumenttina sitä haluaville tahoille. Raportointi tulee suorittaa sellaisessa muodossa, että sitä kyetään ylläpitämään tuotteen koko elinjakson ajan. Ylläpitovaateita tukee mm. tuotteen ominaisuuksissa tapahtuvat merkittävät muutokset.

Raporttiin sisältyvät ainakin seuraavat asiat:

- tuotteen täydellinen kuvaus ja identifiointi
- raportin identifiointi
- lista yksilöidyistä mahdollisista vaaroista ja niiden alullepanevista syistä
- analyysimenetelmä
- kuhunkin vaaraan liittyvän riskin arviointi
- riskin vähimmäistämistapa, kun tarvittu käyttää
- riskin vähimmäistämisen vaikuttavuuden arviointi
- riski/etu-tarkastelu
- analyysin suorittaja
- analyysin hyväksyjä
- analyysin päivitys.

6.5 Riskien merkitysten arviointi

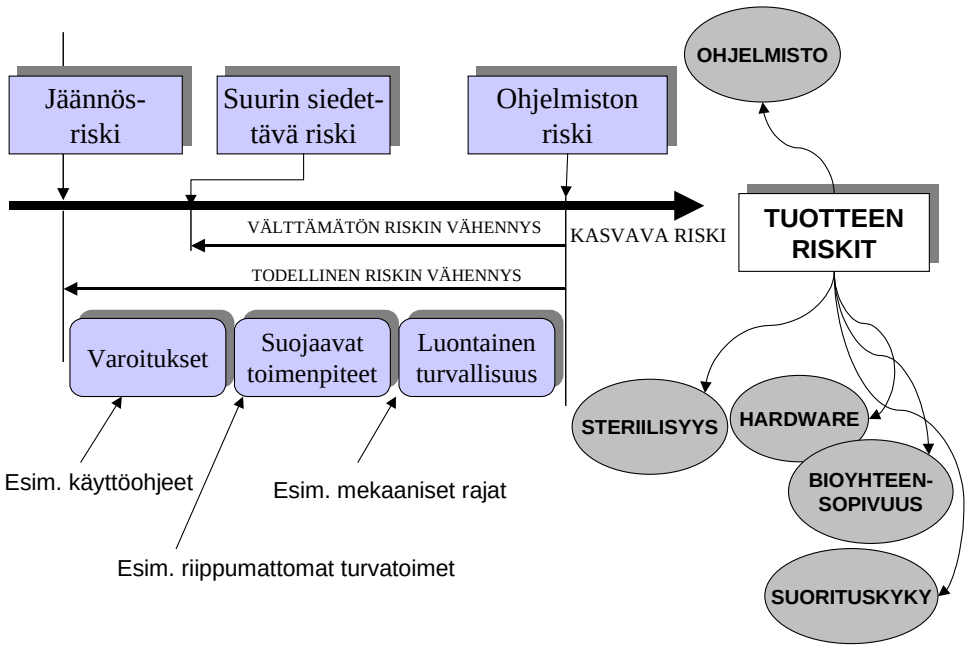
Kuten kohdassa 6.4 esitettiin, valmistajalla tulee olla riskien arvioinnin prosessi, jossa järjestelmällisesti tunnistetaan lääkintälaitteen vaarat ja määritellään vaarojen aiheuttamien riskien suuruus. Kun riskien suuruudet on selvillä, päätetään niiden merkittävyydestä sekä valitaan ja toteutetaan toimet turvallisuuden parantamiseksi, jos on päädytty pienentämään riskiä. Käsitellään tässä kohdassa riskien merkityksen arviointia ja parantavia toimenpiteitä kohdassa 6.6.

Riskien merkitys arvioidaan sekä vaaratekijöittäin että tuotekohtaisesti. Vaarakohtaisessa menettelyssä päätetään riskin siedettävyydestä ja toimenpiteisiin ryhtymisestä jokaiselle riskianalyysissa tunnistetulle vaaratekijälle. Tuotekohtainen riskien merkityksen arviointi tapahtuu, kun kaikki turvallisuutta parantavat vaarakohtaiset toimenpiteet on tehty.

Arviointiprosessissa otetaan huomioon sekä normaalikäyttö että virheellinen käyttö ja niissä määritellään myös tapauskohtaisesti, milloin riski on hyväksyttävissä. Menetelmien tulee sisältää myös vaikuttavuuden arviointi, jolla voidaan arvioida riskin vähentämisen toimenpiteiden tehokkuutta.

Standardi SFS-EN 60601-1-4 ei spesifioi hyväksyttävää riskiä, mutta sen selostavassa osassa suositellaan noudattamaan standardin SFS-EN 60601-1 'yhden vian tapaus' -periaatetta ja/tai tutkimaan jo käytössä olevien vastaavanlaisten lääkintälaitteiden suorituskykyä.

Ennen kuin tuotteen jäännösriskin hyväksyttävyydestä voidaan päättää, pitää olla tiedossa siedettävän eli ALARP-riskialueen rajat. Jos tuotteen jäännösriski on ALARP-alueella, tulee tarkastella riskin hyväksyttävyyttä saavutettuihin hyötyihin verrattuna (ks. myös kohta 6.4.2).



Kuva 15. Riskin vähennyksen yleiset periaatteet lääkintälaitteen ohjelmistolle.

Kuvassa 15 on esitetty riskin vähennyksen yleiset periaatteet sovellettuna lääkintälaitteiden ohjelmiston riskin vähennykselle. Kuvassa riskien vähentämiskeinoina ovat standardin SFS-EN 60601-1-4 mainitsemat menetelmät: luontaisen turvallisuuden suunnittelu, suojaavat toimenpiteet ja varoitukset. Niitä sovelletaan yksittäisiin vaaroihin ja vaaratapahtumiin, joiden kaikkien riskin täytyy olla hyväksyttävä. Kun ne on hyväksytty, päätetään lääkintälaitteen turvallisuuden hyväksyttävyydestä.

Kun jäännösriskiin sisältyy sellaisia vaaratapahtumia, joiden riskiä pitää laitteen operaattorin, hoitajan tai käyttäjän toimenpitein vähentää, tulee ne standardin SFS-EN 60601-1-4 ohjeiden mukaan kirjata käyttöohjeisiin ja riskienhallintakansioon (ks. esim. käyttöohjeet kuvassa 15). Jos jäännösriski ei ole merkittävä, eli se on alle pienimmän siedettävän riskin ilman käyttöohjeitakin, sitä ei tarvitse kuvata laitteen mukana seuraavissa asiakirjoissa.

Käytännössä tämä edellyttää valmistajalta riskin arvon määrittelyohjeistusta, jota voidaan aina tuotekohtaisesti soveltaa. Määrittelyssä kuvataan riskin arvot numeerisina arvoina, esimerkiksi RPN[4]-lukeman avulla.

6.6 Riskien valvonta

Kun riskin vähentäminen on riskin merkityksen arvioinnin tuloksena todettu tarpeelliseksi, aloitetaan eri ratkaisujen keskinäinen vertaaminen. Mahdollisten vaihtoehtojen käyttöönotossa kehottaa lainsäädäntö (MDD 1993) noudattamaan seuraavaa järjestystä:

- pyri poistamaan tai vähentämään riskiä mahdollisimman paljon luontaisesti turvallisella suunnittelulla tai rakenteella
- käytä riittäviä suojakeinoja niiden riskien yhteydessä, joita et voi poistaa, esim. hälytysjärjestelmät
- tiedota käyttäjää jäännösriskeistä, jotka johtuvat käytettyjen suojatoimenpiteiden vaikutuksesta.

Valitun suojatoimenpiteen käyttöönotto ja sen vaikuttavuus arvioidaan. Kaikki toimet analyysin ja riskin vähimmäistämisen aikana dokumentoidaan, jotta saadaan rakennettua kattava jäljitettävyys suunnittelun elinkaarien**,** riskianalyysin ja dokumentaation välille. Syntynyt dokumentaatio talletetaan osaksi tuotteen riskienhallintakansiota.

[4] RPN: Risk Priority Number

Jäännösriskin suuruuden merkitys tulee arvioida kohdan 6.4.2 periaatteiden mukaisesti. Suurimman siedettävän tason alittaminen voi vaatia useita vertailuja ja toimenpiteiden vaikuttavuuden arviointeja.

Jos jäännösriski kaikista toimenpiteistä huolimatta jää liian suureksi, on arvioitava painoarvoja aiotun käyttötarkoituksen, vaikuttavuuden ja ko. jäännösriskin välillä. Tässä tapauksessa on syytä käyttää kliinistä asiantuntijaa päätöksenteon tukena.

Valmistajan tulee luokitella käyttämänsä suojatoimenpiteet teknologian, soveltuvuuden ja tehokkuuden mukaan. Käytettyjä suojatoimenpiteitä voidaan määritellä esimerkiksi osaksi ohjelmoinnin tyylioppaita. Suojatoimet ohjelmistolle voivat olla seuraavat:

- alustustoimenpiteet
- virheenhallintarutiinien määrittely
- samanaikaisesti auki olevien prosessien minimointi
- prosessien priorisointi
- laitteistovarmistus kriittisten ohjelmistomoduulien suojaamisessa
- dynaamisen muistin allokoinnin välttäminen
- ohjelmiston kriittisten toimintojen kaksinkertaiset verifioinnit ja testaukset.

Suojatoimien tehokkuuden arvioimiseksi valmistajan on kerättävä tilastoa järjestelmän ja prosessien käyttäytymisestä. Tilastoja voidaan käyttää osoitusaineistona käytettyjen menetelmien sopivuudesta.

Verifioinnit, validoinnit ja testaukset (V&V&T) löytävät piileviä ohjelmistovirheitä. V&V&T-menettelyt vähentävät siis myös riskiä, vaikka niitä ei voidakaan pitää direktiivin tarkoittamina varsinaisina riskinvähennyksen keinoina. RPN-menettelyssä on kolme lukua vakavuus (severity, *S*), esiintymistiheys (occurance, *O*) ja havaittavuus (detectability, *D*), joista jälkimmäisessä seurataan myös ohjelmistovirheiden havaittavuutta koko ohjelmiston elinkaaren aikana (ks. myös kuva 10, missä APN, Action Priority Number vastaa RPN:ta). Siten testein ja analyysein ym. V&V-menettelyillä vaikutetaan *D*-luvun välityksellä RPN-lukuun. Laitteistoviat ovat satunnaisvikoja ja niiden esiintymistodennäköisyydet voidaan määrittää kvantitatiivisesti esim. vikapuumenetelmällä. Ohjelmistovirheet ovat systemaattisia virheitä, joille ei ole hyviä kvantitatiivisia määritysvälineitä.

Luvussa 5 esitetään yleisimmät V&V&T-tekniikat. Niitä ja monia muita riskien vähentämiseen sopivia menetelmiä ja tekniikoita luettelee, luokittaa ja kuvaa IEC 61508.

6.7 Tuotannon jälkeiset vaiheet

Standardiluonnoksen ISO 14971 mukaan valmistajan tulee luoda ja ylläpitää suunnitelmallista menettelytapaa tarkastaa ja koota tietoa, jotka koskevat lääkintälaitteen tuotannon jälkeistä vaihetta. Kootun tiedon turvallisuusmerkitys arvioidaan ja valmistajan tulee selvittää:

a) esiintyykö tunnistamattomia riskejä

b) voiko tunnistamatonta riskiä hyväksyä

c) onko alkuperäinen arviointi vielä voimassa .

Jos joku yllä olevista kohdista täyttyy, analyysi täytyy uusia riskienhallintaprosessin edellyttämällä tavalla.

Saadun tiedon perusteella harkitaan tarvittavat riskienhallintaprosessin toimenpiteet. Jos on mahdollista, että jäännösriski tai sen hyväksyttävyys on muuttunut, muutoksen vaikutus toteutetuille riskin valvontakeinoille arvioidaan ja arviointitulokset talletetaan riskienhallintakansioon.

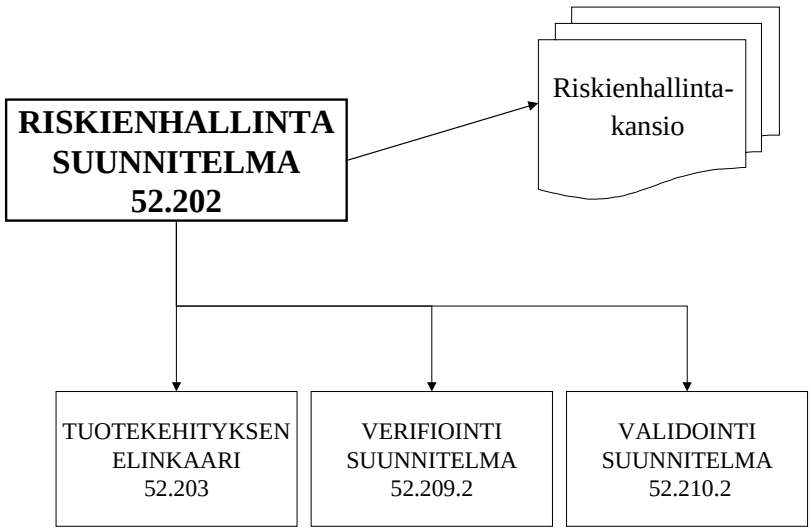
Tuotannon jälkeisten valvontavaatimusten täyttyminen ei ole ainoastaan suunnittelutiimin tehtävä, vaan osa tehtävistä lankeaa myös markkinoinnin ja laadunvarmistuksen vastuulle. Tuotannon jälkeinen valvonta onkin syytä integroida osaksi tuotteenhallintaa.

6.8 Riskienhallintasuunnitelma

Lääkintälaitteiden toimintojen suorittamisessa, valvonnassa ja laitteen valmistuksessa käytettävät teknologiat ovat monimutkaistuneet hyvin nopeasti. Monimutkaisuuden hallitsemiseksi edellytetään valmistajilta hyvin kuvattuja ja valvottuja tuotekehitysmenetelmiä, joiden osaksi on integroitu tehokas riskienhallinta.

Riskienhallinta ei ainoastaan auta varmistamaan, että lopputuote on turvallinen ja vaatimukset täyttävä, sitä voidaan myös käyttää tuotekehitysprojektin aikataulun ja kustannusten valvontaan. Jotta kaikki tämä on mahdollista, edellytetään valmistajaa käynnistämään tuotekehitysprojektin alkuvaiheessa kattava riskienhallintasuunnitelman teko.

Riskienhallintasuunnitelma on koko tuotteen kattava suunnitelma. Se liitetään osaksi tuotteen riskienhallintasuunnitelmaa. Se määrittelee projektin verifiointi- ja validointisuunnitelmat sekä kehitysprosessin ohjelmistopohjaiselle lääkintälaitteelle (kuva 16) standardin SFS-EN 60601-1-4 mukaisesti.



Kuva 16. Riskienhallintasuunnitelman osat standardissa SFS-EN 60601-1-4.

Standardia SFS-EN 60601-1-4 täsmällisemmin riskienhallintaa kuvataan standardiluonnoksessa ISO/FDIS 14971.

Eurooppalainen standardoimisjärjestö CEN on laatinut direktiivin (MDD 1993) tueksi standardin EN 1441, joka antaa puitteet terveydenhuollon tuotteen riskianalyysille. Standardi on omaksuttu myös muille markkina-alueille, joten kansainvälisen ISO-standardoimisjärjestön komitea TC210 käynnisti vastaavan standardin kehittämisen (ISO/DIS 14971) lisäten siihen esimerkiksi riskienhallintasuunnitelman laatimisen. Uusi standardi on nyt lähes lopullisessa vaiheessa (DIS), ja on oletettavaa, että standardin sisältö aikanaan hyväksytään lähes sellaisenaan ISO-standardina. Standardi todennäköisesti myös adoptoidaan EN-standardiksi.

Jotta valmistaja saa tuotettua jokaiselle projektille riittävän laadukkaan ja kattavan riskienhallintasuunnitelman siten, että se täyttää suunnitelmalle asetetut vaatimukset, on valmistajan kuvattava suunnitelmien tekotapa laatujärjestelmädokumenteissa tai muissa menetelmäohjeissa. Ohjeiden tulee sisältää myös määrämuotoiset dokumenttipohjat, joiden avulla tuotekohtaiset suunnitelmat laaditaan.

7. Vaatimustenmukaisuuden osoittaminen

Tässä luvussa käsitellään vaatimustenmukaisuuden arviointia valmistajan ja kolmannen osapuolen näkökulmasta. Kuvataan sellainen yrityksen validointimalli, jonka tulokset ovat myös riittävä osoitusdokumentaatio kolmannen osapuolen arvioinnille. Tässä tekstissä valmistajan toimenpiteitä vaatimustenmukaisuuden osoittamiseksi kutsutaan validoinniksi ja kolmannen osapuolen toimenpiteitä arvioinniksi. Validointisuunnitelman kuvaamat toimenpiteet osoittavat, että oikeat toiminnalliset ja turvallisuuteen liittyvät vaatimukset on implementoitu tuotteeseen. Ohjelmiston validoinnin merkitys koko tuotteen validoinnissa arvioidaan. Validointi eri tilanteissa: oma ohjelmisto, alihankittu tai valmisohjelma ja muutostilanteet. Yhteys laadunvarmistusmalleihin. Tarkastuslistojen käyttö.

7.1 Johdanto

Vaatimustenmukaisuuden osoittamisen käsite voidaan määritellä eri tavoin sen mukaan, otetaanko lähtökohdaksi viranomaismääräysten ja standardien esittämät vaatimukset, valmistajan omat tuotteelle asettamat vaatimukset vai määritelläänkö vaatimus prosessin laadun näkökulmasta.

Tässä tekstissä käsitellään vaatimustenmukaisuuden osoittamista valmistajan näkökulmasta validoimalla ja kolmannen osapuolen vaatimuksia eli arviointia.

Tässä yhteydessä validoinnilla (kelpuutus) tarkoitetaan valmistajan suorittamia toimintoja tuotteelle asetettujen vaatimusten täyttymisen varmistamiseksi. Vaatimusten täyttyminen osoitetaan vertaamalla asetettuja vaatimuksia ja toiveita valmiiseen tuotteeseen. Validointia voidaan suorittaa tuotekehitysprosessin lopussa tai sen eri vaiheissa, ja joskus siihen sisältyy verifiointiraporttien tarkastus. Validoinnin tarkoitus on antaa valmistajalle varmuus, että tuote täyttää sille asetetut kaikki vaatimukset ja sen tuotanto voidaan käynnistää.

Verifiointi eli todentaminen eroaa validoinnista siten, että se kohdistuu osasuoritteiden tai yksittäisten ominaisuuksien arviointiin.

Arvioinnilla taas tarkoitetaan niitä toimintoja, joita riippumaton kolmas osapuoli suorittaa tarkastaakseen valmistajan valmistaman tuotteen vaatimustenmukaisuutta kansallisiin tai kansainvälisiin vaatimuksiin nähden. Arvioinnissa arvioidaan asetettujen lähtötietojen oikeellisuus, riittävyys ja vaikuttavuus sekä toimiiko suunnittelu ja vastaako valmistunut tuote asetettuja toiveita, tarpeita ja vaatimuksia. Lisäksi arvioinnissa tutkitaan valmistajan validointiprosessin toimivuutta kansallisiin säädöksiin nähden.

7.2 Validointi

Validoinnilla varmistetaan tuotteen määriteltyjen ominaisuuksien toteutuminen. Terveydenhuollon tuotteen validointi voidaan jakaa esim. suunnittelun, tuotantoprosessien ja kliinisten tulosten validointiin. Ohjelmistolla voi olla vaikutusta näihin kaikkiin osatekijöihin. Validoinnin avulla osoitetaan, että ohjelmiston vaatimukset on oikein ja täydellisesti toteutettu ja että ne ovat jäljitettävissä tuotteen (järjestelmän, laitteen) vaatimuksiin (turvallisuus ja vaikuttavuus tai turvallisuus vastaan kliiniset hyödyt). Validointitoimenpiteillä saadaan selkeä näyttö siitä, että ohjelmisto suorittaa aiotun toimintansa oikein eikä suorita ei-toivottuja toimintoja sekä tuotetaan tietoa sen laadusta ja luotettavuudesta. Lääkintälaitteen osalta ohjelmiston validoinnilla varmistetaan, että ohjelmiston kautta ei direktiivin (MDD 1993) olennaisten vaatimusten täyttyminen ole uhattuna.

Standardin SFS-EN 60601-1-4 mukaan validointi määritellään seuraavasti:

> ***Validation:*** *Process of evaluating a PEMS or a component of a PEMS during or at the end of the development process, to determine whether it satisfies the requirements for its intended use.*

Validointi kohdistetaan itse tuotteeseen sekä sen tuotannossa tarvittaviin välineisiin ja erityisprosesseihin.

FDA:n mukaan (FDA 1997) validointi määritellään:

> ***Validation:*** *establishing by objective evidence that all software requirements have been implemented correctly and completely and are traceable to system requirements.*

Kuten määritelmistä voi huomata, validoinnin käsite näillä markkina-alueilla vaihtelee hieman; molempiin määritelmiin sopii kuitenkin validoinnin tarkoitus, eli tutkia ja analysoida kokonaisvaltaisesti, että

- ohjelmistotuotannon prosessit toimivat kuvatulla tavalla
- ohjelmistoa tuottavat välineet toimivat tarkoitetulla tavalla
- ohjelmisto suorittaa sille aiotut tehtävät oikein ja että
- ohjelmisto on turvallinen ja luotettava sekä täyttää sille asetetut asiakas-, valmistaja- ja viranomaisvaatimukset. Jos ohjelmisto on osa laitetta, niin tällöin varmistetaan, että ohjelmisto ei vaaranna laitteen/järjestelmän oikeaa toimintaa tai turvallisuutta (huom. ohjelmistovaatimusten jäljitettävyys tuotevaatimuksiin).

FDA:n vaatimukset validoinnille noudattavat edellä mainittuja periaatteita määritellen validoitavien kohteiden sisältöä tarkemmin (FDA 1997). Ohjeen mukaan validoinnin piiriin kuuluu

- johto (development plan summary, tuotteen vakavuustason ollessa merkittävä tai kohtalainen)
- vaatimukset (SRS document, kaikilla vakavuustasoilla)
- suunnittelu (SDS document, tuotteen vakavuustason ollessa merkittävä tai kohtalainen)
- implementointi (unit and integration tests, tuotteen vakavuustason ollessa merkittävä tai kohtalainen)
- integrointi ja testaus (testitulokset ja hyväksy/hylkää-kriteerit).

Jäljitettävyys vaatimusten, tunnistettujen vaarojen ja testauksen välillä osoittaa tuotteen vaatimusten toteutumisen tuotteen vakavuustason ollessa merkittävä tai kohtalainen. FDA suosittaa riskianalyysimenetelmien ja tekniikoiden käyttöä validoinnissa. Esimerkiksi SFMECA:lla voidaan SRS:sta ja SDS:sta tunnistaa ja dokumentoida ne osat, jotka liittyvät tuotteen turvallisuuteen tai toimivuuteen ja siten myös jäljittää ne eteenpäin testitapauksiin.

FDA hyväksyy (FDA 1997) myös, että ohjelmiston validoinnin laajuus ja validointiin käytettävä työmäärä ovat suhteessa laitteen aiheuttamaan riskiin. Laajaa ohjelmistoa validoitaessa korkeamman riskin omaavien moduulien ohjelmiston vaatimusspesifikaatioihin, ohjelmiston suunnitteluspesifikaatioihin ja testitapauksiin on kohdistettava perinpohjaiset ja yksityiskohtaiset tarkastukset.

Vastaava elementti EU:n lainsäädännössä saavutetaan riskianalyysin kautta. Riskianalyysin osoittaessa alhaista riskiä tarvittavien analyysien ja dokumentointien ei tarvitse olla niin täydellisiä.

Silloin kun toiminnan tuloksia ei voida varmistaa lopputuotteen testausmenetelmien avulla, suunnitellaan suunnittelu/tuotantoprosessi siten, että sopivien koeajojen (validointiajojen) avulla voidaan osoittaa tavoitteiden toteutuminen. Suunnitteluprosessi on usein kertaluonteinen tapahtuma sovelletaan näihin yleisiä toimintamalleja, kuten ISO 9001, joihin liitetään tarvittavat tapauskohtaiset lisäykset. Eräs malli lääkintälaitteiden ohjelmistokehitykselle on kuvattu standardissa SFS-EN 60601-1-4.

Seuraavissa luvuissa kuvataan validoinnin kohteita tarkemmin, jotta valmistaja voisi verrata niitä oman ohjelmistotuotantonsa prosesseihin. Validointimenettelyn käyttö edellyttää valmistajalta omien ko. prosessien tunnistamista, määrittelyä sekä ohjeistamista.

7.2.1 Validoinnin tavoite

Validointiprosessin on oltava kiinteästi osana tuotekehitystä. Validointi on tehokas laadunvarmistuskeino, jolla valmistaja voi valvoa ohjelmistotuotannon prosesseja, teknologioita ja ohjelmistokomponentteja, jotka

- valmistaja kehittää
- alihankkija kehittää
- ostetaan kaupallisina valmisohjelmina (COTS-komponentteja)
- ovat osa tuotekehityksessä käytettäviä testaus- , simulointi- ja muita tukiohjelmia.

Koska lääkintälaitteiden turvallisuus on eräs tärkeistä vaatimuksista, validoinnin avulla varmistetaan, että riskianalyysissä havaitut vaarat on muutettu vaatimusmäärittelyyn ominaisuuksiksi, joilla havaittuun vaaraan liittyvää riskiä vähimmäistetään.

Vaikka validointi nähdään usein pelkästään pakollisena toimena, joka vain ja ainoastaan hidastuttaa ja hankaloittaa yrityksen toimintoja, niin se myös auttaa luomaan ja kehittämään niitä tehostamalla korjaavia toimenpiteitä, vähentämällä tuotteen takaisinkutsuja (recall) ja pienentämällä kustannuksia pitkällä aikavälillä.

Tehokkaiden validointimenetelmien avulla myös ohjelmistomuutokset tehdään luotettavammin ja hallittavammin (on huomattava, että hyvin useat ohjelmistoprojektit ovat vanhan olemassa olevan ohjelmiston modifiointia) ja parannetaan tuotekehitystä tukevaa dokumentaatiota ja raportointia sekä varmistetaan laatutavoitteiden toteutumista**.**

Ohjelmiston käyttöön liittyvän kokemusperäisen tiedon on ohjattava myös validointitapahtumaa. Toisin sanoen kentältä saadun palautteen perusteella on tarvittaessa muutettava validoinnin kohdentuvuutta, hyväksyntärajoja tai laajuutta.

Jotta valmistaja kykenee tuottamaan toistettavasti ja kustannustehokkaasti vaatimustenmukaista tuotetta, tarkoittaa tämä käytännössä sitä, että valmistajalla on käytössään dokumentoitu toimintajärjestelmä, esimerkiksi ISO 9001 -malli. Toimiakseen järjestelmä edellyttää jatkuvaa sisäistä auditointia (SFS-EN ISO 9001: kohta 4.17). Sisäinen auditointi alkaa prosessikuvausten arvioinnilla ja kohdistuu tällöin yrityksen laadunvarmistusmenetelmien kuvaukseen ja menettelyjen käyttöönottoon. Auditoitaessa suunnittelutoimintoja tarkastetaan yrityksen

- laadunvarmistuspolitiikka
- suunnitteluhankkeen läpivientiproseduuri sekä vastuiden ja valtuuksien asettaminen
- valvontaproseduuri

- resurssointitavat
- dokumenttihallinta.

Tämän jälkeen sisäinen auditointi kohdistuu erityisesti suunnittelun toteutumisen, suunnittelukatselmusten, validointi- ja riskienhallintasuunnitelmien laadinnan, toimintojen läpiviennin ja raportoinnin arviointiin.

7.2.2 Validoinnin valmistelu

Validointitapahtumasta laaditaan aina suunnitelma ja suunnitelmassa otetaan huomioon aiottu käyttötarkoitus, olosuhteet sekä turvallisuusvaatimukset. Suunnitelman laadinta aloitetaan projektin alkuvaiheessa ja se tarkentuu projektin etenemisen mukaan.

Validoinnin tarkoituksena on osoittaa, että turvallisuus- ja toiminnallisuusvaatimukset ja käyttäjän tarpeet on toteutettu määrittelyjen ominaisuuksien mukaisesti lopullisessa tuotteessa.

Standardin (SFS-EN 60601-1-4 1999) mukaan validointisuunnitelma on aina osa tuotekohtaista riskienhallintasuunnitelmaa, johon kuuluu lisäksi myös verifiointisuunnitelma sekä kulloinkin sovellettava elinkaari. Vaatimuksen huomioiminen on tärkeää, koska se voi aiheuttaa muutoksia esimerkiksi validoinnin sisältöön, kattavuuteen, aikataulutukseen ja vaiheistukseen. Suunnitelman on osoitettava, että

- oikeat turvallisvaatimukset on toteutettu
- validointi on suoritettu suunnitelman mukaisesti
- validointitiimin johtaja on riippumaton suunnittelutiimistä
- ammatilliset suhteet validointitiimin ja suunnittelutiimin on dokumentoitu
- suunnittelutiimin jäsen ei validoi omaa suunnitteluaan.

Validointia ei voida suorittaa ilman ennalta määriteltyjä prosessikuvauksia, vaihejakomallia ja tuotekohtaisia vaatimuksia ja spesifikaatioita. Kun nämä määritelmät on tehty, voidaan validointitoiminnot kohdistaa oikeassa laajuudessaan oikeisiin kohteisiin.

Validointisuunnitelma on dokumentoitava ja suunnitelman tulee kattaa

- kohteet
- kattavuus ja soveltuvuus
- rajaukset

- vastuut, pätevyys ja aikataulutus
- hyväksyntäkriteerit
- käytetyt menetelmät
- kliiniset kokeet ja niiden tulokset
- raportointi.

Mikäli yrityksellä ei ole riittävää asiantuntemusta tai resursseja suorittaa validointia, voidaan validointi ostaa päteväksi todetuilta yrityksiltä. Ratkaisu takaa myös validoinnilta edellytettävän riippumattomuuden, mutta vaatii tiivistä yhteistyötä palvelua tarjoavan yrityksen kanssa (esim. alustavien validointisuunnitelmien teko jne.).

7.2.3 Toteutus ja kohteet

Ohjelmistotuotannon validointi on ensimmäisellä kertaa erittäin laaja toiminto, jossa tarkastava arvioija pyrkii tarkastuksen aikana arvioimaan ohjelmistotuotannon kykyä tuottaa määritellyn mukaista dokumentaatiota ja tuotetta (prosessin arviointi) ja toisaalta siinä syntynyttä tuotetta (tuotearviointi). Arvioija voi olla talon omaa henkilöstöä tai pienessä yrityksessä ulkopuolinen henkilö.

Tuotteen vaatimusten täyttymistä ei välttämättä voida kaikilta osilta todeta, jos ei tunneta tuotetta tuottavien prosessien käyttäytymistä, ominaisuuksia ja tarkkuutta. Siten kunkin tuotetta valmistavan prosessin tai tuotantomenetelmän validointi on täytynyt suorittaa ennen prosessien käynnistämistä. Tuotekohtaisessa validoinnissa voidaan viitata näihin validointiraportteihin tai validoinnin perusteella voi syntyä menetelmäkuvaus, jonka perusteella tuotetta tehdään.

Validoinnin kohteena ovat kaikki tuotteen laatuun ja turvallisuuteen vaikuttavat tekijät, kuten organisaatio, laatujärjestelmä ja projektin hallinta. Kun tuotteen laadulliset ja määrälliset ominaisuudet on tunnistettu, voidaan aloittaa validointisuunnitelman laatiminen. Validointi kohdistuu ensisijaisesti aina tuotteen validointiin, mutta myös tuotteen tekemisessä ja suunnittelussa apuna käytettäviä tukiprosesseja tullaan validoimaan. Keskeisiä kysymyksiä validoinnin valmisteluista kuvataan liitteissä F ja G.

Validointi kohdistuu myös prosessin arviointiin, jolloin siinä arvioidaan seuraavia kohteita:

- projektin hallinta ja aikataulutus
- suunnitelmat, toteutus ja analyysit

- validointitoimenpiteiden valintaan vaikuttavat kriteerit ja validointitoimenpiteiden dokumentointi
- verifiointi ja testaus sekä vaihejakomallin vaatimusten toteutuminen.

7.2.4 Erityiskohteita

7.2.4.1 Eri lähteistä tuleva ohjelmisto

Valmistajalla voi olla tuotteessaan useista eri lähteistä tulevaa ohjelmistoa, josta jokaisen vaatimustenmukaisuus tulisi valmistajan kyetä osoittamaan. Valmistajan on määriteltävä hyvin tarkkaan kunkin ohjelmiston tai ohjelmamoduulin vaikutus koko ohjelmiston turvallisuuteen ja määritellä tämän perusteella kullekin ohjelmistolle sopivat menetelmät vaatimustenmukaisuuden osoittamiseksi.

Eräs selkeä kohta voisi olla arkkitehtuurispesifikaatio, jossa määritellään kaikki tuotteen ohjelmistokomponentit, niiden valmistaja sekä komponenttien tai moduulien merkittävyys tuotteen turvallisuuden ja luotettavuuden kannalta. Tässä olisi myös eräs perustelu sille, miksi jostain moduulista riittäisi pelkkä lähdekoodi ja mitkä moduulit tarvitsevat lähdekoodin lisäksi tarkemman spesifikaation.

Vastaavien prosessien mukaisesti tulisi validoida myös kaikki valmistajan käyttämät tukiohjelmistot, joita käytetään suunnittelun ja valmistuksen tukena.

7.2.4.2 Valmistajan oma ohjelmisto

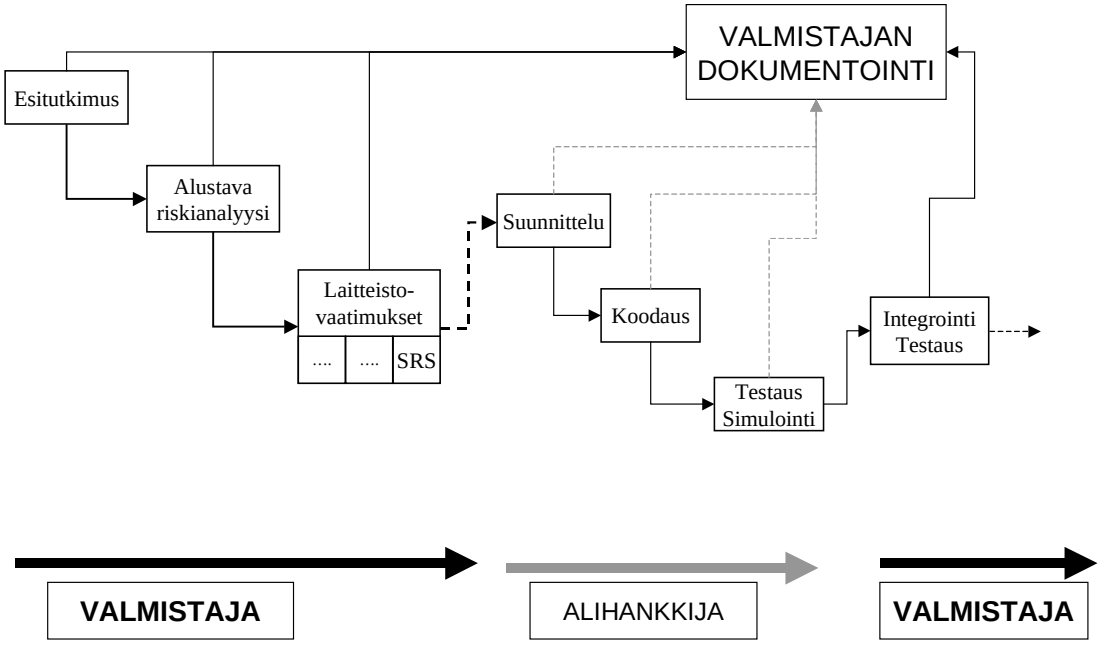
Vaatimusten asettaminen ja vaatimusten toteutumisen varmistus voidaan hoitaa dokumentoidusti ja hallitusti ohjelmistotuotannon eri vaiheiden aikana.

Dokumentoitu ohjelmistotuotannon vaihejakomalli selkeine tehtäväjakoineen on tässä tapauksessa merkitsevässä asemassa. Laajoissa ohjelmistoprojekteissa on huomioitava tiedonkulku eri suunnittelutiimien välillä, joka voidaan hoitaa esim. tarkkaan määritellyillä suunnitelmilla ja suunnittelukatselmuksilla.

7.2.4.3 Alihankinta

Osa tuotteen ohjelmistosta voidaan joko kustannussyistä, ajansäästön tai ko. ohjelmistossa vaadittavan erikoisosaamisen puuttumisen vuoksi teettää alihankkijalla.

Alihankintaohjelmiston osalta on tärkeää, että valmistaja määrittelee vaatimukset mahdollisimman tarkkaan. Käytännössä vaatimus ”täytettävä standardin SFS-EN 60601-1-4 vaatimus” on hyvin laaja, ja ilman tarkempaa määrittelyä voi alihankkijalle olla epäselvää, mitkä vaatimukset soveltuvat ko. ohjelmistolle ja mitkä eivät sovellu. Näissä tapauksissa valmistajan tehtäväksi jää osoittaa ohjelmiston vaatimustenmukaisuus. Tämä edellyttää laajoja testejä, simulointeja ja raportointia. Lisäksi on huomioitava dokumentoinnille esitettävät vaatimukset. Kaikki tämä voi olla jälkikäteen hyvinkin työlästä tai jopa mahdotonta toteuttaa.



Kuva 17. Tarkka määrittely alihankkijalle tuottaa valmiin dokumentaation.

Alihankinnan onnistumista helpottaa, kun valmistaja määrittelee koodaustavan, lähdekoodin rakenteen, työkalut, suoritettavat testit, hyväksyntärajat, raportoinnit sekä raportoinnin sisältövaatimuksen määrämuotoisena lomakkeena. Tarkan määrittelyn avulla alihankkija kykenee tuottamaan valmiin dokumentaation, joka voidaan liittää sellaisenaan osaksi valmistajan tuotekohtaista RMF:a. Kuvan 17 osoittamassa mallissa valmistajan vastuulle jää lopullisen riskianalyysin suorittaminen ja dokumentointi.

Alihankkijan on toimitettava järjestelmästä ainakin seuraavat tulokset:

- alihankintana kehitetyn ohjelmiston riskianalyysi (valmistajan vastuulle jää koko järjestelmän riskianalyysi)
- spesifikaatiot
- testisuunnitelmat ja testitulokset

- simulointi- ja kuormitustulokset sekä verifiointiraportit
- hyväksyntärajat kuormitukselle
- ohjelman kaikki tulot ja lähdöt.

Valmistajan on valvottava alihankkijan toimintaa, mikä tarkoittaa valmistajan suorittamia auditointeja alihankkijan tiloissa. Tilaisuudesta kirjoitetaan auditointiraportti, joka liitetään osaksi valmistajan omaa laatujärjestelmää. Auditointien kattavuutta voidaan rajoittaa silloin, kun alihankkijalla on kolmannen osapuolen valvoma laatujärjestelmä tai jonkinlainen ohjelmistotuotannon kypsyysmittari, esim. CMM[5]-luokitus.

Mikäli alihankkijalla ei ole laatujärjestelmää, on valmistajan ulotettava oma laadunvarmistusjärjestelmänsä koskemaan alihankkijan niitä toimintoja, joilla tuotetaan alihankittava ohjelmisto. Valmistaja valvoo näitä toimintoja esimerkiksi auditoinnein, ja auditoinnista on kirjoitettava ns. auditointiraportit. Tässä tapauksessa jää valmistajan vastuulle suurempi osa SFS-EN 60601-1-4 -vaatimuksenmukaisuuden osoittamisesta liittyen esim. ohjelmistotuotannon vaihejakomalliin, riskienhallintaprosessiin, suunnitteluun ja ohjelmiston vaatimusspesifikaatioon. Valmistajan on myös varmistettava, että alihankkija suorittaa alustavan riskianalyysin saadakseen turvallisuutta valvovat toiminnot osaksi ohjelmiston vaatimusspesifikaatiota.

7.2.4.4 Kolmannen osapuolen ohjelmisto

Kolmannen osapuolen ohjelma määritellään kaupalliseksi valmisohjelmaksi, jota käytetään osana omassa tuotteessa. Siitä käytetään useampaa eri termiä, ja käytännössä tuntuvat vakiintuneen termit Third Party Software (SFS-EN 60601-1-4 1999), Commercial-Off-The-Shelf-Component ja Off-The-Shelf-Component (FDA 1999b).

Käytännössä kolmannen osapuolen ohjelmisto voi olla laiteohjain, käyttöjärjestelmä, signaalinkäsittely-, kuvankäsittelykirjasto tai vastaava algoritmi, tietokanta tai tietoliikenneohjelmisto, käyttöliittymä tai "business-object"-komponentti tai ohjelmistokehitystyökalu.

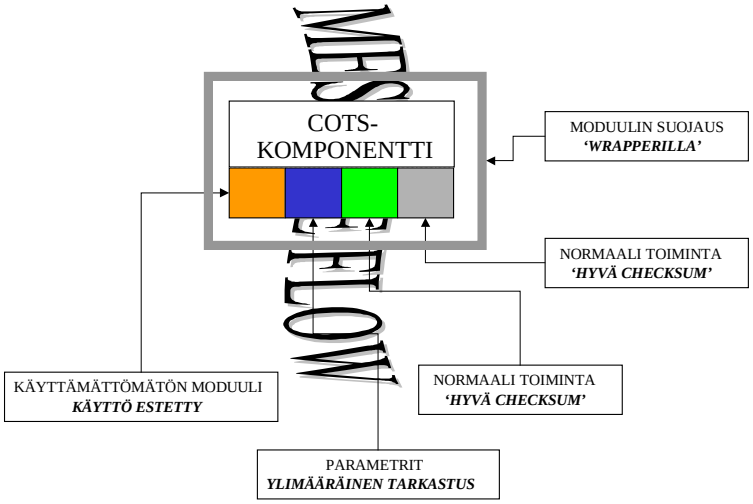
Ohjelmistot toimitetaan tavallisesti komponentteina tai kirjastoina ilman lähdekoodia. Valmistajan vastuulle jää näissä tapauksissa tutkia ohjelmien soveltuvuus suunniteltuun käyttöön. Ohjelmistoissa voi olla myös useita eri ominaisuuksia, joista käytetään vain muutamaa toimintoa. Valmistajan tulisi kyetä validoimaan käytetyt toiminnot ja ominai-

[5] Capability Maturity Model

suudet, mutta myös osoittamaan, että käyttämättömät osat eivät tee mitään asiaan kuulumatonta.

Käytännössä on mahdotonta osoittaa esimerkiksi laiteohjaimen vaatimuksenmukaisuutta ilman suunnitteluspesifikaatioita, riskianalyysejä tai testiraportteja. Valmistaja voi joko testata ostokomponentit hyvin tarkasti tai osoittaa, että järjestelmä sietää niissä esiintyvät virhetoiminnot silloin, kun ne myydään sellaisina kuin komponenttitoimittaja on ne toimittanut. Kaiken kaikkiaan turvallisuuden osoittaminen voi olla hyvin hankalaa, ja näissä tapauksissa vaatimuksenmukaisuuden osoittamiseen voidaan käyttää seuraavia tapoja:

- CERT-suosituksia
- noudatetaan puolustautuvan ohjelmoinnin periaatteita
- laajoja simulointeja, jotka kattavat kuormituksen, stabiilisuuden ja virhekäytön
- black box -testausta
- COTS-komponentin sulkeminen nk. wrapper-ohjelman sisään.



Kuva 18. COTS-komponentti ja puolustautuvan ohjelmoinnin periaatteita.

Kuvassa 18 on esimerkki puolustautuvan ohjelmoinnin periaatteista.

Eräs käyttökelpoinen tapa voi olla myös vetoaminen markkinoilta saatuihin käyttökokemuksiin. Tällöin tieto on saatettava määrämuotoisiksi tilastoiksi. Lisätietoa COTS-komponenteista löytyy luvusta 4**.**

7.2.4.5 Ohjelmiston uudelleenkäyttö

Uudelleenkäytöllä ymmärretään aiemmin toteutettujen vaatimus- ja suunnitteluspesifikaatioiden, ohjelma- ja luokkakirjastojen ja testimenetelmien ja -tulosten käyttöä uusissa ohjelmistotuotantoprojekteissa. Perinteisesti uudelleenkäyttöä on toteutettu ohjelmakirjastoilla. Olio- ja komponenttipohjaiset tekniikat laajentavat aikaisemmin kehitetyn ohjelmiston käyttömahdollisuuksia uusissa kohteissa.

On tavallista, että vanhaa ohjelmaa tai koodia käytetään uuden ohjelman perustana. Tämä on mahdollista ja monesti suositeltavaakin, koska jo käytössä olleesta ohjelmasta on näin saatu kokemuksia, joita voidaan hyödyntää uuden tuotteen suunnittelussa ja riskianalyysissä. Oikeaoppisinta olisi suunnitella oman tuotekehityksen käyttämät ohjelmakokonaisuudet alusta alkaen uudelleenkäytettäviksi.

Käytettäessä olio-ohjelmointia ja olemassa olevia luokkarakenteita on perintähierarkiassa yläpuolella olevien luokkien täytettävä myös uudelle ohjelmalle asetetut vaatimukset. Oleellista on tietenkin myös tutkia perintäominaisuuksien sopivuus uudelle sovellukselle.

Keskeiset tehtävät ohjelmiston uudelleenkäytön hyväksymiselle ovat:

- suunnitelma, jossa vertaillaan vanhan ohjelman vanhaa sovellusaluetta ja uudelleenkäytön sovellusaluetta ja käytettäviä ja lisättäviä ominaisuuksia (tärkeää miettiä esim. luokkien käyttö, perintä ja attribuutit)
- päivitetty riskianalyysi ja vaatimusspesifikaatio
- testaussuunnitelma ja tulokset
- muu tarvittava dokumentaatio.

NIST:n julkaisussa (NIST 500-234 1996) käsitellään ohjelmistoprosessin validointia, jossa on otettu yhdeksi erityiskohteeksi myös koodin uudelleenkäyttö.

7.2.5 Muutosten validointi

Muutokset tuotteen rakenteissa, ominaisuuksissa ja suorituskyvyssä ovat sen elinkaaren aikana hyvin yleisiä. Muutoksia tehdään yleensä käyttäjien toivomuksesta, kustannusten pienentämiseksi, suorituskyvyn parantamiseksi, markkina-alueilla tapahtuvienmuutosten vuoksi sekä mahdollisesti käytössä havaittujen virheiden poistamiseksi.

Lisäongelman aiheuttaa vielä muutokset kauan markkinoilla olleeseen tuotteeseen, jota ei olla hyväksytetty nykyisin voimassa olevien säädösten mukaan. Tuotteelle tulisi suorittaa nykyisten säädösten mukainen riskianalyysi, suunnittelu ja valmistus sekä tuotteesta tulisi laatia tekninen tiedosto, jonka osaksi tulisi tuottaa myös standardin SFS-EN 60601-1-4 edellyttämä RMF.

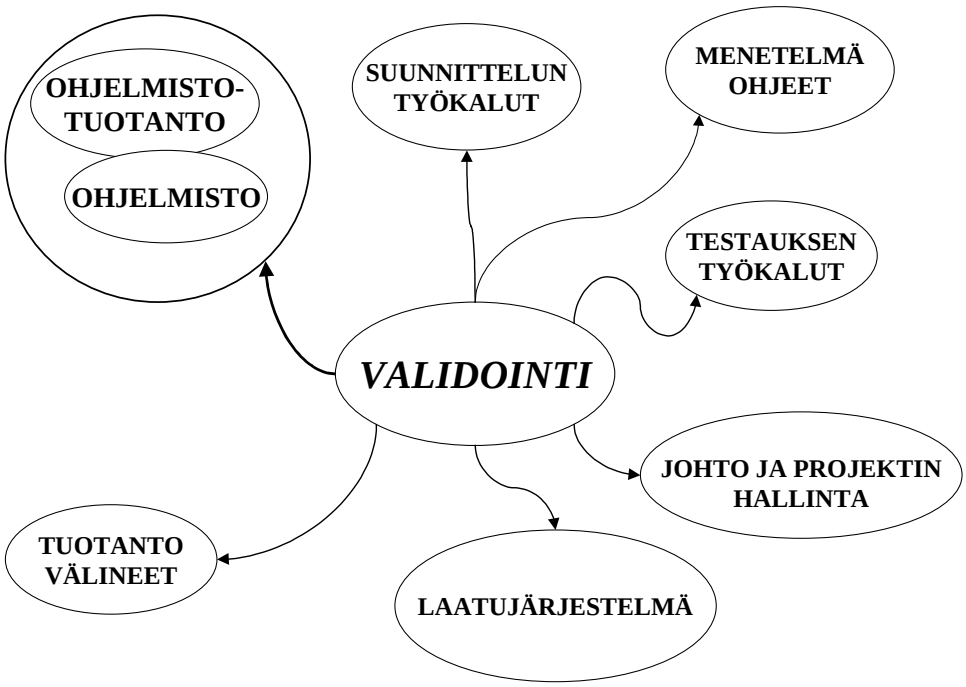
Näissä tapauksissa ohjelmistoa tulisi käsitellä kuten joko kolmannen osapuolen ohjelmistoja (kohta 7.2.4.4) tai ohjelmiston uudelleen käyttöä (kohta 7.2.4.5).

Muutosten hallinnassa on tärkeää muistaa dokumentoinnin pysyminen ajan tasalla, koska pieneltäkin tuntuva muutos voi muuttaa RMF:ssa useita dokumentteja esimerkiksi seuraavilta osin:

- muutosten kuvaus ja vaikutus vanhaan tuotteeseen nähden
- ohjelmiston uudelleenkäyttö
- päivitetty riskianalyysi ja spesifikaatiot
- muutosten ilmoitusmenettely: kenelle ilmoitetaan ja mistä ilmoitetaan
- muutosten verifiointi, validointi ja dokumentointi,

7.2.6 Validoinnin laajuus ja kohdistuvuus

Validointi kohdistuu ensisijaisesti aina tuotteeseen. Validoinnin on kuitenkin katettava kaikki ne toiminnot, joilla on vaikutusta tuotteen suunnitteluun, valmistukseen ja hallintaan. Näiden toimintojen validointi voidaan tehdä kuitenkin kertaluonteisesti, ja tuotekohtaisessa validointiraportissa voidaan viitata näihin validointeihin. Validointi uusitaan tai päivitetään aina, kun toiminto, menetelmä tai työkalu muuttuu.



Kuva 19. Esimerkki validoitavista elementeistä.

Kuvassa 19 voidaan huomata, että tuotteen vaatimusten saavuttamiseen vaikuttavat suunnittelun lisäksi myös muut valmistajan toiminnot, työkalut ja tukiprosessit.

Ohjelmiston validointi ei ole kertatapahtuma. Se alkaa esitutkimuksesta ja päättyy uuden ohjelmiston markkinoille saattamiseen. Ohjelmiston käyttöön liittyvän kokemusperäisen tiedon on ohjattava myös validointitapahtumaa. Toisin sanoen, jos käytön aikana havaitaan ongelmia turvallisuuteen, suorituskykyyn tai käytettävyyteen liittyen, tulee näiden toimintojen muuttaa tuotekehitysprosesseja.

Riskienhallinnan eräs vaatimus on kerätä tuotannon jälkeistä tietoa (vaaratilanteet, bugiraportit jne.). Tämän vaatimuksen perusteella riskienhallinta voisi antaa palautetta myös validointitapahtumaan, jolloin käytön aikana havaitut ongelmat saataisiin paremmin kerättyä validoinnin tueksi.

7.3 Arviointi

Tässä yhteydessä arvioinnilla tarkoitetaan ilmoitetun laitoksen suorittamaa arviointia, jolla todetaan tuotteen suunnittelun, valmistuksen ja lopputarkastuksen täyttävän viranomaisvaatimukset. Arvioinnin pohjana ovat kuvatut kolme EU-mallia ja FDA:n malli. Käytännössä arviointi perustuu ISO 9000 -vaatimuksiin , mutta taustalla ovat direktiivin

olennaiset vaatimukset sekä FDA:n arviointiohjeet. Tämän takia arviointi etenee hyvin syvälle suunnitteluprosessin arviointiin.

Ohjelmistotuotantoprosessia arvioidaan tuotekohtaisten dokumenttien perusteella. Arvioinnissa kiinnitetään erityistä huomiota turvallisuus- ja jäljitettävyysvaatimuksiin sekä direktiivien ja standardien vaatimuksiin. Näiden vaatimusten täyttymisen tulee olla jäljitettävissä yrityksen omaan validointiin.

Arviointilaitos tarkastaa yrityksen suorittaman validoinnin ja raportit. Tarkastuksessa arvioidaan tuotekohtaiset suunnitelmat ja niiden kattavuus ja miten niitä on noudatettu, sekä validoinnin ja raporttien suhde vaatimuksiin. Arvioija hakee kokonaisvaltaista käsitystä vaatimusten ja toteutusten välisistä riippuvuuksista ja niiden hallinnasta.

ISO 9000 -pohjainen laatujärjestelmäarviointi ei yleensä kata validointia niin syvällisesti kuin mitä kansalliset säädökset lääkintälaitteilta vaativat.

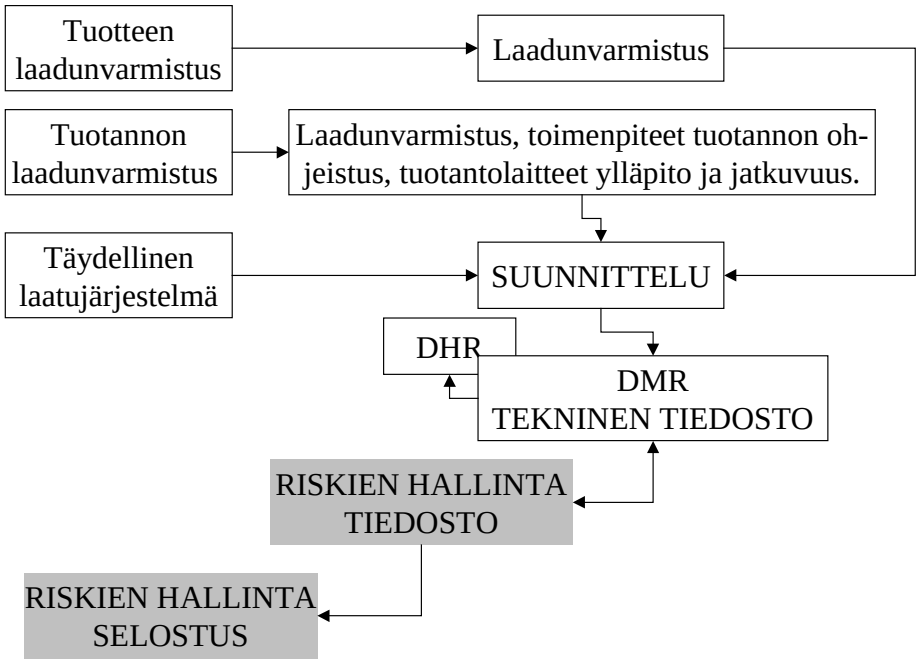
7.3.1 Tuoteluokan vaikutus arvioinnin laajuuteen

Eurooppalaisessa lainsäädännössä terveydenhuollon tuotteet jaetaan neljään tuoteluokkaan I, IIa, IIb ja III (MDD 1993) sen perusteella, millaisia potentiaalisia vaaroja ne aiheuttavat potilaalle tai käyttäjälle. Luokittelun lähtökohtana on valmistajan tuotekuvaus (so. aiottu käyttötarkoitus rajauksineen) ja säädösten antamat luokittelusäännöt, jotka perustuvat ihmisen haavoittuvuuteen.

Vaatimuksenmukaisuuden arviointia varten on modulaarinen järjestelmä, josta tuoteluokan perusteella valmistajalle on annettu tietty vapaus valita tuotteensa vaatimusten täyttymisen varmennustapa. Direktiivissä (MDD 1993) annettujen rajoitusten puitteissa valmistaja voi valita sekä tuotteen suunnittelu- että valmistusvaiheelle soveliaimmiksi katsomansa moduulit. Käytetyt reitit vaikuttavat suunnitteluun (ks. kuva 20). Terveydenhuollon tuotteille on käytettävissä seuraavat reitit:

- itsesertifiointi
- tuotteen laadunvarmistus
- tuotannon laadunvarmistus
- tuotteen tarkastaminen
- tyyppitarkastus
- täydellinen laatujärjestelmä.

HUOM: Reitti koostuu yhdestä tai useammasta liitteestä



Kuva 20. Kaikki reitit johtavat suunnitteluun.

Vaatimustenmukaisuuden arviointimenettelyn olennainen tarkoitus on varmistaa, että . markkinoitavat tuotteet ovat direktiivin (MDD 1993) säännöksissä esitettyjen vaatimusten mukaisia.

Ohjelmiston ollessa kyseessä vaatimuksia kohdistuu aina suunnitteluun, tuotantoon ja itse tuotetta koskevaan tekniseen tiedostoon. Teknisen tiedoston rakennetta ja sisältöä käsitellään tarkemmin liitteessä G.

Tyyppitarkastusreitin osalta ohjelmiston arviointi päätyy käytännössä ohjelmistotuotannon tuottamiin dokumentteihin eli täydellisen laatujärjestelmän arviointiin.

Arviointitoiminta kohdistuu alemman riskiluokan tuotteilla vain tuotteen lopputarkastukseen ja/tai sen tuotantoon. Korkeamman riskiluokan tuotteille myös suunnittelun onnistumista on arvioitava. Tuoteluokittelun vaikutus varmennusmenettelytavan valintaan on kuvattu tarkemmin direktiivin (MDD 1993) artikla 9:ssä.

Lisätietoa tuoteluokan vaikutuksesta on esitetty liitteessä H.

EU-alueella direktiivin (MDD 1993) mukaan luokiteltujen tuotteiden arviointi viranomaisen toimesta kohdistuu ohjelmiston osalta seuraaviin reitteihin:

- täydellinen laatujärjestelmä
- tuotannon laadunvarmistus
- tuotteen laadunvarmistus.

USA-alueella tarkastus perustuu hakemukseen ja sen mukana toimitetun tuotedokumentaation ja suunnitteludokumentaation tarkastukseen. Tarkastus vastaa dokumentointivaatimukseltaan EU-alueen täydellisen laatujärjestelmän arviointia sillä erotuksella, että dokumentoinnin määrään ja kattavuuteen vaikuttaa tuotteen vakavuustaso (Level Of Concern, LOC).

FDA käyttää termiä LOC (FDA 1998b) arvioimaan sen vahingon vakavuutta, jonka laite voi sallia tai aiheuttaa (suoraan tai epäsuorasti) potilaalle tai käyttäjälle piilevän vian seurauksena, suunnitteluvirheenä tai käyttämällä lääkintälaitteen ohjelmistoa. Markkinointilupaprosessin laajuus riippuu LOC:n arvosta.

Kolmannen osapuolen suorittama arviointi kohdistuu valmistajan ohjelmistotuotantoon, tuotekohtaiseen riskienhallintatiedostoon, sen osana olevaan riskienhallintaselostukseen, validointiraportteihin ohjeistuksiin sekä yleisiin laatujärjestelmäelementteihin.

VTT Automaation (ilmoitettu laitos 0537) linjaus arvioitaessa lääkintälaitteen ohjelmistoa on seuraava:

- sovellettaessa direktiivin (MDD 1993) liitettä II on valmistajalla oltava näyttö SFS-EN 60601-1-4 -standardin (tai vastaavan tasoisten) vaatimusten täyttämisestä
- sovellettaessa direktiivin (MDD 1993) liitteitä V ja VI on valmistajalla oltava näyttö ohjelmiston validoinnista (= osa lopputarkastusta)
- edellä mainittu koskee myös ohjelmiston olennaisia muutoksia.

7.3.2 Täydellinen laatujärjestelmä

Suunnittelun ja tuotannon varmistukseen perustuvassa laatujärjestelmässä keskitytään yleisesti kaikkiin suunnitteluelementteihin ja suunnittelun kykyyn tuottaa vaatimustenmukaista ohjelmistoa. Arvioinnin lähtökohtana ovat valmistajan omat suunnitteluprosessit ja niitä tukevat menetelmä-, tuotanto- ja riskienhallintaohjeistukset. Arviointi kattaa tässä tapauksessa kaiken suunnittelusta tuotantoon sisältäen suunnitteluprosessin tukiprosessit.

Arvioinnissa keskitytään standardin SFS-EN 60601-1-4 vaatimusten lisäksi myös kokonaisvaltaisesti suunnitteluprojektin kulkuun ja suoritukseen sekä sitä tukevien tukipro-

sessien arviointiin. Lisäksi erityisesti arvioidaan suunnitteluprosessia, riskienhallinta-prosessia sekä tuotekehityksen elinkaarta.

Suunnitteluprosessissa arvioidaan prosessin kykyä ratkoa suunnittelun aikana esiintyviä ongelmia, prosessikuvauksia, katselmuskäytäntöjä, dokumentointia ja suunnitelmia sekä suunnitelmien noudattamista.

Suunnittelusta arvioidaan myös projektin sisällöllinen kehittyminen ja jäljitettävyys suunnittelun eri vaiheiden aikana. Kehittyminen arvioidaan dokumentaation perusteella suunnittelukatselmus pöytäkirjoista, dokumenttien muutoshistoriasta, päiväyksistä sekä ristiviittauksista.

Riskienhallintaprosessissa arvioidaan prosessikuvauksia, suunnitelmien sisältöä, tuote-kohtaisten riskien määrittelyä, luokittelua, päätöksen tekoa sekä standardin (SFS-EN 60601-1-4 1999) edellyttämän riskienhallintatiedoston sekä riskienhallintaselostuksen sisältöä.

Tuotekehityksen elinkaaressa arvioidaan projektille sovellettua elinkaarta, prosessiku-vauksia, vaihedokumenttien sisältöä sekä verifiointeja ja validointeja sekä niitä tukevia menetelmäohjeita.

Ilmoitettu laitos arvioi täydellisen laadunvarmistuksen SFS-EN ISO 9001 -mallin vaa-timusten mukaan. Ohjelmiston osalta silloin arvioidaan tapauskohtaisesti taulukon 16 mukaisia kohtia standardin SFS-EN 60601-1-4 vaatimusten pohjalta.

Taulukko 16. Standardin SFS-EN 60601-1-4 vaatimustenmukaisuuden arviointi.

MITÄ ARVIOIDAAN	ARVIOINNIN KOHDE	TARKASTETTAVAT DOKUMENTIT TAI TOIMINNOT
Sovellusalue, käyttöympäristö, luokitus, ohjelmistolla ohjatut toiminnot. Mitä kohtia standardista SFS-EN 60601-1-4 dokumentit kattavat. Dokumentoinnin muutoshistoria sekä jäljitettävyys	Tutustu tuotteeseen 6.8, 52.201	Käyttöohjeistus, riskienhallintatiedosto, Arkkitehtuuri
Suunnitelmien kattavuus, missä elinkaarenvaiheissa tapahtuu mitäkin?	Arvioi suunnitelmat 52.202, 52.209, 52.210	Riskienhallintatiedosto ja sen osana olevat suunnitelmat
Tehtäväjako, suunnittelun tulos- ja lähtötiedot, dokumentointi, arviointi vaiheistuksen onnistumisesta (päiväykset). Suunnitteluprojektin hallinta (katselmuskäytännöt) ja asetettujen tavoitteiden täyttyminen	Elinkaari 52.203	Riskienhallintatiedosto Yleiset laatutiedostot
Riskienhallinnan kattavuus, suunnitelman ja analyysien vertailu, käytetyt menetelmät, riskien hyväksyntärajat ja menetelmien vaikuttavuuden arviointi, raportit. Riskienhallintaprosessin toimivuus, riskien hallinta ja riskien pienentäminen	Riskienhallintaprosessi 52.204	Riskienhallintatiedosto, riskienhallintaselostus, yleiset laatutiedostot, jäännösriskien kuvaus, mukana seuraavat asiakirjat
Teknologiaosaaminen, suunnittelukokemus, sovellusalueosaaminen, resurssienhallinta ja henkilöstön pätevyys	Henkilöstö 52.205	Yleiset laatutiedostot, koulutusrekisteri, projektidokumentaatio
Järjestelmän rakenne, toiminnot joihin liittyy riski, toimintojen turvallisuuseheys, arkkitehtuuri jokaisesta järjestelmän osasta, riskienhallintatoimenpiteet alijärjestelmille ja niiden komponenteille, suunnitteluspesifikaatiot ja testispesifikaatiot, kattavuus. Riskienhallinnan vaikutusta ohjelmiston vaatimusspesifikaatioon	Tekninen toteutus 52.206, 52.207, 53.208	Spesifikaatiot, arkkitehtuurisuunnitelmat, riskenhallintadokumentaatio
Menetelmät, tulokset, hyväksyntärajat	Verifiointi 52.209	Yleiset laatutiedostot, testisuunnitelmat, suunnittelukatselmukset, riskienhallintaselostus, riskienhallintatiedosto
Validoinnin suoritusta, raportointia ja riippumattomuutta sisältäen menetelmät, tulokset, hyväksyntärajat, riippumattomuuden, toteutuksen, suunnitelmien vastaavuuden ja henkilöstön	Validointi 52.210	Yleiset laatutiedostot, testisuunnitelmat, riskienhallintaselostus, riskienhallintatiedosto
Muutoshistoria, muutosten arviointi, muutosten kuvaus ja vaikuttavuus	Muutokset 52.211	Riskienhallintatiedosto, spesifikaatiot, raportit
Sisäiset auditointipöytäkirjat, validointiraportit	Arviointi 52.212	Riskienhallintatiedosto, raportit
Lausunnon soveltuvuus, standardit, mallit, jne.	Raportti vaatimustenmukaisuudesta	Lausunnon kirjoitus

Arviointia helpottaa kuvaus dokumentoinnin rakenteesta, jota tarvittaessa voidaan selvittää ns. dokumenttikartalla (ks. myös liite B).

7.3.3 Tuotannon laadunvarmistus

Tuotannon laadunvarmistuksen tehtävänä on huolehtia, että tuotantoketju toimii tehokkaasti ja luotettavasti ja tuotannon aikaiset varmistustoimenpiteet varmentavat, että tuotteet ovat suunnitellun mukaisia.

Tuotannon laadunvarmistusreitistä arvioidaan laadunvarmistustoimenpiteet, tuotannon ohjeistus, tuotantolaitteet, ylläpito ja jatkuvuus. Koska hankkeessa keskitytään ohjelmiston turvallisuuden osoittamiseen, arvioidaan edellä mainittujen toimintojen vaikutusta ainoastaan ohjelmistolle.

Ohjelmiston osalta se tarkoittaa lähinnä tuotettavan median tuotantoa ja tarkastusta. Media voi olla CD-ROM, levyke, nauha tai ROM-piiri, joka soveltuu myös sulautetun ohjelmiston mediaksi.

Tuotannon laadunvarmistuksen toimenpiteillä, mitkä automatisoidussa tuotannossa tarkoittavat poltto-, kopiointilaitteen tai siirtotien virheentarkastuksia, varmistetaan suunnitellun ohjelmiston luotettava siirtyminen asennusmediaan. Tuotantovälineen on lisättävä joko päiväystieto tai versionumero tuotteeseen (media), jolla tuote voidaan tunnistaa. Tunnistusta tarvitaan tuotteen hyväksyntäprosessissa, ns. bugikorjauksissa, muissa muutoksissa ja mahdollisissa tuotteen markkinoilta poisvetotilanteissa.

Valmistajan tulisi harkita myös median varustamista sarjanumerolla, koska joissain tapauksissa tämä saattaa olla merkityksellistä median paikantamisessa kentältä. Asennusohjelman tulisi varmistaa ohjelmiston virheetön asentuminen.

Ilmoitettu laitos arvioi tuotannon laadunvarmistuksen SFS-EN ISO 9002 -mallin mukaan. Ohjelmiston osalta silloin arvioidaan tapauskohtaisesti taulukon 17 mukaisia kohtia. Tuotannon laadunvarmistuksessa voidaan epäilyttävissä tapauksissa arvioida myös osia suunnittelusta, koska suunnittelun tehtävänä on tuottaa lopputarkastusohjeet.

Taulukko 17. Arviointikohteet tuotannon laadunvarmistuksessa.

MITÄ ARVIOIDAAN	ARVIOINNIN KOHDE	TARKASTETTAVAT KOHTEET TAI DOKUMENTIT
Tuotannon varmistus ja toistettavuus Kalibroinnit Soveltuvuus Tuotantovälineet Suorituskyky Virheen tarkastus Tuotteen tunnistettavuus Jäljitettävyys (päiväys, sarjanumerointi)	Tuotanto	Tuotantovälineiden tekniset manuaalit Tuotantovälineiden validointiraportit, kalibrointitodistukset sekä huoltotiedot Tuotannon laadunvarmistusohjeet Tuotteen versionhallinta
Tuotannon soveltuvuus Tuotantokriteerit	Tuotanto & Tuote	Asennustestausta ohjaavat menetelmäohjeet sekä asennustestausraportit. Versionhallintadokumentaatio

7.3.4 Tuotteen laadunvarmistus

Tuotteen lopputarkastuksen tarkoituksena on varmistaa, että valmistunut tuote on sille asetettujen vaatimusten mukainen. Varmistus toteutetaan riittävällä määrällä testauksia ja tarkastuksia.

Arviointi kohdistetaan lopputarkastuksen aikana tapahtuviin toimintoihin ja niitä ohjaaviin menetelmäohjeisiin sekä tuotteen asennustestaukseen, jolla varmistetaan tuotteen virheetön asentuminen käyttöympäristöön.

Ilmoitettu laitos arvioi tuotteen laadunvarmistuksen SFS-EN ISO 9003 -mallin vaatimusten mukaan. Ohjelmiston osalta silloin arvioidaan tapauskohtaisesti taulukon 18 mukaisia kohtia.

Kaikki lopputarkastuksen aikana syntyneet raportit ja mittaustulokset on talletettava osaksi tuotteen tuotantohistoriaa [DHR].

Tuotteen laadunvarmistuksessa voidaan epäilyttävissä tapauksissa arvioida myös osia suunnittelusta, koska suunnittelun tehtävänä on tuottaa lopputarkastusohjeet.

Taulukko 18. Arviointikohteet tuotteen laadunvarmistuksessa.

MITÄ ARVIOIDAAN	ARVIOINNIN KOHDE	TARKASTETTAVAT KOHTEET, TOIMINNOT TAI DOKUMENTIT
Täyttääkö tuote asetetut vaatimukset? Toimiiko tuote aiotulla tavalla?	Tuote	Validointiraportit
Asentuuko tuote ympäristöönsä? Ohjelmistoversio ja kokoonpano (moduulit, kieli-versiot, rinnakkaismallit, lisäoptiot) Versionhallinta	Asennus	Asennustestausta ohjaavat menetelmäohjeet sekä asennustestausraportit. Versionhallintadokumentaatio
Sopiiko tuote ympäristöönsä? Ympäristön identifiointi (käyttöjärjestelmä, versionumero, päivitykset, Hardiksen kuvaus)	Ympäristö Järjestelmä	Virustarkistusraportti (jos sovellettavissa). Ympäristön stabiilisuustestausraportti tai kuormitusmittaus (jos sovellettavissa) Asennustestausraportin kohta: järjestelmävaatimukset ja järjestelmäidentifikaatio
Tuotteella tarkoitetaan tässä tapauksessa asennusmediaa. Tuotteen tunnistus	Tuotteen muut ominaisuudet	Tuotteen pakkaukseen ja käsittelyyn (mekaaninen) liittyvät toiminnot ja sitä tukevat menetelmäohjeet. Staattiseen suojaukseen liittyvät toimenpiteet ja ohjeistukset. Ohjelmiston pakkauksessa arvioidaan, että pakkaus on virheetön ja pakkauksen sisältö on määritellyn mukainen. Arvioinnissa haetaan esim. jonkinlaisen tarkastuslistan käyttöä, jossa määritellään tarvittavat asennusohjeet, käyttöohjeet, median tarkastus ja pakkauksen kunnon tarkastus. Asennustestausraportti SW-version tarkastamiseksi

7.3.5 Itsearviointi

Tuotteen itsearviointi perustuu valmistajan ilmoitukseen siitä, että tuote täyttää direktiivin (MDD 1993) asettamat vaatimukset. Vaatimuksenmukaisuusvakuutusta voidaan käyttää ainoastaan luokan I laitteilla. Näissä tapauksissa ohjelmiston arviointia ulkoisen tahon toimesta ei suoriteta.

Kun tuotteen vaatimustenmukaisuus asetetaan kyseenalaiseksi markkinavalvonnan, käyttäjän tai kilpailijan toimesta, voi viranomainen suorittaa tai suorituttaa arviointitoimenpiteen ohjelmiston vaatimustenmukaisuuden toteamiseksi. Näissä tapauksissa vaatimustenmukaisuus arvioidaan aina suunnitteludokumenttien perusteella (ks. täydellinen laatujärjestelmä).

7.4 Lisätietoa validoinnista

Validointia ja arviointia koskevat luvut käsittelevät vaatimuksia direktiivin (MDD 1993) näkökulmasta, joten kullekin elinkaarivaiheelle soveltuvia validointitehtäviä tai ohjelmistospesifisiä ongelmia eivät validointi- tai arviointiluvut riitä kaikilta osilta ratkomaan.

Taulukkoon 19 on kerätty joitain hyödyllisiä lisälähteitä, jotka käsittelevät lääkintälaitteiden ohjelmistotuotantoprosessin tai ohjelmiston validointia

Taulukko *19. Lisätietoa ohjelmiston validointiin ja elinkaarimalliin.*

Kohde	Selitys	Linkki
Lääkintälaitteiden validointiprosessista ja ohjelmisto-vaatimuksista	General Principles Of Software Validation	http://www.fda.gov/cdrh/comp/swareval.html
	Guidance for the Content of Pre-market Submissions for Software Contained in Medical Devices	http://www.fda.gov/cdrh/ode/57.html
	Off-The-Shelf Software Use in Medical Devices	http://www.fda.gov/cdrh/ode/1252.html
	Design Control Guidance For Medical Device Manufacturers	http://www.fda.gov/cdrh/comp/designgd.html
	Reference Information for the Software Verification and Validation Process NIST Special Publication 500-234	http://hissa.ncsl.nist.gov/HHRFdata/Artifacts/ITLdoc/234/val-proc.html
Elinkaarimallista ja validointiprosessista	AAMI Medical Device Software Standard 15-Jan-99 Draft, Medical Device Software -- Software Life Cycle Processes	
	IEEE 1074:1997 Standard for developing Software Life Cycle Processes (Software)	
	ISO/IEC 12207:1995 Information technology - Software life cycle processes.	
	IEEE 1012:1998 Standard for Software Verification and Validation	

8. Yhteenveto

Julkaisussa esitetään vaatimustenmukaisuuden osoittamismalli, jossa yhdistetään EU:n ja FDA:n asettamat vaatimukset lääkintälaitteiden ohjelmistoille. Mallissa pyritään tehokkaaseen ohjelmiston turvallisuuden ja suorituskyvyn validointiin.

EU:n lääkintälaitteiden ohjelmistoa koskeva keskeinen standardi SFS-EN 60601-1-4 (1996, 1999) on yhdenmukainen FDA:n ohjelmisto-ohjeen (FDA 1998b) kanssa. Kummatkin asiakirjat ovat tätä kirjoitettaessa vielä luonnosteluvaiheessa ja on oletettavaa, että uudet versiot vastaavat toisiaan tärkeimmissä yksityiskohdissa entistä enemmän.

Lääkintälaite on ohjelmistoltaan, elektroniikaltaan ja mekaniikaltaan varsin monimutkainen, niin suunnittelun kuin valmistuksen osalta suurta ammattitaitoa ja osaamista vaativa nk. high tech -tuote. Useimmissa valmistajayrityksissä siten laitteen suunnittelu ja implementointi jakaantuu myös organisatorisesti ja projektirakenteellisesti eri teknologia-alueiden kesken.

Lääkintälaitteella yleensä mitataan suureita potilaasta. Mittaus on joko jatkuvaluonteista, ja mitata voidaan joko yhtä tai useampaa suuretta samanaikaisesti. Mittaustulosta käytetään tyypillisesti joko potilaan tutkimukseen tai tilan seurantaan. Lääketieteelliset kuvauslaitteet puolestaan toimivat suurimmaksi osaksi kuva eli mittaus kerrallaan. Turvallisimmissa mittauksissa kohteena oleva suure voidaan mitata kehoa häiritsemättä tai rasittamatta. Moni mittaus on kuitenkin mahdollista vain joko laitteen itse kehoon kohdistaman, esimerkiksi röntgensäteilyn avulla tai muun järjestelyn, esimerkiksi varjoaineen injektoinnin, avulla. On kuitenkin ilmeistä, että ohjelmistoa sisältäviä lääkintälaitteita jo lähitulevaisuudessa käytetään mitä moninaisimpiin tarkoituksiin, ja toimivat siten yleisesti sekä mittaus- ja/tai toimilaitteina potilaaseen nähden. Lääkintälaite sisältää yhä enemmän tutkimus- ja hoitotilannetta tukevia toimintoja, lähtien esimerkiksi hälytysrajojen asettamisista ja hälytysten hyväksikäytöstä monipuolisiin tietokantaoperaatioihin.

MDD ja FDA vaativat, että ohjelmistotuotantoprosessi on jaettu vaiheisiin, ja edellyttävät dokumentteja, joiden sisältö implikoi tietynlaisen vaihejaon olemassaoloa. Vesiputousmalli, jossa tuotanto etenee hyvin määritellyissä peräkkäisissä vaiheissa vaatimusten määrittelystä testaukseen, sopii periaatteessa lääkintälaitteen tuotantoprosessin kuvaukseksi. Vesiputousmalli edustaa ennen kaikkea hallinnollista näkökulmaa ohjelmistotuotantoon, vaiheet etenevät aikataulussa suunnitelluilla resursseilla, ja tuottavat edellisessä vaiheessa määritellyt tulokset, jotka vaiheiden päättyessä verifioidaan.

Koko ohjelmiston elinkaaren aikana kehitetään ja toteutetaan ohjelmistotuotannon rinnalla erilaisia suunnitelmia. Verifiointi- ja validointisuunnitelmassa määritellään toi-

menpiteet, joilla varmistetaan edeltävän vaiheen määritysten toteutuminen vaiheen tuottamien artefaktien perusteella (verifiointi), ja toimenpiteet, joilla varmistetaan, että valmis ohjelmisto toimii, kuten vaatimusmäärittely edellyttää (validointi). Konfiguraation hallintasuunnitelma määrittää konfiguraation hallinnan alaiset artefaktit ja menetelmät, joilla niiden eheys turvataan läpi ohjelmistotuotannon elinkaaren. Ohjelmiston riskienhallintasuunnitelmassa varaudutaan ohjelmistoon mahdollisesti liittyvien vaaratekijöiden tunnistamiseen ja eliminoimiseen.

Ohjelmiston laadunvarmistussuunnitelmassa kuvataan toimet, joilla varmistetaan, että havaitut puutteet ja virheet korjataan, samoin kuin tunnistetaan poikkeamiset standardeista, muista ohjeista ja suunnitelmista. Esimerkiksi ISO 9000 -standardi asettaa katselmuskäytännöille kovat vaatimukset, mm. laadunvalvonnan ammattitaito on oltava korkealla tasolla jotta kyettäisiin verifioimaan vaihetuotteen täydellisyys, oikeellisuus ja ristiriidattomuus.

Ohjelmistokehityksen laatutavoitteiden täyttymisen osoittamiseksi on kehitetty suuri joukko mittoja. Kuitenkin viranomaisattribuutit (turvallisuus, käytettävyys ja suorituskyky) muuntuvat vain vaivoin laatukriteereiksi ja mitattaviksi suureiksi. Puhutaan kokonaisen mittausohjelman pystyttämisestä, missä mitat voidaan validoida ja mittausprosessi suorittaa helposti. Mittausohjelmassa tulisi olla vertailuasteikko, joka kertoo, onko tulos hyvä vai huono, mikä edellyttää sitä, että huonon tai hyvän suorituskyvyn syyt tunnetaan ja niihin voidaan puuttua.

Mittatyökaluja on saatavilla verkosta. Vaikeutena ei ole mittojen puute vaan niiden valinta. Osa mitoista on epäinformatiivisia yksinään, osa ei täytä mitan määritelmää. Osalla mitoista on käänteinen riippuvuussuhde: prosessin optimoiminen yhden mitan suhteen tuottaa huonoja tuloksia toisen suhteen.

Luotettavuusmitat pyrkivät mittaamaan ja ennustamaan virhetoiminnan todennäköisyyttä tietyllä aikavälillä. Nämä mitat yleensä laaditaan kehitettäessä sovellukselle sopivia luotettavuusmalleja. Niiden tuottamia ennusteita ja arvioita voidaan siis pitää myös mittoina.

Riskienhallinta on keskeinen käsite sekä EU:n että FDA:n standardeissa ja ohjeissa. Riskienhallinnassa päätetään hyväksyttävän riskin tasoista ja menetelmistä, joilla asetetut tasot saavutetaan. Riskienhallintaprosessi on kokonaisvaltainen suunnittelun tukiprosessi, mikä liittää yhteen monia eri elementtejä vaarojen alustavasta tunnistamisesta riskien siedettävyyden arviointiin ja mahdollisten riskiä pienentävien ratkaisujen tunnistamiseen sekä tarkoituksenmukaisten valvonta- ja parannustoimenpiteiden valintaan, toteuttamiseen ja seurantaan. Prosessi vaiheistetaan koko tuotteen elinkaarelle siten, että

kunkin elinkaaren vaiheen jälkeen on arvioitava, miten vaiheen aikana tehdyt ratkaisut ja toimenpiteet vaikuttavat laitteen turvallisuuteen.

Riskianalyysin yksittäisistä tekniikoista tärkeimmät ovat vikapuuanalyysi (FTA) sekä vika- ja vaikutusanalyysi (FMEA). FMEA:lla kyetään tunnistamaan mahdolliset vaarat ja vikatapahtumat varhaisessa vaiheessa laitteen ja sen ohjelmiston kehitysprosessia. Määrittelyvaiheessa tarkastellaan jommallakummalla tekniikalla vaatimusten oikeellisuutta ja täydellisyyttä sekä suunnittelu- ja toteutusvaiheissa vaatimusten virheetöntä täyttymistä ja sitä, ettei uusia vaaroja ole ilmaantunut. Tekniikat soveltuvat elektroniikan ja mekaniikan lisäksi myös ohjelmistolle, jonka virhetilanteiden havaittavuutta ja suhdetta riskiin kyetään seuraamaan kehitysprosessin aikana.

Vaatimustenmukaisuuden osoittamismalli jakaantuu tässä julkaisussa kahteen osaan: valmistajan validointiin ja kolmannen osapuolen arviointiin. Validoinnin tavoitteena on varmistua siitä, että tuote täyttää sille asetetut kaikki vaatimukset. Arvioinnilla tarkoitetaan kaikkia niitä toimintoja, joita riippumaton osapuoli suorittaa tarkistaakseen valmistajan valmistaman ja validoiman tuotteen vaatimustenmukaisuuden kansallisiin ja kansainvälisiin vaatimuksiin nähden.

Jäljitettävyys vaatimusten, tunnistettujen vaarojen, ja testauksen välillä osoittaa tuotteen vaatimusten toteutumisen tuotteen vakavuustason ollessa merkittävä tai kohtalainen. Riskianalyysitekniikoilla, esimerkiksi ohjelmiston FMECA:lla voidaan ohjelmiston vaatimusmäärittelystä ja suunnitteludokumentaatiosta tunnistaa ja dokumentoida ne osat, jotka liittyvät tuotteen turvallisuuteen tai toimivuuteen, ja siten myös jäljittää ne eteenpäin testitapauksiin.

Ohjelmiston validoinnin laajuus ja validointiin käytettävä työmäärä ovat suhteessa laitteen aiheuttamaan riskiin. Laajaa ohjelmistoa validoitaessa korkeamman riskin omaavien moduulien ohjelmistovaatimusmäärittelyihin, suunnitteludokumentaatioon ja testitapauksiin on kohdistettava perinpohjaiset ja yksityiskohtaiset tarkastukset.

Silloin kun toiminnan tuloksia ei voida varmistaa lopputuotteen testausmenetelmillä, suunnitellaan suunnittelu/tuotantoprosessi siten, että sopivien koeajojen (validointiajojen) avulla voidaan osoittaa tavoitteiden toteutuminen. Suunnitteluprosessin ollessa usein kertaluonteinen tapahtuma, sovelletaan näihin yleisiä toimintamalleja, kuten ISO 9001, joihin lisätään tarvittavat tapauskohtaiset lisäykset. Eräs malli lääkintälaitteiden ohjelmistokehitykselle on kuvattu standardissa SFS-EN 60601-1-4.

Julkaisussa kuvataan validoinnin kohteet siten, että valmistaja kykenisi vertaamaan niitä oman ohjelmistotuotantonsa prosesseihin. Validointimenettelyn käyttäminen edellyttää valmistajalta omien ko. prosessien tunnistamista, määrittelemistä sekä ohjeistamista.

Validoinnin kohteena ovat kaikki tuotteen laatuun ja turvallisuuteen vaikuttavat tekijät, kuten organisaatio, laatujärjestelmä ja projektin hallinta. Kun tuotteen laadulliset ja määrälliset ominaisuudet on tunnistettu, voidaan aloittaa validointisuunnitelman laatiminen. Validointi kohdistuu ensisijaisesti aina tuotteen validointiin, mutta myös tuotteen tekemisessä ja suunnittelussa apuna käytettäviä tukiprosesseja tullaan validoimaan.

Validointi kohdistuu myös prosessin arviointiin, jolloin siinä arvioidaan projektin hallintaa ja aikataulutusta, suunnitelmia, toteutusta ja analyyseja sekä validointitoimenpiteiden valintaan vaikuttavia kriteereitä ja dokumentointia. Validoinnin toteutus riippuu ohjelmistotyypistä ja -lähteestä. Valmistajalla voi olla tuotteessaan useista eri lähteistä tulevia ohjelmistoja, joista jokaisen vaatimustenmukaisuus tulisi valmistajan kyetä osoittamaan.

Dokumentoitu ohjelmistotuotannon vaihejakomalli selkeine tehtäväjakoineen on tässä tapauksessa merkitsevässä asemassa. Laajoissa ohjelmistoprojekteissa on huomioitava tiedonkulku eri suunnittelutiimien välillä, mikä voidaan hoitaa esim. tarkkaan määritellyillä suunnitelmilla ja suunnittelukatselmuksilla.

Alihankinnassa on tärkeää, että valmistaja määrittelee vaatimukset mahdollisimman täsmällisesti. Käytännössä vaatimus ”täytettävä standardin SFS-EN 60601-1-4 vaatimus” on hyvin laaja, ja ilman tarkempaa määrittelyä voi alihankkijalle olla epäselvää, mitkä vaatimukset soveltuvat ko. ohjelmistolle ja mitkä eivät. Näissä tapauksissa valmistajan tehtäväksi jää osoittaa ohjelmiston vaatimustenmukaisuus. Tämä edellyttää laajoja testejä, simulointeja ja raportointeja. Lisäksi on huomioitava dokumentoinnille esitettävät vaatimukset. Kaikki tämä voi olla jälkikäteen työlästä tai jopa mahdotonta toteuttaa. Valmistajan on valvottava alihankkijan toimintaa, mikä tarkoittaa valmistajan suorittamia auditointeja alihankkijan tiloissa. Tilaisuudesta kirjoitetaan auditointiraportti, joka liitetään osaksi valmistajan omaa laatujärjestelmää.

Kolmannen osapuolen ohjelmistot toimitetaan tavallisesti komponentteina tai kirjastoina ilman lähdekoodia. Käytännössä niitä ovat laiteohjain, käyttöjärjestelmä, signaalinkäsittely-, kuvankäsittelykirjasto tai vastaava algoritmi, tietokanta tai tietoliikenneohjelmisto, käyttöliittymä tai "business-object"-komponentti tai ohjelmistokehitystyökalu.

Valmistajan vastuulle jää näissä tapauksissa tutkia ohjelmien soveltuvuus suunniteltuun käyttöön. Ohjelmistoissa voi olla myös useita eri ominaisuuksia, joista käytetään vain muutamaa toimintoa. Valmistajan tulisi kyetä validoimaan käytetyt toiminnot ja ominaisuudet, mutta myös osoittamaan, että käyttämättömät osat eivät tee mitään asiaan kuulumatonta.

Kolmannen osapuolen arviointi suoritetaan ISO 9000 -vaatimuksiin perustuen, mutta taustalla ovat direktiivin (MDD 1993) olennaiset vaatimukset sekä FDA:n arviointiohjeet. Ohjelmistotuotantoprosessia arvioidaan tuotekohtaisten dokumenttien pohjalta. Arviointilaitos tarkastaa yrityksen suorittaman validoinnin ja raportoinnin. Arvioija hakee kokonaisvaltaista käsitystä vaatimusten ja toteutusten välisistä riippuvuuksista ja niiden hallinnata.

Lähdeluettelo

Beizer, B. 1989. Software testing techniques, Van Nostrand Reinhold.

Boehm, B. W. 1984. Software Engineering Economics, IEEE Transaction on Software Engineering, SE-10, 1, s. 4–21.

CERT, Computer Emergency Response Team. 1999. Täydellinen asiakirjaluettelo hakuviitteineen on osoitteessa http://www.cert.org/security-improvement/

EN 1050, standard. 1996, Safety of machinery — Principles for risk assessment.

EN 60601-1-4, standard. 1996. Medical Electrical Equipment--Part 4: Collateral Standard: Programmable Electrical Medical Systems. International Electrotechnical Commission, IEC, Geneva, Switzerland.

FDA, Food and Drug Administration. 1991. Reviewer Guidance for Computer-Controlled Medical Devices Undergoing 510(k) Review. Rockville, MD. Center for Devices and Radiological Health.

FDA, Food and Drug Administration. 1995. Premarket Notification 510(k): Regulatory Requirements for Medical Devices. Center for Devices and Radiological Health. 30 s. Saatavissa: http://www.fda.gov/cdrh/manual/510kprt1.html (viittaus marraskuussa 2001).

FDA, Food and Drug Administration. 1997. General Principles of Software Validation. Draft guidance, version 1.1. Center for Devices and Radiological Health. Saatavissa: http://www.fda.gov/cdrh/ode/swareval.html (viittaus marraskuussa 2001).

FDA, Food and Drug Administration. 1998a. Code of Federal Regulations 21 (CFR 21), Parts 800 to 1299. Center for Devices and Radiological Health. 701 s.

FDA, Food and Drug Administration. 1998b. Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. Center for Devices and Radiological Health. 42 s. Saatavissa: http://www.fda.gov/cdrh/ode/software.pdf (viittaus marraskuussa 2001).

FDA, Food and Drug Administration. 1999a. Guidance for Industry, FDA Reviewers and Compliance on: Off-The-Self Software Use in Medical Devices. Issued on 9.9.1999. Center for Devices and Radiological Health. 29 s. Saatavissa: http://www.fda.gov/cdrh/ode/otssguid.pdf (viittaus marraskuussa 2001).

FDA, Food and Drug Administration. 1999b. How To Prepare A Traditional 510(k). Food and Drug Administration. Center for Devices and Radiological Health. Saatavissa: http://www.fda.gov/ cdrh/ devadvice/3143.html (viittaus marraskuussa 2001).

Fenton, N. E. 1995. Software Measurement: A necessary scientific basis. Predictably Dependable Computing Systems. Berlin: Springer-Verlag. S. 67–86. ISBN 3-540-59334-9.

GHTF, The Global Harmonization Task Force. 1999. Design Control Guidance for Medical Device Manufacturers, GHTF.SG3.N99-9. 49 s.

GHTF, The Global Harmonization Task Force. 1999. Essential Principles of Safety & Performance of Medical Devices. GHTF.SG1.N020R5. 12 s.

GHTF, The Global Harmonization Task Force. 1999. Process Validation Guidance, GHTF.SG3.N99-10. 34 s.

Haikala, I. & Märijärvi, J. 1998. Ohjelmistotuotanto, Suomen Atk-kustannus. 385 s.

Harju, H. 2000. Ohjelmiston luotettavuuden kvalitatiivinen arviointi. Espoo: Valtion teknillinen tutkimuskeskus. 111 s. (VTT Tiedotteita 2066). ISBN 951-38-5766-2.

Herrmann, D. S. & Zier, D. A. 1999. Using IEC 601-1-4 to Satisfy FDA Software Guidance Requirements. Saatavissa: http://www.devicelink.com/mddi/archive/95/12/013.html (viitattu marraskuussa 2001).

IEC 60812, standard. 1985. Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA). International Electrotechnical Commission, Geneve.

IEC 61025, standard. 1990. Fault Tree Analysis (FTA). International Electrotechnical Commission, Geneve.

IEC 61508, standard. 2000. Functional Safety of Programmable Electronic Systems: Generic Aspects. International Electrotechnical Commission.

IEEE 1074, standard. 1997. IEEE Standard for Developing Software Life Cycle Processes.

Ippolito, L. & Wallace, D. R. 1995. A Study on Hazard Analysis in High Integrity Software Standards and Guidelines. National Institute of Standards and Technology. 44 s. (NISTIR 5589).

ISO/IEC 12207, standard. 1995. Information technology - Software life cycle processes.

ISO/FDIS, standardi. 2000. Medical devices – Application of risk management to medical devices.

Kitchenham, B. 1990. Software Development Metrics and Models. In: Software Reliability Handbooks, edited by Rook, P. New York: Elsevier Applied Science, s. 441–486.

Kletz, T. 1986. HAZOP and HAZAN. Institution of Chemical Engineers, UK.

L 1505. Laki. 1994. Laki terveydenhuollon laitteista ja tarvikkeista.

Laprie, J.-C. 1998. Dependability: Basic Concepts and Terminology. Dependability Handbook. Toulouse: Laboratory for Dependability Engineering. 290 s. (LAAS Report no 98-346.)

Leveson, N. 1995. Safeware: System Safety and Computers. A guide to preventing accidents and losses caused by technology. 1. p. New York: Addison-Wesley. 680 s. ISBN 0-201-11972-2

Leveson, N. G. & Turner, C. S. 1992. An investigation of the Therac-25 accident. University of Washington. UCI Technical Report 92-108. 59 s.

MDD, Direktiivi. 1993. Medical Devices Directive. Terveydenhuollon laitteita ja tarvikkeita koskeva direktiivi 93/42/ETY.

McCall, J. A. 1994. Quality factors. In: Encyclopedia of Software Engineering (John J. Marciniak, Ed.). New York: John Wiley & Sons, s. 958–969.

Mills, E. E. 1988. Software Metrics, SEI Curriculum Module SEI-CM-12-1.1, December 1988, Carnegie Mellon University, Software Engineering Institute (http://www.sei.cmu.edu/).

Musa, J. D. 1980. Software reliability measurement, J. Sys. And Software, 1, 3, s. 223–241.

Musa, J. D., Iannino, A. & Okumoto, K. 1987. Software Reliability. Measurement, Prediction, Application. New York: McGraw-Hill. 621 s.

Palady, P. 1998. Failure Modes & Effects Analysis, Author's Edition. USA: Practical Applications. 262 s. ISBN 0-9663160-0-2.

Scavo, F. 1994. Software Validation for Pharmaceutical and Medical Device Manufacturers. California, USA: APICS International Conference in San Diego.

SFS-EN 1441, standardi. 1998. Terveydenhuollon laitteet ja tarvikkeet. Riskianalyysi. Helsinki: Suomen Standardisoimisliitto, SFS, 26 s.

SFS-EN 60601-1, standardi. 1998. Sähkökäyttöiset lääkintälaitteet. Osa 1: Yleiset turvallisuusvaatimukset, 221 s.

SFS-EN-60601-1-4, standard. 1996. Medical Electrical Equipment--Part 4: Collateral Standard: Programmable Electrical Medical Systems, 64 s.

SFS-EN 60601-1-4, standard. 1999. International Electrotechnical Commission, IEC. Amendment 1.

SFS-EN ISO 9000-3, standardi. 1997. Laatujohtamisen ja laadunvarmistuksen standardit. Osa 3: Suuntaviivat standardin SFS-EN ISO 9001 soveltamiseksi ohjelmistojen kehittämisessä, toimittamisessa ja ylläpidossa. Helsinki: Suomen Standardisoimisliitto, SFS, 34 s.

SFS-EN ISO 9001, standardi. 2000. Laatuhallintajärjestelmät. Vaatimukset. Helsinki: Suomen Standardisoimisliitto, SFS, 60 s.

SFS-EN ISO 9002, standardi. 1994. Laatujärjestelmät. Malli suunnittelussa tai tuotekehityksessä, tuotannossa, asennuksessa ja toimituksen jälkeisissä palveluissa toteutettavalle laadunvarmistukselle. Helsinki: Suomen Standardisoimisliitto, SFS

SFS-EN ISO 9003, standardi. 1994. Laatujärjestelmät. Malli tarkastuksessa ja testauksessa toteutettavalle laadunvarmistukselle. Helsinki: Suomen Standardisoimisliitto, SFS.

Storey, N. 1996. Safety-critical computer systems, Addison-Wesley. 427 s. ISBN 0-201-42787-7.

Wallace, D. R & Kuhn, D. R. 2000. Lessons from 342 medical device failures Saatavissa: http://hissa.nist.gov/effProject/ (viittaus marraskuussa 2001).

Wilson, W., Rosenberg, L. & Hyatt, L. 1996. Automated quality analysis of natural language requirement specifications, Fourteen Annual Pacific Northwest Software Quality Conference, October, 1996.

Windows NT, ohje. 1997. Windows NT server white paper, Securing Windows NT Server Installation, Microsoft Corporation.

Woodward, M. R., Hennell, M. A. & Hedley, D. 1979. A measure of control flow complexity in program text. IEEE Transaction on Software Engineering, SE-5(1), s. 45–50.

Liite A: Tietoturva

> Tietoturva (information security, computer security) on niiden keinojen muodostama kokonaisuus, joiden avulla tietoriskejä pyritään minimoimaan. Tietoturvaan kuuluvia keinoja ovat mm. tietojen turvaaminen menetelmineen ja välineineen, tietojen turvaamiseen osoitetut resurssit sekä käytettävän välineistön tietoturvallisuuteen liittyvät ominaisuudet.
>
> Lähde: TEPA-termipankki

A.1 Yleistä

Sairaaloissa, terveyskeskuksissa ja tutkimuslaitoksissa on aina luotu, käsitelty, siirretty ja säilytetty potilastietoa, mutta vasta tietokoneistumisen ja erityisesti tietoverkkojen käyttöönoton jälkeen aineiston suojaamista on alettu kutsua tietoturvaksi. Tietojen katoaminen, asiaton muuttaminen tai joutuminen vääriin käsiin ovat olleet aikaisemminkin riskejä, mutta verkottuminen tuo väärinkäytösten mahdollisuuden entistä useampien ulottuville. Tietosuojan murtaminen ei esimerkiksi edellytä läsnäoloa tutkimushuoneessa, johon järjestelmä on sijoitettu. Tietojen muuttaminen tai varastaminen tietokonejärjestelmästä on nopeaa eikä jätä jälkiä.

Silti riskit eivät ole kasvaneet kohtuuttomiksi: amerikkalaisen lähteen mukaan (Schoenfelt 1999) hakkereita ei pidetä vaarallisimpana tietoturvauhkana sairaaloissa vaan omaa henkilökuntaa, jolla on jo nyt pääsy suurimpaan osaan sairaalassa liikkuvasta informaatiosta. Luottamuksen pettämistä ei kuitenkaan voida estää tietoturvajärjestelmällä, eikä tietoturvan toteuttaminen saa johtaa järjestelmään, joka on niin hankala, että se haittaa hoitohenkilökunnan toimintaa. Toimimattomat järjestelmät ovat alttiita tahallisille laiminlyönneille, sillä ihmiset kiertävät määräyksiä voidakseen työskennellä tehokkaasti. Riskien tunnistaminen ja niiden suuruuden arvioiminen kuuluukin olennaisena osana tietoturvan rakentamiseen, sillä toteuttamiskeinot on valittava suhteessa todelliseen riskiin ja toiminnan tehokkuuteen.

Järjestelmävalmistajan on osaltaan huolehdittava potilastiedon luottamuksellisuuden, eheyden ja saatavuuden säilymisestä. Tässä projektissa lähteenä käytetyt standardit eivät esitä selkeitä vaatimuksia tietoturvan toteuttamisesta eivätkä jaa vastuuta valmistajan ja käyttäjän välillä. On selvää, että järjestelmävalmistaja on velvollinen toteuttamaan tuotteensa siten, että käyttäjällä on mahdollisuus rakentaa tietoturva oman strategiansa mukaiseksi, s.o. valmistajan valinnat eivät estä strategian toteuttamista. Toisaalta valmistajaa ei kuitenkaan vedetä vastuuseen esimerkiksi käyttäjän tietoturvastrategian puuttumisesta.

Tietoturvaan kuuluu potilaan yksityisyyden suojaamisen lisäksi myös lääkintälaitteen käytettävyyden varmistaminen. Valmistajan kannalta tietoturvaongelma tiivistyy kolmeen kysymykseen:

- Kuinka varmistetaan tietojen eheys (integrity) ja kuinka havaitaan eheyden menettäminen?.
- Kuinka turvataan järjestelmän käytettävyys ja tietojen saatavuus (availability)?
- Kuinka suojataan tieto ja järjestelmä tahattomilta ja asiattomilta muutos- tai anastamisyrityksiltä (confidenciality)?

Jos tarkastellaan tahallista haitantekoa – kuten hakkerointia tai vakoilua –, lääkintälaitteen ohjelmiston piirteet, puutteet ja virheet muuttuvat tietoturvariskeiksi vasta, jos seuraavat reunaehdot toteutuvat yhtä aikaa:

- Piirre on hyödynnettävissä tietosuojan murtamisessa (exploitability).
- On olemassa tekijä, hyökkääjä, joka tuntee piirteen ja sen hyödynnettävyyden (intruder).
- Murtamisesta on tekijälle hyötyä (motive).
- Järjestelmä on hyökkääjän ulottuvilla (availability).

Lääkintälaitteiden kytkeminen verkkoon – paikalliseen tai julkiseen – muuttaa tietoturvan uhkakuvia eniten juuri saavutettavuuden kautta (Lindqvist & Jonsson 1998): verkko muodostaa fyysisen linjan laitteen ja suuren käyttäjäjoukon välille. Koska suurin osa laitteistoista rakennetaan valmiskomponenteista, joista tärkeimpiä ovat käyttöjärjestelmät, niiden tietoturva-aukkoja koskevaa tietoa ja häirinnän välineitä on runsaasti saatavilla Internetistä. Verkko tarjoaa myös eräänlaisen markkinapaikan, jossa tietoa voidaan myydä ja siirtää nopeasti ja huomaamatta. Uuden turvariskin tuo mukanaan suoritettava koodi, kuten makrot ja Java appletit, joka automaattisesta siirretään verkosta ja suoritetaan käyttäjän koneessa.

A.2 Viranomaisvaatimukset

FDA ei esitä suoria vaatimuksia järjestelmätoimittajille vaan kehottaa ottamaan tietoturvan huomioon järjestelmäkehityksessä. FDA:n ohje (FDA 1998) mainitsee seuraavat aiheet kappaleessa A17:

- Mittausaineiston ja potilastietojen suojaaminen
- Tietojen muuttamisen dokumentointi
- Tietojen muuttamisen salliminen vain auktorisoiduille käyttäjille

- Ohjelmistojen ja datan varmuuskopiointi säännöllisin väliajoin
- Salasanojen käyttö
- Viestien salaaminen (encryption) tarvittaessa

FDA painottaa jäljitettävyyttä ja toipumiskykyä: riippumatta järjestelmälle valitusta tietoturvatasosta tai tietoturvan toteuttamisesta kaikista järjestelmää koskevista muutoksista on jäätävä jälki. On siis oltava keino havaita tietoturvaan kohdistunut hyökkäys. Lisäksi on oltava keino toipua järjestelmän kaatumisesta tai esimerkiksi asiattomasta muutosyrityksestä.

Jäljitettävyyden tekninen toteuttaminen jää todennäköisesti valmistajan vastuulle, sillä mittaukseen liittyvien tietojen (ohjelmiston versionnumero, aika, paikka, mittaaja, potilas) kerääminen mittaavan järjestelmän ulkopuolelta on hankalaa. Laitevalmistajan ei tarvitse myöskään välttämättä tuottaa login-toimintoa, joka yleensä sisältyy käyttöjärjestelmäpalveluihin, mutta hän joutuu rakentamaan versionpäivityksen (huollon) ympärille suojauksen ja pohtimaan käytönestolta suojautumista.

SFS-EN 60601-1-4 sisältää kolme tietoturvaan välillisesti liittyvää kohtaa :

- 52.204.3.1.6 Vaara-analyysissä on tunnistettava [tietoturvan pettämiseen johtavat tai tietoturvan pettämisen aiheuttavat] vaarat, kun vaaran syyt ovat inhimillisiä ja tahallisia tai tahattomia.
- 52.206.2 Vaatimusmäärittelyn tulee eritellä toiminnot, joihin liittyy riski, ja ne toiminnot, jotka valvovat riskiä. Virhetoiminnon vaikutus [tietoturvaan] on tunnistettava.
- 52.207.3 Arkkitehtuurin vaatimusmäärittelyn on sisällettävä suojautumiskeinot inhimillisiltä ja tahattomilta syiltä[, jotka uhkaavat tietoturvaa].

Kuvaus tietoturvan toteuttamisesta sisältyy yhtenä kohtana riskienhallintatiedostoon. Standardi edellyttää siis myös tietoturvariskin dokumentoitua arviointia.

A.3 Tietoturvaratkaisujen kohdistaminen järjestelmän elinkaaren vaiheisiin

Abrams (1995) jakaa tietokoneiden väärinkäytön neljään luokkaan:

1. Resurssivarkaudet: tietokoneen muistia tai keskusyksikköaikaa käytetään asiattomasti.

2. Käytön estäminen: järjestelmä kaadetaan tai lamautetaan toimintakyvyttömäksi.
3. Tietovarkaudet
4. Tietojen muuttaminen

Tietoturvan tasoa voidaan arvioida luokkien avulla tarkastelemalla, mitä tahallinen tai tahaton aie kussakin luokassa merkitsee järjestelmän kannalta: kuinka rikkomus toteutetaan, kuinka sen tulos havaitaan, ja kuinka hyökkäyksestä toivutaan.

Käytetään Abramsin väärinkäyttöluokitusta apuna, kun kuvataan, missä tuotantovaiheissa tietoturvaratkaisut tehdään:

Resurssivarkaudet (theft of computational resources)

Oletetaan, että lääkintälaitteen osana on PC, jossa mittausta ohjaavaa ohjelmistoa ajetaan ja joka on kytketty verkkoon. Oletetaan vielä, että logintoiminto on käytössä. Resurssien väärinkäyttö tarkoittaa keskusyksikköajan tai muistitilan varastamista. Kaikilla auktorisoiduilla käyttäjillä, joilla on kirjoitus- ja suoritusoikeudet, on mahdollisuus sijoittaa muistiin omia tiedostojaan sekä ladata koneeseen ylimääräisiä ohjelmia. Laittomilla käyttäjillä on myös tämä mahdollisuus, mutta se edellyttää järjestelmään murtautumista.

Ellei suojausta voida tehdä esimerkiksi siksi, että asiakas haluaa ajaa muita ohjelmia samassa koneessa, ongelma muuttaa muotoaan ja muuttuu lopulta juridiseksi. Yhteiskäyttö toimii tietyillä edellytyksillä:

- Muut ohjelmat eivät saa vaarantaa lääkintälaitteen ohjelmiston oikeutta keskusyksikköaikaan ja muistitilaan. Ohjelman suoritus ei saa häiriintyä.
- Konfigurointia ei saa muuttaa.

Järjestelmän konfigurointi voi muuttua esimerkiksi siten, että käyttäjä ajaa jonkin toisen ohjelman installointiohjelman ja tämä kirjoittaa alkuperäisten dll-asetusten päälle.

Valmistajan on pohdittava, voiko oman ohjelmiston toiminnan taata näillä edellytyksillä vai kieltäydytäänkö vastuusta.

Resurssien väärinkäytön estävät tietoturvaratkaisut tehdään määrittely- ja arkkitehtuurivaiheessa. Ne otetaan huomioon myös käyttöönotto- ja ylläpitovaiheissa:

- määritellään käyttäjäprofiilit oikeuksineen

- määritellään välttämättömät resurssit (luku-, kirjoitus-, suoritusoikeudet, pääsy hakemistoihin, verkkopalvelut) jokaiselle käyttäjäryhmälle ja tarjotaan vain nämä sisäänkirjoittautuessa
- määritellään laitteen konfigurointi
- laaditaan asennuspaketti ohjeineen.

Laite voidaan esimerkiksi konfiguroida siten, että vain lääkintälaitteen oma ohjelma saa kirjoitus- ja suoritusoikeudet. Uudelleenkonfigurointi levykkeeltä estetään lukitsemalla levyasema fyysisesti koteloon.

Resurssivarkauksien ongelma koskee lähinnä käyttöympäristöä eikä niinkään sovellusohjelmaa, sillä konfigurointi tehdään käyttöjärjestelmätasolla.

Käytönesto (denial of service; disruption of computational services)

Ohjelmallinen käytönesto voidaan toteuttaa verkon kautta pommittamalla työasemaa signaaleilla, jotka käyttöjärjestelmävirheestä johtuen kaatavat tietokoneen. Verkosta tulevaa käytönestohyökkäystä vastaan käyttöjärjestelmät ovat melko suojattomia. Jos käytönesto on esimerkiksi mittaustilanteessa kriittinen, harkitaan kytkeytymistä verkosta käytön ajaksi. On mahdollista, että verkkoyhteys luodaan vain esimerkiksi potilastietojen siirron tai versionpäivityksen ajaksi.

Samanlainen vaikutus voi olla samanaikaisesti ajetuilla ohjelmilla, joista jokin aiheuttaa virhetilanteen, tai lääkintälaitteen omalla ohjelmistolla, jos se esimerkiksi kaatuu vääränlaisten syötteiden takia (esim. buffer overflow, smash-the-stack attack).

Käytöneston kriittisyys on pohdittava vaara-analyysissä, jonka tuloksesta seuraa vaatimusmäärittely käyttöjärjestelmälle ja verkkoyhteyksille. Ohjelmiston osalta käytöneston mahdollistavat puutteet on tunnistettava ohjelmiston toteutusvaiheessa ja niistä on luotava testitapaukset testausta varten.

Tietojen muuttaminen (unauthorized information modification)

Potilastietojen kirjaaminen ja niiden muuttaminen on laitteiston auktorisoidun käyttäjän oikeus. Muutoksista on jäätävä jälki lokiin. Tässä kohdassa tarkoitetaan kuitenkin muuttamista jälkiä jättämättä. Se on mahdollista murtautumalla lokitiedostoon. Ohjelmistovirhe tietojen syöttämisen yhteydessä saattaa saada aikaan saman tuloksen. Lokin siirtäminen arkistoon on yksi riskialtis vaihe. Arkistoiminen ei enää kuulu tämän tarkastelun piiriin.

Lokitiedoston murtaminen edellyttää kirjoitussuojan murtamista ja, jos loki oli kryptattu, salauksen murtamista. Suojan vahvuus riippuu käyttöjärjes-

telmästä ja salausalgoritmista ja riskin suuruus jälleen kerran ohjelmiston alttiudesta verkosta tuleville hyökkäyksille.

Versiopäivitys ja laitteen huolto voivat altistaa myös tahattomille ei-toivotuille muutoksille.

Tietokantaohjelmissa saattaa olla oma käyttäjäluokituksensa, joka ohittaa käyttöjärjestelmän luokituksen. Myös tietokantaohjelman konfigurointi on tehtävä määrittelyvaiheessa.

Vaara-analyysiä varten valmistaja voisi luoda kuvauksen tiedon keräytymisestä, kerääjistä ja tiedon siirtymisestä eri medioissa. Kuvauksesta olisi helppo johtaa vaaratilanteet ja pohtia riskiä.

Tietovarkaudet (unauthorized information disclosure)

Tietovarkaudet ovat käytönaikaisia ongelmia. Kaikki auktorisoidut käyttäjät voivat vapaasti lukea omalle käyttöoikeudelleen luovutettua tietoa. Tieto saattaa vuotaa suojaustasolta toiselle ohjelmistovirheen takia. Vuotamisen mekanismia kutsutaan piilokanavaksi (covert channel). Yksi esimerkki tästä on tiedon lukeminen ennen kirjoittamista: jos käyttäjällä on oikeus valita tietojenkäsittelyn järjestys, on mahdollista lukea tietoa muistialueelta ennen kuin sinne on tällä käyttökerralla kirjoitettu mitään. Jos muistia ei alusteta käytön alussa, ohjelma lukee edellisen istunnon muistiin tallettamaa tietoa. (Abrams 1995)

Tärkein suojauskeino on laitteen login-toiminto. Ohjelmistovirheiden vaikutus poistetaan suunnittelu- ja toteutusvaiheissa (koodaus).

Varkaudet saattavat kohdistua paitsi tietokoneen muistiin myös siirtotilanteeseen. Kuten edellisessä kohdassa mainittiin, tietojen keräytymisestä ja siirtymisestä laadittu kuvaus helpottaa riskianalyysin tekemistä.

Tietoturva on samalla tavalla dynaaminen suure kuin koko tuotekin. Staattiset ratkaisut eivät ole kestäviä. Koska tietoturva on prosessi eikä projekti, riskienarvioimisen tulisi olla jatkuvaa, sillä käyttäjän työympäristö verkkoineen ja tietokoneistettuine toimintoineen muuttuu luultavasti koko ajan. Esimerkiksi käyttöjärjestelmän vaihtaminen vaihtaa samalla osan tietoturvaongelmista toisiin; valmistajalla tulisi olla näkemys, kuinka hoidetaan tunnettujen tietoturva-aukkojen paikkaaminen.

Muutosten jäljitettävyyttä ja virhetilanteista toipumista voidaan pitää tietoturvan vähimmäisvaatimuksina. Jäljitettävyyden toteuttaminen edellyttää tietojen kirjautumista lokiin ja lokin turvaamista muutosyrityksiltä ja järjestelmän kaatumiselta. Toipumista varten ylläpidetään varakopioita.

Jos valmistajan tehtävänä on ohjelmoida järjestelmäänsä lokin kirjautuminen ja sen suojaaminen, se, kenen vastuulle jää lokin analysoiminen hyökkäysten tai virheiden löytämiseksi, määritellään käyttäjän tietoturvastrategiassa. Loki pitäisi tarkistaa aika ajoin ja tietenkin havaittujen ongelmatilanteiden jälkeen, jotta mahdolliset puutteet tietoturvajärjestelmässä opittaisiin korjaamaan.

Asiattomalta käytöltä suojautumiseen riittävät yleensä tavanomaiset keinot, kuten käyttöoikeuksien suojaaminen salasanalla ja käyttäjien jakaminen oikeuksiltaan erilaisiin luokkiin, jos suojaamisen tavoitteena on estää lähinnä vahingot ja uteliaat kokeilut. Käyttöjärjestelmien sisältämien virheiden takia on mahdollista, että ohjelmisto voidaan kaataa murtamatta salasanaa (ks. esim. Stout: Known NT exploits 1999). Käytönestoa vastaan nykyiset käyttöjärjestelmät ovat huonosti suojattuja (Howard 1998).

A.4 Vaatimustenmukaisuuden osoittaminen

IEC-601-1-4:n asettamiin vaatimuksiin on suoraviivaista vastata, koska vaatimukset on eksplisiittisesti annettu. FDA:n ohje ei muotoile selkeitä tietoturvaa koskevia vaatimuksia. Vaatimustenmukaisuuden osoittamiseen tarvitaan seuraavat dokumentit:

Riskienhallintatiedostossa on oltava tietoturvan vaara-analyysi ja riskiarvio. Riskejä ovat esimerkiksi

- inhimillisen virheen aiheuttama tietovuoto
- tahallinen murtautumisyritys
- ohjelmistovirheestä aiheutuva tiedon kulkeutuminen suojaustasolta toiselle
- laitevian aiheuttama tietojen katoaminen.

Järjestelmällä on oltava keino suojautua tietoturvariskeiltä. Keinoja ovat esimerkiksi

- salasanalla suojattu käyttöoikeus (ennaltaehkäisy)
- menetelmä lokitiedoston keräämiseksi, seuraamiseksi ja virheiden tai puutteiden korjaamiseksi (jäljitettävyys, eheys, ennaltaehkäisy, toipuminen)
- varmuuskopiointi (toipuminen)
- käyttäjän opastaminen, huolellinen ohjeistaminen (ennaltaehkäisy).

Tämän lisäksi valmistajan on tunnettava versiopäivitysten ja käyttöympäristön muutosten vaikutus tietoturvaan ja sen ylläpitoon.

Tietoturvaan erikoistunut CERT on toimittanut yksityiskohtaisen listan tietoturvan ylläpidosta käytettävistä keinoista (Simmel 1999; Ford 1999).

A.5 Yhteenveto

Tietosuojan ja -turvan toteuttaminen on toistaiseksi terveydenhuollon yksiköiden kuten sairaaloiden ja terveyskeskusten vastuulla. Tietoturvan merkitys kasvaa, kun tarve tiedonsiirtoon hoitoyksiköiden välillä kasvaa. Uudenmaan sairaanhoitopiiri on hyvä esimerkki terveydenhuollon verkottumisesta ja sen aiheuttamista uusista vaatimuksista tietoturvalle.

Lääkintälaitevalmistajien on omalta osaltaan tarjottava tukensa tietoturvan rakentamisessa. Ilmeisiä ratkaisuja, jotka saattavat muuttua tulevaisuudessa vaatimuksiksi, ovat standardiratkaisujen käyttö tietoturvan toteuttamisessa (salaaminen, todentaminen, lokin kerääminen, ohjelmaversioiden identifiointi).

Jos ohjelmistojen merkitys lääkintälaitteiden osana tai erillisinä palveluina kasvaa, valmistajat joutuvat määrittelemään ehdot mm. eri sovellusten yhtäaikaiselle ajamiselle samassa tietokone- ja verkkoympäristössä. Samasta syystä on kiinnitettävä erityistä huomiota ohjelmiston tarvitsemien resurssien määrittelyyn (CPU-aika, muisti, konfigurointi).

Lähdeluettelo

Abrams, Podell & Jajodia, (toim.). 1995. Information Security – an Integrated Collection of Essays, IEEE Computer Society Press, ISBN 0-8186-3662-9.

FDA, Food and Drug Administration. 1998. Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. CDRH, Center for Devices and Radiological Health.

Lindqvist, U. & Jonsson, E. 1998. A map of security risks associated with using COTS, Computer, June 1998.

Howard, J. 1997. An Analysis of Security Incidents On the Internet 1989–1995, dissertation. Carnegie Mellon University.

Schoenfelt, S. 1999. Next Generation: How Internet Technology Propels the Electronic Medical Record, Journal of AHIMA. Saatavissa: http://www.ahima.org/ publications/2f/feature.0999.3.htm. (viittaus marraskuussa 2001).

Simmel, D. et al. 1999. Securing Desktop Workstation, CMU/SEI-SIM-004.

Ford, G. et al. 1999. Securing Network Server, CMU/SEI-SIM-006.

Stout, B. 1999. Known NT exploits. Saatavissa: http://www.iss.net/vd/bill_stout/Ntexploits.hdr (viittaus marraskuussa 2001).

Liite B: Tarkastuslista standardin SFS-EN 60601-1-4 vaatimuksille sekä opastusta

B.1 SFS-EN 60601-1-4 Checklist

The purpose of this document is to show that the below mentioned software product fulfills the compliance with the requirements of standard SFS-EN 60601-1-4 (1996).

Taulukko B 1. Tarkastuslista standardin SFS-EN 60601-1-4 vaatimuksille.
Note for project team: FDA requirement will be added to this document if FDA-requirement deviates from SFS-EN 60601-1-4 requirement. [remove this text from ready template]

SYSTEM IDENTIFICATION	Revision
[SW-product description]	
[Main SW-version]	
[related Sub/PESS SW-versions]	

Level of Concern for FDA: Note for project team: The manufacturer shall determine Severity Level of SFS-EN 60601-1-4 and how it correlate with Level of Concern of FDA. This can be stated in general quality documents of software production.[remove this text from ready template]			
Minor	Moderate	Major	Reference document for LOC:

Members of design team	Role / qualification / reference documents

Members of validation team	Role / qualification / reference documents

Checked by	Approved by	Date

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
6	**Identification, marking and documents**				
6.8	*ACCOMPANYING DOCUMENTS*				
6.8.201	All relevant information regarding significant RESIDUAL RISK including descriptions of the HAZARDS and any actions by the OPERATOR or the USER necessary to avoid/mitigate them shall be placed in both the INSTRUCTIONS FOR USE and the RISK MANAGEMENT FILE.				
6.8.202	ACCOMPANYING DOCUMENTS for the PEMS shall identify, as a minimum, the MANUFACTURER and a unique identifier such as revision level and date of release/issue. *NOTE: information pertaining to any specific EQUIPMENT that software is intended to be used in conjunction with, and a means by which the MANUFACTURER can be contacted, can be located on the package or in the INSTRUCTIONS FOR USE so that it is available to the USER independently of the software operation.*				
52	**Abnormal operation and fault conditions**				
52.201	*Documentation*				
52.201.1	Documents produced from application of this standard shall be maintained and shall form part of the quality records; see figure 201. This should be done in accordance with 6.3 of ISO 9000-3.				
52.201.2 52.201.3	These documents, herein referred to as the RISK MANAGEMENT FILE, shall be approved, issued and changed in accordance with a formal configuration management system. This should be done in accordance with 6.2 of ISO 9000-3. A RISK MANAGEMENT SUMMARY shall be developed throughout the DEVELOPMENT LIFE-CYCLE as part of the RISK MANAGEMENT FILE**.** It shall contain: a) identified HAZARDS and their initiating causes; b) estimation of RISK**;**				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
	c) reference to the SAFETY measures, used to eliminate or control the RISK of the HAZARD; d) evaluation of effectiveness of RISK control., e) reference to VERIFICATION				
52.202	*RISK management plan*				
52.202.1	The MANUFACTURER shall prepare a RISK management plan.				
52.202.2 52.202.3	This plan shall include the following: a) scope of the plan, defining the project or product and the DEVELOPMENT LIFE-CYCLE phases for which the plan is applicable; b) the DEVELOPMENT LIFE-CYCLE to be applied (see 52.203) including a VERIFICATION plan and a VALIDATION plan; c) management responsibilities in accordance with 4.1 of ISO 9001; d) RISK management process; e) requirements for reviews. If the plan changes during the course of development, a record of the changes shall be kept.				
52.203	*DEVELOPMENT LIFE-CYCLE*				
52.203.1	A DEVELOPMENT LIFE-CYCLE shall be defined for the design and development of the PEMS.				
52.203.2	The DEVELOPMENT LIFE-CYCLE shall be divided into phases and tasks, with a well-defined input, output and activity for each.				
52.203.3	The DEVELOPMENT LIFE-CYCLE shall include integral processes for RISK management.				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.203.4	The DEVELOPMENT LIFE-CYCLE shall include documentation requirements.				
52.203.5	RISK management activities shall apply throughout the DEVELOPMENT LIFE-CYCLE as appropriate; see 52.204.				
52.203.6	Where appropriate, a defined system for problem resolution within and between all phases and tasks of the DEVELOPMENT LIFE CYCLE shall be developed and maintained as part of the RISK MANAGEMENT FILE. Depending upon the problem, the system may have the following characteristics:				
	- be defined as a part of the DEVELOPMENT LIFE-CYCLE;				
	- allow the reporting of potential or existing SAFETY and/or performance problems;				
	- include an assessment of each problem for associated RISKS;				
	- identify the criteria (SAFETY and/or performance) that have to be met for the issue to be closed;				
	- identify the action to be taken to resolve each problem;				
	- identify VALIDATION methods for each action;				
	- identify the steps taken for VERIFICATION of continuing compliance.				
52.204	*RISK management process*				
52.204.1	A RISK management process shall be used that has the following elements: -RISK analysis; -RISK control.				
52.204.2	The process shall be applied throughout the DEVELOPMENT LIFE-CYCLE.				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.204.3	*RISK analysis*				
52.204.3.1	*HAZARD ANALYSIS*				
52.204.3.1.1	HAZARD identification shall be carried out as defined in the RISK management plan; see 52.202.				
	HAZARDS shall be identified for all reasonably foreseeable circumstances including: - NORMAL USE; - incorrect use.				
52.204.3.1.3	The HAZARDS considered shall include, as appropriate: - HAZARDS to PATIENTS; - HAZARDS to OPERATORS; - HAZARDS to service personnel; - HAZARDS to bystanders; - HAZARDS to the environment.				
52.204.3.1.4	Reasonably foreseeable sequences of events, which may result in a HAZARD, shall be considered.				
52.204.3.1.5	Initiating causes considered shall include, as appropriate: - human factors including ergonomic limitations; -hardware faults; -software faults; -integration errors; -environmental conditions.				
52.204.3.1.6	Matters considered shall include, as appropriate:				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
	-compatibility of system components, including hardware and software; -user interface, including command language, warning and error messages; -accuracy of translation of text used in the user interface and INSTRUCTIONS FOR USE; -data protection from human intentional or unintentional causes; -RISK/benefit criteria; -third party software.				
52.204.3.1.7	HAZARD identification methods appropriate to the DEVELOPMENT LIFE-CYCLE phase shall be used.				
52.204.3.1.8	The methods used (e.g. fault tree analysis, failure modes and effects analysis) shall be documented in the RISK MANAGEMENT FILE.				
52.204.3.1.9	The results of the application of the methods shall be documented in the RISK MANAGEMENT FILE.				
52.204.3.1.10	Each identified HAZARD and its initiating causes shall be documented in the RISK MANAGEMENT FILE.				
52.204.3.2	*RISK estimation*				
52.204.3.2.1	For each identified HAZARD the RISK shall be estimated.				
52.204.3.2.2	The estimation of the RISK shall be based on an estimation of the likelihood of each HAZARD and/or the SEVERITY of the consequences of each HAZARD**.**				
52.204.3.2.3	The SEVERITY level categorization method shall be recorded in the RISK MANAGEMENT FILE				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.204.3.2.4	The likelihood estimation method shall be either quantitative or qualitative and shall be recorded in the RISK MANAGEMENT FILE.				
52.204.3.2.5	The estimated RISK shall be recorded against each HAZARD in the RISK MANAGEMENT SUMMARY.				
52.204.4	*RISK control*				
52.204.4.1	RISK shall be controlled so that the estimated RISK of each identified HAZARD is made acceptable.				
52.204.4.2	A RISK is acceptable if the RISK is less than or equal to the MAXIMUM TOLERABLE RISK and the RISK is made as low as reasonably practicable.				
52.204.4.3	Methods of RISK control shall reduce the likelihood of the HAZARD or reduce the SEVERITY of the HAZARD or both. The likelihood that the means for RISK reduction will perform correctly shall be specified quantitatively or qualitatively; see annex CCC.				
52.204.4.4	RISK control methods shall be directed at the cause of the HAZARD (e.g. by reducing its likelihood) or by introducing protective measures which operate when the cause of the HAZARD is present, or both, using the following priority: - inherent safe design; - protective measures including alarms; - adequate USER information on the RESIDUAL RISK.				
52.204.4.5	The requirement(s) to control the RISK shall be documented in the RISK MANAGEMENT SUMMARY (directly or as a cross reference).				
52.204.4.6	An evaluation of the effectiveness of the RISK controls shall be recorded in the RISK MANAGEMENT SUMMARY.				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.205	*Qualification of personnel*				
	The design and modification of a PEMS shall be considered as an assigned task in accordance with 4.18 of ISO 9001.				
52.206	*Requirement specification*				
52.206.1	For the PEMS and each of its subsystems (e.g. for a PESS) there shall be a requirement specification.				
52.206.2	The requirement specification shall detail the functions that are RISK-related. This includes functions that control RISKS arising from a) causes arising from environmental conditions; b) causes elsewhere in the PEMS**;** c) possible malfunctions.				
52.206.3	The requirement specification shall include the information necessary to assure that RISK control measures satisfactorily reduce the identified RISKS				
52.207	*Architecture*				
52.207.1	The architecture shall satisfy the requirement specification.				
52.207.2	For the PEMS and each of its subsystems, an architecture shall be specified.				
52.207.3	Where appropriate, the architecture specification of PEMS and its subsystems shall address the RISK CONTROL requirements by reducing the corresponding likelihood of the HAZARD or by reducing the SEVERITY of the HAZARD or both.				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.207.4	Where appropriate, to reduce the likelihood of the HAZARD, the architecture specification shall make use of: a) highly reliable components; b) fail-safe functions; c) redudancy; d) diversity; e) defensive design; f) limits on potential hazardous effects, for example by restricting the available output power and/or introducing means to limit the travel of actuators.				
52.207.5	The architecture specification shall take the following into consideration:				
	a) allocation of RISK control measures to subsystems and components of PEMS; NOTE - Subsystems and components include sensors, actuators, PESS and interfaces.				
	b) failure modes of components and their effects;				
	c) common cause failures;				
	d) systematic failures;				
	e) test interval, test duration and diagnostic coverage;				
	f) maintainability;				
	g) protection from human intentional or unintentional causes.				
52.208	*Design and implementation*				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.208.1	Where appropriate, the design shall be decomposed into subsystems, each having a design and test specification.				
52.208.2	Descriptive data regarding the design environment shall be included in the RISK MANAGEMENT FILE NOTE - See annex DDD for examples of design environment elements.				
52.209	*Verification*				
52.209.1	VERIFICATION of the implementation of SAFETY requirements shall be carried out.				
52.209.2	A VERIFICATION plan shall be produced to show how the SAFETY requirements for each DEVELOPMENT LIFE-CYCLE phase will be verified. The plan shall include				
	a) the selection and documentation of VERIFICATION strategies, activities and techniques;				
	b) the selection and utilization of VERIFICATION tools;				
	c) coverage criteria for VERIFICATION. NOTE - Examples of methods and techniques are - walkthroughs and inspections; - static / dynamic analyses; - white / black box testing				
52.209.3	The VERIFICATION shall be performed according to the VERIFICATION plan. The results of the VERIFICATION activities shall be documented, analyzed and assessed.				
	A reference to the methods, techniques and results of the VERIFICATION shall be included in the RISK MANAGEMENT SUMMARY.				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.210	*Validation*				
52.210.1	VALIDATION of the SAFETY of PEMS under conditions of the intended use shall be carried out				
52.210.2	A VALIDATION plan shall be produced to show that correct SAFETY requirements have been implemented.				
52.210.3	The VALIDATION shall be performed according to the VALIDATION plan. The results of VALIDATION activities shall be documented, analyzed and assessed.				
52.210.4	The leader of the team carrying out the VALIDATION shall be independent of the design team.				
52.210.5	All professional relationships of the members of the VALIDATION team with members of the design team shall be documented in the RISK MANAGEMENT FILE.				
52.210.6	No member of a design team shall be responsible for VALIDATION of his own design.				
52.210.7	A reference to the methods and results of the VALIDATION shall be included in the RISK MANAGEMENT FILE.				
52.211	*Modification*				
52.211.1	If any or all of a design results from a modification of an earlier design then either all of this standard applies as if it were a new design or the continued validity of any previous design documentation shall be assessed under a modification/change procedure.				

SFS-EN 60601-1-4 Clause	Requirement	Yes	No	N/A	Remark, reference document, document of quality system etc.
52.211.2	All relevant documents in the DEVELOPMENT LIFE-CYCLE shall be revised, amended, reviewed, Approved under a document control scheme in accordance with 4.5.2 of ISO 9001 or equivalent.				
52.212	*Assessment*				
52.212.1	Assessment shall be carried out to ensure that the PEMS has been developed in accordance with the requirements of this standard and recorded in the RISK MANAGEMENT FILE. This may be carried out by internal audit.				

B.2 Riskienhallintatiedoston (RMF) rakenne

Standardin SFS-EN 60601-1-4 lähes jokaisen vaatimuksen arviointi tapahtuu riskienhallintatiedoston [RMF] ja sen osana olevan riskienhallintaselostuksen [RMS] sisällön tarkastamisella.

Standardin SFS-EN 60601-1-4 määritelmät riskinhallintatiedostolle, riskinhallintaselostukselle ja ohjelmiston tuotekehityksen elinkaarelle ovat:

Risk Management File	*That part of the quality records required by this standard.*
Risk Management Summary	*Document, which provides tracealibity for each HAZARD and each cause of the HAZARD to the RISK analysis and to the VERIFICATION that the RISK of the HAZARD is controlled.*
Development Life-Cycle	*Necessary activities occurring during a period of time that starts at the concept phase of project and finishes when the VALIDATION of the PEMS is complete.*

Käytännössä tuotteen RMF voidaan toteuttaa kahdella eri tapaa:

- Tuotekohtainen RMF, jolloin kaikki informaatio siirretään yhteen tiedostoon. Haittana on tiedoston koon kasvaminen ja lisätyömäärä.

- Tuotekohtainen tarkastuslista standardin vaatimuksista, jossa kunkin vaatimuksen kohdalla kuvataan dokumentit, jolla ko. kohdan vaatimus täytetään, vaatimuksen täyttyminen osoitetaan tällöin laatujärjestelmän pysyväisdokumentilla tai RMF:n tuotekohtaisella dokumentaatiolla tai molemmilla.

Pysyväisdokumentit voivat olla esimerkiksi dokumentteja, joissa määritellään yrityksen käytössä olevat riskianalyysitekniikat ja ohjelmiston vaihejakomallin kuvausdokumentit.

RMF muodostaa sen dokumentaation, jolla valmistaja voi osoittaa ohjelmistotuotteensa täyttävän sille asetetut vaatimukset. Osa standardin vaatimuksista sijoittuu luontevasti yleisiin laatutiedostoihin, jolloin tuotekohtainen RMF saadaan sisällöltään pienemmäksi ja sen tuottamiseen käytettyä aikaa pienemmäksi.

B.3 Tarkastuslistan käyttö

Tarkastuslista muodostaa koontidokumentin, jolla kootaan tuotteeseen liittyvät dokumentit yhteen. Tällä tavoin valmistaja varmistaa oman työnsä laatua ja nopeuttaa ulkopuolisten tahojen tarkastusta, koska valmistaja kykenee tuottamaan tarkastuksessa vaaditun aineiston hyvin nopeasti tarkastajalle. Valmistajalle tämä näkyy välittömästi nopeutuneena aikatauluna ja tätä kautta pienentyneinä kustannuksina.

Tarkastuslistaa täytetään sarake kerrallaan, jossa käydään läpi kaikki standardin alakohdat. Listassa on kunkin kysymyksen kohdalla kolme saraketta (Yes, No, N/A), joilla valmistaja vastaa ko. kohdan vaatimuksenmukaisuuteen. Valmistaja vastaa harkintansa mukaan kuhunkin kohtaan: soveltuuko tämä kohta, täyttyykö vaatimus vai eikö se täytä standardin vaatimusta. Lisäksi sarakkeeseen 'references' laitetaan viitedokumentit tai perustelu käytetystä ratkaisusta. Vastausten on kuvattava tuotekohtaisia menettelytapoja, eikä niin kuin olisi hyvä toimia. Sisältö voidaan tarkistaa sisäisin auditoinnein.

Koska standardin SFS-EN 60601-1-4 lähes kaikkien vaatimusten täyttyminen perustuu RMF:n ja RMS:n sisällön tarkastukseen ja osa näistä vaatimuksista voidaan täyttää yleisillä laatudokumenteilla, tulee tarkastuslistan joihinkin kohtiin useampikin dokumenttiviite.

Valmistajan tulisi määritellä RMF ja sen sisältö- ja tiedontuottovaatimukset jossain menetelmäkuvauksissaan, jolloin jokainen yrityksen vastuullinen henkilö täyttäisi RMF:n peruselementit samankaltaisesti vaikkakin sisältö voi vaihdella tuotekohtaisesti.

Esimerkiksi erilaiset ohjelmoinnin tyylioppaat, joissa valmistaja määrittelee ohjelmoinnissa käytettyjä työkaluja ja ohjelmointitapoja voidaan sisällyttää osaksi yleisiä laatutiedostoja. Tällöin on kuitenkin tuotekohtaisessa riskinhallintakansiossa muistettava tehdä ristiviittaus näihin tiedostoihin (esimerkiksi kohdat 52.204.3.1.6, 52.208.2 ks. tyyliopas SG4.1).

Liite C: Esimerkkejä mahdollisista terveydenhuollon tuotteeseen liittyvistä vaaroista ja sen alullepanevista syistä

Taulukossa C1 on pyritty kuvaamaan muutamia esimerkkejä mahdollisista tuotteeseen liittyvistä alullepanevista syistä.

Valmistaja voi laatia omalle tuotteelleen sopivan vastaavan taulukon, jossa kuvataan ko. tuotteeseen liittyviä vaaroja, alullepanevia syitä ja mahdollisia riskin valvontakeinoja. Tällaiseen taulukkoon voidaan liittää kaikki se kokemusperäinen tieto, joka valmistajalle on olemassa omasta tuotteestaan.

Taulukkoa voidaan käyttää lähdedokumenttina alustavan vaara-analyysin suoritukselle valmistajan tuotteelle.

Taulukko C 1. Vaaran aiheuttavia alullepanevia syitä.

Lähde	Kohde	Kuvaus	Valmistajan riskin arviointi	Valvonta
SAHCE-projekti	Käyttäjän virhetoiminto	Käyttäjä antaa virheellisen syötteen Käyttäjä tekee toiminnon aktivoinnin väärään aikaan Käyttäjä tekee väärän toiminnon Ei tiedä mitä pitää tehdä Ei tiedä miten pitää tehdä Ei tiedä miksi pitää tehdä Joudutaan muuttamaan asetuksia kesken toiminnon ja laite kykene muuttamaan toimintamuotoa tai toimintaa	Tähän sarakkeeseen sarakkeessa 'kuvaus' esitetyn asian riskin suuruuden arviointi ja mahdolliset viitedokumentit	Tähän sarakkeeseen se tieto, jolla pyritään eliminoimaan sarakkeessa 'kuvaus' esitetyn asian valvonta
SAHCE-projekti	Järjestelmän oletusarvot	Parametrikohtaiset hälytykset estetty Väärät hälytysluokitukset Väärät parametrikohtaiset mittausalueet, Asetettu käyttäjäryhmäkohtaiset asetukset, jolloin toinen käyttäjä ei osaa käyttää järjestelmää, Väärät suureiden mittayksiköt. Sopimattomat parametrivalinnat		
SAHCE-projekti	Järjestelmän ominaisuudet ja vikasietoisuus	Syötteen tarkastus puuttuu Sallitaan väärä toiminto Laskentavirhe Ohjelma harhautuu väärään ohjelmamoduuliin		

		Mekaaniset/elektroniset varmistukset puuttuu Käytetään piilevän vian sisältävää toimintoa Väärä tai tehoton algoritmi laskennassa Muuttujien ylivuoto Järjestelmä ei kykene laitevian seurauksena indikoimaan käyttäjälle olevansa kykenemätön suorittamaan toimintoa Päivitysversioiden yhteensopivuus vanhempiin versioihin ja jatkossa myös uudempiin versioihin Vikasietoisuus Ajoitus Alustus		
SAHCE-projekti	Ympäristö	Sähkökatkos Liian korkea lämpötila Sivulliset häiritsevät tai käyttävät järjestelmää Tietoliikenneverkon häiriöt Tietoliikenneverkon katkokset Rajapinnat (ohjelmamodulit, tietokannat, laitteistot jne.)		
SFS-EN 1441	Vaaralliset energiat	Sähkö Lämpö Mekaniikka Ionisoiva säteily Ei-ionisoiva säteily Sähkömagneettiset kentät Liikkuvat osat Ripustetut massat Potilasta kannattavien laitteiden rikkoontuminen Paine, akustinen paine, värinä		
SFS-EN 1441	Biologiset vaarat	Virheellinen ulostulo Bio-yhteensopimattomuus Virheellinen sanamuoto Virheellinen mittaus Riittämättömät varoitukset sivuvaikutuksista Riittämättömät varoitukset vaarasta kuten kertakäyttölaitteen uudelleenkäytöstä Kemiallisten (aineiden) myrkyllisyys (cross-)infection pyrogenicity hygieenisen turvallisuuden ylläpidon puute / kyvyttömyys		

SFS-EN 1441	Ympäristön aiheuttamat vaarat	Sähkömagneettinen säteily Riittämätön sähköteho tai jäähdytys Jäähdytyksen rajoitukset Todennäköisyys, että toimitaan kuvattujen ympäristöolosuhteiden ulkopuolella Yhteensopimattomuus toisiin laitteisiin Äkilliset mekaaniset vauriot Kontaminaatio johtuen jätteistä ja/tai tuotteen hävittämisestä		
SFS-EN 1441	Tuotteen käyttöön liittyvät vaarat	Riittämättömät merkinnät Riittämättömät tarvikespesifikaatiot Riittämättömät käyttöohjeet Liian monimutkaiset käyttöohjeet Olemattomat tai erilliset käyttöohjeet Ammattitaidoton käyttöhenkilökunta Kouluttamattoman henkilökunnan käyttö Inhimilliset virheet Virheelliset diagnoosit Virheellinen datansiirto Tulosten virhetulkinnat Virheelliset mittaukset Riittämättömät varoitukset sivuvaikutuksista		
SFS-EN 1441	Toiminnallisesta viasta, huollosta tai vanhenemisesta aiheutuvat vaarat	Riittämättömät suorituskykyominaisuudet aiottuun käyttöön Huoltotietojen puute Riittämätön huolto Riittämätön laitteen eliniän määritys Mekaanisen kestävyyden menetys Riittämätön pakkaus		

Liite D: Riskin esiintymistodennäköisyyden määrittelyn osatekijöitä

Taulukossa D1 on kuvattu standardissa SFS-EN 1050 kuvattuja osatekijöitä, joita voidaan käyttää apuna määriteltäessä esiintymistodennäköisyyttä. Osa tekijöistä on sovellettavissa sellaisenaan terveydenhuollon tuotteen riskien arviointiin.

Valmistaja voi laatia omalle tuotteelleen sopivan taulukon, jossa kuvataan ko. tuotteelle soveltuvia esiintymistodennäköisyyden osatekijöitä. Tällaiseen taulukkoon voidaan liittää kaikki se kokemusperäinen tieto, joka valmistajalle on olemassa omasta tuotteestaan.

Taulukkoa voidaan käyttää lähdedokumenttina alustavassa vaara-analyysissä havaitun riskin esiintymistodennäköisyyden määrittämiseen.

Taulukko D 1. Esiintymistodennäköisyyden osatekijät standardin SFS-EN 1050 mukaisesti.

Lähde	Kohde	Kuvaus
SFS-EN 1050	Altistumisen taajuus ja kesto	- Vaaravyöhykkeelle pääsyn tarve - Pääsytapa - Vaaravyöhykkeellä oloaika - Henkilöiden lukumäärä, joiden on päästävä vaaravyöhykkeelle - Vaaravyöhykkeelle menemisen taajuus
	Vaarallisen tapahtuman esiintymistodennäköisyys	- Luotettavuutta koskevat ja muut tilastolliset tiedot - Tapaturmatiedot - Tiedot terveyshaitoista - Riskin vertailu
	Vahingon vältettävyys ja rajoitettavuus	
	a) koneen käyttäjien vaikutuksesta	- Ammattitaitoiset henkilöt - Ammattitaidottomat henkilöt - Kone on miehittämätön
	b) vaarallisen tapahtuman ilmaantumisnopeus	- Äkillinen - Nopea - Hidas

	c) tietoisuus riskin olemassaolosta	- Yleistietoihin perustuen - Suoraan havaitsemalla - Varoitusmerkintöjen ja merkinantolaitteiden avulla
	d) inhimilliset mahdollisuudet välttää tai rajoittaa vahinkoa (esim. refleksit, notkeus, mahdollisuudet pelastautumiseen)	- Mahdollisia - Mahdollisia tietyissä olosuhteissa - Mahdottomia
	e) käytännön kokemusten ja tietojen avulla	- Kyseisestä koneesta - Vastaavasta koneesta - ei kokemusta

Liite E: Tuotteen riskianalyysi standardin SFS EN 1441 mukaan

E.1 Standardin tarkoitus

Eurooppalainen standardisointiorganisaatio CEN on laatinut ohjestandardin sille, kuinka terveydenhoidon tuotteen valmistajan tulisi varmentaa, todentaa, ja dokumentoida näiden vaatimusten täyttyminen.

Standardi SFS-EN 60601-1-4 edellyttää, että tuotteelle tehdään riskianalyysi, mutta ei määrittele sen suoritustapaa. SFS-EN 60601-1-4 viittaa EN 1441:een. Näin ollen riskianalyysin suorittamista käsitellään standardin SFS-EN 1441 pohjalta.

Tuotteiden riskianalyysi ja sen dokumentointi

Riskianalyysi on menettelytapa, jolla voidaan suorittaa haitallisten seuraamuksien sekä todennäköisyyden että laajuuden tunnistaminen. Analyysillä pyritään löytämään vastaukset seuraaviin kolmeen peruskysymykseen:

Mikä voi mennä väärin?	vaaran tunnistaminen
Kuinka todennäköisesti se tapahtuu?	taajuusanalyysi
Mitkä ovat seuraamukset?	seuraamusanalyysi

Standardi SFS-EN 1441 käsittelee riskienhallinasta vain tuotteen suunnitteluvaiheeseen liittyvien vaaratilanteiden ja niihin liittyvien riskien analyysin.

Standardi antaa ainoastaan menettelytavan tutkia käyttäen saatavilla olevaa informaatiota tuotteen turvallisuus yksilöimällä siihen tai sen käyttöön liittyvät vaarat ja arvioimalla todettujen vaarojen riskit. Erityisen hyödyllinen standardi on silloin, kun tuotteeseen liittyvät turvallisuus-, suorituskyky- tai rakennestandardit ovat puutteellisia tai niitä ei jostain syystä haluta käyttää. Mitään uutta analyysitekniikkaa standardi ei esitä vaan viittaa olemassa oleviin tekniikoihin. Näistä on valmistajan valittava tuotteeseensa tai käyttämäänsä teknologiaan soveltuva menettely.

On huomattava, että tämä standardi ei anna mitään hyväksyttävyystasoja riskeille. Riskin hyväksyttävyys, ollen merkittävä osatekijä riskinhallinnassa, jää valmistajan tai joissain tapauksissa myös loppukäyttäjän punnittavaksi.

Myös riskinhallinta sekä arviointi ja päätöksenteko koskien tietyn laitteen käytön indikaatioita ja kontraindikaatiota eivät kuulu standardin piiriin.

E.2 Proseduuri

Proseduuri kuvataan vuokaavion muodossa. Jos valmistajalla on käytössä laatujärjestelmä, on luontevaa sisällyttää riskianalyysi osaksi sitä. Riskianalyysin tekijöiden on oltava riittävästi koulutettuja tähän tehtävään tai valmistaja voi käyttää kolmannen osapuolen asiantuntijoita.

Osalle terveydenhuollon tuotteiden ominaisuuksista on jo saatettu tehdä riski-analyysi esim. mekaanisen turvallisuuden tarkastelun yhteydessä, mutta se ei kuitenkaan poista sitä vaatimusta, etteikö myös ohjelmistolle tule tehdä riskianalyysiä.

Proseduuri etenee taulukon E1 mukaisesti.

Taulukko E1. Standardin SFS-EN 1441 mukainen riskianalyysin vuokaavio.

ASKEL	TOIMENPIDE
2. Listaa arvioitavan laitteen kaikki ne ominaisuudet, jotka voisivat vaikuttaa turvallisuuteen ja, jos sovellettavissa, määrittele raja-arvot.	Standardi antaa avuksi joukon kysymyksiä listaa laadittaessa: a) Mikä on tuotteen aiottu käyttötarkoitus ja kuinka sitä käytetään? b) Onko tuote kontaktissa potilaaseen? c) Tuotteeseen liittyvät materiaalit ja komponentit d) Annetaanko potilaaseen tai otetaanko potilaasta mitään energiaa? e) Annetaanko potilaaseen tai otetaanko potilaasta mitään aineita? f) Käsitelläänkö tuotteella biologista materiaalia uudelleenkäyttöön? g) Toimitetaanko tuote steriilinä tai tarkoitettu käyttäjän steriloitavaksi? h) Onko tuotteen tarkoitus modifioida potilaan ympäristöä? i) Tehdäänkö mittauksia? j) Suorittaako tuote tulkintaa? k) Onko tuotteen tarkoitus säätää tai olla vuorovaikutuksessa toisen tuotteen tai lääkeaineen kanssa? l) Onko ei-toivottuja energian tai aineen ulostuloja? m) Sietääkö laite ympäristön vaikutuksia? n) Liittyykö tuotteeseen oleellisia kulutustavaroita/tarvikkeita? o) Onko huolto ja/tai kalibrointi tarpeen? p) Sisältääkö tuote ohjelmistoa? q) Onko tuotteella rajoitettu säilyvyysaika? r) Mahdolliset viivästyneen ja/tai pitkäaikaisen käytön vaikutukset? Mitkä mekaaniset voimat kohdistuvat tuotteeseen? t) Mitkä tekijät määrittelevät tuotteen eliniän? u) Onko tuote tarkoitettu kertakäyttöiseksi vai uudelleenkäytettäväksi?

3. Tunnista potentiaaliset vaarat	Tee lista potentiaalisista vaaroista, jotka voivat esiintyä normaalitilassa ja vikatilanteissa. Liitä mukaan kaikki tiedetyt vaarat, jotka ovat tavatut samanlaisen tuotteen yhteydessä.
4. Arvioi kunkin vaaran riskit	Tutki kunkin kohdassa 3. Listatun vaaran riski sekä normaali- että vikatilanteessa. Valitse sopiva analyysitekniikka (esim. FMEA, FTA, HAZOP). Riski tulisi ilmaista sellaisessa muodossa, että päätöksenteko riskinvalvonnasta on mahdollista. Systemaattisten vikojen todennäköisyyksiä ei voida arvioida ja näin ollen niiden riskikin jää määrittämättä. Näiden riskien hallitsemiseksi on käytettävä muita riskienhallintamenetelmiä. Sopivaa tietoa arvioinnin perustaksi voi saada esim. suorittamalla vertailuja; - Soveltuviin standardeihin, - Tieteelliseen dataan, - samanlaisista laitteista kerättyyn dataan, - Tutkimustuloksiin, - Kliiniseen aineistoon. Arviointiin tulisi liittää epävarmuus- tai herkkyysanalyysi, jotta valmistaja saisi paremman käsityksen riskin epävarmuudesta ja tämän vaikutuksesta turvallisuuteen.
5. Onko riski hyväksyttävissä?	Jos riski on hyväksyttävä, niin voidaan edetä askeleeseen 8. Jos tutkittavan vaaran riski ylittää hyväksyttävän tason, jatka askeleella 6. Jos vaara ylittää hyväksyttävyystason ainoastaan vikatilanteessa, vian ilmenemisen todennäköisyyttä tulisi analysoida. Seuraaviin kysymyksiin tulisi hakea vastauksia: - voiko käyttäjä havaita vian ennen kuin vaara toteutuu? - Voitaisiinko vika eliminoida tehokkaammalla valmistuskontrollilla tai ennakkohuollolla? - Kasvattaako väärinkäyttö vian todennäköisyyttä? - Voidaanko lisätä hälytyksiä?

6. Riskin vähentäminen	Kun riskiä on vähennetty riittävästi, etene askeleeseen 7. Jos jokin riski ei ole hyväksyttävissä, se tulisi redusoida hyväksyttävälle tasolle jollakin soveliaalla keinolla, esim.: - Välitön suojakeino (inherent safe design) - Välillinen suojakeino, esim. hälytyksiä (safeguarding) - Käyttäen "kuvailevaa" turvallisuutta, esim. antamalla käyttörajoituksia - rajoittamalla sovellutusaluetta, elinaikaa tai käyttöympäristöä - määrittelemällä uudelleen aiottu käyttötarkoitus.
7. Onko syntynyt muita vaaroja?	Onko uudelleensuunnittelun seurauksena muodostunut uusia vaaroja?
8. Ovatko kaikki potentiaaliset vaarat evaluoitu?	Jos ei niin palaa askeleeseen 4. Jos kyllä, niin jatka askeleella 9.
9. Laadi dokumentti analyysistä.	Dokumentin perusteella on mahdollista päättää, onko jäännösriski hyväksyttävissä huomioiden tuotteen aiotun sovellutuksen ja käytön. Dokumentti muodostaa osan tuotteen teknistä tiedostoa ja sitä säilytetään ja päivitetään lainsäädännön mukaisesti. Sisältö on seuraavanlainen: - tuotteen täydellinen kuvaus ja identifiointi - analyysin suorittaja - analyysin päivitys - Yhteenveto ja päätelmät - Kohde ja laajuus - Rajoitukset, olettamukset ja näiden perustelut - Analyysin metodiikka - listan yksilöidyistä mahdollisista vaaroista - kuhunkin vaaraan liittyvän riskin arviointi - riskin vähimmäistämistapa, kun tarvittu käyttää - Käytetyt mallit, olettamukset ja pätevyys - Käytetty tieto lähteineen - Riskinarvioinnin tulokset - Herkkyys- ja epävarmuusanalyysi - Keskustelua tuloksista - Viitteet

	Jos riskit muuttuvat ajan myötä, riskianalyysiä päivitetään vastaavasti. Nopeasti muuttuva teknologia voi eliminoida, kasvattaa tai vähentää todettuja riskejä. Uusia riskejä saattaa ilmetä analyysin suorituksen jälkeen. Tarkastelun on katettava myös ohjelmiston huollosta ja muusta käsittelystä vastaavat henkilöt, myös muut kuin itse tuotteesta johtuvat vaarat on sisällytettävä analyysiin (esim. energian syötöstä tai käytöstä johtuvat vaarat). Arvioinnin, onko ohjelmiston riski hyväksyttävissä rajoissa vai ei, tulisi tapahtua punnitsemalla riskiä lääketieteelliseen tarpeeseen ja diagnostisiin etuihin nähden. Tuotetta tulee arvioida lähtökohtana kilpailevat tuotteet, vaihtoehtoiset diagnostiset menetelmät ja lääketieteelliset teoriat ja tutkimustulokset. Kunkin vaaran osalta seuraavia seikkoja tulisi tarkastella: - luottamus analyyttiseen tulokseen - uskottavuustarkastukset - valvontakeinojen saatavuus ja käyttö - poikkeamien/virheiden havaittavuus - käyttötilanteet

Liite F: Keskeisiä kysymyksiä validoinnin valmisteluun

Taulukossa F1 on pyritty kuvaamaan kysymysmuotoon niitä seikkoja, jotka vaikuttavat terveydenhuollon tuotteen validoinnin suorittamiseen.

Valmistajan voi laatia omalle tuotteelleen ja tuotannolleen vastaavan taulukon, joka helpottaa tuotekohtaisen validointisuunnitelman ja validoinnin valmistelua.

Taulukko F 1. Validoinnin valmisteluun liittyviä kysymyksiä.

Kohde	Toiminto	Tarkka kuvaus	Soveltuvuus
Tuotteen validointi	Suunnittelun lähtötietojen asettaminen	Onko käyttäjän tarpeet riittävän selvät, jota niistä voidaan tuottaa teknisiä eritelmiä? Onko tuotteen turvallisuuteen liittyvät vaatimukset ja ominaisuudet osana lähtötietoja? Onko tiedostettu standardi- ja viranomaisvaatimukset? Onko sopimuskatselmukset, niiden sisältö ja tavoitteet dokumentoitu? Onko suunnittelukatselmukset ja niiden tavoitteet määritelty? Onko markkina-alueiden vaikutus esimerkiksi lokalisointeihin määritelty? Onko alustava riskienhallintasuunnitelma ja sen kattavuus määritelty? Onko alustavat verifiointi- ja validointisuunnitelmat sekä sovellettava elinkaarimalli määritelty?	Tähän sarakkeeseen voidaan lisätä tarvittaessa arvio validoinnin kohdistuvuudesta ja laajuudesta ja mahdolliset viite- tai tukidokumentit tai arvio ko. kohdan soveltuvuudesta
Prosessin validointi	Riskienhallintaprosessi	Onko käytettävissä riittävä määrä riskienhallintaprosessin edellyttämiä työkaluja? Onko alustava riskianalyysi tunnistanut riskejä, joiden valvonta edellyttää tiettyjä toiminnallisia vaatimuksia? Onko yllämainitut vaatimukset siirtyneet osaksi vaatimusspesifikaatiota? Tukeeko analyysin suorittamista menetelmäohjeet ja valmiit dokumenttipohjat? Onko analyysi raportoitu asianmukaisesti? Onko riskien vaikuttavuus ja merkittävyys arvioitu? Onko riskien hyväksyntä tehty asianmukaisesti? Onko analyyseissä käsitelty vikaantumismallit? Onko käytetyt suojakeinot riittäviä? Suojataanko käytetyt COTS-komponentit? Kattaako riskienhallintaprosessi alihankintaohjelmiston?	

Prosessin validointi	Ohjelmisto-tuotannon vaihejakomalli	Onko tuotteen suunnittelussa käytetty elinkaarimalli määritelty? Onko vaiheiden tehtävät määritelty? Onko kullekin vaiheelle määritelty riittävät tulos- ja lähtötiedot? Onko dokumentointivaatimukset määritelty? Onko vaiheen päätös määritelty? Onko muutostenhallinta määritelty?	
	Dokumentointi	Onko mukana seuraavat asiakirjat riittävät? Onko jäännösriskit kuvattu? Vaikuttaako ohjelmaversiot asiakirjojen sisältöön? Onko näytöt, hälytykset ja avusteet kuvattu riittävän selkeästi? Onko aiottu käyttötarkoitus ja rajoitukset kuvattu? Riittääkö ohjeistus kaikkien toimintojen tekemiseksi turvallisesti ja asianmukaisesti? Selviääkö testi- verifiointi- ja validointiraporteista raportin kattavuus, hyväksyntä- ja hylkäysperusteet, tekijä(t) ja hyväksyjä(t)? Mikä on raporttien ja dokumenttien suhde vaatimuksiin? Onko ne riittävät arkkitehtuurin, spesifikaatioiden, suunnittelun, implementoinnin ja testauksen osalta? Onko ongelmat ja korjaavat toimenpiteet raportoitu? Onko Level Of Concern määritelty sekä hyväksytty ja onko se jäljitettävästi raportoitu? Voidaanko suunnittelun aikaiset toiminnot jäljittää dokumenttien perusteella? Onko tuotteen arkkitehtuuri kuvattu riittävästi? Mistä dokumenteista järjestelmäspesifikaatio on kehittynyt? Tunnistetaanko dokumenteista kaikki tuotteessa käytetyt ohjelmistomodulit? Tunnistetaanko dokumenteista käytetyt COTS-komponentit? Onko mahdolliset alihankkijoiden toimittamat modulit dokumentoitu kaikilta osilta riittävästi? Onko elinkaarimallin edellyttämät suunnitteludokumentit? Täyttääkö tuotekohtainen dokumentointi viranomaisvaatimukset?	
Prosessin validointi	Ylläpito	Onko tuotannollistamissuunnitelma tehty? Salliiko ohjelmisto muunneltavuuden ja päivityksen? Sallitaanko ohjelmiston uudelleenkäytettävyys? Onko määritelty prosessit takaisinkutsuille?	

Tukiprosessien validointi	Välineet ja menetelmät	Onko projektinhallinta määritelty siten, että se tukee suunnitteluprojektia hallitsemaan taloudelliset ja laadulliset tavoitteet sen läpiviemiseksi? Tuetaanko suunnitteluprojektia menetelmäohjeilla ja valmiilla lomakepohjilla? Onko käytetyt työkalut ja menetelmät kelpuutettu käyttöön? Onko suunnittelun elinkaarimalli määritelty? Onko käytössä henkilöstön koulutusrekisteri ja -suunnitelmat? Soveltuvatko käytetyt teknologiat käyttöön? Tukeeko koulutusrekisteri ja koulutussuunnitelma valmistajan käyttämiä teknologioihin, kuten ohjelmointitekniikat, riskianalyysitekniikat ja sovellusaluekoulutus? *HUOM: henkilö ei välttämättä tarvitse aina erityistä koulutusta tekemiinsä tehtäviin. Pätevyys voidaan hoitaa myös pitkäaikaisella kokemuksella vastaavista tehtävistä. Henkilön pätevyyden päättäminen jää aina valmistajan vastuulle.* Kykenevätkö olemassa olevat menetelmäohjeistukset ja tyylioppaat tukemaan ohjelmistotuotantoa jatkuvaan toimivuuteen, toistettavaan tuotantoon ja kykyyn hallita muutoksia? Määritelläänkö verifioinnille yleiset ohjeet, jotka sisältävät verifiointisuunnitelman laatimisen, suorittamisen, hyväksynnän, kattavuuden ja raportoinnin? Määritelläänkö validoinnille yleiset ohjeet, jotka sisältävät verifiointisuunnitelman laatimisen, suorittamisen, hyväksynnän, kattavuuden ja raportoinnin?	

Liite G: Lisätiedot

FDA:n alustavia kysymyksiä ja näkökohtia validoinnin valmisteluun

Alla olevassa luettelossa on kuvattu FDA:n (FDA 1997) alustavia kysymyksiä validoinnin valmisteluun:

- What software quality factors (e.g., reliability, maintainability, usability, etc.) are important to the validation process and how will those factors be evaluated?
- Are there enough staff and facilities resources to conduct the validation?
- What part will the hazard management function play in the software validation process?
- How will off-the-shelf (OTS) software be validated? What are the specific requirements for use of the OTS software? What are the risks and benefits of OTS versus contracted or in-house developed software? What information (description of the software quality assurance program, validation techniques, documentation of validation, "bug lists", etc.) is available from the OTS vendor to help in validating use of the software in the device, or to produce the device? Will the OTS vendor allow an audit of their validation activities? Is the OTS software suitable for its intended use, given the availability of necessary validation information? What level of black-box testing is required to demonstrate that the OTS software is suitable for its intended use? What impact will these factors have on contract negotiations and vendor selection?
- How will contracted software be validated? In addition to the issues above for OTS software, who will control the source code and documentation, and what role will the contractor play in validation of the contracted software.
- For an OTS software automated process or quality system function, will the output of the process or function be fully verified in every case against specifications? If so, the process or function is not dependent upon proper operation of the software and "verification by output" may be sufficient. However, if the output will not be fully verified against the specification, then the software must be validated for its intended use [Ref: 21 CFR 820.70(i)].
- What human factors need to be addressed? One of the most persistent and critical problems encountered by FDA is user error induced by overly complex or counter-intuitive design. Frequently, the design of the software is a factor in such user errors. Human factors engineering should be woven into the entire design and development process, including the device design concept, requirements, analyses, and tests. Safety and usability issues should be considered when developing flowcharts, state diagrams, prototyping tools, and test plans. Also task and function analyses, hazard analyses, prototype tests and reviews, and full usability tests should be performed. Participants from the user population should be included when applying these methodologies.

Teknisen tiedoston rakenne ja sisältö

Tekninen tiedosto on keino, jolla valmistaja voi varmentaa, että hänen tuotteensa on standardien ja direktiivien vaatimusten mukainen sekä kykenee osoittamaan tuotteensa vaatimustenmukaisuuden ilmoitetulle laitokselle. Teknisen tiedoston systemaattinen rakentaminen helpottaa oleellisesti valmistajaa ja tarvittaessa ilmoitettua laitosta arvioitaessa tuotteen yhdenmukaisuutta.

Teknisen tiedoston dokumentoinnin tarve on riippumaton valmistajan käytössä olevista EU:n vaatimustenmukaisuuden varmistusmenetelmistä ns. "reiteistä". Vaikka valmistajalla olisi mahdollisuus tuoteluokasta riippuen käyttää yksinkertaisinta "reittiä" eli vaatimustenmukaisuusvakuutusta (sis. suunnittelun ja tuotannon valvonta), niin tällöinkin suunniteltu ohjelma tuotedokumentaation tuottamiseksi ja hallitsemiseksi eräänlaisena laatujärjestelmänä varmistaa ja helpottaa vaadittujen toimenpiteiden toteutumista.

Tekninen tiedosto ja sen tuotto on mielekästä jakaa kahteen osaan: tuotteen suunnittelua koskevat ja tuotteen tuotantoa koskevat tiedostot. Syntyvät tiedostot, niiden tuotantotapavastuu ja ajan tasalla pito ovat luontevasti osa yrityksen laatujärjestelmää. Näistä käytetään usein nimityksiä. Device history records ja Device master record. Tuotteen suunnittelun lopputulosta koskeva tiedostoon (DMR) sisältyy:

- tuotteen yleinen kuvaus mukaan lukien suunnitellut tuotevaihtoehdot
- suunnittelupiirustukset, valmistusmenetelmät, kaaviot osista osakokoonpanoista, piireistä jne.
- kuvaukset ja selitykset mainittujen piirustusten ja kaavioiden ja tuotteen toiminnan ymmärtämiseksi
- luettelo käytetyistä standardeista tai standardikohdista sekä kuvaukset ratkaisuista, joita on tarvinnut tehdä kaikkien soveltuvien olennaisten vaatimusten täyttymiseksi sekä riskianalyysin tulokset
- kuvaus steriileinä markkinoille saatettavien tuotteiden sterilointimenetelmistä
- tulokset suunnittelulaskelmista ja tarkastustuloksista;
- näyttö, että tuote käytettynä yhdessä toisen tuotteen kanssa täyttää olennaiset vaatimukset
- testausselosteet ja tarvittaessa tuotetta koskevat kliiniset tiedot
- merkinnät ja käyttöohjeet.

Terveydenhuollon tuotteen luokittelu

EU:n lainsäädäntö on antanut terveydenhuollon laitteille ja tarvikkeille yhtenäiset säädökset koskien tuotteiden ja niiden valmistuksen vaatimuksia ja näiden vaatimusten toteutumisen todentamista.

Taloudellisista ja käytännön syistä ei ole mielekästä vaatia jokaiselta terveydenhuollon taitteelta ja tarvikkeelta kaikkein ankarinta vaatimustenmukaisuuden arviointimenetelmän käyttöä. Tuotteen riskin mukaan porrastettu valvontajärjestelmä on katsottu soveltuvammaksi. Tämän toteuttamiseksi on luotu tuotteiden luokittelujärjestelmä, joka ohjaa tuotteen oikeaan vaatimustenmukaisuuden varmennusmenetelmään.

Terveydenhuollon tuotteet ja tarvikkeet luokitellaan direktiivin 93/42/EEC (Medical Device Directive; MDD1993) mukaan luokkiin I, IIa, IIb ja III. Luokittelusäännöt on annettu direktiivin liitteessä IX. Lähtökohtana on valmistajan tuotteelle määrittelemä käyttötarkoitus ja tähän että tuotteeseen liittyvät riskit.

EU:n luokitteluajatus perustuu tuotteen käyttöön ja mahdollisiin vikaantumisiin liittyviin potentiaalisiin vaaroihin. Näin on saatu eri tavoin yhdisteltävien kriteerioiden määrä pieneksi: kehokontaktin kesto, invasiivisuuden aste ja paikallinen vs. systeemin vaikutus.

Luokittelusäännöt perustuvat paljolti ihmisen haavoittuvuuteen tuotetta käytettäessä suunnitellulla tavalla. Säännöstö ottaa huomioon myös tuotteiden toimimattomuuden seuraamukset. Yhdessä käytettävät tuotteet on luokiteltava kuin erilliset tuotteet. Tämä koskee myös tarvikkeita. Tuotteeseen liittyvän ohjelmiston riskiluokka on aina sama kuin ko. tuotteen. Käytännössä riskiluokittelu johtaa yhä tarkempaan tuotteen ja sen tarvikkeiden ja oheislaitteiden spesifiointiin kokonaisvastuussa olevan valmistajan toimesta.

Peruskriteerit direktiivissä MDD (1993) olevalle luokittelulle ovat seuraavat:

- tuotteen kosketus tai vuorovaikutus ihmiskehon kanssa
- kosketus vaurioituneeseen ihoon
- tuotteen invasiivisuus
- implantointi kehoon
- kosketus elintärkeisiin elimiin (keskushermosto, keskeisverenkierto)
- energian tai aineen siirto kehoon tai keholle.

Esitettyjen sääntöjen perusteella voidaan antaa seuraava karkea kuva luokittelusta:

- **Luokka I**:

- tuotteet, jotka eivät ole yhteydessä kehoon, uudelleen käytettävät kirurgiset instrumentit, sidemateriaalit, diagnostiset ja terapeuttiset laitteet, jotka eivät syötä energiaa,

- **Luokka IIa tai IIb:**

- invasiiviset tai implantoitavat tuotteet tai tuotteet, jotka ovat vuorovaikutuksessa ihmiskehon kanssa,

- hammasimplantit, ei-invasiiviset verenkäsittelylaitteet, aktiiviset diagnostiset ja terapeuttiset laitteet, jotka syöttävät energiaa,

- **Luokka III:**

- invasiiviset tuotteet, jotka ovat tekemisissä kehon elintärkeiden elimien kanssa tai joilla on biologinen vaikutus tai absorboituvat tai kemiallisesti muuttuvat kehossa.

Valmistajan tullee kyetä luokittelemaan tuotteensa johonkin näistä luokista ja joillain tuotteilla tämä saattaa olla epäselvää tai vaikeaa. Apua voi kysyä ilmoitetulta laitokselta (VTT Automaatio) tai alan viranomaiselta (Lääkelaitos). Valmistajan on kuitenkin itse tehtävä luokittelu. Lisäavuksi on Euroopan yhteisö julkaisut lääkintälaitteiden luokitteluohjeen Guidelines to the Classification of Medical Devices: (March 1996, 5th draft), jossa kuvataan niitä sääntöjä ja valintakriteereitä, joiden avulla valmistaja voi luokitella tuotteensa. Ohjeessa on annettu tyypillisiä luokitteluesimerkkejä asian havainnollistamiseksi.

Luokittelun vaikutus vaatimustenmukaisuuden varmentamiseen ilmenee seuraavasta taulukosta G1.

Taulukko G1. Terveydenhuollon tuotteen luokittelu.

Varmennus	TUOTELUOKKA					
	I	Isteriili	Imittaus	IIa	Iib	III
Laadunvarmistusjärj. + sunnittelutark.						X
Laadunvarmistusjärj.				X	X	
Tyyppitarkastus					X	X
Yksilö/erätarkastus		X	X	X	X	X
Tuotannonvalvonta		X	X	X	X	X
Lopputarkastus		X	X	X	X	
Valmistajan vakuutus	X	X	X	X		

Julkaisija

Vuorimiehentie 5, PL 2000, 02044 VTT
Puh. (09) 4561
Faksi (09) 456 4374

Julkaisun sarja, numero ja raporttikoodi

VTT Tiedotteita 2150
VTT–TIED–2150

Tekijä(t)
Pöyhönen, Ilpo, Kylmälä, Kaarle, Harju, Hannu, Kemppainen-Kajola, Pia, Kuhakoski, Kalle, Spankie, Greig & Ventä, Olli

Nimeke

Vaatimukset ohjelmistoa sisältäville lääkintälaitteille
Hallinta ja menetelmät vaatimustenmukaisuuden osoittamiseksi

Tiivistelmä

Terveydenhuollossa käytettävien laitteiden ja järjestelmien suorituskyky ja monimutkaisuus lisääntyvät vuosi vuodelta. Osaltaan muutokset johtuvat teknologisista muutoksista, mutta myös potilaan hoitomenetelmät kehittyvät ja lisäävät muutospaineita laitteiden suorituskyvylle ja turvallisuudelle. Hyvin usein myös hoito- ja tutkimusmenetelmien tukena käytettävät järjestelmät sisältävät tietokoneen tai palvelimen, käyttöjärjestelmän ja laajan joukon erilaisia sovellusohjelmia ja ajureita.

Näiden teknologisten ja itse hoitomenetelmissä tapahtuneiden muutosten seurauksena ohjelmistojen osuus lääkintälaitteissa ja lääkintälaitejärjestelmissä on kasvamassa. Lääkintälaitteet ja järjestelmät on ennen markkinoille saattamista hyväksytettävä eri markkina-alueilla voimaan saatettujen säädösten mukaisesti. Mikäli markkinoille saattamisen aikana huomataan, että tuote ei täytä sille asetettuja standardien tai säädösten vaatimuksia, niin tästä voi aiheutua huomattavia lisäkustannuksia yritykselle johtuen uudesta tuotekehitysjaksoista sekä uusista varmennustesteistä. Viivästyksillä, mahdollisista jälkitoimenpiteillä tai jopa markkinoilta poisvetämisellä on myös kielteinen vaikutus ostajan mielikuvaan yrityksen ja sen tuotteiden luotettavuudesta ja laadusta.

Koska ohjelmistopohjaisten laitteiden ja järjestelmien luotettavuuden, turvallisuuden ja suorituskyvyn, s.o. vaatimustenmukaisuuden osoittaminen on itse valmiista tuotteessa vaikeaa, tässä julkaisussa on erityisesti käsitelty arviointia tuotantoprosessissa.

Tässä julkaisussa yhdistetään EU:n ja FDA:n asettamat vaatimukset lääkintälaitteiden ohjelmistoille sekä esitetään vaatimustenmukaisuuden osoittamismalli. Julkaisussa käsitellään laajasti myös riskienhallintaprosessia ja -menetelmiä, joilla pystytään riskien tunnistamisen ja arvioimisen ohella myös tehostamaan kehitysprosessia ja varmistamaan tuotteen markkinoillesaattamisen onnistuminen. Julkaisu on tarkoitettu lääkintälaitteiden valmistajille ohjeistamaan ohjelmistotuotannon eri osa-alueita tavoitteena yhdistää tuoteriskien hallinta ja riskianalyysit kiinteäksi osaksi suunnittelua. Tällöin suunnitteluprosessi tuottaa turvallisempia ja luotettavampia tuotteita lyhentäen tuotekehitysjaksoja. Lisäksi valmistaja kykenee osoittamaan viranomaisille standardien ja direktiivien vaatimuksenmukaisuuden tehokkaammin.

Avainsanat
medical device software, medical devices, medical systems, risk management, risk analysis, manufacturing process assessment

Toimintayksikkö
VTT Tuotteet ja tuotanto, Tekniikankatu 1, PL 1306, 33101 TAMPERE

ISBN	Projektinumero
951–38–6060–4 (nid.) 951–38–6061–2 (URL: http://www.inf.vtt.fi/pdf/)	G2SU00140

Julkaisuaika	Kieli	Sivuja	Hinta
Elokuu 2002	Suomi, engl. abstr.	135 s. + liitt. 40 s.	D

Projektin nimi	Toimeksiantaja(t)
	VTT, Tekes, Instrumentarium Imaging Oy, Orion Soredex Oy, Philips Medical Systems Finland Oy

Avainnimeke ja ISSN	Myynti:
VTT Tiedotteita – Research Notes 1235–0605 (nid.) 1455–0865 (URL: http://www.inf.vtt.fi/pdf/)	VTT Tietopalvelu PL 2000, 02044 VTT Puh. (09) 456 4404 Faksi (09) 456 4374

Published by

Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
Phone internat. +358 9 4561
Fax +358 9 456 4374

Series title, number and report code of publication

VTT Research Notes 2150
VTT–TIED–2150

Author(s)
Pöyhönen, Ilpo, Kylmälä, Kaarle, Harju, Hannu, Kemppainen-Kajola, Pia, Kuhakoski, Kalle, Spankie, Greig & Ventä, Olli

Title

Requirements for medical device software

Abstract

The performance and complexity of devices and systems used in health care increase year by year. Changes are partly result from technological changes but also methods of treatment are developing and adding pressure for changes of performance and safety of devices. Regularly, the systems used to support methods of treatment and examination include a computer or a client, an operating system and a large amount of different kind of application programs and drivers.

Because of the changes in technology and methods of treatment, the portion of software in medical devices and systems is increasing. Before access to the market, medical devices and systems have to be accepted against promulgation of the regulations in different market areas. If during the access to the market, it is detected that the product does not fulfil the issued requirements of standards and regulations, some remarkable additional cost for the company might be due to the new manufacturing processes and new assurance tests.

Because the demonstration of reliability, safety and performance of the product itself is very difficult, this publication concerns assessment of manufacturing processes.

In this publication, EU and FDA requirements for medical device software are integrated and a model for demonstration of compliance with the requirements is introduced. In addition, risk management processes and methods are largely concerned. With these processes and methods one is able, in addition to identify and estimate risks, also improve the development processes and assure the succeed access to the market.

The publication is intended to the manufacturers of medical devices to guide different sectors of software engineering aim to integrate product risk management and risk analysis to the compact part of design. In this way the design process will produce more reliable and safer products at a shorter product development time. In addition, the manufacturer is able to effectively demonstrate the compliance with the requirements of standards and directives for the authorities.

Keywords
medical device software, medical devices, medical systems, risk management, risk analysis, manufacturing process assessment

Activity unit
VTT Industrial systems, Tekniikankatu 1, P.O.Box 1306, FIN–33101

ISBN	Project number
951–38–6060–4 (soft back ed.) 951–38–6061–2 (URL: http://www.inf.vtt.fi/pdf/)	G2SU00140

Date	Language	Pages	Price
August 2002	Finnish, Engl. abstr.	135 p. + app. 40 p.	D

Name of project	Commissioned by
	VTT, Tekes, Instrumentarium Imaging Oy, Orion Soredex Oy, Philips Medical Systems Finland Oy

Series title and ISSN	Sold by
VTT Tiedotteita — Research Notes 1235–0605 (soft back edition) 1455–0865 (URL: http://www.inf.vtt.fi/pdf/)	VTT Information Service P.O.Box 2000, FIN–02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374

VTT TIEDOTTEITA – RESEARCH NOTES

VTT TUOTTEET JA TUOTANTO – VTT INDUSTRIELLA SYSTEM – VTT INDUSTRIAL SYSTEMS

2050 Kotikunnas, Erkki & Heino, Perttu. Turvallisen prosessilaitoksen suunnittelu. STOPHAZ-projektissa syntyneet työkalut. 2000. 44 s.

2058 Konola, Jari. Kunnossapidon tietojärjestelmä käyttövarmuustiedon lähteenä Suomen paperi- ja selluteollisuudessa. 2000. 25 s.

2061 Välisalo, Tero & Rouhiainen, Veikko. Luotettavuusjohtaminen työkoneteollisuudessa. 2000. 43 s. + liitt. 15 s.

2063 Tonteri, Hannele, Vatanen, Saija & Kuuva, Markku. Työkoneiden käytön jälkeisen käsittelyn suunnittelu. 2000. 32 s. + liitt. 4 s.

2064 Tonteri, Hannele, Vatanen, Saija & Kuuva, Markku. Design for end-of-life treatment of work machines. 2000. 32 p. + app. 4 p.

2066 Harju, Hannu. Ohjelmiston luotettavuuden kvalitatiivinen arviointi. 2000. 111 s.

2067 Baumont, Geneviève, Wahlström, Björn, Solá, Rosario, Williams, Jeremy, Frischknecht, Albert, Wilpert, Bernhard & Rollenhagen, Carl. Organisational factors. Their definition and influence on nuclear safety. Final raport. 2000. 65 p.

2077 Solin, Jussi (ed.). Plant life management (XVO). Report 1999. 2001. 68 p. + app. 3 p.

2098 Parikka, Risto, Ahlroos, Tiina, Halme, Jari, Miettinen, Juha, Salmenperä, Pekka, Lahdelma, Sulo, Kananen, Markku & Kantola, Petteri. Monitorointi ja diagnostiikka. 2001. 55 s.

2115 Luoma, Tuija, Mattila, Inga, Nurmi, Salme, Ilmén, Raija, Heikkilä, Pirjo, Salonen, Riitta, Sikiö, Teija, Lehtonen, Mari & Anttonen, Hannu. Elektroniikka- ja kemianteollisuuden suojavaatteet. Sähköstaattiset ominaisuudet ja käyttömukavuus. 2001. 92 s. + liitt. 12 s.

2117 Malm, Timo, Hämäläinen, Vesa & Kivipuro, Maarit. Paperiteollisuuden rullankäsittelyn turvallisuus ja luotettavuus. 2001. 68 s. + liitt. 12 s.

2140 Reiman, Teemu & Oedewald, Pia. The assessment of organisational culture. A methodological study. 2002. 42 p.

2148 Aaltonen, Pertti, Bojinov, Martin, Helin, Mika, Kinnunen, Petri, Laitinen, Timo, Muttilainen, Erkki, Mäkelä, Kari, Reinvall, Anneli, Saario, Timo & Toivonen, Aki. Facts and views on the role of anionic impurities, crack tip chemistry and oxide films in environmentally assisted cracking. 2002. 68 p. + app. 21 p.

2149 Hemilä, Jukka. Information technologies for value network integration. 2002. 97 p. + app. 1 p.

2150 Pöyhönen, Ilpo, Kylmälä, Kaarle, Harju, Hannu, Kemppainen-Kajola, Pia, Kuhakoski, Kalle, Spankie, Greig & Ventä, Olli. Vaatimukset ohjelmistoa sisältäville lääkintälaitteille. Hallinta ja menetelmät vaatimustenmukaisuuden osoittamiseksi. 2002. 135 s. + liitt. 40 s.

2151 Harju, Hannu. Kustannustehokas ohjelmiston luotettavuuden suunnittelu ja arviointi. Osa 1. 2002. 114 s. + liitt. 15 s.

Tätä julkaisua myy
VTT TIETOPALVELU
PL 2000
02044 VTT
Puh. (09) 456 4404
Faksi (09) 456 4374

Denna publikation säljs av
VTT INFORMATIONSTJÄNST
PB 2000
02044 VTT
Tel. (09) 456 4404
Fax (09) 456 4374

This publication is available from
VTT INFORMATION SERVICE
P.O.Box 2000
FIN–02044 VTT, Finland
Phone internat. + 358 9 456 4404
Fax + 358 9 456 4374

ISBN 951–38–6060–4 (nid.)
ISSN 1235–0605 (nid.)
ISBN 951–38–6061–2 (URL: http://www.inf.vtt.fi/pdf/)
ISSN 1455–0865 (URL: http://www.inf.vtt.fi/pdf/)