IEC 61508-3 sisältö ja rakenne

Matti Vuori, Tampereen teknillinen yliopisto

Huom! Esityksessä käytetyt standardin suomenkieliset tekstit, termit ja kaaviot ovat standardin käännöksen vielä hyväksymättömästä versiosta ja saattavat siten osin muuttua standardin julkaistavassa versioon.

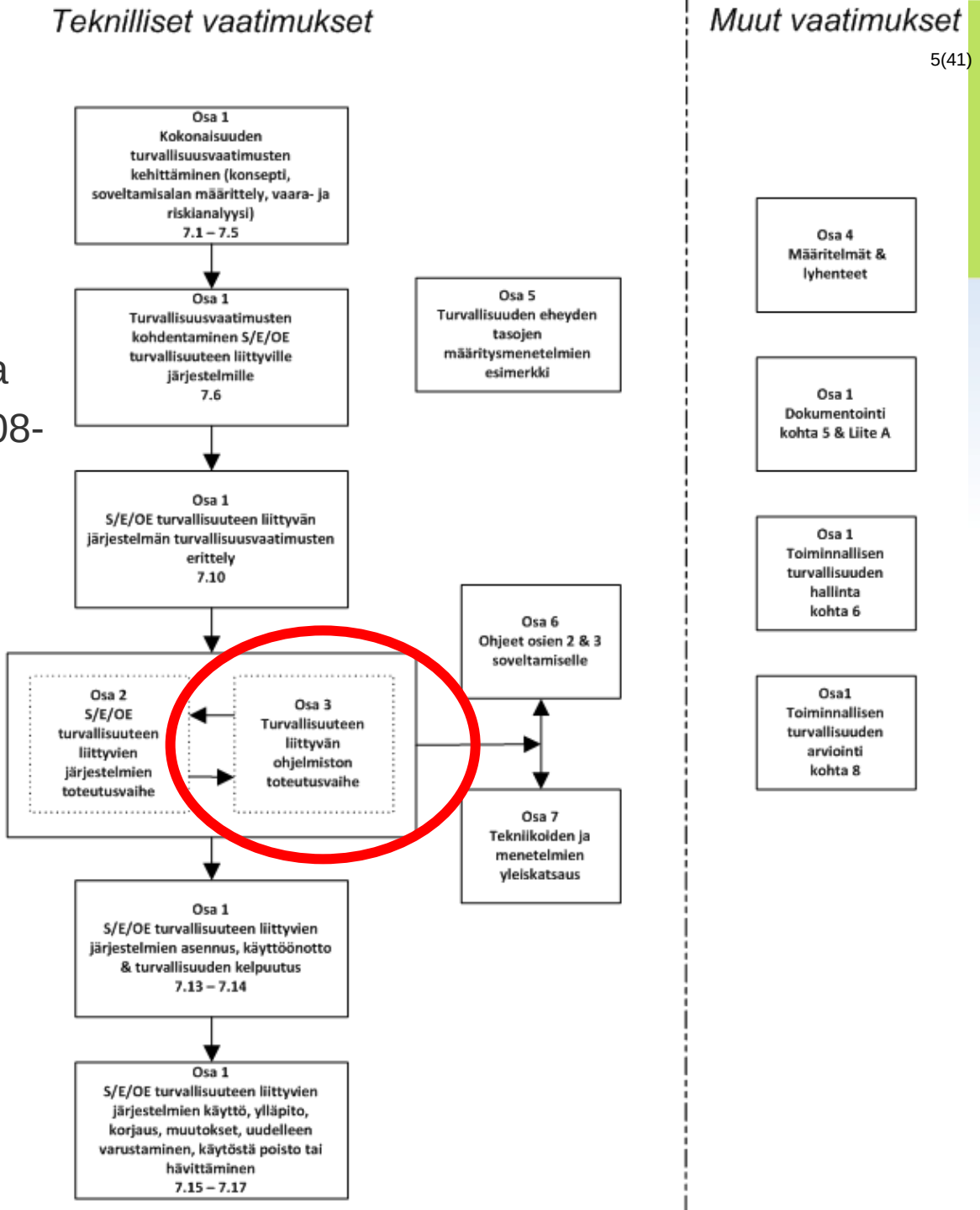
Sisällysluettelo 1/3

Sisällysluettelo 2/3

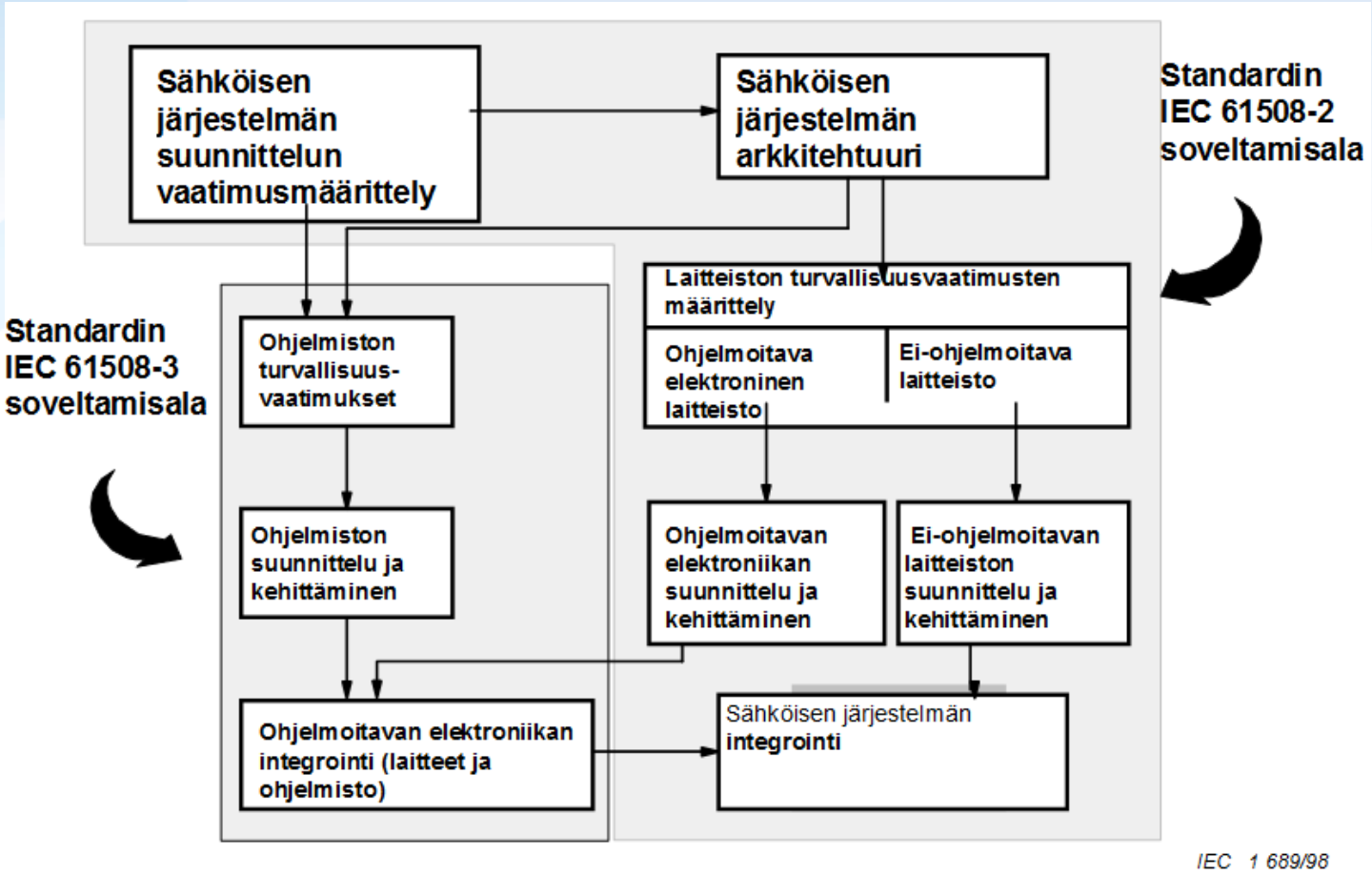
Sisällysluettelo 3/3

Asema koko standardisarjassa

- Tiukasti kiinni 61508-2:ssa
- Menetelmäopastusta 61508-7:ssä
- Ymmärrettävä 61508-1:n konteksti



Tiukasti kiinni ohjelmoitavan elektroniikan kehityksen kanssa



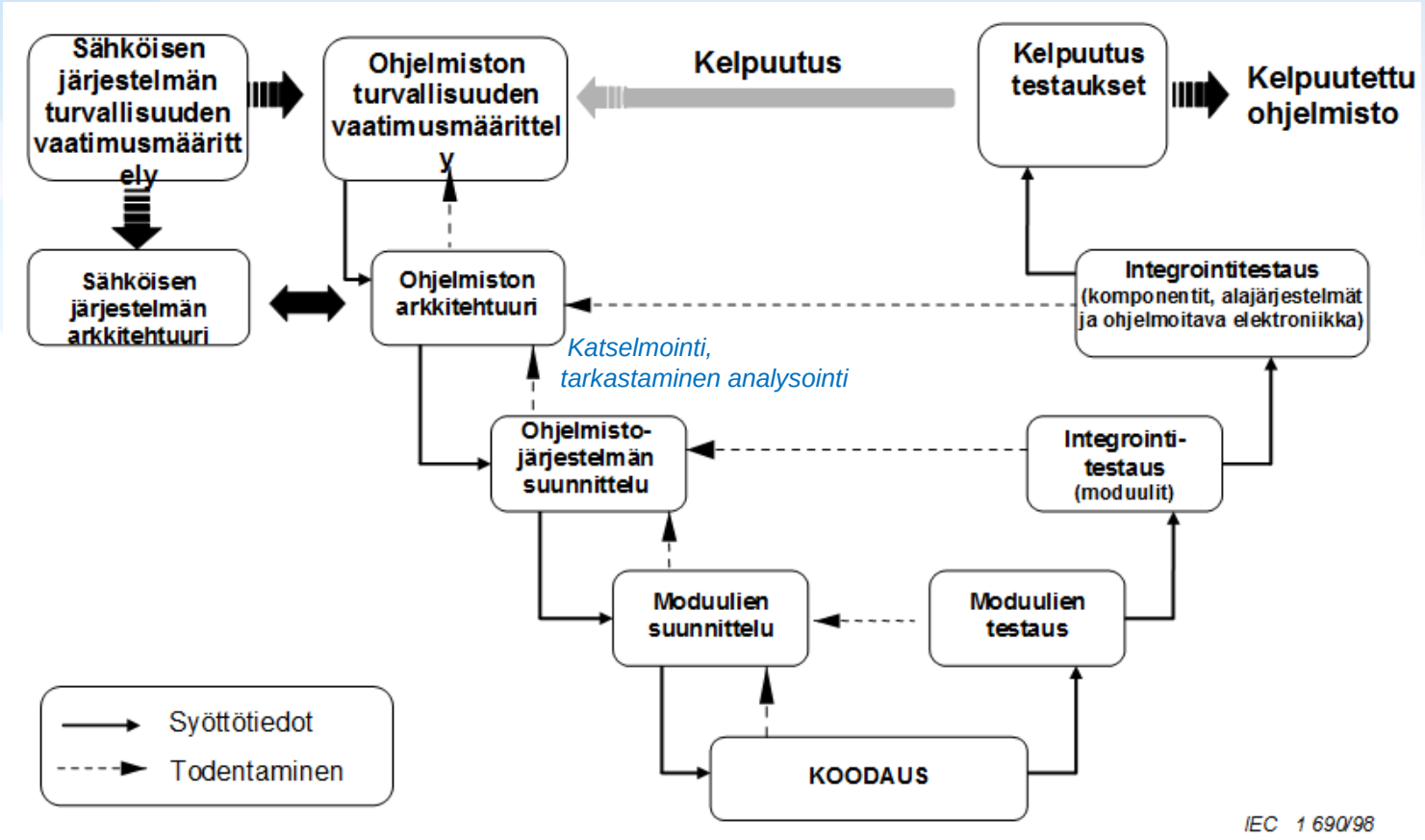
Soveltamisen perusfilosofia

- Perustuu turvallisuuden elinkaari –ajatteluun: turvallisuusvaatimuksista systeemin validointiin ja käyttöjaksoon.
- Ei riskien kvantifiointia, vaan toimia systemaattisten ohjelmistovirheiden estämiseen.
- Ohjelmistolle allokoidut turvallisuusvaatimukset ja SIL-taso ovat lähtötietoina.
- Kehyksenä traditionaalinen näkemys ohjelmistokehityksen prosessista – V-malli (mutta muitakin prosesseja voi käyttää).
  - Systemaattinen ketju vaatimuksista toteutukseen ja verifioinnin kautta paluu ylätasolle – erilaisten testausvaiheiden kautta.
- Traditionaalisia systemaattisen ohjelmistokehityksen menetelmiä ja laatutekniikoita, mutta myös uusia, vaativia tekniikoita (SIL 3-4).
- Menettelyt valitaan SIL-tason perusteella.

Soveltamisen perusfilosofia

- Jämäkkä prosessi:
  - Sekä alku- että loppupää ja kaikki siltä väliltä.
  - Vaatimusmäärittely kokonaisjärjestelmän vaatimuksiin perustuen.
  - Arkkitehtuurin laatu.
  - Suunnittelun ja toteutuksen laatu.
  - Verifioinnin laatu – testauksen yms. laatu.
  - Turvallisuuden arviointi aina, kun suunnitteluratkaisuja muutetaan,
  - Validointi suhteessa siihen, mitä oikeasti tarvitaan, eli kokonaisjärjestelmän puitteissa.
  - Tiukka dokumentointi.
  - Erittäin vahva konfiguraationhallinta ja asioiden jäljitys.
  - Osaamisvaatimukset ja vastuut.
- Vastaa monia traditionaalisia näkemyksiä hyvästä ohjelmistotuotannosta.

Prosessimallina V-malli – mutta muitakin voi soveltaa



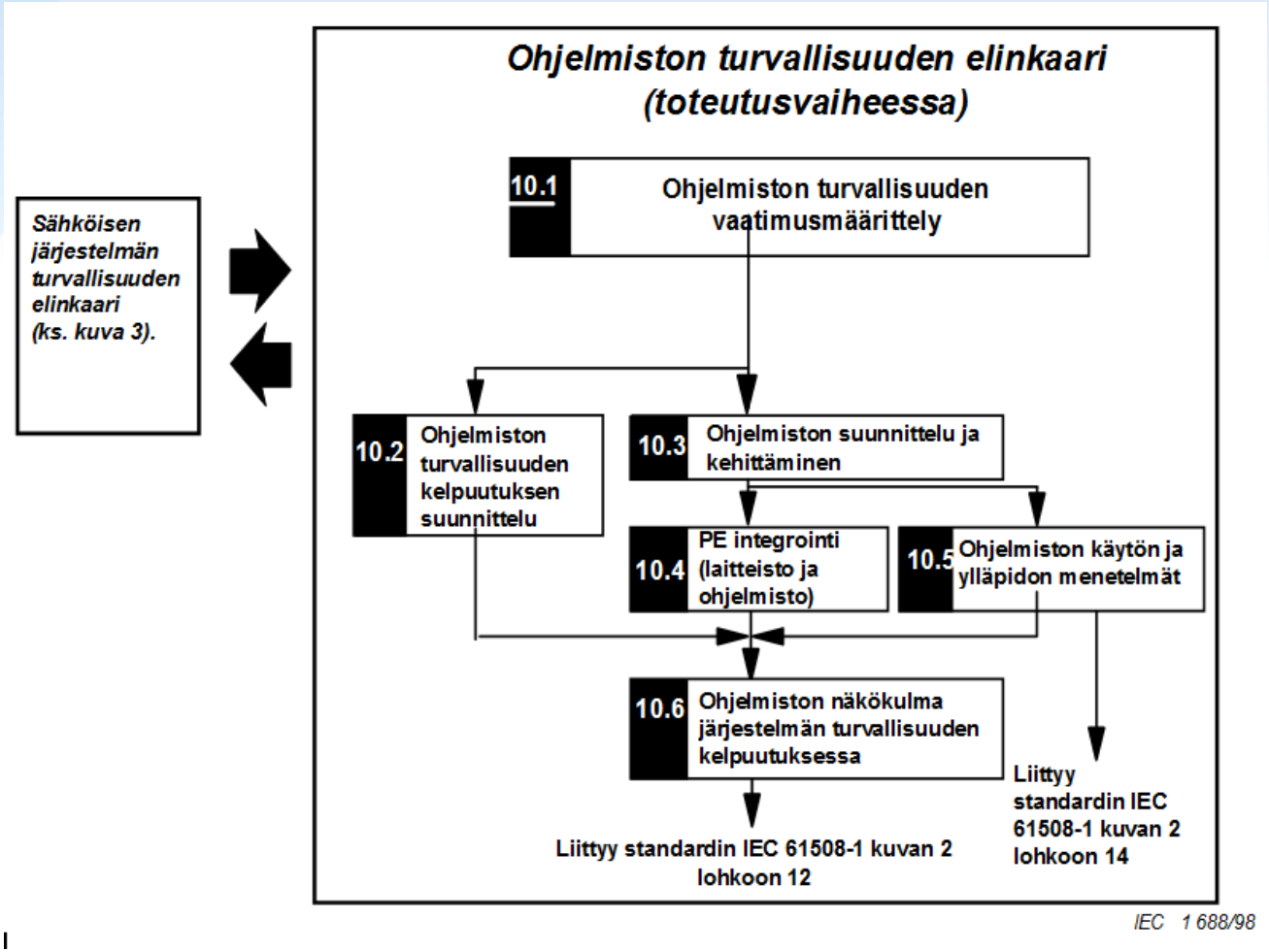
Tukee laadukkaan ohjelmistokehityksen ideaaleja

- Menettelyissä on erilaisia ”parhaita käytäntöjä” ja tiukkaa jämäkkyyttä.
- Standardille saa siksi tukea ohjelmistokehityksen laatuun liittyvistä standardeista ja kehysmalleista.
- Esimerkiksi:
- Laadunhallintajärjestelmästandardi:
  - ISO 9001:n sovellusstandardi ISO 90003 ”Guidelines for the application of ISO 9001:2000 to computer software” – suh. konkreettisia ohjeita – hyvä standardi, mutta vähän tunnettu
- Kypsyysmallistandardi:
  - CMMI for Development, http://www.sei.cmu.edu/library/abstracts/reports/06tr008.cfm

Milloin sovelletaan?

- Ohjelmistojen kehittäminen 61508-1:n määrittämässä kontekstissa
- Rajanveto: mikä on ohjelmointia ja mikä konfigurointia?
- ”Oikea ohjelmointi” purkkien sisällä
  - Tekstuaaliset ohjelmointikielet
  - Yleiskäyttöiset ohjelmointikielet, C#
- Jos turvajärjestelmää ei voi erottaa tuotannollisesta järjestelmästä, kokonaisuus kuuluu tämän standardin piiriin

Ohjelmiston turvallisuuden elinkaari



Rakenne ja sisältö – Luvut 3-6

- 3 Määritelmät ja lyhenteet
  - Viittaus: IEC 61508-4
- 4 Standardinmukaisuus
  - Vaatimukset standardin IEC 61508-1 luvussa 4.
- 5 Dokumentaatio
  - Tavoitteet ja vaatimukset standardin IEC 61508-1 luvussa 5
- 6 Lisävaatimukset turvallisuuteen liittyvän ohjelmiston hallintaan
  - Strategia koko toimintaan
  - Konfiguraationhallinta
  - Muutoksenhallinta: valtuus, dokumentointi, konfiguraationhallinta

Rakenne ja sisältö – Luku 7 / 7.1

- 7 Ohjelmiston turvallisuuden elinkaaren vaatimukset
- 7.1 Yleistä
  - Turvallisuuden elinkaaren ja ohjelmistokehityksen elinkaaren määrittäminen
  - V-mallin voi korvata, kunhan vaatimukset täyttyvät
    - Ketteristä projektimalleista myöhemmin
  - Elinkaaressa vahvasti turvallisuuden ja laadun toimenpiteet
  - Dokumentointi
  - Kun elinkaarimallissa iteroidaan, on tarvittavat muutokset analysoitava

Rakenne ja sisältö – Luku 7.1 & 7.2

- 7.2 Ohjelmiston turvallisuusvaatimusten määrittely
  - Syntetisoidaan vaatimukset lähtien kokonaisjärjestelmästä ja miten turvallisuusvaatimukset on allokoitu
  - Yksityiskohtaisuus ja dokumentointi
- 7.3 Kelpuutussuunnitelma järjestelmän turvallisuuden ohjelmisto-osuudelle
  - Miten turvallisuusvaatimusten täyttyminen osoitetaan
  - Miten, kuka, koska
  - Missä ympäristössä
  - Kaikenlaiset käyttötilanteet
  - Perustelu valinnoille
  - Tarkka esim. testien dokumentointi

Rakenne ja sisältö – Luku 7.4

- 7.4 Ohjelmistosuunnittelu ja ohjelmiston kehittäminen
  - Kuka vastaa - toimittaja, kehittäjä
  - Valittava oikeat suunnittelumenetelmät, joilla hallitaan turvallisuuskriittisten sulautettujen järjestelmien haasteet
  - Suunnittelussa otettava huomioon testattavuus ja muutettavuus
  - Yksinkertaisuus toteutuksissa!
  - Jos turvatoimintoja ei voida erottaa, on koko ohjelmistoa käsiteltävä turvallisuuskriittisenä
  - Hyväksymiskriteerit aiemmille komponenteille kuten ohjelmoitavalle elektroniikalle -- mutta ohjelmiston uudelleenkäyttö aina invalidoi sen käytännössä pitkälti
  - Hyvä arkkitehtuurisuunnittelu

Rakenne ja sisältö – Luku 7.4

- Ajonaikaiset "tukityökalut" ovat ohjelmiston elementti (lähtöjä tuottavat, testausvälineet)
- Ohjelmointikielet ovat ohjelmiston elementti – eli kehittämistyökalut on hallittava yhtä hyvin kuin kehitettävä ohjelma
- Ohjelman koodin oltava hyvää
- Moduulit testattava hyvin: kattavasti, toistettavasti, määritellyssä ympäristössä... tavallista systemaattista testausta
- Todentaminen suunnitelman mukaisesti - ketterä testaus ei riitä
- Negatiivinen testaus oleellista - ohjelma ei saa tehdä mitään, mitä se ei saisi; häiriöiden hallinta, robustius
- Moduulien integrointi suunnitelman mukaisesti
- Huom! Kaksi integrointitasoa: ohjelman moduulien integrointi ja ohjelman integrointi ohjelmoitavaan elektroniikkaan

Rakenne ja sisältö – Luku 7.4

- Ajonaikaiset "tukityökalut" ovat ohjelmiston elementti (lähtöjä tuottavat, testausvälineet)
- Ohjelmointikielet ovat ohjelmiston elementti -- eli kehittämistyökalut on hallittava yhtä hyvin kuin kehitettävä ohjelma
- Ohjelman koodin oltava hyvää
- Moduulit testattava hyvin: kattavasti, toistettavasti, määritellyssä ympäristössä... tavallista systemaattista testausta
- Todentaminen suunnitelman mukaisesti - ketterä testaus ei riitä
- Negatiivinen testaus oleellista - ohjelma ei saa tehdä mitään, mitä se ei saisi; häiriöiden hallinta, robustius
- Moduulien integrointi suunnitelman mukaisesti
- Huom! Kaksi integrointitasoa: ohjelman moduulien integrointi ja ohjelman integrointi ohjelmoitavaan elektroniikkaan

Rakenne ja sisältö – Luku 7.5 & 7.6

- 7.5 Ohjelmoitavan elektroniikan integrointi (laitteisto ja ohjelmisto)
  - Testit määritetään suunnittelu- ja kehitysvaiheessa (eli ei ad-hoc – mutta tietenkin ketterä, tutkiva testaus on tärkeää käytännössä)
  - Erotettava kehittäjän tiloissa tehtävät testit ja ne, joita pitää tehdä asiakkaan tiloissa
  - Jos integroinnin aikana tehdään muutoksia, niille vaikutusanalyysi
  - Testien kattava dokumentointi
- 7.6 Ohjelmiston käytön ja muutosten tekemisen menettelytavat
  - Vaatimukset standardin IEC 61508-2 kohdassa 7.6 ja tämän standardin kohdassa 7.8

Rakenne ja sisältö – Luku 7.7

- 7.7 Järjestelmän turvallisuuden kelpuutus (validointi) ohjelmiston osalta
  - Jos kelpuutus tehdään ohjelmoitavan elektroniikan kelpuutuksen myötä, sitä ei tarvitse toistaa
  - Kelpuutus kuin on määritetty järjestelmän turvallisuuden ohjelmisto-osuuksien kelpuutussuunnitelmassa
  - Vastuu voi jakautua eri osapuolille -- eli toimittaja hoitaa oman osuutensa
  - Tulosten tallennus
  - Jos jokin ei onnistu, tietoinen päätös kelpuutuksen jatkosta tai keskeytyksestä ja muutospyynnöstä
  - Pitää kelpuuttaa kaikki ohjelmiston turvallisuusvaatimukset
  - Testit dokumentoitava riippumatonta arviointia varten

Rakenne ja sisältö – Luku 7.8

- 7.8 Ohjelmiston muutokset
  - Pitää olla määritetty muutosprosessi
  - Valtuutettu muutospyyntö: Kannanotto, mihin vaaroihin vaikuttaa, syy muutokseen
  - Vaikutusanalyysi: pitääkö toistaa riskianalyysi; mihin prosessin vaiheisiin vaikuttaa
  - Muutoksen toteutus suunnitelman mukaan
  - Dokumentointi

Rakenne ja sisältö – Luku 7.9

- 7.9 Ohjelmiston todentaminen (verifiointi)
  - Elinkaaren eri vaiheissa
  - Todennetaan (mm.)
    - a) ohjelmiston turvallisuusvaatimukset,
    - b) ohjelmistoarkkitehtuuri,
    - c) ohjelmistojärjestelmän toteutus,
    - d) ohjelmistomoduulien toteutus,
    - e) koodi,
    - f) tiedot (tietorakenteet, sovellustiedot),
    - g) ajoitukset,
    - h) ohjelmistomoduulit,
    - i) ohjelmiston integrointi,
    - j) ohjelmoitavan elektroniikan integrointi,
    - k) järjestelmän turvallisuuden ohjelmisto-osuus

Rakenne ja sisältö – Luku 7.9

- Testauksen ohella monia muita tekniikoita
- Todentamisen täysi kattavuus elinkaaren edellisen vaiheen suhteen:
  - todentamisen oikeellisuus elinkaaren edellisen vaiheen suhteen (onnistunut loppuun saattaminen), toistettavuus, tarkasti määritelty todentamisen konfiguraatio
- Todentaminen on suunniteltava samanaikaisesti kehittämistoiminnan kanssa jokaiselle ohjelmiston turvallisuuden elinkaaren vaiheelle
- Ohjelmiston todentamisen suunnittelun on viitattava kriteereihin, tekniikoihin ja työkaluihin, joita käytetään
- Dokumentoitava näyttö osoittamaan, että todennettava vaihe on kaikilta osin tyydyttävästi loppuun suoritettu
- Todennettaessa on oltava kaikki tarpeelliset tiedot
- Tiukka todennettavan konfiguraation yksilöinti

Rakenne ja sisältö – Luku 8

- 8 Toiminnallisen turvallisuuden arviointi
  - Perusteet 61508-1:ssä
  - Arvioijien riippumattomuusvaatimukset: omaa porukkaa, toinen yksikkö, ulkopuolinen? IEC 61508-1, luku 8
  - Käytetään A.10:n tuloksia

Standardin liitteet 1/2

- Varsinainen ”ongelma” on liitteissä, joissa kuvataan tarvittavia menettelyjä eri SIL-tasoille
- Miten osataan valita sopivat menetelmät, jotka täyttävät vaatimukset ja joita osataan käyttää? (Standardin uudistus toi haasteita mm. verifiointimenetelmiin)
- Liite A: Ohje tekniikoiden ja toimenpiteiden valintaan
  - Velvoittava
  - Tekniikoiden edellyttäminen: HR, R, –, NR
- Liite B: Yksityiskohtaiset taulukot
  - Opastava
  - Lisätietoa vastaaviin A-taulukoihin

Standardin liitteet 2/2

- Liite C: Ohjelmiston systemaattisen kyvykkyyden ominaisuudet
  - Ohjeita tiettyjen liitteiden A ja B tekniikoiden valintaan ohjelmiston systemaattisen kyvykkyyden saavuttamiseksi
  - Perusteluja niiden tekniikoiden käyttämiseksi, joita ei ole eksplisiittisesti lueteltu liitteissä A ja B.
- Liite D: Turvallisuuskäsikirja vaatimustenmukaisille osille – lisävaatimuksia ohjelmiston elementeille
- Liite E: Standardien IEC 61508-2 ja IEC 61508-3 suhde
  - Opastava
- Liite F: Tekniikat, joilla saavutetaan ohjelmistoelementtien välinen erillisyys yhdessä tietokoneessa
  - Opastava
- Liite G: Ohjeita elinkaaren räätälöintiin tieto-ohjattujen järjestelmien yhteydessä
  - Opastava

Taulukkoesimerkki liitteestä A

- Taulukko A.1 – Ohjelmiston turvallisuusvaatimusten määrittely

	Tekniikka/toimenpide*	Viite	SIL1	SIL2	SIL3	SIL4
1a	Semiformaalit menetelmät	Taulukko B.7	R	R	HR	HR
1b	Formaalit menetelmät	B.2.2, C.2.4	---	R	R	HR
2	Eteenpäin jäljitettävyys järjestelmän turvallisuusvaatimusten ja ohjelmiston turvallisuusvaatimusten välillä	C.2.11	R	R	HR	HR
3	Taaksepäin jäljitettävyys turvallisuusvaatimusten ja havaittujen turvallisuustarpeiden välillä	C.2.11	R	R	HR	HR
4	Tietokoneavusteiset määritystyökalut yllä olevien soveltuvien tekniikoiden/toimenpiteiden tukemiseen	B.2.4	R	R	HR	HR

HUOMAUTUS 1 Ohjelmiston turvallisuusvaatimusten määrittelyyn vaaditaan aina ongelman kuvaus luonnollisella kielellä ja millä tahansa tarpeellisilla sovellusta kuvaavilla matemaattisilla merkinnöillä.

HUOMAUTUS 2 Taulukko kuvastaa selkeästi ja tarkasti lisävaatimuksia ohjelmiston turvallisuusvaatimusten määrittämiseen.

HUOMAUTUS 3 Katso taulukko C.1.

HUOMAUTUS 4 Viitteet (jotka ovat opastavia, eivät velvoittavia) “B.x.x.x”, ”C.x.x.x” sarakkeessa ’Viite’ ilmaisevat yksityiskohtaiset kuvaukset tekniikoista/toimenpiteistä, jotka on annettu standardin IEC 61508-7 Liitteissä B ja C.

*Soveltuvat tekniikat/toimenpiteet on valittava turvallisuuden eheyden tason mukaisesti. Vaihtoehtoiset tai vastaavat tekniikat/toimenpiteet on merkitty numeroa seuraavalla kirjaimella. On tarkoitettu, että vain yksi vaihtoehtoisista tai vastaavista tekniikoista/toimenpiteistä pitäisi toteuttaa. Vaihtoehtoisen tekniikan valinta pitäisi perustella kyseisen sovelluksen haluttujen ominaisuuksien mukaisesti, kuten esitetään Liitteessä C.

Liitteen A taulukot 1/2

- Taulukko A.1 – Ohjelmiston turvallisuusvaatimusten määrittely (Katso 7.2)
- Taulukko A.2 – Ohjelmistosuunnittelu ja ohjelmiston kehittäminen – ohjelmistoarkkitehtuurin suunnittelu (katso 7.4.3)
- Taulukko A.3 – Ohjelmistosuunnittelu ja ohjelmiston kehittäminen - tukityökalut ja ohjelmointikieli (katso 7.4.4)
- Taulukko A.4 – Ohjelmistosuunnittelu ja ohjelmiston kehittäminen – yksityiskohtainen suunnittelu (katso 7.4.5 ja 7.4.6) (Sisältää ohjelmistojärjestelmän suunnittelun, ohjelmistomoduulien suunnittelun ja koodauksen)
- Taulukko A.5 - Ohjelmistosuunnittelu ja ohjelmiston kehittäminen – ohjelmistomoduulien testaus ja integrointi (Katso 7.4.7 ja 7.4.8)

Liitteen A taulukot 2/2

- Taulukko A.6 – Ohjelmoitavan elektroniikan integrointi (ohjelmisto ja laitteisto) (katso 7.5)
- Taulukko A.7 – Järjestelmän turvallisuuden kelpuutus ohjelmiston näkökulmasta (katso 7.7)
- Taulukko A.8 – Muutokset (katso 7.8)
- Taulukko A.9 – Ohjelmiston todentaminen (katso 7.9)
- Taulukko A.10 – Toiminnallisen turvallisuuden arviointi (katso Luku 8)

Liitteen B taulukot – linkittyvät A-taulukoihin

- Taulukko B.1 – Suunnittelu- ja koodausstandardit (Viittaukset taulukosta A.4)
- Taulukko B.2 – Dynaaminen analyysi ja testaus (Viittaukset taulukoista A.5 ja A.9)
- Taulukko B.3 – Toiminnallinen testaus ja mustalaatikkotestaus (Viittaukset taulukoista A.5, A.6 ja A.7)
- Taulukko B.4 – Vika-analyysi (Viittaukset taulukosta A.10)
- Taulukko B.5 – Mallintaminen (Viittaukset taulukosta A.7)
- Taulukko B.6 – Suorituskykytestaus (Viittaukset taulukoista A.5 ja A.6)
- Taulukko B.7 – Semiformaalit menetelmät (Viittaukset taulukoista A.1, A.2 ja A.4)
- Taulukko B.8 – Staattinen analyysi (Viittaukset taulukosta A.9)
- Taulukko B.9 – Modulaarinen lähestymistapa (Viittaukset taulukosta A.4)

Liite C – Tarkoitus

- Kun otetaan huomioon se suuri määrä tekijöitä, jotka vaikuttavat ohjelmiston systemaattiseen kyvykkyyteen, ei ole mahdollista esittää algoritmia niiden tekniikoiden ja toimenpiteiden yhdistämiseen, jotka ovat oikeita mihin tahansa sovellukseen.
- Liitteen C tarkoitus on:
  - Esittää ohjeita tiettyjen liitteiden A ja B tekniikoiden valintaan ohjelmiston systemaattisen kyvykkyyden saavuttamiseksi
  - Hahmotella perustelut niiden tekniikoiden käyttämiseksi, joita ei ole eksplisiittisesti lueteltu liitteissä A ja B.
- Muitakin menetelmiä siis voi käyttää – kunhan niiden käytön perustelee.
- Liite C on täydennys liitteiden A ja B taulukoihin.

Liite C – Peruskäyttö 1/3

- Taulukko A.1 esittää toimenpiteitä turvallisuusvaatimusten määrittämiseen, mutta tässä ei ole vielä koko totuus…

Tekniikat/toimenpiteet *		Viite	SIL 1	SIL 2	SIL 3	SIL 4
1a	Semiformaalit menetelmät	Taulukko B.7	R	R	HR	HR
1b	Formaalit menetelmät	B.2.2, C.2.4	---	R	R	HR
2	Eteenpäin jäljitettävyys järjestelmän ja ohjelmiston turvallisuusvaatimusten välillä	C.2.11	R	R	HR	HR
3	Taaksepäin jäljitettävyys turvallisuusvaatimusten ja havaittujen turvallisuustarpeiden välillä	C.2.11	R	R	HR	HR
4	Tietokoneavusteiset määrittelytyökalut yllä olevien soveltuvien tekniikoiden/toimenpiteiden tukemiseen	B.2.4	R	R	HR	HR

Liite C – Peruskäyttö 2/3

- Taulukko C.1 kertoo, miten toimenpiteet tukevat systeemin ominaisuuksia, kun pyritään toimimaan tietyllä ”tarkkuudella” (rigour, esim. R1, R2, R3)

	Tekniikka/ Toimenpide	Ominaisuudet					
		Kattavuus ohjelmistoon kohdennettavien turvallisuustarpeiden osalta	Oikeellisuus ohjelmistoon kohdennettavien turvallisuustarpeiden osalta	Vapaus luontaisista määritysvirheistä sekä epäjohdonmukaisuuksista	Turvallisuusvaatimusten ymmärrettävyys	Vapaus ohjelmistolle kohdennettavien turvallisuusvaatimuksiin liittymättömien toimintojen haitallisista vaikutuksista	Kyvykkyys tuottaa perusta todentamiselle ja kelpuutukselle
1a	Semiformaalit menetelmät	R1 Sovellusystävällinen tai sovellusaluekohtainen määritysmenetelmä ja toimialan asiantuntijoiden käyttämä notaatio	R1 Sovellusystävällinen tai sovellusaluekohtainen määritysmenetelmä ja toimialan asiantuntijoiden käyttämä notaatio R2 Määritysten	R1 Menetelmä ja notaatio, joka auttaa välttämään tai tunnistamaan sisäisiä epäjohdonmukaisuuksia, puuttuvia toimintoja tai matemaattisesti epäjohdonmukaisia ilmaisuja	R1 Määritelty notaatio, joka rajoittaa väärinkäsitysten mahdollisuutta R2 Monimutkaisuuden rajoitusten soveltaminen määrityksissä	–	R2 Määritelty notaatio, joka vähentää määrittelyn epäselvyyttä

Liite C – Peruskäyttö 3/3

	Tarkkuustasot
R1	Ilman objektiivisia hyväksymiskriteerejä tai rajoitetuilla objektiivisilla hyväksymiskriteereillä, esimerkiksi arviointiin perustuva mustalaatikkotestaus ja kenttäkokeet.
R2	Objektiivisilla hyväksymiskriteereillä, jotka voivat tuoda korkean tason luottamuksen sille, että vaadittu ominaisuus saavutetaan (poikkeukset on yksilöitävä ja perusteltava, esim. testi- tai analyysitekniikat ml. kattavuusmittaus tai tarkistuslistojen kattavuus.)
R3	Objektiivisella, systemaattisella päättelyllä, jolla saavutetaan vaadittava ominaisuus, esim. formaali todistaminen ja pitäytyminen arkkitehtuurin rajoituksiin, jotka varmistavat kyseisen ominaisuuden.
_	Tämä tekniikka ei ole asiaankuuluva tälle ominaisuudelle.

Liite C.2 – Taulukot 1/3

- Kokoelma 1: Systemaattisen turvallisuuden eheyden ominaisuudet
  - Viitteet standardin lukuihin ja A-taulukoihin
  - Taulukko C.1 – Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmiston turvallisuusvaatimusten määrittely
  - Taulukko C.2 – Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmistosuunnittelu ja ohjelmiston kehittäminen – ohjelmistoarkkitehtuurin suunnittelu
  - Taulukko C.3 – Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmistosuunnittelu ja ohjelmiston kehittäminen – tukityökalut ja ohjelmointikielet
  - Taulukko C.4 – Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmistosuunnittelu ja ohjelmiston kehittäminen – yksityiskohtainen suunnittelu (mukaan lukien ohjelmistojärjestelmän suunnittelu, ohjelmistomoduulien suunnittelu ja koodaus)
  - Taulukko C.5 - Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmiston testaus ja kehittäminen – ohjelmistomoduulien testaus ja integrointi

Liite C.2 – Taulukot 2/3

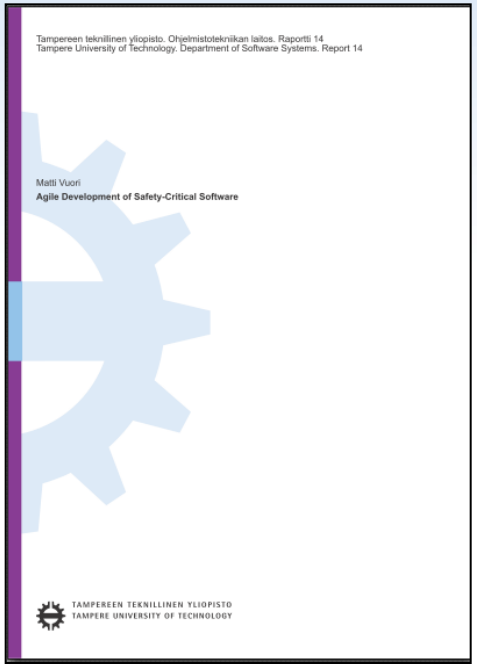
- Taulukko C.6 – Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmoitavan elektroniikan integrointi (laitteisto ja ohjelmisto)
- Taulukko C.7 – Systemaattisen turvallisuuden eheyden ominaisuudet – Systemaattisen turvallisuuden näkökulma ohjelmistoon
- Taulukko C.8 – Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmiston muuttaminen
- Taulukko C.9 – Systemaattisen turvallisuuden eheyden ominaisuudet – Ohjelmiston todentaminen
- Taulukko C.10 – Systemaattisen turvallisuuden eheyden ominaisuudet – Toiminnallisen turvallisuuden arviointi

Liite C.2 – Taulukot 3/3

- Kokoelma 2: Systemaattisen turvallisuuden eheyden ominaisuudet – Yksityiskohtaiset taulukot
  - Viitteet B-taulukoihin
  - Taulukko C.11 – Yksityiskohtaiset ominaisuudet – Suunnittelu ja koodausstandardit
  - Taulukko C.12 – Ominaisuuksien yksityiskohdat – Dynaaminen analyysi ja testaus
  - Taulukko C.13 – Ominaisuuksien yksityiskohdat – Toiminnallinen ja mustalaatikkotestaus
  - Taulukko C.14 – Ominaisuuksien yksityiskohdat – Vika-analyysit
  - Taulukko C.15 – Ominaisuuksien yksityiskohdat - Mallintaminen
  - Taulukko C.16 – Ominaisuuksien yksityiskohdat – Suorituskykytestaus
  - Taulukko C.17 – Ominaisuuksien yksityiskohdat – semiformaalit menetelmät
  - Taulukko C.18 – Systemaattisen turvallisuuden eheyden ominaisuudet – Staattinen analyysi
  - Taulukko C.19 – Ominaisuuksien yksityiskohdat – Modulaarinen lähestymistapa

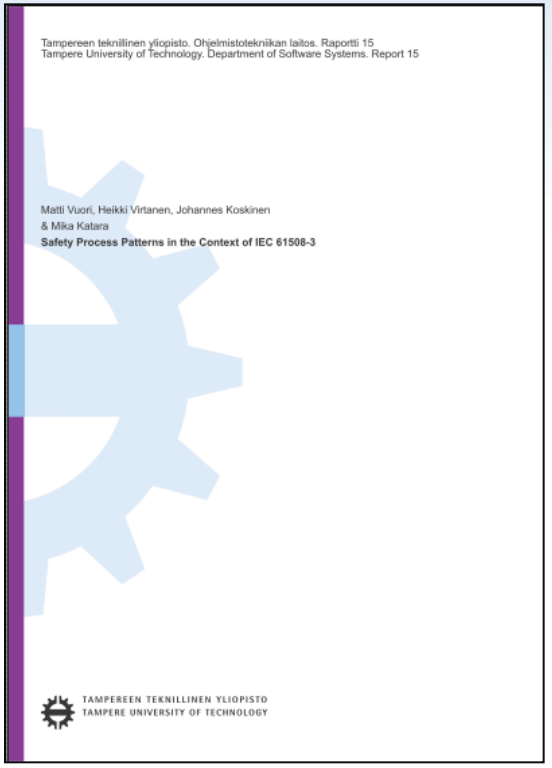
Apu ketterään kehittämiseen standardin mukaisesti

- Standardin ajattelumalli perustuu pitkälti perinteisiin ohjelmistokehitysmalleihin ja varsinkin ns. V-malliin.
- Jos yrityksessä käytetään ketteriä kehittämismalleja, pitää varmistaa, että kaikki standardin vaatima tulee tehtyä.
- TTY:llä tehtiin työ, jossa analysoitiin ketterän kehittämisen soveltamista turvallisuuskriittisten järjestelmien kehittämiseen ja miten 61508:n asiat voidaan toteuttaa ketterässä ohjelmistokehityksessä.
- Raportti ”Agile Development of Safety-Critical Software” löytyy netistä PDF:nä. Osoite: ks. lähteitä-kalvo.



Patterneista apua ymmärtämiseen

- Ns. patterneja käytetään usein toiminnan toistuvien asioiden kuvaamiseen puoliformaalilla, säännönmukaisella tavalla
  - Jokin työnkulku, aliprosessi, tapa tehdä asioita, tapa toimia
  - Systemaattiset tehtävät ja myös organisaation epäviralliset toimintamallit
- TTY:llä tehtiin työ, jossa niitä sovellettiin 61508:n, erityisesti 61508-3:n asioiden kuvaamiseen tiiviinä toimintamalleina
- Raportti ”Safety Process Patterns In the Context of IEC 61508-3” löytyy netistä PDF:nä. Osoite: ks. Lähteitä-kalvo.



Lähteitä ja viitteitä

- Vuori, M., Virtanen, H., Koskinen, J. & Katara, M. 2011. Safety Process Patterns In the Context of IEC 61508-3. Tampere University of Technology. Department of Software Systems. Report 15. 128 p. Available at: http://urn.fi/URN:NBN:fi:tty-201106141470l. (Tai hae sivulla: http://dspace.cc.tut.fi/dpub)

- Vuori, M. 2011. Agile Development of Safety-Critical Software. Tampere University of Technology. Department of Software Systems. Report 14. 95 p. Available at: http://urn.fi/URN:NBN:fi:tty-201106141470l. (Tai hae sivulla: http://dspace.cc.tut.fi/dpub)

Kiitos

Yhteystiedot:

Matti Vuori
Tampereen teknillinen yliopisto
Ohjelmistotekniikan laitos
matti.p.vuori@tut.fi
040 849 0039