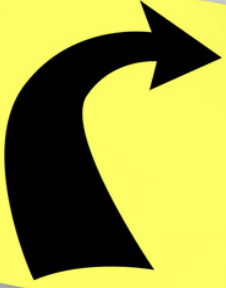
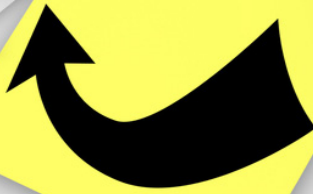


PEOPLE



PRODUCT

PROCESS



Katsaus kompleksisten järjestelmien elinkaaren suunnitteluun

Katsaus kompleksisten järjestelmien elinkaaren suunnitteluun

Göran Granholm (toim.)

ISBN 978-951-38-8045-3 (URL: <http://www.vtt.fi/publications/index.jsp>)

VTT Technology 121

ISSN-L 2242-1211

ISSN 2242-1211 (painettu)

Copyright © VTT 2013

ISBN 978-951-38-8045-3 (URL: <http://www.vtt.fi/publications/index.jsp>)

ISSN 2242-122X (URL: <http://www.vtt.fi/publications/index.jsp>)

Copyright © VTT 2013

JULKAISIJA – UTGIVARE – PUBLISHER

VTT

PL 1000 (Vuorimiehentie 5, Espoo)

02044 VTT

Puh. 020 722 111, faksi 020 722 4374

VTT

PB 1000 (Bergsmansvägen 5, Esbo)

FI-2044 VTT

Tfn +358 20 722 111, telefax +358 20 722 4374

VTT Technical Research Centre of Finland

P.O. Box 1000 (Vuorimiehentie 5, Espoo)

FI-02044 VTT, Finland

Tel. +358 20 722 111, fax + 358 20 722 4374

Katsaus kompleksisten järjestelmien elinkaaren suunnitteluun

A look into the life-cycle design of complex systems. **Göran Granholm (toim.).**
Espoo 2013. VTT Technology 121. 220 s. + liitt. 9 s.

Tiivistelmä

Jatkuva teknologinen kehitys, tiukentuvat asiakasvaatimukset ja kansainvälistyvät verkostot luovat yrityksille yhä vaativamman toimintaympäristön. Monitekniset tuotteet, asiakaskohtainen räätälöinti ja järjestelmien elinkaaren kattavat palvelut edellyttävät paitsi tehokasta tiedonhallintaa myös toimivia käytäntöjä ja tehokasta yhteistyötä eri suunnittelualojen välillä, jotta sekä nykyiset että tulevat vaatimukset voidaan huomioida heti suunnittelun alkumetreiltä.

Tässä kirjassa pyritään antamaan vinkkejä siihen, miten näihin haasteisiin voidaan vastata ja jopa kääntää ne kilpailukyvyksi ja menestykseksi. Valittuja keinoja ja metodiikkoja ovat kompleksisten järjestelmien tekniikka (Systems Engineering – SE), tuotteen elinkaaren hallinta (Product Lifecycle Management – PLM), mallipohjaisuus ja virtuaaliprototyyppi, vaatimusten hallinta sekä näihin liittyvät standardit. Tärkeä lähtökohta on menetelmien, prosessien ja työkalujen käyttöönotto ja soveltaminen sekä yrityksen sisällä että keskeisissä verkostoissa.

Keskeisenä lähtökohtana on järjestelmän käsitteleminen kokonaisuutena, johon vaikuttavat niin sidosryhmät, käyttöympäristö kuin kytkennät muiden järjestelmien muodostamaan ”ekosysteemiin” ja näissä tapahtuvat muutokset koko elinkaaren ajan. Näin ajatellen näennäisesti yksinkertainen laite voi olla osa kompleksista sosio-tekniistä järjestelmää. Tällöin on tärkeä ajatella järjestelmää kokonaisuutena osana arvoketjua ja elinkaarta yksittäisten toimintojen tai teknologian optimoinnin sijaan.

Tietoteknisten järjestelmien rooli tuotekehityksen ja elinkaarenhallinnan mahdollistajana on kiistaton. Järjestelmän kokonaisvaltainen käsittely sekä sisällöllisesti että ajallisesti tarkoittaa lukuisten sidosryhmien, suunnittelualojen ja teknisten ratkaisujen sekä palveluiden yhteensovittamista, mikä ilman edistyksellisiä tietoteknisiä järjestelmiä on nykypäivänä käytännössä mahdotonta. Asteittainen siirtyminen dokumenttipohjaisesta tiedonhallinnasta mallipohjaisiin esityksiin kaikissa elinkaaren vaiheissa mahdollistaa entistä tehokkaamman yhteistyön eri suunnittelualojen ja sidosryhmien välillä. Uusien menetelmien, prosessien ja niitä tukevien työkalujen käyttöönotto ja soveltaminen sekä yrityksen sisällä että keskeisissä verkostoissa vaatii kuitenkin huolellista suunnittelua ja tapauskohtaisten erityspiirteiden huomioimista.

Keskeisenä lähtökohtana on suunnittelukohteiden käsitteleminen osana kokonaisuutta, johon vaikuttavat niin sidosryhmät, käyttöympäristö kuin kytkennät muiden järjestelmien muodostamaan ”ekosysteemiin” ja näissä tapahtuvat muutokset koko elinkaaren ajan.

Avainsanat Systems Engineering, life-cycle management, product development, requirements engineering, model-based design, MBSE, PLM

A look into the life-cycle design of complex systems

Katsaus kompleksisten järjestelmien elinkaaren suunnitteluun. **Göran Granholm (Ed.)**.
Espoo 2013. VTT Technology 121. 220 p. + app. 9 p.

Abstract

The increasing complexity of both technological systems and the business environment in which companies operate continue to produce new challenges to manufacturing industry.

The objective of this book is to present methods and approaches that could help integrating tools, processes and systems thinking to achieve successful and sustainable products, considering all aspects of the product lifecycle. The chosen methodologies include Systems Engineering (SE), Product Life Cycle Management (PLM), model-based approaches, virtual prototyping, requirements management and related standards.

Adopting Systems Engineering principles across an organisation network involves additional challenges. It requires implementation of processes internally but also synchronization with other actors in the network, regarding both procedures and data.

With the increasing complexity of products, smaller time margins and higher demands on quality and compatibility the importance of models on all levels of the product process will increase. Automated design based on system models and efficient model re-use is still in its infancy but system description languages, such as SysML may in the future be the foundation of effective, high quality design processes and product life-cycle models.

One important feature of the model-based process is the possibility to replicate as much as possible of the products different life-cycle stages as virtual, digital model representations. Virtual modelling efficiently integrated into the product life-cycle enables effective user involvement throughout the process, including early design. It can be used in the validation of requirements and design concepts and may help identifying possible problems or design issues before major decisions have been fixed.

This guide book gives as a general introduction to the subject and treats selected topics on a comprehensive but easily approachable level, and gives hints to further reading. The guide will also include practical case studies to illustrate how Systems Engineering has been implemented in real life and what experience has been gained.

Keywords Systems Engineering, life-cycle management, product development, requirements engineering, model-based design, MBSE, PLM

Esipuhe

Tämän kirjan tarkoituksena on antaa yleiskatsaus Systems Engineering lähestymistapaan pohjautuviin suunnittelukäytäntöihin, joiden avulla voidaan paremmin huomioida tuotteiden koko elinkaaren vaatimukset ja sitä kautta aikaansaada kestäviä ja kilpailukykyisiä tuoteratkaisuja. Tuotteella tai järjestelmällä tarkoitetaan tässä yhteydessä mitä tahansa tuote–palvelukokonaisuutta. Kirja on tarkoitettu palvelemaan valmistavan teollisuuden yrityksiä, pk-sektori mukaan lukien. Otsikossa mainittu kompleksisuus ei välttämättä edellytä teknisesti monimutkaista järjestelmää, vaan viittaa ajatukseen, että teknisesti yksinkertainenkin tuote on osa laajaa ja monitahoista kokonaisuutta kun huomioidaan kaikki siihen vaikuttavat kytkökset koko elinkaaren ajalta.

Suunnittelutiedon ja elinkaaritiedon yhdistäminen edellyttää sekä tehokasta tiedonhallintaa että toimivia prosesseja. Tältä osin aihe liittyy kiinteästi yritysten jo toteutettuihin tai suunniteltuihin PLM-järjestelmäinvestointeihin. Toiveena on, että kirja herättäisi kiinnostusta aiheeseen ja auttaisi lukijaa löytämään syventävää lisätietoa monista muista saatavilla olevista lähteistä.

Kirja on syntynyt osana Tekesin Digitaalinen tuoteprosessi -tutkimusohjelman kautta rahoitettua projektia Systems Engineering – PLM integraation haasteet ja mahdollisuudet. Projektin koordinoinnista vastasi VTT, jonka lisäksi hankkeeseen osallistuivat omilla osuuksillaan Cargotec Oyj, Patria Land Systems Oy, Insta DefSec Oy, Eurostep Oy ja Dassault Systèmes Finland Oy. Asiantuntijaorganisaatioina projektin johtoryhmätyöskentelyyn osallistui lisäksi Fimecc Oy, Aalto-yliopisto ja Puolustusvoimat.

Projekti oli osa Fimecc Oy:n GP4V-ohjelmaa.

Kirjan sisällön tuottamiseen ovat osallistuneet Jarmo Alanen, Göran Granholm, Kari Karstila, Iris Karvonen, Simo-Pekka Leino, Tapio Salonen, Juha Sääski, Mikko Uoti, Peetu Valkama ja Juhani Viitaniemi. Kirjoittajien nimet on mainittu kunkin tekstin osan yhteydessä.

Kiitämme kaikkia yhteistyökumppaneita sekä projektiin osallistuneita tutkijoita siitä erinomaisesta yhteistyöstä, jonka pohjalta tämä kirja on syntynyt.

Espoossa 6. kesäkuuta 2013

Göran Granholm

Sisällysluettelo

Tiivistelmä	3
Abstract	4
Esipuhe.....	5
Lyhenteitä.....	8
Johdanto.....	10
OSA I – Näkökulmia monimutkaisten järjestelmien hallintaan	13
1. Kompleksisuuden haasteista mahdollisuuksia.....	14
1.1 Sosiotekniset järjestelmät	15
1.1.1 Sosiotekninen järjestelmä vaatimusten näkökulmasta	16
1.1.2 Sosioteknisten järjestelmien arviointi.....	18
1.2 Arvoverkostot lisäävät kompleksisuutta	18
1.3 Liiketoiminnan reunaehdot	19
1.4 Tiedonhallinnan näkökulma kompleksisuuteen	20
2. Lääkkeitä kompleksisuuden hallintaan ja liiketoiminnan	
tehostamiseen.....	23
2.1 Systeminen ajattelu käytännössä	23
2.2 Prosessit, toimintatavat ja työkalut kuntoon	26
2.3 Organisaatioiden ja roolien näkökulma	27
2.4 Vaatimustenhallinnan prosessi.....	29
2.5 Tuotteen elinkaaren hallinta – PLM	30
2.6 Mallipohjaisuus.....	34
2.7 Mitä ovat mallit?	35
2.8 Virtuaaliprototyyppi	37
2.9 Monimutkaisuuden hallintaan liittyviä standardeja	41
2.10 Menetelmiä prosessien kehittämiseen	41
2.11 PLM:n implementoinnista ja kaupallisista ratkaisuista.....	43
2.12 Verkoston valmiuksien kehittäminen.....	44

3. Johtopäätökset	46
Lähteitä ja lisätietoa (Osa I).....	47
OSA II – Systems Engineering käytännössä.....	49
SE-PLM soveltamisen haasteita	50
Mallipohjaisuus ja virtuaaliprototyointi	54
SE toimintatavan implementointi verkostoympäristössä.....	85
Vaatimusten luokittelu	113
Lähdeluettelo (Osa II)	133
OSA III – Sovellusesimerkkejä.....	135
Esimerkkejä systeemisuunnittelun soveltamisesta.....	135
Saksinostimen teknisten prosessien hallinta.....	136
Tuotekehitys ja virtuaali-katselmointi.....	170
Vaatimustenhallinta järjestelmähankinnassa	172
Ajoneuvon ergonomiavaatimukset	187
Lähdeluettelo (Osa III)	218
Liitteet	
Liite A: Systems Engineering – määritelmiä ja standardeja	
Liite B: Käsitekartta ja teoreettinen viitekehys	

Lyhenteitä

3DXML	Dassault Systèmesin kehittämä XML-pohjainen 3D-tiedostoformaatti
ALM	Application Life-cycle Management
AR	Augmented Reality (Lisätty todellisuus)
ASCII	American Standard Code for Information Interchange
B2B	Business to Business (Yritysten välinen liiketoiminta)
CAD	Computer Aided Design, Tietokoneavusteinen suunnittelu (myös Computer Aided Drafting, Tietokoneavusteinen piirtäminen)
CAE	Computer Aided Engineering,
ConOps	Concept of Operation (Toiminnallinen konsepti)
EDM	Engineering Data Management (Suunnittelutiedon hallinta)
ERP	Enterprise Resource Planning (Yrityksen toiminnanohjausjärjestelmä)
EXPRESS	Standardoitu mallinnuskieli tuotetiedon kuvaamiseen
FEA	Finite Element Analysis
FFBD	Functional Flow Block Diagram
FINSE	Finnish Systems Engineering Association ry
HMD	Head Mounted Display
HTML	Hypertext Markup Language
INCOSE	International Council on Systems Engineering
ITC	Information Technology and Communication Technology
MBSE	Model Based Systems Engineering (Mallipohjainen Systems Engineering)
PDF	Portable Document Format
PDM	Product data management (Tuotetiedon hallinta)
PIA	PLM Impact Analysis

PLM	Product Lifecycle Management (Tuotteen elinkaaren hallinta)
SE	Systems Engineering
SEM	Systems Engineering Manager
SEP	Systems Engineering Process
STEP	Standard for the Exchange of Product model data (ISO 10303)
PLCS	Product Lifecycle Support (ISO 10303-239)
SysML	Systems Modeling Language
U3D	Universal 3D (3D informaation siirtoon kehitetty pakattu tiedostomuoto)
UML	Unified Modeling Language
VE	Virtual environment (Virtuaaliympäristö)
VIP-ASE	Virtual Prototyping Aided MBSE
VR	Virtual reality (Virtuaalitodellisuus)
XLM	Extensible Markup Language

Johdanto

Suomalainen valmistava teollisuus elää haastavia aikoja. Markkinoiden dynamiikka lisääntyy, monitekniset tuotteet ja arverkot tulevat jatkuvasti kompleksisimmiksi, eikä tälle kehitykselle näy loppua. Samanaikaisesti laadukkaita, personoituja, konfiguroitavia ja kustannustehokkaita tuotteita tulee pystyä toimittamaan markkinoille varmasti ja yhä nopeammin. Kilpailun kiristyessä asiakkaiden vaatimukset eivät rajoitu tuotteen hankintavaiheessa määriteltyihin ominaisuuksiin, vaan huomio kiinnitetään yhä enemmän järjestelmän suunniteltuun käyttöaikaan kokonaisuutena. Tämä edellyttää koko elinkaaren huomioimisen jo suunnitteluvaiheessa.

Tarve hallita tuotteen koko elinkaari lisää kompleksisuutta: miten hallita keskenään ristiriitaisia tavoitteita ja vaatimuksia, joita esitetään eri sidosryhmien tahoilta sekä miten tukea päätöksentekoa ristiriitaisuuksien ratkaisuisissa erilaisin kompromissein? Tuotteen suunnittelussa on yhä useammin mukana sekä organisatorisesti että maantieteellisestikin hajautuneita osapuolia, valmistuksessa käytetään kumppaneita ja alihankkijoita ja tuotteen käytönaikaiset palvelut (esim. huolto) hoidetaan verkostolla, jossa on useimmiten mukana myös paikallisia osapuolia.

Tämä kirja on tiivis katsaus ja yhteenveto aineistosta, joka syntyi Tekesin rahoittamassa Systems Engineering – PLM integraatio -tutkimusprojektissa. Projektin tavoitteena oli löytää parhaita käytäntöjä teollisuuden tuotesuunnittelun ja elinkaari-prosessien integroituun hallintaan, sekä edistää kokonaisvaltaisen Systems Engineering -lähestymistavan yleistymistä teollisuudessa osana järjestelmien elinkaarenhallintaprosessia muun muassa hyödyntäen PLM-ohjelmistojen (Product Lifecycle Management) mahdollisuuksia. Projektissa luotiin katsauksia nykyisiin käytäntöihin ja tuleviin trendeihin, sekä kehitettiin malleja, jotka auttavat yrityksiä fokuosoimaan toimintaansa tärkeisiin kehityskohteisiin tuotteidensa suunnittelu- ja elinkaari-prosesseissa ja valitsemaan näitä tukevat tietotekniset ratkaisut. Projektissa hyödynnettiin kansainvälisiä kontakteja ja tuotiin ajankohtaista tietoa ja kokemuksia toimintamalleista ja työkaluista, joita voidaan sovittaa yritysten tarpeisiin ja joita yritykset voivat hyödyntää omien prosessiansa bench marking -toiminnassa.

Projektin alussa johtoryhmä täsmensi käsiteltäviksi aihealueiksi erityisesti:

- Systems Engineering -prosessien hyödyntäminen suomalaisessa teollisuudessa, mukaan lukien pk-sektori
- Systems Engineeringiin ja PLM:ään liittyvien käsitteiden selventäminen suomen kielellä

- Sosiotekninen kokonaisvaltainen näkökulma järjestelmiin ja niiden elinkaareen
- Monitekniisyys ja monialaisuus
- Vaatimustenhallinnan ulkoinen ja sisäinen näkökulma sekä konsepti-suunnittelun tukeminen
- Systems engineering -prosessit sekä niiden suhde PLM:n prosesseihin ja tietojärjestelmiin
- Mallipohjaisuus ja virtuaaliprototyypointi
- Yritysverkostojen ja kotimaisen teollisuuden tarpeet sekä toimialojen väliset eroavaisuudet
- Organisaatioihin ja johtamiseen liittyvät käsitteet ja niiden välinen suhde – kuka on System Engineer?

Aihealueen käsitteitä ja niiden välisiä kytkentöjä tutkittiin käsittekarttojen (Liite B) avulla. Aineistoa hankittiin kirjallisuuden lisäksi työpajatyöskentelyllä, haastatteleamalla yritysten asiantuntijoita ja tutkijoita sekä kotimaassa että ulkomailla (erityisesti Yhdysvalloissa). Myös muissa aihealuetta sivuavissa projekteissa ja toteutetuissa esimerkkisovelluksissa hankittuja kokemuksia hyödynnettiin tutkimuksen aineistossa.

Kirja on suunnattu suomalaisille teollisuusyrityksille yleisesti mutta sillä toivotaan olevan käyttöä myös esimerkiksi tuotesuunnittelun koulutuksessa ammattikorkeakouluissa. Kirja on jaettu kolmeen osioon. Ensimmäisessä osassa lähestytään aihetta kuvailemalla monimutkaisuuden ja kompleksisuuden haasteita ja ilmenemistä ja käsitellään keinoja vastata tunnistettuihin haasteisiin. Erityisesti pohditaan aihetta SE:n ja PLM:n kannalta. Osassa II tarkastellaan erikseen muutamia valittuja osa-alueita ja annetaan vinkkejä toimintatapojen ja prosessien kehittämiseen. Viimeisessä osiossa kuvataan muutamia esimerkkejä, joissa lähestymistapaa on sovellettu eri tavoin.

Kirja käsittelee eri aihealueita hyvin kompaktisti muutamalla sivulla per aihe, minkä lisäksi liitteinä on lisätietoa valittuihin osa-alueisiin liittyen (osa liitteistä on itsenäisiä työraporteja ja niissä on osittain päällekkäisyyttä yhteenvedon kanssa). Tarkoituksena on ollut enemmän herätellä kiinnostusta ja haastaa pohtimaan asiaa, kuin antaa valmiita ratkaisuja. Tämänkaltaisen laajan aihealueen oppaassa kaikille sopivien valmiiden ratkaisujen antaminen ei olisi mahdollistakaan. Tärkeintä on ensin tunnistaa kehityskohteet – niihin sopivia ratkaisuja kannattaa lähteä kehittämään esimerkiksi yritysten ja tutkimuslaitosten yhteistyössä.

Käytännössä lähes koko Systems Engineering -aiheeseen liittyvä terminologia on englanninkielinen. Suomenkielisen käsitteistön kehittäminen ja erityisesti sen vakiinnuttaminen teollisuudessa on erittäin haastavaa. Tämä johtuu sekä englanninkielisen kirjallisuuden valta-asemasta alalla että yritysten toimintaympäristön kansainvälistymisestä. Lisäksi käsitteet ja niiden merkitykset vaihtelevat tekniikan aloittain.

Kirjassa tarjotaan myös vinkkejä kirjallisuuteen, josta eri aihealueista voi lukea enemmän.

OSA I – Näkökulmia monimutkaisten järjestelmien hallintaan

Simo-Pekka Leino, VTT

Juhani Viitaniemi, VTT

Jarmo Alanen, VTT

Iris Karvonen, VTT

Mikko Uoti, VTT

Juha Sääski, VTT

Göran Granholm, VTT

Tässä osassa pohditaan teknisten järjestelmien, tuotteiden tai tuote-palvelukonseptien kokonaisvaltaiseen suunnitteluun ja hallittuun elinkaareen liittyviä asioita yleisellä tasolla. Ensimmäisessä luvussa käsitellään tällaisiin järjestelmiin liittyviä haasteita ja mahdollisuuksia. Toisessa luvussa kuvataan muutamia menetelmiä ja työkaluja, joiden avulla voidaan tarttua näihin haasteisiin.

1. Kompleksisuuden haasteista mahdollisuuksia

Monimutkaisuus ja kompleksisuus luovat suuria haasteita liiketoiminnalle. Haasteita ei välttämättä voida täysin voittaa, mutta niistä kilpailijoita paremmin selviytyminen voi tuoda kilpailuetua ja menestystä markkinoilla. Monimutkaisuus koskee nykyisin suurinta osaa tuotteita. Näennäisesti yksinkertainenkin tuote on usein esimerkiksi viestinnän ym. rajapintojen kautta osa kompleksista järjestelmää (system of systems). Palvelujen ja muiden aineettomien tekijöiden sekä dynaamisten prosessien ja organisaatioiden näkökulma tekee lähes kaikista tuotteista ja järjestelmistä kompleksisia. Kompleksisiin järjestelmiin sisältyy usein epävarmuustekijöitä, joita ei voida täysin hallita. Näiden monimutkaisten ja kompleksisten sosioteknisten järjestelmien ja tuotteiden kehittämistä käsitellään tässä kirjassa.

Järjestelmä eli systeemi on joidenkin periaatteiden mukainen toiminnallinen kokonaisuus. Sosiotekninen järjestelmä koostuu ihmisistä, teknisestä järjestelmästä ja aktiivisesta ympäristöstä. Tämän järjestelmän hyvyys riippuu kolmen osa-alueen yhteistoiminnasta. Yksi keskeinen suunnittelutehtävä on tehtävien allokointi ihmisille ja teknisille järjestelmille hyödyntäen molempien vahvuuksia. Ihmisellä on tiettyjä vahvuuksia verrattuna teknisiin järjestelmiin. Näitä ovat ennen kaikkea kyvykkyys tehdä päätöksiä rajatun informaation perusteella sekä joustavuus erilaisissa tilanteissa. Sosioteknisen järjestelmän mallin tulee kuvata kuka tekee, miten/millä tekee ja missä ympäristössä. Käytännön suunnittelutehtävässä tämä kokonaiskuva saattaa hämärtyä, mutta menetelmillä, kuten SE ja mallipohjaisuus, voidaan tukea systeemistä ajattelua.

Käsitteillä *monimutkainen (complicated)* ja *kompleksinen (complex)* on hyvä nähdä ero, joka liittyy sosioteknisen järjestelmän luonteeseen. Teknisen järjestelmän (esim. lentokone) monimutkaisuus johtuu suuresta määrästä ennalta määrätysti vuorovaikutteisesti toimivia komponentteja, mutta kompleksisessa järjestelmässä on mukana joukko itsenäisiä toimijoita [Amaral & Uzzi 2007]. Ihmiset ja ympäristö aiheuttaa usein sen että suhteellisen yksinkertaisestakin teknisestä järjestelmästä tulee osa kompleksista järjestelmää.

Seuraavien kappaleiden määritelmien keskeisenä lähteenä on käytetty Hubkan ja Ederin teoriaa [1984], joka pyrkii kokonaisvaltaiseen ja neutraaliin sosio-tekni-
järjestelmien kuvaukseen.

1.1 Sosiotekniset järjestelmät

Tuote tarkoittaa sekä virtuaalista (tuotetyyppi, spesifikaatio, portfolio), että fyysistä ja aineetonta tuoteyksilöä (konfiguraatio), joilla on omat elinkaaret. Liiketoiminnan eri sidosryhmät ymmärtävät tuotteen elinkaaren hallinnan käsitteenä sekä siihen liittyvät termit eri tavoilla. Myös tuotteen elinkaaren hallintaan liittyvät odotukset ja vaatimukset ovat hyvin erilaisia, ja usein ristiriitaisia riippuen henkilöistä ja sidosryhmistä. Vaatimuksia on usein vaikea priorisoida, eikä niitä voida kaikkia toteuttaa.

Tuotteiden ja toimintaympäristön monimutkaistuminen vaikeuttaa myös suunnitteluprosesseja. Puhutaan mm. multi-X (multi-site, multi-project, multi-partner)-suunnittelusta, jossa vaatimukset tulevat monesta lähteestä: loppukäyttäjiltä, globaaleilta asiakkailta, eri markkinasegmenteilta, standardeista, viranomaismääräyksistä, olemassa olevista tuotteista, toimituksesta, asennuksesta, käyttöönnotosta ja teknologiatutkimuksesta. Tähän multi-joukkoon voidaan vielä lisätä monitekniisyys, joka on edelleen laajentunut mm. tuotteisiin liittyvien ohjelmistojen kriittisyyden kasvaessa. Tuotteet eivät ole vain yksittäisiä järjestelmiä vaan "systems of systems".

Järjestelmän monimutkaisuus voi perustua esimerkiksi sen rakenteen monimutkaisuuden asteeseen. Monimutkaisuus on suoraan kytketty järjestelmän suunnitteluun, jolloin suunnittelijoilta vaaditaan tiettyä erityisosaamista ja tuoteproses-
silta kyvykkyyttä. Järjestelmät voidaan luokitella myös suunnittelutehtävän monimutkaisuuden perusteella, mikä auttaa esim. suunnittelun ulkoistamisessa. Vaikeusaste liittyy myös järjestelmän omaperäisyyteen, sen funktioiden ja vaikutuksen monimutkaisuuteen, rakenteeseen, mallintamiseen ja analysointiin, kustannusten optimointiin, jne. Järjestelmiä voidaan myös luokitella abstraktiivisemmin niiden rakenteen taksonomian perusteella. [Hubka ja Eder 1984]

Valmistukseen liittyen järjestelmät voidaan jakaa myös yksittäiskappaleisiin (one-off), piensarjoihin, variantteihin, massatuotantoon, jne. Tuotteen prototyyppi on myös yksi erityinen luokka ja joskus prototyyppi on myös one-off myytävä tuote. Prototyyppiä hyödynnetään yleensä järjestelmän vaatimuksiin perustuvien ominaisuuksien/toimintojen verifiointissa ja validoinnissa. Mallipohjaisuudella ja virtuaali-prototypoinnilla voidaan osittain korvata fyysiset prototyypit.

Suunnittelutyöhön vaikuttaa myös valmistettavien tuotteiden lukumäärä. Yksittäisten tuotteiden ja järjestelmien kohdalla toiminnallisuudet ja muut ominaisuudet pitää pystyä suunnittelemaan varmasti, koska prototyyppien valmistaminen ei yleensä tule kysymykseen. Mallipohjaisesti voidaan kuitenkin tutkia myös yksittäisten järjestelmien ominaisuuksia jo suunnitteluvaiheessa. Sarjatuotannon tai suurten erien tapauksessa taas on mahdollista testata ja kehittää tuotteen ja sen valmistuksen ominaisuuksia fyysisten tai virtuaalisten prototyyppien avulla. Valmistettavuus ja kokoonpantavuus ovat keskeisiä teknisen järjestelmän ominaisuuksia. Niihin liittyvät suunnittelussa tehtävät valinnat vaikuttavat noin 60–80 % järjestelmän

kokonaiskustannuksista. Soveltuvat valmistusmenetelmät riippuvat myös erittäin paljon siitä onko kyseessä yksittäinen tuote vai sarjatuote tai jotain siltä väliltä. Sarjatuotannossa kannattaa esimerkiksi usein pyrkiä automaatioon ja robotiikkaan, mikä asettaa tuotteen valmistettavuudelle tiettyjä reunaehtoja. Joka tapauksessa vuorovaikutus suunnittelijoiden ja tuotannon välillä on tärkeää ja mallipohjaisuus tuo mahdollisuuksia parantaa myös sitä.

Haluttu päämäärä, jota varten sosiotekninen järjestelmä on luotu, saadaan aikaan tietyillä järjestelmän käytön ja elinkaaren aikaisilla vaikutuksilla. Käytön aikaiset ominaisuudet määrittävät miten hyvin järjestelmä soveltuu tehtäväänsä. Käytön aikaiset ominaisuudet pitävät sisällään mm. luotettavuuden, turvallisuuden, eliniän, energiankulutuksen, huollettavuuden, käytettävyyden, kustannukset, jne. Sosio-tekniset järjestelmät valmistetaan useimmiten jonkinlaisessa tehtaassa ja ne kokevat monia vaiheita ennen varsinaista käyttöönottoa, esimerkiksi pakkaamista, kuljettamista paikasta toiseen, varastointia. Nämä aiheuttavat myös kustannuksia, joihin vaikuttavat esim. dimensiot, muoto, massa, materiaali, kestävyys, asennettavuus, jne.

1.1.1 Sosiotekninen järjestelmä vaatimusten näkökulmasta

Tuotteelle tai järjestelmälle asetetaan vaatimuksia hyvin monilta tahoilta (kuluttajat, loppukäyttäjät, sidosryhmät, viranomaiset, jne.) ja vaatimukset ovat tyypillisesti hyvin erityyppisiä (käyttötapaukset, suorituskyky, turvallisuus, ulkonäkö, valmistettavuus, jne.). Vaatimukset eivät aina kytkeydy saumattomasti elinkaareen, eikä ole varmuutta kenen vastuulla niiden toteuttaminen on. Vaatimuksia voidaan myös tarkastella eri näkökulmista: toiminnallinen näkökulma, järjestelmän näkökulma, tuotteen näkökulma, käyttö- ja ylläpito näkökulma.

Kun puhutaan järjestelmän vaatimuslistoista, kyseessä on järjestelmä *haluttujen ominaisuuksien määrittelystä* [Hubka & Eder 1984]. Termi *ominaisuus* (property) on haastava, koska se kattaa laajasti ymmärrettynä mm. funktion, kyvykkyyden, dimensiot, huollettavuuden, valmistettavuuden. Kun tekniselle järjestelmälle määritellään haluttuja ominaisuuksia osana suunnittelutehtävää, niitä kutsutaan vaatimuksiksi (requirements). Vaatimusmäärittely voi esiintyä monessa muodossa riippuen tarkoituksesta ja sitä käsittelevistä henkilöistä, esimerkiksi:

- Viranomaiset määrittelevät usein muodollisesti mitä vaatimusmäärittelyn (requirements specification) pitää sisältää
- Vastaava on esimerkiksi asiakkaan määrittelemä tarjouspyyntö (request for tender)
- Sopimuksessa määritellään ja allekirjoitetaan sopimusmäärittely (design contract specification)
- Tämän taas suunnittelijat tulkitsevat suunnittelumäärittelyksi (design specification).

Edellä mainitut koskevat järjestelmiä, jotka eivät vielä ole fyysisesti olemassa. Olemassa olevia järjestelmiä koskevat valmistusmäärittelyt, testimäärittelyt, käytönmäärittelyt (manuaalit) jne. Epätäydellinen varhainen spesifikaatio johtaa usein virheisiin lopputuotteessa. Spesifikaatioiden muotoon ja sisältöön vaikuttavat järjestelmän toimintojen monimutkaisuus, suunnittelutehtävän vaikeus (täysin uusi järjestelmä vs. pienet parannukset), vaatimukset lisäominaisuuksille (turvallisuus, elinkaari jne.), ongelman asettaja tai rahoittaja.

Epäonnistumisen riskit kasvavat yleensä suoraan suhteessa suunniteltavan järjestelmän uutuuteen ja monimutkaisuuteen – ja vähenevät kun vaatimukset ja suunnitelma kypsyvät. Koska vaatimuksia iteroidaan, ne pitää laatia muutos mielessä. Tähän liittyen on myös tärkeää, että vaatimukset kuvataan kyvykkyyksinä ja ominaisuuksina – ei toimintoina ja ratkaisuin. Uutuus tarkoittaa ominaisuuksia ja niiden yhdistelmiä, joita ei aikaisemmin ole vastaavissa järjestelmissä sovellettu, ja jotka johtavat kaupallisiin innovaatioihin. Suunnittelu sisältää silloin tyypillisesti konseptisuunnittelua.

Asiakas voi toteuttaa tarpeensa joko ostamalla valmiin tuotteen tai tilaamalla yksittäisen tuotteen. Näissä kahdessa tapauksessa spesifikaatioita määrittelevät eri tahot. Ensimmäisessä tapauksessa myynti ja markkinointi osastot määrittelevät vaatimusmäärittelyt, jälkimmäisessä asiakas itse osallistuu määrittelyyn. Kun asiakas tilaa tuotteen suoraan valmistajalta, hän vahvistaa vaatimukset. Samalla hän ottaa riskin siitä että vaatimukset ovat väärin määritellyt. Kun markkinointi ja myynti määrittelevät vaatimukset uudelle tuotteelle, esiintyy riski siitä että asiakkaat eivät halua ostaa tuotetta eli vaatimukset eivät kohtaa. Koska vaatimusmäärittelyt eivät voi koskaan olla täydellisiä, suunnittelijoiden pitää täydentää niitä lisäominaisuuksilla. Asiakkaat ja markkinointikin määrittelevät yleensä vain tuotteen ulkoisia ominaisuuksia. Vain tietyn tyyppiset asiakkaat (kuten armeija, viranomaiset) määrittelevät myös tuotteen sisäisiä ominaisuuksia kuten materiaali ja pinnan laatu. Edellä mainitut vaatimukset koskevat lopputuotetta, joka toimitetaan asiakkaalle. Tilanne on toinen kun puhutaan osajärjestelmistä ja niiden elementeistä. Silloin suunnittelijat määrittelevät vaatimuksia lähinnä itselleen. Mukana on myös rajoitteita (constraints), jotka rajaavat pois tiettyjä vaihtoehtoja. Teknisten rajoitteiden lisäksi esiintyy mm. lakeja, sääntöjä, eettisiä periaatteita jne. jotka myös rajaavat suunnitteluvaihtoehtoja. Vaatimukset voidaan luokitella niiden tärkeyden mukaan esimerkiksi kolmeen ryhmään:

- Pakolliset vaatimukset
- Vaatimukset joista voidaan joustaa tietyissä olosuhteissa
- Toiveet, joita ei ole pakko täyttää mutta jotka tuovat lisäarvoa.

Vaatimuksia koskeva luokittelut tulee hyväksyä yhdessä asiakkaan ja toteuttajien kesken.

Yksi vaatimusryhmä koostuu epäsuorista, implisiittisistä, vaatimuksista, jotka suunnittelijat pyrkivät aina täyttämään etiikan ja ammattitaidon perusteella, vaikka niitä ei erikseen määritellä. Näitä ovat: yleiset käyttöön liittyvät vaatimukset (minimoitu energiankulutus, massa jne.), ergonomia, resurssien hyödyntämisen maksimointi, taloudelliset tekijät.

1.1.2 Sosioteknisten järjestelmien arviointi

Teknisillä järjestelmillä on tiettyjä niitä määritteleviä ominaisuuksia kuten koko, massa, muoto, valmistettavuus, luotettavuus. Osa järjestelmien ominaisuuksista on ilmaistavissa kvantitatiivisesti eli määrällisesti, mutta toiset ominaisuudet ovat määriteltävissä vain kvalitatiivisesti eli laadullisesti. Laadullinen arviointi riippuu arvioivan henkilön omasta näkökulmasta. Toisaalta laadullisia ominaisuuksia voidaan kvantifioida todennäköisyyksiin perustuen. Tällaisia ominaisuuksia ovat mm. luotettavuus ja huollettavuus. Osa ominaisuuksista on arvioitavissa vain subjektiivisesti, esimerkiksi ulkonäkö. Näistä syistä johtuen on haastavaa valita ne ominaisuudet, joilla halutaan kuvata tai evaluoida teknistä järjestelmää.

Tilanne on huomattavasti monimutkaisempi silloin kun järjestelmä on vasta abstraktissa konseptivaiheessa. Silloin mallinnustekniikat voivat olla tarkoituksenmukaisia apukeinoja. Erityisesti silloin kun kyseessä on uusi järjestelmä (ottaen huomioon koko elinkaari), mallinnuksesta voi olla huomattavaa hyötyä objektiivisessä arvioinnissa.

1.2 Arvoverkostot lisäävät kompleksisuutta

Yritysverkosto on itsenäisten yritysten yhteenliittymä, joka tekee yhteistyötä pystyäkseen paremmin palvelemaan asiakkaitaan. Globaali markkina-alue, monimutkaiset ja monitekniset tuotteet ja niitä tukevat elinkaari palvelut edellyttävät monialaista osaamista, jota yksi yritys ei yleensä kykene yksin hallitsemaan, vaan sen tulee verkostoitua muiden yritysten kanssa. Yritysten verkostoitumisella on monia tavoitteita ja muotoja. Yritykset pyrkivät saamaan tehokkuutta ja joustavuutta keskittymällä ydinosaamiseensa ja tekemällä yhteistyötä muiden yritysten kanssa. Usein pyritään pitkäkestoisiin yhteistyösuhteisiin, jolloin yhteistyöprosessit voidaan kehittää toimiviksi ja osaaminen voi kumuloitua helpommin kuin lyhytaikaisissa suhteissa. Perinteisen resurssipohjaisen ja osaamiseen perustuvan verkostoitumisen lisäksi uusiksi verkostoitumisen ajureiksi on tullut lisääntyvässä määrin asiakas- suuntautuneisuus ja uudet markkinat, ketteryys ja riskien hallinta sekä kestävä kehityksen tavoitteet.

Verkostoituneessa toiminnassa, itsenäisten yritysten yhteistyössä päätöksenteko hajautuu ja suunnittelun seuranta ja hallinta on haastavampaa. Kukin yritys päättää esim. itse omien resurssiensa käytöstä. Aikataulujen lyhentäminen aiheuttaa paineita rinnakkaissuunnittelulle. Erilaiset käsitteet, prosessit, toimintatavat ja kulttuurit voivat aiheuttaa vääryimmäryksiä, virheitä ja prosessien hidastumista. Tiedonvaihto ja -hallinta ei ole tyypillisesti yhtä avointa kuin yrityksen sisällä. Hajautetussa ympäristössä projektit menettävät helposti kuvan systeemitason tavoitteista, systeemiarkkitehtuurista ja määräajoista. Saatetaan tehdä lokaaleja, yksipuolisia päätöksiä, jotka vaikuttavat systeemitasolle resursseihin ja aikatauluihin, joskus välittämättä aiemmista sitoutmisista. Tällöin tarvitaan näkyvyyttä projektin kuluessa eläviin sitoutmisiin sekä organisaation sisällä ja ulkoisten partnereiden kanssa.

Yritysverkostojen luonteen ja keston mukaan voidaan tunnistaa kaksi keskeistä verkostomuotoa:

- *Pitkäkestoinen verkoston valmiustila*, jossa yhteistyössä kehitetään yhteisiä valmiuksia yhteistyöhön, jotta asiakastilauksen tullessa pystyttäisiin reagoimaan nopeasti ja tehokkaasti tuottamaan asiakkaalle arvoa. Yhteistyöllä voi olla erilaisia muotoja ja kiinteyksasteita, alkaen ns. kasvatusympäristöstä (breeding environment) löyhempiin liiketoiminta-ekosysteemeihin ("business ecosystems").
- *Tiettyyn, usein aikataulutettuun tavoitteeseen tähtäävä yhteistyö* esim. projektiluonteisesti. Tällaista verkostomuotoa kutsutaan joskus "virtuaaliorganisaatioksi". Sen odotetaan toimivan kuten yksi organisaatio, vaikka se muodostuu useista eri organisaatioista. Virtuaaliorganisaatio luodaan tiettyä tehtävää varten ja puretaan, kun tehtävä on saatu suoritetuksi.

Vaikka verkostoituminen voi tuottaa yrityksille paljon hyötyä, verkostotoiminnassa on myös paljon haasteita, jotka liittyvät mm. toimijoiden omiin tavoitteisiin ja päätöksentekoon, erilaisten tavoitteiden yhteensovittamiseen sekä osaamisen hallintaan. Verkostoympäristössä tuotteen elinkaari prosessit ulottuvat organisaatorajojen yli. Tuotteen suunnittelussa voi olla mukana useita organisatorisesti ja maantieteellisestikin hajautuneita osapuolia, valmistuksessa käytetään kumppaneita ja alihankkijoita ja tuotteen käytön aikaiset palvelut (esim. huolto) hoidetaan verkostolla, jossa on useimmiten mukana myös paikallisia osapuolia.

1.3 Liiketoiminnan reunaehdot

Taloudelliset ominaisuudet yhdistävät monia edellä mainittuja ominaisuuksia niitä kuvaaviksi tunnusluvuiksi – kustannuksiksi ja tuloiksi. Kustannuksia ovat suunnitteluun, hankintaan, valmistukseen, jakeluun, yms. liittyvät kustannukset sekä operointiin liittyvät käyttökustannukset. Nämä yhdessä määrittävät järjestelmän elinkaarikustannukset. Suunnittelutyön hyvyttä voidaan myös tutkia analysoimalla yksittäisiä kustannuslajeja, joita tarvitaan vaadittujen järjestelmän ominaisuuksien toteuttamiseen, sekä niiden välisiä vuorovaikutteisia kytkentöjä. Kustannuksia voidaan usein pienentää maksimoimalla uudelleenkäytettävyys suunnittelussa ja valmistuksessa. On myös tärkeää analysoida kustannusten näkökulmasta onko taloudellisinta valmistaa tuote tai komponentit itse, ostaa ne toimittajalta tai ostaa valmistus alihankintana. Menestyvän liiketoiminnan kannalta on tärkeää nähdä ja kehittää järjestelmää kokonaisuutena osana arvoketjua ja elinkaarta yksittäisten toimintojen tai teknologian optimoinnin sijasta. Teknisen järjestelmän elinkaaren lopussa se poistetaan käytöstä ja likvidoidaan mikä pitää ottaa huomioon järjestelmän suunnittelussa.

Liiketoiminnan näkökulmasta järjestelmä kokonaisuudessaan on tärkeä. Mitkä ovat järjestelmän rajapinnat ja osa-järjestelmät arvonverkoston ja elinkaaren näkökulmasta, mitkä ovat sidosryhmät ja niiden väliset vuorovaikutukset koko järjestelmän elinkaaren ajan? Tämä kokonaisuus on kompleksinen eli se sisältää lukuisia

itsenäisiä toimijoita, jotka eivät usein näe kokonaiskuvaa tai joilla on jopa ristiriitaisia tavoitteita.

Liiketoiminta määrittelee tavoitteet ja kriteerit järjestelmän kehittämiseen. Tämä tulisi tapahtua tiiviissä yhteistyössä loppukäyttäjien, asiakkaiden ja elinkaaren sidosryhmien kanssa. Projektiprosessit toteuttavat liiketoiminnan asettamat tavoitteet, jäsentävät päätöksenteon sekä johtavat käytännön tekemistä teknisten prosessien transformaatioiden eli järjestelmän toteuttamiseen liittyen.

Suunnittelijat luovat teknisen järjestelmän ja suurimman osan sen ominaisuuksista. He eivät suoraan vaikuta valmistukseen, käyttöön, ylläpitoon, mutta luovat edellytykset niille. Teknisen järjestelmän arvo luodaan suunnitteluprosessissa. Mallipohjaisuutta voidaan hyödyntää järjestelmän ominaisuuksien arvioinnissa ja kokonaisuoptimoinnissa. Teknisen järjestelmän arvo toteuttaa jonkun tahon tarpeen tai tuottaa muuta lisäarvoa. Arvot voidaan ryhmitellä seuraavanlaisiksi ominaisuuksiksi:

- Tekninen arvo (funktiot, valmistettavuus jne.)
- Taloudellinen arvo
- Ergonominen ja esteettinen arvo
- Käyttöön liittyvä arvo
- Kokonaisarvo.

Järjestelmän toteuttavia arvoja voidaan verrata absoluuttiseen ideaaliseen ratkaisuun. Arvostus on yhdistelmä niistä järjestelmän ominaisuuksista, jotka vaikuttavat ihmisten haluun omistaa se. Se viittaa järjestelmän käyttöön, ergonomiaan, esteetiikkaan, yhteiskunnallisiin ja taloudellisiin ominaisuuksiin.

1.4 Tiedonhallinnan näkökulma kompleksisuuteen

Tietotekniikka nähdään keinona tukea kompleksisten järjestelmien ja prosessien hallintaa. Ilman niitä nykyisen kaltainen liiketoiminta ei olisi mahdollista. Toisaalta tiedonhallintaan liittyy useita haasteita, joita ratkaisemalla voidaan tehostaa liiketoimintaa. Sosioteknisten järjestelmien kohdalla Product Lifecycle Management (PLM), tuotteen elinkaaren hallinta, on eräs keskeinen teknologia, jonka selkärangan on Product Data Management (PDM), tuotetiedon hallinta.

PLM ratkaisujen ja installaatioiden historiallinen tausta nähdään yleensä olevan sarjatuotantotyyppisissä, konfiguroituissa tuotteissa (esimerkiksi autonvalmistus) ja mekaniikan detalji-suunnittelussa ja tuoterakenteiden hallinnassa. PLM on tavallaan kasvanut PDM:stä ja Engineering Data Management (EDM), CAD suunnittelutiedon hallinta, kohti elinkaaren loppupäätä. PDM ja PLM on tyypillisesti ollut tunnettua ja käytetty lähinnä tuotekehitys- ja suunnittelu-organisaatioissa. Tällä hetkellä vahvana trendinä on laajentaa PLM:n kattavuutta myös toiseen suuntaan mekaniikasta ja osarakenteesta kohti moniteknistä ja abstraktia systeemi-kuvausta, toimintarakenteita, konseptisuunnittelua ja vaatimustenhallintaa. PLM:ään liittyvä ymmärrys on perinteisesti usein ollut suhteellisen pienellä joukolla ihmisiä yrityksen sisällä. Kun trendinä on ollut laajentaa yhteistä PLM:ää globaalien konsernien ja verkostojen sisällä, on ollut tarpeen lisätä myös yhteistä ymmärrystä.

Lisäksi on tarvittava mittava määrä nimikkeiden, prosessien, menetelmien, terminologian, organisaatioiden yms. harmonisointia. PLM:n implementaatiota suunniteltaessa tarvittavien resurssien määrää on usein aliarvioitu [Sääksvuori 2011].

Yleisiä PLM:n haasteita Systems Engineeringiin liittyen:

- *Tuotteen kompleksisuus:* Tuhansia osia ja konfiguraatioita (tuotetyypit ja tuoteyksilöt), moniteknisuus ja monialaisuus, abstraktit tuotespesifikaatiot, sumeat ihmisvaatimukset, aineettomat tuotteet ja palvelut
- *Tuote-prosessien ja liiketoiminnan kompleksisuus:* Tuhansia PLM-transaktioita päivittäin, pieniä ja suuria yrityksiä, yritysverkostot, erilaiset osaamistasot, asenteet sekä ajattelutavat, erilaiset liiketoimintamallit, ketteryyden vaatimus, heterogeeniset prosessit ja menetelmät, muutostenhallinta ja pimeät muutokset, työnkierrot ja statukset, systeeminen ajattelu, takaisinkytkentä elinkaaresta
- *PLM:n elinkaari:* Yritysten strategiat vs. evoluutio, liiketoiminnan dynaamisuus, ylläpito ja kehittäminen
- *PLM:n IT-arkkitehtuuri:* Yrityksen sisäinen arkkitehtuuri sekä verkostot, heterogeeniset työkalut ja järjestelmät, monoliittinen vs. modulaarinen arkkitehtuuri, kaupalliset vs. in-house, useita toimittajia, erilaisia konfiguraatioita samastakin järjestelmästä, järjestelmä- ja työkaluintegraatiot
- *Informaation kompleksisuus:* Relaatiot objektien välillä, versio- ja revisiohallinta, eri dataformaattit, datan lisäksi muu informaatio (paperi-dokumentit, mallit, hiljainen tieto know-how, know-who jne.), abstrakti informaatio ja spesifikaatiot, rikkaat PLM-tietomallit, rinnakkaiset tuoterakenteet, datan laatu ja elinkaari, datan saanti, turvallisuus, omistajuus.

Organisaatiomielessä haaste PLM:n implementoinnissa suunnitteluosaston ulkopuolelle johtuu paljolti siitä että järjestelmät eivät usein tue tuotekehitysprosessin sumeaa alkupäätä ja abstrakteja, käsitteellisiä sekä monitekniisiä tuotteen määrittelyjä. Osittain tämä myös johtuu siitä että kyseistä prosessia ei ole edes määritelty yrityksessä. Tämän prosessin määrittely ja implementointi eivät ole helppoja, mutta ilman niitä IT-järjestelmistäkään ei kannata odottaa suuria apuja.

Puutteellisista PLM-implementoinneista on havaittu mm. seuraavanlaisia seurouksia [Leino et al. 2012]:

- Huonolaatuinen tuotetieto (ei voida käyttää IT-järjestelmässä, tuotetietoon ei luoteta, tuotetiedon huono jäljitettävyyys)
- Huono ennustettavuus (johtaminen vaikeaa, virheet toistuvat, osaoptimointi eri toimintojen välillä)
- Turvallisuus- ja ympäristöriskit
- Huonontunut asiakastyytyväisyys
- Epäkelpoja tuotteita tai komponentteja, väärä lukumäärä komponentteja

- Viiveet, tuotantokatkokset, uudelleen tekeminen
- Logistiikka-ongelmat
- Lisääntyneet kustannukset
- Sekoilu hankinnassa ja toimitusketjussa
- Tullimaksut, takuukustannukset, puutteelliset ohjeet.

Tuotetiedon virheistä kärsivät usein muut kuin niiden aiheuttaja sekä yrityksen sisällä että sidosryhmät elinkaaren eri vaiheissa. Pienetkin ongelmat voivat johtaa suuriin kustannuksiin vuositason tasolla.

2. Lääkkeitä kompleksisuuden hallintaan ja liiketoiminnan tehostamiseen

Edellä on kuvattu määrättyjä haasteita ja näkökulmia sosioteknisten järjestelmien monimutkaisuuteen ja kompleksisuuteen liittyen. Tässä luvussa pyritään antamaan joitain vinkkejä siihen, miten näitä haasteita voidaan voittaa ja jopa kääntää kilpailukyvyksi ja menestykseksi. Valittuja keinoja ja metodiikkoja ovat kompleksisten järjestelmien tekniikka, Systems Engineering, tuotteen elinkaaren hallinta, mallipohjaisuus ja virtuaaliprototyyppi, vaatimustenhallinta ja standardit. Näiden ja yleisesti muidenkin prosessien ja organisaatioiden implementointi ja kehittäminen verkostotasolla ovat tärkeitä lähtökohtia. Niiden kehittämiseen annetaan myös joitain vinkkejä.

2.1 Systeminen ajattelu käytännössä

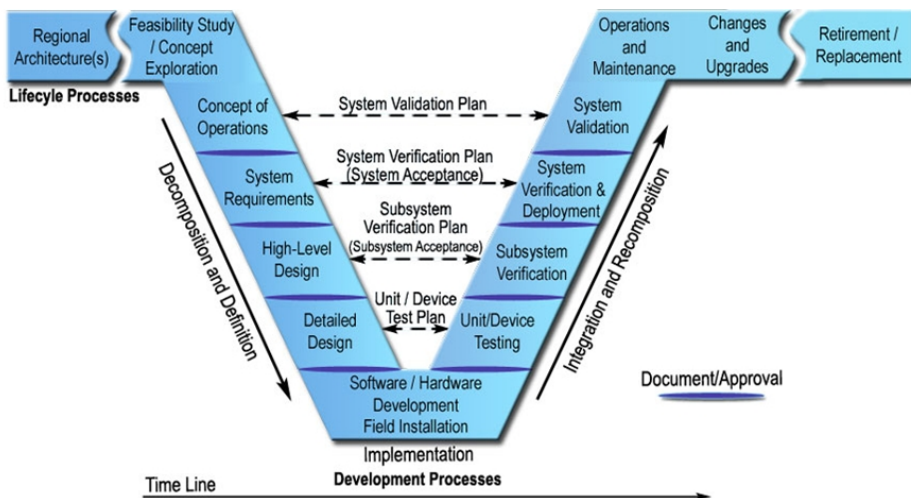
Sosioteknisen järjestelmän kompleksisuutta tulee lähestyä kokonaisvaltaisesti ja ymmärtää myös se ettei järjestelmän toimintaa pysty kaikilta osin selittämään analyyttisesti ja determinisesti. Systeminen ajattelu (system thinking) ja systeemi-teoriat (esimerkiksi [Hubka & Eder 1984]) pyrkivät ongelmanratkaisuun mallintamalla järjestelmiä ja ongelmia kokonaisvaltaisesti. Systems Engineering voidaan nähdä erääksi käytännönläheiseksi systeemisen ajattelun metodiikaksi. SE on standardoitu ja pitkälle vakioitu lähestymistapa moniteknisten kompleksisten järjestelmien suunnitteluun, jossa keskeisiä piirteitä ovat tuotetta koskevien tarpeiden, ominaisuuksien ja vaatimusten systemaattinen käsittely, vaatimuksiin vastaavien suunnitteluratkaisujen ja toteutusten toiminnallisuuden todentaminen, suunnittelu-päätösten jäljitettävyyden ja koko elinkaaren kattava tarkastelu. Järjestelmän kokonaisvaltainen käsittely sekä sisällöllisesti että ajallisesti tarkoittaa lukuisten sidosryhmien, suunnittelualojen ja teknisten ratkaisujen yhteensovittamista, mikä ilman edistyksellisiä tietoteknisiä järjestelmiä on käytännössä mahdotonta. SE:n keskeisiä piirteitä ovat:

- järjestelmällinen ja kattava vaatimusten määrittely ja hallinta
- suunnitteluratkaisujen ja toteutusten systemaattinen todentaminen eli verifiointi ja kelpuus eli validointi
- suunnitelman pilkkominen hallittaviin osa-ongelmiin (decomposition).

Systems Engineering aiheeseen liittyvää englanninkielistä kirjallisuutta on julkaistu vuosien saatossa runsaasti. Suomalaisesta näkökulmasta näissä teoksissa on yleensä eräitä keskeisiä puutteita:

- *Tuoteprosessit perustuvat tänä päivänä vahvasti verkottuneeseen toimintaan*, jossa eri suunnittelualojen ja elinkaarivaiheiden osapuolet toimivat yhdessä, usein hyvin dynaamisissa ja tapauskohtaisesti vaihtelevissa kokoonpanoissa. Systemaattiset suunnittelukäytännöt, järjestelmän elinkaarenhallinta ja niitä tukevien järjestelmien tulisi siis toimia tehokkaasti myös tällaisissa heterogeenisissa verkostoissa, jossa monien toimijoiden valmiudet voivat olla rajalliset.
- Kompleksisten järjestelmien tekniikan (SE) käytännöt, sellaisina kun ne on esitetty standardeissa ja käsikirjoissa, *koetaan yrityksissä yleisesti rasakaaksi kustannus- ja aikataulupaineita lisääväksi byrokratiaksi*, ja lupaukset säästöistä tai laadun parannuksesta epämääräisiksi.

Systems Engineeringin määrittelemän prosessin vaiheittaista etenemistä ja eri vaiheiden välisiä kytkentöjä on usein tapana kuvata niin sanotun V-mallin avulla (kuva 1). Varsinkin eräiden yksinkertaistettujen V-mallien eräänlaisena heikkoutena voidaan ehkä pitää sen suoraviivaista kaavamaisuutta ja rajoittumista yksittäisen tuotteen tai tuotesarjan toteuttamiseen suunnittelusta valmistukseen. Kuvasta jää silloin puuttumaan monia SE:lle keskeisiä piirteitä, kuten elinkaaren alku ja loppujaksot, järjestelmän liittyminen muihin järjestelmiin, eri suunnittelualojen väliset rajapinnat (rinnakkaissuunnittelun tarve) jne. Tuotekehitys lähtee harvoin tyhjältä pöydältä ja yhä harvemmin koko prosessi on saman yrityksen hallinnassa. SE:n soveltaminen monivaiheisessa arvoketjussa muodostaa oman haasteensa.



Kuva 1. Systems Engineering V-malli [U.S. Department of Transportation 2007].

Vaikka SE-toimintatapa perustuu standardeihin, eivät sen käytännön sovellukset eri organisaatioissa ole yhteneväisiä. SE-toimintatapa esitetäänkin skaalautuvana menettelynä. Skaalautuminen voi tarkoittaa erilaista SE-prosessien soveltamista, tuotearkkitehtuureja, vaatimusten hallintatasoja, organisatorista toteutusta jne. Ei ole olemassa yhtä ainoaa SE-sovellustapaa. SE-käytäntöjen toteuttamistapa on aina riippuvainen tavoitteista, ulkoisista ja sisäisistä ajureista, sovelluskohteen vaatimuksista, kulloisestakin tilanteesta, organisaation osaamistasosta (kypsydestä) sekä käytettävissä olevista resursseista. Myös organisaation sisällä voidaan soveltaa erilaisia prosesseja uuden kehittämiseen, vanhan modifiointiin ja valmiin tuotteen käyttämiseen tai eri tuoteryhmiin.

Aloilla, joilla tuotteiden käyttöön liittyy suuria riskejä, vaaditaan luotettavuuden ja turvallisuuden varmistamista. Hankalasti huollettavalle tuotteelle voidaan asettaa korkea käytettävyysovaatimus. Näissä tapauksissa usein asiakas voi vaatia SE-prosessien noudattamista, mutta SE-menettelyä voidaan soveltaa myös tukemaan viranomaisvaatimusten täyttämistä. Tällöin SE:n soveltaminen määrittyy yleensä alan yleisten käytäntöjen ja asiakkaan hyväksymän tason mukaan. Toisaalta SE:n soveltamiseen voidaan myös lähteä yrityksen sisäisestä tavoitteesta esim. systematisoida ja harmonisoida suunnittelukäytäntöjä, kun tuotteet monimutkaistuvat, suunnittelu hajautuu ja on vaikea hallita kokonaisuutta. SE ja mallipohjainen suunnittelu tukevat toisiaan. SE:tä soveltamalla voidaan myös tukea vanhojen tuotteiden ylläpitoa. Kun kiinnostus SE:n soveltamiseen lähtee yrityksen sisältä, voidaan SE:n sovellustaso määritellä siten, että se tukee parhaiten yrityksen omia tavoitteita.

SE-käsitettä on myös pyritty täsmentämään erilaisin määritelmien. Eri määritelmät eivät yleensä ole ristiriidassa toistensa kanssa vaan tarve omalle määritelmälle liittyy yleensä johonkin toimialan erityispiirteeseen, jonka suhdetta SE lähestymistavalle on ollut tarpeen korostaa. Yhtenäistä eri SE-kuvauksille on tavoite kuvata prosessi, jonka tarkoituksena on hallita kompleksisten järjestelmien suunnittelu ja toteutus siten, että järjestelmälle asetettavien vaatimusten täyttyminen lopputuotteessa varmistetaan. Suunnittelussa huomioidaan kaikkien elinkaaren vaiheiden vaikutukset ja sidosryhmien näkökulmat. Vaatimustenhallinnan korostaminen ja järjestelmän pilkkominen hallittavampiin osajärjestelmiin on keskeisiä lähestymistapoja. Systems Engineering -prosessia kehitetään myös jatkuvasti sen perinteisillä teollisuusaloilla yrityksen tai toimialan omista lähtökohdista (NASA, MITRE¹ jne.).

Yritykset ovat erilaisia ja ne toimivat erilaisilla toimialoilla ja hyvinkin erilaisin toimintatavoin. Tällä on myös suora vaikutus siihen kuinka Systems Engineeringiä tulisi eri tapauksissa lähestyä ja kuinka se kannattaisi eri organisaatioissa toteuttaa. Joissakin tapauksissa SE nähdään enemmän kaikkia yhdistävänä toimintona, laatuprosessina vastaavana prosessina, ja joissakin tapauksissa taas hyvin teknisenä (tai jopa tieteellisenä) suunnitteluprosessin johtamismenetelmänä.

¹ <http://www.mitre.org>

Eriaisissa käytännön Systems Engineering -sovelluksissa ihmiset toimivat erilaisissa rooleissa. Roolien tulisin olla linjassa sen suhteen, missä luokassa yritys useimmiten tai pääasiassa toimii.

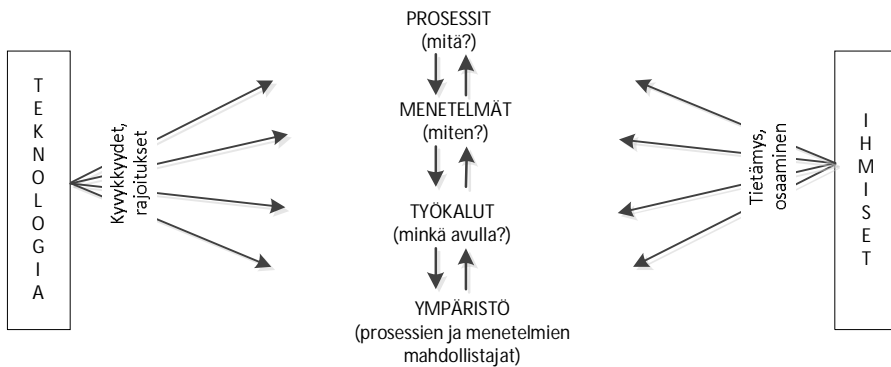
Aracic [2012] tunnistaa SE:n hyväksikäytölle erilaisia ohjaavia periaatteita kommunikaatiolähtökohdasta: valitaan missä perusprosesseissa (vaatimusten määrittely, arkkitehtuuri, analyysi ja allokointi, systeemin hallinta) pääosin käsitellään tietoja tai dokumentteja, joita vaihdetaan eri osapuolien välillä. Näin tunnistetaan kolme hyväksikäyttö tapaa, jotka eivät ole toisensa poissulkevia ja voivat toimia myös toistensa yhdistelmänä:

- *Vaatimuslähtöinen SE*: Projektissa on suuri määrä alihankkijoita, joilla on kaikilla hyvin erilaiset osaamisalueet. Tällöin vaatimusten hallinta -prosessi yleensä valitaan keskeiseksi ja keskeiset kommunikoitavat tiedot liittyvät vaatimuksiin, sopimuksiin ja oikeaksi osoittamiseen.
- *Mallipohjainen SE*: Referenssiprosessiksi valitaan arkkitehtuuri, analyysi ja allokointi. Mallipohjaisuutta pidetään tapana, jolla saadaan suurimmat tuotavuushyödyt.
- *Palautelähtöinen SE*: Projekti käyttää "ketterää" lähestymistapaa, säännöllistä validointia ja verifointia ennen seuraavan vaatimusjoukon määrittelyä.

2.2 Prosessit, toimintatavat ja työkalut kuntoon

Systems Engineering on määritelty prosesseina mm. useissa standardeissa. Myös yleiseen tuotekehitykseen ja arvoketjuihin liittyviä referenssimalleja [Esim. Ulrich & Eppinger 2003] löytyy kirjallisuudesta lukuisia. Yrityksissä on nykyään yleensä kuvattu prosesseja vähintäänkin laatukäsikirjassa esimerkiksi projektien päätöksenteon ja dokumentoinnin tasolla. Prosessit saattavat kuitenkin olla liian yleisellä tasolla määriteltyjä tai ne eivät kohtaa todellista tekemistä. Prosessit voivat myös olla liian jäykkiä ja raskaita, mikä ei tietenkään ole tarkoitus. Hyvät ja virtaviivaiset prosessit tukevat ja nopeuttavat tekemistä, mutta niiden määrittely vaatii aikaa ja investointeja, sekä sidosryhmien osallistamista. Ilman riittäviä prosesseja ja toimintatapoja aikaa ja rahaa tuhlaantuu siihen että toistuvasti ratkaistaan samoja ongelmia, odotetaan jotain tai etsitään tietoa. Pitää myös muistaa että prosessit ovat dynaamisia ja niitä pitäisi päivittää jatkuvasti kuin toimintaympäristö muuttuu.

Prosesseista puuttuu usein tarkemmat määrittelyt, kuten niihin liittyvät menetelmät ja työkalut, sekä syötteiden ja tulosten määrittely. Prosessit saatetaan kuvata lohkokaavioina, mutta ei sitä mitä kaavion laatikoiden sisällä ja niiden välillä tapahtuu. Menetelmien ja työkalujen määrittely on erityisen tärkeää mallipohjaisen Systems Engineeringin (MBSE) kohdalla. Prosesseihin, menetelmiin ja työkaluihin liittyy olennaisesti myös mallien ja muun informaation hallinta esimerkiksi PDM-järjestelmässä.



Kuva 2. Teknologian ja ihmisten vaikutusten elementit. [Estefan 2008].

Nykytilanteessa prosesseja tukevia IT-ratkaisuja eri tarkoituksiin on paljon tarjolla ja voi olla vaikeus valita niistä mahdollisimman sopiva. Useat järjestelmät ovat monimutkaistuneet, kun niitä voidaan konfiguroida käyttöön monella tavalla. Tällöin keskeiseksi ongelmaksi usein nousee, miten tietotekniikkaa pitäisi kussakin tapauksessa soveltaa. On syytä pohtia sitä, mitä tavoitteita sen soveltamisella halutaan saavuttaa. IT-ratkaisujen pitäisi lähtökohtaisesti tukea yrityksen prosesseja, mutta käyttöönotto voi myös muuttaa ja parantaa organisaation toimintaprosesseja (kuva 2). Vastuut, roolit ja päätöksenteko voivat muuttua, organisaatioiden välinen vuorovaikutus ja riippuvuudet lisääntyvät. Keskeinen haaste on saada prosessi jalkautettua ja organisaatio ymmärtämään se. Muutoksen aikaansaaminen vaatii, että organisaation toimijat ymmärtävät tavoitteet ja uudet toimintatavat ja sitoutuvat niihin. Tietotekniikan tai SE:n käyttöönotto aiheuttaa usein lisätyötä tietyissä suunnitteluvaiheissa ja hyödyt eivät välttämättä näy niille, joiden pitää tehdä tämä lisätyö. Jos SE:n soveltamisella lähinnä vastataan ulkoisiin vaatimuksiin, eikä mietitä sen soveltamista omista lähtökohdista, voivat hyödyt omalle organisaatiolle jäädä näkymättömiksi.

Yhtenä johtamiseen liittyvänä haasteena on käytännössä asenteiden muuttaminen. Nykyisin esimerkiksi suunnittelijat eivät usein halua näyttää ”keskeneräistä” työtä muille, vaikka palautteen saaminen esimerkiksi tuotannosta mahdollisimman aikaisin olisi tärkeää. SE:n liittyvä rinnakkaissuunnittelu edellyttää keskeneräisen työn jakamista ja läpinäkyvyyttä suunnittelun aikana.

2.3 Organisaatioiden ja roolien näkökulma

Systems Engineer nimikkeellä olevaa henkilöä organisaatiossa on joskus kuvattu orkesteriaan johtavaksi maestroksi, joka tietää miltä musiikin tulisi kuulostaa (”ulkonäkö ja toiminnallisuus”) ja jolla on taito ohjata orkesteriaan (”tiimiään”) halutun kuuluisen lopputuloksen saamiseksi (”vaatimusten saavuttaminen”).

Systems Engineeringissä on kyse ennen kaikkea oikean suunnitelman löytämisestä ja määrittämisestä ja vasta tämän jälkeen teknisen yhtenäisyyden ylläpitämi-

sestä ja kompleksisen prosessin johtamisesta oikein prosessein. Tällöin Systems Engineer voidaan jakaa kahteen eri ryhmään:

- *Tekninen johtajuus* – Systems Engineering -osaaminen – keskittyen järjestelmän tekniseen suunnitteluun ja eheyteen koko elinkaaren yli
- *Systeemien johtaminen* – Systems Engineering -tiede – keskittyen useiden teknisten disipliinien, organisaatioiden ja ihmisten muodostaman erittäin teknisen ja monitahaisen toiminnan johtamiseen.

Organisaatiot ovat rakentuneet tyypillisesti seuraaviin vaihtoehtoihin.

- *Funktionaalinen*: Perinteinen ja hyvin yleinen lähestymistapa on jakaa organisaatio toimintojen mukaan.
- *Projekti*: Tässä lähestymistavassa keskiössä ovat yksittäiset projektit toimintojen sijaan.
- *Matriisi*: Matriisiorganisaatio on tietynlainen välimuoto kahdesta edellä mainitusta. Projektitiimin jäsenet raportoivat sekä projekti- että toimintoihin perustuvalla linjajohdolle.
- *Projektitoimisto*: Tässä lähestymistavassa projektijohto, SE ja suunnittelu tiimit on organisoitu projektien mukaan, jotka pyytävät tarvittaessa apuaan eri toiminnoissa työskenteleviltä asiantuntijoilta.
- *Integroitu tuotetiimi*: Tällainen tiimi koostuu sekä asiakkaan, että toimittajan edustajista, jotka työskentelevät yhdessä projektin tarpeita vastaavan järjestelmän kehittämiseksi.

Prosessien soveltamiseen hajautetussa organisaatiossa ei ole olemassa yhtä ratkaisua, vaan toimivin käytäntö riippuu aina tavoitteista ja tilanteesta. Periaatteessa vastuiden määrittelylle ja jakamiselle voidaan erottaa kaksi "ääripäävaihtoehtoa":

- Verkostoyritys toimii päähankkijana tai päähankkijan kumppanina ja on vastuussa monimutkaisen tuotekokonaisuuden, systeemin, kehittämisestä, suunnittelusta ja mahdollisesti valmistuksestaakin. Tuotteen vaatimukset voivat tulla annettuina asiakkaalta tai yritys on itse vastuussa niiden tunnistamisesta ja määrittelystä. Yritys vastaa myös ko. kokonaisuuden vaatimusten verifioinnista ja validoinnista.
- Verkostoyritys on vastuussa tuotteen osan tai moduulin suunnittelusta tai valmistuksesta. Vaatimukset tulevat loppuasiakkaalta tai päähankkijalta.

Nämä vaihtoehdot ovat luonnollisesti ääripäitä ja tyypillisesti sopiva ratkaisu löytyy niiden väliltä. Verkostoympäristössä on kuitenkin syytä tarkastella kriittisesti, mitä vaatimuksia eri osapuolille kannattaa asettaa, koska liian suuret vaatimukset pienelle yritykselle voivat aiheuttaa viivästyksiä ja aina ei prosesseja saada tällöin toimimaan. Valintaan vaikuttaa toki asiakaskunta ja vaatimustaso; jos nähdään välttämättömäksi, että yhteistyökumppanit omaksuvat SE-käytännöt, tulee varata aikaa ja resursseja käytäntöjen rakentamiseen.

On yleisesti tiedossa, että ihmiset kommunikoivat parhaiten yhteisissä tiloissa, joissa on tilaa kasvokkain tapahtuvalle kanssakäymiselle. Nykyinen hyvin hajautunut toimintaympäristö ei kuitenkaan välttämättä mahdollista tätä. Tämä on haaste, jolle on löydettävä ratkaisu varsinkin monimutkaisia järjestelmiä kehitettäessä. Systems Engineering osaltaan vastaa tähän haasteeseen, tarjoamalla toimintamallin, jolla pyritään takaamaan, että tarvittava tieto on käytettävissä, missä ja milloin sitä tarvitaan. Mutta SE:n soveltaminenkaan ei välttämättä ratkaise ihmisten väliseen kommunikaatioon liittyviä haasteita. Näitä haasteita voidaan tunnistaa hajautetun projektinhallinnan näkökulmasta:

- selkeä määritelmä ratkaistavasta ongelmasta – jaettu yhteinen ymmärrys siitä, mitä suunnitteluprojektilla ollaan ratkaisemassa
- selkeä määritelmä kohteesta, tavoiteltavasta ratkaisusta – jaettu yhteinen ymmärrys siitä, mihin suunnitteluprojektilla tähdätään
- yhteinen ymmärrys päätöksenteon säännöistä – käytäntöjen läpinäkyvyys.

2.4 Vaatimustenhallinnan prosessi

Vaatimustenhallinta on tunnistettu yhdeksi keskeiseksi prosessiksi, joka on tällä hetkellä kehityskohteenä useassa yrityksessä. Vaatimustenhallinnan yhteydessä tulisi aina pohtia myös muutostenhallinnan prosessia, koska vaatimukset ja niitä vastaavat ratkaisut muuttuvat. Vaatimustenhallinta koostuu useasta eri tehtävästä, joista tärkeimmät on lueteltu seuraavassa:

- *Vaatimusten kartoitus* (sidosryhmien tunnistaminen; asiakkaan, lainsäädännön ja muiden sidosryhmien vaatimusten esiin kaivaminen ja niiden systemaattinen kirjaaminen).
- *Vaatimuksen kuvaaminen* (hyvälaatuisten vaatimuslauseiden; vaatimuslauseurunkojen hyväksikäyttö; mallien ja diagrammien käyttö).
- *Vaatimusten todentaminen ja kelpuuttaminen* (tarkistetaan, että vaatimukset ovat hyvälaatuisia ja että ne kuvaavat sen järjestelmän ja palvelun, mitä asiakas on haluamassa).
- *Vaatimusten analyysi* (vaatimusten muuntaminen insinöörimaailmaan sellaisessa muodossa, että järjestelmä voidaan niiden perusteella suunnitella; usein tässä vaiheessa vaatimuksiin tulee mitattavia ilmaisuja).
- *Vaatimusten tarkentaminen ja kohdentaminen järjestelmäarkkitehtuuriin* (kohdennetaan laitteistoon ja ohjelmistoon yksityiskohtaisia toteutettavissa olevia vaatimuksia).
- *Vaatimusten luokittelu ja kommunikointi* (erilaisia näkymiä vaatimusmassasta erilaisissa muodoissa eri tahoille, kuten alihankkijoille tai sidosryhmille).

- *Vaatimusprosessin ja vaatimusten elinkaaren hallinta* (vaatimusten numeerointi ja organisointi, tallennus ja versionhallinta sekä priorisointi; tuki suunnitelman ja tuotteen todennukseen ja kelpuutukseen; vaatimusten työnkulkujen tuki; jäljitettävyysetkijujen hallinta; vaatimusten luojien ja käyttäjien yhteistyön hallinta).

Tässä yhteydessä paneudutaan vaatimusten luokitteluun ja sitä kautta osittain myös vaatimusten kommunikointiin. Tämä valinta on tehty siksi, että käytännön projekteissa vaatimusten luokittelun on havaittu olevan ongelmallista, mutta siitä huolimatta siihen annetaan verraten suppeita ohjeita kirjallisuudessa. Hyvin suunnitellun luokittelun merkitys on suuri, kun vaatimuksia on useita satoja. Luokittelua voidaan tehdä mm. vaatimuksen tyyppin, lähteen, kohteen, tieteenalan, prioriteetin, kriittisyyden, elinkaaren tai toteuttajan mukaan. Lisäksi luokittelua voidaan tehdä sen perusteella onko vaatimus rajoite, sidosryhmävaatimus tai prosessivaatimus.

Vaatimustenhallintaan on olemassa lukuisia ohjelmistotyökaluja. Kun mukana on myös useita osapuolia ja toimijoita eri suunnittelualoista, vaatimusten jäljittäminen ja ylläpitäminen on hyvin haastavaa.

Vaatimukseen liittyy kaksi keskeistä käsitettä:

- Verifiointi: yksittäisen vaatimuksen todentaminen.
- Validointi: palveleeko kokonaisuus alkuperäistä tarvetta.

Vaatimustenhallinnan oikean soveltamistason löytäminen on haaste. Esimerkiksi jos vaatimuksia tarkastellaan liian yksityiskohtaisella tasolla, hukutaan yksityiskohtiin ja kokonaisuus ei hahmotu. Toisaalta liian suurina "nippuina" käsiteltävien vaatimusten jäljittäminen, validointi ja muutosprosessit voivat käydä vaikeiksi.

2.5 Tuotteen elinkaaren hallinta – PLM

Tämän luvun tavoitteena on kuvaus siitä, mikä on SE:n (systems engineering) ja PLM:n (tuotteen elinkaaren hallinta) keskinäinen suhde prosessien, työkalujen, tietomallien ja järjestelmien näkökulmasta, sekä mitä edellytyksiä PLM:ltä vaaditaan SE-käytäntöjen toteuttamiseen. Käsitteiden ja niiden välisien relaatioiden tunnistamisessa hyödynnetään kirjallisuudesta, yrityshaastatteluista ja omista havainnoista saatua dataa. Tämän opuksen PLM-määritelmä perustuu amerikkalaisen puolueettoman PLM-alueen tutkimus- ja konsultti-organisaation CIMdatan² määritelmään:

- "A strategic business approach that applies a consistent set of business solutions that support the collaborative creation, management, dissemination, and use of product definition information.

² www.cimdata.com

- Supporting the extended enterprise (customers, design and supply partners, etc.).
- Spanning from concept to end of life of a product or plant.
- Integrating people, processes, business systems, and information”.

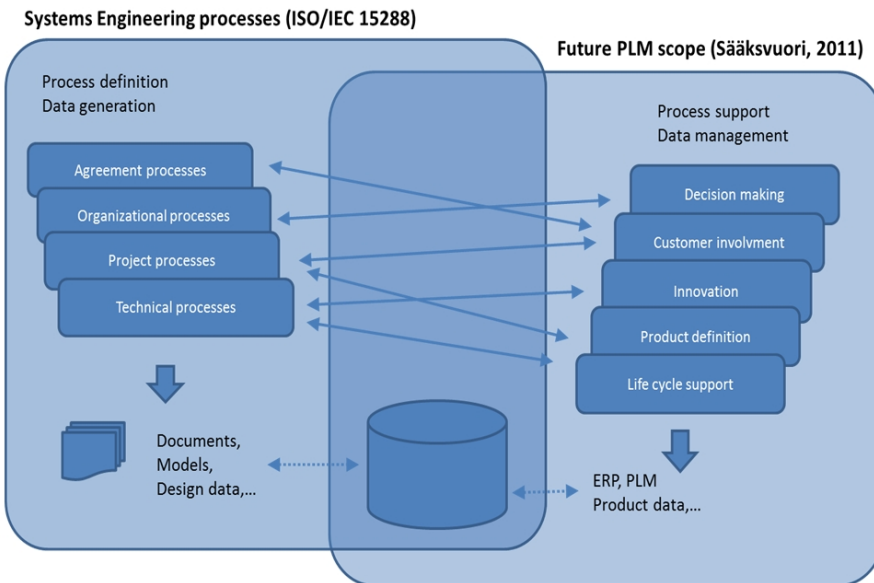
”PLM on siis liiketoiminnan lähestymistapa, joka hyödyntää johdonmukaisesti erilaisia sovelluksia informaation luomiseen, hallintaan, jakamiseen ja hyödyntämiseen. PLM tukee verkostoitunutta toimintatapaa, kattaa koko tuotteen elinkaaren sekä integroi yhteen ihmiset, organisaatiot, prosessit, järjestelmät ja informaation.” CIMdatan määritelmän mukaan on huomattava, että PLM ei ole vain teknologinen asia.

PLM:n näkökulmasta SE on monella tavalla haasteellinen menetelmä. Se on lähtökohtaisesti monitekninen ja holistinen lähestymistapa, jonka kohteena ovat tyypillisesti mekatroniset tuotteet (järjestelmät), jotka pitävät sisällään esimerkiksi mekaniikkaa, automaatiota, ohjelmistoja jne. Näiden tekniikan osa-alueiden suunnitteluun on kehittynyt omat menetelmät ja työkalut. Osa-alueiden tuotetieto on myös hyvin erityyppistä ja on eri formaateissa. Tuotetieto voi olla lukuisissa eri formaateissa: digitaalisena, paperilla, hiljaisena tietona, piirustuksina, 3D-malleina, kuvina, videoina jne. Eri elinkaaren toimijat ja prosessit tuottavat erityyppisiä spesifikaatioita ja ymmärtävät tuotteen, sen piirteet ja ominaisuudet, sekä tuotetiedon omasta näkökulmastaan hieman eri tavoilla.

Jotta monitekninen tuote (järjestelmä) voidaan suunnitella parhaalla tavalla, se pitää ymmärtää myös abstraktilla tasolla niin että se pystytään mallintamaan ja digitalisoimaan. Järjestelmän osajärjestelmiä ei tule ajatella tekniikan osa-alueina, vaan tiettyjä vaatimuksia, toimintoja, logiikoita ja moduuleita toteuttavina osajärjestelminä. Kun esimerkiksi mekaniikassa on tuotetta perinteisesti lähestytty osarakenteen kautta, tarvitaan kokonaan erilainen filosofia siirryttäessä kohti mekatronista tuoterakennetta. Systems Engineeringin omaa aluetta on moniteknisyyden lisäksi tyypillisesti suunnitteluprosessin sumea alkupää, jossa järjestelmän kuvaukset ovat usein hyvin abstrakteja lähtien asiakkaan tarpeista ja asiakasvaatimuksista kohti toimintorakenteita, loogista rakennetta ja viimein osarakennetta, sekä suunnitteluratkaisujen todentamista (verifiointia) ja järjestelmän kelpuutusta (validointia). Abstraktien rakenteiden, ja niihin liittyvän suunnittelumenetelmien tuotetiedon ristikkäinen hallinta PDM/PLM-järjestelmissä on haasteellista.

SE standardien näkökulmasta PLM:n tulisi tukea teknisten prosessien lisäksi projektiprosesseja, sopimusprosesseja ja liiketoimintaprosesseja. Taso, jolla standardeja implementoidaan PLM-prosesseiksi, riippuu kunkin yrityksen politiikasta. Prosessien käytännön toteuttamisen puitteissa PLM:n pitää antaa tukensa myös niihin liittyvien menetelmien ja työkalujen käyttöön perinteisen CAD-integraation lisäksi. Monimutkaisien ja moniteknisten järjestelmien rinnakaissuunnittelussa erilaisia sidosryhmiä sisältävissä verkostossa tarvittaisiin automatisoituja ja varmatoimisia prosesseja ja työnkiertoja tuotetiedon hallintaan ja jakamiseen reaaliaikaisesti sekä kyvykkäitä kommunikaatiovälineitä.

Uuden tekniikan tai uusien menetelmien käyttöönotto organisaatiossa tuo mukanaan monia haasteita. Mitä monimutkaisemmasta järjestelmästä on kyse, ja mitä useampia organisaation osia ja toimijoita se koskee, sitä suuremmat haasteet todennäköisesti ovat. Tyypillisesti esim. PLM-järjestelmiä pidetään monimutkaisina ja niiden implementointi vaatii usein valtavan projektin ja pitkän kalenteriajan. SE-prosessien implementoinnin haastavuus riippuu siitä, millä vaativuustasolla ja roollituksella ne toteutetaan; miten moniin osapuoliin ja toimijoihin se vaikuttaa. Tietotekniikan kehitys on ajan kuluessa siirtynyt konfiguroitavien tietojärjestelmien suuntaan. Työkaluja käytetään enenevästi selaimen kautta, joten aina ei tarvita edes työasemakohtaisia asennuksia ja päivityksiä. Varsinainen tekninen käyttöönotto on siten yleensä helpottunut. Kehityskulun tavoitteena on, että tarvittavat palvelut ja jopa data olisivat helposti saatavissa ”pilvestä” ja että käyttäjät itse voisivat enenevässä määrin muokata järjestelmän kunkin yrityksen tarpeisiin sopivaksi.



Kuva 3. Systems engineeringin ja PLM:n prosessien samankaltaisuuksia.

Yksi keskeinen tutkimusaihe ja kehityskohde liittyen PLM:ään ja SE-implementointiin on erilaisten tuotetiedon näkymien luominen eri sidosryhmien ja toimintamallien tarpeisiin (kuva 3). Tämä vaatii PLM-installaatiolta kyvykkyyttä hallita rinnakkaisia tuoterakenteita ja niiden välisiä relaatioita ja muutoksia. Esimerkiksi suunnittelu, valmistus ja vaikkapa virtuaaliprototyyppiin tarvitsevat erilaiset rakenteet. Sääksvuoren [2011] mukaan PLM:n merkittävät tulevaisuuden trendit ja tutkimusaiheet liittyvät tuotteen ja tuoteyksilön aitoon elinkaaren hallintaan ja tukemiseen, takaisinkytkentään eri elinkaaren vaiheista ja sidosryhmistä ja innovaatioprosessien tukemiseen sekä asiakkaan ja muiden sidosryhmien kannalta. Samoin aiheet

liittyvät 3D-mallien ja virtuaalimallien kyvykkyyksien ja merkityksen kasvamiseen, tietomallien kehittymiseen tukemaan paremmin moniteknisistä ja abstrakteja kuvauksia tuotteista sekä kevyempiin IT-ratkaisuihin PK-sektorin tarpeisiin (esim. pilvipalvelut).

Kuten edellä on todettu, tuotekehitys ja -suunnittelutoiminta ovat nykypäivänä usein hyvin verkostoitunutta ja hajautunutta toimintaa. Tuotekehitystiimit koostuvat yhä useammin eri koulutuksen ja taustan omaavista, eri tiloissa ja jopa organisaatioissa työskentelevistä ihmisistä, jotka käyttävät työhönsä erilaisia työkaluja ja järjestelmiä. Tietotekniikassa määritellään, että organisatorinen käyttöönotto sisältää koko prosessin lähtien tarpeiden analyysistä ja teknisen ratkaisun valinnasta siihen, kun teknologian tuoma hyöty on toteutunut organisaation toiminnassa [Munkvold 1999]. Se ei siis rajoitu teknisen järjestelmän määrittelyyn, toteutukseen ja asentamiseen, vaan sisältää myös ”todellisen käytön”, käyttäjien osallistamisen, koulutuksen ja sitouttamisen käyttöön. Usein käyttöönotto tehdään askelittain tai vaiheittain; tuoteryhmä, moduuli tai organisaation osa kerrallaan, jolloin seuraavassa vaiheessa voidaan käyttää hyväksi edellisten vaiheiden oppia. Myös SE-toimintatavan käyttöönotto osana PLM:ää vaatii tarpeiden tunnistamista ja tavoitteiden asettamista. Se muuttaa toimijoiden toimintatapoja ja prosesseja ja vaikuttaa eri osapuolien yhteistyöhön. Se vaatii uuden osaamisen ja uusien roolien kehittämistä organisaatioon. Jotta SE-prosessi voidaan viedä läpi, toimijoiden pitää ymmärtää sen tavoitteet ja sitoutua sen toimintatapoihin. Tätä tukee osapuolien osallistaminen ja tiedotus hankkeen aikana.

PLM:n kehityksessä IT-arkkitehtuurimielessä on nähtävissä monenlaisia trendejä (ks. esim. Sääksvuori [2011]). Yhtenä suuntauksena on pyrkimys kohti yhtä monoliittista järjestelmää. Toinen suunta on modulaarinen arkkitehtuuri ja kolmantena voidaan nähdä erilaiset ”pilvi-PLM” palvelut. Sopiva strategia arkkitehtuurin kehittämiseen riippuu kunkin yrityksen liiketoiminnan luonteesta. SE:n näkökulmasta tällä hetkellä on kuitenkin vaikea löytää yhtä järjestelmää, joka kykenee hallitsemaan samanaikaisesti mekatronisen tuotteen monien suunnittelualojen informaatiotyypit ja rakenteet sekä objekteja liittyen esimerkiksi palveluliiketoimintaan. PLM-arkkitehtuuri sisältää liiketoiminnan näkökulmasta laajasti:

- *Prosessit*: Yhteiset ja yhteneväiset puitteet tekemiselle toistettavasti – milloin ja mitä tehdään.
- *Menetelmät*: Menettelytavat ja tehtävät – miten tehdään
- *Informaatio*: Kaikki tuotteeseen liittyvä data, informaatio ja tietämys
- *Infrastruktuuri*: Ihmiset, osaaminen ja IT
- *Työkalut*: Tukevat menetelmiä.

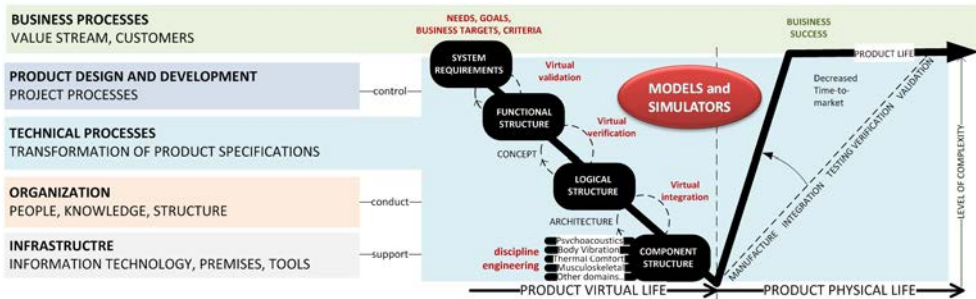
PLM-arkkitehtuuriin liittyvä haaste on myös se, että markkinoilla olevat kaupalliset järjestelmät ovat usein raskaita ja kalliita PK-yritysten tarpeisiin ja resursseihin nähden. Perinteisesti erityisesti koneenrakennuksessa alihankkijat ovat toimittaneet päämiehen määrittelyjen mukaisia osia tai osajärjestelmiä. Kuitenkin, toimiminen osana SE-verkosta vaatii tulevaisuudessa usein myös PK-yrityksiltä kyvykkyyttä osallistua PLM:n prosesseihin ja informaation jakamiseen [Sääksvuori 2011].

Alihankkijoita halutaan myös entistä enemmän mukaan tuotekehitykseen ja innovointiin. Silloin vaihtoehtoina voi olla esimerkiksi kytkeytyminen päämiehen IT-järjestelmiin ja prosesseihin tai omien PK-sektorille suunnattujen kevyiden ja helpokäyttöisten järjestelmien käyttöönotto. Haasteena tällä hetkellä on kuitenkin se, että monilla PK-yrityksillä (alihankkijoilla) on useita päämiehiä. Toisaalta markkinoilla ei juuri ole PK-yrityksille suunnattuja, helposti käyttöönotettavia, edullisia järjestelmiä. PLM:n IT-arkkitehtuuriin liittyen on huomattava myös että vaikka verkoston eri osapuolilla olisi käytössä sama kaupallinen PLM-järjestelmä, niiden tietomalli ja konfiguraatio voi olla erilainen, jolloin suora yhteensopivuus ei toteudu.

2.6 Mallipohjaisuus

Mallipohjaisuus on yleistymässä teollisuudessa. Tämä ilmiö näyttää liittyvän osaltaan lisääntyvään kiinnostukseen Systems Engineering lähestymistapaa kohtaan. Ilmiö liittyy myös PLM:n laajenemiseen suunnitteluprosessin alkupäähän ottaen haltuun SE:n osa-alueita ja tuotteeseen liittyviä malleja. Toisaalta mallipohjaisuuteen laajasti ajateltuna liittyvät teknologiat alkavat olla riittävän kypsiä yrityskäyttöön, jolloin kehitystä kohdistetaan enemmän prosessien, menetelmien, organisaatioiden, informaation hallinnan ja infrastruktuurin kehittämiseen.

Yritysten strategioita ohjaa tarve parantaa innovaatio-kyvykkyyttä ja kompetenssia yleensä. Asiakkaiden ja muiden sidosryhmien tarpeiden ja tavoitteiden todellinen ymmärtäminen ja todentaminen ovat nousseet yhdeksi keskeiseksi ydinkompetenssiksi niin B2B kuin kuluttajatuote puolella. Tuotekehitystä ohjaavia tekijöitä ovat mm. käyttäjäkokemus, turvallisuus, päivitettävyyys ja kestävä kehitys. Markkinoiden dynamiikka lisääntyy jatkuvasti ja monitekniset tuotteet sekä arvoverkot tulevat jatkuvasti monimutkaisemmaksi. Tuotteen elinkaaren hallinta lisää systeemin monimutkaisuutta, jolloin ristiriitaisia tavoitteita ja vaatimuksia esitetään eri sidosryhmien tahoilta. Samanaikaisesti laadukkaita, personoituja, konfiguroitavia ja kustannustehokkaita tuotteita tulee pystyä toimittamaan varmasti ja yhä nopeammin. Tuotteiden ja kokonaisjärjestelmiä tulee pystyä kehittämään ketterästi. Näiden kaikkien ilmiöiden ymmärtämiseen ja hallitsemiseen mallipohjaisuus ja siihen liittyvät uudet teknologiat ja lähestymistavat voivat tuoda helpotusta. Mallipohjaisuus mahdollistaa myös sidosryhmien osallistamisen järjestelmien kehittämiseen.



Kuva 4. Mallipohjaisuuden sijoittuminen liiketoiminnan prosesseihin.

Prosessit ovat luonteeltaan usein dynaamisia. Koska inhimillinen toiminta pitää sisällään myös epärationaalisuutta ja epäennustettavuutta, järjestelmiä ja prosesseja tulee tutkia ja kehittää myös johtamisen, sosiologian, politiikan yms. tieteiden keinoin. Kuva 4 havainnollistaa mihin SE:n tekniset prosessit sijoittuvat liiketoiminnan yhteydessä. MBSE:n päätavoite on tukea teknisiä prosesseja, mutta niillä on liityntä myös monimutkaisiin liiketoimintaprosesseihin, projekti- ja TK-prosesseihin, organisaatiomalleihin, tiedonhallintaan sekä muuhun infrastruktuuriin. On siis tärkeää mallintaa sekä tekninen järjestelmä että sen kehittämiseen osallistuvat prosessit menetelmineen, työkaluineen ja tekijöineen. Teknisiä prosesseja ylemmältä tasolta nähtynä järjestelmän määrittelyssä ja kehittämisessä on neljä vaihetta:

1. Järjestelmän kontekstin ymmärtäminen ja määrittely: reunaehdot, rajapinnat, ihmiset ja organisaatiot sekä vuorovaikutukset
2. Järjestelmän käyttäminen; ketkä käyttävät ja miten
3. Järjestelmän toteuttaminen
4. Käyttö ja elinkaarituki.

2.7 Mitä ovat mallit?

Insinööriäitöön on aina kuulunut erilaisten mallien (pienoismallit, piirustukset, laskelmat, digitaaliset mallit jne.) käyttö. MBSE eli mallipohjainen SE määritellään eri lähteissä hieman eri tavoin. Jotkut rajaavat MBSE:n määrittelyn lähinnä kuvailevien (deskriptiivisten) mallien käyttöön järjestelmän vaatimusten ja konseptien kuvaamiseen. Toiset näkevät sen hyvinkin laajasti kaikenlaisten mallien hyödyntämisenä järjestelmän elinkaaren aikana mukaan lukien erilaiset virtuaaliympäristöjen ja lisätyn todellisuuden sovellukset. Tässä oppaassa mallipohjaisuutta käsitellään laajasti. Oppaan osassa II käsitelty on laajemmin mallipohjaisuuden teoriaa.

INCOSEn [2007] mukaan mallipohjaisuus tarkoittaa formaalin mallinnuksen soveltamista järjestelmän (tuotteen) koko elinkaaren aikana ja mallien tärkeyden korostamista [Estefan 2008]. Vitech [2011] on määrittellyt MBSE:n suppeammin projektin ongelma- ja ratkaisuvastuuden esittämisen kuvauskielen avulla. Vitech mainitsee mm. seuraavat: SysML, Activity Diagram, Functional Flow Block Diagram (FFBD). Vitech toteaa myös, että oikean kielen kehittäminen järjestelmien kuvaamiseen jatkuu.

Olennaista mallipohjaisuudessa on sen kytkeminen osaksi yrityksen toiminta-prosesseja. Voidaan ajatella että SE on ylätasoinen prosessi, jonka sisällön hallintaa mallipohjaisuus tukee. Mallipohjaisuus pyrkii osittain korvaamaan dokumenttipohjaista informaation hallintaa esimerkiksi järjestelmän vaatimusten määrittelyssä.

Tuotteet ovat nykyisin usein moniteknisiä sosioteknisiä mekatronisia järjestelmiä, jotka ovat yhdistelmiä mekaniikkaa, automaatiota, elektroniikkaa ja ohjelmistoja. Mukana järjestelmässä on myös aina ihminen ja ympäristö vähintäänkin jossain järjestelmän elinkaaren vaiheessa. Mallipohjaisuus mahdollistaa tällaisten moniteknisten järjestelmien tarkastelun ja simuloinnin, mutta samalla moniteknisyys ja sosioteknillinen näkökulma tuovat suuria haasteita mallien tuottamiseen, hallintaan ja ymmärtämiseen. Eri tekniikan alat ja muut sidosryhmät käyttävät erilaisia käsitteitä, malleja ja mallinnustyökaluja. Niiden yhteensovittaminen on tällä hetkellä keskeinen tutkimus- ja kehityskohde.

Järjestelmän mallinnuksessa on kiinnitettävä huomiota siihen mitkä ominaisuudet ovat keskeisiä mallissa, ja mitä tarkoitusta varten ne palvelevat. Mallilla on aina tietty tarkoitus. Se voi olla esimerkiksi vain tietyn ominaisuuden mittaaminen, mutta mallia voidaan hyödyntää myös mm. verifiointiin, kommunikointiin, koulutukseen ja ohjeistamiseen. Fyysinen prototyyppi pitää yleensä sisällään lähes kaikki samat ominaisuudet kuin lopputuotekin, mutta malliin on määritelty vain tietyt tarkastelun kohteena olevat ominaisuudet kuten esim. toiminta. Mallit helpottavat erityisesti abstraktien järjestelmien (esim. täysin uudet järjestelmät ja konseptit) tutkimista. Mallin ilmaisuväline (medium) voi olla verbaalinen, matemaattinen, symbolinen, graafinen tai visuaalinen. [Hubka & Eder 1984]

Mallin konteksti vaihtelee:

- Abstraktista konkreettiseen
- Aineellisesta käsitteelliseen
- Yleisestä spesifiin

Mallin funktio ja tarkoitus voi olla yksi tai kombinaatio seuraavista:

- Kuvaileva (deskriptiivinen) – mallin tai todellisuuden tietyn aspektin selittämiseen
- Ennustava – tietyn käyttäytymisen ennakoimiseen
- Tutkiva – tietyn käyttäytymisen muuttuminen eri olosuhteissa
- Suunnittelu – uudenlaisen sovelluksen ehdottamiseen
- Määräävä – ohjeistava

Lähtien tarpeista ja tavoitteista, järjestelmän tulisi pystyä kuvaamaan sen ominaisuuksina (properties), vaatimuksina, toimintorakenteena, loogisena rakenteena ja komponentti-rakenteena (osarakenne, tuoterakenne) [Hubka & Eder 1984]. Tässä yhteydessä on huomattava, että vaikka järjestelmää tai tuotetta lähdetään harvoin kehittämään tyhjästä, tarve edellä mainituille abstraktiotasoille on olemassa. Innovaatiot voivat liittyä esimerkiksi tuotteen uudenaikaiseen tuotannon modulointi-periaatteeseen, jolloin toimintorakenteen ja loogisen rakenteen hallinta on välttämätöntä.

Mallipohjaisuuden näkökulmasta tärkeää on kyvykkyys mallintaa ja simuloida edellä mainittuja aspekteja. Staattiset ja deskriptiiviset mallit kuvaavat järjestelmän arkkitehtuuria, rakennetta ja siinä ilmeneviä riippuvuuksia, kun taas dynaamisia malleja käytetään simuloimaan järjestelmän tiloja ja datavirtaa. Deskriptiivisiä (kuvailevia) malleja ovat mm. geneeriset UML- ja SysML-mallit sekä esim. suunnittelualakohtaiset hydraulii-, sähkö-, automaatio- ja prosessikaaviot.

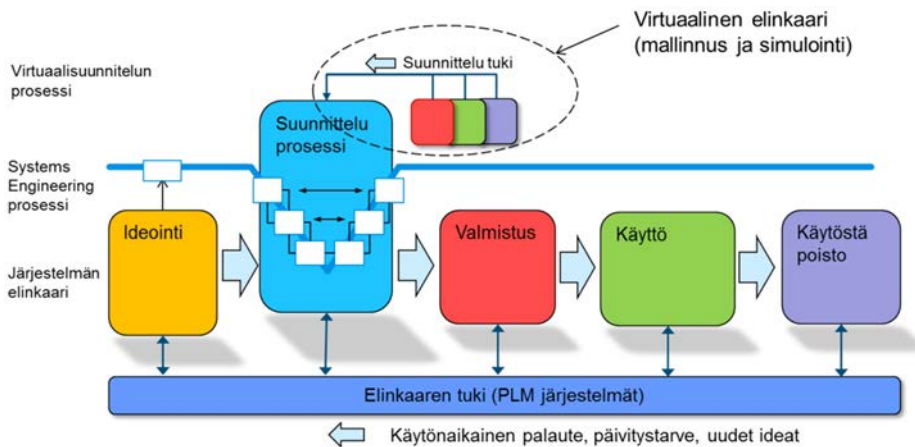
Mallipohjaisuuden tuomia hyötyjä ovat mm. järjestelmän kokonaisvaltaisen ymmärtämisen parantuminen, symbolisen kielen käyttö abstraktien tarpeiden, vaatimusten ja järjestelmien ymmärtämiseen ja järjestelmän rakenteen ja toiminnallisuuden simuloiminen.

2.8 Virtuaaliprototyyppi

Tuoreessa kirjallisuudessa esiintyy käsite "laajennettu mallipohjainen SE" (esim. Eigner et al. [2012]). Tämä tarkoittaa sitä että mallipohjaisuuden käsitettä on laajennettu kattamaan kuvailevien (deskriptiivisten) mallien lisäksi myös virtuaaliprototyyppiin eli monitekniisiin simulointimalleihin (kuten Matlab) sekä suunnittelualakohtaisiin simulointeihin ja visualisointeihin. Virtuaaliprototyyppi on tässä yhteydessä termi, jolla tarkoitetaan joukkoa digitaalista tuotetietoa tuottavia, analyysoivia ja hyödyntäviä työkaluja ja menetelmiä, esimerkiksi erilaisia simulointeja, virtuaaliympäristöjä, CAE-työkaluja jne. Virtuaaliprototyyppi käsittelee tyypillisesti dynaamisia malleja. Tämän luvun tavoitteena on kuvata virtuaaliprototyypin asema osana liiketoiminnan prosesseja sekä sen suhde tekniseen järjestelmään ja tuoterakenteisiin, tuoteprosessiin ja -projektiin, luovaan suunnitteluprosessiin, tuotteen elinkaareen, yms. Virtuaaliprototyypin voidaan hyödyntää mm. abstraktien ja kompleksisten mallien visualisoinnissa ja tulkinnassa, vaatimusten määrittelyssä ja validoinnissa, suunnitelmien arvioinnissa, suunnittelualojen välisessä kommunikoinnissa, yms. Tämän kirjan osassa II on esitelty referenssi-malli, jossa virtuaaliprototyypin osa-alueita ja hyödyntämistä on käsitelty yksityiskohtaisemmin.

Virtuaaliprototyyppi ja tuotekehityksen simulointimenetelmät ovat lyömässä vihdoin läpi laajemmin teollisuudessa. Tähän trendiin ovat syynä mm. kehittyneet työkalut ja laskentakapasiteetti yhdistettynä alentuneisiin hankintakustannuksiin, mallien helpompi siirtäminen työkalusta toiseen sekä onnistuneet pilotit, jotka ovat osoittaneet saavutettuja hyötyjä. Kustannustehokkuuden todentaminen on kuitenkin edelleen hankalaa, koska yritykset tarvitsevat konkreettista näyttöä ennen suuria investointeja ja skaalaamista koskemaan koko yrityksen prosesseja. Tuote-

tiedon hallinnan näkökulmasta virtuaaliprototyypoinnin ja simuloinnin tuottaman, muokkaamaan ja hyödyntämän informaation ja mallien konsolidointi, versiointi, konvertointi ja jakaminen ovat edelleen ongelmia. Tuotetiedosta pitäisi pystyä tarjoamaan näkymiä ja rinnakkaisia tuoterakenteita virtuaaliprototyypoinnin käyttöön. Virtuaalimalleja pitäisi pystyä generoimaan helposti, jopa automaattisesti, elinkaaren eri vaiheiden tarpeisiin sekä muiden liiketoimintojen tukemiseen kuten markkinointi ja myynti, tuotteen toiminnallisuus, kokoonpantavuus ja huollettavuus. Lisäksi virtuaaliprototyyppeihin ja simulointimalleihin pitäisi pystyä sisällyttämään järjestelmän abstrakteja kuvauksia kaikkien suunnittelualojen näkökulmasta. Silloin virtuaalimallit palvelisivat paremmin eri alojen välisenä havainnollisena kommunikointimediana. Teknisten haasteiden lisäksi on huomattava että tähän liittyy paljon inhimillisiä tekijöitä, esimerkiksi erilaiset työkalut, prosessit ja kulttuurit eri suunnittelualoilla.



Kuva 5. Virtuaalisen tuotteen ja reaalisen tuotteen suhde, sekä tuotteen elinkaaren prosessin suhde SE prosessiin. Elinkaaren aikaisia vaiheita voidaan simuloida virtuaalimalleilla.

Perinteisesti virtuaaliprototyypointia on sovellettu usein järjestelmän teknisten ratkaisujen verifiointiin hyvin myöhäisessä vaiheessa. Mallipohjaisessa menetelmässä asia pitäisi olla päinvastainen, eli dokumenttien sijaan malleilla ja simuloinneilla lähdetään määrittelemään tarpeita, tavoitteita, vaatimuksia – jotka sitten konkretisoituvat suunnittelussa (kuva 5). Tämän prosessin kehittäminen ja hallinta on toki haastavaa, mutta monet ovat jo ottaneet askeleita siihen suuntaan. SE tavallaan oikeuttaa toiminnallisten simulointimallien tekemisen jo tuoteprosessin alkupäässä.

Virtuaaliprototyypointiin liittyvistä käsitteistä ja termeistä ei vielä muodostunut yksimielisyyttä. Syynä lienee nuorehko teknologia sekä eri toimialoilla ja yrityksissä muodostunut kulttuuri- ja slangisanasto. Tieteessä virtuaaliprototyypointi ei muodosta omaa koulukuntaa vaan on enemmän tai vähemmän osana esimerkiksi mekaniikkaa, tietojenkäsittelytieteitä tai suunnittelutieteitä. Virtuaaliprototyypointiin liittyen käytetään yleisesti myös esimerkiksi termejä virtuaalisuunnittelu (virtual engineering), virtuaalitekniikka (virtual technology) jne. Määritellyt riippuvat myös siitä mitä teknologioita, työkaluja ja menetelmiä luetaan tähän löyhään kategoriaan kuuluvaksi.

Tässä käytetään ylätasoa käsitteinä virtuaaliprototyypointia (tekeminen) ja virtuaaliprototyyppi (kohde, järjestelmän malli). Virtuaaliprototyypin ilmentymänä voi olla esimerkiksi virtuaaliympäristö (VE, virtual environment) ja käyttöliittymänä virtuaalitodellisuus (VR, virtual reality).

Wang [2002] on määritellyt käsitteet näin: *“Virtuaaliprototyyppi (tai digitaalinen malli) on fyysisen tuotteen/järjestelmän tietokone-simulointi. Sillä voidaan esittää, analysoida ja testata asioita järjestelmän elinkaaren eri vaiheista, kuten suunnittelusta, valmistuksesta, huollosta, kierrätyksestä, kuten todellisella fyysisellä mallilla. Virtuaaliprototyypin rakentamista ja testaamista kutsutaan virtuaaliprototyypinniksi”*. Tähän voisi lisätä että virtuaaliprototyypinnillä voidaan tutkia myös sellaisia asioita, joita fyysisellä järjestelmällä ei pystytä tutkimaan.

Jotta virtuaaliprototyypointi tukee SE- ja MBSE-prosesseja, mallin (eli virtuaaliprototyypin) on kyettävä ilmaisemaan järjestelmän ominaisuudet, rakenteet, toiminnot, muodot. Virtuaaliprototyypinnin tulee siis tukea teknisen prosessin transformaatiota tarve- ja vaatimusmäärittelystä valmiiksi järjestelmäksi. Virtuaaliprototyypinnin osa-alueita ovat mm:

- Virtuaaliympäristöt (VE)
- Virtuaalitodellisuus (VR)
- Lisätty todellisuus (AR)
- Virtuaalisimulaattorit tuotekehityksessä ja koulutuksessa
- Fysiikkasimuloinnit: monikappaledynamiikka, FEA yms.

Edellä mainittuja käytetään eri tarkoituksiin järjestelmän määrittelyn ja prosessien eri vaiheissa esimerkiksi tarpeiden ja vaatimusten määrittelyyn, verifiointiin ja validointiin sekä testaukseen. Perinteisesti virtuaaliprototyypit ja niihin liittyvät simuloinnit ovat jakaantuneet reaaliaikaisiin (virtuaaliympäristöt, simulaattorit, yms.) ja ei-reaaliaikaisiin simulointeihin. Tietokoneiden laskentatehon ja ohjelmistojen kehittyessä tämä jaottelu on kuitenkin hämärtyvässä.

Virtuaaliprototyypinnin yhtenä tavoitteena on perinteisesti nähty tuoteprosessien tehostaminen ja nopeuttaminen mm. vähentämällä fyysisten prototyypien tarvetta ja parantamalla päätöksentekokykyä. Teollisuudessa on otettu käyttöön virtuaali-

prototyyppointia metodiikkana enenevässä määrin nyt, kun teknologia alkaa olla riittävän kypsää. Kansainvälisesti varsinkin auto- ja ilmailuteollisuus ovat olleet pioneereja virtuaaliprototyyppoinnissa, mutta myös kotimaisia esimerkkejä löytyy. On selvää, että virtuaaliprototyyppoinnin ja sen alalajien, kuten virtuaaliympäristöjen ja simulaattorien hyödyt ilmenevät eri tavalla massatuotannossa (kuten autot) tai lentokonevalmistajilla (suuret resurssit) kuin suomalaisella teollisuudella. Juuri tästä syystä on tärkeää pohtia mallipohjaisuutta laajasti kunkin toimialan, liiketoiminnan sekä tuotteiston näkökulmasta. Virtuaaliprototyyppointi nähdään yleisesti metodologiana joka:

- Tehostaa tuoteprosesseja
- Lyhentää tuotekehityksen läpäisyaikaa
- Vähentää fyysisen prototyyppoinnin kustannuksia
- Auttaa rakentamaan laitoksia ja muita yksittäisiä järjestelmiä kerralla oikein eli kustannustehokkaammin
- Mahdollistaa useampien konseptien ja vaihtoehtoisten ratkaisujen vertailun
- Parantaa kommunikointia ja päätöksentekoa
- Mahdollistaa rinnakkaissuunnittelun eri toimintojen ja suunnittelualojen kesken paremmin
- Mahdollistaa tuotteen tai järjestelmän elinkaaren aikaiset tarkastelut suunnitteluvaiheessa
- Auttaa ymmärtämään loppukäyttäjien, asiakkaiden ja muiden sidosryhmien tarpeita, sekä järjestelmän kontekstia paremmin
- Parantaa tuotteen laatua
- Toimii käyttöliittymänä ja ilmaisuvälineenä monimutkaiseen ja abstraktiin tuotetietoon.

Nämä yleiset väitteen hyödyistä ja kustannustehokkuudesta ovat olleet jo pitkään tiedossa, mutta niiden todentamisesta käytännössä on vähän tieteellistä evidenssiä. Tähän on useita syitä. On esimerkiksi erittäin vaikeaa verrata toteutuneita projekteja, koska harvoin on löydettävissä kahta samanlaista projektia joissa virtuaaliprototyyppoinnin vaikutusta toisen projektin onnistumiseen voidaan arvioida. On myös vaikeaa osoittaa yleistettäviä syy-seuraussuhteita aukottomasti. Toisaalta yksittäisiä menestystarinoita voidaan analysoida ja osoittaa tapauksia, joissa virtuaaliprototyyppointi on säästänyt merkittävästi aikaa ja rahaa. Näyttääkin siltä, että virtuaaliprototyyppoinnin hyötyjä kannattaa tarkastella kokonaisuutena, strategisena kyvykkyyden parantajana ja arvona liiketoiminnalle.

2.9 Monimutkaisuuden hallintaan liittyviä standardeja

Kansainväliset standardoimisjärjestöt ja niitä tukevat järjestöt ovat kehittäneet joukon Systems Engineering -standardeja. Standardeja ylläpidetään ja kehitetään jatkuvasti. Viime aikoina on eri lähtökohdista alkunsa saaneita standardeja pyritty harmonisoimaan mm. termistöjensä osalta.

Tällä hetkellä keskeisinä Systems engineering -standardeina voidaan pitää:

- ISO/IEC 15288 [2010] Systems and Software Engineering – Systems Life-Cycle Processes
- IEEE 1220 [2005] Standards for Application and Management of the Systems Engineering Process
- EIA 632 [1999] Processes for Engineering a System.

Näistä ISO/IEC 15288 [2010] on yleispätevin ja kattaa laajimmin koko elinkaaren. Standardissa elinkaari on jaettu neljään pääprosessiin:

- Sopimusprosessi (Agreement process)
- Organisaation tukiprosessi (Organizational Project-Enabling Processes)
- Projektiprosessi (Project process) ja
- Tekninen prosessi (Technical process).

Jokainen pääprosessi koostuu edelleen useista osavaiheista ja prosesseista.

Laajaan ISO 10303 [2012] standardiperheeseen (Automation systems and integration — Product data representation and exchange, epävirallisesti STEP Standard for the Exchange of Product model data) kuuluu osia jotka liittyvät läheisesti elinkaaritiedon esittämistapoihin. Tällaisia ovat esimerkiksi AP 239 Product Life-Cycle Support (PLCS) ja AP 233 Systems Engineering.

2.10 Menetelmiä prosessien kehittämiseen

Fiatech³ on etenkin amerikkalaisen teollisuuden yhteenliittymä, joka pyrkii edistämään suunnittelu- ja projektitietojen hallintaa ja mallipohjaista suunnittelua sekä tuotetiedon integraation edistämistä. Fiatech on tunnistanut seuraavia esteitä tämänkaltaisen lähestymistavan käyttöönotolle:

- *Hyvien esimerkkitapausten näkyvyyden puute:* Yritykset haluaisivat nähdä, miten muut yritykset ovat soveltaneet toimintatapaa ja mitä tuloksia on saavutettu. Halutaan varmistaa, että investoinnilla kehittämiseen saataisiin tulosta aikaan. Aihepiirissä pidemmällä olevat yritykset taas eivät ole välttämättä valmiita jakamaan osaamistaan mahdollisille kilpailijoille.

³ www.fiatech.org

- *Ristiriita projektin ja elinkaaren tavoitteiden välillä:* Koko elinkaaren huomiointo voi lisätä kustannuksia projektivaiheessa (suunnittelu ja valmistus), ja projektivaiheessa pidetään tärkeänä pysyä tavoitekustannusten sisällä. Jos asiaa ei pystytä hoitamaan, koko tuotteen elinkaaren tavoitteita ei todennäköisesti saavuteta.
- *Voimakkaat ja itsenäiset projektipäälliköt:* Projektipäälliköitä mitataan tyypillisesti yksittäisen projektin tulosten perusteella (aikataulu ja budjetti). He ovat usein haluttomia omaksuma uusia lähestymistapoja ja näkevät uudet lähestymistavat lisääntyneenä projektiiriskinä. Avoimuus innovaatioille on harvinaista.
- *Vaikeus muuttaa toimintatapoja ja työkäytäntöjä:* Yritykset eivät halua muuttaa systeemeitä, jotka "toimivat". Kun teknologia kehittyy, yritykset käyttävät uuteen teknologiaan siirtymistä vasta ponttimena prosessien parantamiseen.
- *Riskien jakaminen ja palkitseminen koko tuotteen elinkaaren huomioimisesta:* Jos suunnittelutoimittajat pakotetaan käytäntöön, joka ei ole heille ennestään tuttu, ne pyrkivät laskuttamaan korkeamman hinnan kattaakseen "tuntemattoman" työn osuuden. Joissakin tapauksissa taas suunnittelutoimisto voi olla kykenevä tuottamaan hyödyllistä sisältöä, joka säästäisi asiakkaan rahoja, mutta kannustimet puuttuvat.

SE:stä, mallipohjaisuudesta ja virtuaaliprototypoinnista, kuten myös tuotekehityksestä ja muista vastaavista prosesseista, on olemassa useita yleisiä referenssimalleja ja standardeja. Ne pitää kuitenkin räätälöidä ja sovittaa kunkin yrityksen ja arvoverkoston luonteeseen. Näitä luonteenpiirteitä ovat esimerkiksi liiketoimintamallit, tuotteen tyyppi ja sen monimutkaisuuden aste ja tuotantotapa. Prosessien räätälöintiin ja kehittämiseen on olemassa menetelmiä:

- Kypsyysmallit
- Vaikuttavuusanalyysit
- Simulointipelit
- Prosessimallinnus (as-is ja to-be)
- Roadmapping
- Benchmarking.

Seuraavaksi käsitellään tarkemmin kypsyysmallien käyttöä ja vaikuttavuusanalyysia. Kypsyysmallit (maturity model) ovat hyviä lähtökohtia nykytilanteen arvioinnissa ja kehityskohtien tunnistamisessa. Kirjallisuudesta löytyy monia eri tarkoituksiin kehitettyjä malleja, jotka painottavat eri tavoin esimerkiksi teknologiaa, prosesseja ja organisaatiota. PLM:n implementointi on tyypillinen kohde, jossa hyödynnetään kypsyysmalleja. Mallipohjaisuuden ja virtuaaliprototypoinnin kypsyyttä voidaan arvioida mm. seuraavista näkökulmista [Aromaa et al. 2013]:

- Vaikutus liiketoimintaan
- Tuoteprosessit ja elinkaari

- Virtuaaliprototyypoinnin ja mallipohjaisuuden prosessit
- Virtuaaliprototyypoinnin ja mallipohjaisuuden teknologiat
- Henkilöresurssit
- Yritysinfrastruktuuri
- Yrityskulttuuri ja organisaatio.

Vaikuttavuusanalyysillä voidaan tarkemmin tutkia mitä positiivisia (hyötyjä) ja negatiivisia (esim. kustannuksia) vaikutuksia edellä mainittujen näkökulmien kehittämisellä on. Vaikuttavuutta voidaan analysoida esimerkiksi arvoketjun, verkoston, yrityksen toimintojen, disipliinien tai tuotteen elinkaaren näkökulmasta. Esimerkki vaikuttavuusanalyysistä on VTT:n kehittämä PIA – PLM Impact Analysis [Leino et al. 2012]. Vaikuttavuusanalyysissä ja yleensä prosessien, menetelmien, yms. kehittämisessä voidaan hyödyntää simulointipelejä ja prosessimallinnusta. Tyypillisesti niihin osallistuu sidosryhmiä, jotka tuovat omat näkemyksensä kehittämiseen. Näitä menetelmiä on hyödynnetty mm. virtuaaliprototyypoinnin prosessien ja niihin liittyvien tiedonhallinnan tekniikoiden ja tietomallien kehittämiseen.

2.11 PLM:n implementoinnista ja kaupallisista ratkaisuista

PLM:n implementoinnin ja kehittämisen tulee tapahtua liiketoimintalähtöisesti tarpeista, ei IT-lähtöisesti. On myös otettava huomioon, että yksi PLM-konsepti ei välttämättä sovellu suuren yrityksen tai konsernin kaikkiin liiketoimintoihin. Liiketoimintamielessä on huomioitava esimerkiksi minkälaisia yrityksen tuotteet ovat, millainen on tuotantotapa (yksittäisiä järjestelmiä, konfiguroituvia, sarjatuotteita) sekä minkälainen on liiketoimintaympäristö. Ei voi myöskään liikaa korostaa sitä, että PLM-implementoinnin tulisi lähteä liikkeelle yhteisestä PLM-ymmärryksen rakentamisesta organisaatioissa, minkä jälkeen kerätään ja käsitellään eri sidosryhmien vaatimukset. Näiden pohjalta rakennetaan konsepti tarvittavista prosesseista ja tietomallista, sekä askeleittain etenevä roadmap PLM-implementointiin. Vasta sen jälkeen lähdetään valitsemaan IT-järjestelmää, toimittajaa ja mahdollisia konsultteja. Haasteena organisaatioissa on usein siiloutuminen, eli yrityksen eri funktiot ja osastot toimivat omien tavoitteidensa, kannustimiensa, prosessiensa ja kulttuuriensa mukaan. Lisäksi käytännön ruohonjuuritason työ tehdään usein eri tavalla kuin prosesseissa on määritelty. Hyvä johtaminen ja organisaation muutoshallinta ovat keskeisiä tekijöitä PLM:n implementoinnin onnistumisessa. Johtamiseen liittyviä asioita ovat mm. motivointi, kouluttaminen, kommunikointi, palkitseminen. PLM:n implementointiin käytännöllisenä lähestymistapana voi suosittelaa lean-periaatteita, jotka pitävät sisällään konkreettisia ja mitattavia toimenpiteitä.

Systems Engineering -standardit eivät määrittele mitä prosesseja PLM:n pitäisi tukea. PLM-ratkaisusta pitäisi kuitenkin löytyä kyvykkyksiä tukemaan ja realisoimaan tyypillisiä SE-prosesseja: Vaatimusten käsittely ja relaatiot sekä jalostaminen ja hallinta, järjestelmän tekninen erittely (breakdown) ja integrointi, monitekniisen suunnittelutiedon hallinta, ratkaisuvaihtoehtojen todentaminen (verification) ja kokonaisuuden kelpuuttaminen (validation), testaaminen, valmistus, elinkaari, V-mallin tukeminen, tietämyksen hallinta, suunnittelupäätösten hallinta, jäljitettävyys (vaa-

timuksiin ja ratkaisuihin) ja analysointi, projektin hallinta, liiketoimintaprosesseihin kytkytyminen jne. PLM:n implementointi ei missään tapauksessa saa tehdä suunnittelusta ja muista prosesseista hitaampia ja kömpelömpiä, vaan päinvastoin sen tulee nopeuttaa niitä ja tehdä ketterämmäksi systemaattisen transformaation asiakasvaatimuksista ja -palautteesta, teknisten vaatimusten ja ominaisuuksien kautta asiakkaalle lisäarvoa tuottaviksi tuotteen ominaisuuksiksi. Tämä on mahdollista silloin kun vaatimukset ja roadmap PLM:n implementointiin tehdään huolella, mutta luonnollisesti se vaatii alussa kohtuullisen suuria investointeja.

Markkinoilla olevat PLM-ratkaisut tukevat SE-prosesseja ja tietomalleja vaihtelevasti. Ne ovat historiallisesti kehittyneet eri lähtökohdista ja usein kasvu on tapahtunut ostamalla markkinoilta tiettyyn tarkoitukseen rakennettuja ohjelmistoja. Kaupalliset PLM-järjestelmät voidaan myös karkeasti jakaa suunnittelulähtöisiin (lähellä CADiä), tuotantolähtöisiin (lähellä ERPiä) ja verkoston yhteistoimintaan keskittyviin. Näistä lähtökohdista kaupallisia PLM-ratkaisuja on hankalaa verrata suoraan keskenään. Lähtökohtana tulisikin olla selkeä ymmärrys oman yrityksen tarpeista ja prosesseista, työkaluista ja menetelmistä, joita halutaan PLM:llä tukea.

PLM:n IT-arkkitehtuurin näkökulmasta vaihtoehtoja on valittavana nykyään monenlaisia: modulaarinen vs. monoliittinen, pilvipalvelu, open-source jne. Kaupallisissa ratkaisuihin on enemmän tai vähemmän suljettuja rajapintoja. Huomioon pitää yleensä ottaa myös yrityksen itse tehdyt ja perinteiset (legacy) sovellukset. Riskienhallintamielessä kannattaa pohtia, mennäänkö vain yhden toimittajan ratkaisuun. Toisaalta tarpeet ovat usein sellaisia, ettei yhdellä toimittajalla ole tarjota ratkaisuja kaikkiin tarpeisiin.

Harkittavaksi tulee myös minkä tyyppiseen tietomalliin sitoutuu, onko se jonkin standardin mukainen vai toimittajakohtainen. Standarditietomallien etuina voidaan nähdä tietynlainen neutraalius ja pitkä informaation elinkaari. Toimittajakohtaisissa tietomalleissa etuna voi olla mm. informaation parempi integrointi sovellusten ja työkalujen välillä. Saattaa olla myös haasteellista löytää ja määritellä tietomalli, joka toimii koko tuotteen elinkaaren läpi ja huomioi eri suunnittelualueet ja organisaatioyksiköt yli disiplina- ja organisaatio-rajoiden, sekä eri järjestelmien välillä. Usein tuotetiedon näkökulmasta PLM-implementoinnin työläs haaste on eri lähteistä tulevan informaation harmonisointi ja konsolidointi.

2.12 Verkoston valmiuksien kehittäminen

Verkostotoiminnan valmiuksilla tarkoitetaan yleisesti sitä, miten paljon resursseja tai toimenpiteitä tehdään ennen kuin varsinainen tehtävä suoritetaan. Jos prosessit on mietitty etukäteen ja toimintamalleista sovittu, voidaan tarjouspyynnön tai asiakastilauksen tullessa nopeasti edetä itse tehtävän suorittamiseen ja arvon tuottamiseen. Pitkälle menevien valmiuksien kehittäminen ei ole kuitenkaan itse-tarkoitus eikä aina kannattavaa; sopiva taso riippuu yhteistyön tavoitteista, kestosta, toistuvuudesta ja luonteesta.

SE:n soveltaminen esim. projektissa voidaan nähdä potentiaalisena riskitekijänä, joka aiheuttaa lisätyötä ja viivästyksiä. Näin varmaan onkin, jos SE:stä ei ole

mitään tietoa ja kokemusta ennen projektia. SE-lähestymistavan soveltaminen vaatii siis omat valmiutensa. Asiakasprojektissa ei ole aikaa enää opiskella SE-prosesseja; siihen pitäisi valmistautua etukäteen. Se ei tarkoita, että jokaisen projektiosallistujan tulee olla yhtä syvällä SE:n hallinnassa. On kuitenkin mietittävä etukäteen, miten SE-käytännöt viedään läpi, miten paljon SE-osaamista pitää kehittää ja jakaa, millaisella tiedonvaihdolla ja työkaluilla sitä voidaan tukea ja miten ne saadaan käyttöön. Jos on tarve soveltaa SE-menettelyä vain rajatulle alueelle eikä kaikissa projekteissa, kannattaa valita erikseen se joukko, jonka kanssa näitä vaativia hankkeita tehdään.

SE-valmiuksiin sisältyy riittävän osaamisen kehittäminen ja kouluttaminen sekä seuraavien asioiden määrittely:

- Selkiytetään, mitä asiakas vaatii SE-prosesseilta
- Miten SE viedään läpi omassa yrityksessä; prosessit, roolit ja johtaminen
- SE-prosessit yli yritysrajojen, miten SE-käytännöt hoidetaan yhteistyökumppaneiden ja alihankkijoiden kanssa
- Käytännöt ja työkalut sekä niiden osaaminen, esim. vaatimusten hallinnan työkalut
- Yhteiset käsitteet ja termit
- Verkoston yhteinen visio
- Verkoston ydinosamisten tunnistaminen ja kehittäminen
- Yhteiset prosessit, säännöt ja toimintaperiaatteet, roolit, verkoston johtaminen ja hallinta
- Yhteisen tiedon hallinta, kommunikaatio ja tiedonvaihto + työkalut, järjestelemät
- Sopimusasiat
- Verkostopartnerien valinta, kvalifointi ja suorituksen seuranta
- Luottamuksen rakentaminen ja ylläpito.

3. Johtopäätökset

Tähän oppaaseen tiivistettiin havainnot ja tulokset tutkimushankkeesta, jonka tavoitteena oli löytää parhaita käytäntöjä teollisuuden tuotesuunnittelun ja elinkaari-prosessien integroituun hallintaan. Erityisesti tarkasteltiin Systems Engineering-lähestymistavan ja PLM:n hyödyntämisen mahdollisuuksia. Aihetta analysoitaessa pohdittiin mitä lisäarvoa SE ja PLM tuovat. Varsinkin SE:n ajatellaan usein soveltuvan vain monimutkaisten teknisten systeemien suunnitteluun, jota lähdetään toteuttamaan tyhjältä pöydältä. Nykyisin kuitenkin lähes kaikki teolliset tuotteet ja järjestelmät ovat monimutkaisia, kun otetaan huomioon koko elinkaari ja kaikki siihen liittyvät toimijat. Juuri tämän haasteen hallinnassa SE:n ja PLM:n vahvuudet yhdessä tulevat esiin. Ne pitää nähdä tavallaan kattauksena erilaisia prosesseja, toimintatapoja ja työkaluja, joista kunkin yrityksen tulee rakentaa omaan liiketoimintaan sopiva konfiguraatio. Konfiguraatio luo vaatimukset niin tietoteknisten ratkaisujen, kuin organisaation ja prosessien jatkuvaan kehittämiseen. Jatkossa tarvitaan vielä tutkimusta sekä tiedonhallinnan teknologiaan (esimerkiksi tietomallit, formaatit, standardit, simuloinnit) OASIS että toimintatapoihin liittyen. Esimerkiksi mallipohjaisuuden laaja implementaatio vaatii organisaatio- ja verkostotason tutkimusta.

Lähdeluettelo (Osa I)

- Amaral, L.A.N. & Uzzi, B. 2007. Complex Systems – A New Paradigm for the Integrative Study of Management, Physical, and Technological Systems. Management Science, Vol. 53, No. 7.
- Aracic, J. 2012. Deploying Systems Engineering – An introduction. Crescendo Technologies. <http://blogs.crescendo-technologies.com/en/deploying-systems-engineering-an-introduction/>. [Accessed 25.3.2013].
- Aromaa, S., Leino, S-P. & Viitaniemi, J. 2013. Are companies ready for the revolution in design – modelling maturity for virtual prototyping. International Conference on Engineering Design. Seoul, 19–22. 8. 2013.
- EIA 632. 1999. Processes for Engineering a System.
- Eigner, M., Gilz, T. & Zafirov, R. 2012. Proposal for functional product description as part of a PLM solution in interdisciplinary product development. DESIGN2012.
- Estefan, J.A. 2008. Survey of Model-Based Systems Engineering (MBSE) Methodologies IN-COSE MBSE Initiative.
- Hubka, V. & Eder, V.E. 1984. Theory of Technical Systems – A Total Concept Theory for Engineering Design. Springer-Verlag. ISBN 3-540-17451-6.
- IEEE 1220. 2005. Standards for Application and Management of the Systems Engineering Process.
- INCOSE. 2007. Systems Engineering Vision 2020. International Council on Systems Engineering (INCOSE). http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf.
- ISO 10303. 2012. Automation systems and integration — Product data representation and exchange.
- ISO/IEC 15288. 2010. System Engineering Supported by PLM. White Paper. InsidePLM GmbH.
- Leino, S.-P., Anttila, J.-P., Heikkilä, J., Aaltonen, J. & Helin, K. 2012. PLM Impact Analysis Model. PIA. 9th IFIP WG 5.1 International Conference on Product Lifecycle Management, PLM 2012, 9–11 July 2012, Montreal, QC. IFIP Advances in Information and Communication Technology. Springer. Vol. 388 AICT, s. 501–511.

- Munkvold, B.E. 1999. Challenges in IT implementation for supporting collaboration in distributed organizations. *European Journal of Information Systems* 8, s. 260–272.
- Sääksvuori, A. 2011. *PLM vision 2016 and beyond*. Sirkus Publishing. ISBN 978-952-67529-2-1.
- Ulrich, K.T. & Eppinger, S.D. 2003. *Product Design and Development*. McGrawHill.
- U.S. Department of Transportation. 2007. *Systems Engineering for Intelligent Transportation Systems – An introduction for Transportation Professionals*. <http://ops.fhwa.dot.gov/publications/seitsguide/index.htm>.
- Vitech. 2011. *A primer for model-based system engineering*.
- Wang, G.G. 2002. Definition and Review of Virtual Prototyping. *Information Science in Engineering*, Vol. 2, No. 3.

Muuta kirjallisuutta

- ESA. Virtual spacecraft design. The “Virtual Spacecraft Design”. <http://www.vsd-project.org/>.
- Long, D. & Scott, Z. 2011. *A Primer for Model-Based Systems Engineering*. Vitech Corporation, Blacksburg. ISBN 978-1-105-58810-5.
- OASIS. Advancing open standards for the information society. <https://www.oasis-open.org/>.
- Shamieh, C. 2011. *Systems Engineering for Dummies*. IBM limited edition. Wiley Publishing Inc. ISBN 978-1-118-10001-1.

OSA II – Systems Engineering käytännössä

Tässä osiossa syvennytään muutamiin aihealueisiin ja niihin liittyviin käytännön haasteisiin ja mahdollisiin ratkaisuihin. Luvut on toteutettu erillisinä, itsenäisinä lukuina.

Iris Karvonen

SE-PLM soveltamisen haasteita..... 50

Simo-Pekka Leino, Juhani Viitaniemi, Tapio Salonen

Mallipohjaisuus ja virtuaaliprototypointi..... 54

Mikko Uoti, Iris Karvonen, Jarmo Alanen

Systems Engineering -toimintatavan implementointi verkostoympäristössä..... 85

Jarmo Alanen

Vaatimusten luokittelu..... 113

SE-PLM-soveltamisen haasteita

Iris Karvonen, VTT

Uuden tekniikan tai uusien menetelmien käyttöönotto organisaatiossa tuo mukanaan monia haasteita. Mitä monimutkaisemmasta järjestelmästä on kyse ja mitä useampia organisaation osia ja toimijoita se koskee, sitä suuremmat haasteet todennäköisesti ovat. Tyypillisesti esim. PLM-järjestelmiä pidetään monimutkaisina ja niiden implementointi vaatii usein valtavan projektin ja pitkän kalenteriajan. Systems Engineering -prosessien implementoinnin haastavuus riippuu siitä, millä vaativuustasolla ja roolituksella ne toteutetaan; miten moniin osapuoliin ja toimijoihin se vaikuttaa. Vaikka SE:n implementointi ei vaadi varsinaista PLM-järjestelmää, on edullista tukea SE-prosesseja, esim. vaatimusten hallintaa, sopivilla tietotekniikkatyökaluilla. Jos organisaatiolla ei ole jo käytössä sopivaa PLM-järjestelmää, voidaan käyttää erityisiä vaatimustenhallinta-työkaluja. Siten SE:n toteuttamisessa usein on mukana myös tietotekniikan jonkinasteista käyttöönottoa. Kokonaisen PLM-järjestelmän toteuttaminen yhtäaikaaisesti Systems Engineering -toimintatavan läpiviennin kanssa voi olla riski.

Kirjallisuudessa esitetään jopa 50–70 %:n todennäköisyyksiä tietotekniikan käyttöönottohankkeiden epäonnistumiselle; onpa ERP:n osalta esitetty jopa 90 %:n todennäköisyyttä sille, että ERP-hanke epäonnistuu ensimmäisellä yrityksellä. Epäonnistuminen voi tarkoittaa sitä, että kaikkia suunniteltuja toimintoja ei toteuteta, järjestelmä otetaan käyttöön vain osittain, projektin aikataulu viivästyy, kustannukset ylittyvät tai/ja odotetut hyödyt eivät toteudu. On myös raportoitu tapauksista, joissa ongelmat ovat johtaneet implementoinnista luopumiseen.

Tietotekniikan kehitys on ajan kuluessa siirtynyt konfiguroitavien järjestelmien suuntaan, eikä nykyisin tarvitse räätälöidä tai kehittää alusta uutta ohjelmistoa. Työkaluja käytetään enenevästi selaimen kautta, joten aina ei tarvita edes työasemakohtaisia asennuksia ja päivityksiä. Varsinainen tekninen käyttöönotto on siten yleensä helpottunut. Kehityskulun tavoitteena on, että tarvittavat palvelut ja jopa data olisivat helposti saatavissa ”pilvestä” ja että käyttäjät itse voisivat enenevässä määrin muokata järjestelmän kunkin yrityksen tarpeisiin sopivaksi.

Nykytilanteessa IT-ratkaisuja eri tarkoituksiin on paljon tarjolla ja voi olla vaikeus valita niistä mahdollisimman sopiva. Useat järjestelmät ovat monimutkaistuneet, kun

niitä voidaan konfiguroida käyttöön monella tavalla. Tällöin keskeiseksi ongelmaksi usein nousee, *miten tietotekniikkaa pitäisi kussakin tapauksessa soveltaa*. On syytä pohtia sitä, mitä tavoitteita sen soveltamisella halutaan saavuttaa. Käyttöön-otto muuttaa organisaation toimintaprosesseja, vastuut, roolit ja päätöksenteko voivat muuttua, organisaatioiden välinen vuorovaikutus ja riippuvuudet lisääntyvät. Keskeinen haaste on saada prosessi jalkautettua ja organisaatio ymmärtämään se. Muutoksen aikaansaaminen vaatii, että organisaation toimijat ymmärtävät tavoitteet ja uudet toimintatavat ja sitoutuvat niihin. Tietotekniikan tai SE:n käyttöönotto aiheuttaa usein lisätyötä tietyissä suunnitteluvaiheissa ja hyödyt eivät välttämättä näy niille, joiden pitää tehdä tämä lisätyö. Jos SE:n soveltamisella lähinnä vastaan ulkoisiin vaatimuksiin, eikä mietitä sen soveltamista omista lähtökohdista, voivat hyödyt omalle organisaatiolle jäädä näkymättömiksi.

Samoin kuin IT:n soveltamisessa, ei SE:nkään käyttöönotto aina johda suora- viivaisesti odotettuihin tuloksiin. SE:n soveltaminen käynnistyvässä toteutusprojek- tissa voidaan nähdä jopa riskinä. Ongelmia tulee todennäköisesti, jos menettely ollaan ottamassa käyttöön asiakasprojektissa äkkiseltään, ilman aiempaa koke- musta ja SE-käytäntöjen "sisäänajoa". Pelkkä "SE-teoria", SE:n mekaaninen so- veltaminen, ei riitä, vaan pitää olla sen soveltamiseen ko. toimialalla liittyvää ym- märrystä. SE:n soveltamishaasteina on mainittu mm. seuraavia:

- Vaikka SE esitetään skaalautuvana menettelynä, on oikean soveltamistason löytäminen käytäntöön viennille haaste. Esim. jos vaatimuksia tarkastellaan liian yksityiskohtaisella tasolla, hukutaan yksityiskohtiin ja kokonaisuus ei hahmotu. Toisaalta liian suurina "nippuina" käsiteltävien vaatimusten jäljit- täminen, validointi ja muutosprosessit voivat käydä vaikeiksi. Systeemi- arkkitehtuurin osalta "system breakdown" – miten vaatimukset (product), tuotteen funktiot, osat, geometria (product decomposition), projektin "work breakdown", projektin organisaatio (project decomposition) – kannattaa pa- lastella, ei löydy mitään yleisiä ohjeita.
- SE:n soveltamisen erilaiset lähtökohdat uuden kehittämisessä, vanhan modifioinnissa ja valmiin käyttämisessä vaativat erilaisia prosesseja. Haas- teena ovat myös olemassa olevat tuotteet, joilla on pitkä historia, mutta ei esim. olemassa olevia vaatimuksia.
- Tuotteiden ja toimintaympäristön monimutkaistuminen vaikeuttaa myös suunnitteluprosesseja. Puhutaan mm. multi-X (multisite, multiproject, multi- partner) -suunnittelusta, jossa vaatimukset tulevat monesta lähteestä: lop- pukäyttäjiltä, globaaleilta asiakkailta, eri markkinasegmenteiltä, standar- deista, viranomaismääräyksistä, olemassa olevista tuotteista, toimituksesta, asennuksesta, käyttöönotosta, teknologiatutkimuksesta. Tähän multi- joukkoon voidaan vielä lisätä moniteknicisyys, joka on edelleen laajentunut mm. tuotteisiin liittyvien ohjelmistojen kriittisyyden kasvaessa. Tuotteet eivät ole vain järjestelmiä vaan "systems of systems".
- Verkostoituneessa toiminnassa, itsenäisten yritysten yhteistyössä päätök- senteko hajautuu ja suunnittelun seuranta ja hallinta on haastavampaa.

Kukin yritys päättää esim. itse omien resurssiensa käytöstä. Aikataulujen lyhentäminen aiheuttaa paineita rinnakkaissuunnittelulle. Erilaiset käsitteet, prosessit, toimintatavat ja kulttuurit voivat aiheuttaa väärinymmärryksiä, virheitä ja prosessien hidastumista. Tiedonvaihto ja -hallinta eivät ole tyypillisesti yhtä avointa kuin yrityksen sisällä.

- Hajautetussa ympäristössä projektit menettävät helposti kuvan systeemitason tavoitteista, systeemiarkkitehtuurista ja määräajoista. Saatetaan tehdä lokaaleja, yksipuolisia päätöksiä, jotka vaikuttavat systeemitasolle resursseihin ja aikatauluihin, joskus välittämättä aiemmista sitoutmisista. Tällöin tarvitaan näkyvyyttä projektin kuluessa eläviin sitoutmisiin sekä organisaation sisällä ja ulkoisten partnereiden kanssa.
- Vaikka työkalut ovat kehittyneet, niiden suuri määrä ja yhteentoimimattomuus tuottavat haasteita. Joudutaan käyttämään esim. ”välimuotoja” tiedon siirrossa. Vaikka uusissa versioissa alkaa kehittyä esim. vaatimusten hallintaa tukevia osioita, yritykset eivät ole tietoisia näiden olemassaolosta ja hyödyllisyydestä.

Fiotech (www.fiotech.org) on etenkin amerikkalaisen teollisuuden yhteenliittymä, joka pyrkii edistämään ns. automaattista suunnittelua projektitoimintaympäristössä. Automaattinen suunnittelu perustuu vahvasti suunnittelu- ja projektitietojen hallintaan ja mallipohjaisen suunnittelun ja projektitoiminnan, sekä tietojen integraation edistämiseen. Vaikka Fiotechissä ei puhuta SE-käytännöistä, lähestymistavoilla on yhteisiä piirteitä. Mm. seuraavat Fiotechin tunnistamat esteet lähestymistavan käyttöönotolle ovat relevantteja myös SE:n kannalta:

- Hyvien esimerkitapausten näkyvyyden puute – Yritykset haluaisivat nähdä, miten muut yritykset ovat soveltaneet toimintatapaa ja mitä tuloksia on saavutettu. Halutaan varmistaa, että investoinnilla kehittämiseen saataisiin tulosta aikaan. Aihepiirissä pidemmällä olevat yritykset taas eivät ole niin valmiita jakamaan osaamistaan mahdollisille kilpailijoille.
- Ristiriita projektin ja elinkaaren tavoitteiden välillä – Koko elinkaaren huomioonotto voi lisätä kustannuksia projektivaiheessa (suunnittelu ja valmistus), ja projektivaiheessa pidetään tärkeänä pysyä tavoitekustannusten sisällä. Jos asiaa ei pystytä hoitamaan, koko tuotteen elinkaaren tavoitteita ei todennäköisesti saavuteta.
- Voimakkaat ja itsenäiset projektipäälliköt – Projektipäälliköitä mitataan tyypillisesti yksittäisen projektin tulosten perusteella (aikataulu ja budjetti). He ovat usein haluttomia omaksumaan uusia lähestymistapoja ja näkevät uudet lähestymistavat lisääntyneenä projektiriskinä. Avoimuus innovaatioille on harvinaista.
- Vaikeus muuttaa toimintatapoja ja työkäytäntöjä – Yritykset eivät halua muuttaa systeemeitä, jotka ”toimivat”. Kun teknologia kehittyy, yritykset käyttävät uuteen teknologiaan siirtymistä pontimena prosessien parantamiseen. Työprosessit ja teknologia pitäisi sovittaa toisiinsa.

- Riskien jakaminen ja palkitseminen elinkaarilähestymistavasta. – Jos suunnittelutoimittajat pakotetaan käytäntöön, joka ei ole heille ennestään tuttu, he pyrkivät laskuttamaan korkeamman hinnan kattaakseen ”tunte mattoman” työn osuuden. Joissakin tapauksissa taas suunnittelutoimisto voi olla kykenevä tuottamaan hyödyllistä sisältöä, joka säästäisi asiakkaan rahoja, mutta kannustimet puuttuvat.

Alla olevassa taulukossa on tarkasteltu, mitä eri osapuolien tulee ottaa huomioon SE-toimintatavan soveltamisessa.

Taulukko 1. SE-toimintatavan vaatimukset eri osapuolille.

Osapuoli	Vaatimukset/ huomioitavaa
Asiakkaat, loppukäyttäjät	Projekti/suunnitteluvaihe voi vaatia enemmän resursseja, voi nostaa hintaa suunnitteluvaiheessa. Toteutusta ei pidä arvioida vain hankintahinnan vaan myös elinkaarivarvon ja -kustannusten suhteen. Asiakkaan oma toiminta esim. vaatimusten määrittelyssä vaikuttaa projektin onnistumiseen. Löydettävä sopiva vaatimustaso SE:n soveltamiseen -> asiakkaan ymmärrys prosesseista auttaa. Toimittajan SE-osaaminen selvitettävä ennen projektia.
Tuotteen toimittaja	SE-toimintaprosessien ja roolien määrittely, sekä prosesseja tukevat työkalut/ohjelmistot Vaatii ymmärrystä paitsi SE-soveltamisesta myös sovellusalueesta ja monitekniisten järjestelmien suunnittelusta. Vanhojen tuotteiden suunnittelutiedon hallintaan löydettävä menettelytavat (vaatimukset puuttuu).
Projekti-päällikkö ja pääsuunnittelija	Vaatii ymmärrystä paitsi SE-soveltamisesta myös sovellusalueesta ja monitekniisten järjestelmien suunnittelusta. Vaatimusten allokointi osajärjestelmille/ komponenteille. Tiedonhallintajärjestelmien tulee tukea SE-prosessia. Mallipohjainen suunnittelu käytössä.
Suunnittelijat	Mallipohjainen suunnittelu käytössä. Projektipäällikkö/ pääsuunnittelija hoitaa eri suunnittelualueiden integrointia.
Suunnittelu-partnerit/ Alihankintasuunnittelijat	Määriteltävä, mitä alihankkijoiden tulee ymmärtää SE:stä Vaatiiko alihankkijoilta uusien työkalujen hallintaa ja käyttöä? Pitääkö alihankkijan prosessit validoida?
Valmistus	Jäljitettävyyden hoidettava myös valmistusvaiheessa. Muutoksissa noudatettava määriteltyä prosessia ja dokumentaatiota.
Materiaali/osatoimittajat	Muutoksissa noudatettava määriteltyä prosessia ja dokumentaatiota.
Käyttö, huolto ja ylläpito, modernisointi, päivitys	Jos muutoksia ei dokumentoida, tuotetieto ”vanhenee” eikä vastaa enää todellisuutta. Minkä tason muutokset tulee dokumentoida? Minkä osapuolien pitää tietää muutoksista?

Mallipohjaisuus ja virtuaaliprototyyppi

Simo-Pekka Leino, VTT

Juhani Viitaniemi, VTT

Tapio Salonen, VTT

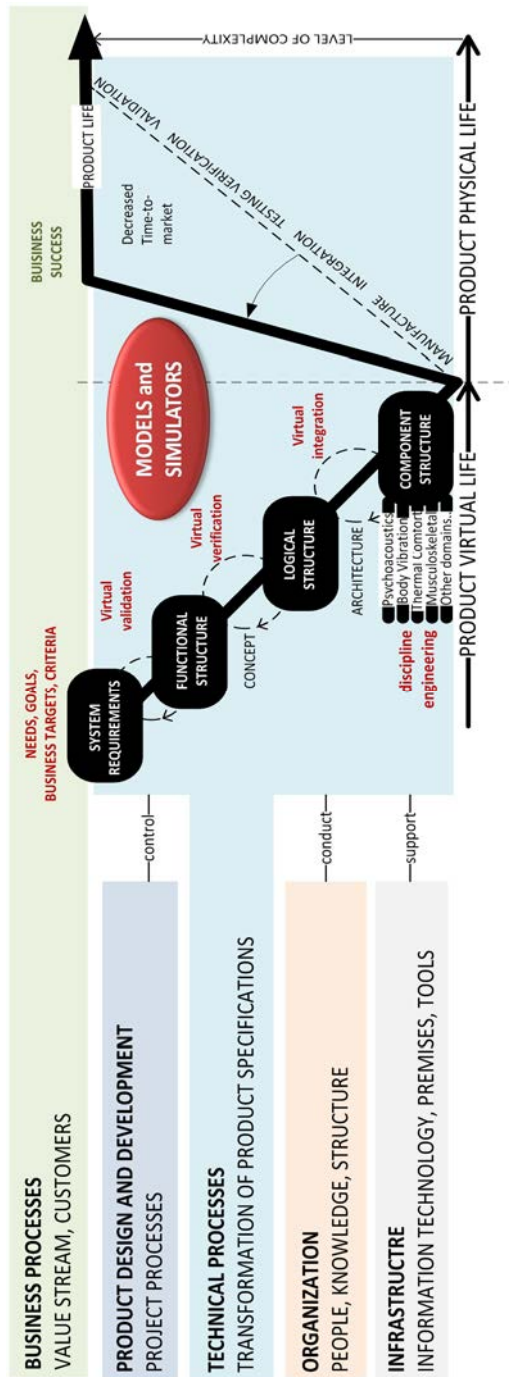
Sisältö

Johdanto.....	55
Mitä on MBSE eli mallipohjainen SE?	57
Mitä ovat mallit?	57
Mallinnusta ohjaavia tekijöitä	58
Teknisten järjestelmien ja prosessien mallit.....	59
Järjestelmien ja mallien abstraktio-tasot	61
Järjestelmien moniteknisyys ja luokittelut	63
Systems Engineeringin tekniset prosessit	64
Model-Based Systems Engineering liiketoiminnan näkökulmasta.....	68
Laajennettu Model-Based Systems Engineering ja virtuaaliprototyyppi ...	70
Virtuaaliprototyypin määritelmää	71
Prosessit, menetelmät ja työkalut.....	72
Virtuaaliprototyypin hyödyt ja niiden todentaminen.....	73
Virtuaaliprototyypin referenssimalli	74
Mallipohjaisuuden suhde muihin prosesseihin	76
Mallinnuksen ja simuloimisen näkökulmat.....	79
Mallinnuskielet UML ja SysML.....	81
Unified Modeling Language (UML).....	81
Systems Modeling Language (SysML).....	82
Menetelmiä prosessien, menetelmien ja tietomallien kehittämiseen	84

Johdanto

Mallipohjaisuus on yleistymässä teollisuudessa. Tämä ilmiö näyttää liittyvän osaltaan Systems Engineeringin (SE) uudelleentulemisen trendiin ja lisääntyvään kiinnostukseen sitä kohtaan. Ilmiö liittyy myös PLM:n laajenemiseen tuoteprosessien alkupäähän ottaen haltuun SE:n osa-alueita ja tuotteeseen liittyviä malleja laajemmin. Toisaalta mallipohjaisuuteen laajasti ajateltuna liittyvät teknologiat alkavat olla tänä päivänä riittävän kypsiä yrityskäyttöön, jolloin kehitystä kohdistetaan enemmän prosessien, menetelmien, organisaatioiden, informaation hallinnan ja infrastruktuurin kehittämiseen.

Prosessit ovat luonteeltaan usein dynaamisia. Koska inhimillinen toiminta pitää sisällään myös epärationaalisuutta ja epäennustettavuutta, järjestelmiä ja prosesseja tulee tutkia ja kehittää myös johtamisen, sosiologian, politiikan yms. tieteiden keinoin. Kuva 1 havainnollistaa, mihin SE:n tekniset prosessit sijoittuvat liiketoiminnan kontekstissa. MBSE:n päätavoite on tukea teknisiä prosesseja, mutta niillä on liittynyt myös monimutkaisiin liiketoimintaprosesseihin, projekti- ja TK-prosesseihin, organisaatiomalleihin, tiedonhallintaan sekä muuhun infrastruktuuriin. On siis tärkeää mallintaa sekä tekninen järjestelmä että sen kehittämiseen osallistuvat prosessit menetelmineen, työkaluineen, tekijöineen jne. Teknisiä prosesseja ylemmältä tasolta nähtynä järjestelmän määrittelyssä ja kehittämisessä on neljä vaihetta: 1) Järjestelmän kontekstin ymmärtäminen ja määrittely: Reunaehdot, rajapinnat, ihmiset ja organisaatiot, vuorovaikutukset; 2) järjestelmän käyttäminen – ketkä käyttävät, miten, miksi jne. esimerkiksi tarinan muodossa; 3) järjestelmän toteuttaminen: rakenne ja arkkitehtuuri, toiminnot ja käyttäytyminen, elementit; 4) käyttö ja elinkaarituki.



Kuva 1. Mallipohjaisuuden sijoittuminen liiketoimintaan.

Mitä on MBSE eli mallipohjainen SE?

Kompleksisten järjestelmien tekniikka (Systems Engineering, SE) on osa insinööri-taito-käsitettä (engineering) johon on aina kuulunut erilaisten mallien (pienoismallit, piirustukset, laskelmat, digitaaliset mallit jne.) käyttö. MBSE (Model Based Systems Engineering) eli mallipohjainen SE määritellään eri lähteissä hieman eri tavoin. Jotkut rajaavat MBSE:n määritelmän lähinnä kuvailevien (deskriptiivisten) mallien käyttöön järjestelmän vaatimusten ja konseptien kuvaamiseen. Toiset näkevät sen hyvinkin laajasti kaikenlaisten mallien hyödyntämisenä järjestelmän elinkaaren aikana mukaan lukien erilaiset virtuaaliympäristöjen ja lisätyn todellisuuden sovellukset. *Tässä oppaan luvussa MBSE:tä ja mallipohjaisuutta käsitellään laajasti. Lähestymistapana luvussa on luoda ensin teoreettinen katsaus malleihin, käsitteisiin ja prosesseihin, ja peilata niitä sitten käytännön tekemiseen. Jälkimmäinen pohjautuu paljolti havaintoihin lukuisissa VTT:n tutkimus-projekteissa, joissa aihe on jossain muodossa ollut mukana.*

Olennaista mallipohjaisuudessa on sen kytkeminen osaksi SE:n prosesseja. *Voidaan ajatella että SE on ylätason prosessi, jonka sisällön hallintaa MBSE tukee.* INCOSE:n mukaan MBSE on formaalin mallinnuksen soveltamista järjestelmän (tuotteen) koko elinkaaren ajaksi [INCOSE 2007] ja mallien tärkeyden korostamista eri systeemin (tuotteen) elinkaaren vaiheissa [Estefan 2008]. Vitech [2011] on määritellyt MBSE:n suppeammin projektin ongelma- ja ratkaisuväyryyden esittämiseksi järjestelmän kuvaus-kielillä, jolloin kuvauskielinä (system definition language) Vitech mainitsee mm. SysML, Activity Diagram, Functional Flow Block Diagram (FFBD). Vitech toteaa myös että oikean kielen kehittäminen systeemin kuvaamiseen jatkuu. Mallipohjaisuus pyrkii osittain korvaamaan dokumenttipohjaista informaation hallintaa esimerkiksi järjestelmän vaatimusten määrittelyssä. PLM:n kehitys ajaa myös tekemistä tähän suuntaan.

Mitä ovat mallit?

Malli kuvaa tai simuloi kohteena olevaa järjestelmää. Järjestelmän mallinnuksessa on kiinnitettävä huomiota siihen mitkä ominaisuudet ovat keskeisiä mallissa, ja mitä tarkoitusta varten ne palvelevat. Mallilla on aina tietty tarkoitus. Se voi olla esimerkiksi vaikka vain järjestelmän tietyn ominaisuuden mittaaminen, mutta mallia voidaan hyödyntää myös mm. verifiointiin, kommunikointiin, koulutukseen ja ohjeistamiseen. Kun fyysinen prototyyppi pitää yleensä sisällään lähes kaikki samat ominaisuudet kuin lopputuotekin, niin malliin on yleensä määriteltävä vain tietyt tarkastelun kohteena olevat ominaisuudet, kuten käyttäytyminen, rakenne, muoto. Mallit helpottavat erityisesti abstraktien järjestelmien (esim. täysin uudet järjestelmät ja konseptit) tutkimista. Mallin ilmaisuväline (medium) voi olla verbaalinen, matemaattinen, symbolinen, graafinen tai visuaalinen. [Hubka & Eder 1984] Primäärinen tarve olisi luoda ensin kokonaisjärjestelmän malli. Usein käytännössä oikaistaan – mallilla pyritään heti pelkästään johonkin tiettyyn tavoitteeseen kuten mm. erilaiset analyttiset tarkastelut. Ongelmaksi muodostuu se, että tällöin mallista

jää uupumaan ihmisen ja ympäristön realistinen vuorovaikutus sekä muiden järjestelmän osien relaatiot. Pahimmassa tapauksessa malli johtaakin vain harhaan sen sijaan, että auttaisi päätöksen tekoa oikeaan suuntaan. Tärkeätä on myös aina selkeästi tuoda esille mitä mallilla pyritään tuomaan esille ja mitä seikkoja mallin tekemisessä ei ole otettu lainkaan huomioon.

Mallin konteksti vaihtelee:

- Abstraktista konkreettiseen
- Aineellisesta käsitteelliseen
- Yleisestä spesifiin

Mallin funktio ja tarkoitus voi olla yksi tai kombinaatio seuraavista:

- Kuvaileva (deskriptiivinen) – mallin tai todellisuuden tietyn aspektin selittämiseen
- Ennustava – tietyn käyttäytymisen ennakoimiseen
- Tutkiva – tietyn käyttäytymisen muuttuminen eri olosuhteissa
- Suunnittelu – uudenlaisen sovelluksen ehdottamiseen
- Määräävä – ohjeistava

Mallinnusta ohjaavia tekijöitä

Yritysten strategioita ohjaa tarve parantaa innovaatiokyvykkyyttä ja kompetenssia yleensä. Asiakkaiden ja muiden sidosryhmien tarpeiden ja tavoitteiden todellinen ymmärtäminen ja todentaminen ovat nousseet yhdeksi keskeiseksi ydin-kompetenssiksi niin B2B- kuin kuluttajatuotepuolella. Tuotekehitystä ohjaavia tekijöitä ovat mm. käyttäjäkokemus, turvallisuus, päivitettävyyden, kestävä kehitys. Ne eivät saisi kuitenkaan jäädä ainoiksi tuotekehitystä ohjaaviksi tekijöiksi, tärkeätä on oivaltaa niiden taustalla olevat asiakas- ja sidosryhmätarpeet ja niiden toteuttaminen.

Markkinoiden dynamiikka lisääntyy jatkuvasti ja monitekniset tuotteet sekä arvoverkot tulevat jatkuvasti monimutkaisemmaksi; kyse on kompleksisesta kokonaisjärjestelmästä, joka pitää hallita. Tuotteen elinkaaren hallinta lisää monimutkaisuutta: miten hallita keskenään ristiriitaisia tavoitteita ja vaatimuksia, joita esitetään eri sidosryhmien tahoilta, miten tukea päätöksentekoa ristiriitaisuuksien ratkaisuisissa erilaisin kompromissein. Samanaikaisesti laadukkaita, personoituja, konfiguroitavia ja kustannustehokkaita tuotteita tulee pystyä toimittamaan varmasti ja yhä nopeammin. Tuotteiden käytön perustana olevia kokonaisjärjestelmiä tulee myös pystyä kehittämään ketterästi. Näiden kaikkien ilmiöiden ymmärtämiseen ja hallitsemiseen kokonaisjärjestelmän lähestymistavasta lähtevä järkevästi toteutettu mallipohjaisuus ja siihen liittyvät uudet teknologiat ja lähestymistavat voivat tuoda helpotusta. Mallipohjaisuus mahdollistaa myös ympäristön huomioon ottamisen sekä sidosryhmien ja toimijoiden osallistamisen järjestelmien kehittämiseen ja toteuttamiseen.

Teknisten järjestelmien ja prosessien mallit

Kokonaisjärjestelmä koostuu ympäristöstä, ihmisistä ja teknisestä järjestelmästä. Termiä tekninen järjestelmä (technical system) käytetään tässä yhteydessä tarkoittamaan ihmisen tekemiä artefakteja, kuten teknisiä tuotteita ja prosesseja. *Tekninen järjestelmä on siis suunnittelutoiminnan kohde, joka ottaa huomioon kokonaisjärjestelmän. Kompleksisten järjestelmien tekniikka (SE) on prosessi, joka tukee teknisten järjestelmien suunnittelua ja toteutusta, ja MBSE on siihen liittyvä malleja systemaattisesti hyödyntävä metodiikka.* Suunnittelutoiminta pitää sisällään laajasti informaation tuottamista, hakemista, käsittelyä ja siirtämistä liittyen tuotteisiin.

Yhtenä tämän kirjan lähteenä käytetty Hubkan ja Ederin yleinen teknisten järjestelmien teoria (Theory of Technical Systems [Hubka & Eder 1984], TTS) pyrkii kokonaisvaltaiseen kuvaukseen, joka yhdistää insinöörien, tieteen, talouden, ergonomian, johtajien, loppukäyttäjien, sosiologien, jne. näkökulmat ja vuorovaikutuksen, sekä selittää eri näkökulmien vaikutuksen teknisiin tuotteisiin. Se on moniteknillinen, monitieteellinen teoria, joka kattaa myös tuotteen elinkaaren näkökulman. Teorian yhtenä tavoitteena on ollut myös mallintaa ”abstrakti kone tai järjestelmä”, jolla voidaan kuvata mikä tahansa suunnittelun kohteena oleva tuote tai prosessi.

Huom! Tässä oppaan luvussa teknisiin järjestelmiin liittyvä käsitteistö ja terminologia perustuu TTS:ään. Käytännössä terminologia tosin vaihtelee esim. yrityksittäin ja toimialoittain.

Tehokas mallipohjainen SE vaatii mallinnukselta tiettyjä kyvykkyyksiä. Malli on aina rajattu yksinkertaistus kokonaisjärjestelmästä, mutta sen tulee tarjota kokonaisjärjestelmästä tarvittava ”näkyvä” kulloiseenkin tarpeeseen. Teknisten järjestelmien olemusta kuvaavat yhdessä näiden järjestelmien (yhtenevästi kokonaisjärjestelmän kanssa):

- Tarkoitus ja päämäärä
- Toimintatapa
- Komponentit ja rakenne
- Järjestelmän tilat.

Sosiotekninen järjestelmä koostuu ihmisistä, teknisestä järjestelmästä ja aktiivisesta ympäristöstä. Tämän järjestelmän hyvyys riippuu kolmen osa-alueen yhteistoiminnasta. Yksi keskeinen suunnittelutehtävä on tehtävien allokoiminen ihmisille ja teknisille järjestelmille hyödyntäen molempien vahvuuksia. Ihmisellä on tiettyjä vahvuuksia verrattuna teknisiin järjestelmiin. Näitä ovat ennen kaikkea kyvykkyys tehdä päätöksiä rajatun informaation perusteella sekä joustavuus erilaisissa tilanteissa. Sosioteknisen järjestelmän mallin tulee kuvata kuka tekee, miten/millä tekee ja missä ympäristössä. Käytännön suunnittelutehtävässä yksinomaan teknisen järjestelmän parissa työskenneltäessä tämä kokonaiskuva tekniseen järjestelmään liittyvästä kokonaisjärjestelmästä saattaa helposti hämärtyä, mutta SE:llä ja mallipohjaisuudella voidaan tukea systeemistä ajattelua.

SE on metodiikka monimutkaisten ja kompleksisten järjestelmien kehittämiseen. Käsitteillä monimutkainen (complicated) ja kompleksinen (complex) on hyvä nähdä ero, joka liittyy sosioteknisen järjestelmän luonteeseen. Teknisen järjestelmän (esim. lentokone) monimutkaisuus johtuu suuresta määrästä ennalta määrätysti vuorovaikutteisesti toimivia komponentteja, mutta kompleksisessa järjestelmässä on mukana joukko itsenäisiä toimijoita [Amaral & Uzzi 2007]. Ihmiset ja ympäristö aiheuttaa usein sen että suhteellisen yksinkertaisen teknisestä järjestelmästä tulee osa kompleksista järjestelmää.

Haluttu päämäärä, jota varten sosiotekninen järjestelmä on luotu, saadaan aikaan tietyillä järjestelmän käytön aikaisilla vaikutuksilla. Käytönaikaiset ominaisuudet määrittävät miten hyvin järjestelmä soveltuu tehtäväänsä. Käytönaikaiset ominaisuudet pitävät sisällään mm. luotettavuuden, turvallisuuden, eliniän, energiankulutuksen, huollettavuuden, käytettävyyden, kustannukset jne. Tekniset järjestelmät valmistetaan useimmiten tuotantoympäristössä ja ne kokevat monenlaisia vaiheita ennen varsinaista käyttöönottoa, esimerkiksi pakkaamista, kuljettamista paikasta toiseen ja varastointia. Nämä aiheuttavat myös kustannuksia, joihin vaikuttavat dimensiot, muoto, massa, materiaali, kestävyys, asennettavuus, jne.

Tuotteen toimituksen määräaika on ominaisuus, joka usein vaikuttaa hankintapäätökseen. Toimituksen määräaika asettaa määräajan myös suunnittelutyölle. Samoin suunnittelutyöhön vaikuttaa valmistettavien tuotteiden lukumäärä. Yksittäisten tuotteiden ja järjestelmien kohdalla toiminnallisuudet ja muut ominaisuudet pitää pystyä suunnittelemaan varmasti, koska prototyyppien valmistaminen ei yleensä tule kysymykseen. Mallipohjaisesti voidaan kuitenkin tutkia myös yksittäisten osajärjestelmien ominaisuuksia jo suunnitteluvaiheessa. Sarjatuotannon tai suurten erien tapauksessa taas on mahdollista testata ja kehittää tuotteen ja sen valmistuksen ominaisuuksia fyysisten tai virtuaalisten prototyyppien avulla. Valmistettavuus ja kokoonpantavuus ovat keskeisiä teknisen järjestelmän ominaisuuksia. Niihin liittyvät suunnittelussa tehtävät valinnat vaikuttavat noin 60–80 % järjestelmän kokonaiskustannuksista. Siksi suunnittelijoiden tulisi tietää riittävästi valmistusmenetelmistä ja niiden soveltuvuudesta erilaisiin tapauksiin. Soveltuvat

valmistusmenetelmät riippuvat myös erittäin paljon siitä onko kyseessä yksittäinen tuote vai sarjatuote tai jotain siltä väliltä. Sarjatuotannossa kannattaa esimerkiksi usein pyrkiä automaatioon ja robotiikkaan, mikä asettaa tuotteen valmistettavuudelle tiettyjä reunaehtoja. *Joka tapauksessa vuorovaikutus suunnittelijoiden ja tuotannon välillä on tärkeää ja mallipohjaisuus tuo mahdollisuuksia parantaa myös sitä.*

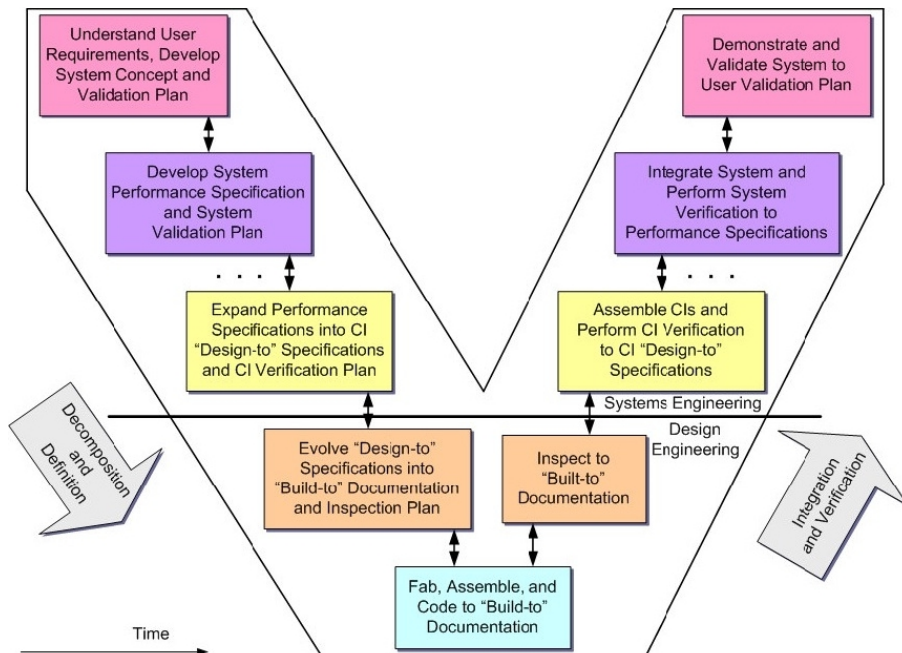
Järjestelmien ja mallien abstraktiotasot

Sosioteknisten järjestelmien mallintamiseen on kehitetty teorioita (esim. Hubka & Eder 1984), jotka pyrkivät kuvaamaan ja käsitteellistämään järjestelmän sekä sen suunnittelun ja kehittämisen geneerisesti ja neutraalisti ottamatta kantaa toimialaan, järjestelmän/tuotteen tyyppiin, tekniseen disiplinaan, jne. Järjestelmä kuvataan lähtien siitä mihin tarpeeseen se suunnitellaan tai modifioidaan ja mitkä ovat sen vaaditut ominaisuudet. Tarve uuden järjestelmän luomiseen tai olemassa olevan järjestelmän kehittämiseen voi tulla monelta taholta, esimerkiksi loppukäyttäjiltä, liiketoimintastrategiasta, tuotannon automatisoinnista, viranomaisilta, jne. Nämä *tarpeet jalostetaan vaatimusmäärittelyksi (requirements specification) ja edelleen suunnittelumäärittelyiksi (design specification), jotka ohjaavat järjestelmän suunnittelua.* Suunnittelumäärittely on teknisen järjestelmän mallin kaikkein abstraktein muoto. Se määrittelee ainoastaan tarpeet, vaatimukset, ja rajoitteet ottamatta kantaa prosessiin tai laitteisiin. Tämän transformaation ja abstraktiotasojen hallinta on keskeinen osa SE-prosesseja ja MBSE:tä. Kun teknisen järjestelmän kuvaus tulee abstraktimmaksi, mallin täydellisyys vähenee. Käytännön suunnitteluprosessissa pitäisi kiinnittää huomiota siihen, että prosessin aikana pidetään erillään tavoitteet ja keinot, sekä että tuotetaan riittävästi vaihtoehtoisia ratkaisuja arvioitavaksi ja valittavaksi.

Lähtien tarpeista ja tavoitteista, järjestelmä tulisi pystyä kuvaamaan sen ominaisuuksina (properties), vaatimuksina, toimintorakenteena, loogisena rakenteena ja komponenttirakenteena (osarakenne, tuoterakenne). Tässä yhteydessä on huomattava, että vaikka järjestelmää tai tuotetta lähdetään harvoin kehittämään tyhjästä, tarve edellä mainituille abstraktiotasoille on olemassa. Innovaatiot voivat liittyä esimerkiksi tuotteen uudenlaiseen tuotannon modulointi-periaatteeseen, jolloin toimintorakenteen ja loogisen rakenteen hallinta on välttämätöntä.

Toimintorakenne ja looginen rakenne liittyvät järjestelmän konseptivaiheeseen, jolloin toiminnot kohdennetaan toteuttaviin alijärjestelmiin, laaditaan järjestelmän arkkitehtuuri. Tämän jälkeen alkaa nk. disiplinaatio, jossa eli tekniikan alat tuottavat detaljiratkaisuja teknisiin vaatimuksiin. Tuloksena tästä prosessista muodostuu komponentti- ja osarakenne. Edellä mainitut aspektit havainnollistetaan usein ns. V-mallin (kuva 2) avulla. On huomattava, että kyse on aivan eriasista kuin kronologisesti määrittelyä lähestyvässä "vesiputousmallissa", kyse on nimenomaan järjestelmän spesifikaation eri abstraktiotasoista. Se on myös ymmärrettävä itera-

tiivisena prosessina, jossa tarpeen mukaan palataan muokkaamaan vaatimuksia, rakenteita, jne. Nykyaikaisiin PLM-järjestelmiin on alkanut ilmestyä kyvykkyyksiä tulemaan tätä nk. RFLP (Requirements – Functions – Logical – Part) -prosessia.



Kuva 2. Järjestelmän määrittelyn ja toteutuksen vaiheita havainnollistava V-malli [lähde: Estefan 2008].

MBSE:n näkökulmasta tärkeää on tieteenkin kyvykkyys mallintaa ja simuloida edellä mainittuja aspekteja. Staattiset, deskriptiiviset mallit kuvaavat järjestelmän arkkitehtuuria, rakennetta ja siinä ilmeneviä riippuvuuksia, kun taas dynaamisia malleja käytetään simuloimaan järjestelmän tiloja ja data-virtaa. Deskriptiivisiä (kuvailevia) malleja ovat mm. generiset UML- ja SysML-mallit sekä disiplinaariset hydraulikaaviot, sähkö- ja automaatio- sekä prosessikaaviot.

Teknisiä järjestelmiä voidaan havainnoida ja luokitella eri näkökantojen perusteella (Hubka & Eder 1984):

- Tehtävä ja tarkoitus, jonka järjestelmä suorittaa, sekä vaikutus, jonka se antaa (toimintorakenne).
- Sisäinen toimintaperiaatteen prosessi ja funktiot toteuttavat elimet (looginen rakenne).
- Miten järjestelmä valmistetaan ja kokoonpannaan yksittäisistä komponenteista (osarakenne).
- Järjestelmän elinkaaren *vaiheet*, mukaan lukien testaus, verifiointi ja validointi.

Teknisen järjestelmän kolme erilaista rakennetta (toiminto, looginen, osa) ovat saman *järjestelmän eri abstraktion tasoisia kuvauksia*. Koneinsinööreillä (mechanical engineers) voi joskus olla vaikeuksia hahmottaa mitä loogisella rakenteella ajetaan takaa, koska he ovat usein tottuneet käsittelemään fyysisiä kappaleita ja niiden osarakenteita. Muut insinöörialat (kuten sähkö, tietotekniikka, automaatio, jne.) ovat käyttäneet tätä periaatetta jo pitkään. Loogisella toimielinrakenteella voidaan myös kuvata suurten laitosten, kuten voimalaitosten toiminnot niiden toteutusprojekteissa. Rakenteen ja elimien kuvaamiseen on kehitetty monia symbolisia standardoituja ”kieliä”. Esimerkkejä ovat mm. sähkö- ja hydraulikka-kaaviot. Järjestelmän rakenne ja käyttäytyminen (behavior) ovat sen tärkeimmät ominaisuudet, ja rakenne määrittää käyttäytymisen.

Osarakenne kuvaa teknisen järjestelmän kaikkein konkreettisimmin ja sillä on suora yhteys järjestelmän valmistukseen. Osarakenne on riippuvainen kontekstista, esimerkiksi koneenrakennuksessa ja talonrakennuksessa käytetään erityyppisiä osarakenteen kuvauksia. Osarakenne ei riipu vain sen tuottamasta vaikutuksesta, vaan myös siitä millä periaatteella se tullaan rakentamaan/valmistamaan. Rakenne sisältää yleensä valmistus- ja kokoonpanopiirustukset. Rakenteen kehittäminen mahdollisimman hyväksi kokonaisuuden kannalta edellyttää hyviä menetelmiä, työkaluja (esim. CAE), kommunikointia ja yhteistyötä eri disipliinien välillä.

Tekniset järjestelmät siirtyvät elinkaarensa aikana tilasta (state) toiseen – konseptista suunnitteluun, valmistukseen, käyttöön ja käytöstä poistoon. Teknisestä järjestelmästä keskusteltaessa ja sitä mallinnettaessa pitää mainita mikä on sen kulloinenkin tila. Suunnittelijoiden pitäisi käydä mielessään läpi kaikki suunnittelemansa järjestelmän tilat ja niiden vaatimukset. *Mallien ja simulointien avulla tehdyt kokeet ja testit voivat toimia myös apuna järjestelmää hahmotellessa.*

Järjestelmien moniteknisyyden ja luokittelut

Systems Engineeringin kohteena on yleensä kompleksiset ja monitekniset sosiotekniset mekatroniset järjestelmät, jotka ovat yhdistelmiä mekaniikkaa, automaatiota, elektroniikkaa ja ohjelmistoja. Mukana järjestelmässä on myös aina ihminen ja ympäristö vähintäänkin jossain järjestelmän elinkaaren vaiheessa. Mallipohjaisuus mahdollistaa tällaisten monitahoisten järjestelmien tarkastelun ja simuloinnin, mutta samalla moniteknisyyden ja sosioteknillinen näkökulma tuovat suuria haasteita mallien tuottamiseen, hallintaan ja ymmärtämiseen.

Eri tekniikan alat (kuten mekaniikka, hydraulikka, ohjelmistotekniikka, ergonomia) ja muut sidosryhmät käyttävät erilaisia käsitteitä, malleja ja mallinnustyökaluja. Niiden yhteensovittaminen on tällä hetkellä keskeinen tutkimus- ja kehityskohde. Toisaalta, tekniset järjestelmät voidaan luokitella (Hubka & Eder 1984) monesta näkökulmasta niiden samankaltaisuuden ja sukulaissuhteen perusteella. Tämä mahdollistaa mm. teknologiasiirron eri kategorioiden välillä. Mallipohjaisuuden näkökulmasta luokittelu voi parantaa mallien monikäyttöisyyttä ja modulaarisuutta.

Tätä voidaan tehostaa standardoimalla ja luomalla tietokantoja yritysten järjestelmistä ja komponenteista. Luokittelukriteereitä ovat mm. järjestelmän

- Tarkoitus ja toimintaperiaate
- Kohdemuuttujan tyyppi (esim. materiaali, energia, informaatio jne.) ja prosessin vaikutusperiaate (esim. mekaaninen, hydraulinen, sähköinen, kemiallinen jne.)
- Olosuhteet (esim. trooppinen/arktinen ilmasto jne.)
- Monimutkaisuuden aste
- Valmistusprosessin samankaltaisuus (esim. yksittäiskappale, konfiguraatiot, massatuotanto jne.)
- Muoto, valmistusmateriaali
- Valmistaja.

Järjestelmä voi tietenkin kuulua samanaikaisesti moneen luokkaan ja kategoriaan. SE:n näkökulmasta järjestelmän monimutkaisuus on kiinnostava aspekti. Järjestelmän monimutkaisuus voi perustua esimerkiksi sen rakenteen monimutkaisuuden asteeseen. Rakenne voidaan myös jakaa esimerkiksi osakokoonpanoihin, jolloin monikäyttöisyys paranee. Luokittelu mahdollistaa järkevän tuotannon suunnittelun sekä tehtaan lattiatasolla, että globaalisti verkostoissa. Valmistukseen liittyen järjestelmät voidaan jakaa myös yksittäiskappaleisiin (one-off), piensarjoihin, variantteihin, massatuotantoon jne. Tuotteen prototyyppi on myös yksi erityinen luokka ja joskus prototyyppi on myös myytävä tuote. Prototyyppiä hyödynnetään yleensä järjestelmän vaatimuksiin perustuvien ominaisuuksien/toimintojen verifiointissa ja validoinnissa. *Mallipohjaisuudella ja virtuaaliprototyypoinnilla voidaan osittain korvata fyysiset prototyypit.*

Monimutkaisuus on suoraan kytketty järjestelmän suunnitteluun, jolloin suunnittelijalta vaaditaan tiettyä erityisosaamista. Järjestelmät voidaan luokitella myös suunnittelutehtävän monimutkaisuuden perusteella, mikä auttaa esim. suunnittelun ulkoistamisessa. Vaikeusaste liittyy myös järjestelmän omaperäisyyteen, sen funktioiden ja vaikutuksen monimutkaisuuteen, rakenteeseen, mallintamiseen ja analysointiin, kustannusten optimointiin, jne. Järjestelmiä voidaan myös luokitella abstraktiivisemmin niiden rakenteen taksonomian perusteella.

Systems Engineeringin tekniset prosessit

Teknisellä prosessilla tarkoitetaan tässä yhteydessä sitä tekemistä, joka käytännössä toteuttaa teknisen järjestelmän tai tuotteen määrittelyn ja toteuttamisen osana liiketoiminnan kokonaisuutta (ks. kuva 1, sivu 56). Teknistä prosessia ohjaa usein esimerkiksi tuotekehitysprojektit (prosessit) ja laatu prosessit. SE nähdään tässä ylätasoinen prosessina, joka pitää kokonaisuudesta huolen taloudellisteknisestä näkökulmasta. Teknistä prosessia kuvataan usein kuvan (kuva 2, sivu 62) ”V-mallilla”. Tekninen prosessi pitää sisällään sekä luovaa toimintaa, joka lähestyy

jopa taidetta esimerkiksi asiakastarpeiden tulkinnassa teknisiksi vaatimuksiksi ja spesifikaatioiksi, että kurinalaista disiplinaatiota. Suomen kielessä ei löydy tämän päivän toiminnan huomioon ottaen hyviä suoria vastineita englannin ”engineering” ja ”designing” termeille. Suomen kielessä suunnittelu tarkoittaa sekä luovaa ”taiteilua” ja kokonaisuuden hallintaa, että yksityiskohtaista toteuttamista. SE:n tärkeimpiä tehtäviä teknisen prosessin kannalta on kokonaisuuden optimoiminen, eri abstraktiotasojen hallinta ja moniteknisyyden ja eri disiplinaalien yhdistäminen sekä kommunikoinnin parantaminen. Tähän mallipohjaisuus nähdään potentiaalisena keinona. SE voidaan nähdä myös yhtenä teknisten järjestelmien tieteellisten teorioiden käytäntöön panijana.

Järjestelmän ulkoiset ja sisäiset ominaisuudet

Kuten aikaisemmin on todettu, teknisten järjestelmien ja tuotteiden suunnittelu ja kehittäminen tulisi perustua asiakkaiden, loppukäyttäjien ja muiden elinkaaren aikaisten sidosryhmien tarpeisiin. Loppukäyttäjillä ja muillakin sidosryhmillä on kuitenkin usein ns. ”black-box”-näkökulma tekniseen järjestelmään. He ovat kiinnostuneita lähinnä järjestelmän ulkoisista ominaisuuksista. Käyttäjät asettavat tiettyjä syötteitä systeemiin ja saavat tiettyjä haluamiaan tuotoksia, mutta eivät välttämättä tiedä (eivätkä ole kiinnostuneita) mitä systeemin sisällä tapahtuu. Black-boxin sisäisiä ominaisuuksia kutsutaan myös *suunnittelu ominaisuuksiksi* (*design properties*). Suunnitteluominaisuudet voidaan luokitella kolmeen pääryhmään [Hubka & Eder 1984]:

1. Erityiset suunnitteluominaisuudet, jotka liittyvät suoraan haluttuun teknisen järjestelmän toteuttamaan prosessiin.
2. Yleiset suunnitteluominaisuudet, esim. lujuus, jäykkyys, kovuus, korroosionkestävyys jne.
3. Perusominaisuudet, joiden avulla suunnittelija saavuttaa kaikki tarvittavat järjestelmän ominaisuudet; rakenne, muoto, dimensiot, materiaali, pinnanlaatu, toleranssit, valmistusmenetelmät sekä näiden kombinaatiot.

Suunnittelun tärkein tavoite liittyy tekniseltä järjestelmältä vaadittujen toiminnallisuuksien (funktioiden) toteuttamiseen. Toisaalta suunnittelussa pitäisi pyrkiä tasapainoon järjestelmän teknisten, ergonomisten ja esteettisten ominaisuuksien kesken. On tärkeää että eri alojen asiantuntijat otetaan mukaan suunnittelutehtävään riittävän aikaisin. Systeeminen ajattelu (system thinking) tarjoaa mahdollisuuden ymmärtää ja ratkaista ongelmia kokonaisuutena. Yhteinen terminologia ja abstrakti järjestelmäajattelu mahdollistavat myös paremman kommunikoinnin eri insinööri- ja tieteenalojen välillä. Se myös helpottaa tuoteprosessien kuvaamista loogisista operaatioista koostuvaksi prosessiksi.

Tietokoneiden aikakaudella on myös erittäin tärkeää ymmärtää kokonaisuus ja ihmisten tarpeet niin ettei insinööri kehity vain työkalujen lähtökohdista. Suunnittelu on myös ihmisten intuitiivista luovaa työtä, joka vaikea tai mahdotonta muuntaa algoritmeiksi.

Järjestelmä vaatimusten ja spesifikaatioiden näkökulmasta

Kun puhutaan järjestelmän vaatimuslistoista, kyseessä on teknisen järjestelmän haluttujen ominaisuuksien määrittelystä. Spesifikaatio voi esiintyä monessa muodossa riippuen tarkoituksesta ja sitä käsittelevistä henkilöistä, esimerkiksi [Hubka & Eder 1984]:

- Viranomaiset usein määrittelevät muodollisesti mitä vaatimusmäärittelyn (requirements specification) pitää sisältää
- Vastaava määrittely on esimerkiksi asiakkaan määrittelemä tarjouspyyntö (request for tender)
- Sopimuksessa määritellään ja allekirjoitetaan sopimusmäärittely (design contract specification)
- Tämän taas suunnittelijat tulkitsevat suunnittelumäärittelyksi (design specification).

Edellä mainitut koskevat järjestelmiä, jotka eivät vielä ole fyysisesti olemassa. Olemassa olevia järjestelmiä koskevat valmistusmäärittelyt, testimäärittelyt, käyttömäärittelyt (manuaalit) jne. Epätäydellinen varhainen spesifikaatio johtaa usein virheisiin lopputuotteessa. Spesifikaatioiden muotoon ja sisältöön vaikuttavat järjestelmän toimintojen monimutkaisuus, suunnittelutehtävän vaikeus (täysin uusi järjestelmä vs. pienet parannukset), vaatimukset lisäominaisuuksille (turvallisuus, elinkaari jne.), ongelman asettaja tai rahoittaja.

Epäonnistumisen riskit kasvavat yleensä suoraan suhteessa suunniteltavan järjestelmän uutuuteen ja monimutkaisuuteen – ja vähenevät kun vaatimukset ja suunnitelma kypsyvät. Koska vaatimuksia iteroidaan, ne pitää laatia muutosmielessä. Tähän liittyen on myös *tärkeää, että vaatimukset kuvataan kyvykkyyksinä ja ominaisuuksina – ei toimintoina ja ratkaisuinä*. Uutuus tarkoittaa ominaisuuksia ja niiden yhdistelmiä, joita ei aikaisemmin ole vastaavissa järjestelmissä sovellettu, ja jotka johtavat kaupallisiin innovaatioihin. Suunnittelu sisältää silloin tyypillisesti konseptisuunnittelua.

Asiakas voi toteuttaa tarpeensa joko ostamalla valmiin tuotteen tai tilaamalla yksittäisen tuotteen. Näissä kahdessa tapauksessa spesifikaatioita määrittelevät eri tahot. Ensimmäisessä tapauksessa myynti ja markkinointi osastot määrittelevät vaatimusmäärittelyt, jälkimmäisessä asiakas itse osallistuu määrittelyyn. Kun asiakas tilaa tuotteen suoraan valmistajalta, hän vahvistaa vaatimukset. Samalla hän ottaa riskin siitä että vaatimukset ovat väärin määritellyt. Kun markkinointi ja myynti määrittelevät vaatimukset uudelle tuotteelle, esiintyy riski siitä että asiakkaat eivät halua ostaa tuotetta eli vaatimukset eivät kohtaa. Koska vaatimusmäärittelyt eivät voi koskaan olla täydellisiä, suunnittelijoiden pitää täydentää niitä

lisäominaisuuksilla. Asiakkaat ja markkinointikin määrittelevät yleensä vain tuotteen ulkoisia ominaisuuksia (external). Vain tietyn tyyppiset asiakkaat (kuten armeija, viranomaiset) määrittelevät myös tuotteen sisäisiä ominaisuuksia, kuten materiaalit, pinnan laadut jne. Edellä mainitut vaatimukset koskevat lopputuotetta, joka toimitetaan asiakkaalle. Tilanne on toinen kun puhutaan osajärjestelmistä ja niiden elementeistä. Silloin suunnittelijat määrittelevät vaatimuksia lähinnä itselleen. Mukana on myös rajoitteita (constraints), jotka rajaavat pois tiettyjä vaihtoehtoja. Teknisten rajoitteiden lisäksi esiintyy mm. lakeja, sääntöjä, eettisiä periaatteita jne., jotka myös rajaavat suunnitteluvaihtoehtoja. Vaatimukset voidaan luokitella niiden tärkeyden mukaan esimerkiksi kolmeen ryhmään:

- Pakolliset vaatimukset
- Vaatimukset joista voidaan joustaa tietyissä olosuhteissa
- Toiveet, joita ei ole pakko täyttää, mutta jotka tuovat lisäarvoa.

Spesifikaation vaatimuksia koskeva luokittelu tulee hyväksyä yhdessä asiakkaan ja toteuttajien kesken.

Yksi vaatimusryhmä koostuu epäsuorista implisiittisistä (implicit) vaatimuksista, jotka suunnittelijat pyrkivät aina täyttämään etiikan ja ammattitaidon perusteella, vaikka niitä ei erikseen määritellä. Näitä ovat: yleiset käyttöön liittyvät vaatimukset (minimoitu energiankulutus, massa jne.), ergonomia, resurssien hyödyntämisen maksimointi, taloudelliset tekijät. Näitä asioita määritellään monenlaisissa oppaissa ja suosituksissa.

Model-Based Systems Engineering liiketoiminnan näkökulmasta

Taloudelliset ominaisuudet yhdistävät monia edellä mainittuja ominaisuuksia niitä kuvaaviksi tunnusluvuiksi – kustannuksiksi ja tuloiksi. Kustannuksia ovat suunnitteluun, hankintaan, valmistukseen, jakeluun yms. liittyvät kustannukset sekä opeointiin liittyvät käyttökustannukset. Nämä yhdessä määrittävät järjestelmän elinkaarikustannukset. Suunnittelutyön hyvyttä voidaan myös tutkia analysoimalla yksittäisiä kustannuslajeja, joita tarvitaan vaadittujen järjestelmän ominaisuuksien toteuttamiseen, sekä niiden välisiä vuorovaikutteisia kytkentöjä. Kustannuksia voidaan usein pienentää maksimoimalla uudelleen käytettävyys niin suunnittelussa, kuin valmistuksessa. On myös tärkeää analysoida kustannusten näkökulmasta onko taloudellisinta valmistaa tuote tai komponentit itse, ostaa ne toimittajalta tai ostaa valmistus alihankintana. Menestyvän liiketoiminnan kannalta on tärkeää nähdä ja kehittää järjestelmää kokonaisuutena osana arvoketjua ja elinkaarta yksittäisten toimintojen tai teknologian optimoinnin sijasta.

Taloudellisiin ominaisuuksiin liittyy kaksi keskeistä tunnuslukua:

- Tehokkuus (effectiveness): Prosessin vaikutus / kustannus
- Kannattavuus (profitability): Tuotto tietynä ajanjaksona / sidottu pääoma.

Teknisen järjestelmän elinkaaren lopussa se poistetaan käytöstä ja likvidoidaan. Tämä on myös ominaisuus, joka pitää ottaa huomioon järjestelmän suunnittelussa.

Kuva 1, sivu 56 havainnollistaa mihin SE:n tekniset prosessit sijoittuvat liiketoiminnan kontekstissa. MBSE:n päätavoite on tukea teknisiä prosesseja, mutta niillä on liityntä myös monimutkaisiin liiketoimintaprosesseihin, projekti- ja TK-prosesseihin, organisaatiomalleihin, tiedonhallintaan sekä muuhun infrastruktuuriin. On siis tärkeää mallintaa sekä tekninen järjestelmä että sen kehittämiseen osallistuvat prosessit menetelmineen, työkaluineen, tekijöineen jne. Teknisiä prosesseja ylemmältä tasolta nähtynä järjestelmän määrittelyssä ja kehittämisessä on neljä vaihetta:

1. *Järjestelmän kontekstin ymmärtäminen ja määrittely.* Reunaehdot, rajapinnat, ihmiset ja organisaatiot, vuorovaikutukset
2. Järjestelmän käyttäminen; ketkä käyttävät, miten, miksi, jne. esimerkiksi tarinan muodossa

3. Järjestelmän toteuttaminen: rakenne, toiminnot ja käyttäytyminen, osat
4. Käyttö ja elinkaarituki.

Liiketoiminnan näkökulmasta järjestelmän konteksti kokonaisuudessaan on tärkeä. Mitkä ovat järjestelmän rajapinnat ja osa-järjestelmät arvoverkoston ja elinkaaren näkökulmasta, mitkä ovat sidosryhmät ja niiden väliset vuorovaikutukset koko järjestelmän elinkaaren ajan? Tämä kokonaisuus on kompleksinen, eli se sisältää lukuisia itsenäisiä toimijoita jotka eivät usein näe kokonaiskuvaa tai joilla on jopa ristiriitaisia tavoitteita.

Liiketoiminta määrittelee myös tavoitteet ja kriteerit järjestelmän kehittämiseen. Tämä tulisi tapahtua tiiviissä yhteistyössä loppukäyttäjien, asiakkaiden ja elinkaaren sidosryhmien kanssa. Projektiprosessit toteuttavat liiketoiminnan asettamat tavoitteet, jäsentävät päätöksenteko-prosessin sekä johtavat käytännön tekemistä teknisten prosessien transformaatioiden eli järjestelmän toteuttamiseen liittyen.

Mallipohjaisuuden tuomia hyötyjä ovat mm:

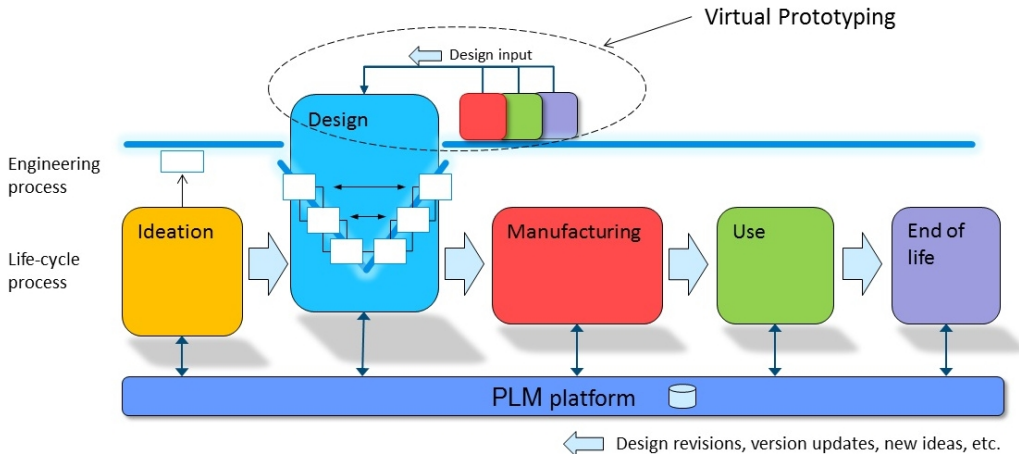
- Järjestelmän kokonaisvaltaisen ymmärtämisen parantuminen
- Symbolinen kieli abstraktien tarpeiden, vaatimusten ja järjestelmien ymmärtäminen
- Järjestelmämallit päätöksenteon tukena
- Järjestelmän rakenteen ja toiminnallisuuden sekä muiden ominaisuuksien simuloiminen.

Suunnittelijat luovat teknisen järjestelmän ja suurimman osan sen ominaisuuksista. He eivät suoraan vaikuta valmistukseen, käyttöön, ylläpitoon, mutta luovat edellytykset niille. Mallipohjaisuus tukee kokonaisoptimaaliseen ratkaisuun pääsemistä.

Laajennettu Model-Based Systems Engineering ja virtuaaliprototyointi

Tuoreessa kirjallisuudessa (esimerkiksi [Eigner et al. 2012]) esiintyy käsite "laajennettu mallipohjainen SE" ("Extended MBSE"). Tämä tarkoittaa sitä että mallipohjaisuuden käsitettä on laajennettu kattamaan kuvailevien (deskriptiivisten) mallien lisäksi myös virtuaaliprototyointiin eli monitekniisiin simulointimalleihin (kuten Matlab) sekä disipliini-simulointeihin ja visualisointeihin.

Virtuaaliprototyointi on tässä yhteydessä termi, jolla tarkoitetaan joukkoa digitaalista tuotetietoa tuottavia, analysoivia ja hyödyntäviä työkaluja ja menetelmiä, esimerkiksi erilaisia simulointeja, virtuaaliympäristöjä, CAE-työkaluja jne. Virtuaaliprototyointi käsittelee tyypillisesti dynaamisia malleja. Tämän luvun tavoitteena on kuvata virtuaaliprototyoinnin asema osana SE:tä ja sen suhde tekniseen järjestelmään ja tuoterakenteisiin, tuoteprosessiin ja -projektiin, luovaan suunnittelu-prosessiin, tuotteen elinkaareen yms. Virtuaaliprototyointia voidaan hyödyntää mm. abstraktien ja kompleksisten mallien visualisoinnissa ja tulkinnassa, vaatimusten määrittelyssä ja validoinnissa, suunnitelmien evaluoinnissa, disipliinien välisessä kommunikoinnissa yms.



Kuva 3. Virtuaalisen tuotteen ja reaalisen tuotteen suhde, sekä tuotteen elinkaaren prosessin suhde SE-prosessiin. Elinkaaren aikaisia vaiheita voidaan simuloida virtuaalimalleilla.

Perinteisesti virtuaaliprototyyppiä on sovellettu usein järjestelmän teknisten ratkaisujen verifiointiin hyvin myöhäisessä tuoteprosessin vaiheessa tavallaan omana disiplinaan suunnittelun jatkeena. Mallipohjaisessa paradigmassa tulokulma pitäisi olla päinvastainen, eli dokumenttien sijaan malleilla ja simuloineilla lähdetään määrittelemään tarpeita, tavoitteita, vaatimuksia – jotka sitten konkretisoituvat suunnittelussa (kuva 3). Tämän prosessin kehittäminen ja hallinta on toki haastavaa, mutta monet ovat jo ottaneet askeleita siihen suuntaan. SE tavallaan oikeuttaa toiminnallisten simulointimallien tekemisen jo tuoteprosessin alkupäässä.

Virtuaaliprototyyppien määritelmiä

Virtuaaliprototyyppiin liittyvistä käsitteistä ja termeistä ei vielä muodostunut yksimielisyyttä sen paremmin käytännön elämässä, tieteessä, kuin standardeissa. Syynä lienee nuorehko teknologia sekä eri toimialoilla ja yrityksissä muodostunut kulttuuri- ja slangisanasto. Tieteessä virtuaaliprototyyppi ei muodosta omaa koulukuntaa vaan on enemmän tai vähemmän osana esimerkiksi mekaniikka, tietojenkäsittelytieteitä tai suunnittelutieteitä. Virtuaaliprototyyppiin liittyen käytetään yleisesti myös esimerkiksi slanginomaisia termejä kuten virtuaalisuunnittelu (virtual engineering) tai virtuaalitekniikka (virtual technology). Määrittelyt riippuvat myös siitä mitä teknologioita, työkaluja ja menetelmiä luetaan tähän löyhään kategoriaan kuuluvaksi.

Tässä luvussa käytetään ylätasoina käsitteinä virtuaaliprototyypointia (tekeminen) ja virtuaaliprototyyppi (kohde, järjestelmän malli). Virtuaaliprototyypin instanssina voi olla esimerkiksi virtuaaliympäristö (VE, virtual environment) ja käyttöliittymänä virtuaalitodellisuus (VR, virtual reality). Wang [2002] on määritellyt käsitteet näin: *“Virtuaaliprototyyppi (tai digitaalinen malli) on fyysisen tuotteen/järjestelmän tietokone-simulointi. Sillä voidaan esittää, analysoida ja testata asioita järjestelmän elinkaaren eri vaiheista, kuten suunnittelusta, valmistuksesta, huollosta, kierrätyksestä, kuten todellisella fyysisellä mallilla. Virtuaaliprototyypin rakentamista ja testaamista kutsutaan virtuaaliprototyypoinniksi”*. Tähän voisi lisätä että virtuaaliprototyypoinnilla voidaan tutkia myös sellaisia asioita, joilla fyysisellä järjestelmällä ei pystytä tutkimaan.

Jotta virtuaaliprototyypointi tukee SE- ja MBSE-prosesseja, mallin (eli virtuaaliprototyypin) on kyettävä ilmaisemaan järjestelmän ominaisuudet, rakenteet, toiminnot, muodot. Virtuaaliprototyypoinnin tulee siis tukea teknisen prosessin transformaatiota tarve- ja vaatimusmäärittelystä valmiiksi järjestelmäksi. Virtuaaliprototyypoinnin osa-alueita ovat mm:

- Virtuaaliympäristöt (VE)
- Virtuaalitodellisuus (VR)
- Lisätty todellisuus (AR)
- Virtuaalisimulaattorit tuotekehityksessä ja koulutuksessa
- Simuloinnit: monikappaledynamiikka, FEA yms.

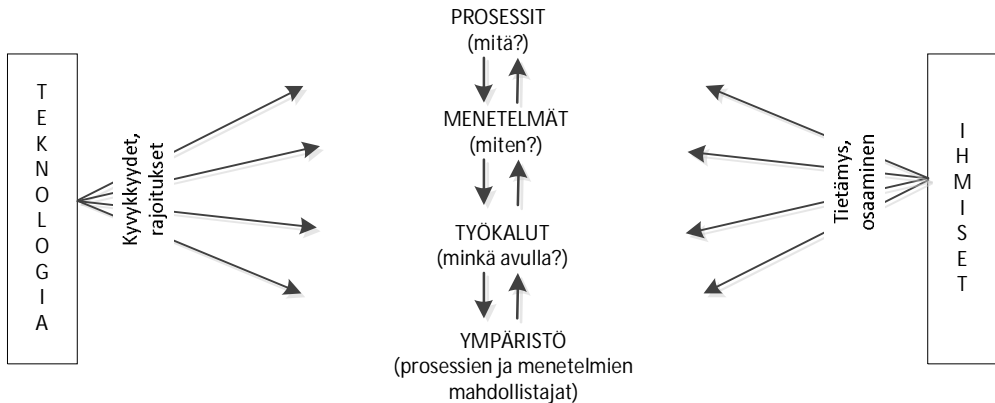
Edellä mainittuja käytetään eri tarkoituksiin järjestelmän määrittelyn ja prosessien eri vaiheissa, esimerkiksi tarpeiden ja vaatimusten määrittelyyn, verifointiin ja validointiin, testaukseen. Perinteisesti virtuaaliprototyypit ja niihin liittyvät simuloinnit ovat jakaantuneet reaali-aikaisiin (virtuaaliympäristöt, simulaattorit yms.) ja ei-reaali-aikaisiin simulointeihin. Tietokoneiden laskentatehon ja ohjelmistojen kehittyessä tämä jaottelu on kuitenkin hämärtymässä.

Prosessit, menetelmät ja työkalut

Systems Engineering on määritelty prosesseina mm. useissa standardeissa. Myös yleiseen tuotekehitykseen ja arvoketjuihin liittyviä referenssimalleja löytyy kirjallisuudesta lukuisia (esimerkiksi [Ulrich & Eppinger 2003]). Yrityksissä on nykyään yleensä kuvattu prosesseja vähintäänkin laatukäsikirjassa esimerkiksi projektien päätöksenteon ja dokumentoinnin tasolla. Prosessit saattavat kuitenkin olla liian yleisellä tasolla määriteltyjä tai ne eivät kohtaa todellista tekemistä. Prosessit voivat myös olla liian jäykkiä ja raskaita, mikä ei tietenkään ole tarkoitus. Hyvät ja virtaviivaiset prosessit tukevat ja nopeuttavat tekemistä, mutta niiden määrittely vaatii aikaa ja investointeja, sekä sidosryhmien osallistamista. Ilman riittäviä prosesseja ja toimintatapoja aikaa ja rahaa tuhlaantuu siihen että toistuvasti ratkaistaan

samoja ongelmia, odotetaan jotain tai etsitään tietoa. Pitää myös muistaa että prosessit ovat dynaamisia ja niitä pitäisi päivittää jatkuvasti kuin toimintaympäristö muuttuu.

Prosesseista puuttuu usein tarkemmat määrittelyt, kuten niihin liittyvät menetelmät ja työkalut, sekä syötteiden ja tulosten määrittely. Prosessit saatetaan kuvata lohkokaaviona, mutta ei sitä mitä kaavion laatikoiden sisällä ja niiden välillä tapahtuu. Menetelmien ja työkalujen määrittely on erityisen tärkeää MBSE:n kohdalla. Prosesseihin, menetelmiin ja työkaluihin liittyy olennaisesti myös mallien ja muun datan hallinta esimerkiksi PDM-järjestelmässä.



Kuva 4. Teknologian ja ihmisten vaikutusten elementit. [Estefan 2008]. [INCOSE MBSE initiative].

Virtuaaliprototyyppien hyödyt ja niiden todentaminen

Virtuaaliprototyyppien yhtenä tavoitteena on perinteisesti nähty tuoteprosessien tehostaminen ja nopeuttaminen mm. vähentämällä fyysisten prototyyppien tarvetta ja parantamalla päätöksentekokykyä. Teollisuudessa on otettu käyttöön virtuaaliprototyyppiä metodiikkana enenevässä määrin nyt, kun teknologia alkaa olla riittävän kypsää. Kansainvälisesti varsinkin auto- ja ilmailuteollisuus ovat olleet pioneereja virtuaaliprototyyppien käytössä, mutta myös kotimaisia esimerkkejä löytyy. On selvää, että virtuaaliprototyyppien ja sen alalajien, kuten virtuaali-ympäristöjen ja simulaattorien hyödyt ilmenevät eri tavalla massatuotannossa (kuten autot) tai lentokonevalmistajilla (suuret resurssit), kuin suomalaisella teollisuudella. Juuri tästä syystä on tärkeää pohtia mallipohjaisuutta laajasti kunkin toimialan, liiketoiminnan sekä tuoteiston näkökulmasta. Virtuaaliprototyyppi nähdään yleisesti metodologiana joka:

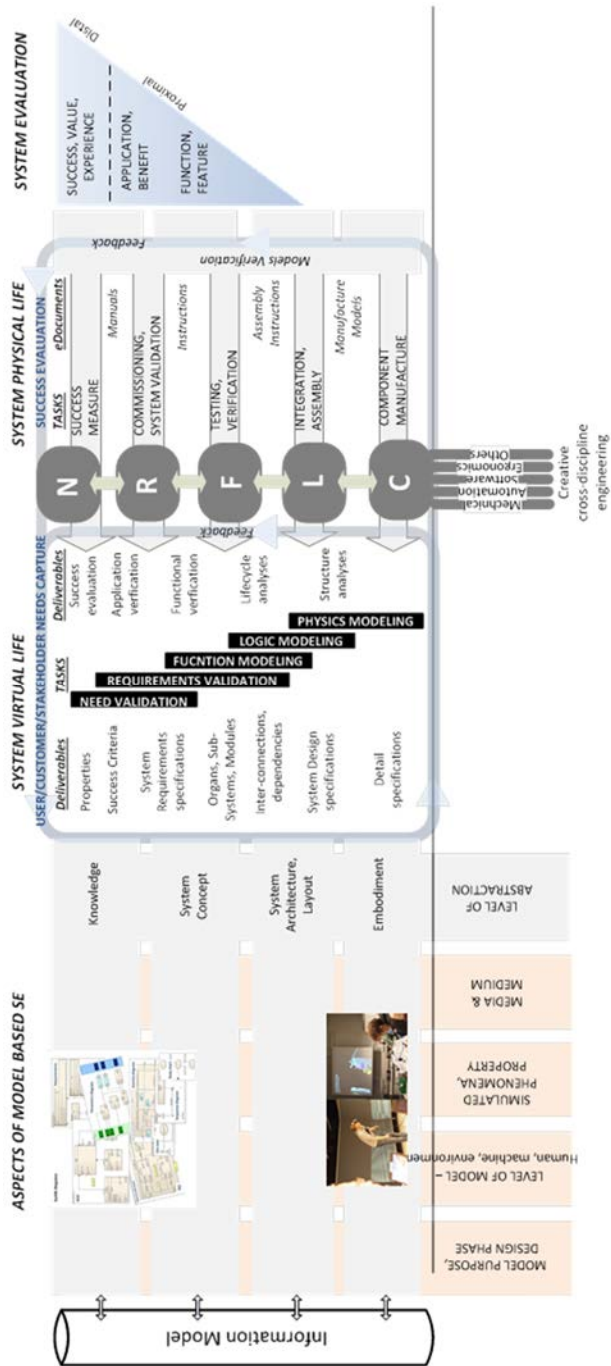
- Tehostaa tuoteprosesseja
- Lyhentää tuotekehityksen läpäisyäikää
- Vähentää fyysisen prototyyppien kustannuksia

- Auttaa rakentamaan laitoksia ja muita yksittäisiä järjestelmiä kerralla oikein eli kustannustehokkaammin
- Mahdollistaa useampien konseptien ja vaihtoehtoisten ratkaisujen vertailun
- Parantaa kommunikointia ja päätöksentekoa
- Mahdollistaa rinnakkaissuunnittelun eri toimintojen ja disipliinien kesken paremmin
- Mahdollistaa tuotteen tai järjestelmän elinkaaren aikaiset tarkastelut suunnitteluvaiheessa
- Auttaa ymmärtämään loppukäyttäjien, asiakkaiden ja muiden sidosryhmien tarpeita, sekä järjestelmän kontekstia paremmin
- Parantaa tuotteen laatua
- Toimii käyttöliittymänä ja ilmaisuvälineenä monimutkaiseen ja abstraktiin tuotetietoon.

Nämä yleiset väitteen hyödyistä ja kustannustehokkuudesta ovat olleet jo pitkään tiedossa, mutta niiden todentamisesta käytännössä on vähän tieteellistä evidenssiä. Tähän on useita syitä. On esimerkiksi erittäin vaikeaa verrata toteutuneita projekteja, koska harvoin on löydettävissä kahta samanlaista projektia joissa virtuaaliprototyyppin vaikutusta toisen projektin onnistumiseen voidaan arvioida. On myös vaikeaa osoittaa yleistettäviä syy-seuraussuhteita aukottomasti. Toisaalta yksittäisiä menestystarinoita voidaan analysoida ja osoittaa tapauksia, joissa virtuaaliprototyyppi on säästänyt merkittävästi aikaa ja rahaa. Näyttääkin siltä, että virtuaaliprototyyppin hyötyjä kannattaa tarkastella kokonaisuutena, strategisena kyvykkyuden parantajana ja arvona liiketoiminnalle.

Virtuaaliprototyyppin referenssimalli

Edellisissä luvuissa on kuvattu mitä tarkoitetaan mallipohjaisuudella, minkälaisia malleja on olemassa eri tarkoituksiin ja miten ne liittyvät liiketoiminnan erilaisiin prosesseihin. Tässä kappaleessa esitellään *VIP-ASE-malli (Virtual Prototyping Aided MBSE)*, joka kokoaa eri aspektit sekä prosessit, menetelmät ja työkalut yhteen (kuva 5).



Kuva 5. VIP-ASE-mallit.

Mallissa on samoja elementtejä, kuin perinteisessä Systems Engineering V-mallissa, mutta tässä on ollut tavoitteena korostaa prosessien iteratiivisuutta ja mallipohjaisuutta. Tarkoituksena on myös korostaa sitä, että mallipohjaisuus mullistaa prosesseja, toimintatapoja, organisaatiota ja infrastruktuuria yrityksissä. Tekemällä asioita vanhalla tavalla uusilla välineillä ei todennäköisesti ole tarkoituksenmukaisinta. Visiona nähdään ympäristö, jossa järjestelmän spesifikaation transformaatiota tukee kokonaisjärjestelmän erilaiset tarkoituksenmukaiset mallit, sekä niitä tukevat tiedonhallintajärjestelmät ja prosessit. Ei ole kuitenkaan realistista pyrkiä kovin yksityiskohtaiseen malliin, koska yritysten tarpeet ovat erilaisia. VIP-ASE onkin tavallaan referenssimalli, joka tuo esille toisaalta mallipohjaisuuden mahdollisuuksia ja kytkentää yritysten prosesseihin, ja toisaalta niitä edellytyksiä joilla hyötyjä voidaan lunastaa. Malli tulee räätälöidä ja täsmentää kunkin yrityksen liiketoiminnan tarpeisiin. Tähän on myös tarjolla metodiikkaa, jota on lyhyesti kuvattu luvussa 6.

Mallipohjaisuuden suhde muihin prosesseihin

Kuten on aikaisemmin todettu, mallipohjaisuus liittyy teknisiin prosesseihin, projekti-prosesseihin, liiketoimintaprosesseihin ja niitä tukeviin menetelmiin, työkaluihin ja tietojärjestelmiin.

Mallipohjaisuuden näkökulmat

Malleja voidaan tarkastella eri näkökulmasta. Toisaalta malleja peilataan teknisen prosessin vaiheisiin, spesifikaatioiden abstraktiotasoihin ja kehitysprojektiin, toisaalta niiden käyttötarkoitukseen, mallin tasoon, kuvattavaan ja simuloitavaan ilmiöön, järjestelmän ominaisuuteen, viestintävälineeseen ja käyttöliittymään. Näistä näkökulmista voidaan muodostaa kuvan vasemman laidan mukaisesti yleinen matriisi. Luonnollisesti matriisi tulee täydentää yksityiskohtaisesti yritysten omista lähtökohdista ja kypsyydestä riippuen.

Järjestelmän virtuaalinen ja fyysinen elinkaari

Virtuaalisuus tarkoittaa tässä yhteydessä järjestelmää tai tuotetta ennen sen valmistusta ja sitä seuraavaa fyysistä elinkaarta. Toisaalta virtuaalinen ja fyysinen elinkaari esiintyvät osittain limittäin. Esimerkiksi järjestelmän komponentteja saatetaan valmistaa jo siinä vaiheessa, kun arkkitehtuuria ja tuotantojärjestelmää muokataan vielä virtuaalisesti. Virtuaalimalleja voidaan myös hyödyntää monin tavoin fyysisen elinkaaren aikana mm. kokoonpano-, huolto- ja käyttöohjeissa. Mallipohjaisuuden tavoitteena on virtaviivaistaa järjestelmän määrittely- ja toteutusvaihetta mahdollistamalla mm. verifiointi- ja validointiprosesseja sekä parempaa päätöksentekoa aikaisemmin. Järjestelmän fyysiseen elinkaareen liittyviä prosesseja voidaan tarkastella, kehittää ja todentaa osana virtuaalista elinkaarta. Erimerkkeinä tästä ovat mm. järjestelmän valmistettavuuden, kokoonpantavuuden, logistiikan, operoinnin, huollon ja kierrätyksen prosessit. Fyysinen elinkaari tuottaa myös taikaisinkytkentää virtuaaliseen vaiheeseen. Malleja voidaan todentaa reaali maailman

mallien ja prosessien perusteella. Fyysisestä elinkaaresta tulee myös palautetta, joka toimii syötteenä uusien kehitystarpeiden, vaatimusten ja prosessien kehittämiseen.

Tiedonhallintajärjestelmien tuki

Mallipohjaisuuden ja virtuaaliprototyypoinnin laajamittainen hyödyntäminen vaatii myös yritysten infrastruktuurin kehittämistä. Tässä tuotetiedon hallinta ja informaation hallinta yleisesti ovat keskeisiä tekijöitä. Jos mallipohjaisuudesta tulee keskeinen SE-prosessin sisällön väline, mallit ja data ei voi enää sijaita hajallaan erilaisia IT-järjestelmissä tai pahimmillaan yksittäisten käyttäjien tietokoneiden kovalevyillä. Mallipohjaisuus ja virtuaaliprototyypointi tulisikin nähdä osana PLM:n käyttöönottoa ja kehittämistä. Tällöin on tärkeää ymmärtää PLM integroituna liiketoimintaympäristönä, jonka prosessit, menetelmät, työkalut ja organisaatio ovat hyvin määriteltäviä.

Perinteisesti PDM-järjestelmien eli PLM:n tiedonhallinnan selkärangan vahvuus on liittynyt fyysisen komponenttirakenteen (tuoterakenne) ja nimikkeiden hallintaan. Kustannustehokkuus edellyttää, että tuki löytyy mallien abstraktiotasojen kehitykselle, mallien uudelleenkäytettävyydelle ja jakamiselle jne. Yksi suurimpia haasteita tällä hetkellä onkin hallita moniteknisten mallien prosessointi kuvailevista malleista aina CAE-työkalujen ja simulaattoreiden malleihin saakka. Vastaavasti tuki on usein puutteellista jos halutaan prosessoida malleja ylävirtaan päin, eli esimerkiksi reaaliaikaisista virtuaaliympäristöistä suunnittelutyökaluihin (CAD) päin. Automaatiikkaa pitäisi tarjota vähintään informaation, kuten muutospyyntöjen ja vaatimusten jäljittämisen hallintaan tietojärjestelmissä. Tämä on esimerkki siitä, miten PLM-järjestelmän tulisi pystyä yhdistämään teknisen prosessin ja projektiprosessin näkymät. Tuotekehitysprojektien ja tuotteen elinkaaren näkökulma vaatii myös kyvykkyyttä ns. rinnakkaisten tuoterakenteiden hallintaan, eli mahdollisuutta tarkastella komponenttirakenteita esimerkiksi suunnittelun, kokoonpanon ja huollon näkökulmasta säilyttämällä relaatiot tuotemallin objektien välillä. Useat PDM/PLM-toimittajat ovat jo tuoneet markkinoille tuotteita, joissa tukea em. asioille löytyy. Käytännön suuren mittakaavan implementaatioita on kuitenkin vielä vähän.

Järjestelmän evaluointi

Tuotteen tai järjestelmän evaluointi tehdään usein teknisen prosessin, eli teknisten ominaisuuksien testauksen, verifiointin ja validoinnin näkökulmasta (ns. proksimaalinen evaluointi). Liiketoiminnan ja projektin näkökulmasta evaluointi tulisi kuitenkin ulottaa käyttäjien ja asiakkaiden kokemukseen sekä muiden sidosryhmien lisäarvon näkökulmaan (ns. distaalinen evaluointi). Haasteellista on se että proksimaalinen evaluointi voidaan tehdä suoraan teknisestä järjestelmästä, mutta distaalinen evaluointi vaatii epäsuoraa tarkastelua yleensä pitemmällä ajanjaksolla. Mallipohjaisuus mahdollistaa sekä proksimaalisen että distaalisen evaluoinnin osittain jo aikaisessa järjestelmän kehitysvaiheessa. Tärkeää on kuitenkin se, että evaluointi määritellään integroiduksi osaksi eri SE-prosesseja.

Teknisillä järjestelmillä on tiettyjä niitä määritteleviä ominaisuuksia (properties), kuten koko, massa, muoto, valmistettavuus, luotettavuus jne. Osa järjestelmien ominaisuuksista on ilmaistavissa kvantitatiivisesti eli määrällisesti, mutta toiset ominaisuudet ovat määriteltävissä vain kvalitatiivisesti eli laadullisesti. Kvalitatiivinen arviointi riippuu arvioivan henkilön omasta (subjektiivisesta) näkökulmasta. Toisaalta kvalitatiivisia ominaisuuksia voidaan kvantifioida todennäköisyyksiin perustuen. Tällaisia ominaisuuksia ovat mm. luotettavuus ja huollettavuus. Osa ominaisuuksista on arvioitavissa vain subjektiivisesti, esimerkiksi ulkonäkö. Näistä syistä johtuen on joskus haastavaa valita ne ominaisuudet, joilla halutaan kuvata tai evaluoida teknistä järjestelmää. Termi ominaisuus (property) saattaa tuottaa insinööreille vaikeuksia ymmärtää, koska se kattaa laajasti ymmärrettyä mm. funktion, kyvykkyyden, dimensiot, huollettavuuden, valmistettavuuden jne. Kun tekniselle järjestelmälle määritellään haluttuja ominaisuuksia osana suunnittelu-tehtävää, niitä kutsutaan vaatimuksiksi (requirements).

Ominaisuudet toteutuvat tietyllä tasolla, joka määrittää järjestelmän laadun (quality). Ominaisuudet voidaan luokitella monella eri tavalla, niitä voidaan vertailla, evaluoida ja verifioida. Ominaisuuksia voidaan selvittää mittaamalla, arvioimalla, koe-mallintamalla, laskemalla, vertailemalla, määrittämällä optimiarvot. Ominaisuuksien selvittämisessä voidaan myös hyödyntää epäsuoria mittauksia jos tiedetään tekijöiden väliset riippuvuudet. Ei-mitattavien suureiden arviointi on vaikeaa. Jotta arviointi voidaan suorittaa mahdollisimman objektiivisesti, ominaisuuksien kriteerit pitää määritellä tarkasti. Jos esimerkiksi arvioidaan järjestelmän huollettavuutta, arvion peruste pitää olla todettu, esimerkiksi tietyn komponentin helppo vaihdettavuus jne.

Tilanne on huomattavasti monimutkaisempi silloin kun järjestelmä on vasta abstraktissa konseptivaiheessa. Silloin mallinnustekniikat voivat olla tarkoituksenmukaisia apukeinoja. Erityisesti silloin kun kyseessä on uudenlainen järjestelmä (ottaen huomioon koko elinkaari), mallinnuksesta voi olla huomattavaa hyötyä objektiivisessa arvioinnissa.

Periaatteessa arviointi (evaluointi) voidaan tehdä kolmella eri tavalla, joita kuvaa seuraavat kysymykset:

1. *Todellinen järjestelmä*; miten hyvä järjestelmä on?
2. Ratkaisuehdotus tai prototyyppi; onko järjestelmä tai malli vaatimusten mukainen?
3. Useita vaihtoehtoisia ratkaisuja; mikä niistä on paras tai optimaalisin ratkaisu?

Arviointi voidaan tehdä kahdella perustavalla – *subjektiivisesti* tunteeseen perustuen tai *objektiivisesti* tiettyihin kriteereihin perustuen. Subjektiivinen arviointi on hyödyllinen silloin kun se perustuu pitkään kokemukseen ja systemaattiseen toimintatapaan. Molempiin arviointitapoihin tulisi sisältyä: Syntetisoitujen ominaisuuksien valinta, arviointikriteerien valinta, kriteerien tarkempi määrittely ja arviointi, vertaileminen. Arvioinnin tavoitteena on päätöksien tekeminen. Kaikki arvioinnit pitäisi erikseen todentaa (verifioida) objektiivisesti. Tällä varmistetaan että virheitä

ei ole tehty ja että päätöksien perusteista ollaan samaa mieltä. Tähän voi liittyä esimerkiksi *suunnittelukatselmuksset ja auditoinnit päätöksenteon tueksi*.

Teknisen järjestelmän arvo toteuttaa jonkin tahon tarpeen tai tuottaa muuta lisä-arvoa. Arvot voidaan ryhmitellä seuraavanlaisiksi ominaisuuksiksi:

- Tekninen arvo (funktiot, valmistettavuus jne.)
- Taloudellinen arvo
- Ergonominen ja esteettinen arvo
- Käyttöön liittyvä arvo
- Kokonaisarvo.

Järjestelmän toteuttavia arvoja voidaan verrata esimerkiksi kertoimen avulla absoluuttiseen ideaaliseen ratkaisuun. Arvostus on kombinaatio niistä järjestelmän ominaisuuksista, jotka vaikuttavat ihmisten haluun omistaa se. Se viittaa järjestelmän käyttöön, ergonomiaan, estetiikkaan, yhteiskunnallisiin ja taloudellisiin ominaisuuksiin. Arvostus heijastaa usein subjektiivisia arviointeja.

Mallinnuksen ja simuloinnin näkökulmat

Mallipohjaisuutta voidaan tarkastella myös mallien sisällön, menetelmien ja työkalujen näkökulmasta

Mallien käyttötarkoitus ja tasot

Mallien käyttötarkoitus liittyy liiketoiminta-, projekti- ja teknisiin prosesseihin. Liiketoimintaan liittyen malleilla voidaan etsiä ja testata uusia ansaintamahdollisuuksia sekä parannuksia nykyisiin tuotteisiin ja prosesseihin. Projektiprosessien tasolla mallit liittyvät usein päätöksentekoon projektin eri vaiheissa. Teknisissä prosesseissa malleilla ja simuloineilla voidaan tarkastella lukuisia eri ilmiöitä ja ominaisuuksia.

Yleisesti mallien käyttötarkoitus voi olla kuvaileva tai selittävä, käyttäytymistä ennustava, tutkiva, ohjeistava tai suunnitteluun käytettävä. Mallin taso liittyy järjestelmän kuvauksen abstraktion ja täydellisyyden tasoon ja itse mallin sisällön ja kontekstin abstraktion, käsitteellisyden ja spesifisyyden tasoon. Tarvittava mallin taso liittyy edellä kuvattuihin käyttötarkoituksiin. Mallit voidaan yleisesti jakaa kuvaileviin (deskriptiivisiin) malleihin, yleisiin simulointimalleihin ja displiinimalleihin (CAE, VP yms.).

Simuloitavat ilmiöt ja ominaisuudet

Mallipohjaisesti voidaan tarkastella ja simuloida lähes kaikkia sosioteknisen järjestelmän ilmiöitä ja ominaisuuksia jo järjestelmän virtuaalisessa vaiheessa. Itse asiassa simuloimalla voidaan saada näkymä jopa sellaisiin asioihin, joita ei todellisesta järjestelmästä pystytä ainakaan kustannustehokkaasti tarkastelemaan. Simuloimalla voidaan esimerkiksi laskea koneen mekaanisia jännityksiä kohdasta, jonka anturointi on vaikeaa tai mahdotonta. Simuloimalla voidaan myös testata

skenaarioita, jotka todellisessa järjestelmällä olisivat liian kalliita tai vaarallisia. Tärkeimpiä simuloitavia ilmiöitä ja ominaisuuksia ovat järjestelmän toiminta, rakenne ja muoto. Niitä ei tosin tarvita aina kaikkia samanaikaisesti. Esimerkiksi lujustarkasteluissa voi riittää numeerinen data, mutta uuden automallin esteettisyyden tarkastelussa muodon visualisointi on tärkeää. Yleensä muodon (geometrian) yhdistäminen numeeriseen simulointiin tuo kuitenkin lisäarvoa.

Olennaista mallipohjaisuuden laajassa hyödyntämisessä on järjestelmän elinkaaren aikana esiintyvien simuloitavien ilmiöiden ja ominaisuuksien järjestelmällinen määrittäminen osana SE-prosesseja. Tämäkin riippuu hyvin paljon kehiteltävien järjestelmien ja tuotteiden sekä liiketoiminnan luonteesta. Ei ole kovin tehokasta pohtia tapauskohtaisesti mitä ilmiöitä tai ominaisuuksia esimerkiksi järjestelmän verifiointi-katselmuksessa halutaan simuloida.

Mallien viestintäväline ja käyttöliittymä

Malleihin liittyen pitäisi tehdä ero sisällön (content) ja viestintävälineen (media) välillä. Sisältö on tuotetietoa eli järjestelmän määrittelyä kokonaisvaltaisesti ajateltuna. Viestintäväline on tavallaan vuorovaikutteinen käyttöliittymä tuotetietoon. Esimerkiksi uuden tuotteen arkkitehtuuria voidaan suunnitella ja arvioida eri sidosryhmien toimesta konseptivaiheessa virtuaaliympäristössä. Tällöin virtuaaliympäristöteknologia tarjoaa havainnollisen (visuaalinen, luonnollinen) käyttöliittymän abstraktiin tuotetietoon.

Mallinnuskielet UML ja SysML

Unified Modeling Language (UML)

Unified Modeling Language eli UML on graafinen kuvauskieli ja se on alun perin kehitetty oliopohjaisen ohjelmistotuotannon tehostamiseen. Se on kuitenkin yleinen kuvauskieli ja sitä käytetään myös muiden kuin ohjelmistopohjaisten järjestelmien suunnittelemiseen. Luotava kohde kuvataan korkealla abstraktiotasolla, johon kuuluu oliot ja niiden väliset relaatiot. Luotavasta kohteesta tehdään UML-työkalun avulla malli, josta UML-ohjelmisto luo haluttuja standardin mukaisia graafisia kaavioita, luokkakuvauksia, generoi valmista tai puolivalmista koodia riippuen siitä kuinka pitkälle mallia on kuvattu ja mitä ominaisuuksia UML-työkalu tukee.

UML on standardoitu jo vuonna 1997. Standardia (ISO 13485) on päivitetty useamman kerran ja siitä vastaa nykyään Object Management Group (OMG) (<http://www.omg.org/>). Se on kansainvälinen, avoin ja ei-kaupallinen konsortio ja sitä kautta UML-dokumentaatio on vapaasti saatavilla. [UML]

OMG:n määritelmä UML:lle:

“The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system’s blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components.”

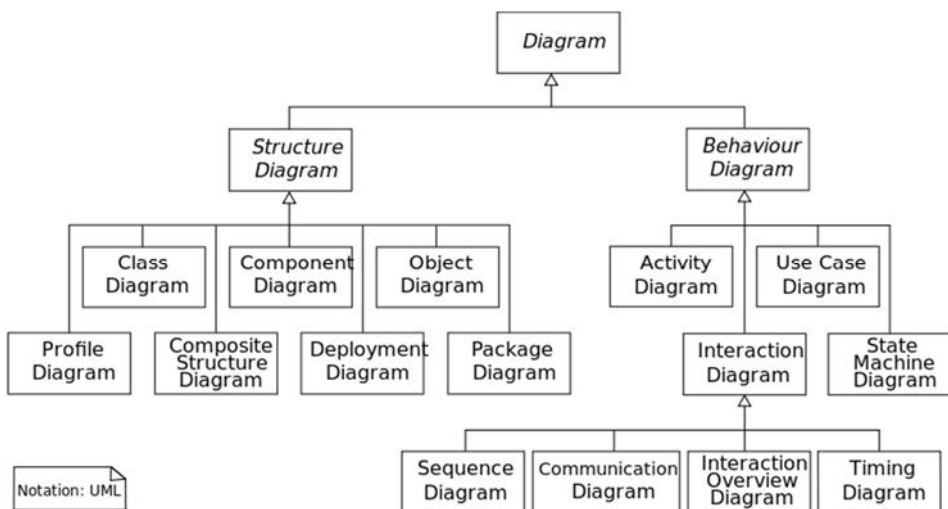
Vapaasti käännettynä:

”Unified Modeling Language (UML) on graafinen kieli ohjelmistointensiivisten järjestelmien visualisointiin, määrittelyyn, konstruointiin ja dokumentointiin. UML tarjoaa standardin tavan kirjoittaa järjestelmän suunnitelmia, mukaan lukien käsitteellisiä asioita kuten liiketoimintaprosessit ja järjestelmätoiminnot, sekä konkreettiset asiat kuten ohjelmointikielen lauseet, tietokantakuvaukset ja uudelleen käytettävät ohjelmisto-osat.”

Merkittävä päivitys tapahtui vuonna 2005, jolloin myös UML:n murre SysML hyväksyttiin standardiksi. Uusin UML-versio on 2.4.1 ja se on vuodelta 2009. [UML]

UML:n rakenne on esitetty kuvassa 6. Siinä on 14 diagrammityyppiä jaettuna kahteen kategoriaan: rakenne ja käyttäytyminen. Taulukossa 1 on listattu UML:n diagrammit.

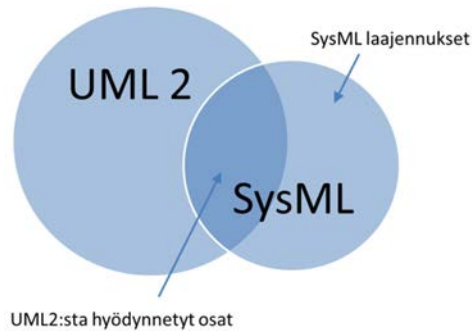
UML-työkaluohjelmistoja on runsaasti saatavilla. Mukana on sekä kaupallisia että maksuttomia. Wikipediassa on laajahko lista UML-ohjelmistoista (http://en.wikipedia.org/wiki/List_of_UML_tools).



Kuva 6. UML:n rakenne.

Systems Modeling Language (SysML)

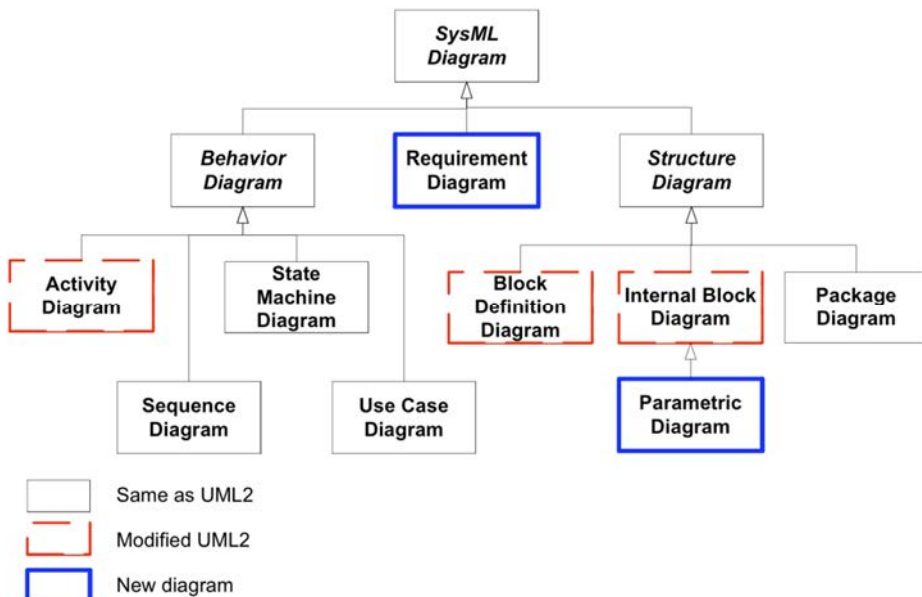
The Systems Modeling Language (SysML) on yleiskäyttöinen graafinen mallinnuskieli Systems Engineering -sovelluksille. Se tukee laajojen järjestelmien ja systems of systems:ien määrittelyä, analyysiä, suunnittelua, verifointia ja validointia. SysML oli alkujaan kehitetty avoimen lähdekoodin määrittelyprojektina ja sisältää siten avoimen lähdekoodin lisenssin. SysML on määritelty UML:n osajoukon laajenuksena. Kieleen on lisätty SE:n tarvitsemat osat ja siitä on poistettu ohjelmistospesifiset osat (kuva 7).



Kuva 7. UML ja SysML. SysML on UML2:sta räätälöity mallinnuskieli Systems Engineering -alueelle.

SysML on The International Council on Systems Engineering (INCOSE) aloitteesta ja pitkälti INCOSE:n jäsenten määrittelemä mallinnuskieli, joka siirrettiin OMG-organisaatiolle hyväksyttäväksi ja jatkokehittettäväksi. INCOSE on ei-kaupallinen organisaatio ja on omistautunut edistämään Systems Engineering -asiaa. Vuonna 2007 OMG hyväksyi SysML 1.0 version. SysML-kieltä kehittää OMG ja viimeisin SysML-versio v1.3 on vuodelta 2012. [OMG SysML]

SysML:n rakenne on kuvassa 8. Systems Engineering -sovelluksia varten UML:stä on tehty oma "murre". Siinä Block Definition Diagram on modifioitu UML:n Class Diagramista ja Internal Block Diagram on vastaavasti UML:n Component Diagramista. SysML:ään on määritelty uudet diagrammit: Requirement Diagram ja Parametric Diagram.



Kuva 8. SysML:n rakenne verrattuna UML-rakenteeseen.

Menetelmiä prosessien, menetelmien ja tietomallien kehittämiseen

Systems Engineeringistä, MBSE:stä ja virtuaaliprototyypoinnista, kuten myös tuotekehityksestä ja muista vastaavista prosesseista, on olemassa useita yleisiä referenssimalleja ja standardeja. Ne pitää kuitenkin räätälöidä ja sovittaa kunkin yrityksen ja arvoverkoston luonteeseen. Näitä luonteenpiirteitä ovat esimerkiksi liiketoimintamallit, tuotteen tyyppi ja sen monimutkaisuuden aste, tuotantomoodi, yms. Prosessien räätälöintiin ja kehittämiseen on olemassa menetelmiä, kuten simulaatiopelit, prosessimallinnus, vaikuttavuusanalyysit ja kypsyyssmallit.

Kypsyyssmallit (maturity model) ovat hyviä lähtökohtia nykytilanteen arvioinnissa ja kehityskohteiden tunnistamisessa. Kirjallisuudesta löytyy monia eri tarkoituksiin kehitettyjä malleja, jotka painottavat eri tavoin esimerkiksi teknologiaa, prosesseja ja organisaatiota. PLM:n implementointi on tyypillinen kohde, jossa hyödynnetään kypsyyssmalleja. Mallipohjaisuuden ja virtuaaliprototyypoinnin kypsyyttä voidaan arvioida mm. seuraavista näkökulmista:

- Vaikutus liiketoimintaan
- Tuoteprosessit ja elinkaari
- Virtuaaliprototyypoinnin ja mallipohjaisuuden prosessit
- Virtuaaliprototyypoinnin ja mallipohjaisuuden teknologiat
- Henkilöresurssit
- Yritysinfrastruktuuri
- Yrityskulttuuri ja organisaatio.

Vaikuttavuusanalyyseinä voidaan tarkemmin tutkia mitä positiivisia (hyötyjä) ja negatiivisia (esim. kustannuksia) vaikutuksia edellä mainittujen näkökulmien kehittämisellä on. Vaikuttavuutta voidaan analysoida esimerkiksi arvoketjun, verkoston, yrityksen toimintojen, disipliinien tai tuotteen elinkaaren näkökulmasta. Esimerkki vaikuttavuusanalyysistä on VTT:n kehittämä PIA – PLM Impact Analysis [Leino et al. 2012]. Vaikuttavuusanalyyseissä ja yleensä prosessien, menetelmien yms. kehittämisessä voidaan hyödyntää simulointipelejä ja prosessimallinnusta. Tyypillisesti niihin osallistuu sidosryhmiä, jotka tuovat omat näkemyksensä kehittämiseen. Näitä menetelmiä on hyödynnetty mm. virtuaaliprototyypoinnin prosessien ja niihin liittyvien tiedonhallinnan tekniikoiden ja tietomallien kehittämiseen.

SE-toimintatavan implementointi verkostoympäristössä

Mikko Uoti, VTT

Iris Karvonen, VTT

Jarmo Alanen, VTT

Sisältö

Systems Engineering -organisointi – roolit.....	86
Kolme tapaa lähestyä Systems Engineeringiä	86
Systems Engineering ja insinöörien roolit	88
Systems Engineering -roolit	89
Systems Engineering -johtamisroolit	91
Systems Engineering verkostoympäristössä	95
Mitä verkostolla tarkoitetaan? Verkostoitumisen haasteet	95
Suunnittelu verkostossa	96
Hierarkkinen suunnittelu verkostossa.....	98
Systems Engineeringin soveltaminen verkostossa.....	99
Verkostovalmiudet – Systems Engineering -valmiudet.....	100
Kommunikaatio osana yhteisen ymmärryksen rakentamista	101
Systems Engineering – toimintatavan implementointi	105
SE-toimintatavan soveltaminen – erilaiset käyttöönottilanteet.....	105
Systems Engineering ja monimutkaiset systeemit.....	108
Lean-periaatteet SE:n soveltamisessa	108
Organisatorinen käyttöönotto ja sen onnistumistekijät.....	109

Systems Engineering -organisointi – roolit

Tässä luvussa esitellään ja arvioidaan erilaisia rooleja – joko pysyviä tehtäviä tai vaihtuvia rooleja – joita Systems Engineering -prosessissa toimivilla ihmisillä on. Roolit ovat merkittävän riippuvaisia SE-sovellustavasta, joka yrityksessä on valittu – vaikuttaen rooleihin, vastuunjakoon ja tehtäviin.

Kolme tapaa lähestyä Systems Engineeringiä

Yritykset ovat erilaisia ja ne toimivat erilaisilla toimialoilla ja hyvinkin erilaisin toimintatavoin. Tällä on myös suora vaikutus siihen kuinka Systems Engineeringiä tulisi eri tapauksissa lähestyä ja kuinka se kannattaisi eri organisaatioissa toteuttaa. Joissakin tapauksissa SE nähdään enemmän kaikkia yhdistävänä toimintona, laatuprosessia vastaavana prosessina, ja joissakin tapauksissa taas hyvin teknisenä (tai jopa tieteellisenä) suunnitteluprosessin johtamismenetelmänä.

Systems Engineering -toiminta voidaan jakaa karkeasti kolmeen eri luokkaan, jotka eivät kuitenkaan ole toisensa poissulkevia. Eri luokat voivat esiintyä projektin eri vaiheissa ja ne ovat myös osittain päällekkäisiä. Suurin ero **SE-innovaatiojärjestelmän** ja **toimintatavan** välillä on se, että ensimmäinen on analyyttisempi ja keskittyy enemmän ongelmakenttään ja jälkimmäinen taas enemmän ratkaisujen löytämiseen. **SE-toimintatapa** ja **-lähestymistapa** ovat puolestaan molemmat järjestelmien luomiseen keskittyneitä, ero on lähinnä projektien laajuudessa ja johtamisessa.

Erilaisissa käytännön Systems Engineering -sovelluksissa ihmiset toimivat erilaisissa rooleissa. Roolien tulisikin olla linjassa sen suhteen, missä luokassa yritys useimmiten tai pääasiassa toimii.

Taulukko 1. Kolme tapaa soveltaa Systems Engineering -ajattelua.

	Innovaatiojärjestelmä	Toimintatapa	Lähestymistapa
Voidaan nähdä myös	Asiantuntija-SE Käsitteiden tutkiminen ja kehittäminen "SYSTEMS engineering" "The engineering discipline applied to the science of complexity" [Senglaub 1996] Disipliini	Yleis-SE Projektin johtamisen tekninen puoli "Systems ENGINEERING" "Reduction to practice" [Dahlberg 1999] Concurrent engineering ("saman-aikais- tai rinnakkais suunnittelu")	SE-prosessi Maalaisjärjen käyttöä Ongelmien ratkaisu tieteellisesti menetelmällä Kuinka jokaisen insinöörin tulisi toimia
Päämäärä	Ongelman ratkaisumahdollisuuksien todentaminen ja toteuttamiskelpoisen ratkaisun löytäminen. Monimutkainen haaste ongelma- kentässä .	Asiakastarpeeseen keskittyminen, eri vaihtoehtojen tunnistaminen ja kustannustehokkaiden ratkaisujen löytäminen, pysyen annetussa aikataulussa ja budjetissa. Haaste ratkaisukentässä ja mahdollisesti organisaatiossa.	Ratkaisuun kiirehtimisen välttäminen, toiminnot ennen kohteita, huomion pitäminen todellisissa asiakasvaatimuksissa, integraation varmistaminen. Haaste sovellusten varioinnissa ja tuotantojärjestelmissä.
Ajoitus	Voi olla ensimmäinen vaihe suuren järjestelmän kehittämisessä	Voi seurata <i>innovaatioprosessia</i> suuren järjestelmän kehittämisessä	"Kehdosta hautaan"
Pitää sisällään	Systeemianalyysiä ja mallintamista, tehtäväänalyysiä, uhka-analyysiä – täysin uusia ratkaisuja aiemmin tuntemattomiin ongelmiin	Hyödynnetään olemassa olevia komponentteja uusilla ennen käyttämättömillä tavoilla luoden uusia ratkaisuja jo tiedettyyn tarpeeseen. Uusia ratkaisuja jo tunnetuihin ongelmiin.	Tapa tai prosessi kaikenlaisen insinöörityön tekemiseen. Ennalta tunnetut ongelmat ja ratkaisut, joissa on hyödynnetty uusia innovaatioita joissakin kohdin.
Esimerkkejä	Toisen maailmansodan jälkeiset suuret projektit kuten Atlas, SAGE, lennonjohtojärjestelmä, avaruuslennot, sodankäyntivälineet	Satelliittien, lentokoneiden, avioiikkajärjestelmien, ohjausjärjestelmien ja suurten informaatijärjestelmien toteuttaminen.	Hissit, kuluttajatuotteet, ohjelmistot, alijärjestelmät ja komponentit, sovellukset insinööritieteiden ulkopuolelta
Asiakkaat	Yleisesti valtiohallinto ja armeija	Valtiohallinto tai toisinaan kaupalliset yritykset	Yleensä kaupalliset yritysasiakkaat
Arvo	Ratkotaan ongelmia, joita ei muuten pystytä ratkaisemaan. Luodaan innovaatioita ja vallankumouksellisia lähestymistapoja.	Poistaa ongelmia toimitusketjun rajapinnoilla ja parantaa asiakastytyvääisyyttä. Vähentää ongelmia testauksessa ja siitä seuraavaa lisätöitä. Auttaa työn organisoinnissa.	Kuten SE-toimintatavassa , mutta pienemmässä mittakaavassa.
Pääroolit	System Analyst, Requirements Owner, Information Manager, Technical Manager	System Designer / Architect, Coordinator, Technical Manager, Customer Interface, Glue, Validation and Verification, Requirements Owner	Requirements Owner, System Designer, Validation and Verification, Logistics and Operations, also Customer Interface and Technical Manager

Systems Engineering ja insinöörien roolit

Miltä näyttää Systems Engineer?

SE-insinööriä on joskus kuvattu orkesteriaan johtavaksi *maestroksi*, joka tietää miltä musiikin tulisi kuulostaa ("ulkonäkö ja toiminnallisuus") ja jolla on taito ohjata orkesteriaan ("tiimiään") halutun kuuluisen lopputuloksen saamiseksi ("vaatimusten saavuttaminen"). SE-insinöörillä on myös oltava seuraavanlaisia ominaisuuksia:

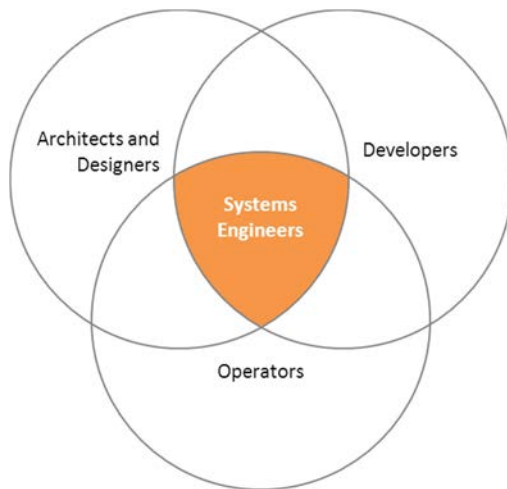
- Matematiikan, fysiikan ja muiden olennaisten tieteiden perusteiden sekä ihmisten ja eri disipliinien voimavarojen ymmärtäminen
- Vähintään yhden disipliinin täydellinen hallinta, osaamista useiden disipliinien osalta
- Täydellinen ymmärrys "pöytäkirjan" päämäärästä ja tavoitteista
- Vision ja lähestymistavan luominen päämäärien saavuttamiseksi
- Mahdollisesti arkkitehti tai suunnittelija
- Monitieteellisten tiimien pohdittavaksi asetettavien teknisten asioiden valinta ja hahmottelu
- Kyky tulkita ja kommunikoida tavoitteita, vaatimuksia, järjestelmäarkkitehtuureja ja suunnitelmia
- Vastuu suunnitelman teknisestä eheydestä
- Monitieteellisten tiimien organisointi ja johtaminen
- Vastuu monimutkaisen tuotteen tai palvelun menestyksekkästä toimittamisesta asiakkaalle.

Systems Engineering – laajuus ja päämäärä

Eräässä näkökulmassa Systems Engineering nähdään *holistisena* ja *yhdistävänä* toimintana. Se toimii tasapainottavana tekijänä rakennesuunnittelun, mekaanisen suunnittelun, sähkö-, ohjelmisto-, turvallisuus- ja energiasuunnittelun sekä monien muiden disipliinien välillä johdonmukaisen kokonaisuuden luomiseksi. SE:ssä on enemmän kyse kompromissien löytämisestä ja "yleismiehinä" toimimisesta kuin erikoisasiantuntemuksesta. Systems Engineeringissä on kyse ennen kaikkea *oikean suunnitelman löytämisestä ja määrittämisestä* ja vasta tämän jälkeen teknisen yhtenäisyyden ylläpitämisestä ja kompleksisen prosessin johtamisesta oikein prosessein. Tällöin SE voidaan jakaa kahteen eri ryhmään:

- **Tekninen johtajuus** – Systems Engineering -osaaminen – keskittyen järjestelmän tekniseen suunnitteluun ja eheyteen koko elinkaaren yli
- **Systemien johtaminen** – Systems Engineering -tiede – keskittyen useiden teknisten disipliinien, organisaatioiden ja ihmisten muodostaman erittäin teknisen ja monitahoisen toiminnan johtamiseen.

Organisaatioiden on hallittava molemmat yllä mainitut ryhmät. Vain toiseen keskittymällä tuotetaan vain joko tuotteita, jotka eivät vastaa tavoitteitaan (systemien johtaminen) tai tuotteita jotka ovat toimimattomia tai liian kalliita (tekninen johtajuus). Kokonaisvaltaista lähestymistapaa on kuvattu kuvassa 1. Hyvällä Systems Engineer -insinöörillä on hyvä yhdistelmä tietotaitoa ja kokemusta suunnittelusta, kehittämisestä ja arkipäiväisestä toiminnasta.



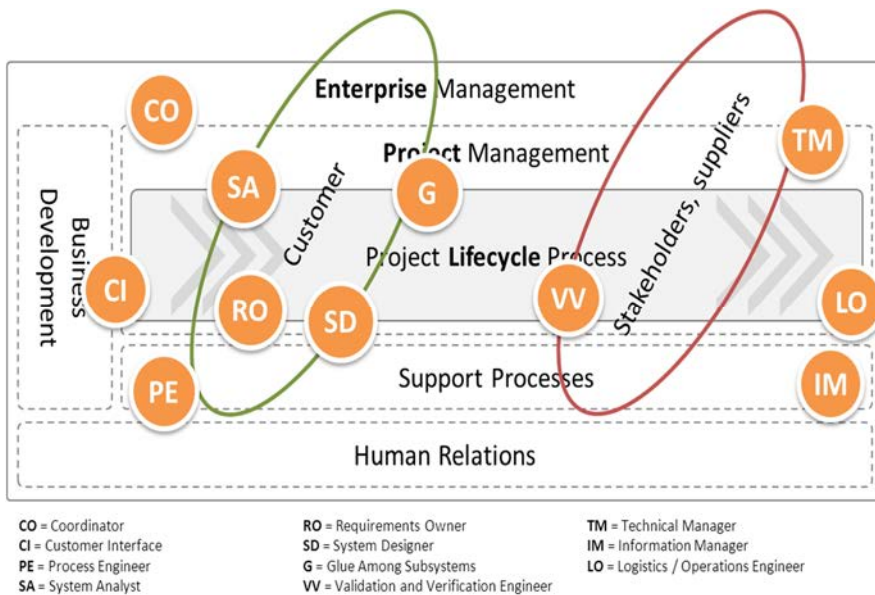
Kuva 1. Systems Engineering -lähestymistapa.

Systems Engineering -roolit

SE-rooleja on yleisesti tutkittu enemmän informaatioteknologian sovelluksissa. Kyseisellä alalla on tunnistettu useita tuotekehityksen elinkaaren alle eri ajankohtiin sijoittuvia tehtäviä eli rooleja. Näistä yksitoista on tunnistettu yleisimmiksi ja eniten päivittäiseen toimintaan vaikuttaviksi. Nämä roolit on esitetty alla kuvassa 2 ja lueteltu taulukossa 2.

Kuvassa roolit on elinkaaren lisäksi jaettu myös tuotekehityksen, sen johtamisen ja tukitoimintojen näkökulmasta eri ulottuvuuksiin. Myös liittyvyys asiakkaaseen ja muihin sidosryhmiin (kuten toimittajat) on esitetty. Kuvasta voidaan nähdä, että asiakkaaseen liittyvät roolit korostuvat projektin alkuvaiheissa (mm. vaatimusmäärittely) kun taas muut sidosryhmät ovat usein enemmän esillä vasta myöhemmissä elinkaaren vaiheissa.

Edellä esitetyt roolit voidaan jakaa myös elinkaari- ja projektin johtamisrooleiksi, sekä rooleiksi, jotka liittyvät riskien ja laadun seuraamiseen ja johtamiseen. Tämä jaottelu kertoo roolien erilaisesta luonteesta ja toimintaympäristöstä. Roolit ovat luonteeltaan myös sellaisia, että ne on harvemmin täysin sidottu tiettyyn organisaatioon tehtävään, rooleissa ei toisaalta myöskään yleensä toimita samanaikaisesti ja moni insinööreistä ei koko uransa toimi jokaisessa roolissa.



Kuva 2. SE-roolit ja organisaation prosessit.

Taulukko 2. SE-insinöörin erilaiset roolit.

Lyh.	Roolin nimi	Kuvaus
RO	Requirements Owner	Järjestelmävaatimukset asiakkaan tarpeista, yhdistää vaatimusmäärittelyyn liittyvät tehtävät toisiinsa.
SD	System Designer / Architect	Luo korkeantason järjestelmä-määrittelyn ja -suunnitelman sekä valitsee pääkomponentit.
SA	System Analyst	Vastaa siitä, että suunniteltu järjestelmä vastaa vaatimuksia.
VV	Validation / Verification Engineer	Suunnittelee ja toteuttaa verifiointijärjestelmän, jolla taataan, että suunniteltu ja toteutettu järjestelmä vastaa vaatimuksia.
LO	Logistics / Operations Engineer	Vastaa suunnittelun elinkaaren loppuvaiheesta.
G	Glue Among Subsystems	Proaktiivinen ongelmanratkaisija, joka pyrkii tunnistamaan ja ratkaisemaan ongelmia projektin aikana.
CI	Customer Interface	Vastaa asiakkaan näkökulmasta ja siitä, että se ohjaa toimintaa koko projektin ajan.
TM	Technical Manager	Sisältää kustannusten seuraamista, aikataulutusta, resurssien jakoa ja tukitoimintojen (tietojärjestelmät, talous ym.) ylläpitoa.
IM	Information Manager	Konfiguraation hallinta, tiedonhallinta ja mittarit.
PE	Process Engineer	Dokumentoi, seuraa, omistaa ja parantaa projektin ja organisaation SE-prosesseja.
CO	Coordinator	Koordinoi eri sidosryhmiä ja ratkoo systeemiä sekä eri osapuolien välisiä haasteita, vähintään yhteisen konsensuksen saavuttamiseksi tai suositusten tekemiseksi.

Elinkaariroolit

Requirements Owner, System Designer / Architect, System Analyst, Validation / Verification Engineer, and Logistics / Operations Engineer

Nämä roolit esiintyvät teknisillä fokusalueilla ja ovat aikataulullisesti normaalisti siinä järjestyksessä kuin edellä on lueteltu, *System Analyst* -rooli tosin esiintyy toistuvasti elinkaaren aikana. Nämä roolit pitävät sisällään tehtäviä, joiden pitää tapahtua järjestelmän elinkaaren aikana.

Projektin johtamisroolit

Technical Manager, Glue, Information Manager, Coordinator, and Customer Interface.

Nämä roolit esiintyvät johtamiseen keskittyvillä fokusalueilla projektissa.

Muut roolit

Edellä mainittujen pääryhmien lisäksi on tunnistettu myös muita rooleja, joissa SE-projekteissa toimitaan. Näitä on mm. riskienhallinta (G, SA, TM), seuranta ja erilaisen suunnittelun tarkistuskierrokset (TM, CI, G) ja laadunvarmistus (PE, TM).

Usein asiakasrajapinta nähdään kriittisenä pisteenä SE:n menestyksekkäälle soveltamiselle. Puhutaan kuilusta asiakkaan ja insinöörien välillä – projektin osapuolilla on usein toistensa kanssa ristiriitaisia tavoitteita, joka korostuu erityisesti suurissa hankkeissa, erityisesti julkisissa hankinnoissa. Tällaisiin ongelmatilanteisiin on ehdotettu kahdenlaista roolia tai tehtävää.

Asiakkaan tekninen edustaja

- Työskentelee asiakkaalle, fyysisesti toimittajan tiloissa
- Varmistaa, että spesifikaatiota noudatetaan, seuraa projektin etenemistä, vastaa teknisiin kysymyksiin.

Asiakkaan SE insinööri

- Asiakkaan tai kolmannen osapuolen palkkalistoilla
- ”Silta asiakkaan ja toimittajan välillä.” Auttaa SE-prosessin ensivaiheissa, toimittajasta riippumaton
- Johtaa parempaan vaatimusmäärittelyyn.

Systems Engineering -johtamisroolit

Systems Engineering -rooleja käsiteltäessä keskitytään yleisemmin lähinnä insinöörien työhön liittyviin tehtäviin ja myös hyvin tärkeät johtamisroolit jäävät pienemmälle huomiolle. Tässä luvussa on tarkoitus esitellä ja arvioida rooleja tai niiden yhdistelmiä, jotka ovat tärkeitä SE-prosessin johtamisessa.

Chief Engineer

Ajatuksena *Chief Engineer* juontuu Toyotan johtamismalleista, laatujohtamisesta ja lean-ajattelusta. Toyotalla Chief Engineer on ylin päätöksentekijä tuotannossa: henkilö, jolla on pitkä monialainen ura yrityksessä ja valtaa sekä arvostusta tarvittaessa jopa pysäyttää koko tuotanto laatupoikkeamien korjaamiseksi. Tällainen toimintamalli on erityisen yleistä tuotantoyrityksissä, jotka soveltavat lean-ajattelua tuotannossaan.

Samaa konseptia voidaan soveltaa myös SE-prosessin johtamiseen, jossa myös lopulta on kyse riittävän laadun ylläpidosta ja laatujohtamisesta. SE-kontekstissa Chief Engineer on

- "Halutuin tehtävä koko yrityksessä", "Menestystuotteet minimibudjetilla ja -aikataululla"
- Kokenut ja tunnustettu edelläkävijä/johtaja
- Revenue Assurance Analyst koko tuotekehitykselle (konseptit, kompromissit, merkittävät suunnittelupäätökset, koordinaatio, tavoitteet, aikataulu ja budjetti).

SE kontekstissa Chief Engineerin tulisi olla monialainen huippuosaaja ja tunnustettu esikuva / johtaja – erityisosaaja SE:ssä, rajapinnoilla ja kompromisseissa sekä ihmisten välisessä kommunikaatiossa, kulttuurissa ja yhteisen konsensuksen rakentamisessa – erittäin haastava tehtävä siis. Chief Engineerillä pitäisi myös olla riittävästi valtaa ja vaikutusmahdollisuuksia kriittisten päätösten tekemiseksi tarvittaessa – "pysäyttää tuotantolinjat tarvittaessa" – sekä lähes pakkomieltäinen kiinnostus tuotteiden tinkimättömyyteen, suunnittelupäämääriin ja hyvään insinööriyöhön.

Chief Engineerin pitäisi myös työskennellä samassa tilassa projektin johdon kanssa, mutta keskittyen lean-päämääriin, karsimaan arvoa tuottamatonta budjetin, hallinnon ja yritysjohtamisen kautta.

Organisaatorakenteet

Tässä esitellyt organisaatorakenteet ovat yleisiä malleja joita esiintyy organisaatioissa myös systems engineeringin ulkopuolella. Rakennevaihtoehtojen esittäminen on kuitenkin hyödyllistä, jotta voidaan arvioida millaista lähestymistapaa erityyppisissä organisaatioissa voidaan SE-prosessien johtamiseen käyttää.

Funktionaalinen

Perinteinen ja hyvin yleinen lähestymistapa on jakaa organisaation toimintojen mukaan. Jokainen toiminnallinen erikoisalue tai disiplinaari määritetään erillisille organisaatioyksiköille, esim. SE-tiimi, joka toteuttaa SE-prosessin jokaisessa yrityksen projektissa. Tämä lähestymistapa toimii parhaiten *pienissä projekteissa*, jolloin ihmiset työskentelevät samalla useissa projekteissa.

Jotta voitaisiin taata, että disipliinkohtaista osaoptimointia tapahtuisi mahdollisimman vähän ja toiminta olisi integroitunutta, tarvitaan jatkuvaa ristikkäistä kommunikointia toimintojen yli sekä systeemisten asioiden läpikäymistä.

Projekti

Tässä lähestymistavassa keskiössä ovat yksittäiset projektit niidet ylittävien toimintojen sijaan. Kaikki projektissa työskentelevät, oli heidän toimintonsa tai tehtävänsä mikä tahansa, vastaavat toiminnastaan projektipäällikölle. Tämä lähestymistapa toimii usein *suurissa ja pitkäkestoisissa projekteissa*, joissa asiantuntijat ja erityisosaajat voivat käyttää koko työajan yhdelle projektille pitkäksi aikaa.

Matriisi

Matriisiorganisaatio on tietynlainen välimuoto kahdesta edellä mainitusta. Projekti-tiimin jäsenet raportoivat sekä projekti- että toimintoihin perustuvalla linjajohdolle. Tämä lähestymistapa on hyödyllinen *suurissa ja pitkäkestoisissa projekteissa*.

Projektitoimisto

Tässä lähestymistavassa projektijohto, systems engineering ja suunnittelutiimit on organisoitu projektien mukaan, jotka pyytävät tarvittaessa apuaan eri toiminnoissa työskenteleviltä asiantuntijoilta. Tämä lähestymistapa toimii kohtuullisen kokoisissa projekteissa, joissa vain avainhenkilöt voivat käyttää kaiken aikansa projektille. Erityisasiantuntijat työskentelevät edelleen useissa projekteissa.

Integroitu tuotetiimi

Tällainen tiimi koostuu sekä asiakkaan, että toimittajan edustajista, jotka työskentelevät yhdessä projektin tarpeita vastaavan järjestelmän kehittämiseksi. Usein molemmissa organisaatioissa on peilikuvat eri toiminnoista. Jokainen disiplineista on myös edustettuna tiimissä, jolla taataan toiminnot ylittävä kommunikaatio. Tällainen lähestymistapa on usein mahdollinen *vain suurissa projekteissa*.

Systems Engineering -johtamisroolit

Taulukkoon 3 on kerätty erilaisia lähestymistapoja Systems engineering -johtamisen toteuttamiseksi.

Taulukko 3. Erilaisia lähestymistapoja Systems Engineering -prosessin johtamiseen.

Rooli	Toiminnot, vastuut	Vaatimukset	Sopivuus
<i>Erillinen SEM & Chief Engineer</i>	SE Manager (SEM): Vastaa SE prosessista (sen olemassaolosta ja implemen- toinnista) koko yrityksen laajuisesti Organisatorinen rooli Chief Engineer: Vastaa, että projekti (suunnitte- lu-) seuraa SE-prosessia ja toimii sen vaatimusten mukaan Projektin/tuotannon johtamis- rooli	Molemmissa rooleissa tarvitaan laajaa monia- laista osaamista, mutta SEM on enemmän SE-fokusoitunut ja Chief Engineer keskittyy enemmän suunnittelu- prosessin johtamiseen Vaatii kaksi erittäin kyvykästä henkilöä.	Engineering-fokus, kaikki projektit eivät välttämättä toimi SE-periaatteiden mukaisesti
<i>Yhdistetty SEM ja Chief Engineer</i>	Vastaa SE-prosessista ja sen käytöstä "Vaatimustenmukaisuuden johtaja" Työskentelee läheisesti projektipäällikköiden kanssa, mutta rooli ei ole projektirooli	Syvä monialainen osaaminen ja tietämys sekä engineering- osaaminen Syvä SE osaaminen Voimakas ja arvostettu henkilö	Erittäin riippuvainen yhdestä henkilöstä
<i>SEM tiimi (jolla vetäjä)</i>	SE-ammattilaisten tiimi, jota vetää SE Manager Jokainen tiimin jäsen voi olla mm. jonkin disiplinaarin erityisosaaja	Yhden henkilön ei tarvitse hallita kaikkea Yksi erityisosaaja ja ryhmä disiplinaari-osaajia ("yleismiehiä")	Suuret organisaatiot tai organisaatiot joissa tehdään valtavaa kokoisia projekteja missä yksi henkilö ei voi olla mukana kaikessa
<i>Jaettu useiden toimijoiden kesken, sisältäen myös vastuut</i>	Systems Engineering on "tapa toimia" koko yrityksessä – kaikilla on riittävästi tietoa SE:stä Ei erityistä SE-johtamisroolia	Jokaisella on vähintään perustiedot SE:stä Jokainen ymmärtää ja "näkee suuren kuvan" Jokaisella on tiedostettu oma rooli SE prosessin toimijana Prosessin ylläpito vaatii jatkovaa seurantaa ja työtä, osaoptimointia esiintyy helposti	Organisaatiot, jotka toimivat täysin SE-toimintamallein alalla, jossa SE-mallit ovat laajalti hyväksytyjä Voi olla helpompaa pienemmille yrityksil- le, joissa kaikki työntekijät ovat yhdessä tilassa

Systems Engineering verkostoympäristössä

Tässä luvussa kuvataan, miten SE-prosessi voisi toimia verkostossa organisaatio-rajojen yli, ja mitä se edellyttää. Mitä ovat verkostovalmiudet? Kommunikaatio ja yhteisen ymmärryksen saavuttamisen merkitys.

Mitä verkostolla tarkoitetaan? Verkostoitumisen haasteet

Yritysverkosto on itsenäisten yritysten yhteenliittymä, joka tekee yhteistyötä pystyäkseen palvelemaan asiakkaitaan. Globaali markkina-alue, monimutkaiset ja monitekniset tuotteet ja niitä tukevat elinkaaripalvelut edellyttävät monialaista osaamista, jota yksi yritys ei kykene yksin hallitsemaan, vaan sen tulee verkostoitua muiden yritysten kanssa.

Yritysten verkostoitumisella on monia tavoitteita ja muotoja. Yritykset pyrkivät saamaan tehokkuutta ja joustavuutta keskittymällä ydinosaamiseensa ja tekemällä yhteistyötä muiden yritysten kanssa. Usein pyritään pitkäkestoiisiin yhteistyösuhteisiin, jolloin yhteistyöprosessit voidaan kehittää toimiviksi ja osaaminen voi kumuloidua helpommin kuin lyhytaikaisissa suhteissa. Perinteisen resurssipohjaisen ja osaamiseen perustuvan verkostoitumisen lisäksi uusiksi verkostoitumisen ajureiksi on tullut lisääntyvässä määrin asiakassuuntautuneisuus ja uudet markkinat, ketteryys ja riskien hallinta sekä kestävä kehityksen tavoitteet (taulukko 4).

Taulukko 4. Verkostoitumisen ajurit.

Verkostoitumisen ajurit:

- *Ydinosaamiseen keskittyminen, resurssien ja kustannusten säästö*
- *Rinnakkaisuus, ajan säästö*
- *Kasvun tukeminen, joustavuus, ketteryys*
- *Osaamisen laajentaminen ja uuden osaamisen saaminen käyttöön*
- *Uudet tuotteet, teknologiat tai palvelut*
- *Markkinoiden laajentaminen uusille alueille*
- *Riskien pienentäminen ja siirto*
- *Kestävän kehityksen tavoitteet*

Yritysverkostojen luonteen ja keston mukaan voidaan tunnistaa kaksi keskeistä verkostomuotoa:

- Pitkäkestoinen verkoston valmiustila, jossa yhteistyössä kehitetään yhteisiä valmiuksia yhteistyöhön, jotta asiakastilauksen tullessa pystyttäisiin reagoimaan nopeasti ja tehokkaasti tuottamaan asiakkaalle arvoa. Yhteistyöllä voi olla erilaisia muotoja ja kiinteyksasteita, alkaen ns. kasvatusympäristöstä (breeding environment) löyhempiin liiketoiminta-ekosysteemeihin ("business ecosystems").
- Tiettyyn, usein aikataulutettuun tavoitteeseen tähtäävä yhteistyö esim. projektiluonteisesti. Tällaista verkostomuotoa kutsutaan joskus "virtuaaliorganisaatioksi". Sen odotetaan toimivan kuten yksi organisaatio, vaikka se muodostuu useista eri organisaatioista. Virtuaaliorganisaatio luodaan tiettyä tehtävää varten ja puretaan, kun tehtävä on saatu suoritetuksi.

Vaikka verkostoituminen voi tuottaa yrityksille paljon hyötyä, verkostotoiminnassa on myös paljon haasteita, jotka liittyvät mm. toimijoiden omiin tavoitteisiin ja päätöksentekoon, erilaisten tavoitteiden yhteensovittamiseen sekä osaamisen hallintaan (taulukko 5).

Taulukko 5. Verkostoitumisen haasteita.

Verkostoitumisen haasteita:

- Itsenäiset yritykset tekevät itse päätöksensä – eivät suoraan ohjattavissa.
- Verkoston jäsenet voivat toimia useassa verkostossa.
- Yksittäisen toimijan etu ei aina ole sama kuin kokonaisuuden etu, ristiriitoja voi esiintyä.
- Resurssit eivät välttämättä ole käytettävissä tarvittaessa.
- Koordinaatiokustannukset lisääntyvät
- Kustannusten säästämiseksi tehtäviä siirretään osapuolille, joilla on osaamistaso ja kyvykkyydet ovat heikkomat.
- Hiljaisen tiedon puuttuminen; kokemustiedon hajautuminen ja menetys
- Laadun ylläpito, toimintakulttuurin erot
- Erilaiset työkalut ja järjestelmät ja niiden yhteensopimattomuus
- Riskiä siirretään partnereille, jotka eivät ole kyvykkäitä sitä hallitsemaan

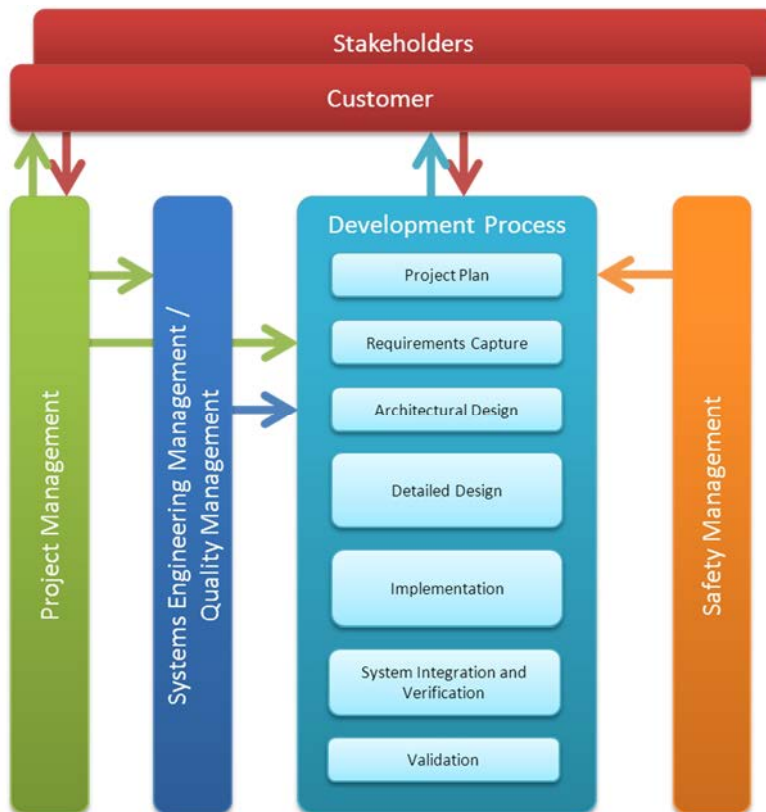
Suunnittelu verkostossa

Suunnitteluprosessiin (kuva 3) liittyy yleensä useita osapuolia. Riippuen tuotetyypistä asiakasvaatimukset voivat tulla joko spesifisinä vaatimuksina (asiakaskohtainen tuote) tai tunnistettuina / määriteltyinä vaatimuksina (tuotekehitys). Asiakaskohtaisen

tuotteen suunnittelussa asiakas on mukana läpi prosessin jollakin tasolla, tuotekehityksessä on erilaisia käytäntöjä.

Koska monimutkaisten tuotteiden suunnittelu vaatii useita eri osaamisaloja, liittyy varsinaiseen suunnitteluunkin usein monia tahoja; mm. alihankintasuunnittelua ulkoisilta osapuolilta. Suunnittelu yhteistyössä suunnittelua voidaan jakaa eri osapuolille mm. suunnitteluvaiheen (konsepti- – yksityiskohtainen suunnittelu), suunnittelualueen (mekaniikka-, sähkö- jne.) tai osajärjestelmien mukaan. Asiakasvastuullinen OEM (original equipment manufacturer) yleensä haluaa pitää itsellään kokonaisuuden hallinnan.

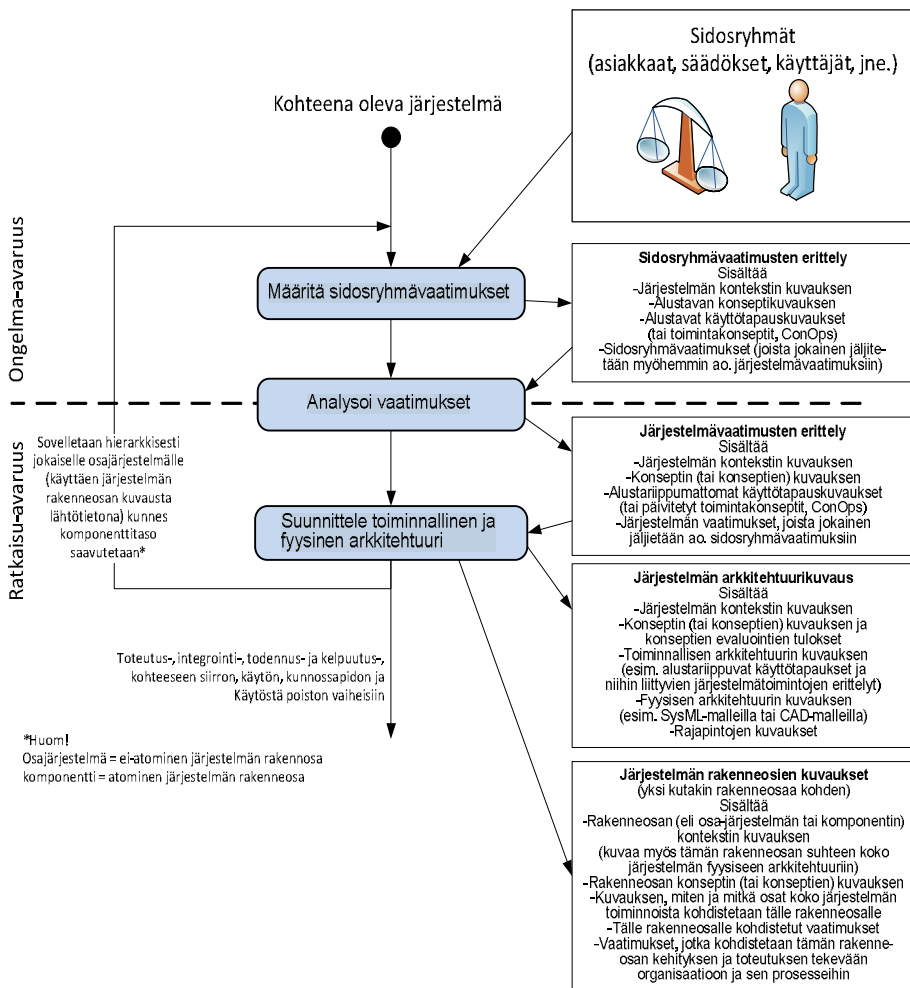
Varsinaisen suunnittelun lisäksi tulevat johtamisprosessit ja niiden toimijat: projektinhallinta, turvallisuusjohtaminen, laatujohtaminen ja Systems Engineering -johtaminen. Kuten edellä on todettu, johtamiseen (mukaan lukien SE-käytännöt) liittyvät roolit voidaan toteuttaa monella tavalla ja usein eri roolit yhdistyvät (esim. laatu- ja SE-). Vastaavasti roolit voivat jakautua verkostossa eri tavoin: myös yhteistyökumppaneille / alihankintakumppaneille voi aiheutua joitakin vastuuta ja roolituksia liittyen SE-prosesseihin; etenkin, jos partneri on vastuussa jostakin kokonaisuudesta/alijärjestelmästä.



Kuva 3. Suunnitteluprosessi.

Hierarkkinen suunnittelu verkostossa

Kohteena oleva järjestelmä suunnitellaan hierarkkisesti ylhäältä alaspäin niin, että jokaisessa haarassa päädytään lopulta atomisiin rakenneosiin (komponentteihin), jotka eivät enää jakaudu osajärjestelmiksi. Osajärjestelmän suunnittelua varten tarvitaan riittävät lähtötiedot pääjärjestelmän suunnittelijalta. Ne saadaan pääasiassa kahdessa eri dokumentissa, järjestelmän arkkitehtuurikuvauksessa ja järjestelmän rakenneosan kuvauksessa (ks. kuva 4). Näistä jälkimmäinen on osajärjestelmäkohtainen, ja se muodostaa pääasiallisen lähteen osajärjestelmän suunnitteluun, mutta myös järjestelmäarkkitehtuurin kuvaus on tärkeä kokonaisuuden ymmärtämiseksi.



Kuva 4. Tuotekehitysvaiheessa siirtyvä informaatio prosessista toiseen ja osajärjestelmien tuotekehitysprosessiin; kuvassa suunnitteluvaihe.

Jos järjestelmäsunnittelija tekee konseptivaiheessa rakenneosista virtuaalisia toteutuksia, niistä tulee rakenneosan suunnittelijalle konseptimalleja, ja ne (tai viite niihin) sisällytetään ko. rakenneosan kuvaukseen, eli virtuaalimallit täydentävät rakenneosan konseptikuvausta.

Vastaavasti kun järjestelmää lähdetään toteuttamaan, integrointi-, todennus-, kohteeseen siirto- ja kelpuutusprosessit, tehdään myös hierarkkisesti, mutta nyt alhaalta ylöspäin. Vastaavasti tällöin siirtyy tietoa, lähinnä integrointi- ja käyttöohjeita, teknisiä kuvauksia sekä todennus- ja kelpuustietoa, osajärjestelmistä pääjärjestelmään päin. Osajärjestelmien todennus- ja kelpuustiedot tarvitaan todisteeksi pääjärjestelmän kelpuutusvaiheessa.

Systems Engineeringin soveltaminen verkostossa

Verkostoympäristössä tuotteen elinkaari prosessit ulottuvat organisaatorajojen yli. Tuotteen suunnittelussa voi olla mukana useita organisatorisesti ja maantieteellisestikin hajautuneita osapuolia, valmistuksessa käytetään kumppaneita ja alihankkijoita ja tuotteen käytön aikaiset palvelut (esim. huolto) hoidetaan verkostolla, jossa on useimmiten mukana myös paikallisia osapuolia. Miten Systems Engineering -prosessit voidaan toteuttaa tällaisessa ympäristössä? Miten vastuut jaetaan ja mitä informaatiota siirretään eri osapuolien välillä?

SE:n soveltamiseen hajautetussa organisaatiossa ei ole olemassa yhtä ratkaisua, vaan toimivin käytäntö riippuu aina tavoitteista ja tilanteesta, siitä, miksi SE:tä halutaan soveltaa, mihin sen soveltamisella pyritään, mitkä ovat eri osapuolien valmiudet ja osaaminen, minkälaista yhteistyökokemusta partnereilla on aiemmin jne. Periaatteessa SE-vastuiden määrittelylle ja jakamiselle voidaan erottaa kaksi ”ääripäävaihtoehtoa”:

1. Verkostoyritys toimii päähankkijana tai päähankkijan kumppanina ja on vastuussa monimutkaisen tuotekokonaisuuden, *systemin*, kehittämisestä, suunnittelusta ja mahdollisesti valmistuksestaakin. Tuotteen vaatimukset voivat tulla annettuina asiakkaalta tai yritys on itse vastuussa niiden tunnistamisesta ja määrittelystä. Yritys vastaa myös ko. kokonaisuuden vaatimusten verifiointista ja validoinnista.
 - Tässä tapauksessa verkostoyritykseltä vaaditaan vahvaa SE-prosessien ja periaatteiden osaamista ja ymmärrystä. Yrityksen tulee määrittellä omat SE-prosessit ja niitä toteuttavat henkilöt, kehittää riittävä osaaminen ja viedä SE-toimintatavat läpi organisaatiossa sekä seurata ja varmistaa prosessien toimivuus. Yhteistyössä asiakkaaseen tai päähankkijan ja partnerin välillä tulee sopia SE-prosessien linkkautumisesta yli organisaatorajojen, mikä toteutuu esim. sopimisella SE:tä tukevien tietojen tai dokumenttien vaihdosta. Edullisinta olisi, jos kumppanit käyttäisivät samoja tai yhteensopivia työkaluja esim. vaatimusten hallintaan, mutta mikäli se ei ole mahdollista, joudutaan usein eri työkalujen tiedot muokkaamaan yhteensopiviksi.

2. Verkostoyritys on vastuussa tuotteen osan tai moduulin suunnittelusta tai valmistuksesta. Vaatimukset tulevat loppuasiakkaalta tai päähankkijalta.

➔ Tässä tapauksessa asiakkaan/päähankkijan kannattaa harkita, onko tarpeen vaatia yritykseltä (alihankkijalta tai toimittajalta) paneutumista SE-käytäntöihin ja omien SE-prosessien ja vastuiden määrittelemistä. Kevein vaihtoehto voi olla, että asiakas/päähankkija antaa selkeän ohjauksen alihankkijalle siitä, millaista tietoa tämän pitää toimittaa ja missä muodossa. Jos alihankkija on vastuussa myös verifiointista ja validoinnista, asiakas/päähankkija voi varmistaa sen, että SE-prosesseja noudatetaan, antamalla esim. tiukat testaus- ja dokumentointiohjeet.

Nämä vaihtoehdot ovat luonnollisesti ääripäitä ja tyypillisesti sopiva ratkaisu löytyy niiden väliltä. Verkostoympäristössä on kuitenkin syytä tarkastella kriittisesti, mitä vaatimuksia eri osapuolille kannattaa asettaa, koska liian suuret vaatimukset pienelle yritykselle voivat aiheuttaa viivästyksiä ja aina ei prosesseja saada tällöin toimimaan. Valintaan vaikuttaa toki asiakaskunta ja vaatimustaso; jos nähdään välttämättömäksi, että yhteistyökumppanit maksuvat SE-käytännöt, tulee varata aikaa ja resursseja käytäntöjen rakentamiseen.

Verkostovalmiudet – Systems Engineering -valmiudet

Verkostotoiminnan tehokkuutta voidaan kehittää verkostovalmiuksia kehittämällä. Valmiuksilla tarkoitetaan yleisesti sitä, miten paljon resursseja tai toimenpiteitä tehdään ennen kuin varsinainen tehtävä suoritetaan. Jos prosessit on mietitty etukäteen ja toimintamalleista sovittu, voidaan tarjouspyynnön tai asiakastilauksen tullessa nopeasti edetä itse tehtävän suorittamiseen ja arvon tuottamiseen. Pitkälle menevien valmiuksien kehittäminen ei ole kuitenkaan itsetarkoitus eikä aina kannattavaa; sopiva taso riippuu yhteistyön tavoitteista, kestosta, toistuvuudesta ja luonteesta.

Systems Engineering -lähestymistapaa pidetään yleisesti vaativana. SE:n soveltaminen esim. projektissa voidaan nähdä potentiaalisena riskitekijänä, joka aiheuttaa lisätyötä ja viivästyksiä. Näin varmaan onkin, jos SE:stä ei ole mitään tietoa ja kokemusta ennen projektia. Systems Engineering -lähestymistavan soveltaminen vaatii siis omat valmiutensa. Asiakasprojektissa ei ole aikaa enää opiskella SE-prosesseja; siihen pitäisi valmistautua etukäteen. Se ei tarkoita, että jokaisen projektiosallistujan tulee olla yhtä syvällä SE:n hallinnassa. On kuitenkin mietittävä etukäteen, miten SE-käytännöt viedään läpi, miten paljon SE-osaamista pitää kehittää ja jakaa, millaisella tiedonvaihdolla ja työkaluilla sitä voidaan tukea ja miten ne saadaan käyttöön. Jos on tarve soveltaa SE-menettelyä vain rajatulle alueelle, ei kaikissa projekteissa, kannattaa valita erikseen se joukko, jonka kanssa näitä vaativia hankkeita tehdään.

SE:n soveltaminen verkostoympäristössä vaatii:

- valmiuksia SE-prosessien ja käytäntöjen läpiviemiseen
- valmiuksia yhteistyöhön yritysten välillä

SE-valmiuksiin sisältyy riittävän osaamisen kehittäminen ja kouluttaminen sekä seuraavien elementtien määrittely:

- Selkiytetään, mitä asiakas vaatii SE-prosesseilta
- Miten SE viedään läpi omassa yrityksessä; prosessit ja roolit, johtaminen
- SE-prosessit yli yritysrajojen, miten SE-käytännöt hoidetaan yhteistyökumppaneiden ja alihankkijoiden kanssa
- Käytännöt ja työkalut sekä niiden osaaminen, esim. vaatimusten hallinnan työkalut.

Yhteistyövalmiudet voivat merkitä esim. seuraavien asioiden kehittämistä:

- Yhteiset käsitteet ja termit
- Verkoston yhteinen visio tai yhteiset tavoitteet, asiakkaan tavoitteiden ymmärtäminen
- Ydinosaamisten tunnistaminen ja kehittäminen, sekä tekeminen näkyväksi
- Yhteiset prosessit, säännöt ja toimintaperiaatteet, roolit, verkoston johtaminen ja hallinta
- Yhteisen tiedon hallinta, kommunikaatio ja tiedonvaihto + työkalut, järjestelmät
- Sopimusasiat
- Verkostopartnerien valinta, kvalifointi ja suorituksen seuranta
- Luottamuksen rakentaminen ja ylläpito

Kommunikaatio osana yhteisen ymmärryksen rakentamista

Kuten edellä on todettu, tuotekehitys ja -suunnittelutoiminta ovat nykypäivänä usein hyvin verkostoitunutta ja hajautunutta toimintaa. Tuotekehitystiimit koostuvat yhä useammin eri koulutuksen ja taustan omaavista, eri tiloissa ja jopa organisaatioissa työskentelevistä ihmisistä, jotka käyttävät työhönsä erilaisia työkaluja ja järjestelmiä.

On yleisesti tiedossa, että ihmiset kommunikoivat parhaiten yhteisissä tiloissa, joissa on tilaa kasvokkain tapahtuvalle kanssakäymiselle. Nykyinen hyvin hajautunut toimintaympäristö ei kuitenkaan välttämättä mahdollista tätä. Tämä on haaste,

jolle on löydettävä ratkaisu varsinkin monimutkaisia järjestelmiä kehitettäessä, työssä, jossa toimiva informaation jakaminen ja saatavuus ovat elintärkeitä. Systems Engineering osaltaan vastaa tähän haasteeseen, tarjoamalla (teknisen) rakenteen ja toimintamallin, jolla pyritään takaamaan, että tarvittava tieto on käytettävissä, missä ja milloin sitä tarvitaan. Mutta SE:n soveltaminenkaan ei välttämättä ratkaise ihmisten väliseen kommunikaatioon liittyviä haasteita. Näitä haasteita voidaan tunnistaa hajautetun projektinhallinnan näkökulmasta:

- selkeä määritelmä ratkaistavasta ongelmasta – jaettu yhteinen ymmärrys siitä, mitä suunnitteluprojektilla ollaan ratkaisemassa
- selkeä määritelmä kohteesta, tavoiteltavasta ratkaisusta – jaettu yhteinen ymmärrys siitä, mihin suunnitteluprojektilla tähdätään
- yhteinen ymmärrys päätöksenteon säännöistä – käytäntöjen läpinäkyvyys.

Haasteita informaation ja tietämyksenhallinnan näkökulmasta:

- Erilainen ymmärrys asioista, koska kaikilla ei ole kaikkea samaa tietoa käytettävissään. Tämä sisältää mm. monialaisissa tiimeissä työskentelevien ihmisten tai organisaatioiden erilaiset taustat, toimintaympäristöt, teollisuudenalat, koulutuksen ja ajattelutavat.
- Ihmiset kommunikoivat parhaiten kasvotusten, miten tämä ratkaistaan globaalisti hajautuneessa verkostossa?
- Tieto ja ymmärrys siitä, mitä muut osaavat ja tietävät. Kyky hyödyntää projektiin osallistuvien ihmisten tietämys.

Näiden lisäksi hajautuneessa, useita yrityksiä ja yksilöitä sisältävässä verkostossa toimittaessa tiedetään harvoin tarkkaan mitä kukin osaa ja hallitsee, ja millä tasolla heihin voidaan luottaa.

Kuinka kommunikoida?

Kommunikaation haasteisiin ei ole mitään yleispätevää ratkaisua. Perinteisesti kommunikaation haaste on yritetty ratkaista tuomalla ihmiset yhteiseen tiimityötä ja ajatustenvaihtoa tukevaan tilaan. Tämä ei kuitenkaan aina ole mahdollista. Informaatioteknologioilla on puolestaan pyritty luomaan sähköisiä työkaluja, joilla kommunikaatio toimisi myös hajautetussa ympäristössä. Nämä työkalut ovat kuitenkin vain **mahdollistajia**, jotka eivät ratkaise itsestään mitään.

Jotta tiedon jakamista ja kommunikaatiota voitaisiin parantaa, pitää ymmärtää sitä kuinka ihmiset kommunikoivat – muodostavat ja käyttävät informaatiota. Ihmisillä on aina oma näkemyksensä asioista ja heillä on usein taipumuksena olettaa, että muut näkevät asiat samalla tavoin heidän jakaessaan tietonsa aiheesta. Vastoin tätä olettamusta ihmiset kuitenkin aina heijastavat vastaanottamansa informaation omiin kokemuksiinsa. Onkin hyvin tärkeää, että verkostomaisessa suunnittelutoiminnassa kommunikaation rajapinnat, käytännöt ja kieli jaetaan ja ymmärretään samoin. Tätä voidaan edistää esimerkiksi:

- käyttämällä prosessia tai toimintamallia (työstäminen yhdessä ja keskustellen), mikä auttaa yhteisen ymmärryksen muodostamisessa koskien ongelmakenttää, haasteita ja tavoitteita
- olemalla tietoisia siitä, että ihmiset tulkitsevat jaettua tietoa eri tavoin ja peilaavat sitä omaan kokemukseen
- rakentamalla jokaisen yksilön omaa sosiaalista pääomaa – ihmissuhteiden, yhteyksien, luottamuksen, ymmärryksen sekä arvojen muodostamaa "varastoa", tarkoittaen yritystoiminnassa erityisesti luottamusta muiden osamiseen ja tietotaitoon.

SOSIAALINEN PÄÄOMA

- **Tiedä mitä** – ymmärrys tieteenalasta, perustuu esimerkiksi koulutukseen ja sertifiointeihin
- **Tiedä miten** – taito hyödyntää ennen opittua, tieteenalan sääntöjen soveltaminen ja hyödyntäminen monimutkaisissa käytännön ongelmissa
- **Tiedä miksi** – tietämys ja ymmärrys tieteenalan sääntöjen riippuvuuksista ja suhteista
- **Välitä miksi** – motivaatio, innokkuus ja mukautuminen työhön
- **Tiedä kuka** – ihmissuhteet, joiden avulla tiedetään "kuka tietää", "keneltä kysyä" ja "keneen voi luottaa"

Hajautuneessa verkostomaisessa toiminnassa on tärkeää, että kommunikaatioon liittyvät haasteet tunnistetaan ja niitä pyritään ratkaisemaan – informaatiota jaetaan ja se kommunikoidaan niin, että se myös ymmärretään. Yhteinen ymmärrys ratkaistavasta haasteesta ja tavoitteista sekä yhteinen sanasto ovat erityisen tärkeitä. Informaatio on usein hiljaisessa ("tacit") muodossa, jolloin oikeanlaisella kommunikaatiolla vältetään väärinymmärryksiä.

Kommunikaatiotyökalut

Suunnittelutoiminnassa ja muussa insinööriyössä informaation siirto IT-välineitä käyttäen on usein arvokkainta silloin, kun sitä tuetaan ihmisten välisellä suoralla kommunikaatiolla. Informaatiota pitää usein selittää, koska tietojärjestelmiin voidaan tallentaa vain osa tiedosta (hiljainen ja eksplisiittinen tieto). Jos tietoa vain jaetaan sen kommunikoimisen sijaan, on vaarana, että ihmiset eivät ymmärrä sitä, kuten oli tarkoitettu. Tällöin ihmiset lähinnä toimivat toistensa kanssa (engl. cooperation) aidon yhteistyö sijaan (engl. collaboration).

ILLUUSIO KOMMUNIKAATIOSTA

Teknisiä työkaluja (esim. sähköposti ja verkkokokoukset) käytetään yhä enemmän korvaamassa kasvotusten tapahtuvaa kommunikaatiota ja toistuvia tapaamisia. Keskitytään kehittämään järjestelmiä, prosesseja ja tuotteita ihmisten tarpeiden sijaan.

Ei voida tietää ja varmistaa, että oikea tieto saavuttaa oikean henkilön oikeaan aikaan. Ei esimerkiksi voida tietää, että hyvissä ajoin lähetetty sähköposti löytyisi enää silloin kun siinä oleva tieto olisi tarpeen.

Ihmisillä on tapana ajatella, että muut ymmärtävät ja näkevät asiat samalla tavalla kuin he itse. Ja olettavat, että kun asia on kommunikoitu, on se myös ymmärretty.

On tärkeää muistaa myös *kommunikaation* ja *yhteisön* välinen ero.

Systems Engineering – toimintatavan implementointi

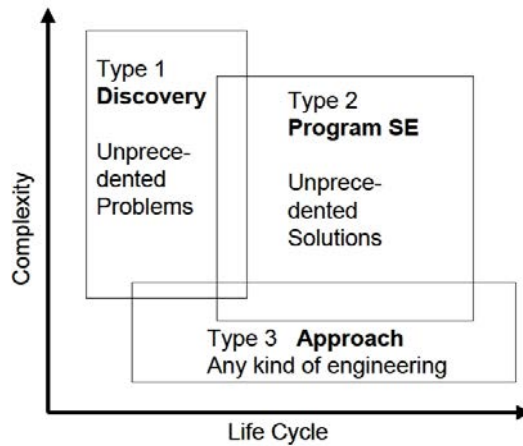
Tässä luvussa käsitellään SE-toimintatavan käyttöönottoa organisaatiossa. Mitä tekijöitä pitää ottaa huomioon? Miten lean-periaatteita voidaan soveltaa SE-prosesseissa?

SE-toimintatavan soveltaminen – erilaiset käyttöönottilanteet

Vaikka Systems Engineering -toimintatapa perustuu standardeihin, eivät sen käytännön sovellukset eri organisaatioissa ole yhteneväisiä. SE-toimintatapa esitetäänkin skaalautuvana menettelynä. Skaalautuminen voi tarkoittaa erilaista SE-prosessien soveltamista, tuotearkkitehtuureja, vaatimusten hallintatasoja, organisatorista toteutusta jne. Ei ole olemassa yhtä ainoaa SE-sovellustapaa. SE-käytäntöjen toteuttamistapa on aina riippuvainen tavoitteista, ulkoisista ja sisäisistä ajureista, sovelluskohteen vaatimuksista, kulloisestakin tilanteesta, organisaation osaamistasosta (kypsydestä) sekä käytettävissä olevista resursseista. Myös organisaation sisällä voidaan soveltaa erilaisia prosesseja uuden kehittämiseen, vanhan modifiointiin ja valmiin tuotteen käyttämiseen tai eri tuoteryhmiin.

Aloilla, joilla tuotteiden käyttöön liittyy suuria riskejä, vaaditaan luotettavuuden ja turvallisuuden varmistamista. Hankalasti huollettavalle tuotteelle voidaan asettaa korkea käytettävyyksivaatimus. Näissä tapauksissa usein asiakas voi vaatia SE-prosessien noudattamista, mutta SE-menettelyä voidaan soveltaa myös tukemaan viranomaisvaatimusten täyttämistä. Tällöin SE:n soveltaminen määrittyy yleensä alan yleisten käytäntöjen ja asiakkaan hyväksymän tason mukaan.

Toisaalta SE:n soveltamiseen voidaan myös lähteä yrityksen sisäisestä tavoitteesta esim. systematisoida ja harmonisoida suunnittelukäytäntöjä, kun tuotteet monimutkaistuvat, suunnittelu hajautuu ja on vaikea hallita kokonaisuutta. SE ja mallipohjainen suunnittelu tukevat toisiaan. SE:tä soveltamalla voidaan myös tukea vanhojen tuotteiden ylläpitoa. Kun kiinnostus SE:n soveltamiseen lähtee yrityksen sisältä, voidaan SE:n sovellustaso määrittellä siten, että se tukee parhaiten yrityksen omia tavoitteita.



Kuva 5. 3 SE:n sovellustapaa (Sheardin mukaan) (Sheard, 2000).

Yksi tapa kuvata SE:n erilaisia toteutustapoja on kuvassa 5. Sheard on tunnistanut kolme tapaa:

- Discovery – Havaitseva – ongelmien tunnistaminen
- Program SE – ratkaisujen tunnistaminen
- Approach – sovellusvariaatiot.

”Discovery”-tapa soveltuu parhaiten tilanteisiin, joissa on tavoitteena ymmärtää monimutkaista ongelmakenttää. Usein kyse on uusista kehitettävistä systeemeistä ja mukana on suuri joukko ”systeemi-insinöörejä”. ”Program SE”-toteutuksessa on tyypillisesti nimetty erityinen joukko ”systeemi-insinöörejä” tai vain yksittäinen henkilö hoitamaan SE-prosessia. Tässä tapauksessa ongelmakenttä on tunnetumpi ennalta ja keskeisenä tavoitteena voi olla uuden tuotevariaation suunnittelu kustannustehokkaasti ja täyttäen laatu- ja aikataulukriteerit. ”Approach” soveltuu pienempiin projekteihin tai järjestelmiin ja koko elinkaarelle ja se tarkoittaa sitä, että SE toteutetaan prosessina tai lähestymistapana, jota kukin suunnittelija noudattaa: riskien ymmärtäminen, tarpeen ymmärtäminen, vaatimusten selkiyttäminen ennen ratkaisua, integroinnin varmistaminen.

Aracic [2012] tunnistaa SE:n hyväksikäytölle erilaisia ohjaavia periaatteita kommunikaatiolähtökohdasta: valitaan, missä perusprosesseissa (vaatimusten määrittely, arkkitehtuuri, analyysi ja allokointi, systeemin hallinta) pääosin käsitellään tietoja tai dokumentteja, joita vaihdetaan eri osapuolien välillä. Näin tunnistetaan kolme hyväksikäytön moodia, jotka eivät ole toisensa poissulkevia ja voivat toimia myös toistensa kombinaationa:

- Vaatimuslähtöinen SE: Projektissa on suuri määrä alihankkijoita, joilla on kaikilla hyvin erilaiset osaamisalueet. Tällöin vaatimusten hallinta -prosessi yleensä valitaan keskeiseksi ja keskeiset kommunikoidtavat tiedot liittyvät vaatimuksiin, sopimukseen ja oikeaksi osoittamiseen. Tämä on yleisin lähes-

tymistapa SE:n soveltamisessa. Se tukee myös sopimusten hallintaa sekä asiakkaiden että alihankkijoiden suuntaan.

- Mallipohjainen SE: Projekti, jossa on pieni määrä alihankkijoita ja jossa pääsopijaosapuolella on keskeinen erikoisosaaminen. Referenssiprosessiksi valitaan usein ”arkkitehtuuri, analyysi ja allokointi”, vaihdettavat tiedot voivat olla esim. Catia-osakuvaustiedosto, geometriaparametrit, materiaalit. Mallipohjaisuutta pidetään moodina, jolla saadaan suurimmat tuottavuushyödyt. Tosin kääntöpuolelta mainitaan, että hyötyjen realisoiminen voi viedä aikaa, koska pitää päästä yhteisymmärrykseen malleista ja työkaluista ja tarvitaan uutta osaamista jne. Voidaan vielä läpi asteittain, jos tehdään kunnollinen SE:n hyväksikäyttösuunnitelma.
- Palautelähtöinen SE: Projekti käyttää ”ketterää” lähestymistapaa, säännöllistä validointia ja verifiointia ennen seuraavan vaatimusjoukon määrittelyä. Tällöin ”Systeemin hallinta” -prosessi voi olla keskeinen kommunikaatio-prosessi ja vaihdettavia tietoja ovat suorituskyvyn mittaukset ja poikkeamareportit. Tämä soveltuu systeemeille, joiden laatu voidaan arvioida tehokkaasti vasta kun niitä käytetään. Avainsana on käytännöllisyys; palaute voi olla jotakin hyvin käytännöllistä ja luotettavaa, jolla voidaan nopeasti rakentaa korkeatasoinen tietämuskanta. Tämä tapa ei myöskään ”sekoita” käytössä olevia suunnittelumenetelmiä. Ongelmana on, että palaute on yleensä ”negatiivista”. Lisäksi pitää määrittellä selvä strategia palautetiedon käyttöön.

Tilanneriippuvainen SE:n käyttöönottopaatus: SE:n soveltaminen rautatie-sektorilla. Rautatiesektorilla tunnetaan kasvavaa kiinnostusta SE-konseptien soveltamiseen (mm. Lontoon maanalainen, Hollannin rautatieinfrastruktuuri). Osittain samoja periaatteita on aiemmin sovellettu eri termeillä. SE-käytäntöjen soveltamisessa törmättiin kuitenkin esteisiin, jotka johtuvat rautatiesektorin ja perinteisen SE:n soveltamiskentän eroista: 1. Rautatieprojekteissa on yleensä kysymys enemmän olemassa olevien järjestelmien laajentamisesta kuin uusien luomisesta. 2. Rautatieprojekteissa yleensä järjestelmien pitää pystyä toimimaan myös muutoksen aikana. Perinteinen SE-opastus ei ota näitä tilanteita huomioon. Koska ei ole pystytty sopimaan, millä laajuudella ja rakenteella SE:tä sovelletaan, on ollut vaikea saada perinteisen rautatietoimialan jäsenien joukossa yhteistä ymmärrystä siitä, miten toimitaan. Myös SE-sanastot ja -termit ovat tuottaneet vaikeuksia. Erityisenä piirteenä on otettava huomioon, että rautatiealalla on jo olemassa käytäntöjä, jotka toteuttavat joitakin SE-aktiiviteetteja. Jos nämä jätetään huomiotta, voidaan tarpeettomasti muuttaa tehokkaita olemassa olevia prosesseja tai tehdä samat asiat kahteen kertaan. Voidaan myös joutua ristiriitaan olemassa olevien rautatiealan käytäntöjen kanssa. Ongelman ratkaisemiseksi esitetään, että tarvitaan alalle sovitettujen käytäntöjen määrittelyä, SE:n esittämistä yksinkertaisella kielellä, rajojen (esim. SE – projektinhallinta) määrittelyä, SE:n tuomien uusien toimintojen sovittamista jo olemassa oleviin, eri osapuolien ottamista mukaan käytäntöjen määrittelyyn ja toteutukseen sekä asteittaista käyttöönottoa.

Systems Engineering ja monimutkaiset systeemit

Perinteisesti SE:n avulla on pyritty ratkaisemaan määritelty ongelma suunnittelemalla ja rakentamalla sille ratkaisu. Tosielämässä ongelmia ei lähdetä ratkaisemaan "tyhjältä", vaan systeemit ovat jo olemassa. Systeemeissä voi olla eri osapuolien eri aikoina ja eri tarkoituksiin kehitettyjä komponentteja. Teknologiat ja käyttäjien tarpeet ja vaatimukset muuttuvat. Monimutkaiset systeemit ja systeemeistä muodostuvat systeemit ("systems of systems") voivat olla lähempänä kaaosta kuin järjestystä, vaikka ne voivatkin sisältää hyvin määriteltyjä komponentteja. Rajapinnat muodostuvat usein kehityksen kuluessa "suunnittelemattomiksi".

Sheardin [2009] mukaan kompleksisia järjestelmiä pitääkin ajatella enemmän *kehittyvinä*, kuin valmiiksi suunniteltuina systeemeinä. Määritellään tavoitteet, jotka toteuttavat halutun lopputuloksen, jota kohti mennään. Käytetään hyväksi todettuja elementtejä ja ohjataan koko systeemiä kohti haluttua lopputulosta. Tätä voidaan soveltaa myös itse Systems Engineering -prosesseihin: Ajan myötä kehittyneiden prosessien selvittäminen voi olla parempi lähtökohta kuin määritellä suoraan lopulliset uudet prosessit. Muita suositeltavia periaatteita ovat mm.:

- Tunnista lokaalit toimenpiteet, jolla voi olla globaalit seuraukset. Valmistaudu muutoksiin.
- Säilytä useita ratkaisuvaihtoehtoja. Luo tarkoituksella monimuotoisuutta systeemiin.
- Tunnista eri muutosnopeudella kehittyvät osiot ja eristä ne toisistaan rajapintakerroksilla.
- Luovu täydellisestä optimoinnista.
- Luo rikkaat mielikuvamallit ongelmakentän ymmärtämiseksi. Koordinoi ja yhdistä ihmisiä ja ryhmiä.

Lean-periaatteet SE:n soveltamisessa

Lean-ajattelun ydinajatus on tuottaa asiakkaalle suurin mahdollinen arvo samalla minimoiden syntynyt hukka. Hukkaa ovat mm. ylituotanto, turha odottaminen ja etsiminen, tarpeettomat kuljetukset ja varastointi. Hukkaa ovat myös hyödyntämättömät resurssit – koneet, tilat ja työntekijöiden hyödyntämätön potentiaali.

Lean-ajattelu sai alkunsa 1980-luvun lopulla, kun Toyotan autotehtaiden toimintamallia alettiin kutsua termillä "lean". Johtuen tuotantokeskeisestä historiastaan lean mielletään usein tuotannon tehostamisen menetelmäksi, koska se on enemmänkin koko organisaation toimintafilosofia, on se sovellettavissa myös muunlaisessa liiketoiminnassa, kuten palveluissa.

Lean ei ole kehitystyökalu, vaan enemmänkin yrityksen tapa toimia, toimintafilosofia, jossa pyritään tuottamaan täydellinen asiakasarvo täydellisellä arvonluontiprosessilla, joka ei tuota lainkaan hukkaa. Täydellisyttä tavoitellaan muuttamalla

(johtamis-) toimintaa eri teknologioiden, omaisuuden ja vertikaalisten osastojen osaoptimoinnin sijaan tuotteiden ja palveluiden koko horisontaalisen arvoketjun optimointiin – yli teknologioiden, omaisuuden ja osastojen aina asiakkaaseen asti. Koko arvoketjun hukkaa vähentämällä, yksittäisten pisteiden sijaan, pyritään luomaan prosessi, joka tarvitsee vähemmän työtä, tilaa, pääomaa ja aikaa tuotteiden ja palveluiden tuottamiseksi. Samalla kustannukset pienenevät ja virheiden määrä laskee. Toimintamallilla pyritään myös valmiuteen nopeasti muuttuviin asiakas- ja markkinavaatimuksiin reagointiin. Etuja saavutetaan myös informaation hallinnassa, yksinkertaisemman prosessin kautta.

Tarkoitus, prosessi, ihmiset

Muutoksessa kohti lean-ajattelua korostetaan kolmea olennaista asiaa, jotka muutosta läpi vievien johtajien tulisi pitää mielessään ja antaa ohjata kohti päämäärää:

- Tarkoitus: Mitkä asiakkaan ongelmat yritys pyrkii ratkaisemaan pyrkiessään kohti omaa menestystään?
- Prosessi: Kuinka yritys arvioi ja määrittää jokaisen merkittävän arvoketjun, jotta voidaan varmistua että jokainen otettava askel tuottaa arvoa, on kykenevä, saavutettavissa, riittävä, sekä joustava ja että jokainen askel on liitetty toisiinsa virta- (flow), imu- (pull) tai tasausohjauksen (levelling) avulla.
- Ihmiset: Kuinka yritys voi varmistua, että jokaiselle tärkeälle prosessille löytyy vastuuhenkilö, joka jatkuvasti arvioi ja kehittää prosessiaan liiketoiminnan ja lean-periaatteiden lähtökohdista? Kuinka jokainen arvoketjussa työskentelevä saadaan osallistettua toimimaan oikein ja pyrkimään jatkuvaan parantamiseen.

INCOSEn (International Council for Systems Engineering) lean SE -työryhmä on laatinut ohjeita siitä, miten lean-periaatteita voidaan soveltaa SE:hen. Materiaalia aiheesta löytyy INCOSEn verkkosivuilta:

<http://www.lean-systems-engineering.org/downloads-products/lean-enablers-for-systems-engineering/downloads-lean-enabler-for-systems-engineering/>.

Organisatorinen käyttöönotto ja sen onnistumistekijät

Tietotekniikan puolella määritellään, että organisatorinen käyttöönotto sisältää koko prosessin lähtien tarpeiden analyysistä ja teknisen ratkaisun valinnasta siihen, kun teknologian tuoma hyöty on toteutunut organisaation toiminnassa (Munkvold 1999). Se ei siis rajoitu teknisen järjestelmän määrittelyyn, toteutukseen ja asentamiseen, vaan sisältää myös "todellisen käytön", käyttäjien osallistamisen, koulutuksen ja sitouttamisen käyttöön. Usein käyttöönotto tehdään askelittain tai vaiheittain; tuoteryhmä, moduuli tai organisaation osa kerrallaan, jolloin seuraavassa vaiheessa voidaan käyttää hyväksi edellisten vaiheiden oppia.

Systems Engineering -toimintatavan käyttöönotto muistuttaa monessa suhteessa tietotekniikan käyttöönottoa ja usein tueksi tarvitaankin myös uusia ohjelmistotyökaluja. Se vaatii tarpeiden tunnistamista ja tavoitteiden asettamista. Se muuttaa toimijoiden toimintatapoja ja prosesseja ja vaikuttaa eri osapuolien yhteistyöhön. Se vaatii uuden osaamisen ja uusien roolien kehittämistä organisaatioon. Jotta SE-prosessi voidaan viedä läpi, toimijoiden pitää ymmärtää sen tavoitteet ja sitoutua sen toimintatapoihin. Tätä tukee osapuolien osallistaminen ja tiedotus hankkeen aikana.

Tietotekniikan puolella on määritelty ”kriittisiä onnistumistekijöitä” (Critical success factors, CSF) tukemaan monimutkaisten järjestelmien (kuten ERP tai PLM) käyttöönottoa. Kriittiset onnistumistekijät määritellään ”niinä suorituksen avainalueina, jotka ovat oleellisia tehtävän tai projektin onnistumiselle” (Araujo 2006). Onnistumistekijät eivät kuvaa ainoastaan sitä, että hanke pysyy budjetissa ja aikataulussa, vaan kiinnittää huomiota suunniteltujen hyötyjen saavuttamiseen. Onnistumistekijöitä voi käyttää apuna tarkistuslistoina käyttöönoton suunnittelussa ja läpiviennissä.

SE-prosessien käyttöönotto koskettaa usein eri toimijoita organisaatiossa ja jopa sen ulkopuolella. Tällöin ei riitä, että yksittäinen pääsuunnittelija tai Systems Engineer hoitaa sen implementoinnin, vaan tarvitaan organisoitua käyttöönottoa. Monet IT-käyttöönoton onnistumistekijät ovat relevantteja myös SE-käyttöönotossa. Alla taulukossa 6 useista lähteistä peräisin olevia onnistumistekijöitä on vedetty yhteen aihepiireittäin ja muokattu sopivaksi SE-käyttöönottoon.

Taulukko 6. SE:n käyttöönoton onnistumistekijät.

Onnistumistekijät SE:n käyttöönottoon organisaatiossa	Selitys
1 Visio ja johtaminen	
1. Yritysjohdon tuki, sitoutuminen ja jatkuva osallistuminen koko käyttöönoton ajan.	Yritysjohdo tukee käyttöönottoa ja tekee sen myös näkyväksi, ei ainoastaan antamalla resursseja, vaan myös kiinnostuksella, osallistumalla ja seurannalla.
1.2 SE-käyttöönoton visio ja tavoitteet ja suunnitellut sekä hyödyt (myös pitkällä tähtäimellä) ovat selvät. Ne on jaettu ja dokumentoitu organisaatiossa ja tarvittaessa sen ulkopuolella.	Osallistujat ymmärtävät SE:n hyödyt ja tulevaisuuden vision. Jos implementointi vaatii myös alihankkijoita tai muita ulkoisia tahoja muuttamaan toimintatapojaan tai prosessejaan, osapuolet osallistetaan päätöksentekoon.
2 Osaaminen ja käyttöönoton suunnittelu ja organisointi	
2.1 Organisaation osaaminen on riittävää, kulttuuri myönteinen muutoksille ja käyttöönototiimiin saadaan riittävät taidot ja osaaminen eri toiminnoista.	Käyttöönotto-tiimissä riittävät kompetenssit. On hyvä, jos on aiempaa kokemusta prosessin muuttamisesta. Luottamus toimintojen ja ihmisten välillä.
2.2 Projektinhallinta hyvällä tasolla, pätevä ja sitoutunut projektin vetäjä = "project champion".	Riippuen käyttöönoton koosta ja laajuudesta: Projektin tarkoituksenmukainen suunnittelu ja seuranta, johtoryhmän perustaminen ja osallistuminen, riskien hallinta
2.3 Asteittainen/vaiheittainen käyttöönotto. Hyötyjen näyttäytyminen matkan varrella. Oppiminen käyttöönoton kuluessa.	Valitaan pilottikohde, jolla suuri onnistumistodennäköisyys.
2.4 Tarvittaessa otettava huomioon eri organisaatiot, toimipaikat -> paikalliset "championit".	Hajautetussa organisaatiossa kussakin toimipisteessä on nimetty "project champion", joka kääntää vision todellisuudeksi toimipisteen näkökulmasta, hoitaa yhteyksiä kokonaisprojektiin, ratkaisee ongelmia ja pitää yllä etenemistä.
2.5 Jälkiseuranta ja parantaminen	Jatketaan prosessia ensimmäisen käyttöönoton jälkeen, seurataan hyötyjä, tarvittaessa muokataan prosesseja.
3 Prosessien mukauttaminen ja toimijoiden osallistaminen	
3.1 Nykyisten käytäntöjen ja prosessien mukauttaminen; uusien prosessien määrittely ja organisointi.	Tunnistetaan tarvittavat prosessimuutokset (mihin toimintoihin ja tiedonvaihtoon SE-käytäntö vaikuttaa). Määritellään roolit ja vastuut, muutoksen hallinta; tarvittaessa myös organisaatioiden välillä.

Onnistumistekijät SE:n käyttöönottoon organisaatiossa	Selitys
3.2 SE-prosessiin osallistujien ja siihen liittyvien toimijoiden osallistaminen käyttöönottoon mahdollisimman aikaisin.	Osallistaminen tukee ihmisten motivaatiota, sopivan toimintatavan löytämistä, yhteisen ymmärryksen luomista ja odotusten hallintaa. Auttaa toimijoita ymmärtämään, miten oma työ muuttuu. Vähentää ihmisten kokemaa epävarmuutta ja siten myös muutosvastarintaa. Annetaan resursseja, mutta estettävä turhautuminen. Otettava huomioon potentiaaliset valtaristiriidat. Kannustimet muutokseen.
3.3 Riittävä koulutusohjelma ja -resurssit.	Osallistujien kouluttaminen, tehokas organisaation oppimisprosessi, focus tavoitteiden ymmärtämisessä.
3.4 Vahva kommunikaatio muutoksesta sekä organisaation sisällä että tarvittaessa ulospäin.	Tietoisuus tavoitteista, motivointi, vaikutukset organisaatioon, mitä tapahtuu milloinkin, missä mennään.

Nämä tekijät tarvitaan myös PLM-käyttöönotossa. Ne ovat tärkeitä mm. loppukäyttäjien hyväksynnän ja sitoutumisen kannalta. Lisäksi tarvitaan joitakin lisätekiäjiä:

4 PLM-järjestelmän valinta ja toteutus (edellisten lisäksi)	
4.1 Valitaan organisaation tarpeisiin mahdollisimman hyvin sopiva järjestelmä.	Määritellään järjestelmän yleiset vaatimukset (mm. haluttu tuki toimittajalta, hajautetun käytön tarve jne.) ja yleisen tason toiminnalliset vaatimukset (ei vielä yksityiskohtaisesti). Valitun systeemin pitää pystyä mukautumaan olemassa olevaan valmistusrakenteeseen ja prosesseihin.
4.2 Tarkistetaan linkit olemassa oleviin järjestelmiin ja datan saatavuus.	Mihin järjestelmiin PLM:n tulee liittyä? Onko tuotetdata saatavilla/olemassa PLM:n edellyttämässä muodossa?
4.3 Tarjoa PLM-järjestelmä mahdollisimman laajasti eri osastojen ja osapuolien käyttöön siltä osin kuin ne käsittelevät tuotetietoa; myös yhteistyökumppaneille.	Tämä tukee yhteistyötä ja digitaalista työnkulkua, mutta vaatii myös muiden osapuolien osallistamisen/kouluttamisen käyttöön (riippuen käyttötavasta).

Vaatimusten luokittelu

Jarmo Alanen, VTT

Tässä luvussa tavoitteena on antaa ohjeita vaatimusten luokitteluun, jotta vaatimusten jäsentäminen ei olisi työlästä ja vaatimusluokat olisivat selkeät. Kontekstina on järjestelmätaso, vaikka esimerkkejä ohjelmistovaatimusmäärittelyistä käytetäänkin.

Sisältö

Johdanto	114
Esimerkkejä luokittelusta	116
Toiminnalliset – ei-toiminnalliset.....	116
Vaatimustyyppin mukaan luokittelu	117
Lähteen mukaan luokittelu.....	120
Kohteen mukaan luokittelu	121
Tieteenalan mukaan luokittelu	121
Prioriteetin mukaan luokittelu.....	121
Elinkaaren mukaan luokittelu	122
Vaatimuksen toteuttajan mukaan luokittelu.....	122
Rajoitteet vs. vaatimukset.....	122
Sidosryhmävaatimukset	123
Prosessivaatimukset	124
Luokittelun yleisiä sääntöjä ja ohjeita	124
Ehdotus luokitteluksi	126
Esimerkkejä ehdotetun luokittelun mukaisista vaatimuksista	130
Lähdeluettelo (Osa II)	133

Johdanto

Vaatimustenhallinta koostuu useasta eri tehtävästä, joista tärkeimmät on lueteltu seuraavassa:

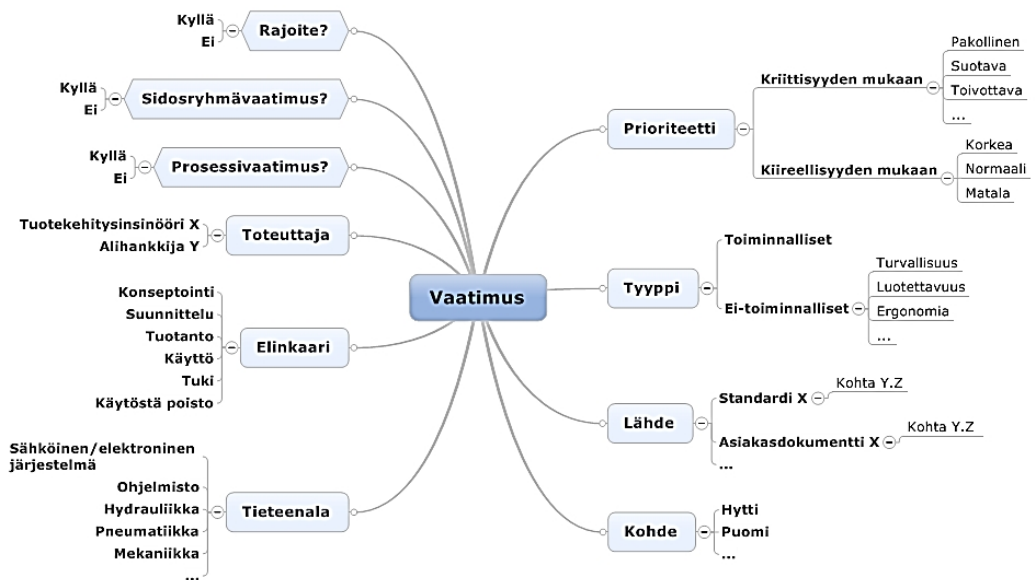
1. **Vaatimusten kartoitus** (sidosryhmien tunnistaminen; asiakkaan, lainsäädännön ja muiden sidosryhmien vaatimusten esiin kaivaminen ja niiden systemaattinen kirjaaminen)
2. **Vaatimuksen kuvaaminen** (hyvälaatuiset vaatimuslauseen; vaatimuslause-runkojen hyväksikäyttö; mallien ja diagrammien käyttö)
3. **Vaatimusten todentaminen ja kelpuuttaminen** (tarkistetaan, että vaatimukset ovat hyvälaatuisia ja että ne kuvaavat sen järjestelmän ja palvelun, mitä asiakas on haluamassa)
4. **Vaatimusten analyysi** (vaatimusten muuntaminen insinöörimaailmaan sel-laisessa muodossa, että järjestelmä voidaan niiden perusteella suunnitella; usein tässä vaiheessa vaatimuksiin tulee mitattavia ilmaisuja)
5. **Vaatimusten tarkentaminen ja kohdentaminen järjestelmäarkkitehtuu-riin** (kohdennetaan laitteistoon, ohjelmistoon jne. yksityiskohtaisia toteutet-tavissa olevia vaatimuksia)
6. **Vaatimusten luokittelu ja kommunikointi** (erilaisia näkymiä vaatimus-massasta erilaisissa muodoissa eri tahoille, kuten alihankkijoille tai sidos-ryhmille)
7. **Vaatimusprosessin ja vaatimusten elinkaaren hallinta** (vaatimusten numerointi ja organisointi, tallennus ja versionhallinta sekä priorisointi; tuki suunnitelman ja tuotteen todennukseen ja kelpuutukseen; vaatimusten työkulkujen tuki; jäljitettävyyssketjujen hallinta; vaatimusten luojien ja käyt-täjien yhteistyön hallinta).

Tässä yhteydessä paneudutaan vaatimusten luokitteluun (ja sitä kautta osittain myös vaatimusten kommunikointiin). Tämä valinta on tehty siksi, että käytännön projekteissa vaatimusten luokittelun on havaittu olevan ongelmallista, mutta siitä huolimatta siihen annetaan verraten suppeita ohjeita kirjallisuudessa.

- tyyppin mukaan

- lähteen mukaan
- kohteen mukaan
- tieteenalan mukaan
- prioriteetin tai kriittisyyden mukaan
- elinkaaren mukaan
- toteuttajan mukaan.

Lisäksi luokittelua voidaan tehdä sen perusteella onko vaatimus rajoite, sidosryhmävaatimus tai prosessivaatimus (ks. kuva 1).



Kuva 1. Vaatimusten luokittelutapoja.

Kun vaatimuksia on paljon, hyvää luokittelutapaa on vaikea löytää, varsinkin kun suunnittelun eri vaiheissa tarvitaan erilaisia näkökulmia vaatimusmassaan. Esimerkiksi vaatimusten kartoituksessa saatetaan tarvita erilaista luokittelua kuin varsinaisessa tuotekehitysprojektissa.

Esimerkkejä luokittelusta

Toiminnalliset – ei-toiminnalliset

Tyypillisin vaatimusten luokittelutapa on luokittelu toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Tällaisen luokittelun ongelmana on liian karkea jako (vain kaksi luokkaa), joten ainakin ei-toiminnallisia vaatimuksia täytyy luokitella lisää. Esimerkiksi IEEE 830:1998 luokittelee vaatimukset kuuteen luokkaan seuraavasti:

1. Toiminnot
2. Ulkoiset rajapinnat
3. Suorituskykyvaatimukset
4. Loogiset tietokantavaatimukset
5. Suunnitelman rajoitteet
6. Ohjelmiston kyvykkyyttä kuvaavat attribuutit.

Näistä ensimmäinen kuuluu luokkaan toiminnalliset vaatimukset ja loput luokkaan ei-toiminnalliset vaatimukset. Eli vaatimuksia ei jaeta suoranaisesti toiminnallisiin ja ei-toiminnallisiin vaatimuksiin, vaan toiminnalliset vaatimukset ovat yksi vaatimustyyppi muiden joukossa. Tästä seuraa se, että esimerkiksi vaatimuksen, joka kuuluu luokkaan kolme, ei eksplisiittisesti tiedetä olevan ei-toiminnallinen vaatimus; toisin sanoen vaatimuksen ollessa irrallaan siinä ei ole tietoa ei-toiminnallisuudesta, vaan se voidaan päätellä implisiittisesti asiayhteydessään, kun tarkkailijalla on tiedossa yllä listattu kuuden luokan luokittelu.

Kyseinen standardi antaa esimerkin suorituskykyvaatimuksesta (eli luokkaan kolme kuuluvasta vaatimuksesta):

95 % transaktioista täytyy suorittaa alle sekunnissa.

Tämä vaatimus ei siis standardin luokittelun mukaisesti ole toiminnallinen vaatimus. Mutta keskiverto insinööri voi helposti luokitella vaatimuksen toiminnalliseksi, koska se asettaa vaatimuksen toiminnolle, vaikkakaan se ei kuvaa, mitä järjestelmän täytyy osata tehdä, vaan miten hyvin järjestelmän täytyy pystyä se tekemään. Siksi on hyvä tähdentää, että toiminnalliset vaatimukset ovat vastaus kysymykseen, *mitä* järjestelmän tai järjestelmällä täytyy pystyä tekemään.

Jako toiminnallisiin ja ei-toiminnallisiin vaatimuksiin voi siis olla joissakin tapauksessa häilyvää. Esimerkiksi turvallisuusvaatimuksia pidetään yleensä ei-toiminnallisina vaatimuksina, mutta voi olla vastahakoista luokitella, ainakaan kaikkia, toiminnallisen

turvallisuuden standardista löytyviä vaatimuksia tai turvatoiminnon vaatimuksia ei-toiminnallisiksi. Jako toiminnallisiin ja ei-toiminnallisiin vaatimuksiin toimii hyvin ainoastaan prosessin alkupäässä, eli silloin kun määritellään asiakasvaatimuksia tai muita sidosryhmävaatimuksia: toiminnallisista vaatimuksista voidaan analyysin kautta määritellä käyttötapaukset tai toimintakonseptit (engl. Concept of Operations, ConOps), eli järjestelmäsuunnittelijalle ne antavat lähtökohdan, mistä vaatimuksista järjestelmää lähdetään suunnittelemaan (tässä oletetaan normaali suunnittelu-prosessi, jossa toiminnallisuus määritellään ensin alustariippumattomasti, jonka jälkeen suunnitellaan toiminnallisuuden toteuttava fyysinen arkkitehtuuri).

Vaatimustyyppien mukaan luokittelu

Vaatimustyypeillä tarkoitetaan esimerkiksi turvallisuusvaatimuksia, luotettavuusvaatimuksia, ergonomiavaatimuksia jne. Selkeää ja kattavaa listaa vaatimustyypeistä on vaikea tehdä. Tässä luetellaan kuitenkin joitakin vaatimustyyppisiä:

1. liiketoimintavaatimukset
2. noudatettava lainsäädäntö ja muut määräykset
3. käytettävyyksivaatimukset
4. toiminnalliset vaatimukset
5. tekniset vaatimukset
6. ergonomiavaatimukset
7. liityntärajapintojen vaatimukset (sisäiset ja ulkoiset rajapinnat)
8. yhteensopivuusvaatimukset
9. luotettavuusvaatimukset (sis. toimintavarmuus-, kunnossapidettävyyss- ja kunnossapitovarmuusvaatimukset eli yhteensä käyttövarmuusvaatimukset)
10. ympäristöolosuhteista johtuvat vaatimukset (sis. EMC, ilmastolliset, seis-miset, mekaaniset, kemialliset jne. ympäristöolosuhteet)
11. infrastruktuurin vaatimukset
12. suorituskykyvaatimukset
13. tehonsyötön vaatimukset (sis. vaatimukset energialähteille)
14. turvallisuusvaatimukset
15. tietoturvaan ja käyttöoikeuksiin liittyvät vaatimukset
16. ympäristövaikutusten vaatimukset
17. materiaalivaatimukset
18. tuotantovaatimukset
19. valmistajaorganisaation kyvykkyyksivaatimukset
20. kuljetusvaatimukset
21. varastointivaatimukset
22. kokoonpanon vaatimukset
23. asennusvaatimukset
24. testattavuusvaatimukset
25. todennus-, kelpuutus- ja hyväksyntävaatimukset
26. koulutusvaatimukset
27. käyttöönoton vaatimukset

- 28. käytöstä poiston vaatimukset
- 29. laajennettavuus-, muunneltavuus- ja skaalautuvuusvaatimukset
- 30. dokumentaatiovaatimukset
- 31. prosessivaatimukset
- 32. laatuvaatimukset
- 33. elinkaarivaatimukset.

Vaatimustyyppejä on siis huomattava joukko. Käytännössä vaatimustyyppit valitaan ja mukautetaan kehitettävänä olevan järjestelmän mukaisesti. Valintoihin vaikuttavat myös sekä kehitettävän järjestelmän tilaajan (ja muiden sidosryhmien) että sen kehittäjän, ml. alihankkijat, tavat jäsentää vaatimuksia.

Vaatimustyyppejä käytetään usein vaatimusmäärittelydokumenttien otsikoina. Seuraavassa on esimerkki järjestelmän vaatimusmäärittelydokumentin rungosta standardista ISO/IEC/IEEE 29148 [2011].

1. JOHDANTO

- 1.1 Järjestelmän tarkoitus
- 1.2 Järjestelmän sovellusala
- 1.3 Järjestelmän yleiskatsaus
 - 1.3.1 Järjestelmän konteksti
 - 1.3.2 Järjestelmän toiminnot
 - 1.3.3 Käyttäjän ominaisuudet
- 1.4 Määritelmät

2. VIITTEET

3. JÄRJESTELMÄVAATIMUKSET

- 3.1 Toiminnalliset vaatimukset
- 3.2 Käytettävyyksivaatimukset
- 3.3 Suorituskykyvaatimukset
- 3.4 Järjestelmän rajapinta
- 3.5 Järjestelmän käyttö
- 3.6 Järjestelmän tilat ja toimintatavat
- 3.7 Fyysiset ominaisuudet
- 3.8 Ympäristöolosuhteet
- 3.9 Järjestelmän turvallisuus
- 3.10 Tiedonhallinta
- 3.11 Ohjeet ja säännökset
- 3.12 Järjestelmän elinkaaren
- 3.13 Pakkaaminen, käsittely ja perille toimittaminen

4. TODENTAMINEN

(rinnakkaisesti kohdan kolme otsikoitten kanssa)

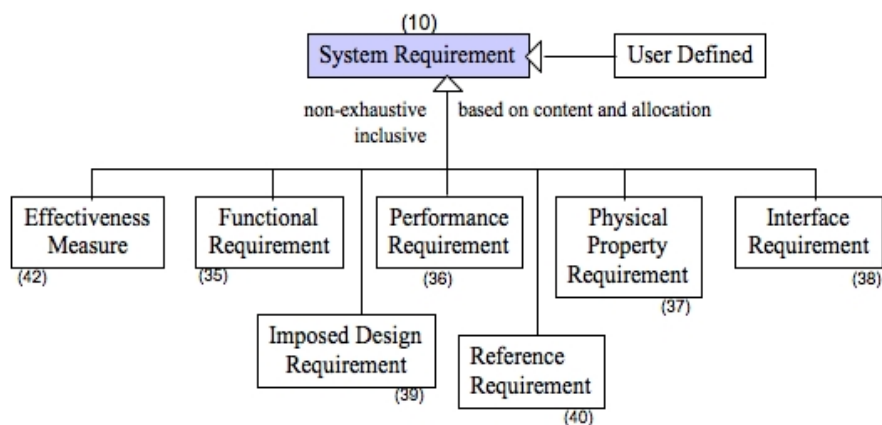
On huomattava, että ennen järjestelmän vaatimusmäärittelydokumenttia voidaan tehdä sidosryhmävaatimusdokumentti. Esimerkki sellaisen rungosta löytyy myös standardista ISO/IEC/IEEE 29148 [2011]. Samoin kyseisestä standardista löytyy selitykset yllä olevan vaatimusmäärittelyrungon otsikoille; esimerkiksi otsikon *Järjestelmän käyttö* alta löytyy yllättäen toimintavarmuus- ja kunnossapitovaatimukset.

Kuten yllä olevassa esimerkissäkin, vaatimusmäärittelydokumentin alkuluvuissa kuvataan yleensä kehitettävän järjestelmän konsepti sillä tasolla, kun se vaatimusmäärittelydokumentin tekovaiheessa on tiedossa. Dokumenttirunkoesimerkin kohdassa kolme esitellään sitten varsinaiset vaatimukset, ja tässä nimenomaan vaatimustyyppin mukaan luokiteltuina.

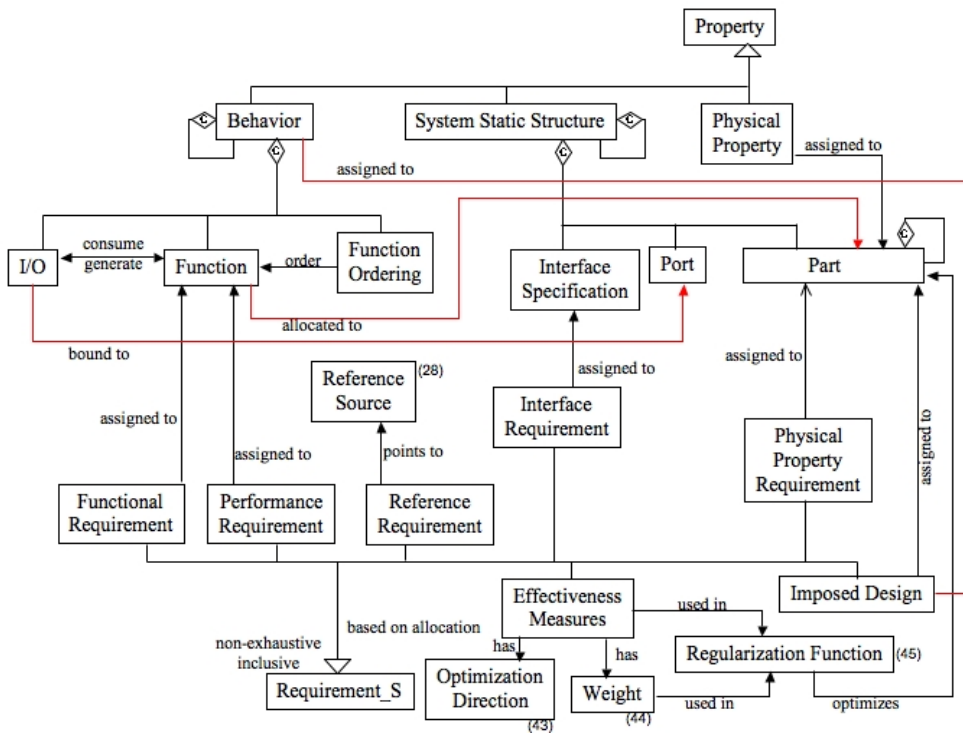
Vaatimustyyppin mukaan luokittelu on usein ongelmallista siksi, että vaatimus kuuluu usein useampaan luokkaan; esimerkiksi vaatimus voi olla sekä turvallisuusvaatimus että ergonomiavaatimus. Tietokannassa se ei ole ongelma, kunhan sallitaan usean vaatimustyyppin valinta. Mutta dokumenttipohjaisessa vaatimushallinnassa, erityisesti jos dokumentti toimii vaatimusten varastona, useaan vaatimustyyppiin kuuluminen on ongelma kopiointitarpeen takia, eli vaatimus täytyy olla kopioituna useamman otsikon alla. Jo tämä argumentti yksistään osoittaa dokumenttipohjaisen vaatimusmäärittelyn heikkouden.

Vaatimustyyppien valinnassa täytyy myös olla huolellinen. Esimerkiksi tämän luvun alun esimerkkiluettelosta löytyy vaatimustyyppi *liityntäraajapintavaatimukset*, joka on itse asiassa kohteen mukaista luokittelua (ks. luku Kohteen mukainen luokittelu), tai vaatimustyyppi *tuotantovaatimukset*, joka on itse asiassa elinkaaren vaiheen mukaista luokittelua (ks. luku Elinkaaren mukaan luokittelu). Tällainen epätarkka vaatimusten tyypittäminen aiheuttaa turhaan tapauksia, joissa vaatimus kuuluu kahteen vaatimustyyppiinluokkaan.

ISO 10303-233 [2012] -standardi esittää luokittelua kuvan 2, mukaisesti. Kuvassa 3 esitetään, miten AP233-standardin vaatimusluokittelu kohdentuu järjestelmän tietomallin mukaisiin suunnitteluartefakteihin.



Kuva 2. ISO AP233 -standardin esittämästä mallista vaatimusten luokitteluun [ISO 10303-233 2012].



Kuva 3. Esimerkki vaatimusten kohdentamisesta järjestelmän suunnitteluartefakteihin [ISO 10303-233:2012].

Lähteen mukaan luokittelu

Vaatimuksen lähde on joka tapauksessa tärkeä vaatimuksen metatieto, vaikka luokittelua ei sen perusteella tehtäisikään. Varsin usein herää kysymys, mistä vaatimus on peräisin, että voitaisiin tarkistaa sen pätevyys ja mahdollisesti kysyä tarkennuksia. Yleensä vaatimuksen lähteenä on jokin dokumentti, joka voi olla esimerkiksi standardi tai asiakkaan kanssa tehdyn sopimuksen liite. Lähdedokumenttia voidaan useissa tapauksissa päivittää, joten lähdeattribuutin lisäksi on syytä järjestää jäljitettävyysslinkki lähdedokumenttiin, jolloin dokumentin päivittäminen saattaa vaatimuksen epäilyksenalaiseksi.

Mutta lähettä voidaan käyttää myös luokitteluun. Varsinkin kartoitusvaiheessa vaatimukset on helpointa luokitella niiden lähteen mukaan, ja vielä hetken siitä eteenpäinkin suunnitteluinsinöörin voi olla helpointa selata vaatimuksia niiden lähteen perusteella. Vaatimusmäärittelydokumentin runkona lähteen mukaista luokittelua harvemmin käytetään.

Kohteen mukaan luokittelu

Usein on hyödyllistä liittää vaatimukseen attribuuttina sen kohde, kuten "koneen hytti" tai "puomi". Tässä havaitaan, että kun attribuutteja lisätään – kuten tässä "kohde" – herää kysymys, kuinka paljon tietoa lopulta attribuutteihin siirretään. Esimerkiksi tieto kohteesta on usein itse vaatimulauseessa, esimerkiksi: "Puomin maksimi paino saa olla X kg". Periaatteessa riittäisi lause: "Maksimi paino saa olla X kg", koska kohdeattribuutti kertoo vaatimuksen kohteen. Joka tapauksessa: on hyvä että vaatimulause on itsessään ymmärrettävä ja kohde-attribuuttia käytetään luokitteluun ja suodatukseen. Tämä vaatii että kehitettävän järjestelmän fyysinen arkkitehtuuri on suunniteltu, eli voidaan tunnistaa rakenneosat (osajärjestelmät ja komponentit), jotka on yksikäsitteisesti nimetty; nämä nimet tarjotaan sitten kohdeattribuuteiksi valintalistana, eli vapaamuotoisia tekstejä ei sallita. Täten kirjoitusvirheet eivät voi aiheuttaa vaatimuksen jäämistä pois, kun vaatimusmassasta suodatetaan näkyviin tietyn kohteen vaatimuksia.

Kohdeartefaktityypin mukaan luokittelu on myös mahdollista. Tällöin vaatimusta ei kohdisteta yksilöityyn artefaktiin, kuten "puomi", vaan artefaktityyppiin, "rakenneosa". Voidaan käyttää myös molempia, kuten "rakenneosa: puomi", joko yhdessä attribuuttikentässä tai kahdessa kentässä.

Tieteenalan mukaan luokittelu

Joissakin tapauksissa voi olla tarpeen luokitella vaatimuksia sen mukaan, liittyvätkö ne sähkötekniikkaan, ohjelmistotekniikkaan, mekaniikkaan, hydraulikkaan, pneumatiikkaan tai optiikkaan. Tavoitteena on, että vaatimusmassasta voidaan suodattaa näkyviin esim. mekaniikkasuunnittelijoille relevantit vaatimukset. Toisaalta jos järjestelmän fyysinen arkkitehtuuri määrittellään siten, että järjestelmä jaetaan tieteenalan mukaisesti mekaaniseen osajärjestelmään, sähköiseen osajärjestelmään jne., silloin sama tavoite saavutetaan kohteen mukaisessa luokittelussa (ks. luku Kohteen mukainen luokittelu). Näkymiä fyysiseen arkkitehtuuriin voi olla useita, joten joko tieteenalan mukaisiin osajärjestelmiin ei estä luomasta muitakin arkkitehtuurinäkymiä, kuten mekatroninen arkkitehtuuri. Siksi tieteenalan mukainen fyysinen arkkitehtuuri voidaan todennäköisesti luoda aina, joten erillistä tieteenalan mukaista luokittelua ei tarvita, koska tieteenalat paljastuvat tieteenalaperustaisina osajärjestelmäkohteina.

Prioriteetin mukaan luokittelu

Varsin usein vaatimukseen liitetään prioriteetti. Prioriteetti voi tarkoittaa joko toteuttamisjärjestystä (tärkeimmät tehdään ensin) tai kriittisyyttä, eli kuinka tärkeää on, että vaatimus toteutetaan tai ei toteuteta. Toteuttamisjärjestysprioriteetti voi olla kolmesta kymmeneen -portainen. Kriittisyys voi olla esimerkiksi "pakollinen", "suotava", 'toivottava' ja 'salliva'. Kriittisyys näkyy usein myös itse vaatimulauseessa;

pakollisuus ilmaistaan käskymuotona ("täytyy", "on oltava" jne.) ja suotavuus konditionaalimuotona ("pitäisi toimia", "olisi oltava" jne.).

Elinkaaren mukaan luokittelu

ISO/IEC/IEEE15288 määrittelee seuraavan elinkaarimallin:

- konseptivaihe
- tuotekehitysvaihe
- tuotantovaihe
- käyttövaihe
- tukivaihe
- käytöstä poisto.

Riskin arvioinnin standardi SFS-EN ISO 12100 [2010] taas esittelee seuraavat elinkaarivaiheet:

- kuljetus
- kokoonpano ja asennus; käyttöönotto
- asetusten teko; opettamalla ohjelmointi tai ohjelmointi ja/tai prosessin muuttaminen
- käyttötoiminta
- puhdistus ja kunnossapito
- vianetsintä/ongelman selvitys
- purku; käytöstä poisto.

Mitä elinkaarimallia tahansa käytetäänkin, vaatimukset on mahdollista luokitella sen mukaan, mitä elinkaaren vaihetta ne koskevat, eli missä vaiheessa vaatimus täytyy ottaa huomioon. Elinkaarivaiheitten lista auttaa myös kartoittamaan vaatimukset kattavasti.

Vaatimuksen toteuttajan mukaan luokittelu

Vaatimuksen luokittelun avainkysymys on, mihin ja kuka tarvitsee luokittelua. Toisin sanoen: kuka lukee vaatimuksen tehdäkseen sen perusteella johtopäätöksiä esimerkiksi suunnitteluun tai toteutukseen. Tällöin vaatimukset kannattaa luokitella toteuttajan mukaan.

Vaatimuksella voi olla enemmän kuin yksi toteuttaja, joten vaatimushallintatyökalussa toteuttajaa vastaava kenttä täytyy olla monivalintainen. Vaatimusmäärittelydokumenteissa luokittelu toteuttajan mukaan ei ole järkevää.

Rajoitteet vs. vaatimukset

Rajoitteet ovat vaatimusten erikoistapaus, tai toisaalta ne voidaan ilmaista siten, että ne eivät lauseopillisesti ole vaatimuksia, vaan ilmaisevat faktoja kehitettävänä

olevan järjestelmän ympäristöstä, infrastruktuurista sekä muista järjestelmistä, rakenteista ja tiloista, joiden yhteyteen tai sisään järjestelmää kehitetään. Varsinaiset vaatimukset taas kuvaavat kehitettävää järjestelmää tai kehittämisen prosessia. Rajanveto voi olla vaikeata, ja joskus voi olla turhauttavaa eritellä rajoitteet ja vaatimukset.

Rajoitteet ovat periaatteessa joustamattomia reunaehtoja toteutettavalle järjestelmälle. Esimerkkinä voisi olla kone, joka asennetaan alustalle, jonka kantokyky on 1 000 kg. Rajoitteena on 1 000 kg, mutta varsinaiseksi vaatimukseksi koneen painolle voidaan asettaa: ”pienempi tai yhtä suuri kuin 800 kg”. Tässä siis 1 000 kg on kiinteä ja 800 kg on valittu sopivalla varmuuskertoimella, että varauduttaisiin mahdollisiin tulevaisuuden lisälaitteiden painoon. Eli 800 kg on enempi valintaan kuin pakkoon perustuva vaatimus. Toisaalta vaatimuksia voisi olla kaksi: ”Koneen kokonaispaino tulevaisuuden lisälaitteineen täytyy olla pienempi kuin 1 000 kg” (ja tämä merkitty pakolliseksi vaatimukseksi) sekä tästä johdettu vaatimus ”Koneen paino saa olla enintään 800 kg”. Tässä kuitenkin ensimmäinen tapa, rajoite + vaatimus, on parempi tapa, sillä rajoite ilmaisee selkeästi vaatimuksen syyn: ”Alustan kantokyky on 1 000 kg”.

Rajoitteet voidaan esittää samassa formaatissa kuin vaatimuksetkin; vaatimusobjektissa vaaditaan ainoastaan attribuutti ”Rajoite”, jonka arvo on ”Kyllä”, kun kyseessä on rajoite.

Sidosryhmävaatimukset

Lähtökohtaiset vaatimukset tulevat sidosryhmiltä. Mahdollisia sidosryhmiä ovat esimerkiksi:

- asiakkaat
- viranomaiset
- asiakkaan asiakkaat ja järjestelmän käyttäjät
- järjestelmän kehittäjät
- järjestelmän toteuttajat (valmistajat)
- kouluttajat
- huoltohenkilöstö.

On huomattava, että yllä olevassa listassa on myös rooleja, jotka ovat pääasiassa vaatimusten käyttäjiä, kuten kehittäjät ja toteuttajat, mutta voivat myös olla niiden tuottajia. Joka tapauksessa oleellista sidosryhmävaatimuksissa on, että kehitetty ja valmistettu järjestelmä kelpuutetaan niitä vasten. Toisin sanoen ne täytyy pystyä erottamaan vaatimusmassasta siinä vaiheessa, kun järjestelmä esitellään hyväksyntää varten sidosryhmien edustajille. Toisaalta sidosryhmävaatimukset täytyy olla tiedossa tuotekehitysprosessin alkupäässä, sillä kun järjestelmävaatimuksia luodaan, peruseriaate on, että jokaista sidosryhmävaatimusta täytyy vastata yksi tai useampia johdettuja järjestelmävaatimuksia, ja toisaalta järjestelmävaatimukset täytyy pystyä jäljittämään sidosryhmävaatimuksiin. Jos sidosryhmävaatimuksia ei

pystytä erottelamaan vaatimusmassasta, järjestelmävaatimusten kattavuutta ja jäljitettävyyttä ei kyetä osoittamaan.

- Sidosryhmävaatimukset voidaan erottaa järjestelmävaatimuksista yksinkertaisesti lisäämällä vaatimusobjektiin attribuutti ”Sidosryhmävaatimus”, joka saa arvon ”Kyllä”, kun kyseessä on sidosryhmävaatimus.

Kun puhutaan kelpuutuksesta tai todennuksesta, on syytä käyttää tarkkoja ilmaisuja, mitä kelpuutetaan tai todennetaan: Kun kelpuutetaan ja todennetaan sidosryhmävaatimuksia, silloin tarkistetaan, ovatko vaatimukset juuri ne, mitä sidosryhmät ovat esittäneet, ja ovatko ne hyvälaatuisia. Asiakkaalle luovutusta edeltävässä hyväksyntävaiheen kelpuutuksessa ei siis puhuta sidosryhmävaatimusten kelpuutuksesta, vaan järjestelmän kelpuutuksesta sidosryhmävaatimuksia vasten.

Prosessivaatimukset

Varsinkin ohjelmistojen turvallisuusvaatimukset ovat usein prosessivaatimuksia eikä itse ohjelmistoon kohdistuvia vaatimuksia. Taustalla on ajatus, että kun ohjelmisto tuotetaan laadukkailla prosesseilla, siitä tulee laadukkaampaa, ja siten turvallisempaa, kuin satunnaisilla menetelmillä tuotettuna. Koska vaatimukset kohdistuvat prosesseihin, voi olla tarpeen luokitella vaatimukset prosessivaatimuksiksi, että prosessista vastuussa olevat henkilöt ja organisaatiot voisivat löytää helposti heitä koskevat vaatimukset. Prosessivaatimukseen liittyy usein myös ajallinen aspekti: Ne täytyy ainakin tarkistaa ensimmäiseksi, sillä mitään tekemistä ei voi aloittaa ennen kuin on varmistettu, että tekeminen tehdään prosessivaatimusten mukaisesti.

Osa prosessivaatimuksista voi kohdistua todennukseen ja kelpuutukseen.

Luokittelun yleisiä sääntöjä ja ohjeita

Vaatimukset kuuluvat usein kahteen tai useampaan luokkaan; vaatimus voi esimerkiksi olla yhtä aikaa sekä ergonomiavaatimus että turvallisuusvaatimus. Täten vaatimusmäärittelydokumentti, jonka otsikointi on tehty vaatimustyyppin mukaisesti, ei kykene esittämään tätä seikkaa ilman kopiointia. Tämä jo yksistään herättää tarpeen tietokantapohjaiselle vaatimushallintatyökalulle, jossa vaatimuksia voidaan luokitella ja suodattaa attribuuttien avulla. Toisaalta useat vaatimushallintatyökalut on rakennettu siten, että niiden käyttöliittymä rohkaisee käyttäjää tekemään tekstinkäsittelydokumentin kaltaista vaatimusmäärittelyä, jossa on otsikot ja aliotsikot, ja varsinaiset vaatimukset ovat dokumentin kappaleita. Tällainen menettely on hyvä kompromissi puhtaan tietokantaratkaisun ja tekstinkäsittelydokumentin välillä, ja se toimii matalana kynnyksenä tekstinkäsittelypohjaisesta vaatimusmää-

rittelystä tietokantapohjaiseen vaatimusmäärittelyyn, mutta sen ongelmana on otsikoinnin kiinnittäminen, jolloin edellä mainittu ongelma vaatimuksen kuulumisesta kahteen tai useampaan luokkaan tulee helposti esille. Otsikointi tulisi kyetä valitsemaan sellaiseksi, että kukin vaatimus kuuluu vain yhden otsikon alle. Vaatimuksen lähde voisi olla mahdollinen otsikoinnin perusta, koska vaatimus on useimmiten peräisin vain yhdestä lähteestä; jos vaatimus on peräisin kahdesta lähteestä, normaalisti niiden sanamuodot ovat erilaisia, joten ne voidaan käsitellä erillisinä vaatimuksina; jos niiden sanamuodot ovat täsmälleen samat, silloin voi useimmissa tapauksissa olla mahdollista jättää toinen lähteistä pois; jos sekään ei ole mahdollista, silloin vaatimus on kannassa kahteen kertaan, mutta se on kuitenkin järkevää, koska ne ovat kuitenkin eri lähteestä, jolloin niiden tulevaisuuden versiohistoriat voivat mennä eri suuntiin. Vaatimuksen lähdetä käytetään kuitenkin harvoin vaatimusmäärittelydokumentin sisällysluettelon perustana.

Otsikointi voi palvella myös vaatimusten kartoituksessa, sillä etukäteen kiinnitetyt otsikot toimivat avainsanoina, joiden perusteella vaatimuksia kartoitetaan, eli etukäteisotsikointi estää unohtamasta vaatimuksia joukoittain.

Dokumenttipohjaisen vaatimusten selailun sijaan tietokantapohjainen vaatimushallintatyökalu tarjoaa mahdollisuuden selailla vaatimuksia erilaisten suodatettujen näkymien läpi. Samoin vaatimuskannasta voidaan luoda ei-editoitavia raportteja, jotka näyttävät täsmälleen samanlaisilta kuin tekstinkäsittelyperiaatteella tehdyt vaatimusmäärittelydokumentit. Raporttien etu on siinä, että vaatimukset, jotka kuuluvat esimerkiksi kahteen vaatimustyyppiluokkaan, voidaan huoletta esittää siinä kahteen kertaan, eli kummankin otsikon alla (jos otsikointi tehdään vaatimustyyppin mukaan), koska itse vaatimus on olemassa vain kertaalleen tietokannassa; raportti on vain näkymä vaatimukseen, eikä sen kautta editoida vaatimuksia.

Vaatimustyyppien, prioriteetin, vaatimuksen kohteen jne. valinnoissa on syytä käyttää esivalintalistoja, eli vapaamuotoista tekstiä ei sallita. Tämä varmistaa sen, että väärin kirjoitetut metatiedot eivät pudota relevantteja vaatimuksia pois tietokantahauista.

Ehdotus luokitteluksi

Tässä ehdotetaan luokittelua **toiminnallisiin vaatimuksiin ja ei-toiminnallisiin vaatimuksiin, vaatimuksen tyyppin, lähteen, vaatimuksen toteuttajan, prioriteetin, prosessin ja vaatimuksen kohteen** mukaan. Lisäksi ehdotetaan yhtä kenttää sellaisten vaatimusten merkitsemiseen, jotka viittaavat joihinkin lisävaatimuksiin, esimerkiksi jonkin standardin vaatimuksiin; kyseiseen kenttään merkitään viitattujen dokumenttien riittävät tunnistetiedot. Kyseessä ei ole siis lähdeviittaus, vaan **vaatimuksen tarkennusviittaus**. Näiden lisäksi vaatimukseen ehdotetaan liitettäväksi kenttä, johon voi kirjoittaa **vapaamuotoisen avainsanan**. Jos **sidosryhmävaatimuksia** säilytetään samassa taulussa kuin järjestelmävaatimuksiakin, silloin sidosryhmävaatimukset erotellaan järjestelmävaatimuksista yhdellä Kyllä/Ei-lipulla. Samoin **rajoitteet** voidaan erottaa varsinaisista vaatimuksista Kyllä/Ei-valinnalla.

Toiminnalliset vaatimukset erotellaan muista vaatimuksista Kyllä/Ei-lipulla.

Vaatimustyyppit valitaan siten, että ne eivät sisällä kohteita, esim. 'liityntäraja-pintavaatimukset' eivät ole yksi vaatimustyyppi. Vaatimustyyppiksi voidaan valita erikoisala, esimerkiksi seuraavasti:

- turvallisuus
- rikosturva
- luotettavuus
- luotettavuus
- inhimilliset tekijät (ergonomia)
- ympäristötekijät (kestävä kehitys)
- logistiikka
- liiketoiminta.

Työkalun täytyy sallia usean vaatimustyyppin valinta per vaatimus. Myös tyhjä valinta täytyy sallia.

Lähteeksi kirjoitetaan riittävän tarkka viite lähdedokumenttiin, että vaatimuksen alkuperäinen ilmaisu voidaan tarkistaa. On järkevää luoda kaksi kenttää lähdeä varten, varsinainen lähde ja lähteen tarkennus. Tämä siksi, että joissakin työkaluissa vaatimusten pikalajittelu helpottuu, kun lähdeä ei kirjoiteta yhtenä merkkijonona, vaan esimerkiksi seuraavasti:

- Lähde: Konedirektiivi 2006/42/EY
- Lähteen tarkennus: Liite 1; luku 1.3.7.

Tällä tavalla voidaan nopeasti lajitella näkyviin esimerkiksi vain Konedirektiivin asettamat vaatimukset järjestelmälle, ja toisaalta vaatimus voidaan tarkasti identifioida.

Vaatimuksen toteuttaja -kategoriaan valitaan henkilö- tai organisaationimen sijasta rooli, esimerkiksi seuraavasti (roolit ovat tapauskohtaisia):

- Ohjausjärjestelmän toimittaja
- Laatu- ja prosessivastaava
- Sähkö-CAD piirtäjä
- jne.

Toteuttajakenttä ei saa olla täysin vapaavalintainen, vaan projektin konseptointivaiheessa ennen vaatimusmäärittelyn alkua pyritään määrittelemään tuotekehitysorganisaation ja alihankintaverkoston roolit niin pitkälle kuin mahdollista. Näistä rooleista tehdään esivalintalista. Esivalintalista täytyy kuitenkin päivittää myöhemmin, kun verkoston rakenne selkiytyy suunnittelun ja toteutuksen edetessä. Esivalintalistalta täytyy pystyä valitsemaan useita rooleja yhtä vaatimusta kohden.

Prioriteetti voi olla esimerkiksi jokin seuraavista:

- **pakollinen** (vaatimusklauseissa on tällöin useimmiten käskymuodossa, "on oltava", "täytyy" jne.; englanninkielisissä teksteissä "must" tai "shall")
- **suotava** (vaatimusklauseet ovat tällöin useimmiten konditionaalissa; englanninkielisissä teksteissä "should"; voidaan esimerkiksi vaatia, että jos vaatimuksen tyyppi on *suotava*, vaatimuksen toteuttamatta jättäminen on perusteltava)
- **toivottava** (vapaaehtoinen, tulevaisuudessa toteutettava; ei tarvitse perustella, jos vaatimusta ei toteuteta)
- **salliva** (vaatimus ei tarkkaan ottaen ole vaatimus, vaan sallii jonkin vapauden, esimerkiksi "Vikaemulointia voi tehdä joko rakenneosatasolla tai osajärjestelmätasolla"; tämä prioriteettiarvo voi olla tarpeeton, sillä ainakin joissakin tapauksissa salliva virke voidaan lisätä, ja kannattaakin lisätä, varsinaiseen vaatimukseen esim. "Vikaemulointitestejä on tehtävä todennusvaiheessa; vikaemulointia voi tehdä joko rakenneosatasolla tai osajärjestelmätasolla").

Tässä pyritään välttämään tyypillisiä prioriteettiluokitteluja, kuten korkea, keskinertainen ja matala, sillä ne ovat relatiivisia ilmaisuja, ja niiden skaalautuminen tapahtuu ilman selkeää ohjeistusta hyvinkin eri tavoilla eri ihmisten ajatuksissa (insinööri, joka on tottunut määrittelemään turvallisuuskriittisiä järjestelmiä, voi jossakin ei-kriittisessä järjestelmässä määritellä kaikki vaatimukset matalalle prioriteetille, kun taas ei-kriittisten järjestelmien insinöörit voisivat käyttää prioriteettiarvojen koko dynamiikkaa).

Prioriteettiattribuutin tueksi voi ottaa Kyllä/Ei-valinnan, jolla kerrotaan, onko vaatimus **dominoiva** eli vaatimus, jonka vaikutus toteutukseen on olennaisen suuri. Tätä valintaa ei välttämättä osata tehdä vaatimusmäärittelyvaiheessa, vaan

vasta kun suunnittelu etenee ja huomataan, mitkä vaatimukset ovat merkittävimpiä suunnittelua ohjaavia vaatimuksia. Vaatimus voidaan merkitä oletusarvoisesti ei-dominoivaksi tai normaaliksi, tai valinta voidaan jättää tyhjäksi. Dominoiva vaatimus voi kuulua mihin tahansa edellä esitetystä prioriteettiiluokista; toivottavakin vaatimus voi olla sellainen, että sen toteutus aiheuttaa ratkaisevan muutoksen suunnitelmaan.

Kiireellisyysprioriteettia ei käytetä, sillä se voidaan hoitaa projektiprosessien puolella. Käytännössä se voisi tarkoittaa, että järjestelmäsuunnittelija luo tehtäviä, joissa jollekin suorittajalle annetaan tehtäväksi toteuttaa tehtäväkuvauksessa mainitut tai tehtävään linkatut vaatimukset. Eli vaatimusten suoritusjärjestys hoidetaan tehtävänannoissa. Toisaalta kiireellisyysprioriteetti voi olla sidosryhmävaatimuksissa mukana, jos sidosryhmien edustajat ilmaisevat kiireellisyysprioriteetteja. Se helpottaa järjestelmäsuunnittelijan tehtävänantojen luomista.

Prosessin mukaan luokittelua käytetään vaatimuksille, jotka ovat prosessivaatimuksia, eli eivät kohdistu suunnittelun kohteena olevaan järjestelmään, vaan prosesseihin, joilla järjestelmää kehitetään, todennetaan, kelpuutetaan, operoidaan, ylläpidetään ja poistetaan käytöstä. Tämä vaatii tuekseen edellä kuvatun *vaatimuksen kohde* -luokittelun tai jäljitettävyysslinkin kohteeseen, että vaatimus kohdistuisi oikean järjestelmän prosessiin; esimerkiksi kohde voisi olla ”SW-osajärjestelmä” (eli ohjelmisto) ja prosessi voisi olla ”todennusprosessi”. Tällöin tiedetään että vaatimus liittyy ohjelmiston todennukseen (eli esimerkiksi testaukseen tai analyysiin).

Prosessimalli täytyy olla tiedossa, kun prosessia käytetään vaatimusten luokittelussa. Prosessimallin mukaiset prosessit näytetään esivalintalistassa; vapaa-
muotoista tekstiä ei sallita.

Vaatimuksen kohteena voi olla esimerkiksi jokin seuraavista:

- toimija
- järjestelmä (kohteena oleva koko järjestelmä)
- rakenneosaa (osajärjestelmä tai jakamaton rakenneosaa)
- käyttötapa
- käyttötapaus
- käyttötapausakti
- järjestelmätoiminto
- portti
- vuoro
- järjestelmäparametri
- malli
- dokumentti.

Kohde ilmaistaan yksilöiden esim. seuraavasti: *Toimija: operaattori* tai *Järjestelmä: moottorinohjausjärjestelmä*; tai ilman artefaktityyppejä: *operaattori*, *moottorinohjausjärjestelmä*. Voidaan myös käyttää kahta erillistä kenttää, kohde (artefaktityyppi) ja kohteen tarkennus (yksilöity artefakti). Kohteen tarkennus yksilöi kohteen. Esivalintaa kannattaa käyttää sekä kohteessa että sen tarkennuksessa. Kohteen tarkennus ei tosin ole tiedossa, ennen kuin fyysinen arkkitehtuuri on suunniteltu, eli kohde-attribuutti valitaan vastaan arkkitehtuurisuunnitteluvaiheen

lopussa, kun vaatimuksia kohdennetaan osajärjestelmille ja rakenneosille. Esivalintalista tehdään käyttäen fyysisen arkkitehtuurikuvauksen nimityksiä osajärjestelmille ja rakenneosille. Kohteita voi yhdelle vaatimukselle olla useita, joten monivalinta täytyy sallia. Kahden eri kentän käyttö kohteelle ja sen tarkennukselle on tosin silloin hankalaa. Kohde ei ole relevantti silloin, kun kyseessä on prosessivaatimus.

Edellä olevassa listassa oletetaan, että toiminnallinen arkkitehtuuri esitetään käyttötapausina ja järjestelmätoimintoina, joilla kummallakin voi olla aliartefakteja. Käyttötapaus voidaan alimmalla tasolla jakaa atomisiin akeihin, kuten kuvassa 15. Tällöin vaatimuksen kohteena voi olla myös akti.

Vaatimuksen kohde on tarpeeton luokittelu, jos käytössä on työkalu, jolla jäljitettävyyden vaatimuksista suunnitteluartefakteihin voidaan toteuttaa linkkeinä. Tällöin tiettyyn artefaktiin liittyvät vaatimukset voidaan suodattaa näkyviin jäljitettävyyden linkkien kautta.

Vaatimuksen tarkennus on tarpeellinen kenttä silloin, kun halutaan nopeasti näkyviin esimerkiksi kaikki standardiviittaukset, että osataan hankkia tarvittavat standardit, joissa varsinaiset vaatimukset ovat. Esimerkiksi vaatimus voi olla ilmaistuna seuraavasti: *"On noudatettava SFS-EN 13849-1 [2008] -standardin vaatimuksia ohjausjärjestelmän turvallisuuden varmistamiseksi."* Tällöin *Vaatimuksen tarkennus* -kenttään kirjataan: *SFS-EN 13849-1 [2008]*. Tarkennusviittauksen voi periaatteessa tehdä ilman erityistä kenttää, esimerkiksi jäljitettävyyden linkeillä viiteluettelon.

Edellä kuvattujen luokittelujen lisäksi voidaan käyttää vapaamuotoista **avainsana**-kenttää, johon voi kirjata avainsanoja vapaamuotoisesti, kuitenkin niin, että avainsanakenttä tarjoaa avainsanoiksi aikaisemmin käytettyjä sanoja, siten että monivalintakin on mahdollinen. Vaatimushallintatyökalu ei välttämättä tarjoa tällaista toimintaa.

Edellä kuvattu jäsentely on suunniteltu erityisesti järjestelmävaatimuksille, mutta sitä voi käyttää myös sidosryhmävaatimuksille.

Esimerkkejä ehdotetun luokittelun mukaisista vaatimuksista

Seuraavassa on viisi esimerkkiä, jotka noudattavat edellä kuvattua luokittelutapaa. Kohde-attribuuttia ei ole käytetty, koska tässä vaatimukset oletetaan linkatuiksi kohteisiinsa jäljitettävyysslinkeillä. Sen sijaan avainsanoihin on lisätty karkeaa tietoa, mihin osajärjestelmään tai rakenneosaan vaatimus liittyy. Kaikki vaatimukset liittyvät saksinostimen turvallisuuteen.

Vaatus 1. Kosketukset liikkuviin osiin

Koneen liikkuvat osat on suunniteltava ja rakennettava niin, että kosketuksesta aiheutuvat ja onnettomuuksiin mahdollisesti johtavat riskit estetään, tai jos riskejä ei saada poistetuksi, ne on varustettava suojuksilla tai turvalaitteilla.

Lähde: Konedirektiivi 2006/42/EY	Lähteen tarkennus: Liite 1; luku 1.3.7	Tyyppi: Turvallisuus
Prosessi: Ei prosessivaatimus	Prioriteetti: Pakollinen Dominoiva: Kyllä	Tila: Hyväksytty
Ehdotettu kelpuutusmenetelmä: Tarkastus	Vaadittu kelpuutusmenetelmä:	Onko sidosryhmävaatimus: Kyllä
Onko rajoite: Ei Onko toiminnallinen: Ei	Avainsanat: suojuukset; turvalaitteet	Tarkennus:

Vaatus 2. Hallintalaitteiden loogisuus ja liikkeiden suuntien merkitseminen

SHN:n kaikkien liikkeiden suunnat on merkittävä selvästi hallintalaitteisiin tai niiden läheisyyteen opastustekstien tai kuvatunnusten avulla. Kaikki hallintalaitteet on järjestettävä toimimaan loogisesti mikäli mahdollista.

Lähde: SFS-EN 280+A2 [2009]	Lähteen tarkennus: Luku 5.7.3	Tyyppi: Turvallisuus, Inhimilliset tekijät (ergonomia)
Prosessi: Ei prosessivaatimus	Prioriteetti: Pakollinen Dominoiva: Ei	Tila: Hyväksytty
Ehdotettu kelpuutusmenetelmä: Vaadittu menetelmä (ks. erill. sarake)	Vaadittu kelpuutusmenetelmä: Silmämääräinen tarkastus ja toimintakoe	Onko sidosryhmävaatimus: Kyllä
Onko rajoite: Ei Onko toiminnallinen: Ei	Avainsanat: hallintalaitteet	Tarkennus:

Vaatus 3. Päätyrajapysäytys

Turvallisuuteen liittyvä luokan 0 pysäytys toteutettava päätyrajoilla ISO 13849-1 kohdan 5.2.1, ISO 12100-2 kohdan 4.11.3 sekä IEC 60204-1 kohtien 9.2.2, 9.2.5.3 ja 9.2.5.5 mukaan.

Lähde: ISO 13849-1	Lähteen tarkennus: Luku 5.1, Taulukko 8, ens. rivi	Tyyppi: Turvallisuus
Prosessi: Ei prosessivaatimus	Prioriteetti: Pakollinen Dominoiva: Ei	Tila: Hyväksytty
Ehdotettu kelpuutusmenetelmä: Testi, analyysi ja demonstraatio	Vaadittu kelpuutusmenetelmä:	Onko sidosryhmävaatimus: Kyllä
Onko rajoite: Ei Onko toiminnallinen: Ei	Avainsanat: turvallisuuteen liittyvä pysäytys	Tarkennus: ISO 13849-1 (luku 5.2.1), ISO 12100-2 (luku 4.11.3), IEC 60204-1 (kohdat 9.2.2, 9.2.5.3 ja 9.2.5.5)

Vaatus 4. Sallintakytkin oltava

Nostotason ja koko koneen siirtämiseen tarkoitetussa ohjainsauvassa täytyy olla sallintakytkin, joka estää tahattoman liikkeen silloin, kun ohjainsauvaa poikkeutetaan vahingossa.

Lähde: Riskin arviointi	Lähteen tarkennus: Vaara PHA_0014	Tyyppi: Turvallisuus, Inhimilliset tekijät (ergonomia)
Prosessi: Ei prosessivaatimus	Prioriteetti: Pakollinen Dominoiva: Ei	Tila: Hyväksytty
Ehdotettu kelpuutusmenetelmä: Testi, Analyysi	Vaadittu kelpuutusmenetelmä:	Onko sidosryhmävaatimus: Kyllä
Onko rajoite: Ei Onko toiminnallinen: Ei	Avainsanat: hallintalaitteet; turvatoiminto	Tarkennus:

Vaatus 5. Sovellusohjelmiston kehityksen elinkaarimalli

Turvallisuuteen liittyviin [rajoitetun käskynkannan] sovellusohjelmistoihin, jotka on tarkoitettu komponenteille, joille vaadittavat suoritustasot ovat PLr a...e, on sovellettava seuraavia perustoimenpiteitä:

- elinkaarimallin mukainen ohjelmistonkehitys, johon kuuluu todentamista ja kelpuutusta (ks. kuva 6 [standardissa SFS-EN ISO 13894-1 [2008]])

Lähde: SFS-EN ISO 13849-1	Lähteen tarkennus: Luku 4.6.3	Tyyppi: Turvallisuus
Prosessi: Vaatimushallinta, Arkkitehtuurisuunnittelu, Toteutus, Integrointi, Todennus, Kelpuus	Prioriteetti: Pakollinen Dominoiva: Ei	Tila: Hyväksytty
Ehdotettu kelpuutusmenetelmä: Suunnittelu-dokumenttien katselmointi	Vaadittu kelpuutusmenetelmä:	Onko sidosryhmävaatimus: Kyllä
Onko rajoite: Ei Onko toiminnallinen: Ei	Avainsanat: ohjelmisto	Tarkennus:

Näistä ainakin vaatimus 4 on syytä linkata jäljitettävyysslinkillä lähteeseensä ("Vaara PHA_0014"), sillä riskin arviointi saatetaan päivittää, jolloin vaatimus ei välttämättä enää ole pätevä. Ko. vaatimuksen avainsanoissa on myös sana "turvatoiminto", joka ilmaisee, että tämä vaatimus aiheuttaa tarpeen turvatoiminnon ("tahattoman liikkeen esto") suunnittelulle.

Lähdeluettelo (Osa II)

- Amaral, L.A.N. & Uzzi, B. 2007. Complex Systems – A New Paradigm for the Integrative Study of Management, Physical, and Technological Systems. *Management Science*, Vol. 53, No 7.
- Aracic, J. 2012. Deploying Systems Engineering – An introduction. Crescendo Technologies. <http://blogs.crescendo-technologies.com/en/deploying-systems-engineering-an-introduction/>. [Accessed 25.3.2013].
- Araujo, I. 2006. Critical Success Factors for ERP Deployments in International Federation for Information Processing. *Research and Practical Issues of Enterprise Information Systems*. Tjoa, A.M., Xu, L., Chaudhry, S. (eds.). Vol. 205, Boston: Springer, s. 319–324.
- Dahlberg, S. 1999. Message posted to INCOSE list server, October 1999.
- Eigner, M., Gilz, T. & Zafirov, R. 2012. Proposal for functional product description as part of a PLM solution in interdisciplinary product development. DESIGN 2012.
- Estefan, J.A. 2008. Survey of Model-Based Systems Engineering (MBSE) Methodologies INCOSE MBSE Initiative.
- Hubka, V. & Eder, W.E. 1984. *Theory of Technical Systems – A Total Concept Theory for Engineering Design*. Springer-Verlag.
- IEEE 830:1998. IEEE Recommended Practice for Software Requirements Specifications. IEEE Standards Association.
- INCOSE. 2007. Systems engineering vision 2020. International Council on Systems Engineering (INCOSE).
- ISO 10303-233. 2012 Industrial automation systems and integration – Product data representation and exchange – Part 233: Application protocol: Systems engineering.
- ISO/IEC/IEEE 29148. 2011. Systems and software engineering – Life cycle processes – Requirements engineering.
- Konedirektiivi 2006/42/EY. Euroopan parlamentin ja neuvoston direktiivi 2006/42/EY, koneista ja direktiivin 95/16/EY muuttamisesta (uudelleenlaadittu). Lähteen

tarkennus: Liite 1, luku 1.3.7. Liikkuvista osista aiheutuvat riskit. L 157/35, 9.6.2006. Lähteen tarkennus Luku 5.7.3.

Leino, S-P., Anttila, J-P., Heikkilä, J., Aaltonen, J. & Helin, K. 2012. PLM Impact Analysis Model – PIA. 9th IFIP WG 5.1 International Conference on Product Lifecycle Management, PLM 2012, 9–11 July 2012, Montreal, QC. IFIP Advances in Information and Communication Technology. Springer. Vol. 388 AICT, s. 501–511.

Munkvold, B.E. 1999. Challenges in IT implementation for supporting collaboration in distributed organizations. European Journal of Information Systems 8, s. 260–272.

OMG SysML. OMG Systems Modeling Language. <http://www.omgsysml.org/>.

Senglaub, M. 1996. Systems Engineering: A problem of perception. Proceedings of INCOSE.

SFS-EN 13849-1:2008. Koneturvallisuus. Turvallisuuteen liittyvät ohjausjärjestelmien osat. Osa 1: Yleiset suunnitteluperiaatteet.

SFS-EN 280 + A2:2009. Siirrettävät henkilönostimet. Suunnittelulaskelmat. Vakavuus. Rakenne. Turvallisuus. Tarkastukset ja testit. Lähteen tarkennus Luku 5.7.3.

SFS-EN ISO 12100:2010. Koneturvallisuus. Yleiset suunnitteluperiaatteet, riskin arviointi ja riskin pienentäminen.

Sheard, S. A., 2000. Three Types of Systems Engineering Implementation. Proceedings of the Tenth International Symposium of the International Council on Systems Engineering, Minneapolis, MN, July 2000.

Sheard, S. A. & A. Mostashari 2009. Principles of complex systems for systems engineering. Systems Engineering 12(4). Ulrich, K.T. & Eppinger, S.D. 2003. Product Design and Development. McGrawHill.

UML. Unified Modeling Language™ (UML®). OMG Formally Released Versions of UML®. OMG. <http://www.omg.org/spec/UML/>.

Vitech. 2011. A primer for model-based system engineering.

Wang, G. G. 2002. Definition and Review of Virtual Prototyping. Information Science in Engineering, Vol. 2, No. 3.

OSA III – Sovellusesimerkkejä

Esimerkkejä systeemisuunnittelun soveltamisesta

Systems Engineering -lähestymistapaa sovelletaan harvoin, jos koskaan, täysin standardien kuvaamalla tavalla. Lisäksi standardeissa on usein huomioitu tarve mukauttaa sovellettavia menetelmiä yrityksen, tuotteen ja toimintaympäristön mukaan. Tässä on esitelty lyhyet yhteenvedot muutamasta esimerkistä, joissa on tavalla tai toisella sovellettu Systems Engineering -ajattelua tai -menetelmiä käytännössä.

Jarmo Alanen, Peetu Valkama

Saksinostimen teknisen prosessin hallinta..... 136

Simo-Pekka Leino

Tuotekehitys ja virtuaalikatselmointi..... 170

Kari Karstila

Vaatimustenhallinta järjestelmähankinnassa..... 172

Juhani Viitaniemi, Peetu Valkama

Ajoneuvon ergonomiavaatimukset..... 186

Lähdeluettelo..... 218

Saksinostimen teknisten prosessien hallinta

Jarmo Alanen, VTT

Peetu Valkama, VTT

Sisältö

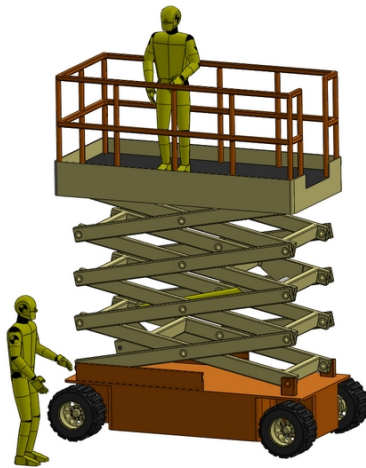
Johdanto	137
Järjestelmän fyysisen arkkitehtuurin malli	140
Systemaattiset prosessit	143
Tietomalleja	148
Järjestelmän päämalli.....	148
Malli vaatimusten jäljitettävyyteen	149
Toiminnallisen arkkitehtuurin konseptimalli.....	150
Järjestelmän fyysisen arkkitehtuurin eli struktuurin konseptimalli	151
Muita tietomalleja	153
Saksinostimen SE-suunnitteluympäristö	154
Konteksti.....	155
Suunnitteluartefaktit.....	156
Erikoisalat	159
Parametrit	161
Työkalut.....	161
Dokumentit	161
Päätason keskiosa: tekniset prosessit.....	164
Päätason keskiosa: muu osuus	166

Johdanto

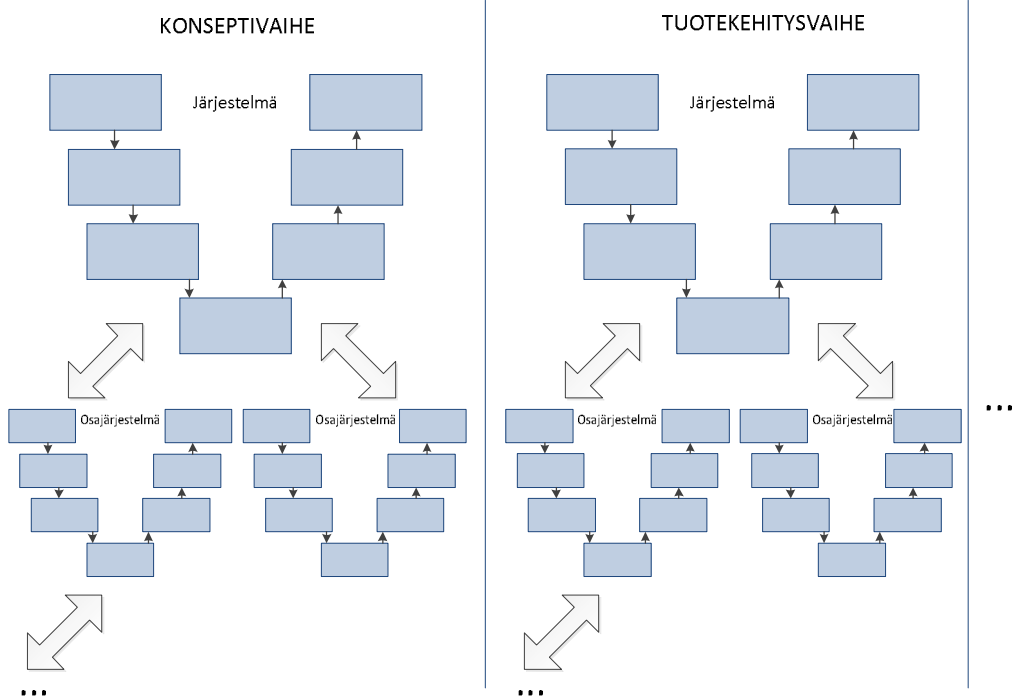
Tässä luvussa esitellään esimerkki yksinkertaisen koneen, saksinostimen (ks. kuva 1), SE-prosessin hallinnasta, lähinnä koskien teknisiä prosesseja, ei projekti-toimintaan, ostotoimintaan jne. liittyviä prosesseja. Hallintatyökaluksi on valittu Microsoft SharePoint, mutta periaatteessa mikä tahansa työkalu käy, joka tarjoaa dokumenttienhallinnan sekä tietokantamaisten listojen käytön, raskaan sarjan PLM-työkaluista puhumattakaan.

Saksinostin on monitekninen järjestelmä, sillä se sisältää sähköisen (tässä tapauksessa ohjelmoitavan) ohjausjärjestelmän sekä mekaanisen järjestelmän. Tässä esimerkkitapauksessa käytetään myös hydraulista järjestelmää. Saksinostimen voidaan kuvitella käyvän läpi kaikki ISO/IEC/IEEE 15288 [2008] -standardin elinkaaren vaiheet: konseptivaihe, tuotekehitysvaihe, tuotantovaihe, käyttövaihe, tukivaihe ja käytöstä poiston vaihe. Täten tekniset prosessit ovat periaatteessa kolmiulotteiset: saksinostimen omat tekniset prosessit, sen prosessista hierarkkisesti porautuvat osajärjestelmien vastaavat prosessit, ja tämä kaikki, ainakin osittain, eri elinkaarivaiheissa (ks. kuva 2).

Erityisesti virtuaalitekniikoita sovellettaessa konseptivaihe ja tuotekehitysvaihe voivat prosessimalleiltaan toistua lähes identtisinä: Konseptivaiheessa luodaan, käytetään ja kunnossapidetään, ehkä jopa harjoitellaan käytöstä poistoa, käyttäen virtuaalimalleja; tuotekehitysvaiheessa suunnitellaan sitten varsinainen fyysinen järjestelmä virtuaalimalleista saadun palautteen perusteella. Suunnitellusta laitteesta voidaan tehdä fyysinen prototyyppi, jota operoidaan, sen kunnossapitoa kokeillaan ja sen käytöstä poistoa voidaan harjoitella.



Kuva 1. Saksinostin.



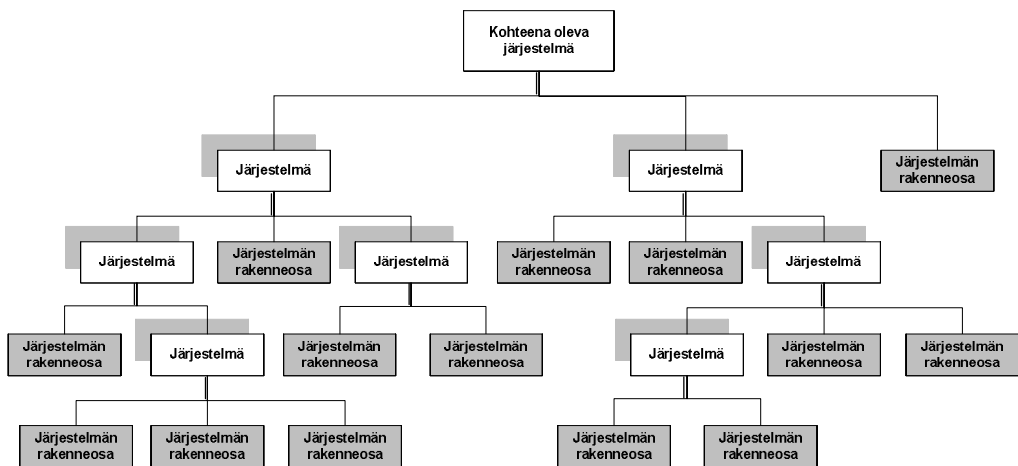
Kuva 2. SE-prosessien kolmiulotteisuus.

SE-prosessin hallintaa varten täytyy määrittää **systemaattiset prosessit** ja **systemaattinen tietomalli**. Kolmantena tärkeänä SE-prosessin onnistumistekijänä on **hyvin määritellyt projektiroolit**. Tietomallilla tarkoitetaan mallia siitä, mitä ja minkä sisältöisiä artefakteja SE-prosessissa tuotetaan ja miten ne liittyvät toisiinsa. Artefaktit ovat perinteisesti dokumentteja tai piirustuksia, yleensä siis tiedostoja, mutta käytännössä on tarve tarkempaan jyvitykseen, esimerkiksi vaatimusmäärittelydokumentin sijasta yksittäinen vaatimus olisi jäljitettävä ja versioitava artefakti. Täten jos yksi vaatimus muuttuu, vain siihen jäljitetyt artefaktit täytyy tarkistaa; dokumentin tapauksessa yksittäisen vaatimuksen muuttaminen aiheuttaa uuden version vaatimusmäärittelydokumentista, jolloin kaikki siihen jäljitetyt artefaktit täytyy tarkistaa, eli yksittäisen vaatimuksen muuttaminen johtaa periaatteessa koko suunnitelman tarkistamiseen.

Sekä prosessit että tietomallit riippuvat kohteena olevasta järjestelmästä. Siksi mitään yleispäteviä sitovia prosessimalleja ja tarkkoja tietomalleja ei voida esittää; sen sijaan toimialakohtaisia referenssimalleja voidaan luoda. Seuraavissa jaksoissa esitellään saksinostimen SE-prosessiin valittuja teknisiä prosessimalleja ja tietomalleja, mutta sitä ennen esitellään tapa, miten järjestelmän fyysinen arkkitehtuuri jäsenellään, sillä jäsentely on oleellinen sekä prosessimallien että tietomallien luomisessa ja ymmärtämisessä.

Järjestelmän fyysisen arkkitehtuurin malli

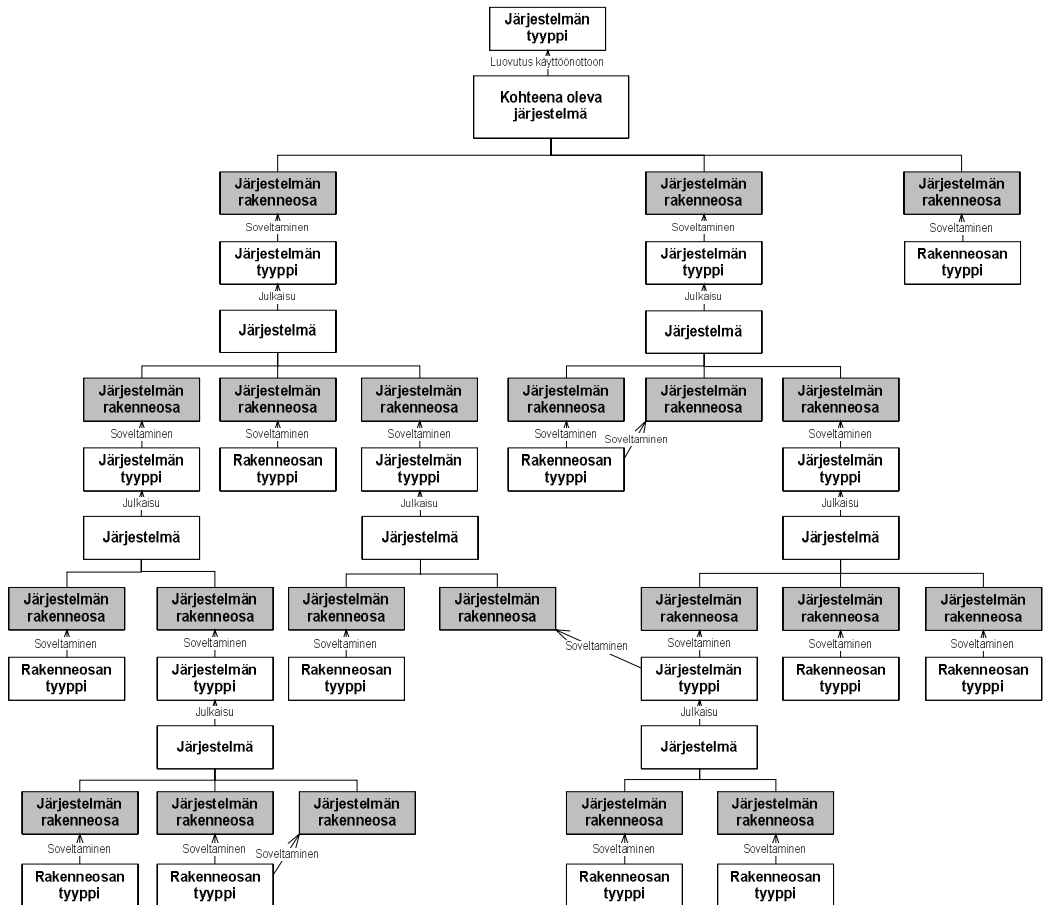
ISO/IEC/IEEE 15288 [2008] määrittelee järjestelmän struktuurin eli fyysisen arkkitehtuurin kahdella käsitteellä, järjestelmä ja järjestelmän rakenneosa: järjestelmä koostuu rakenneosista. Rakenneosat taas voivat koostua alemman tason rakenneosista, eli ovat siinä tapauksessa myös järjestelmiä (ks. kuva 3).



Kuva 3. Järjestelmän struktuuri ISO/IEC/IEEE 15288 [2008] mukaan.

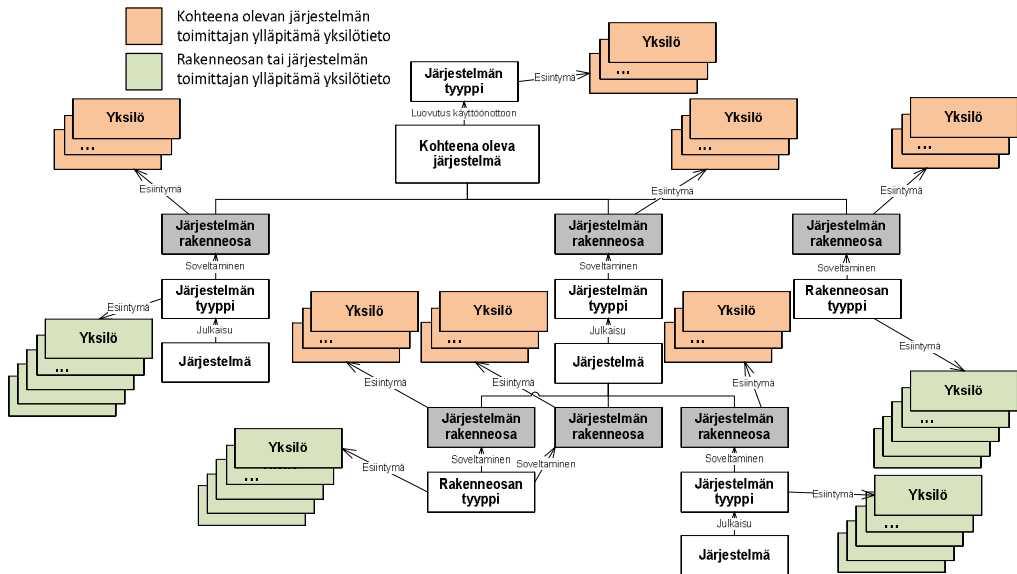
Diagrammi on piirretty uuteen muotoon kuvassa 4 siten, että kuvan 3 varjostetut laatikot on piirretty erikseen, eli ne näkyvät selkeämmin järjestelmän rakenneosina. Lisäksi kuvassa 4 järjestelmän ja sitä soveltavan järjestelmän rakenneosan välille on piirretty uusi laatikko, järjestelmän tyyppi, joka vastaa sitä kuvaa järjestelmästä, joka julkaistaan järjestelmää käyttävien, eli järjestelmäintegraattoreiden, käyttöön. (Kaikkea järjestelmän suunnittelutietoa ei siis useimmiten toimiteta integraattorille tai asiakkaalle.) Kyseessä on siis minimissään ohjelehti, jossa kuvataan järjestelmä, sen spesifikaatiot sekä integrointi-, käyttö-, kunnossapito- ja käytöstä poiston ohjeet. Järjestelmäintegraattori voi kirjastoida tällaisia tyyppitietoja, eli hän voi luoda kirjaston käytössä olevista tai potentiaalisista rakenneosista ja osajärjestelmistä. Kuva 4 paljastaa myös sen, että rakenneosa- tai järjestelmätyyppejä voi

käyttää useassa roolissa kohteena olevassa järjestelmässä; tämä seikka ei näy kuvassa 3.



Kuva 4. ISO/IEC/IEEE 15288 [2008] -standardin struktuurikuva uudelleen piirrettynä.

Kuvan 4 käsitteiden lisäksi tarvitaan vielä tietoelementtityyppi, joka tallentaa tietoa järjestelmän tai rakenneosan yksilöistä, jotka on toimitettu käyttöön. Tällaista tietoa ovat esimerkiksi takuutiedot tai luotettavuuden seuranta varten kerättävät kunnossapitotiedot. Kuvassa 5 on esitetty malli yksilötiedon huomioimiseksi.



Kuva 5. Kuvan 4 yläosa uudelleen piirrettynä siten, että yksilötiedot huomioidaan.

Kaikki edellä esitetyt kuvat struktuurista esittävät vain yhtä näkymää järjestelmän fyysisestä arkkitehtuurista. Esimerkiksi saksinostin voidaan ajatella koostuvan seuraavista ei-atomisista rakenneosista eli osajärjestelmistä:

- ohjelmoitava ohjausjärjestelmä
- sähköinen järjestelmä (esim. valot, merkinantotorvi jne. silloin kun niitä ei ohjata ohjelmoitavalla järjestelmällä)
- hydraulinen järjestelmä
- toimilaitteet
- teholähteet
- mekaaninen rakenne.

Tällainen rakenne palvelee tuotekehitysvaihetta, joka usein tapahtuu tieteenala-kohtaisesti. Esimerkiksi ohjausjärjestelmä voidaan haluta tilata alihankkijalta yhte-näisenä osajärjestelmänä.

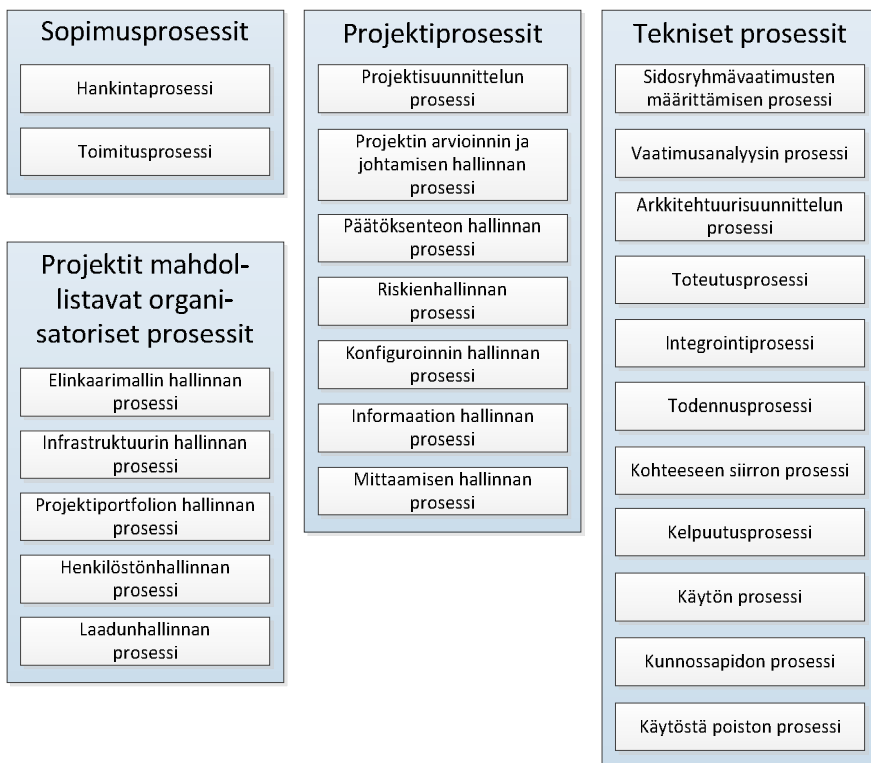
Toisaalta tuotantovaiheessa järkevämpi näkymä fyysiseen arkkitehtuuriin voisi olla esimerkiksi seuraava:

- työtaso
- nosterakenne
- alusta.

Tällainen mekatroninen rakenne on periaatteessa tosin mahdollinen myös tuotekehitysvaiheessa, jos mekatronisten osajärjestelmien rajapinnat on hyvin määritetty.

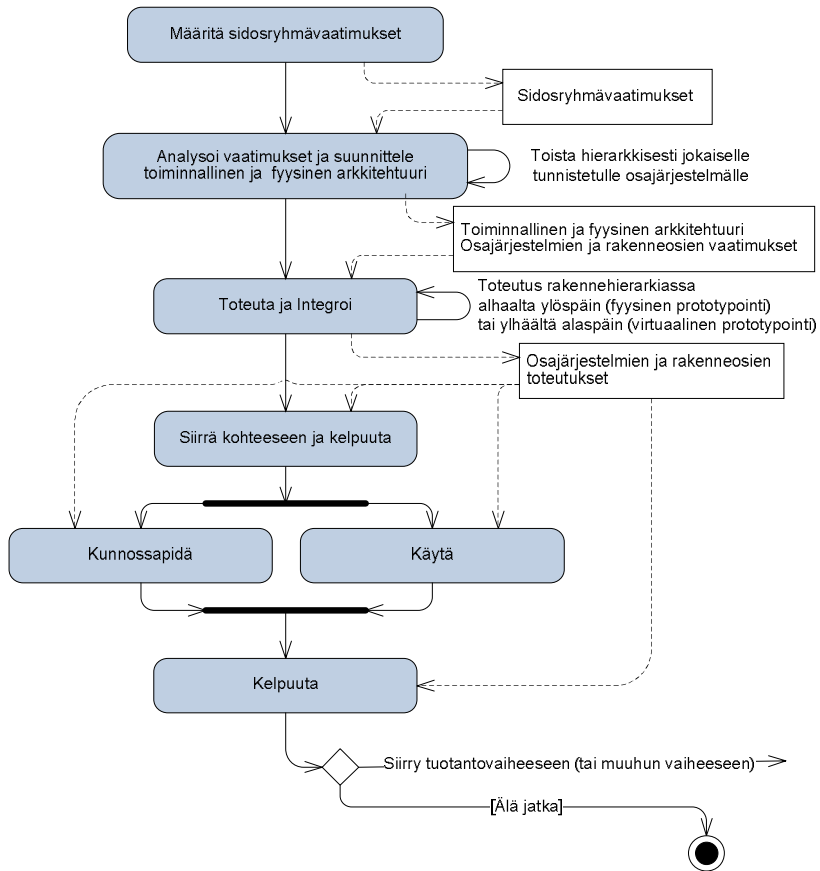
Systemaattiset prosessit

ISO/IEC/IEEE 15288 [2008] määrittelee joukon SE-prosesseja seuraavasti:



Kuva 6. ISO/IEC/IEEE 15288 [2008] SE-prosessit.

Saksinostinesimerkkiä varten ISO/IEC/IEEE 15288 [2008] -standardin teknisistä prosesseista luotiin räätälöity toimintokaavio, joka on esitetty kuvassa 7.



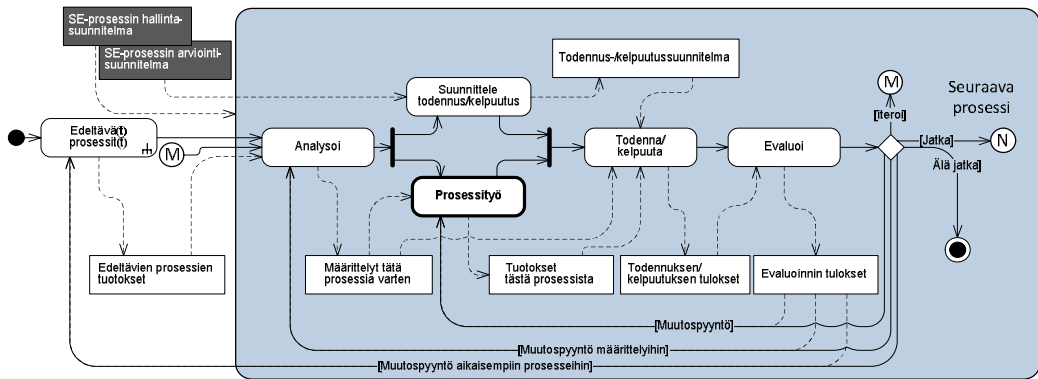
Kuva 7. Räätelöity toimintokaavio ISO/IEC/IEEE 15288 [2008] teknisistä prosesseista; kuvassa tuotekehitysvaiheen toimintokaavio.

Kuva 7 esittää nimenomaan tuotekehitysvaiheen toimintokaavion. Tuotekehitysvaihetta edeltävä konseptivaihe voi käydä läpi hyvin samankaltaisen vuon; tällöin toimitaan malleilla, jotka voivat olla fyysisiä prototyyppisiä tai virtuaalisia malleja.

Kuvan 7 toimintokaaviosta nähdään, että verrattuna ISO/IEC/IEEE 15288 [2008] -malliin vaatimusanalyysiprosessi ja arkkitehtuurisuunnitteluprosessi on yhdistetty yhdeksi prosessiksi, samoin toteutus ja integrointi⁴ ja samoin kohteeseen siirto ja kelpuutus. Lisäksi on huomattava, että tuotekehitysvaiheessa voidaan kokeilla käyttöä ja ylläpitoa; sen takia tarvitaan vielä toiseen kertaan kelpuutusprosessi ennen tuotantovaiheeseen siirtymistä. Todennusprosessi ei näy kuvan 7

⁴ Integrointi on toteuttamista osista kokoamalla.

toimintokaaviossa, sillä todennukseen liittyvät aktiviteetit on sisällytetty prosessien sisään kuvassa 8 esitettävän metamallin mukaisesti.



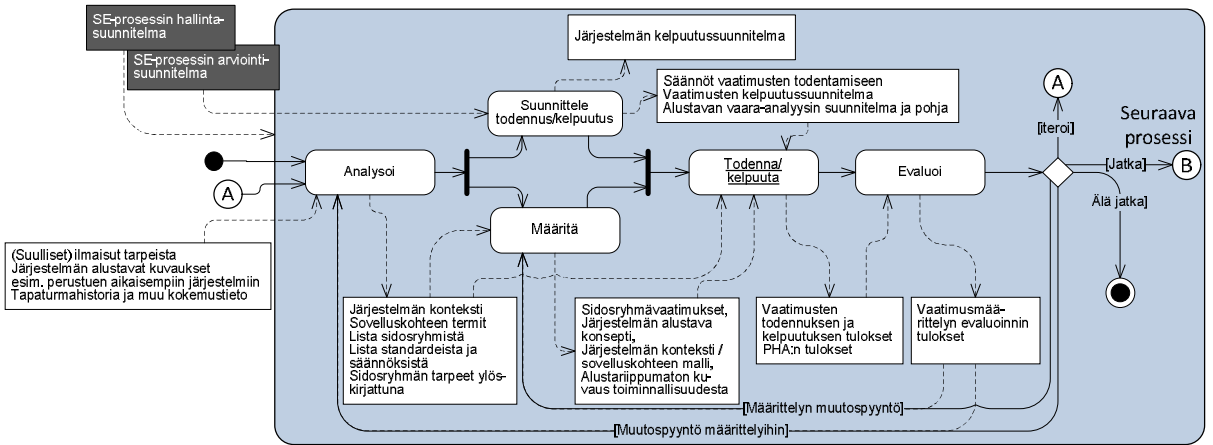
Kuva 8. Neljän ensimmäisen prosessin metamalli.

Kuvan 8 metamalli luotiin neljää ensimmäistä prosessia varten. Metamalli esittää, että jokaista varsinaista prosessityövaihetta edeltää analyysivaihe, jossa edeltävien prosessien tuotoksista luodaan määrittelyt tätä prosessia varten. Taustalla on ajatus, että määrittelyt (specifications) ja kuvaukset (descriptions) erotetaan selkeästi toisistaan siten, että määrittelyt ovat lähtötietoa ja kuvaukset ovat tulostietoa; edellisen vaiheen tulostiedoista (kuvauksista) syntyy seuraavan vaiheen lähtötietoja (määrittelyjä) analyysivaiheen kautta. Esimerkiksi, jos elektroniikkalaitetta suunniteltaessa kytkentäkaavion suunnittelijan tuotos on kytkentäkaavio ja siihen liittyvät kuvaukset, piirilevysuunnittelija ei käytä niitä sellaisenaan lähtötietoinaan, vaan välissä on analyysivaihe, jossa joku asiaan perehtynyt (ehkä piirilevysuunnittelija itse) määrittää piirilevyn kerrosten määrän ja muut piirilevyn suunnitteluun vaikuttavat attribuutit ja ohjeet. Ne kirjataan yhdessä kytkentäkaavion ja sen kuvauksen kanssa määrittelydokumenttiin, jonka perusteella piirilevy suunnitellaan.

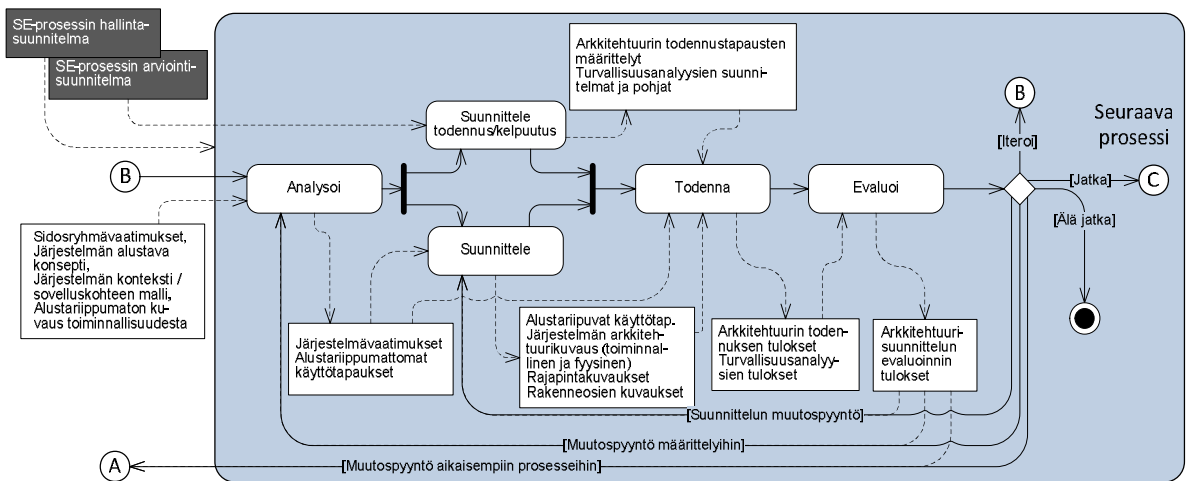
Jokaiseen prosessiin liittyy aina prosessissa syntyneiden tuotosten todennus tai joissakin tapauksissa kelpuutus. Siksi prosessiin liittyy oleellisesti toteutus- tai kelpuutus-suunnitelman teko. Kun todennus tai kelpuutus on tehty, todennuksen tai kelpuutuksen tulokset evaluoidaan jatkopäätöksen tekemiseksi. Metamallissa näyttäisi olevan kahteen kertaan sama jatkoreitti prosessin alkuun, muutospyynnön kautta ja iteroinnin kautta (M-kirjaimella merkitty reitti); niiden ero on kuitenkin selkeä: iterointi on suunniteltu, mutta muutospyyntöreitti on ei-toivottu reitti.

Seuraavassa esitellään kuvan 7 neljä ensimmäistä prosessia noudattaen kuvan 8 metamallia. Kuvat ovat iteseselittyviä⁵.

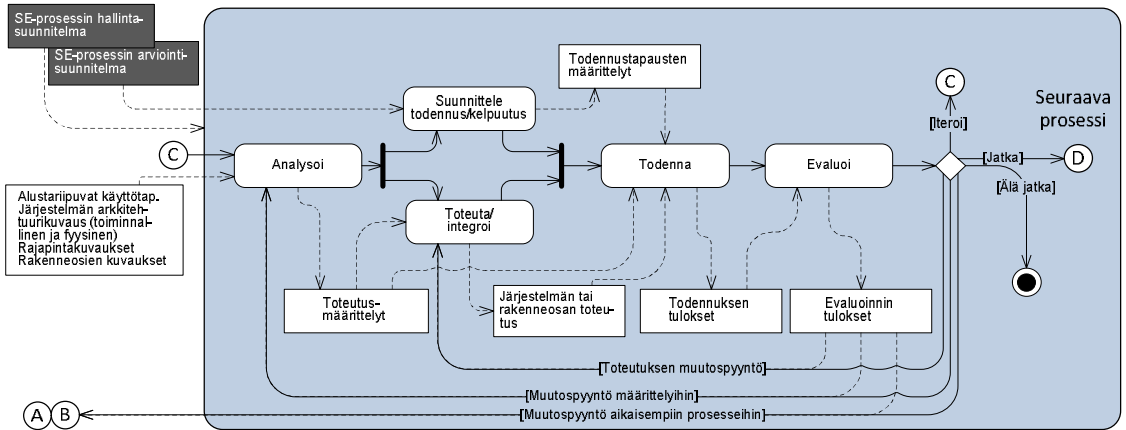
⁵ Kaavioiden esitystapa noudattaa pääpiirteissään UML/SysML-toimintokaavioiden notaatiota.



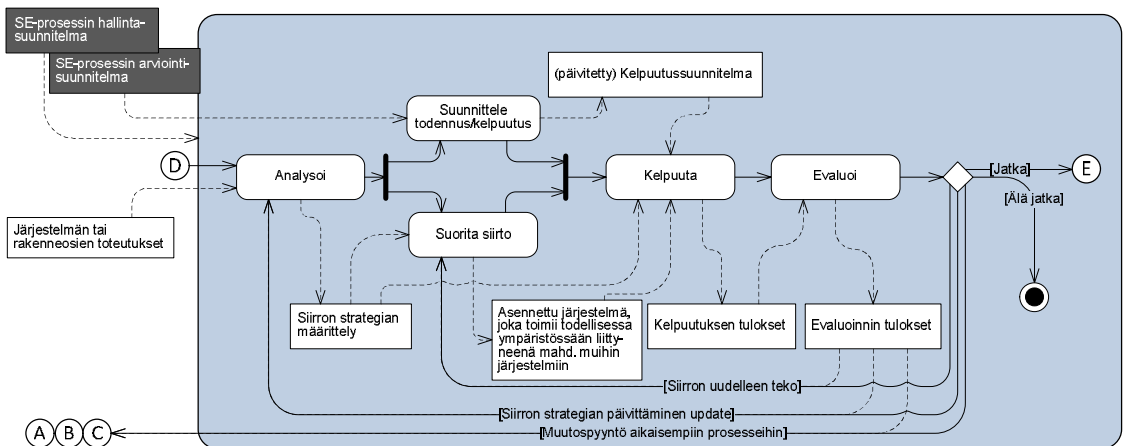
Kuva 9. Sidosryhmävaatimusten kartoituksen prosessi.



Kuva 10. Vaatimusten analyysin sekä toiminnallisen ja fyysisen arkkitehtuurin suunnittelun prosessi.



Kuva 11. Toteutuksen ja integraation prosessi.



Kuva 12. Kohteeseen siirron ja kelpuutuksen prosessi.

Kuvissa 9–12 pyörätiedot suorakulmiot kuvaavat aktiviteetteja ja tavalliset suorakulmiot aktiviteettien tuotoksia eli artefakteja. Seuraavassa luvussa (luku Tietomalleja) kuvatut tietomallit kuvaavat tarkemmin, mitä artefakteja SE-prosessissa syntyy ja mitkä ovat niiden väliset suhteet.

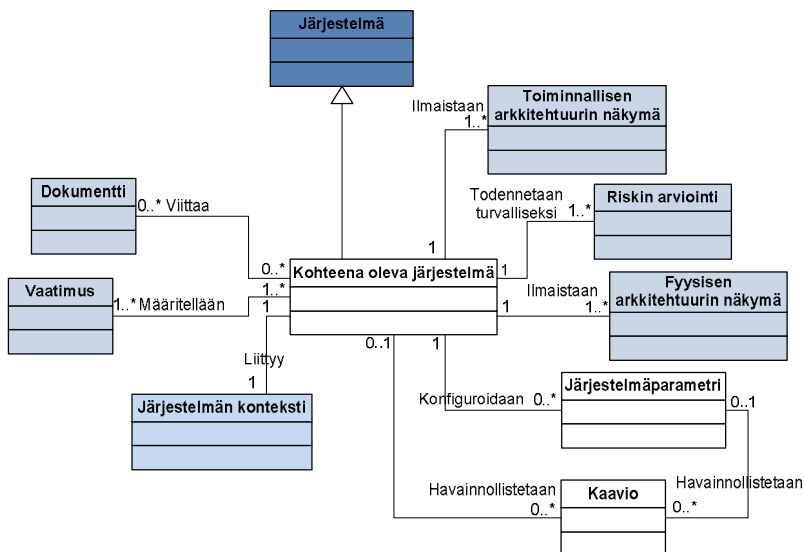
Tietomalleja

Seuraavassa on esitelty joidenkin tärkeimpien tietomallien konsepteja. Ne perustuvat TIKOSU-projektissa⁶ suunniteltuihin tietomalleihin⁷. Kuvat ovat iteseselittyviä, mutta joitakin ohjeita kuvien tulkitsemiseen annetaan, varsinkin niiltä osin, kun poiketaan TIKOSU-projektin tietomalleista.

Tietomallit ovat konseptimalleja; niissä esiintyvien artefaktityyppien sisältämiä attribuutteja ei tässä ole kuvattu.

Järjestelmän päämalli

Kuva 13 esittää kehityksen kohteena olevan järjestelmän konseptimallin päätasolla.



Kuva 13. Kehityksen kohteena olevan järjestelmän konseptimalli; päätasoa; vaaleansinisille artefaktityypeille luodaan tarkennetut alimallit.

⁶ Ks. www.hankegalleria.fi → TIKOSU [viitattu 12.11.2012].

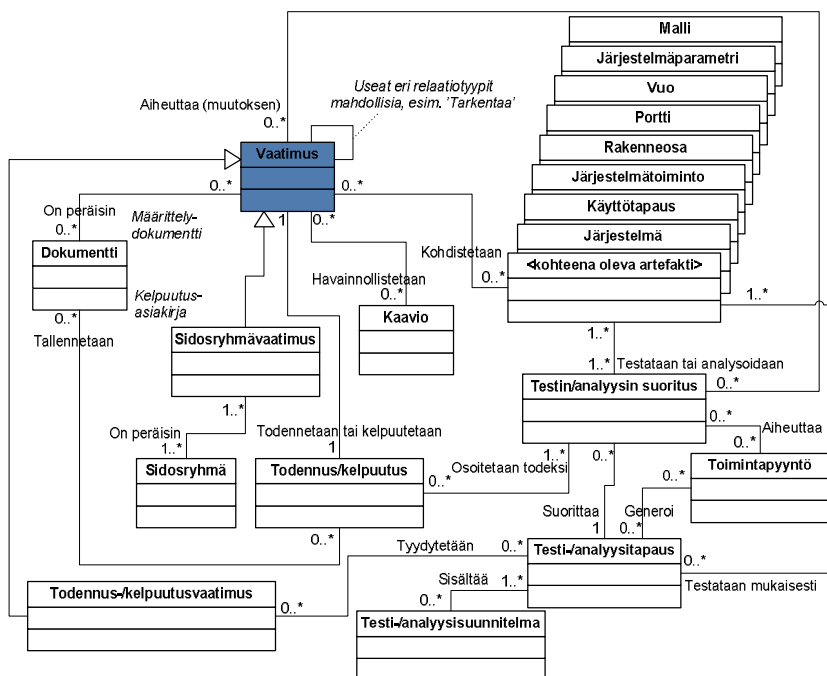
⁷ Kaavioiden esitystapa noudattaa pääpiirteissään UML-luokkakaavioiden notaatiota.

Kuvan 13 konseptimallin pääartefaktityyppi on *Järjestelmä*. Sen erikoistapauksena on *Kohteena oleva järjestelmä* eli järjestelmähierarkian ylin järjestelmä.

Järjestelmän tärkeimmät suunnittelutiedot ovat vaatimukset, toiminnallinen arkkitehtuuri ja fyysinen arkkitehtuuri. Sekä toiminnalliseen että fyysiseen arkkitehtuuriin tarjotaan tarvittaessa useita näkymiä.

Malli vaatimusten jäljitettävyyteen

Kuva 14 esittää konseptimallin vaatimusten jäljitettävyydestä.



Kuva 14. Vaatimusten jäljitettävyyden konseptimalli.

Vaatimukset voivat jäljittyä tarkennettuihin vaatimuksiin ja niistä varsinaisiin suunnitteluartefakteihin, jotka voivat olla toimintokuvauksia, CAD-malleja jne. Suunnitteluartefaktit todennetaan tai kelpuutetaan vaatimuksia vasten siten, että luodaan testi- tai analyysitapausten määrittelyt⁸, joiden perusteella testit tai analyysit tehdään ja raportoidaan. Testi- ja analyysitapauksiin itseensäkin voi liittyä vaatimuksia; esimer-

⁸ Testit ja analyysit eivät ole ainoat todentamisen ja kelpuutuksen menetelmät, muita ovat esimerkiksi katselmoinnit, demonstraatiot ja tarkastukset. Tässä yhteydessä testeistä ja analyysistä puhuttaessa tarkoitetaan kaikkia mahdollisia todentamisen ja kelpuutuksen menetelmiä.

kiksi turvallisuusstandardit määrittelevät usein (pakollisia) testi- ja analyysitapauksia. Kun testit tai analyysit on suoritettu ja raportoitu, varsinainen todennus tai kelpuus vaatimusta vasten raportoidaan artefaktissa *Todennus/kelpuus*.

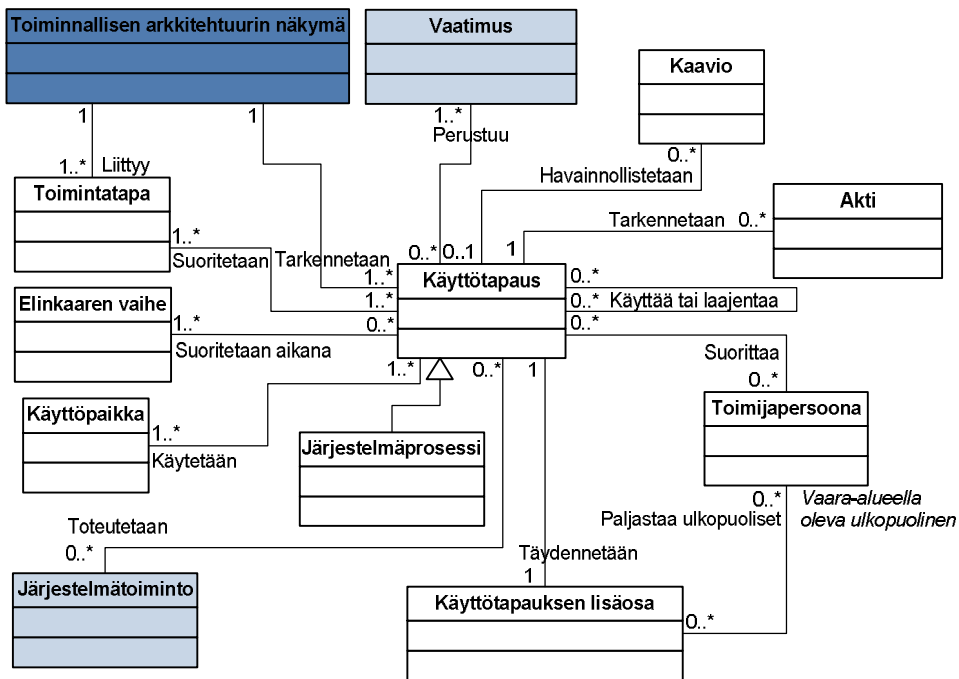
Toiminnallisen arkkitehtuurin konseptimalli

Toiminnallisen arkkitehtuurin konseptimalli perustuu UML- ja SysML-standardien mukaisten käyttötapauskuvausten käyttöön ylempillä abstraktiotasoilla ja toiminnallisten kuvausten (*Järjestelmätoiminto*) käyttöön alemmilla tasoilla. Käyttötapausten erikoistapaus on löydettävissä implisiittisesti itse käyttötapausistakin). Käyttötapausta laajennetaan lisäosalla, joka tarjoaa paikan turvallisuusanalyysien kannalta olennaiselle käyttötapaustiedolle (kuten altistumistaajuus ja käyttäjän pätevyysvaatimukset). Käyttötapausten ytimenä ovat peräkkäin suoritettavien aktien kuvaukset, joista jokainen tallennetaan omana artefaktinaan, että niistä identifioidut vaarat voitaisiin jäljittää tarkasti.

Käyttötapaukset perustuvat toiminnallisiin vaatimuksiin, mutta niiden sisältöön voivat vaikuttaa muutkin kuin toiminnalliset vaatimukset.

Järjestelmätoiminnon tarkempaa alimallia ei esitellä tässä.

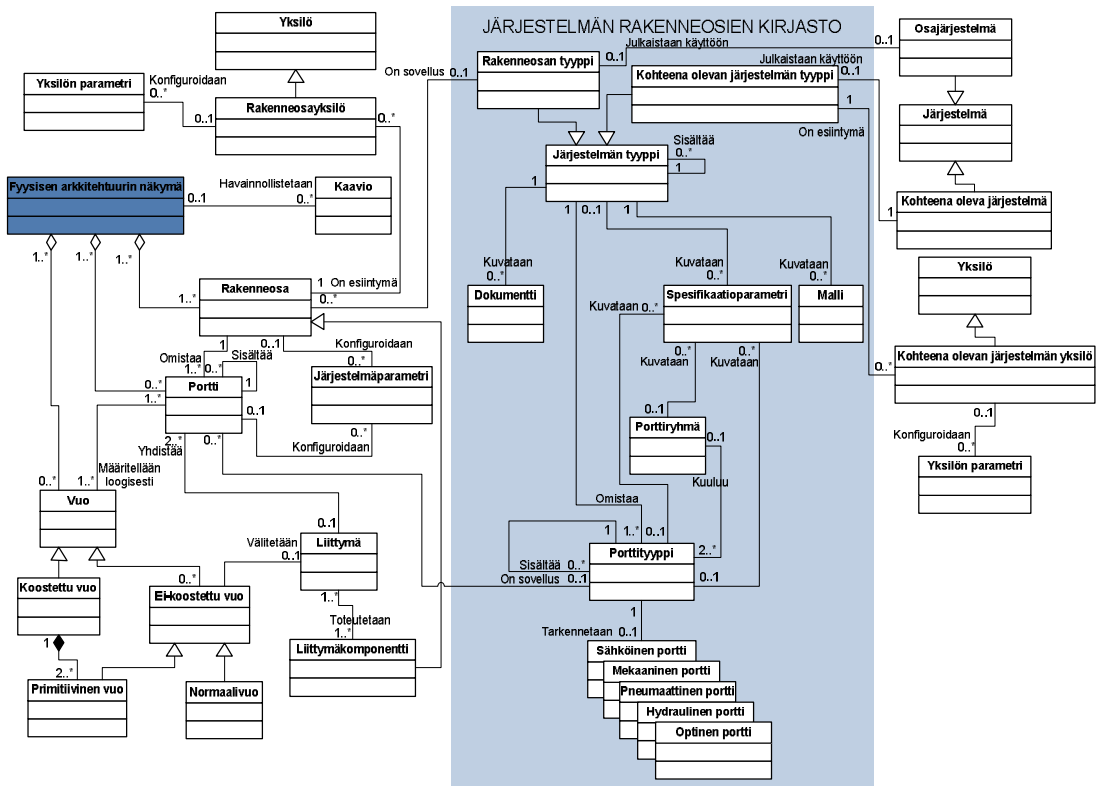
Toiminnallisen arkkitehtuurin konseptimalli on esitetty kuvassa 15.



Kuva 15. Toiminnallisen arkkitehtuurin konseptimalli.

Järjestelmän fyysisen arkkitehtuurin eli struktuurin konseptimalli

Seuraavassa on kuvattu fyysisen arkkitehtuurin konseptimalli (kuva 16).



Kuva 16. Järjestelmän fyysisen arkkitehtuurin konseptimalli.

Fyysisen arkkitehtuurin konseptimalli on selkeästi monimutkaisempi kuin edellä kuvatut muut konseptimallit. Erityisesti sen tekee hankalaksi tarve tukea eri tieteenaloja, kuten mekaanisia järjestelmiä ja ohjelmoitavia ohjausjärjestelmiä. Samoin hankaluutta aiheuttaa tarve tukea järjestelmän rakenneosatyyppien (eli kompo-

nentti- ja osajärjestelmätyyppien) kirjastointia. Tämä siksi että järjestelmän suunnittelutiedoissa ei tarvitsisi toistaa rakenneosien tietolehdistä löytyviä tietoja. (Esimerkiksi jos järjestelmässä olisi sata esiintymää jostakin komponenttityypistä, komponentin relevantit tietolehdistiedot pitäisi kopioida suunnittelutietoihin sataan kertaan.)

Fyysinen arkkitehtuuri koostuu rakenneosista, joilla on portteja, joita voidaan kytkeä toisiinsa liittymällä, joka välittää yhden tai useamman vuon. Kaksi porttia voidaan liittää toisiinsa vain jos niiden voiden spesifikaatiot ovat yhteensopivat. Liittymä on abstrakti käsite, joka toteutetaan fyysisesti yhdellä tai useammalla liittymäkomponentilla, jotka taas ovat rakenneosien erikoistapauksia. (Esimerkiksi anturin lähtöpiste ja mittauslaitteen sisääntulopiste liitetään toisiinsa liittymällä, joka fyysisesti toteutetaan kaapelilla ja liittimillä. Liittymällä on kuitenkin esiintymänsä piirroksissa; ne kuvataan sähköjärjestelmien tapauksissa viivoilla ja kytkentäpisteillä, jotka sähköpiirtäjä realisoi rakenneosiksi eli kaapeleiksi, jatkoskohdiksi ja liittimiksi.)

Vuot voivat olla koostettuja voita tai yksittäisiä. Koostettu vuo koostuu useasta primitiivivuosta. Tarve tällaiseen jakoon tulee erityisesti sähköisistä ohjausjärjestelmistä. Esimerkiksi kvadratuurien kooderilla on kaksikanavainen lähtöaste; kanavat ovat 90°:n vaihesiirrossa toisiinsa nähden; vaihesiirron etumerkki määrittelee enkooderilla mitattavan kohteen pyörimissuunnan ja pulssien määrä tai taajuus kohteen kiertymiskulman tai nopeuden; kummatkin kanavat eli primitiivivuot tarvitaan varsinaisen mittaussignaalin eli koostetun vuon tuottamiseen. Toinen esimerkki sellaiset absoluuttiset digitaaliset anturit, joissa tieto välitetään rinnakkaismuotoisesti.

Rakenneosista on kentällä useita yksilöitä. Yksilöt voivat olla kuitenkin konfiguroinniltaan erilaisia. Siksi yksilöistä tallennetaan myös parametritietoa. Yksilöihin liittyy myös muita tietoja, joita ei ole mukana kuvan 16 mallissa, kuten kenttäpaulutteena saatava vikatieto ja huoltotapahtumat.

Rakenneosat liittyvät rakenneosatyyppeihin, jotka on kirjastoitu. Ne tarjoavat tarpeellisen tiedon rakenneosien, niin atomisten kuin ei-atomisten (eli osajärjestelmien), järjestelmään integrointia varten. Niissä on siksi tieto myös rakenneosatyypin rajapinnoista. Tieto voidaan välittää perinteisissä tietolehdistä tai yksilöllisesti tallennetuissa spesifikaatioparametreissa tai virtuaalimalleina. Rakenneosien portit esitellään erikseen. Portit voivat kuulua porttiryhmiin.

Kohteena oleva järjestelmä (eli pääjärjestelmä) julkaistaan lopulta myös kyseisen järjestelmän rakenneosien kirjastossa. Se ei periaatteessa ole tarpeellista, mutta rakenneosakirjasto tarjoaa hyvän tallennuspaikan julkaistulle järjestelmälle. Kun lopulta toteutetaan yksi tai useampia kohteena olevan järjestelmän yksilöitä⁹,

⁹ Tietomallissa kardinaliteettina on silti 0..*, sillä välttämättä järjestelmää ei lopulta toteuteta, ja toisaalta tuotekehitysvaiheen aikana, jolloin suunnittelutiedot luodaan, yhtäkään järjestelmää ei ole vielä toteutettu, joten hyvin tehty työkalu hälyttäisi koko tuotekehitysprosessin ajan, että suunnittelutietojen relaatiot eivät ole kunnossa.

yksilöt voidaan jäljittää julkaistuun tietoon, jolloin voidaan varmistaa jäljitettävyys yksilön ja sitä vastaavan teknisen ja muun dokumentaation, jopa käyttöohjeiden välille. Yksilöt voivat poiketa toisistaan yksilöityjen parametrien mukaisesti. Yksilökohtaisissa parametreissa voidaan myös kertoa, mitä järjestelmän konfiguraatiota tämä yksilö noudattaa, esimerkiksi jos järjestelmiä tuotetaan erilaisilla kokoonpanoilla tai toiminnallisuuksilla.

Muita tietomalleja

Edellä kuvatut tietomallit ovat esimerkkejä tarvittavista tietomalleista. Näiden lisäksi tarvitaan tarkempi malli esimerkiksi järjestelmätoiminnoista, riskien arvioinnista ja sähköisten osajärjestelmien kommunikaatiojärjestelmistä. Tällaisia malleja esitellään esimerkiksi lähteessä [Alanen et al. 2011]. Kyseisessä lähteessä esitellään myös ISO STEP-standardiperheeseen kuuluva AP233 tietomalli [ISO 10303-233:2012] sekä saksalaisen autoteollisuuden tietomallit (kuten MSRSYS ja MSRSW), joita voi myös käyttää perustana omien tietomallien luonnissa.

Saksinostimen SE-suunnitteluympäristö

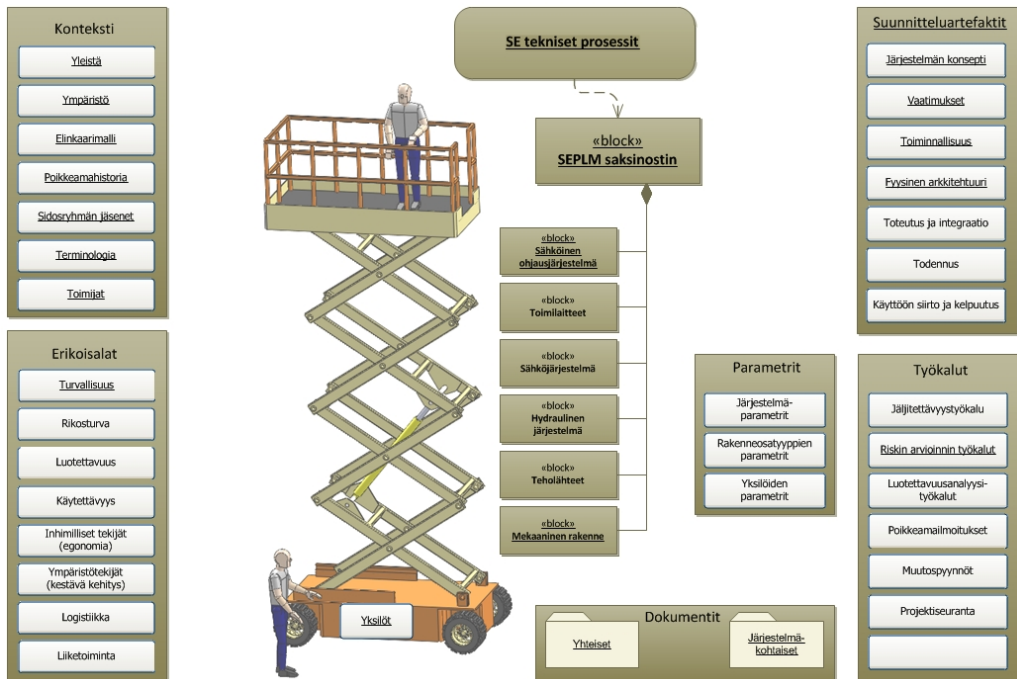
Saksinostimen suunnittelua varten luotiin SE-suunnitteluympäristö, jonka avulla edellä kuvattuja teknisiä prosesseja ja tietomallia voidaan hallita ja käyttää. Suunnitteluympäristö toteutettiin Microsoft SharePoint -alustalle. Prosessikaaviot ja käyttöliittymä piirrettiin Microsoft Visio -piirustustyökalulla, josta ne talletettiin HTML-muodossa SharePoint-alustaan. Tietomalli toteutettiin SharePoint-listoina luvussa Tietomalleja esitettyjen tietomallien mukaisesti. Listojen väliset relaatiot toteutettiin relaatiolistoilla, jotka mahdollistavat monesta-moneen relaatiot sekä epäilyksenalaisuuslippujen käytön. Tällaiset jäljitettävyysominaisuudet ovat välttävät; parempien ominaisuuksien (kuten jäljitettävyysetketjujen visualisointi) toteutus vaatisi lisäohjelmointia. Varsinaisissa PLM- ja ALM-työkaluissa tällaiset ominaisuudet ovat, tai ainakin pitäisi olla, luonnostaan. Tässä esimerkissä haluttiin kuitenkin demonstroida, miten pitkälle päästään työkaluilla, jotka ovat jo ennestään koneenrakentajayrityksessä käytössä, kuitenkin niin, että otetaan askel eteenpäin perinteisestä tekstinkäsittely- ja taulukkolaskin-pohjaisesta suunnittelusta.

Saksinostimen SE-työpöytä on esitetty kuvassa 17.

Käyttöliittymä tarjoaa pääsyn seuraaviin SE-osa-alueisiin:

- **Konteksti:** Järjestelmän toimintaympäristö, elinkaarimalli, kokemustieto (erityisesti poikkeamahistoria), sidosryhmien erittely, sovellusalaan liittyvä terminologia sekä järjestelmän toimijat
- **Suunnitteluartefaktit:** Tässä kategoriassa esitellään varsinaiset suunnittelutyön tulokset. Kuvasta puuttuu tuotanto-, käyttö-, kunnossapito- ja käytöstäpoiston vaiheiden artefaktit.
- **Erikoisalat:** Tietty osa-alueet on eritelty omiksi näkymiksi, kuten turvallisuus ja luotettavuus. Tämä siksi, että tällaiset osa-alueet vaativat usein erityisiä prosesseja, työkaluja ja henkilöresursseja. Tällaisia erikoisaloja voi olla useitakin ja ne riippuvat sovelluksesta.
- **Parametrit:** Järjestelmään, sen rakenteeseen ja sen yksilöihin liittyvät parametrit
- **Työkalut:** Kaikkiin elinkaaren aikana käytettäviin työkaluihin pääsee suoraan tästä valikosta. Työkalut avautuvat muualtakin, mutta tästä niihin pääsee oikeitsem.

- **Dokumentit:** Dokumentteihin pääse käsiksi tästä, mutta myös suoraan SharePoint dokumenttikirjastosta.
- **Keskiosa:** Tästä pääsee teknisiin prosesseihin sekä osajärjestelmiin. Kuvassa oleva osajärjestelmärakenne on valittu tuotekehitykseen sopivaksi. Tuotantoa varten näkymä olisi erilainen. Yksilöt-painikkeesta pääsee selaimaan kentällä olevia järjestelmän yksilöitä.



Kuva 17. Saksinostimen Systems Engineering -työpöytä.

Konteksti

Esimerkiksi painettaessa Elinkaarimalli-painiketta konteksti-kategoriassa avautuu seuraavanlainen tietosivu (kuva 18):

KONEEN KOKO ELINKAAREN ERI VAIHEIDEN KUVAUS

Tässä yhteydessä käytetään sekä ISO/IEC TR 24748-1 -standardin elinkaarimallia sekä ISO 12100 riskinarvioinnin standardin elinkaarimallia. Edellistä käytetään järjestelmän SE-prosessin ylempään tason elinkaarimallina. Jälkimmäistä käytetään turvallisuusprosessin yhteydessä.

Taulukko 1. ISO/IEC 24748-1 -standardin elinkaarien vaiheet

Title	Explanation	Reference	link_to_process
Konseptivaihe	Konseptointivaiheella tarkoitetaan saksinostimen kehityksen sitä vaihetta, jossa erilaisia konsepteja kokeillaan virtuaalisilla malleilla. Jopa huoltotyötä voidaan kokeilla ao. malleilla.	ISO/IEC TR 24748-1:2010	Linkki konseptivaiheen prosessiin
Tuotekehitysvaihe	Saksinostimen tapauksessa tuotekehitysvaihe on vaihe, jossa järjestelmä varsinaisesti suunnitellaan käyttäen konseptivaiheen ivaalluksia, jotka voidaan kirjata järjestelmävaatimuksiksi tähän tuotekehitysvaiheeseen. Tuotekehitysvaiheen tuotoksena on piirustukset ja muut ohjeet, jonka perusteella kone voidaan rakentaa.	ISO/IEC TR 24748-1:2010	Tuotekehitysvaiheen prosessi
Tuotantovaihe	Saksinostimen tapauksessa tuotantovaihe tarkoittaa saksinostimen rakentamista tilatuista komponenteista sekä tuotannossa tapahtuvaa testausta.	ISO/IEC TR 24748-1:2010	
Käyttövaihe	Saksinostimen tapauksessa käyttövaiheella tarkoitetaan vaihetta, jossa kone on käytössä loppukäyttäjällä, esim. varastossa. Käyttövaiheeseen liittyvät myöskin toiminnalliset tarkistukset.	ISO/IEC TR 24748-1:2010	
Tukivaihe	Tukivaiheella tarkoitetaan saksinostimen tapauksessa huoltotilanteita ja määräaikaistarkistuksia.	ISO/IEC TR 24748-1:2010	
Käytöstä poisto	Saksinostimen tapauksessa käytöstä poisto tarkoittaa vaihetta, jossa kone puretaan metalli- ym. romuksi.	ISO/IEC TR 24748-1:2010	

➤ Add new item

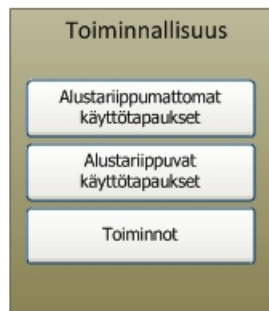
Taulukko 2. ISO 12100 standardin elinkaaren vaiheet

Title	Explanation	Reference
Kuljetus	Taulukko 2. ISO 12100 standardin elinkaaren vaiheet	ISO 12100:2010

Kuva 18. Esimerkki: elinkaarivaiheitten tietosivu (vain yläosa sivusta näkyvissä).

Tietosivu on toteutettu Wiki-sivuna, johon on tuotu näkyviin elinkaarivaiheet dynaamisena SharePoint-listana. Toisin sanoen, jos elinkaarivaiheita editoidaan muualla, muutokset näkyvät myös tällä sivulla. Ja toisaalta elinkaarivaiheita voi lähteä editoimaan tältä sivulta.

Suunnitteluartefaktit



Kuva 19. Toiminnallisuuden valikko.

Toiminnallisuuden suunnittelu kehittyi valikon mukaisesti alustariippumattomista käyttötapauksista järjestelmien toimintoihin alustariippuvien käyttötapauksien kautta. Kun valitaan alustariippumattomat käyttötapaukset, saadaan seuraava tietosivu (kuva 20).

Alustariippumattomat käyttötapaukset								
<p>Alustariippumattomat käyttötapaukset kuvaavat järjestelmän halutun toiminnallisuuden ottamatta kantaa, millaisella laitealustalla toiminnallisuus toteutetaan. Käyttötapauksukuvauksissa ei siksi ole virkkeitä, joihin mainitaan esimerkiksi jokin käyttöliittymälaite: "Operaattori ohjaa ohjauksussivellä puomia...", vaan kuvaus tehdään esim. tähän tapaan, "Operaattori ohjaa puomia käyttöliittymästä, josta voidaan ohjata kolmea vapausastetta, X, Y ja Z". Tämä antaa mahdollisuuden suunnittelijoille harkita esimerkiksi painonappeja ohjauksuun sijasta.</p> <p>Seuraavassa taulukossa on lueteltu saksinostimen alustariippumattomat käyttötapaukset</p>								
Alustariippumattomat käyttötapaukset								
<input type="checkbox"/> Select	use_case_code	use_case_name	explanation	actors	preconditions	operating_positions	lifecycle_phases	
<input checked="" type="checkbox"/>	UC_59	Työtason nosto tai lasku työtasolta	Operaattori ohjaa työtason työtasolta käsin ylös tai alas, jos nostoa tai laskua ei ole estetty.	Asentaja, Operaattori, Huoltohenkilö, Kouluttaja, Tuotekehitysinööri			Käyttötoiminta; Käyttövaihe; Kokoonpano ja asennus; Kunnossapito; Käyttöönnotto; Tuotekehitysvaihe; Vianetsinta/ongelman selvitys	
<input checked="" type="checkbox"/>	UC_60	Työtason nosto tai lasku etäkäyttöpaneelilla maasta käsin					Kokoonpano ja asennus; Kunnossapito; Käyttötoiminta; Käyttövaihe; Käyttöönnotto; Tuotekehitysvaihe; Vianetsinta/ongelman selvitys	
<p>Add new item</p>								
Jäljitettävyysslinkit vaatimuksiin								
<input type="checkbox"/> Parent: Requirement				Direction	Relation type	Child: UseCase▼	Suspect incoming	Suspect outgoing
		Venttiilit suunniteltava pysäyttämään liike energiansyötön katketessa		<-	Perustuu	Työtason nosto tai lasku työtasolta	No	No
<p>Add new item</p>								
Jäljitettävyysslinkit elinkaaren vaiheisiin								
<input type="checkbox"/> Parent: LifeCycle				Direction	Relation type	Child: UseCase▼	Suspect incoming	Suspect outgoing
		Käyttötoiminta		<-	Suoritetaan aikana	Työtason nosto tai lasku työtasolta	No	No
		Käyttövaihe		<-	Suoritetaan aikana	Työtason nosto tai lasku työtasolta	No	No
<p>Add new item</p>								

Kuva 20. Alustariippumattomien käyttötapauksien tietosivu.

Kuvasta nähdään, että alustariippumattomista käyttötapauksista on jäljitettävyysslinkit vaatimuksiin ja elinkaaren vaiheisiin kuvan 15 konseptimallin mukaisesti (kuvassa 20 ei ole esitetty kaikkia kuvan 15 relaatioita eikä kaikkia käyttötapauksia). Jäljitettävyysslinkit näytetään valitulle käyttötapaukselle (tummempi vinonnuoli). Jäljitettävyysslinkeistä nähdään relaatiotyypin lisäksi myös epäilyksenalaisuuslippujen tilat; liput ovat EI-tilassa, joka tarkoittaa sitä, että linkkien päissä olevat artefaktit eivät ole muuttuneet sen jälkeen, kun jäljitettävyyssliput on viimeksi nollattu.

Jäljitettävyysslinkit on tehty omiin SharePoint-listoihin, eli jokaista relaatiotyyppiä varten luodaan oma relaatiolista.

Vastaavasti painettaessa Fyysisen arkkitehtuurin -painiketta saadaan seuraava tietosivu (kuva 21):

Järjestelmän rakennenäkömät

Järjestelmälle voidaan määritellä useita rakennenäkömiä eli fyysisen arkkitehtuurin näkökulmia. Pääasialliset näkömät ovat yleensä tuotekehityksen aikainen näkömä sekä tuotantovaiheen aikainen näkömä. Edellinen on useimmiten tieteenalaan perustuva (mekaaninen -, hydraulinen -, sähköinen järjestelmä,...), ja jälkimmäinen perustuu mekatronisiin moduleihin, jotka ovat tuotannon kannalta selkeitä asennettavia kokonaisuuksia.

Seuraavassa luetellaan tämän järjestelmän rakennenäkömät ja niihin liittyvät rakenneosat. (Valitse Select-sarakkeesta nuoli-ikonista rakennenäkömä, jonka rakenneosat haluat nähdä alla).

Rakennenäkömät

Select	Title	Description	Image
<input checked="" type="checkbox"/>	Tuotekehitysnäkömä	Näkömä, jota käytetään tuotekehityksen ajan eli konseptivaiheen ja tuotekehitysvaiheen ajan. Näkömä perustuu tieteenaloihin perustuvaan rakenteeseen, eli osajärjestelmiin kuten mekaniikka, sähköinen ohjausjärjestelmä, hydraulikka, jne.	
<input type="checkbox"/>	Tuotantonäkömä	Tämä näkömä on käytössä tuotantovaiheessa. Näkömä perustuu mekatroniseen rakenteeseen, joka vastaa kokoonpanon kannalta optimaalista rakennetta.	Mekatroninen rakenne

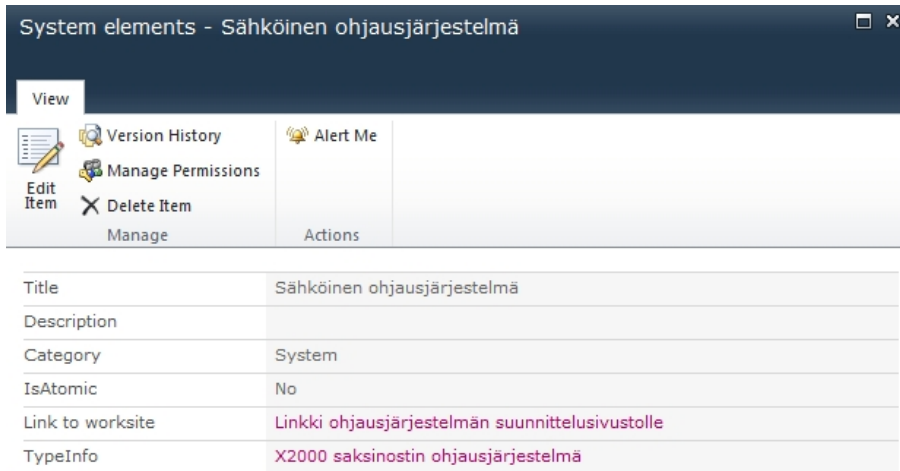
[Add new item](#)

Rakennenäkömään kuuluvat rakenneosat

Parent: Structure view	Suspect incoming	Direction	Suspect outgoing	Relation type	Child: System element
Tuotekehitysnäkömä	No	<-	No	Kuuluu	Sähköinen ohjausjärjestelmä
Tuotekehitysnäkömä	No	<-	No	Kuuluu	Sähköinen järjestelmä
Tuotekehitysnäkömä	No	<-	No	Kuuluu	Hydraulinen järjestelmä
Tuotekehitysnäkömä	No	<-	No	Kuuluu	Toimilaitteet
Tuotekehitysnäkömä	No	<-	No	Kuuluu	Teholähteet
Tuotekehitysnäkömä	No	<-	No	Kuuluu	Mekaaninen rakenne

Kuva 21. Järjestelmän rakennenäkömien tietosivu.

Järjestelmän fyysinen arkkitehtuuri ilmaistaan kuvan 16 konseptimallin mukaan erilaisina näkyminä. Kuvassa 21 on valittu tuotekehitysnäkömä, joka näkyy myös SE-työpöydän pääsivun keskellä. Osajärjestelmiin päästään valitsemalla haluttu rakenneosa (tässä: osajärjestelmä) ja valitsemalla halutaanko päästä osajärjestelmän omalle SE-sivulle (jos se suunnitellaan saksinostimen rakentajan toimesta) tai osajärjestelmän tyyppitietoihin siinä vaiheessa kun kyseinen osajärjestelmä on valmis ja dokumentoitu tyyppitietoihin. Dialogi valintaan on esitetty kuvassa 22.



Kuva 22. Dialogi osajärjestelmätietoihin pääsemiseksi.

Suunnitteluartefakteja voidaan selailia myös tietomallien välityksellä. Tietomallit voidaan avata napsauttamalla Suunnitteluartefaktit-tekstiä. Tällöin avautuu seuraava valikko (kuva 23).

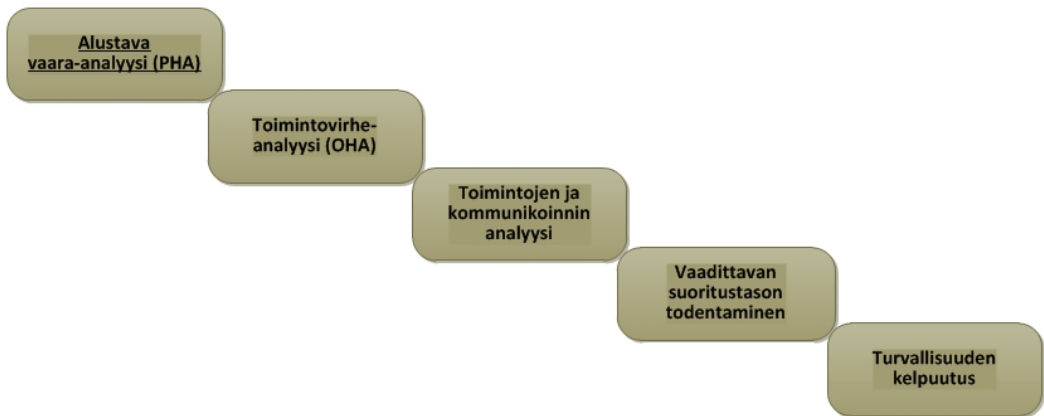


Kuva 23. Tietomallien valikko.

Kun napsautetaan *Vaatimukset sekä todennus ja kelpuus* -painiketta, avautuu kuvan 14 mukainen tietomalli. Tietomallin kuvakkeista voi sitten avata yksittäisiä artefaktilistoja, kuten vaatimuslistan.

Erikoisalut

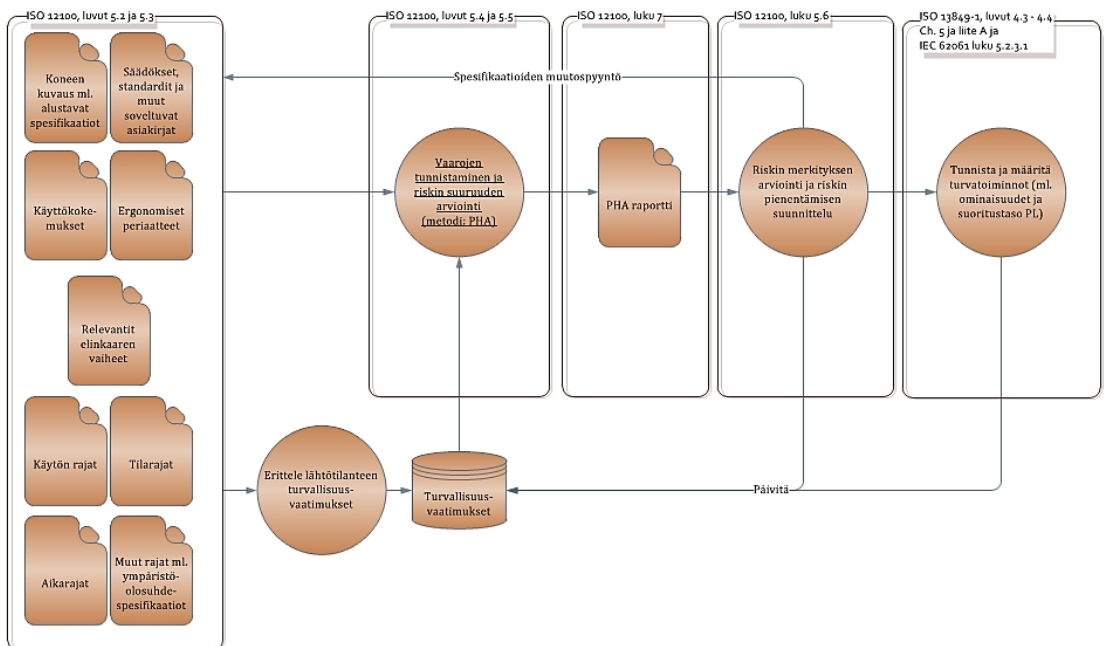
Erikoisaloja voi olla useita. Tässä annetaan esimerkki turvallisuusprosessista. Kun pääsivulta painetaan Turvallisuus-painiketta, saadaan näkyviin valikko, josta päästään turvallisuusprosessin eri vaiheisiin (kuva 24).



Kuva 24. Turvallisuusprosessin vaiheet.

Kun valitaan alustava vaara-analyysi, saadaan seuraava prosessikuva (kuva 24).

Alustava vaara-analyysi (PHA)



Kuva 25. Alustavan vaara-analyysin prosessikuva.

Prosessikuvasta pääsee taas porautumaan analyysin lähtötietoihin, raporttiin sekä analyysityökaluihin sekä turvallisuusvaatimuksiin. Kuvasta pääsee myös avaamaan analyysiin liittyvät standardit ja ohjeet (standardit avautuvat standardiviitteistä ja ohjeet avautuvat yhdessä työkalun kanssa alleviivatusta kuvaobjektista).

Parametrit

Parametrivalikosta voidaan avata tarkasteltavaksi järjestelmään, sen rakenneosiin ja sen yksilöihin liittyvät parametrit. Yksilöparametreilla ei tarkoiteta tässä loppuasiakkaan tai operaattorin tekemiä parametreita, vaan konfiguraatioon liittyviä parametreja, jotka kertovat miten kyseinen koneyksilö on varustettu ja konfiguroitu.

Parametrit esitetään normaaleina SharePoint-listoina.

Työkalut

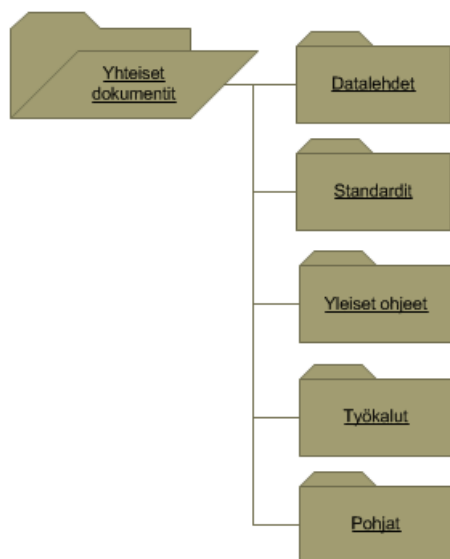
Työkalukategorista esimerkkinä annetaan riskien arvioinnin työkalujen valikko (ks. kuva 26).



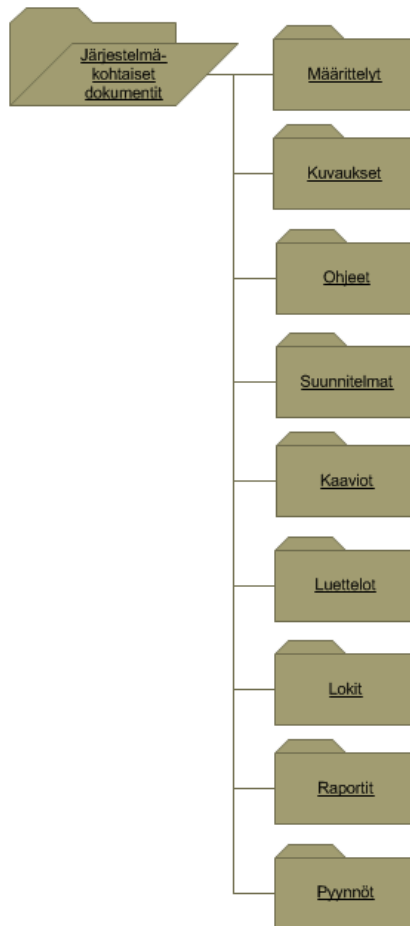
Kuva 26. Riskien arvioinnin työkaluvalikko.

Dokumentit

Dokumenttivalikosta voidaan avata joko kaikille järjestelmille yhteiset dokumentit (kuva 27) tai kohteena olevan järjestelmän (kuva 28) dokumentit.



Kuva 27. Järjestelmien yhteisten dokumenttien valikko.



Kuva 28. Kohteena olevan järjestelmän dokumenttien valikko.

Vastaavat näkymät löytyvät myös suoraan SharePoint-dokumenttikirjastoina (ks. kuva 29).



SE_PLM > Yhteiset dokumentit > All Documents >
Kehitettävänä olevien järjestelmien yhteiset dokumentit

me Saksinostin SE-työt

Type	Name	Modified	Modified By
Folder	Datalehdet	10.10.2012 9:31	Alanen Jarmo Albert
Folder	Pohjat	10.10.2012 9:32	Alanen Jarmo Albert
Folder	Standardit	10.10.2012 9:31	Alanen Jarmo Albert
Folder	Työkalut	10.10.2012 9:32	Alanen Jarmo Albert
Folder	Yleiset ohjeet	10.10.2012 9:32	Alanen Jarmo Albert

+ Add document

Kuva 29. Järjestelmien yhteisten dokumenttien SharePoint -dokumenttikirjasto.

Päätason keskiosa: tekniset prosessit

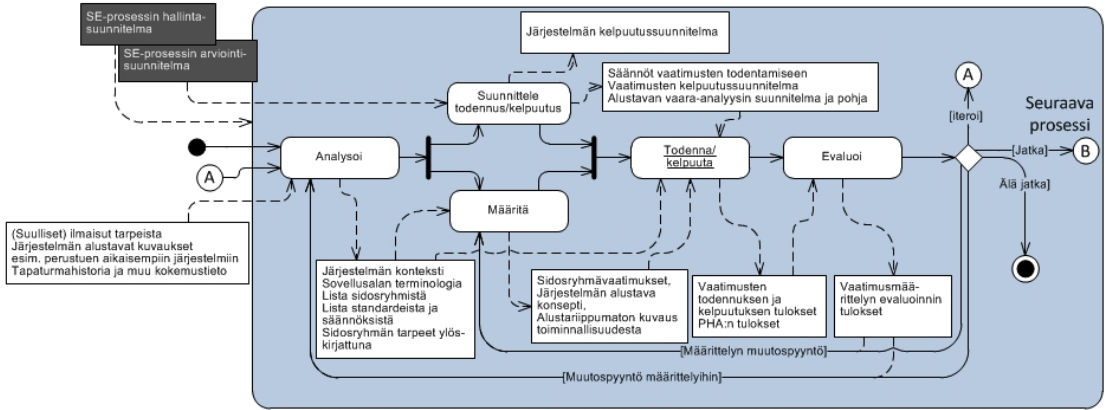
Teknisiin prosesseihin pääsee pääsivulta. Vastaavasta objektista pääsee porautumalla seuraavaan valikkoon, josta valitaan haluttu elinkaaren vaihe (kuva 30).

Valitse elinkaarivaihe, jonka prosessiin haluat mennä



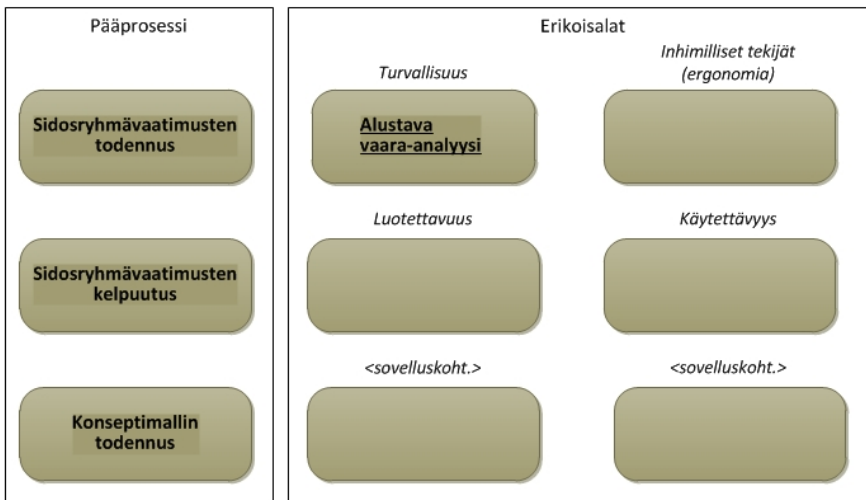
Kuva 30. Elinkaaren vaiheen valinta.

Elinkaaren vaiheen valinnasta päästään varsinaiseen teknisen prosessin toimintokaavioon. Esimerkiksi tuotekehitysvaiheen valinta avaa kuvan 7 toimintokaavion, ja toimintokaavion eri vaiheista avautuu kuvien 9–12 mukaiset tarkemmat toimintokaaviot. Esimerkiksi kun poraudutaan konseptivaiheen prosessiin ja sieltä sidosryhmävaatimuksiin saadaan kuvan 31 mukainen seuraava tarkempi toimintokaavio. Tarkempien toimintokaavioiden objekteista voidaan porautua työkaluihin, artefakteihin ja dokumentteihin.



Kuva 31. Sidosryhmävaatimusten määrittysten prosessikaavio.

Esimerkiksi valittaessa Todenna/kelpuuta-aktiviteetti, saadaan seuraava valikko (kuva 32).



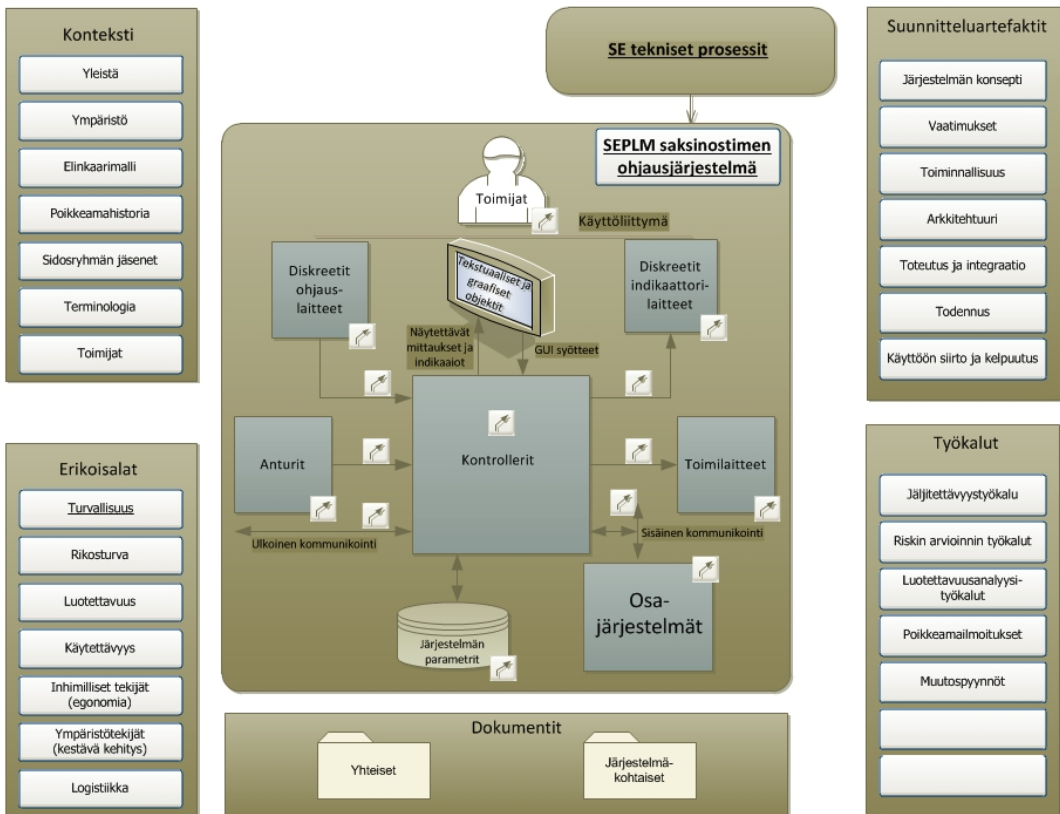
Kuva 32. Todennus- ja kelpuutusaktiviteettien valinta.

Kuvasta nähdään, että turvallisuusanalyysit luetaan todennusaktiviteetteihin (järjestelmä todennetaan turvallisuusnäkökulmasta). Turvallisuusprosessin valikko kuvassa 24 ei ota kantaa, missä vaiheessa eri turvallisuusanalyysit tehdään. Sen sijaan teknisten prosessien kaavioista, erityisesti kuvan 31 mukaisten toimintokaavioiden todennus- ja kelpuutusaktiviteettien kautta turvallisuusanalyysien oikeat suorituskohdat tulevat määritellyiksi.

Päätason keskiosa: muu osuus

Päätason keskiosan SEPLM-saksinostin-objektista voi avata tietosivun, jossa kohteena oleva järjestelmä tunnistetaan ja sille annetaan lyhyt yleiskuva ja alustava konsepti.

Osajärjestelmistä voi avata kohteena olevan järjestelmän kehitteillä olevat osajärjestelmät. Esimerkiksi ohjausjärjestelmä-objektista avautuu seuraava näkymä (kuva 33).

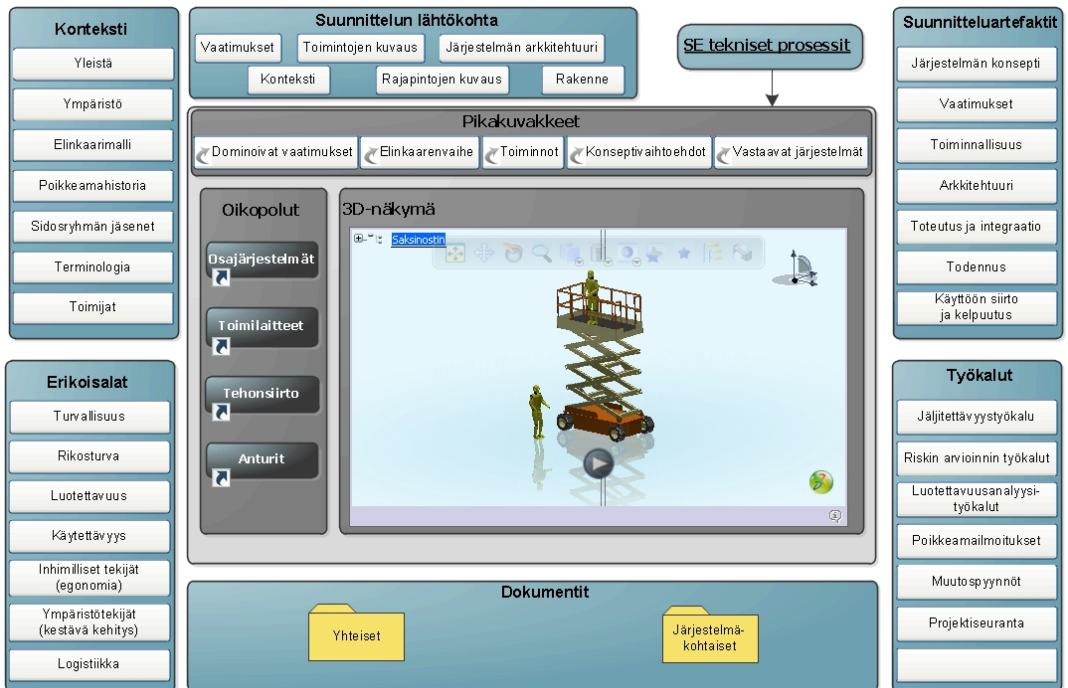


Kuva 33. Ohjausjärjestelmän SE-työpöytä.

Kuten kuvasta 33 havaitaan, työpöytä on periaatteessa samankaltainen kuin saksinostimellakin. Tässä siis noudatetaan kuvan 4 struktuurimallia, eli osajärjestelmät ovat itsessään järjestelmiä sille organisaatiolle, jonka vastuulla sen suunnittelu ja muu elinkaari on.

Osajärjestelmille luodaan oma SharePoint-sivusto työpöytineen ja kuvan 4 artefaktimalleineen. Samoin osajärjestelmille on omat sisäiset tekniset prosessit. Prosessimalli voi olla hyvinkin erilainen verrattuna päätason prosessimalliin; esimer-

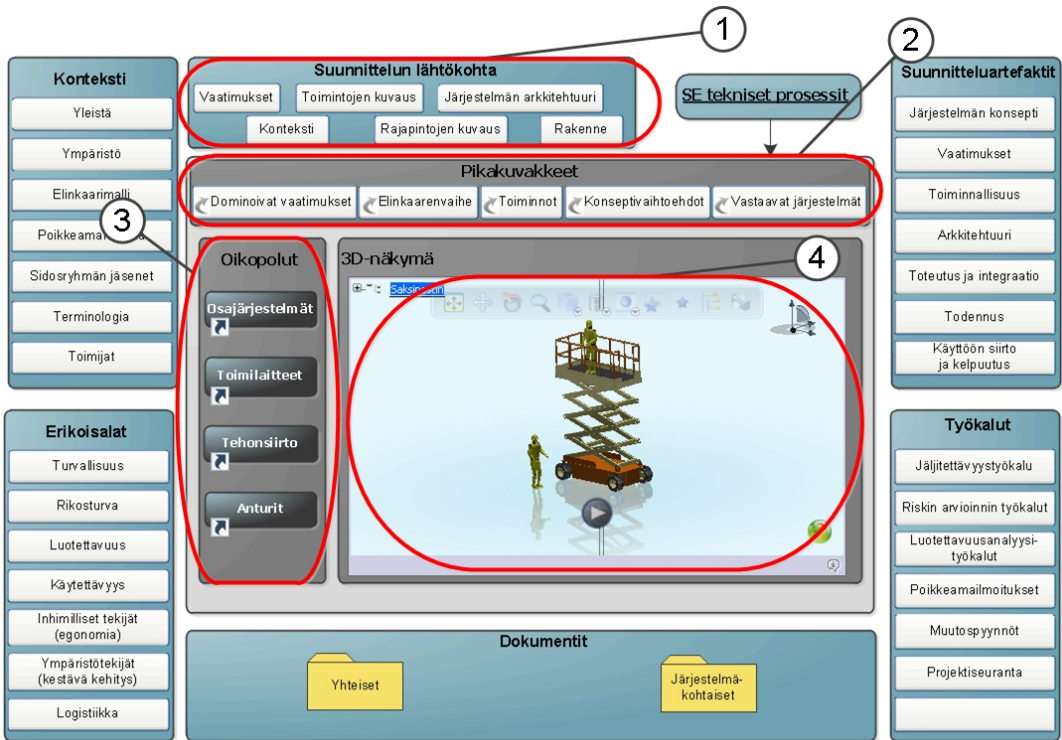
kiksi ohjelmiston suunnittelussa noudatetaan omaa prosessimallia, usein ns. V-mallia. Myöskään työpöydät eivät ole identtisiä päätason työpöydän kanssa, vaan jokaisen osajärjestelmän työpöydällä on oma tarkoituksenmukainen sisältönsä ja ilmeensä. Päätason järjestelmästä, eli saksinostin-tason artefakteista, voidaan tehdä jäljitettävyyssyöntejä osajärjestelmien artefakteihin. Päätason mekaaninen rakenne-objektista avautuu mekaanisen rakenteen SE-työpöytä (kuva 34).



Kuva 34. Mekaanisen rakenteen SE-työpöytä.

Kuvassa 34 esitetty työpöytä pohjautuu jo esiteltyyn saksinostimen päätason näkymään, samankaltaisesti kuin myös ohjausjärjestelmän näkymä. Mekaanisen rakenteen käyttöliittymän näkymässä on painotettu mekaniikkasuunnittelijaa hyödyttäviä osa-alueita.

Mekaanisen rakenteen SE-työpöydän keskiosan valikot on korostettu kuvassa 35.



Kuva 35. Mekaanisen rakenteen valikot.

Mekaanisen rakenteen käyttöliittymän tarjoaa pääsyn mm. seuraaviin objekteihin:

- Suunnittelun lähtökohta (1): Järjestelmän suunnittelussa tarvittavat lähtötiedot. Jäädytetyt versiot kyseisen järjestelmän suunnittelua varten toimitetuista aineistoista. Aineisto toimii myös pohjana järjestelmän suunnittelussa täydentyville osa-alueille.
- Pikakuvakkeet (2): Suunnittelijan valitsema pikalinkkejä järjestelmän tietoihin. Esimerkiksi pikakuvakkeesta "Dominoivat vaatimukset" saadaan näkymä suodatettuun vaatimuslistaan. Dominoivilla vaatimuksilla (Design Drivers) tarkoitetaan tässä yhteydessä vaatimuksia, jotka toteutuessaan varmistavat että järjestelmän rakenne voidaan toteuttaa. Toteuttamalla dominoivat vaatimukset saadaan toteutettua yleensä myös suuri joukko muita vaatimuksia. Muut pikakuvakkeet tarjoavat suoran näkymän järjestelmän tietoihin.
- Oikopolut (3): Suunnittelija voi halutessaan tarkastaa mekaaniseen rakenteeseen liittyvien muiden järjestelmien toteutuksia.
- 3D-näkymä (4): Näkymässä on suunniteltavan järjestelmän 3D-malli. Järjestelmän 3D-malli on tallennettu 3DXML-muodossa SharePointiin (Järjes-

telmäkohtaiset dokumentit/kuvaukset-kansio), tiedosto esitetään selaimessa 3DXML-viewer¹⁰ avulla. Malli ei päivity CAD/PDM-järjestelmästä automaattisesti, 3D-mallin päivityksen jälkeen 3DXML-tiedosto pitää generoida uudestaan. Toimenpiteen voi automatisoida esimerkiksi hyödyntämällä CAD-ohjelman komentosarjoja. Vaihtoehtoinen esitystapa on upottaa 3D-malli U3D-tiedostomuodossa PDF-tiedostoon ja esittää tämä tiedosto mekaniikkasuunnittelijan työpöydällä. PDF-esitystavan etuina on yleinen tiedostomuoto ja mahdollisuus lisätä kommentteja suoraan 3D-näkymään. Tiedoston päivittämisen automatisointi on kuitenkin hankalaa, joten tässä toteutuksessa on pitäydytty 3DXML-tiedostomuodossa. 3D PDF-tiedostomuotoa voisi hyödyntää tehokkaasti esimerkiksi suunnittelun katselmoineissa.

Osajärjestelmien Share Point -sivustot voivat olla periaatteessa samalla hierarkiatasolla kuin pääjärjestelmäkin¹¹. Ei-hierarkkinen sivustorakenne on järkevä siksi, että jotakin saksinostimen osajärjestelmää, esimerkiksi tehölähteet, voidaan käyttää jossakin toisessa saman valmistajan koneessa. Tällöin järjestelmien ja sen rakenneosien hierarkiaa ei rakenneta näkyviin Share Point -sivustojen hierarkiana vaan fyysisen arkkitehtuurin tietomallin kautta, koska kyseessä on monestamoneen relaatio järjestelmien ja osajärjestelmien välillä. Toisaalta hierarkkinen rakenne ei estä käyttämästä samaa osajärjestelmää muuallakin, koska – kuten kuvan 4 struktuurimalli esittää – osajärjestelmät julkaistaan rakenneosatyyppien kirjastossa, josta kukin pääjärjestelmä ottaa ne käyttöön. Toisin sanoen rakenneosatyyppien kirjasto pitää olla sellaisella tasolla, että kaikilla sitä käyttävillä järjestelmillä on siihen pääsy (eli se ei saa olla jonkin tietyn järjestelmän omistuksessa).

¹⁰ Selaimen tulee asentaa 3DXML-viewer, jotta 3D-mallin saa näkymään. Mekaanisen rakenteen työpöytä ehdottaa automaattisesti kyseisen tiedoston asennusta.

¹¹ Jäljitettävyysslinkkien tekoa järjestelmästä osajärjestelmään kokeiltiin ainoastaan hierarkkisessa sivustorakenteessa.

Tuotekehitys ja virtuaalikatselmointi

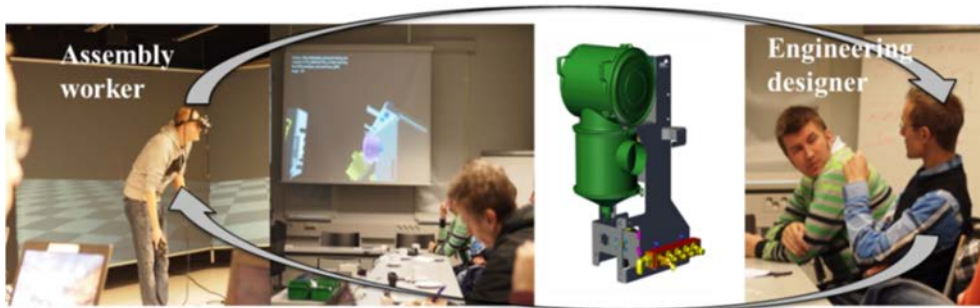
Simo-Pekka Leino, VTT

Katselmukset ovat tärkeitä tuotekehitysprojektin tarkastuspisteitä, joissa varmistetaan, että tuotteen vaatimukset ja evaluoinnin kriteerit ovat kunnossa ja että suunnitteluratkaisut ovat niiden mukaisia. Katselmuksia toteutetaan tuoteprosessin eri vaiheissa ja niillä on vaiheeseen liittyviä erilaisia tavoitteita. Katselmus voi liittyä tuotteen elinkaaren vaiheiden aspekteihin, esimerkiksi käyttäjätarpeiden kartoitukseen, vaatimusmäärittelyjen validointiin, konseptin ja arkkitehtuurin suunnitteluun ja todentamiseen, uuden tuotteen esittelyyn omalle organisaatiolle, tuotteistukseen eli tuotantoon valmistautumiseen, tuotteen kokoonpantavuuteen ja huollettavuuteen tai markkinointiin. Katselmointien tarkoituksena on sekä tunnistaa virheitä ja poikkeamia että kerätä uusia ideoita ja palautetta eri tuotteen elinkaaren sidosryhmiltä. Katselmointeja voidaan hyödyntää sekä yrityksen sisäisesti eri toimintojen välillä, että yritysverkostossa tai asiakkaan kanssa.

EU-ManuVAR-tutkimushankkeessa (www.manuvar.eu) kehitettiin virtuaalikatselmoinnin teknologiaa sekä toimintatapoja ja prosesseja kaivos- ja rakennusteollisuuden koneita valmistavan yrityksen kanssa. Katselmoinnin kohteena oli erityisesti uuden tuotteen ja sen moduulien tuotteistaminen ja manuaalinen kokoonpano. Virtuaalikatselmointeihin osallistui sidosryhmiä laajasti suunnittelusta, projektinhallinnasta ja tuotannosta. Keskeisenä tavoitteena oli parantaa näiden toimijoiden välistä yhteistyötä ja kommunikointia. Erityisesti esimerkiksi suunnittelijoiden ja kokoonpanotyöntekijöiden kommunikointi oli mielenkiinnon kohteena. Perinteisesti kommunikoinnin haaste liittyy sekä prosesseihin ja toimintatapoihin että viestintävälineisiin. Kaikki toimijat eivät esimerkiksi välttämättä ymmärrä CAD-malleja tai 2D-piirustuksia samalla tavalla. Malleja ei välttämättä edes haluta näyttää ennen kuin ne ovat täysin valmiita, mikä estää niiden kommentoimista ajoissa. Tuotekehityksessä ei toisaalta välttämättä tiedetä miten tuote valmistetaan ja kokoonpannaan todellisuudessa. Virtuaaliympäristöt voivat parantaa tilannetta, jos toimintatavat ja prosessit päivitetään samalla.

Esimerkin virtuaalikatselmoinneissa käytettiin tyypillisesti virtuaaliympäristöjen konfiguraatiota, jossa kokoonpanotyöntekijöillä oli ns. HMD-silmikko (Head Mounted Display) ja katselmoinnin muut osallistajat tarkkailivat tapahtumia suurilta valko-

kankailta. Katselmoinnissa keskusteltiin tietystä aiheesta, kun työntekijä teki kokoonpanotyötä uppoutuneena virtuaali-ympäristöön (kuva 1).



Kuva 1. Virtual Environments aided design review meeting.

Hankkeessa kehitettiin myös mallinnustekniikkaa ja metodiikkaa sosioteknisestä näkökulmasta. Teknisen järjestelmän rakenteen lisäksi mallinnettiin tehtävärakennetta ja prosesseja, jolla tuotannon työntekijät toimivat. Tuotetiedon hallintaan liittyen kehitettiin rajapintaa PDM-järjestelmään ja katselmointiin liittyviä prosesseja PLM:ssä.

Vaatimustenhallinta järjestelmähankinnassa

Kari Karstila, Eurostep Oy

Tausta	173
Tavoite	173
Perusteet.....	174
Ratkaisu ja toteutus.....	174
Järjestelmänäkymät ja rakenteet.....	174
Vaatimukset ja niiden rakenteet	178
Vaatimusten verifiointi ja validointi.....	181
Järjestelmän yksilöhallinta.....	183
Tulokset	185
Keskustelua	186

Tausta

Kohteena on pientalon lämmitysjärjestelmän uusimishanke. Hanketta haluttiin käyttää testi- ja malliesimerkinä siitä, miten ISO 10303:2012 Product Lifecycle Support (PLCS) -tuotetietostandardin määrittelemiä elinkaaritietoja voidaan soveltaa järjestelmän elinkaarenhallintaan, mukaan lukien vaatimustenhallinta. Aikaisemman, ajoneuvon tuoteyksilön hallintaan liittyneen toteutuksen kautta, oli jo tiedossa, että ISO PLCS -standardia ja sen toteuttavaa PLM-järjestelmää (PLM – Product Lifecycle Management, tuotteen elinkaaren hallinta) voidaan hyvin soveltaa tuoteyksilön elinkaarenhallintaan. Myös tässä lämmitysjärjestelmän tapauksessa yksilöhallinta on kiinnostuksen kohteena, sisältäen konfiguraation muutokset, vikatilanteet sekä huolto- ja korjaustoimenpiteet.

Vaatimustenhallinnan osalta päämotiivi oli dokumentoida järjestelmähankinnan perusteena olevat vaatimukset ja tehdä esimerkki siitä, miten eritasoiset vaatimukset, niiden jäljitys sekä verifiointi ja validointi voidaan toteuttaa, joskin tietysti tapaus huomioon ottaen datan määrän osalta pienimuotoisesti.

Tavoite

Tavoitteena oli saada PLM-järjestelmällä hallittavaksi seuraavat toisaalta järjestelmän hankintaan liittyvät tiedot, ja toisaalta järjestelmän käyttöön ja ylläpitoon liittyvät tiedot:

- Hankintaan liittyvät sidosryhmävaatimukset sekä niistä johdetut järjestelmä- ja tuotevaatimukset, sekä vaatimukset järjestelmän käyttöön ja ylläpitoon. Tarkoituksena oli dokumentoida järjestelmävalinnan ja hankintapäätöksen perustana olleet vaatimukset, sekä toisaalta vaatimukset järjestelmän elinkaaren hallintaan käyttöä ja ylläpitoa varten.
- Järjestelmärankenteen kuvaus kattaen eri näkökulmat (toiminnallinen, järjestelmä, tuote, tuoteyksilö)
- Järjestelmärankenteita käytettäisiin sekä vaatimusten että käyttöön ja ylläpitoon liittyvien tietojen ja dokumentaation liittämiseen kontekstiinsa
- Järjestelmän konfiguraatiomuutokset, tapahtumatiedot (vikatilanteet), sekä tehdyt huolto- ja korjaustoimenpiteet.

Esimerkki on laadittu järjestelmän hankkijan/omistajan/käyttäjän näkökulmasta ja häntä varten, eikä tietojen hyödyntämiseen ole toistaiseksi osallistunut muita osapuolina. Järjestelmän elinkaaren osalta PLM-järjestelmän on tarkoitus toimia käytännönläheisenä sovelluksena järjestelmän käytön ja ylläpidon kaikkien tapahtumien ja toimenpiteiden dokumentoinnin ja seurannan välineenä. Kohdassa *Keskustelua* on kuitenkin kuvattu esimerkkitoteutuksen laajentamismahdollisuuksia useamman toimijan kollaboraatioon.

Perusteet

Sovellusesimerkissä haluttiin testata ja samalla luoda malliesimerkki ISO 10303 Product Lifecycle Support (PLCS) -tuotetietostandardin soveltamisesta vaatimusten ja elinkaarihallintaan käyttäen standardin määrittelemiä tuote- ja ylläpitotietoja. ISO PLCS -standardi määrittelee kattavasti tuotetietoja tuotteen elinkaaridimensiossa lähtien tuotevaatimuksista, tuotteen määrittelyyn, valmistukseen ja tuoteyksilönhallintaan, mukaan lukien tuotteen ylläpidon ja huollon suunnitelma- ja toteumatiedot.

Vaatimus- ja tuotetietojen hallinnan sovelluksena käytettiin Share-A-space™ PLM-kollaboraatiosovellusta, joka toteuttaa ISO PLCS -standardin mukaisten tuotetietojen hallinnan ja tarvittaessa myös usean toimijan kollaboraatiotoiminnallisuudet.

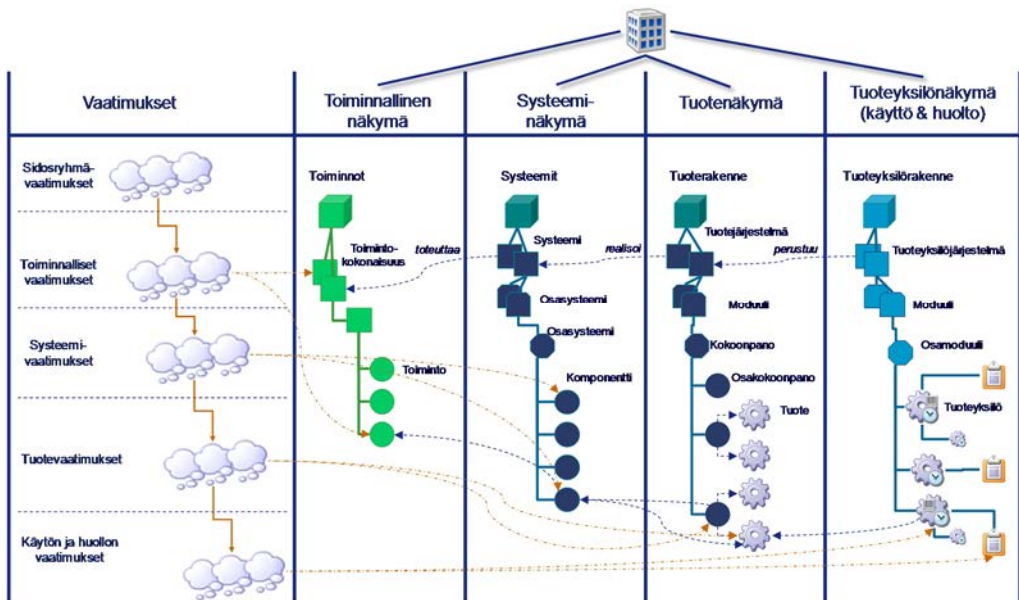
Ratkaisu ja toteutus

Sovelletussa vaatimus- ja tuotetietojen hallinnan ratkaisussa voidaan toteutetut tiedot jakaa neljään eri osa-alueeseen:

- Järjestelmäkuvaus, joka pitää sisällään useamman näkymän järjestelmään ja näkymien osittelun rakenneosiin. Näitä näkymiä ovat toiminnallinen, systeemi-, tuote- ja tuoteyksilönäkymät.
- Vaatimukset eri tasoilla ja eri näkymiin kohdistuen: toiminnalliset, systeemi-, tuote- sekä käytön ja huollon vaatimukset.
- Vaatimusten verifiointin ja järjestelmän validoinnin tiedot
- Tuoteyksilön hallintaan liittyen järjestelmän käytön, huollon ja korjauksen suunnitelma- ja toteumatiedot. Tähän sisältyy myös muutos- ja korjaustoimenpiteiden aiheuttamat järjestelmän konfiguraatiomuutokset.

Järjestelmänäkymät ja rakenteet

Järjestelmäkokonaisuuden eri elinkaarinäkymät ja niiden osittelurakenteet ovat järjestelmäkuvauksen perusta. Näihin näkymiin kohdistetaan erityyppiset vaatimukset, sekä tuoteyksilönäkymään järjestelmän elinkaaritiedot, esimerkiksi huolto- toimenpiteet ja konfiguraatiomuutokset.



Kuva 1. Järjestelmänäkymät ja vaatimukset eri tasoilla.

Siten järjestelmäkuvaus koostui useammasta, järjestelmän määrittelyn etenemiseen ja järjestelmän elinkaaren vaiheisiin liittyvästä näkymästä (kuva 1):

- Toiminnallinen näkymä: Kuvaa järjestelmän yleisen toiminnallisen osittelun toimintoihin. Esimerkiksi lämmön keruu, lämmön siirto, lämmön jakelu, jne. Sidosryhmävaatimuksista johdetut toiminnalliset vaatimukset liitettiin toimintoihin. Toiminnallinen näkymä ei ota kantaa millä teknisellä ratkaisulla järjestelmä toteutetaan. Toiminnallisen näkymän tarkoituksena on tarjota rakenne, johon voidaan kytkeä teknisestä ratkaisusta riippumattomat toiminnalliset vaatimukset. Saman toiminnallisuuden toteuttamiseksi voi sitten olla olemassa useita erilaisia teknisiä ratkaisuja.
- Systeminäkö: Järjestelmän valitun toteutusratkaisun tyyppin perusteella tarkennettu järjestelmän osittelu toiminnot toteuttaviin osiin (osajärjestelmiin). Siten esimerkiksi lämmön lähde tarkentui tässä tapauksessa maalämmönlähteeksi, lämmön tuotto/vaihtotoiminto tarkentui lämpöpumppusysteemiksi. Toiminnalliset vaatimukset on tarkennettu systeemivaatimuksiksi, jotka liitettiin systeminäkömön osajärjestelmiin. Kaikki systeemivaatimukset eivät kuitenkaan ole johdettu toiminnallisista vaatimuksista, vaan systeemivaatimuksia voitiin ja oli tarpeen joiltakin osin tarkentaa, kun toteutuksen periaateratkaisu oli päätetty.

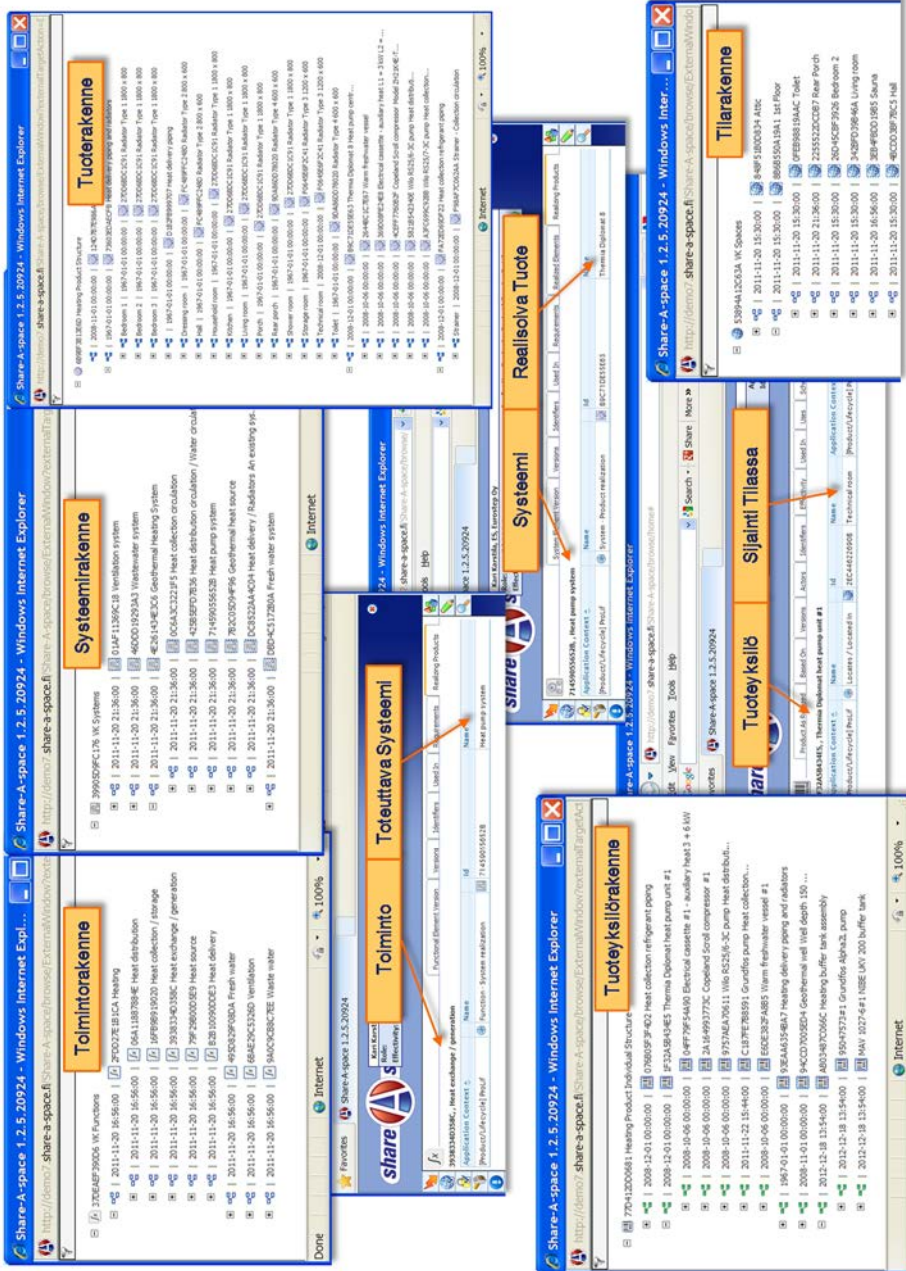
Huomautus: Järjestelmä ja systeemi ovat sinänsä synonyymejä. Tässä yhteydessä järjestelmä termillä viitataan konkreettiseen lämpöä tuottavaan laitekokonaisuuteen.

Systeemillä tarkoitetaan tuotevalinnoista riippumatonta järjestelmän ja sen osien määrittelynäkökulmaa.

- Tuotenäkymä: Järjestelmän tuotevalintojen perusteella syntyvä tuoterakenne, joka koostuu kaupallisista tuotteista, joilla järjestelmä, sen osajärjestelmät ja komponentit toteutetaan. Tässä tapauksena keskeisin tuotevalinta oli lämpöpumppulaitteisto. Muut osat, lämpökaivo, lämmönkeruuputkisto sekä liitännät lämmitys- ja käyttövesiputkistoihin olivat paikalla tehtäviä tuoteosia.
- Tuoteyksilönäkymä: Kuvaa järjestelmän yksilöllistä, mahdollisesti ajassa muuttuvaa konfiguraatiota, jossa keskeisimmät, sarjanumerolla tunnistettavat komponentit voivat olla yksilö seurannan kohteena. Huoltotoimenpiteet, vikatilanteet, muutos- ja korjaustoimenpiteiden aiheuttamat laitteiden vaihdot koskevat tuoteyksilörakennetta ja sen osia. Tuoteyksilörakenteeseen kohdistettiin myös järjestelmän käyttöä ja huoltoa koskevat vaatimukset – esimerkiksi vuosittaiset (joskin vähäiset) huoltotoimenpiteet. Tuoteyksilönäkymään liittyi omana rakenteenaan rakennuksen tilajärjestelmän yksittäiset tilat (huoneet), johon liittyen lämmitysjärjestelmän komponenteilla on relaatio tilaan, jossa ne sijaitsevat.

Edellä mainittujen järjestelmänäkymien osien keskinäisiä suhteita (relaatioita), jotka on myös kuvattu PLM-järjestelmään, on myös esitetty kuvassa 1: Toiminnallisen näkymän toiminnot *toteutuvat* systeeminäkymän osasysteeminä; osasysteemit *realisoituvat* tuotenäkymän tuotteilla. Esimerkiksi: Lämmönlähde-toiminto toteutuu periaateratkaisun systeemiosana maalämpölähde, joka realisoituu tuotenäkymän maalämpökaivona (150 m syvä kallioreikä).

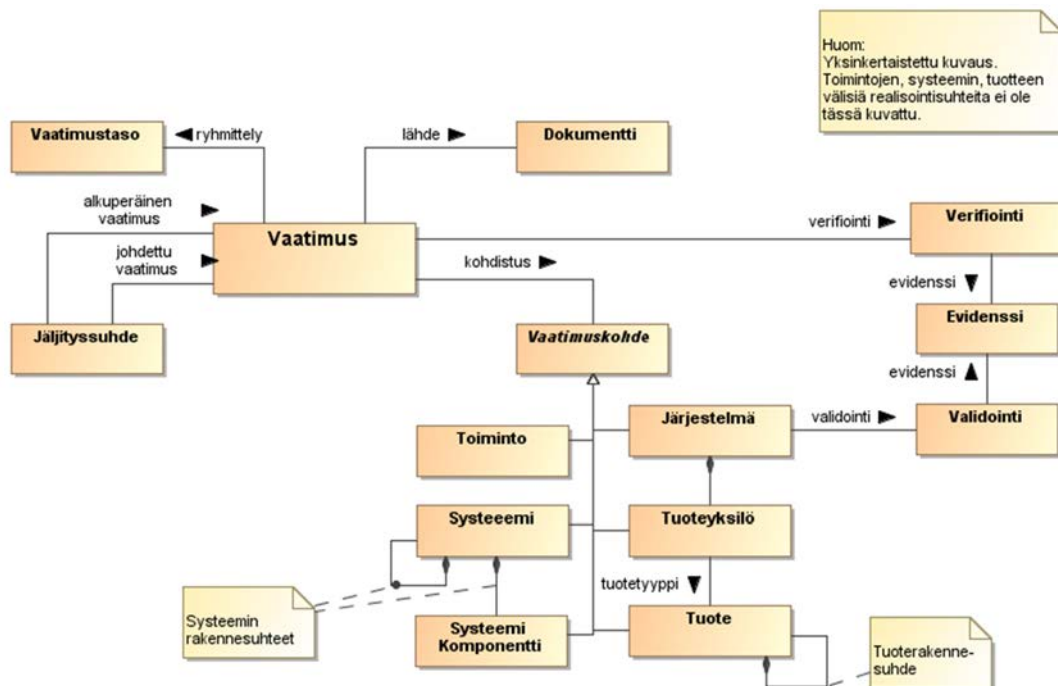
Eri järjestelmän näkymiä PLM-järjestelmässä on esitetty kuvassa 2.



Kuva 2. Eri järjestelmänäkymiä ja niiden välistä relaatioita PLM-järjestelmässä.

Vaatimukset ja niiden rakenteet

Vaatimustietojen hallinnan kannalta keskeisiä asioita ratkaisussa olivat (kuva 3): Vaatimustasot, vaatimusten lähde, vaatimusten kohde ja vaatimusten jäljitys (johtaminen) toisista vaatimuksista.



Kuva 3. Yksinkertaistettu käsitekaavio (UML-luokkakaavio notaatiolla) vaatimuksista ja niihin liittyvistä tiedoista.

Vaatimustasot osittelevat tai ryhmittelevät vaatimukset niiden tyypin ja kohteen mukaan sekä navigoinnin helpottamiseksi. Osittain vaatimustasot heijastavat myös vaatimusten johtamisen vaiheistusta. Vaatimustasoja olivat:

- Sidosryhmävaatimukset (hankkijan, omistajan ja käyttäjän keskeiset vaatimukset)
- Toiminnalliset vaatimukset – kohdistuvat järjestelmätoimintoihin
- Systeemivaatimukset – kohdistuvat järjestelmän periaateratkaisun systeemi-ositteluun
- Tuotevaatimukset – kohdistuvat järjestelmän tuotevalintojen tuotteisiin

- Käytön vaatimukset – kohdistuvat järjestelmän tuoteyksilöihin sekä käytön ja seurannan toimenpiteisiin
- Ylläpidon vaatimukset – kohdistuvat järjestelmän tuoteyksilöihin ja ylläpidon toimenpiteisiin.

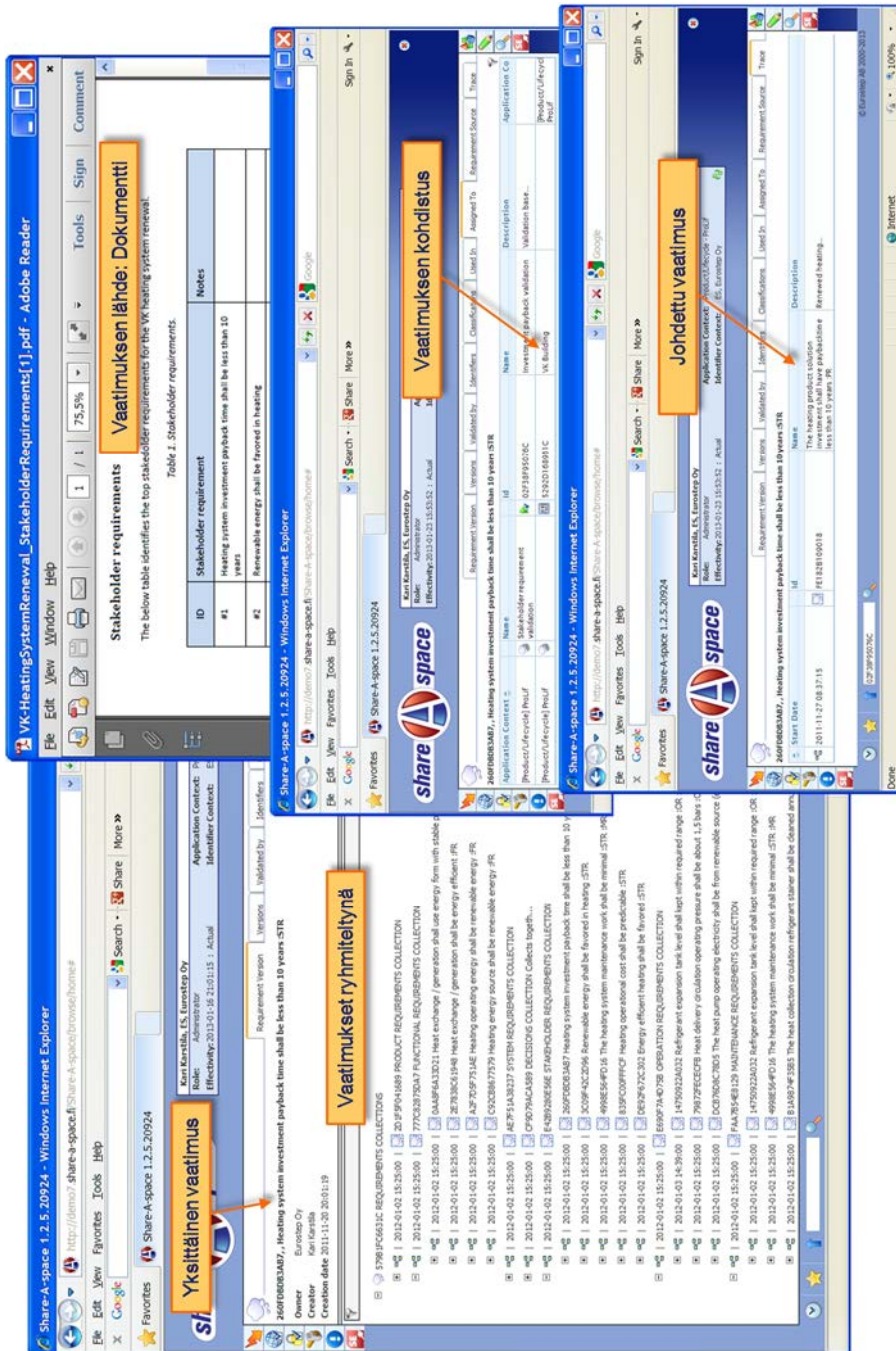
Vaatimusten lähteenä haluttiin tunnistaa kuka vaatimuksen on määritellyt tai mistä (dokumentista) se on tullut. Lähtökohtana olevia sidosryhmävaatimuksia sitten tarkennettiin johtamalla niistä uusia, toiminnallisia yms. vaatimuksista. Esimerkiksi toiminnallisista vaatimuksista johdetaan tarkennettuja systeemivaatimuksia, kun systeemin periaateratkaisu on päätetty. Systeemivaatimuksista johdettiin edelleen tuotevaatimukset.

Vaatimukset myös haluttiin kohdentaa asiaan, jota ne koskevat: Toiminnalliset vaatimukset kohdistuvat tyypillisesti koko järjestelmään tai yksittäiseen toimintoon, systeemivaatimukset periaateratkaisun kuvaavaan systeemiosittelun osasysteemeihin ja niiden komponentteihin, ja tuotevaatimukset tuoterakenteen tuoteosiin.

Vaatimusten keskinäisiä suhteita kuvaa jäljittämisrelaatio (engl. tracing), joka linkittää toisiinsa ”ylemmän tason” vaatimuksen ja siitä johdetun tarkennetun vaatimuksen; esimerkiksi sidosryhmävaatimuksesta johdetun toiminnallisen vaatimuksen tai toiminnallisesta vaatimuksesta johdetun systeemivaatimuksen. Vaatimusten johtaminen voi myös tapahtua järjestelmänäkymän sisällä: Tuotevalinnasta voi seurata joukko tarkentavia tuotevaatimuksia. Esimerkiksi lämmitysjärjestelmän keskuskojeen valinnasta seurasi mitoitusvaatimuksia lämpökaivon syvyydelle.

Määritellyt sidosryhmävaatimukset koskivat järjestelmän yleistä energiatehokkuutta ja mahdollisuutta käyttää uusiutuvaa energiaa, järjestelmän takaisinmaksuaikaa, muutostoimenpiteiden minimointia lämmitysjärjestelmän uusimisessa sekä vähäistä huoltotarvetta. Näistä on edelleen johdettu systeemi- ja tuotenäkymiä koskevia, osittain tarkennettuja vaatimuksia. Näissä näkymissä on myös tullut periaateratkaisun ja tuotevalintojen myötä kokonaan uusia vaatimuksia. Yksityiskohtaisimmillaan vaatimukset tuotenäkymässä koskivat esimerkiksi järjestelmän tehon mitoitusta sekä tilantarvetta keskuskojeen viemiseksi ja sijoittamiseksi tekniiseen tilaan.

Kuvassa 4 on esitetty PLM-järjestelmä näyttöjä vaatimukseen liittyvistä näkymistä.



Kuva 4. Esimerkki vaatimuksen, sen lähteen, kohdistamisen ja jäljittämisen tiedoista PLM-järjestelmässä.

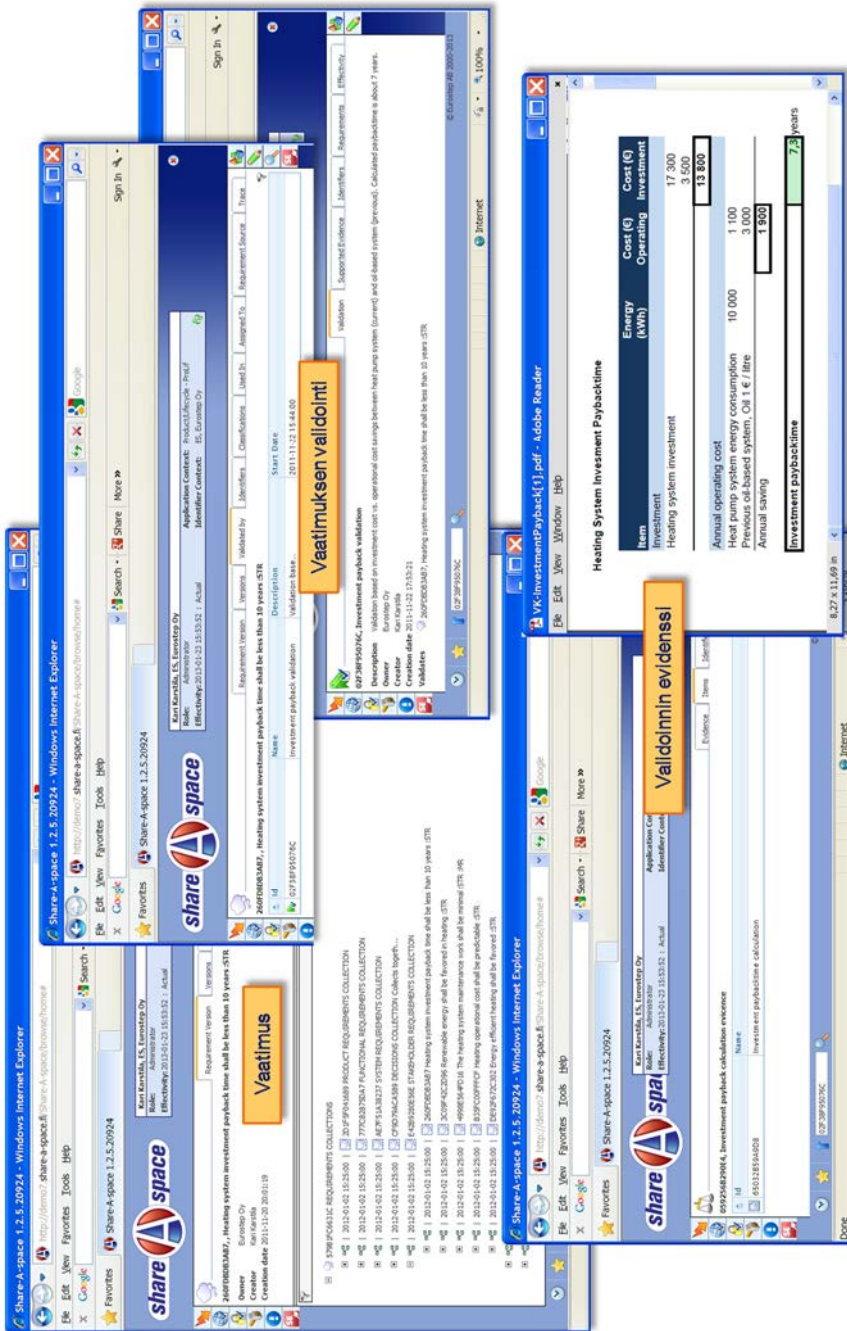
Vaatimusten verifiointi ja validointi

Toteutuksessa on PLM-järjestelmään myös tallennettu tiedot lämmitysjärjestelmän vaatimusten verifioinnista ja validoinnista. Tässä verifioinnilla tarkoitetaan yksittäisten vaatimusten toteutumisen tarkastelua ja vahvistamista objektiivisin todistein; validoinnilla tarkoitetaan tarkastelua siitä, että järjestelmä kokonaisuudessaan toteuttaa toiminnallisuuden (tai suorituskyvyn) ja vaatimukset, joita siltä odotettiin.

Yksittäisten vaatimusten verifiointiin evidenssinä toimivat erilaiset tietolähteet: mm. yleinen ja asiantuntijatieto ko. järjestelmätyypin toiminnasta, energiatehokkuuskertoimista ja mitoituksesta, tekniset piirustukset tuotteen tilan tarpeesta jne.

Lämmitysjärjestelmän keskeisin validoinnin kohde oli järjestelmän yleinen toimivuus ja sen käyttökustannukset/takaisinmaksuaika. Siten järjestelmän validointi ei ollut mahdollista heti käyttöönoton hetkellä, vaan edellytti järjestelmän käyttöä ja käytön tilastointia sähkönkulutuksen osalta joidenkin vuosien ajan. Validoinnissa todettiin vertailussa aikaisempaan järjestelmään ja sen käyttökustannuksiin takaisinmaksuajan tavoitteen (alle kymmenen vuotta) selkeää alittumisen. Käytössä järjestelmä on myös osoittautunut toimivaksi, lukuun ottamatta muutamia laiterikkoja (ks. Järjestelmän yksilöhallinta).

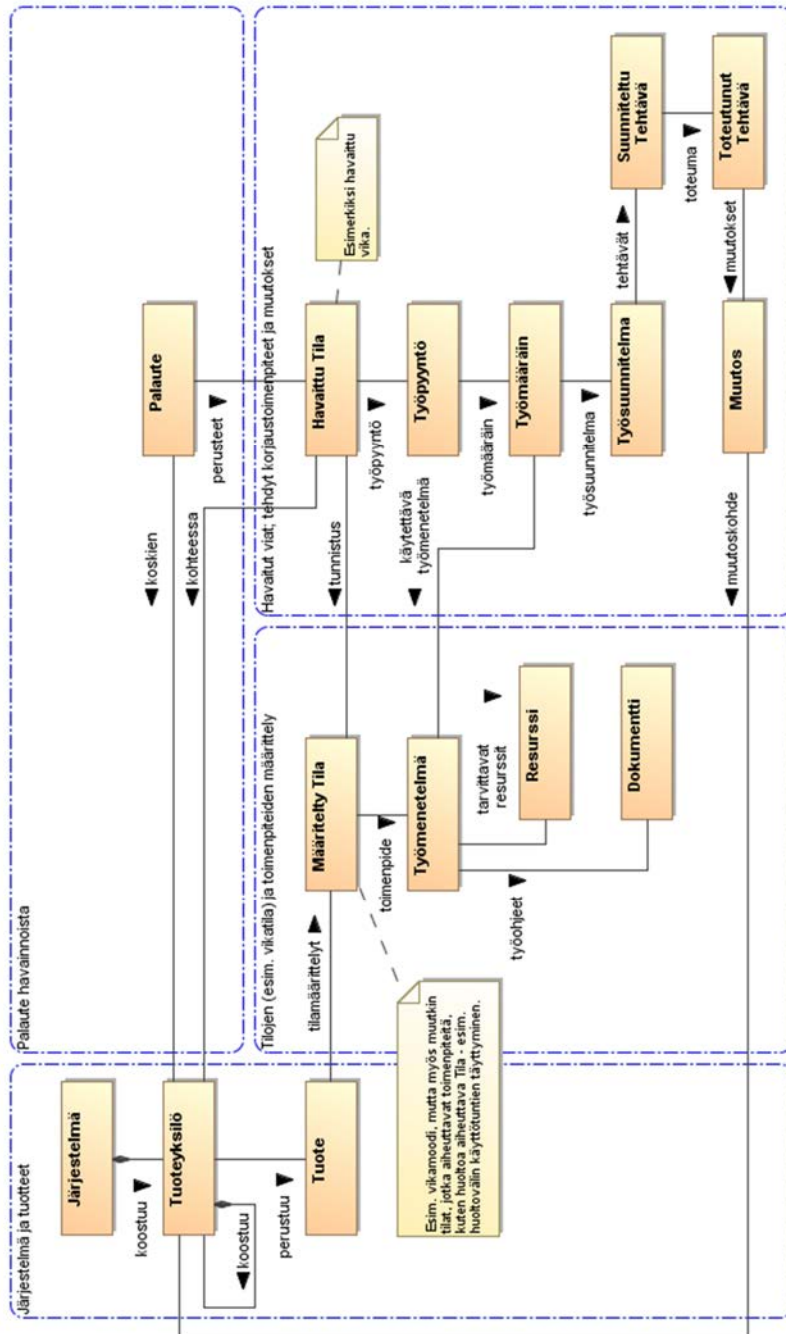
Lämmitysjärjestelmän vaatimusten verifiointit ja validoinnit (esimerkki kuvassa 5) ja niihin liittyvä aineisto on kirjattu PLM-järjestelmään ja liitetty ao. vaatimuksiin.



Kuva 5. Esimerkki vaatimuksen, sen validoinnin ja validoinnin evidenssin tiedoista PLM-järjestelmässä.

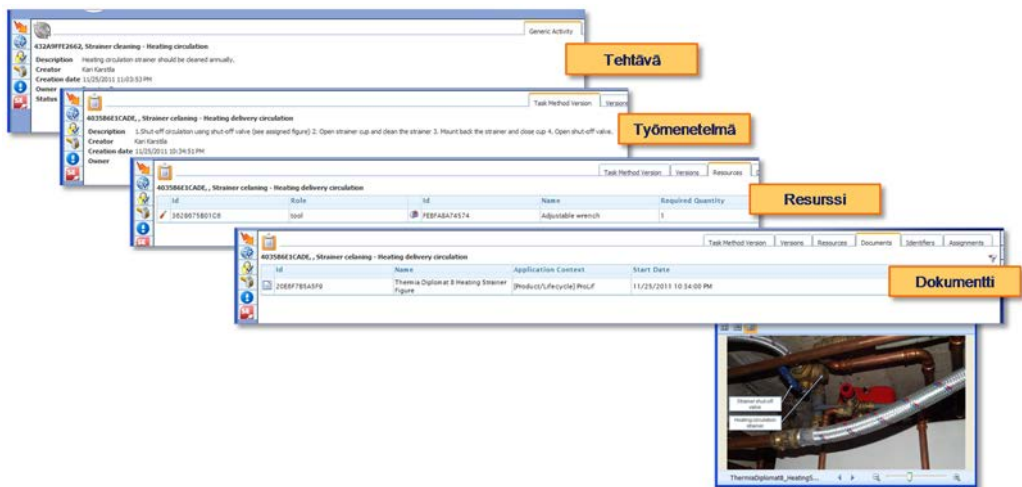
Järjestelmän yksilöhallinta

Järjestelmän elinkaaren yksilöhallinnalla tarkoitetaan tässä tapauksessa järjestelmän käytön, seurannan, vikatilanteiden, huollon ja korjauksen sekä konfiguraatiomuutosten tietojen dokumentointia ja hallintaa PLM-järjestelmän avulla. Tuoteyksilön hallintaan liittyvät keskeiset tiedot ja niiden väliset relaatiot on esitetty kuvan 6 käsitekaaviossa.



Kuva 6. Yksinkertaistettu kaavio tuoteyksilön hallintaan, erityisesti korjaukseen ja huoltoon liittyvistä tiedoista, ml. käyttökokemuksien perusteella kerätty palaute.

PLM-järjestelmän soveltamisessa on lämmitysjärjestelmän käyttöhistorian keskeiset tapahtumatiedot sekä järjestelmäkonfiguraation muutostiedot otettu hallintaan. Lisäksi PLM-järjestelmästä löytyy joitakin keskeisiä järjestelmän manuaaleja sähköisessä muodossa, mm. käyttäjämankaali ja joidenkin huoltotoimenpiteiden ohjeita. Vikatilanteiden tunnistamista varten on PLM-järjestelmään tallennettu lämpöpumppukojeen vikakoodit, niiden selitykset ja toimenpiteet niiden ilmetessä. Vähäisistä vuosittaisista huoltotoimenpiteistä on PLM-järjestelmästä löydettävissä huoltotehtävän kuvaus, tarvittavat resurssit (työkalut) sekä ohje huollon kohteesta (kuva 7).



Kuva 7. Esimerkki huoltotoimenpiteeseen liittyvistä tiedoista PLM-järjestelmässä.

Lämmitysjärjestelmän tähänastisesta käyttöhistoriasta on PLM-järjestelmään kirjattu tehdyt huoltotoimenpiteet (ml. ajankohta ja tekijä), muutama laiterikko ja niihin liittyvät pumppuyksilöiden vaihdot, sekä viimeisimpänä muutoksena lämmityskierron varaajatankin ja kiertopumpun lisäys järjestelmäkonfiguraatioon. Siten PLM-järjestelmästä voidaan todeta järjestelmäkonfiguraatio, keskeisten osien sarjanumeroineen, tietyllä ajanhetkellä koko käyttöhistorian ajalta. Laiterikkojen perusteella on myös kirjattu palaute tiettyjen pumppujen lyhyestä kestoikästä. Konfiguraation muutoksen osalta (varaajatankki) on siihen kytketty tieto muutoksen perusteluista, so. palaute eräistä käytön aikaisista havainnoista.

Tulokset

Lämmitysjärjestelmän vaatimustenhallinnan ja tuoteyksilön elinkaaren hallinnan toteutuksesta ja kokemuksista voidaan todeta seuraavaa:

- Järjestelmähankintaan liittyvät vaatimukset ja hankintaperusteet on kirjattu PLM-järjestelmään
- Lämmitysjärjestelmän käytön, huollon ja korjauksen sekä konfiguraatiomuutosten tiedot on kirjattu tähänastisesta käyttöhistoriasta kattavasti ja näin tullaan jatkossakin tekemään
- ISO PLCS -standardin määrittelemät vaatimusten, järjestelmäkuvauksen ja tuotetuen tietojen sekä niiden hallintaa tukevan PLM-järjestelmän toiminnallisuus kattaa hyvin tämän sovelluksen tarpeet vaatimustenhallinnan ja tuoteyksilön elinkaarenhallinnan osalta.

Keskustelua

Tässä esimerkkitapauksessa ei ole ollut mukana yritysverkostojen kollaboraatiönäkökulmaa, joka on kuitenkin tyypillinen useimmissa, esimerkiksi valmistavan teollisuuden PLM-toteutuksissa. Tässä esitetty toteutus on kyllä melko helposti laajennettavissa visioksi usean toimijan kollaboraatioon, jonka osapuolia voisivat olla:

- Järjestelmä- ja huoltopalveluiden toimittaja, joka päätoimijana ottaa haltuunsa toimitettujen järjestelmien yksilötiedot sekä järjestelmien elinkaaren aikaiset huolto- ja korjaustoimenpiteet ja järjestelmäkonfiguraation muutokset. Siten tehtävää huolto- tai korjaustyötä varten saataisiin ko. järjestelmän historiatiedot ja ajantasainen konfiguraatio. Vikatilanne yms. tiedot voisivat myös olla perustana palautteelle komponenttitoimittajien suuntaan.
- Komponenttitoimittajat voisivat tuoda ja ylläpitää PLM-kollaboraatiojärjestelmään omien komponenttiensa ajantasaisia tuotetietoja sekä dokumentaatiota järjestelmätoimittajan käyttöön.
- Järjestelmien käyttäjillä voisi olla pääsy oman järjestelmänsä tietoihin, kuten huoltohistoriaan, sekä mahdollisuus tehdä huoltopyyntöjä järjestelmän kautta.

Vision kaltaisen PLM-kollaboraation toteuttaminen edellyttää PLM-järjestelmältä luonnollisesti kyvykkyyttä eri osapuolien tietojen näkyvyyden ja pääsyoikeuksien hallintaan kollaboratooriota ja tiedon suojaustarpeita vastaavasti.

Ajoneuvon ergonomiavaatimukset

Juhani Viitaniemi, VTT

Peetu Valkama, VTT

Johdanto	188
Ongelman kuvaus	188
Tavoite	189
Kokonaisjärjestelmän periaate.....	189
Menetelmät	191
Järjestelmän ja sen vaatimusten määrittäminen	192
Kokonaisjärjestelmää hahmottavat lähestymistavat.....	193
Kokonaisjärjestelmän ja sen teknisen järjestelmän osan rakentumisesta	193
Järjestelmän määrittäystä ergonomian näkökulmasta	197
Järjestelmän määrittäystä toimintojen ja tehtävien näkökulmasta	197
Järjestelmän ergonomiavaatimusten johdonmukainen rakentuminen	199
Järjestelmän suunnitteluun liittyvistä vaatimusmäärittelyn menetelmistä....	199
Järjestelmän suunnittelutekniikan prosessimalli	200
Loogisiin rakenteisiin kytkeytyvät rajapintaelementit.....	201
Tietojen käsittelyssä useita tietojärjestelmiä.....	202
Suunnitelman todentaminen ja toteutuksen kelpuus.....	203
Ratkaisuehdotus ergonomiavaatimusten hallintaan	204
Käytännöt vaatimusten muodostamiseen ja hallintaan	204
Tunnistetaan ja ymmärretään kokonaisjärjestelmä.....	205
Käyttäjän toiminnan piirteiden ja hallittavan kohteen ominaisuuksien malli.....	208
Ergonomiavaatimusten ja määrittelyiden rakentuminen.....	209
Vaatimusten kytkös toimintorakenteisiin ja tehtävarakenteisiin	212
Malli ergonomiavaatimusten hallintajärjestelmäksi.....	212
Mallin DOORS -toteutus	214
Esimerkki ehdotuksen toteutusvuosta	216

Johdanto

Tämä case kuvaa ratkaisuehdotusta ajoneuvon, erityisesti ajajan alueen, ergonomiavaatimusten hallintaan. Käsiteltävät vaatimukset liittyvät ajajan tehtäviin, huoltotoimiin ja ympäristöön. Vaatimusmäärittelyssä pyrkimys on huomioida erilaisten markkina-alueiden yhteneväisyydet ja eroavaisuudet, jotka voivat ajoneuvon yksilöinti- ja räätälöintitilanteissa tuoda muutospaineita vaatimusmäärittelyiden ja sitä kautta myös tuotemäärittelyiden läpikäyntiin ja muutoksiin.

Ongelman kuvaus

Laajan ja eri kohdealueille suuntautuvan ergonomiatietämyksen hyödyntäminen ajoneuvon suunnittelussa pohjaa ergonomiavaatimusten määrittelyyn, jota puolestaan yhtäältä ohjaavat ergonomian soveltamisen periaatteet suunnittelussa sekä standardit, josta esimerkkejä ovat mm. SFS-EN ISO 9241-210 [2010], SFS-EN ISO 6385 [2004] tai SFS-EN 614-2 [2009].

Suunnittelussa ei kuitenkaan riitä, että pelkästään on kyetty muodostamaan käypä (valid) ergonomiaan liittyvä vaatimusmassa. Vähintään yhtä tärkeää on kyetä ergonomiavaatimusten hallintaan – suunnitteluvaiheen läpiviemiseksi ja suunnitelman toteuttamiseksi, myös pitkäjänteisesti mm. ajoneuvoon liittyvien asiakasprojektien tuomissa muutoshaasteissa, joissa vaaditaan uudelleensuunnittelua.

Muutoshaasteita syntyy, kun ergonomiavaatimusten täsmällisempi määrittely pitää voida jäljittää taaksepäin, jotta selviää, mikä osa suunnitelmasta säilyy ja mikä osa vaatii suunnitelman uudistamista. Miten tällainen voidaan toteuttaa – miten kytkeä vaikkapa ylätasoon vaatimuksen tai järjestelmän toiminto- ja tehtävä-rakenteiden kautta suunnittelun edetessä täsmällisempää vaatimusmäärittelyä erilaisten populaatioiden erityistietoihin tai kansallisten ja vastaavien kansainvälisten määräysten mukaisesti erityisvaateisiin tai, kun muutostarve koskee ajoneuvon toimintoja tai kuljettajalle allokoituja tehtäviä.

Tässä sovellusesimerkissä haetaan ratkaisua seuraaviin tutkimusongelmiin:

- Miten voidaan hallita vaatimusmassan yksityiskohtainen vaatimusmäärittely vastaamaan erityisesti ergonomian määrittelytiedoissa tapahtuviin muutoksiin?
- Miten erityisesti ergonomiavaatimusta koskevaa vaatimusmassan pitäisi rakentua, että muutoshallinta on tehtävissä?

Tavoite

Tavoitteena on käytäntö ergonomiavaatimusten muodostamiseen, käsittelyyn ja hallintaan. Pyrkimyksenä on, että tämän tavoitteen mukainen käytäntö liittää yhteen keskeisten suunnittelukohteen liiketoiminnallisten tavoitteiden, käyttäjätarpeiden sekä järjestelmälle asetettujen toiminnallisten ja ei-toiminnallisten tavoitteiden tehokkaan käsittelyyn. Tarkoituksena on helpottaa modulaarisen tuotteen suunnittelua ottaen huomioon suunnitteluprosessi ja suunnitelman versionhallinnan peruspiirteet ja siten helpottaa myös mm. perustuotteen yksilöintiä ja räätälöintiä kulloisenkin asiakastarpeen mukaan.

Kokonaisjärjestelmän periaate

Tässä sovellusesimerkissä tavoitteena oleva käytäntö pyrkii ottamaan huomioon koko järjestelmän (ihminen–ympäristö–tekniikka). Kokonaisjärjestelmällä tässä yhteydessä ymmärretään sitä laajempaa kokonaisuutta, johon sen osina kuuluvat perustuote (ajoneuvo) ja sen eri osajärjestelmät, kuljettajat, matkustajat ja ajoneuvon ylläpidosta vastaavat henkilöt sekä erilaiset ympäristöt, jossa perustuotteen on tarkoitus toimia. Kokonaisjärjestelmään kuuluvat siten myös ajoneuvon erilaiset käyttöympäristöt; ympäristöolosuhteet maanteistä erilaisiin maastoihin samoin kuin myös erilaiset sääolosuhteet vuoden ja vuorokauden eri aikoina. Kokonaisjärjestelmän osia ovat siis ihmiset, ympäristöt ja olosuhteet sekä erilaiset tekniset järjestelmän osat. Tässä tapauksessa tarkastelujen peruslähtökohtana on järjestelmälähtöisyys, jossa kohdennettujen analyttisten tarkastelujen rinnalla merkittävässä asemassa ovat myös järjestelmän osien väliset relaatiot ja vuorovaikutukset.

Tässä kokonaisjärjestelmän keskeistä osaa edustaa konkreettinen kohde, joka on eräs ajoneuvotuoteperheen ohjaamotilaa ja sitä voidaan pitää ajoneuvotyypille melko tavanomaisena tilana (istuin turvavöineen, jalka- ja käsikäyttöisine hallintalaitteineen, erilaisten toimintatilojen ja suoritusarvojen näyttölaitteet, tilaa rajaavat lattiat, seinät ja katto sekä tuulilasi), jossa kuljettaja suorittaa määriteltyjä tehtäviään. Tehtäviä ovat mm. tarkastukset ennen ajoon lähtöä, moottorin käynnistykseen liittyvät toimet, ajoneuvon perustoimintojen tarkistukset kuten rengaspaineet, jarrujärjestelmien toiminnot sekä ohjausjärjestelmän toiminnot.

Ohjaamotilan perusrakennetta kuvataan esimerkissä DOORS-työkalulla. Ohjaamomoduli sisältää jaottelun ajajan sekä matkustajien alueeseen. Ajajan alue on jaettu seuraavin alaotsikoin:

- Järjestelmän toiminnot (System Functions)
- Järjestelmän arkkitehtuuri (System Architecture)
- Järjestelmän suunnittelu (System Design)
- Huolto (Maintenance)
- Käytövarmuus (Availability)
- Järjestelmän kustannukset (System Cost)
- Järjestelmän paino (System Weight).

Kyseinen järjestelmän ja järjestelmän osan jäsentely noudattaa osittain suunnittelun elinkaaren mukaista jäsentelyä:

- Toiminnot (Functions)
- Arkkitehtuuri (Architecture)
- Järjestelmän suunnittelu (System Design)

ja osittain vaatimuksen tyyppin mukaista jäsentelyä:

- Huolto (Maintenance)
- Käyttövarmuus (Availability)
- Järjestelmän kustannukset (System Cost)
- Järjestelmän paino (System Weight).

Jäsentelyssä kaikkien tason otsikoiden alle on sijoitettu vaatimuksia.

Menetelmät

Suurin osa reaali maailman kokonaisuuksista tai järjestelmistä ovat kompleksisia [Peltoniemi et al. 2004]. Kompleksisuus (Complex, Complexity) on todellisuuden ominaisuus, joka on seurausta vuorovaikutuksista ja riippuvuussuhteista, joita voidaan tutkia järjestelmä lähestymistavalla (Systems Approach); havainnoija ei voi nähdä kompleksisuutta havainnoidessaan todellisuuden osia erillään muista osista ja siksi kompleksisuutta ei voi ymmärtää tarkastelemalla vain yksittäisiä osajärjestelmiä ilman niiden yhteyksiä tai vuorovaikutuksia muihin. Monimutkaisuus (Complicated, Complicatedness) on puolestaan osan, osajärjestelmän ominaisuus; siksi monimutkaisuutta voidaan havainnoida huomioimatta havainnoitavan osan vuorovaikutuksia muiden osien kanssa. Bertalanffyn mukaan järjestelmän rajausta on oltava niin laaja, että kaikki tarkasteltavan tapauksen kannalta tärkeät tekijät ovat siinä mukana; osien optimointi on liian yksipuolista ja osittaista eikä anna riittäviä optimointituloksia. [Peltoniemi et al. 2004, Ecmovic et al. 2002, von Bertalanffy 1968] Järjestelmiä voidaan kuvata muuttujien, tapahtumien, ryhmien tai ideoiden avulla [Umpleby 1994], mutta on huomattava, että kompleksisten järjestelmien kuvaaminen, mallintaminen on vaikeata eikä toisaalta ole olemassa yhtä ainoata oikeaa tapaa kuvata järjestelmiä [Peltoniemi et al. 2004].

Järjestelmän määrittelyn keskeinen tavoite on ilmaistuista elinkaaren asiakas-tarpeesta ja toiminnallisista tavoitteista, toimintakonseptin hahmotelmasta (Concept of Operations) sekä järjestelmän perustehtävästä muodostaa järjestelmän ylätasoa vaatimukset (High-Level Requirements). Näistä myös tuotteen ergonomi-aa koskevat vaatimuksien määrittelyt kehittyvät edelleen aikanaan johdetuiksi ja allokoiduiksi (Derived and Allocated Requirements) järjestelmän vaatimusmäärittelyiksi niin asiakkaan esille nostamista toiminnallisista ja suorituskyvyn tarpeista ja olemassa olevista yleisistä ergonomiatiedoista, kuin usein myös suunnittelutyön perustana olevan perustuotteen myötä kertyneistä ergonomiatiedosta sekä käyttä-jä käyttäjäkokemuksista. Suunnittelun edetessä nämä vaatimukset, järjestelmän arkkitehtuuri (järjestelmän osien loogiset rakenteet) ja toimintakonsepti sekä perustuotteen ergonomi ominaisuudet jalostuvat ensin yhtenäiseksi vaatimusmäärit-telyksi sekä suunnittelun edetessä lopulta järjestelmän tuotemäärittelyksi. [NASA 2007, Suh 1998, Savioja 2003, Sage & Armstrong 2000, ISO/IEC 15288 2008, ISO/IEC 26702 2007]

Tuoteprosessissa, lähtien asiakkaan käyttäjätarpeesta aina valmiiksi tuotteeksi käyttöön otettuna, pyrkimys on tarkastella kokonaisjärjestelmän (ihminen-ympäristö-

tekniikka) toiminnallisuuksia sekä tästä lähtökohdasta käsin erilaisten palveluiden sekä teknisten järjestelmän osien toteutusta ottaen huomioon koko järjestelmän elinkaaren vaiheet ja sen eri tilanteisiin liittyvät käyttäjät ja käyttöympäristöt. Tämä edellyttää kokonaisjärjestelmän lähestymistavan soveltamista, järjestelmäorientoitunutta suunnittelutekniikkaa ja toimintaprosessia sekä tähän soveltuvaan tietomallia. Yhdessä näiden avulla voidaan varmistua validin vaatimusmassan määrittelystä ja sen hallinnasta etenkin muutostilanteissa, sen ristiriidattomuudesta ja jäljitettävyydestä sekä tarpeellisesta, systemaattisesta dokumentoinnista. Kokonaisjärjestelmä tässä yhteydessä on ilmentymä sosioteknisestä järjestelmästä.

Järjestelmän ja sen vaatimusten määrittäminen

Kokonaisjärjestelmän tarkastelussa käytetään järjestelmälähestymistapaa (Systems Approach), jossa tarkastelun kohteena ovat myös järjestelmän eri osien väliset relaatiot ja vuorovaikutukset. Nämä piirteet ovat lisäksi tavanaomaisten yksittäisten järjestelmän osien analyttiseen tarkasteluun. Järjestelmä on ryhmä toisiinsa yhteydessä olevia osia, osatekijöitä (elements, components), jotka toimivat yhdessä, yhteisen päämäärän (purpose, objective) eteen. Järjestelmän painotus voi vaihdella ja olla palvelussa, teknologiassa tai prosessissa.

Vaatimusten ylätasoinen (High-Level Requirements) määrittely tarkentava määrittely edellyttää kokonaisjärjestelmän kuvausta toiminnallisena ja loogisena hajoitelmana (Functional and Logical Decomposition). Se voidaan tehdä ja esittää hierarkisesti (esimerkiksi kokonaisjärjestelmä–järjestelmä–osajärjestelmä–komponentit-tyyppisesti) lähtien kokonaisjärjestelmästä, josta järjestelmän osien ja osajärjestelmien sekä komponenttien tarvittavat kuvaukset, vaatimukset ja määrittelyt sitten johdetaan tai periytetään sekä allokoidaan. Tätä järjestelmän kuvausta ja sen laadintaa tukevia rakenteita on esitetty mm. ISO/IEC 15288 [2008] (IEEE Std 1220-2005) sekä ISO/IEC 26702 [2007] (IEEE Std 1220-2005) standardeissa [Doran 2005] sekä niihin pohjautuvissa käsikirjoissa kuten INCOSE SE Handbook [INCOSE 2007].

Vaatimukset ovat osaltaan lähtöisin asiakkaan ja sidosryhmien esittämistä tarpeista, osaltaan käytettävissä olevista teknologioista tai jo käytössä olevista teknologioista (perustuotteessa), osaltaan markkinoiden muodollisesta hyväksynnästä ja käyttäjien hyväksynnästä sekä osaltaan pakollisista määräyksistä ja muista tietolähteistä. Tiedon keruun kanssa täsmennetään iteratiivisesti kokonaisjärjestelmän (ihminen–ympäristö–tekniikka) kuvausta muodostaen ymmärrystä sen toiminnallisuuksista.

Kyse on itse asiassa eräänlaisesta kokonaisjärjestelmän yhdistetystä elinkaaren prosessien ja järjestelmien arkkitehtuurin mallista, joka on strukturoitu esitys vaatimuksista, käyttäytymisestä, rakenteista, ominaisuuksista ja vuorovaikutussuhteista. Mutta ergonomiavaatimusten näkökulmasta kuitenkin oleellinen seikka, joka pitää ottaa huomioon, on se, että aktiviteetti on jaettua yhteistä toimintaa käyttöliittymien ja rajapintojen sekä järjestelmän välillä. Näihin toiminnallisiin aktiviteetteihin pitää voida viitata, jotta saadaan muodostettua kytkökset toiminto- ja

tehtävärakenteiden sekä muihin keskeisten kokonaisjärjestelmän rakenteiden välille. Tässä apuna ergonomiavaatimusten näkökulmasta on myös kognitiivinen työn analyysi.

Kokonaisjärjestelmää hahmottavat lähestymistavat

Useimmat käsikirjat kuten INCOSE SE Handbook [2010] pohjaavat kokonaisjärjestelmän käsitteistössään pääosin standardin ISO/IEC 15288 [2008] esitystapaan. Tämä järjestelmän näkökulma ja ymmärrys esittelee kokonaisjärjestelmän, mutta painottuu usein vain sen tekniseen järjestelmään ja sen rakentumiseen, joka siten asiaan perehtymättä tapahtuu helposti ilman kokonaisjärjestelmänäkemyksiä ja järjestelmälähestymistapaa, jolloin ei välttämättä muodostu oikeaa kuvaa järjestelmän kompleksisuudesta (Complex) ja käytännön kehitystyö ja toteutus saattaa hämärtä todellisen järjestelmän kokonaiskuvan puutteeseen.

Kokonaisjärjestelmän ja sen teknisen järjestelmän osan rakentumisesta

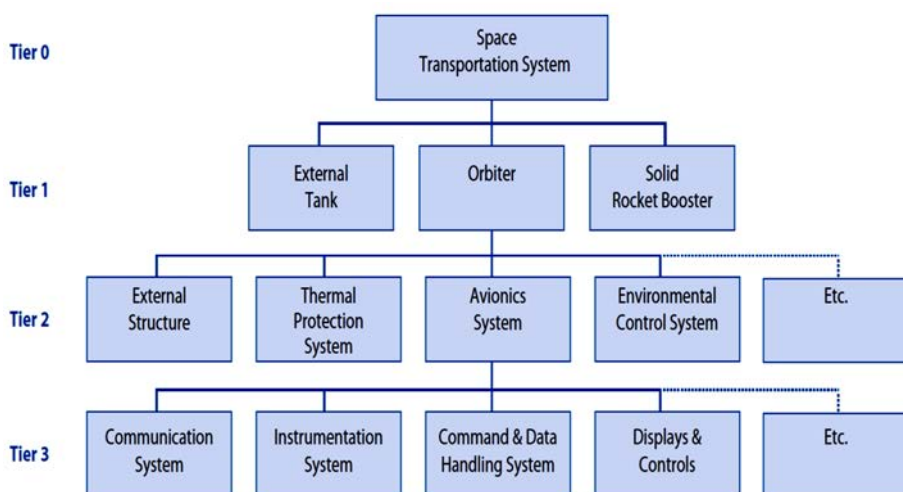
Asiakkaan käyttäjätarpeista sekä olemassa olevista, hyödynnettävistä ja käyttökelpoisista teknologioista käsin tavoite on muodostaa ymmärrys niistä järjestelmän osista, josta kyseessä oleva kokonaisjärjestelmä muodostuu ja mitä sen toiminnallisuuksilta odotetaan. Yhtenäisen ymmärryksen kannalta merkittävää ovat kysymykset, (1) miten nämä järjestelmä eri osat linkittyvät yhteen, (2) miten nämä osat toimivat erikseen ja mitä tehtäviä ne kykenevät tekemään yhdessä sekä (3) mikä on kokonaisjärjestelmän toiminnan päämäärä. Tähän liittyy myös kokonaisjärjestelmää koskevien reunaehtojen ja rajoitteiden tunnistaminen. Teknisen järjestelmän osalta määrittäminen tapahtuu iteratiivisesti osia yhteen sovittamalla ja valitsemalla siihen soveltuvat elementit, jotka toteuttavat yhdessä yhteisen päämäärän.

Jotta saadaan aikaan kokonaisjärjestelmän (ihminen–ympäristö–tekniikka) määrittäminen ja erityisesti sen teknisen järjestelmän osan suoritustason määrittäminen sekä toiminnalliset määrittäykset, on ensimmäisessä vaiheessa muodostettava ylätasoa vaatimukset (High Level Requirements) so. kaikkia koskevat ylätasoa määrittäykset (mm. koskien suunnittelun periaatteita ja menettelyitä, joita on noudatettava, sekä keskeisiä asiakastarpeista tulleita tarvetta kuvaavia ylätasoa vaatimuksia). Ne muodostavat selkärangan, jonka avulla on peilattava kokonaisjärjestelmän eri osien toimintojen rakentumista, periyttämistä ja allokointia kokonaisjärjestelmän (Functional and Logical Decomposition) eri osien välille mukaan lukien tekninen järjestelmän osa.

Kokonaisjärjestelmän toimintorakenteen määrittäminen tehdään yleensä koskien teknistä järjestelmän osaa. Tavallisesti se edellyttää toimintakonseptin kehittämistä sekä osin myös jonkin asteista teknisen järjestelmän osien loogisten rakenteiden hahmottamista ja kuvaamista mallilla (esimerkki kuva 1) sekä teknisen järjestelmän osien konseptuaalista mallintamista edelleen ratkaisuvaihtoehtoiksi kuitenkin unohtamatta kokonaisjärjestelmän muita osia.

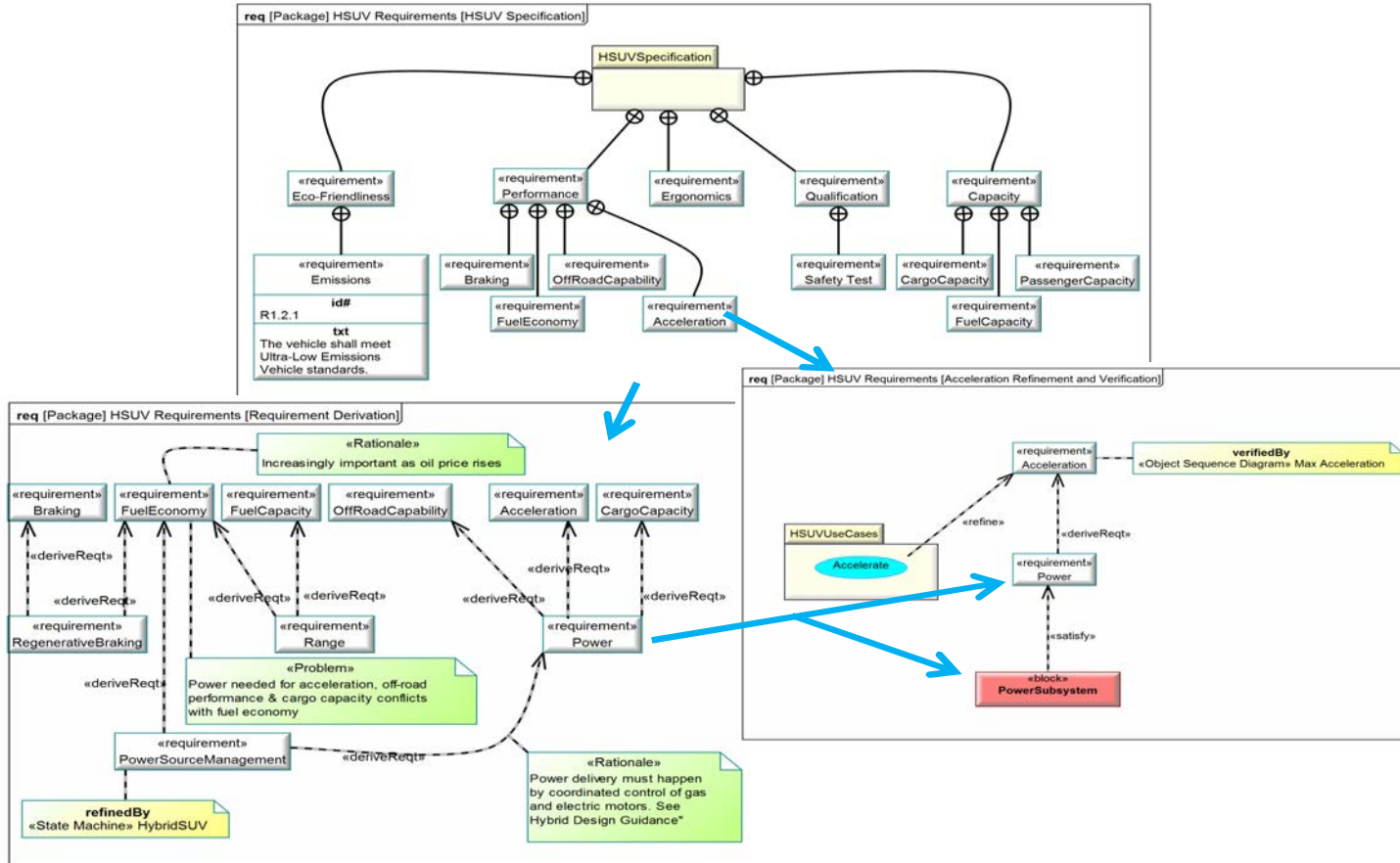
Kokonaisjärjestelmän toiminnallisuutta ja toiminnallisia ominaisuuksia voidaan mallintaa ja tarkastella myös käyttäen esimerkiksi IDEF0 -notaation pohjautuvaa toimintomallia. Se kuvaa dynaamista prosessia kuten toimintoa, aktiviteettia, toimintaa tai tehtävää. IDEF0:n peruspiirre on kyky muodostaa ja kuvata toimintojen hajoitelmia.

Kokonaisjärjestelmän toiminnallisia ominaisuuksia voidaan mallintaa ja tarkastella myös käyttäen esimerkiksi SysML-mallinnusta [OMG SysML 2011, Artisan Studio SysML Tutorial]. SysML -mallinnus lähtee liikkeelle esimerkiksi vaatimuksien määrittämisestä (kuva 2), joita täydennetään mm. toimintojen ja rakenteen kuvaamisella sekä käyttötapausten kuvauksilla (kuva 1) sekä johdettujen vaatimusten määrittämisellä ja toimintojen tarkentamisella. Mallinnus rakentuu iteratiivisesti ja sitä muodostettaessa voidaan tarkastella järjestelmän vaatimusten rakentumista, toteutumista ja relaatioita suhteessa järjestelmään eri toimintoihin.

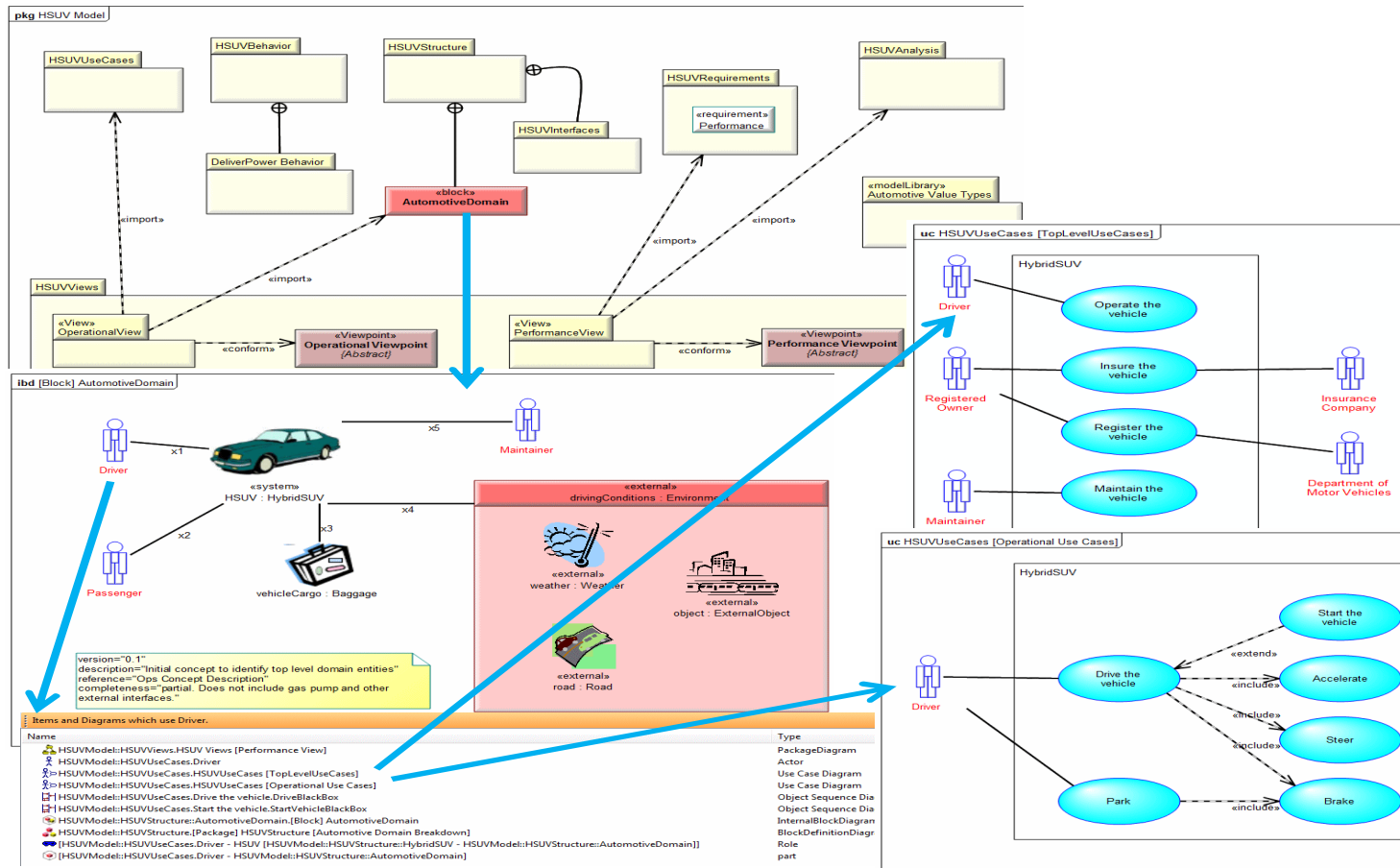


Kuva 1. Tyypillinen teknisen järjestelmän osien, loogisten rakenteiden hajoitelma vaihe vaiheelta pienempiin järjestelmän osiin [NASA 2007].

SysML-mallinnusta käyttäen voidaan tunnistaa järjestelmän toimintojen ja tehtävien kytkeytymistä järjestelmän osien välisten rajapintojen ja rajapintaelementtien kautta. SysML tukee myös tapaa muodostaa niitä formaalisti sekä johdonmukaisesti allokoida järjestelmän toimintoja järjestelmän eri osille mukaan lukien käyttäjät. Kun kokonaisjärjestelmän eri osilla kullakin on selkeä sille allokoitu toiminto tai tehtävä, se tukee myös järjestelmän toiminnallista ja rakenteellista modulaarisuutta.



Kuva 2. Järjestelmän vaatimuksien määrittystä SysML-mallinnusta käyttäen [Artisan Studio Uno].



Kuva 3. Järjestelmää ja sen toimintojen määrittystä SysML-mallinnusta käyttäen [Artisan Studio Uno].

Järjestelmän määrittystä ergonomian näkökulmasta

Ergonomiaa koskevien vaatimuksien (shall/mandatory) osalta ylätason vaatimuksia koskevia perusteita, niitä tukevia ohjeita ja viittauksia suositeltaviin käytäntöihin löytyy mm. standardeista SFS-EN ISO 9241-210 [2010], SFS-EN ISO 6385 [2004] ja SFS-EN 614-2 [2009]. Esteettömyyttä koskevia ohjeita ja vaatimuksia on mm. standardeissa SFS-EN ISO 9241-20 [2009] ja SFS-EN ISO 9241-171 [2009].

Vastaavanlaisia yksityiskohtaisempia määrittelytietoja sisältäviä standardeja on olemassa eri maantieteellisille alueille (markkina-alueille). Erot eri standardeihin juontuvat mm. paikallisten populaatioiden eroista (antropometriset mittauerot, joista johtuen mm. erilaiset ulottuvuudet, muut mitoitukselliset erot sekä vastaavasti voimantuottoon ja kestävyysvaikutteeseen vaikuttavat erot), joilla voi tietyissä tilanteissa olla suurikin merkitys.

Pelkästään standardien sisältämien vaatimusten ja niistä johdettujen tapauskohtaisten täsmällisempien määrittelyjen ongelmana on etenkin ergonomiavaatimusten hallinnan näkökulmasta niiden luettelomainen luonne. Ne eivät vielä sellaisenaan muodosta selkeää perustaa, keinoa tai rakennetta koko vaatimusmassan tai edes ergonomiavaatimusmassan hallintaan, vaatimusten jäljitettävyyteen tai niiden ristiriidattomuuden selvittämiseen.

Niinpä selkeä tarve olisi tehdä järjestelmän toimintorakenteesta (a) johdonmukainen toimintojen allokointi myös tiettyihin rooleihin omistautuneille henkilöille, toimijoille järjestelmässä ja (b) samalla kytkeä se myös heidän tehtävärakenteensa relevantteihin kohtiin. Tämä mahdollistaisi sen, että ergonomiavaatimuksia voidaan linkittää siten, että ne ovat jäljitettävissä ja muutoksia tehtäessä tiedetään tai voidaan helposti etsiä muutkin muutostarpeiden kohteet. Edellytyksenä on järjestelmämalli, jossa on esitettyä kyseiset järjestelmään kytkeytyvät tehtävärakenteet, toimintorakenteet ja loogiset rakenteet.

Järjestelmän määrittystä toimintojen ja tehtävien näkökulmasta

Kognitiivisen työn analyysi (Cognitive Work Analysis, CWA) [Vicente 1999] on viisivaiheinen metodologia monimutkaisten sosioteknisten järjestelmien suunnitteluun, jota on sovellettu työkohteen analyysin (1. vaihe) ja hallintatehtävien analyysin (2. vaihe) osalta sekä vähäisessä määrin myös organisaation ja yhteistyön analyysin (4. vaihe) osalta [Jenkins 2008].

Kognitiivisen työn analyysin tavoitteena on sellaisen järjestelmän kehittäminen, joka sallii poikkeamat rutiineista. Järjestelmässä on tietyt reunaehdot, rajoitteet sille, mitä missäkin tilanteessa voidaan tehdä, mutta ei välttämättä edellytetä tiettyä (teknistä tai muuta toiminnallista) ratkaisumallia aina tiettyyn tilanteeseen. Juuri tällaista mm. tämän työn kohteena olevan ajoneuvon ajaminen ja muut henkilöiden tekemien tehtävien pitäisi ehkä ollakin, jotta muuttuneeseen asiakastarpeeseen liittyen perustuen voidaan tehdä tarvittavat muutokset joustavasti.

Kognitiivisen työn analyysin ensimmäisen vaiheen, työkohteen analyysin (Work Domain Analysis, WBA) tavoitteena on selvittää systemaattisesti ympäristön työlle

asettamat reunaehdot ja rajoitteet. Tuotoksena on määritelmä järjestelmän ympäristön lähtökohdille so. suunniteltavalle järjestelmälle asetettavia vaatimuksia. Nämä vaatimukset ovat työntekijästä riippumattomia esim. fyysisen tai sosiaalisen ympäristön asettamia ja erityisesti ohjattavasta prosessista johdettavia vaatimuksia.

Kognitiivisen työn analyysin ensimmäisen vaiheen pyrkimys on kuvata työympäristö mallina siltä osin, joka pysyy muuttumattomana järjestelmän tai ohjattavan prosessin tilasta riippumatta. Samaan malliin voidaan liittää tieto siitä, mikä informaatio on merkityksellistä käyttäjän kannalta. Mallinnustekniikkana voidaan käyttää esimerkiksi ADS (Abstraction-Decomposition Space) -mallia. Se on kaksiulotteinen taulukko, jossa vaaka-akselilla kuvataan mallinnettavan järjestelmän hajotelma (decomposition) koko järjestelmästä osajärjestelmiin ja komponentteihin asti ja pystyakselilla kuvataan abstraktiohierarkia (AH) eli järjestelmän tavoitteiden eri abstraktiotasoja korkeimmasta alkaen; korkeimmat tasot kuvaavat järjestelmää tarkoituksen ja toiminnallisten tavoitteiden kautta ja alemmat kuvaavat järjestelmän fyysisen toteutuksen ominaisuuksia – AH kuvaa kuilua järjestelmän tarkoituksen ja fyysisten ominaisuuksien välillä, se ei kuvaa mitään erityistä tapahtumaa tai käyttäjän tehtäviä, koska ne eivät auta käyttäjää selviytymään ennakoimattomissa tilanteissa. Malli kertoo vain mitä käyttäjä tekee.

Kognitiivisen työn analyysin toisen vaiheen hallintatehtävien analyysin (Control Task Analysis, CTA) tavoitteena on jaksollisten aktiviteettien yksityiskohtaisempi panos-tuotostehtäväanalyysi, jossa se sekä ympäristön analyysi liittyvät tiukasti toisiinsa. Ympäristön jälkeen analysoidaan tehtävät, joita kyseisessä ympäristössä tulee suorittaa. Tällä tavalla tehtävät saadaan pysymään yhteydessä käyttökontekstiinsa. Kognitiivisen työn analyysissä lähtökohta on järjestelmän kokonaistavoitteet eikä käyttäjäyksilöiden lähtökohdat. Tavoitteena hallintatehtävän analyysissä ovat vaatimukset, jotka liittyvät tunnistettuihin tapahtumiin ja tavoitteisiin tunnetussa ympäristössä. Monimutkaiset sosiotekniset järjestelmät ovat avoimia systeemejä, josta syystä panos-tuotos mallia käytetään tehtäväanalyysissä, koska mahdolliset ulkoiset häiriötekijät vaikuttavat tehtäviin. Siksi häiriötekijöistä riippuvia ratkaisuvaihtoehtoja on olemassa useita. Hallintatehtävien analyysissä mallinetaan toimintoja, joita kuvataan verbeillä, mutta vain siltä osin, mitä tehtävillä pitää saavuttaa, ei sitä, miten se saavutetaan. Pyrkimyksenä on, että kehitetyt ratkaisut ovat laite-, tekniikka- yms. riippumattomia.

Kognitiivisen työn analyysin toisen vaiheen eräitä mallinnustyökaluja ovat CTA-mallit ja päätöstikapuut (Decision Ladder), jonka avulla voidaan mallintaa tehtävien panostuotossuhteita; mikä on edellytys suunnitteilla olevan tehtävän suorittamiselle ja toisaalta sitä, mitä suunniteltu tehtävä tuottaa panokseksi seuraavalle tehtävälle. Malli kertoo, miksi käyttäjä toimii niin kuin hän toimii.

Tässä esimerkissä keskitytään kognitiivisen työn analyysin suhteen kuitenkin mainittuihin kahteen ensimmäiseen vaiheeseen, työkohteen ja hallintatehtävien analyysin, ja niiden erilaisten analyysimenetelmien käyttöön, joiden avulla pyritään muodostamaan perustaa ergonomiavaatimusten rakentumiselle. Eräs keskeinen perusta on tunnistaa mitä käyttäjä tekee ja miksi käyttäjä toimii niin kuin hän toimii; näin saadaan muodostettua työhön ja tekemiseen liittyvät rajat ja reunaehdot sekä

edelleen näihin liittyvät erityiset rajapintaelementit (toimintorakenteen ja tehtävära-
kenteen konseptuaalinen kytkös).

Muita kognitiivisen työn analyysin vaiheita ovat:

- strategian analyysi (miten yksittäinen työntekijä edellisessä vaiheessa määritellyt tavoitteet saavuttaa)
- organisaation ja yhteistyön analyysi (miten eri strategioihin pohjautuvat on-
gelmanratkaisumallit jaetaan ihmisen ja automaation välillä)
- työntekijän osaamisen analyysi (tavoitteena on tunnistaa työntekijöiden eli
järjestelmän käyttäjien kognitiivisten kykyjen asettamat reunaehdot järjes-
telmän suunnittelulle).

Järjestelmän ergonomiavaatimusten johdonmukainen rakentuminen

Käyttäjätiedon avulla kehitetään ja luodaan tietoa ergonomiavaatimusten määrittelyn sekä järjestelmän tarkan spesifikaation tueksi. Tämä on prosessi, joka jatkuu koko järjestelmän tuotekehityksen elinkaaren ajan vaatimusten hallintana; vaatimusmäärittelyssä esitetyt vaatimuksia käytetään joka tapauksessa hyväksi tuotekehitysprosessin elinkaaren eri vaiheissa. Myös vaatimuksissa ilmenneisiin muutostarpeisiin on voitava vastata tuotekehitysprosessin elinkaaren myöhemmissäkin vaiheissa; järjestelmän uudelleen suunnittelu on esimerkki tästä. Järjestelmän ominaisuuksia koskevat päätökset on siis voitava jäljittää taaksepäin aina vaatimusmäärittelyn vaatimuksiin asti.

Tarkoituksena on muodostaa ergonomiavaatimusten hallintaan edellä esitettyjen analyysien tulosten valossa yhteys järjestelmän toimintorakenteen ja sen loogisten rakenteiden sekä tehtävärakenteen välille. Tässä hyödynnetään toimintojen allokointia teknisten järjestelmän osien sekä henkilöiden tehtäviksi. Tämä luo ensimmäisen yhteyden rakenteiden välille.

Konseptuaalisilla, yleisillä teknisen järjestelmän rajapintaelementeillä luodaan tämän lisäksi sisältörikkaampi yhteys toimintorakenteen ja tehtävärakenteen välille ja edelleen yhteys vaatimusmäärittelyjen täsmentämisen kautta järjestelmän suunnitteluratkaisuihin sen eri vaiheissa. Näin suunnittelun ratkaisusta (tuotemäärittely) saadaan muodostettua jäljitettävyyttä takaisin aina vaatimusmäärittelyihin ja niitä edeltäneisiin vaatimuksiin asti ja pitämällä vaatimusprosessi aktiivisena myös myöhempi muutoshallinta on mahdollinen tapa jatkaa asiakastarpeiden edellyttämiin suunnitelmamuutoksiin.

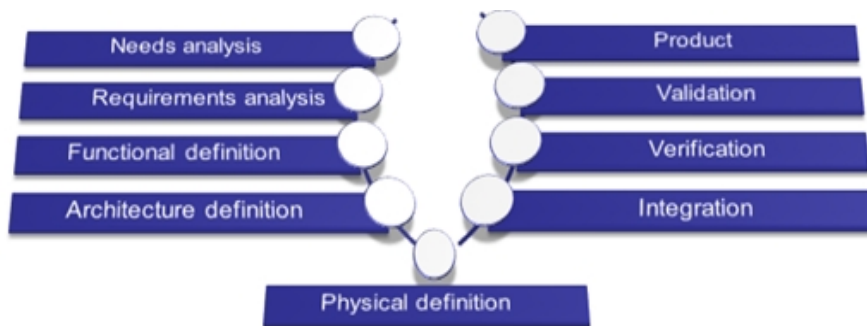
Järjestelmän suunnitteluun liittyvistä vaatimusmäärittelyn menetelmistä

Järjestelmän suunnittelussa V-mallin mukaisella suunnitteluvuolla eteneminen on yhdenaikaista osavaiheiden prosessien toteutusta rinnakkaisesti; aloittava vaihe voi vaihdella ja vaiheet eivät suinkaan välttämättä aktivoidu suunnitteluvuon mu-

kaisessa vaihejärjestyksessä. Alussa on usein olemassa jo jonkinlainen teknologinen ratkaisumalli tai perusta, joka toimii suunnittelun lähtökohtana, pohja-aihiona ja otetaan järjestelmän suunnittelussa analyysi- ja määrittäytöyön perustaksi. Tällöin on jo heti alusta alkaen olemassa koko joukko vaatimuksia sekä niihin pohjautuvia täsmällisempiä määrittäytöksiä.

Kyse on selkeästi rinnakkaisesta, yhdenaikaisesta, iteratiivisesti etenevästä suunnittelusta eikä peräkkäisvaiheiden mukaisesta suunnittelusta. Tämä tarkoittaa, että V-mallin mukaiset suunnitteluvuon vaiheet ovat heti alusta alkaen lähes tulkoon kaikki aktiivisia ja niiden välinen iteratiivinen työ on suuressa määrin heti alusta lähtien käynnissä. Tilanne koskee myös ihmisestä lähtöisin olevia ergonomiavälitteisiä toimintoja ja vaatimuksia sekä hyödynnettävän teknologian käytettävyyden ergonomialle aiheuttamia paineita, oli kyse sitten täysin uuden tuotteen suunnittelusta tai jo olemassa olevan tuotteen uudelleen suunnittelusta tai kehittämisestä; merkittävä osa tavoitteista, tarpeista ja toimintavaatimuksista on löydettävissä jo olemassa olevista käyttäjätarpeista.

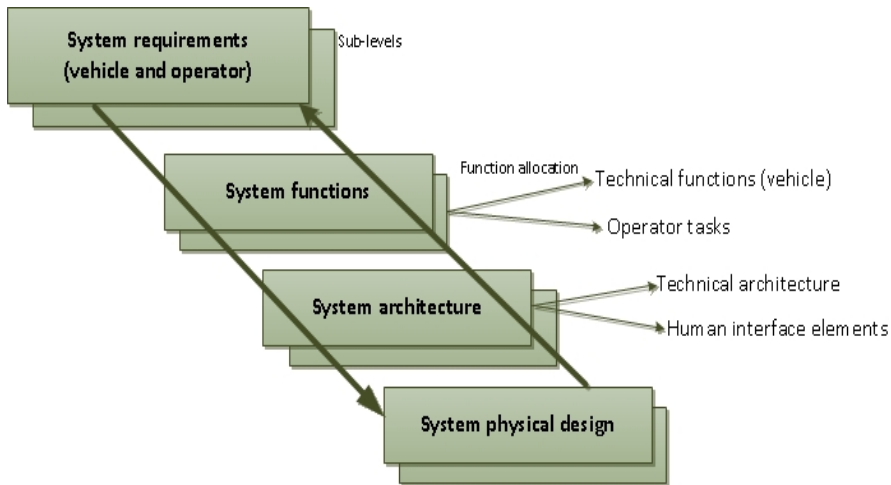
V-mallin mukainen suunnitteluvuon (kuva 4) tukee järjestelmä lähestymistapaa (Systems Approach) sekä on osa järjestelmän suunnittelua, järjestelmän suunnittelutekniikkaa (Systems Design, Systems Design Engineering, saksalaisten käyttämiä nimityksiä termille Systems Engineering).



Kuva 4. Esimerkki systeemisuunnittelun V-mallista.

Järjestelmän suunnittelutekniikan prosessimalli

Kuvassa 5 nähdään osa ISO/IEC 15288 [2008] sekä ISO/IEC 26702 [2007] standardeihin perustuvasta Systems Engineering -prosessimallista (SEP, Systems Engineering Process). Tässä prosessimallissa on korostettu kahden järjestelmän osan eli teknisen järjestelmän ja käyttäjän toiminnallisuuksia ja näiden toiminnallisuuksien allokoimista. Kun nämä järjestelmän osien toiminnot allokoidaan teknisiin toimintoihin ja operaattorin toimintoihin, tuotteen ergonomiata koskevat vaatimukset on helpompi tunnistaa, jäljittää ja todentaa. Melkoinen osa vaatimuksista tulee myös kolmannesta järjestelmän osasta eli ympäristöstä, joka tässä prosessimallissa ajatellaan kuuluvan osaksi järjestelmää (System).

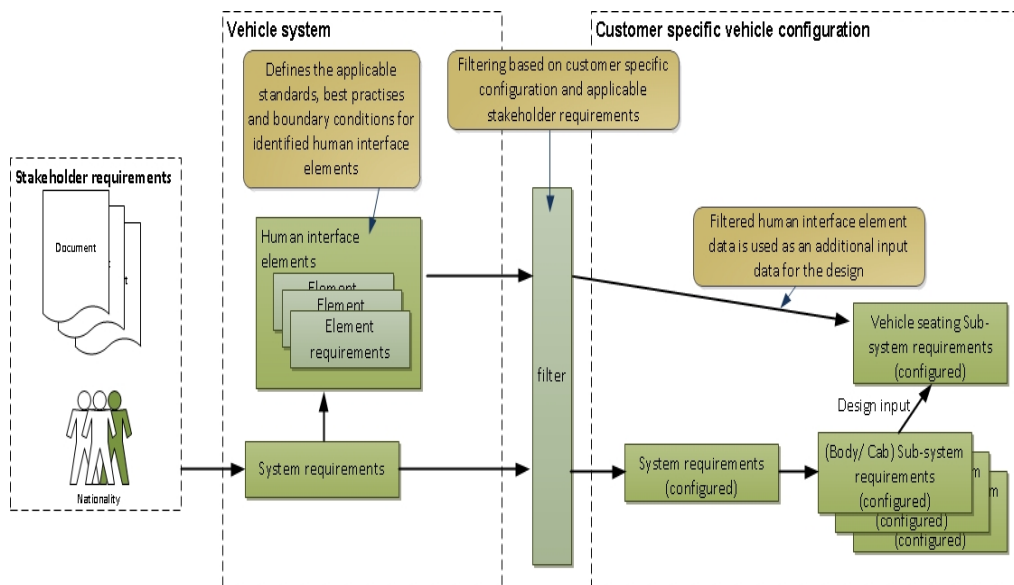


Kuva 5. Erityisesti teknisen järjestelmän osan vaatimusten kehittyminen suunnitteluprosessin aikana lopulta sen tuotemäärittelyksi.

Loogisiin rakenteisiin kytkeytyvät rajapintaelementit

Teknisen järjestelmän loogisen rakenteen, arkkitehtuurin määrittelyssä hyödynnetään edellä tunnistettuja kokonaisjärjestelmän toiminnallisuuksia. Näin arkkitehtuurin suunnittelussa voidaan määrittellä tehokkaasti mm. käyttäjän ja hänen toimintaympäristönsä väliset erilaiset rajapinnat. Näihin rajapintoihin liittyy tyypillisesti joukko teknisiä rajapintaelementtejä sekä myös muun tyyppisiä rajapintaelementtejä.

Rajapintaelementtien geneeriset määrittelyt kootaan järjestelmätasolla tietokantaan (kuva 6). Muita kuin teknisiä rajapintaelementtejä ovat tehtäviin kytkeytyvät rajapintaelementit mm. näkymät ja muut yhteydet ohjaamotilasta, äänitila ja kommunikation sisällöt ja ohjaamon olosuhteet sekä käyttäjän eri tilanteista muodostuvat tehtävien tavoitteet. Teknisten rajapintaelementtien yleinen määrittely sisältää yhteyden niitä koskevaan tietojärjestelmään, sen tietokantoihin; kirjallisuuteen, standardeihin, parhaisiin käytäntöihin ja reunaehtoihin. Tiettyä ajoneuvon kokoonpanoa varten tekniseen järjestelmän osaan liittyvät vaatimukset suodatetaan, samalla suodatetaan myös näihin teknisiin rajapintaelementteihin sekä muihin rajapintaelementteihin sisältyvät ergonomiavaatimukset. Suunnittelija hyödyntää siten tietokannasta löytyviä erilaisten rajapintaelementtien määrittelyjä varsinaisen vaatimusmassan lisäksi.



Kuva 6. Kokonaisjärjestelmän eri osien vaatimukset muodostavat asiakasspesifisesti konfiguroidun tuotteen suunnittelun lähtötiedot ja toisaalta niihin liittyvien vaatimusten muutoshallinnan.

Lähtökohdallisesti teknisen rajapintaelementin (esimerkiksi istuimen) loogisen rakenteen, arkkitehtuurin määrittelyssä ei otettaisi kantaa yksityiskohtaiseen tekniseen toteutukseen, vaan määriteltäisiin näihin elementtiin liittyvät vaatimukset, reunaehdot ja ohjeet. Vasta teknisen järjestelmän fyysisessä suunnittelussa määriteltäisiin istuimen tarkka tekninen toteutus, huomioiden tietokannasta löytyvät määritelmät kyseiselle elementille. Niin tekniset kuin muutkin rajapintaelementit voidaan koota koskemaan myös muita tuotteita, ääni-, lämpötila-, valaistus- tai näkyvyysolosuhteita, kommunikaatiota tai vaikkapa käyttäjäkokemuksia, näin ergonomiavaatimukseen liittyvää tietomäärää voidaan hallita paremmin ja eri järjestelmän osien modulaarisuutta voidaan parantaa.

Tietojen käsittelyssä useita tietojärjestelmiä

Tietojärjestelmä on yrityksen käyttöön räätälöity kokoelma tietokantoja ja muita tiedonlähteiden linkkejä. Tässä tapauksessa se sisältää mm. standardointijärjestöjen ja niiden kautta erilaisten standardien tietokantoja, kirjallisuutta menetelmistä ja ergonomiatietoudesta. Keskeinen osa tietojärjestelmää on myös suunnittelun tietojärjestelmä, yleisemmin tuotetiedon hallintajärjestelmä (PDM, Product Data Management) tai laajempi tuotteen elinkaaren hallintajärjestelmä (PLM, Product Life cycle Management). Monista eri tekijöistä johtuen tärkeäksi muodostuu myös toiminnanohjausjärjestelmä (ERP, Enterprise Resource Planning).

Järjestelmän vaatimusten määrittelyiden näkökulmasta tietojärjestelmien välinen sujuva yhteistoiminta on tärkeää samoin kuin niiden prosessimistajien välinen kitkaton yhteistyö selkeän toimivan työnjaon ja yhteisen päämäärän saavuttamiseksi. Erityisesti tämä tulee esiin suunniteltavan järjestelmän koko elinkaaren aikaisten vaatimusten määrittelyjen rakentumisessa, niiden sisältämien ristiriitojen tunnistamisessa ja niiden poistamisessa.

Keskeisiä tietojärjestelmien tietokannoista on mm. vaatimustietokanta, suunnittelun tietokanta sekä informaatiolähteiden tietokanta. Informaatiolähteiden tietokanta pitää yllä tietoa kumulatiivisesti mm. vaatimuksissa ja suunnittelussa käytetyistä standardeista ja niiden muutoksista. Vaatimustietokanta pitää sisällään viittauksia myös suunnittelun ja informaatiolähteiden tietokantoihin.

Suunnitelman todentaminen ja toteutuksen kelpuus

Vaatimusmäärittely on iteratiivisesti etenevä suunnitteluun kytkeytyvä prosessi. Vaatimusten ja niiden täsmällisempien määrittelyjen kehittäminen edellyttää suunnittelun edetessä tapahtuvaa suunnitelman todentamista, verifiointia vertaamalla ja testaamalla sekä myös toteutuksen kelpuutusta. Nämä vaatimukset pitää myös määrittellä osana kokonaisjärjestelmän vaatimuksia.

Ratkaisuehdotus ergonomiavaatimusten hallintaan

Seuraavassa on pyritty kuvamaan erästä mahdollista ratkaisuehdotusta ergonomiavaatimusten muodostamiseen ja hallintaan. Se koostuu käytännöistä ergonomiavaatimusten rakentumiseksi, vaatimusten käsittelystä täsmällisemmiksi määrittelyiksi sekä sen lisäksi tietomallista vaatimusten hallintajärjestelmäksi.

Käytännöt vaatimusten muodostamiseen ja hallintaan

Käytäntöjen toteutus jakautuu seuraaviin pääosin rinnakkaisesti eteneviin osiin:

1. Tunnistetaan ja ymmärretään kuvaamalla kokonaisjärjestelmä (ihminen–ympäristö–tekniikka) kuten sen käyttötilanne, käyttäjän toiminnan piirteet ja käyttäjien tarpeet [SFS-EN ISO 9241-210 2010, SFS-EN ISO 6385 2004, SFS-EN 614-2 2009, Savioja 2003], mm.
 - a. nostamalla esiin järjestelmän monimutkaisuutta ja epävarmuutta lisääviä piirteitä käytön, toiminnan tai ymmärrettävyyden näkökulmasta
 - b. nostamalla esiin työn kannalta merkityksellisiä käyttäjän toiminnan piirteitä
 - c. lähestymällä järjestelmää sillä hallittavan kohteen ominaisuuksien kautta.
 - d. ymmärtämällä ympäristöt ja olosuhteet, jossa järjestelmä on käytössä
2. Luodaan kokonaisjärjestelmästä malleja [Savioja 2003, Vicente 1999], mm.
 - a. kuvaamalla käyttäjän tarvitsemaa informaatiota
 - b. kuvaamalla hallittavasta kohteesta saatavaa informaatiota
 - c. kuvaamalla järjestelmän piirteitä ja yhteistoiminnallisuuden sekä vuorovaikutteisuu den luonnetta
 - d. tunnistamalla järjestelmän ja sen osien reunaehdoja ja rajoitteita.

3. Määritellään rajapintaelementit sekä järjestelmän toimintorakenteet ja tehtävärakenteet mm.
 - a. kuvaamalla käyttäjän liitynnät muihin järjestelmän osiin
 - b. kuvaamalla järjestelmän toiminnot
 - c. kuvaamalla käyttäjälle allokoituvat tehtävät
4. Yhdistetään järjestelmän rakenteiden mallien tiedot vaatimuksiin ja vaatimusmäärittelytietoihin.

Vaatimuksia kokonaisjärjestelmätasolle ja sen osille muodostettaessa on tärkeää, että (a) vaatimukset rakentuvat formaalisesti siten, että ne määrittelevät johdonmukaisesti tietyt järjestelmän reunaehdot ja rajoitteet sekä (b) määrittelysisältöön liitetään myös tietoa, jolla muutosten vaikutusta voidaan selvittää ja jäljittää kuten yhteydet toiminto- ja tehtävärakenteisiin. Vaatimukset selkeyttävät mitä käyttäjän tekemillä tehtävillä pitää saavuttaa rajoittamatta miten se saavutetaan. Vaatimusmäärittelyllä pyritään tuomaan esiin järjestelmältä edellytettävät keskeiset ominaisuudet. Tuotemäärittely toteuttaa vaatimukset ja kuvaa ratkaisun.

Tunnistetaan ja ymmärretään kokonaisjärjestelmä

Menetelmällisesti tässä vaiheessa hyödynnetään ryhmämenetelmää, erityisesti osallistavaa suunnittelua ja kehittämistä. Ryhmän koko kannattaa pitää melko pienenä: n. 3–6 henkilöä koostuen eri alueiden avainhenkilöistä. Ryhmä hyödyntää aiemmassa kappaleessa kuvattuja muita menetelmiä tarpeen mukaan tehdesään analyysijä.

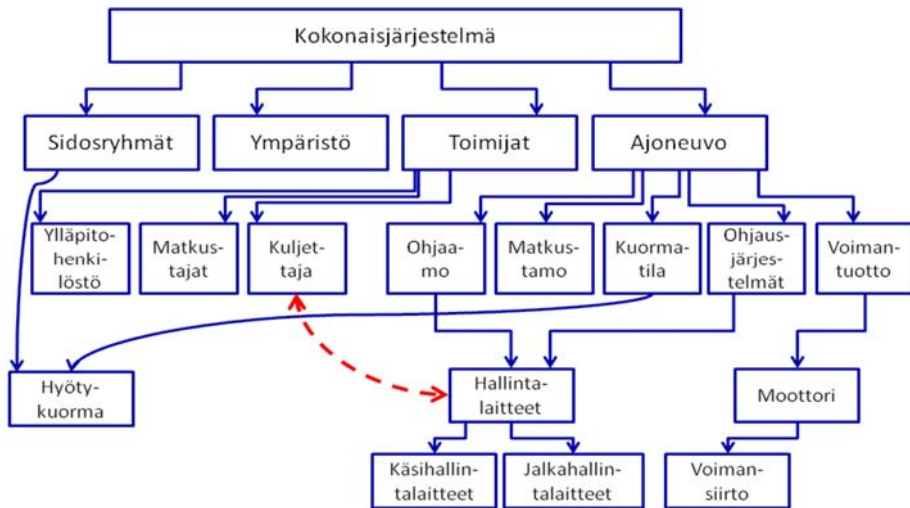
Kokonaisjärjestelmä (ihminen–ympäristö–tekniikka) koostuu painotetusti esimerkeissä kuljettajasta, maastosta ja sen kohteista sekä ajoneuvosta. Kohteena oleva ajoneuvo on miehitetty. Sillä on tarkoitus kuljettaa ympäri vuoden erilaisissa sää- ja maasto-olosuhteissa kulloisenkin tehtävän mukaiseen kohteeseen miehistöä ja hyötykuormaa. Ajoneuvon huoltoa ja muuta ylläpitoa ja siihen kytkeytyviä henkilöitä ja tehtäviä samoin kuin miehistön tehtäviä ei käydä läpi sen koommin tässä esimerkeissä, mutta niiden käsittelyperiaatteet ovat samoja.

Käyttäjien toiminnan ymmärtämisen menetelmiä ovat pääasiassa testaaminen ja käyttäjien toiminnan hahmottaminen, jossa lähestymistapana käytetään mm. toiminta- ja käyttötarinoita; niistä saatavan tiedon hyödyllisyyden vuoksi pitää tarinoita koota myös oikeilta käyttäjiltä. Tarinoiden avulla voidaan löytää myös usein käyttökelpoisia termejä tarvittaville toiminnoille. Tarinoista usein selviää myös toiminnan saavutettavuudelle asetettavia tavoitteita. Tarinoiden avulla saadaan myös tietoja käyttötilanteista, tehtävistä, mahdollisuuksista ja rajoitteista sekä tavoitteista.

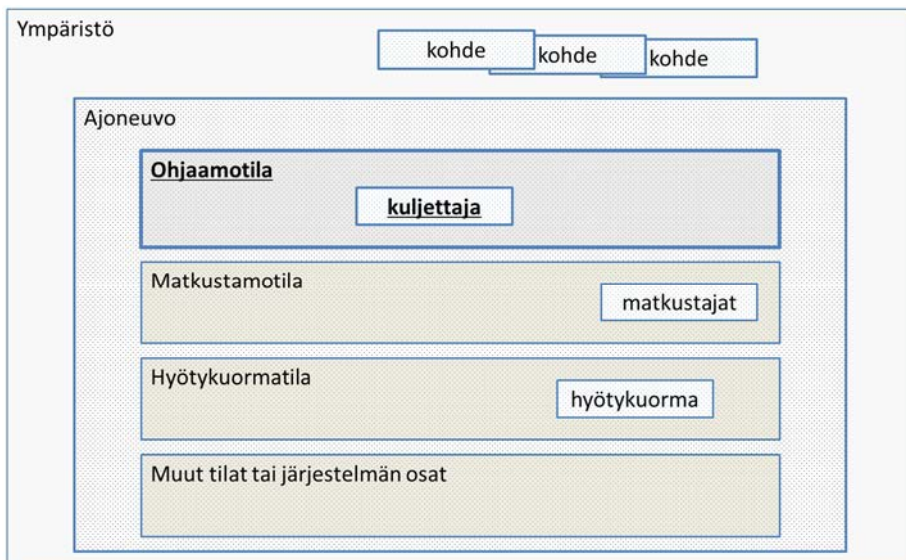
Edellä olevan pohjalta ryhmämenetelmällä on kerätty tietoa ja sitä analysoidessa on käynyt ilmi, että kuljettajan tehtävä sisältää ainakin seuraavia piirteitä ja hallittavia kohteita:

1. ajotehtävä ja kuljetustehtävä kumpikin edellyttävät valmistelujen suorittamista lähtöpaikalla mm.
 - a. kuljetettava hyötykuorma ja matkustajat on saatava ajoneuvoon mukaan ja on huolehdittava niiden turvallisesta sijoittamisesta ajoneuvon tiloihin
 - b. yhteyshenkilöt on selvitettävä sekä mahdolliset kommunikointitavat
 - c. ajoreitti, välikohteet ja loppukohde sekä matkan pituus on selvitettävä navigointia varten
 - d. polttoaineen riittävydestä on varmistuttava samoin kuin mahdollisista välitankkauskohteista.
2. ennen liikkeelle lähtöä on tehtävä ajoneuvolle tietyt perustarkistukset, säädöt, ajoneuvon käynnistys ja lämmityskäyttö mm.
 - a. nestetasojen yms. tarkistukset
 - b. hallintalaitteiden, informaationäyttöjen sekä sähköjärjestelmien ja tietojen syöttölaitteiden toimivuus sekä säädöt kuljettajalle sopiviksi
 - c. kommunikointi- ja navigointilaitteiden toimivuus, asetukset ja säädöt
 - d. varmistettava, että liikkeellelähtö voi tapahtua turvallisesti.
3. ajotilanteessa on hallittava ajoneuvon käyttö mm.
 - a. hallintalaitteiden avulla pysyttävä ajoreitillä ja sovitettava kulkunopeus eri tilanteisiin
 - b. oltava tarpeen mukaan kommunikointiyhteydessä ulkopuolisiin tahoihin ja kyettävä suorittamaan vaadittuja muutoksia tehtävään ja sen suoritukseen tai suoritusjärjestykseen
 - c. navigoitava tehtävän mukaisiin kohteisiin
 - d. suoritettava käyntikohteissa määritellyt tehtävät
 - e. saatettava ajoneuvo loppukohteessa asianmukaiseen pysäköintipaikkaan
4. tehtävän suorituksen lopuksi palautettava ajoneuvo oikeaan tilaan ja tarpeen mukaan raportoitava tehtävän suoritus.

Näiden tietojen pohjalta on selvää, että ajoneuvon ajamiseen liittyviä teknisen järjestelmän toimintoja on allokoitava kuljettajan vastuulle suoritettavaksi tehtäviksi. Kuvat 7 ja 8 esittävät alustavia kokonaisjärjestelmän loogisten rakenteiden hajoitelmia. Kokonaisjärjestelmä muodostuu ihmisistä, ympäristöstä ja tekniikasta.



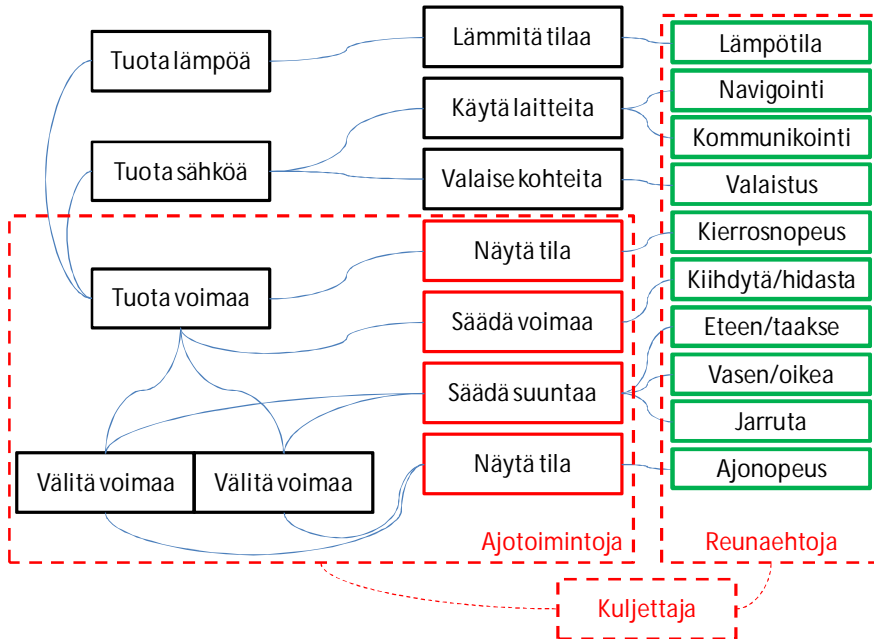
Kuva 7. Esimerkki kokonaisjärjestelmän rakenteiden karkeasta alustavasta hajoitelmasta.



Kuva 8. Esimerkki kokonaisjärjestelmän rakenteiden karkeasta alustavasta hajoitelmasta.

Tässä kokonaisjärjestelmän loogisten rakenteiden hajoitelmassa on useita järjestelmän osia kuten ajoneuvo, ympäristö ja muita kohteita ympäristössä, kukin niistä on kokonaisjärjestelmän osa. Ajoneuvo puolestaan koostuu teknisistä osajärjestelmistä ohjaamotilasta sekä matkustamo-, hyötykuorma- ja muista tiloista ja järjestelmän osista. Kun ajoneuvoa käytetään tarkoitettuun toimintaansa, sen ohjaamotilassa on kuljettaja, joka myös on silloin kokonaisjärjestelmän osa (kuva 9).

Kuljettaja tarvitsee hänelle allokoitujen toimintojen, tehtävien suorittamiseksi siis, tiettyjä hallintalaitteita (tässä ne ovat myös rajapintaelementtejä), jotka muodostavat kokonaisjärjestelmän toimintojen ja kuljettajan tehtävien välille kytköksen. Näiden rajapintaelementtien avulla on tarkoitus myös muodostaa ergonomiavaatimusten muutosten hallintaan soveltuva menetelmä. Toisin sanoen ihmiseen liittyvät ergonomiavaatimukset siirretään tällä tavalla, rajapintaelementtistatuksella, osaksi teknistä järjestelmää.



Kuva 9. Eräitä kokonaisjärjestelmän toimintoja ja niiden osittaista allokoointia kuljettajalle reunaehtoineen.

Käyttäjän toiminnan piirteiden ja hallittavan kohteen ominaisuuksien malli

Ergonomiavaatimusten muodostaminen perustuu tiedon keruulle ja analysoinnille mm. kognitiivista työn analyysin avulla sekä käsittelylle asiantuntija-arvioita hyödyntäen. Sen lisänä hyödynnetään ryhmämenetelmää. Ryhmämenetelmän idea on, että kun käyttäjiä on muutama pohtimassa samaa asiaa, he inspiroivat toisiaan ja näin saadaan nostettua esille enemmän tietoa ja yksittäisiä mielipiteitä kuin silloin, kun vain yhtä henkilöä haastatellaan kerrallaan. Menetelmän onnistumisen edellytys on syntyvän keskustelun hyvä henki ryhmässä sekä se, että ryhmähaastatteluja pidetään vähintään kaksi eri kokoonpanoilla. Ryhmässä yhden käyttäjän ei saa antaa dominoida keskustelua. Ns. hiljaisista tiedoista ryhmämenetelmällä on vaikea saada esiin; hiljaiselle tiedolle määritelmän mukaan on ominaista se, että

se pystytään ilmaisemaan ainoastaan luonnollisessa käyttötilanteessa. Tällaisissa tilanteissa simulaatioita ja virtuaalisia ympäristöjä voidaan hyödyntää erilaisten artefaktien ohella käyttäjävaatimusten esiin nostamisessa konseptuaalisissa käyttötilanteissa. Simulointien avulla voidaan aktivoida käyttäjiä tuottamaan uusia ideoita.

Tässä esimerkkitarkastelussa kokonaisjärjestelmä kuvataan vaatimusten hallinnan näkökulmasta aluksi karkeasti, ks. esimerkki (kuva 8), jolla päästään alkuun. Tämä kuvaus tai malli noudattaa soveltuvia yleisperiaatteita kokonaisjärjestelmän hierarkian rakennetasoista (kokonaisjärjestelmä, järjestelmä, alijärjestelmä, komponentit), jotta se on yhtenevä myöhemmin käytettävään täsmällisempään rakennepuumalliin; tässä esimerkissä hyödynnetään DOORS-ohjelmaa kokonaisjärjestelmän kuvaukseen, ks. luku Mallin DOORS-toteutus, s. 217. Siinä kokonaisjärjestelmän rakenteen jäsentelyssä sen eri rakennetasoilla (L1...L5) esitetään niin teknistä järjestelmää, sidosryhmiä, ympäristöä kuin toimijoita. Tämän hierarkian rakennetaso on vaatimusmassan hallinnassa myös yksi vaatimusten hakuavain, hakutieto.

Ergonomiavaatimusten ja määrittelyiden rakentuminen

Ajoneuvon toimintojen tarkastelussa on mm. päädytty siihen, että ajoneuvon ajamiseen liittyviä toimintoja on allokoitu kuljettajan suoritettaviksi tehtäviksi. Standardeista voidaan suoraan johtaa ylätasoa vaatimuksia suunnitteluperiaatteille ja menetelmille, joita suunnittelussa on noudatettava ergonomiaan liittyvien vaatimusten toteuttamiseksi riittävässä määrin henkilöiden työskentelyolosuhteiden takaamiseksi. Standardeista löytyy myös suosituksia sille, mitä fyysisiä raja-arvoja, säätöalueita, toiminta-alueita tai ulottuvuusalueita kunkin hallittavan tehtävän suorittaminen edellyttää.

Kognitiivisen työn analyysissa vaatimusten määrittely ei ole erillinen kokonaisuus, vaan koko prosessi käsittelee sitä, miten erilaisia järjestelmävaatimuksia muodostetaan työtä analysoimalla. Aluksi tehdään työkohteen analyysia laatien siitä abstraktiohierarkkinen malli (AH, ADS), jossa työkohteena olevia järjestelmän osia tarkastellaan kokonaisuutena ja hajoitelmana järjestelmän osiin osa ja komponentitasoille asti (kuva 10). Tarkoituksena on luoda ymmärrys siitä mitä käyttäjä tekee sekä mitä mahdollisuuksia järjestelmän osat tarjoavat selviytymisessä myös ennakoimattomissa tilanteissa. Käyttäjän tehtäviä ei kuvata tässä normatiivisella tavalla vaan pyrkimys on päästä formatiiviseen tekemisen kuvaamisen tuoden esiin ympäristön rajoitteet käyttäjän tekemiselle. AH/ADS-mallin suhteen kuvassa (kuva 10) oleva malli on eräs ratkaisu, muitakin ratkaisuja on olemassa. Malli voidaan tehdä myös tilanteesta, jossa tarkastelun kohteena on useampi ajoneuvo ja esimerkiksi komentokeskuksen järjestelmä. Myös abstraktiotasojen lukumäärää voi vähentää jättämällä jonkin pois/yhdistämällä niitä. Seuraavaksi tehdään hallintatehtävän analyysi ja laaditaan AH-malli (kuva 11).

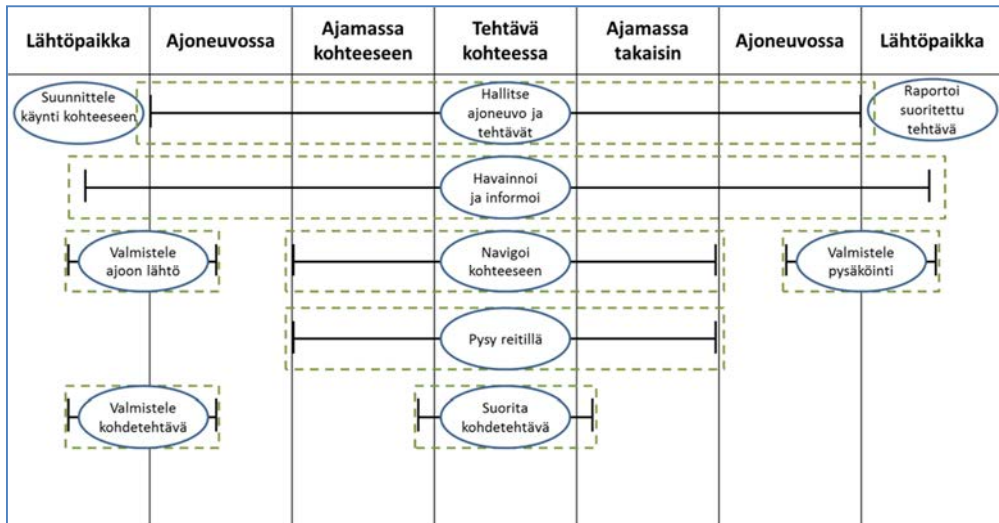
Työkohteen ja hallintatehtävien analyyseistä (ADS ja CTA) käy selville aihioita vaatimuksiksi, jotka liittyvät tunnistettuihin tapahtumiin ja tavoitteisiin ympäristössä ja ovat riippumattomia itse työntekijästä. Työntekijästä riippumattomat vaatimukset ovat ympäristön asettamia rajoitteita ja prosessin asettamia tavoitteita kuten aikarajoja tai vasteaikoja.

<i>Hajotelma</i> <i>Abstraktio</i>	Järjestelmä	Alijärjestelmä	Osa, komponentti
Toiminnallinen tarkoitus	Ajaminen aiottuun kohteeseen ja takaisin		
Abstrakti toiminto / Arvot ja priorisoinnit	Ajon valmistelu Ajon toteuttaminen Tehtävistä huolehtiminen Joustavuus Sovittaminen	Kommunikointi Navigointi Ajonhallinta Rinnakkaisten aktiviteettien hallinta	
Yleistetty toiminto / Tarkoitukseen liittyvät toiminnot		Hyötykuorma Tehtävävarustelu Informaatiot ja datat Ajoituslaskenta	
Fyysinen toiminto / Kohteeseen liittyvät toiminnot		Ajamisen toiminnot Ajamisen tukitoiminnot Resurssien seuranta Suorituksen seuranta	Ajamisen seuranta Tila ja rajoitteet Reitti ja kohteeseen saapuminen Olosuhdemittaus
Fyysinen muoto / Fyysiset kohteet		Ohjaamotila	Ympäristön seurannan ja navigoinnin laitteet Istuin ja hallintalaitteet Kommunikointilaitteet Valaisimet

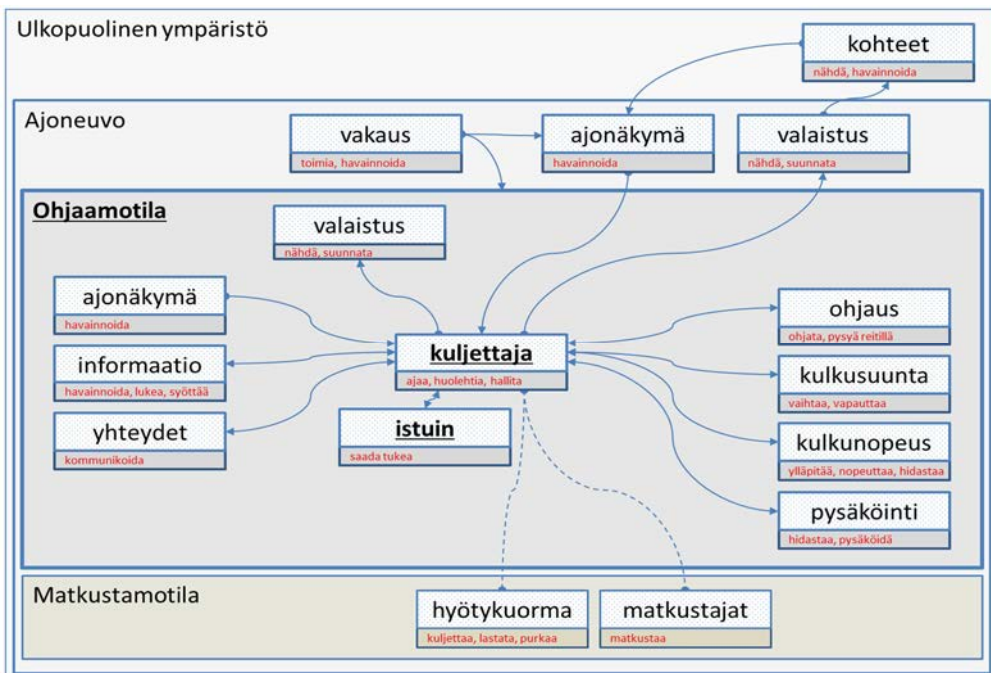
Kuva 10. Esimerkki järjestelmän AH/ADS -mallista.

Ajoneuvossa pitää olla ohjaamotila, jotta mainittuja toimintoja voidaan suorittaa suojassa ulkoympäristön rasitteilta. Ohjaamotila on suunnittelun tekemä ratkaisuvaihtoehto logiseksi rakenteeksi. Ohjaamotilassa pitää olla myös istuin, josta käsin kuljettaja voi hallita ajoneuvoa (kuva 12). Istuin on eräs tapa ratkaista kuljettajan tarve saada tukea hallintatehtävien suorittamiseen. Istuin on suunnittelun tekemä ratkaisuvaihtoehto logiseksi rakenteeksi. Kuten huomataan, istuin on eräs varsin keskeinen tekijä kuljettajan ohella suunnittelun vaatimia määrittelyjä täsmennettäessä. Ajoneuvon hallintaa (kuva 12) varten pitää olla hallintalaitteita kuten kulkusuunnan ohjauslaite, kulkusuunnan ja nopeuden hallintalaite, katseluyhteys kulkusuuntiin ja reitillä pysymiseen sekä myös tarpeelliset informaation, navigointi ja kommunikointi laitteet. Nämä hallintalaitteet ovat myös suunnittelun tekemiä ratkaisuvaihtoehtoja logisiksi rakenteiksi. Suunnittelun tekemien loogisten rakenteiden ratkaisuvaihtoehtojen määrittelyä on edelleen täsmennettävä rajoitteilla, raja-arvoilla, mitoitusarvoilla ja tavoitearvoilla.

Testien suorittamisessa hyödynnetään ohjaamon toiminnallisen tarkoituksen kuvausta sekä toimintorakenteiden ja tehtävärakenteiden linkittymistä (kuva 10 ja kuva 12). Kutakin hallintatehtävän osaa (kuva 11 ja kuva 12) voidaan tarkentaa skenaariokuvauksella ja käyttötapauskuvauksella.



Kuva 11. Esimerkki järjestelmän CTA-mallista.



Kuva 12. Ohjaamotilan toimintojen ja kuljettajan tehtävien tarkastelua, kun keskeisten loogisten rakenteiden, ohjaamotilan ja istuimen valintoja on jo tehty.

Vaatimusten kytkös toimintorakenteisiin ja tehtävärakenteisiin

Määritellään tunnistetut rajapintaelementit edellä tehtyjen analyttisten tarkastelun pohjalta; tässä rajapintaelementtejä ovat ne hallintaelimet, jotka välittävät käyttäjän aktiviteetteja teknisiin osajärjestelmiin. Muita kuin teknisiä rajapintaelementtejä ovat tehtäviin kytkeytyvät rajapintaelementit kuten näkymät ja muut yhteydet ohjaamotilasta, äänitila ja kommunikaation sisällöt ja ohjaamon olosuhteet sekä käyttäjän eri tilanteista muodostuvat tehtävien tavoitteet. Rajapintaelementin kuten istuimen, hallintaelimien ja näkymää tuottavan ikkunan tai periskoopin loogisen rakenteen, arkkitehtuurin määrittelyssä ei vielä oteta kantaa sen yksityiskohtaiseen tekniseen toteutukseen, vaan määritellään näihin elementtiin liittyvät ylätasoon vaatimukset, reunaehdot ja ohjeet; vasta suunnittelun edetessä määritetään johdettujen vaatimusten määrittelyt, jotka ohjaavat myös tehtäviä suunnittelun valintoja.

Rajapintaelementtien yleinen määrittely tehdään DOORS-tietojärjestelmään, sen tietokantoihin. Sen jälkeen tiettyä ajoneuvon kokoonpanoa varten tekniseen järjestelmän osaan liittyvät vaatimukset suodatetaan edellä tehdyn analyttisen tarkastelun pohjalta. Suunnittelija voi siten hyödyntää DOORS-tietojärjestelmästä löytyviä erilaisten rajapintaelementtien määrittelyjä varsinaisen vaatimusmassan määrittelyssä; teknisen järjestelmän fyysisessä suunnittelussa suunnittelija määrittelee istuimen tarkan teknisen toteutuksen, joka perustuu tietokannasta löytyvään istuimen rajapintaelementin perusmäärittelyihin. Keskeistä tässä on se, että noissa perusmäärittelyissä luodaan myös yhteys ihmisen roolin tehtävien suorittajana ja sitä kautta muodostetaan yhteys teknisen järjestelmän toimintorakenteisiin ja toisaalla DOORS-tietojärjestelmässä määritettyihin teknisen järjestelmän toimijoiden tehtävärakenteisiin; täten mahdollistetaan jäljitettävyyden ja muokattavuuden ergonomiavaatimusten hallinnassa.

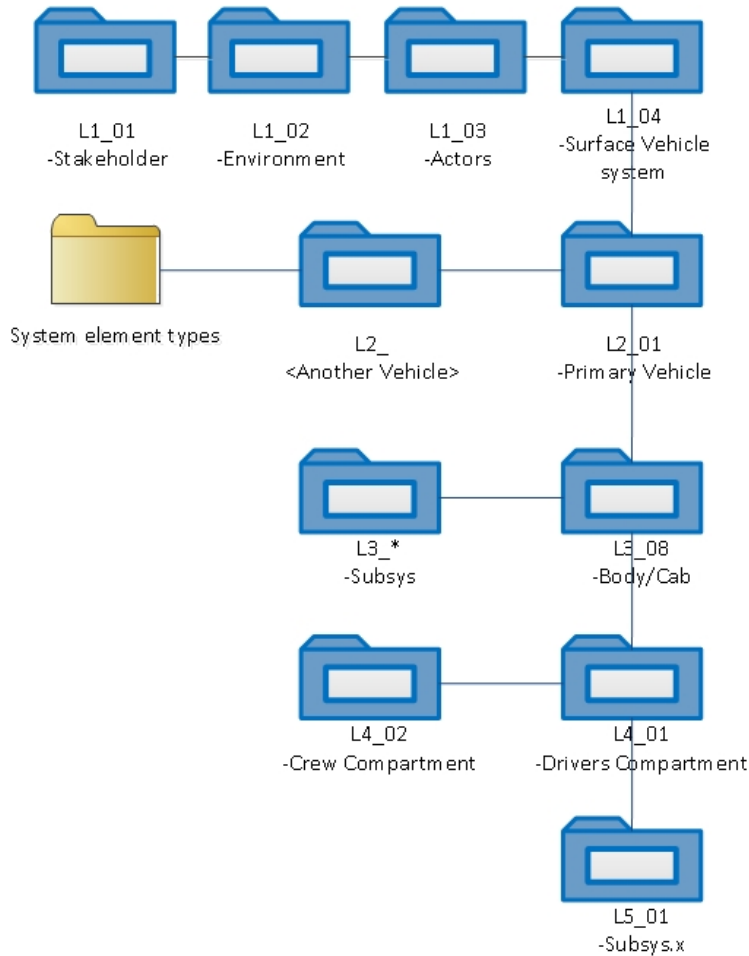
Malli ergonomiavaatimusten hallintajärjestelmäksi

Ehdotettu järjestelmä ja rakenteen jäsentely ergonomiavaatimusten hallintaan DOORS-toteutuksena on esitetty kuvassa 13. Siinä L1-tasolle on teknisen järjestelmän osan (L1_04) ohella lisätty muut järjestelmän osat eli sidosryhmät (L1_01), ympäristö (L1_02) ja toimijat (L1_03).

Teknisen järjestelmän osan tasolla (L2) hyödynnetään rakenneosatyypikansiota johon kootaan teknisen järjestelmän osaan liittyvien yhteisten komponenttien määrittelyt. Rakenneosatyypikansioon kootaan myös yhteisten teknisten ja muiden rajapintaelementtien määrittelyt, kyseinen rajapintaelementtien periaate on esitetty myös kuvassa 6 (Human interface elements) sivulla 210 teknisten rajapintaelementtien osalta. Ajoneuvossa saattaa olla esimerkiksi useita ovia, joista jokainen on sijoitettu eri puolille ajoneuvoa. Jokainen ovi esitetään ko. järjestelmän arkkitehtuurin määrittelyssä (SBS).

Ehdotetussa toteutuksessa on käytetty DOORS-ohjelman projekteja ja kansioita, projekti kuvaa aina järjestelmää. Käyttämällä projekteja järjestelmä on helpommin

ylläpidettävissä (projektista voidaan tehdä arkistovarmuuskopio, kansioista ei) ja myös ulkoasu on selkeämpi.



Kuva 13. Ehdotettu DOORS-toteutuksen kansiorakenne.

Jokaiseen projektiin liitetään alla esitetty kansiorakenne:

- Vaatimukset (Requirements)
- Toiminta (Behaviour)
- Arkkitehtuuri/ Rakenne (Architecture)
- Suunnittelu (Design)
- Todennus ja kelpuus (Verification & Validation).

Ehdotettu kansiorakenne mukaillee V-mallin mukaista suunnitteluvuota, asiakkaan tarpeiden määrittely (Needs analysis) on sijoitettu DOORS-rakenteen päätasolle

L1_01 Stakeholder/Needs. V-mallissa esitetty fyysinen määrittely/ toteutus ja järjestelmän integrointi on yhdistetty suunnittelukansioon ehdotetussa DOORS-toteutuksessa (kuva 15). Jokaiseen kansioon on liitetty moduuleja, jotka ovat listattu alla.

Vaatimuskansion sisältö

- Dokumenttimoduuli (DOC, Document)
- Vaatimusmoduuli (SyRS, System Requirements Specification)

Toimintakansion sisältö

- Dokumenttimoduuli (DOC, Document)
- Toiminnallisuuden määrittely (FNS, Functional Specification)
- Käyttötapausten / skenaarioiden määrittely (UCS, Use Case Specifications / Scenarios)

Arkkitehtuurikansion sisältö

- Dokumenttimoduuli (DOC, Document)
- Järjestelmän osittelu (SBS, System Breakdown Structure)

Suunnittelukansion sisältö

- Dokumenttimoduuli (DOC, Document)

Todennus- ja kelpuutuskansion sisällöt

- Dokumenttimoduuli (DOC, Document)
- Testitapausten määrittelyt (TCS, Test Case Specifications)
- Todennuksen ja kelpuutuksen tulokset (VVR, Verification/ Validation Reports).

Mallin DOORS-toteutus

DOORS-toteutuksen rakenne noudattaa päätasolla kohteen mukaista jäsentelyä ja suunniteltavan järjestelmän tasolla (järjestelmät/rakenneosat) elinkaaren mukaista jäsentelyä (kuva 14).

Kuvassa 15 tätä perusrakennetta laajennetaan tuoterakenteella. Tässä esimerkissä on käytetty julkisesti saatavilla olevaa DoD:n julkaisemaa tuoterakennetta [MIL-HDBK-881 1998].

L1_01 Stakeholder

- Dokumenttimoduuli (DOC, Document)
- Asiakkaan vaatimukset ja tarpeet (NDS, Customer Needs)

L1_02 Environment

L2_01 Mission Environment

L2_02 Logistics Environment

L2_03 Tactical Environment

L1_03 Actors

- Toimijoiden määrittely (mm. kieli, kansallisuus, populaatio)

L2_01 Vehicle Crew

Driver

- Tiettyyn käyttötapaukseen liittyvän toimijan valinta (ACS, Actor Specification)
- Käyttäjän toimien määrittely (UAS, User Action Specification)
- Käyttötapauksen / skenaarioiden määrittely (UCS, Use Case Specifications / Scenarios)

L2_02 Maintenance Crew

L1_04 Surface Vehicle System

Requirements

Behaviour

Architecture

Design

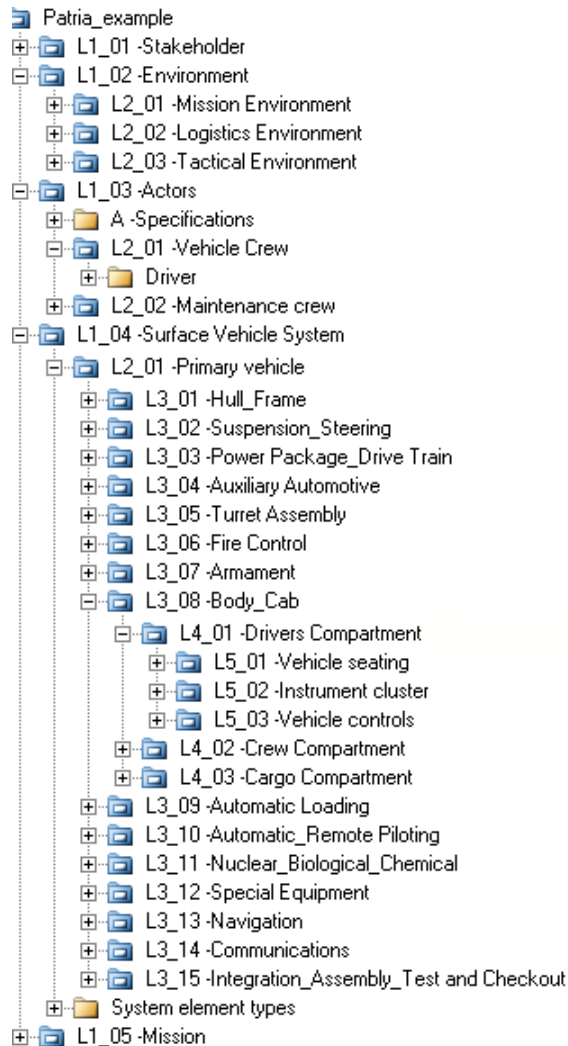
Verification and Validation

L2_01 Primary Vehicle

System element types

L1_05 Mission

Kuva 14. DOORS-toteutuksen perusrakenne.



Kuva 15. DOORS-toteutus noudattaa standardia perusrakennetta ja mukaillee tyypillistä tuoterakennejakoa, jollainen esimerkiksi ajoneuvolla voisi olla.

Esimerkki ehdotuksen toteutusvuosta

Seuraavassa on esitetty pelkistetty vuo siitä, miten ehdotettua ratkaisua voidaan soveltaa ergonomiavaatimusten hallintaan. Mainitut vuon pääkohdat eivät ole perättäisiä toimintoja vaan ne etenevät rinnakkaisesti ja iteratiivisesti vuorovaikutuksessa toisiinsa.

Määritä kokonaisjärjestelmä (ihminen–ympäristö–tekniikka):

- Kognitiivisen työanalyysi
 - Käyttäjän toiminnan piirteiden ja hallittavan kohteen ominaisuuksien malli
 - Aktiviteetit, toiminnot, reunaehdot
 - Toimintaympäristöt, reunaehdot ja rajoitteet
- Kokonaisjärjestelmän mallit sisältäen erilaisia näkymiä kuten
 - Fyysiset yhteydet
 - Toiminnalliset yhteydet
 - Rajapintojen aktiviteetit
 - Ympäristö- ja sidosryhmämallit

Laadi tietojärjestelmään tarvittavat rakenteet:

- Teknisen järjestelmän tuoterakenne
 - Toimintorakenne
 - Rajapintaelementtien rakenne
- Toimijarakenne
 - Tehtäväraakenne
- Sidosryhmärakenne
- Ympäristörakenne

Määritä vaatimukset:

- Sidosryhmätarpeet
- Ylätason vaatimukset
- Johdetut vaatimukset
- Tuotemäärittelyt

Todenna suunnittelu:

Todenna tehty suunnittelu ja tuotemäärittelyt johdettuihin vaatimuksiin nähden.

Kelpuuta toteutus:

Suorita toteutuksen kelpuus hyödyntäen sidosryhmätarpeita ja ylätason vaatimuksia todellisessa käyttötilanteessa oikeiden käyttäjien tehdessä työtä tuotteen kanssa.

Lähdeluettelo (Osa III)

- Alanen, J., Vidberg, I., Nikula, H., Papakonstantinou, N., Pirttioja, T. & Sierla, S. 2011. Engineering data model for machine automation systems. Espoo, VTT. VTT Tiedotteita – Research Notes 2583, s. 1–137.
- Artisan Studio SysML. Artesan Studio SysML Tutorial v7.3. Artisan Studio™, Atego™ 21.1.2013. <http://www.atego.com/downloads/support/tutorials/SysMLTutorial7.3.exe>.
- Artisan Studio Uno. Artesan Studio Uno 7.4. HSUV-malli. 21.1.2013
- Doran, T. 2005. IEEE 1220: For Practical Systems Engineering. Computer May Vol. 39, No. 5, s. 92–94.
- Ecimovic, T., Mulej, M. & Mayur R. 2002. System Thinking and Climate Change System (Agains a Big “Tragedy of Commons” of all of us). SEM Institute of Climate Change, Korte, Slovenia. 286 s.
- INCOSE SE Handbook. 2007. Systems Engineering Handbook. A Guide for system life cycle processes and activities. Volume 3.2. INCOSE.
- ISO 10303-233. 2012. Industrial automation systems and integration — Product data representation and exchange. Part 233: Application protocol: Systems engineering.
- ISO/IEC 15288. 2008. Systems and Software Engineering – System Life Cycle Processes. 2008-02-01.
- ISO/IEC 26702. 2007. Systems engineering — Application and management of the systems engineering process. 2007-07-15, (IEEE Std 1220-2005)
- Jenkins, D.P., Stanton, N.A., Salmon, P.M. & Walker, G.H. 2008. Using Cognitive Work Analysis to Explore Activity Allocation within Military Domains, Human Factors Integration Defence Technology Center, HFIDTC/2/WP1.1.1.3/1 Version 1/ 5th February 2008.
- MIL-HDBK-881. 1998. MIL-HDBK-881 Department of Defense. Handbook. Work Breakdown Structure, 2.1.1998
- NASA 2007. Systems Engineering Handbook, NASA/SP-2007-6105. National Aeronautics and Space Administration. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20080008301_2008008500.pdf.

- OMG SysML. 2011. OMG Systems Modeling Language (OMG SysML™) Version 1.3. 10.8.2011. <http://www.omg.org/spec/SysML/1.3>.
- Peltoniemi, M., Isoaho, S., Hämäläinen, T., Nurmi, P. & Nummela, E. 2004. Kat-
saus systeemiteorioihin – Järjestelmäajattelu. Materiaalivirtaustutkimus-
ryhmä, Bio- ja ympäristötekniikan laitos, Tampereen yliopisto. 33 s.
[http://www.utu.fi/en/units/ffrc/research/project-archive/environment/Docu-
ments/etu_7.pdf](http://www.utu.fi/en/units/ffrc/research/project-archive/environment/Documents/etu_7.pdf).
- Sage, A.P. & Armstrong, J.E. Jr. 2000. Introduction to Systems Engineering.
Wiley, New York.
- Savioja, P. 2003. Käyttäjakeskeiset menetelmät monimutkaisten järjestelmien
vaatimusten kuvaamisessa. Espoo, VTT. VTT Tiedotteita – Research
Notes 2216. 132 s. + liitt. 10 s. [http://www.vtt.fi/inf/pdf/tiedotteet/2003/
T2216.pdf](http://www.vtt.fi/inf/pdf/tiedotteet/2003/T2216.pdf).
- SFS-EN 614-2 + A1. 2009. Koneturvallisuus. Ergonomiset suunnitteluperiaatteet.
Osa 2: Työtehtävien ja koneen suunnittelun väliset vuorovaikutukset [Safety
of machinery. Ergonomic design principles. Part 2: Interactions between
the design of machinery and work tasks]. 30.3.2009.
- SFS-EN ISO 6385. 2004. Työjärjestelmien ergonomiset suunnitteluperiaatteet
[Ergonomic principles in the design of work systems]. 17.5.2004.
- SFS-EN ISO 9241-20. 2009. Ihmisen ja järjestelmän vuorovaikutuksen ergonomia.
Osa 20: Tieto- ja viestintäteknologian laitteiden sekä palvelujen esteettä-
myyttä koskevat ohjeet [Ergonomics of human-system interaction. Part 20:
Accessibility guidelines for information/communication technology (ICT)
equipment and services]. 22.6.2009.
- SFS-EN ISO 9241-171. 2009. Ihmisen ja järjestelmän vuorovaikutuksen ergonomia.
Osa 171: ohjelmistojen esteettömyyttä koskevaa opastusta [Ergonomics
of human-system interaction. Part 171: Guidance on software accessibility],
16.2.2009.
- SFS-EN ISO 9241-210. 2010. Ihmisen ja järjestelmän vuorovaikutuksen ergonomia.
Osa 210: Vuorovaikutteisten järjestelmien käyttäjakeskeinen suunnittelu [Er-
gonomics of human-system interaction. Part 210: Human-centred design for
interactive systems]. 22.11.2010.
- Suh, N.P. 1998. Axiomatic Design Theory for Systems. Research in Engineering
Design, Vol. 10, No. 4, s. 189–209.

Umpleby S.A. 1994. The Cybernetics of Conceptual Systems. Paper prepared for the Institute for Advanced Studies, Vienna, Austria. http://www.gwu.edu/~umpleby/Conceptual_Systems.txt.

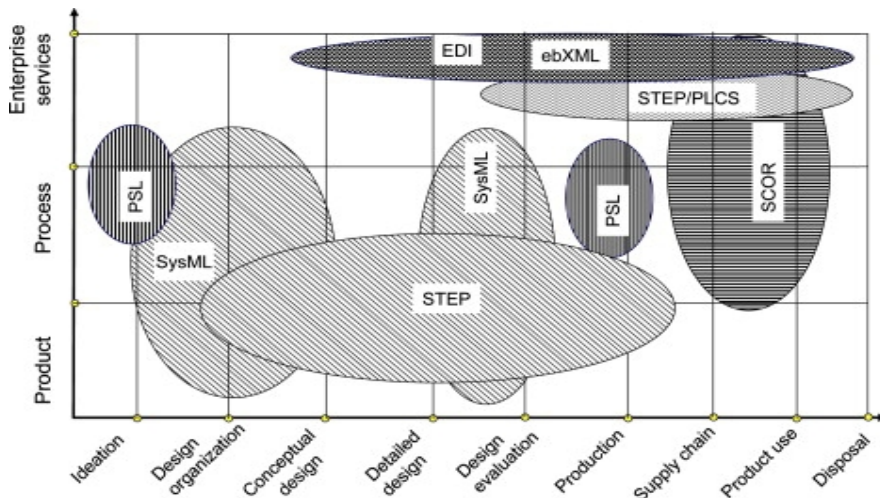
Vicente, K.J. 1999. Cognitive Work Analysis. Toward Safe, Productive, and Healthy Computer-Based Work. Mahwah, New Jersey: Lawrence Erlbaum.

von Bertalanffy, L. 1968. General System Theory. Foundations, Development, Applications. New York.

Liite A: Systems Engineering – määritelmiä ja standardeja

Tuotetietostandardeja

Standardointielimet (mm. ISO) ovat määritelleet tuotetiedon esittämiseen standardeja, joita hyödyntämällä sovellukset pystyvät kommunikoimaan ilman manuaalista välikäsitelyä. Standardien kattavuutta liiketoiminnassa on tutkinut Rachuri et al. [2008].



Kuva A1. Standardien kattavuus tuotteen elinkaaren aikana.

Tällä hetkellä ns. STEP-standardi (ISO 10303 [2012]) on käytetyin. STEP-standardi (the **ST**andard for the **E**xchange of **P**roduct model data) kattaa perinteisen suunnittelutiedon lisäksi tuotteen toiminnallisia ominaisuuksia sekä tuotteen elinkaaritietoja. STEP-standardi on laaja kokonaisuus, joka määrittelee perustan toimiala- tai sovellusaluekohtaisille ratkaisuille. STEP voidaankin ajatella meta-standardina, joka määrittelee toimialakohtaiset standardit. STEPin kattavuutta suhteessa tuotteen elinkaareen on esitetty kuvassa A1 (standardien kattavuus)

STEP koostuu:

- EXPRESS-kielestä, jolla voidaan kuvata oliopohjaisia tietomalleja
- valmiiksi määritellyistä yleishyödyllisistä tietomalleista (esim. geometria, visualisointi, tuoterakenne)
- näiden päälle rakennetuista alakohtaisista tuotemalleista (Application Protocols, AP).

Tiedonsiirtoon STEPissä on määritelty kolme ratkaisua: ASCII, XML ja ohjelmointirajapinta. Käytännössä STEPiä eniten on hyödynnetty CAD-järjestelmien välisessä tiedonsiirrosta geometrian ja tuoterakenteiden osalta.

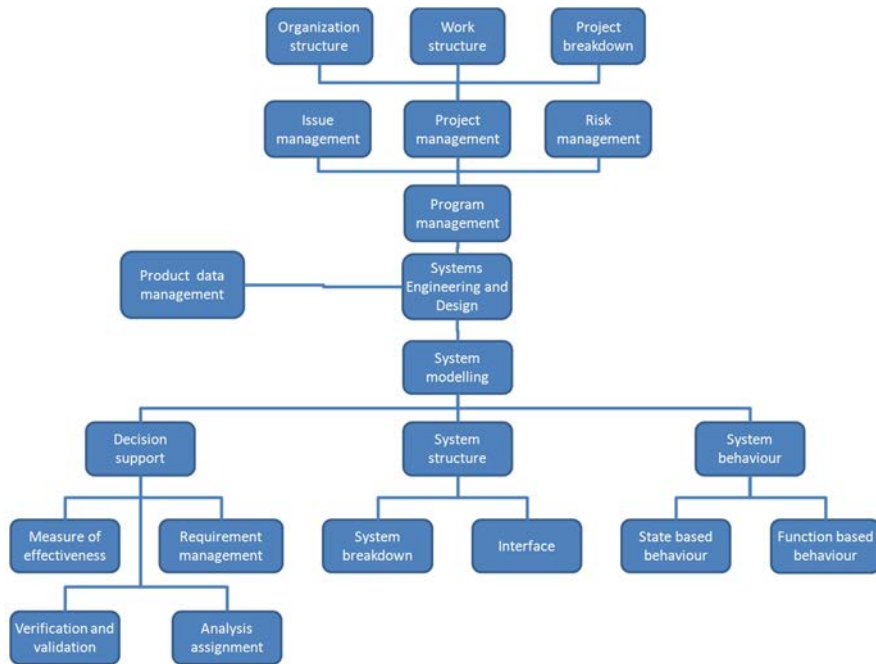
STEP-standardin kehittämisen lähtökohtana on ajatus siitä, että standardia voitaisiin soveltaa minkä tahansa tuotteen elinkaaren hallintaan. STEP on siis joukko toisiaan tukevia sovellusprotokollia, jotka yhdessä määrittävät lähes kaiken teollisten tuotteiden valmistamisessa välitettävän ja tallennettavan tiedon. Peruseriaatteena on siirtää tietoa standardoidussa muodossa koko arvoketjun läpi. Sovellusprotokollat tukevat toinen toisiaan ja muodostavat yhdessä aukottoman tiedonsiirtoketjun tuotteen elinkaaren ajan. STEP-standardin tarkoituksena on levittäytyä koko arvoketjuun. Tällöin tuote tieto olisi saman standardin mukaista ja helposti järjestelmästä toiseen siirrettävissä. STEP-standardia ovat voimakkaasti ajamassa eteenpäin mm. auto- ja ilmailuteollisuus, laivanrakennus ja puolustusvälineiteollisuus. Näillä teollisuuden aloilla tiedon siirtäminen lukuisten alihankkijoiden ja tavaramittajien välillä aiheuttaa usein paljon ylimääräistä työtä. Olisi kaikkien etu, jos eri osapuolet kykenisivät jonkun toimijan johdolla sopimaan mm. olemassa olevien standardien käytöstä, toimintatavoista, pelisäännöistä ja tietoteknisistä ratkaisuista.

Systems Engineering

STEP tuoteperheeseen kuuluu myös suoraan SE:tä tukeva tietomalli (ISO 10303-233 [2012]). Standardin ajavana voimana on ollut ohjelmistotyökalujen määrän kasvu, useiden organisaatioiden välinen yhteistyö ja tästä aiheutuvat tiedonhallintaongelmat. Tietomalli jakaantuu systeemin mallintamiseen, tuotetiedon ja koko tuoteperheen (program) hallintaan (kuva A2).

Kun huomioidaan koko tuotteen elinkaari ja varsinkin tuotteen omistajan näkökulmasta, niin STEP-perheeseen kuuluu myös tietomalli (Industrial automation systems and integration – Product data representation and exchange – Part 239: Application protocol: Product life cycle support [ISO 10303-239 2005]) tai lyhyesti AP-239. AP-239 tietomalli on lähtökohtaisesti tarkoitettu investointituotteille, joiden elinkaari on pitkä (laiva, lentokone, laitos jne.), mutta sitä voidaan soveltaa myös vähemmän vaativille tuotteille kuten kirjan osassa III, *Vaatimustenhallinta järjestelmähankinnassa* on osoitettu. AP-239 pyrkii ratkaisemaan ongelman, kuinka hallita tuotetietoa, jota tarvitaan, jotta tuotetta voidaan käyttää ja ylläpitää. Tuotteen elinkaaren aikana tuote saa uusia "päivityksiä" ja saattaa olla elinkaaren loppupuolella jo muuttunut melkoisesti alkuperäisestä versiosta.

System Engineering -käsitettä on myös pyritty täsmentämään erilaisin määritelmien. Seuraavassa on joitakin yleisimpiä määritelmiä:



Kuva A2. AP 233 -tietomallin rakenne.

NASA¹²:

“An interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and lifecycle balanced set of system people, product, and process solutions that satisfy customer needs. Systems engineering encompasses (a) the technical efforts related to the development, manufacturing, verification, deployment, operations, support, disposal of, and user training for, system products and processes; (b) the definition and management of the system configuration; (c) the translation of the system definition into work breakdown structures; and (d) development of information for management decision making.”

¹² <http://www.nasa.gov>

FINSE (Finnish Systems Engineering Association ry – The Finnish Chapter of INCOSE)¹³

”Systems Engineeringillä tarkoitetaan järjestelmällistä toimintatapaa, jolla pyritään varmistumaan siitä, että järjestelmä tai yleisesti ottaen palvelu, jota ollaan tuottamassa vastaa sille asetettuja vaatimuksia. Oleellista on ymmärtää, että suorituskkyisen järjestelmän kehittäminen edellyttää, että tiedetään mitä ollaan tekemässä, miksi ollaan tekemässä ja missä toimintaympäristössä järjestelmää tullaan käyttämään. Systems Engineering pitää ymmärtää prosessina, jossa on useita osa-alueita.”

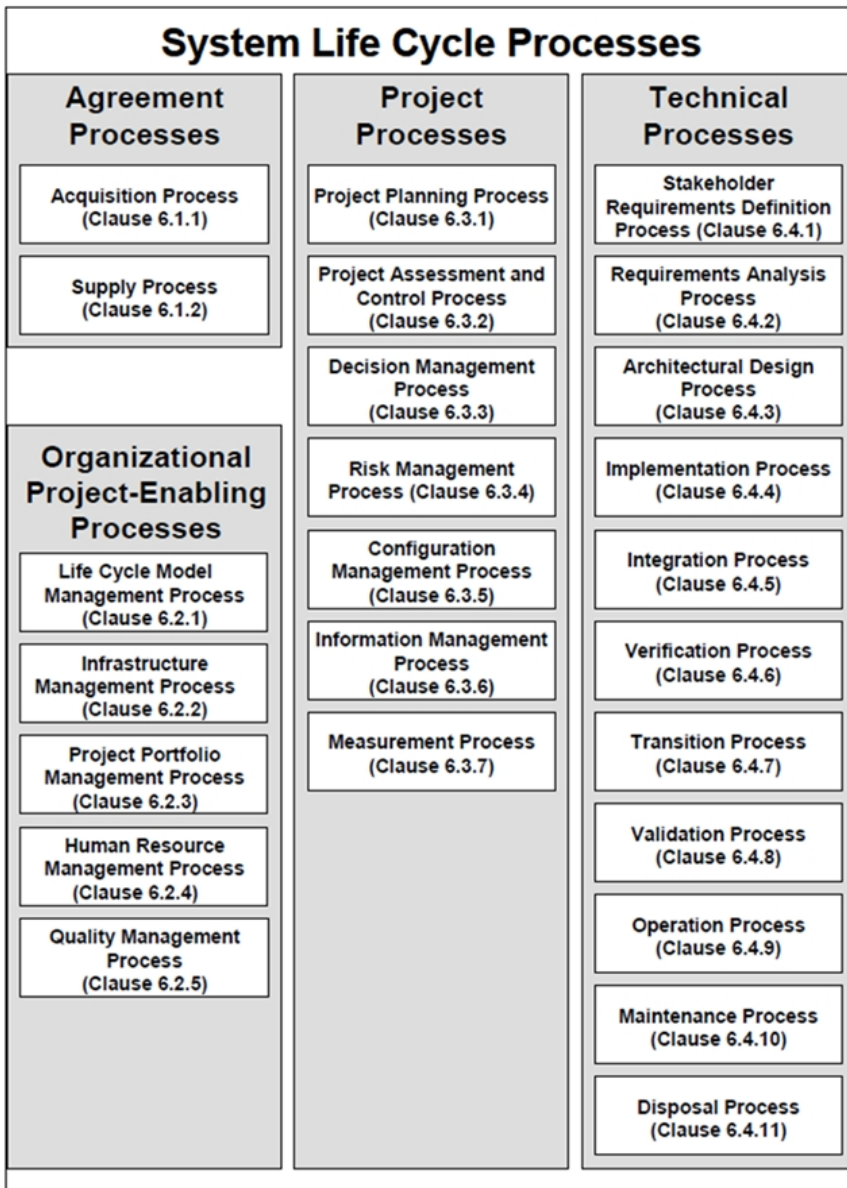
International Council of Systems Engineering (INCOSE):

”Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: Operations, performance, test, manufacturing, cost & schedule, training & support, and disposal. Systems engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.”

ISO/IEC 15288

ISO/IEC 15288 [2008] Systems and software engineering – System life cycle processes määrittelee yhtenäisen tavan kuvata järjestelmän elinkaari määrittelemällä joukon prosesseja ja niihin liittyvää terminologiaa. Standardi kuvaa myös prosesseja elinkaarivaiheiden määrittelyyn, hallintaan ja parantamiseen organisaation tai projektin puitteissa. Standardi määrittelee 25 eri prosessia, jotka on ryhmitelty teknisiin prosesseihin, projektiprosesseihin, organisaation prosesseihin ja sopimusprosesseihin. Jokaiselle prosessille on määritelty nimi ja tarkoitus, sen sisältämät aktiviteetit ja tehtävät sekä prosessin tuottama tulo.

¹³ <http://www.finse.org>



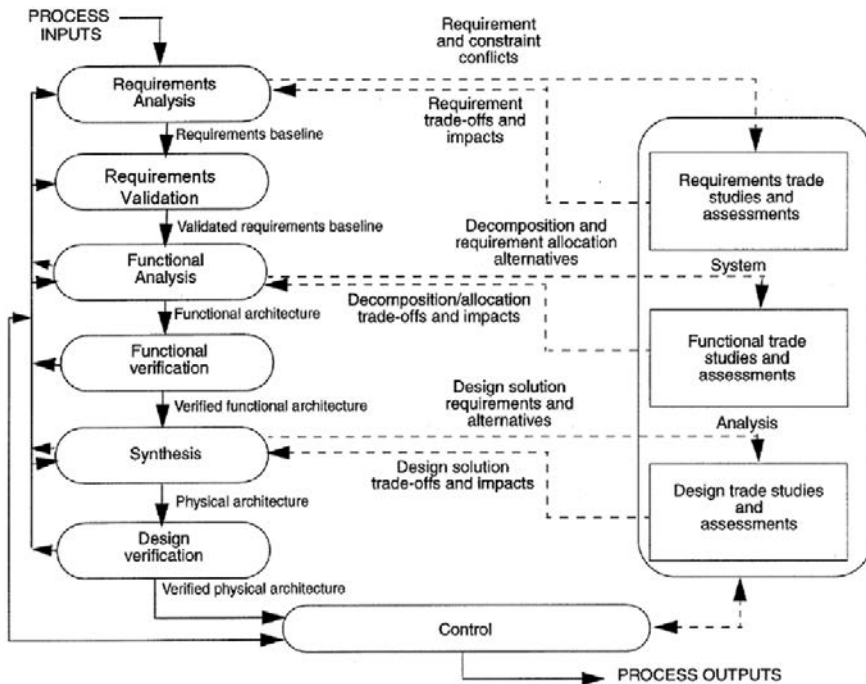
Kuva A3. Järjestelmän elinkaari prosessit. ISO/IEC 15288 [2008].

IEEE 1220

IEEE 1220 [2005] Standard for Application and Management of the Systems Engineering Process on ISO/IEC 15288 [2008] yksityiskohtaisempi ja käytännöllä-

heisempi ja sisältää myös käytännön ohjeistusta siitä mitä eri vaiheessa tulisi huomioida. Samalla se on myös soveltamisalaltaan rajoituneempi. IEEE 1220 [2005] jakaa elinkaari-prosessin kuvan A3 mukaisesti.

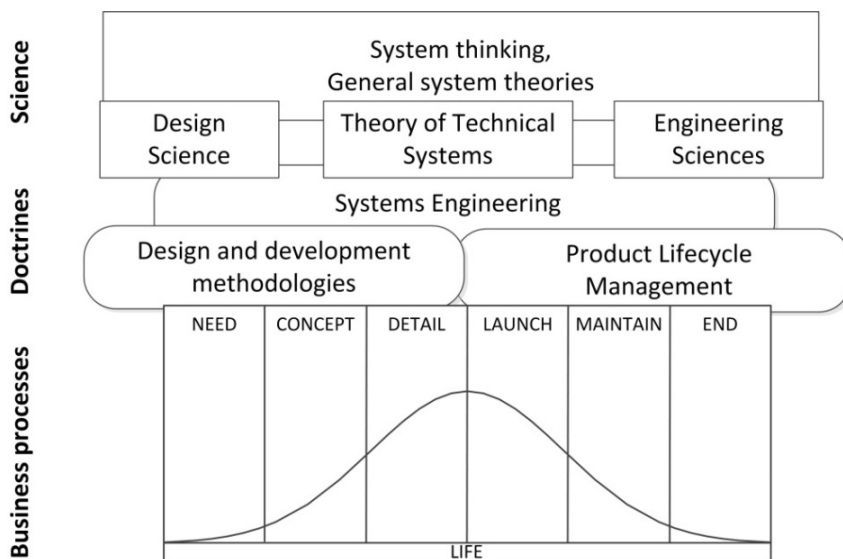
Standardien lisäksi on yrityksissä ja organisaatioissa tuotettu omia ohjeistuksia, joita on usein myös julkaistu internetissä vapaasti selattaviksi. Tällaisia ovat esimerkiksi NASA:n ja MITRE:n Systems Engineering -oppaat.



Kuva A4. Systems Engineering prosessin (SEP) päävaiheet, ISO 1220 [2005].

Yritykset toimivat verkottuneessa globaalissa ympäristössä, jossa valokaapeli siirtää dataa paikasta toiseen salamannopeasti. Internet mahdollistaa luotettavan datasiirron maailmassa, jossa etäisyys on sivuseikka. Haasteeksi onkin muodostunut tuotetiedon tulkinta eri sovellusten välillä.

Kuvassa B1 on havainnollistettu miten kirjassa on ajateltu tieteellisten teorioiden, käytännöllisten oppien ja metodiikkojen (esimerkiksi tuotekehitys ja projektinhallinta) sekä yritysten prosessien ja liiketoiminnan välisiä suhteita. SE:n ajatellaan olevan eräs käytännön sovellus yleisistä systeemi-teorioista teknisten järjestelmien kehittämiseen. Oppaassa on myös viitattu yhteen erityiseen teknisten järjestelmien teoriaan (Hubka & Eder 1984), jota on hyödynnetty käsitteellistämään SE:hen liittyviä asioita mallintamisen ja teknisten prosessien näkökulmasta. Kirjassa tuodaan esiin myös järjestelmien suunnittelun ja kehittämisen käytäntöjä eri tasoilla lähtien hyvin abstrakteista tarpeista yksityiskohtaiseen disipliini-työhön. Tämä liittyy erityisesti siihen miten sumeista tarpeista jalostetaan luovasti vaatimuksia ja asetetaan oikeita ongelmia suunnittelulle – ja miten ne ongelmat ratkaistaan kustannustehokkaasti. Miten määritellään ja mallinnetaan hyvä tuote tai järjestelmä (design science) – ja miten sellainen saadaan aikaan (engineering sciences). Tämän prosessin aikana tarvitaan erilaisia osaamisia ja organisaation rooleja. Erilaisia metodiikkoja ja näkökulmia tarvitaan järjestelmän määrittelyn ja elinkaaren eri vaiheissa. Tähän liittyy myös oppaan yksi keskeinen tavoite, eli pohtia SE:n ja PLM:n välistä suhdetta. Ensimmäinen on perinteisesti keskittynyt enemmän laajojen ja usein yksittäisten järjestelmien kehittämiseen, ja jälkimmäinen sarjatuotteiden hallintaan. Tällä hetkellä eletään vaihetta jossa nämä paradigmat menevät päällekkäin ja erityisesti PLM laajentaa kyvykkyyttään myös SE:n prosessien hallintaan.



Kuva B1. Kirjan teoreettinen viitekehys.

Lähdeluettelo

- Hubka, V. & Eder, V.E. 1984. Theory of Technical Systems – A Total Concept Theory for Engineering Design. Springer-Verlag. ISBN 3-540-17451-6.
- IEEE 1220. 2005. Standards for Application and Management of the Systems Engineering Process.
- ISO 10303. 2012. Automation systems and integration – Product data representation and exchange.
- ISO 10303-233. 2012 Industrial automation systems and integration – Product data representation and exchange – Part 233: Application protocol: Systems engineering.
- ISO 10303-239. 2005. Industrial automation systems and integration – Product data representation and exchange – Part 239: Application protocol: Product life cycle support.
- ISO/IEC 15288. 2008. Systems and Software Engineering – System Life Cycle Processes. 2008-02-01.
- Rachuri, S., Subrahmanian, E., Bouras, A., Fenves, S.J., Foufou, S. & Sriram, R.D. 2008. Information sharing and exchange in the context of product lifecycle management: Role of standards. Computer-Aided Design, Vol. 40, No. 7, s. 789–800.

Nimeke	Katsaus kompleksisten järjestelmien elinkaaren suunnitteluun
Tekijä(t)	Göran Granholm (toim.), Jarmo Alanen, Kari Karstila, Iris Karvonen, Simo-Pekka Leino, Juha Sääski, Mikko Uoti & Juhani Viitaniemi
Tiivistelmä	<p>Kirjassa käsitellään menetelmiä teknisten järjestelmien elinkaarivaatimukset huomioivaan suunnitteluun. Valittuja keinoja ja metodiikkoja ovat kompleksisten järjestelmien tekniikka (Systems Engineering – SE), tuotteen elinkaaren hallinta (Product Lifecycle Management – PLM), mallipohjaisuus ja virtuaaliprototyypointi, vaatimusten hallinta sekä näihin liittyvät standardit.</p> <p>Tietoteknisten järjestelmien rooli tuotekehityksen ja elinkaarenhallinnan mahdollistajana on kiistanaton. Asteittainen siirtyminen dokumenttipohjaisesta tiedonhallinnasta mallipohjaisiin esityksiin kaikissa elinkaaren vaiheissa mahdollistaa entistä tehokkaamman yhteistyön eri suunnittelualojen ja sidosryhmien välillä. Uusien menetelmien, prosessien ja niitä tukevien työkalujen käyttöönotto ja soveltaminen sekä yrityksen sisällä että keskeisissä verkostoissa vaatii kuitenkin huolellista suunnittelua ja tapauskohtaisten erityispiirteiden huomioimista.</p> <p>Keskeisenä lähtökohtana on suunnittelukohteiden käsitteleminen osana kokonaisuutta, johon vaikuttaa niin sidosryhmät, käyttöympäristö kuin kytkennät muiden järjestelmien muodostamaan ”ekosysteemiin” ja näissä tapahtuvat muutokset koko elinkaaren ajan.</p>
ISBN, ISSN	ISBN 978-951-38-8045-3 (URL: http://www.vtt.fi/publications/index.jsp) ISSN-L 2242-1211 ISSN 2242-122X (verkkojulkaisu)
Julkaisu-aika	Syyskuu 2013
Kieli	Suomi, englanninkielinen tiivistelmä
Sivumäärä	220 s. + liitt. 9 s.
Projektin nimi	SE-PLM integ
Toimeksiantajat	Tekes
Avainsanat	Systems Engineering, Life-cycle management, product design, product data, design process, requirements management, PLM, PDM
Julkaisija	VTT PL 1000, 02044 VTT, Puh. 020 722 111

Title	A look into the life-cycle design of complex systems
Author(s)	Göran Granholm (Editor), Jarmo Alanen, Kari Karstila, Iris Karvonen, Simo-Pekka Leino, Juha Sääski, Mikko Uoti, Juhani Viitaniemi
Abstract	<p>Methods and approaches to help integrating tools, processes and systems thinking to achieve successful and sustainable products, considering all aspects of the product lifecycle are presented. Chosen methodologies include Systems Engineering (SE), Product Life Cycle Management (PLM), model-based approaches, virtual prototyping, requirements management and related standards.</p> <p>One important feature of the model-based process is the possibility to replicate as much as possible of the products different life-cycle stages as virtual, digital model representations. Virtual modelling efficiently integrated into the product life-cycle enables effective user involvement throughout the process, including early design.</p> <p>Adopting new tools and processes across an organisation network involves challenges that have to be addressed case by case.</p> <p>This guide book gives as a general introduction to the subject and treats selected topics on a comprehensive but easily approachable level, and gives hints to further reading. The guide will also include practical case studies to illustrate how Systems Engineering has been implemented in real life and what experience has been gained.</p>
ISBN, ISSN	ISBN 978-951-38-8045-3 (URL: http://www.vtt.fi/publications/index.jsp) ISSN-L 2242-1211 ISSN 2242-122X (Online)
Date	September 2013
Language	Finnish
Pages	220 p. + app. 9 p.
Name of the project	Systems Engineering – PLM integraatio
Commissioned by	TeKes
Keywords	Systems Engineering, Life-cycle management, product design, product data, design process, requirements management, PLM, PDM
Publisher	VTT Technical Research Centre of Finland P.O. Box 1000, FI-02044 VTT, Finland, Tel. 020 722 111

Katsaus kompleksisten järjestelmien elinkaaren suunnitteluun

ISBN 978-951-38-8045-3 (URL: <http://www.vtt.fi/publications/index.jsp>)
ISSN-L 2242-1211
ISSN 2242-122X (verkkojulkaisu)

