JavaScript Työ

Tommi Jokinen

Miikael Lehtimäki

Antti Leppänen

Syyskuu 2016

Ohjelmistotekniikka

SISÄLLYS

[1 TOGAF 3](#_Toc462064021)

[1.1 Dokumentaatio 3](#_Toc462064022)

[1.2 Esittely 3](#_Toc462064023)

[1.3 Arkkitehtuurin kehitysmetodit 3](#_Toc462064024)

[1.4 Ohjenuorat ja tekniikat 4](#_Toc462064025)

[1.5 Arkkitehtuurin sisältörakenne 4](#_Toc462064026)

[1.6 Yhtiöjatkumo ja välineet 5](#_Toc462064027)

[1.7 TOGAF -referenssimalli 5](#_Toc462064028)

[1.8 Arkkitehtuurin kykyverkko 6](#_Toc462064029)

[2 DoDAF 7](#_Toc462064030)

[2.1 Yleiskäsitys 7](#_Toc462064031)

[2.2 Arkkitehtuurimalli 8](#_Toc462064032)

[3 Refaktorointi 10](#_Toc462064033)

[4 MIL-STD-882D -standardi 11](#_Toc462064034)

[4.1 Yleistä 11](#_Toc462064035)

[4.2 Dokumentaatio ja käyttö 11](#_Toc462064036)

[LÄHTEET 13](#_Toc462064037)

# TOGAF

## Dokumentaatio

TOGAF on avoimen ryhmän standardi yhtiöarkkitehtuureihin, The Open Group Architecture Forum, johon kuuluu yli 200 jäsentä ylläpitää ja kehittää standardia. Sen dokumentti on jaoteltu seitsemään osaan: esittely, arkkitehtuurin kehitysmetodit, ohjenuorat ja tekniikat, arkkitehtuurin sisältöverkosto, yhtiöjatkumo ja välineet, TOGAF -referenssimalli ja arkkitehtuurin kykyverkosto.

## Esittely

Käydään läpi edeltävä tieto, arkkitehtuurin hyödyt ja syyt miksi se pitäisi adoptoida. Tämä osio on lyhyt osio, jossa on tiivistettynä TOGAF:n periaatteet.

## Arkkitehtuurin kehitysmetodit

TOGAF on yhteensopiva neljän arkkitehtuuriryhmän kanssa, liiketoiminta arkkitehtuuri, data arkkitehtuuri, applikaatio arkkitehtuuri ja teknologia arkkitehtuuri ryhmien kanssa. Arkkitehtuurin kehitysmallissa on kahdeksan kohtaa joissa käydään läpi edellä käydyt neljä ryhmää omaan arkkitehtuurivisioon sopivina, pohditaan tilaisuuksia ja ratkaisuja ongelmiin, siirtymän suunnittelu, implementaation seuranta ja arkkitehtuurin muutosten johtaminen. Tämä prosessi tuottaa lopuksi ulosmenevää tietoa jotka voi kasata kolmeen ryhmään.

Vietävät, jotka ovat sovittuja ja allekirjoitettuja tuotteita jotka on sovittu etukäteen ja projektit jotka valmistuessa menevät suoraan arkistoihin. Artefaktit, jotka ovat arkkitehtuurisia töitä jotka kuvaavat jotain arkkitehtuurin osaa. Rakennuspalikat, jotka kuvaavat uudelleenkäytettäviä osia yrityksestä, arkkitehtuurista, tietoverkosta tai muusta osasta jonka voi yhdistää toisten rakennuspalikoiden kanssa tuottaakseen arkkitehtuureja ja ratkaisuja.

## Ohjenuorat ja tekniikat

Ohjenuorat ja tekniikat -kappaleessa käydään tarkemmin läpi iteraatiomallin kehityksestä, välianalyysistä, muutoksen suunnittelutekniikoista ja riskianalyysistä. Ohjeistuksesta löytyy myös kuinka TOGAF:ia voidaan käyttää muiden kehitysmallien kanssa.

## Arkkitehtuurin sisältörakenne

Kappaleessa tutustutaan arkkitehtuurin kehitysprosessin ulostulevan datan tarkempaan analyysiin, mallintamiseen ja vaihtoehtoisenmallin kuvaamiseen sen ulkopuolelle. TOGAF:n metamallia käytettäessä käytetään seuraavia ydintermejä pohjana:

Näyttelijä: henkilö, organisaatio tai järjestelmä joka on arkkitehtuurin ulkopuolella, mutta käyttää sitä.

Sovellus komponentti: sovelluksen toiminnat tiivistettynä, jotka on kohdistettu arkkitehtuuriin.

Liiketoiminta palvelu: tukee yritysten toimintaa määritellyn rajapinnan kautta, joka on jonkin organisaation hallitsema.

Tiedon kokonaisuus: tiivistelmä tiedoista, jotka ovat liiketoiminnan ammattilaisen hyväksymiä. Nämä tiedot voivat olla liitoksissa sovelluksiin, arkistoihin ja palveluihin.

Tehtävä: toimittaa liiketoiminta edellytykset sen läheisellä organisaatiolla, mikä ei tarvitse olla kyseisen organisaation hallitsemaa.

Tietojärjestelmä palvelu: yrityksen palvelun automatisoidut elementit. Tietojärjestelmä palvelu voi samanaikaisesti hoitaa vain osaa tai kaikkea yhden yrityksen palveluista.

Järjestöyksikkö: itsenäinen resursseista koostuva yksikkö joilla on päämäärät, tavoitteet ja toimenpiteet. Nämä yksiköt voivat sisältää kolmansia osapuolia ja liikekumppani organisaatioita.

Pohjapalvelu: teknillinen edellytettävä valmius, joka mahdollistaa tuen uusille intrastruktuurin tarjoamille sovelluksille.

Rooli: näyttelijä, joka ottaa roolin tehtävän suorittamiseksi.

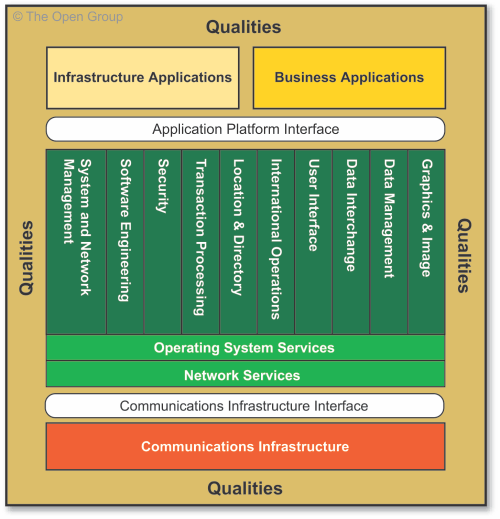
Teknologia komponentti: infrastruktuurin teknologian tiivistelmä, joka edustaa teknologia luokan tuotetta tai erityistä teknologian tuotetta.

## Yhtiöjatkumo ja välineet

Tässä käydään tarkemmin, miten arkkitehtuurien kehitys loogisesta fyysiseen ja abstraktista konkreettiseen vaikuttaa yhtiöön ja miten yhtiön eri arkkitehtuurit voivat toimia yhdessä ja ristiriitojen minimoimisen.

## TOGAF -referenssimalli

Referenssimallissa käydään läpi eri esitystapoja sekä tekninen referenssimalli, joka on yleistettävä malli kaikkien tilanteiden soveltamiseen arkkitehtuurin sisällä. Kuvassa 1 nähdään yksityiskohtaisimmat asiat referenssimallista ohjelman tuottamisessa.



KUVA 1. Yksityiskohtainen tekninen referenssimalli

## Arkkitehtuurin kykyverkko

Kirjoitus kertoo arkkitehtuurin käyttöä vaativista organisaation rakennelma-, rooli-, prosessi-, vastuu- ja kykyvaatimuksista. Tärkeä osa arkkitehtuurin johtamisstrategiasta on ryhmän valitseman ylitsekatsomaan sitä, nämä ryhmät luodaan tarkoin vastuualuein ja päätösvalta rajojen kanssa.

# DoDAF

## Yleiskäsitys

DoDAF eli Department of Defense Arhcitecture Framwork on arkkitehtuuri malli Yhdysvallan puolustuslaitokselle, jolla saadaan visuaalinen kuva koko infrastruktuurista. Sen tarkoitus on määritellä käsitteet ja mallit puolustuslaitoksen kuudesta ydinprosessista:

JCIDS – Joint Capabilities Integration and Development

PPBE – Planning, Programming, Budgeting, Execution

DAS - Acquisition System

SE – Systems Engineering

Operations Planning

CPE - Capabilities Portfolio Management

Uusin DoDAF 2.0 koostuu seuraavista näkökulmista, joiden pohjalta tarkastellaan eriasioita:

All Viewpoint (AV) - Kuvaa yleisiä näkökulmia arkkitehtuurista. AV:t toimittavat oleellista tietoa koko arkkitehtuurin kuvaukseen käyttämällä hyväksi kaikki arkkitehtuurin eri näkökulmia.

Capability Viewpoint (CV) – Ilmaisee valmiuksien vaatimukset, toimituksien ajankohdat ja lähetettyjen valmiudet.

Data and Information Viewpoint (DIV) – Kuvailee operaatioita ja liiketoiminnan tietojen vaatimuksia ja sääntöjä joita hallinnoidaan sekä käytetään rajoittamaan organisaation liiketoimintaa.

Operational Viewpoint (OV) – Ilmaisee tehtävät ja toiminnat, operaatioiden elementtejä ja resurssien kulkua operaatioiden suorittamiseksi. Voidaan käyttää kuvaamaan vaatimuksena ”tuleville” arkkitehtuureille loogisin termein tai ”kuten-on” arkkitehtuureille.

Project Viewpoint (PV) – Ilmaisee kuinka ohjelmat, projektit, portfoliot tai toimituskykyjen aloitteet, organisaation avustukset niille tai niiden keskeisiä riippuvaisuuksia.

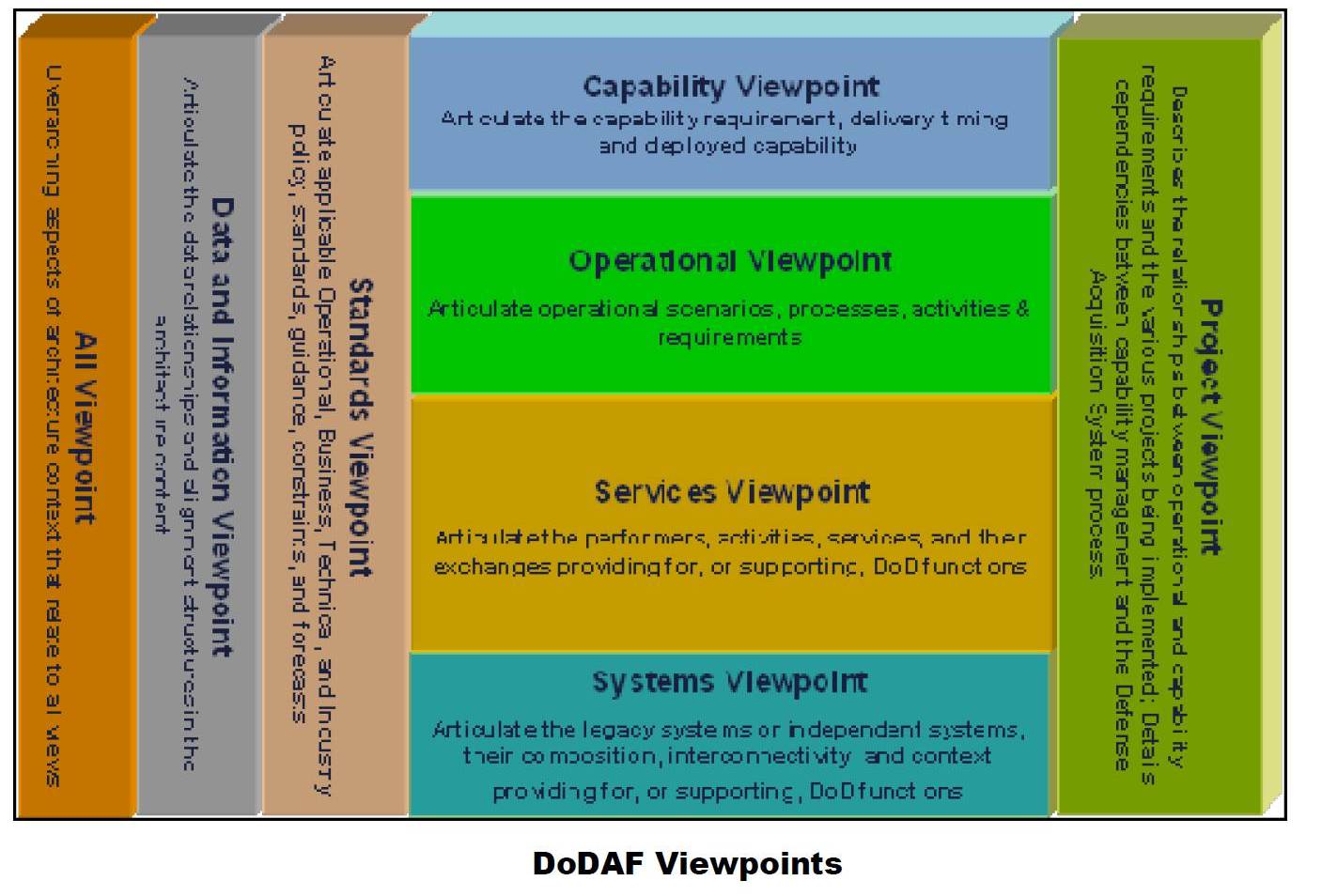
Services Viewpoint (SvcV) – Ilmaisee palveluita ja niiden liitoksia tarjota tai avustaa puolustuslaitokselle toimintoja, kuten sotaa tai liiketoiminnallisia asioita.

Standards Viewpoint (StdV) – Ilmaisee sovellettavia menettelytapoja, standardeja, ohjeita, rajoitteita, ennusteita joita JCIDS, DAS, järjestelmän tekniikka, PPBE, toiminnat sekä muiden prosessien haltijat ja päätöksentekijät.

Systems Viewpoint (SV) – Ilmaisee suunnittelun ratkaisujen ilmaisemiseen järjestelmille ja niiden koostumukseen, yhteenliitettävyyteen ja kontekstin tukemaan operaatioita tai valmius toimintoja. Järjestelmällä voidaan tarkoittaa muutakin kuin tietokone komponentteja tai -ohjelmia.

## Arkkitehtuurimalli

Jokaisella yksittäisellä arkkitehtuurin elementillä on ainutlaatuinen tunniste mitä käytetään kaikkialla tuotteissa ja näkymissä arkkitehtuurin sisällä, tarkoittaen että niihin viitataan suoraan.

Kuva 2. DoDAF 2.0 arkkitehtuurin mallit.

Tietyillä malleilla on tiettyjä tarkoituksia jotka prosessien omistajat ovat määritelleet. Kaikkia malleja arkkitehtuurista ei tarvitse luoda käyttääkseen sitä. Kun toiminta malli luodaan, avain prosessien omistajat päättävät mitä arkkitehtuurin tietoja vaaditaan DoDAF -malleissa.

DoDAF -arkkitehtuurimallia käytetään siis Yhdysvaltojen puolustuslaitoksessa uusien erilaisten järjestelmien liittämistä tai kehittämistä heidän piiriinsä, olivat ne sitten tietokoneista valmistettuja asioita tai esineitä taikka pienorganisaatioita. Sen eri näkymillä voidaan kategorisoida järjestelmien eri osa-alueita, jotta niiden toimintaa voidaan paremmin tarkastella puolustusvoimien näkökulmaa ajatellen, esim. mitenkä järjestelmää voidaan hyödyntää tietyissä tilanteissa taikka minkälaiset vaatimukset sen käyttöön tarvitaan.

# Refaktorointi

Tiivistettynä refaktorointi on koodin organisoimista järkevämmäksi, helppo lukuisemmaksi tulevaisuudessa tarvittavien muutosten tekoja varten. Kun toinen henkilö joutuu esimerkiksi lisäämään ominaisuuksia eri henkilön tuottamaan koodiin, niin ominaisuuksien lisääjällä on helpompi ymmärtää alkuperäistä koodia, jos sitä on refaktoroitu.

(1)

Refaktorointia tyypillisesti tehdään pienissä osissa, kerran päivässä hieman ja seuraavana toinen pieni pätkä. Se voi olla muuttujien nimeämisestä selkeämmiksi metodien esittelyyn kolmannen osapuolen luokkia varten.

Esimerkkinä kaavassa 1 havaitaan kuinka funktion oikealla puolella oleva yhtälö on lyhyesti ja tehokkaasti esitelty, mutta ensisilmäyksellä ei heti voi saada selville sen alkuperää. Vasemmanpuolinen osa on yksinkertaisempi ymmärtää, koska se käyttää yksinkertaisempia operaattoreita sekä kertoo syvemmin funktion toiminnasta. Tätä voidaan kuvata refaktoroinniksi.

Refaktorointi ei kuitenkaan päde vain koodin kirjoittamisessa, mutta sitä voidaan soveltaa myös arkkitehtuureissa ja suunnitteluissa. Kun arkkitehtuurista saadaan parempi käsitys ja on helpompaa selvittää sen osia sekä niiden toimivuuksia. Liiallinen refaktorointi voi kuitenkin koitua haitalliseksi ja varsinkin arkkitehtuurien refaktorointi on kallista sekä sisältää myös suuria riskejä.

# MIL-STD-882D -standardi

## Yleistä

MIL-STD-882D standardi on Yhdysvallan puolustusviraston standardi, joka määrittelee järjestelmille turvallisuus vaatimuksia koko laitteen elinajalle. Oikein käytettynä nämä vaatimuksen poistavat tai alentavat vaaratekijät hyväksytyille tasoille.

Laitteille on oltava valmistajan ja ohjelmiston johtajan hyväksymät lähestymistavat ja tämän dokumentaation pitää sisältää seuraavat ominaisuudet:

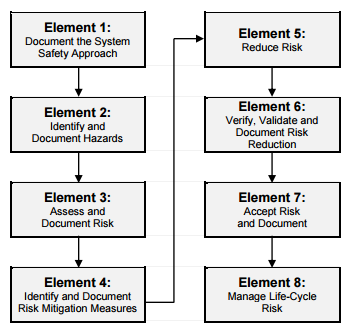
- Kuvata ohjelman käyttö ja tunnisteet kaikille vaara- ja vahinkotekijöille.

- Tiedot itse turvajärjestelmästä ja miten se on osa ohjelman kokonaisuutta.

- Määrittää miten vaara ja kommellus riskit kommunikoidaan sopiville virkahenkilöille ja miten niitä seurataan.

## Dokumentaatio ja käyttö

Laitteiden turvallisuutta taataan mahdollisimman pitkälle ensin laite ja ohjelmavalinnalla, sitten lisälaitteiden tai hälytyslaitteiden avulla ja viimeseksi koulutuksella ja proseduureilla, tässä tärkeänä osana on järjestelmän turvallisuuslähestymisdokumentointi. Tämä tapahtuu useassa vaiheessa, ensin riskin hoitamismenetelmä kuvaillaan ja miten ohjelma integroi sen, merkitään mahdolliset etukäteen määritellyt ja jälkikäteen määritellyt vaatimukset järjestelmälle, määritelmät sille miten vaarat ja niihin liittyvät riskin ovat virallisesti hyväksytty sopivien virkahenkilöiden toimesta ja jotka tapahtuivat käyttäjäedustajan ollessa paikalle. Vaarat dokumentoidaan suljetulla kierteellä toimivalla järjestelmällä joka sisältää vähintään virheen, siihen liittyvät kommellukset, riskien arvioinnin, riskien vähentämiseen käytettävissä olevat toimenpiteet, käytetyt toimenpiteet, vaaran tilanteen ja vahvistuksen riskien vähentämistoimenpiteille. Kuva liittyy aiheeseen.



KUVA 3. Järjestelmän turvaamisen kahdeksan kohtaa.

Kun järjestelmä on otettu käyttöön, järjestelmän hoitokeskus tulee käyttää sovittuja proseduureja vaarojen tunnistamiseen ja laitteen ylläpitoon sen eliniän ajan, tämä elinikäiseen vaiheeseen kuuluu käyttöjärjestelmät, käyttäjät, konerauta ja ohjelmistot, profiilit ja järjestelmän terveyttä mittaava tieto. Jos uusi vaara havaitaan tai jokin ennestään tunnistettu vaara oli kategorisoitu väärin, uusi tai uudelleen kategorisoitu riski pitää virallisesti hyväksyä järjestelmään puolustusviraston standardin mukaan. Lisäksi puolustusviraston vaatii tutkimuksia kaikkiin tällaisiin tapauksiin, joissa vaarat ja riskit määritellään ja riskiä vähentävien tapojen ja proseduurien pohtimista, keskittyen eritysiesti niihin jotka vähentävät ihmispohjaisia virheitä joita voi välttää.

Ohjelmistojen riskiselvitys ja vastaavasti ohjelmistojen ohjaamien laitteiden riskejä ei voi kategorisoida pelkästään vaaran ja todennäköisyyden perusteella. Ohjelman yksittäisen funktion virhettä on hankala ennustaa ja ei voi perustaa historialliseen tietoon. Ohjelmistojen yksittäiskohtainen käyttö hankaloittaa tätä ennestään ja siksi on aina tärkeä olla tietoinen ohjelmiston vaikutusvallasta laitteiston rautaan. Tämän takia ohjelmilla on tarkat rajat sille, kuinka paljon vaikutusvaltaa niille annetaan tilanteesta riippuen ja tätä määritellään dokumentoinnissa useilla kuvilla.

LÄHTEET

TOGAF Open Group Standard architecture document, version 9.1, luettu 12.9.2016.

<http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html>

Department of Defense Architecture Framework, version 2.02, luettu 12.9.2016

<http://dodcio.defense.gov/Library/DoD-Architecture-Framework>

What is Refactoring, luettu 19.9.2016

<http://c2.com/cgi/wiki?WhatIsRefactoring>

Department of Defense Standard Practice document, luettu 5.9.2016

<http://www.system-safety.org/Documents/MIL-STD-882E.pdf>