

Patric Kangasmäki

**TIETOJOUKKOJEN VISUAALISEN ESI-
TYSTAVAN SUUNNITTELU OHJELMIS-
TOKEHITYKSESSÄ**
VISDOM Roadmapper

TIIVISTELMÄ

Patric Kangasmäki: Tietojoukkojen visuaalisen esitystavan suunnittelu ohjelmistokehityksessä.
Pro gradu -tutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Huhtikuu 2023

Erilaisen tiedon visualisoinneilla on juuret syvällä historiassa. Ensimmäiset löydetty visuaaliset esitystavat ovat lähtöisin jo varhaisesta Egyptistä. Nykyään visualisoinnit ovat yleistyneet paljon, ja erityisesti ohjelmistokehityksessä ne ovat keskeisessä roolissa. Niiden avulla voidaan edistää sidosryhmien välistä kommunikaatiota, seurata projektin etenemistä, sekä kehittää toimintamalleja tiimien sisällä.

Tietoa on erityyppistä. Tutkielmaan on valittu esiteltäväksi ohjelmistoprojektissa käytettävien menetelmien lisäksi numeerisen tiedon tekniikoita, kuten esimerkiksi lämpökartat ja erilaiset graafit. Graafeja tulkittaessa on erityisen tärkeää ottaa huomioon, että huonosti suunniteltu graafi saattaa helposti johtaa sen tulkitsijaa harhaan. Tietyissä tilanteissa harhaanjohtavia kaavioita saatetaan jopa suunnitella tarkoituksella.

Ohjelmistoprojekteissa pelkät numeerisen tiedon esitystavat eivät usein riitä. Ohjelmistoprojekteissa tärkeää on pystyä seuraamaan projektin. Tutkielmassa esitellään eri visualisoinnin tekniikoita projektin eri vaiheisiin. Esimerkkeinä näistä ovat Kanban-taulut, etenemiskaaviot, sekä tietojen välisten suhteiden visuaaliset esitystavat.

Tässä tutkimuksessa perehdytään eri visualisoinnin metodeihin, sekä kehitetään Vincit Oyj:llä VISDOM projektissa visualisoinnin tekniikoita. Tarkoituksena on myös kehittää tietojoukkojen sovellukseen tehtävien välisille suhteille visuaalinen esitystapa. Tutkimuksessa on käytetty materiaalina aiheeseen liittyvää kirjallisuutta ja tutkimuksia.

Avainsanat: Visualisointi, visuaaliset notaatiot, ohjelmistoprojekti, Kanban, etenemiskaavio, kommunikaatio, tietojen väliset suhteet, kaavio, graafi

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

Sisällys

1. Johdanto	1
2. Datan visualisointi	2
2.1 Visualisoinnin historiaa	4
2.2 Visualisoinnin tavoitteet	5
2.3 Visualisoinnin prosessi	6
2.4 Visualisoinnin konkreettiset hyödyt	7
2.5 Visualisoinnin potentiaaliset ongelmat	8
2.5.1 Käsitteen muuttuminen odotetun tuloksen perusteella	9
2.5.2 Semioottinen selkeys	9
3. Visualisoinnin käytänteet.....	11
3.1 Visuaaliset merkinnät	11
3.1.1 Visuaalisten merkintöjen kommunikaatio	12
3.1.2 Ohjelmiston mallintaminen UML:lla	14
3.2 Värien merkitys visualisoinneissa	15
3.2.1 Värien symboliikka historiassa	17
3.2.2 Värien merkitys nykyajassa	17
3.2.3 Värien erottelu	18
3.2.4 Kirkkaus ja saturaatio ja kontrasti	19
3.3 Hyvän suunnittelun käytänteet	20
4. Numeerisen tiedon visualisointi.....	23
4.1 Lämpökartta	23
4.2 Viivadiagrammi	24
4.3 Pylväsdiagrammi	25
4.4 Ympyrädiagrammi	27
5. Ohjelmistoprojektin visualisointi.....	28
5.1 Kanban-taulu	28
5.2 Tehtävien väliset suhteet	30
5.3 Projektin etenemisen seuranta ja sen visualisoiminen	32
5.4 Ohjelmistoprojektin suunnitelma ja aikataulus.	34
5.4.1 Kriittisen polun menetelmä	34
5.4.2 Gantt-kaavio (Gantt-Chart)	35
6. Case VISDOM Roadmapper	37

6.1	VISDOM Roadmapper projekti	37
6.2	Tehtävien haasteen ja arvon esittäminen lämpökartalla	37
6.3	Suunnitelmien vertaaminen optimaaliseen arvoon	38
6.4	Tehtävien visuaalinen esitystapa	39
6.4.1	Vaatimukset tehtäväkartalle	39
6.4.2	Tietojen väliset suhteet	40
6.4.3	Tehtäväkartan suunnitelma	40
6.4.4	Käyttäjätestaukset	41
6.4.5	Raahaa ja pudota	43
6.4.6	Värien käyttäminen tehtäväkartassa	44
6.4.7	Tehtäväkartan visuaalinen selkeys ja suorituskyky suurilla datajoukoilla	45
6.4.8	Kehityskohteet	47
7.	Yhteenveto	49
8.	Johtopäätökset	50
9.	Viiteluettelo	51

1. Johdanto

Jo vuosituhsia erilaista informaatiota on visualisoitu sanoista kuviksi. Ensimmäiset löydettyt visualisoinnit ovat ajoittuneet ajanjaksolle 200 eaa. Informaation ja datan visualisointi on erityisen tärkeää ohjelmistokehityksessä, jossa dataa saattaa olla jopa tuhansia rivejä. Tiedon visualisoinnilla tarkoitetaan sanallisten esitysten sijaan tietojen kuvitettua esitystapaa.

Erityisen haasteellista tiedon tulkitsemisesta tulee silloin, kun kyseisillä tietojoukoilla on riippuvuuksia toisten tietojoukkojen välillä. Ohjelmistokehityksessä tiedon visualisointi voidaan jakaa tyypilliseen numeerisen tiedon visualisointiin, joka pitää sisällään esimerkiksi diagrammit ja kaaviot, sekä eri ohjelmistoprojektin vaiheiden visualisointiin. Ohjelmistojen suunnittelussa myös kaaviovisualisointi on ollut käytössä jo pitkään. Perinteisenä tapana toiminnallisuuksien esittämiseen on käytetty vuokaavioita. Yleisenä tiedon visualisoinnin mallina on käytetty ER-mallia (Chen 1976).

Tutkimuksen tavoitteena on kehittää Vincit Oyj:llä VISDOM Roadmapper sovelluksessa visualisoinnin tekniikoita, sekä luoda projektiin sopiva visuaalinen esitystapa tietojoukkojen välisille suhteille. Lisäksi työ pyrkii antamaan kirjallisuuskatsauksen tiedon visualisoinnin eri tekniikoihin ohjelmistoprojektissa.

Luvussa 2 käsitellään tiedon visualisointia yleisellä tasolla. Siinä käydään läpi visualisoinnin historiaa, nykyaikaa, tavoitteita sekä hyötyjä ja mahdollisia haittoja, joita visualisoinneilla voidaan saada aikaiseksi. Luku 3 pitää sisällään visuaalisten notaatioiden esittelyn. Luvussa käydään läpi visuaalisten notaatioiden suunnittelua, sekä niiden välistä kommunikaatiota. Luku sisältää myös visualisoinneissa värien merkityksien esittelyä, sekä yleisiä hyviä käytänteitä suunnitteluun. Lukuun 4 valittiin esiteltäväksi neljä yleistä numeerisen tiedon visuaalista esitystapaa. Valintaan vaikutti myös VISDOM Roadmapperissa jo käytetyt tekniikat. Luku 5 käsittelee ohjelmistoprojektin eri vaiheiden visualisointeja. Luku sisältää visualisointeja alkaen projektin suunnittelusta läpi projektin toteuttamisvaiheen. Myös ohjelmoinnin apuun luotua tekniikoita käydään läpi. Luvussa 6 käymme läpi VISDOM Roadmapper projektissa käytettyjä visualisoinnin tekniikoita, joilla pyritään luomaan käyttäjälle mahdollisimman selkeä ja nopeasti ymmärrettävä kokonaisuus isoista ja monimutkaisista tietorakenteista. Tämän lisäksi esittelemme kehitysvaiheet tietojen välisten suhteiden visuaalisesta esitystavasta ohjelmaan kehitetyssä tehtäväkartassa. Luku 7 luo yhteenvedon tutkimuksen löydöistä ja luvusta 8 löytyy työssä käytetyt viitteet.

2. Datan visualisointi

Visuaalinen mallinnus on yleinen tapa esittää tietoa. Visuaalisilla malleilla tarkoitetaan esitystapaa, jossa käytetään tekstin sijaan kuvia. Tyypillinen käyttäjä käyttää visuaalisia malleja päivittäin, monesti huomaamattaan. Niiden tavoitteena on esittää tieto ymmärrettävästi prosesseista, datasta ja kehitettävän ratkaisun vuorovaikutuksesta. (Chen & Beatty, 2012)

Kun tiedon määrä kasvaa, keskeiset kohdat tiedon tuomasta viestistä saattaa helposti jäädä piiloon. Visuaaliset kommunikointitavat auttavat tällaisessa tilanteessa keskittämään oleellisen tiedon helposti sisäistettävään visuaaliseen kerrontamuotoon. Visualisointien tärkeimpänä tehtävänä on kertoa tarina kaavioiden avulla ja tuoda esiin usein piilossa oleva viesti. Yleisiä visualisoinnin kohteina ovat datan arviointi, kollegoille kommunikoiminen, asiakkaille tilastojen esittäminen ja esimerkiksi raportointi erilaisten säännösten noudattamisesta (Rodriguez & Kaczmarek, 2016) Datalla tarkoitetaan tässä havaintoja, laskelmointeja tai mittauksia, jotka sisältävät tietoa, jota voidaan käyttää tietolähteenä (Australian Bureau of Statistics, 2023). Vaikeista ja monimutkaisista tiedoista voidaan saada helposti ymmärrettävää käyttämällä visuaalisia esitystapoja. Kuviot ja kaavat, joita ei muuten välttämättä datasta paljaalla silmällä huomasi, saattavat tulla selkeästi nähtäväksi. (Krause, 2012)

Toisaalta huonosti käytettynä visualisoinneilla voi olla myös haittavaikutuksia. Tärkeää tietoa saattaa jäädä havaitsematta tai se saattaa luoda vääriä tulkintoja. Jos taas ei käytä visualisointeja tai graafeja ollenkaan, niin datan tuomat tiedot saattavat jäädä datan tulkitsijalta kokonaan havaitsematta. Jos dataa hyödynnetään esimerkiksi turvallisuustehävissä, silloin huonosti esitetty visualisointi voi pahimmassa tapauksessa peittää datan esittämän turvasignaalin kokonaan. Kun datasta puuttuu rakenne, siitä ei välttämättä edes huomaa, ettei kyseinen datanesitysmalli ole tilanteeseen sopiva, jolloin ongelmaa ei edes tiedosteta. (Krause, 2012)

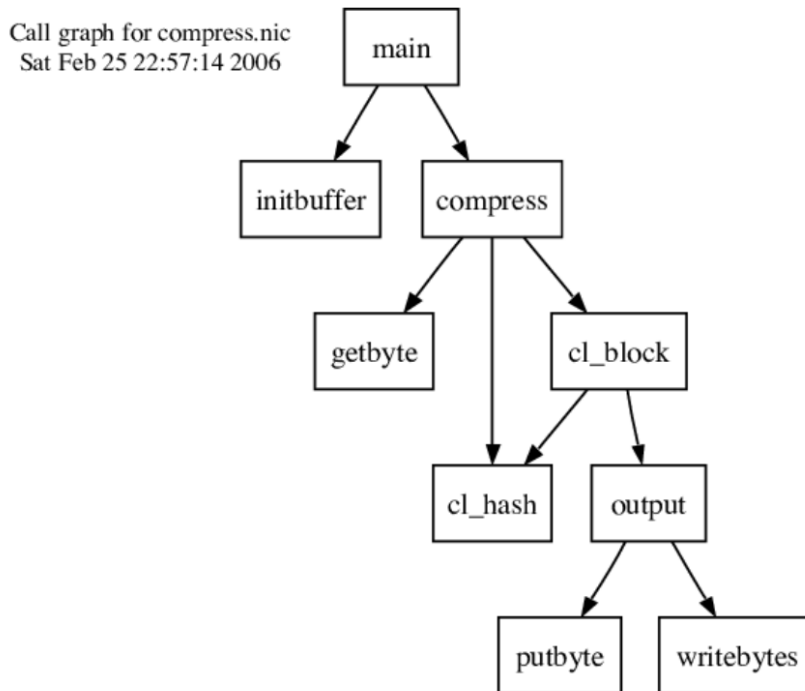
Dataa on montaa eri tyyppiä. Numeerisella datalla tarkoitetaan tietoa, joka nimensä mukaisesti käsittelee numeroita. Sille tyypillistä on, että arvoilla on numeeriset merkitykset, sekä data sisältää lukumääriä tai mittoja. Datan luonteen vuoksi sitä kutsutaan kvantitatiiviseksi dataksi. Tällä tarkoitetaan sitä, että kyseinen data kuvastaa määriä. (Griffiths, 2009)

Dataa voidaan myös jakaa erillisiin lohkoihin sen ominaisuuksien mukaan. Erilaisista kategorioista esimerkkinä toimii videopelien lajityypit, kuten esimerkiksi toiminta- tai

seikkailupelit. Jokainen eri lajityyppi näistä luo oman kategoriansa. Kategorista dataa voi myös kutsua laadulliseksi dataksi. Laadullinen data eroaa numeerisesta datasta siten, että arvoilla ei ole numeerisia merkityksiä. (Griffiths, 2009)

Statistiikalla tarkoitetaan numeroita, jotka esiin jotain tietoa faktoista. Pelkät numerot yksinään eivät ole siis статистиikkaa. Statistiikan avulla pyritään esittämään niitä asioita, joita ei välttämättä pelkästä datasta suoraan huomaisikaan. Statistiikan tulisi olla sellaista, että siitä pystyisi tekemään selkeästi johtopäätöksiä ilman väärintulkintoja. Esimerkkinä статистиikasta voidaan käyttää jalkapallopelien tulosten tulostauluja. Sen sijaan, että saisit ison listan joukkueita eri tuloksilla, voit nopeasti tarkistaa suosikkijoukkueesi sijainnin tilastoissa, joka onnistuu nopeasti katsomalla jalkapallostatistiikkoja. (Griffiths, 2009)

Visualisointi on keskeisessä roolissa myös ohjelmistokehityksessä. Aikaisemmin esitettyjen visualisointien lisäksi ohjelmistokehityksessä visualisoidaan ohjelmistoartefakteja, kuten lähdekoodia, ohjelmien käsittelemää dataa, ja niiden ominaisuuksia. Näihin ominaisuuksiin voidaan lukea tietojen tai joukkojen koko, kompleksisuus ja erilaiset tietojen tai järjestelmien väliset riippuvuudet. Järjestelmäriippuvuuksien visualisoinnissa päätavoitteena on esittää ohjelman osien suhteita ja riippuvuuksia toisiinsa. Esimerkkeinä näistä voidaan käyttää ohjausriippuvuutta, datariippuvuutta ja tiedonkulkua. Näihin on yleensä käytössä niille tyypilliset graafit, kuten CFG, PDG/SDG, ohjelman pilkkomiskaavio (program slicing) ja esimerkiksi kutsumiskaavio (call graph) (kuva 1). Visualisoinneilla voidaan auttaa ohjelmistokehittäjiä tai ohjelmistojen ylläpitäjiä ymmärtämään heidän käyttämiensä järjestelmien ja tietojen toimintoja paremmin. (Mao, 2009)

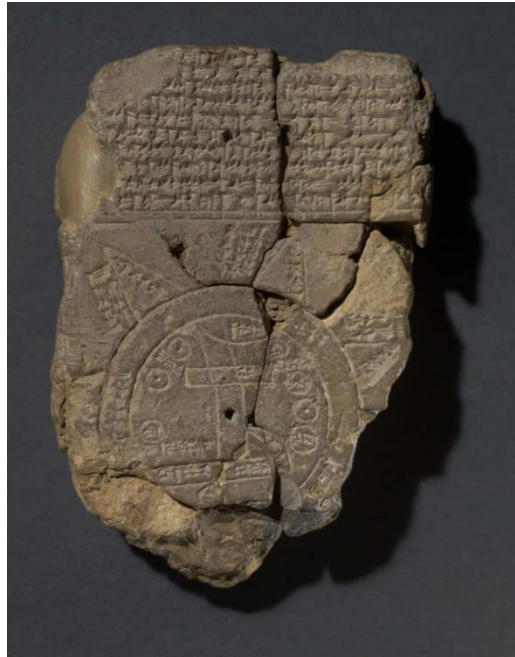


Kuva 1 Esimerkki kutsumiskaaviosta (Researchgate¹)

2.1 Visualisoinnin historiaa

Usein tilastollisia grafiikoita ja tietojen visualisointia pidetään suhteellisen tuoreina konsepteina tilastotieteessä, vaikka todellisuudessa ne juurtuvat jo kartanvalmistuksen historiaan. (Chen et al., 2008) Ensimmäisen kartan, Babylonian maailmankartan (kuva 2), arvioidaan ajoittuvan yhdeksännen ja kuuden vuosisadan eaa. välille (Delnero, 2017). Tämän jälkeen visualisoinnin tekniikoita käytettiin muun muassa muihin temaattisiin karttoihin, tilastoihin, sekä lisäksi myös monet tieteenalat kehittyessään ottivat käyttöön monenlaisia grafiikoita. Visualisoinnilla on juuria myös geometrisissa kaavioissa sekä tähtien ja muiden taivaankappaleiden sijaintitaulukoissa. Tähtien sijainteja hyödynnettiin muinaisessa Egyptissä kaupunkien sijaintien koordinaattien määrittelyssä varhaisimmillaan jo vuonna 200 eaa. Näiden lisäksi 1800-luvulla tilastollisen ajattelun ja tiedonkeruun suosion kasvaessa tiedon visualisointi oli alalla keskeisessä roolissa. Kaikki nämä ovat yhteydessä datan visualisointiin ja sen kehitykseen, mitä meillä on käytössä nykyään. (Chen et al., 2008)

¹ https://www.researchgate.net/figure/Example-of-a-call-graph-for-a-benchmark-program-compress_fig1_221552165



Kuva 2 Babylonian maailmankartta (Britishmuseum²)

2.2 Visualisoinnin tavoitteet

Usein pelkkä data yksinään on vaikeasti tulkittavissa, varsinkin, jos tietoa on kymmeniä tai jopa satoja rivejä. Hyvänä esimerkkinä tästä toimii projektinhallintaohjelmistoissa ison projektin tehtävälista. Ihmisen aivojen sietokyky tulee tällaisen tietomäärän skannaamisessa nopeasti vastaan. Kognitiivinen psykologi George A. Miller tutki jo vuonna 1950, että ihmisen on mahdollista prosessoida kerrallaan vain viidestä yhdeksään asiaa. 2000-luvun alkupuolella tehdyt tutkimukset ovat havainneet, että tämä luku saattaa olla jopa tätäkin pienempi. Professori Cowanin (2001) mukaan tämä luku voisi olla lähempänä kolmea tai neljää. (Chen & Beatty, 2012)

Tämän takia data tulisi aina joko pilkkoa pienempiin osioihin tai visualisoida kuvien avulla, sillä kuvien sanotaan vastaavaan arvoltaan tuhat sanaa. Tällä sanonnalla tarkoitetaan, että vaikeita ja monimutkaisiakin ideoita pystyy esittämään helposti ja ymmärrettävästi visualisoinnilla ja graafeilla. Vaikka tällaisten kuvien selittämiseen ei usein vaaditaakaan tuhatta sanaa, kertoo jokainen kuva silti oman tarinansa huomattavasti selkeäm-

² https://www.britishmuseum.org/collection/object/W_1882-0714-509

min, kuin esimerkiksi monta sivua taulukoituja numeroita ja selitteitä. Kuvan moniulotteisilla ominaisuuksilla pystytään ylittämään numeroiden ja sanojen rajoitteet. (Krause, 2012)

Jos pelkkä kuva ei onnistu selittämään tilannetta tai sen merkitystä, voidaan kuvien avuksi ottaa niiden yhteyteen myös tekstimuotoinen selite. Tällaista mallia kutsutaan visualisoinnin hybridimalliksi, sillä siinä yhdistetään sanallisen ja kuvallisen selityksen ominaisuudet. (Wu et al., 2020)

Datan esittämisessä yleisenä tavoitteena pidetään datan sisältämien tietojen muuttamista ymmärrettäväksi kokonaisuudeksi, josta saadaan selkeästi datan päätarkoitus irti. Tärkeimpinä visualisointien ominaisuuksina Krause (2012) listaa tarkkuuden, helppouden, sekä nopeuden, jolla asia ymmärretään. Graafien tarkoituksena ei ole vain olla tilantäyttäjänä, josta ei saa tietoa irti, vaan avata datan sisältämät informaatiot sen katsojalle. Tämän takia graafien ja visualisointien suunnittelussa on oleellista ottaa huomioon kohderyhmänsä tausta ja osaaminen. Jos graafin suunnittelija on esimerkiksi ohjelmistokehittäjä, hän saattaa helposti pitää joitakin oman alansa asioita itsestäänselvytenä. Jos graafin tulkitsija olisikin esimerkiksi lääketieteen alalla, suunnittelijan mielestä olleet itsestäänselvytykset ovatkin täyttä mystiikkaa lukijalle, ja graafin koko tarkoitus menee hukkaan. (Krause, 2012)

Tilastoihin perustuvalla visualisoinnilla voi Clevelandin (1993) mukaan olla neljä erilaista tavoitetta. Näitä ovat tietojoukon sisällön tutkiminen, datan rakenteen selvittäminen, tilastollisista malleista oletuksien tarkistaminen, sekä analyysin tuloksista keskusteleminen. (Krause, 2012)

2.3 Visualisoinnin prosessi

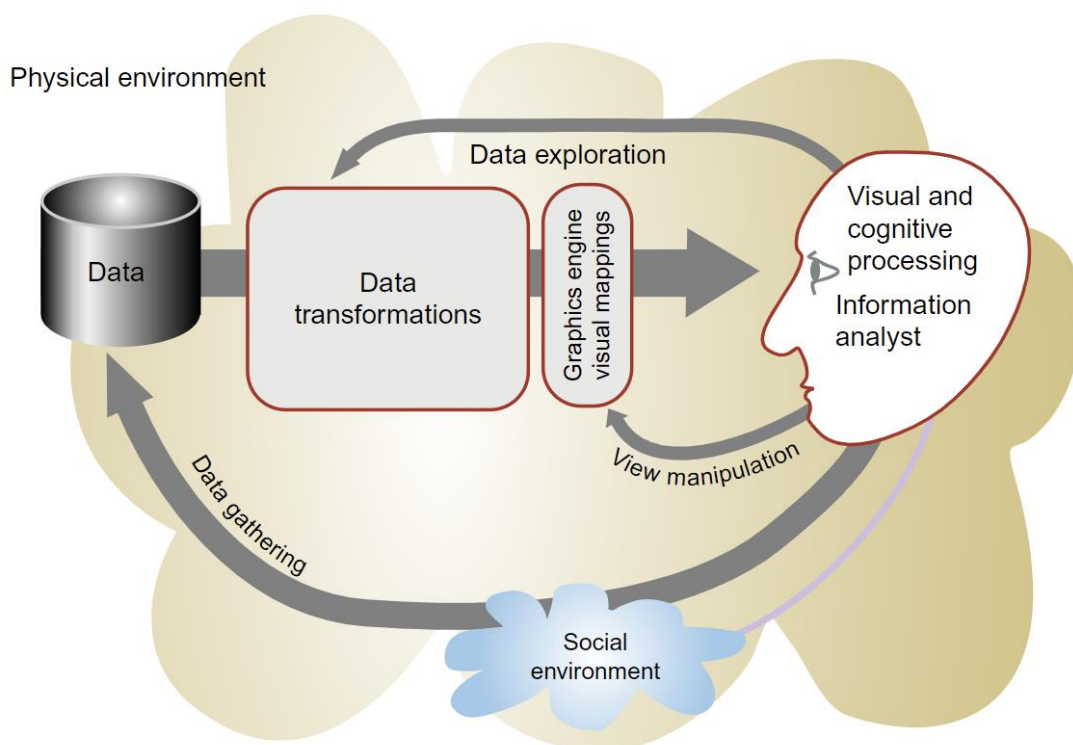
Datan visualisointi on prosessi, jossa käydään läpi neljä eri vaihetta (kuva 3). Jokaisessa vaiheessa on myös hyvä pitää käytössä jatkuvia palautesilmukoita. Ensimmäinen vaihe sisältää tiedon keräämisen ja tallettamisen. (Ware, 2012)

Seuraavaksi datasta tulisi saada helposti ymmärrettävämpää. Ylimääräiset ja tarpeettomat tiedot karsitaan ja vain vaaditut kokonaisuuden kannalta oleelliset asiat säilötään. (Ware, 2012)

Kolmannessa vaiheessa aikaisemmin kerätystä ja käsitellystä datasta luodaan visuaaliset esitystavat. Usein tässä käytetään erilaisia ohjelmistoja luomaan kerätystä datasta

kyseiset näkymät käyttäjän näytölle. Tässä vaiheessa usein myös käyttäjälle annetaan mahdollisuus vaikuttaa näkymiin eri toimintojen avulla. (Ware, 2012)

Viimeinen vaihe on kyseisen visualisoinnin vastaanottaminen ihmisen, tässä tapauksessa visualisoinnin tulkitsijan, havainto- ja kognitiivisella järjestelmällä. Visualisoinnin katsoja tulkitsee visualisoinnin sanoman, ja täten prosessi on valmis. Jokainen vaihe on erityisen tärkeä, jotta viimeisen vaiheen tulkinta olisi mahdollisimman nopea ja todenmukainen. (Ware, 2012)



Kuva 3 Visualisoinnin vaiheet (Ware, 2012)

2.4 Visualisoinnin konkreettiset hyödyt

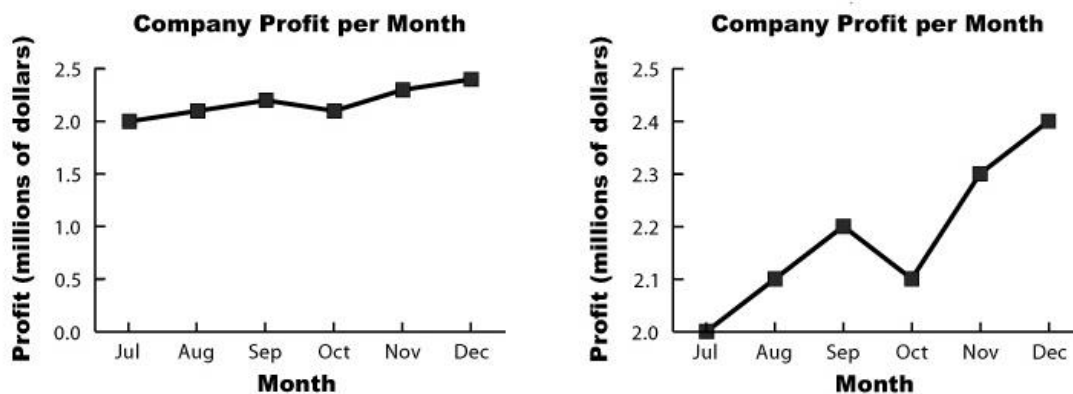
Visualisoinneilla voidaan esittää suuriakin määriä monimutkaista tietoa tehokkaasti ja ymmärrettävästi. Tuhansien rivien sarja numeroita tai muita tietoja olisi lähes mahdotonta ihmisen käsitellä nopeasti ja ymmärrettävästi. Ihmisen on lähes mahdotonta käsitellä nopeasti tuhansia rivejä erilaista tietoa. Tiedon visuaalisella esitystavalla tämä este voidaan rikkoa. Parhaimmillaan tuosta suuresta määrästä dataa saadaan lähes välittömästi ymmärrettävä selkeä kaavio.

Visualisoinneilla pystyy tehokkaasti tiivistämään tiedon, jota raaka data sisältää. Tuhansien rivien selaamisen ja analysoinnin sijaan, nopealla katseella kaavion saada yleiskuva selväksi. Siksi oikeanlaisten kaavioiden ja asteikoiden valitseminen on tärkeää, jotta et niitä tehdessäsi vahingossa ohjaa katsojaa väärään suuntaan ja anna viestiä, mitä et tarkoittanut. Myös katsojan täytyy pitää mielessä se, että jopa hyvinkin yksinkertaisen oloinen kaavio saattaa tarkoituksella yrittää ohjata mielikuvaasi kaavion tekijää hyödyttävään suuntaan. (Griffiths, 2009)

2.5 Visualisoinnin potentiaaliset ongelmat

Tiedon visualisointi ei kuitenkaan hyödyistään huolimatta ole aina oikea ratkaisu. Sen lisäksi, että tulisi valita oikea esitystapa, tulee myös arvioida kyseisen visualisoinnin hyöty.

Visualisoinnit eivät ole aivan yksiselitteisiä. Niitä ei vain voida laittaa jokaiseen tilanteeseen satunnaisesti, vaan ne tulee valita harkiten. Myös visualisoinnin poisjättäminen on omalla tavallaan päätös. Väärin käytettynä visualisoinnit saattavat ohjata sen katsojaa väärin tulkintoihin (kuva 4), mutta puutteelliset visualisoinnit saattavat puolestaan jättää oleellista tietoa piiloon. (Krause, 2012)



Kuva 4 Harhaanjohtava asteikko (Griffiths, 2009)

Tufte viittaa tekstissään musteella pikseleihin, joita visualisoinneissa on käytetty. Hänen mukaansa on olemassa tietty havaintojen ja musteen suhde, joka tulisi pitää kohtal- laan. Tavoitteena tulisi olla pitää havaintojen määrä mustetta kohden mahdollisimman

korkeana. Liiallinen määrä mustetta havaintoa kohden tarkoittaa, että visualisoinnissa olisi vielä kehittämisen varaa. Vähäinen määrä mustetta taas indikoi sitä, että visualisointi on tehty tehokkaasti. (Krause, 2012) Tässäkin tilanteessa täytyy ottaa huomioon mahdollinen tietojen piiloutuminen, jos mustetta käytetään liiankin vähän.

Visualisoinneissa myös negatiivisen tilan, myös kutsutaan valkoiseksi tilaksi, käyttäminen on erityisen tärkeää. Se on osa mitä tahansa visualisointia. Aivan kuten saattaisit lisätä asettelussa muita näkyviä asioita, voit myös käyttää yhtä lailla negatiivista tilaa. Negatiivisen tilan avulla voidaan kiinnittää katsojan huomio haluttuihin pisteisiin. Toisaalta se antaa myös katsojan silmälle levähdysalueen, jolla ei ole laisinkaan tietoa tarjolla. (Samara, 2014)

Negatiivisen tilan käyttämisessäkin tulee olla tarkka. Liian vähäinen negatiivinen tila taas saattaa aiheuttaa sen, että esityksesi on aivan liian täynnä visuaalisia kohteita, joka taas saattaa tuntua katsojasta raskaalta tai ahdistavalta. (Samara, 2014)

2.5.1 Käsityksen muuttuminen odotetun tuloksen perusteella

Ihmiselle tyypillistä on ylhäältä alas huomio ominaisuus- ja alkeismallianalyysin mukaan. Se saattaa aiheuttaa ongelmia, sillä silloin katsoja huomioi herkemmin ne asiat, joita hän etsii. Esimerkiksi, jos katsoja etsii punaisia pisteitä, silloin punaisen pisteiden ilmaisimet antavat kovempia signaaleja muihin verrattuna, ja käyttäjä löytää herkemmin juuri näitä pistettä mitä etsikin. Sama pätee myös esimerkiksi vinojen linjojen etsimisessä. Tämä tapahtuu nähdyn käsittelyn jokaisessa vaiheessa ja tilanne usein päättyykin siihen, että kaikki mitä havaitsemme, on kallistunut sitä kohti, mitä tavoittelemme. (Spinde, 2022)

2.5.2 Semioottinen selkeys

Symboleilla ja niiden merkityksillä ja konsepteilla tulisi olla yksi-yhteen suhde. Tämä perustuu Goodmanin symbolien teoriaan, jolla pyritään merkintöjen osuvan merkintäjärjestelmän vaatimuksiin. Näihin vaatimuksiin sisältyy tarkkuus, ilmeisyys, sekä vähävaraisuus. Tämä tekee merkinnöistä haluttuja ja käytännöllisiä ohjelmistokehityksessään. Toisin kuin luonnollisessa kielessä, merkintäjärjestelmien pitäisi olla yksiselitteisiä. Monilla keinotekoisilla kielillä toisaalta löytyy merkintäjärjestelmän ominaisuudet, kuten esimerkiksi musiikin ja matematiikan merkistöillä. Jos merkinnät eivät täytä yksiselitteisyyden määritelmää, on sillä riskinä aiheuttaa näitä:

1. Symbolien päällekkäisyys, jolloin useat merkinnät osoittavat samaan merkitykseen.
 2. Symbolien ylikuormittaminen. Silloin samaa asiaa voidaan merkitä usealla eri symbolilla.
 3. Symbolien ylimääräisyys. Symboleita esiintyy, joilla ei ole mitään merkitystä.
 4. Symbolien alijäämä. On merkityksiä, joille ei ole olemassa symboleita laisinkaan.
- (Moody, 2009)

3. Visualisoinnin käytänteet

Suunnittelijan päätavoitteena on esitellä ideoita ja tietoa, jotka pyritään yhdistämään käyttäjälle kokemukseksi. Lopputuloksella tulisi saada käyttäjälle aikaiseksi tietty reaktio tai tunnetila. Suunnittelussa apuna voidaan käyttää kuvia, symboleja, väriä ja eri materiaaleja. (Samara, 2014)

Semiotiikan ymmärtäminen on suunnittelijalle välttämätöntä. Tällä tarkoitetaan prosessia, jossa visuaalisilla elementeillä tai sanallisella esityksellä suunnittelija viestii tarkoittamansa merkityksen loppukäyttäjälle. Suunnittelu vaatii kuvien analyttistä ja teknistä ammattitaitoa. Suunnittelijan tulee ymmärtää muotojen, värien ja tekstuurien eri tavat kuvata ideoita ja luoda esteettistä koheesiota ja dynaamisuutta. Näiden tulisi herättää käyttäjässä halutunlaisia vahvoja tunnereaktioita. (Samara, 2014)

3.1 Visuaaliset merkinnät

Visuaalinen merkintä (visual notation) pitää sisällään kolme pääkohtaa: graafinen merkintä, tietyt ennalta määritellyt sommittelusäännöt, sekä merkityksestä, jota merkintä symboloi. Nämä yhdessä luovat visuaalisen syntaksin. Symboleilla on aina tarkoituksena ilmaista jokin semanttinen merkitys. Visuaalisesta merkinnästä voidaan käyttää myös termejä visuaalinen kieli, graafinen merkintä ja kaaviomerkintä. (Moody, 2009)

Visuaaliset merkinnät ovat oleellisessa roolissa kaikissa ohjelmistokehityksen vaiheissa, jo vaatimusten määrittelystä ylläpitoon asti. Visuaalisille merkinnöille uskotaan olevan ominaista se, että niiden avulla pystytään välittämään tietoa paremmin henkilöille, joilla ei ole entuudestaan suurta erityysosaamista käsiteltävästä aiheesta. Tämä tekee niistä erityisen tärkeitä varsinkin asiakkaan ja loppukäyttäjien kanssa kommunikoinnissa, sen sijaan, että turvauduttaisiin tekstiin, jota vain alan ammattilaiset ymmärtävät. (Moody, 2009)

Visuaalisten esitystapojen tehokkuus perustuu ihmisen visuaaliseen järjestelmään. Ihminen pystyy käsittelemään visuaalista informaatiota tehokkaasti, ja noin neljäsosa ihmisten aivoista onkin omistettu näköaisteille. Tämä on enemmän kuin kaikki muut aistit yhteensä. Visuaalisissa esitystavoissa myös etuna on se, että kuvien paremmuusefektin (picture superiority effect) vuoksi niiden sanotaan jäävän helpommin muistiin. Näiden lisäksi kaavioilla pystytään luonnollista kieltä tarkemmin ja selkeämmin esittämään vaadittu tieto. (Moody, 2009)

Jotta visuaalisesta merkinnästä saa luotua mahdollisimman tehokkaan, täytyy ymmärtää niiden käyttötarkoitus. Sillä visuaaliset notaatiot ovat ohjattu ihmisille, tulee ne

suunnitella mahdollisimman selkeästi ihmismielelle käsiteltäväksi. Kognitiivinen tehokkuus merkitsee sitä, kuinka nopeasti, helposti ja tarkasti esitys voidaan käsitellä ihmismielellä. Moody (2009) esittää tätä ensisijaiseksi muuttujaksi visuaalisten merkintöjen laadulliseen arviointiin ja vertailuun, sekä merkintöjen tärkeimmäksi tavoitteeksi niitä suunnitellessa. Merkintöjen kognitiivinen tehokkuus on erittäin tärkeää, sillä se määrittää merkinnän kyvyn viestiä tietoa, sekä avustaa kehittäjiä ongelmien ratkaisussa. (Moody, 2009)

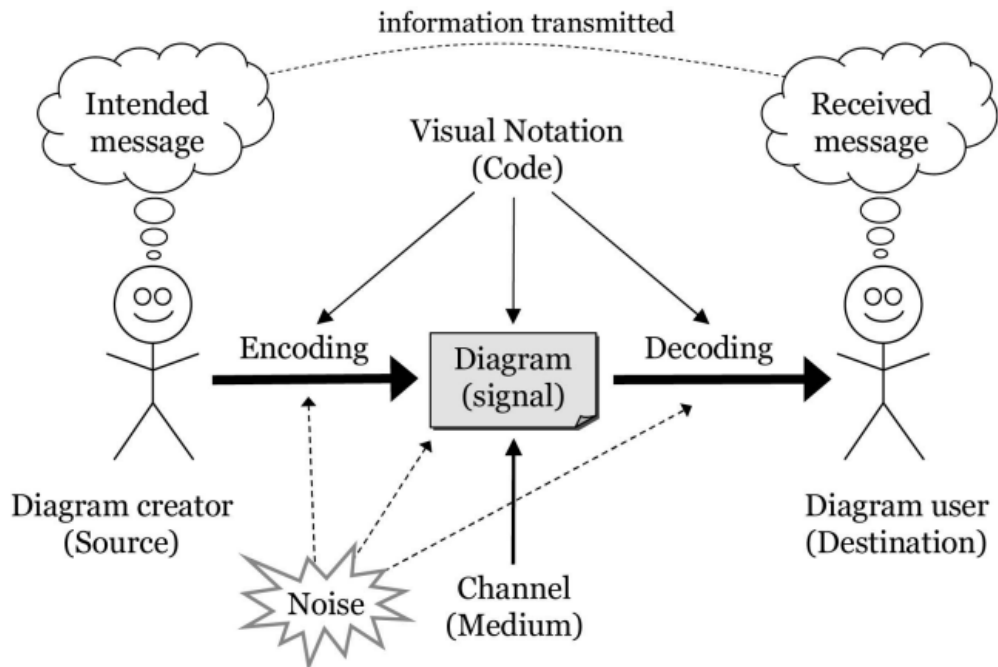
Kognitiivinen tehokkuus ei kuitenkaan ole visuaalisten merkintöjen luontainen ominaisuus, joten se tulee suunnitella merkintöihin tarkasti. Pelkkä tiedon muuttaminen visuaaliseksi merkinnäksi ei tarkoita sitä, että se toimisi tehokkaasti tiedon välityksessä ja ongelmanratkaisussa. Jos visuaalinen merkintä suunnitellaan huonosti, voi se olla huonompi vaihtoehto kuin pelkkä tekstiesitys. (Moody, 2009)

Visuaaliset merkinnät eivät kuitenkaan ole saaneet suurta huomiota ohjelmistokehityksessä, vaikka niiden käyttö on alalla yleistä. Tähän ei ole saatu yhtä selitystä, mutta yhtenä syynä on epäilty visuaalisten merkintöjen näkemistä epävirallisina matemaattistaustaisten tutkijoiden keskuudessa. Tämä ei kuitenkaan pidä paikkaansa, sillä visuaaliset kielet eivät ole vähemmän virallisia kuin tekstikielet. Toisena selityksenä on pidetty sitä, että visuaalisten esitystapojen analysointimenetelmät ovat tuorempi ilmiö kuin esimerkiksi matemaattiset- tai tekstianalyysit. Mahdollisena pidetään myös sitä, että tutkijat ja suunnittelijat ohjelmistokehityksen alalla eivät vain pidä visuaalista syntaksia tärkeänä. Niitä saatetaan helposti pitää turhina ja vain esteettisinä lisukkeina sen sijaan, että ne toisivat lisäarvoa. (Moody, 2009)

3.1.1 Visuaalisten merkintöjen kommunikaatio

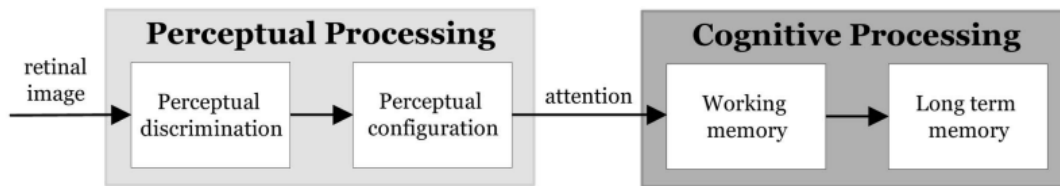
Jotta hyvän visuaalisen merkinnän pystyy suunnittelemaan, tulee suunnittelijan ymmärtää miten, miksi ja mitä ne kommunikoivat. Kaavioviestinnän teorian (kuva 5) mukaan viestintä sisältää kaksi prosessia: koodauksen (encode) ja dekodeauksen (decode). Tämän teorian mukaan koodauksella tarkoitetaan ilmaisua, ja dekodeauksella ilmaistun viestin tulkintaa. Jotta viestintä on mahdollisimman tehokasta, tulee molemmat puolet ottaa suunnittelussa huomioon. Koodauksessa tulee huomioida mahdolliset visuaaliset tekni-

kat ja vaihtoehdot tiedon välittämiseksi ja dekodauksessa tulee ymmärtää miten ihmis-
mieli ymmärtää merkinnän. Näihin valintoihin saadaan luotua pohja ymmärtämällä ihmi-
sen tiedonkäsittelyn periaatteita. (Moody, 2009)



Kuva 5 Visuaalisten merkintöjen kommunikaatio koodauksen ja dekodauksen kautta.
(Moody, 2009)

Kuvassa 6 esitellään ihmisen käsittelyn vaiheita graafisen tiedon vastaanottamisessa. Ensimmäisenä vaiheena on havaitseminen eli visuaalisissa merkinnöissä näkeminen. Tämä prosessi ei vaadi tietoisia toimia, vaan käsittely toimii automaattisesti erittäin nopeasti. Tässä prosessissa käsittely toimii myös rinnakkain suorittaen. Seuraavaksi tulee kognitiivinen käsittely eli havaitun kohteen ymmärtäminen. Tämä prosessi on hidas ja peräkkäin toimiva. Se myös toimii tietoisien huomion hallinnassa ja on usein kognitiivisesti vaativaa. Visuaalisten yksi tehokkuus perustuu näiden käsittelyprosessien hyödyntämiseen. Niissä osa prosessoinnista siirretään kognitiiviselta järjestelmältä havaintojärjestelmälle, jolloin käsittely tapahtuu helpommin ja nopeammin, ja kognitiivisia resursseja jää yli muihin tehtäviin. Erot kaavioiden tehokkuuden välillä usein selittyvät sillä, kuinka paljon kyseinen kaavio hyödyntää havaintokäsittelyä. (Moody, 2009)

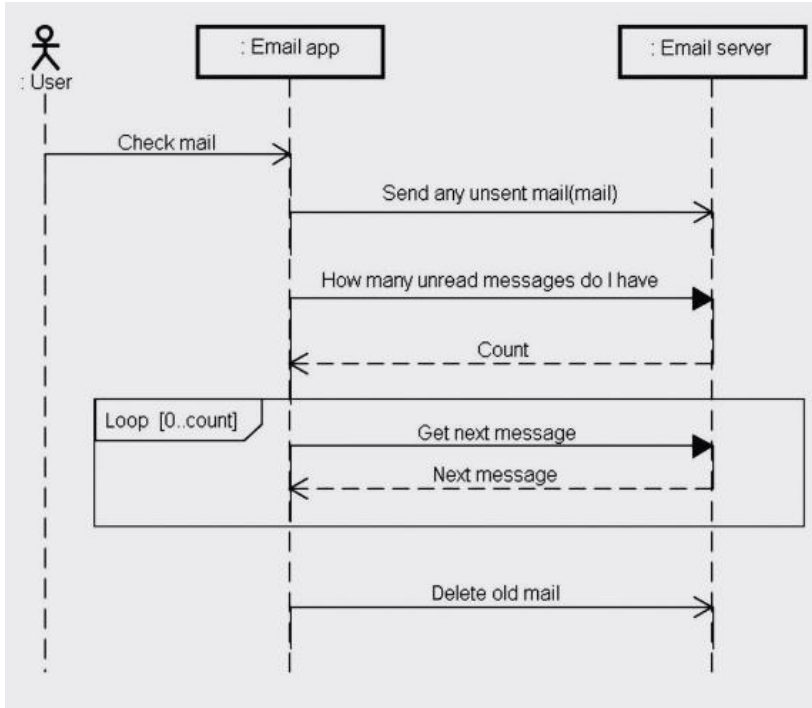


Kuva 6 havainto- ja kognitiivisen käsittelyn prosessi (Moody, 2009)

3.1.2 Ohjelmiston mallintaminen UML:lla

Edellä mainituissa luvuissa esitellyt mallit esiintyvät ohjelmistokehityksessä isolta osin UML:n merkeissä. Sitä on kuvailtu graafiseksi kieleksi järjestelmien visualisointiin, määrittelyyn, rakentamiseen ja dokumentointiin. (Moody, 2009) Nykyään se onkin käytetössä standardina ohjelmistojärjestelmien mallintamisessa. Ennen UML:llä, ohjelmistokehityksen alalla ei ollut yhteistä visuaalista kieltä, vaan erilaiset käsitteet ja mallit kilpailivat keskenään. (Gašević et al., 2009) Tämä vaikeutti ja hidasti olioihin perustuvien kielten ja työkalujen kasvua. (Chonoles, 2017)

UML 90-luvun loppupuolella kehitetty visuaalinen mallinnuskieli, jota käytetään laajasti ohjelmistojen mallintamiseen. UML ei vaadi värien käyttöä ja sitä pystyykin tekemään lähes millä työkalulla tahansa, esimerkiksi UML mallinnusohjelmilla, PowerPointilla tai vaikka valkotaululla. Se eroaa ohjelmointikielistä paljon ja sen syntaksin on sanottu olevan lähempänä englannin kieltä kuin ohjelmointikieliä. UML:n tavoitteena on olla mallinnuskieli, joka on ymmärrettävää sekä koneille, että ihmisille. Sen käyttäminen sopii sekä pieniin, että laajoihinkin järjestelmiin. Kuvassa 7 näkyy esimerkkinä UML sekvenssikaavio sähköpostien käsittelemisestä. (Chonoles, 2017)



Kuva 7 Sekvenssikaaviolla esitetty kuva sähköpostin prosessoinnista (Chonoles, 2017)

3.2 Värien merkitys visualisoinneissa

Visualisointeja suunniteltaessa yhtenä tärkeänä vaiheena on värien käytön suunnittelu. Värit sisältävät suuren määrän erilaisia psykologisia ja tunteellisia merkityksiä. (Samara, 2014). Sen sijaan, että valittaisiin vain satunnaiset värit, tulisi niillä olla jokin käyttötarkoitus. Väreillä voidaan merkitä, mitata ja elävöittää dataa. Oikein käytettynä sävyillä voidaan auttaa tulkitsijaa tunnistamaan ja erottamaan kohteita ja kategorioita toisistaan. Säättämällä kylläisyyttä ja valonvoimakkuutta voidaan myös tukea kvantitatiivisia vertailuja. (Lin et al., 2013)

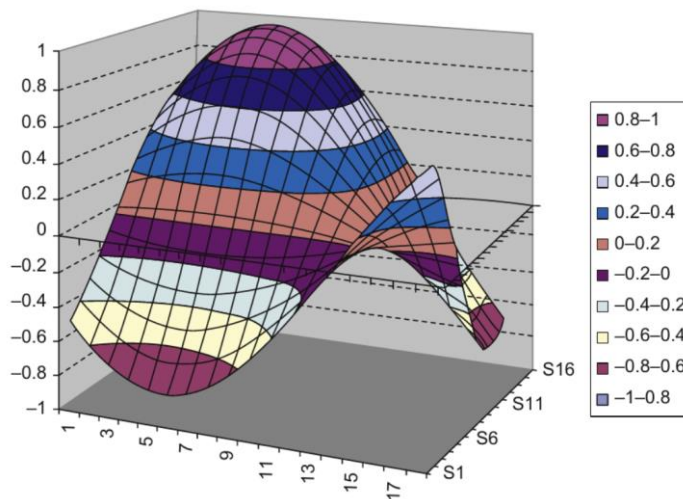
Hyvin käytettyinä värit voivat olla yksiä parhaimpia visualisointitekniikoita. Siitä huolimatta värit ovat yksi eniten väärinkäytettyjä ja laiminlyötyjä tekniikoita visuaalisointien suunnittelussa. Väärinkäytöillä tässä tarkoitetaan huonoja värivalintoja, ja laiminlyönneillä sitä, että tyydytään huonoihin oletusasetuksiin, vaikka hyvällä suunnittelulla olisi mahdollista valita tilanteeseen sopivimmat. (Steele & Iliinsky, 2010).

Oikein käytettynä väreillä pystytään kiinnittämään käyttäjän huomio haluttuun osa-alueeseen. Tämän lisäksi niillä pystytään yhdistelemään asiakohteita ja kokonaisuuksia toi-

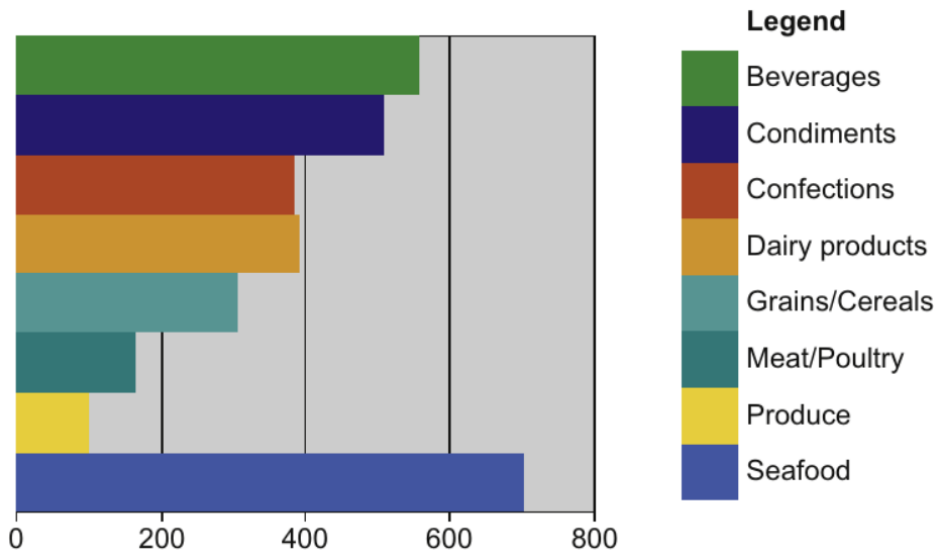
siinsa, sekä osoittaa eri merkityksiä. (Lidwell et al., 2010) Esimerkiksi punainen virheviesti on huomattavasti huomiota kiinnittävämpi, kuin esimerkiksi vihreä, sillä punainen yhdistetään usein vaaraan. Värejä käyttämällä pystytään myös parantamaan yleisesti sivuston tai sovelluksen ulkonäköä. (Lidwell et al., 2010)

Työskentelyvälineillä on usein isompi rooli suunnittelussa kuin oletetaan. Esimerkkinä tästä toimivat näytöt. Eri näytöissä värien kalibroinnit saattavat oletusasetuksilla poiketa huomattavasti toisistaan. Tämä taas saattaa johtaa siihen, että omalla näytöllä, joka on huonosti kalibroitu, värit näyttävät juuri oikeilta ja sopivilta. Kuitenkin, kun muut katsovat visualisointeja omilla näytöillensä, saattavat värit näyttää hyvin erilaisilta. Tämän takia on oleellista, että oman työpisteen välineet ovat sekä hyvä laatuaisia, että asetuksiltaan kunnossa, jotta näet, miltä värivalintasi todellisuudessa näyttävät. (Hardy, 2011)

Datakaavioille on tyypillistä pienet värimerkinnät. Varsinkin valmiita graafeja tarjoavien pakkausten mukana graafin yhteyteen tulee usein myös värikoodeilla varustettu kuvateksti tai seloste. Selosteissa on usein ongelmana se, että väreinä on käytetty toisistaan heikosti erottuvia, kuten kalpeita värejä, sekä värien laatikot ovat liian pieniä (Kuva 8). Kuten aikaisemmin esitettiin, tämä vaikeuttaa niiden erottamista ja heikentää graafin luettavuutta. Tämän sijaan värien tulisi olla voimakkaita ja isolla merkittyjä, jotta niiden erottelu on mahdollista ilman tarkempaa tutkimusta (Kuva 9). Myös tyypillisenä ongelmana on seurattujen ja ei-seurattujen linkkien erottaminen toisistaan. Usein näissä käytetään liian samankaltaisia värejä ja ne ovat tyypillisesti vielä sinisen sävyjä. Lisäongelmaa sinisen värin käyttäminen tuo siksi, että se on silmillemme yksi vaikeimmista väreistä erottaa. (Johnson, 2014)



Kuva 8 Epäselkeät värit kaaviossa (Johnson, 2014)



Kuva 9 Selkeästi eroteltavissa olevat värit kaaviossa (Johnson, 2014)

3.2.1 Värien symboliikka historiassa

Historiassa, muinaisen Egyptin aikaan, väreillä oli yleiset vertauskuvalliset merkitykset. Mustalla värillä viitattiin hedelmällisyyteen, uudelleensyntymiseen sekä uudistumiseen. Valkoisella puhtautta, pyhyyttä ja yksinkertaisuutta. Niiden vastakohtana pidettiin punaista, kaaoksen ja häiriön väriä. Nämä merkitykset eivät kuitenkaan olleet yksiselitteiset, vaan samalla värillä saattoi olla myös useampia merkityksiä. Edellä mainittujen merkitysten lisäksi punainen symbolisoi elämää ja voimaa veren ja tulen välityksellä. (Mehl, 2013)

Väreillä oli näiden lisäksi myös taivaallisia merkityksiä. Sinisillä kuvastettiin Amunia, Jumalaa, joka loi maailman. Sininen liitettiin taivaan, sekä veden väriin. Ne edustivat jumalan valtakuntaa sekä vuotuisia tulvia. Aamunkoitteen aurinkoa, kuuta sekä tähtiä kuvattiin hopealla. Vihreä kuvasti hyvinvointia ja parantumista. Kultaista ja keltaista käytettiin maalauksissa kuvaamaan ihoa. (Mehl, 2013)

3.2.2 Värien merkitys nykyajassa

Nykyään värit yhdistetään usein tiettyihin asioihin tai konsepteihin ja värien näkemisen uskotaan luovan mielikuvia konsepteista. Sama pätee myös toisinpäin. Jokin tietty asia, jonka tiedetään olevan usein tietyn värinen, yhdistetään helposti siihen väriin alitajuntai-

sesti. Esimerkkinä tästä toimii Yhdysvalloissa tehty tutkimus, jonka mukaan usein keltaisen yhdistetään banaaneihin, punainen vihaan, sekä vihreä väri rahaan. Näiden johdannaisen epäillään liittyvän vahvasti esineiden ulkonäköön, metaforiin tai yleisiin kulttuurisiin tekijöihin. Tällaisista värivalinnoista, joilla luodaan mielikuvia tiettyihin asioihin, voidaan käyttää termiä semanttisesti resonoiva. (Lin et al., 2013)

Värien ja konseptien yhteneväisyys ei kuitenkaan ole yksiselitteistä. Useassa eri kulttuurissa samat värit voidaan liittää täysin eri aiheisiin. Nämä kulttuuriin tai kielelliseen perustuvat assosiaatiot saattavat pahimmillaan aiheuttaa joidenkin käyttäjien keskuudessa väärinymmärryksiä. Se mikä tarkoittaa yhdelle kohderyhmälle jotakin positiivista, saattaa toiselle ryhmälle osoittautua negatiivisena tai jopa loukkaavana. Myös värien tekniset vaikeudet värien esittämisessä eri alustoilla, kuten näytöillä, painettuna ja muilla materiaaleilla, voivat tuoda kehittäjälle epävarmuuksia. Oikein käytettynä väreillä on kuitenkin suuria positiivisia vaikutuksia. Parhaimmassa tapauksessa värien käyttäminen saattaa vaikuttaa käyttäjien ja potentiaalisten asiakkaiden päätöksiin käyttää palveluita. (Sherin, 2012)

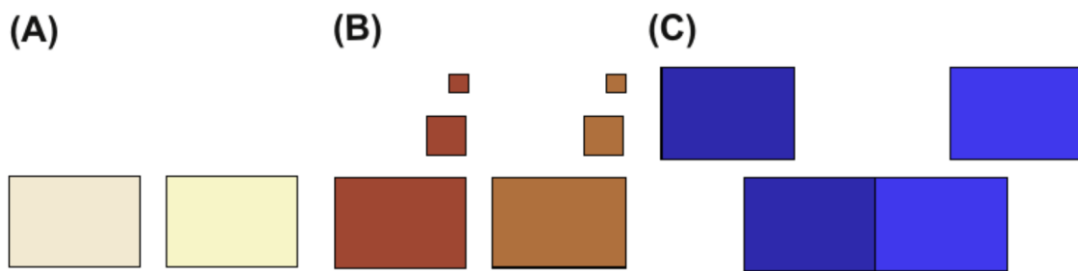
Jokaiseen tilanteeseen värien käyttäminen ei kuitenkaan sovi. Erityisesti kliinisten tulosten raportoinnissa tämä on hyvin kiistanalainen aihe, sillä graafien tulisi olla luettavia ja ymmärrettäviä, vaikka tulostamiseen käytettäisiin väritöntä tulostinta. Tästä syystä kliinisten tutkimusten standardina on edelleen käyttää vain harmaan sävyjä tai mustaa ja valkoista. Myös ylimääräiset elementit kuten varjostus, liukuvärit tai tekstuurit saattavat olla haitallisia selkeyden kannalta, jos ne eivät tuo lisäarvoa tulkinnalle. (Krause, 2012)

Värejä valitessa tulee myös ottaa huomioon kohdeyleisö, jolle kohde on osoitettu. Vaikka värivalinnoissa ei tarvitse aina päätyä stereotyyppisiin valintoihin, on tietyissä tilanteissa värien tarkalla valinnalla silti keskeinen rooli. Jos esimerkiksi asiakkaana on hautausseuran toimisto, kyseisen alan teemaan ei yleisesti sovi monipuoliset ja kirkkaat värit, sillä ne ovat ristiriidassa alan herkän tuotekuvan kanssa. (Hardy, 2011)

3.2.3 Värien erottelu

Värien tosistaan erottaminen riippuu täysin niiden esitystavasta. Johtuen ihmisen visuaalisesta järjestelmästä, meidän kykymme erottaa värejä toisistaan on rajallista. Värien esitystavassa on kolme tekijää, jotka vaikuttavat siihen, kuinka ne hyvin pystytään erottamaan toisistaan. Ensimmäinen näistä on värin kalpeus (A), eli värin puute. Mitä kalpeampia värit ovat, sitä vaikeampi ne ovat erottaa toisistaan. Isona tekijänä on myös värien

alueen koko (B). Mitä pienemmäksi värialue menee, sitä vaikeammaksi värien erottelu toisistaan menee. Hyvänä esimerkkinä tästä toimii teksti, jolla on tietty väri. Koska teksti on usein ohutta ja suhteellisen pientä, sen tarkka väri on vaikea havaita ilman tarkempaa tutkimista. Viimeisenä on värien sijainti suhteessa muihin väreihin (C). Mitä kauemmaksi värialueet menevät toisistaan, sitä vaikeammaksi niiden erottelu menee. Kaikista vaikein on tilanne, jossa kahden värialueen välimatka on niin suuri, että silmien on liikuttava nähdäkseen toinen väreistä. Kuvassa 10 esitellään visuaaliset esimerkit näistä tapauksista. Käytännön esimerkki huonosti toimivien väriyhdistelmien vaikutuksesta oli matkavaurusten verkkosivu, jossa käytettiin kahta vaaleata väriä erottelemaan vaihe, missä käyttäjä oli varauksessaan. Sivusto sai useita valituksia siitä, että käyttäjät eivät onnistuneet erottamaan, missä vaiheessa varausta he olivat. (Johnson, 2014)



Kuva 10 Värien erottelu (Johnson, 2014)

3.2.4 Kirkkaus ja saturaatio ja kontrasti

Värejä voidaan säätää joko saturaation tai kirkkauden avulla. Saturaatiolla tarkoitetaan lisättyä harmaan määrää värisävyssä. Mitä korkeampi saturaatio on, sitä vähemmän harmaata sävyyn on lisätty. Saturaation pienentyessä harmaan määrä taas nousee. Värien kirkkaudella taas viitataan sävyssä olevaan valkoisen määrään. Mitä korkeampi kirkkaus, sitä enemmän valkoista sävyssä on ja mitä pienempi kirkkaus, sitä vähemmän valkoista. (Lidwell et al., 2010)

Kontrastilla tarkoitetaan kahden värin, jotka ovat joko päällekkäin tai vierekkäin, valtiheyden (luminanssi) eroavaisuutta. Värien välisen kontrastisuhteen pystyy määrittelemään vertaamalla värien välistä kirkkauden eroa. Kontrastin noustessa kahden värin erottelu toisistaan helpottuu. Tämän takia vahvat kontrastit ovat erityisen tärkeitä niille käyttäjille, joiden kontrastiherkkyys on heikentynyt. Ne ovat erityisen tärkeitä käyttäjille,

joilla on esimerkiksi kaihi, silmänpohjan rappeuma, glaukooma sekä niille, joilla on värisokeutta. (Firth, 2019) Varsinkin tekstin luettavuuden kannalta olisi hyvä aina pitää mielessä taustan ja tekstin kontrastisuhde.

Ihmisen näkökyvyille värien kirkkautta oleellisempaa on värien välinen kontrasti. Näköjärjestelmämme on huomattavasti herkempi värin ja kirkkauden muutoksille kuin värien absoluuttiselle kirkkaustasolle. Jos kahden toisissaan kiinni olevan osan eroavaisuus kontrastissa on suuri, silloin ne ovat todella helppo erotella toisistaan. Tästä syystä kontrastiset värit sekä reunukset tulisi ottaa huomioon selkeyden parantamiseksi. Kuvassa 11 näkyy, kuinka harmaa sisäpalkki näyttää eri sävyiseltä taustan värin muuttuessa, vaikka todellisuudessa palkki on tasaisesti saman värinen. (Johnson, 2014)



Kuva 11 Kontrastin vaikutus värin havaitsemiseen (Johnson, 2014)

3.3 Hyvän suunnittelun käytänteet

Graafisessa suunnittelussa jokaisessa visuaalisella elementillä on tärkeää olla jokin tehtävä tai viesti, mitä se kommunikoi elementin katsojalle. Tämän takia suunnittelijan on oleellista tietää jokaisen visuaalisen elementin tehtävä ja tarkoitus, joita on joko jo asetettu, tai tullaan asettamaan. Suunnittelussa tyypillinen virhe on kommunikoinnin sijaan esityksen koristelemineen. Jos kuva tai toiminto ei tuo uutta käyttötarkoitusta, eikä lisää aikaisempaan aiheeseen mitään, se on vain koriste, jota ei tule luokitella suunnitteluksi. Kuvien, värien ja tehosteiden kanssa kokeileminen on toki tärkeää, mutta ylimääräistä koristetta, jolla ei ole viestiä, tarinaa, ideaa, kerrontaa eikä hyödyllistä kokemusta, ei tulisi lisätä. (Samara, 2014)

Suunnittelun tulee kohdistua suurelle yleisölle, sen sijaan, että asiat tehtäisiin pienelle asiantuntijaryhmälle. Poikkeuksena on tietty sellainen tilanne, jossa pieni asiantuntijaryhmä on suunnitelman ainut kohdeyleisö. Tärkeänä on siis tietää, kenelle suunnitelmaa tekee. Kohderyhmän pitää pystyä ymmärtämään käyttämiesi muotojen, värien ja kuvien tarkoitus. Tyypillisenä virheenä on tilanne, jossa visualisointisi ymmärtää pieni ryhmä muita ammattilaisia, sekä tekijä itse. Ongelmana tässä on se, että käyttäjinä on useita eri tyyppisiä, joista suuri osa ei ole osa tätä pientä ryhmää. Hyödynnä yhteiskunnan yleisiä tunnettuja tapoja ja metaforia, ja luo yhteyksiä, jotka ovat tyypillisiä tavallisten käyttäjien keskuudessa. Hyvänä tapana varmistaa suunnitelmasi ymmärrettävyys, on esittää se jollekin, kuka ei tunne asian kontekstia ammattilaisen tasolla. (Samara, 2014)

”Vähemmän on enemmän” on yleinen sanonta, ja se pätee myös suunnittelussa. Mitä enemmän pieneen tilaan asetetaan katselukohteita, sitä vaikeampi yksittäisiin kohteisiin on keskittyä ja kohteiden erottelu toisistaan heikkenee. Usein tällaisissa tilanteissa saattaa haluttu kohde jäädä näkemättä kokonaan. Tarvittavan määrän lisäksi lisäominaisuuksien lisäämistä voitaisiin kutsua liljan kultaamiseksi. (Samara, 2014)

Ylimääräisen tiedon välittämistä ja samojen asioiden toistamista tulisi pyrkiä välttämään. Projektissa esiintyvien kuvien ja tekstien kertomia asioita tulisi myös vertailla, sillä jos toinen niistä kertoo tarvittavan tiedon jo, ei sitä ole tarpeellista kertoa uudelleen. Tämän sijaan tulisi pohtia, mitä kuvat tai tekstit jättävät kertomatta ja esittää puuttuvat asiat niitä tukevilla visualisoinneilla ja avustavilla teksteillä. Kuvien ja tekstin ei tulisi ainoastaan toimia yhdessä ja kertoa sama asia kahdella eri tekniikalla, vaan lisätä myös uutta ymmärrettävyyttä aiheeseen. Tämänkaltaisilla tekniikoilla katsoja saadaan vahvasti kiinni esitettyyn asiaan. (Samara, 2014)

Suunnittelussa myös päättäväisyys on oleellinen asia. Sen sijaan, että vain lisäät satunnaisesti osia, tulisi jokaisen ratkaisun olla päämäärätietoisesti tehty. Asioiden tulisi näyttää juuri siltä, miltä olet niiden tarkoittanut näyttävän. Tämän lisäksi suunnittelijan tulisi luottaa omaan katseeseensa ja hahmotuskykyynsä lisätessään visuaalisia osia. Tarkan matemaattisilla mittaamisilla menettelyn sijaan, käytä mittauksen apuna myös omaa harkintaa ja hahmotuskykyä. Jos juuri lisätty osa on matemaattisilta ominaisuuksiltaan täysin oikean kokoinen ja värinen, sekä oikeassa paikassa, mutta silti se ei näytä silmääsi oikealta, harkitse uudelleen. Jos kaksi osaa on luvuiltansa linjassa, mutta muotojensa takia näyttävät olevan vinossa, onko luvuilla silloin oikeasti väliä? Sen sijaan, että käyttäjä ymmärtäisi tämän silmissä tapahtuvan hämäyksen, hän suurella todennäköisyydellä vain olettaa osien olevan huonosti aseteltuna. Vakuuta käyttäjä sillä, mitä käyttäjä näkee, älä

numeroilla. Esimerkkinä tästä toimivat ympyrä ja neliö, jotka ovat kooltansa täysin samat. Niiden eri muodon vuoksi, ne näyttävät kuitenkin eri kokoisilta silmään. Myös kaksi kohdetta, jotka tulisi olla linjattuna, eivät aina välttämättä näytä siltä, vaikka matematiikka toisin väittää. Tällaisessakin tilanteessa suunnittelijan tulisi luottaa omaan hahmotuskykyynsä vakuuttaakseen yleisö, ei todellisiin mittoihin. (Samara, 2014)

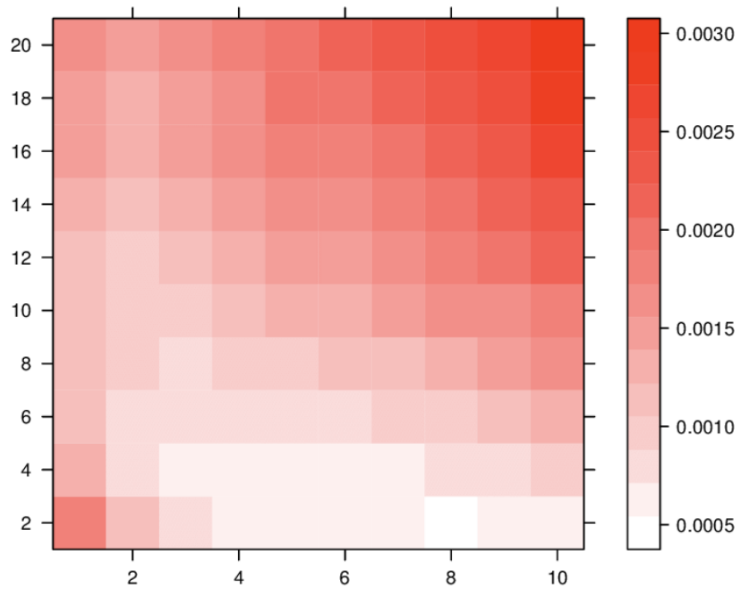
4. Numeerisen tiedon visualisointi

Numeerista tietoa voidaan visualisoida erilaisilla kaavioilla. Numeerisella tiedolla tarkoitetaan tietoa, joka kuvastaa määriä. Tähän lukuun valittiin neljä yleistä visuaalista esitystapaa numeeriselle tiedolle. Numeerista tietoa visualisoidaan, sillä se auttaa paljastamaan tiedosta niissä esiintyvät merkitykset, sekä auttaa lukijaa hahmottamaan mitä tiedolla yritetään kertoa.

4.1 Lämpökartta

Yksi yleisimmistä numeerisen tiedon visualisointitekniikoista on lämpökartta (heatmap) Lämpökartat ovat tärkeitä visualisoinnin tekniikoita, sillä niillä pystytään purkamaan laajastakin datajoukosta tiedot näkyviin suhteellisen yksinkertaiseen muotoon (kuva 12). Esimerkiksi genomitietojen tutkimisessa lämpökartat ovat yleinen menetelmä. (Gu et al., 2016)

Lämpökartassa datan graafinen esitys perustuu värien ja sävyjen käyttöön. Jokainen arvo sisältyy matriisiin ja sille määritellään värin sävy valittujen äärivärien välistä sen perusteella, minkä arvon se saa. Tyypillisesti mitä vahvemmaxi väri menee, sitä suurempi arvo on. Näillä matriisin arvoilla ja väreillä luodaan kartalle visuaalinen esitystapa. (Shahid, 2014) Lämpökartat ovat erittäin suosittu menetelmä. Vuonna 2012 tehtiin tutkimus viidestä johtavasta aikakauslehdessä, jossa tutkittiin lämpökarttojen suosiota. Keskimäärin lähes kolmasosa (202/664) julkaisuista tutkimuksista käytti lämpökarttoja visualisoidakseen dataa. (Deng et al., 2014)



Kuva 12 lämpökartta (Researchgate³)

4.2 Viivadiagrammi

Jos tieto on sen tyyppistä, että sille pystytään luomaan sen havaintoarvojen perusteella tietty järjestys, sopii sen visualisointitekniikaksi viivadiagrammi (line graph). Viivadiagrammeissa (line graph) havaintoarvot merkitään diagrammille pisteinä, ja nämä havaintoarvot yhdistetään toisiinsa viivojen avulla, jolla saadaan luotua arvoille tietty järjestys (kuva 13). Tämä arvojen järjestys helpottaa graafin luettavuutta ja ymmärrettävyyttä, joka tekee viivadiagrammista hyvän valinnan, kunhan arvojoukko on oikeata muotoa. Hyvänä esimerkkinä viivadiagrammin oikeaoppisesta käytöstä on lääkkeen vaikutuksen kuvaaminen verrattuna lääkepitoisuuteen. Päinvastaisena esimerkkinä toimii sukupuolijakauma, jolla taas ei minkäänlaista järjestystä ole. (Krause, 2012)

³ https://www.researchgate.net/figure/Heatmap-for-the-polynomial-mixing-rate-example-n-200_fig3_320296663

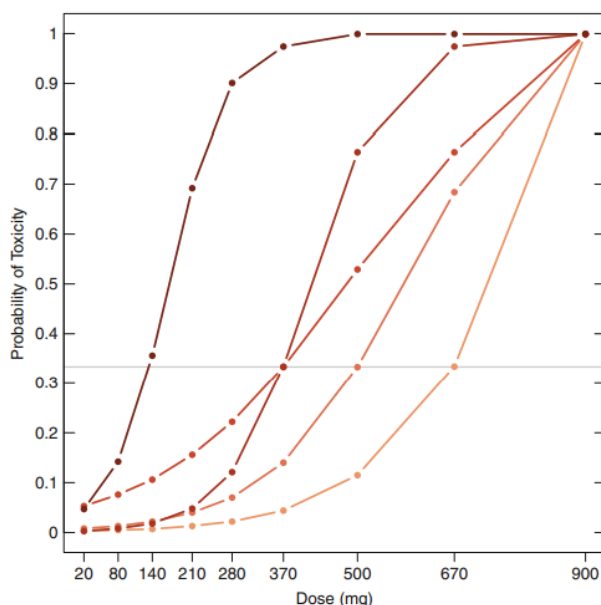


Fig. 9.1 Line plot illustrating simple dose-toxicity profiles for a new agent based on the logistic function $p(\text{dose}) = 1/(1 + \exp(-\alpha - \beta \times \text{dose}))$ with varying model parameters α and β

Kuva 13 Viivadiagrammi (Krause, 2012)

4.3 Pylväsdiagrammi

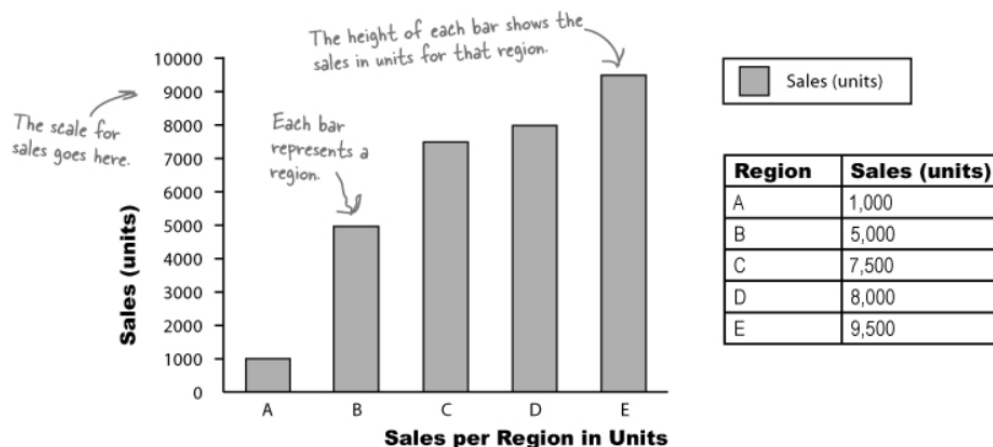
Pylväsdiagrammissa (bar chart) data esitetään kategorioittain pylväillä. Pylvään pituus määrittelee kategorian arvon lukeman. Jo pelkästään vilkaisemalla pystyy arvioimaan arvojen eroavaisuuksia, sillä mitä pidempi pylväs on suhteessa muihin, sitä suurempi sen arvo on. Selkeyden vuoksi pylväiden leveys pidetään tasaisena säilyttääkseen niiden vertailuarvon keskenään. (Griffiths, 2009)

Myös ympyrädiagrammeilla pystyisi vertailemaan havaintoarvoja, mutta pylväsdiagrammien etuna on korkeampi tarkkuus. Pylväsdiagrammit ovat siis parhaita tilanteisiin, joissa havaintoarvot ovat lähellä toisiaan, sillä pienet eroavaisuudet on helppo visualisoida niillä. (Griffiths, 2009)

Kuten aikaisemmin esiteltiin, visualisoinneista ja graafeista voidaan käyttää viittausta ”pieni havaintoarvojen määrä suhteessa pikseleihin”. Tällä tarkoitetaan tilannetta, jossa tietoa on huomattavan vähän, kun taas grafiikkaa on paljon. Esimerkkinä tällaisesta toimii pylväsdiagrammit. Yhtä arvoa kohden piirretään alue, joka vie suuren tilan näytöltä. Tämän takia tietojoukon ollessa todella pieni, pylväsdiagrammin käyttöä saatetaan pitää huonompana keinona kuin taulukon käyttöä. (Krause, 2012)

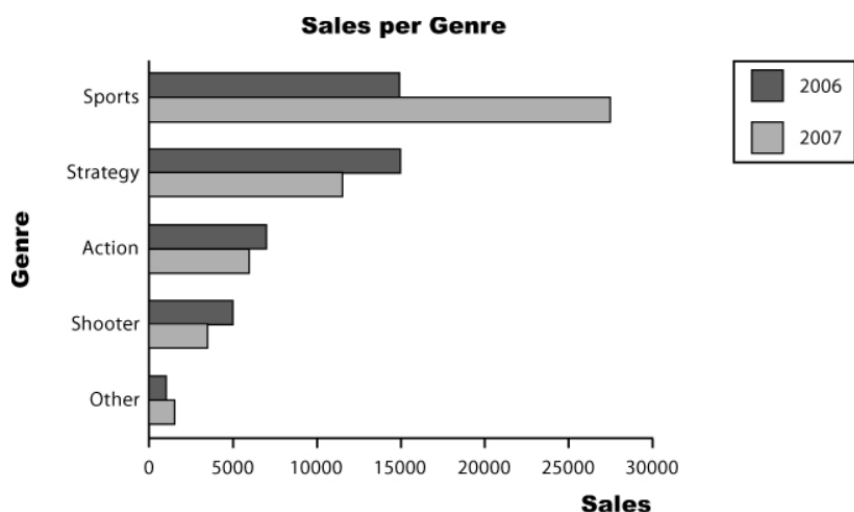
Pylväsdiagrammit voidaan jakaa kahteen eri luokkaan. Pystysuuntaiset pylväsdiagrammeissa pylväiden suunta on vertikaalinen. Niissä pylväiden leveys pysyy samana ja korkeudella määritellään kategorian arvo. Kategoriat esitetään horisontaalisella akselilla,

ja arvojen frekvenssit vertikaalisella. Kuvassa 14 näkyy esimerkki eri yksiköiden myyntitulosten visualisoinnista pystysuuntaisella pylväsdiagrammilla. (Griffiths, 2009)



Kuva 14 Pystysuuntainen pylväsdiagrammi (Griffiths, 2009)

Toinen mahdollinen tapa määrittellä pylväsdiagrammit ovat vaakasuuntaiset diagrammit. Ne ovat muuten täysin vastaavia pystysuuntaisten kanssa, mutta arvot määritellään pylväiden leveydellä korkeuden sijaan sekä akseleiden sijainti on vaihdettu keskenään. Toisin sanoen kategoriat ovat vertikaalisella asteikolla, ja arvot horisontaalisella. Kuvan 15 esimerkissä olevalta vaakasuuntaiselta pylväsdiagrammilta pystyy erottamaan nopeasti, mitkä kategoriat ovat saaneet suurimmat arvot. (Griffiths, 2009)

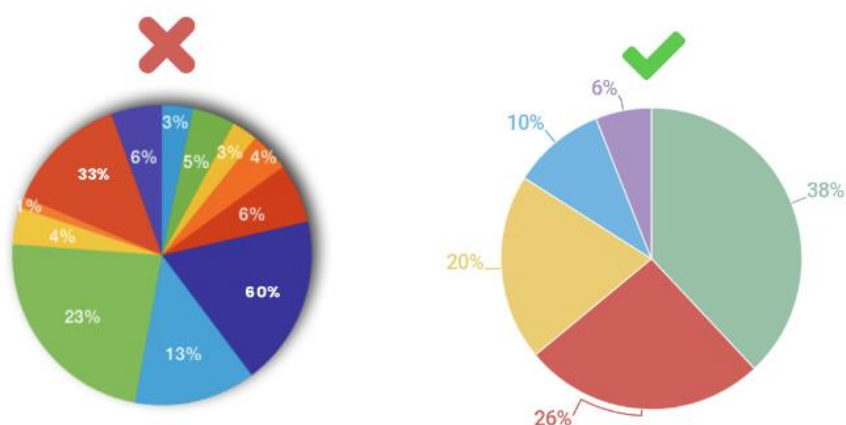


Kuva 15 Vaakasuuntainen pylväsdiagrammi (Griffiths, 2009)

4.4 Ympyrädiagrammi

Aikaisemmin mainitussa ympyrädiagrammissa (tunnetaan myös nimillä piirakka- ja sektorikaavio) data jaetaan pylväsdigrammin lailla erillisiin kategorioihin tai ryhmiin (kuva 16). Nimensä mukaisesti kategoriat sijoitetaan niiden suhteellisen koon mukaisesti ympyrän sisään. Kategorioiden muoto on auramainen. Kategorioiden koko määräytyy niiden suhteellisesta suuruudesta verrattuna muihin kategorioihin. Kategorioiden yhteenlasketun frekvenssin tulisi olla 100 %. Frekvenssillä ympyrädiagrammeissa tarkoitetaan kategorian suhteellista prosentuaalista osuutta. Tämän ansiosta arvojen vertailu on ympyrädiagrammeissa helppoa. Jo pelkällä vilkaisulla usein pystyy näkemään arvojen suhteelliset eroavaisuudet. (Griffiths, 2009)

Ympyrädiagrammi on muista kaavioista poikkeava siten, että siinä ei ole laisinkaan akseleita. Tyypillisten X- ja Y-akselien avulla tapahtuvan esityksen sijaan ympyrädiagrammi jakaa datan osiksi datan osuuksien, usein prosentuaalisesti, mukaan ympyrän sisällä. Kaavion ulkonäön takia sitä kutsutaan usein myös piirakkakaavioksi. Ympyrädiagrammit sopivat tilanteisiin, jossa havaintoarvot halutaan esittää suhteellisesti toisiinsa verrattuna, tai prosentteina. (Shahid, 2014) Arvojen tulisi kuitenkin olla toisistaan eroavat, sillä tarkempiin vertailuihin pylväsdigrammi sopii paremmin. Ympyrädiagrammeissa tämä toisistaan erottaminen menee pienien eroavaisuuksien kohdalla vaikeaksi. (Griffiths, 2009) Tyypillisesti prosenttiyksiköt esitetään myös tekstimuodossa oman kategoriansa kohdalla. (Shahid, 2014)



Kuva 16 esimerkki oikeaoppisesta ympyrädiagrammista (Infogram ⁴)

⁴ <https://infogram.com/blog/do-this-not-that-pie-charts/>

5. Ohjelmistoprojektin visualisointi

Myös ohjelmistoprojektien sujuvuuden kannalta on oleellista, että tärkeitä tietoja ei jää piiloon isojen datamäärien taakse. Lisäksi sujuvan kommunikoinnin onnistumiseksi on hyvä olla käytössä erilaisia menetelmiä, joilla sekä tiimin, että sidosryhmien välistä kommunikointia projektista voidaan helpottaa. Tähän lukuun on valittu tekniikoita, joilla projektin suunnittelua, aikataulutusta, etenemistä, sekä onnistumista voidaan seurata ja mitata.

Japanissa visualisoinnilla tarkoitetaan asioiden esittämisen visuaalisessa, helposti ymmärrettävässä muodossa, lisäksi myös läpinäkyvyyttä ja tiedon jakamista työntekijöiden ja sidosryhmien välillä. Näiden avulla pyritään kehittämään organisaation toimivuutta ja tehokkuutta. Termillä voidaan myös viitata prosesseihin, joissa käytetään visualisointien tuloksia ja tehdään niiden pohjalta toimenpiteitä. (Hammarberg, 2014)

5.1 Kanban-taulu

Kanban-tyylisessä tehtävien visualisoinnissa käytetään joko valkotaulua tai tyhjää seinää, jolle asetetaan kortteja, esimerkiksi muistilappuja, visualisoimaan työtehtäviä ja niiden kehitystä. Tästä tulee yleisesti käytetty nimitys Kanban-taulu. Kanban-taulu toimii erinomaisesti työnkulun ja informaation jakamiseen. Tärkeimpiä tietoja ovat prioriteetit, työskentelykohteet ja eteneminen työtehtävissä. Visuaalisella esitystavalla nämä saadaan erinomaisesti esille. Kanban-taulun avulla voidaan myös saada esille uusia kehitysideoita, sillä monesti kaikkia mahdollisuuksia ei näe ilman visuaalista esitystapaa. (Hammarberg, 2014)

Kanban-taulua voidaan käyttää joko fyysisenä tauluna, tai sähköisenä sovelluksen muodossa. Molemmat ovat suuressa käytössä, sekä molemmilla on omia positiivisia sekä negatiivisia puolia. Vielä ei ole tultu tulokseen, että toinen olisi parempi ja juuri sitä tulisi käyttää, joten tiimin tulee itse selvittää, kumpi sopii heidän tilanteeseensa paremmin. (Hammarberg, 2014)

Kanban-taululla myös työtehtävien kulkua on tärkeä seurata. Yleisin tapa tähän on luoda kaikille työtehtäville omat kortit. Korttina voi toimia esimerkiksi arkistokortti tai muistilappu. Näillä korteilla edustetaan tehtäviä, joka on joko työn alla tai se tulee tehdä. Fyysisissä korteissa on oleellista, että niitä on helppo liikuttaa ympäriinsä, sillä kortteja tulee pystyä järjestelemään taululla. Nämä kortit ovat yksinkertainen ja toimiva tapa

saada näkyviin nykyinen työtilanne. Niiden avulla voidaan havaita kehitystä, pullonkauloja ja jonoja työkulussa. Ne ovat tehokas keino saada nämä jatkuvasti esille työttömille. (Hammarberg, 2014)

Kanban-työkalun luominen on yksinkertainen prosessi. Tiivistettynä siinä lisätään työkalulle sarakkeet, lisätään niihin työtehtävät oikeille kohdille ja liikutellaan työtehtäviä tapahtumien mukaan, usein tilauksesta loppuun asti, ja asetetaan ne taululle tai seinälle. Kuvassa 17 on esitetty yksinkertainen Kanban-työkalu. (Hammarberg, 2014)

Backlog	Ready (4)	Analyse (2)		Develop (3)		Test (2)	
		Doing	Done	Doing	Done	Doing	Done

Kuva 17 Esimerkki Kanban-työkalusta (Damij & Damij, 2021)

Kanban-työkaluissa on myös työtapojen ja työkalun kannalta etuja. Kun visualisoidaan työkalu, luodaan samassa myös tiettyjä yhteisiä käytänteitä työnteon ympärille. Esimerkkinä toimii ominaisuuspyynnön elinkaari pyynnöstä valmistumiseen. Kaikilla tiimin jäsenillä saattaa olla oma ajatus siitä, miten tämä elinkaari toimii, mutta jos se esitetään tarkasti Kanban-työkalun avulla kaikille nähtäväksi, silloin oletukset poistuvat ja kaikki pääsevät kiinni samoihin työtapoihin. Tässä tilanteessa myös kaikki epäselkeydet tulevat esille, ja tarvittaessa tiimi voi keskustella ja kehittää toimintamallejansa jatkoa varten. (Hammarberg, 2014)

Jotta tiedoista on hyötyä, tulee sen olla ajan tasalla. Jotta tähän päästään, tulee työkalun sisältämiä tehtäviä jatkuvasti päivittää. Yleinen tapa tähän on käyttää päivittäisiä, lyhyitä tapaamisia, jossa tiimin jäsenet käyvät läpi mitä työkalulla on tapahtunut edellisen tapaamisen jälkeen. Nämä tapaamiset auttavat tiimiä koordinoimaan yhdessä, jakamaan tietoa, sekä oppimaan toisiltaan. (Hammarberg, 2014)

Kanban työkalun käytöstä tulevista hyödyistä suurimmat ovat ohjelmistojen kannalta nopeutunut toimitusaika sekä niiden laadun parantuminen. Kokonaisuudessa projektille koituva hyöty sisältää myös parantuneen kommunikation ja koordinaation, ominaisuuk-

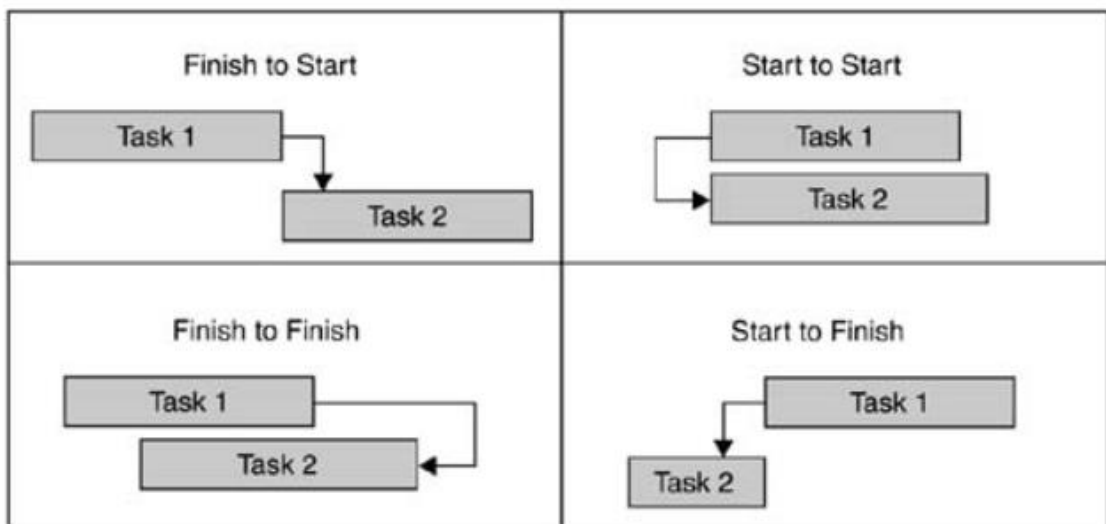
sien tasainen ja johdonmukaistunut toimitus sekä asiakkaiden ilmoittamien vikojen vähentyminen. Nämä hyödyt riippuvat vahvasti siitä, kuinka tasaisesti taululla korttien kulku etenee. (Damij & Damij, 2021)

5.2 Tehtävien väliset suhteet

Tehtävien suorittamisen ei tarvitse aina tapahtua peräkkäin. Jos projektista löytyy tarpeeksi henkilöstöä, tehtäviä pystyy suorittamaan myös rinnakkain. Tärkein vaihe suunnittelussa on selvittää aluksi tehtävien väliset suhteet. Siinä tulee selvittää mitä tehtäviä pystyy tehdä rinnakkain ja niiden perusteella luoda kriittinen polku (critical path). Useilla projektinsuunnittelutyökaluilla on käytössä neljää erityyppisiä tehtävien välisiä suhteita (kuva 18). (Moss & Atre, 2003)

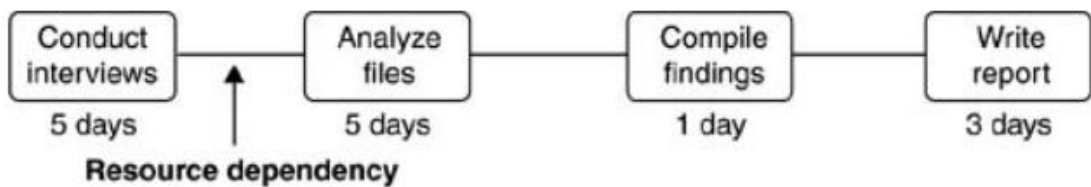
1. Lopusta alkuun suhde tarkoittaa, että seuraavaa tehtävää ei voi aloittaa, ennen kuin edellinen tehtävä on tehty loppuun.
2. Alusta alkuun suhteessa tehtävät voidaan aloittaa samaan aikaan.
3. Lopusta loppuun suhteessa tehtävä voidaan lopettaa vasta kuin toinen tehtävä loppuu myös eli tehtävää 2 ei voida lopettaa ennen kuin tehtävä 1 on myös lopetettu.
4. Alusta loppuun suhteessa tehtävää 2 ei voida lopettaa, ennen kuin tehtävä 1 alkaa.

Näistä yleisin lopusta alkuun ja alusta alkuun, ja alusta loppuun on harvinaisin. Kun suhteet on selvitty, tehtävien tulee tehtävä allokoida tiimin jäsenille. (Moss & Atre, 2003)

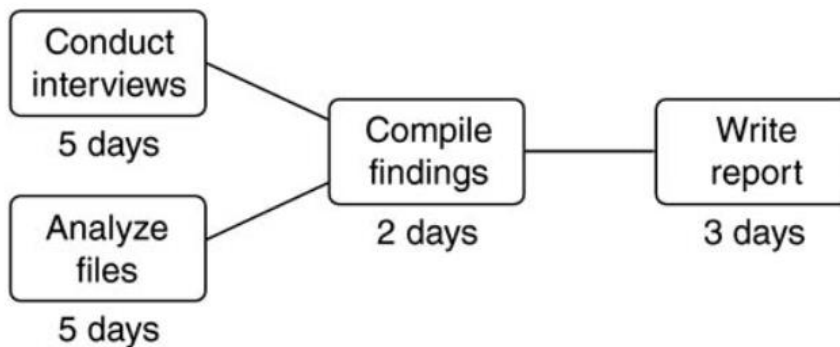


Kuva 18 Yleisimmät tehtävien väliset suhteet (Moss & Atre, 2003)

Myös projektissa käytössä olevilla resursseilla on vaikutuksia tehtävien suhteisiin ja niiden valmistumiseen. Resurssienhallinta on erityisen tärkeää, sillä mitä useampia tehtäviä voidaan tehdä samaa aikaan, sitä nopeammin projekti etenee. Jos henkilöstöä ei ole riittävästi, ei tehtävien toisistaan riippumattomuus usein auta. Vaikka tehtävät pystyttäisiin suorittamaan samanaikaisesti riippuvuuksien puolesta, tekijöiden puuttuminen aiheuttaa sen, että ne ovat silti suoritettava yksi toisensa jälkeen. Kuvassa 19 ja 20 esitellään yhden ja kahden työntekijän projektin eroavaisuus tehtävien suorittamiseen vaaditussa ajassa. Kuvista huomataan, että samojen tehtävien suorittamiseen vajaalla työntekijämäärällä (kuva 19) kestää 14 päivää, kun taas kahdella kestää verrattuna 10 päivää (kuva 20). Vertailussa otettiin myös huomioon se, että löytöjen yhteenvedossa kestää enemmän työntekijämäärän kasvaessa. (Moss & Atre, 2003)



Kuva 19 Työhön kuluva aika resurssivajeen kanssa (Moss & Atre, 2003)



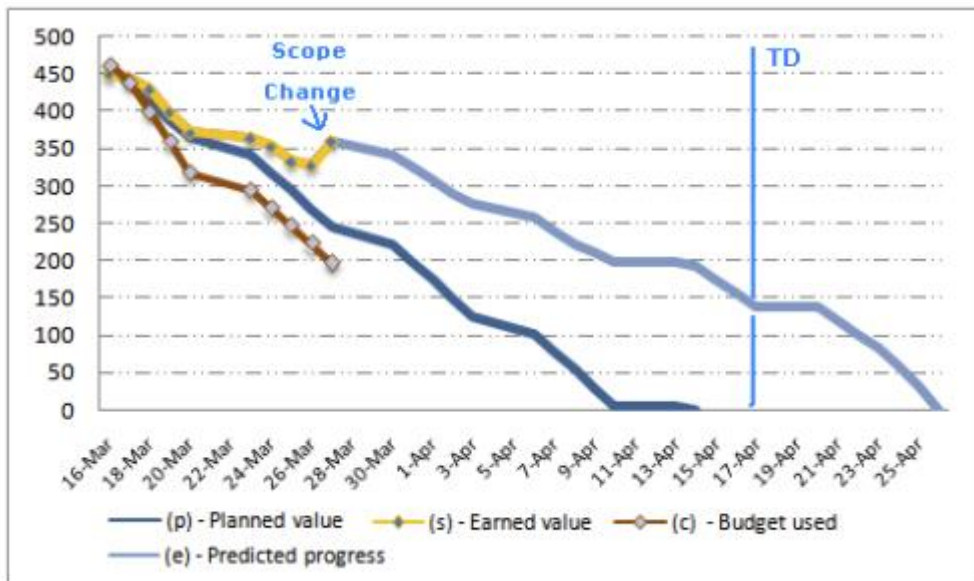
Kuva 20 Työhön kuluva aika ilman resurssipulaa (Moss & Atre, 2003)

5.3 Projektin etenemisen seuranta ja sen visualisointi

Kaikilla projekteilla on tietyt vaatimukset ja velvoitteet, joiden mukaan niiden tulee toimia. Yleisimmät näistä vaatimuksista liittyvät projektin laajuuteen, ajankäyttöön tai budjettiin. Nykyään oleellisina vaatimuksina ovat myös projektin laatu ja riskienhallinta. Projektille tärkeää on myös vastata sille etukäteen asetettuihin kriteereihin, sekä tiimin tulee pysyä vastuullisena ja johtoryhmän on oltava jatkuvasti ajan tasalla projektin tilanteesta. Projektinhallinnassa suorituskyvyn hallinnan työtehtävänä pitää huolta siitä, että näihin tarpeisiin vastataan. Projektin tulisi pystyä jatkuvasti tarjoamaan vastauksia kysymyksiin projektin sen hetkisestä tilanteesta ja etenemisestä. Tämän vastauksen tulisi ideaalissa tilanteessa pohjautua suorituskykymalleihin, jotka sisältävät mittauksia ja indikaattoreita projektin velvoitteista ja vaatimuksista. (Vajda, 2009)

Edistymiskaaviot (burn chart) sopivat edellä mainittuihin mittauksiin hyvin. Näitä kaavioita on kahta eri tyyppiä. Ensimmäinen niistä on nouseva edistymiskaavio (burnup chart), jossa edistymistä seurataan kasvavilla viivoilla tehdyn työn perusteella. Toisessa, laskevassa edistymiskaaviossa (burndown chart), seurataan jäljellä olevaa työtä. Tässä kaaviossa viivat laskevat ja sillä edustetaan jäljellä olevaa työtä. (Vajda, 2009)

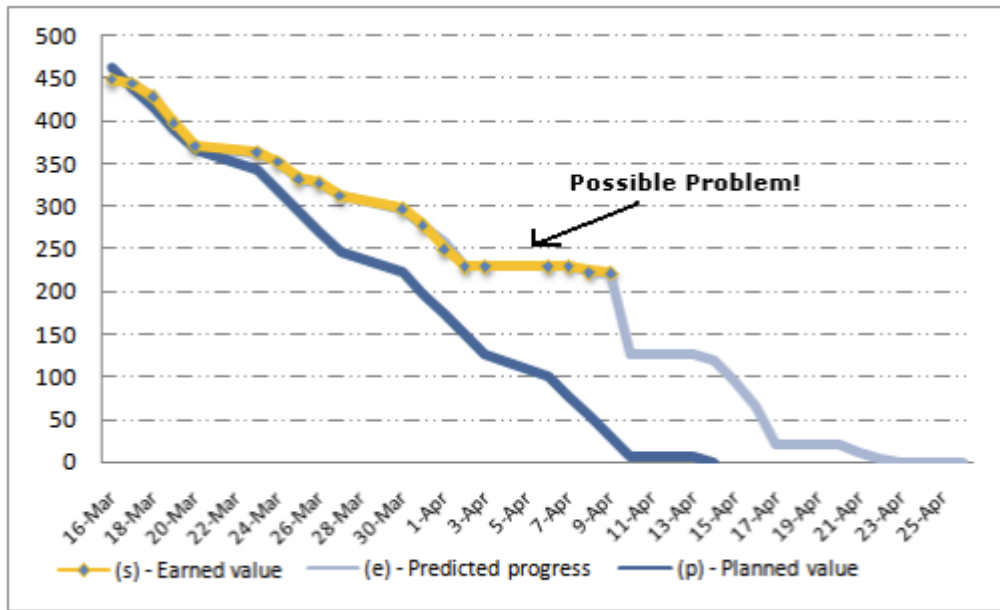
Laskevan edistymiskaavion käyttämiseksi tulee ensiksi projektin tavoitteet, laajuus ja rajoitteet olla selvillä. Jos projekti on liian laaja, se voidaan myös jakaa omiin aliprojekteihin. Näiden jälkeen tulisi käydä läpi, mitä projektissa tulee tehdä, jotta se saavuttaa tavoitteensa ja listataan eri tehtävät, jotka tulee tätä varten suorittaa. Kun tehtävät on tunnistettu, niihin vaaditut resurssien määrän voidaan arvioida, ja ne voidaan jakaa tiimin jäsenten kesken. Kuvan 21 esimerkissä voidaan nähdä, kuinka esimerkiksi projektin laajuuden muuttumisen näkee selkeästi palamiskaaviosta. (Vajda, 2009)



Kuva 21 Projektin laajuuden muuttuminen (Vajda, 2009)

Kun pohjatyö on tehty, voidaan aloittaa tuloksen ja kehityksen mittaaminen. Jokaisen tehtävän edistymistä tulisi seurata säännöllisesti ja jatkuvasti tehdä uudelleenarvioita työn etenemisestä ja vaatimuksista. Jos muutoksia tulee, tulee kaaviot päivittää. Jatkuva projektin keston seuraaminen on myös oleellista. Kun näitä seurataan, voidaan jatkuvasti tehdä korjausliikkeitä, jos huomataan, että tavoitteisiin ei olla pääsemässä. (Vajda, 2009)

Edistymiskaavioiden viivoja seuraamalla pystytään myös havaitsemaan projektissa mahdollisten ongelmien esiintymistä (kuva 22). Jos viivat laskevat hitaasti tai liikkuvat lähes vaakasuunnassa, silloin on mahdollista, että projektissa on ongelmia jollain osalla. Tyypillisiä ongelmia ovat tehtävien aliarviointi, vaatimusten epäselkeys tai toisiaan estävät tehtävät (blocking issues). Edistymiskaavion ansioista tällaiset ongelmat havaitaan nopeasti, ja tiimi pystyy reagoimaan niihin. (Vajda, 2009)



Kuva 22 Ongelmien esiintyminen editysmiskaaviossa (Vajda, 2009)

Laskevissa editysmiskaavioissa etuna myös tavoitteiden luoma tunteisiin vetoava vaikutus. Niiden avulla saadaan luotua tiimille tehtävä, jossa tavoite on päästä alas numeroon nolla asti. Tämä auttaa tekijöitä saamaan positiivisia tunteita onnistumisista ja työn etenemisestä. Tässä mallissa etuna on myös se, että useimmat laskelmat ja indikaattorit ovat samassa yksikössä, jolloin ne voidaan esittää saman kaavion sisällä. Tämän avulla yhdellä kaaviolla pystytään tarjoamaan laajemman kuvan nykyisestä tilanteesta. (Vajda, 2009)

5.4 Ohjelmistoprojektin suunnitelma ja aikataulus.

Projektissa aikataulujen suunnittelu ja niiden kommunikoiminen sidosryhmille on usein tärkeää. Sen sulavoittamiseksi voidaan käyttää seuraavaksi esiteltäviä visualisointitekniikoita.

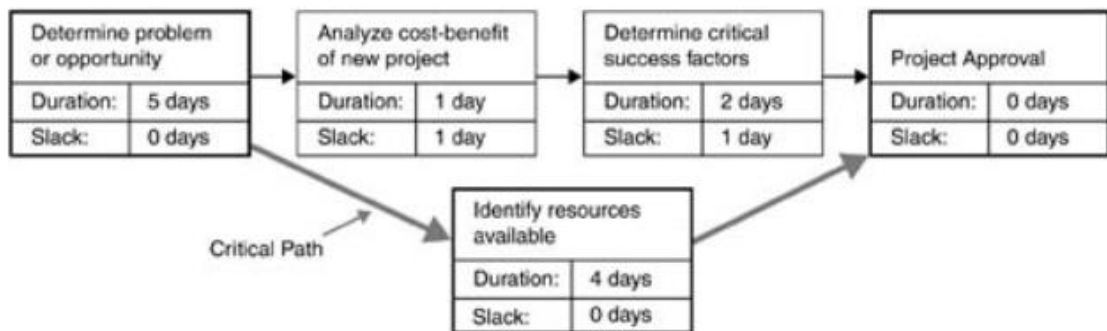
5.4.1 Kriittisen polun menetelmä

Tehtävien suhteiden tunnistamisen ja resurssien jakamisen jälkeen voidaan hyödyntää kriittisen polun menetelmää (critical path method). Tämän menetelmän avulla voidaan hahmotella tehtävien kestoja ja suorittamisjärjestystä. Tällä saadaan havainnollistettua nii-

den tehtävien viiveet, joilla on vaikutusta projektin kokonaiskeston. Tämän tiedon pohjalta voidaan jaotella resursseja uudelleen tai neuvotella uudelleen projektin vaatimuksia. (Moss & Atre, 2003)

Kriittisen polun menetelmän tarkoitus on avustaa projektin keston ja tuloksen suunnittelussa, arvioinnissa ja ohjauksessa. Menetelmä purkaa projektin pienempiin osiin, joiden kesto voidaan mitata. Tämän avulla pystytään tekemään ennustuksia koko projektin aikatauluista ja suunnitelmista. Tiivistettynä kriittisen polun menetelmä luo kaavion, josta voidaan seurata tehtävien kestoja ja niiden suorittamisen järjestystä. Tämä järjestys luo erilaisia polkuja, joista pisimpään kestävä kutsutaan kriittiseksi poluksi. Tämä johtuu siitä, että tämän kriittisen polun kokonaiskesto on suurempi kuin minkään muun tehtävien polun. Tällöin kaikki viiveet kriittisellä polulla vaikuttavat koko projektin aikatauluihin. (East, 2015)

Kuvan 23 esimerkissä esitetään, kuinka kriittisen polun viivästymisellä on vaikutus koko projektin kulkuun. Jos esimerkin kriittisen polun neljä päivää kestävä työ viivästyy päivällä, koko projektin valmistuminen siirtyy tämän verran eteenpäin. Jos taas samaa aikaa työn alla olevat yhteensä kolme päivää kestävä tehtävä venyvät päivällä, se ei vaikuta projektin valmistumiseen, sillä kriittisen polun alla oleva tehtävä kestää joka tapauksessa mainitun neljä päivää. (Moss & Atre, 2003)



Kuva 23 kriittinen polku (Moss & Atre, 2003)

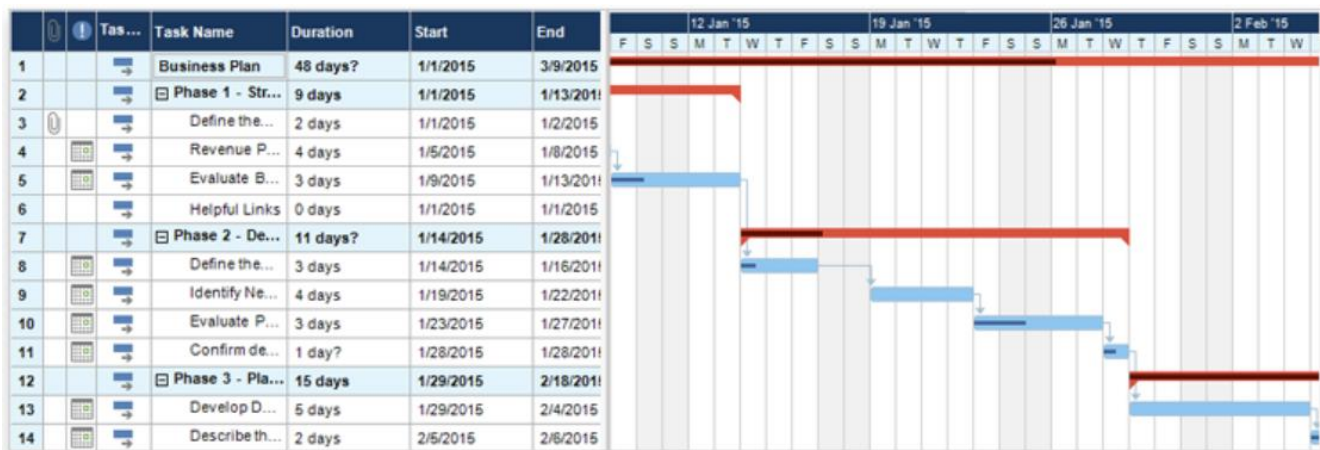
5.4.2 Gantt-kaavio (Gantt-Chart)

Projektin suunnitteleminen on aina ollut työläs osuus ja sen ylläpitäminen ja päivittäminen on ennen vaatinut paljon resursseja. Nykyään projektinhallintatyökalujen ansiosta

tästä prosessista on saatu kevyempi. Jotta tarjolla olevia työkaluja pystyttäisiin käyttämään tehokkaasti, tulee projektinhallinnastakin olla hyvä perusymmärrys. Kaavioiden luominen työkalujen avulla auttaa hallinnassa paljon, sillä kaikki päivitykset, mitä teet joko tehtäviin, resursseihin tai muihin arvioihin, päivittyvät automaattisesti työkalun avulla kaikkiin kaavioihin. Usein on silti suositeltua tarkistaa päivityksen jälkeen tulokset, ettei niihin ole tullut virheitä. (Moss & Atre, 2003)

Kun projektissa on selvitetty kaikki tehtävät ja niiden riippuvuudet, sekä tehtävien suorittamiseen vaadittavat resurssit ja aika-arviot, voidaan projektin aikataulua alkaa suunnittelemaan. Yleisin visualisoinnin tekniikka tähän on Gantt-kaavio. (Moss & Atre, 2003)

Gantt-kaaviolla kuvastetaan tehtäviä ja niiden kestoa suhteessa koko projektiin (kuva 24). Tehtävät asetetaan allekkain ja niille asetetaan niiden kestoa kuvaavat vaakasuuntaiset palkit, jotka kuvaavat aikaa, joka tehtävien suorittamiseen kuluu. Kaavion tärkein tehtävä on suunnitella projektin kokonaiskesto. Alkuperäisessä Gantt-kaaviossa tehtävien välisiä suhteita ei visualisoitu, mutta 90-luvulla alusta loppuun ja lopusta alkuun suhteita alettiin hyödyntämään kaaviossa. Tämän avulla voidaan seurata tehtävien viivästymisen vaikutusta koko projektin keston. (Sharon & Dori, 2017)



Kuva 24 Gantt-kaavio (Gantt⁵)

⁵ <https://www.gantt.com/>

6. Case VISDOM Roadmapper

Tässä kappaleessa esitellään, kuinka edellä mainittuja visualisoinnin tekniikoita on hyödynnetty VISDOM Roadmapper projektissa Vincit Oyj:llä. Tärkeimpänä tavoitteena oli tietorakenteiden suhteiden visualisointi käyttäjälle, sekä käytettävyyden yksinkertaistaminen ja helpottaminen visuaalisilla ominaisuuksilla.

6.1 VISDOM Roadmapper projekti

Roadmapper on osana laajempaa VISDOM projektia tuotettu työkalu, jonka tavoitteena on tuoda esille visuaalisesti ohjelmistoprojektin virstanpylväiden (milestone) ja tehtävien (task) etenemistä ja ominaisuuksia. Roadmapper on jaettu kolmeen pääosaan: metriikoihin, tehtävälistaan, sekä virstanpylväiden suunnitteluun. Sen käyttäjiä ovat projektissa kehittäjät, tiiminvetäjät, asiakkaat, sekä liiketoiminnan edustajat.

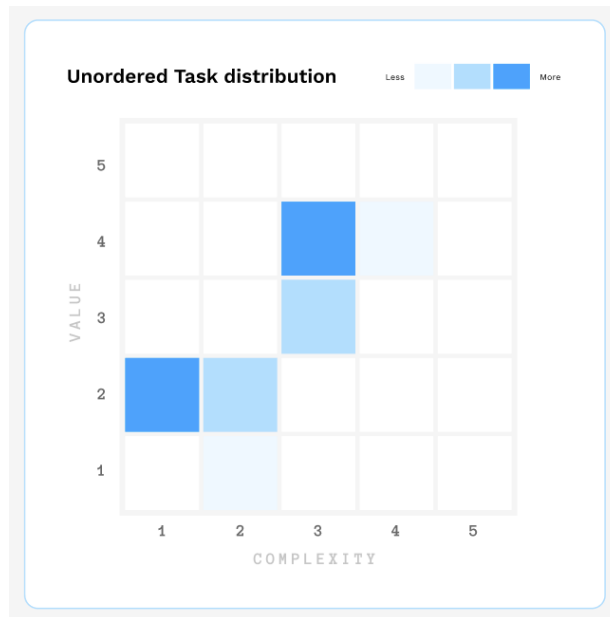
Etusivulla (dashboard) ovat ensimmäisenä projektiin liittyvät metriikat. Siinä kuvataan tehtävien arvoja lämpökartan avulla, projektin virstanpylväiden suunniteltuja arvoja, sekä projektin etenemistä. Nämä arvot ovat tyypillistä numeerista dataa, johon on valittu niitä kuvastavat graafit. Toisena tärkeänä sivuna on tehtävälista, joka jakautuu kahteen osaan. Toisessa osassa on tavallinen listarakenne, jossa tehtävät on luokiteltu allekkain. Tehtävät jaotellaan siinä tehtäviin, joille on annettu arvot, sekä niihin, joille ei ole. Sivun toinen osa sisältää tehtäväkartan, joka esitellään seuraavissa luvuissa.

Virstanpylväiden suunnittelun sivulla on listattuna projektin kaikki tehtävät ja virstanpylväät. Tehtäviä voidaan vapaasti liikutella virstanpylvästä toiseen, sekä virstanpylväitä voidaan luoda haluttu määrä. Tehtävien arvot lasketaan virstanpylväissä ja niiden avulla etusivun metriikat saavat sitten arvonsa.

6.2 Tehtävien haasteen ja arvon esittäminen lämpökartalla

Luokittelimme tehtävät niiden tuottaman arvon, sekä niiden toteuttamiseen vaaditun ajan mukaan. Tehtävien arvot oli luokiteltu seuraavasti: kompleksisuus asteikoilla 1–5 ja tehtävän tuottava arvo asteikolla 1–5. Tämän tyyppisen numeerisen tiedon visualisoimiseksi sopi lämpökartta, sillä sen avulla pystytään nopeasti arvioimaan, minkä tyyppisiä tehtäviä projektissa on eniten vielä valitsematta. Jos projektissa on paljon tehtäviä, joilla on korkea arvo, mutta matala kompleksisuus, voi tämän perusteella tulkita, että asiakkaalle olisi mahdollista saada paljonkin arvoa vähäisellä työllä. Jos taas tilanne on päinvastainen, eli pienen arvon omaavia haastavia tehtäviä löytyy paljon, voi miettiä asiakkaan kanssa, että onko niiden tehtäviä tekeminen taloudellisesti kannattavaa.

Väreiksi valittiin lämpökartoille tyypillisesti yhden värin eri sävyjä, ja tässä tapauksessa valittu väri oli sinertävä. Värivalintaan päädyttiin Roadmapperin yleisen sinertävän teeman vuoksi. Asteikoksi päädyttiin valitsemaan 1–3, jotka nimettiin käyttäjäystävällisesti ”vähemmän” ja ”enemmän”. Jos arvo osuu näiden väliin, se tarkoittaa keskimääräistä. Kuvassa 25 esitellään työkaluun suunniteltu lämpökartta.



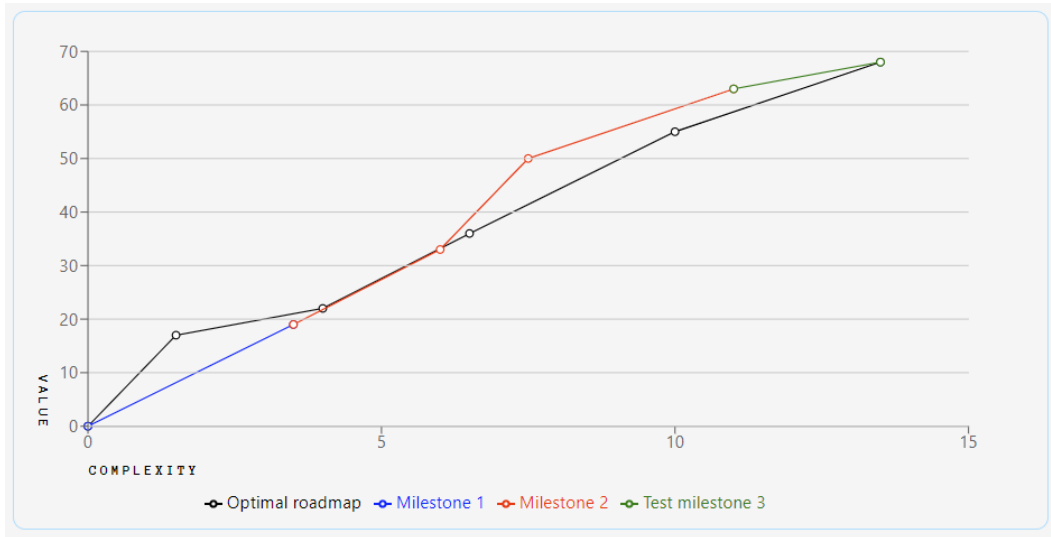
Kuva 24 Työkaluun suunniteltu lämpökartta

6.3 Suunnitelmien vertaaminen optimaaliseen arvoon

Yhtenä Roadmapperin toiminnoista on optimaalisen suunnitelman (roadmap) tekeminen projektille. Projekti jaetaan virstainpylväisiin, sekä näihin lisätään kyseisessä jaksossa työn alle otettavat tehtävät. Jokaisella tehtävällä on Roadmapperissa määriteltyä asiantuntijoiden toimesta sekä taloudellinen arvo, että työmäärän arvio. Näiden perusteella saadaan arvon ja kompleksisuuden suhde, jonka perusteella voidaan arvioida tehtävän tärkeys.

Roadmapper laskee ja piirtää automaattisesti optimaalisen suunnitelman projektille nousevaan etenemiskaavioon. Tätä kuvataan kuvassa (kuva 26) ”optimal roadmap” viivalla. Samalle asteikolle myös piirretään tämän hetken suunnitelman perusteella piirretty graafi. Sillä arvot ovat numeerisia, sekä niitä on kaksi (kompleksisuus ja arvo), sopi viivadiagrammi tilanteeseen. Tällä kaaviolla pyritään helpottamaan optimaalisen suunnitel-

man ja toteutetun suunnitelman poikkeamien havaitsemista ja tarvittaessa korjausliikkeiden tekemistä. Sillä pystytään myös välittömästi havaitsemaan, jos jokin virstanpylväs on muita selkeästi poikkeava arvoltaan tai haasteellisuudeltaan.



Kuva 25 Projektin virstanpylväiden suunnittelu

6.4 Tehtävien visuaalinen esitystapa

Sen sijaan, että tehtävät esitettäisiin vain listamuodossa, haluttiin tehtäville myös visuaalinen esitystapa. Sen tarkoituksena on helpottaa tehtävien ymmärtämistä, käsittelemistä, sekä projektin suunnittelua.

6.4.1 Vaatimukset tehtäväkartalle

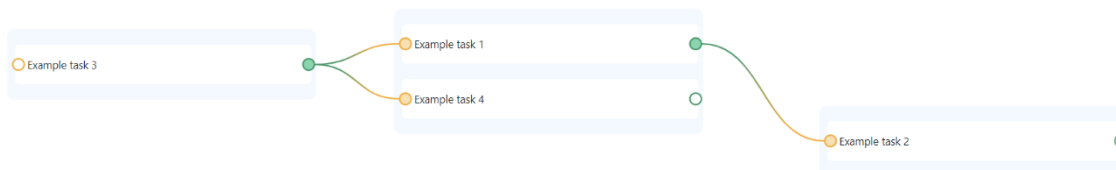
Tehtäväkartan tavoitteena on visualisoida käyttäjälle tehtävien väliset suhteet. Tällä tarkoitetaan sitä, että mitkä tehtävät ovat toistensa esivaatimuksia, mitkä liittyvät vahvasti toisiinsa ja mitkä tehtävät voidaan tehdä vasta kun edelliset ovat valmiita. Tämän avulla voidaan esimerkiksi sprinttejä tai virstanpylväitä suunnitella ottaen huomioon tehtävien tekemisen järjestys.

Alun perin tehtäväkartan vaatimukset näyttivät yksinkertaiselta. Toteutuksen yhteydessä huomattiinkin, että visualisoinnin selkeyttämiseksi tarvitaankin monia muitakin pieniä visuaalisia esityksiä, kuten värien käyttöä, kursoria sekä opasteita. Tehtäväkartasta piti myös onnistua tekemään käyttäjälle selkeä, etteivät oleelliset ominaisuudet jäisivät huomaamatta.

6.4.2 Tietojen väliset suhteet

Erillisiä datapisteitä ja tehtäviä oli helppo visualisoida esimerkiksi käyttämällä suoraviivaisesti sanallista taulukkolistausta. Ongelmaksi tulikin taulukossa se, ettei sen välityksellä pystynyt selkeästi näkemään tietojen riippuvuuksia toisiinsa. Roadmapperissa tehtävillä on kolmenlaisia eri suhteita: edellytykset, synergiat ja vaatimukset. Jos tehtävällä on edellytyksiä, ne tulisi tehdä ennen kyseisen tehtävän aloittamista. Synergiassa olevat tehtävät taas liittyvät toisiinsa oleellisesti ja niiden tekemistä suositellaan yhdessä. Vaatimus-suhteella olevia tehtäviä ei voi suorittaa, ennen kyseisen tehtävän valmistumista.

Nämä vastaavat lähes aikaisemmin esiteltyjä lopusta alkuun, alusta alkuun ja alusta loppuun suhteita. Jos tehtäväkarttaa luetaan vasemmalta oikealle, silloin suhteet ovat muotoa lopusta alkuun. Isoimpana erona on se, että alusta alkuun suhteet tehtäväkartassa ovat suosituksia vaatimusten sijaan. Kuvassa 27 esitellään työkalussa käytössä oleva visuaalinen esitystapa tehtävien välisille suhteille.



Kuva 26 Tietojen välisen suhteiden visuaalinen esitystapa

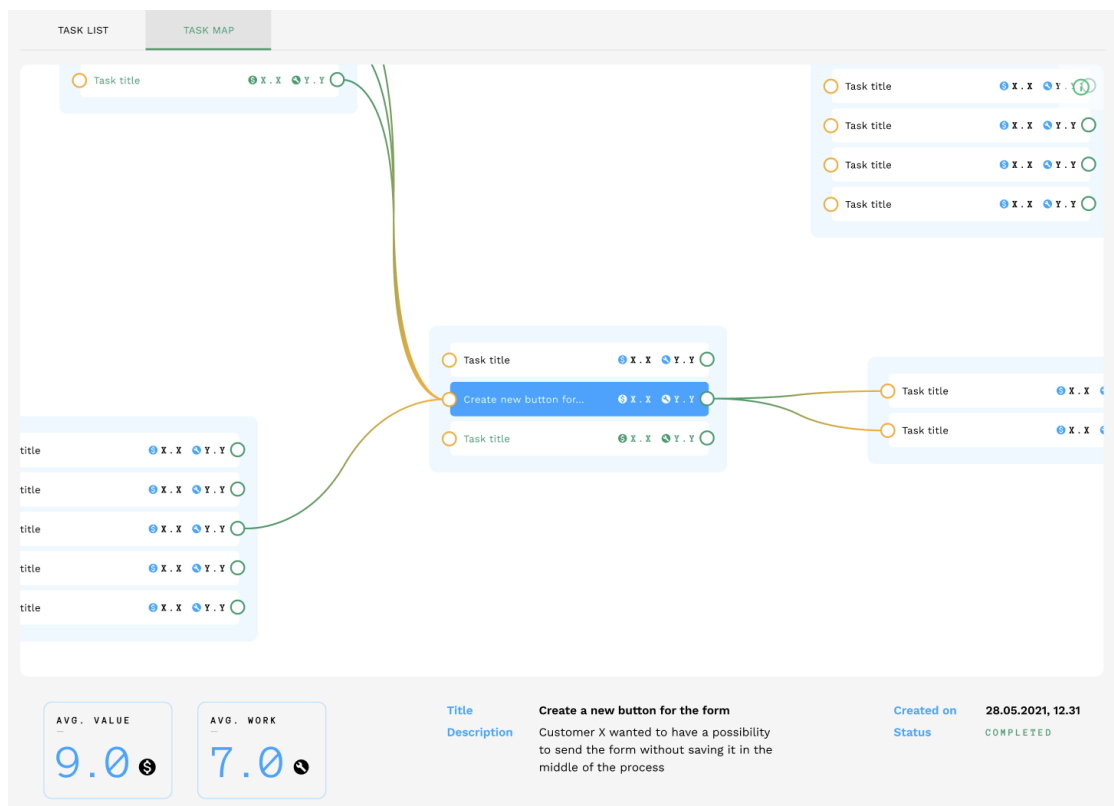
6.4.3 Tehtäväkartan suunnitelma

Tehtäväkartan alkuperäinen suunnittelu sisälsi kolme pääkohtaa: tehtäväryhmät, tehtävien väliset aikaisemmin esitellyt suhteet, sekä tehtävien lisätiedot. Tehtäväryhmät ovat sinisellä taustalla varustettuja nelikulmioita, joiden sisälle toisiinsa liittyvät tehtävät lisätään. Näillä ryhmillä viestitään käyttäjälle, että kyseiset tehtävät olisivat hyvä tehdä yhdessä. Nämä tehtävät eivät ole edellytyksiä eikä vaatimuksia toisillensa, mutta liittyvät niin vahvasti toisiinsa, että ne oli hyvä tehdä samalla kerralla. Tämä vastaa lähes aikaisemmin esitettyä alusta alkuun suhdetta. Erona tässä on se, että vaatimuksen sijaan, Roadmapperissa ne ovat suosituksia.

Tehtäväryhmien lisäksi tärkeänä visualisointina ovat tehtävien suhteet. Niillä kuvastetaan tehtävien toteuttamisen järjestystä. Järjestys on pyritty saamaan mahdollisimman

nopeasti ymmärrettäviksi, joten siinä seurataan länsimaissa tyypillistä vasemmalta oikealle suuntaa. Jos tehtävä on liitetty viivalla toiseen, ja alkuperäinen tehtävä on vasemmalla, tulee silloin tämä tehdä ennen oikealla puolella olevaa (kuva 28).

Viimeisenä selkeyttämisen keinona on käytetty tehtävien lisätietolomaketta. Siitä käyttäjä saa tarvittaessa näkyviin tehtävistä tietoja kuten tehtävän kuvaus, tila sekä sen luomispäivämäärä. Tämä on oleellista, sillä tehtävälaatikkoihin ei saisi mitenkään mahdumaan näitä kaikkia tietoja ilman, että näytölle tulisi aivan liian paljon asiaa kerralla, joka vaikeuttaisi sekä ymmärrettävyyttä, selkeyttä, että yleistä visuaalista ilmettä.



Kuva 27 Tehtäväkartan suunnitelma

6.4.4 Käyttäjätestaukset

Roadmapperissa järjestettiin tasaisesti käyttäjätestauksia. Niiden yhteydessä myös tehtäväkartta oli testauksen kohteena. Käyttäjätestauksissa valittiin asiantuntijoiden ryhmä toteuttamaan kuvitteellista skenaariota, jossa yritys käyttää projektinhallinnassaan tätä työkalua. Asiantuntijat jaettiin projektitiimeihin, sekä heille annettiin kaikille tietyt roolit. Osa testaajista oli kehittäjiä, osa asiakkaita, ja osa liiketoimintavastaavia. Heille annettiin tietty tehtävä, joka heidän piti tiiminä suorittaa neljän tunnin aikana. Käyttäjätestauksiin

osallistui koko Roadmapperin kehitystiimi, sekä muita osapuolia. Käyttäjien toimia seurattiin ja analysoitiin ja niiden perusteella tehtiin kehitysehdotuksia. Nämä tilaisuudet myös videoitiin tarkempaa analysointia varten. Kun edellisessä testauksessa esille tulleet tarpeet oli huomioitu ja tarvittaessa lisätty, pidettiin aina uusi vastaava tilaisuus uudella skenaariolla.

Tehtäväkarttaa testatessa käyttäjien oli tarkoitus luoda olemassa oleville tehtäville riippuvuudet, joidenka avulla tehtäväkartasta pystyisi päättämään tehtävien suositellun tekojärjestyksen.

Käyttäjille ei ollut selkeää, että oikea klikkaus poistaa suhteen. Tämä ominaisuus jäi osalta kokonaan huomaamatta, ja osa taas onnistui sen avulla vahingossa poistamaan haluamatta suhteita. Tulimme tehtäväkartan suunnittelijoiden kanssa lopputulokseen, että oikea klikkaus ei ollut oikea ratkaisu tähän ja vaihdoimme sen avaamaan kontekstimenun, jossa on mahdollisuus suhteiden poistamiseen.

Myöskään tehtäväkartan ohjeistuksen avaamiseen tarkoitettu painike jäi usealta huomaamatta. Se oli alun perin liian pieni, sekä epäkäytännöllisessä paikassa. Päädyimme tämän jälkeen siirtämään kuvakkeen käyttäjän huomion kiinnittämiseksi oikeaan yläreunaan. Tämä tehtäväkartan ohjeistus (kuva 29) oli välttämätön osa tehtäväkarttaa, sillä ilman sitä, usea ominaisuus jäi käyttäjiltä huomaamatta.

Overview ^

TASK RELATIONS

Relations Tasks connected by a line

Synergies Tasks in a same group (light blue box)

VIEW


Task map Includes tasks with relations and synergies


Task circles Connect task to other tasks on the map

Unstaged tasks list Holds tasks without relations or synergies

Task info Shows selected task's info below the task map



SCREEN FUNCTIONS



 **Fit groups into the viewport** Shows the center of the map

 **Reset group positions** Removes saved group positions and generates a new layout

Actions ^

Start by drag and dropping tasks from Unstaged tasks list into the task map area. Task map has the following mouse actions:

 Click or Click & drag  Grab (a task)

 Move (a group)  Zoom in and out (scroll wheel)

ADDING AND REMOVING TASK RELATIONS & SYNERGIES

Add relation **Left-click on a task circle to show possible connections and hold left-click to drag a line** to another task circle. (Orange creates dependencies, green creates prerequisites)

Remove relation **Left or right click on the line** to open a context menu.

Add synergy **Drag and drop a task into** another light blue area.

Remove synergy **Remove synergy actions** the light blue area.

REORGANIZING THE VIEW AND VIEWING INFO

Move a task or group **Hold left-click on a task or group** to move it around the map.

Show task info **Left-click on a task** to open task info.

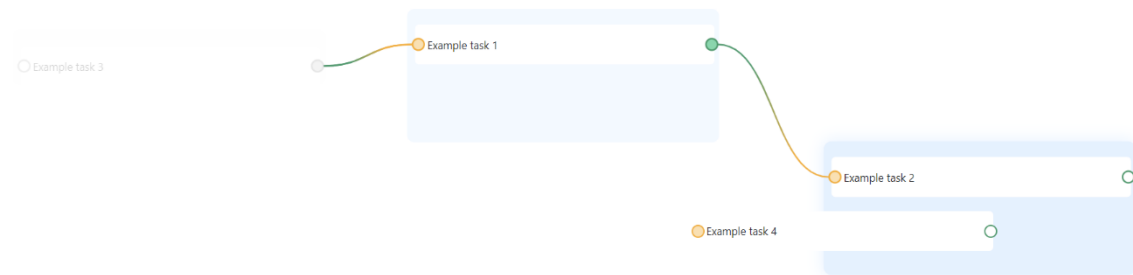
Kuva 29 Tehtäväkartan ohjeet

6.4.5 Raahaa ja pudota

Sen sijaan, että tehtäväkartta oli pelkästään kuvaus, jopolla visualisoidaan tehtävien suhteita, päätimme myös lisätä sinne käyttäjälle mahdollisuuksia muokata tehtävryhmien kokoonpanoja. Tämän ominaisuuden toteuttamiseksi lisäsimme raahaa ja pudota -ominaisuuden tehtäviin. Tämän avulla käyttäjä voi siirrellä tehtäviä ryhmistä toisiin, poistaa tehtäviä synergiaryhmistä tai jopa poistaa yksittäisiä tehtäviä näkymästä kokonaan. Tehtävien piilottaminen näkymästä liittyy vahvasti myös näkymän suorituskyvyn parantamiseen, johon perehdymme lisää kappaleessa 6.4.7.

Raahatessa tehtäviä ryhmien välillä, on käyttäjälle tärkeää visualisoida kaikki mahdolliset toiminnot (kuva 30). Tehtäväkartassa päädyimme visualisoimaan raahatessa kaksi käyttäjälle tärkeintä asiaa: kielletyt ja sallitut pudotusalueet. Kielletyt pudotusalueet visualisoidaan käyttäjälle harmaannuttamalla kiellettyjen alueiden tehtävien, sekä taustan väri. Sallitut pudotusalueet taas jätetään samanvärisiksi, kuin ne ovat normaalisti. Täten

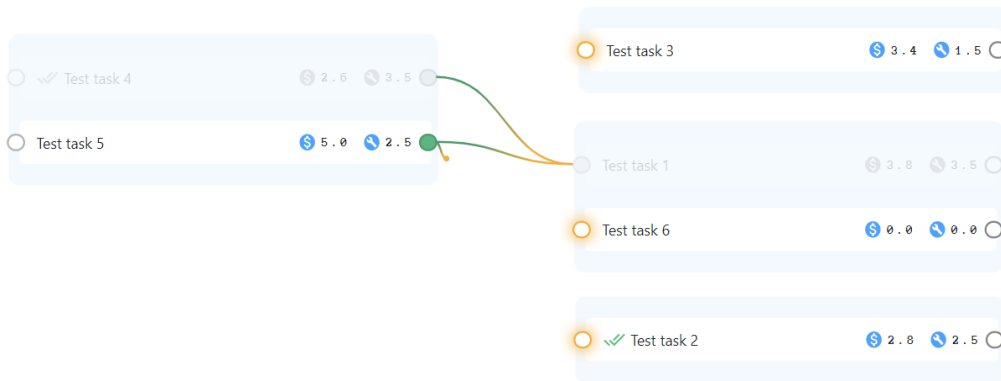
saadaan käyttäjälle heti näkyväksi hänelle oleelliset tehtäväryhmät. Sillä tehtävän pystyy myös pudottamaan tehtäväryhmien ulkopuolelle, joka erottaa tehtävän nykyisestä ryhmästään, on myös oleellista visualisoida käyttäjälle tilanne, jolloin tehtävä on pudotussa ryhmään. Tämä saatiin aikaiseksi tummentamalla ja laajentamalla tehtäväryhmää, johon tehtävä ollaan pudottamassa. Jos mikään tehtäväryhmä ei visualisoi tätä, silloin tehtävä pudotessaan irtoaa nykyisestä ryhmästään. Raahaa ja pudota -ominaisuuden toteuttamiseen käytimme valmista kirjastoa nimeltä React-beautiful-dnd.



Kuva 30 Tehtävien siirtäminen raahaamalla

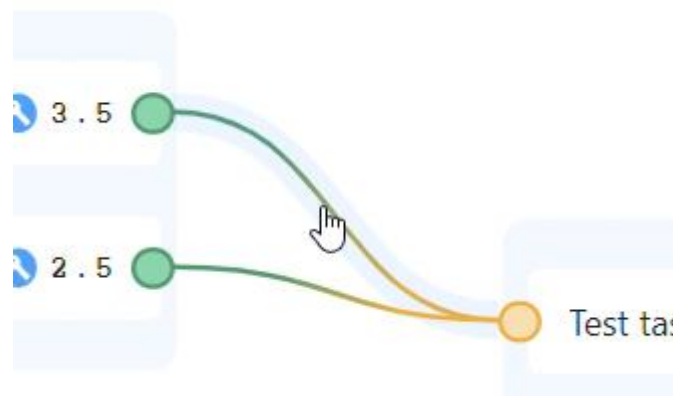
6.4.6 Värien käyttäminen tehtäväkartassa

Jotta tehtäväkartan käyttäminen olisi mahdollisimman sulavaa, pyrimme tekemään sen käyttöliittymästä mahdollisimman itsestään selittyvän. Tähän tavoitteeseen pääsimme käyttämällä eri tilanteisiin sopivia värejä. Esimerkkinä tästä voidaan käyttää tilannetta, jossa käyttäjä haluaisi luoda uuden suhteen kahden tehtävän välille. Jotta käyttäjän ei tarvitse tietää tarkalleen jokaista tehtävää, johon valitun tehtävän voi liittää, päätimme korostaa mahdollisia suhteita värien avulla. Ne tehtävät, jotka voi yhdistää valittuun tehtävään korostuvat hohtovärillä, ja ne, joihin ei, muuttuvat harmaaksi (kuva 31). Tällä pyrimme siihen, että käyttäjä näkee välittömästi mitä toimintoja sillä hetkellä pystyy tekemään.



Kuva 31 Mahdolliset vaihtoehdot osoitettuna korostetulla värillä

Myös suhteiden viivoja valitessa lisäsimme taustalle korostavan värin. Tämä oli tärkeää varsinkin tilanteissa, joissa suhdeviivat ylittävät toisiaan ja niitä on useita. Jotta käyttäjä tietää varmasti, minkä suhteen yllä hänen hiirensä on, tuli se visualisoida käyttäjälle (kuva 32). Käyttäjätestauksissa huomasimme ennen taustavärin lisäämistä usein tilanteita, että vääriä suhdeviivoja poistettiin, kun se ei ollut selkeää, mikä viivoista oli käyttäjän valitsemana.



Kuva 32 Suhteiden korostus

6.4.7 Tehtäväkartan visuaalinen selkeys ja suorituskyky suurilla datajoukoilla

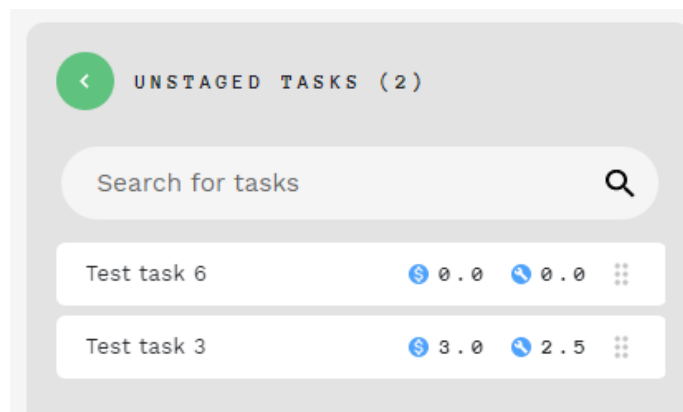
Web-sovellusta suunniteltaessa käyttäjäkokemuksen laadun ylläpitämiseksi tulee myös sovelluksen tehokkuus ottaa huomioon. Usein jo pienenkin hetken lataaminen saa käyttäjän kokemaan epämieltyä työkälyä käyttäessään. Myös semioottinen selkeys on otettava

huomioon, joten näytöllä pitäisi pyrkiä näyttämään pääosin vain niitä asioita, joita käyttäjä tarvitsee. Liiallinen määrä visuaalisia elementtejä kerralla aiheuttaa helposti sekaannusta ja oleelliset asiat jäävät huomaamatta.

Vähentääkseen visuaalista sekaannusta, lisäsimme käyttäjälle mahdollisuuden päättää, mitä tehtäviä kartalla näytetään. Isommassa projektissa voi olla satoja, tai jopa tuhatta, tehtäviä, joille ei ole merkitty laisinkaan toisiinsa liittyviä suhteita. Tämä tarkoittaisi, että tehtäväkartalla oli pahimmassa tapauksessa tuhat yksittäistä tehtävää kerralla näkyvissä. Tämä johtaisi myös siihen, että löytääkseen etsimänsä tehtävän, käyttäjän tulisi liikkua toivottoman suurilla matkoilla kartalla, sillä kartalle mahtuu kerrallaan pystysuunnassa lukematon määrä tehtäviä.

Suuret tietojoukot aiheuttivat aluksi suorituskykyyn ongelmia, sillä yksittäinen projekti voi sisältää useita satoja tehtäviä ja tehtäväkartta piirsi oletuksena kaikki tehtävät kerralla kankaalle (canvas) näkyviin. Ratkaisuna tähän, vaihdoimme tehtävien piirtämislogiikan siten, että ainoastaan ne tehtävät, joilta löytyi joko riippuvuus- tai synergiasuhteita, piirretään oletuksena. Muille yksittäisille tehtäville teimme oman osion näyttämättömät tehtävät (unstaged tasks) (kuva 33). Kun lataat tehtäväkartan sivun, aina ennen tehtävien piirtämistä kankaalle, ajetaan näyttämättömiin tehtäviin edellytykset täyttävät tehtävät.

Listan voi joko avata, tai piilottaa omien tarpeiden mukaan. Listan avaamisessa on tietty logiikka. Jos tehtäväkartalle on asetettu näkyviin tehtäviä, lista on oletuksena kiinni. Mutta jos tehtäväkartta on tyhjä, silloin näyttämättömien tehtävien lista avautuu oletuksena käyttäjälle. Perustelut tälle käytökselle on siinä, että jos käyttäjä siirtyy tehtäväkartta sivulle, eikä siellä ole vielä yhtään kuvannettua tehtävää, niin ensimmäinen asia mitä käyttäjä tekisi, on tehtävälistan avaaminen aloittaakseen tehtävien siirtelyn kartalle.



Kuva 33 Lisäämättömien tehtävien lista

Tämän lisäksi tehtäväkartan suorituskykyä on pyritty parantamaan vähentämällä uudelleenpiirtämistä mahdollisimman paljon.

6.4.8 Kehityskohteet

Projektien loppuvaiheilla tulee usein ongelmaksi tehtävien priorisointi. Asiakkaan toiveista ja muista projektiin liittyvistä tehtävistä on valittava kaikista oleellisimmat ja tärkeimmät. Myös VISDOM projekti oli loppuillaan ja siksi meidänkin täytyi tietyissä kohdissa priorisoida ajankäyttöä. Tämän takia tehtäväkarttaankin jäi muutamia asioita, joita tulisi vielä mahdollisessa jatkokehityksessä huomioida.

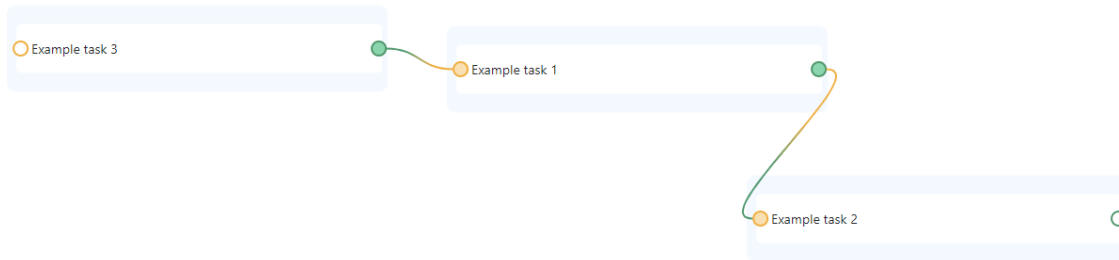
Yhtenä ongelmana tuli tehtävien suhteita kuvaavat viivat, sekä niiden värimaailma. Tehtävien väliset suhdeviivat piirretään siten, että viiva piirretään aluksi metsänvihreänä, ja kun viivaa on edetty kolmekymmentä prosenttia vasemmalta oikealle, se alkaa hitaasti vaihtamaan väriään avulla oranssiksi. Seitsemänkymmenen prosentin kohdalla viiva on jo täysin oranssi (kuva 34).

```
<linearGradient
  id="linearGradient"
  className={classes({ [css.loading]: isLoading })}
>
  <stop className={classes(css.firstStop)} offset="30%" />
  <stop className={classes(css.secondStop)} offset="70%" />
</linearGradient>
```

Kuva 34 Värien piirtämisen logiikka.

Ongelmaksi tässä ratkaisussa tuli se, että koodi olettaa aina, että lähtöpisteen X-koordinaatit ovat pienemmät kuin saapumispisteen X-koordinaatit. Ratkaisuksi tähän kokeiltiin useita eri ratkaisuita. Esimerkkinä toimii tilanne, jossa X-koordinaatin sijainnit laskettiin näytöltä ja käänteisessä tilanteessa myös oranssin ja metsänvihreän värit vaihdettiin keskenään, jolloin saataisiin haluttu lopputulos. Ongelmaksi tässä tuli tehtäväkartassa käytetyn valmiin kirjaston viivojen piirtämislogiikka, johon emme pystyneet vaikuttamaan. Ohjelmointilogiikan toimivuuden pystyi tarkistamaan konsoliin tulostamisella, mutta vaikka siellä näytti kaikki olevan kohdallaan, siltikään viivojen värit eivät vastanneet odotettua. Testasimme myös sellaista vaihtoehtoa, että viivoista olisi tehty yksivärisiä. Tämä ei kuitenkaan näyttänyt esteettisesti yhtä miellyttävältä kuin sulavasti

vaihtuva väri, joten päädyimme jättämään tällä kertaa tämän virheen tehtäväkarttaan (kuva 35). Tilannetta ei kuitenkaan kovin usein tule vastaan, jolloin edellä oleva tehtävä raahattaisiin tehtäväkartalla edellisen taakse, eikä tämä vaikuta tehtäväkartan käyttöön laisinkaan.



Kuva 35 Virheellinen värimaailma

Jos nyt tehtäväkartan tekeminen pitäisi aloittaa täysin alusta, suurimpana erona olisi se, ettei tukeudu valmiiden kirjastojen käyttämiseen näin paljoa. Kirjastojen valitseminen tehtiin ajan ja resurssien säästämiseksi, sillä useinhan sanotaan, ettei pyörää kannata keksiä uudelleen. Tässä tapauksessa pyörän alusta asti itse kehittäminen olisi voinutkin lopulta olla tehokkaampi ratkaisu, sillä tehtäväkarttaa tulikin tehdä niin tarkaksi ja spesifiksi, että useammassakin tilanteessa tuli ongelmia vastaan eri kirjastojen taipumisen kanssa. Vaikka nämä kirjastot olivatkin toiminnallisuudeltaan erinomaisia ja niiden käyttötarkoituksiin todella hyviä, liian tarkat vain tehtäväkarttaan liittyvät ominaisuudet hankaloittivat sen käyttöä ja sen rinnalla kehitystyötä huomattavasti. Isoin syy näille ongelmille oli oletettavasti se, että monen kirjaston rinnakkaiskäytössä kirjastojen väleillä saattaa tulla ristiriitoja, joihin pystyisi vain vaikuttamaan kirjaston kehittäjä. Ja tätä ei tietenkään voi kirjaston kehittäjältä vaatia, vaan tällaisessa tilanteessa tulisi kehittää itse omaan sovellukseen sopiva ratkaisu.

7. Yhteenveto

Erilaisia tietoja ja dataa on visualisoitu jo vuosituhansien ajan. Nykyään visualisoinneilla on keskeinen rooli ja erityisen tärkeitä ne ovat ohjelmistokehityksessä ja -projekteissa. Visualisoinneilla voidaan tarkoittaa esimerkiksi datan esittämistä graafin muodossa tai muita kuvallisia esitystapoja. Ohjelmistokehityksessä tyypillisenä graafisena metodina on ohjelmistojen mallintaminen UML-kaavioilla. Visuaalisten esitystapojen päätavoitteena on esittää tiedon sisältämä viesti, mikä usein raasta datasta jää huomaamatta.

Visuaalisille merkintätavoille on tärkeää, että ne ovat helposti ymmärrettäviä. Selkeyden lisäksi merkintöjä suunniteltaessa tulee ottaa huomioon myös värien käyttäminen. Esimerkiksi tilanteissa, joissa visualisointeja täytyy pystyä tulostamaan tai piirtämään manuaalisesti taululle, tulisi värien käyttöä välttää. Värit voivat myös olla tehokas keino ohjata käyttäjää toimimaan oikein. Esimerkiksi tietyillä tehosteilla pystytään helposti opastamaan käyttäjää eri toiminnoista.

Tyypillisiä numeerisen datan visuaalisia esitystapoja ovat lämpökartta, viivadiagrammi, pylväsdiagrammit, ympyrädiagrammit, puoliympyrädiagrammit. Niiden tärkein ominaisuus on tuoda esille tietty kaava, mikä datasta saattaa olla piilossa. Ohjelmistoprojekteissa visualisoinneilla on keskeinen merkitys varsinkin kommunikoinnin, suunnittelun, aikatauluttamisen ja seurannan kannalta. Näiden eri mittarien avulla voidaan saada esille mahdolliset ongelmat ja sen pohjalta pystytään tekemään korjaavia liikkeitä. Myös erilaiset tilastot ovat tärkeitä kommunikoinnin välineitä sidosryhmien välillä, erityisesti tilanteissa, joissa kaikilla ei ole sama tausta. Projektin sisällä yleinen metodi työnkulun seurantaan on Kanban-taulu. Siinä työnkulun vaiheet kirjataan tauluun ja korteilla kuvastetaan työtehtäviä. Korteja siirtelemällä pystytään visualisoimaan työn etenemistä. Projektin muita hyviä mittareita ovat esimerkiksi etenemiskaaviot, kriittinen polku, sekä Gantt-kaavio.

Tehtävillä on myös usein riippuvuuksia muiden tehtävien välillä. Joitakin tehtävien ei esimerkiksi voi aloittaa, ennen kuin edelliset ovat viimeisteltynä. Tällaisia tilanteita varten visualisoinnit ovat erityisen hyviä, sillä jo nopealla vilkaisulla pystyy tarkistamaan, että onko tehtävän esivaatimukset suoritettuna.

8. Johtopäätökset

Tutkimuksen tavoitteena oli kehittää VISDOM Roadmapperin visualisoinnin tekniikoita, sekä luoda visuaalinen esitystapa tietojoukkojen välisille suhteille. Mielestäni tietojoukkojen väliset suhteet saatiin visualisoitua tehokkaasti ja selkeästi tehtäväkartan avulla. Tehtäväkartta näyttää onnistuneesti käyttäjille tehtävien esivaatimukset, ryhmät, sekä jälkivaatimukset. Näiden avulla käyttäjien on helpompi rakentaa tehtävien pohjalta virstanpylväät, sillä tehtävien tekemiseen vaadittu tietty järjestys on visuaalisesti nähtävissä. Ennen tehtäväkarttaa, virstanpylväitä suunnitellessa olisi täytynyt muistaa tarkasti mitkä tehtävät liittyvät toisiinsa, tai muuten riskinä olisi ollut tilanne, jossa toisistaan riippuvat tehtävät tulisi suunnitelman mukaan tehdä väärässä järjestyksessä. Tämä taas johtaisi siihen, että mahdollisesti kesken kehitystyön suunnitelmaa tulisi muokata ja korjata. Tästä voidaan tulkita, että näiden suhteiden visuaalinen esitystapa oli tarpeellinen tähän tilanteeseen, sillä ennen tehtäväkarttaa projektin suunnitteleminen oli huomattavasti haastavampaa. Tehtäväkartta nopeuttaa myös huomattavasti tehtävien välisten suhteiden asettamista, sillä sen sijaan, että käyttäjien täytyisi mennä yksitellen tehtäviä läpi, nyt saadaan tehtäväkartan avulla kaikki tarvittavat tehtävät näkyville samaan aikaan. Tässä näkymässä eri siirtojen teko on erittäin nopeaa.

Tehtäväkarttaa ja tehtävien välisiä suhteita luodessa nousi myös esille automaattiset varoitukset käyttäjille virstanpylväiden suunnitteluun. Jos käyttäjä on tekemässä ristiriitaisia siirtoja tehtävillä, osaa suunnitelmasivulle luotu algoritmi myös varoittaa ristiriidoista pohjautuen tehtäväkartalla tehtyihin vaatimuksiin ja suhteisiin. Tehtäväkartan ja tämän algoritmin avulla tulevien virstanpylväiden suunnittelun, sekä tehtävien välisten suhteiden muovaaminen, on huomattavasti yksinkertaisempaa Roadmapper-sovelluksessa, kuin ennen.

9. Viiteluettelo

- Australian Bureau of Statistics (2023). Statistical Terms and Concepts: Data, <https://www.abs.gov.au/statistics/understanding-statistics/statistical-terms-and-concepts/data>. Haettu 18.4.2023
- Chen, A., & Beatty, J. (2012). *Visual models for software requirements*. Pearson Education.
- Chen, C. H., Härdle, W. K., & Unwin, A. (Eds.). (2007). *Handbook of data visualization*. Springer Science & Business Media.
- Chonoles, M. J. (2017). *Ocup 2 certification guide: Preparing for the omg certified uml 2.5 professional 2 foundation exam*. Morgan Kaufmann.
- Damij, N., & Damij, T. (2021). An approach to optimizing Kanban board workflow and shortening the project management plan. *IEEE Transactions on Engineering Management*.
- Delnero, P., (2017). A Land with No Borders: A New Interpretation of the Babylonian “Map of the World”. *Journal of Ancient Near Eastern History*, 4(1-2), 19-37.
- Deng, W., Wang, Y., Liu, Z., Cheng, H. and Xue, Y., (2014). *HemI: a toolkit for illustrating heatmaps*. PloS one, 9(11), p.e111988.
- East, W., (2015). *Critical Path Method (CPM) tutor for construction planning and scheduling*. McGraw Hill Professional.
- Firth, A., (2019). *Practical web inclusion and accessibility: A comprehensive guide to access needs*. Apress.
- Gašević, D., Lämmel, R., & Van Wyk, E. (Eds.). (2009). *Software Language Engineering: First International Conference, SLE 2008 Toulouse, France, September 29-30, 2008, Revised Selected Papers* (Vol. 5452). Springer.
- Griffiths, D. (2008). *Head first statistics*. O'Reilly Germany.
- Gu, Z., Eils, R. & Schlesner, M., (2016). *Complex heatmaps reveal patterns and correlations in multidimensional genomic data*. *Bioinformatics*, 32(18), pp.2847-2849.
- Hammarberg, M., & Sunden, J. (2014). *Kanban in action*. Manning Publications Co..
- Hardy, G., (2011). *Smashing logo design: The art of creating visual identities* (Vol. 24). John Wiley & Sons.
- Huang, M. L., Nguyen, Q. V., & Zhang, K. (2010). *Visual information communication*. Berlin: Springer.
- Johnson, J., (2020). *Designing with the mind in mind: simple guide to understanding user interface design guidelines*. Morgan Kaufmann.
- Krause, A., & OConnell, M. (Eds.). (2012). *A picture is worth a thousand tables: graphics in life sciences*. Springer Science & Business Media.

- Lin, S., Fortuna, J., Kulkarni, C., Stone, M., & Heer, J. (2013). Selecting semantically-resonant colors for data visualization. *In Computer Graphics Forum* (Vol. 32, No. 3pt4, pp. 401-410). Oxford, UK: Blackwell Publishing Ltd.
- Mao, C., (2009). Visualizing table dependency relations to reveal network characters in database applications. *In Visual Information Communication* (pp. 283-297). Springer, Boston, MA.
- Mehl, R., (2013). *Playing with Color: 50 Graphic Experiments for Exploring Color Design Principles*. Rockport Pub.
- Moody, D. (2009). The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on software engineering*, 35(6), 756-779.
- Moss, L. T., & Atre, S. (2003). *Business intelligence roadmap: the complete project lifecycle for decision-support applications*. Addison-Wesley Professional.
- Park, B. K., Kang, G. H., Son, H. S., Jeon, B., & Kim, R. Y. C. (2020). Code Visualization for Performance Improvement of Java Code for Controlling Smart Traffic System in the Smart City. *Applied Sciences*, 10(8), 2880.
- Rodriguez, J., & Kaczmarek, P. (2016). *Visualizing financial data*. John Wiley & Sons.
- Saito, S., Iimura, Y., Massey, A. K., & Antón, A. I. (2018). Discovering undocumented knowledge through visualization of agile software development activities: Case studies on industrial projects using issue tracking system and version control system. *Requirements Engineering*, 23, 381-399.
- Samara, T. (2014). *Design Elements: Understanding the rules and knowing when to break them-Updated and Expanded*. Adams Media.
- Shahid, B. (2014). *Highcharts essentials*. Packt Publishing Ltd.
- Sharon, A., & Dori, D., (2017). Model-Based Project-Product Lifecycle Management and Gantt Chart Models: A Comparative Study. *Systems engineering*, 20(5), 447-466.
- Sherin, A., (2012). *Design elements, Color fundamentals: A graphic style manual for understanding how color affects design*. Rockport Publishers.
- Spinde, T., Jeggle, C., Haupt, M., Gaissmaier, W., & Giese, H. (2022). How do we raise media bias awareness effectively? Effects of visualizations to communicate bias. *Plos one*, 17(4), e0266204.
- Steele, J., & Iliinsky, N. (2010). *Beautiful visualization: Looking at data through the eyes of experts*. " O'Reilly Media, Inc."
- Vajda, A. (2009). PROJECT MONITORING AND CONTROL USING BURNDOWN CHARTS. *In The International Conference Interdisciplinarity in Engineering INTER-ENG* (p. 196). Elsevier Limited.

- Wang, Z., Wang, D., & Li, Q. (2021). Data Visualization on the Life Cycle of Science and Technology Projects. In 2021 *IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)* (pp. 48-52). IEEE.
- Ware, C. (2012). *Information visualization: perception for design*. Morgan Kaufmann.
- William/Holden Kritina/Butler Lidwell (Jill), (2010). *Universal principles of design: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Quayside Publishing Group.
- Wu, R., Wu, H. H., & Wang, C. L. (2021). Why is a picture ‘worth a thousand words’? Pictures as information in perceived helpfulness of online reviews. *International Journal of Consumer Studies*, 45(3), 364-378.
- Yuk, M., & Diamond, S. (2014). *Data visualization for dummies*. John Wiley & Sons.