



## **KOMPLEKSISUUDESTA KÄYTETTÄVYYTEEN: ERIKOISOHJELMISTOJEN KEHITYS**

Lappeenrannan–Lahden teknillinen yliopisto LUT

Tietotekniikan kandidaatintyö

2022

Janne Pennanen

Tarkastaja: Associate Professor Ari Happonen

## TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tietotekniikan koulutusohjelma

Janne Pennanen

### **Kompleksisuudesta käytettävyyteen: Erikoisohjelmistojen kehitys**

Tietotekniikan kandidaatintyö

36 sivua, 10 kuvaa ja 1 taulukko

Tarkastaja: Associate Professor Ari Happonen

Avainsanat: digitalisaatio, kompleksisuus, käytettävyys, web-kehitys, erikoisohjelmistot, tapaustudkimus, asiakassuhde

Nykypäivän maailmassa panostetaan digitalisaatioon paljon ja tiedon saattaminen helposti saatavilla olevaan sekä käytettävään muotoon on tärkeämpää kuin koskaan. Tämä tosiasia koskee myös sellaista tietoa, jonka oletettu käyttäjäkunta ja -tarkoitus ovat suppeat. Hyvä esimerkki tällaisesta tiedosta on LUT-yliopiston konetekniikan osastoon kuuluvien Aholan et al. tekemä tutkimus, jossa kehitettiin nk. 4R-metodi tietynlaisten teräshitsien väsymislaskentaan.

Tässä kandidaatintyössä perehdytään ensin erikoisohjelmistojen kehityksen ominaispiirteisiin sekä ohjelmistotuotteiden käytettävyyteen. Tämän jälkeen toteutetaan yhteistyössä konetekniikan osaston kanssa selainpohjainen työkalu, joka suorittaa 4R-metodin mukaisen laskennan käyttäjän antamalla lähtöarvoilla. Näin tuodaan metodi sellaiseen muotoon, jossa se on helposti saatavilla ja sitä on helppo käyttää.

Kirjallisuudessa painotettiin sitä, että vastaavanlaisissa kehitysprojekteissa harvoin tiedetään heti alussa mitä lopputulokselta halutaan. Tämän takia asiakassuhteen ja -kommunikaation merkitys korostuu, sillä projektia ohjataan kohti asiakkaan haluamaa lopputulosta jatkuvasti koko kehitysprosessin ajan.

Työkalu saatiin valmiiksi aikataulussa ja se vastaa konetekniikan osaston odotuksia. Tähän lopputulokseen päästiin jatkuvan asiakaskommunikaation ansiosta, ja työn tulokset vaikuttavatkin vahvistavan kirjallisuuden luomaa käsitystä siitä, että tiivis asiakassuhde on välttämättömyys erikoisohjelmistojen kehityksessä.

## ABSTRACT

Lappeenranta-Lahti University of Technology LUT

School of Engineering Science

Software Engineering

Janne Pennanen

### **Converting complexity to usability: Developing specialized software**

Bachelor's thesis

2022

36 pages, 10 figures and 1 table

Examiner: Associate Professor Ari Happonen

Keywords: digitalization, complexity, usability, web-development, specialized software, case study, customer relationship

In today's world, a lot is invested in digitalization and easy access to information is more important than ever. This fact also applies to information whose assumed user base and use case are limited. A good example of this kind of information is a study conducted by Ahola et al. from LUT Mechanical Engineering, in which the so-called 4R method for calculating the structural fatigue of steel weld joints was developed.

Firstly, this thesis investigates the characteristics of developing specialized software and the usability of software products. Afterwards, a tool that, when given an input by the user, performs calculations according to the 4R method, is developed in cooperation with LUT Mechanical Engineering. Thus, the method is brought into a format where it is easy to use and access.

In related literature, the fact that, in specialized software projects, it is rarely known exactly what is wanted from the result is highlighted. This underlines the importance of the customer-developer relationship and communication in specialized software development since these projects are steered towards the result desired by the customer throughout the whole development process.

The tool was completed on schedule and satisfies the needs of LUT Mechanical Engineering. This result was made possible by continuous customer communication and the findings of this thesis seem to indeed strengthen the idea that a close customer-development relationship is vital to the success of specialized software projects.

## KIITOKSET

Kiitos konetekniikan osastolle, etenkin Antti Aholalle ja Timo Björkille mielenkiintoisesta aiheesta ja yhteistyöstä työn toteutuksen aikana.

Kiitos työni ohjaajalle, Ari Happoselle neuvoista ja yleisesti työn ohjauksesta.

Kiitos myös ystäväilleni ja sukulaisille, joiden kanssa on voinut vaihtaa ajatuksia työstä ja joille on voinut purkaa stressiä.

Lopuksi kiitos avopuolisolleni Roosalle tuesta, patistamisesta ja luottamuksesta niinä hetkinä joina niitä tarvittiin.

## LYHENTEET JA TERMIT

5G	Viidennen sukupolven mobiiliverkko
4R-metodi	Laskentametodi, jolla voidaan analysoida teräshitsien väsymistä
Function-plot	Kuvaajien piirron mahdollistava kirjasto JavaScriptille
JavaScript	Selainpohjaiseen kehitykseen soveltuva ohjelmointikieli
LUT-yliopisto	Lappeenrannan-Lahden Teknillinen Yliopisto
Matlab	Laskentaohjelmisto ja -kieli
Plotly.js	Kuvaajien piirron mahdollistava kirjasto JavaScriptille
PNG	Portable Network Graphics; Kuvien tallennusformaatti
React	Käyttöliittymäkirjasto JavaScriptille
Teams	Microsoftin kehittämä viestintäalusta
UHSS	Ultra-high-strength steel; Ultraluja teräs
USD	Yhdysvaltain dollari
Web	Verkko; Internet

## SYMBOLIT

$\Delta\sigma_k(r)$	Lovijännitysvaihtelu
$\sigma$	Paikallisjännitys
$\Delta\sigma$	Paikallisjännitysvaihtelu
$\sigma_{\max}$	Paikallisen jännityksen maksimi
$\sigma_{\min}$	Paikallisen jännityksen minimi
$\sigma_{\text{res}}$	Jäännösjännitys
$E$	Materiaalin kimmokerroin
$R$	Ulkoisen kuormituksen jännityssuhde

$R_m$	Materiaalin murtolujuus
$R_{local}$	Paikallinen jännityssuhde
$H$	Ramberg-Osgoodin muokkauslujittumisvakio
$n$	Ramberg-Osgoodin muokkauslujittumiseksponentti
$\Delta\sigma_{k,ref}$	Lovijännitysvaihtelun referenssiarvo
$N_f$	Kestoikä
$C$	Väsymiskapasiteetti
$m$	Kaltevuuseksponentti

## Sisällysluettelo

Tiivistelmä

Abstract

Kiitokset

Symboli- ja lyhenneluettelo

1	Johdanto .....	8
1.1	Tausta .....	8
1.2	Tavoitteet, rajaukset ja tutkimuskysymykset .....	10
1.3	Tutkimusmenetelmä.....	11
1.4	Työn rakenne.....	12
2	Kirjallisuuskatsaus .....	13
2.1	Erikoistuneiden ohjelmistojen kehittäminen .....	13
2.2	Käytettävyys ohjelmistotuotteissa.....	14
2.3	Yhteenveto kirjallisuuskatsauksesta.....	16
3	Case: Väsymislaskentatyökalu.....	17
3.1	Projektin lähtökohdat .....	17
3.2	Projektissa tarvittavan koneteknisen teorian esittely .....	18
3.2.1	Yhtälöt.....	18
3.2.2	Laskenta työkalun taustalla .....	20
3.3	Valittu toteutustapa .....	21
3.4	Toteutusprosessi.....	22
4	Prosessin reflektointi.....	31
5	Yhteenveto .....	33
	Lähteet.....	34

# 1 Johdanto

Digitalisaatio muuttaa tiedon jakamisen ja kuluttamisen tapoja ja palveluita sähköisiksi. Tiedon odotetaan olevan helposti saatavilla ja jaettavissa kaikkina vuorokaudenaikoina, vuoden jokaisena päivänä, käyttäjien tarpeet huomioivilla tavoilla. [1] Erikoistuneen tutkimustiedon kohdalla näin ei kuitenkaan usein ole, vaan tehdystä tutkimuksesta julkaistaan joskus hyvinkin vaikealukuinen raportti ja prosessi loppuu siihen. Kuinka erikoistunut tieto voidaan saattaa sellaiseen muotoon, että sen hyödyntäminen on helppoa? Miten tietojen pohjalta rakennetaan erikoistunut työkalu tai tuote, jota voi kuitenkin käyttää perehtymättä alkuperäiseen tutkimukseen? Mitä haasteita siihen liittyy? Tässä luvussa tehdään katsaus digitalisaatioon, tarkennetaan työn tavoite, kuvataan ratkaisumenetelmä ja esitellään työn rakenne.

## 1.1 Tausta

Digitalisaatio, eli tietotekniikan kasvava rooli jokapäiväisissä toiminnoissa, on osa nykypäivää. Maailmamme digitalisoituu jatkuvasti, ja digitalisaation nopeus kiihtyy päivä päivältä. Digitalisaatio on koko yhteiskunnan läpipleikkaava ilmiö. Se vaikuttaa yhtä lailla yritysten tuotanto- [2], [3] ja toimintojen suoritusprosesseihin [4], opiskelijoiden työllistymiseen [5], kuin myös liiketoimintamallien muutokseen [6] ja uusien, palvelumallien syntymiseen [7]. Esimerkiksi Euroopan Komission julkaisemassa digitalisaatioon liittyviä lähestymistapoja ja tavoitteita kuvaavassa *2030 digitaalinen kompassi* -tiedonannossa [8] esitetään, että vuoteen 2030 mennessä kaikilla eurooppalaisilla kotitalouksilla olisi gigabittiyhteys ja 5G-verkko kattaisi kaikki asutut alueet, kun vuonna 2020 mahdollisuus gigabittiyhteyteen oli 59 %:lla talouksista ja 5G-verkon kattavuus 14 %. Yritysten digitalisaatioon liittyen tavoitellaan, että 75 % eurooppalaisista yrityksistä käyttäisi pilvipalveluja, massadataa ja tekoälyä, vuoden 2020 lukemien ollessa pilvipalveluille 26, massadatalle 14 ja tekoälylle 25 prosenttia. Yhdysvalloissa hyväksyttiin juuri infrastruktuuripaketti, joka sisältää muun muassa 65 miljardin Yhdysvaltain dollarin investoinnin laajakaistayhteyksien käyttöönottoon, tavoitteenaan taata internetyhteys jokaiselle amerikkalaiselle [9]. Kaiken kaikkiaan maailmanlaajuisen tietotekniikkakulutuksen ennustetaan nousevan lähes 4,5 biljoonaan dollariin vuoden 2022 aikana (taulukko 1).



Taulukko 1. Ennuste maailmanlaajuisesta IT-kulutuksesta 2021–2023, summat miljoonia USD pyöristettynä tuhansien tarkkuudelle [10].

	2021 Kulutus (USD)	2021 Kasvu (%)	2022 Kulutus (USD)	2022 Kasvu (%)	2023 Kulutus (USD)	2023 Kasvu (%)
Datakeskusjärjestelmät	216 000	11,4	227 000	4,7	238 000	4,7
Yritysohjelmistot	605 000	14,4	672 000	11,0	752 000	11,9
Laitteet	787 000	13,0	814 000	3,3	804 000	-1,2
IT-palvelut	1 186 000	10,7	1 280 000	7,9	1 392 000	8,8
Tietoliikennepalvelut	1 444 000	3,4	1 463 000	1,3	1 494 000	2,2
<b>Yhteensä</b>	<b>4 239 000</b>	<b>9,0</b>	<b>4 454 000</b>	<b>5,1</b>	<b>4 679 000</b>	<b>5,0</b>

Edellä kuvatun kaltaisen kehityksen seurauksena yhä suurempi osa jakamastamme ja kuluttamastamme tiedosta on digitaalisessa muodossa; muun muassa oppimateriaalit sähköistyvät halki koulutusasteiden, ja sanomalehtien digitaalijulkaisut kasvattavat suosiotaan. Lisäksi jo vuosituhannen taitteessa esitettiin, että käytettävyyden merkitys kasvaa tietotekniikan muuttuessa yleiseksi hyödykkeeksi ja että kuluttajat odottavat käytettävyyden olevan keskeinen osa tuotekehitystä [11]. Digitaalisen tiedon kulutuksen yleistyessä käytettävyys, eli käyttämisen helppous on tullut tärkeäksi osaksi myös tiedonjakoa. Esimerkiksi tieteellisen tutkimuksen tuloksena saatu tieto tulisi pystyä esittämään sekä digitaalisessa muodossa että siten, että se on saatavilla helposti ja sitä on helppo hyödyntää.

Tiedonjaon helppoutteen pyrkiminen ei kuitenkaan ole uusi asia. Avoimesta datasta ja sen potentiaalisista positiivisista vaikutuksista muun muassa tieteen saralla on konseptina puhuttu jo vuosia [12], mutta sen tavoitetta eli yhteiskuntaa jossa kuka tahansa voi hyödyntää dataa käytännössä mistä tahansa, ei ole kuitenkaan koskaan saavutettu. Avoimen datan haasteet ovat lukuisat, esimerkiksi yritysten liiketoimintamallien perustuminen juuri datan arvoon ja sitä johtuva haluttomuus datan jakamiseen, julkisen yleisön tietämättömyys avoimesta datasta tai sen käyttökohteista ja saatavilla olevan datan liika kompleksisuus [13]. Ei siis riitä, että dataa ja tietoa on saatavilla, vaan yhä täytyy ottaa askelia, joilla ne saatetaan käytettävään muotoon: tuotteiksi ja työkaluiksi.

LUT-yliopiston konetekniikan osastoon kuuluvat Ahola et al. [14] määrittelevät artikkelissaan tietynlaisten ultralujien terästen (engl. ultra-high-strength steel; UHSS) hitsien väsymislaskentaa heidän kehittämänsä nk. 4R-metodin avulla. Metodia käyttämällä voidaan muun muassa arvioida hitsien kestoikiä, ja se soveltuu myös muista kuin UHSS-teräksistä tehdyille hitsiliitoksille. Artikkelin on hyvä esimerkki tiedosta, joka on julkaistu mutta jota voi työstää edelleen käytännölliseen muotoon siten, että siitä pääsee hyötymään ilman artikkelin läpikäymistä ja syvällistä tietämystä siihen liittyvästä teoriasta. Konetekniikan osasto haluaakin, että hitsiliitosten väsymislaskennalle kehitettäisiin graafisella käyttöliittymällä varustettu, helppokäyttöinen ja helposti jaettava työkalu. Konetekniikan osastolla on olemassa komentoriviltä käytettävä Matlab-ohjelma, joka kykenee suorittamaan osan 4R-metodin vaatimasta laskennasta, mutta tavoitteena olisi graafisella käyttöliittymällä varustettu työkalu, joka pystyisi tuottamaan kestoikatulokset suoraan.

## 1.2 Tavoitteet, rajaukset ja tutkimuskysymykset

Tämän kandidaatintyön tavoitteena on ensin selvittää kirjallisuuskatsauksena, miten erikoistuneita, esimerkiksi tieteelliseen tutkimustietoon perustuvia ohjelmistoja kannattaa kehittää ja minkälaisia haasteita niiden kehittämiseen yleensä liittyy. Minkälainen merkitys asiakas-suhteella on ohjelmistokehityksessä ylipäätään ja kuinka asiakkaan kanssa kannattaa kommunikoida, etenkin tilanteessa, jossa ohjelmistoa kehitetään asiakkaalle tutun, mutta kehittäjälle vieraan tiedon pohjalta? Tietoa etsitään myös käytettävyyteen liittyen, sekä tutkitaan syitä sille, miksi käytettävyys on nykyään odotusarvo valtaosalle tuotteista ja työkaluista sekä ohjelmistoalalla että sen ulkopuolella.

Löydettyjä tietoja hyödyntämällä ja niihin peilaamalla kehitetään Aholan et al. [14] tutkimukseen ja laskentametodeihin perustuva selainpohjainen, hitsiliitosten väsymistä laskeva ja tuloksia graafein kuvaava työkalu LUT-yliopiston konetekniikan osastolle. Työkalusta pyritään tekemään mahdollisimman helppokäyttöinen, sillä halutaan osoittaa, että erikoistuneen ohjelman ei tarvitse olla vaikea käyttää. Kehitettävä työkalu on hyvä esimerkki erikoistuneesta ohjelmistosta; sitä ei kehitetä suurille massoille, vaan sen käyttäjäkunta ja käyttötapaukset ovat melko rajatut.

Työn tavoitteisiin ei kuulu erilaisten ohjelmistokehitysalustojen, -kielten ja -prosessien ominaisuuksien vertailu ja toteutustavan valinta tehdyn vertailun perusteella, vaan ohjelman toteutustapa ja prosessin kulku on täysin työn toteuttajan itse valittavissa. Valittu toteutustapa kuitenkin esitellään ja lyhyt perustelu sen soveltuvuudelle tähän tapaukseen annetaan. Työssä ei myöskään käydä läpi hitsien väsymislaskennan teoreettista puolta yksityiskohtaisesti, vaan ainoastaan niiltä osin, joita ohjelman kehityksessä tarvitaan. Työkalun vienti tuotantoon eli käyttäjien saataville rajataan myös pois tästä työstä, sillä se tullaan julkaisemaan LUT-yliopiston verkkosivujen yhteyteen ja julkaisuprosessi voi kestää kauankin.

Työn tutkimuskysymykset ovat seuraavat:

- 1) Mitä ominaispiirteitä erikoisohjelmistojen kehitykseen liittyy ja mitkä asiat vaikuttavat kehitysprosessiin?
- 2) Minkälaisia haasteita lähtötiedon kompleksisuus asettaa toisaalta kehitysprosessille ja toisaalta lopputuloksen saattamiselle käytettävään muotoon?
- 3) Millä tavoin edellä mainittuihin haasteisiin tulee vastata, jotta päästään haluttuun lopputulokseen?

### 1.3 Tutkimusmenetelmä

Työ toteutetaan tapaustutkimuksena, sillä se sopii sekä tehtävän luonteeseen että ohjelmistokehitysohjelmien ylipäättään hyvin. Tapaustutkimuksissa tutkitaan valittua asiaa tai ilmiötä sen luonnollisessa kontekstissa, mikä sopii hyvin ohjelmistokehitykseen liittyviin tutkimuksiin, sillä niiden tavoitteena on usein ymmärtää, kuinka ohjelmistokehitysprosessi etenee tietyissä olosuhteissa [15]. Tässä työssä se tarkoittaa sitä, että pyritään saavuttamaan parempi ymmärrys erikoisohjelmistojen kehityksestä, sen haasteista ja niihin vastaamisesta kehittämällä aiemmin kuvailtu työkalu konetekniikan osastolle. Työssä käydään läpi työkalun teknisen toteutuksen vaiheet ja peilataan kirjallisuuden teoriaa tosielämän toteutukseen: Vastaako toteutus kirjallisuuskatsauksen luomia odotuksia, löytyikö kirjallisuudesta esimerkiksi toimintatapoja tai menetelmiä, joita projektissa pystyttiin hyödyntämään? Ilmenikö toteutuksessa haasteita tai muita asioita, joihin kirjallisuudessa ei oteta kantaa? Havainnot käydään läpi ja niiden perusteella otetaan kantaa siihen, miten erikoistuneen

ohjelmistokehityksen ominaispiirteet ilmenivät projektissa ja miten niitä käsiteltiin. Tämän perusteella annetaan ohjeita jatkotutkimukselle ja vastaaville projekteille tulevaisuudessa.

#### 1.4 Työn rakenne

Johdannossa käsiteltiin tiedon jakamisen ja kuluttamisen helppouden kasvavaa merkitystä ja sitä, ettei avoimen datan ideaalia ole vielä saavutettu. Toinen luku paneutuu kirjallisuuteen ja pyrkii löytämään sieltä tietoa, jota toteutuksessa voidaan käyttää hyödyksi. Kolmannessa luvussa siirrytään käytäntöön eli väsymislaskennan teorian esittelyn kautta ohjelman toteutuksen läpikäyntiin ja erittelyyn. Neljännessä luvussa keskustellaan siitä, miten teoria ja käytäntö kohtasivat tässä työssä käsitellyssä tapauksessa. Viides luku vetää yhteen työn havainnot ja pyrkii antamaan ohjeita vastaavanlaisille projekteille tulevaisuudessa.

## 2 Kirjallisuuskatsaus

Tässä luvussa etsitään kirjallisuudesta tietoa liittyen erikoisohjelmistojen rakentamiseen, käytettävyyteen ja asiakassuhteen vaikutukseen ohjelmistokehityksessä. Kirjallisuuskatsauksen tavoitteena on luoda pohja sille, että ohjelmiston toteutus onnistuu perehtymällä ongelmiin, joita vastaavanlaisten ohjelmistojen rakentamisessa usein kohdataan, miettimällä tapoja, joilla niitä voidaan välttää ja etsimällä metodeja ja käytäntöjä, joista voi olla hyötyä projektin toteutuksessa.

### 2.1 Erikoistuneiden ohjelmistojen kehittäminen

Yksi ensimmäisistä vaiheista ohjelmistokehityksessä on vaatimusmäärittely; mitä kehitettävältä ohjelmistolta tai työkalulta vaaditaan, minkälainen sen halutaan olevan? Erikoisohjelmiston kehittäminen spesifiin tarkoitukseen eroaa kaupallisesta ohjelmistotuotannosta jo tässä vaiheessa. Carverin et al. [16] mukaan erikoistuneiden ohjelmistojen kehitykselle on ominaista se, ettei asiakas usein pysty antamaan tyhjentävää selvitystä vaatimuksista projektin alussa, vaan vaatimukset tarkentuvat projektin edetessä. Morrisin ja Segalin [17] tapaus tutkimus vahvistaa tätä väitettä lisäten huomion siitä, että myöskään loppukäyttäjä ei välttämättä näissä tapauksissa ole hyvä tiedonlähde vaatimuksia määriteltäessä. Loppukäyttäjän vaatimukset voivat olla niin spesifejä, että niiden ymmärtäminen ja täyttäminen ilman tarkkaa ymmärrystä pohjatiedoista on usein vaikeaa, jopa mahdotonta [18]. Tässä työssä kyseiseen haasteeseen pyritään vastaamaan siten, että tyhjentävän vaatimusmäärittelyn sijaan pyritään alussa saamaan yleisen tason kuva suunnasta, johon projektia aletaan viemään. Projektin edetessä vaatimukset tarkentuvat ja kehitystä voidaan ohjata haluttuun suuntaan.

Tyypillistä on myös se, että erikoistuneiden ohjelmistojen kehityksessä helppokäyttöisyyttä ei pidetä niin tärkeänä kuin kaupallisessa ohjelmistokehityksessä [19]. Esimerkiksi ohjelman tai työkalun tuottamien numeeristen tai graafisten tulosten oikeellisuutta saatetaan pitää ainoana onnistumiskriteerinä varsinkin, jos ohjelman kehittäjät ovat itsekin alan asiantuntijoita ulkoisten ohjelmistokehittäjien sijaan. Tällainen tilanne on itse asiassa yleinen ja koska asiantuntijoilla harvoin on laajaa koulutusta tai kokemusta ohjelmistokehityksen saralta [20],

ohjelmistokehittäjiä tarvitaan viimeistään siinä vaiheessa, kun ohjelmisto halutaan saattaa suuremman joukon käytettäväksi tai konseptitodistustasoa (engl. proof of concept) pidemmälle [21]. Tämän työn toteutuksessa pyritään välttämään tämän käytettävyyden huomiotta jättämisen stereotyypin toteutuminen ja ohjelmalta tavoitellaan alusta asti nimenomaan helpokäyttöisyyttä, tehokkuutta ja yksinkertaisuutta. Tämä onnistuu siksi, että työn alusta alkaen kehittäjän ja asiakkaan välillä käydään tiivistä keskustelua. Kommunikaatio asiakkaan kanssa onkin ensiarvoisen tärkeää ohjelmistokehityksessä, mutta siihen liittyy haasteita. Etenkin tehokkaan kommunikaation saavuttaminen voi olla vaikeaa, ja samanaikaisesti tehon kommunikaatio aiheuttaa usein suuriakin ongelmia ohjelmistokehitysprojekteissa [22], [23].

Keil ja Carmel [24] korostavat suorien asiakas-kehittäjä-linkkien (engl. customer-developer links) merkitystä ohjelmistokehitysprojektien onnistumisessa. Tämän työn puitteissa niiden merkitys korostuu entisestään, sillä kehitettävän ohjelman erikoistuneen luonteen vuoksi tiivistä yhteistyötä ja kommunikointia kehittäjän ja asiakkaan välillä vaaditaan. Kuten vaatimusmäärittelyyn liittyen mainittiin, heti projektin alussa ei todennäköisesti vielä tiedetä tarkalleen, mitä lopputulokselta halutaan vaan vaatimukset tarkentuvat ja muuttuvat kehitystyön edetessä. Näistä syistä pyritään siihen, että asiakkaaseen ollaan projektin aikana yhteydessä mahdollisimman usein. Rampersad et al. [25] kutsuvat tätä jatkuvaa yhteistyötä painottavaa lähestymistapaa osallistuvaksi suunnitteluksi (engl. participatory design) ja ehdottavat sitä mahdolliseksi ratkaisuksi myös käytettävyysoongelmiin erikoisohjelmistoissa.

## 2.2 Käytettävyys ohjelmistoalalla ja sen ulkopuolella

Käytettävyys, eli se, kuinka helppoa jonkin tuotteen tai palvelun käyttäminen on, on nykymaailmassa odotusarvo lähes kaikille tuotteille sekä ohjelmistoalalla että sen ulkopuolella, ja sen tärkeyttä korostetaan enemmän ja enemmän. Esimerkiksi web-kehityksen alalla on ymmärretty, että vaikka internetin kautta voidaan teoreettisesti saavuttaa suuri joukko asiakkaita, käytettävyys on elintärkeää asiakaskunnan kartuttamisessa ja säilyttämisessä. Kukaan ei käytä kauaa sellaista sivustoa, jonka käyttäminen on turhauttavaa. [26]

Käytettävyys ei kuitenkaan liity ainoastaan ohjelmistoalaan. Käytettävyyteen panostetaan nykyisin eri aloilla niin paljon, että kuluttajille on muodostunut olettamus siitä, että kaikki

tuotteet ja palvelut ovat helppokäyttöisiä. Hanin et al. [27] mukaan esimerkiksi kuluttaja-elektroniikan suunnittelussa on painotettu jo kauan käytettävyyden merkitystä, joskin ohjelmistojen ja elektroniikan käytettävyydessä on eroavaisuuksia. He määrittelevät kuluttaja-elektroniikan käytettävyyden ”käyttäjien tyydyttämisenä sekä suorituskyvyn että heidän saamansa kuvan ja vaikutelman puitteissa”, joka pätee myös ohjelmistokehityksessä tavoiteltavaan käytettävyyteen melko hyvin, alojen eroavaisuuksista huolimatta.

Myös esimerkiksi mobiililaitteiden käyttöliittymiin ja etenkin niille kehitettyjen ohjelmistojen standardisointiin on viimeisten vuosikymmenten aikana panostettu todella paljon aikaa ja rahaa muuan muassa Googlen ja Applen kaltaisten teknologiajättiläisten toimesta. Tästä on seurannut yhtenäinen käyttökokemus, helppokäyttöisyys ja intuitiivisuus ja se, että käytännössä kuka tahansa oppii käyttämään mobiililaitteita nopeasti. Mobiililaitteiden käytettävyyden taso yhdistettynä niiden yleisyyteen – Suomessa 88 prosentilla 16-89-vuotiaista oli käytössään älypuhelin vuonna 2021 [28] – on yksi syistä sille, että valtaväestö on tottunut käyttöliittymien helppouteen.

Edellä kuvailtujen kaltaiset tekijät asettavat odotusarvon käytettävyydestä myös tämän kandidaatintyön puitteissa kehitettävälle työkalulle, vaikka kyseessä onkin erikoistunut työkalu, jonka oletettu käyttäjäryhmä on suhteellisen pieni. Historiallisesti vastaavien ohjelmistojen käytettävyyttä ei ole pidetty kovinkaan tärkeänä, mutta aiemmin mainittujen tekijöiden takia nykypäivänä minkä tahansa työkalun tai käyttöliittymän yksinkertaisesti on oltava käytettävä koska käytettävyyden ontuminen johtaa helposti ongelmiin. Hyvänä esimerkkinä tästä toimivat Australian Hobartissa vuonna 2018 käyttöön otetut pysäköintimittarit, joissa oli digitaalinen käyttöliittymä. Voidaan olettaa, että ko. järjestelmää suunniteltaessa käytettävyys oli jätetty vähälle huomiolle, sillä niiden saamaan kritiikkiin kuului muun muassa se, että digitaalista näyttöä oli mahdotonta lukea auringonvalossa. [29]

Tässä työssä käytettävyys pyritään saavuttamaan yksinkertaisen käyttöliittymän kautta, sillä yksinkertaisuus on tärkeä osa käytettävyyden rakentamista [30]. Karvonen [31] esittää artikkelissaan monia yksinkertaisuuden puolesta puhuvia väitteitä, kuten sen, että yksinkertaisemmat käyttöliittymät luovat käyttäjäänsä uskoa ja että kun käyttöliittymän koetaan olevan kaunis tai miellyttävä, sen koetaan todennäköisesti myös olevan helppokäyttöinen. Liittyen erikoistuneisiin ohjelmistoihin, myös Queiroz et al. [32] kehottavat suosimaan minimalismia pitäen kuitenkin mielessä mahdolliset erityistarpeet liittyen näkyvillä olevaan dataan, sekä suunnittelemaan ohjelmistoja tarkkuus edellä. Tämä voi heidän mukaansa tarkoittaa

esimerkiksi sitä, että käyttäjän antamaa syötettä valvotaan tarkasti eikä sallita datan syöttämistä virheellisessä muodossa.

### 2.3 Yhteenveto kirjallisuuskatsauksesta

Erikoisohjelmistojen kehitykseen liittyen kirjallisuudessa painotetaan ylivoimaisesti eniten jatkuvaa, aktiivista kommunikaatiota kehittäjän ja asiakkaan välillä, ettei kompleksinen lähtötieto muodostu ongelmaksi, vaan kehittäjä tietää mitä tekee ja pystyy mm. vahvistamaan aikaansaamansa tulosten oikeellisuuden asiakkaalta. Kommunikaatiolla pyritään takaamaan se, että kehittäjä tietää mitä asiakas haluaa, varsinkin kun ohjelmistokehityksessä, etenkin erikoisohjelmistojen tapauksessa, vaatimukset tarkentuvat usein vasta kehityksen edetessä. Eritoten hyvä yhteydenpito takaa sen, että työtä ohjataan koko projektin ajan asiakkaan haluamaan suuntaan eikä jouduta tilanteeseen, jossa kehittäjä haluaa pitää kiinni tietystä ratkaisusta sen takia, että siihen on kulutettu paljon aikaa, vaikka se ei olisi paras mahdollinen. [33] Tämä tilanne on nimeltään uponneiden kustannusten dilemma (engl. sunk cost dilemma). Tiivis yhteistyö kehittäjän ja asiakkaan välillä voi myös olla ratkaisu vastaaville ohjelmistoille yleisiin käytettävyysongelmiin. Käytettävyys onkin asia, jota nyky-yhteiskunnassa odotetaan lähestulkoon kaikelta, eikä tämän kandidaatintyön puitteissa kehitettävä työkalu ole poikkeus.



## 3 Case: Väsymislaskentatyökalu

Tässä luvussa siirrytään teoriasta käytäntöön. Aluksi esitellään Aholan et al. tutkimus tarpeellisilta osin sekä muut projektin lähtökohdat, kuvaillaan työkalulle valittu toteutustapa ja käydään läpi työkalun toteutuksen eri vaiheet. Tavoitteena on rakentaa hitsiliitosten väsymistä lähtöarvojen perusteella 4R-metodia käyttäen laskeva ja tuloksia kuvaava työkalu LUT-yliopiston konetekniikan osastolle.

### 3.1 Projektin lähtökohdat

Projektin määrittämisvaihe aloitettiin asiakastarpeiden läpikäymisellä. Läpikäynnissä keskusteltiin muun muassa työkalun minimivaatimuksista sekä asiakkaan toiveista liittyen työkalun jakamiseen ja julkaisuun. Konetekniikan osaston toiveena oli se, että työkalua olisi helppo sekä käyttää että jakaa käytettäväksi, jonka myötä päädyttiin selainpohjaiseen toteutukseen sillä se tekee valmiin työkalun jakamisesta helppoa.

Kuten erikoisohjelmistojen kehitysprojekteille todettiin aiemmin olevan tyypillistä, ei tyhjentävää vaatimusmäärittelyä voitu projektin alussa asiakkaan toimesta antaa. Työkalua alettaisiin kehittämään niiden lähtötietojen perusteella, jotka saadaan ja kehitystä tultaisiin ohjaamaan haluttuun suuntaan projektin edetessä. Asiakastarpeiden läpikäynnissä eriteltiin kuitenkin eri ominaisuuksien tärkeyksiä ja muun muassa sitä, kuinka vapain käsin kehittäjä voi suunnitella työkalun ulkoasun. Tärkeimmät läpikäynnistä saadut tiedot olivat:

- 1) Työkalun tulee laskea kestoikäarviot ja piirtää havainnollistavat kuvaajat annettujen lähtöarvojen perusteella. Työkalun ulkoasu tulee näin ollen koostumaan syöttökentistä ja graafisten sekä numeeristen tulosten esityksistä.
- 2) Lähtöarvot tulee voida syöttää sekä syöttökenttien kautta että tiedoston välityksellä. Tiedoston tapauksessa voidaan kuitenkin olettaa, että data on tekstitiedostossa tietyssä muodossa, eikä työkalun näin ollen tarvitse tukea useita tiedostomuotoja. Jos työkalulle syötetty tiedosto ei ole odotetunlainen, se voi yksinkertaisesti ilmoittaa virheestä ja pyytää uutta tiedostoa.

3) Työkalulle syötettävä tiedosto voi sisältää monta riviä, kullakin rivillä yksi joukko lähtöarvoja. Tässä tapauksessa tulokset tulee laskea ja kuvaajat piirtää jokaiselle lähtöarvojoukolle. Manuaalisen lähtöarvojen syötön tapauksessa lasketaan tulos ja piirretään kuvaajat vain syötetyille arvoille.

4) Osa kuvaajista piirretään vain tietyllä alueella, ja tiettyjen kuvaajien leikkauspisteitä tulee korostaa.

### 3.2 Projektissa tarvittavan koneteknisen teorian esittely

Aholan et al. tutkimuksessa esitellään hitsien väsymislaskentaa 4R-metodia käyttäen. Koko metodin teorian selittäminen, yhtälöiden johtaminen tai niiden toimivuuden perustelu ei kuulu tämän kandidaatintyön tavoitteisiin, vaan teoriasta esitetään ne vaiheet, jotka ovat merkityksellisiä työkalun kehittämisen kannalta i.e. ne vaiheet, jotka ohjelman täytyy suorittaa pystyäkseen tuottamaan oikeita lopputuloksia. Esitellään ensin tarvittavat yhtälöt ja niissä esiintyvät symbolit. Toiston välttämiseksi kukin symboli esitellään ainoastaan sen yhtälön yhteydessä, jossa se esiintyy ensimmäisen kerran.

#### 3.2.1 Yhtälöt

$$\sigma_k = \frac{\Delta\sigma_k(r)}{1 - R} \quad (1)$$

jossa  $\sigma_k$  on maksimilovijännitys,  $\Delta\sigma_k(r)$  on lovijännitysvaihtelu ja  $R$  on ulkoisen kuormituksen jännityssuhde.

$$f(\sigma) = \frac{\sigma}{E} + \left(\frac{\sigma}{H}\right)^{\frac{1}{n}} - \frac{(\sigma_k + \sigma_{res})^2}{\sigma E} \quad (2)$$

jossa  $\sigma$  on jännitys,  $E$  on materiaalin kimmokerroin,  $H$  on Ramberg-Osgoodin muokkauslujittumisvakio,  $n$  on Ramberg-Osgoodin muokkauslujittumiseksponentti ja  $\sigma_{res}$  on jäännösjännitys.

$$f'(\sigma) = \frac{1}{E} + \frac{1}{n} \cdot \frac{\sigma^{\frac{1}{n}-1}}{H^{\frac{1}{n}}} + \frac{(\sigma_k + \sigma_{res})^2}{\sigma^2 E} \quad (3)$$

joka on yhtälön (2) ensimmäinen derivaatta  $\sigma$ :n suhteen.

$$y(\Delta\sigma) = \frac{\Delta\sigma}{E} + 2 \left( \frac{\Delta\sigma}{2H} \right)^{\frac{1}{n}} - \frac{\Delta\sigma_k(r)^2}{\Delta\sigma E} \quad (4)$$

$$y'(\Delta\sigma) = \frac{1}{E} + \frac{2}{n} \cdot \frac{\Delta\sigma^{\frac{1}{n}-1}}{(2H)^{\frac{1}{n}}} + \frac{\Delta\sigma_k(r)^2}{\Delta\sigma^2 E} \quad (5)$$

joka on yhtälön (4) ensimmäinen derivaatta  $\Delta\sigma$ :n suhteen.

$$\sigma_{\min} = \sigma_{\max} - \Delta\sigma \quad (6)$$

jossa  $\sigma_{\max}$  ja  $\sigma_{\min}$  ovat paikallisen jännityksen maksimi- ja minimiarvot.

$$R_{local} = \frac{\sigma_{\min}}{\sigma_{\max}} \quad (7)$$

jossa  $R_{local}$  on paikallinen jännityssuhde.

$$\Delta\sigma_{k,ref} = \frac{\Delta\sigma_k(r)}{\sqrt{1 - R_{local}}} \quad (8)$$

jossa  $\Delta\sigma_{k,ref}$  on keskijännityskorjattu referenssilovijännitys.

$$N_f = \frac{C}{\Delta\sigma_{k,ref}^m} \quad (9)$$

jossa  $N_f$  on kestoikä ja  $C$ ,  $m$  ovat tunnettuja vakioarvoja.

Funktioiden (10) – (13) arvoja käytetään graafien piirrossa:

$$\varepsilon = \frac{\sigma}{E} + \left(\frac{\sigma}{H}\right)^{\frac{1}{n}} \quad (10)$$

$$\Delta\varepsilon = \frac{\Delta\sigma}{E} + 2\left(\frac{\Delta\sigma}{2H}\right)^{\frac{1}{n}} \quad (11)$$

$$\varepsilon = \frac{\left(\frac{\Delta\sigma_k(r)}{1-R} + \sigma_{res}\right)^2}{\sigma E} \quad (12)$$

$$\Delta\varepsilon = \frac{\Delta\sigma_k^2}{\Delta\sigma E} \quad (13)$$

Laskennan toiminta voidaan varmistaa seuraavalla yhtälöllä:

$$\frac{\Delta\sigma_k(r)}{1-R} + \sigma_{res} > 0 \quad (14)$$

, jossa jos suuremmusehto ei toteudu, niin lähtöarvot ovat virheelliset.

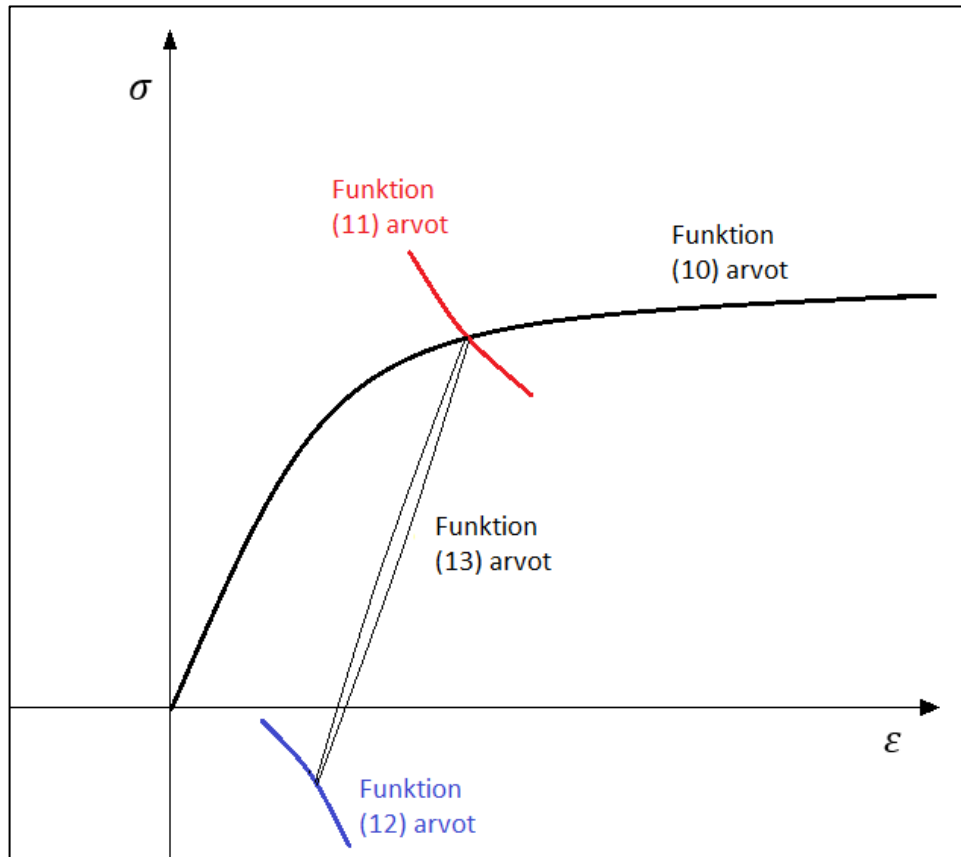
### 3.2.2 Laskenta työkalun taustalla

Seuraavaksi esitellään vaiheet, joilla lähtöarvoista päästään kestoikätulokseen 4R-metodia käyttäen. Tunnetut lähtöarvot laskennalle ovat  $\Delta\sigma_k(r)$ ,  $H$ ,  $n$ ,  $\sigma_{res}$ ,  $E$ ,  $C$ ,  $m$  ja  $R$ . Laskennan ensimmäinen vaihe on  $R_{local}$ -arvon laskeminen, joka tapahtuu siten, että funktiota (1) käyttäen lasketaan maksimilovijännitys  $\sigma_k$ , jonka jälkeen voidaan sijoittaa arvot funktioihin (2) ja (3). Arvo  $\sigma_{max}$  ratkaistaan soveltamalla Newtonin menetelmää [34] ko. funktioille.

Tämän jälkeen Newtonin menetelmää sovelletaan funktioille (4) ja (5), jonka tuloksena saadaan arvo  $\Delta\sigma$ , joka on paikallisen jännityksen vaihtelu. Sitten voidaan kaavaa (6) käyttämällä ratkaista arvo  $\sigma_{min}$ , jolloin tiedetään kaikki tarvittava  $R_{local}$ -arvon ratkaisuuun, joka tapahtuu käyttäen kaavaa (7). Kun  $R_{local}$ -arvo tunnetaan, voidaan laskea kestoian ratkaisussa

tarvittava  $\Delta\sigma_{k,ref}$ -arvo käyttämällä kaavaa (8) ja lopulta kestoikä  $N_f$  ratkaistaan kaavan (9) mukaisesti. Kestoikien laskennan lisäksi piirretään neljä kuvaajaa käyttäen hyväksi yhtälöitä (10) – (13). Koordinaatiston piirrettyine kuvaajineen tulee näyttää, lähtöarvoista riippuen, noin kuvan 1 mukaiselta.

Kuva 1. Hahmotelma piirrettävistä kuvaajista.



### 3.3 Valittu toteutustapa

Työkalu toteutetaan JavaScriptin React-kirjastoa käyttäen. Valinta perustuu seuraaviin tekijöihin:

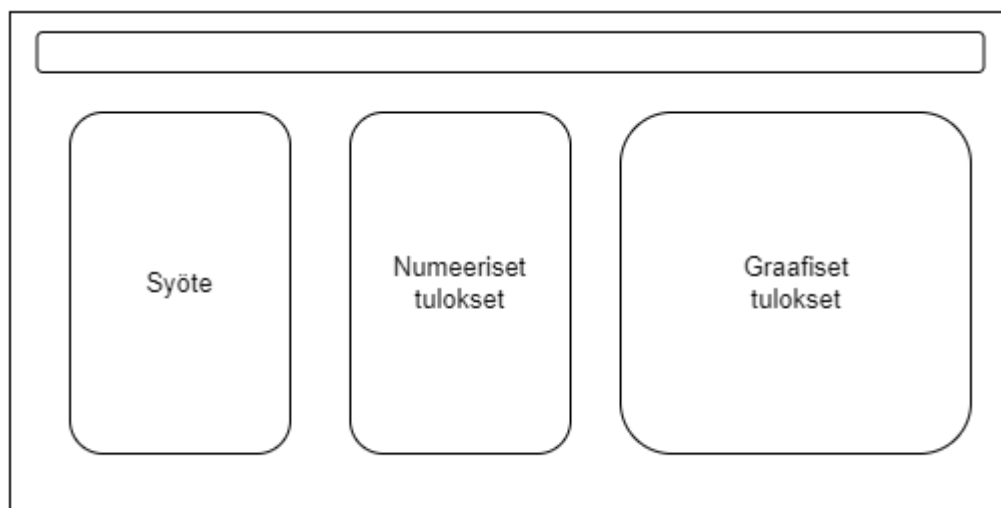
- 1) Työkalusta tehdään selainpohjainen helpon jaettavuuden takia. JavaScript-kirjastona React sopii tähän tarkoitukseen hyvin, lisäksi työn tekijällä on eniten kokemusta web-kehityksestä Reactilla.

2) React on yksi tämän hetken suosituimmista käyttöliittymäkirjastoista [35]. Tämän ansiosta siihen liittyvää tietoa on saatavilla helposti, kuten myös valtava määrä käyttäjien kehittämiä kirjastoja, jotka mahdollistavat erilaisten toimintojen implementoinnin, kuten vaikkapa kuvaajien piirtämisen, jota tässä työssä tullaan tarvitsemaan.

### 3.4 Toteutusprosessi

Työkalun toteutus aloitettiin suunnittelemalla työkalun rakennetta asiakkaalta saatujen toiminnallisten vaatimusten perusteella. Työkalun toiminta koostuu kolmesta osasta: käyttäjän syötteestä eli laskennan lähtöarvoista, numeerisista tuloksista sekä graafisista tuloksista. Näin ollen, sekä Reactin komponentteihin perustuvan luonteen takia, oli luontevaa, että työkalussa tulisi olemaan kolme pääkomponenttia, joista yksi ottaa vastaan käyttäjän syötteen, toinen esittää numeeriset ja kolmas graafiset tulokset. Näistä komponenteista kaksi viimeistä tulisivat olemaan poissa näkyvistä, kunnes käyttäjä on antanut syötteen. Kolmen pääkomponentin lisäksi esimerkiksi virheilmoitusten näyttämistä varten voitaisiin toteuttaa myös yksinkertainen komponentti. Kuvassa 2 esitetään hahmotelma työkalun rakenteesta.

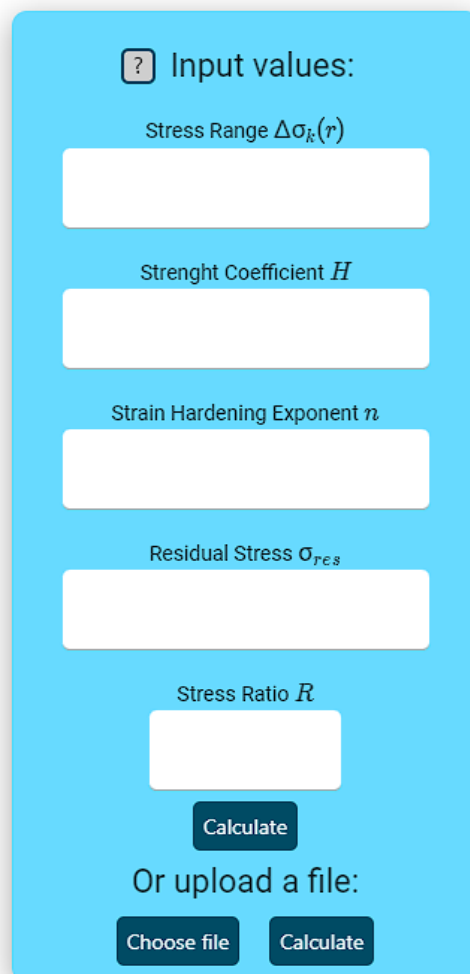
Kuva 2. Suunnitelma työkalun rakenteesta.



Kun suunnitelma työkalun rakenteesta oli valmis, alettiin suunniteltuja komponentteja toteuttaa. Tämä alkoi syötekomponentista, johon rakennettiin kentät lähtöarvojen manuaalista syöttämistä varten sekä vaihtoehto palauttaa tiedosto. Syötettäviä lähtöarvoja ovat  $\Delta\sigma_k(r)$ ,

$H$ ,  $n$ ,  $\sigma_{res}$  ja  $R$ . Käyttäjän syötettä rajattiin tässä ainoastaan arvon  $R$  osalta, sillä se voi saada arvoja ainoastaan välillä  $[0, 1[$  ja arvon 1 syöttäminen sille aiheuttaisi nolllalla jakamisen yhtälöissä (1) ja (12). Muita syötekenttiä ei rajattu syötteen tyyppiä (kokonaisluku, desimaaliluku) enempää, sillä jos syöte on virheellinen, voidaan yhtälön (14) tuloksen avulla ilmoittaa käyttäjälle, jos syöte on virheellinen hänen yrittäessä suorittaa laskennan. Syötekomponentti esitetään kuvassa 3.

Kuva 3. Syötekomponentti.



? Input values:

Stress Range  $\Delta\sigma_k(r)$

Strength Coefficient  $H$

Strain Hardening Exponent  $n$

Residual Stress  $\sigma_{res}$

Stress Ratio  $R$

Calculate

Or upload a file:

Choose file Calculate

Numeeristen tulosten komponentin rakentamista varten tuli luonnollisesti toteuttaa luvussa 3.2.2 kuvailtu laskentaprosessi työkalussa, jotta esitettävät tulokset voitaisiin laskea. Laskennan toteuttaminen oli työn ensimmäinen suurempi kokonaisuus, jonka aikana pidettiin tiivistä yhteyttä konetekniikan osaston edustajiin, kuten kirjallisuudessa kehoitettiin. Kirjallisuus vastasi tässä suhteessa käytäntöä hyvin; yhteydenpito osoittautui erittäin tärkeäksi

tekijäksi toteutuksen etenemisessä, sillä asiakkaan jatkuvaa palautetta ja neuvoja tarvittiin sekä väsymislaskennan teorian ymmärtämisessä että tulosten oikeellisuuden tarkastamisessa sitä mukaa kun niitä saatiin. Tässä auttoi myös se, että kandidaatintyön tekijällä on kokemusta konetekniikasta sivuaineen kautta. Asiakassuhteen tärkeyden lisäksi toinen laskennan toteuttamiselle ominainen tekijä oli matemaattisten yhtälöiden koodiin siirtämisen ja etenkin mahdollisten virheiden alkuperän etsinnän vaikeus. Esimerkkejä tästä ovat se, että JavaScript ei tue lainkaan negatiivisia kantalukuja eksponenteille, esimerkiksi  $-2^2$  tulee korvata  $-(2^2)$ :lla ja se, että yhtälöjä muotoillessa koodiin suluilla on paljon suurempi merkitys kuin paperilla. Kuvassa 4 annetaan esimerkki yhtälöiden eroavaisuuksista koodissa ja sen ulkopuolella.

Kuva 4. Yhtälön (11) muoto matemaattisena kaavana ja JavaScript-koodina.

$\Delta\varepsilon = \frac{\Delta\sigma}{E} + 2 \left( \frac{\Delta\sigma}{2H} \right)^{\frac{1}{n}}$	<pre>function function_3(x) {   let value = x / E + 2 * ((x / (2 * H))) ** (1 / n);   return value; }</pre>
---	---

Kun laskenta oli toteutettu koodissa ja tulokset vahvistettu, tuli toteuttaa komponentti, joka näyttää syötteestä riippuen joko listan tuloksista tai yhden tuloksen. Konetekniikan osastolta saatiin tässäkin kohtaa arvokasta palautetta, he tarkensivat, että myös kaksi kuhunkin tulokseen liittyvää välitulosta,  $R_{local}$  ja  $\Delta\sigma_{k,ref}$  tulee näyttää varsinaisten kestoikätulosten  $N_f$  yhteydessä. Komponentin toteutus sujui muutoin mutkitta, mutta alaindeksien esittäminen tuloksissa osoittautui yllättävän vaikeaksi. Tähänkin ongelmaan löytyi kuitenkin ratkaisu, kirjasto react-latex [36], joka on suunniteltu matemaattisten yhtälöiden esittämiseen. Kuvassa 5 esitellään valmista tuloskomponenttia kuudella lähtöarvojoukolla.

Kuva 5. Tuloskomponentti.

Results:		
$R_{local} = -0.56$	$\Delta\sigma_{k,ref} = 488.65$	$N_f = 125705$
$R_{local} = -0.56$	$\Delta\sigma_{k,ref} = 492.27$	$N_f = 120399$
$R_{local} = -0.38$	$\Delta\sigma_{k,ref} = 408.20$	$N_f = 360100$
$R_{local} = -0.37$	$\Delta\sigma_{k,ref} = 371.12$	$N_f = 628564$
$R_{local} = -0.53$	$\Delta\sigma_{k,ref} = 424.02$	$N_f = 288246$
$R_{local} = -0.53$	$\Delta\sigma_{k,ref} = 421.27$	$N_f = 299437$



Seuraavaksi siirryttiin graafisten tulosten esittämiseen. Kuvaajien piirtämistä varten tuli etsiä valmis kirjasto, sillä koko piirto-ominaisuuden toteuttaminen alusta alkaen olisi vienyt koh- tuuttoman paljon aikaa. Etsittävältä kirjastolta vaadittiin joko sitä, että se pystyy piirtämään kuvaajat pelkkien funktioiden perusteella, tai sitä että se pystyy muodostamaan kuvaajat funktioille lasketuista arvoista.

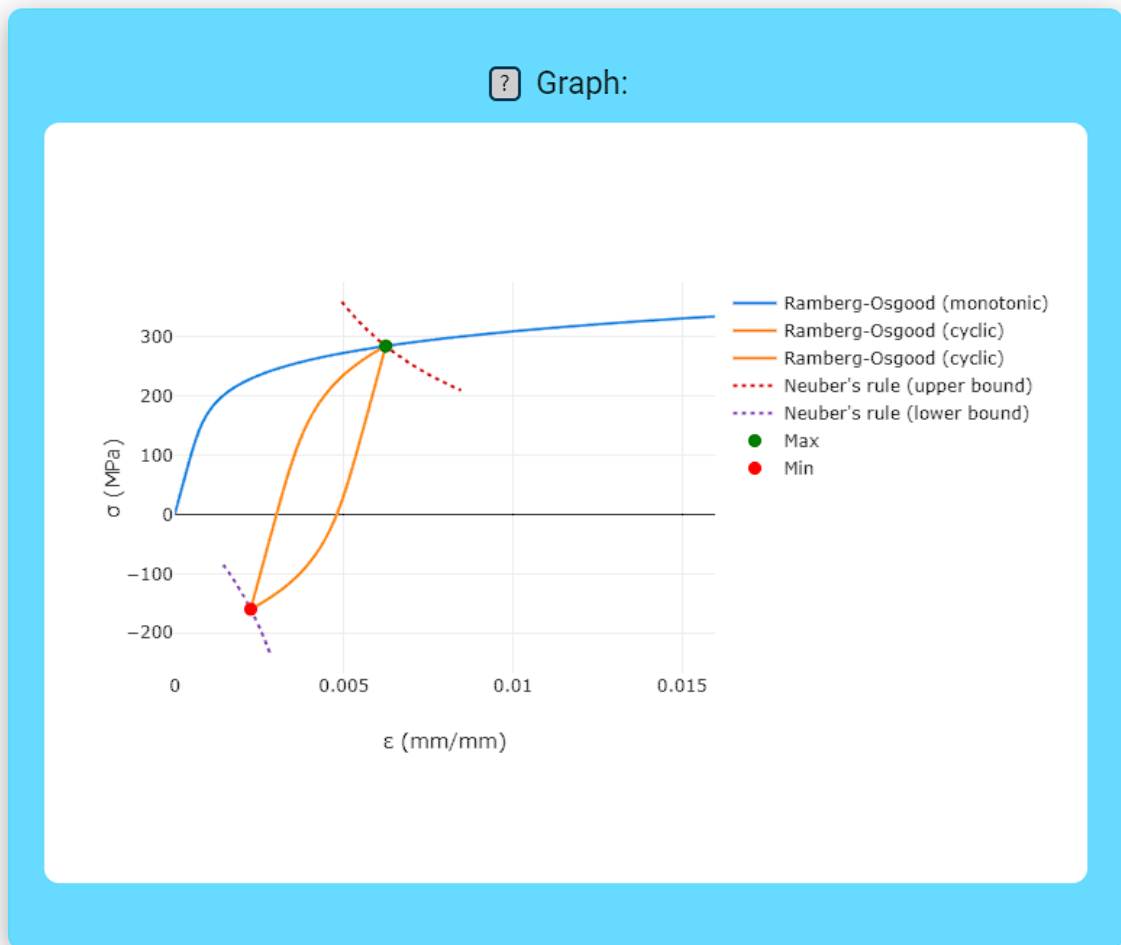
Kirjasto function-plot [37] täytti vaatimuksista ensimmäisen, eli se pystyi muodostamaan kuvaajat yhtälöiden perusteella ilman arvojen laskemista erikseen. Lisäksi ko. kirjastossa pystyttiin määräämään akselien määrittelyalueet, eli arvojoukot kummallekin akselille. Tämä on tärkeää, sillä jotta kuvaajista voi saada selvää, tulee x-akselilla olla arvoja  $n$ . väliltä  $[0, 0.1]$  ja y-akselilla  $n$ . väliltä  $[-100, 500]$ . Kyseisen kirjaston käytössä kohdattiin kuitenkin seuraavanlainen ongelma: piirrettävät funktiot ovat muotoa  $\varepsilon = f(\sigma)$ , mutta kuvaajat tuli piirtää  $\varepsilon$ ,  $\sigma$ -koordinaatistoon, eli akselit olisi tullut peilata, mutta siihen kirjasto ei tarjonnut toiminnallisuutta. Tämän lisäksi yhtälöiden (3) ja (5) kohdalla kuvaajat haluttiin piirtää vain tietyllä alueella, joka ei myöskään vaikuttanut olevan mahdollista Function Plot:lla. Siispä sopivan kirjaston etsintää jatkettiin, tavoitteena löytää toisen vaatimuksen täyttävä kirjasto, eli sellainen, jolla voidaan piirtää laskettujen arvojen joukon perusteella kuvaaja. Tällöin kuvaajien pelaaminen ja piirtäminen vain tietyltä väliltä onnistuisi helpommin.

Seuraavaksi löytynyt kirjasto Plotly.js [38] vaikutti sopivan tarkoitukseen hyvin, sillä sen avulla kyettiin piirtämään kuvaajat arvojoukkojen perusteella. Koska graafien kunkin pis- teen x- ja y-koordinaatit syötetään itse, onnistui myös akselien pelaaminen, eli  $\sigma$ -arvojen asettaminen x-akselille ja  $\varepsilon$ -arvojen y-akselille sekä yhtälöiden (3) ja (5) kuvaajien piirto vain tietyltä väliltä. Lisäksi akselien määrittelyalueiden asettaminen onnistui, ja kirjasto tar- joaa myös paljon ominaisuuksia käyttäjälle, joihin lukeutuu muun muassa koordinaatiston zoom-tason säätäminen, koordinaatistossa liikkuminen kursoria käyttäen sekä PNG-muotoi- sen kuvan lataamisen koordinaatistosta. Näiden tekijöiden takia kuvaajien piirto toteutettiin Plotly.js:llä.

Myös tässä työvaiheessa asiakassuhteen merkitys oli suuri, sillä ilman palautetta konetekniikan osastolta ei olisi voitu olla varmoja siitä, ovatko kuvaajat täysin oikeannäköisiä. Tästä syystä tiivis kommunikaatio tapahtui luonnollisesti, sillä säännöllisiä kommentteja vaadit- tiin, jottei kehitys lähtenyt väärille raiteille. Lisäksi kirjastojen haussa oli aluksi ongelmia siksi, että sellaisia kirjastoja, jotka piirtävät kuvaajan pelkän yhtälön perusteella on ilmeisesti olemassa melko vähän, kun taas valmiiksi laskettujen pisteiden näyttämiseen

koordinaatistossa kykenee lähes mikä tahansa kaaviokirjasto. Tämä kielii siitä, että JavaScriptiä tai Reactia ei käytetä kovinkaan paljoa tällaisiin projekteihin. Valmiit kuvaajat näyttävät eräillä alkuarvoilla kuvan 6 mukaisilta ja vastaavat täten aiemmin kuvassa 1 esitettyä hahmotelmaa.

Kuva 6. Valmiit, Plotly.js:llä toteutetut kuvaajat kuvaajakomponentissa.



Kun valmiita kuvaajia esiteltiin konetekniikan osastolle, tunnistettiin lisävaatimus työkalulle. Jos käyttäjä syöttää tiedoston, jossa on monta lähtöarvojoukkoa, tulee kuvaajat piirtää niistä jokaiselle. Lähtöarvojoukkoja voi kuitenkin olla käytännössä kuinka suuri määrä tahansa, joten niiden näyttäminen käyttäjälle samanaikaisesti ei olisi järkevää. Niinpä päätettiin toteuttaa ominaisuus, jonka myötä kutakin lähtöarvojoukkoa vastaavan tuloksen klikkaaminen tuloskomponentissa näyttää ko. tuloksen kuvaajan. Tämän tavan todettiin olevan intuitiivisempi kuin pudotusvalikko tai muu komponentti, josta kuvaajat valittaisiin.

Kuvaajakomponentin toteutuksen jälkeen oli päästy tilanteeseen, jossa jokainen kolmesta suunnitellusta pääkomponentista oli valmis. Kehitysprosessin aikana oli kuitenkin havaittu, että olisi järkevää kehittää myös yksinkertainen komponentti, jossa voitaisiin näyttää mahdollisia virheitä käyttäjälle. Yleisimmät työkalun kohtaamat virheet voisivat liittyä esimerkiksi virheellisiin lähtöarvoihin tai vääränlaisen tiedoston syöttämiseen. Koska virhekomponentin ei tarvitse tehdä muuta kuin näyttää käyttäjälle kuhunkin virheeseen liittyvä teksti, ei sen kehittäminen ollut vaikeaa. Kuvassa 7 esitetään syöte- ja virhekomponentti tilanteessa, jossa käyttäjä on syöttänyt vääränlaisen tiedoston työkalulle. Virhekomponentti ilmoittaa tilanteesta ja tarjoaa käyttäjälle mahdollisuuden ladata esimerkkitiedosto, jonka rakenne on oikeanlainen työkalun käsiteltäväksi.

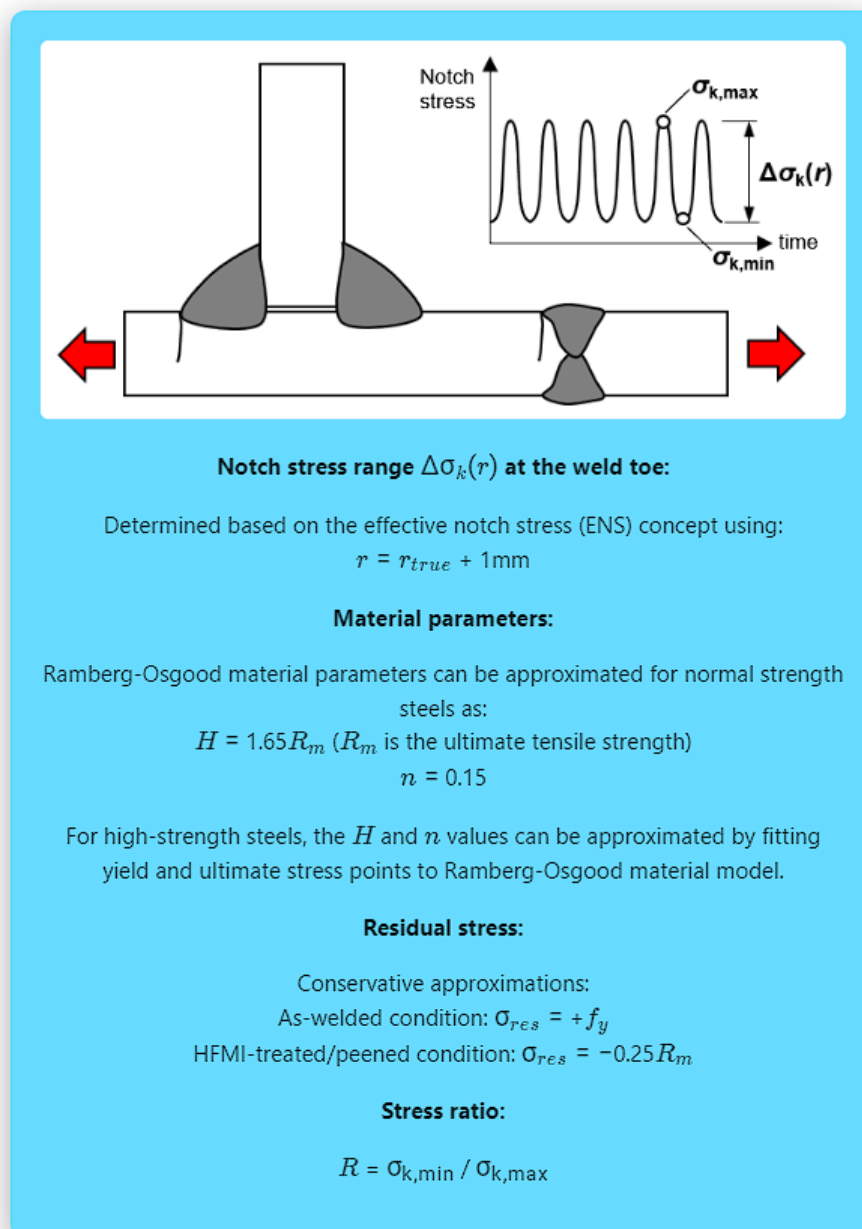
Kuva 7. Virhetilanne ja -ilmoitus.

The image shows a software interface with a light blue background. At the top, there is a section titled "Input values:" with a question mark icon. Below this, there are five input fields, each with a label: "Stress Range  $\Delta\sigma_R(r)$ ", "Strength Coefficient  $H$ ", "Strain Hardening Exponent  $n$ ", "Residual Stress  $\sigma_{res}$ ", and "Stress Ratio  $R$ ". Below the input fields is a "Calculate" button. At the bottom of the blue area, there is a section titled "Or upload a file:" with a "Choose file" button, the filename "wrong-input-test.txt", and another "Calculate" button. To the right of the blue area, there is an orange error message box with the text: "The file wasn't of the correct format. It should be a text file with 5 values on each line, separated by tab-characters. Try another file." Below the error message is a "Download example file" button.

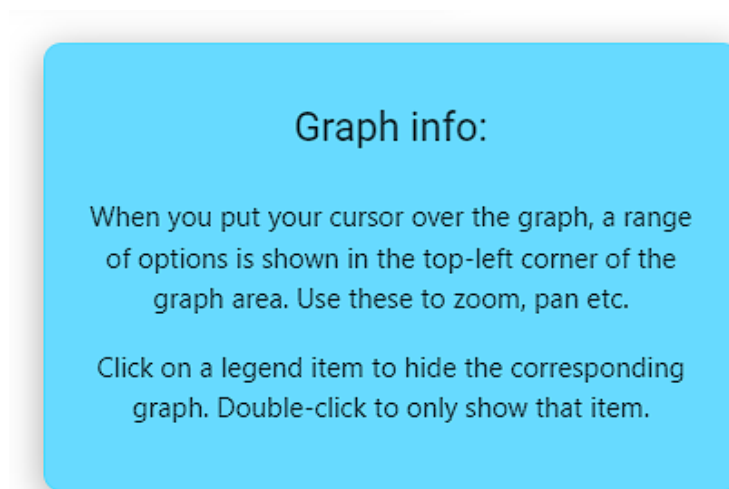
Virhekomponentin lisäksi tunnistettiin konetekniikan osaston edustajien toimesta tarve infokomponentille, johon liittyvät syöte- ja kuvaajakomponenteissa näkyvät kysymysmerkkipainikkeet. Samaa infokomponenttia käytettäisiin kahdesti eri sisällöillä, sillä lisätietoja haluttiin näytettävän sekä syötteeseen että kuvaajiin liittyen. Näihin lisätietoihin kuului muun

muassa tarkempi kuvaus siitä, missä muodossa käyttäjän tulee syöttää tiedot ja miten kuvaajakomponenttia käytetään. Infokomponentin luonne on yksinkertainen, sen tulee ainoastaan näyttää lisätiedot käyttäjälle, jos sitä vastaava kysymysmerkinappia on painettu. Kuvassa 8 esitetyn syötteeseen liittyvän infokomponentin sisältö saatiin suoraan konetekniikan osastolta, joten kehittäjän ei tarvinnut itse miettiä, millä tavalla asiat tulisi käyttäjälle selittää. Kuvaajien infokomponenttiin (kuva 9) taas avattiin muutamalla lauseella sitä, miten kuvaajakomponenttia käytetään.

Kuva 8. Syötteen infokomponentti.



Kuva 9. Kuvaajien infokomponentti.



Kun infokomponentit oli saatu toteutettua, oli työkalun rakenne valmis. Tässä vaiheessa olttiin yhteydessä asiakkaaseen ja esiteltiin heille lopputulosta. Koska asiakas oli tyytyväinen, oli kehitysprosessi valmis ja projekti siirtyi seuraavaksi julkaisuvaiheeseen, joka ei kuulu tämän kandidaatintyön puitteisiin, koska sen toteuttaa asiakas. Seuraavalla sivulla (kuva 10) esitetään kokoelma kuvakaappauksia valmiista työkalusta.

Kuva 10. Kuvankaappauksia valmiista työkalusta.

**Input values:**

Stress Range  $\Delta\sigma_k(r)$

Strength Coefficient  $H$

Strain Hardening Exponent  $n$

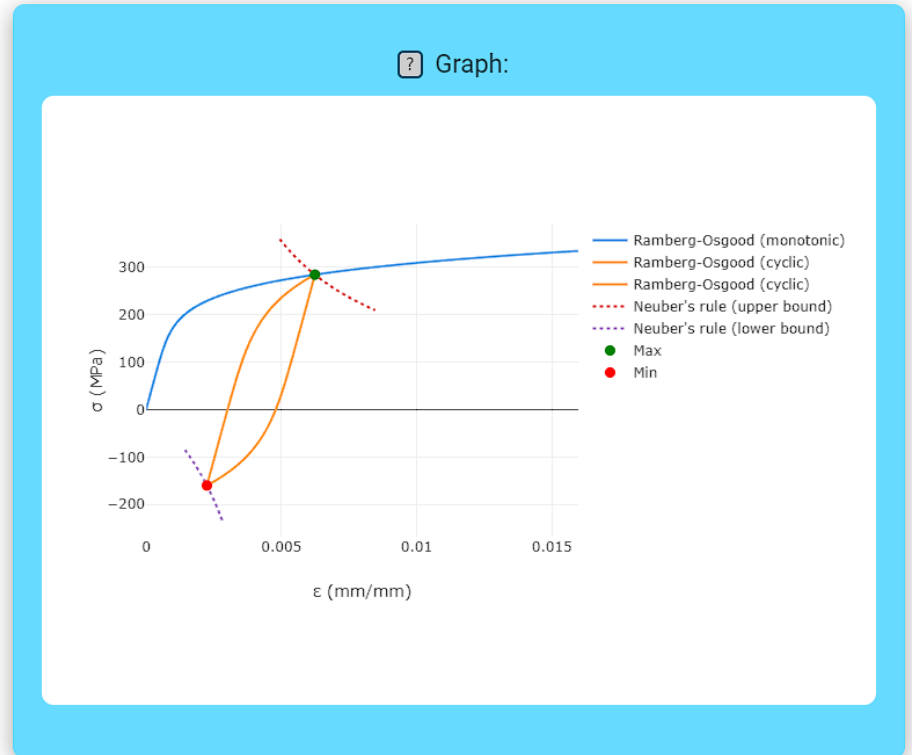
Residual Stress  $\sigma_{res}$

Stress Ratio  $R$

**Calculate**

Or upload a file:

**Choose file** Kanava – kopio.txt **Calculate**



The inputted values aren't realistic or some of them were empty.

[Download example file](#)

The file wasn't of the correct format. It should be a text file with 5 values on each line, separated by tab-characters. Try another file.

[Download example file](#)

### Results:

$R_{local} = -0.56$	$\Delta\sigma_{k,ref} = 488.65$	$N_f = 125705$
$R_{local} = -0.56$	$\Delta\sigma_{k,ref} = 492.27$	$N_f = 120399$
$R_{local} = -0.38$	$\Delta\sigma_{k,ref} = 408.20$	$N_f = 360100$
$R_{local} = -0.37$	$\Delta\sigma_{k,ref} = 371.12$	$N_f = 628564$
$R_{local} = -0.53$	$\Delta\sigma_{k,ref} = 424.02$	$N_f = 288246$
$R_{local} = -0.53$	$\Delta\sigma_{k,ref} = 421.27$	$N_f = 299437$

## 4 Prosessin reflektointi

Tämän työn kirjallisuuskatsausvaiheessa kerätyt neuvot ja ohjeet erikoisohjelmien kehitysprojektia varten osoittautuivat arvokkaiksi, etenkin tiivis kommunikointi asiakkaan kanssa oli ratkaisevan tärkeää projektin onnistumisen kannalta. Asiakassuhteen merkitystä korostettiin paljon myös kirjallisuudessa, joten kirjallisuuden voidaan todeta vastaavan käytäntöä tältä osin hyvin. Sama pätee myös siihen, että kehitysprosessin alussa ei pystytty antamaan seikkaperäistä vaatimusmäärittelyä toteutettavalle työkalulle, vaan kehitysprojektia ohjattiin oikeaan suuntaan sen edetessä. Tämä oli asiakassuhteen lisäksi toinen yleisimmistä asioista, jota kirjallisuudessa painotettiin. Eikä ihme, sillä asiakassuhde ja projektin ohjaaminen kulkevat luonnollisesti käsi kädessä, koska ilman tiivistä asiakassuhdetta ei kehitysprosessia voitaisi ohjata haluttuun suuntaan.

Kirjallisuuskatsauksessa etsittiin tietoa liittyen myös käytettävyyteen. Sen tuloksena ymmärrettiin, että käytettävyys on tänä päivänä odotusarvo kaikille tuotteille ja palveluille sekä ohjelmistoalalla että sen ulkopuolella. Tästä syystä käytettävyys otettiin huomioon myös tämän työn puitteissa kehitettävässä työkalussa, vaikka kyseessä onkin asiantuntijoiden käyttöön kehitetty ohjelmisto. Tämä saavutettiin siten, että työkalu pidettiin yksinkertaisena: kullekin toiminnallisuudelle tehtiin oma, selkeä komponenttinsa. Komponenttien kokonaismäärä pidettiin myös alhaisena.

Toteutettu väsymislaskentatyökalu valmistui aikataulussa, kevään 2022 aikana ja vastaa asiakkaan projektille asettamia odotuksia. Kehitysprosessin aikana ei kohdattu sellaisia haasteita, jotka olisivat vaarantaneet toteutuksen onnistumisen, vaan prosessi eteni alusta loppuun asti sulavasti. Suurin tähän vaikuttanut tekijä oli jo aiemmin mainittu tiivis asiakassuhde, jota ilman projekti ei olisi onnistunut. Pienempi, mutta varmasti osaltaan projektin onnistumista edistänyt tekijä oli se, että kehittäjällä oli kokemusta konetekniikasta sivuaineopintojen muodossa.

Kirjallisuudessa painotettiin asiakassuhteen korostunutta merkitystä erikoisohjelmistojen kehityksen erityispiirteenä, mutta avoimeksi kysymykseksi jää se, että eroaako tämä muista ohjelmistoprojekteista mitenkään? Eikö asiakassuhde ole tärkeä osa mitä tahansa nykypäivän ketterää ohjelmistokehitysprojektia? Lisäksi koska kyseessä oli yhden tapauksen

tapaustutkimus, ei tuloksia voida yleistää koskemaan kaikkia vastaavanlaisia projekteja. Siitä huolimatta työn tulokset vaikuttaisivat vahvistavan kirjallisuuden näkemystä prosessin jatkuvan ohjauksen sekä sen vaatiman asiakassuhteen tärkeydestä erikoistuneiden ohjelmistojen kehitysprojekteissa.



## 5 Yhteenveto

Tämä kandidaatintyö alustettiin kertomalla nykymaailman digitalisaatiosta ja antamalla esimerkkejä siitä, miten digitalisaatio konkreettisesti näyttäytyy yhteiskunnassa. Todettiin, että tiedon saattaminen saavutettavaan ja käytettävään muotoon on tärkeää nykypäivänä. Työ toteutettiin tapaustutkimuksena, jossa esimerkkinä tiedosta, joka voidaan saattaa käytännölliseen muotoon toimi LUT-yliopiston konetekniikan osastoon kuuluvien Aholan et al. tutkimus. Kyseisessä tutkimuksessa oli kehitetty nk. 4R-metodi, jolla voi laskea teräshitsien väsymistä. 4R-metodin saattaminen selainpohjaiseksi, helppokäyttöiseksi työkaluksi oli tämän työn tapaustutkimuksen kohde.

Tutkimus alkoi kirjallisuuskatsauksella, jossa perehdyttiin erikoisohjelmistojen kehitykseen liittyviin erityispiirteisiin ja haasteisiin, niihin vastaamiseen sekä ohjelmistotuotteiden käytettävyyteen. Kirjallisuuskatsauksen tuloksena ymmärrettiin, että etenkin tiiviillä asiakassuhteella ja kehityksen jatkuvalla ohjaamisella on suuri merkitys erikoisohjelmistojen kehityksessä. Käytettävyyteen liittyen taas todettiin, että vaatimus käytettävyydestä koskee nykyään lähes kaikkia tuotteita ohjelmistoalalla ja sen ulkopuolella. Tästä syystä myös tämän työn puitteissa kehitetyn työkalun tulisi olla helppokäyttöinen, vaikka vastaavanlaisten, pienelle käyttäjäryhmälle spesifiin käyttötapaukseen suunniteltujen ohjelmistojen käytettävyyttä ei ole perinteisesti pidetty kovin tärkeänä.

Kirjallisuuskatsauksesta siirryttiin varsinaiseen kehitykseen. Prosessin alussa asiakas ei pystynyt antamaan tyhjentävää vaatimusmäärittelyä, joka myös mainittiin kirjallisuudessa erikoisohjelmistojen kehityksen ominaispiirteeksi. Kehitysprojekti aloitettiin niillä tiedoilla, jotka alussa saatiin ja sitä ohjattiin tiiviillä asiakaskommunikaatiolla sen edetessä. Kommunikaatio oli tärkeää myös 4R-metodin teorian ymmärtämisessä. Työkalu valmistui ajallaan ja konetekniikan osasto oli lopputulokseen tyytyväinen. Prosessi sujui pitkälti siten, miten kirjallisuus antoi olettaa; tiiviillä kommunikaatiolla taattiin projektin onnistuminen.

Työn tulokset vaikuttavat todistavan sen, että tiivis asiakassuhde on elintärkeä osa erikoisohjelmistojen kehitystä. Tuloksiin täytyy kuitenkin suhtautua varauksella, sillä yhden tapauksen tapaustutkimuksen tulosten ei voida olettaa kuvaavan erikoisohjelmistojen kehitysprosesseja yleispätevästi.

## Lähteet

- [1] V. Palacin, S. Gilbert, S. Orchard, A. Eaton, M. A. Ferrario, ja A. Happonen, ”Drivers of Participation in Digital Citizen Science: Case Studies on Järviwiki and Safecast”, *Citiz. Sci. Theory Pract.*, vsk. 5, nro 1, ss. 1–20, 2020, doi: 10.5334/cstp.290.
- [2] H. Piili *ym.*, ”Digital Design Process and Additive Manufacturing of a Configurable Product”, *Adv. Sci. Lett.*, vsk. 19, nro 3, ss. 926–931, 2013, doi: 10.1166/asl.2013.4827.
- [3] T. Widmaier *ym.*, ”Digital design and manufacturing process comparison for new custom made product family – a case study of a bathroom faucet”, *Est. J. Eng.*, vsk. 19, nro 1, ss. 76–89, 2013, doi: 10.3176/eng.2013.1.07.
- [4] A. Vatousios ja A. Happonen, ”Renewed talent management: more productive development teams with digitalization supported HR tools”, *Int. J. Eng. Technol.*, vsk. 10, nro 2, ss. 170–180, 2021, doi: 10.14419/ijet.v10i2.31705.
- [5] A. Happonen, L. Manninen, M. Hirvimäki, ja A. Nolte, ”Expectations for young job applicants’ digital identity related to company’s social media brand development strategies”, *Small Enterp. Res.*, vsk. 10, nro 2, ss. 170–180, 2021, doi: 10.1080/13215906.2021.2000482.
- [6] H. Kortelainen, A. Happonen, ja J. Hanski, ”From Asset Provider to Knowledge Company—Transformation in the Digital Era”, teoksessa *Asset Intelligence through Integration and Interoperability and Contemporary Vibration Engineering Technologies*, Cham, 2019, ss. 333–341.
- [7] L. Metso, A. Happonen, V. Ojanen, M. Rissanen, ja T. Kärri, ”Business model design elements for electric car service based on digital data enabled sharing platform”, Cambridge, Iso-Britannia, syys 2019, s. 6. doi: 10.17863/CAM.45886.
- [8] ”2030 digitaalinen kompassi: eurooppalainen lähestymistapa digitaalista vuosikymmentä varten”. Euroopan Komissio, 9. maaliskuuta 2021. Viitattu: 24. tammikuuta 2022. [Verkossa]. Saatavissa: [https://eur-lex.europa.eu/resource.html?uri=cellar:12e835e2-81af-11eb-9ac9-01aa75ed71a1.0006.02/DOC\\_1&format=PDF](https://eur-lex.europa.eu/resource.html?uri=cellar:12e835e2-81af-11eb-9ac9-01aa75ed71a1.0006.02/DOC_1&format=PDF)
- [9] ”Fact Sheet: The Bipartisan Infrastructure Deal”, *The White House*, 6. marraskuuta 2021. <https://www.whitehouse.gov/briefing-room/statements-releases/2021/11/06/fact-sheet-the-bipartisan-infrastructure-deal/> (viitattu 25. tammikuuta 2022).
- [10] ”Gartner Forecasts Worldwide IT Spending to Grow 5.1% in 2022”, *Gartner*. <https://www.gartner.com/en/newsroom/press-releases/2022-01-18-gartner-forecasts-worldwide-it-spending-to-grow-five-point-1-percent-in-2022> (viitattu 29. tammikuuta 2022).
- [11] P. Thomas ja R. Macredie, ”Introduction to The New Usability”, *ACM Trans Comput-Hum Interact*, vsk. 9, ss. 69–73, kesä 2002, doi: 10.1145/513665.513666.
- [12] P. Uhlir ja P. Schröder, ”Open Data for Global Science”, *Data Sci. J.*, vsk. 6, ss. 36–53, kesä 2007, doi: 10.2481/dsj.6.OD36.
- [13] M. Janssen, Y. Charalabidis, ja A. Zuiderwijk, ”Benefits, Adoption Barriers and Myths of Open Data and Open Government”, *Inf. Syst. Manag.*, vsk. 29, ss. 258–268, syys 2012, doi: 10.1080/10580530.2012.716740.

- [14] A. Ahola, T. Skriko, ja T. Björk, ”Fatigue strength assessment of ultra-high-strength steel fillet weld joints using 4R method”, *J. Constr. Steel Res.*, vsk. 167, s. 105861, marras 2019, doi: 10.1016/j.jcsr.2019.105861.
- [15] P. Runeson ja M. Höst, ”Guidelines for conducting and reporting case study research in software engineering”, *Empir. Softw. Eng.*, vsk. 14, nro 2, s. 131, joulu 2008, doi: 10.1007/s10664-008-9102-8.
- [16] J. C. Carver, R. P. Kendall, S. E. Squires, ja D. E. Post, ”Software Development Environments for Scientific and Engineering Software: A Series of Case Studies”, teoksessa *29th International Conference on Software Engineering (ICSE '07)*, touko 2007, ss. 550–559. doi: 10.1109/ICSE.2007.77.
- [17] C. Morris ja J. Segal, ”Lessons Learned from a Scientific Software Development Project”, *IEEE Softw.*, vsk. 29, nro 4, ss. 9–12, heinä 2012, doi: 10.1109/MS.2012.93.
- [18] C. Macaulay *ym.*, ”Usability and User-Centered Design in Scientific Software Development”, *IEEE Softw.*, vsk. 26, nro 1, ss. 96–102, tammi 2009, doi: 10.1109/MS.2009.27.
- [19] Z. Ahmed, S. Zeeshan, ja T. Dandekar, ”Developing sustainable software solutions for bioinformatics by the “Butterfly” paradigm”. F1000Research, 1. elokuuta 2014. doi: 10.12688/f1000research.3681.2.
- [20] G. Wilson, ”Where’s the Real Bottleneck in Scientific Computing?”, *Am. Sci. - AMER SCI*, vsk. 94, tammi 2006, doi: 10.1511/2006.1.5.
- [21] J. Segal, ”Some challenges facing software engineers developing software for scientists”, teoksessa *2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, touko 2009, ss. 9–14. doi: 10.1109/SECSE.2009.5069156.
- [22] M. Korkala ja P. Abrahamsson, ”Communication in Distributed Agile Development: A Case Study”, teoksessa *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)*, elo 2007, ss. 203–210. doi: 10.1109/EUROMICRO.2007.23.
- [23] M. Korkala, M. Pikkarainen, ja K. Conboy, ”Distributed Agile Development: A Case Study of Customer Communication Challenges”, teoksessa *Agile Processes in Software Engineering and Extreme Programming*, Berlin, Heidelberg, 2009, ss. 161–167.
- [24] M. Keil ja E. Carmel, ”Customer-developer links in software development”, *Commun. ACM*, vsk. 38, nro 5, ss. 33–44, touko 1995, doi: 10.1145/203356.203363.
- [25] L. Rampersad, S. Blyth, E. Elson, ja M. M. Kuttel, ”Improving the usability of scientific software with participatory design: a new interface design for radio astronomy visualisation software”, teoksessa *Proceedings of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '17*, Thaba 'Nchu, South Africa, 2017, ss. 1–9. doi: 10.1145/3129416.3129899.
- [26] G. G. Chowdhury, *Information Users and Usability in the Digital Age*. Lontoo: Facet Publishing, 2011.
- [27] S. Han, M. Yun, J. Kwahk, ja S. Hong, ”Usability of Consumer Electronic Products”, *Int. J. Ind. Ergon.*, vsk. 28, ss. 143–151, syys 2001, doi: 10.1016/S0169-8141(01)00025-7.
- [28] ”Liitetaulukko 13. Käytössä älypuhelin ja internetin käyttö televisiolla 2021, %-osuus väestöstä”, *Suomen virallinen tilasto (SVT)*. [https://www.stat.fi/til/sutivi/2021/sutivi\\_2021\\_2021-11-30\\_tau\\_013\\_fi.html](https://www.stat.fi/til/sutivi/2021/sutivi_2021_2021-11-30_tau_013_fi.html) (viitattu 19. helmikuuta 2022).
- [29] ”’Slow, hard to use, impossible to read’: Have you tried Hobart’s new parking meters?”, *ABC News*, 1. marraskuuta 2018. Viitattu: 22. helmikuuta 2022. [Verkossa]. Saatavissa: <https://www.abc.net.au/news/2018-11-01/issues-with-hobarts-new-parking-meters/10454890>

- [30] J. Nielsen, *WWW-suunnittelu*. Helsinki: Edita, IT Press, 2000.
- [31] K. Karvonen, "The beauty of simplicity", teoksessa *Proceedings on the 2000 conference on Universal Usability*, 2000, ss. 85–90.
- [32] F. Queiroz, R. Silva, J. Miller, Sandor Brockhauser, ja H. Fangohr, "Track 1 Paper: Good Usability Practices in Scientific Software Development", 2017, doi: 10.6084/M9.FIGSHARE.5331814.V3.
- [33] G. Pan, "The Hidden Dilemmas in Software Development Project Decisionmaking: Persist or Desist?", esitetty tilaisuudessa Pacific Asia Conference on Information Systems, 2006. Viitattu: 11. toukokuuta 2022. [Verkossa]. Saatavissa: <https://aisel.aisnet.org/pacis2006/92>
- [34] A. Galántai, "The theory of Newton's method", *J. Comput. Appl. Math.*, vsk. 124, nro 1, ss. 25–44, 2000, doi: [https://doi.org/10.1016/S0377-0427\(00\)00435-0](https://doi.org/10.1016/S0377-0427(00)00435-0).
- [35] "Stack Overflow Developer Survey 2020", *Stack Overflow*. [https://insights.stackoverflow.com/survey/2020/?utm\\_source=social-share&utm\\_medium=social&utm\\_campaign=dev-survey-2020](https://insights.stackoverflow.com/survey/2020/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2020) (viitattu 12. maaliskuuta 2022).
- [36] "react-latex", *npm*. <https://www.npmjs.com/package/react-latex> (viitattu 11. huhtikuuta 2022).
- [37] "Function Plot - 2d function plotter powered by d3". <https://mauriciopoppe.github.io/function-plot/> (viitattu 18. maaliskuuta 2022).
- [38] "Plotly". <https://plotly.com/javascript/> (viitattu 20. maaliskuuta 2022).