

# **Pätevyysalueen laskennan automatisointi hitsausohjeille**

**Case: Kemppi Oy**

**Tiivistelmä**

|   |                                     |                         |
|---|-------------------------------------|-------------------------|
| Tekijä(t)<br>Hirvonen, Tuomas   | Julkaisun laji<br>Opinnäytetyö, AMK | Valmistumisaika<br>2024 |
|   | Sivumäärä<br>49                     |                         |
| Työn nimi<br><b>Pätevyysalueen laskennan automatisointi hitsausohjeille</b><br>Case: Kemppi Oy  |                                     |                         |
| Tutkinto<br>esim. Insinööri (AMK), Tieto- ja viestintätekniikka   |                                     |                         |
| Toimeksiantajan nimi, titteli ja organisaatio<br>Kemppi Oy  |                                     |                         |
| Tiivistelmä<br><p>Opinnäytetyössä toteutettiin menetelmäkoepöytäkirjan luonti sekä siihen kuuluvan pätevyysalueen laskennan automatisointi. Palvelun tarkoituksena on tarjota käyttäjille helppo tapa laatia sekä ylläpitää menetelmäkokeita ja luoda niiden pohjalta hitsausohjeita. Työssä keskitytään palvelun toteuttamisprosessiin sekä siihen liittyvien työkalujen ja teknologioiden käyttöön.</p> <p>Palvelun käyttöliittymä toteutettiin Angular-ohjelmistokehyksellä sekä siihen liittyvillä eri kirjastoilla. Palvelussa tarvittavien pätevyysalueiden laskennat toteutettiin luomalla niitä varten oma npm-paketti, jota käytetään palvelussa.</p> <p>Työ toteutettiin Kemppi Oy:lle 2022 ja julkaistiin saman vuonna syyskuussa. Työ oli osa suurempaa projektia, jossa osana haluttiin tarjota käyttäjille mahdollisuus hallinnoida menetelmäkokeita ja automatisoida menetelmäkokeelle täytettävän pätevyysalueen täyttö jo olemassa olevassa WeldEye-palvelussa.</p> <p>Toteutuksen myötä käyttäjillä on nyt helppo tapa täyttää, sekä ylläpitää menetelmäkoepöytäkirjoja eri standardeille WeldEye-palvelussa.</p> |                                     |                         |
| Asiasanat<br>Angular, hitsausohje, menetelmäkoepöytäkirja, pätevyysalue   |                                     |                         |

## Abstract

|   |                                    |                   |
|---|------------------------------------|-------------------|
| Author(s)<br>Hirvonen, Tuomas   | Type of Publication<br>Thesis, UAS | Published<br>2024 |
|   | Number of Pages<br>49              |                   |
| Title of Publication<br><b>Automation of qualification range calculation for welding procedure specifications.</b><br>Case: Kemppi Oy   |                                    |                   |
| Name of Degree<br>Bachelor of Engineering, Information and Communications Technology  |                                    |                   |
| Name, title and organization of the client<br>Kemppi Oy   |                                    |                   |
| Abstract<br><p>The objective of the thesis was to produce a feature into an already existing web application where users can easily create and manage welding procedure qualification records and automate the process of filling the range of qualification in them.</p> <p>The user interface was implemented with the Angular framework. The calculations for the qualification ranges were implemented by creating a npm package which was then imported and used in the application.</p> <p>The thesis was done for the Kemppi Oy in 2022 and was published later in the same year. The thesis was part of a larger project, where a part of the goal was to provide the possibility to manage and automate some of the processes in welding procedure qualification records in the already existing WeldEye-web application.</p> <p>As a result of the thesis, the users now have an easy way to manage their welding procedure qualification records for different standards and automate the process of filling the range of qualification in them.</p> |                                    |                   |
| Keywords<br>Angular, welding procedure specification, welding procedure qualification record, qualification range   |                                    |                   |

## Sisällys

|       |   |    |
|-------|---|----|
| 1     | Johdanto.....   | 1  |
| 2     | Hitsauksen menetelmäkokeen periaate .....                       | 2  |
| 3     | Työkalut ja teknologiat .....                                   | 4  |
| 3.1   | TypeScript .....  | 4  |
| 3.2   | Angular-ohjelmistokehys .....                                   | 6  |
| 3.2.1 | Dekoraattorit.....  | 6  |
| 3.2.2 | Komponentit .....   | 7  |
| 3.2.3 | Näkymät .....   | 9  |
| 3.2.4 | Palvelut .....  | 10 |
| 3.2.5 | Direktiivit.....  | 11 |
| 3.3   | RxJs .....  | 12 |
| 3.4   | Npm .....   | 16 |
| 3.5   | Mocha .....   | 16 |
| 3.6   | Chai.....   | 17 |
| 4     | Toteutus .....  | 19 |
| 4.1   | Vaatimukset .....   | 19 |
| 4.2   | Yleiskatsaus .....  | 20 |
| 4.3   | Npm-paketti.....  | 21 |
| 4.4   | Käyttöliittymä.....   | 28 |
| 4.4.1 | Menetelmäkoepöytäkirjakomponentti.....                          | 28 |
| 4.4.2 | Pätevyysalueen laskennan ja täytön automatisointi .....         | 39 |
| 4.4.3 | Hitsausohjeiden luominen menetelmäkoepöytäkirjan pohjalta ..... | 44 |
| 5     | Yhteenveto .....  | 49 |
|       | Lähteet .....   | 49 |

## 1 Johdanto

Kemppi on suomalainen, vuonna 1949 Lahdessa perustettu hitsausalan perheyrittäjä, jolla on 20 tytäryhtiötä ja lähes 800 työntekijää maailmanlaajuisesti ja jonka liikevaihto oli 195 miljoonaa euroa vuonna 2022. Kemppi kehittää, valmistaa ja tarjoaa hitsauslaitteita, hitsaukseen liittyviä lisä- ja suojarusteita, hitsaustuotannon hallintaohjelmistoja sekä asiantuntijapalveluita. WeldEye on Kemppi Oy:n kehittämä ja ylläpitämä järjestelmä hitsauksen tukemiseksi, jossa on muun muassa mahdollisuus hallita ja ylläpitää hitsaajien tietoja, kuten pätevyyskatsauksia, hitsausohjeita, jotka ovat tärkeässä osassa teollisen hitsauksen laadun varmistamisessa. Näiden lisäksi WeldEye tarjoaa myös projektien sekä hitsauksen seurantaan liittyviä työkaluja, kuten hitsien tallentamista ja niiden parametrien, sekä niistä löytyvien poikkeamien tarkastelua, laitteiden hallintaa ja monia muita ominaisuuksia. (Kemppi Oy. Tietoja Kempistä)

Kuvittele, että olet öljynporauslautalla keskellä merta. Merituuli puhaltaa ja painaa kehoasi kohti lautan kaidetta, joka on hitsattu kauttaaltaan yhteen kestävyuden takaamiseksi. Mikä takaa, ettei kaide petä? Hitsausohjeet ovat tärkeässä roolissa teollisessa hitsauksessa. Niiden ensisijaisena tehtävänä on varmistaa hitsattujen tuotteiden laatu ja kestävyys, mutta ne myös tarjoavat hitsaajille selkeän suunnitelman ja ohjeet hitsausprosessin suorittamiseen turvallisesti ja tehokkaasti.

Tämän työn tavoitteena on automatisoida menetelmäkokeeseen merkittävän pätevyysalueen laskenta sekä lasketun pätevyysalueen täyttö menetelmäkoepöytäkirjaan. Menetelmäkoepöytäkirjan pohjalta on myös pystyttävä luomaan hitsausohje, joille hitsin eri parametrit ja ominaisuudet kopioituvat suoraan menetelmäkoepöytäkirjaan merkatuista pätevyysalueista. Työ tehdään Kemppi Oy:lle asiakkaan tilaustyönä.

Tämä työ kattaa menetelmäkoepöytäkirjan pätevyysalueiden laskennan automatisoinnin, menetelmäkoepöytäkirjan pätevyysalueen automatisoidun täytön sekä menetelmäkoepöytäkirjan pätevyysalueen kopioinnin menetelmäkokeesta luoduille hitsausohjeille WeldEye-palvelussa.

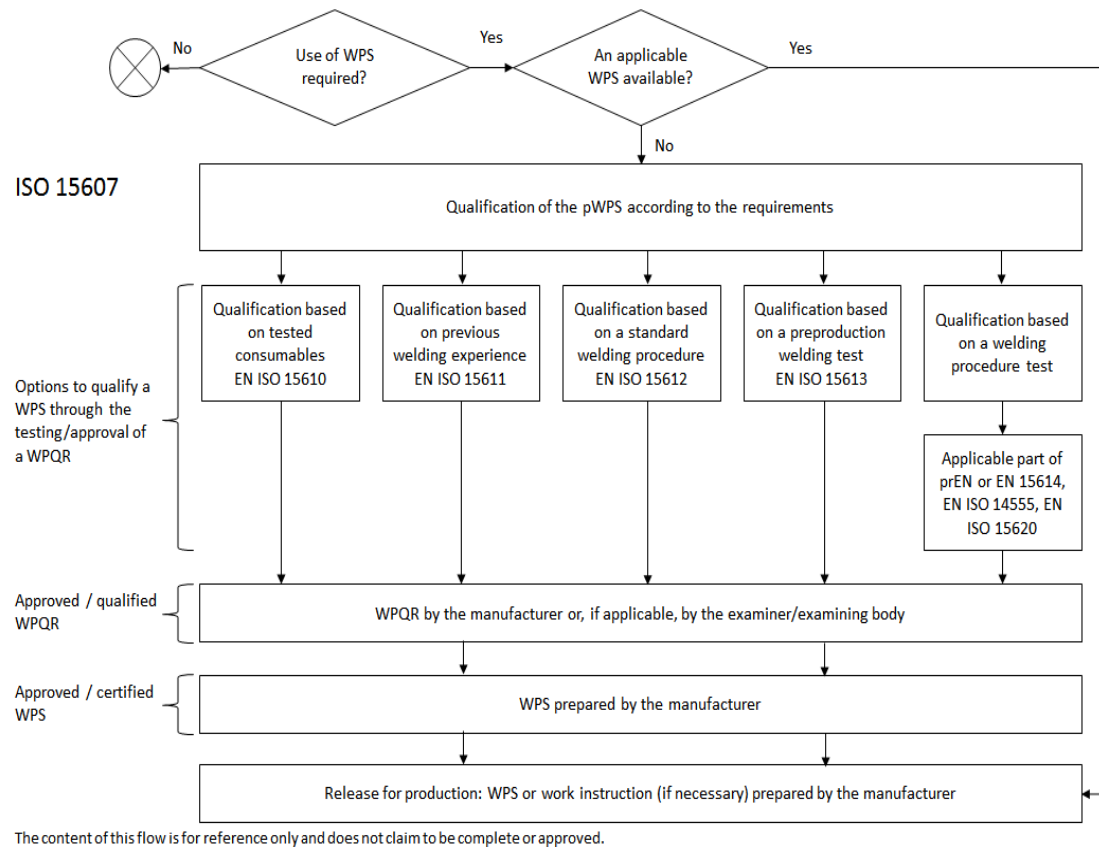
## 2 Hitsauksen menetelmäkokeen periaate

Hitsauksen pohjimmaisena periaatteena on yhdistää metallikappaleita toisiinsa pysyvästi lämpövaikutuksen avulla. Tämä voidaan saavuttaa sulattamalla liitoskohtaan metallia, joka jäähtyttyään muodostaa sauman. Liikaa lämpöä, ja ympäröivä metalli sulaa. Liian vähän lämpöä, ja sauma ei sulaudu kunnolla yhteen. Molemmat tilanteet voivat aiheuttaa halkeilua, heikkoa tarttumista tai jopa sauman irtoamisen. Mutta millä ja miten varmistetaan, että hitsaaja saa hitsattua sauman, joka on tarpeeksi vahva ja tulee kestäväksi sille kohdistuvat ulkoiset rasitukset? Menetelmäkokeen tarkoituksena on hitsata koekappale ja tarkastella kaikkia hitsiin kohdistuvia muuttujia, kuten sauman valmistelua, lämmöntuontia ja lämpökäsittelyä, käytettyjä materiaaleja sekä aineita sekä kaikkia muita muuttujia, joilla on vaikutus hitsaussauman laatuun ja lujuuteen.

Yleisesti ottaen hitsausohjeet on jaoteltu kahteen eri kategoriaan: alustaviin hitsausohjeisiin sekä hitsausohjeisiin. Alustavien hitsausohjeiden pohjalta suoritetaan menetelmäkoe, jossa alustavaan hitsausohjeeseen perustuen hitsataan koekappale. Tämän jälkeen hitsille suoritetaan testejä ja tutkimuksia. Mikäli hitsi täyttää tietyt kriteerit, jotka eri standardit määrittävät, hitsille voidaan määrittää pätevyysalue, jonka rajoissa olevat hitsit ovat standardin mukaan valideja ja niiden laatu voidaan perustella menetelmäkokeessa suoritettujen hitsien tuloksiin.

Hitsauksen menetelmäkokeen (WPQR, welding procedure qualification record) tarkoituksena on varmistaa hitsausmenetelmien ja -prosessien laatu sekä hitsattujen liitosten kestävyys ja luotettavuus. Tämä on erityisen tärkeää esimerkiksi teollisuudessa, rakentamisessa ja konepajoilla, missä hitsauksen laatu vaikuttaa suoraan lopputuotteen turvallisuuteen ja kestävyteen.

Menetelmäkokeessa tyypillisesti hitsin toteuttava organisaatio on laatinut alustavan hitsausohjeen (pWPS), joka pohjautuu suunniteltuun lopputuotteeseen. Tämän jälkeen alustavan hitsausohjeen pohjalta valmistetaan koekappale tai useita kappaleita, jotka vastaavat todellista lopputuotetta. Koekappaleet sitten hitsataan hitsausohjeen määrittämien parametrien mukaisesti, jonka jälkeen kappaleille suoritetaan erilaisia tarkastuksia sekä mittauksia. Tämä voi sisältää röntgenkuvauksia, ultraäänitarkastuksia, veto-, isku- ja taivutuskokeita sekä monia muita testausmenetelmiä. Tämän jälkeen testien ja tarkastusten tuloksia verrataan hitsaus-standardien ja vaatimusten kanssa. Mikäli tulokset ovat hyväksyttäviä, voidaan alustavan hitsausohjeen pohjalta luoda hitsausohjeita (WPS), joita voidaan käyttää lopputuotteiden hitsauksessa. Tämä prosessi on tarkemmin avattu kuviossa 1.



Kuvio 1. Hitsausohjeen toteuttamisen vuokaavio (ECE Global 2024)

Hitsauksen menetelmäkokeen tarkoituksena on siis varmistaa, että hitsausprosessit ja -liitokset täyttävät tarvittavat laatuvaatimukset ja standardit. Näin varmistetaan lopputuotteiden luotettavuus ja turvallisuus.

### 3 Työkalut ja teknologiat

#### 3.1 TypeScript

JavaScript on dynaaminen ohjelmointikieli, jossa on mahdollista määrittää muuttujia minkä tahansa tyyppiseksi arvoiksi, tai jopa vaihtaa niiden arvoja eri tyyppiin ajon aikana. JavaScriptissä voi myös viitata muuttujiin, joita ei ole edes olemassa, kuten kuvassa 1 ilmenee. Mikäli tämä tapahtuu ohjelman ajon aikana, syntyy virhe, jonka seurauksena ohjelma ei välttämättä toimi oletetulla tavalla.

```
typescript > JS index.js > ...  
1   const value = 'a'  
2   value.toFixed();  
      .....
```

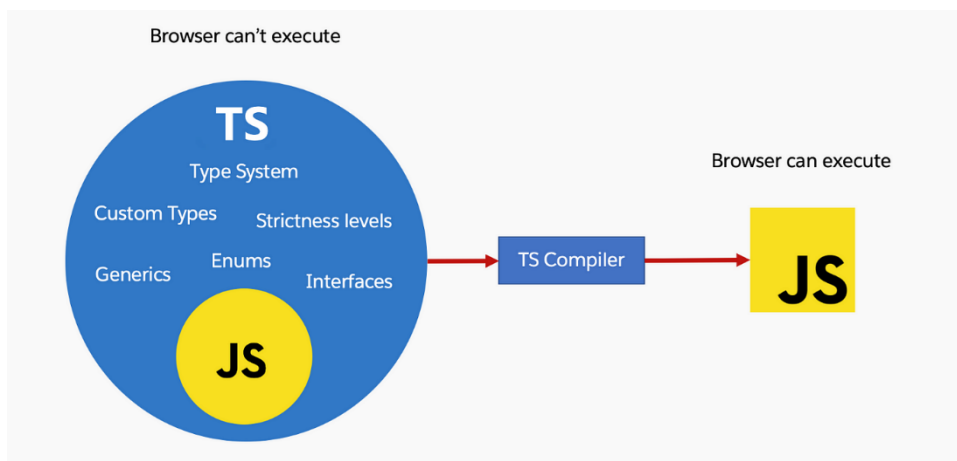
Kuva 1. JavaScript-referointi määrittämättömään muuttujaan

TypeScript on Microsoftin kehittämä ja ylläpitämä avoimen lähdekoodin ohjelmointikieli, jonka tarkoituksena on ratkaista JavaScriptin edellä mainittuja ongelmakohtia. Se on JavaScriptin yläjoukko (superset), joka laajentaa kieltä mahdollistamalla muun muassa tyyppitarkastuksen koodissa ja täten ehkäisee ajon aikana ilmeneviä ongelmia, kuten esimerkkinä kuvassa 2.

```
typescript > src > TS index.ts > ...  
1   const value: string = 'a'  
2   value.toFixed();  
3  
Property 'toFixed' does not exist on type 'string'.
```

Kuva 2. TypeScript-referointi määrittämättömään muuttujaan

Koska TypeScript on JavaScriptin yläjoukko, se myös tukee täysin pelkkää JavaScript-koodia. Lisäksi tarvittaessa tyyppitarkastuksen voi ohittaa antamalla muuttujan tyyppiä minkä tahansa (any) tyyppiin. TypeScriptille on oma kääntäjä (compiler), jolla se käännetään (transpile) takaisin JavaScriptiksi selaimia, tai muita ajoympäristöjä varten. TypeScriptin ominaisuudet ja toimintaperiaate kuviossa 2. (TypeScript a)



Kuvio 2. TypeScript'in toimintaperiaate

TypeScriptille täytyy määrittää tiedosto (tsconfig, Kuva 3), jossa määritellään TypeScriptin käyttäytyminen projektissa. Tähän tiedostoon voidaan määrittellä muun muassa kohdekansio (src), joka kertoo mistä polusta TypeScript lukee koodin, "include" sekä "exclude", joihin voi määrittää mitkä tiedostot tai hakemistot sisällytetään käännökseen. "Target" kertoo mille ECMAScript versiolle käännetty JavaScript muunnetaan yhteensopivaksi. Lisäksi tiedostoon voi määrittää myös monia tyyppi tarkistuksiin liittyviä asetuksia, kuten "strict", jolla voidaan määrittää kuinka tiukasti TypeScript tarkistaa tyyppien yhteensopivuuden. (TypeScript b)

```

typescript > tsconfig.json > ...
1  {
2    "compilerOptions": {
3      "module": "CommonJS",
4      "target": "ES2022",
5      "outDir": "dist",
6      "strict": true,
7      "skipLibCheck": true
8    },
9    "include": ["src/**/*"],
10   "exclude": ["redundant_files"]
11  }

```

Kuva 3. Esimerkki tsconfig-tiedostosta

## 3.2 Angular-ohjelmistokehys

Angular on Googlen kehittämä avoimen lähdekoodin verkko-ohjelmistokehys. Angularin arkkitehtuuri mukaillee pohjimmiltaan MVC-arkkitehtuuria ja perustuu komponentteihin, näkymiin sekä palveluihin, jotka on jaoteltu erillisiin moduuleihin. Se on kirjoitettu TypeScriptillä ja suunniteltu erityisesti yksisivuisten verkkosovellusten kehittämistä varten. Esimerkiksi Microsoft Office, Gmail, Forbes, PayPal sekä Samsung käyttää sivuillansa Angularia. Angularilla on myös oma rikas ekosysteeminsä ja sille löytyy suuri määrä erinäisiä komponenttikirjastoja, joista moni, kuten muun muassa Angular Material kirjasto, joka sisältää valmiiksi määriteltäviä komponentteja esim. input elementeille, on Angularin itse kehittämiä sekä ylläpitämiä. (Angular c)

### 3.2.1 Dekoraattorit

Dekoraattori on suunnittelumalli (design pattern) ja tarkalleen ottaen rakennesuunnittelumalli, jonka avulla voidaan liittää toiminnallisuutta olemassa olevan objektin tai luokan toiminnallisuuteen, käärimällä (wrapping) se dekoraattori-funktioon, joka lisää toiminnallisuutta, muuttamatta olemassa olevaa toiminnallisuutta. Kuvassa (Kuva 4) yksinkertaistettu esimerkki, miten dekoraattori toimii käytännössä.

```

1  const logger = (message) => {
2  |   console.log(message); 'Message to log'
3  | }
4  |
5  function loggerDecorator (logger) {
6  |   return function (message) {
7  |     logger.call(this, message);
8  |     console.log(`${message}, in decorator`); 'Message to log, in decorator'
9  |   }
10 | }
11 |
12 const decoratedLogger = loggerDecorator(logger);
13 |
14 decoratedLogger('Message to log');
15 |

```

Kuva 4. Esimerkki dekoraattorin toiminnasta

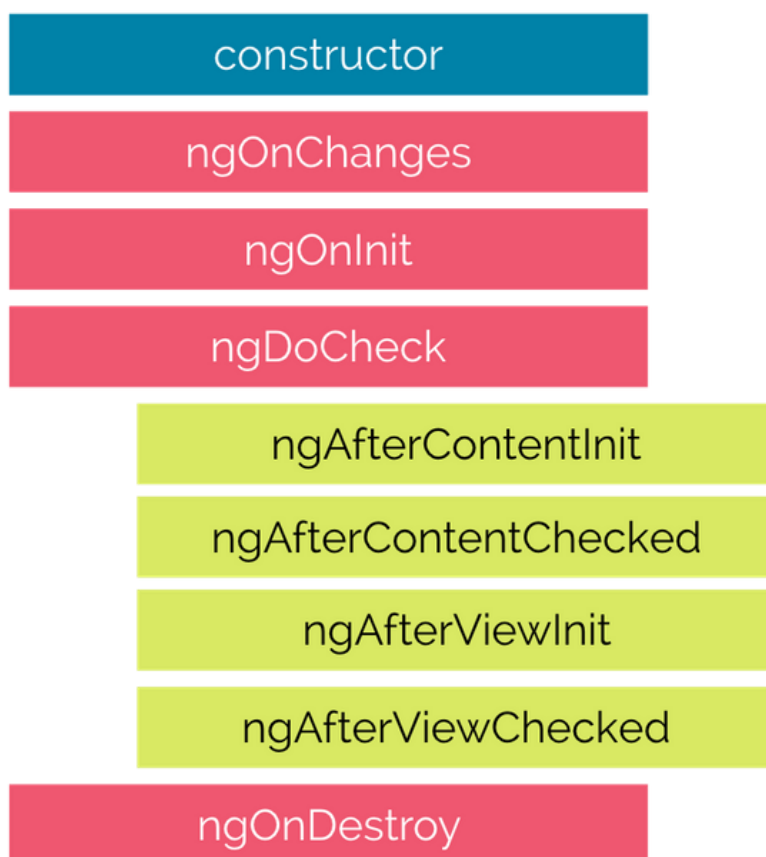
Angular käyttää dekoraattoreita useissa eri yhteyksissä, kuten komponenteissa, palveluissa, moduuleissa ja injektioissa. Dekoraattorit ovat TypeScriptin ominaisuus, jota Angular hyödyntää metadatan lisäämiseen luokkiin ja niiden jäseniin. Tätä käytetään esimerkiksi, kun luodaan komponentti. Komponentti luokalle annetaan @Component-dekoraattori, jotta Angular osaa tulkita sen komponentiksi ja sitoa sille dekoraattorin parametrien sisälle määritetyn metadatan. Dekoraattoreita voi olla myös komponentin property-tyyppinä. Näitä ovat esimerkiksi @Input- ja @Output-dekoraattorit. @Input-

dekoraattorilla voidaan mahdollistaa datan syöttäminen ylemmältä komponentilta lapsi komponentille. @Output-dekoraattori mahdollistaa datan lähettämisen komponentistä ylöspäin esimerkiksi tapahtuman yhteydessä. Komponenttien dekoraattorit ovat olennainen osa Angularin komponenttipohjaista arkkitehtuuria.

### 3.2.2 Komponentit

Komponentit ovat Angular sovelluksessa uudelleenkäytettäviä osia, jotka koostuvat kolmesta osasta: komponentin luokasta, näkymästä sekä tyyleistä. Nämä kolme osaa muodostavat näkymän, joka voidaan esittää sovelluksessa.

Komponentin luokka (component), joka on kirjoitettu TypeScript-koodina vastaa komponentin datasta sekä logiikasta. Se sisältää komponentin toiminnallisuuden ja logiikan, joka määrittelee, miten komponentti käyttäytyy ja miten se reagoi esimerkiksi käyttäjän toimintaan. Komponenteilla on myös metodeja, joilla komponentin tilaa voidaan hallita sen eri vaiheissa (Kuvio 3). Näihin metodeihin kuuluu muun muassa `ngOnInit`, `ngOnChanges` sekä `ngOnDestroy`. Angular hallitsee näiden metodien kutsumista erinäisissä tilanteissa. Esimerkiksi `ngOnInit`-metodia kutsutaan, kun komponentti luodaan näkymässä. `ngOnChanges`-metodia kutsutaan, kun komponentille sidottu data muuttuu ylemmällä komponentilla. `ngOnDestroy`-metodia kutsutaan, kun komponenttia ei enää käytetä näkymässä.



Kuvio 3. Angular komponentin elinkaari

Komponenteille voi määrittää muuttujia ja ominaisuuksia (property) joiden toiminnallisuutta voidaan laajentaa erinäisillä Angularista löytyvillä dekoraattoreilla. Näistä esimerkkinä @Input-dekoraattori (Kuva 5), jolla kyseisen muuttujan arvon voi sitoa ylemmältä komponentilta annettuun arvoon. Komponenteille voi myös määrittää omia metodeja, jotka toimivat kuten normaalit JavaScript luokan metodit. Näiden avulla voidaan helposti dynaamisesti hallita komponentin tilaa tarpeiden mukaan.

```

3  √ @Component({
4      selector: 'app-example-component',
5  √  template: `
6      |     <p>Message: {{ message }}</p>
7      | `
8      ,
9      styles: ['p { font-size: 12 }']
10 })
11 √ export class ExampleComponentComponent {
12     | @Input() message: string;
13     | }

```

Kuva 5. Esimerkki @Input-dekoraattorista

Näkymä (template) määrittelee komponentin ulkoasun ja rakenteen käyttämällä HTML-syntaksia. Näkymässä voidaan myös käyttää Angularin templaattikielen tarjoamia laajennuksia, kuten direktiivejä ja muuttujia, jotka mahdollistavat dynaamisen sisällön luomisen ja komponenttien välisten suhteiden hallinnan.

Tyylit (styles) määrittelevät komponentin ulkoasun käyttämällä CSS-tyylisääntöjä. Tyylit, kuten myös näkymän voi määrittää erillisillä tiedostoilla, joka sidotaan komponentti-dekoraattorin avulla komponentille metadataksi. Angularissa voi myös käyttää erilaisia CSS:n laajennuksia, kuten SCSS (Sassy CSS). SCSS-tiedostot käännetään tavallisiksi CSS-tiedostoiksi, kun Angular ohjelma käännetään. Angular tukee myös globaalien tyylien määrittelyn projektin juuresta löytyvän styles CSS-tiedoston avulla. Tyylit valuvat (cascade) komponenttien hierarkian mukaisesti ylemmältä alemmille.

### 3.2.3 Näkymät

Näkymät (template) näyttävät samalta ja toimivat kuten tavallinen HTML. Ne tukevat myös Angularin omaa näkymäsyntaksia sekä laajennuksia, jotka mahdollistavat muun muassa kaksisuuntaisen data sidonnan. Kaksisuuntainen data sidonta tarkoittaa, että komponentin dataa voidaan näyttää ja päivittää suoraan näkymässä interpolaation avulla. Lisäksi näkymät tukevat direktiivien käyttöä, kuten for-silmukkaa, jonka avulla voidaan renderöidä useita elementtejä dynaamisesti näkymässä. Tapahtumasidonta (event-binding) on myös osa näkymien toiminnallisuutta, mikä mahdollistaa erilaisten tapahtumien, kuten hiiren painallusten kuuntelun ja niihin reagoinnin suoraan näkymässä. Lisäksi Angular HTML tukee myös putkia, lomakkeiden käsittelyyn liittyvää valvontaa ja hallintaa, sekä monia

muita ominaisuuksia. Angular tarjoaa myös templaatti muuttujia (template variable), joiden avulla kyseiseen elementtiin voidaan referoida mistä tahansa muualta komponentista, kuten muualla templaattissa (Kuva 6), tai vaikkapa komponentin luokasta. (Angular d)

```
src/app/app.component.html

<input #phone placeholder="phone number" />

<!-- lots of other elements -->

<!-- phone refers to the input element; pass its `value` to an event handler -->
<button type="button" (click)="callPhone(phone.value)">Call</button>
```

Kuva 6. Templaatti muuttuja

### 3.2.4 Palvelut

Palvelut (services) ovat olennainen osa sovelluksen rakennetta. Ne ovat Angularissa oletuksena ainokaisia (singleton), jotka tarjoavat mahdollisuuden jakaa yhteisiä toimintoja tai dataa useiden komponenttien välillä, sekä eristää liiketoimintalogiikan. Palvelut myös edesauttavat modulaarisuutta, komponenttien ylläpidettävyttä sekä keveyttä. Palvelut tukevat myös riippuvuusinjektiota (dependency injection), mikä tarkoittaa, että palveluita voidaan injektoida myös muihin palveluihin tarpeen mukaan. Kuvassa 7 esimerkki Angular palvelusta. (Angular e)

```

TS example.service.ts U
src > app > shared > services > TS example.service.ts > ExampleService > getMessage
1  import { Injectable } from '@angular/core';
2
3  @Injectable({
4    providedIn: 'root'
5  })
6  export class ExampleService {
7
8    constructor() {}
9
10   getMessage() {
11     return 'hello from example service!';
12   }
13
14 }
15

TS example-component.component.ts U
src > app > shared > components > example-component > TS example-component.compo
1  import { Component } from '@angular/core';
2  import { ExampleService } from 'app/shared/services/example.service';
3
4  @Component({
5    selector: 'app-example-component',
6    template: `
7      <p>Message: {{ message }}</p>
8    `,
9    styles: [],
10   providers: [ExampleService]
11 })
12 export class ExampleComponentComponent {
13   message: string;
14
15   constructor(exampleService: ExampleService) {
16     this.message = exampleService.getMessage();
17   }
18
19 }

```

Kuva 7. Esimerkki palvelusta Angularissa

### 3.2.5 Direktiivit

Direktiivit (directive) ovat kuin ohjeita, joita voidaan antaa Angular HTML:ssä elementeille. Pohjimmiltaan ne ovat luokkia. Myös komponentit itsessään ovat direktiivejä.

Komponenttien lisäksi on myös rakenteellisia direktiivejä sekä attribuutti direktiivejä.

Angulariin kuuluu sisäänrakennettuja direktiivejä, esimerkiksi NgIf, NgFor (kuvassa 8) sekä NgSwitch, jotka ovat rakenteellisia direktiivejä ja vaikuttavat HTML-rakenteeseen.

NgClass, NgStyle ja NgModel ovat attribuutti direktiivejä, joista esimerkiksi NgClass direktiivillä pystytään muokkaamaan HTML elementin luokkia suoraan näkymässä.

NgStyle on kuin NgClass, mutta sillä voidaan muokata elementin tyyliä. NgModel

direktiivillä voidaan lisätä HTML elementille data sidonta. Direktiivejä voi myös luoda itse.

Direktiivit Angularissa ovat hyvin voimakas työkalu, koska ne tarjoavat monipuolisen ja joustavan tavan manipuloida HTML-rakennetta, käyttäytymistä ja ulkoasua sovelluksessa. (Angular a, Angular b ja Angular f)

```

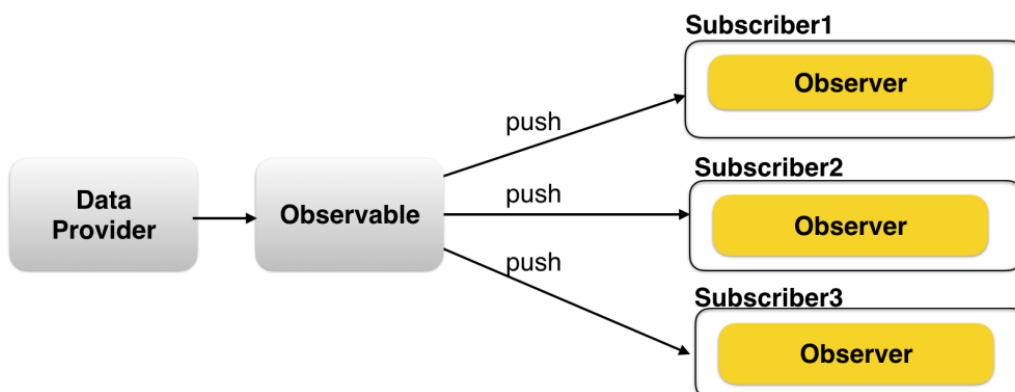
3   @Component({
4     selector: 'app-example-component',
5     template: `
6       <ng-container *ngFor="let message of messages">
7         <p *ngIf="message; else noMessage">
8           |   Message: {{ message }}
9         </p>
10      <ng-template #noMessage>
11        |   No message!
12      </ng-template>
13    </ng-container>
14  `,
15  styles: []
16  })
17  export class ExampleComponentComponent {
18    |   @Input() messages: string[];
19  }

```

Kuva 8. Esimerkki direktiiveistä Angularin näkymässä

### 3.3 RxJs

RxJs on kirjasto, joka perustuu tarkkailtaviin arvoihin (observable), subjekteihin, tilauksiin (subscription) ja tarjoaa suuren määrän funktioita reaktiiviseen ohjelmointiin näiden pohjalta (Kuvio 4). RxJs on myös sisälletty Angulariin yhtenä ydinkirjastoista.



Kuvio 4. RxJs:n toimintaperiaate

RxJs mahdollistaa myös JavaScript lupauksen (promise) kääntämisen tarkkailtaviksi arvoiksi ja päin vastoin, kuten kuvasta (Kuva 9) ilmenee. RxJs:n perusideana on tilata (subscribe) tarkkailtavia arvoja, joita voidaan sitten käsitellä niiden muuttuessa. Tilaus metodi ottaa parametrina takaisinkutsu funktion (callback function), joka ajetaan, kun tarkkailtava arvo muuttuu.

```
1  import { from } from "rxjs";
2
3  const somePromise = new Promise<number>((resolve) => {
4    |   return resolve(1);
5  });
6
7  const asObservable = from(somePromise);
8
9  asObservable.subscribe(resolved => {
10 |   console.log(resolved); 1
11 | });
```

Kuva 9. Lupauksen kääntäminen tarkkailtavaksi arvoksi

Kuten aiemmin mainittiin, RxJs tarjoaa suuren määrän valmiita funktioita (operators) joita voidaan putkittaa (pipe) ja täten käsitellä jatkuvasti tulevaa dataa halutulla tavalla. Esimerkissä (Kuva 10) kuunnellaan dataa, mutta reagoidaan vain tietyn tyyppisiin muutoksiin käyttämällä suodatin-funktiota (filter).

```

1  import { Subject, filter } from "rxjs";
2
3  const subject = new Subject<number>();
4
5  const subscriptionToNumbersLessThanFour = subject.pipe(
6    |   filter(number => number < 4)
7  ).subscribe(number => {
8    |   console.log(number); 1, 2
9  });
10
11 subject.next(1);
12 subject.next(2);
13 subject.next(5);
14 subject.next(10);
15

```

Kuva 10. RxJs filter –funktio

Tämän tyyppisiä funktiota voidaan ketjuttaa tarpeen mukaisesti. Esimerkissä (Kuva 11) käytetään yhdessä suodatin–funktioita sekä ota–funktioita (take), jolla voidaan määrittää, että kuuntelija ottaa vain ensimmäisen subjektilta lähetetyn arvon.

```

1  import { Subject, filter, take } from "rxjs";
2
3  const subject = new Subject<number>();
4
5  const subscriptionToNumbersLessThanFour = subject.pipe(
6    |   filter(number => number < 4),
7    |   take(1)
8  ).subscribe(number => {
9    |   console.log(number); 1
10  });
11
12 subject.next(1);
13 subject.next(2);
14 subject.next(5);
15 subject.next(10);

```

Kuva 11. RxJs-funktioiden ketjuttaminen

RxJs tilaukset (subscription) elävät kunnes ne puretaan (unsubscribe). Tämä voi aiheuttaa tilausten kasaantumisen muistiin, ellei tilausta pureta oikein. Kuvassa 11 ota-funktio toteuttaa tilauksen purun saatuaan yhden arvon subjektilta, kun taas kuvassa 10 tilaus jää päälle, kunnes tilaus puretaan manuaalisesti käyttämällä tilauksen purku -metodia (unsubscribe). Kuvissa 12 ja 13 esitetään, miten tilauksen purku toimii.

```
1  import { Subject, filter, take } from "rxjs";
2
3  const subject = new Subject<number>();
4
5  ∨ const subscriptionToNumbersLessThanFour = subject.pipe(
6    |   filter(number => number < 4)
7  ∨ ).subscribe(number => {
8    |   console.log(number); 1, 1
9    | });
10
11  subject.next(1);
12
13  // subscriptionToNumbersLessThanFour.unsubscribe();
14
15  subject.next(1);
```

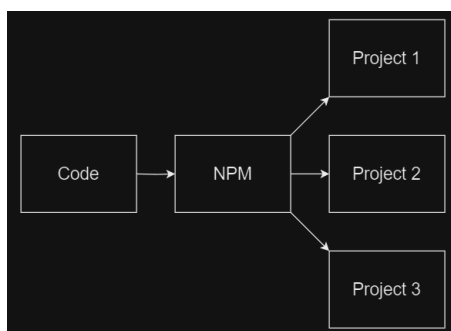
Kuva 12. RxJs-tilaus

```
1  import { Subject, filter, take } from "rxjs";
2
3  const subject = new Subject<number>();
4
5  ∨ const subscriptionToNumbersLessThanFour = subject.pipe(
6    |   filter(number => number < 4)
7  ∨ ).subscribe(number => {
8    |   console.log(number); 1
9    | });
10
11  subject.next(1);
12
13  subscriptionToNumbersLessThanFour.unsubscribe();
14
15  subject.next(1);
```

Kuva 13. RxJs-tilauksen purku

### 3.4 Npm

Npm (Node package manager) on työkalu, jonka avulla voi ladata, hallita ja jakaa koodipaketteja. Npm rekisteri toimii keskitettynä paikkana, josta kehittäjät voivat löytää ja asentaa valmiita koodipaketteja projekteihinsa. Npm rekisteri tukee myös yksityisiä rekisterejä, joissa käyttäjien on mahdollista pitää yksityisiä paketteja, joihin vain tiettyjen käyttäjien on mahdollisuus päästä käsiksi. Npm yhdessä npm rekisterin kanssa mahdollistaa helpon tavan uudelleenkäyttää ja ylläpitää toistuvaa koodia eri projektien välillä (Kuva 14). (Npm. About npm)



Kuva 14. Npm rekisterin toimintaperiaate

### 3.5 Mocha

Mocha on testikehys, joka tarjoaa helpon, tehokkaan ja joustavan tavan testata koodia. Mochan avulla voidaan helposti varmistaa koodin luotettavuus ennen kuin se julkaistaan tai otetaan käyttöön. Mocha tarjoaa helppokäyttöisen ja selkeän syntaksin testien laatimiselle sekä ryhmittelylle. Testit voidaan sitten suorittaa käyttäen mochaa, joka tulostaa testien tulokset ja niistä voidaan luoda raportti. Mocha käyttää JavaScript koodin ajamiseen Node.js:sää, joka pohjautuu verkkoselain Chromen V8 JavaScript moottoriin ja toimii JavaScript ajoympäristönä selainten ulkopuolella. Mocha ajaa sille määritetyt testit sarjallisesti ja se voi ajaa testejä useista eri poluista. Mochalle voi määrittää globaaleja järjestelyitä, kuten esimerkiksi testi tietokannan pystyttämisen ennen testien ajoa, sekä sen purkamisen testien päätyttyä. Lisäksi Mocha tarjoaa erilaisia metodeja, joita voidaan ajaa joko ennen tiettyjä testejä, tai näiden testien jälkeen, kuten kuvassa (Kuva 15) näkyy. (Mocha JS)

```

12 describe('/devices', () => {
13     let testStorage;
14     let mockConfig;
15
16    nock.disableNetConnect();
17
18     before(() => {
19         nock(`${process.env.API_URL}`)
20             .persist()
21             .post('/devices')
22             .reply((uri, body) => {});
23     })
24
25     beforeEach(() => {
26         testStorage = sinon.createStubInstance(ObjectStorage);
27         mockConfig = {
28             storages: testStorage
29         };
30     });
31
32     describe('create-device | POST /devices', () => {
33         it('should successfully create a device', async () => {
34             const res = await createDevice(mockConfig);
35             expect(res.statusCode).to.equal(200);
36         });
37     });
38 });

```

Kuva 15. Mocha testi tiedosto

### 3.6 Chai

Chai on todennuskirjasto (assert library), jolla voidaan kirjoittaa selkeitä ja helposti luettavia testejä. Chai tarjoaa erilaisia todennusmetodeja, kuten should ja expect, jotka tekevät testien kirjoittamisesta sekä lukemisesta helpompaa. Chai myös mahdollistaa todennusmetodien ketjuttamisen ja täten helpottaa huomattavasti testien kirjoittamista sekä lukemista. Näiden todennusmetodien avulla voidaan tarkistaa, että koodi käyttäytyy odotetulla tavalla. (Chai JS a)

Kuvassa 16 on esimerkki Chain todennusmetodien käytöstä. Kuvassa käytetään Chain expect-funktiota yhdessä ketjutettavien metodien kanssa. Kuvan 16 koodin tarkoituksena on tarkistaa, ettei defaultBucket-muuttujan get-metodia ole kutsuttu invalidKey-muuttujaa vastaavalla arvolla. (Chai JS b)

```
it('Should skip files with an invalid key', async () => {  
  |   expect(defaultBucket.get).to.not.have.been.calledWith(invalidKey);  
  | });
```

Kuva 16. Esimerkki Chain todennusmetodista

## 4 Toteutus

### 4.1 Vaatimukset

Käyttäjien on pystyttävä lisäämään ja täyttämään WeldEye-palvelun hitsausohje – moduulissa uusi menetelmäkoepöytäkirja, jonka formaatti tukee kaikkia vaadittuja standardeja. Menetelmäkoepöytäkirjan pätevyysalueen laskenta sekä täyttö tulee olla mahdollista automatisoida. Automatisoidun laskennan tuottamien arvojen tulee vastata standardien määrittämiä arvoja. Laskettujen arvojen tulee täytyä automaattisesti menetelmäkoepöytäkirjan pätevyysalueen kentille. Lisäksi menetelmäkoepöytäkirjan pohjalta tulee pystyä kopioimaan hitsausohjeita, ja arvojen näille ohjeille tulee kopioitua menetelmäkoepöytäkirjaan lasketulta pätevyysalueelta.

### Standardit

Standardit ovat julkaisuja, joihin on kirjattu yhteisesti sovittuja vaatimuksia, suosituksia tai vaikkapa ominaisuuksia tuotteille ja niiden valmistukselle tai testaukselle sekä järjestelmille tai palveluille (SFS ry. Mikä on standardi). Työssä toteutettujen automaattisesti laskettujen pätevyysalueiden on vastattava menetelmäpöytäkirjaan valitun standardin mukaisia arvoja. Menetelmäkoepöytäkirjan on tuettava muun muassa EN ISO 15614-1:2017-, Bureau Veritas NR216-, DNV-RU-SHIP-, Norsok M-101- sekä Lloyd's register –standardeja. Jokaisella standardilla on omat pätevyysalueiden laskentaan vaikuttavat säännöt, jotka tulee huomioida niin menetelmäkoepöytäkirjan rakenteessa, näkymässä, että pätevyysalueiden laskennassa.

### Brändi

Yksinkertaistettuna brändi on mielikuva kuulijan ja kokijan pään sisällä. Brändit ovat kuitenkin myös rahan arvoista pääomaa, ja asiakkaalle brändi on lupaus, joka helpottaa päätöksentekoa. (Helsingin yliopisto)

Kempillä brändi on tärkeässä osassa ja kuten kuvasta 17 ilmenee, Kempillä on myös tarkat ohjeet brändin toteutumisen varmistamiseksi. (<https://brand.kemppi.com/en-US/> ). Menetelmäkoepöytäkirjan ulkoasun on mukailtava WeldEye-palvelun sekä Kemppi Oy:n brändi ohjeita. Lisäksi toiminnallisuuden on oltava tasolla, joka vastaa Kemppi Oy brändin asettamia odotuksia. (Kemppi Oy b)

# Online Brand Guide

The external appearance of a company is the most visible element contributing to its image and the message it conveys.

This **Online Brand Guide** introduces you to the Kemppi brand and gives practical instructions on the use of graphic elements for communicating our company image in a coherent manner.



## THE BRAND MAKES US LIVE OUR PURPOSE

Kemppi's purpose as a business is "Building Safer Infrastructure". Our corporate values are People, Courage, Integrity, and Performance. The brand ensures our values are a part of daily decision-making, which consequently guides us to fulfill our purpose.

The brand and image are important assets that have an effect on the perceived value of a company, its products, the quality of operations, reputation, and people.

## Kuva 17. Kemppi brändi

### 4.2 Yleiskatsaus

WeldEye on Kemppi Oy:n kehittämä ja ylläpitämä järjestelmä hitsauksen tukemiseksi. WeldEye-palvelun käyttöliittymä on toteutettu Angular-ohjelmistokehyksellä. Tässä työssä jo olemassa olevan WeldEye -palvelun hitsausohje-moduuliin lisättiin uusi menetelmäkoepöytäkirjadokumentti, jonka näkymä sekä rakenne perustuu eri standardien mukaisiin menetelmäkoepöytäkirjamalleihin. Tämän menetelmäkoepöytäkirjadokumentin tulee olla dynaaminen ja tukea kaikkia vaadittuja standardeja. Käyttöliittymässä menetelmäkoepöytäkirjalle valittu standardi määrittää menetelmäkoepöytäkirjan formaatin, sekä minkä standardin pätevyysalueen laskentaa käytetään pätevyysalueen kenttien laskentaan. Pätevyysalueen laskennan tulee tapahtua automatisoidusti, kun menetelmäkoepöytäkirjaan kirjataan arvo kentälle, jolla on suora vaikutus pätevyysalueeseen.

Itse laskennat toteutettiin luomalla npm paketti, jossa jokaiselle standardille on oma luokkansa, joissa pätevyysalueiden laskenta käsitellään. Tämä paketti sitten tuodaan

käyttöliittymään, missä sen toiminnallisuutta käytetään pätevyysalueiden laskennassa ja täytössä. Pätevyysalueiden laskennan oikeanmukaisen toiminnallisuuden varmistamiseksi kaikille laskennoille on oltava yksikkötestit, missä varmistetaan itse laskentojen oikeellisuus ja niiden vastaavuus määritettyihin standardeihin ja vaatimuksiin. Tämä toteutetaan käyttämällä hyväksi Mocha testauskehystä sekä Chai todennuskirjastoa. Paketti pidetään yksityisessä npm rekisterissä, josta se voidaan helposti hakea kaikkiin sovelluksiin, joissa sitä mahdollisesti tarvitaan.

### 4.3 Npm-paketti

Laskentoja varten luotiin npm-paketti. Paketista löytyy index.ts-tiedosto (Kuva 18), joka toimii paketin pääsisääntipisteinä, mistä tuodaan kaikki eri standardien laskentaluokat ja mistä ne voidaan viedä eteenpäin. Paketti bundlataan JavaScript tiedostoksi webpack-bundlerin avulla, joka ladataan Kempin omaan npm rekisteriin mistä sen voi ladata muihin projekteihin käytettäväksi. Paketin package.json:niin määriteltiin paketin nimi, kuvaus, versio, sekä pääsisääntipiste mistä paketin koodi tuodaan (import) muihin koodikantoihin. Kun versio ladataan rekisteriin, sen versio päivittyy package.json:ista löytyvän version mukaan. Package.json:iin on määritely myös mihin rekisteriin paketti ladataan.

```
wp > qualification-range-calculation > src > TS index.ts
You, 1 second ago | 1 author (You)
1 export { ISO1561412017Calculation } from './ISO-15614-1-2017/ISO-15614-1-2017-calculation';
2 export { DNVRUSHIPCalculation } from './DNV-RU-SHIP/DNV-RU-SHIP-calculation';
3 export { PACACAL056Calculation } from './P-ACA-CAL-056/P-ACA-CAL-056-calculation';
4 export { BureauVeritasNR216Calculation } from './Bureau-Veritas-NR216/Bureau-Veritas-NR216-calculation';
5 export { NorsokM101Calculation } from './Norsok-M-101/Norsok-M-101-calculation';
6 export { LlyodsRegisterCalculation } from './Lloyds-Register/Llyods-Register-calculation';
7
```

Kuva 18. Paketin pääsisääntipiste

Paketissa on myös mukana taulu, missä on kaikkien tuettujen standardien tunnisteen (kuva 19), sekä tyypitykset kaikille paketissa tarvittaville tiedoille.

```
155 export const supportedStandards = [
156   'DNV-RU-SHIP' ,
157   'ISO 15614-1:2017' ,
158   'P-ACA-CAL-056' ,
159   'NR 216 DT R13' ,
160   'LR RULES 2021' ,
161   'NORSOK M-101' ,
162 ] as const;
163
164 export type SupportedStandard = typeof supportedStandards[number];
```

Kuva 19. Paketin tukemat standardit

Laskennan luokan tulee implementoida laskenta standardi –tyyppi (Kuva 20), joka määrittää luokan rajapinnat. Laskentaluokalla on vain yksi metodi, joka ottaa parametrina laskentoja varten tarvittavat parametrit ja palauttaa pätevyysalueiden laskentojen tulokset.

```
export type CalculationStandard = {
  constructor: Function;
  calculateRangeOfQualification: (essentialVariables: EssentialVariables) => RangeOfQualification;
};
```

Kuva 20. Laskentaluokan tyyppi

Laskentaluokat toimivat seuraavalla tavalla: Laskentaluokan instanssi initialisoidaan käyttöliittymässä, kun laskentaluokkaa tarvitaan. Luokan pätevyysalueen laskenta –metodin tarkoituksena on laskea dokumentilta tulleiden arvojen pohjalta standardiin perustuvat pätevyysalueet eri kentille. Näiden arvojen pohjalta luodaan pätevyysalue –objekti, joka sisältää kaikki lasketut pätevyysalueet ja palautetaan käyttöliittymälle. Koska eri standardeilla on eri vaatimukset, tulee oleellisten muuttujien ja pätevyysalue-tyyppien olla omanlaisensa jokaiselle standardille (Kuva 21).

```
export type EssentialVariables =
  { standard: 'DNV-RU-SHIP' } & DNVRUSHIPEssentials |
  { standard: 'ISO 15614-1:2017' } & ISO15614Essentials |
  { standard: 'P-ACA-CAL-056' } & PACACAL056Essentials |
  { standard: 'NR 216 DT R13' } & BureauVeritasNR216Essentials |
  { standard: 'LR RULES 2021' } & LlyodsRegisterEssentials |
  { standard: 'NORSOK M-101' } & NorsokM101Essentials ;

export type RangeOfQualification =
  { standard: 'DNV-RU-SHIP' } & DNVRUSHIPQualificationRange |
  { standard: 'P-ACA-CAL-056' } & PACACAL056QualificationRange |
  { standard: 'ISO 15614-1:2017' } & ISO15614QualificationRange |
  { standard: 'NR 216 DT R13' } & BureauVeritasNR216QualificationRange |
  { standard: 'LR RULES 2021' } & LlyodsRegisterQualificationRange |
  { standard: 'NORSOK M-101' } & NorsokM101QualificationRange ;
```

Kuva 21. Tyypit laskennoille

Laskentaluokilla on oma toteutuksensa pätevyysalueen laskenta –metodista. Alussa kaikissa toteutuksissa parsitaan käyttöliittymältä tulleista arvoista oleelliset arvot käyttämällä JavaScript–ohjelmointikielen dekonstruointi ominaisuutta. Sen jälkeen dokumentilta löytyville prosesseille lasketaan prosessi kohtaiset pätevyysalueet. Tämän jälkeen jokaiselle kentälle, jolle tulee laskea pätevyysalue, lasketaan pätevyysalue ja lisätään laskennan tulos oikeaan kenttään käyttöliittymälle palautettavaan objektiin (Kuva 22).

```

61 public calculateRangeOfQualification(essentialVariables: EssentialVariables): RangeOfQualification {
62     if (essentialVariables.standard === 'ISO 15614-1:2017') {
63         const {
64             calculationLevel,
65             isImpactTestRequired,
66             parentMaterials,
67             productType,
68             weldingProcesses,
69             branchConnectionAngle,
70             // ...
71         } = essentialVariables;
72
73         const processesWithQualificationRanges = new Map<ProcessIdentifier, WeldingProcessRoQ>();
74
75         for (const [ processIdentifier, processUsed ] of weldingProcesses.entries()) {
76
77             const RoQForProcess: WeldingProcessRoQ = {
78                 // Most of the process data should be 1:1, override properties that should be calculated
79                 ...processUsed,
80                 depositedWeldMetalThickness: this.calculateDepositedWeldMetalThicknessRange(parentMaterials,
81                 currentAndPolarity: this.getAllowedCurrentsAndPolarities(processUsed.currentAndPolarity, p
82                 processTypes: this.getAllowedWeldingProcessTypes(calculationLevel, processUsed.processTypes
83             };
84
85             processesWithQualificationRanges.set(processIdentifier, RoQForProcess);
86         }
87
88         const rangeOfQualification: RangeOfQualification = {
89             standard: essentialVariables.standard,
90             weldingProcesses: processesWithQualificationRanges,
91             parentMaterialThicknessRanges: {
92                 A: this.calculateParentMaterialThicknessRange(parentMaterials.A.thickness, parentMaterials
93                 B: this.calculateParentMaterialThicknessRange(parentMaterials.A.thickness, parentMaterials
94             },
95             branchConnectionAngleRange: this.calculateBranchConnectionAngleRange(productType, branchConnect
96             outsidePipeDiameterRange: this.calculateOutsidePipeDiameterRange(calculationLevel, parentMater
97             parentMaterialGroups: this.getAllowedMaterialGroups(parentMaterials.A.materialGroup, parentMat
98             allowedProcesses: this.getAllowedWeldingProcesses(weldingProcesses),
99             //...
100         };
101
102         return rangeOfQualification;
103     }
104 }

```

Kuva 22. Pätevyysalueen laskenta

Koska jotkut standardit perustuvat toisiin standardeihin, voidaan laskentaluokkien toteutuksessa käyttää samankaltaista lähestymistapaa. Eli joissakin laskentaluokissa voidaan hyödyntää jo olemassa olevaa luokkaa. Tällaisissa tapauksissa luokkaa laajennetaan ja muutetaan vain joidenkin laskentojen toiminta. Kuva 23 havainnollistaa tätä. Tämä rajoittaa kirjoitettavan koodin määrää, testaamista sekä ylläpidettävyyttä.

```

26 export class PACACAL056Calculation extends ISO1561412017Calculation {
27
28     constructor() {
29         super();
30     }
31
32     useHLESCalculations = () => {
33         this.calculateParentMaterialThicknessRange = calculateParentMaterialThicknessRangeForHLESMaterials;
34         this.calculateHeatInputRange = calculateHeatInputRangeForHLESMaterials;
35         this.calculateDepositedWeldMetalThicknessRange = calculateDepositedWeldMetalThicknessRangeForHLESMaterials;
36     }
37
38     useDefaultCalculations = () => {
39         this.calculateParentMaterialThicknessRange = defaultParentMaterialThicknessRangeCalculation;
40         this.calculateHeatInputRange = defaultHeatInputRangeCalculation;
41         this.calculateDepositedWeldMetalThicknessRange = defaultWeldMetalThicknessRangeCalculation;
42     }
43
44     calculateRangeOfQualification = (essentialVariables: EssentialVariables): RangeOfQualification => {
45         if (essentialVariables.standard === 'P-ACA-CAL-056') {
46             const {
47                 calculationLevel,
48                 parentMaterials,
49                 isImpactTestRequired,
50                 weldingProcesses,
51                 // ...
52             } = essentialVariables;
53
54             if (parentMaterials.A.HLES || parentMaterials.B.HLES) {
55                 this.useHLESCalculations();
56             } else {
57                 this.useDefaultCalculations();
58             }
59         }
60     }
61 }

```

Kuva 23. Esimerkki standardin laajentamista toisesta standardista

Standardista löytyvät pätevyysalueiden laskennat (Kuva 24) on pilkottu yksittäisiksi funktioiksi (Kuva 25), jotta koodin ylläpitäminen, testaaminen, sekä luettavuus olisi helpompaa.

**Table 18 Qualified range for pipe and branch connection diameters**

| Diameter of the test piece, D [mm] <sup>1)</sup> | Qualification range [mm] |
|--|--------------------------|
| D ≤ 25   | 0.5 × D to 2 × D         |
| D > 25   | ≥ 0.5 × D and plates     |

1) D is the outside diameter of the pipe or outside diameter of the branch pipe

*Angle of branch connections*

A WPQT carried out on a branch connection with angle  $\alpha$  shall qualify all branch connection angles in the range of  $\alpha$  to 90°.

Kuva 24. Esimerkki standardista löytyvästä laskentaohjeesta (DNV AS)

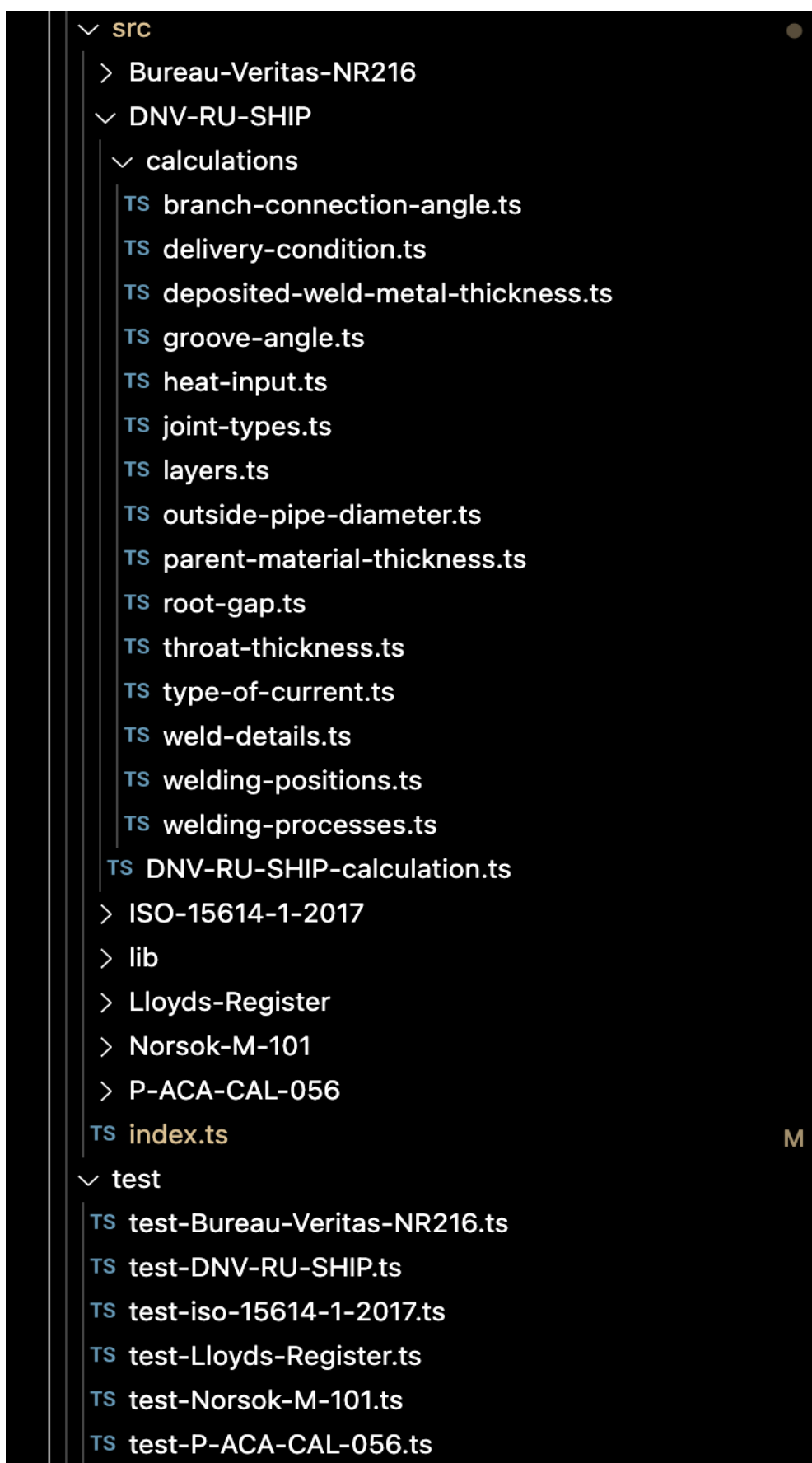
```

3  /**
4  *
5  * Calculates the minimum and maximum allowed outside pipe diameter for the weld
6  * based on the test piece outside pipe diameter and other parameters according to the DNV-RU-SHIP standard.
7  *
8  * @see {@link https://kempi.atlassian.net/wiki/spaces/WAL/pages/3913875521/Range+of+qualification+for+DNV-R}
9  *
10 * @param { MinMax } diameterA Test piece A diameter
11 * @param { MinMax } diameterB Test piece B diameter
12 *
13 * @returns { QualificationRange } The minimum and maximum allowed outside pipe diameter for the weld.
14 */
15 export const calculateOutsidePipeDiameterRange = (
16   diameterA: MinMax,
17   diameterB: MinMax
18 ): QualificationRange => {
19   let min: number | null = null;
20   let max: number | null = null;
21
22   /** Diameter of the test piece */
23   const D = getMinNumber([ diameterA, diameterB ]);
24
25   if (D <= 25) {
26     min = 0.5 * D;
27     max = 2 * D;
28   }
29
30   if (D > 25) {
31     min = 0.5 * D;
32     max = D;
33   }
34
35   return {
36     min: roundToTwoDecimals(min) ?? null,
37     max: roundToTwoDecimals(max) ?? null
38   };
39 };

```

Kuva 25. Laskennan toteuttaminen koodissa

Koodi on lajiteltu niin, että jokaiselle standardille on oma kansionsa, josta löytyy itse laskentaluokan toteutus. Lisäksi kaikille yksittäisille laskennoille on oma kansionsa. Myös testit kaikille laskennoille on omassa kansiossaan (Kuva 26).



Kuva 26. Paketin kansiorakenne

Pätevyysalueiden laskenta-funktioiden sekä laskentaluokkien testaaminen on toteutettu käyttämällä Mocha testauskehystä sekä Chai todennuskirjastoa. Kuvassa 27 on esimerkki testeistä, joita pakettile ajetaan. Kuvassa 28 näkyy testien tulokset. Jokaiselle laskentaluokalle on määritelty testit sekä yksittäisten arvojen laskennalle, että laskentaluokan pätevyysalueen laskenta –metodille. Testien on varmistettava, että laskentojen tulokset ovat oikein ja vastaavat standardissa määriteltyjä arvoja. Paketin package.json -tiedostoon on lisätty skriptit paketin testaamista ja päivittämistä varten. Skripteihin on myös sidottu, että kun paketista ladataan uusi versio npm rekisteriin, pakettile ajetaan ensin testit. Näin varmistetaan, ettei rekisteriin pääse viallisia versioita.

```
23 describe('DNV-RU-SHIP', () => {
24   describe('Calculations', () => {
25     describe('Branch connection angle calculation | branch-connection-angle.ts | calculateBranchConnectionAngleRange()', () => {
26       describe('Should correctly calculate the range for branch connection angle when the angle is', () => {
27         const testCases = [{
28           description: '0',
29           angle: 0,
30           expected: 0
31         }, {
32           description: '"0"',
33           angle: '0',
34           expected: 0
35         }, {
36           description: '29.99',
37           angle: 29.99,
38           expected: 29.99
39         }, {
40           description: 'undefined',
41           angle: undefined,
42           expected: null
43         }, {
44           description: '"some string"',
45           angle: 'some string',
46           expected: null
47         }
48         //...
49       ];
50
51       for (const { description, angle, expected } of testCases) {
52         it(`${description}`, () => {
53           const { min, max } = calculateBranchConnectionAngleRange(angle as any);
54           expect(max).to.equal(90);
55           expect(min).to.equal(expected);
56         });
57       }
58     });
59   });
60 }
```

Kuva 27. Esimerkki laskennan testeistä

```

DNV-RU-SHIP
Calculations
Branch connection angle calculation | branch-connection-angle.ts | calculateBranchConnectionAngleRange()
Should correctly calculate the range for branch connection angle when the angle is
  ✓ 0
  ✓ "0"
  ✓ 29.99
  ✓ "29.99"
  ✓ null
  ✓ undefined
  ✓ "undefined"
  ✓ "some string"
Outside pipe diameter calculation | outside-pipe-diameter.ts | calculateOutsidePipeDiameter()
Should correctly calculate the minimum allowed diameter when diameters are
  ✓ 0 && 1
  ✓ 1 && 0
  ✓ 0.1 && 0.2
  ✓ 0.2 && 0.1
  ✓ 1 && 1.01
  ✓ null && null
  ✓ null && 1
  ✓ 1 && 2
  ✓ undefined && 2
  ✓ undefined && undefined
  ✓ 25 && 25
  ✓ 24.99 && 25.01
  ✓ 25.02 && 25.01
Should correctly calculate the maximum allowed diameter when diameters are
  ✓ 0 && 1
  ✓ 1 && 0
  ✓ 0.1 && 0.2
  ✓ 0.2 && 0.1
  ✓ 1 && 1.01
  ✓ null && null
  ✓ null && 1
  ✓ 1 && 2
  ✓ undefined && 2
  ✓ undefined && undefined
  ✓ 25 && 25
  ✓ 24.99 && 25.01
  ✓ 25.02 && 25.01

```

Kuva 28. Laskennan testien tulokset

## 4.4 Käyttöliittymä

### 4.4.1 Menetelmäkoepöytäkirjekomponentti

WeldEye-palvelu on toteutettu Angular-ohjelmistokehyksellä. WeldEye-palvelussa on jo olemassa hitsausohjemoдуули, joka tukee erilaisten hitsausohje-dokumenttien hallintaa palvelussa. Tässä työssä hitsausohje-moduuliin lisätään tuki uudelle menetelmäkoepöytäkirja-dokumentille. Olemassa olevat hitsausohjeet on jaoteltu hitsausohje-moduulin sisällä omiin moduuleihinsa. Täten myös menetelmäkoepöytäkirjalle lisätään oma moduulinsa (Kuva 29). Hitsausohje-moduulissa on olemassa valmiiksi jo paljon ennalta luotua toiminnallisuutta, kuten komponentteja ja palveluita, joihin uuden menetelmäkoepöytäkirja-moduulin, palveluiden, sekä komponenttien tulee integroitua sujuvasti ja saumattomasti.

```

37 @NgModule({
38   imports: [
39     BrowserModule,
40     WpWpsBaseModule,
41     WpLayoutComponentsModule,
42   ],
43   declarations: [
44     Wpqr,
45     WpqrCoverPageComponent,
46     WpqrRecordOfWeldTestComponent,
47     WpqrTestResultsComponent,
48     PageDisclaimerComponent,
49     WpqrTensileTestsComponent,
50     WpqrBendTestsComponent,
51     WpqrToughnessTestsComponent,
52     WpqrHardnessTestsComponent,
53     WpqrTestResultsRowComponent,
54     WpqrNonDestructiveExaminationsComponent,
55     WpqrOtherTestsComponent,
56     WpqrTestInformationComponent,
57     WpqrTestInfoRowComponent,
58     FilletWeldFractureTestsComponent,
59     RangeOfQualificationComponent,
60     WeldingProcessesComponent,
61   ],
62   exports: [
63     Wpqr,
64   ]
65 })
66 export class WpWpqrModule {
67   constructor(
68     private WpTemplateConfigurationService: WpTemplateConfigurationService,
69     private dropDownHandlerService: DropDownHandlerService,
70   ) {
71     this.WpTemplateConfigurationService.registerTemplate('wpqr', {
72       parse,
73       generate,
74       validate,
75       copyTargetFilter,
76       layoutFunctionsForRoQCalculation: new WpqrLayoutFunctions(this.dropDownHandlerService),
77       prefetchDropdowns: [
78         'commonClient',
79         'commonCompanies',
80         'commonLocation',
81         'commonProject',
82         'commonTechnique',
83         'commonWeldingEquipment'
84       ]
85     });
86   }
87 }

```

Kuva 29. Menetelmäkoepöytäkirja-moduuli

Menetelmäkoepöytäkirjalle luodaan myös oma komponentti, joka lisätään menetelmäkoepöytäkirja-moduuliin tuotavaksi komponentiksi. Tämä komponentti jaetaan osiksi pienempiin komponentteihin uudelleenkäytettävyyden, luettavuuden, sekä ylläpidon helpottamiseksi. Koska menetelmäkoepöytäkirja koostuu kolmesta eri pääsivusta: pätevyysalueesta (range of qualification), menetelmäkokeesta (record of weld test) sekä testituloksista (test results), voidaan itse menetelmäkoepöytäkirja komponentille luoda suoraan kolme erillistä komponenttia, jotka vastaavat näitä edellä mainittuja sivuja (Kuva

30). Näin kunkin sivun rakenne, muotoilu, data, sekä toiminnallisuus on helpommin hallittavissa ja niitä voidaan tarvittaessa käyttää uudelleen myös eri komponenteissa.

```

1  <!-- ----- -->
2  <!-- Page 1: Cover Page -->
3  <!-- ----- -->
4  <wpqr-cover-page
5      *ngIf="state"
6      [state]="state"
7      [automaticRoqCalculationIsSupported]="automaticRoqCalculationIsSupported"
8      [dropdownOptions]="dropdownOptions"
9      (onLoad)="printWaiter.ready('cover')"
10     (wpqrStandardChange)="refreshWpqrStandard($event)"
11     (roqDataChange)="updateRangeOfQualification()"
12     (roqToggle)="toggleRangeOfQualificationCalculation($event)">
13 </wpqr-cover-page>
14
15 <!-- ----- -->
16 <!-- Page 2: Record of Weld Test -->
17 <!-- ----- -->
18 <wpqr-record-of-weld-test
19     *ngIf="state"
20     #recordOfWeldTest
21     [state]="state$.recordOfWeldTest"
22     [dropdownOptions]="dropdownOptions"
23     (onLoad)="printWaiter.ready('recordOfWeldTest')"
24     (weldingProcessChanges)="handleWeldingProcessChanges($event)"
25     (refreshReadOnlyStatus)="refreshReadOnlyStatus()">
26 </wpqr-record-of-weld-test>
27
28 <!-- ----- -->
29 <!-- Page 3: Test results -->
30 <!-- ----- -->
31 <wpqr-test-results
32     *ngIf="state"
33     [state]="state"
34     (onLoad)="printWaiter.ready('testResultsPage')"
35 ></wpqr-test-results>

```

Kuva 30. Menetelmäkoepöytäkirja-komponentin näkymä

Myös näille sivuille luodaan erillisiä komponentteja esimerkiksi erilaisille tauluille, tai kenttäryhmille samasta edellä mainitusta syystä. Tämä ilmenee myöhemmin esimerkiksi menetelmäkoe sivun kohdalla, koska menetelmäkoe sivu vastaa lähes täysin itse hitsausohjetta. Tämä mahdollistaa saman komponentin käytön niin menetelmäkoepöytäkirjan menetelmäkoe sivulla, kuin myös itse hitsausohje dokumentilla.

Kun käyttäjä luo uuden menetelmäkoepöytäkirjadokumentin, sen data alustetaan alkuun tyhjällä kaaviolla (Kuva 31). Kaaviossa on mukana kaikki dokumentille tarvittavat kentät, eri standardien mukaan vaaditut kentät, sekä monia dokumentin tilan hallintaan tarvittavia

kenttiä. Näillä kentillä ylläpidetään tietoa esimerkiksi siitä, onko automaattinen laskenta kytketty päälle vai ei.

```

656 export const wpqrSchema: WpqrFormSchema = {
657   $: null,
658   active: true,
659   approvals: {},
660   data: {
661     companyLogo: null,
662     name: '',
663     reference: null,
664     attachments: [],
665     additionalPages: [],
666     dateOfIssue: null,
667     details: {
668       process: [],
669       processType: [],
670       joint: {
671         type: [],
672       },
673       weldDetails: [],
674       position: [],
675       metalThickness: {
676         min: null,
677         max: null,
678       },
679       throatThickness: {
680         min: null,
681         max: null,
682       },
683       grooveAngle: {
684         min: null,
685         max: null,
686       },
687       rootGap: {
688         min: null,
689         max: null,
690       },
691       flux: {
692         designation: '',
693       },
694       waveformControl: [],
695       metalTransferMode: [],

```

Kuva 31. Kaavio objekti

Datan sidonta komponentille toteutetaan hyödyntämällä olemassa olevaa hitsausohjeen templaatti sivun konfiguraatio palvelua, joka toimii hitsausohje templaattisivu-komponentin apuna. Palvelussa hallitaan eri dokumenttien tilaa. Kun käyttäjä lisää tai valitsee menetelmäkoepöytäkirjan, käyttäjä uudelleenohjataan hitsausohje templaatti sivulle ja dokumentin tyyppi asetetaan URL-osoitteen kyselyparametreihin mukaan. Sivua

ladattaessa kyselyparametreihin määritelty dokumentin tyyppi määrittää sivulla, mikä dokumentti näkymälle ladataan (Kuvat 32, 33).

```
this.layout = this.route.snapshot.queryParams['layout'];
```

Kuva 32. Valitun dokumentin parsinta URL-osoitteesta

```
52 <wpqr
53   *ngIf="layout === 'wpqr'"
54   [state]="state"
55   [can]="can"
56   [print]="printing"
57   [fullReadOnly]="fullReadOnly"
58   [useAutomatedPrintHeaders]="useAutomatedPrintHeaders"
59   [licenses]="licenses"
60   [settings]="settings"
61   (onLoad)="setPageLoaded()"
62   (shortenBarcode)="shortenBarcode()"
63 ></wpqr>
```

Kuva 33. Menetelmäkoepöytäkirjan näyttäminen templaattisivulla

Kuten kuvasta 33 näkyy, templaattisivu mahdollistaa muun muassa latauksen hallinnan näyttämällä sivulla lataus indikaattoria, kunnes ladattava komponentti lähettää signaalin (Kuva 33, rivi 61) kertoakseen olevansa valmis näytettäväksi.

Menetelmäkoepöytäkirjan ensimmäiseltä sivulta (Kuva 34) löytyy pätevyysalue (range of qualification), sekä muita dokumentille kuuluvia kenttiä, kuten dokumentin nimi, standardi, jonka vaatimuksia dokumentti vastaa, tilan hallintaan liittyviä kytkimiä kuten automaattisen laskennan ja täytön kytkin sekä muita kenttiä. Seuraavilta sivuilta löytyvät menetelmäkokeen tiedot (record of weld test) sekä lopusta menetelmäkokeen testitulokset (test results). Sivujen määrä määräytyy dynaamisesti riippuen prosessien ja merkattujen hitsipalko rivien määrästä. Pätevyysalueen, menetelmäkokeen, sekä testisivujen kentät vastaavat standardeilta löydettäviä kenttiä.

WELDING PROCEDURES > NEW PROCEDURE WeidEye Teemu Hirvonen

**KEMPPI**  
Manufacturer

Address:

Office:

Location:

Revision level: **Level 1** **Level 2**  **Off**  **On**

Approved:  Impact tests required  Hardness tests required  Approval test fulfills requirements: PED 2014/68/EU

**WELDING PROCEDURE QUALIFICATION RECORD (WPQR)**

**ISO 15614-1:2017**

WPQR number:

Revision date: 07.04.2024 Revision: 0

Referenced standards:  Notified body:

Reference specification:  Date of issue:

1

2

3

4

5

**RANGE OF QUALIFICATION**

Welding process(es):  Single run / Multi run:  Joint type(s):  Weld details:

Welding position(s):

Parent material group(s):  Delivery condition:  Notes:

Throat thickness (mm):  Root gap (mm):  Parent material thickness (mm):  Outside pipe diameter (mm):

Min:  Max:  Min:  Max:  Min:  Max:

Preheat temperature:  Interpass temperature:  Post Temperature (°C):  Heating rate:

Min:  Max:  Min:  Max:  Min:  Max:  Duration:  Cooling rate:

Notes:

Multi-process joint:  Yes  No

Welding process(es):  Welding process type:  Type of welding current and polarity:  Mode of metal transfer:

Weld metal thickness (mm):  Filler material size (mm):  Shielding gas:  Backing gas:

Min:  Max:  Min:  Max:

Filler material trade name:  Filler material type:  Filler material classification(s):  Flux designation:

**WELDING PARAMETERS**

| Pass                 | Welding process      | Filler # (mm)        | Current (A)   | Voltage (V)   | Travel speed (mm/min)                               | Heat input (kJ/mm)                                  |
|----------------------|----------------------|----------------------|---|---|---|---|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | Min: <input type="text"/> Max: <input type="text"/> | Min: <input type="text"/> Max: <input type="text"/> | Min: <input type="text"/> Max: <input type="text"/> | Min: <input type="text"/> Max: <input type="text"/> |

[+ ADD PARAMETER ROW](#)

**OTHER INFORMATION**

Certified that test welds prepared, welded and tested satisfactorily in accordance with the requirements of the code/testing standard indicated above.

Add signature and save to lock editing of this document.

**KEMPPI**  
WPQR number:   
Company name:

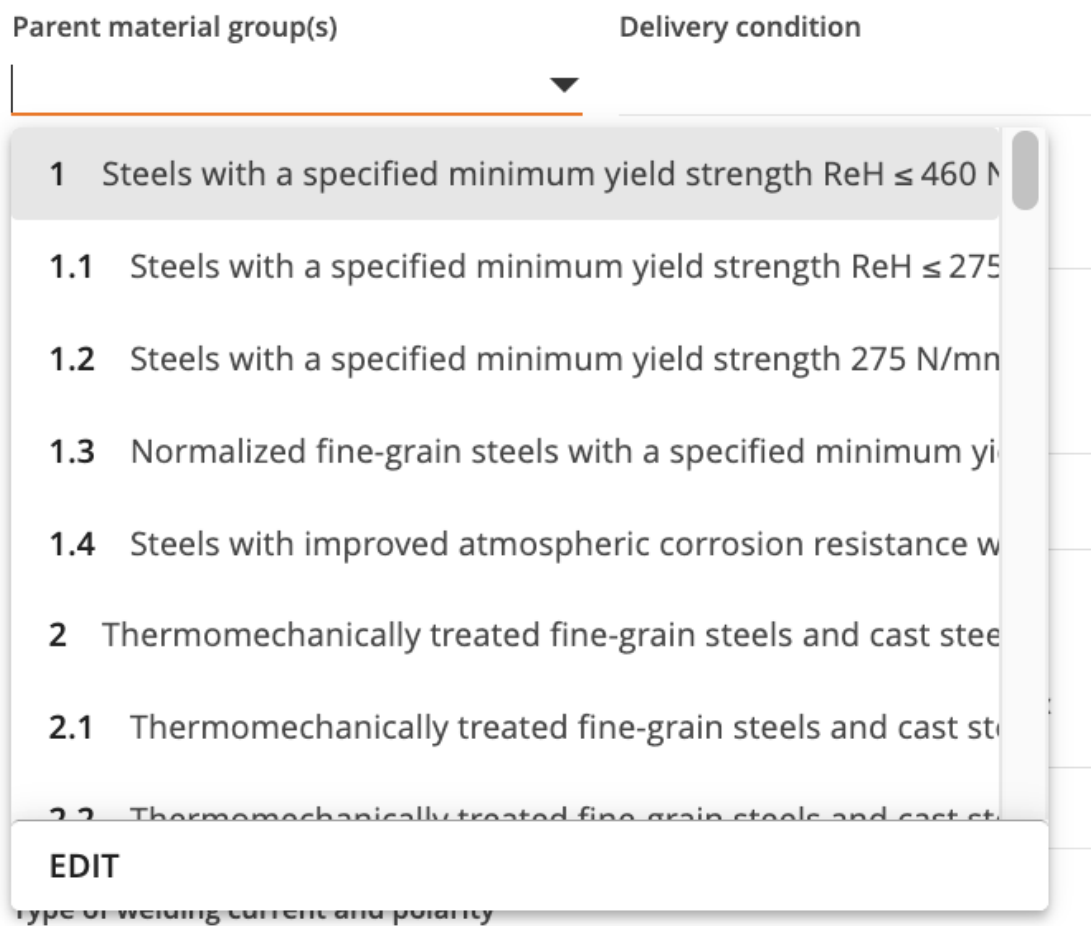
**Record of weld test**

Revision date: 07.04.2024 Revision: 0

REVISIONS: **Active**

Kuva 34. Menetelmäkoepöytäkirjan ensimmäinen sivu

Digitaalisen pöytäkirjan etuna on, että tietyille kentille voidaan muun muassa määrittää alusvetovalikoihin ennalta määritellyjä arvoja, jotka nopeuttavat ja helpottavat pöytäkirjan täyttöö (Kuva 35).



Kuva 35. Esimerkki alasvetovalikko

Lisäksi digitaalisen pöytäkirjan dynaaminen luonne mahdollistaa saman dokumentin käyttöä tilanteissa, joissa prosesseja, testejä, hitsipalko rivejä tai mitä vain muita arvoja voi olla mikä tahansa määrä. Dokumentin tilaa voi myös helposti hallita jälkikäteen, ja siihen voi tehdä muutoksia tarvittaessa helposti. Esimerkkinä sivun otsikko riviltä löytyvä standardi on valittava teksti (Kuva 36). Tämä mahdollistaa saman menetelmäkoepöytäkirjan käytön, kun tarvitaan menetelmäkoepöytäkirjaa, joka vastaa eri standardia. Kun käyttäjä klikkaa standardi tekstiä, aukeaa alasvetovalikko mistä käyttäjä voi valita jonkun toisen standardin (Kuva 37).

**KEMPPI**  
Manufacturer

Address

Office

Location

Referred standards

Reference specification

**WELDING PROCEDURE QUALIFICATION RECORD (WPQR) DNV-RU-SHIP**

Kuva 36. Valittava standardi teksti

**KEMPPI**  
Manufacturer

Address

Office

Location

Referred standards

Reference specification

Autofill  
**Off** On

Fracture tests required

**RANGE OF QUALIFICATION**


WPQR r  
isowp  
Referer

ISO 15614-6 Welding procedure test...  
DNV-RU-SHIP DNV-RU-SHIP  
P-ACA-CAL-056 P-ACA-CAL-056  
NORSOK M-101 NORSOK M-101  
LR RULES 2021 LR RULES 2021  
NR 216 DT R13 NR 216 DT R13

Kuva 37. Standardi alasvetovalikko

Koska eri standardeilla voi olla eri vaatimuksia, kuten erilaisia täytettäviä kenttiä, tämän avulla yhdestä menetelmäpöytäkirjasta voi luoda vastaavia dokumentteja kaikille tarvittaville standardeille. Tämä on hallittu näkymässä pääsääntöisesti käyttämällä nglf-direktiiviä, jolla kontrolloidaan mitä kenttiä näytetään milloinkin dokumentille valitun standardin mukaisesti.

Seuraavilta sivuilta löytyvät menetelmäkoe sivut (Kuvat 38, 39), joihin täytetään menetelmäkokeessa hitsatun hitsin tiedot ja parametrit. Näiden sivujen määrä riippuu käytettyjen prosessien ja hitsattujen palkojen määrästä.



WPQR number  
**WPQR Demo**  
Company name  
**JPP Soft Oy**

## Record of weld test

Revision date  
**04.05.2023**

Revision  
**0**

JOINT DESIGN

ADD
SKETCH LIBRARY
DRAW

WELDING SEQUENCE

ADD
SKETCH LIBRARY

DETAILS

| Joint type | Product type | Sides     | Groove angle (°) |             | Root gap [mm] |            |
|------------|--------------|-----------|------------------|-------------|---------------|------------|
|            |              |           | Min              | Max         | Min           | Max        |
| <b>BW</b>  | <b>P</b>     | <b>SS</b> | <b>45</b>        | <b>- 65</b> | <b>4</b>      | <b>- 6</b> |
| Remarks    |              |           |                  |             |               |            |

BACKING

| Backing   | Backing material         | Dimensions |
|-----------|--------------------------|------------|
| <b>mb</b> | <b>Ceramic Backing</b> × |            |

JOINT PREPARATION

|   |             |                               |                               |
|---|-------------|-------------------------------|-------------------------------|
| Initial or Interpass cleaning<br><b>Brushing and grinding without s</b> | Shop primer | Jigs, fixing and pre-plucking | Tack welding length and pitch |
|---|-------------|-------------------------------|-------------------------------|

IDENTIFICATION OF PARENT MATERIAL

Parent materials

| Name/Grade                        | Material group                       | Range                           | Specification                       |
|-----------------------------------|--------------------------------------|---------------------------------|-------------------------------------|
| <b>EN 10025 S460 G2+M</b>         | <b>2.1</b>                           |                                 | <b>ISO TR 15608:2006 2.1</b>        |
| Thickness [mm]                    | Weld metal thickness [mm]            | Diameter [mm]                   | HLES                                |
| Min <b>15.0</b> Max <b>- 15.0</b> | Min <b>15</b> Max <b>- 15.0</b>      | Min <b>N/A</b> Max <b>-</b>     |                                     |
| Angle (°)                         | Product type <b>P</b>                | SMYS (MPa)                      | Manufacturer <b>[object Object]</b> |
| Delivery condition                | <input type="checkbox"/> Is abutting |                                 |                                     |
| <b>A</b>                          |                                      |                                 |                                     |
| <b>EN 10025 S460 G2+M</b>         | <b>2.1</b>                           |                                 | <b>ISO TR 15608:2006 2.1</b>        |
| Thickness [mm]                    | Weld metal thickness [mm]            | Diameter [mm]                   | HLES                                |
| Min <b>15.0</b> Max <b>- 15.0</b> | Min <b>15</b> Max <b>- 15</b>        | Min <b>N/A</b> Max <b>- N/A</b> |                                     |
| Angle (°)                         | Product type <b>P</b>                | SMYS (MPa)                      | Manufacturer <b>281150-02</b>       |
| Delivery condition                | <input type="checkbox"/> Is abutting |                                 |                                     |
| <b>B</b>                          |                                      |                                 |                                     |
| Cast <b>397932</b>                | Remarks                              |                                 |                                     |

PREHEAT

| Preheat temperature [°C]          | Interpass temperature [°C] | Temperature control | Method     |
|-----------------------------------|----------------------------|---------------------|------------|
| Min <b>20.0</b> Max <b>- 20.0</b> | Min <b>&lt; 250.0</b>      | <b>N/A</b>          | <b>N/A</b> |

INTENTIONALLY INCREASE OF PREHEATING TEMPERATURE BEFORE CAPPING RUN

PWHT

| Temperature [°C]                | Temperature increase rate        |
|---------------------------------|----------------------------------|
| Min <b>N/A</b> Max <b>- N/A</b> |                                  |
| Duration [min]                  | Temperature decrease rate        |
| <b>N/A</b>                      |                                  |
| Procedure no.                   | Temperature for free change rate |
|                                 |                                  |

POST HEATING FOR HYDROGEN REMOVAL

| Temperature [°C]          | Temperature increase rate        |
|---------------------------|----------------------------------|
| Min <b>-</b> Max <b>-</b> |                                  |
| Duration [min]            | Temperature decrease rate        |
|                           |                                  |
| Procedure no.             | Temperature for free change rate |
|                           |                                  |

Kuva 38. Menetelmäkoe sivu

### WELDING PROCESS 2

Index **Welding process 2\***  
**1 121**

---

#### IDENTIFICATION OF FILLER MATERIALS

|  |  |                      |
|--|--|----------------------|
| Flux trade name<br><b>OE-SD3 1Ni 1/4 Mo/OP 121 TT - OERLIK</b> | Classification(s)<br><b>EN 756</b> × <b>EN 760</b> × <b>SO / SA</b> ×<br><b>FB 1 55</b> × <b>AC H5</b> × | Specification        |
| Filler metal size (mm)<br>Min <b>4</b> Max <b>- 4</b>          | Deposited Weld Metal (mm)<br>Min <b>7.5</b> Max <b>- 7.5</b>   | Electro-flux (Class) |
| Cast   |  | Remarks              |

---

#### POSITIONS

Welding position  
**PA** ×

Welding progression

Position of Fillet

#### GAS

|  |             |   |
|--|-------------|---|
| Gas(es)<br>Shielding<br><b>ISO 14175-M21</b> | Mixture (%) | Gas flow [l/min]<br>Min <b>21</b> Max <b>- 21</b> |
| Backing                                      |             | Min      Max                                      |

---

#### TECHNIQUE

|   |  |  |  |
|---|--|--|--|
| String or weave bead<br><b>Straight/Weave</b> | Maximum width<br><b>6mm</b>  | Oscillation [mm]<br>Min      Max<br><b>± 6.0</b> | Root face [mm]<br>Min      Max<br><b>-</b> |
| Peening                                       | Method of back gauging<br><b>Air arc or grinder only if needed</b> |  | Depth and shape                            |
| Contact Tube to Work [mm]<br><b>20-25mm</b>   | Single/Multi-pass (per side)<br><b>ml</b>                          |  | Single/Multiple electrodes<br><b>1</b>     |
| Orifice or Gas Cup Size [mm]                  | Nozzle size<br><b>16mm</b>   | Wire angle (°)<br><b>60-70</b>                   | Welding process type<br><b>Automatic</b> × |
| Oscillation amplitude<br><b>N/A</b>           | Oscillation frequency<br><b>N/A</b>                                | Oscillation speed<br><b>N/A</b>                  |  |
| Oscillation dwell time<br><b>N/A</b>          | Oscillation stop time<br><b>N/A</b>                                |  |  |

---

#### ELECTRICAL CHARACTERISTICS

|   |                             |                             |
|---|-----------------------------|-----------------------------|
| Wire feed speed (m/min)<br>Min      Max | Voltage [V]<br>Min      Max | Current [A]<br>Min      Max |
| Current & Polarity                      | Manufacturer                | Power source                |
| Waveform control mode                   | Synergic curve              | Mode of metal transfer      |

**+ ADD WELDING PROCESS**

---

WPQR number  
**WIK 34 PA**

#### WELDING PARAMETERS

| Pass | Pass group | Electrode | Welding process | Filler ø [mm] | Wire feed speed [m/min] | Current [A] | Voltage [V] | Travel speed [mm/min] | Heat input [kJ/mm] |
|------|------------|-----------|-----------------|---------------|-------------------------|-------------|-------------|-----------------------|--------------------|
|      | 1          | Root      | 136             | 1.2           | 9.0                     | 190         | 23.0        | 17                    | 1.5                |
|      | 2          | Fill      | 136             | 1.2           | 9.0                     | 190         | 23.0        | 26                    | 1                  |
|      | 3          | Fill      | 121             | 4.0           |                         | 500         | 30.0        | 45                    | 2                  |
|      | 4          | Fill      | 121             | 4.0           |                         | 560         | 30.0        | 45                    | 2.2                |
|      | 5          | Cap       | 121             | 4.0           |                         | 560         | 30.0        | 54                    | 1.9                |
|      | 6          | Cap       | 121             | 4.0           |                         | 560         | 30.0        | 54                    | 1.9                |

**+ ADD PARAMETER ROW**

---

Remarks

#### REMARKS

---

#### SIGNATURES

Add signature and save to lock editing of this document.


Manufacturer      Examiner or examining body

ADD SIGNATURE
ADD SIGNATURE

Kuva 39. Menetelmäkokeen prosessi ja palko tiedot

Myös menetelmäkoe sivuilta löytyvät kentät tukevat alavetovalikoita, joista löytyy kentälle sopivat ennalta määritellyt arvot. Tämä on tärkeää laskennan kannalta, koska ilman ennalta määriteltyjä arvoja laskennan toteuttaminen olisi mahdotonta. Lisää monimutkaisuutta tuo WeldEye-palvelussa olemassa oleva kielituki, jonka myötä kenttien arvot näytetään valitulla kielellä, mutta arvolla on myös mukana metadataa, jotta järjestelmässä voidaan käsitellä arvoa muilla kielellä sekä myös laskennoissa. Menetelmäkoe sivuilla kaikille kentille, joilla on suora vaikutus pätevyysalueeseen, on lisätty toiminnallisuus, joka kuuntelee muutoksia arvossa ja lähettää tiedon arvon muuttuessa. Tätä tarvitaan, kun lisätään tuki pätevyysalueen laskennan automatisoinnille.

Menetelmäkoepöytäkirjadokumentin viimeisiltä sivuilta löytyvät menetelmäkokeen testitulokset (Kuva 40). Testituloksilla ei ole suoraa vaikutusta pätevyysalueen laskentaan, mutta ne ovat oleellinen osa menetelmäkoepöytäkirjaa, koska testien tulokset määrittävät, täyttääkö menetelmäkoeksessa hitsattu hitsi standardin asettamat vaatimukset.



# KEMPPI

## Test results

WPQR number  
WPQR Demo  
Reference  
Test results reference

---

**HARDNESS TESTS**

By  Temperature [°C]

| Actions | Area Spot | Metal base |     |     | Z.A.T. |     |     |     | Weld |     |     | Z.A.T. |     |     |     | Metal base |     |     |     |     |
|---------|-----------|------------|-----|-----|--------|-----|-----|-----|------|-----|-----|--------|-----|-----|-----|------------|-----|-----|-----|-----|
|         |           | 1          | 2   | 3   | 4      | 5   | 6   | 7   | 8    | 9   | 10  | 11     | 12  | 13  | 14  | 15         | 16  | 17  | 18  | 19  |
|         | Line 1    | 279        | 279 | 279 | 279    | 279 | 279 | 279 | 279  | 279 | 279 | 279    | 279 | 279 | 279 | 279        | 279 | 279 | 279 | 279 |
|         | Line 2    | 279        | 279 | 279 | 279    | 279 | 279 | 279 | 279  | 279 | 279 | 279    | 279 | 279 | 279 | 279        | 279 | 279 | 279 | 279 |
|         | Line 3    | 279        | 279 | 279 | 279    | 279 | 279 | 279 | 279  | 279 | 279 | 279    | 279 | 279 | 279 | 279        | 279 | 279 | 279 | 279 |

+ADD ROW

Hardness  
**VICKERS HV10.**

Requirement

Microscopic examination

Test results  
**Accepted**

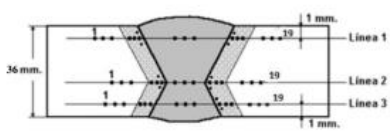
Report no  
**192386**

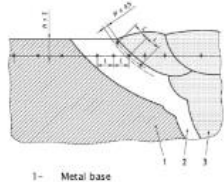
Standard  
**UNE-EN-ISO 6507-1: 2018** ×

Remarks

---

**SKETCHES**

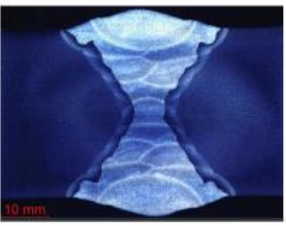




1- Metal base  
2- Z.A.T.  
3- Soldadura

---

**MACRO EXAMINATION (SKETCH)**



Test results  
**Accepted**

Report no  
**192386**

Standard  
**UNE-EN ISO 17639: 2013** ×

Remarks

---

**NON DESTRUCTIVE EXAMINATIONS**

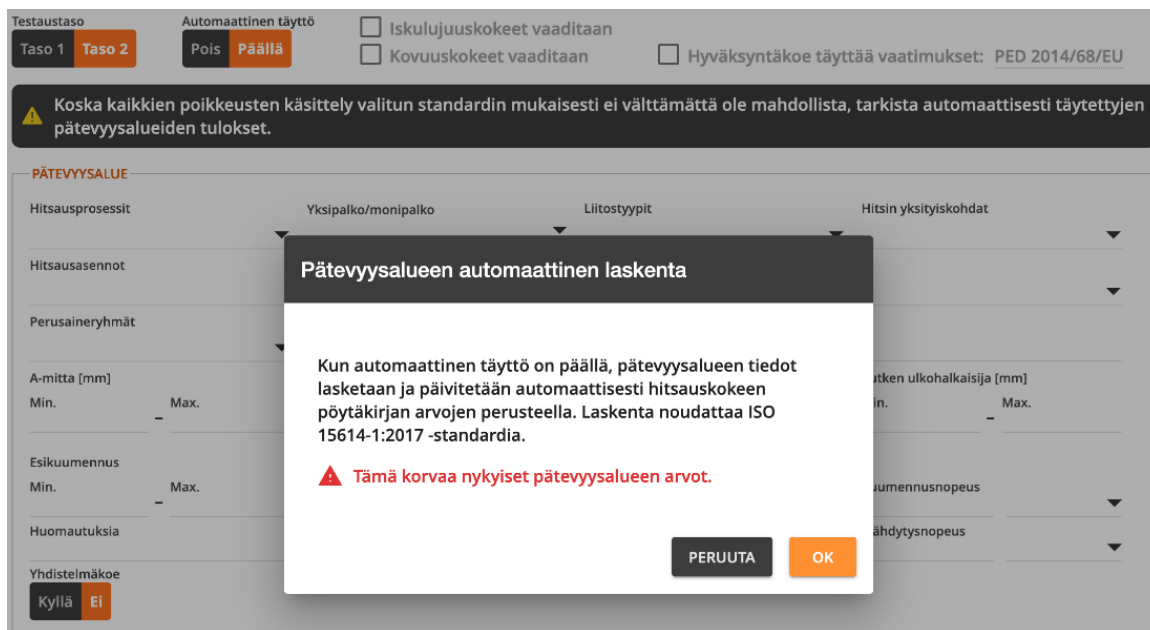
|                                 | Result   | Report no     | Standard               | By   | Date |
|---------------------------------|----------|---------------|------------------------|--|------|
| Visual Inspection               | Accepted | <b>D90681</b> | UNE-EN ISO 17637: 2013 | <span style="border-bottom: 1px solid black; display: inline-block; width: 100px;"></span> |      |
| Visual Inspection before-during |          |               |                        | <span style="border-bottom: 1px solid black; display: inline-block; width: 100px;"></span> |      |
| Magnetic Particle Inspection    | Accepted | <b>D90681</b> | UNE-EN ISO 17637: 2013 | <span style="border-bottom: 1px solid black; display: inline-block; width: 100px;"></span> |      |
| Penetrant Inspection            |          |               |                        | <span style="border-bottom: 1px solid black; display: inline-block; width: 100px;"></span> |      |
| Radiographic Testing            | Accepted | <b>D90681</b> | UNE-EN ISO 17637: 2013 | <span style="border-bottom: 1px solid black; display: inline-block; width: 100px;"></span> |      |
| Ultrasonic Testing              | Accepted | <b>D90681</b> | UNE-EN ISO 17637: 2013 | <span style="border-bottom: 1px solid black; display: inline-block; width: 100px;"></span> |      |

Kuva 40. Sivu menetelmäkokeen testituloksille

#### 4.4.2 Pätevyysalueen laskennan ja täytön automatisointi

Menetelmäkoepöytäkirjan ensimmäiseltä sivulta pätevyysalueen yläpuolelta löytyy kytkin, jolla voidaan kytkeä automaattinen pätevyysalueiden laskenta ja täyttö päälle. Sitä

painettaessa avautuu vielä varmistus dialogi (Kuva 41). Tämä varmistus tehdään, koska pätevyysalueita laskiessa pätevyysalueiden kentät täytetään automaattisesti pätevyyslaskennan tuottamilla arvoilla, jolloin niihin ennalta täytetyt arvot yliajetaan. Laskenta on kytköksissä valittuun standardiin, ja standardin vaihtaminen laskennan ollessa päällä näyttää automaattisen pätevyysalueen laskennan tulokset sille standardille, joka valitaan.



Kuva 41. Varmennus dialogi

Mikäli käyttäjä hyväksyy dialogin, siirrytään menetelmäkoee-sivulle missä ovat kentät, joihin täytetään tieto menetelmäkoeeessa hitsatusta hitsistä ja siinä käytetyistä parametreista. Menetelmäkoepöytäkirja komponentin sisällä rekisteröidään menetelmäkoepöytäkirjan tila pätevyysalueelaskenta-palveluun, jossa itse pätevyysalueen laskenta toteutetaan (Kuva 42).

```

309   toggleRangeOfQualificationCalculation(wasToggledOn: boolean) {
310     if (wasToggledOn) {
311       this.roqCalculationService.registerState(this.state, this.wpqrStandard.systemCode);
312       this.roqCalculationService.changes.pipe(takeUntil(this.unregister$)).subscribe((state) => {
313         this.state.data = state.data;
314       });
315       this.recordOfWeldTest?.nativeElement?.scrollIntoView({ behavior: 'smooth' });
316     } else {
317       this.unregister$.next();
318     }
319   }

```

Kuva 42. pätevyysalueelaskenta-palveluun rekisteröityminen komponentissa.

Pätevyysalueelaskenta-palvelussa on olemassa metodi, joka ottaa parametreina menetelmäkoepöytäkirja-komponentin tilan, sekä sille valitun standardin. Kun

rekisteröinti-metodia kutsutaan, se tarkistaa onko annetulle standardille olemassa tukea pätevyysaluelaskennalle (Kuvat 43, 44).

```
supportedStandards: {
  [key in SupportedStandard]: CalculationStandard
} = {
  'ISO 15614-1:2017': new ISO1561412017Calculation,
  'DNV-RU-SHIP':      new DNVRUSHIPCalculation,
  'P-ACA-CAL-056':    new PACACAL056Calculation,
  'NR 216 DT R13':    new BureauVeritasNR216Calculation,
  'NORSOK M-101':     new NorsokM101Calculation,
  'LR RULES 2021':    new LlyodsRegisterCalculation,
};
```

Kuva 43. Paketin tukemat laskentastandardit

```
125  calculationsExistForStandard(standard: string): standard is SupportedStandard {
126  |   return Object.keys(this.supportedStandards).includes(standard);
127  }
```

Kuva 44. Palvelun tarkistus-funktio

Mikäli standardi tukee pätevyysalueiden laskentaa, asetetaan pätevyysaluelaskenta-palvelussa kyseisen standardin laskentaluokka aktiiviseksi laskentaluokaksi (Kuva 45).

```
64  if (this.standardHasCalculations) {
65  |   this.calculationStandard = this.supportedStandards[standard];
66  }
```

Kuva 45. Palvelun laskenta standardin asettaminen aktiiviseksi

Pätevyysaluelaskenta –palvelussa asetetaan käytettäviksi adapteri funktioiksi dokumentille määritellyt adapteri funktiot. Tämä mahdollistaa samojen laskentojen käytön myös erilaisilla dokumenteilla. Menetelmäkoepöytäkirja-komponentissa tilataan pätevyysaluelaskenta-palvelun RxJs subjekti, kun automaattinen pätevyysalueen laskenta kytketään päälle. Subjektin välittäessä arvoja menetelmäkoepöytäkirjassa päivitetään dokumentin tila subjektin välittämällä arvoilla. Kun automaattinen pätevyysalueen laskenta on päällä ja käyttäjä muuttaa menetelmäkoepöytäkirjassa olevien kenttien arvoja, mikäli muutetulla kentällä on vaikutus pätevyysalueeseen, lasketaan pätevyysalueelle uudet arvot. Tämä tapahtuu käyttäen käyttämällä Angularin @Output-dekoraattoria (Kuva 46). Arvon muuttuessa komponentti lähettää signaalin komponentille, joka hallitsee pätevyysaluelaskentaa ilmoittaakseen, että pätevyysalueeseen vaikuttava arvo on

muuttunut. Pätevyyslaskentaa hallitseva komponentti sitten päivittää pätevyysalueen tilan käyttämällä pätevyyslaskentaa (Kuva 47).

```

<!-- joint type -->
<ks-select
  label="wp.form.iso.joint_type"
  class="align-with-chips"
  [dropdown]="dropdownOptions.jointType"
  [readOnly]="readOnly.readOnly"
  [emitChangesOnlyOnClose]="true"
  (valueChange)="essentialVariableChanged()"
  [(ngModel)]="state.data.details.joint.type"
></ks-select>

```

Kuva 46. Signaali arvon muuttuessa

```

243     essentialVariableChanged() {
244     |     this.roqCalculationService.updateState(this.wpqrState);
245     |     }

```

Kuva 47. Signaaliin reagointi

Lähetetty arvo on tyhjä, koska pätevyysalueen laskennan kannalta yksittäisen kentän arvo ei ole merkityksellinen, koska pätevyysalue perustuu kaikkien kenttien arvojen “summaan”. Eli käytännössä esimerkiksi hitsausasento voi vaikuttaa myös lämmöntuonnin laskentaan. Kun menetelmäkoe komponentti saa tiedon, että jollain kentällä on muuttunut laskennalle oleellinen arvo, se kutsuu pätevyyslaskenta-palvelun metodia pätevyysalueen uudelleen laskemiseksi (Kuva 48). Se antaa metodille argumentiksi uuden päivitetyn tilansa, sekä sen hetkisen dokumentille valitun standardin. Pätevyyslaskenta-palvelun metodissa tarkistetaan, onko sille rekisteröity dokumenttia, sekä tarkistetaan, onko valittu standardi vielä sama. Mikäli ei, se poistaa sen hetkisen rekisteröinnin ja kutsuu omaa rekisteröinti-metodiaan uudella tilalla ja standardilla. Jos standardi ei ole muuttunut ja dokumentin tilassa pätevyyslaskenta on kytketty päälle, pätevyyslaskenta-palvelu uudelleen laskee pätevyysalueet ja lähettää muutokset subjettiin, johon on menetelmäkoepöytäkirja komponentissa tilaus. Menetelmäkoepöytäkirja komponentti päivittää sitten dokumentin tilan laskennalta saaduilla arvoilla.

```

97   updateState(state: any, selectedStandard?: string) {
98     if (!this.isRegistered) {
99       return;
100    }
101
102    if (this.standard && selectedStandard && this.standard !== selectedStandard) {
103      this.unregister();
104      this.registerState(state, selectedStandard);
105      return;
106    }
107
108    if (state?.$?.useAutomaticRoqCalculation) {
109      this.autofillRangeOfQualification(state);
110    }
111  }
112
113  autofillRangeOfQualification(state: any) {
114    this.layoutFunctions.autofillPassesToRoQ(state);
115
116    if (this.standardHasCalculations) {
117      this.rangeOfQualification.next(
118        this.calculationStandard.calculateRangeOfQualification(
119          |   this.layoutFunctions.parseEssentialsFromState(state, this.standard as SupportedStandard)
120          |   )
121        );
122    }
123  }

```

Kuva 48. Laskenta palvelussa laskennan toteuttaminen

Koska pätevyysaluelaskennan tulisi tukea myös muitakin dokumentteja, kuin vain tässä työssä toteutettua menetelmäkoepöytäkirjaa, tulee jokaisella dokumentilla olla omat adapteri funktionsa (Kuva 49). Näiden funktioiden tehtävänä on parsia pätevyysaluelaskentaa varten tarvittavat arvot dokumentista, sekä päivittää pätevyysaluelaskennan laskemat tulokset dokumenttiin.

```

8   export interface LayoutFunctions {
9     parseEssentialsFromState(state: any, standard: SupportedStandard): EssentialVariables;
10    applyQualificationRangeToState(state: any, rangeOfQualification: RangeOfQualification): void;
11    autofillPassesToRoQ(state: any): void;
12  }

```

Kuva 49. Adapteri funktioiden määrittely

Adapteri funktioiden tulee parsia sen hetkisestä dokumentista laskennalle tarvittavat arvot tarvittavassa muodossa, kuten kuvasta 50 näkyy, sekä päivittää laskennalta saadut tulokset oikeille kentille dokumentissa (Kuvat 51 ja 52).

```

382 | public parseEssentialsFromState = (state: WpqrFormSchema, standard: SupportedStandard): EssentialVariables => {
383 |     const recordOfWeldTest = state?.$?.recordOfWeldTest;
384 |
385 |     const essentials: EssentialVariables = {
386 |         standard: null,
387 |         weldingProcesses: this.parseWeldingProcesses(recordOfWeldTest?.$?.weldingProcesses),
388 |         parentMaterials: this.parseParentMaterials(recordOfWeldTest?.data?.materials?.parents),
389 |         passes: this.parsePasses(state?.data?.passes),
390 |         holdingTemperature: recordOfWeldTest?.data?.details?.heatTreatment?.postWeldTemp,
391 |         preheatTemperature: recordOfWeldTest?.data?.details?.heatTreatment?.preheatTemp,
392 |         throatThickness: recordOfWeldTest?.data?.details?.throatThickness,
393 |         isHardnessTestRequired: state?.data?.isHardnessTestRequired,
394 |         isImpactTestRequired: state?.data?.isImpactTestRequired,
395 |         grooveAngle: Number(recordOfWeldTest?.data?.details?.grooveAngle?.max),
396 |         // ...
397 |     };
398 |
399 |     essentials.standard = standard;
400 |
401 |     return essentials;
402 | };

```

Kuva 50. Parsinta-funktion toteutus

```

419 | public applyQualificationRangeToState = (state: WpqrFormSchema, rangeOfQualification: RangeOfQualification): void => {
420 |     this.applySharedQualificationRangeToState(state, rangeOfQualification);
421 |
422 |     if (
423 |         rangeOfQualification.standard === 'ISO 15614-1:2017' ||
424 |         rangeOfQualification.standard === 'P-ACA-CAL-056' ||
425 |         rangeOfQualification.standard === 'NORSOK M-101'
426 |     ) {
427 |         this.applyISO15614QualificationRangeToState(state, rangeOfQualification);
428 |     }
429 |
430 |     if (rangeOfQualification.standard === 'DNV-RU-SHIP') {
431 |         this.applyDNVQualificationRangeToState(state, rangeOfQualification);
432 |     }
433 |
434 |     if (rangeOfQualification.standard === 'NR 216 DT R13') {
435 |         this.applyBureauVeritasNR216QualificationRangeToState(state, rangeOfQualification);
436 |     }
437 |
438 |     if (rangeOfQualification.standard === 'LR RULES 2021') {
439 |         this.applyLloydsRegisterQualificationRangeToState(state, rangeOfQualification);
440 |     }
441 |
442 | };

```

Kuva 51. Laskennan tulosten päivittäminen

```

326 | private applyBureauVeritasNR216QualificationRangeToState = (state: WpqrFormSchema, rangeOfQualification: BureauVeritasNR216QualificationRange) => {
327 |     state.data.details.heatTreatment.postHeatingTemp = rangeOfQualification.postWeldHeatTreatmentTemperatureRange;
328 |     state.data.materials.parents.A.deliveryCondition = this.getDropdownOptions(rangeOfQualification.materialDeliveryConditions, 'commonDeliveryCondition');
329 |     state.data.materials.parents.B.deliveryCondition = this.getDropdownOptions(rangeOfQualification.materialDeliveryConditions, 'commonDeliveryCondition');
330 |     state.data.details.weldDetails = this.getDropdownOptions(rangeOfQualification.weldDetails.systemCodes, 'dnvWeldDetails');
331 |     state.data.details.process = this.getDropdownOptions(rangeOfQualification.allowedProcesses, 'isoWeldingProcess');
332 |     state.data.details.joint.type = this.getDropdownOptions(rangeOfQualification.jointTypes, 'isoJointType');
333 |     state.data.details.position = this.getDropdownOptions(rangeOfQualification.weldingPositions, 'isoWeldingPosition');
334 |     state.data.materials.parents.isoMaterialGroups = this.getDropdownOptions(rangeOfQualification.steelGrades, 'bvSteelGrade');
335 | };
336 |
337 | private applyLloydsRegisterQualificationRangeToState = (state: WpqrFormSchema, rangeOfQualification: LloydsRegisterQualificationRange) => {
338 |     state.data.details.position = this.getDropdownOptions(rangeOfQualification.weldingPositions, 'isoWeldingPosition');
339 |     state.data.materials.parents.isoMaterialGroups = this.getDropdownOptions(rangeOfQualification.steelGrades, 'lrSteelGrade');
340 | };

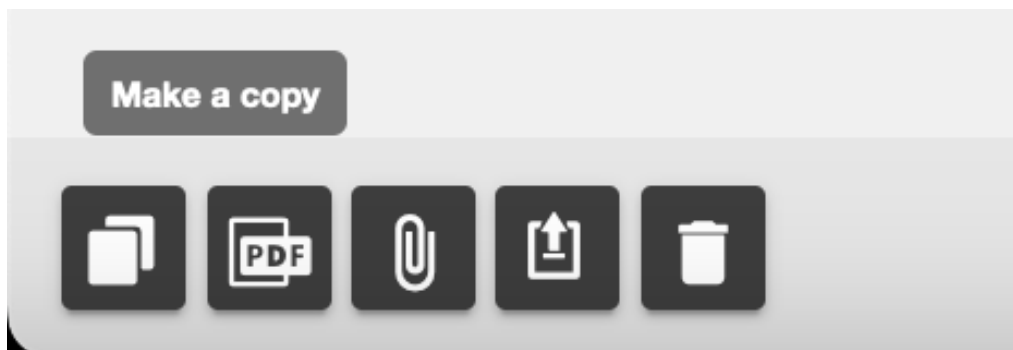
```

Kuva 52. Laskennan tulosten päivittäminen oikeille kentille

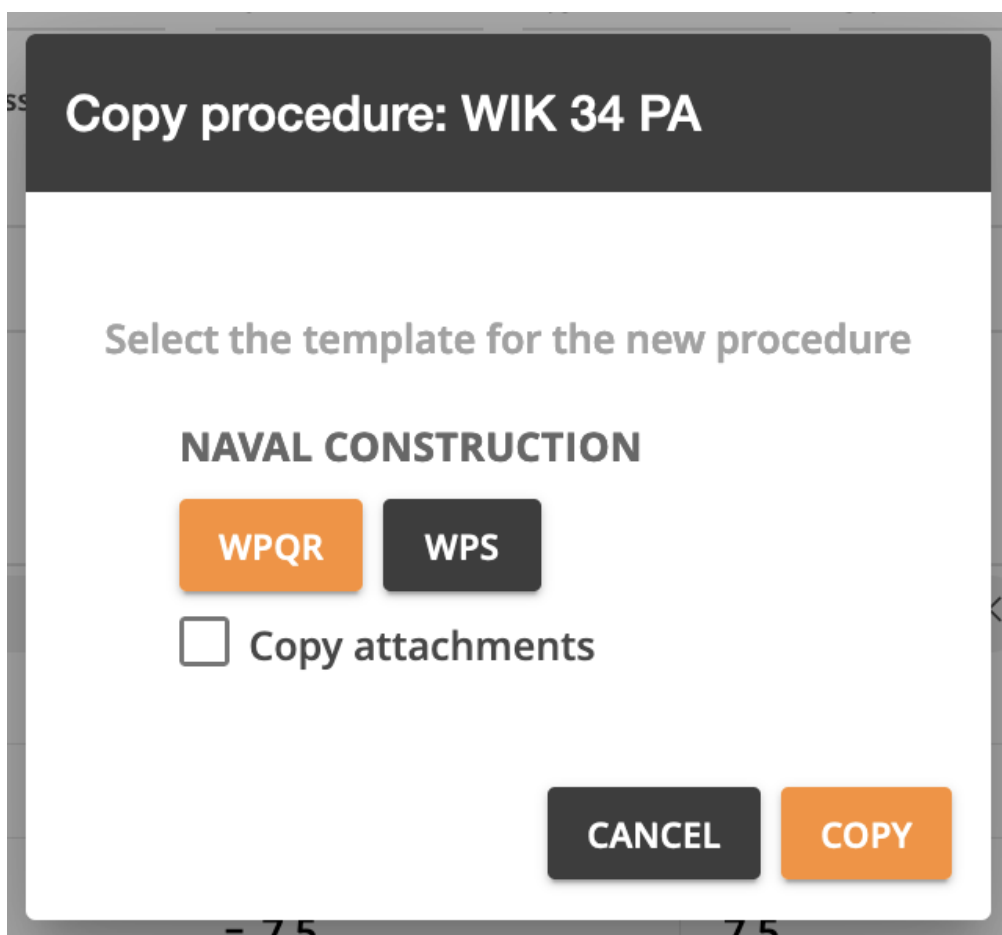
#### 4.4.3 Hitsausohjeiden luominen menetelmäkoepöytäkirjan pohjalta

Kun käyttäjä on täyttänyt menetelmäkoepöytäkirjan ja haluaa luoda sen pohjalta hitsausohjeen, löytyy templaattisivun alapalkista (Kuva 52) paikine, jota painamalla

avautuu dialogi (Kuva 53). Dialogissa käyttäjä voi valita minkä tyyppisen dokumentin hän haluaa luoda menetelmäkoepöytäkirjan pohjalta. Käyttäjä voi valita joko uuden menetelmäkoepöytäkirjan, tai hitsausohjeen.



Kuva 52. Kopiointi painike templaattisivulla



Kuva 53. Dokumentin kopiointi dialogi


Mikäli käyttäjä valitsee menetelmäkoepöytäkirjan, menetelmäkoepöytäkirjasta luodaan uusi kopio ennalta täytetyn menetelmäkoepöytäkirjan tiedoilla. Nimeä, tai muita tunnisteita ei kopioida uudelle menetelmäkoepöytäkirjalle. Jos käyttäjä valitsee dialogissa hitsausohjeen, luodaan uusi hitsausohje dokumentti. Hitsausohjeelle kopioituu

menetelmäkoepöytäkirjasta löytyvät pätevyysalueiden arvot oikeille kentille. Tämä tapahtuu käytännössä käyttämällä dokumenttia ladatessa adapteri funktiota (Kuva 54), joka tarkistaa minkä tyyppisesti dokumentista kopio ollaan luomassa ja kääntämällä dataa luodulle dokumentille sopivaan muotoon kopioidusta kohteesta.

```
11 export default function copyTargetFilter(sourceDoc: any) {
12   let targetDoc = jsonCloneDeep(sourceDoc);
13   targetDoc = clearUniqueIds(targetDoc);
14
15   if (sourceDoc.schema.type === 'WPQR') { // copying from WPQR
16     // Copy Record of Weld Test data
17     _set(targetDoc, 'data', _get(sourceDoc, 'data.recordOfWeldTest'));
18
19     // Use qualification range passes
20     _set(targetDoc, 'data.passes', _get(sourceDoc, 'data.passes'));
21
22     // Overwrite details that can have calculated qualification ranges
23     if (_get(sourceDoc, 'data.details.grooveAngle.min') != null || _get(sourceDoc, 'data.details.grooveAngle.max') != null) {
24       _set(targetDoc, 'data.details.grooveAngle', _get(sourceDoc, 'data.details.grooveAngle'));
25     }
26     if (_get(sourceDoc, 'data.details.rootGap.min') != null || _get(sourceDoc, 'data.details.rootGap.max') != null) {
27       _set(targetDoc, 'data.details.rootGap', _get(sourceDoc, 'data.details.rootGap'));
28     }
29   }
```

Kuva 54. Dokumentin kopiointi -funktio

Kuvassa 55 esimerkki täytetyn menetelmäkoepöytäkirjan pohjalta luodusta hitsausohjeesta. Kuvassa 55 näkyy, kuinka jotkut kentät ovat automaattisesti täyttyneet täytetyn menetelmäkoepöytäkirjan pohjalta.




Welding procedure name \*

Reference \*

Reference specification

Company name



WP-QWPvX5\_vY

### WELDING PROCEDURE SPECIFICATION (WPS)

Revision date  
07.04.2024

Revision \*  
0

Code/testing standard

Department

Referenced WPQRs  
**WPQR Demo** X

**JOINT DESIGN**

ADD
SKETCH LIBRARY
DRAW

**WELDING SEQUENCE**

ADD
SKETCH LIBRARY

**DETAILS**

| Joint type | Product type | Sides | Groove angle (°) |      | Root gap [mm] |      |
|------------|--------------|-------|------------------|------|---------------|------|
|            |              |       | Min              | Max  | Min           | Max  |
| BW         | P            | SS    | 60               | - 85 | 1             | - 16 |

Remarks

**BACKING**

| Backing | Backing material  | Dimensions |
|---------|-------------------|------------|
| mb      | Ceramic Backing X |            |

**JOINT PREPARATION**

| Initial or interpass cleaning | Shop primer | Jigs, fixing and pre-plucking | Tack welding length and pitch |
|-------------------------------|-------------|-------------------------------|-------------------------------|
| Brushing and grinding without |             |                               |                               |

**IDENTIFICATION OF PARENT MATERIAL**

**Parent materials**

| Name/Grade                           | Material group | Range      | Specification         |
|--------------------------------------|----------------|------------|-----------------------|
| EN 10025 S460 G2+M                   | 2.1            |            | ISO TR 15608:2006 2.1 |
| <b>Thickness [mm]</b>                |                |            |                       |
| Min                                  | Max            | Min        | Max                   |
| 7.5                                  | - 30           | 15         | - 15.0                |
| <b>Weld metal thickness [mm]</b>     |                |            |                       |
| Min                                  | Max            | Min        | Max                   |
| 7.5                                  | - 30           | 15         | - 15                  |
| <b>Diameter [mm]</b>                 |                |            |                       |
| Min                                  | Max            | Min        | Max                   |
| 0                                    | ≤ 0            | 0          | - 0                   |
| <b>Angle (°)</b>                     |                |            |                       |
| Product type                         |                | SMYS (MPa) | Manufacturer          |
| P                                    |                |            | 281150-02             |
| <b>Delivery condition</b>            |                |            |                       |
| <input type="checkbox"/> is abutting |                |            |                       |
| <b>Cast</b>                          |                |            |                       |
| 397932                               |                |            |                       |
| <b>Remarks</b>                       |                |            |                       |
|                                      |                |            |                       |

**PREHEAT**

| Preheat temperature [°C] |        | Interpass temperature [°C] |         | Temperature control | Method |
|--------------------------|--------|----------------------------|---------|---------------------|--------|
| Min                      | Max    | Min                        | Max     |                     |        |
| 20.0                     | - 20.0 |                            | < 250.0 | N/A                 | N/A    |

INTENTIONALLY INCREASE OF PREHEATING TEMPERATURE BEFORE CAPPING RUN

Kuva 55. Menetelmäkokeen pohjalta luotu hitsausohje

Lisäksi hitsausohjeen dataan liitetään mukaan linkitys menetelmäkoepöytäkirjaan, josta hitsausohje on kopioitu (Kuva 56). Tätä linkkiä painamalla päästään takaisin menetelmäkoepöytäkirja dokumenttiin, jonka pohjalta kyseinen hitsausohje on luotu.

**KEMPPi**

Welding procedure name \*

Reference \*

Reference specification

Company name

**WELDING PROCEDURE SPECIFICATION (WPS)**

WP-qWPlvx5\_tv

Revision date 07.04.2024

Revision \* 0

Code/testing standard

Department

Referenced WPQRs

WPQR Demo X

Kuva 56. Menetelmäkoepöytäkirjan linkitys

## 5 Yhteenveto

Opinnäytetyön tavoitteena oli automatisoida pätevyysalueiden laskenta ja täyttö Weldeye-palvelussa. Lisättyjen toimintojen tuli integroitua saumattomasti jo olemassa olevaan WeldEye-palveluun ja mukailla olemassa olevaa toiminnallisuutta. Toiminnallisuutta varten tarvittavat pätevyysalueiden laskennat toteutettiin luomalla erillinen npm-paketti, jota voidaan helposti ylläpitää ja lisätä tarvittaessa muihin sovelluksiin, ja jonka toiminnallisuus on varmistettu monipuolisella yksikkö testauksella. Menetelmäkoepöytäkirjojen hallinta ja siitä hitsausohjeiden luonti WeldEye-palvelussa mukailee toiminnallisuudeltaan jo olemassa olevaa hitsausohjeiden hallintaa WeldEye-palvelussa, ja on täten helppo ottaa käyttöön käyttäjien näkökulmasta. Menetelmäpöytäkirjan ulkoasu mukailee WeldEye-palvelun, ja täten Kemppi brändin asettamia vaatimuksia. Pätevyysalueen automaattinen laskenta, sekä menetelmäkoepöytäkirjan formaatti on toteutettu niin, että standardien asettamat vaatimukset menetelmäkoepöytäkirjalle toteutuvat.

Toteutuksen myötä käyttäjillä on nyt helppo tapa lisätä ja ylläpitää menetelmäkoepöytäkirjoja, joista tarvittaessa voi kopioida joko uusia menetelmäkoepöytäkirjoja, jotka täyttävät eri standardien asettamat vaatimukset, tai luoda hitsausohjeita, jotka pohjautuvat täytettyyn menetelmäkoepöytäkirjaan.

Menetelmäkoepöytäkirjaan täytettävän pätevyysalueen kenttien täyttö on myös mahdollista tehdä automatisoidusti ja standardien vaatimusten mukaisesti, jolloin käyttäjien ei tarvitse itse laskea ja kirjata arvoja uusille menetelmäkoepöytäkirjoille. Työ on täyttänyt asiakkaiden asettamat vaatimukset ja on ollut käytössä syksystä 2022 lähtien.

Toteutus on sellaisenaan valmis, eikä ominaisuus itsessään vaadi mitään ylläpitoa. Sen jatkokehittäminen on tehty helpoksi, koska laskennat ovat toteutettu erillisenä pakettina. Tämä mahdollistaa laskentojen päivittämisen, tai uusien standardien lisäämisen ilman suurempia muutoksia infrastruktuurissa tai käyttöliittymässä. Erillinen paketti myös mahdollistaa laskentojen käytön tarvittaessa muissa sovelluksissa, tai projekteissa.

## Lähteet

Angular a. Attribute directives. Viitattu 7.4.2024. Saatavissa

<https://angular.io/guide/attribute-directives>

Angular b. Built in directives. Viitattu 7.4.2024. Saatavissa [https://angular.io/guide/built-in-](https://angular.io/guide/built-in-directives)

[directives](https://angular.io/guide/built-in-directives)

Angular c. Introduction to Angular concepts. Viitattu 7.4.2024. Saatavissa

<https://angular.io/guide/architecture>

Angular d. Introduction to components and templates. Viitattu 7.4.2024. Saatavissa

<https://angular.io/guide/architecture-components>

Angular e. Introduction to services and dependence injection. Viitattu 7.4.2024.

Saatavissa <https://angular.io/guide/architecture-services>

Angular f. Structural directives. Viitattu 7.4.2024. Saatavissa

<https://angular.io/guide/structural-directives>

Chai JS a. Chai JS. Viitattu 7.4.2024. Saatavissa <https://www.chaijs.com/>

Chai JS b. Expect. Viitattu 7.4.2024. Saatavissa

<https://www.chaijs.com/guide/styles/#expect>

DNV AS July 2021. 2021. Rules for classification. DNV AS.

ECE Global. WPQR & WPS Approval Options and Relationships. Viitattu 7.4.2024.

Saatavissa <https://www.eceglobal.com/services/business-consulting/welding-manufacturing/pqr-wps-approval-flow/>

Helsingin yliopisto. Mikä on brändi, ja miksi se on tärkeä liiketoiminnalle. 2019. Viitattu

29.4.2024. Saatavissa <https://hyplus.helsinki.fi/mika-on-brandi-ja-miksi-se-on-tarkea-liiketoiminnalle/>

Kemppi Oy a. Tietoja Kempistä. Viitattu 7.4.2024. Saatavissa [https://www.kemppi.com/fi-](https://www.kemppi.com/fi-FI/yritys/kemppi/tietoa-yrityksesta/)

[FI/yritys/kemppi/tietoa-yrityksesta/](https://www.kemppi.com/fi-FI/yritys/kemppi/tietoa-yrityksesta/)

Kemppi Oy b. Online Brand Guide. Viitattu 7.4.2024. Saatavissa

<https://brand.kemppi.com/en-US/>

Mocha JS. Mocha JS. Viitattu 7.4.2024. Saatavissa <https://mochajs.org/>

Npm. About npm. Viitattu 7.4.2024 Saatavissa <https://docs.npmjs.com/about-npm>

SFS ry. Mikä on standardi. Viitattu 29.04.2024. Saatavissa

<https://sfs.fi/standardeista/mika-on-standardi/>

TypeScript a. TypeScript for JavaScript programmers. Viitattu 7.4.2024. Saatavissa

<https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>

TypeScript b. Tsconfig. Viitattu 7.4.2024. Saatavissa

<https://www.typescriptlang.org/tsconfig>