



# **Automatisert klienttesting**

## **Visjonsdokument**

### **Versjon 2.0**

*Dokumentet er basert på Visjonsdokument utarbeidet ved NTNU. Revisjon og tilpasninger til bruk ved IDER, DATA-INF utført av Carsten Gunnar Helgesen, Svein-Ivar Lillehaug og Per Christian Engdal. Dokumentet finnes også i engelsk utgave.*



## REVISJONSHISTORIE

Dato	Versjon	Beskrivelse	Forfatter
19.01.2023	1.0	Fylt ut førsteutkast	Henrik Engdal, Jan William Jensen
25.02.2023	2.0	Retting	Henrik Engdal, Jan William Jensen

## INNHOLDSFORTEGNELSE

<b>1 INNLEDNING</b>	<b>1</b>
<b>2 SAMMENDRAG PROBLEM OG PRODUKT</b>	<b>2</b>
2.1 PROBLEMSAMMENDRAG	2
2.2 PRODUKTSAMMENDRAG	2
<b>3 BESKRIVELSE AV INTERESSENER OG BRUKERE</b>	<b>3</b>
3.1 OPPSUMMERING INTERESSENER	3
3.2 OPPSUMMERING BRUKERE	3
3.3 BRUKERMILJØET	3
3.4 SAMMENDRAG AV BRUKERNES BEHOV	4
3.5 ALTERNATIVER TIL VÅRT PRODUKT	4
<b>4 PRODUKTOVERSIKT</b>	<b>5</b>
4.1 PRODUKTETS ROLLE I BRUKERMILJØET	5
4.2 FORUTSETNINGER OG AVHENGIGHETER	5
<b>5 PRODUKTETS FUNKSJONELLE EGENSKAPER</b>	<b>6</b>
<b>1. FUNKSJONELLE KRAV FOR DASHBORD</b>	<b>6</b>
<b>1.1. KJØRE TESTER</b>	<b>6</b>
<b>1.2. SE OVERSIKT AV TESTRESULTATER</b>	<b>6</b>
<b>1.3. REGISTRERE NYE TESTER</b>	<b>6</b>
<b>1.4 SE STATISTIKK PÅ TESTRESULTATER OVER TID</b>	<b>6</b>
<b>2. FUNKSJONELLE KRAV FOR API</b>	<b>6</b>
<b>6 IKKE-FUNKSJONELLE EGENSKAPER OG ANDRE KRAV</b>	<b>7</b>
<b>7 REFERANSER</b>	<b>8</b>

# 1 INNLEDNING

Dokumentet inneholder en oversikt av behov, produktet sin rolle og funksjonelle egenskaper. Prosjektet går ut på å automatisere testing i klientene til Uni Micro.

Prosjektets eier er Uni Micro AS, et selskap som ble startet i Modalen 1986 da Hans Jørgen Neset lagde et økonomisystem som skulle forenkle arbeidsdagen til små- og mellomstore bedrifter. I dag er Uni Micro en av de fremste økonomisystemene i Norge og har de siste årene blitt kjøpt opp av DNB og SpareBank 1.

Prosjektet går ut på å utvikle et API for uthenting og prosessering av tester, samt et grafisk brukergrensesnitt for kjøring av tester og henting av testresultater. Produktet vil bli benyttet internt hos oppdragsgiver.

## 2 SAMMENDRAG PROBLEM OG PRODUKT

### 2.1 Problemsammendrag

Problem med	dagens løsning er at det er mye tid og kostnader knyttet til å verifisere at klientene fungerer som de skal.
berører	alle utviklere i bedriften som blir nødt til å manuelt gjenskape bugs som er innrapportert i frontend, før de kan fikses.
som resultatet av dette	brukes det mye ressurser på å manuelt gjenskape og deretter fikse bugs som er innrapportert
en vellykket løsning vil	spare bedriften for tid og kostnader ved å effektivisere testingen.

### 2.2 Produktsammendrag

For	UniMicro
som	har behov for kontinuerlig testing av UI-funksjonaliteter
produktet navngitt	SoftRigTestTool
som	automatisk tester funksjonaliteter i ulike klienter
I motsetning til	dagens løsning hvor testere/utviklere må gjøre alt manuelt
Har vårt produkt	effektivisert test-prosessen

## 3 BESKRIVELSE AV INTERESSENER OG BRUKERE

### 3.1 Oppsummering interessenter

Navn	Utdypende beskrivelse	Rolle under utviklingen
UniMicro	Geir Bjellebø Kvam - "Head of DevOps"	Bidrar med aktiv tilbakemelding og godkjenning av løsninger.
Utviklere ansatt i UniMicro	Utviklere som ønsker å teste frontend funksjonalitet underveis i utviklingen	Teste programmet og gi tilbakemelding til forbedringer og hva som fungerer bra
Testere ansatt i UniMicro	Testere som ønsker å teste at all frontend funksjonalitet fungerer som det skal, før utrulling av nye versjoner	Teste programmet og gi tilbakemelding til forbedringer og hva som fungerer bra

### 3.2 Oppsummering brukere

Navn	Utdypende beskrivelse	Rolle under utviklingen	Representert av
Utviklere	Utviklere må sjekke at det de har utviklet fungerer som tiltenkt og kan da bruke dette systemet til å verifisere at dette stemmer	Teste programmet og gi tilbakemelding til forbedringer og hva som fungerer bra	Seg selv
Testere	I dag gjennomfører testere repetitive manuelle tester for å avdekke feil og verifisere at programmet fungerer som tiltenkt	Teste programmet og gi tilbakemelding til forbedringer og hva som fungerer bra	Seg selv

### 3.3 Brukermiljøet

Systemet er bygget opp av Angular og Typescript, med en back-end i .NET og skal i hovedsak benyttes av utviklere i UniMicro. Ettersom at systemet starter opp Uni

Economy klienten i en nettleser for UI testing, må det fungere på flere typer nettlesere. Her er det satt et minimumskrav for Google Chrome, Mozilla Firefox og Edge.

Systemet vil bli brukt når testere jobber med regresjonstester før en ny versjon. Da vil det bli gjennomført repetitive tester som systemet kan kjøre for testerne.

### 3.4 Sammendrag av brukernes behov

Behov	Prioritet	Påvirker	Dagens løsning	Foreslått løsning
Kjøre tester	1	Bedrift	Manuelt kjøre tester	Tester kjøres automatisk av maskinen
Skrive steg til tester	1	Bedrift	Manuelt av tester	Lage kode som kan kjøre stegene
legge inn testresultat	1	Bedrift	Manuelt av tester	Systemet lager resultat
Se teststatistikk over en tidsperiode	2	Bedrift	Ingen løsning	Systemet lager statistikk

### 3.5 Alternativer til vårt produkt

LambdaTest (Lambdatest, 2017), BrowserStack (BrowserStack, 2011), Katalon Platform (Katalon, 2015) og DevExpress's TestCafe Studio (DevExpress, 2019).

LambdaTest og BrowserStack er basert på Selenium (Selenium, 2004-a) som er et open source prosjekt som har "WebDrivers" (Selenium, 2004-b) som kan automatisere nettleseren til å gjennomføre tester. TestCafe er et eget bibliotek som ikke bruker WebDrivers for å automatisere. Det er mye lettere for en tester som ikke har erfaring med kode og lage tester gjennom TestCafe språket.

TestCafe Studio mangler funksjonalitet og integrasjon mot TestRail, som er nettsiden hvor oppdragsgiver har lagret oppskriften på alle tester som gjennomføres manuelt, og resultatene fra testene som blir gjennomført. Det er mulig å sende resultater til TestRail, men ikke å hente informasjon fra TestRail. I et lengre perspektiv er det også

ønskelig å kunne populere testene med data fra produksjonsmiljøer for å gjenskape feil som kunder opplever.

Både Selenium og TestCafe vil være mulig å benytte i APIet ettersom det ikke vet noe om kjøring av testene. Det sender kun ut kommandoer og tar imot resultater.



## 4 PRODUKTOVERSIKT

### 4.1 Produktets rolle i brukermiljøet

Testere og utviklere har et ansvar for at koden som ender opp i produksjon fungerer som det skal. Produktet vil fungere som en automatisert verifiseringsprosess med et dashboard for oversikt over testresultater og tilbakemeldinger. Testresultatene loggføres slik at de kan enkelt hentes frem i dashboardet og sammenlignes over tidsperioder.

### 4.2 Forutsetninger og avhengigheter

For å bruke systemet kreves det at brukeren har en datamaskin med internett tilgang. Når systemet publiseres vil det også kreves at en bruker har tilgang til Uni Micros intranett.

For å kjøre tester lokalt vil det være nødvendig å laste ned programmet som utvikles.

Prosjektet er avhengig av .Net, Angular og TestCafe for å kjøre tester.  
APIet er avhengig av .Net

Brukergrensesnittet er avhengig av Angular og nodeJS  
tester er avhengig av TestCafe

# **5 PRODUKTETS FUNKSJONELLE EGENSKAPER**

## **1. Funksjonelle krav for Dashbord**

- 1.1. Kjøre tester
- 1.2. Se oversikt av testresultater
- 1.3. Registrere nye tester
- 1.4 Se statistikk på testresultater over tid

## **2. Funksjonelle krav for API**

- 2.1. Starte en test
- 2.2. Starte en kategori med tester
- 2.3. Starte alle tester
- 2.4. Lage rapporter på testene
- 2.5. Starte tester i forskjellige nettlesere
- 2.6. Hente resultater fra tester
- 2.7. Se hvor lenge en test har kjørt

## **6 IKKE-FUNKSJONELLE EGENSKAPER OG ANDRE KRAV**

Teknologier vi skal bruke under utviklingen:

- C# .Net, Angular, Typescript, Nodejs, SQL Server

Hvis prosjektet kommer så langt at det vil være mulig å teste på kunders data vil det bli nødvendig å ta hensyn til personlig informasjon. All statistikk som lages må følge GDPR.

## 7 REFERANSER

1. LambdaTest. 2017. "LambdaTest." [Online] URL: <https://www.lambdatest.com/>. [Hentet: 03 Februar 2023].
2. BrowserStack. 2011. "BrowserStack." [Online] URL: <https://www.browserstack.com/>. [Hentet: 03 Februar 2023].
3. Katalon Platform. 2015. "Katalon Platform." [Online] URL: <https://katalon.com/katalon-platform/>. [Hentet: 03 Februar 2023].
4. DevExpress TestCafe Studio. 2019. "DevExpress TestCafe Studio." [Online] URL: <https://www.devexpress.com/products/testcafestudio/>. [Hentet: 03 Februar 2023].
5. Selenium. 2004-a. "Selenium." [Online] URL: <https://www.selenium.dev/>. [Hentet: 03 Februar 2023].
6. Selenium. 2004-b. "WebDriver." [Online] URL: <https://www.selenium.dev/documentation/webdriver/>. [Hentet: 03 Februar 2023].