



OSSQA

28 mars 2024

# Testplan

Edvin Gibro  
Bacilika Glansholm  
Viktor Holta  
Simon Karlsson  
Jessica Kjellin  
Max Randow  
Erik Simonson  
Jakob Söderström

28 mars 2024

Version 0.6



Status

Granskad	NAMN	2024-xx-xx
Godkänd	NAMN	2024-xx-xx



### Projektidentitet

Hemsida: <https://github.com/OSSQA-PUM>

Kund: Ola Angelsmark, Advenica AB  
E-post: ola.angelsmark@advenica.com

Handledare: Eric Ekström  
E-post: eric.ekstrom@liu.se

Kursansvarig: Kristian Sandahl  
E-post: kristian.sandahl@liu.se

### Projektdeltagare

Namn	Ansvar	E-post
Edvin Gibro	Teamledare	edvgi966@student.liu.se
Bacilika Glansholm	Arkitekt, Vice teamledare	bacgl188@student.liu.se
Viktor Holta	Testledare	vikho305@student.liu.se
Simon Karlsson	Analysansvarig	simka157@student.liu.se
Jessica Kjellin	Kvalitetssamordnare	jeskj559@student.liu.se
Max Randow	Dokumentansvarig	maxra518@student.liu.se
Erik Simonson	Konfigurationsansvarig	erisi409@student.liu.se
Jakob Söderström	Utvecklingsledare	jakso277@student.liu.se



## INNEHÅLL

Dokumenthistorik	IV
1 Introduktion	1
1.1 Syfte . . . . .	1
1.2 Definitioner . . . . .	1
2 Detaljer	2
2.1 Testobjekt och deras identifierare . . . . .	2
2.2 Testspårbarhet . . . . .	2
2.3 Funktionalitet som kommer testas . . . . .	2
2.4 Funktionalitet som inte kommer testas . . . . .	3
2.5 Tillvägagångssätt . . . . .	3
2.6 Kriterium för godkännande av testobjekt . . . . .	3
2.7 Testleveranser . . . . .	3
3 Testförvaltning	4
3.1 Planerade aktiviteter och uppgifter . . . . .	4
3.2 Inläring . . . . .	4
3.3 Risker . . . . .	4
4 Testfall	5
4.1 Enhetstest . . . . .	5
4.2 Integrationstest . . . . .	6
4.3 Systemtest . . . . .	7
Referenser	8



## DOKUMENTHISTORIK

Version	Datum	Utförda ändringar	Utförda av	Granskad
0.1	2024-02-19	Första utkast	VH	MR, JK
0.2	2024-02-22	Skrev fler testfall och flyttade dess sektion till slutet av dokumentet	VH	SK, EG
0.3	2024-03-01	Skrev tabell för testspårbarhet för alla testfall	VH	BG, SK
0.4	2024-03-05	Korrigerade testobjekt och funktionalitet som kommer testas så att de stämmer överens med arkitekturplan, kravspecifikation, och testfall	VH	SK, JK
0.5	2024-03-07	Uppdaterade planerade aktiviteter och uppgifter, som nu beskriver generering av testloggar och skapande av testrapporter mer utförligt	VH	EG, SK
0.6	2024-03-08	Utökning av definitioner, bokstavsordning av definitioner, tillägg av brödtext, tillägg av tabelltexter, och finskrivning	VH	EG



# 1 INTRODUKTION

Detta dokument omfattar information om testning av systemet OSSQA, vilket innefattar vilka typer av tester som ska utföras, hur de ska utföras, och vad de innebär. Dokumentet följer standarden IEEE Std 829-2008 [1], med anpassningar för att vara enlig med andra dokument och underlätta planeringen av tester.

## 1.1 Syfte

Syftet med detta dokument är att förse teammedlemmarna med den information som krävs för att utföra alla aktiviteter, som berör testning av systemet, på ett effektivt sätt.

## 1.2 Definitioner

- **API** - Förkortning av *application programming interface*. Ett gränssnitt för att möjliggöra interaktion mellan två separata datorprogram eller bibliotek.
- **Black-box-testning** - En typ av testning där den interna strukturen av koden inte känns till.
- **CLI** - Förkortning av *command-line interface*. Ett gränssnitt för att interagera med datorprogram via en kommandotolk.
- **CycloneDX** - En standard för hur en SBOM representeras.
- **Frontend** - Det visuella gränssnittet som användaren använder för att interagera med systemet.
- **Git** - Ett versionshanteringsverktyg som användas för att hantera ändringar och samarbete inom ett projekt.
- **GitHub** - En webbplattform för att lagra och hantera *repos*.
- **GitHub Actions** - Ett verktyg för att automatiskt exekvera kommandon då repos på GitHub uppdateras.
- **GUI** - Förkortning av *graphical user interface*. Ett gränssnitt för att interagera med datorprogram via grafiska element.
- **Repo** - Från engelskans *repository*. Ett Git-repo är en katalog eller lagringsenhet där alla filer, historik, och konfiguration för ett projekt sparas.
- **REST** - Förkortning av *representational state transfer*. En mängd arkitektoniska begränsningar som innefattar, bland annat, tillståndslös kommunikation mellan en server och en klient.
- **SBOM** - Förkortning av *Software Bill of Materials*. En innehållsdeklaration för en mjukvara som definierar samtliga programbibliotek som används.
- **SUS** - Förkortning av *System Usability Score*. En etablerad enkät för att mäta användbarhet av ett system.



## 2 DETALJER

I denna sektion beskrivs detaljer om vad som ska testas, hur det ska testas, samt vilka leveranser det resulterar i.

### 2.1 Testobjekt och deras identifierare

Följande tabell beskriver vilka mjukvarumoduler som ingår i systemet. För mer information om varje modul se arkitektplanen [2].

**Tabell 1:** Mjukvarumoduler

ID	Beskrivning
M1	<i>CLI</i>
M2	<i>GUI</i>
M3	<i>Frontend-gränssnitt</i>
M4	Analysator
M5	<i>REST-API</i>
M6	Databas

### 2.2 Testspårbarhet

Följande tabell beskriver kopplingen mellan krav och testfall, som beskrivs i sektion 4. För information om krav se kravspecifikationen [3].

**Tabell 2:** Kopplingar mellan testfall och krav

Testfalls-ID	Krav-ID
TF1	4
TF2	6
TF3	2
TF4	7
TF5	8
TF6	1, 3
TF7	1, 2, 3, 4, 6, 7, 9
TF8	5
TF9	5
TF10	1, 2, 3, 4, 6, 7, 9, 14

### 2.3 Funktionalitet som kommer testas

Följande funktioner ska testas:

- Mottagning av *SBOM* i *CycloneDX-format* via *CLI*.
- Mottagning av *SBOM* i *CycloneDX-format* via *GUI*.



- Analys av en SBOM utifrån kriteriernas prioriteter.
- Lagring av analysresultat i en databas.
- Hämtning av tidigare analysresultat.
- Automatisering av exekvering via *GitHub Actions*.

Följande egenskaper ska testas:

- Andel godkända testkörningar.
- Andel funktionstäckning.
- Antal poäng enligt *SUS* [4]
- Exekveringstid för analys av en SBOM.

Dessa funktioner och egenskaper utgår kraven i kravspecifikationen [3].

## 2.4 Funktionalitet som inte kommer testas

Alla ovanstående funktioner och egenskaper ska testas.

## 2.5 Tillvägagångssätt

Metoden som ska användas för att utföra testning på systemet är *black-box-testning*. Syftet med detta är att förenkla testerna i så stor grad som möjligt.

Indatan och den förväntade utdatan som ska tas hänsyn till vid användning av metoden beskrivs i testfallen i sektion 4.

## 2.6 Kriterium för godkännande av testobjekt

Tester anses godkända när deras resultat stämmer överens med eller överträffar deras förväntade resultat som angivet i motsvarande testfall i sektion 4.

## 2.7 Testleveranser

Följande dokument och filer ska framställas eller uppdateras ifrån testaktivitet:

- Testplan
- Testloggar
- Testrapport



## 3 TESTFÖRVALTNING

I denna sektion beskrivs detaljer om vilka aktiviteter som ska utföras, vad som krävs för att utföra dem, samt risker angående utveckling av systemet.

### 3.1 Planerade aktiviteter och uppgifter

Testning ska ske veckovis, vilket betyder att alla testloggar som genereras under en vecka ska kompileras till en veckolig testrapport, som beskriver aktiviteter, resultat, avvikelser från testplanen, och beslut som berör testobjekten.

Generation av testloggar ska ske automatiskt via GitHub Actions, som fyller dem med följande information:

- ID av berörda testfall
- Datum och tid
- Filnamn av testmodul
- Resultat
- Eventuellt felmeddelande

I slutet av veckan ska testloggarna ses över och kompileras till en testrapport. Samt ska testplanen uppdateras om det döms nödvändigt.

### 3.2 Inlärnin

Alla teammedlemmarna ska gå på en workshop för att lära sig hur olika tester skrivs, samt vart man kommer åt testresultat. Denna workshoppen är preliminär och kan komma att ändras.

### 3.3 Risker

För information om risker se projektplanen [5].





## 4 TESTFALL

I denna sektion beskrivs detaljer om systemets testfall, som ska användas som en guide när tester skrivs.

### 4.1 Enhetstest

**Tabell 3:** Testfall för Analys av ett enstaka beroende

<b>ID</b>	TF1
<b>Beskrivning</b>	Analys av ett enstaka beroende
<b>Indata</b>	Länk till Github-repo
<b>Förväntad utdata</b>	Bedömning utifrån OpenSSF Scorecards områden
<b>Beroenden</b>	Inga
<b>Mjukvaru-ID</b>	M4

**Tabell 4:** Testfall för Sammanställning av övergripande bedömning

<b>ID</b>	TF2
<b>Beskrivning</b>	Sammanställning av övergripande bedömning
<b>Indata</b>	Lista av beroendebestämningar
<b>Förväntad utdata</b>	Bedömning framställd från listan
<b>Beroenden</b>	Inga
<b>Mjukvaru-ID</b>	M4

**Tabell 5:** Testfall för Parsning av en SBOM

<b>ID</b>	TF3
<b>Beskrivning</b>	Parsning av en SBOM
<b>Indata</b>	Sökväg till SBOM
<b>Förväntad utdata</b>	Lista med beroendena som specificerades i SBOM:en
<b>Beroenden</b>	Inga
<b>Mjukvaru-ID</b>	M4

**Tabell 6:** Testfall för Skapande av bedömning

<b>ID</b>	TF4
<b>Beskrivning</b>	Skapande av bedömning
<b>Indata</b>	Repoversion och dess bedömning
<b>Förväntad utdata</b>	En bedömning skapas i databasen
<b>Beroenden</b>	Inga
<b>Mjukvaru-ID</b>	M6

**Tabell 7:** Testfall för Hämtning av bedömning

<b>ID</b>	TF5
<b>Beskrivning</b>	Hämtning av bedömning
<b>Indata</b>	ID av bedömning
<b>Förväntad utdata</b>	Den sparade bedömningen
<b>Beroenden</b>	TF4
<b>Mjukvaru-ID</b>	M6

**Tabell 8:** Testfall för Parsning av argument

<b>ID</b>	TF6
<b>Beskrivning</b>	Parsning av argument
<b>Indata</b>	En sträng med argument
<b>Förväntad utdata</b>	Ett objekt med argument av rätt datatyp
<b>Beroenden</b>	Inga
<b>Mjukvaru-ID</b>	M1

## 4.2 Integrationstest

**Tabell 9:** Testfall för Analys av en SBOM

<b>ID</b>	TF7
<b>Beskrivning</b>	Analys av en SBOM
<b>Indata</b>	Sökväg till SBOM
<b>Förväntad utdata</b>	Bedömning framställd från SBOM:ens beroenden och bedömning sparad i databasen
<b>Beroenden</b>	Inga

**Tabell 10:** Testfall för Hämtning av översikt av föregående bedömningar

<b>ID</b>	TF8
<b>Beskrivning</b>	Hämtning av översikt av föregående bedömningar
<b>Indata</b>	Ingen
<b>Förväntad utdata</b>	En lista av sparade bedömningar
<b>Beroenden</b>	TF7

**Tabell 11:** Testfall för Hämtning av specifik bedömning

<b>ID</b>	TF9
<b>Beskrivning</b>	Hämtning av specifik bedömning
<b>Indata</b>	ID av bedömning
<b>Förväntad utdata</b>	Den sparade bedömningen
<b>Beroenden</b>	TF7, TF8



### 4.3 Systemtest

**Tabell 12:** Testfall för Automatiserad analys via Github Actions

<b>ID</b>	TF10
<b>Beskrivning</b>	Automatiserad analys via Github Actions
<b>Indata</b>	Sökväg till SBOM
<b>Förväntad utdata</b>	Bedömning framställd från SBOM:ens beroenden
<b>Beroenden</b>	Inga



## REFERENSER

- [1] IEEE Computer Society, “Ieee standard for software and system test documentation,” <https://ieeexplore.ieee.org/document/4578383>, 2008.
- [2] PUM14, “Arkitektplan,” [https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96\\_Arkitektplan.pdf](https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96_Arkitektplan.pdf), 2024.
- [3] —, “Kravspecifikation,” [https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96\\_Kravspecifikation.pdf](https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96_Kravspecifikation.pdf), 2024.
- [4] J. R. Lewis, “The system usability scale: past, present, and future,” *International Journal of Human–Computer Interaction*, vol. 34, no. 7, pp. 577–590, 2018.
- [5] PUM14, “Projektplan,” [https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96\\_Projektplan.pdf](https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96_Projektplan.pdf), 2024.