



OSSQA

19 februari 2024

# Testplan

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

19 februari 2024

Version 0.1



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	1
2.1 Testobjekt och deras identifierare . . . . .	1
2.2 Funktionalitet som kommer testas . . . . .	2
2.3 Funktionalitet som inte kommer testas . . . . .	2
2.4 Tillvägagångssätt . . . . .	2
2.5 Objektens godkännandes/misslyckades kriterium . . . . .	3
2.6 Testleveranser . . . . .	3
3 Testförvaltning	3
3.1 Planerade aktiviteter och uppgifter . . . . .	3
3.2 Inläring . . . . .	3
3.3 Risker . . . . .	3
Referenser	4



## DOKUMENTHISTORIK

Version	Datum	Utförda ändringar	Utförda av	Granskad
0.1	2024-02-19	Första utkast	PUM14	Max Randow, Jessica Kjellin



# 1 INTRODUCTION

## 1.1 Syfte

Syftet med detta dokument är att förse den information som krävs för att planera och utföra alla testprocesser som behövs för testningen av systemet.

## 1.2 Definitioner

- **CLI** - Förkortning av *command-line interface*. Ett gränssnitt för att interagera med datorprogram via en kommandotolk.
- **GUI** - Förkortning av *graphical user interface*. Ett gränssnitt för att interagera med datorprogram via grafiska element.
- **API** - Förkortning av *application programming interface*. Ett gränssnitt för att möjliggöra interaktion mellan två separata datorprogram eller bibliotek.
- **Git** - Ett versionshanteringsverktyg som användas för att hantera ändringar och samarbete inom ett projekt.
- **Black-box-testning** - En typ av testning där den interna strukturen av koden inte känns till.

# 2 DETALJER

## 2.1 Testobjekt och deras identifierare

Följande hårdvara ingår i systemet:

ID	Beskrivning
H1	Lokal PC
H2	Serverdator

Följande mjukvarumoduler ingår i systemet:

ID	Beskrivning	Hårdvaru-ID
M1	CLI	H1
M2	GUI	H1
M3	Klient-API	H1
M4	Klient	H1
M5	Server-API	H1
M6	Databas	H2



### 2.1.1 Testfall

ID	Beskrivning	Indata	Förväntad utdata
TF1	Beräkna exekveringstid för en SBOM-fil med 200 beroenden	Sökväg till SBOM-fil	< 5 timmar

## 2.2 Funktionalitet som kommer testas

Följande funktioner och egenskaper ska testas:

- Exekveringstid för analys av enstaka SBOM-filer.
- Inställning av och analys utifrån kriteriernas prioriteter.
- Automatisering av exekvering i en git pipeline.
- Verifiering av gränssnittet mellan klienten och CLI samt GUI.
- Verifiering av gränssnittet mellan klienten och databasen.

## 2.3 Funktionalitet som inte kommer testas

Följande funktioner och egenskaper inte ska testas:

- Val av SBOM-fil, eftersom det är ett problem som kan lösas via paket.

## 2.4 Tillvägagångssätt

Följande metoder kommer att användas för att utföra testning på systemet.

### 2.4.1 Enhetstester

Exekveringstid för analys av enstaka SBOM-filer ska testas via *Black-box-testning*, där tiden för analys av flera SBOM-filer beräknas.

Inställning av och analys utifrån kriteriernas prioriteter ska testas via *Black-box-testning*, där en SBOM analyseras, vars resultat sedan jämförs med de inmatade prioriteterna.

### 2.4.2 Integrationstester

Verifiering av gränssnittet mellan klienten och CLI samt GUI ska testas via *Black-box-testning*, där användarinput simuleras och korrektheten av resultatet verifieras.

Verifiering av gränssnittet mellan klienten och databasen ska testas via *Black-box-testning*, där förfrågningar skickas till varje slutpunkt, vars responsstatus verifieras.



### 2.4.3 Systemtester

Automatisering av exekvering i en *Git-pipeline* ska testas via *Black-box-testning*, där en SBOM av systemet analyseras som en del av dess egna *Git-pipeline*.

## 2.5 Objektens godkännandes/misslyckades kriterium

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

## 2.6 Testleveranser

Följande dokument som ska framställas eller uppdateras ifrån testaktivitet:

- Testplan
- Testloggar
- Testrapport

# 3 TESTFÖRVALTNING

## 3.1 Planerade aktiviteter och uppgifter

Testningen för systemet ska främst ske automatiskt som del av en *Git-pipeline*.

I slutet av varje sprint ska testresultaten ses över och kompileras till en testrapport. Samt ska testplanen uppdateras om det döms nödvändigt.

## 3.2 Inläring

Alla teammedlemmarna kommer gå på en workshop för att lära sig hur testningen sker, var man kommer åt testresultat, och hur man lägger till, ändrar, eller tar bort tester.

## 3.3 Risker

För information om risker se projektplanen [1].



## REFERENSER

- [1] 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.