



OSSQA

6 mars 2024

Testplan

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

6 mars 2024

Version 0.4



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 Testspårbarhet	2
2.3 Funktionalitet som kommer testas	2
2.4 Funktionalitet som inte kommer testas	3
2.5 Tillvägagångssätt	3
2.6 Objektens godkännandes/misslyckades kriterium	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	4
4.1 Enhetstest	4
4.2 Integrationstest	5
4.3 Systemtest	6
Referenser	7



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



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.
- **Frontend** - Det visuella gränssnittet som användaren använder för att interagera med systemet.
- **REST** - Förkortning av *representational state transfer*. En mängd arkitektoniska begränsningar som innefattar, bland annat, tillståndslös kommunikation mellan en server och en klient.
- **SUS** - Förkortning av *System Usability Score*. En etablerad enkät för att mäta användbarhet av ett system.
- **SBOM** - Förkortning av *Software Bill of Materials*. En innehållsdeklaration för en mjukvara som definierar samtliga programbibliotek som används.
- **CycloneDX** - En standard för hur en SBOM representeras.

2 DETALJER

2.1 Testobjekt och deras identifierare

Följande mjukvarumoduler ingår i systemet:

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



För mer information om varje modul se arkitektplanen [1].

2.2 Testspårbarhet

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

För information om krav se kravspecifikationen [2].

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 i en git pipeline.

Följande egenskaper ska testas:

- Andel godkända testkörningar.
- Andel funktionstäckning.
- Antal poäng enligt *SUS* [3]
- Exekveringstid för analys av en *SBOM*.

Dessa funktioner och egenskaper utgår kraven i kravspecifikationen [2].



2.4 Funktionalitet som inte kommer testas

Alla ovanstående funktioner och egenskaper ska testas.

2.5 Tillvägagångssätt

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

2.5.1 Enhetstester

Exekveringstid för analys av enskilda 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.

2.5.2 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.

2.5.3 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*.

2.6 Objektens godkännandes/misslyckades kriterium

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

2.7 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 [4].

4 TESTFALL

4.1 Enhetstest

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

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



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

ID	TF4
Beskrivning	Skapande av bedömning
Indata	Reversion och dess bedömning
Förväntad utdata	En bedömning skapas i databasen
Beroenden	Inga
Mjukvaru-ID	M6

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

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

4.2 Integrationstest

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



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

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

4.3 Systemtest

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



REFERENSER

- [1] PUM14, “Arkitektplan,” https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96_Arkitektplan.pdf, 2024.
- [2] —, “Kravspecifikation,” https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96_Kravspecifikation.pdf, 2024.
- [3] 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.
- [4] PUM14, “Projektplan,” https://github.com/OSSQA-PUM/OSSQA/blob/main/documents/TDDD96_Projektplan.pdf, 2024.