

# Pflichtenheft *DIgScriptManager*

Version 0.1

# Table of Contents

1. Zusammenfassung .....	2
2. Aufgabenstellung und Zielsetzung .....	3
3. Produktnutzung .....	4
4. Interessensgruppen (Stakeholders) .....	5
5. Systemgrenze und Top-Level-Architektur .....	6
5.1. Kontextdiagramm .....	6
5.2. Top-Level-Architektur .....	6
6. Anwendungsfälle .....	7
6.1. Akteure .....	7
6.2. Überblick Anwendungsfalldiagramm .....	7
6.3. Anwendungsfallbeschreibungen .....	8
7. Funktionale Anforderungen .....	15
7.1. Muss-Kriterien .....	15
7.2. Kann-Kriterien .....	16
8. Nicht-Funktionale Anforderungen .....	17
8.1. Qualitätsziele .....	17
8.2. Konkrete Nicht-Funktionale Anforderungen .....	17
9. GUI Prototyp .....	18
10. Datenmodell .....	19
10.1. Überblick: Klassendiagramm .....	19
11. Akzeptanztestfälle .....	21
12. Glossar .....	26

Version	Status	Bearbeitungsdatum	Autoren(en)	Vermerk
0.1	Grober Entwurf, Überarbeitung	27.10.2018	David Stein	Bitte noch Mal drüber lesen und ggf. korrigieren/ergänzen.
0.2	Überarbeitung	27.10.2018	Jannusch Bigge	
0.3	Überarbeitung	5.11.2018	Nico Hahn	

# Chapter 1. Zusammenfassung

Dieses Dokument enthält die Spezifikation der Softwareanforderung des Projektes **DigSkriptManager**. Es beinhaltet einen Überblick für das zu entwickelnde Softwareprodukt und als Basis der Kommunikation zwischen den Projektbeteiligten des Projekts und dem Entwicklungsteam. Dieses Dokument bildet die Grundlage für einen Vertrag zwischen Auftragnehmer und Klient. Daher sollte es auf Konsistenz geprüft werden. Die Softwarespezifikation beschreibt, was das gewünschte System zu erfüllen hat und teilweise, wie das Entwicklungsteam vor hat, die Lösung zu implementieren.

Die Spezifikation sollte komplett, korrekt und konsistent sein. Da es zu Projektende genutzt wird, um zu prüfen, ob die geforderte Softwarelösung geliefert worden ist, sollte es verifizierbar sein. Während des gesamten Projekts wird das Pflichtenheft verwendet. Daher sollten die nötigen Schritte nachverfolgbar sein. Ferner sollten die Spezifikationen flexibel anpassbar und entwickelbar sein. Allerdings sollten sich diese Anpassungen, nachdem der Auftraggeber dem Inhalt zugestimmt hat, auf ein Minimum beschränken. Anpassungen sind nichtsdestotrotz zu erwarten und müssen dokumentiert werden.

# Chapter 2. Aufgabenstellung und Zielsetzung

Um die Skriptverwaltung von DIGSILENT zu verbessern, soll eine zentrale Speichermöglichkeit geschaffen werden. Wichtig ist hier insbesondere eine performante und intuitive Suche basierend auf den enthaltenen Metadaten. Die zu entwickelnde Applikation DIGScriptManager (DIGSILENT Script Manager) soll die Verwaltung von Skripten unterstützen, diese in menschenlesbarem Format speichern und eine performante Suche an Hand ausgewählter Metadaten gewährleisten. Die Datenabfrage soll über einen Webservice erfolgen, der mit beliebigen kompatiblen Clients kommunizieren kann. PowerFactory, eine Softwarelösung von DIGSILENT soll dabei nur ein spezieller Client sein. Der Webservice muss Funktionen zum Upload und Download, sowie zur Informationsabfrage bereitstellen. Der DIGScriptManager muss neben Webservice- und Datenverwaltungsfunktionen auch Analyse- und Auswertungsmechanismen bereitstellen. Die Applikation soll so aufgebaut werden, dass deren verwaltete Skripte später in ein Versionsverwaltungssystem (z.B. SVN oder GIT) eingebettet werden kann. Die Implementierung soll basierend auf dem .Net Framework (bzw. .Net Core) in der Sprache C# erfolgen.

# Chapter 3. Produktnutzung

Das Produkt wird firmenintern von DIgSILENT benutzt. Das Produkt wird auf einem Server laufen. Man soll auf das Produkt über eine REST Anfrage oder über einen Browser zugreifen können. Der Zugriff soll dabei nur über das Intranet von DIgSILENT möglich sein. Die zu verwaltenden Skripte werden persistent gespeichert und für die Mitarbeiter von DIgSILENT verfügbar sein. Das Produkt sollte keine technische Wartung benötigen.

# Chapter 4. Interessensgruppen (Stakeholders)

Die Anforderungen an das Projekt werden vom Kunden DIgSILENT (vertreten durch Vinzent Krauße) vorgegeben. Ferner hat der Lehrstuhl für Softwaretechnologie (vertreten durch Peter Heisig) Einfluss auf die Anforderungen des Softwareentwicklungsprozesses.

Name	Priority (1..5)	Description	Goals
DIgSILENT	5	Der Auftraggeber für das Projekt.	<ul style="list-style-type: none"><li>• zentrale Skriptverwaltung für Skripte in der firmeneigenen Sprache DPL</li></ul>
Mitarbeiter von DIgSILENT	3	Nutzer, die Skripte hoch- bzw. runterladen oder nach Skripten suchen	<ul style="list-style-type: none"><li>• einfache Möglichkeit des Zugriffs auf bereits vorhandene Skripte (Download und Suchanfragen)</li><li>• Teilen von Skripten mit anderen Mitarbeitern (Upload)</li></ul>
Entwickler (Studenten der TU Dresden)	1	Studenten, die das Produkt im Rahmen des Softwaretechnologiepraktikums entwickeln	<ul style="list-style-type: none"><li>• erweiterbare Anwendung</li><li>• Praxis in der Softwareentwicklung</li></ul>

# Chapter 5. Systemgrenze und Top-Level-Architektur

## 5.1. Kontextdiagramm

[Zielarchitektur der Anwendung] | `./images/Zielarchitektur.png`

*Figure 1. Zielarchitektur*

Der rote Rahmen stellt den Kern der Anwendung dar. Es handelt sich um eine Serveranwendung, die über HTTP-Anfragen mit dem Client (links) oder einem Browser (unten) kommuniziert. Dabei nutzt die Anwendung den Server ebenfalls, um Skriptdaten zu speichern.

## 5.2. Top-Level-Architektur

[Top-Level-Architektur] | `./models/analysis/Top-Level-Architecture.png`

*Figure 2. Top-Level-Architektur*



# Chapter 6. Anwendungsfälle

## 6.1. Akteure

Name	Beschreibung
Mitarbeiter (DIgSILENT)	Mitarbeiter der Firma DIgSILENT, die Skripte nutzen wollen und mit dem Produkt interagieren

## 6.2. Überblick Anwendungsfalldiagramm

[Anwendungsfälle der Anwendung] | `.\models\analysis\UseCaseDiagram1.jpg`

*Figure 3. Anwendungsfalldiagramm*

Die Funktionalität des Programms beläuft sich grundlegend auf den Upload und Download von Skripten sowie eine gezielte Such- und Filterfunktion, um Skripte nach spezifischen Kriterien zu suchen. Dabei werden zahlreiche Prozeduren durchlaufen, unter anderem das Prüfen von Abhängigkeiten, der Integrität eines Skripts oder der Skriptversion. (s. Diagramm) Ebenso sind im Diagramm mögliche Suchkriterien dargestellt.

## 6.3. Anwendungsfallbeschreibungen

[Sequenzdiagramm] | `./models/analysis/Sequence.jpg`

Figure 4. Sequenzdiagramme der Anwendung

<b>ID</b>	<b>[UC0010]</b>
<b>Name</b>	Upload eines neuen Skripts
<b>Beschreibung</b>	Ein Nutzer soll ein Skript zum Webservice hochladen können.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Upload</i> : Nutzer möchte ein Skript hochladen. Er stellt dabei die XML-Beschreibung des Skriptes bereit.
<b>Vorbedingung(en)</b>	<i>Upload</i> : Skript liegt noch nicht auf dem Webserver bereit.
<b>Essentielle Schritte</b>	<i>Upload</i> :  1. Bereitstellen der XML-Beschreibung des Skripts 2. Nutzer sendet Skript an die Webservice-Schnittstelle

<b>ID</b>	<b>[UC0020]</b>
<b>Name</b>	Download eines neuen Skripts
<b>Beschreibung</b>	Ein Nutzer soll ein Skript vom Webservice runterladen können.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Download</i> : Nutzer möchte ein Skript runterladen. Er stellt dabei den Skriptnamen bereit. Der Webservice sendet die XML-Beschreibung des Skripts zurück an den Aufrufer.
<b>Vorbedingung(en)</b>	<i>Download</i> : Skript liegt noch nicht beim Nutzer bereit.
<b>Essentielle Schritte</b>	<i>Download</i> :  1. Bereitstellen des Skriptnamens 2. Nutzer sendet Anfrage zum Download mit dem Skriptnamen an den Webservice 3. Webservice sendet XML-Beschreibung des Skripts an den Aufrufer.

<b>ID</b>	<b>[UC0030]</b>
<b>Name</b>	Suchanfrage für Skript anhand von Metadaten
<b>Beschreibung</b>	Ein Nutzer soll in den auf dem Webserver liegenden Skripten nach bestimmten Skripten suchen können.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query</i> : Nutzer möchte Skripte suchen, welche bestimmte Metadaten haben.

<b>Vorbedingung(en)</b>	<i>Query: ...</i>
<b>Essentielle Schritte</b>	<i>Query:</i> <ol style="list-style-type: none"> <li>1. Bereitstellen von Metadaten für die Suchanfrage</li> <li>2. Benutzer gibt diese in einer Eingabemaske auf einer Webseite (aus dem Intranet zugänglich) ein</li> <li>3. Nutzer drückt "Suchen"</li> <li>4. Webservice sendet Liste von allen Skripten zurück, die die angegebenen Metadaten haben</li> <li>5. Anzeige der Skripte in einer sinnvollen Darstellung auf der Webseite</li> </ol>

<b>ID</b>	<b>[UC0031]</b>
<b>Name</b>	Suchanfrage für Skript anhand des Namens des Skriptes
<b>Beschreibung</b>	Ein Nutzer soll in den auf dem Webserver liegenden Skripte nach Skripten mit bestimmten Namen suchen können
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte Skripte suchen, welche bestimmten Namen haben.
<b>Vorbedingung(en)</b>	<i>Query: ...</i>
<b>Essentielle Schritte</b>	<i>Query:</i> <ol style="list-style-type: none"> <li>1. Bereitstellen des Namens für die Suchanfrage</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für die Suche nach Skripten mit bestimmten Namen zuständig ist. (getScriptsByName)</li> <li>3. Webservice sendet Liste von Metadaten von Skripten mit dem gesuchten Namen.</li> </ol>

<b>ID</b>	<b>[UC0032]</b>
<b>Name</b>	Suchanfrage für Skript anhand eines Tags des Skriptes
<b>Beschreibung</b>	Ein Nutzer soll in den auf dem Webserver liegenden Skripte nach Skripten mit bestimmten Tag suchen können
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte Skripte suchen, welche bestimmten Tag haben.
<b>Vorbedingung(en)</b>	<i>Query: ...</i>

<b>Essentielle Schritte</b>	<p><i>Query:</i></p> <ol style="list-style-type: none"> <li>1. Bereitstellen des Tags für die Suchanfrage</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für die Suche nach Skripten mit bestimmten Tag zuständig ist. (getScriptsByTag)</li> <li>3. Webservice sendet Liste von Metadaten von Skripten mit dem gesuchten Tag.</li> </ol>
-----------------------------	---

<b>ID</b>	<b>[UC0033]</b>
<b>Name</b>	Suchanfrage für Skript anhand der Category des Skriptes
<b>Beschreibung</b>	Ein Nutzer soll in den auf dem Webserver liegenden Skripte nach Skripten mit bestimmter Category suchen können
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte Skripte suchen, welche bestimmte Category haben.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<p><i>Query:</i></p> <ol style="list-style-type: none"> <li>1. Bereitstellen der Category für die Suchanfrage</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für die Suche nach Skripten mit bestimmter Category zuständig ist. (getScriptsByCategory)</li> <li>3. Webservice sendet Liste von Metadaten von Skripten mit der gesuchten Category.</li> </ol>

<b>ID</b>	<b>[UC0034]</b>
<b>Name</b>	Suchanfrage für Skript anhand des Autors des Skriptes
<b>Beschreibung</b>	Ein Nutzer soll in den auf dem Webserver liegenden Skripte nach Skripten mit bestimmtem Autor suchen können
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte Skripte suchen, welche bestimmten Autor haben.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<p><i>Query:</i></p> <ol style="list-style-type: none"> <li>1. Bereitstellen des Autors für die Suchanfrage</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für die Suche nach Skripten mit bestimmtem Autor zuständig ist. (getScriptsByAuthor)</li> <li>3. Webservice sendet Liste von Metadaten von Skripten mit dem gesuchten Autor.</li> </ol>

<b>ID</b>	<b>[UC0035]</b>
<b>Name</b>	Suchanfrage für Skript anhand der Firma (company) des Skriptes
<b>Beschreibung</b>	Ein Nutzer soll in den auf dem Webserver liegenden Skripten nach Skripten mit bestimmter Firma (company) suchen können
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte Skripte suchen, welche von einer bestimmten Firma (company) erstellt worden sind.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<i>Query:</i> <ol style="list-style-type: none"> <li>1. Bereitstellen der Firma (company) für die Suchanfrage</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für die Suche nach Skripten mit bestimmter Firma (company) zuständig ist. (getScriptsByCompany)</li> <li>3. Webservice sendet Liste von Metadaten von Skripten mit der gesuchten Firma (company).</li> </ol>

<b>ID</b>	<b>[UC0036]</b>
<b>Name</b>	Suchanfrage für Skript anhand einer Freitextsuche in der Beschreibung (description).
<b>Beschreibung</b>	Ein Nutzer soll in den auf dem Webserver liegenden Skripten nach Skripten mit bestimmten String in der Beschreibung suchen können.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte Skripte suchen, welche einen bestimmten String in der Beschreibung (description) enthalten.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<i>Query:</i> <ol style="list-style-type: none"> <li>1. Bereitstellen des Suchstrings für die Suchanfrage</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für die Freitextsuche zuständig ist. (getScriptsByDescription)</li> <li>3. Webservice sendet Liste von Metadaten von Skripten mit bestimmten String in der Beschreibung (description).</li> </ol>

<b>ID</b>	<b>[UC0040]</b>
<b>Name</b>	Abhängigkeiten abrufen
<b>Beschreibung</b>	Der Webservice soll eine Funktion bereitstellen, die für den UniqueName eines Skriptes die Metadaten von referenzierten Skripten zurückgibt.
<b>Akteure</b>	Nutzer

<b>Trigger</b>	<i>Query:</i> Nutzer möchte die Metadaten aller von einem Skript referenzierten Skripte erhalten.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<i>Query:</i> <ol style="list-style-type: none"> <li>1. Bereitstellen des UniqueNames des Skripts, dessen Abhängigkeiten geprüft werden soll.</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für das Prüfen von Abhängigkeiten von Skripten zuständig ist.</li> <li>3. Webservice sendet Liste von Metadaten von Skripten zurück, welche vom Skript mit gegebenem UniqueName referenziert werden.</li> </ol>

<b>ID</b>	<a href="#">[UC0041]</a>
<b>Name</b>	Abhängigkeiten prüfen
<b>Beschreibung</b>	Der Webservice soll eine Funktion bereitstellen, die für den UniqueName und einer Liste von UniqueNames von installierten Skripten prüft, welche Abhängigkeiten neu installiert werden müssen bzw. eine neue notwendige Version notwendig ist.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte die Metadaten aller Skripte, die er noch installieren muss, um ein Skript zu installieren.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<i>Query:</i> <ol style="list-style-type: none"> <li>1. Bereitstellen des UniqueNames des Skripts sowie einer Liste von UniqueNames von installierten Skripten.</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für das Prüfen von Abhängigkeiten von Skripten zuständig ist.</li> <li>3. Webservice sendet Liste von Metadaten von abhängigen Skripten zurück, welche installiert werden müssen.</li> </ol>

<b>ID</b>	<a href="#">[UC0042]</a>
<b>Name</b>	Skriptintegrität prüfen
<b>Beschreibung</b>	Der Webservice soll eine Funktion bereitstellen, die für den UniqueName eines Skriptes prüft, ob alle abhängigen Skripte auf dem Webservice vorhanden sind.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte die Integrität eines Skriptes prüfen.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...

<b>Essentielle Schritte</b>	<p><i>Query:</i></p> <ol style="list-style-type: none"> <li>1. Bereitstellen des UniqueNames des Skripts</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für das Prüfen der Skriptintegrität zuständig ist.</li> <li>3. Webservice sendet Liste von ungültigen Referenzen zurück bzw. eine leere Liste, falls das Skript integer ist.</li> </ol>
-----------------------------	---

<b>ID</b>	<b>[UC0043]</b>
<b>Name</b>	Inverse Abhängigkeit prüfen
<b>Beschreibung</b>	Der Webservice soll eine Funktion bereitstellen, die für den UniqueName eines Skriptes die Metadaten aller direkt oder indirekt abhängigen Skripte zurückgibt.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte inverse Abhängigkeiten eines Skriptes erhalten.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<p><i>Query:</i></p> <ol style="list-style-type: none"> <li>1. Bereitstellen des UniqueNames des Skripts</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für das Ermitteln der inversen Abhängigkeiten zuständig ist.</li> <li>3. Webservice sendet Liste von Metadaten von Skripten zurück, die das Skript mit dem UniqueName direkt oder indirekt referenzieren.</li> </ol>

<b>ID</b>	<b>[UC0050]</b>
<b>Name</b>	Neueste Version abrufen
<b>Beschreibung</b>	Der Webservice soll eine Funktion bereitstellen, die für den FullName eines Skriptes die Metadaten des Skriptes mit der höchsten Versionsnummer für diesen FullName zurückgeben.
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte die neueste Version eines Skriptes erhalten.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...

<b>Essentielle Schritte</b>	<p><i>Query:</i></p> <ol style="list-style-type: none"> <li>1. Bereitstellen des FullName eines Skriptes</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für das ermitteln der neuesten Version zuständig ist auf</li> <li>3. Webservice sendet Metadaten des Skriptes zurück, das diesen FullName hat.</li> </ol>
-----------------------------	---

<b>ID</b>	<b>[UC0051]</b>
<b>Name</b>	Versionen abrufen
<b>Beschreibung</b>	Der Webservice soll eine Funktion bereitstellen, die für den FullName eines Skriptes die Metadaten aller Skripte zurückgibt, die diesen FullName haben. (d.h. alle Versionen des Skriptes)
<b>Akteure</b>	Nutzer
<b>Trigger</b>	<i>Query:</i> Nutzer möchte alle Versionen eines Skriptes mit bestimmten FullName abrufen.
<b>Vorbedingung(en)</b>	<i>Query:</i> ...
<b>Essentielle Schritte</b>	<p><i>Query:</i></p> <ol style="list-style-type: none"> <li>1. Bereitstellen des FullName eines Skriptes</li> <li>2. Nutzer ruft Funktion des Webservice auf, die für das Ermitteln aller Versionen zuständig ist.</li> <li>3. Webservice sendet Metadaten aller Skripte zurück, die den FullName haben.</li> </ol>



# Chapter 7. Funktionale Anforderungen

ID	Version	Name	Description
[F0010]	v0.1	Upload	Das System muss in der Lage sein, Skripte auf den Webservice zu laden. Der Nutzer gibt dabei die XML-Beschreibung des Skripts vor.
[F0011]	v0.1	Upload - Existenzcheck	Das System muss in der Lage sein, bereits vorhandene Skripte trotz versuchter Uploads, nicht zu überschreiben. Der aufrufende Client wird darüber per Fehlermeldung informiert.
[F0020]	v0.1	Download	Das System muss in der Lage sein, Skripte zum Download bereitzustellen. Der Nutzer gibt dabei den konkreten Skriptnamen vor. Ist es nicht vorhanden, wird eine Fehlermeldung zurückgegeben.
[F0030]	v0.1	Suche	Das System muss in der Lage sein, in den verwalteten Skripten zu suchen. Das heißt: Der Nutzer gibt Metadaten der zu suchenden Skripte vor. Der Nutzer bekommt eine Liste aller Skripte zurück, die diese Metadaten aufweisen.

## 7.1. Muss-Kriterien

- Es sind keine kommerziellen Tools, Bibliotheken etc. zur Benutzung nötig.
- Die Nutzung ist Intranet-tauglich. Verbindungen ins Internet dürfen nicht nötig sein.
- Es gibt eine Konfigurationsdatei für alle notwendigen Parameter.
- Funktionalität wird über automatisierte Integrationstests sichergestellt.
- Die Softwarelösung ist skalierbar bzgl. der Anzahl der Skripte (inkl. Metadaten).
- Die Skripte (inkl. Metadaten) werden persistent gespeichert.
- Die Skripte können z.B. beim Upload einfach validiert werden. (z.B. der Objektreferenzen)
- Definition des Skriptinterfaces (z.B. XML). Skripte umfassen Metadaten sowie die eigentlichen Nutzdaten.
- Upload eines Skriptes: Der Upload von einem Client umfasst die gesamte Skriptdefinition, d.h. alle Metadaten und internen Abhängigkeiten
- Download eines Skriptes: Mittels Skriptnamen wird die gesamte Skript-Definition an den anfragenden Client zurückgegeben.
- Definition des Query-Interfaces (z.B. XML) ähnlich zum Skript-Interface, nur ohne Nutzdaten
- Skript-Suche: Der Client schickt eine Suchanfrage zum Server. Dieser schickt als Antwort eine Liste aller möglicher Skripte.

- einfache Installation des Skript-Managers

## 7.2. Kann-Kriterien

- Sicherung der Skripte
- Objektreferenzen mit Versionsbeschränkungen
- Der Server enthält eine Webseite zur Suche nach Skripten. Die Skripte können dort eingesehen werden.
- Erweiterbarkeit des Query-Interfaces
- Intelligente Filter für das Query-Interface, welche logisch verknüpft werden können.
- Versionierung der Skripte
- Zusammenstellen von Listen für externe Objektreferenzen

# Chapter 8. Nicht-Funktionale Anforderungen

## 8.1. Qualitätsziele

Die folgende Tabelle zeigt, welche Qualitätsansprüche mit welcher Priorität erreicht werden müssen.

1 = nicht wichtig .. 5 = sehr wichtig

Qualitätsanspruch	1	2	3	4	5
Wartbarkeit			x		
Erweiterbarkeit			x		
Benutzerfreundlichkeit	x				
Skalierbarkeit					x
Verlässlichkeit					x
Performanz				x	

Sicherheit wird über das Intranet des Kunden geschaffen und muss in diesem Projekt nicht beachtet werden. Benutzerfreundlichkeit hat eine niedrige Priorität, da lediglich ein Webservice implementiert wird, der Anfragen über HTTP verarbeitet.

## 8.2. Konkrete Nicht-Funktionale Anforderungen

ID	Name	Description
<a href="#">[NF0010]</a>	Performanz - Programmiersprache	Das System (vor allem die Suche nach Skripten) soll performant sein. Dazu wird die Sprache C# als kompilierte Sprache benutzt.
<a href="#">[NF0020]</a>	Skalierbarkeit - Datenstrukturen	Das System soll skalierbar bzgl. der Anzahl der Skripte sein.
<a href="#">[NF0030]</a>	Erweiterbarkeit - Klassenvererbung	Das System soll erweiterbar gestaltet sein, insbesondere bzgl. möglicher anderer Abhängigkeiten der Skripte untereinander wie auch bzgl. der verwendeten Skripttypen.

# Chapter 9. GUI Prototyp

In diesem Abschnitt wird der GUI-Prototyp der Webseite vorgestellt.

[Hauptseite der Webseite] | `./models|analysis|mainpage.png`

*Figure 5. Hauptseite der Webseite*

Die Hauptseite der Browseranwendung Besteht aus 3 Teilen: Die obere Navigationsleiste, einen Suchbutton enthaltend; ein Panel auf der linken Seite, welches die hierarchie der auf dem Server gespeicherten Skripte und sonstigen Objekte enthält; und ein Panel auf der rechten Seite, das nach Auswahl eines Objektes im Panel links Informationen über das entsprechende Objekt anzeigt. Der Klick auf den Suchbutton leitet zur Suchmaske weiter.

[Suchmaske der Webseite] | `./models|analysis|searchpage.png`

*Figure 6. Suchmaske der Webseite*

Die Suchmaske ist ein Eingabeformular bestehend aus Textboxen für jedes spezifische Suchkriterium. Hierbei ist der Name des Kriteriums im "Hint" (Text, der angezeigt wird, wenn die Textbox eigentlich leer ist) der Textbox angegeben. Für die Suche wird jede nichtleere Textbox berücksichtigt. Ein Klick auf den "Suchen" Button startet die Suchanfrage. Die Ergebnisse werden danach auf einer neuen Seite in Form einer Tabelle präsentiert.

[Interaktion auf der Webseite] | `./models|analysis|gui1.jpeg`

*Figure 7. Interaktion auf der Webseite mit Dialogbeschreibung*

# Chapter 10. Datenmodell

## 10.1. Überblick: Klassendiagramm

[Analyseklassendiagramm] | `./models|analysis|analysisclassdiagramm_v2.jpg`

Figure 8. Analyseklassendiagramm

Die Klasse **Webserver** besitzt beliebig viele Objekte vom Type **Skript**. Sie kann über Http ein neues Skript zugesendet bekommen und dieses dann speichern, ebenso kann sie ein Skript über http versenden. Sie kann auch ein bestimmtes Skript anhand von verschiedenen Kriterien suchen. Die Klasse **Skript** enthält den Inhalt(content), eine lange Beschreibung(longDiscription), einen eindeutigen Namen(uniqueName), welcher aus Name, Version und Kategorie besteht, so wie den vollen Namen(fullName), welcher aus Name und Kategorie besteht. Die Klasse **Skript** besitzt genau ein Objekt vom Typ **MetaDaten**. **MetaDaten** enthält den Typ des Skriptes(scriptType), den Namen(name), die Kategorie(category), die Version(version), den Author(author), die Firma die das Skript erstellt hat(company), eine kurze Beschreibung(shortDescription), den Zeitstempel der letzten Änderung(modificationDate), die Sichtbarkeit welche genutzt wird um innere Skripte zu verwalten(isPublic), sowie beliebige Tags zur Identifikation des Skriptes(tags). Die Klasse verfügt außerdem über eine Funktion die alle Meta Daten in ein XML Dokument umwandelt. Jedes Skript kann beliebig viele Objekte vom Type **InputParameter** besitzen. In **InputParameter** wird der Datentype des Inputparameters gespeichert(dataType), der Name(name), die Einheit(unit), eine Beschreibung(description) sowie den eigentlichen Wert(value). Ebenso besitzt **Skript** ein Objekt vom Type **OutputParameter** welches wieder den Datentype(dataType), den Namen(name), die Einheit(unit) und eine Beschreibung(description) enthält. Jedes Skript kann Abhängigkeiten aufweisen welche in einer eigenen Klasse **Dependencies** verwaltet werden. **Dependencies** besitzt das Ziel für die Abhängigkeit(target). Die Klassen **Objectreference**, **InnerScript** und **InnerObject** erben von der Klasse **Dependencies**. **Objectreference** stellt die Referenz auf ein anderes Objekt dar welches ebenso durch den **DigScriptManager** verwaltet wird. Es besitzt zusätzlich einen vollen Namen des zu referenzierenden Objektes(sourceFullName), den Pfad zu dem Objekt(sourcePath). Das **InnerScript** beinhaltet noch den Namen des Inneren Skriptes(name) und die Version(version). Das **InnerObject** besitzt ebenso den Namen des Skriptes(name), sowie den Type des Objektes(objectType). === Klassen und Enumerationen Dieser Abschnitt gibt eine Kurzbeschreibung der Klassen/Enumerationen im Analyseklassendiagramm an.

Klasse/Enumeration	Beschreibung
<b>Script</b>	Ein Skript ist das zu verwaltende Objekt im <b>DigScriptManager</b> . Es enthält alle nötigen Daten des Skripts (eventuell als referenziertes Metadatenobjekt).
<b>MetaDaten</b>	Das <b>MetaDaten</b> -Objekt enthält Merkmale von Skripten wie Kategorie, Autor etc. . Es kann daher nicht ohne ein referenzierendes Skript bestehen.
<b>Dependency (abstrakt)</b>	Eine <b>Dependency</b> ist eine Abhängigkeit eines Skriptes von anderen Skripten oder Objekten. Sie kann unterschiedliche Ausprägungen haben. Vererbung von dieser Klasse stellt ferner den Erweiterungspunkt für andere mögliche Abhängigkeiten dar.

Klasse/Enumeration	Beschreibung
<b>InnerScript</b>	Ein InnerScript ist eine mögliche Ausprägung von Dependency. Diese Klasse beschreibt lediglich, wo das referenzierte innere Skript liegt, nicht das innere Skript selbst. Das innere Skript liegt beliebig geschachtelt unterhalb des Vaterskriptes.
<b>InnerObject</b>	Ein InnerObject ist eine mögliche Ausprägung von Dependency. Diese Klasse beschreibt, wo das referenzierte Objekt liegt.
<b>Objectreference</b>	Eine Objectreference ist eine mögliche Ausprägung von Dependency. Es beschreibt die Abhängigkeit von einem beliebigen anderen Skript oder Objekt, welches ebenfalls vom DIgScriptManager verwaltet wird.
<b>InputParam</b>	Ein InputParam ist ein Eingabeparameter für ein Skript. Er enthält Beschreibung, Einheit, Name, Datentyp und den Standardwert des Parameters.
<b>OutputParam</b>	Ein OutputParam ist ein Ausgabeparameter für ein Skript. Er enthält Beschreibung, Einheit, Name und Datentyp des Parameters.
<b>Webservice</b>	Der Webservice stellt Funktionen zum Suchen sowie Up- und Download von Skripten bereit.

# Chapter 11. Akzeptanztestfälle

Mithilfe von Akzeptanztests wird geprüft, ob die Software die funktionalen Erwartungen und Anforderungen im Gebrauch erfüllt. Diese sollen und können aus den Anwendungsfallbeschreibungen und den UML-Sequenzdiagrammen abgeleitet werden. D.h., pro (komplexen) Anwendungsfall gibt es typischerweise mindestens ein Sequenzdiagramm (welches ein Szenarium beschreibt). Für jedes Szenarium sollte es einen Akzeptanztestfall geben. Listen Sie alle Akzeptanztestfälle in tabellarischer Form auf. Jeder Testfall soll mit einer ID versehen werden, um später zwischen den Dokumenten (z.B. im Test-Plan) referenzieren zu können.

<b>ID</b>	[AT0010]
<b>Use Case</b>	[UC0010]
<b>Vorbedingung(en)</b>	Das Skript liegt noch nicht beim Webservice bereit.
<b>Event</b>	Ein Nutzer sendet die XML-Beschreibung eines Skriptes an den Webservice.
<b>Erwartetes Resultat</b>	<ul style="list-style-type: none"><li>• Das Skript liegt auf dem Webserver und wird vom Manager verwaltet.</li><li>• Das Skript erscheint in der Suche.</li></ul>

<b>ID</b>	[AT0011]
<b>Use Case</b>	[UC0010]
<b>Vorbedingung(en)</b>	Das Skript liegt bereits beim Webservice bereit.
<b>Event</b>	Ein Nutzer sendet die XML-Beschreibung desgleichen Skripts an den Webservice.
<b>Erwartetes Resultat</b>	<ul style="list-style-type: none"><li>• Der Nutzer bekommt eine Fehlermeldung zurück, die aussagt, dass das Skript bereits auf dem Webservice liegt und nicht überschrieben werden darf.</li></ul>

<b>ID</b>	[AT0020]
<b>Use Case</b>	[UC0020]
<b>Vorbedingung(en)</b>	Skript liegt beim Webservice bereit
<b>Event</b>	Ein Nutzer sendet Skriptnamen an den Webservice mit der Intention, dieses Skript herunterzuladen.
<b>Erwartetes Resultat</b>	Rückgabe der XML-Beschreibung des Skripts an den Nutzer.

<b>ID</b>	<a href="#">[AT0021]</a>
<b>Use Case</b>	<a href="#">[UC0020]</a>
<b>Vorbedingung(en)</b>	Skript liegt nicht beim Webservice
<b>Event</b>	Ein Nutzer sendet Skriptnamen an den Webservice mit der Intention, dieses Skript herunterzuladen.
<b>Erwartetes Resultat</b>	Eine Fehlermeldung wird zurückgegeben, die angibt, dass das Skript nicht vorhanden ist.

<b>ID</b>	<a href="#">[AT00301]</a>
<b>Use Case</b>	<a href="#">[UC0030]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, nach Suchkriterien zutreffende Skripten liegen im Server.
<b>Event</b>	Ein Nutzer sendet eine Anfrage an den Webservice mit gewünschten Metadaten, nach denen gesucht werden soll.
<b>Erwartetes Resultat</b>	Rückgabe einer Liste von zutreffenden Skripten.

<b>ID</b>	<a href="#">[AT00302]</a>
<b>Use Case</b>	<a href="#">[UC0030]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, keine nach Suchkriterien zutreffende Skripten liegen im Server.
<b>Event</b>	Ein Nutzer sendet eine Anfrage an den Webservice mit gewünschten Metadaten, nach denen gesucht werden soll.
<b>Erwartetes Resultat</b>	Anzeige, dass keine Skripts mit den gegebenen Metadaten gefunden wurden.

<b>ID</b>	<a href="#">[AT00311]</a>
<b>Use Case</b>	<a href="#">[UC0031]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, zumindest ein Skript mit dem zu suchenden Namen befindet sich in der Serverseite.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem gewünschten Skriptnamen, nach dem gesucht werden soll.
<b>Erwartetes Resultat</b>	Rückgabe einer zutreffenden Liste von Skripten.



<b>ID</b>	<a href="#">[AT00312]</a>
<b>Use Case</b>	<a href="#">[UC0031]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, kein Skript mit dem zu suchenden Namen befindet sich in der Serverseite.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem gewünschten Skriptnamen, nach dem gesucht werden soll.
<b>Erwartetes Resultat</b>	Anzeige, dass kein Skript mit dem gesuchten Namen gefunden wurde.

<b>ID</b>	<a href="#">[AT00321]</a>
<b>Use Case</b>	<a href="#">[UC0032]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, zumindest ein Skript mit dem zu suchenden Tag befindet sich in der Serverseite.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem gewünschten Tag, nach dem gesucht werden soll.
<b>Erwartetes Resultat</b>	Rückgabe einer zutreffenden Liste von Skripten.

<b>ID</b>	<a href="#">[AT00322]</a>
<b>Use Case</b>	<a href="#">[UC0032]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, kein Skript mit dem zu suchenden Tag befindet sich in der Serverseite.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem gewünschten Tag, nach dem gesucht werden soll.
<b>Erwartetes Resultat</b>	Anzeige, dass kein Skript mit dem gesuchten Tag gefunden wurde.

Die restlichen Akzeptanztests für [\[UC0033\]](#), [\[UC0034\]](#), [\[UC0035\]](#), [\[UC0036\]](#) sind analog.

<b>ID</b>	<a href="#">[AT00401]</a>
<b>Use Case</b>	<a href="#">[UC0040]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden. Es existieren Skripte, die durch das mit UniqueName bezeichneten Skript referenziert werden.

<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem UniqueName, für den eine Liste der Metadaten der referenzierten Skripten zurückgegeben werden sollen.
<b>Erwartetes Resultat</b>	Rückgabe einer Liste zutreffenden Metadaten von Skripten, welche vom Skript mit UniqueName referenziert sind.

<b>ID</b>	<a href="#">[AT00402]</a>
<b>Use Case</b>	<a href="#">[UC0040]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden. Es existieren keine Skripte, die durch das mit UniqueName bezeichneten Skript referenziert werden.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem UniqueName, für den eine Liste der Metadaten der referenzierten Skripten zurückgegeben werden sollen.
<b>Erwartetes Resultat</b>	Anzeige, dass keine Skripts durch das mit dem angegeben UniqueName bezeichnete Skript referenziert werden.

<b>ID</b>	<a href="#">[AT00411]</a>
<b>Use Case</b>	<a href="#">[UC0041]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, zumindest ein referenziertes Skript befindet sich NICHT in der aktuellen version bzw. es ist nicht vorhanden.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem UniqueName, für den eine Liste der Metadaten der noch nicht bereits vorhandenen (Seite des Nutzers) Skripten zurückgegeben werden sollen.
<b>Erwartetes Resultat</b>	Rückgabe einer Liste mit Metadaten von Skripten, welche vom Skript mit UniqueName referenziert sind und sich noch nicht auf der aktuellsten Version befinden bzw. installiert werden müssen (Clientseite).

<b>ID</b>	<a href="#">[AT00412]</a>
<b>Use Case</b>	<a href="#">[UC0041]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, alle referenzierte Skripte befinden sich in der aktuellen Version.

<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem UniqueName, für den eine Liste der Metadaten der noch nicht bereits vorhandenen (Seite des Nutzers) Skripten zurückgegeben werden sollen.
<b>Erwartetes Resultat</b>	Anzeige, dass sich alle Abhängigkeiten bereits in der aktuellsten Version befinden.

<b>ID</b>	<a href="#">[AT00421]</a>
<b>Use Case</b>	<a href="#">[UC0042]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden, alle abhängige Skripte eines Skripts befinden sich auf dem Webservice.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem UniqueName, für den geprüft werden soll, ob alle abhängige Skripte auf dem Webservice vorhanden sind.
<b>Erwartetes Resultat</b>	Rückgabe einer leeren Liste.

<b>ID</b>	<a href="#">[AT00501]</a>
<b>Use Case</b>	<a href="#">[UC0050]</a>
<b>Vorbedingung(en)</b>	Skriptmanager ist vorhanden.
<b>Event</b>	Der Nutzer sendet eine Anfrage an den Webservice mit einem FullName, für welchen die Metadaten des entsprechenden Skriptes in der aktuellen Version zurückgegeben werden müssen.
<b>Erwartetes Resultat</b>	Rückgabe der Metadaten des Skriptes, das diesen Fullname hat.

# Chapter 12. Glossar

- Kunde: Auftraggeber (Kunde des Projekts)
- Nutzer: Mitarbeiter von DigSILENT, die Skripte verwalten wollen
- REST: Representational State Transfer
- GUI: Graphical User Interface:
- XML: Extensible Markup Language: Darstellungssprache für hierarchische Daten