

# Software Requirements Specification

Version 1.0  
21. Mai 2013

ISGCI - Information System on Graph Classes and their Inclusions  
Team *Graph Maga*

# Inhaltsverzeichnis

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Einleitung / Introduction</b>   | <b>3</b> |
| 1.1      | Zweck (des Dokuments) / Purpose . . . . .  | 3        |
| 1.2      | Umfang (des Softwareprodukts) / Scope . . . . .  | 3        |
| 1.3      | Erläuterungen zu Begriffen und / oder Abkürzungen (Glossar) / Definitions,<br>acronyms and abbreviations . . . . . | 3        |
| 1.4      | Verweise auf sonstige Ressourcen oder Quellen / References . . . . .   | 3        |
| 1.5      | Übersicht/Überblick (Wie ist das Dokument aufgebaut?) / Overview . . . . .   | 3        |
| <b>2</b> | <b>Allgemeine Beschreibung</b>   | <b>4</b> |
| 2.1      | Produktperspektive . . . . .   | 4        |
| 2.2      | Produktfunktionen . . . . .  | 4        |
| 2.3      | Benutzermerkmale . . . . .   | 4        |
| 2.4      | Einschränkungen . . . . .  | 4        |
| 2.5      | Anteil der Anforderungen . . . . .   | 4        |
| <b>3</b> | <b>Spezifische Anforderungen</b>   | <b>5</b> |
| 3.1      | Funktionale Anforderungen (Stark abhängig von der Art des Softwareprodukts) .                                      | 5        |
| 3.2      | Nicht-funktionale Anforderungen . . . . .  | 5        |
| 3.3      | Externe Schnittstellen - Anforderungen . . . . .   | 5        |
| 3.4      | Design Constraints (ja das englisch war so vorgegeben und darf gerne geändert<br>werden :-) ) . . . . .            | 6        |
| 3.5      | Qualitätsanforderungen . . . . .   | 6        |
| 3.6      | Sonstige Anforderungen . . . . .   | 6        |
| 3.7      | Ergänzende Kommentares . . . . .   | 6        |

# 1 Einleitung / Introduction

{definiere erwartete Leserschaft des Dokuments & Version History.

Grund für die Erstellung der neuen Version, bzw. Zusammenfassung der Veränderungen}

## 1.1 Zweck (des Dokuments) / Purpose

{Wieso braucht man das System. Kurze Beschreibung der System Funktionen & wie es mit anderen Systemen arbeiten soll.

(wie passt das System in gesamte Unternehmen || stragische Ziele der Organisation d. Software)}

## 1.2 Umfang (des Softwareprodukts) / Scope

{...}

## 1.3 Erläuterungen zu Begriffen und / oder Abkürzungen (Glossar) / Definitions, acronyms and abbreviations

{ Definiere technische Begriffe, die im Dokument verwendet werden und nicht "klar" sind. Keine Annahmen über die Erfahrungen und Fachkenntnis des Lesers machen.}

## 1.4 Verweise auf sonstige Ressourcen oder Quellen / References

{Hyperlinks? <http://www.graphclasses.org/>

ISGCI, JGraphX als Grafikbibliothek, JGraphT}

## 1.5 Übersicht/Überblick (Wie ist das Dokument aufgebaut?) / Overview

{Ist dieser Abschnitt notwendig?!.. zur Not einfach kippen}

## 2 Allgemeine Beschreibung

### 2.1 Produktperspektive

{Beschreibung, dass unseres Produkt als "Schnittstelle" zwischen JGraphX und ISGCI fungieren wird  
und wie diese miteinander interagieren, bzw. wie darauf aufgebaut wird.}

### 2.2 Produktfunktionen

{Beschreibung der Funktionen und entsprechende tabellarische Übersicht.}

### 2.3 Benutzermerkmale

Die Zielgruppe des fertigen umgesetzten Systems sind in erster Linie diejenigen, die sich mit den grundlegenden Eigenschaften bereits auskennen und dementsprechend weitere Nachforschungen in Bereich Graphen und deren Verbindungen anstellen wollen. Daher werden viele Grundlagen nicht näher erläutert. Vielmehr sollen Personen, die sich mit Zusammenhängen verschiedener Graph-Klassen näher beschäftigen wollen, durch das fertige System einfach Zugang zu Informationen verschaffen können, ohne dass sie weitere (intensive) Hilfe zur Nutzung des gesamten Systems benötigen. Letztendlich dient das System also zu Forschung und zum besseren Verständnis im Studium oder Beruf von Graphen.

### 2.4 Einschränkungen

Die Implementation des Systems ist Java-basiert und die aktuelle Inkompatibilität mit aktuellen Java-Versionen erfordert eine Erneuerung des Systems. Dies führt zu verschiedenen Beschränkungen, die beachtet werden müssen.

Die Umsetzung findet für Java-Versionen ab Java 1.6 statt. Dies besagt, dass bei niedrigeren installierten Java-Versionen keine volle Funktionalität gewährleistet wird.

Die graphische Verbesserung des bestehenden Systems muss insbesondere über folgende Bibliotheken umgesetzt werden: JGraphX wird als neue Zeichenbibliothek fungieren.

Die Umsetzung findet statt, indem die Funktionalität von JGraphT auf JGraphX übertragen wird und anschließend ISGCI an JGraphX angebunden wird. So wird JGraphT letztendlich nicht zur Ausführung der Software verwendet.

Dementsprechend ist ausdrücklich gesagt, dass alle Beschränkungen, die für die genannten Bibliotheken gelten, insbesondere für die fertige Implementation gelten.

Die Software wird nach Umsetzung auf jedem System ausführbar sein, welches die angegebenen Voraussetzungen erfüllt (Java 1.6 +).

Die Software wird als Open-Source-Project gehandhabt. Daher wird jede Dokumentation, die zur Erweiterung des Source-Codes stattfindet, in englischer Sprache formuliert, sodass unsere Implementation zur weiteren Nutzung/Forschung beitragen kann.

### 2.5 Anteil der Anforderungen

Im nächsten Kapitel beschriebene funktionale Anforderungen müssen umgesetzt werden, damit das System/deren Erweiterung die entsprechende Funktionalität hat, damit ISGCI an JGraphX gebunden werden kann und alle bisherigen und hinzukommende Funktionen (außer evtl. Widersprüche) ausgeführt werden können.

## 3 Spezifische Anforderungen

### 3.1 Funktionale Anforderungen (Stark abhängig von der Art des Softwareprodukts)

{Requirements Elicitation !!!, Funktionen / Functions}

### 3.2 Nicht-funktionale Anforderungen

- Technische Grundlage des Softwareprojekts bietet die Programmiersprache Java.
- Kompatibilität: Die Software ISGCI wird Java-Runtime-Umgebungen mit Java 1.6 und Nachfolger vollständig unterstützen. Für ältere Versionen wird die Lauffähigkeit nicht garantiert.
- Zur Umsetzung graphischer Anwendungen wird die Grafikbibliothek JGraphX verwendet.
- Das Projekt wird als OpenSource veröffentlicht.
- Das Benutzerinterface soll interaktiver und userfreundlicher sein als bei der Vorgängerversion (Umsetzung: vgl. Functional Requirements).
- Dokumentationen werden auf Englisch verfasst, um sich dem OpenSource Standard und der restlichen Dokumentation anzupassen.
- Zeichnungen im ISGCI sollen visuell attraktiver dargestellt werden.
- Datenbank-basiertes zeichnen: Flexible Einspeisung von Datensätzen anstatt statischer Einbindung führt zu einer bessern Wartbarkeit.

### 3.3 Externe Schnittstellen - Anforderungen

#### *Benutzerschnittstellen:*

Der Benutzer benötigt zum vollständigen Nutzen des Systems verschiedene Schnittstellen. Er benötigt Peripheriegeräte (Maus, Tastatur) und deren Treiberunterstützung entsprechen der Hersteller, damit die Funktionalitäten des Systems nutzbar sind.

Der Benutzer benötigt eine grafische Ausgabe (Display/Bildschirm) um das Programm anzeigen lassen zu können. Die Anzeige kann gemäß der benutzerdefinierten Auflösung durch Ziehen des Fensters beliebig angepasst werden.

#### *Hardware Schnittstellen:*

Die fertige Software wird in einer Java-Umgebung dargestellt. Daher wird gefordert, dass die Hardware mit der Java-Umgebung ab der in den nicht-funktionalen Anforderungen genannten Version 1.6 kompatibel sein muss, damit die Software reibungslos funktionieren kann.

#### *Software Schnittstellen:*

JGraphX: Eine Java Swing Visualisierungsbibliothek von mxGraph.

ISGCI: Eine bereits umgesetzte Graphenbibliothek, welche viele Informationen über verschiedene Graphklassen beinhaltet (wie Sub-/Superklassen) und ausführliche Informationen derer Eigenschaften.

JGraphT: Eine freie Java Klassenbibliothek, die mathematische graphentheoretische Ziele und Algorithmen unterstützt. Diese läuft allerdings nur auf der Java 2 Plattform, welche mindestens JDK 1.6 voraussetzt.

Die Software wird eine Schnittstelle zwischen diesen beiden Bibliotheken bilden, indem die

Graphzeichnungsmöglichkeiten von JGraphX genutzt werden. Dementsprechend werden die aktuellen Funktionen von ISGCI, die bis dato mit JGraphT umgesetzt sind, auf JGraphX übertragen, indem eine Schnittstelle zwischen JGraphT und JGraphX umgesetzt wird und deren Funktionalität dann von ISGCI auf JGraphX über diese Schnittstelle übertragen wird.

JGraphT verwendet Maven um den Build Process zur Kompilierung der Java-Klassen und um eine gute Dokumentation zu gewährleisten. Insbesondere führt Maven JUnit-Tests aus, um eben die Kontrolle über die Implementierung zu behalten. Dazu müssen allerdings entsprechende Tests vom Entwickler definiert werden.

### 3.4 Qualitätsanforderungen

- **Benutzerfreundlichkeit**

Das bestehende System wird vor Allem erweitert um die Benutzerfreundlichkeit zu steigern. Zwar besteht die Zielgruppe für die Software aus technisch interessierten und fähigen Leuten, dennoch soll die Software auch für Laien leicht durchschaubar und selbsterklärend sein. Intuitiv angeordnete Reiter in der Menüleiste, sowie ein interaktives Kontextmenü sollen dazu beitragen, dass benötigte Funktionen von Benutzern leicht gefunden werden können. Des Weiteren sorgen Funktionen wie Grab&Pull oder Zoom (*vgl. Funktionale Anforderungen*) für einen interaktiven Umgang mit dem System. Um die Übersichtlichkeit über komplexere Graphen zu gewährleisten, gibt es zwei Varianten des Expanding/Collapsing, mit denen man einen komplizierten Graphen in einen überschaubaren Teilgraphen unterbrechen kann (*vgl. Funktionale Anforderungen 08*).

- **Integrität**

Alle Datensätze werden zentral auf einem Server gehalten. Die Software bietet ausschließlich lesende Funktionen an. Somit ist die Datenbank durch negative Manipulationen geschützt.

- **Flexibilität**

Nutzer können ihren Wunschgraphen zeichnen lassen, diesen auf verschiedene Arten manipulieren (z.B.: neu anordnen, reduzieren, Knoten-Hierarchien anzeigen lassen) und den entstandenen Graphen exportieren. Dazu lassen sich zu jedem Graph-Knoten(Graphen) Informationen aus der Online-Datenbank einsehen.

- **Portabilität**

ISGCI ist eine Java Anwendung. Das bedeutet, dass sie plattformunabhängig ausgeführt werden kann, solange eine Java-Laufzeit-Umgebung installiert ist (als Standard vorausgesetzt). Die Online-Datenbank gewährleistet

- **Wartbarkeit**

Dadurch, dass die Datenbank des Programmes online, zentral gelagert wird, lassen sich Datensätze leicht und ohne Updates im Programm manipulieren oder hinzufügen.