

Software Requirements Specification

Version 1.3
01. Juli 2013

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

Inhaltsverzeichnis

1	Änderungsverlauf	2
2	Einleitung	3
2.1	Zweck	3
2.2	Umfang	3
2.3	Erläuterungen zu Begriffen und Abkürzungen	3
2.4	Verweise auf sonstige Quellen	4
3	Allgemeine Beschreibung	5
3.1	Produktperspektive	5
3.2	Produktfunktionen	5
3.3	Benutzermerkmale	6
3.4	Einschränkungen	6
3.5	Zuweisung der Anforderungen	7
4	Spezifische Anforderungen	8
4.1	Funktionale Anforderungen	8
4.2	Nicht-funktionale Anforderungen	11
4.3	Externe Schnittstellen - Anforderungen	11
4.4	Qualitätsanforderungen	14

1 Änderungsverlauf

Version	Datum	Beschreibung der Änderung
1.0	21.05.2013	Veröffentlichung
1.1	23.05.2013	Änderungen nach Meeting
		Formulierungen
1.2	07.06.2013	SRS den Aktuellen Mockups anpassen
	07.06.2013	Einschränkungen ergänzt durch: ISGCI benötigt Internet
	07.06.2013	Funktionale Anforderungen ergänzt durch:
		Kontextmenü komplett überarbeitet
		Selection (Menü-Bar-Parent) ergänzt
		Sidebar hinzugefügt (Mockup)
1.3	01.07.2013	Zusätzliches Feature: „Mark Neighbors“ (vgl. 4.1 - 22)

2 Einleitung

2.1 Zweck

Der Zweck dieses Dokuments ist es eine detailreiche Beschreibung des aktualisierten ISGCI-Projektes zu geben. Es wird neue Funktionalitäten, Schnittstellen, Einschränkungen und seine Reaktionen auf externe Einflüsse beschreiben. Dieses Dokument richtet sich sowohl an den Kunden als auch an die Entwickler.

2.2 Umfang

Das gelieferte Produkt ist eine Aktualisierung des bestehenden ISGCI. Das fertige Produkt wird ein Update auf eine neuere Java Version beinhalten. Das bestehende ISGCI wird mit Funktionalität der Graphzeichenbibliothek JGraphX erweitert. Außerdem werden der neuen Version verschiedene neue Funktionalitäten hinzugefügt, die im Punkt 3 genauer beschrieben werden. Zweck des Updates ist es, nicht nur die Funktionalität des Tools in Zukunft zu gewährleisten und neue Funktionen zu implementieren, sondern auch die Instandhaltungskosten zu senken und das Tool für interessierte Entwickler und Nutzer ansprechender zu gestalten.

2.3 Erläuterungen zu Begriffen und Abkürzungen

mxGraph mxGraph ist eine Produktfamilie mehrerer Bibliotheken, die mit verschiedenen Technologien geschrieben sind, welche eine Ausstattung besitzen, mit der interaktive Diagramme und Graphen visualisiert werden können. Zu diesen Bibliotheken gehören unter anderem JGraphX und JGraph (vorige Versionen von JGraphX).

ISGCI Information System on Graph Classes and their Inclusions

JGraphT JGraphT ist eine freie Java Graphenbibliothek, welche mathematische graphentheoretische Objekte und Algorithmen unterstützt.

JGraphX JGraphX ist eine Java Swing Bibliothek von mxGraph

Swing Swing ist das primäre Java GUI Widget Werkzeug, welche die Bereitstellung von grafische Benutzerschnittstellen ermöglicht.

Click&Drag Bezeichnet eine Funktion, welche das Greifen und Ziehen von Graphenelementen beinhaltet, welche den entsprechenden Graphen manipulieren können.

Menu-Bar-Parent Bezeichnet einen Reiter in der oberen Navigationsbar.

Menu-Bar-Child Bezeichnet ein Item unterhalb des Reiters in der Menu-Bar.

2.4 Verweise auf sonstige Quellen

Tabelle 1: Verweise

Name	Quelle	Datum
ISGCI	Graphclasses.org - ISGCI Homepage	13.05.2013
JGraphT	JGraphT Website	13.05.2013
JGraphX	JGraph Website	13.05.2013
	JGraph Dokumentation	13.05.2013
	JGraphX Library Quellcode	13.05.2013
Swing	Swing Dokumentation	13.05.2013
Java	Java 6 Dokumentation SE	13.05.2013

3 Allgemeine Beschreibung

3.1 Produktperspektive

Unser Produkt wird als Schnittstelle zwischen JGraphX, JGraphT und ISGCI fungieren. Durch die Erweiterung von JGraphX wird dessen Funktionalität verbessert. Mit der Verknüpfung von JGraphX als Zeichenbibliothek werden verschiedene neue Interaktionen ermöglicht und ein Grundbaustein für die Umsetzung des ISGCI als Webapp gelegt. Vor allem, da JGraphX ein OpenSource Projekt ist, verspricht unser Produkt auch in der Zukunft weiterhin neue Funktionen zu erhalten und die Erweiterbarkeit zu vereinfachen.

3.2 Produktfunktionen

1. Das System soll einen Graphen zeichnen können, welcher Relationen zwischen Graphklassen zeigt.
2. Das System soll den erzeugten Graphen exportieren können.
3. Das System soll die Möglichkeit bieten im gezeichneten Graphen eine Graphklasse zu suchen.
4. Das System soll dazu in der Lage sein, die Namensgebung der Graphklassen innerhalb des erzeugten Graphen nach vorgegebenen Mustern zu ändern.
5. Das System soll unklare Inklusionen markieren können. (unklare Inklusionen: Beziehungen zwischen zwei Graph-Klassen, bei denen nicht sicher ist, ob die Menge der Graphen der einen Klasse echte Teilmenge der Menge der Graphen der anderen Graphenklasse sind, oder der Menge selbst entsprechen. Keine Information bzgl. \subset / \subseteq in der Datenbank vorhanden.)
6. Das System soll die Graphdatenbank und die Eigenschaften der Graphklassen anzeigen können.
7. Das System soll die Möglichkeit bieten, nach Relationen zwischen Graphklassen zu suchen.
8. Das System soll die Möglichkeit bieten, zu einem Graph-Problem die entsprechenden Laufzeitklassen zu visualisieren.
9. Das System soll die Möglichkeit bieten, Graph-Probleme im erzeugten Graphen anzuzeigen.
10. Das System soll einen Hyper-Link zu `small graphs` besitzen.
11. Das System soll einen Hyper-Link zur Hilfe Website von ISGCI besitzen.
12. Das System soll eine About Funktion besitzen, welche Versions-, Entwickler- und Software-Informationen enthält.
13. Das System soll die Möglichkeit bieten, einen Knoten im erzeugten Graphen zu markieren.

14. Das System soll die Möglichkeit bieten, einen Knoten im erzeugten Graphen zu verschieben.
15. Das System soll die Möglichkeit bieten, über ein Kontextmenü zu einer Graphklasse im erzeugten Graphen weitere Informationen zu dieser zu erhalten.
16. Das System soll die Möglichkeit bieten, im erzeugten Graphen zu zoomen.
17. Das System soll, wenn ein Graph gezeichnet wird, die für die Zeichnung ausgewählte Graphklasse im Bild zentriert darstellen.
18. Das System soll die Möglichkeit bieten, durch das Gedrückthalten der linken Maustaste und Ziehen den Bildausschnitt zu verschieben.
19. Das System soll Funktionen beinhalten, mithilfe derer man Nachbarknoten beziehungsweise Super- und Subklassen der gewählten Graphklasse ein-/ausblenden kann. Dies soll auch über ein Kontextmenü möglich sein.

3.3 Benutzermerkmale

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

3.4 Einschränkungen

Die Implementation ist Java-basiert und die Inkompatibilität mit aktuellen Java-Versionen (ab 1.6) 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.

Deshalb kann bei niedrigeren Java-Versionen keine Funktionalität gewährleistet werden.

Die graphische Verbesserung des bestehenden Systems wird insbesondere über JGraphX als neue Zeichenbibliothek umgesetzt.

Die Umsetzung findet statt, indem die Funktionalität von JGraph(5) auf JGraphX übertragen wird und anschließend ISGCI an JGraphX angebunden wird. Die Software wird nach Umsetzung auf jedem System ausführbar sein, welches die angegebenen Voraussetzungen erfüllt (Java 1.6 +).

Die Software wird als Open-Source-Projekt 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.

Des weiteren wird eine Internetverbindung vorausgesetzt um das ISGCI zu starten. Die Verwendung ist dann aber ohne Internetverbindung möglich.

3.5 Zuweisung der Anforderungen

Die im nächsten Kapitel beschriebenen funktionalen Anforderungen müssen umgesetzt werden, damit das System bzw. deren Erweiterung die entsprechende Funktionalität hat. Das heißt, dass ISGCI an JGraphX gebunden werden kann und alle bisherigen und hinzukommende Funktionen (außer evtl. Widersprüche) ausgeführt werden können.

4 Spezifische Anforderungen

4.1 Funktionale Anforderungen

Die bisher vorhandenen funktionalen Anforderungen werden, soweit nicht abweichend angegeben, übernommen.

	Name	Beschreibung
1	Knoten Verschieben	Die Funktion wird dahingehend abgeändert dass ein Knoten vor dem Verschieben markiert werden muss.
2	Zoom (Mausrad)	Durch das Scrollen mit dem Mausrad wird das Bild gezoomt, Mausrad nach oben entspricht hinein zoomen, Mausrad nach unten entspricht heraus zoomen.
3	Zoom (Menu-Bar)	Dem Menu-Bar-Parent View werden die Funktionen SZoom in und SZoom out hinzugefügt, diese können auch per Shortcut ausgeführt werden (alt + +, alt + -).
4	Fit in window	Der Zoom wird so angepasst, dass der komplette Graph im Canvas sichtbar ist.
5	Tooltip Werkzeuge	Verweilt die Maus auf einem Menü-Punkt, wird ein Tooltip angezeigt. Der angezeigte Tooltip beinhaltet Informationen zur jeweiligen Funktion.
6	Tooltip Knoten	Verweilt die Maus auf einem Knoten im Graphen, werden alle äquivalenten Graphklassen im Tooltip angezeigt (vgl. Rechtsklick, change name).
7	Zentrierung	Wird ein neuer Graph gezeichnet ist der zum Zeichnen ausgewählte Knoten im Mittelpunkt des Bildes.
8	Scrolling / click and drag	Klickt man in den freien Raum und hält die linke Maustaste gedrückt so kann man durch das Bewegen der Maus den angezeigten Bildausschnitt verschieben.
9	Kontextmenü	Man kann Knoten mit der rechten Maustaste anklicken um ein Kontextmenü zu öffnen.
10	Kontextmenü Show information	Beinhaltet die bereits vorhandene Funktionalität.
11	Kontextmenü Show details	Blendet die Sidebar ein. (vgl. Sidebar Punkt 21)
12	Kontextmenü Change name	Beinhaltet die bereits vorhandene Funktionalität.
13	Kontextmenü Show neighbours	Kontextmenü-item um vom jeweiligen Knoten alle Nachbarknoten einzublenden.
14	Kontextmenü Hide neighbours	Kontextmenü-item um vom jeweiligen Knoten alle Nachbarknoten auszublenden.
15	Kontextmenü Show superclasses	Kontextmenü-item um vom jeweiligen Knoten alle Superklassen einzublenden.
16	Kontextmenü Hide superclasses	Kontextmenü-item um vom jeweiligen Knoten alle Superklassen auszublenden.
17	Kontextmenü Show subclasses	Kontextmenü-item um vom jeweiligen Knoten alle Subklassen einzublenden.
18	Kontextmenü Hide subclasses	Kontextmenü-item um vom jeweiligen Knoten alle Subklassen auszublenden.

19	Selection (Menu-Bar-Parent)	Es soll einen neuen Reiter in der Menu-Bar geben, der alle Funktionen des Kontextmenüs, außer 12 Change Name, für einen ausgewählten Knoten zur Verfügung stellt (Funktionen 10,11,13-18). Wenn kein Knoten ausgewählt ist, werden alle Child-Items grau dargestellt und besitzen keine Funktionalität.
20	Details visible (Menu-Bar-Child)	Im Menu-Bar-Parent View soll ein Punkt Details visible existieren. Dieser signalisiert (mit einem Häkchen) ob, die Sidebar ein- (Häkchen) oder ausgeblendet (kein Häkchen) ist. Außerdem kann über diesen auch die Sidebar ein und ausgeblendet werden.
21	Sidebar	Die Sidebar soll sich durch 3 verschiedene Wege öffnen lassen. Im Kontextmenü über Show details, im Menu-Bar-Parent Selection mit Show details und im Menu-Bar-Parent View über Details visible, falls noch nicht sichtbar. Die Sidebar soll auf der linken Seite sein und wenn nötig eine Scrollbar auf der rechten Seite haben. Die Sidebar wird in ihrer Breite anpassbar sein. Durch einen Button oben in der Sidebar lässt sie sich ausblenden. Außerdem über Details visible in View. Durch das Markieren eines anderen Knoten wird der Inhalt der Sidebar auf die ausgewählte Graphklasse angepasst und deren Informationen in dargestellt. Ist kein Knoten markiert, werden die Informationen der zuletzt ausgewählten Graphklasse angezeigt. Sollte die Sidebar ohne eine Auswahl getroffen zu haben, angezeigt werden, wird die Sidebar leer dargestellt. In der Sidebar wird es möglich sein vertikal in einer Accordion-Navigation zu navigieren. Diese besitzt drei Selektionsmöglichkeiten: Graphclasses, Inclusions, Problems. Dabei führt die Auswahl einer Selektion, zum Einklappen der anderen Selektionsmöglichkeiten.
21	Sidebar - Selection:Graphclasses	In der Selection:Graphclasses werden allgemeine Informationen über die ausgewählte Graphklasse angezeigt.
21	Sidebar - Selection:Inclusions	In der Selection:Inclusions werden Informationen über die Inklusionen der ausgewählten Graphklasse angezeigt.
21	Sidebar - Selection:Problems	In der Selection:Problems werden die Probleme der ausgewählten Graphklasse angezeigt.

4.2 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, solange diese abwärtskompatibel zu Java 1.6 sind. 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. Dafür muss der Standard der bisherigen Dokumentation eingehalten werden.
- Zeichnungen im ISGCI sollen visuell attraktiver dargestellt werden.
- Online Datenbank: Flexible Einspeisung von Datensätzen anstatt statischer Einbindung führt zu einer bessern Wartbarkeit. Auslagerung der Daten auf einen Online-Server statt lokaler Speicherung.

4.3 Externe Schnittstellen - Anforderungen

Benutzerschnittstellen:

Der Benutzer benötigt zum vollständigen Nutzen des Systems verschiedene Schnittstellen.

Er benötigt Peripheriegeräte (Maus, Tastatur) und deren Treiberunterstützung entsprechender 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.

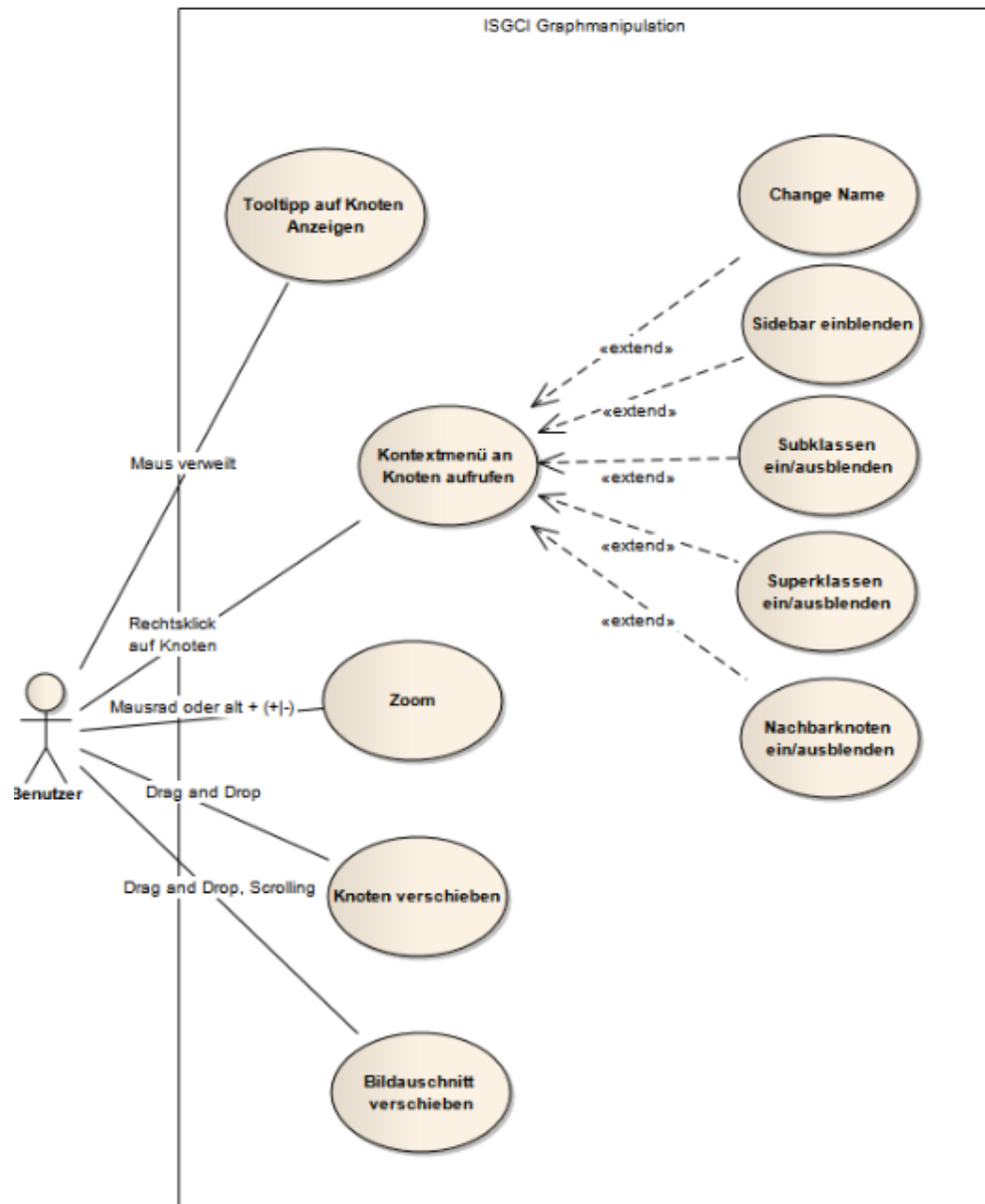


Abbildung 2: Dieses Usecase zeigt, wie ein Benutzer mit einem Graphen interagieren kann und ihn manipulieren. Dazu gehören Funktionen wie Zoom, Knoten verschieben, Bildausschnitt Verschieben und Tooltips anzeigen.

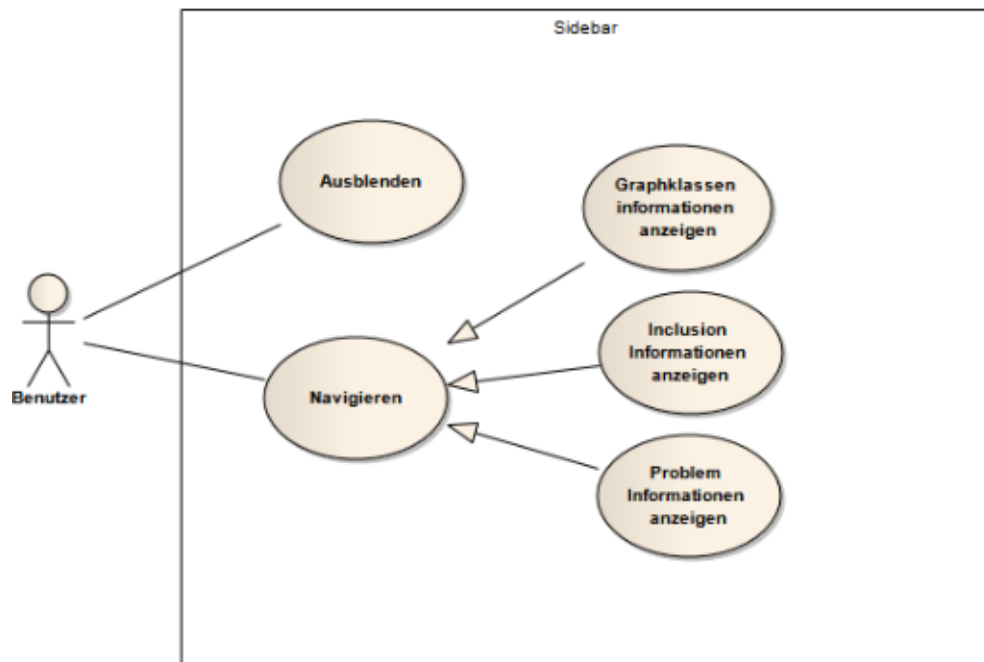


Abbildung 3: Dieses Usecase veranschaulicht die möglichen Interaktionsmöglichkeiten mit der Sidebar, welche neu im ISGCI sein wird.

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 Bibliotheken bilden, indem die

Graphzeichnungsmöglichkeiten von JGraphX genutzt werden. Dementsprechend werden die aktuellen Funktionen von ISGCI, die bis dato mit JGraph 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 die Kontrolle über die Implementierung zu behalten. Dazu müssen entsprechende Tests vom Entwickler definiert werden.

4.4 Qualitätsanforderungen

- **Benutzerfreundlichkeit**

Das bestehende System wird vor allem erweitert, um die Benutzerfreundlichkeit zu steigern. Zwar besteht ein Teil der Zielgruppe für die Software aus technisch interessierten und fähigen Personen, dennoch gibt es auch eine breite Nachfrage im Bereich der Computer-Laien, weshalb die Software auch für diese leicht durchschaubar und selbsterklärend sein soll. 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 herunterbrechen 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 (Graphklasse) 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 ist unabhängig von unserem System.

- **Wartbarkeit**

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