|  |  |
| --- | --- |
| Versions Nr.: | 1.0 |
| Autor: | John Truong  Daniel Xander |
| Date Last Revision: | 06.01.2011 |
| Dokument State: | Final |
| Projektnummer: | SP3S201002 |
|  |  |



|  |
| --- |
| **Newtonsche Axiom** |
| Software Requirements Specification |

Biomedizinische Informatik

Revision

| Version | Date | Comment | Autor |
| --- | --- | --- | --- |
| 0.1 | 05.12.10 | Funktionale Anforderungen mit Priorisieren, nicht-Funktionale Anforderungen | John Truong, Daniel Xander |
| 0.2 | 12.12.10 | *Vision, Sinn und Zweck des Dokuments, Übersicht* | Daniel Xander |
| 0.3 | 13.12.10 | *Benutzereigenschaft, Aufgaben und Ziele des Benutzers* | Daniel Xander |
| 0.4 | 20.12.10 | *Analyse Modell* | Daniel Xander |
| 0.5 | 04.01.11 | *Klassendiagramm* | John Truong |
| 1.0 | 07.07.10 | *Appendix, Korrektur* | John Truong, Daniel Xander |

Inhaltsverzeichnis

Revision i

Inhaltsverzeichnis ii

1. Einleitung 1

1.1 Sinn und Zweck des Dokuments 1

1.2 Vision 1

1.3 Übersicht 1

2. Allgemeine Beschreibung 2

2.1 Benutzereigenschaften 2

2.2 Aufgaben und Ziele der Benutzer 2

2.3 Annahmen und Abhängigkeiten 2

2.4 Realisation der Anforderungen 2

2.4.1 Risiken 2

2.4.2 Ressourcen 2

2.4.3 Werkzeuge 2

3. Funktionale Anforderungen 3

3.1 Use Cases Überblick 3

3.1.1 Prioritäten 3

3.1.2 Use Case 1.1: Messreihen einlesen (Excel, Labview) 4

3.1.3 Use Case 1.2: Tabellarische Darstellung 4

3.1.4 Use Case 1.3: X-Y Plot 5

3.1.5 Use Case 1.4: Vergleich zweier Messreihen 5

4. Nicht-funktionale Anforderungen 6

4.1 Entwurfsbedingungen 6

4.2 Effizienz 6

4.3 Dokumentation 6

4.4 Schnittstellen 6

5. Analysis Modell 7

5.1 Konzeptuelles Modell (Domänen Modell) 7

5.2 Daten Modell 8

5.2.1 SQLite Definition 8

5.2.2 Datenstruktur während der Laufzeit 9

5.3 System Interaktionen 9

5.3.1 System Sequenz Diagramm für Use Case 1.1: Messreihen einlesen 9

5.3.2 System Sequenz Diagramm für Use Case 1.2: Tabellarische Darstellung 10

5.3.3 System Sequenz Diagramm für Use Case 1.3: XY-Plot 11

5.4 Klassendiagramm 12

Appendix 13

# Einleitung

## Sinn und Zweck des Dokuments

Der eigentliche Zweck dieses Dokuments ist es, allen Anspruchsgruppen ein gemeinsames Verständnis der Problemstellung zu ermöglichen. Die Anforderungen werden genau beschrieben, so dass Entwickler und Benutzer ihre Ziele synchronisieren können. Weiterhin werden organisatorische Rahmenbedingungen gesetzt.

## Vision

Ziel des Projekts ist es ein Programm zu erstellen welches einen übersichtlichen Vergleich von numerischen Messdaten ermöglicht. Das Programm soll es Experimentalgruppen erleichtern ihre Daten auszuwerten. Arbeitsaufwand, welcher bei der Bedienung von komplexeren Plot-Programmen entstehen kann, soll durch die Applikation minimiert werden.

Das Programm soll in der Lage sein Daten, die manuell oder automatisch generiert wurden, einzulesen. Nachdem Einlesen soll es möglich sein diese grafisch mittels eines zweidimensionalen Diagramms darzustellen. Die Applikation soll in der Lage sein einzelne Messreihen farblich zu unterscheiden. Das Programm wird in Python entwickelt, was zu einer breiten Nutzbarkeit auf verschiedenen Betriebssystemen führt. Das Endprodukt soll sich von vorhandenen Technologien vor allem durch einfache Nutzbarkeit ohne tieferes Programmverständnis auszeichnen.

## Übersicht

Nachdem im ersten Kapitel die grundlegende Idee kurz erläutert wurde folgt in Kapitel 2 eine allgemeine Beschreibung der definierten Grundaufgaben und Randbedingungen des Projekts. Daraufhin werden in Kapitel 3 funktionale Anforderungen besprochen und in Form von Use Cases dargestellt. Nach dieser Darstellung wird in Kapitel 4 auf nicht-funktionale Anforderungen eingegangen. Abschliessend werden im Analyse Modell (Kapitel 5) verschiedene System-Interaktionen aufgezeigt und Grundstrukturen des Ablaufs und der Daten beschrieben.

# Allgemeine Beschreibung

## Benutzereigenschaften

* **Zugehörige Medizinaltechnik Gruppe:** Gruppe will allgemein möglichst wenig Zeit mit dem erlernen und nutzen des Programms verschwenden, und schnell Ergebnisse einsehen können. GUI soll auf ihre Ergebnisse zugeschnitten sein und möglichst viel Metainformation neben dem eigentlichen Graph zu ihren Versuchen darstellen können.
* **Andere Medizinaltechnik Gruppen:** Programm soll in der Lage sein ihre Daten vollständig zu plotten. In Bezug auf Darstellung von Metainformation können hier Einschränkungen gemacht werden.

## Aufgaben und Ziele der Benutzer

Die Benutzer haben die Aufgaben durch mehrere experimentelle Versuche die Newton’sche Axiome zu studieren. Sie führen Messungen manuell, semiautomatisch und vollautomatisch durch. Ihr Ziel ist es nun, neben der allgemeinen Dokumentation ihrer Messungen und resultierenden Ergebnissen, ihre Versuchsergebnisse durch einen grafischen Plot interpretierbar zu machen.

## Annahmen und Abhängigkeiten

Das Programm wird auf dem Betriebssystem Windows 7 entwickelt. Später folgen Tests auf Mac OSX Snow Leopard sowie Linux Ubuntu 10.10. Am Ende sollte das Programm auf allen genannten Systemen uneingeschränkt lauffähig sein. Andere Systemarchitekturen können leider in der Testphase nicht berücksichtigt werden.

## Realisation der Anforderungen

### Risiken

* Geringe Erfahrung in der Programmierung mit Python erschwert Planung
* Potentielle Fehlerquellen bei Betrieb auf verschiedenen Betriebssystemen.
* Mögliche Schwierigkeiten bei der Kommunikation mit Benutzer sowie mit anderen Entwicklergruppen.

### Ressourcen

Das Kernentwicklerteam besteht aus zwei Programmierern mit etwa gleichwertiger Erfahrung. Gleichzeitig können Informationen mit zwei weiteren Zwei-Mann-Entwicklerteams ausgetauscht werden.

Es stehen täglich ca. 8 Stunden Entwicklungszeit über einen Zeitraum von 4 Woche zur Verfügung. Das macht bei zwei Arbeitskräften eine Entwicklungszeit von 320 Stunden inklusive Dokumentation.

### Werkzeuge

Als Entwicklungsumgebung wird Eclipse Helios verwendet. Programmiersprache ist Python, Version 2.7. Die graphische Oberfläche wird mit der Tkinter Library realisiert. Zur Generierung eines Domänen Modells und Darstellung der Systeminteraktionen wird die Freeware Umlet genutzt.

# Funktionale Anforderungen

## Use Cases Überblick

\*Produktnutzen, \*\*Technische Schwierigkeit, \*\*\*Priorität

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Akteur | Ziel | Hauptablauf | \* | \*\* | \*\*\* |
| 1. Medtech. Gruppen | 1.1 Messreihen einlesen (Excel, Labview) | Das System soll von Excel oder Labview generierte Daten einlesen können. | Hoch | Hoch | Hoch |
| 1.2 Tabellarische Darstellung | System stellt die eingegeben Messreihen tabellarisch dar. | Tief | Mittel | Mittel |
| 1.3 X-Y Plot | System stellt gleichzeitig mehrere Messreihen in einem X-Y Plot dar, und unterscheidet diese farblich. | Hoch | Hoch | Hoch |
| 1.4 Vergleich zweier Messreihen | Es muss möglich sein die Werte zweier Messreihen zu vergleichen und unterschiedlich darzustellen | Hoch | Hoch | Hoch |

### Prioritäten

Die Prioritäten werden bestimmt aus der Kombination von Produktnutzen und technischer Schwierigkeit.

|  |  |
| --- | --- |
| **Priorität** | **Bedeutung** |
| Hoch | Diese Anforderung ist unabdingbar und notwendig für das korrekte Funktionieren der Software; sie muss realisiert werden. |
| Mittel | Diese Anforderung ist nicht unabdingbar ihre Realisierung trägt zur wesentlichen Verbesserung der Software bei. Sie soll wenn möglich realisiert werden. |
| Tief | Diese Anforderung trägt zur Verbesserung der Software bei, ist jedoch nicht unbedingt notwendig. Es wäre aus wünschenswert, wenn die Anforderung realisiert würde. |

### Use Case 1.1: Messreihen einlesen (Excel, Labview)

**Akteur:** Medtech. Gruppen

**Vorbedingungen:** Die Messreihen müssen in kompatiblen Format erstellt sein

**Nachbedingungen (Erfolgsgarantie):** System soll von Excel oder Labview generierte Daten entgegennehmen, validieren und präsentieren.

**Trigger:** Benutzer will die erstellten Messreihen mit dem System darstellen.

**Hauptablauf:**

1. Benutzer startet das System
2. Benutzer sucht via Benutzeroberfläche bestimmte Excel oder Labview Dateiformate auf der Festplatte und kann diese hochladen.
3. System validiert die hochgeladenen Daten.
4. System speichert die Daten programmintern und präsentiert diese tabellarisch.

**Erweiterung (Alternativen, Varianten):**

2.a Hochgeladene Datei wird nicht erkannt

1. System verlangt einen erneuten Versuch und macht einen Hinweis auf das Dateiformat.

3.a Daten können nicht validiert werden.

1. System versteht die Daten in der Datei nicht und macht einen Hinweis.

### Use Case 1.2: Tabellarische Darstellung

**Akteur:** Medtech. Gruppen

**Vorbedingungen:** Messreihen sind erfolgreich validiert worden.

**Nachbedingungen (Erfolgsgarantie):** Messreihen werden tabellarisch dargestellt.

**Hauptablauf:**

1. Nach erfolgreicher Validierung der Messreihen werden diese automatisch in einer Tabelle dargestellt.

**Erweiterung (Alternativen, Varianten):**

1.a Tabelle wird nicht angezeigt.

1.b Tabelle wird nicht richtig dargestellt.

1.c Tabelle zeigt keine Werte an.

### Use Case 1.3: X-Y Plot

**Akteur:** Medtech. Gruppen

**Vorbedingungen:** Messreihen sind erfolgreich validiert worden und werden tabellarisch dargestellt.

**Nachbedingungen (Erfolgsgarantie):** Messreihe wird im X-Y Plot dargestellt

**Hauptablauf:**

1. Messreihe ist tabellarisch vorhanden und kann mit Bestätigung automatisch im X-Y Plot dargestellt werden.

**Erweiterung (Alternativen, Varianten):**

1.a X-Y Plot wird nicht angezeigt

1.b X-Y Plot wird nicht richtig angezeigt.

1.c X-Y Plot zeigt keine Werte an.

### Use Case 1.4: Vergleich zweier Messreihen

**Akteur:** Medtech. Gruppen

**Vorbedingungen:** Messreihen sind erfolgreich eingelesen worden.

**Nachbedingungen (Erfolgsgarantie):** Zwei Messreihen sind verglichen und Unterschiede sind dargestellt.

**Hauptablauf:**

1. Messreihen sind tabellarisch vorhanden und können mit Bestätigung im X-Y Plot dargestellt werden.
2. Nach Auswahl von mindestens zwei Messreihen werden diese mit unterschiedlichen Farben dargestellt.

**Erweiterung (Alternativen, Varianten):**

1.a Gleichzeitige angezeigte Messreihen sind beschränkt

1. Hinweis, dass keine neue Messreihe mehr im Plot angezeigt werden kann.

# Nicht-funktionale Anforderungen

## Entwurfsbedingungen

* Es soll ausschliesslich die Tkinter Library zur Gestaltung der grafischen Oberfläche genutzt werden.
* Das Programm muss auf den Betriebssystemen Mac OX und Windows lauffähig sein.

## Effizienz

Das Programm soll auch bei grösseren Datenmengen performant sein und nicht unnötig Prozessorleistung nehmen. Auch soll die Applikation aus Benutzersicht einfach und schnell zu bedienen sein, so dass die Auswertung möglichst simultan zur Messung geschehen kann.

## Dokumentation

Die Source soll ausreichend Dokumentiert sein, so dass mögliche Modifikationen und Zusatzmodule von Dritten einfach eingebunden werden können.

## Schnittstellen

* **Benutzerschnittstellen:**

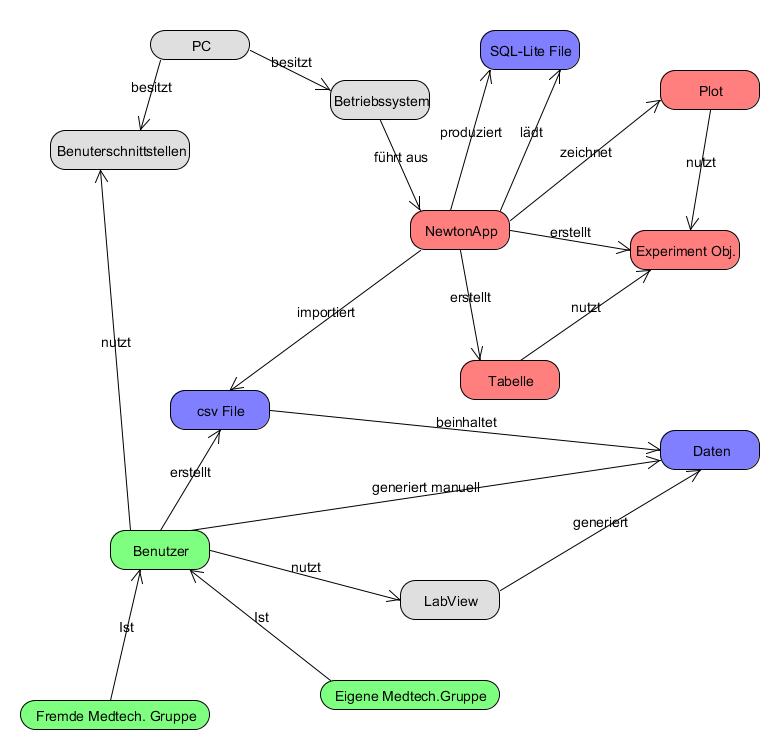
Eine graphische Benutzeroberfläche erlaubt es dem Benutzer die Operationen (einlesen, speichern, plotten, vergleichen) intuitiv zu bedienen. Diese Oberfläche muss frei skalierbar sein, dass heisst, sie muss sich bei Änderung der Fenstergrösse anpassen.

* **Softwareschnittstellen:**

Es besteht die Möglichkeit Importer-Module der anderen Informatik Gruppen in unser Programm einzubinden. Nach deren Einbindung ist es möglich die Daten sämtlicher Medtech. Gruppen einzulesen und auszuwerten.

# Analysis Modell

## Konzeptuelles Modell (Domänen Modell)



## Daten Modell

### SQLite Definition

Bemerkung:

Die SQLite Definition wurde von Michel Heininger und Tobias Thüring erstellt.

**Tabelle Messwerte**

Tabellenname: “**values“**   
Datentyp Spalte “t”: FLOAT  
Datentyp Spalte/n „vx“: FLOAT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| t | v1 | v2 | v3 | v... |
| 0,1 | 1,2 |  |  |  |
| 0,2 | 1,3 |  |  |  |

**Tabelle Metadaten**

Tabellenname: **“metadata“**  
Datentyp Spalte “name”: TEXT  
Datentyp Spalte “value”: TEXT

|  |  |
| --- | --- |
| name | value |
| date | Format: „YYYYMMDDmmss“  Ex. „201012011640“ = 1. Dez. 2010, 16:40 Uhr |
| exp\_name | Name des Experiments |
| actor\_name | Namen der durchführenden Person  Ex. „Hans, Rita, Petra“ |
| series\_nr | Ex. 1 |
| v*n*\_unit | Einheit der Werte v*x*  *Ex. „m/s^2“* |
| v*n*\_desc | Art der Werte v*x*  *Ex. „Geschwindigkeit“* |
| v*n*\_fault | Fehlertoleranz mit relativer Abweichung/Fehler[1](" \l "sdfootnote1sym) der Werte v*x* in Prozent  *Ex. „10“ = 10 % oder „0.2“ = 0.2 %* |
| additional\_info | Weitere Metadaten in Textform |

### Datenstruktur während der Laufzeit

Über die data\_access Klasse können SQLite Daten in das Programm geladen werden. Die Messwerte werden in ein 2-dimensionales Array gespeichert.

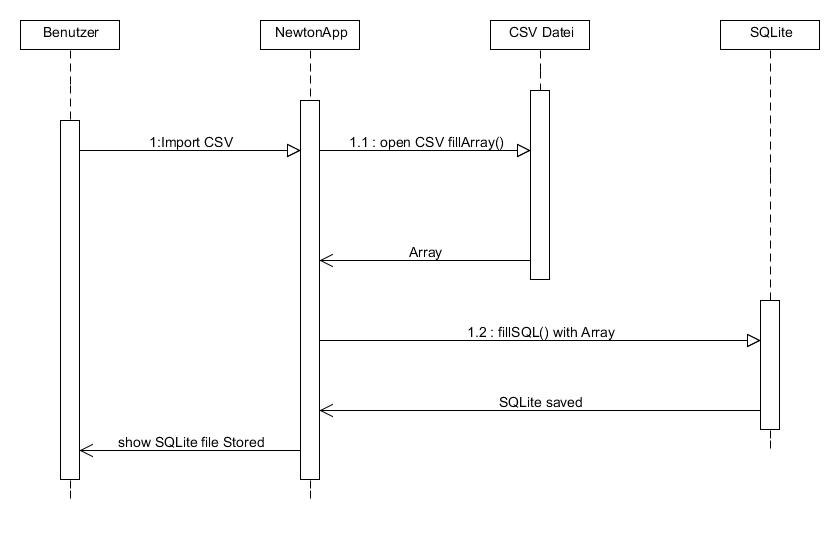
Array Messwerte : [(t,v1,v2,……),(t,v1,v2,……),……]

Die Metainformationen werden in einem Dictionary Objekt gespeichert. Dies ermöglicht einen schnellen Zugriff auf benötigte Metawerte über Keys.

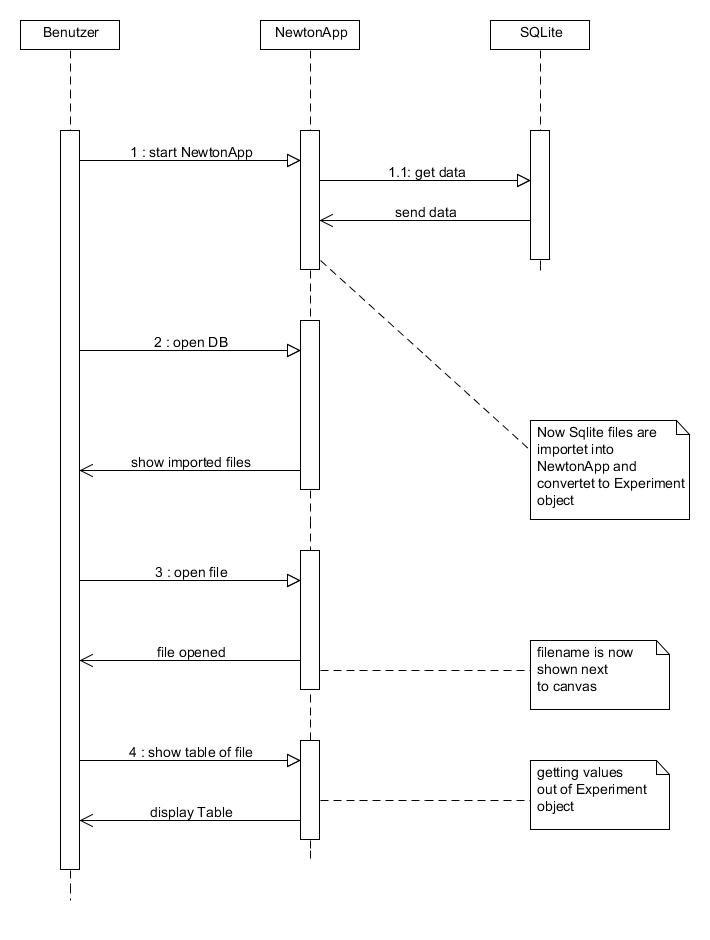
Meta-Dictionary: {“name“ : “value”,…..}

## System Interaktionen

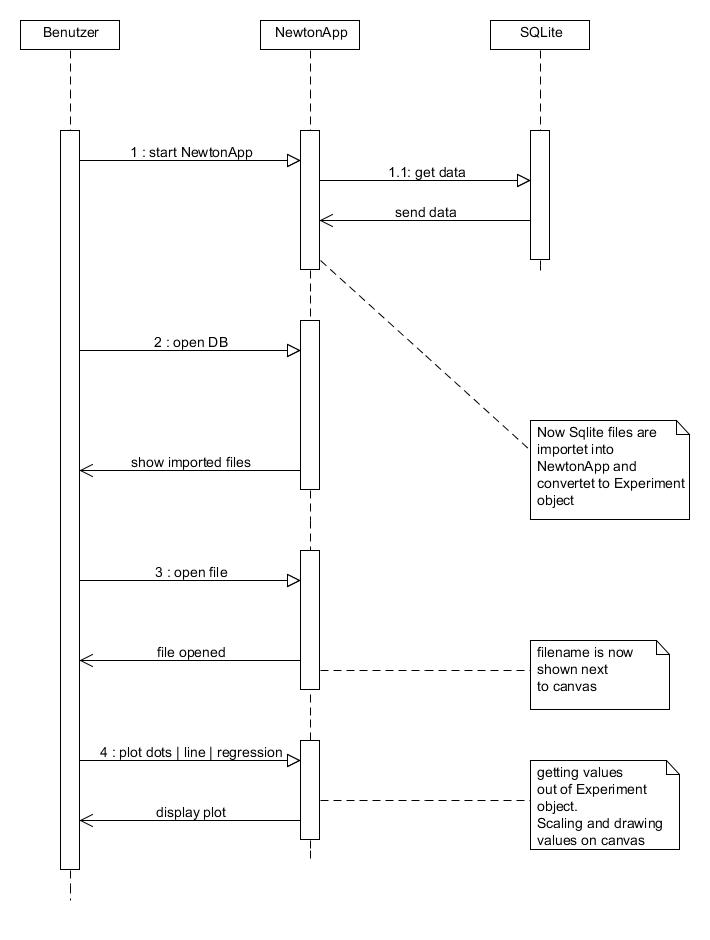
### System Sequenz Diagramm für Use Case 1.1: Messreihen einlesen



### System Sequenz Diagramm für Use Case 1.2: Tabellarische Darstellung



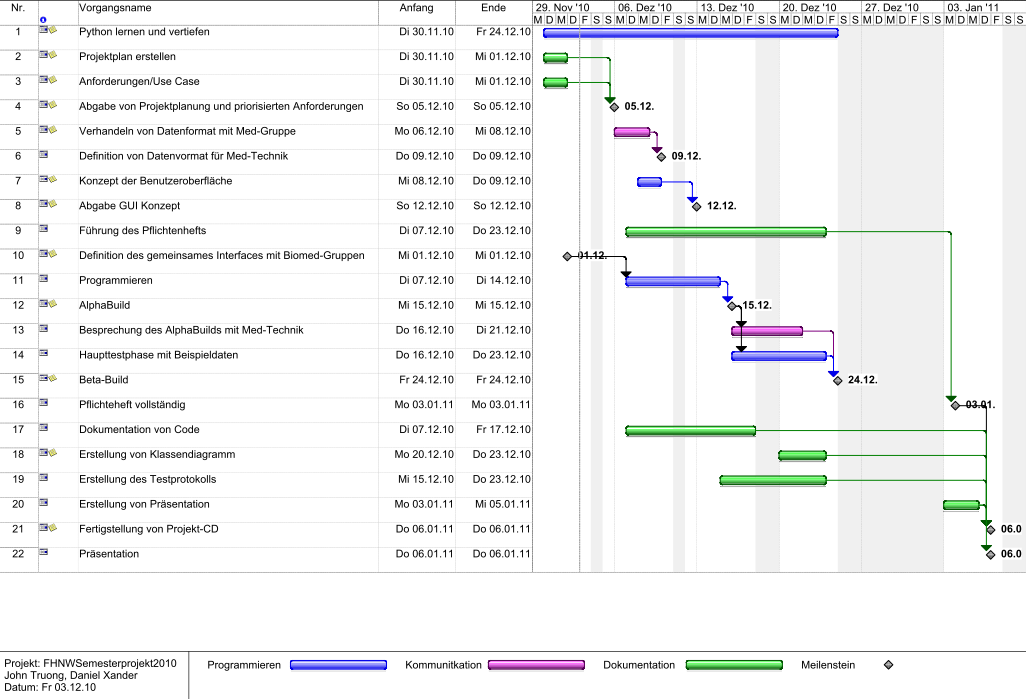
### System Sequenz Diagramm für Use Case 1.3: XY-Plot



Use Case 1.4, also der Vergleich von Messreihen, entspricht grob dem von 1.3. Es werden einfach mehrere Datensätze nacheinander geöffnet und dann gezeichnet.

## Klassendiagrammklassendiagramm.PNGuml.PNG

Appendix

**Projektplan:**