

# SPRINT 3

Sprintdokumentation

14.04.-28.04.2016

Slider 2

Otahal, Holecek, Jevtic,

## Inhaltsverzeichnis

<b>1</b>	<b>Sprintbericht.....</b>	<b>3</b>
1.1	Tasks.....	3
<b>2</b>	<b>GUI mittels Python programmieren .....</b>	<b>5</b>
2.1	Prototyp .....	6
<b>3</b>	<b>SQLite Datenbank .....</b>	<b>5</b>
3.1	Warum SQLite.....	5
3.2	Installation.....	5
3.3	Erstellen einer SQLite3 Datenbank .....	5
<b>4</b>	<b>Überarbeitung eines Webinterfaces.....</b>	<b>11</b>
4.1	Einleitung.....	11
<b>5</b>	<b>Burndown Charts .....</b>	<b>13</b>
5.1	Sprint Burndown Chart .....	13
5.2	Product Burndown Chart.....	13

## 1 Sprintbericht

### 1.1 Tasks

#### 1.1.1 Von Sprint übernommen

ID	Name	Zugewiesen	Akzeptanzkriterium	anfängl. Schätz.	verstrichen	verbleibend	Status
1335	GUI mittels Python programmieren	MO	GUI muss funktionstüchtig sein	20 hrs			Nicht fertiggestellt
1283	Prototyp von Grafischer Live Darstellung	RH	GUI soll mittels den Testdaten angezeigt werden	6 hrs			fertiggestellt
1273	Datenbank am Beaglebone aufsetzen	BJ	Zugriff auf Server möglich	6 hrs			Nicht fertiggestellt
1333	Überarbeitung des Webinterfaces	BJ	Webinterface wird angezeigt	4 hrs			fertiggestellt

## 1.1.2 Neu hinzugefügt

ID	Name	Zugewiesen	Akzeptanzkriterium	anfängl. Schätz.	verstri- chen	verblei- bend	Status
1271	Serielle Kommunikation herstellen	MO	Testdaten				Nicht fertiggestellt

## 2 SQLite Datenbank

### 2.1 Warum SQLite

Unser Projektteam hat sich für das DBMS SQLite wegen den folgenden Punkten entschieden:

- SQLite benötigt viel weniger Ressourcen als andere herkömmliche DBMS. Dies ist wichtig, da der Beagle Bone nicht viel Rechenleistung zur Verfügung stellen kann.
- Es ist File-basiert (die ganze Datenbank besteht aus 1. File) und ist damit sehr Portable.
- Es ist zum Testen von Daten geeignet, da es nicht zusätzliche Prozesse für diesen Vorgang benötigt.

### 2.2 Installation

In unserem Fall wird die SQLite Version 3 verwendet. Um den Download der .tar.gz Datei zu initialisieren, wird der Befehl

```
wget http://www.sqlite.org/sqlite-autoconf-3070603.tar.gz
```

verwendet. Darauf folgend wird die jetzt heruntergeladene Datei mit dem tar-Befehl

```
tar xvfz sqlite-autoconf-3070603.tar.gz
```

die Datei entpackt. Danach wird in eines der extrahierten Verzeichnisse mit dem Befehl `cd sqlite-autoconf-3070603` gewechselt. Hier befindet sich dann die Datei, die mit dem `./configure` Befehl in ein `make`-fähiges gemacht wird. Mit den Befehlen `make` und darauf folgend `make install` wird dann SQLite3 installiert.

Nachdem die `make` Befehle ausgeführt werden, erscheint ein Output, der besagt, dass die SQLite3 Binaries im Verzeichnis `/usr/local/bin` installiert werden.

### 2.3 Erstellen einer SQLite3 Datenbank

Für das Projekt wird eine sehr simple Datenbank erstellt, die die Peak-Werte der Differenz der beiden Laser speichert. Die Datenbank wird mit dem Befehl ***sqlite3 beaglebone*** erstellt.

Es wird dann ein Table mit dem Namen ***PEAK*** mit dem Datentypen ***INTEGER*** mit dem Befehl ***create table PEAK(value INTEGER)*** erstellt.

Die Tabelle wird dann den folgenden Testdaten gefüllt: ***Insert into PEAK value(5)***

## 3 GUI mittels Python programmieren

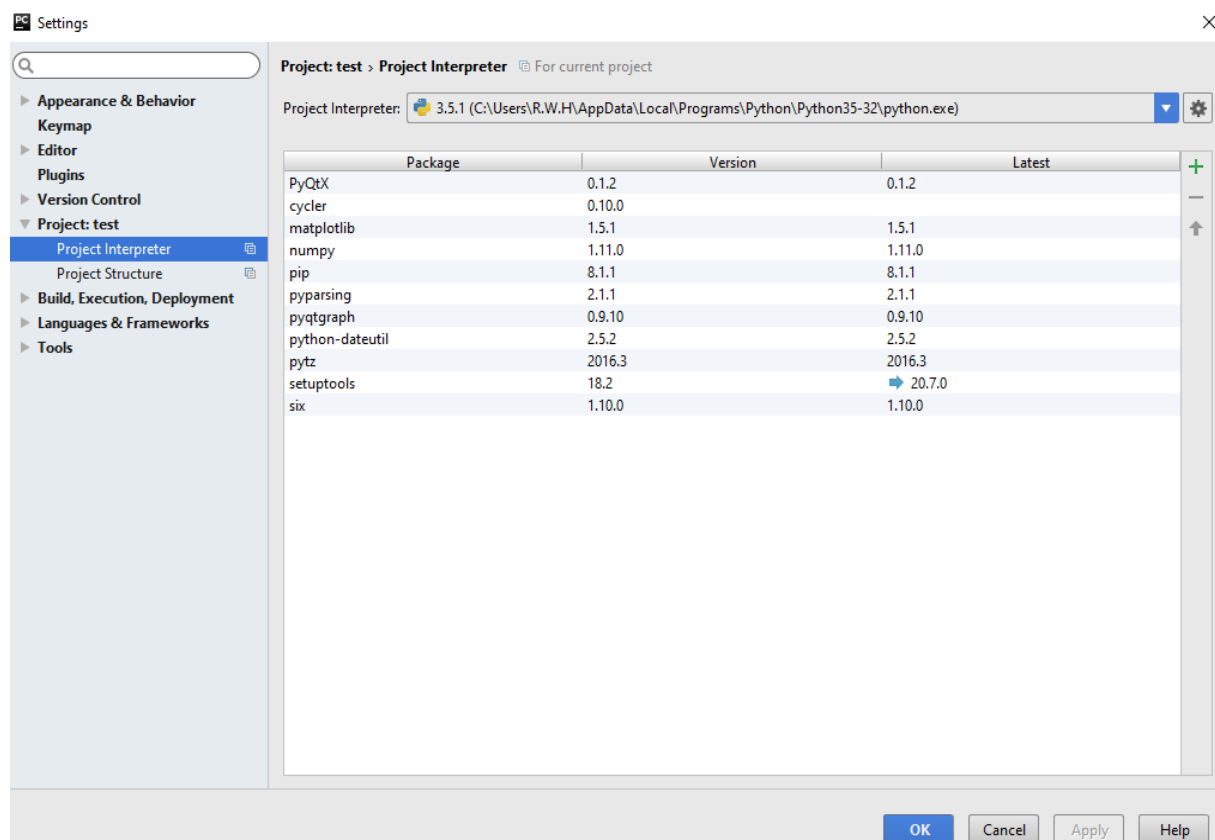
### 3.1 Prototyp

Der erste Gedanke der zur Konzipierung der GUI einfiel war sie in python(Pycharm) selber zu realisieren. Dies wurde mit den Modulen Tkinter (GUI) und Matplotlib(Graph) ermöglicht. Als IDE wurde eine Windows x64 version von Netbeans Pycharm verwendet.

#### 3.1.1 Installieren der Module

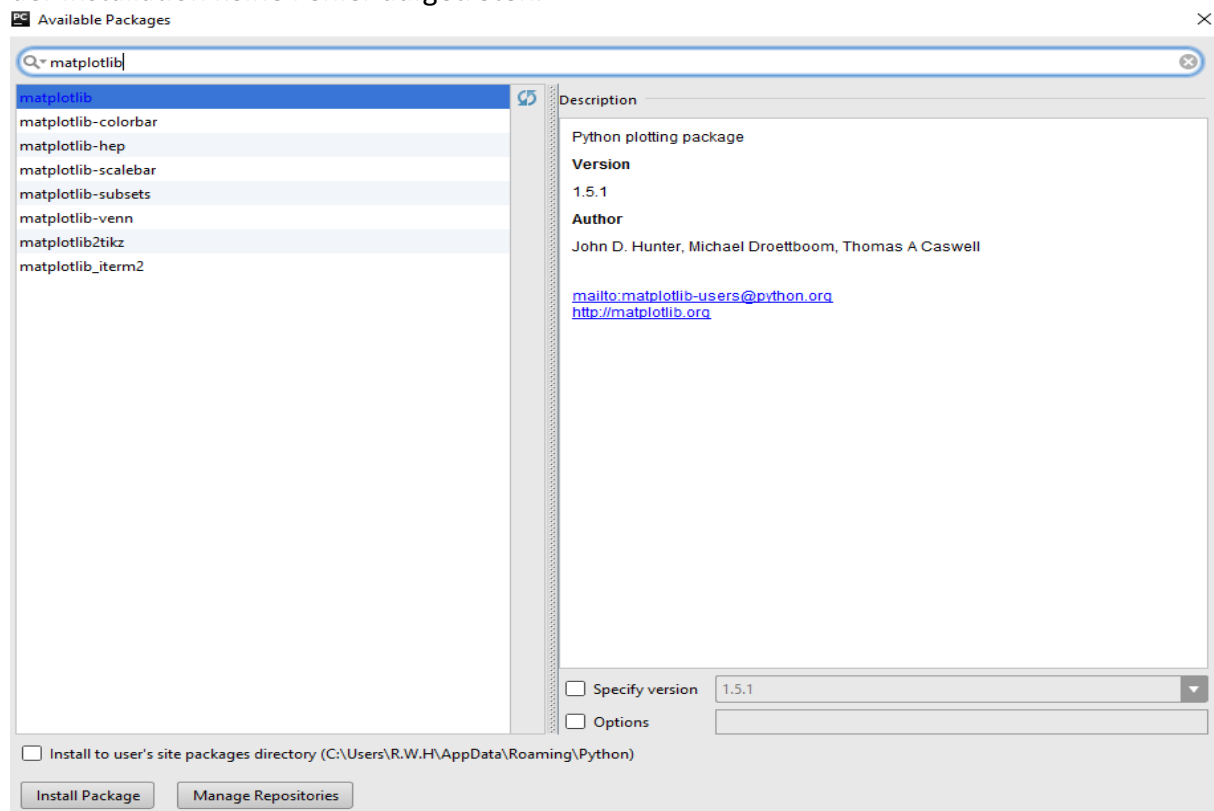
Tkinter ist bei Python 2.7 per default dabei und wird mit den Befehl *from tkinter import \** in eine Klasse importiert. Wenn man nicht den Konstruktor von Tkinter direkt aufrufen will wenn man ihm eine Variable zuweist, kann man den Befehl *import tkinter as varname* verwenden.

Das Modul Matplotlib hingegen muss mittels pip noch extra installiert werden. In Pycharm wird die installation über pip mit einer GUI erleichtert. Diese wird in *File -> Settings -> Project -> Project Interpreter* gefunden. Zu der Suche der verschiedenen Module gelangt man über das Betätigen des *+ Buttons*.



Nach der Betätigung erfolgt die Öffnung des folgenden Fensters.

In diesem können mittels des Suchfeldes alle möglichen Module gefunden und mit betätigung des *Install Package* Buttons dann installiert werden. In unserem fall sind während der Installation keine Fehler aufgetreten.



Danach kann matplotlib in einer beliebigen Klasse mit dem Befehl *import matplotlib* verwendet werden.

### 3.1.2 Matplotlib Probeklasse

In der folgenden gut dokumentierten Klasse wird gezeigt wie man ein einfaches Programm mittels matplotlib erstellt:

```
import matplotlib.pyplot as plt
# a simple plot with the module matplotlib

# The figure module contains all the plot elements
fig = plt.figure()
# signature for subplot is: numrows (1), numcols(1), plot number(1)
a = fig.add_subplot(111)

# lists of x and y values to plot
listx = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
listy = [0, 1, 2.3, 4.1, 6.2, 8.3, 10.1, 9.4, 11, 12]

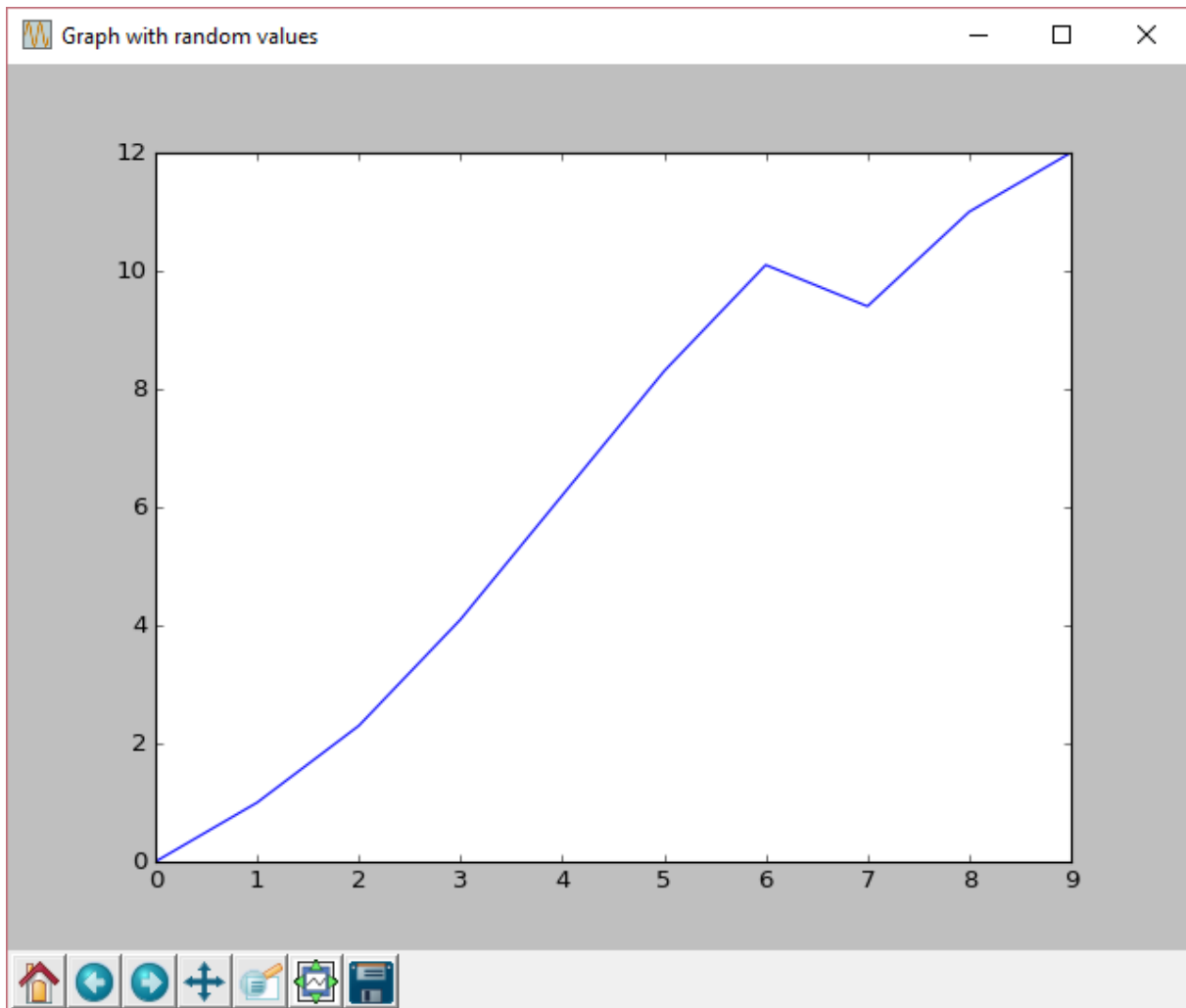
# just to make sure the program is running
print("test")

# plots the 2 lists of values it is given
a.plot(listx, listy)

# makes the graph visible
plt.show()
```

Die ausgegebene GUI sieht wie folgt aus:

Die Buttons im unteren Teil des Fensters sind Standardmäßig bei einem Plot dabei.



### 3.1.3 Embedden in Tkinter

Die nächste Aufgabe war die veraltet aussehenden Buttons mit selbst erstellten zu ersetzen.

Nach einer Recherche hat sich die folgende Lösung ergeben:



```

import matplotlib
import sys
from numpy import arange, sin, pi
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg, NavigationToolbar2TkAgg
from matplotlib.figure import Figure

matplotlib.use('TkAgg')

if sys.version_info[0] < 3:
    import Tkinter as Tk
else:
    import tkinter as Tk

root = Tk.Tk()
root.wm_title("Embedding in TK")
root.resizable(width=False, height=False)

root.geometry('{}x{}'.format(500, 600))
f = Figure(figsize=(5, 4), dpi=100, facecolor='lightgrey')
a = f.add_subplot(111)
t = arange(0.0, 3.0, 0.01)
s = sin(2 * pi * t)

a.set_axis_bgcolor('lightgreen')
listx = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
listy = [0, 1, 2.3, 4.1, 6.2, 8.3, 10.1, 9.4, 11, 12]

print("tests")
a.plot(listx, listy)

canvas = FigureCanvasTkAgg(f, master=root)
canvas.show()
canvas.get_tk_widget().pack(side=Tk.TOP, fill=Tk.BOTH, expand=1)

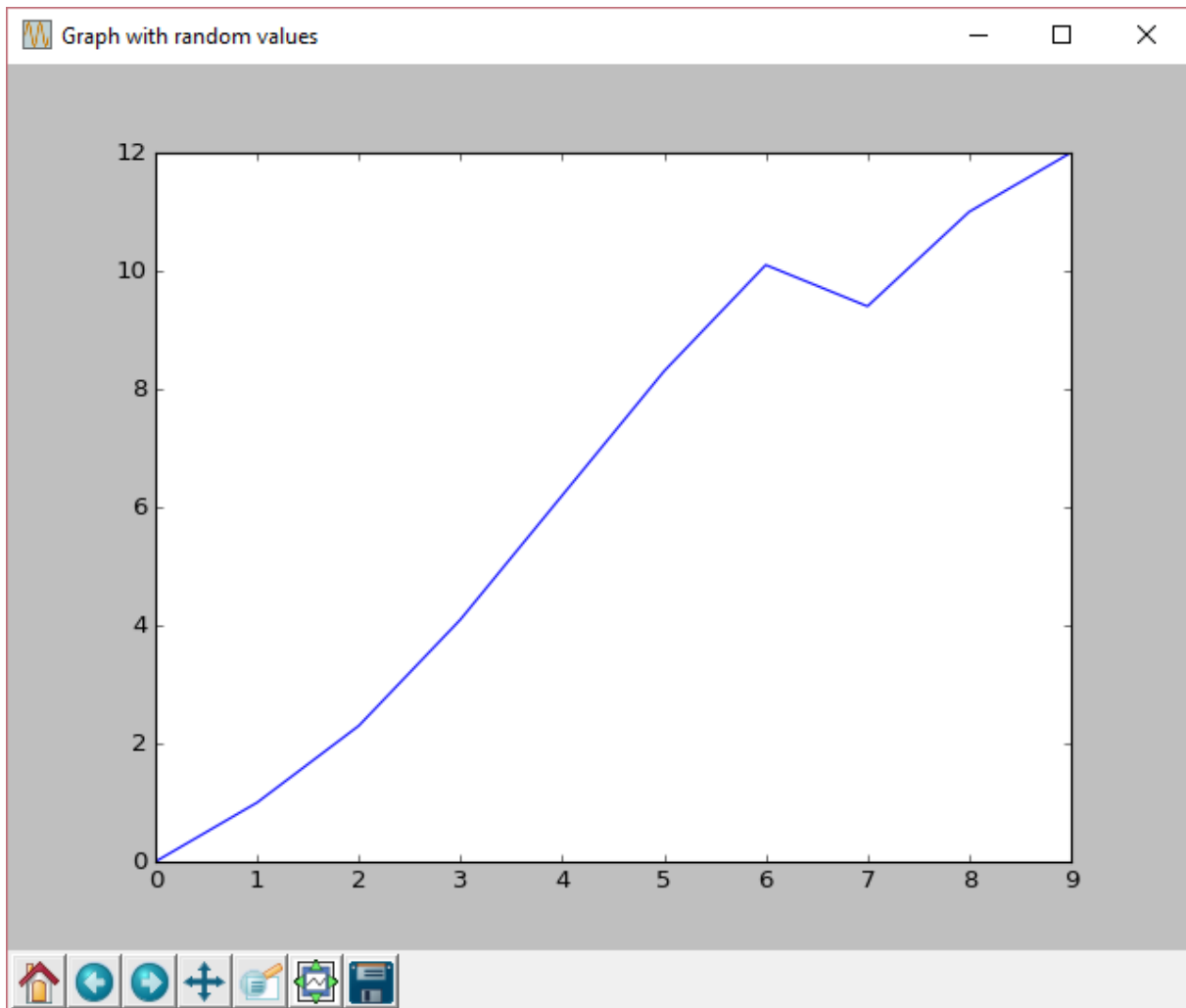
def _quit():
    root.quit() # stops mainloop
    root.destroy() # this is necessary on Windows to prevent
                  # fatal python error: PyEval_RestoreThread: NULL tstate

button = Tk.Button(master=root, text='Quit', command=_quit)
label = Tk.Label(root, text="Spannungsverlauf")
button.config(height=2, width=15)
button.config(bg="grey")
root.configure(background='lightgrey')
button.pack(side=Tk.BOTTOM)

Tk.mainloop()

```

Die Ausgabe dieses Codes sieht wie gefolgt aus:



Die Buttons im unteren Teil wurden durch einen Button des Moduls Tkinter ersetzt.  
Eigentlich wird gehört der ganze südliche Teil zu Tkinter.

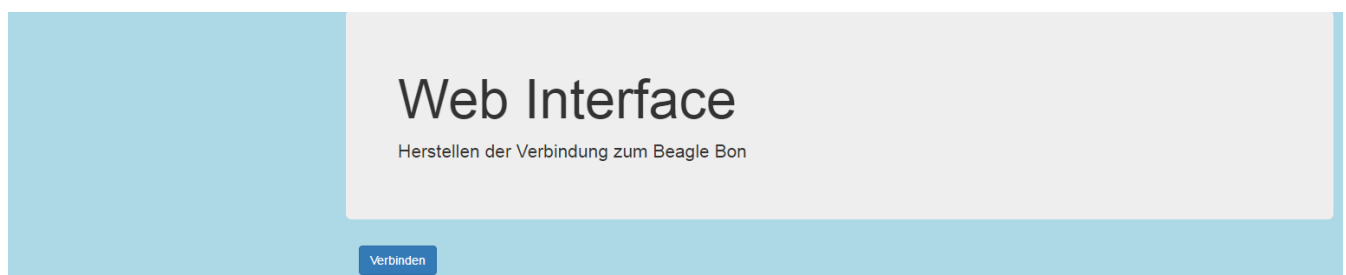
## 4 Überarbeitung eines Webinterfaces

### 4.1 Einleitung

Das Webinterface muss noch entsprechend der Benutzerfreundlichkeit überarbeitet werden. Dabei wird das Webdesign verbessert und verschönert. Jedoch ist die Überarbeitung der Webschnittstelle noch nicht vollständig abgeschlossen, weil die Dropdown Liste (dort, wo die einzelne Laser ausgewählt werden sollen) noch bearbeitet werden muss. Das Webinterface wird noch entsprechend angepasst, um auch wirklich die Verbindung mit dem Beagle Bon herzustellen.

Beispielsweise wurden die Überschriften, Unterüberschriften und Buttons verschönert. Dafür wurde das freie CSS Framework Bootstrap für die Programmierung verwendet, um das Design etwas mehr zu verbessern. Die Hintergrundfarbe ist jedoch gleich geblieben.

connection.html



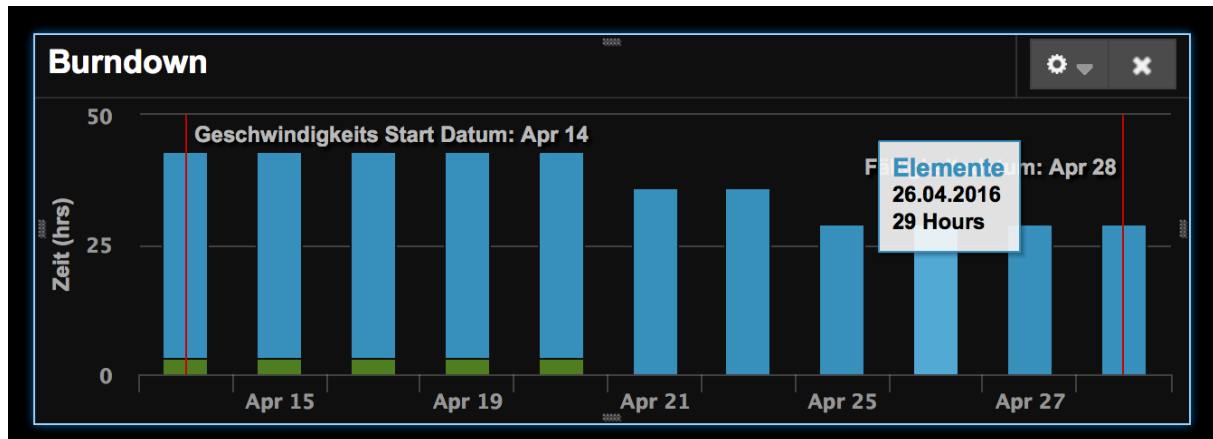
webinterface.html

The screenshot shows a web interface with a light blue background. At the top, there is a white rectangular box containing the title "Web Interface" in a large, bold, black font, and below it, the subtitle "Simulation der Laserkanonen" in a smaller, regular black font. Below this box, there are two blue buttons with white text: "Start!" and "Stop!". In the center of the page, there is another white rectangular box containing the text "Datenbank beaglebon" in a large, bold, black font. At the bottom of the interface, there is a blue button with white text that says "Wählen Sie einen Laser aus!". Below this button is a dropdown menu with a white background and a blue border, showing the text "Wählen Sie einen Laser aus" and a small downward arrow. Below the dropdown menu, there is a line of small black text that reads "Hier werden die Informationen der einzelnen Lasern angezeigt."

Als eines der weiteren Verbesserungsvorschläge wäre noch, ein schönes Template zu finden und dementsprechend anzupassen.

## 5 Burndown Charts

### 5.1 Sprint Burndown Chart



### 5.2 Product Burndown Chart

