MATPLOTLIB: PROJECT DELIVERABLE 5

Team 16 (Do It Tomorrow)

CSCD01

March 31st, 2017

Team Members:
Mengzhe Lu
Jubin Patel
Derek Chow
Daniel Karas
Gauravjeet Kala

Table of Contents

Documentation of Feature

Description:

The feature that our team has implemented is issue 3293: “markerfacecolor / mfc not in rcparams”. In issue 3293 of matplotlib, when creating a line plot, there is currently no way of setting the default marker’s face color. In this case, there does not exist a parameter under the rcParams object such as ‘lines.markerfacecolor’. Therefore for this issue to be resolved, we will need to include a ‘lines.markerfacecolor’ key into our rcParams object, and potentially store the selected default color as the value of this newly created key. However, to fully implement this feature, we will need to adjust and allocate checks to the original rcParams: ‘*linestyle’* to check this newly added rcParams properly.

User Guide:

The default marker face colour of a line plot can be modified directly for example:

```
import matplotlib as mpl
mpl.rcParams['lines.markerfacecolor'] = 'r'
```

Matplotlib also provides a couple of functions for modifying other rc settings. For more details about other functionalities of rcParams, it can be found at matplotlib’s official documentation found on this URL:

matplotlib's rcParams Documentation

Initially the value for rcParams[‘lines.markerfacecolor’] is defaulted to the first color of the color cycle which is normally blue.
See line 893 of lib/matplotlib/rcsetup.py.

The default marker face colour can now be modified by simply changing this value: mpl.rcParams['lines.markerfacecolor'] which can be found in the dictionary-like variable called `matplotlib.rcParams` where all the of the
rc settings are stored.

For example if the user wish to change the default marker face color to red, and incorporate it into their line plot, they would be able to do so without explicitly defining the marker’s face color in their plot:

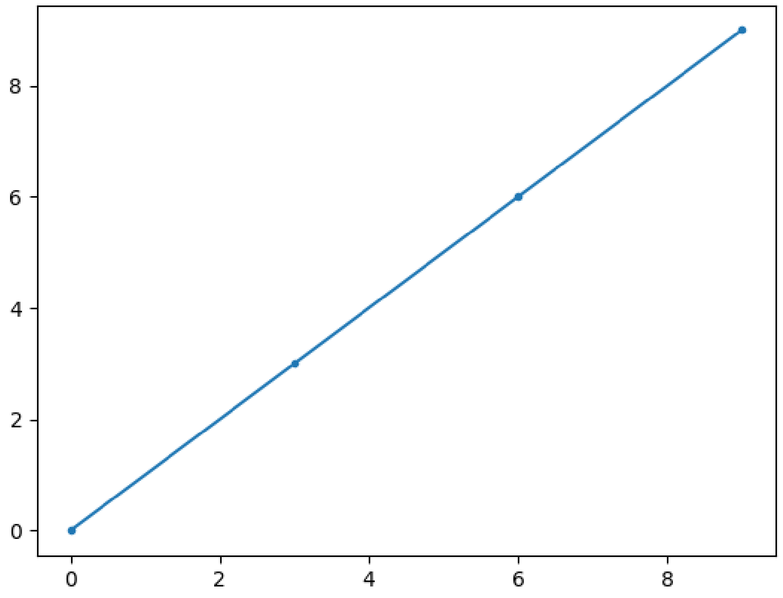
```
import matplotlib as mpl
mpl.rcParams['lines.markerfacecolor'] = 'r'
line_up, = plt.plot(range(10), markevery=3, marker=”o”)
```

Therefore, it is now possible to set default colour for your markers without the user needing to explicitly state it in the code.

Documentation of Code Design

Documentation of Code:

Our feature for issue 3293 was to provide a way to let the user set the default colour of the marker's that are plotted onto a graph.



As you can see the markers or dots at (0, 0), (3, 3), (6, 6) and (9, 9) are blue by default. Before our implementation, the user was unable to set the default colour to something else.

In order to implement this feature, we had to think about interactions between the Artist layers rather than code logic, since code logic was very simple but the interactions were complex. We had to examine various Artist subclasses to check to see if there were any interactions between what we were trying to implement and these classes, and if so we would need to implement the changes.

Our implementation was across few files including ***rcsetup.py*** and ***lines.py***:

The changes were to add the key, 'lines.markerfacecolor' to the mutable map, '*defaultParams'* found on line 893 of the file rcsetup.py. We added the following line to the mutable map:

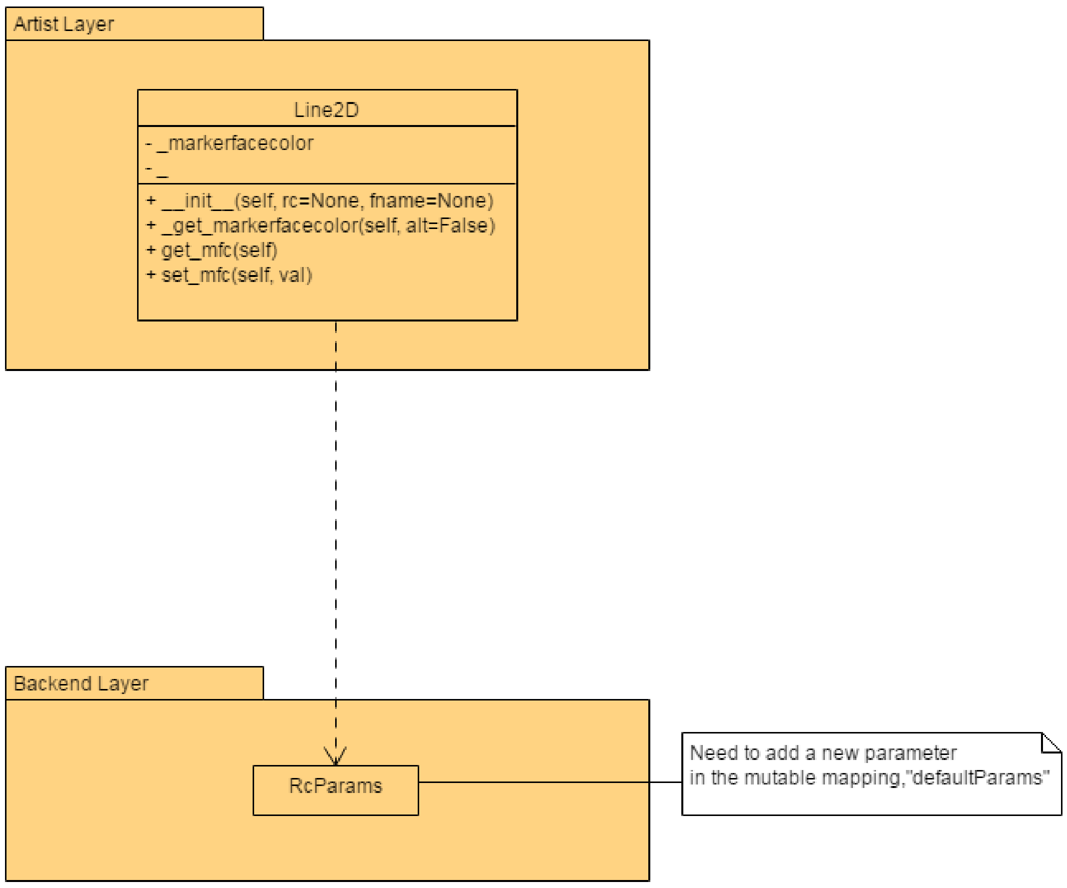
```
    'lines.markerfacecolor': ['C0', validate_color],  # first color in color cycle
```

From this line, we not only will have created a new rcParams but also have it cycle through the first color of the color cycle similar to the rcParams, '*lines.color*'.

The next change we added was to the file *lines.py* where we had to include the logic of whether the user explicitly passes in the marker face color or not. If so, the code will use the explicitly passed marker face color rather than the defaulted one. The changes can be found on lines 342-343.

Organization of Code:

From our previous UML diagram seen in Deliverable 4, we removed the Axes class as it was only using attributes pertaining to linestyle, marker and color. In turn, the code stores these in a tuple and various code found in triplots.py uses this tuple. Therefore, there was no need for our UML Diagram to include the Axes class. Therefore, our newly UML Diagram can be seen below:

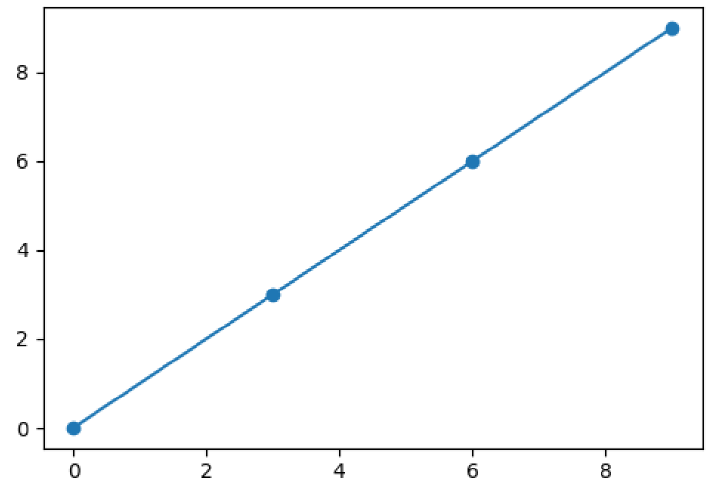

Artist Layer
Line2D
- _markerfacecolor
- _
+ __init__(self, rc=None, fname=None)
+ _get_markerfacecolor(self, alt=False)
+ get_mfc(self)
+ set_mfc(self, val)
Backend Layer
RcParams
Need to add a new parameter in the mutable mapping,"defaultParams"

We added several lines to ‘lines.py’, specifically the __init__ method, which allows the user to set the default marker face colour.

The logic of the code is simple however the implementation interacts with other files which includes both *lines.py* and *rcparams.py*. They interact by rcparams.py first initializing the mutable mapping, ‘defaultParams’, by setting ‘lines.markerfacecolor’ key to the first color of the color cycle. The logic is then applied to *lines.py* such that whenever the user does not specify a marker face color for the line plot, the default marker face color taken from ‘defaultParams’ will be used instead.

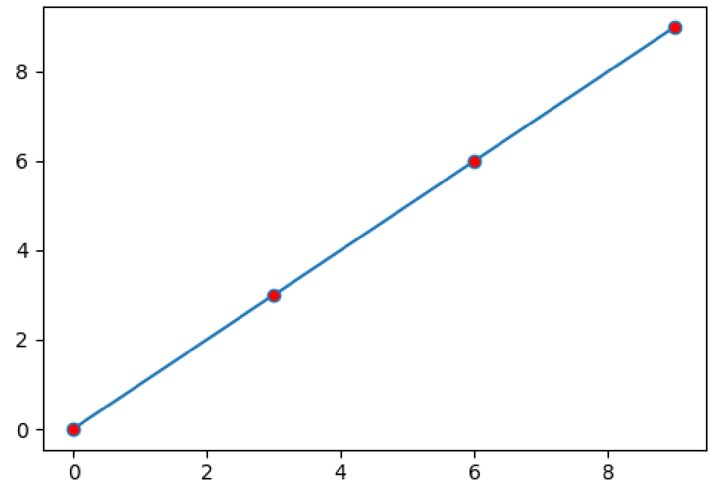
For example, if we were not to set the marker face color of a line graph such as the code below, it will display the following (assuming matplotlib.pyplot is imported as plt):

```
line_up, = plt.plot(range(10), markevery=3, marker=”o”)
```



However, if we add the following to set the default marker face color to red, using the line of code seen below, it’ll generate the following plot (assuming matplotlib is imported as mpl):

```
mpl.rcParams[‘lines.markerfacecolor’] = ‘r’
```



As you can see the difference between the picture above and the previous picture, the dots are now a colour that the user wants and no longer needs to worry about setting them to that colour for each line they are plotting as it will now act as the default.

Testing

Acceptance Testing:

Currently matplotlib doesn’t allow the user to set the default marker face color, and when the user tries to set said parameter, they are encountered with the following error:

Key Error: ‘lines.markerfacecolor is not a valid rc parameter.
See rcParams.keys() for a list of valid parameters.

To recreate the error, the user can type the following python code, assuming they have matplotlib and its necessary dependencies installed.

```
import matplotlib as mpl

mpl.rcParams['lines.markerfacecolor'] = 'r'
```

Furthermore to see if the colours are changing, you can edit the second line where you have ‘r’ which is indication for the colour red or ‘b’ which is for blue.

Alternatively you can also check if the option to set the default marker face color is available through the following method as well:

1 - Run the following script:

```
from matplotlib import rcParams

print(rcParams['lines.markerfacecolor'])
```

2 - It should not give a Key Error or crash. Rather, it will print out the default marker face color (as declared it would be declared in the line properties in lib/matplotlib/rcsetup.py:913).

Lastly, to check if our implementation is working, you can run the following lines:

1 - Run the following script:

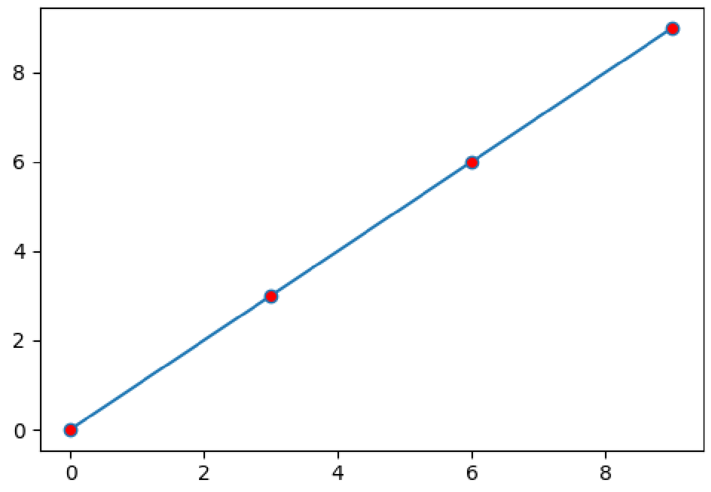
```
import matplotlib.pyplot as plt

from matplotlib import rcParams


rcParams[“lines.markerfacecolor”] = ‘r’

plt.plot(range(10), markevery=3, marker=”o”)
plt.show()
```

2 - It should not give a Key Error or crash. Rather, it will produce the following plot:



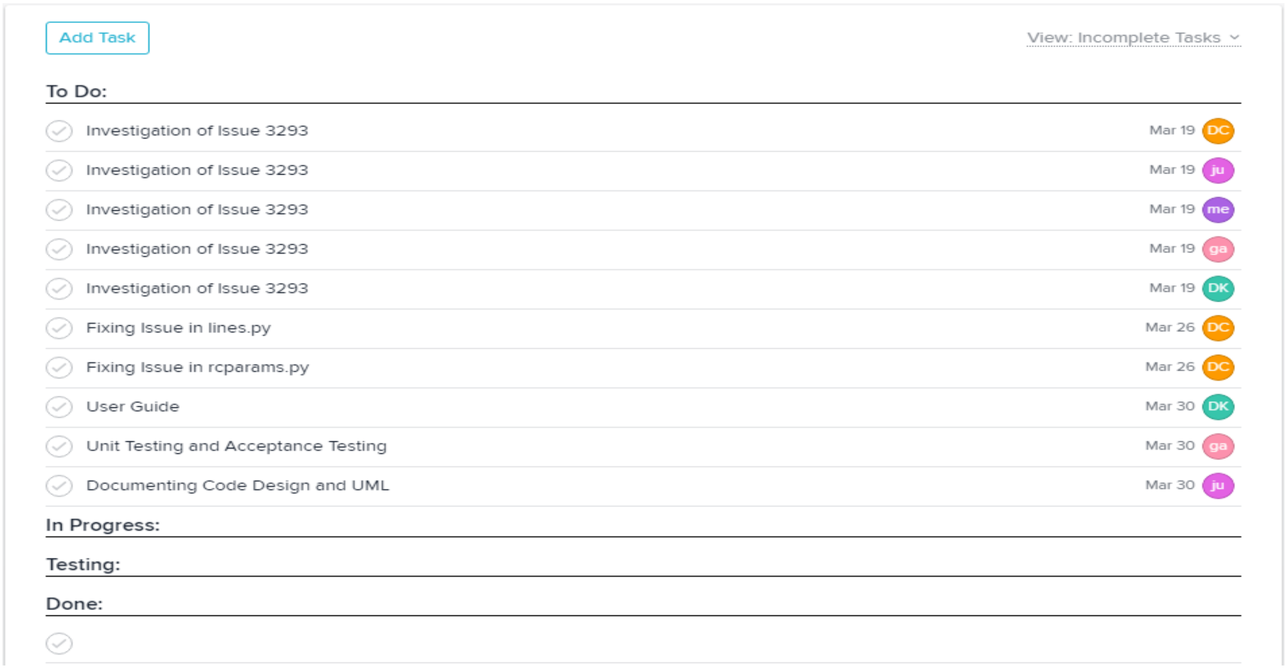
Unit Testing:

If the fix has been implemented, you can run the following unit test suite and make sure it passes all tests to ensure the feature was implemented, as well as running the previous code that raised the error. The unit test suite can be found in the file “test_issue3293.py” under the “Deliverable 5” folder.

Project Management

We used Asana as our task board for our sprint and took screenshots at the beginning, middle and end of our fourteen day sprint:

Beginning of Sprint:



To Do:

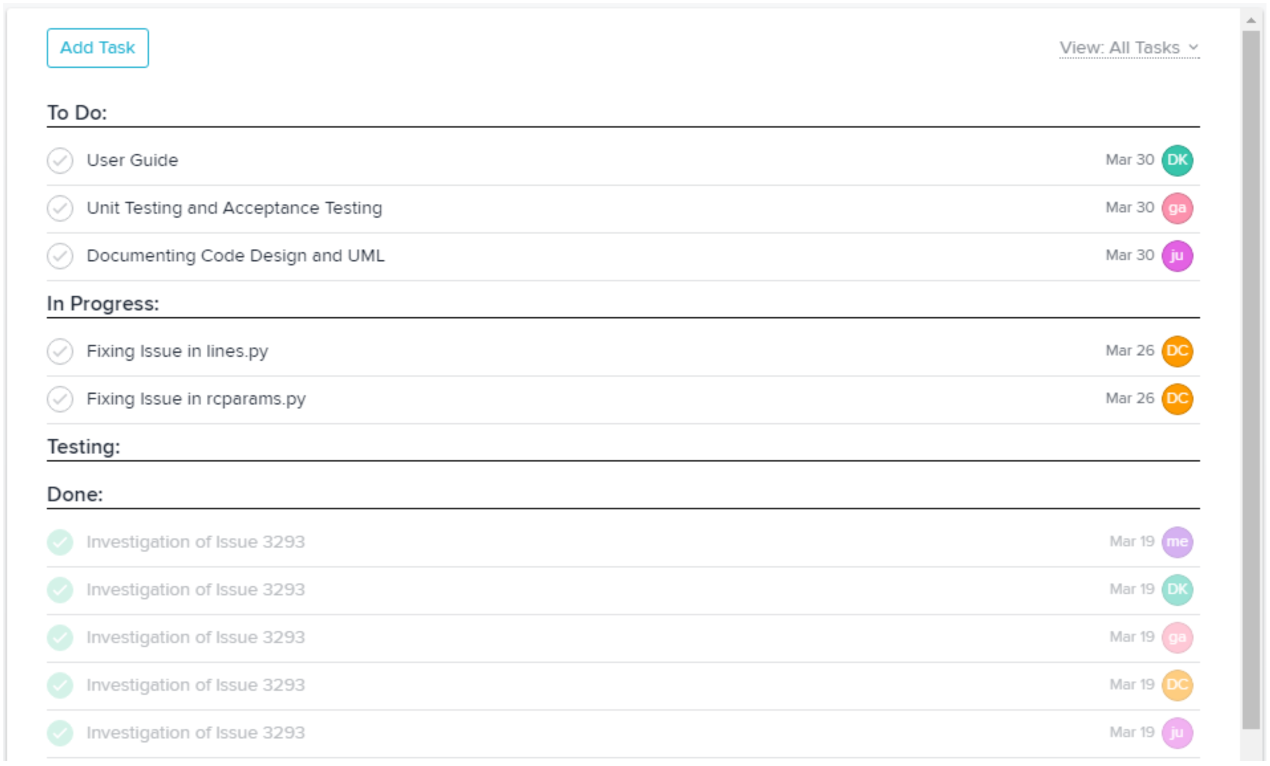
Investigation of Issue 3293	Mar 19	DC
Investigation of Issue 3293	Mar 19	ju
Investigation of Issue 3293	Mar 19	me
Investigation of Issue 3293	Mar 19	ga
Investigation of Issue 3293	Mar 19	DK
Fixing Issue in lines.py	Mar 26	DC
Fixing Issue in rcparams.py	Mar 26	DC
User Guide	Mar 30	DK
Unit Testing and Acceptance Testing	Mar 30	ga
Documenting Code Design and UML	Mar 30	ju

In Progress:

Testing:

Done:

Middle of Sprint



To Do:

User Guide	Mar 30	DK
Unit Testing and Acceptance Testing	Mar 30	ga
Documenting Code Design and UML	Mar 30	ju

In Progress:

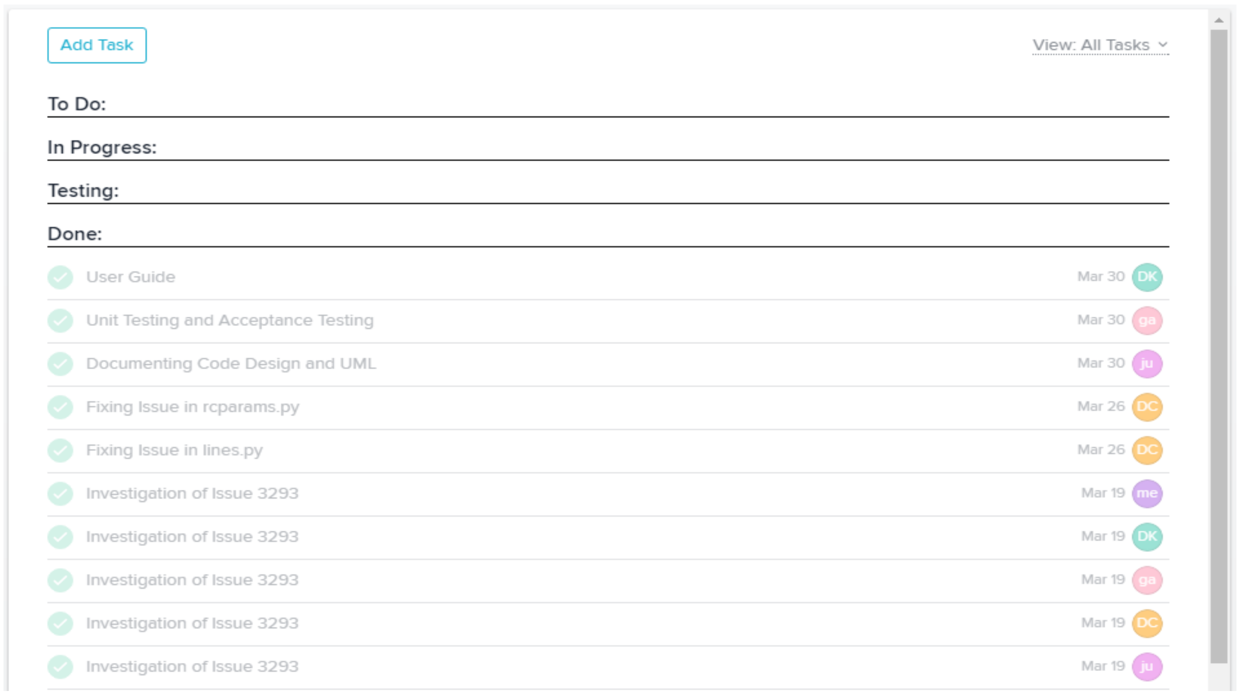
Fixing Issue in lines.py	Mar 26	DC
Fixing Issue in rcparams.py	Mar 26	DC

Testing:

Done:

Investigation of Issue 3293	Mar 19	me
Investigation of Issue 3293	Mar 19	DK
Investigation of Issue 3293	Mar 19	ga
Investigation of Issue 3293	Mar 19	DC
Investigation of Issue 3293	Mar 19	ju

End of Sprint:



To Do:

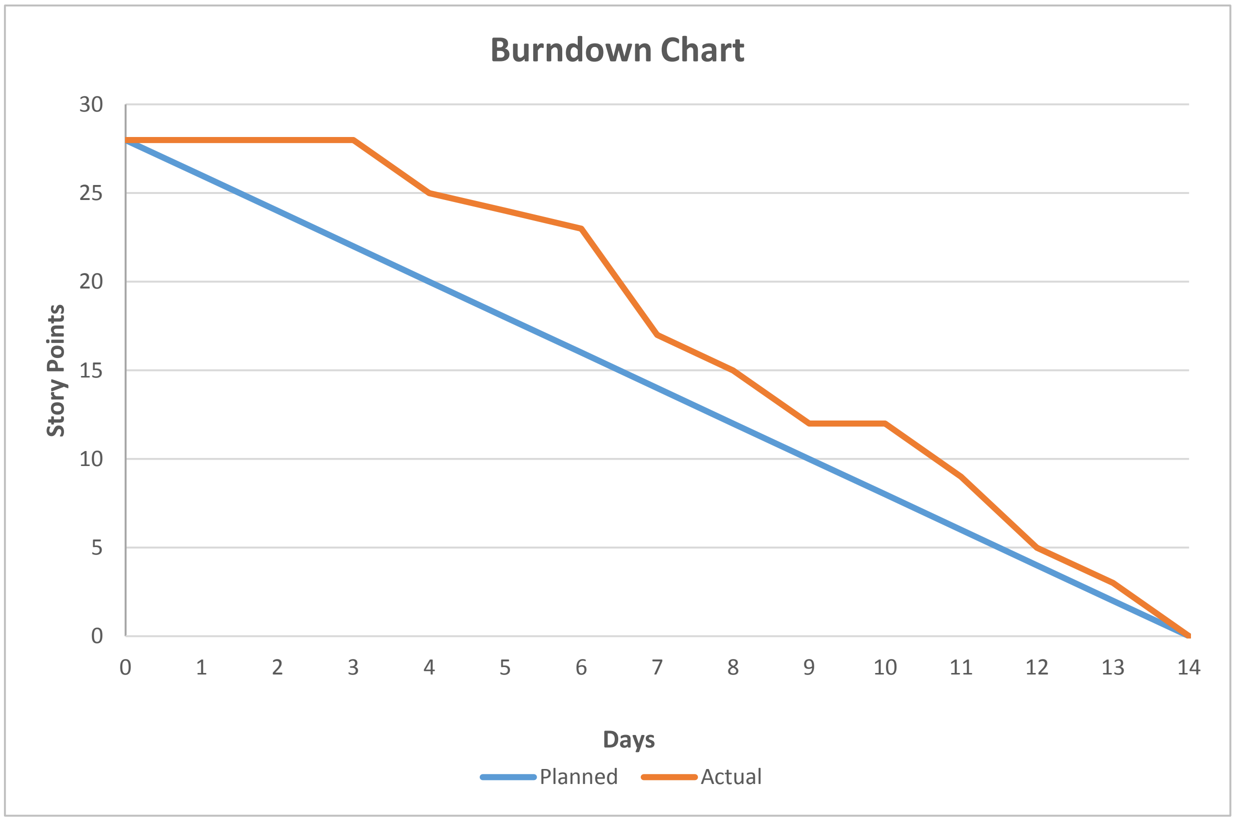
In Progress:

Testing:

Done:

User Guide	Mar 30	DK
Unit Testing and Acceptance Testing	Mar 30	ga
Documenting Code Design and UML	Mar 30	ju
Fixing Issue in rcparams.py	Mar 26	DC
Fixing Issue in lines.py	Mar 26	DC
Investigation of Issue 3293	Mar 19	me
Investigation of Issue 3293	Mar 19	DK
Investigation of Issue 3293	Mar 19	ga
Investigation of Issue 3293	Mar 19	DC
Investigation of Issue 3293	Mar 19	ju

Burndown Chart:



Burndown Chart
Story Points
Days
Planned
Actual