SE 3XA3: Requirements Document
xPyCharts

Team 4, xPy
Hatim Rehman (rehmah3)
Louis Bursey (burseylj)
Sarthak Desai (desaisa3)

October 11, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
Oct. 10, 2016	1.0	Revision 0

This document describes the requirements for .... The template for the Software Requirements Specification (SRS) is a subset of the Volere template (Robertson and Robertson, 2012). If you make further modifications to the template, you should explicity state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

The project being constructed is a recreation of JCharts which is a graphing library available to developers in Java. The purpose of XPYCharts is to be able to allow users to incorporate graphing capabilities into programs and generate simple line graphs. There are several existing programs that allow graphing on various platforms such as graphing calculators and online graphing tools however, graphing libraries are often hard to incorporate into programs. XPYCharts aims to be a simple to use product that can be used to accurately graph data points and produce smooth graphs.

1.2 The Stakeholders

1.2.1 The Client

The client of the project is a Company requiring a graphing library. The company is interested in deploying the graphing library as an educational tool to students and to other end users that may be developers. The client will be the one reviewing all the prototypes of the project and conducting evaluation before the project is deployed into the market.

1.2.2 The Customers

The Customers of the project are the end users who will be using the graphing library in their programs. This includes students of computer science in schools who are looking to graph their research work or use graphs within their applications. This group also include developers who require graphs in their applications. Furthermore, data scientists can use this library to enter large datasets in order to study trends and conduct research.

1.2.3 Other Stakeholders

Other stakeholders include the developers who are creating the project and who will be responsible for all future updates and maintainance.

1.3 Mandated Constraints

Description: The User must have Python 2.7 installed on their device. Rationale: User can only import and use the library inside Python 2.7 Fit Criterion: The library will be able to be imported into Python 100% of the time if Python 2.7 is installed on the system. If the user doesn't have Python 2.7 a link in the User Manual will lead the user to the appropriate download location.

1.4 Naming Conventions and Terminology

Table 2: Terminology

Term	Definition
API	Are routines and protocols that allow software development by specifying how software components should interact with each other.
JCharts	The existing graphing library that is being recreated in this project.
XPYCharts	This is the name of the project/library that is being created.
Library	Is a collection of already made routines that can be incorporated into a program to enable additional functionalities.
Client	Is the party that wants the project to be built.The individuals who are the main investors/stakeholders.
Stakeholders	The individuals/parties that have something to gain/loose from the outcome of the project or someone who is interested/concerned about the outcome of the project.
MATLAB	A high level language that is used for technical computing, visualization, graphing, and programming practices.
Python 2.7	Is a high level programming language. This is language in which the library will be developed and the language for the library will be available for use.

1.5 Relevant Facts and Assumptions

It is assumed here that graphing libraries are a valuable resource in educational, research and development environments and that each party involved will benefit from the production of this software. It is also assumed that the client currently only wants the project to produce line graphs and any further graphing capabilities may or may not be added depending on the client's decision and the time constraints.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

The context can be seen by the following visual, describing user interaction with the program and program response.

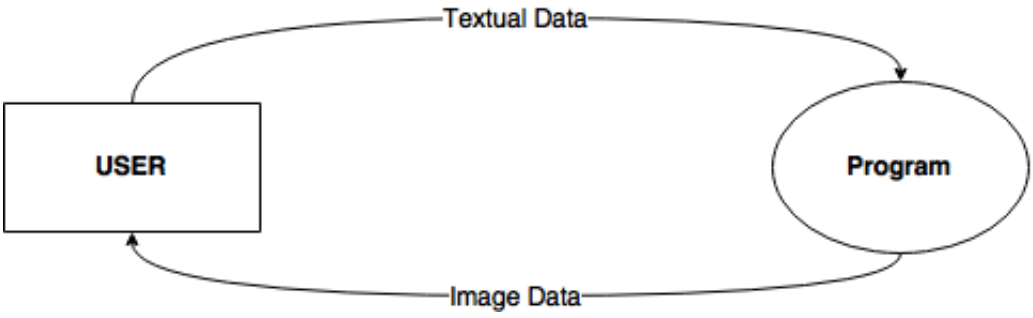

Figure 1: Context Diagram

2.1.2 Work Partitioning

Table 3: Work Partitioning

Event No.	Event
1	Create the (cartesian) coordinate system that is centered within a window.
2	Add labels to the coordinate system.
3	Plot sample points.
4	Construct a line that joins two points together.
5	Finishing edits (i.e input checking and error handling).

2.1.3 Individual Product Use Cases

Because of the nature of the product, a universal use case exists:

Use Case #: 1

Scenario:	Constructing a graph from a set of data.
Trigger:	User request.
Precondition:	Data inputted in the correct format.
PostCondition	Graph generated and outputted to the user's screen.

2.2 Functional Requirements

Requirement #: 1
Description: The software shall read data given to it.
Rationale: Data is needed to construct a graph.
Originator: Hatim Rehman
Fit Criterion: The data used by the program is identical to the data given to it.
Customer Satisfaction: 5 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** 2,3,4,5
History: Created October 10, 2016

Requirement #: 2
Description: The software will raise an exception if the data format cannot be plotted, and stop the program.
Rationale: It is safer to stop the program after it is realized the data points do not meet the expected format, versus letting the program proceed to unexpected behaviour.
Originator: Hatim Rehman
Fit Criterion: The program execution halts when improper data is entered.
Customer Satisfaction: 5 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** 3,4,5
History: Created October 10, 2016

Requirement #: 3
Description: The software will construct a coordinate system that will fit all the data points.

Rationale: This ensures the coordinate system is always dynamically generated to work for all data sets.
Originator: Hatim Rehman
Fit Criterion: The maximum value on the xy axes is $\geq$ maximum x, y in data set.
Customer Satisfaction: 5 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** 4,5
History: Created October 10, 2016

Requirement #: 4
Description: The software shall plot all the data points.
Rationale: The user will want all the data plotted.
Originator: Hatim Rehman
Fit Criterion: All data points exist on the generated graph.
Customer Satisfaction: 5 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** 5
History: Created October 10, 2016

Requirement #: 5
Description: The software will connect a line that passes through all the data points if the data points are a function of x.
Rationale: Imposing a constraint on only graphing functions ensures validity and correctness (a function only has one interpretation), whereas graphing relations introduces ambiguity in the shape of the line.
Originator: Hatim Rehman
Fit Criterion: A line passes through all the points if there is only one y value for each x.
Customer Satisfaction: 5 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
History: Created October 10, 2016

3 Non-functional Requirements

3.1 Look and Feel Requirements

Requirement #: 1 Requirement Type: 10a Event/Use case #:
Description: The graphs produced should be visually appealing and look professional
Rationale: The programmer may be producing graphs for presentations, and will appreciate a good looking product
Originator: Louis Bursey
Fit Criterion: 70% of people surveyed believe that graphs are visually appealing and look professional
Customer Satisfaction: 4 **Customer Dissatisfaction:** 2
Priority: Medium **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

3.2 Usability and Humanity Requirements

Requirement #: Requirement Type: 11a Event/Use case #:
Description: The product should be easy to use for novice Python programmers
Rationale: The programmer using this library should be able to focus on their program, not on using this library
Originator: Louis Bursey
Fit Criterion: 80% of programmers familiar with Python successfully use the product
Customer Satisfaction: 4 **Customer Dissatisfaction:** 3
Priority: Medium **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 11b Event/Use case #:
Description: When natural language is required, this product will use English
Rationale: Python is written in English

Originator: Louis Bursey
Fit Criterion: No non-English natural language is used in the product
Customer Satisfaction: 1 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 11c Event/Use case #:
Description: The programmer using this product should quickly be able to learn how to use this product
Rationale: Programmers who face a steep learning curve will be discouraged from using this product
Originator: Louis Bursey
Fit Criterion: Programmers familiar with Python are able to produce graphs within an average twenty minutes of acquiring the library
Customer Satisfaction: 4 **Customer Dissatisfaction:** 4
Priority: Medium **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 11d Event/Use case #:
Description: When used incorrectly, the product should generate error messages that are easy to understand
Rationale: Knowing when and how the library is being used incorrectly will help developers use the library more efficiently.
Originator: Louis Bursey
Fit Criterion: 80% of programmers using the library for the first time can understand the error messages they create
Customer Satisfaction: 5 **Customer Dissatisfaction:** 4
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

3.3 Performance Requirements

Requirement #: Requirement Type: 12a Event/Use case #:
Description: The product should generate graphs in a timely manner
Rationale: The program should not take so long that it slows down the programmer's workflow
Originator: Louis Bursey
Fit Criterion: The library takes under 20 seconds to generate graphs of a reasonable size
Customer Satisfaction: 4 **Customer Dissatisfaction:** 4
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 12c Event/Use case #:
Description: The product should produce accurate graphs
Rationale: Visual representations of data are useless if they don't represent data faithfully
Originator: Louis Bursey
Fit Criterion: Graphs produced should have no less than 20% difference between it and a graph generated by JCharts
Customer Satisfaction: 5 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 12d Event/Use case #:
Description: The product should always be available
Rationale: The product cannot unexpectedly go out of service, as programmers will depend on its availability
Originator: Louis Bursey
Fit Criterion: The product is always available
Customer Satisfaction: 1 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 12e Event/Use case #:
Description: The library will not stall out, if used incorrectly it will always display error messages and abort
Rationale: Programmers using the library will depend on graphs not stalling out their programs
Originator: Louis Bursey
Fit Criterion: Errors in use always create error messages and aborts, not stalls
Customer Satisfaction: 1 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 12f Event/Use case #:
Description: The library will be able to produce graphs with up to 500 data points
Rationale: Programmers using the library will want to build graphs from large data sets
Originator: Louis Bursey
Fit Criterion: Graph with up to 500 data points can be generated without problems
Customer Satisfaction: 3 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

3.4 Operational and Environmental Requirements

Requirement #: Requirement Type: 13a Event/Use case #:
Description: The product should operate on laptops and desktops
Rationale: Programmers work on laptops and desktops and the library should work in this environment
Originator: Louis Bursey
Fit Criterion: Personal computer users can run programs that use the library

Customer Satisfaction: 3 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 13a Event/Use case #:
Description: The product should be usable as a Pyton library
Rationale: The Python language is the supported language of this project
Originator: Louis Bursey
Fit Criterion: The product is importable in a Python program
Customer Satisfaction: 1 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 13c Event/Use case #:
Description:
Rationale: The product should be distributed as a zip file that is importable in Python programs
Originator: Louis Bursey
Fit Criterion: The product is importable in a Python program
Customer Satisfaction: 3 **Customer Dissatisfaction:** 4
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

Requirement #: Requirement Type: 13d Event/Use case #:
Description: Future releases of the project will be backwards compatible
Rationale: Backwards compatibility keeps programmers from having to update their code when we make changes
Originator: Louis Bursey
Fit Criterion: Releases are backwards compatible
Customer Satisfaction: 2 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None

History: Created October 5, 2016

3.5 Maintainability and Support Requirements

Requirement #: Requirement Type: 13d Event/Use case #:
Description: The product should work in Windows, Linux and Mac OSX environments
Rationale: Programmers working in all of these environments will need graphing capabilities
Originator: Louis Bursey
Fit Criterion: The library is usable in all of these environments
Customer Satisfaction: 4 **Customer Dissatisfaction:** 5
Priority: High **Conflicts:** None
Supporting Materials: None
History: Created October 5, 2016

3.6 Security Requirements

There are no security requirements for this project

3.7 Cultural Requirements

There are no cultural requirements for this project

3.8 Legal Requirements

There are no legal requirements for this project

3.9 Health and Safety Requirements

A graphing library does not pose any serious health and safety risks.

4 Project Issues

4.1 Open Issues

The major issue currently under investigation is of testing. Although manual testing of the graphs created by the library will be relatively easy to check against existing implementations of graphing libraries and calculators, automated testing is a challenge. The problem exists that there is no existing solution that the team has agreed upon as to how to check graphs that are generated during automatic testing. One solution that was initially suggested was to export the graph generated by the library as a JPEG file and compare it against another image file created by an existing implementation. However, variances in the scales or the differences in the formatting of the two graphs can lead to inaccurate outcomes to the image comparisons. The team is actively looking to solve this issue.

4.2 Off-the-Shelf Solutions

Existing off the shelf solutions include the Jchart Library that is currently available for Java on multiple platforms such as Mac and Windows. MATLAB is another widely known solution often used by students and professionals. There are also Python developed online graphing technologies such as Plotty. The team will be further looking into each of these existing products in order to gather information and for guidance on the project.

4.3 New Problems

There are no new problems that have been encountered by the team.

4.4 Tasks

Table 4: Tasks

Tasks	Completion Date
Develop skeleton axes	10/10/16
Add axes numbering/Scaling capabilities	10/14/16
Plotting basic (x,y) coordinates	10/20/16
Generate lines to connect plotted data points	10/28/16
Implement smooth curves for connecting data points	11/4/16
Product release/ Final Client Demonstration	11/28/16

Development Phases (Note: Highlights the major milestones for the project development)

Phase 1: Developing the basic output window once the user enters in the information is the first thing that the program must be able to do. This window should display a simple x and y axis. At this point no numbering or labels are required.

Phase 2: The program will implement the axes numbering and add the functionality of allowing the users to label axes, titles, change scales (more customizable).

Phase 3: In this phase the library should be able to be used to plot multiple points onto the generated graph.

Phase 4: The graphs in this phase should include lines that are connecting the data points together. Although the lines may not necessarily be smooth curves.

Phase 5: The library should be able to create smooth graphs with curved lines that connect data points. This is an important phase because if the library's functional requirements are changed to include graphing different types of functions then smooth curves will be required to properly depict functions.

Phase 6: By this point all final changes should be made and the product should be ready to be deployed to end users.

Note: Throughout all these phases of development testing will be a reoccurring phase. Testing will be done through both manual and automated methods. However, currently automated testing is still an open issue.

4.5 Migration to the New Product

Migration will not be an issue in this project. The library will be created for Python 2.7 only. The project will not pursue migration to Python 3.

4.6 Risks

There are existing risks regarding the scaling and intervals of the data points, given that the project has currently set up to 500 data points as a upper limit to accurate functionality. Given that a user has a larger amount of data points that need plotting the program may not allow to do so without compromising the non functional requirement of generating graphs in reasonable timings. However, if the program is completely limited to only allowing users to enter a certain number of data points the users may yield inaccurate results in terms of data analysis if they can only input partial data.

4.7 Costs

There are no monetary costs associated with the project. All software being used is accessible for free.

4.8 User Documentation and Training

The user document will have two major components to support the user. The first document will provide users with a guide on how to unpack the library provided as an archived file and import it correctly into their workspace. The second document will be an API specification documentation that will

provide support for the developers allowing them to quickly and efficiently incorporate the library functionalities into their program.

References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

5.2 Gantt Chart

The schedule of the project is shown by the following Gantt Chart.