

SE 3XA3: Development Plan xPyCharts

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

Contents

1	Abstract	ii
2	Team Meeting Plan	ii
3	Team Communication Plan	ii
4	Team Member Roles	iii
5	Git Workflow Plan	iii
6	Proof of Concept Demonstration Plan	iii
7	Technology	iv
8	Coding Style	iv
9	Project Schedule	iv
10	Project Review	iv
10.1	Hatim	iv
10.2	Louis	v
10.3	Sarthak	v

List of Tables

1	Revision History	i
---	----------------------------	---

List of Figures

Start filling this table, especially for deliverables - CM

Table 1: Revision History

Date	Version	Notes
Sept. 30, 2016	0	Revision 0
Dec. 7, 2016	1	Revision 1 - Table of Contents - Added "Prior to Demo" section to PoC plan - External links where appropriate - Project Review

1 Abstract

The purpose of the xPyCharts Software Development plan is to collect all the necessary information to complete the project in one central document. Team members will refer to the development plan to understand what they need to do and why they need to do it. It will also be used as a reference for decisions that are made in the future

This document describes the overall plan to be used by the xPyCharts project. It includes a Gantt Chart in the Project Schedule that shows a tentative flow of events required for the completion of the project. **Consider just using a table of contents, and the above can be in a Abstract section - CM**

2 Team Meeting Plan

Our team will hold regular meetings on Tuesdays and Thursdays, for about two hours. We will meet either in a free room in MDCL or in the Thode Library.

Meetings will begin with a discussion of the tasks completed for this meeting. If a task has not been completed, a plan will be made to complete it, and we will discuss the reasons this task was not completed.

Meetings will be chaired by the person most knowledgeable about the meeting's focus. Team members should come to the meeting with their tasks completed, aware of the current state of the project, and aware of the next steps planned. They should also prepare agenda items that they feel important to discuss.

The Scribe will take minutes and draft agendas from the items brought by team members and upcoming milestones. Each agenda item should be phrased as a question, and should be unbiased, e.g. *What Language Should We Use* is better practice than *Should We Use C++?*.

The meeting should follow the structure of the agenda. Important decisions will ideally be decided by consensus. If a discussion stagnates and moves beyond its allotted time without the group reaching a consensus, a decision will be made by vote.

Meetings will end with the scribe writing a brief summary of the decisions made and the tasks to be completed. We will also briefly reflect on the effectiveness of the meeting, and the things that can be done to make meetings more productive.

3 Team Communication Plan

The primary communication method for all issues regarding the project will be Facebook. We have created a Facebook page dedicated to our project. In this manner any important announcements can be posted on the page and discussions can be carried out in a chat. We have exchanged phone numbers in case we need to discuss issues verbally.

4 Team Member Roles

- Team Leader/Programmer: Hatim Rehman
- Scribe/Programmer: Sarthak Desai
- LaTeX Expert/Programmer: Louis Bursey

We have selected project roles according to each project member's skill set. The experts in each of the category will be supporting the other team members. All members of the team are also working towards learning Git and familiarizing themselves with other required applications.

5 Git Workflow Plan

The team will be using the Centralized Workflow Plan. This plan will be ideal for this project due to the small size of the team. All members will pull from the remote repository before they start making changes and then push their changes from their local repository once they are finished. There will not be any branching involved in the project and it will only have a linear commit history.

6 Proof of Concept Demonstration Plan

Despite checking all the boxes of what should be included, try to include more requirements on what needs to be done prior to the demonstration - CM

The demonstration will consist of a graphical interface displaying the X and Y axes that are dynamically generated. The resulting Cartesian Coordinate system must be able to plot (x, y) pairs.

- **Difficulties that may arise in implementation:**

Plotting will be a challenge once the axes are drawn. The coordinate system in the code will have to refer to the the system drawn on the graphics template, while being written in the coordinate system of the panel window.

- **Difficulties that may arise in testing:**

Manual testing will not be an issue as we can simply compare data plotted by the program against the same data plotted on an online tool (such as WolframAlpha).

Automated testing, however, will be a challenge if we want to compare two image files to check if the graph is accurate. One way to automate would be to create a bitmap of the program's output, and superimpose it onto another bitmap created by an online tool (such as WolframAlpha) and measure the differences. A discrepancy of z (to be determined) will allow us to measure correctness.

- **Required libraries and portability:**

The program will use as many standard Python libraries as possible to ensure maximum portability. When an external library is required, preference will be given to the most updated library to ensure future maintainability.

- **Prior to demonstration:**

What must be prioritized prior to the demonstration is finding a suitable, graphical interface library for python that is cross-platform, easily accessible, and easy to use. Development cannot begin until this is done so it is very important. A layout of module and functions would follow after this to make sure work is distributed properly and all the work is realized at the beginning stage so it can be fairly distributed.

7 Technology

This project will be developed in the Python programming language. We will use standard Python libraries whenever possible, and otherwise will prefer more recently updated libraries. To aid in project management, we will use the free version of the PyCharm IDE. For automated testing, we will use the ImageMagick image comparison tool and the test files provided in the JCharts project.

8 Coding Style

Provide links to external content whenever possible - CM

We will adhere to the PEP 8 for Python style. Thorough and meaningful commenting is expected in all code.

9 Project Schedule

Use more appropriate text to tell reader it is a link. With respect to the chart, adjust the resource section so that no person is over 100 percent. - CM
Gantt Chart.

10 Project Review

10.1 Hatim

xPycharts was an ambitious project aimed at python users, such as myself. Had there been more time, the library would have features more types of graphs and even more functionality for the current xy-chart.

The things I enjoyed about this project would be the experience in git that I take away from it. I am more confident in my understanding of what it means to push, pull, merge etc now than I was beforehand, and the idea of using it in

future projects does not make me as nervous any more. As for xPycharts, I am happy with what we were able to complete in the 3 months of development we had, despite small glitches and bugs that may exist.

Things that I disliked about the project would have been the documentation, though I realize it is necessary and a vital part of the software development project experience.

In the future, if I would get the opportunity to manage a project like this again what I would do differently would be to first figure out what functionality we want in the graphs and outline the modules in the beginning. That way, we would be able to work in parallel a lot more efficiently, and manage our time a lot better.

10.2 Louis

XPYcharts was a graphing library aimed at providing simple graphs to python users. For myself, it was a great opportunity to brush up on my Python skills, learn a variety of useful project management tools, and produce an exciting application.

I was grateful for the opportunity to learn Git. It's a tool that I wouldn't have considered using, but with some experience, it's easy to see how much it can improve development. I have started using it in my own projects outside of class, and will continue to explore it's functionality. I also appreciated the opportunity to apply software documentation to a "realer" or more involved project than those I have previously worked on. Although documentation isn't the most exciting part of programming, it is certainly necessary, and having experience with a real project helped me to understand the value of doing things systematically and thoroughly.

Working with graphics was an aspect of this project that I didn't particularly enjoy. It was a good experience, and I'm glad to have explored this subject, but I prefer working with back-end aspects of projects. With that being said, I will benefit from the working knowledge of TKinter I developed in this project.

In the future, I would attempt to lay out the structure of our project earlier in development. This would have made documentation an easier task, and could have prevented later issues in the code. In addition, I would have redeveloped a project that was more accessible, as JCharts's code was often difficult to understand, and was not a valuable resource for this project beyond the conceptual level.

10.3 Sarthak

Having formed our group for the course, when we started to look for projects my initial thought was to choose something we were all experienced with, a simple 2D game. However, we eventually ended up deciding to create xPycharts, a

graphing library. At first I was not so confident in our ability to explore this type of project and thought it may be a little too ambitious. However, as the weeks progressed the team worked very well together and we were able to see the project through and I am very proud of what we have created.

My most valuable experience out of this project was the exposure to Git. Having always used zip files to transfer projects through e-mail before was always a tedious task. However, the fact that Git was integrated as part of the project forced me to learn it and over time I was able to understand its importance.

My major issues with the course and the project came from the documentation. This was mostly because given such a short time of a few months and a lot of documentation it seemed as if they were very closely packed together and there was always another document due, having just completed one.

If I were to take part in another project like this in the future I would like that there was more planning that happened going into the project first. It would be much more ideal having to spend more time first planning the project out correctly rather than diving into the actual code early into the project.