SOEN 6471

ADVANCED SOFTWARE ARCHITECTURE

SUMMER 2023

Deliverable 1

Declaration

We, the members of the team, have read and understood the Fairness Protocol and the Communal Work Protocol, and agree to abide by the policies therein, without any exception, under any circumstances, whatsoever.

Group D

Jigar Maheshbhai Borad
Bhavye Budhiraja
Rancy Chadha
Payal Raj Chaudhary
Raviraj Bhaveshbhai Savaliya

Table of Contents

Details of Deliverable 1

Deliverable 1 focuses on providing a comprehensive understanding of **matplotlib**, which is our selected OPENARCH for this project. This deliverable consists of three distinct problems that need to be addressed. Problem 1 requires a concise description of matplotlib. This description encompasses the purpose of matplotlib, highlighting unique defining characteristics, and outlining the key software activities undertaken during its development. Problem 2 involves the creation of a context of use model for matplotlib which entails defining and describing each entity within the model. Lastly, Problem 3 entails constructing a stakeholder model for matplotlib, which can be represented through a UML Class Diagram or a mind map. The stakeholder model should identify and define each stakeholder involved in the matplotlib project.

These three problems collectively contribute to an in-depth understanding of matplotlib's purpose, usage context, and stakeholder involvement.

Contributions on Deliverable 1

Group D	Contributions
Jigar Maheshbhai Borad	- Researched and gathered information about matplotlib. - Wrote the description of matplotlib (Problem 1).
Bhavye Budhiraja	- Conducted in-depth analysis of matplotlib's purpose and unique characteristics. - Worked on document formation following the given template. - Assisted in defining stakeholders and dividing them into three categories.
Rancy Chadha	- Collaborated in defining the entities in the context of use model (Problem 2). - Designed and created the stakeholder model class diagram(Problem 3). - Collaborated with Bhavye in document formation.
Payal Raj Chaudhary	- Researched and analyzed the Context of use of matplotlib. - Collaborated with Jigar in researching and gathering information about matplotlib. - Designed and created context of use model (Problem 2).
Raviraj Bhaveshbhai Savaliya	- Reviewed and provided feedback on the description of matplotlib(Problem 1). - Assisted in creating the mindmap of stakeholders(Problem 3) - Collaborated with Bhavye in document formation.

GitHub Repo Link: https://github.com/Bhavye27/SOEN-6471-Team-D-Matplotlib.git

PROBLEM 1

1.1 Introduction

Matplotlib [1] is a powerful data visualization library widely used in Python programming. With its extensive functionality, Matplotlib allows users to create a variety of high-quality plots, charts, and figures for effective data representation. It provides a flexible and intuitive interface for generating line plots, scatter plots, bar charts, histograms, and more. Matplotlib supports customization of every aspect of the plot, including axes, labels, colors, and styles, enabling users to create visually appealing and informative visualizations. Its integration with NumPy, another popular Python library, makes it a preferred choice for data analysis and scientific computing. Overall, Matplotlib is an essential tool for data scientists and researchers seeking to visualize their data effectively.

1.2 Characteristics

1.2.1 Easy to Use
Matplotlib's extensive documentation and vast collection of examples [2] makes it simple for users to build plots and visualizations with only a few lines of code. Its syntax is designed to be accessible to both beginners and experienced programmers.

1.2.2 Comprehensive Plotting Capabilities
Matplotlib offers a wide range of plotting options, including line plots, scatter plots, bar charts, histograms, pie charts, 3D plots, and more. It supports a range of plot styles and offers flexibility in expressing various sorts of data.

1.2.3 Customization & Compatibility
Matplotlib allows users to customize every aspect of a plot, giving them full control over the visual representation of their data. This flexibility enables the creation of tailored visualizations. Matplotlib integrates well with other scientific computing libraries, making it a compatible choice for data analysis tasks.

1.2.4 Support for Interactive Plotting
Matplotlib offers interactive functionality, such as zooming, panning, and selecting data points, allowing users to interactively explore and analyze data. To produce interactive visualizations, it can be integrated with other Python packages like Jupyter Notebook.

1.2.5 Matplotlib's Backend Support
Matplotlib offers multiple backends, allowing plots to be displayed in different environments, including interactive GUIs, web applications, and static images. This flexibility ensures compatibility with various platforms and requirements.

1.3 Salient Features

During the development of Matplotlib, various essential software activities are carried out:

1.3.1 Requirement Gathering and Designing Architecture

To begin, developers engage with the user community, review feedback, and identify areas for improvement through requirements gathering. Following this, they create a design and architecture plan, defining the codebase structure, modules, and components, and determining how the changes will integrate with existing functionality.

1.3.2 Implementation

The actual implementation of new features and bug fixes is done through coding. Developers write and test code, adhering to Matplotlib's design guidelines and coding standards.

1.3.3 Testing

It is a critical activity in software development. Unit tests to ensure that individual components and functions operate as intended. Integration testing to verify interactions between different parts of the library. Regression testing was conducted to ensure that existing functionality remains unaffected.

1.3.4 Issue Tracking

Matplotlib has an issue tracking system, enabling users to report bugs, suggest enhancements, or raise concerns. Developers review and prioritize these issues, actively working towards their resolution.

1.3.5 Release management:

It involves coordinating the release process, including packaging the library, preparing release notes, and ensuring backward compatibility. Developers follow a release schedule, aiming to deliver stable and tested versions of Matplotlib.

PROBLEM 2

Context of use of matplotlib has various entities as specified below.

2.1 Users

Scientists and researchers use Matplotlib to visualize experimental results, research findings, and simulation data. Matplotlib can be used in conjunction with other libraries, such as Basemap or Cartopy, to create maps and visualize geographic data.[8]

2.2 Developers

Matplotlib is developed by the core and the community developer. A platform that plays an important role during development is Github, which is used for code versioning, issue tracking and project management.[8]

2.3 Testing Tools

Through the use of continuous integration tools, all code is tested and integrated. The pytest and Tox packages are used for code testing, while Codecov is used to check the code coverage.

2.4 Software Dependencies

The majority of dependencies [4] are image format libraries, such as libpng, as well as common libraries like pyparsing and dateutil. NumPy, which is used for numerical operations, and six, which is required for backward compatibility with Python 2, are two libraries on which Matplotlib strongly depends.

2.5 User Interface Backend

Matplotlib uses Tkinter and wxPython for creating the interactive user interface. Tkinter to create a simple GUI and embed a Matplotlib plot within it to build an interactive interface. wxPython is another popular Python library used to create a GUI and integrate Matplotlib for plotting within the application.[8]

2.6 Cross Platform

Matplotlib is cross-platform and will work on Windows, macOS, and Linux.

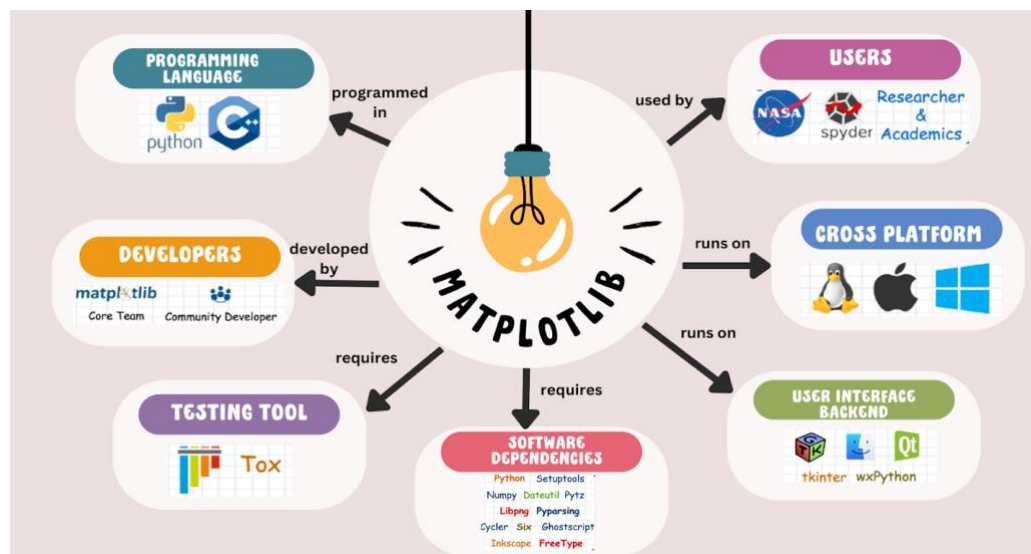For wholesome view of matplotlib's context of use [3] model we have given below context diagram:

Figure 1: Context of Use

PROBLEM 3

Stakeholders [5] in software architecture provide a wealth of benefits. Their diverse perspectives, influential decision-making, alignment with business goals, and representation of various

interests contribute to the effectiveness, relevance, and success of the software architecture. Collaborating with stakeholders throughout the process fosters a holistic and well-rounded approach to architectural design and implementation.

As identified by ISO/IEC/IEEE 42010 [6] Standard the class diagram below illustrates the relation of Matplotlib with various software system aspects.
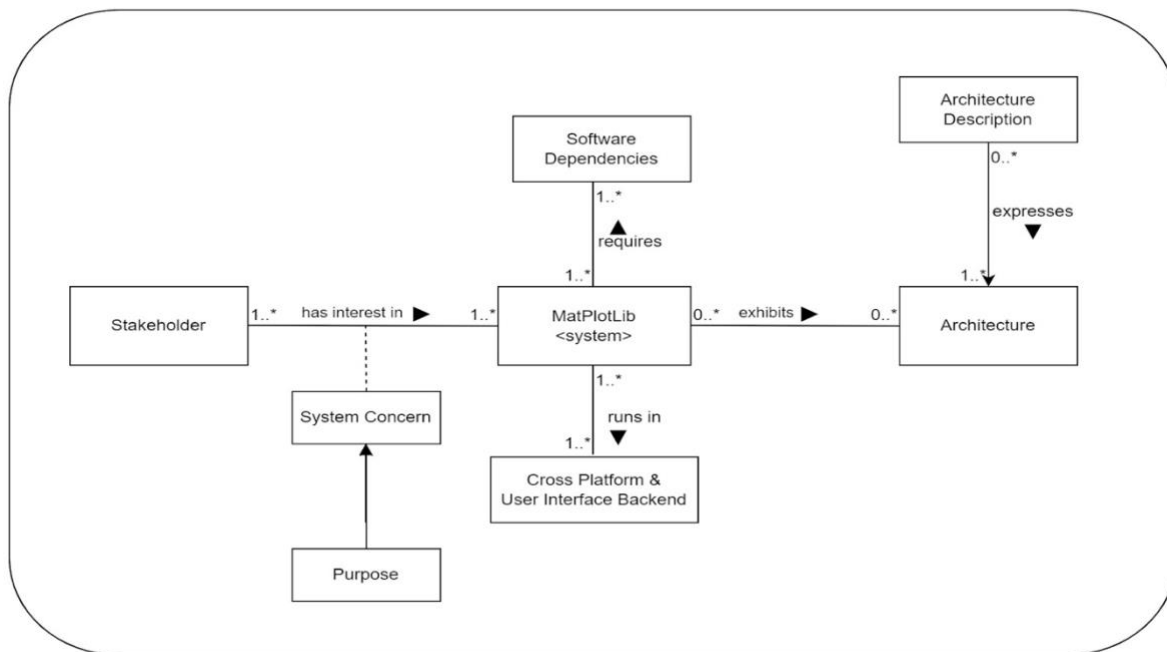

Figure 2: Stakeholder Model Class Diagram of Matplotlib as per ISO/IEC/IEEE 42010

Stakeholders of Matplotlib [7] are provided below and have been classified into three tiers: primary, secondary and tertiary. **Primary** stakeholders have a direct and significant interest in the system and are main beneficiaries or directly affected by the outcomes of the project. **Secondary** stakeholders have an indirect or moderate interest in the system and may not be directly affected by the system's outcomes, but they still have a stake in its success or failure. **Tertiary** stakeholders have a general interest or concern related to the project or system but are not directly involved or significantly impacted by its outcomes.
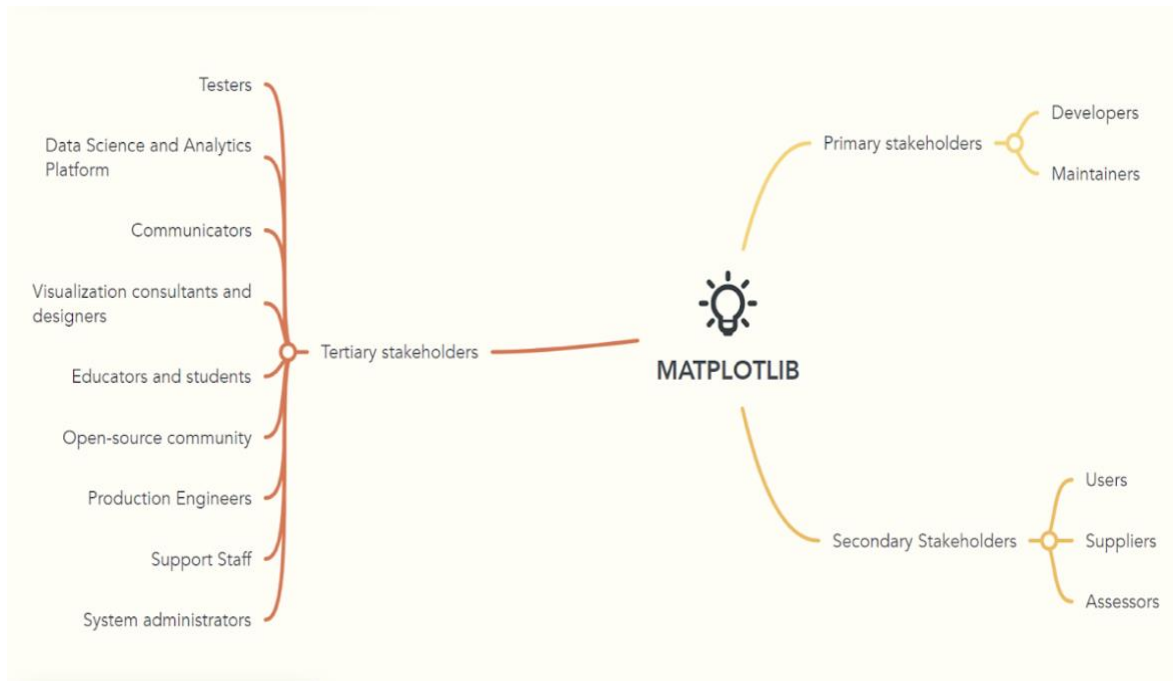
Figure 3: Mind map of Stakeholders

3.1 Primary stakeholders

3.1.1 Developers

Developers are individuals who contribute code, fix bugs, and implement new features in Matplotlib. They play a crucial role in the evolution and improvement of the library by writing and reviewing code, collaborating with other developers, and participating in discussions about future development.

3.1.2 Maintainers

The maintainers are individuals or a team responsible for overseeing the development and overall maintenance of Matplotlib. They ensure that the library remains up-to-date, fix bugs, review and merge contributions from developers, and manage the release process.

3.2 Secondary stakeholders

3.2.1 Users

Users represent individuals or organizations that utilize Matplotlib for their data visualization needs. They range from data scientists and researchers to industry professionals who rely on Matplotlib to create meaningful and informative visualizations for analysis, presentations, and decision-making.

3.2.2 Suppliers

Suppliers in the context of Matplotlib can refer to individuals or organizations that provide additional resources, such as plugins, extensions, or related tools that enhance the functionality of Matplotlib. These suppliers contribute to the ecosystem surrounding

Matplotlib   by   expanding   its   capabilities.

3.2.3 Assessors

Assessors may refer to individuals who evaluate the quality, performance, and compliance of Matplotlib. They may conduct code reviews, assess the library's adherence to standards, and provide feedback on its overall robustness and reliability.

3.3 Tertiary stakeholders

3.3.1 Testers

Testers play a vital role in ensuring the quality of Matplotlib. They write and execute tests to identify bugs, validate the functionality of the library, and verify that new features or changes do not introduce regressions. Testers help maintain a stable and reliable software product.

3.3.2 Data Science and Analytics Platforms

Data science and analytics platforms, such as Jupyter Notebooks, Anaconda, and Google Collab, are major stakeholders in Matplotlib. These platforms often come bundled with Matplotlib as a fundamental visualization tool, providing users with a seamless and integrated experience for data analysis and visualization.

3.3.3 Communicators

Communicators encompass individuals who actively engage in disseminating information about Matplotlib. They may include bloggers, technical writers, conference speakers, or community managers who promote the library, write tutorials, create documentation, and foster communication within the Matplotlib user community.

3.3.4 Open-source community

Contributors, bug reporters, and community members who actively participate in the development   and   improvement   of   Matplotlib.

3.3.5 Production Engineers and Support Staff

Production engineers focus on the operational aspects of Matplotlib. They ensure that the library can be deployed, scaled, and maintained effectively in production environments. They may be involved in optimizing performance, managing dependencies, and ensuring the library's reliability in real-world scenarios. Support staff refers to individuals who provide assistance and technical support to Matplotlib users. They may be part of the Matplotlib project or associated organizations, responding to user inquiries, troubleshooting issues, and providing guidance on how to use Matplotlib effectively.

3.3.6 System administrators

System administrators manage the infrastructure and systems required for running Matplotlib. They ensure the availability, reliability, and security of the platforms on which Matplotlib is deployed, such as servers or cloud-based environments.

REFERENCES

[1] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[2] [John Hunter, Darren Dale, Eric Firing, Michael Droettboom, 2002-2012] Available: https://matplotlib.org/stable/gallery/index.html

[3] [Pankaj Kamthan, 2023] Understanding Context, Available: http://users.encs.concordia.ca/~kamthan/courses/soen-6471/understanding_context.pdf

[4] [John Hunter, Darren Dale, Eric Firing, Michael Droettboom, 2002-2012] Dependencies, Available: https://matplotlib.org/stable/devel/dependencies.html

[5] [Pankaj Kamthan, 2023] Stakeholders Of Software Architecture, Available: http://users.encs.concordia.ca/~kamthan/courses/soen-6471/software_architecture_stakeholders.pdf

[6] ISO/IEC/IEEE 42010, 2011; ISO/IEC/IEEE 42010, 2022

[7] [Andreas Maruli C Pangaribuan, Helmiriawan, Rizky Dharmawan, and Sindunuraga Rikarno Putra] Matplotlib - The Python 2D Plotting Library, Available: https://delftswa.gitbooks.io/desosa-2017/content/matplotlib/chapter.html#rw

[8] Matplotlib. (n.d.). Retrieved from https://matplotlib.org/