Development of Data Products
Project Assignment
September 2022

Student: Octavio Alcazar Sanchez - 18697

Free University of Bozen-Bolzano Professor: Matteo Camilli

Subject Name: Development of Data Products - 73007

Contents

1 Project Introduction

Context

COVID-19 shaped our current lifestyle and changed the health measurements forever. The actual decade has seen several cases of people getting affected by the coronavirus pandemic, and special attention and priority should be given to every health, government and involved research areas institutions. Data is everywhere, is generated every day from an infinite number of sources and is on constant processing by machines programmed by the responsible professional groups.

When relating data over a certain period, time series data is the core of analysis. Thus, time series data related to COVID-19 cases is collected by multiple organizations and ready to be retrieved, filtered, processed and visualized for human understanding. After all, the purpose of working with data is the benefit of the human well-being, by prevention and smart decision-making after conclusions have been proposed with the working data.

All the personnel interacting directly with health prevention, those being health care professionals, researchers, and government experts, need a user-friendly way to access real-time critical data regarding the monitoring of the different cases at any timeframe in any specific region or country. As collected information comes in form of time series data, it is the job of the proper data analysts, engineers and other data-oriented members to properly load and merge the data from different sources, clean and filter the data, aggregate and transform the data, and present it to the customer via clear visualizations or inferred conclusions from hypotheses for supporting governments and involved institutions in decision making related to the COVID-19 situation.

Starting Point

During pandemic, every newspaper, website, television broadcast and any other communication source has presented the statistics for a city, region, or country of interest regarding the number of cases, the number of deceased and number or recovered people. There has been in most of the cases a common data source: the Johns Hopkins University. This research and academic institution, located in Baltimore, United States, has been always a leading reference point for reliable statistics about most of the political, health, and science aspects of the entire world. Therefore, the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) has shared their knowledge data on a public GitHub repository. The available source aggregates pandemic data from primary sources, such as the World Health Organization, national, and regional public health institutions.

Another source is the Oxford COVID-19 Government Response Tracker (OxCGRT). It collects systematic information on policy measures that governments have taken to tackle the pandemic, they are adapted on a scale to reflect the extent of government action, and scores are aggregated as policy indexes. The summarized index is called a Stringency Index (SI).

Both data sources can be merged for creating a robust data frame that is going to be used for retrieving any data selection for visualization about the available variables at any region of interest for a given timeframe selection.

Functional Objective

The objective of creating a data product with source information about the coronavirus cases is to aid the customers in reacting the smartest way possible, this meaning in applying, freezing or suspending

policies and restrictions, always prioritizing the health of the population. A shift on how data analysis and visualization methodologies are executed under time pressure has been introduced after the pandemic situation, as nowadays a significant load of data is on the market ready to be processed and presented to end-users, allowing the creation of data products related to the monitoring of the COVID-19 cases, via web applications, key visualizations inside slide presentations, real-time software products, and more interesting alternatives.

The conducted project intends to track and compare government responses to the coronavirus outbreak by collecting and classifying different measures used by government bodies around the world. In short, the developed product should:

- collect and integrate pandemic and government responses from multiple sources
- merge, filter, clean and aggregate the collected data by different chosen criteria
- present the data to the customer via visualizations and found insights

The product focuses heavily on how governments determine the immediacy of restrictions and prevention regulations, this measured by the computed Stringency Index. A comparison of different government entities, different countries, is to be carried out by collecting, aggregating and presenting time series data. The product main priorities are listed below:

1. Display epidemic and Stringency Index evolution over a selected region or country through time
2. Elaborate comparisons of cases, death and recovered patients, between two selected regions or countries given a time window

The tools to use for developing the product idea are given as free choice options. As Python is the most popular programming language, and it has several user-friendly packages for creating visualizations and user interfaces, it is the chosen solution for the backbone of the project.

Non-Functional Objective

During product design stage, there are important requirements that should be taken into account, listed as non-functional objectives, they do not describe any product functionality that is reflected in code data preprocessing, data filtering, data aggregation or data visualization, but more in how the product can be easily adaptable to unforeseen changes and information updates without the need to modify the core blocks of code, which is normally presented as class modules, class methods or independent functions:

- Data is collected from different sources, meaning that the retrieved information can be of any data format, making the product stage of data collection more complicated. By design principle, every class, module, or function in the program should have just one dedicated purpose. The clear separation of functional responsibilities is related to the Single Responsibility Principle (SRP), part of the well-known SOLID design principles for object-oriented software products.
- The produced code should be designed to be extensible for additional data sources and for any new user query, as the user's data of interest can change as the pandemic situation is in constant evolution and the implied government bodies can implement new regulations. The capability to be easily extended relies on the Open-Closed Principle (OCP) from the SOLID design principles.

Project Overview

Every project implementation, regarding its scope, background and research focus should be preceded by a planning phase, where the overview questions should be formulated: What is the goal of this project? How the project should be split into tasks and subtasks? Who are the team members and which tasks are assigned to each partner? How to validate the developed product?

The planning process starts with a sketch of brainstorming ideas for answering the previous questions, which could be written in a notebook, a whiteboard, any software organizing tool. The information should be clear and available to every team member, as well for the end-user. As there is no need to reinvent the wheel, the product development methodology can be referenced from sources related to product and project management, where experienced people can state how a product planning should be executed.

By consulting websites dedicated to product and project management, a draft overview on how the product should be planned is presented below. The product overview begins with defining the functional objective, followed up by the creation of tasks and subtasks, assigning priorities to the most crucial ones. Then, a list of roles is done for the team members, also listing every human, data, and technical resource available for the product design. At the end, a test process is expected for validating the product.

Product planning proposal can be translated in a purpose-oriented workspace by utilizing diagrams, charts, tables and brainstorming maps for collecting business ideas. An online visual workspace is used for sketching diagrams about the conducted project steps and methodology. The description of each mentioned step in the product overview is explained in more detail on section 5, Product Methodology.

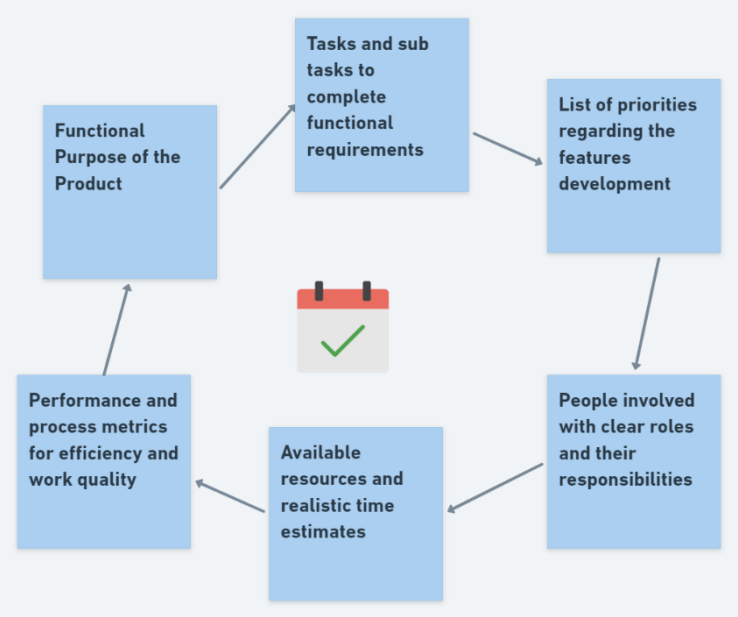

Fig. 1 Product Overview

Report Structure

The project report is divided into six chapters for comfortable reading to the user. The first chapter consists of an explained introduction of the product: the actual context, the project objectives and the starting reference points are mentioned in detail. An introductory overview of the project methodology is also listed and represented as a connected diagram. The second and third chapters are related to the theoretical concepts related to the collected information, to design principles, to product development planning and organization frameworks. The second chapter briefly describes the indexes and coefficients used by the government entities for tackling the situation. The third chapter contains software product development concepts and the framework definitions which are used on this project; well-known methodologies for product planning, monitoring and organization. The fourth chapter lists the utilized resources for the project achievement: literature sources, data sources, and the software tools run in the local computer for storing, executing and updating the project. The fifth and sixth chapters are the backbone of the project, as they describe the carried-out planning and task decomposition, the implementation, the collected insights and key visualizations, ending with a description of validation tools. The report concludes with the consulted bibliography listed in the seventh chapter.

2 Theoretical Background

Stringency Index

The conducted project intends to track and compare government responses to the coronavirus outbreak by collecting and classifying different measures used by government bodies around the world, turning them into a Stringency Index (SI).

The Oxford Covid-19 Government Response Tracker (OxCGRT) records information regarding government responses towards movement and travel restrictions, school and workplaces closures, event cancellations, social gatherings regulations. The measurements are split by indicators, which combined into a formula generate the Stringency Index, a value fluctuating between 0 and 100. The indicators and the mathematical formula for the SI are presented below.

Indicator Code	Indicator Name	Indicator Value Ranges
C1	School Closing	[0, 1, 2, 3]
C2	Workplace Closing	[0, 1, 2, 3]
C3	Cancel Public Events	[0, 1, 2]
C4	Restrictions on Gatherings	[0, 1, 2, 3, 4]
C5	Close Public Transport	[0, 1, 2]
C6	Stay at Home Requirements	[0, 1, 2, 3]
C7	Movement Restrictions	[0, 1, 2]
C8	International Travel Controls	[0, 1, 2, 3, 4]
H1	Public Information Campagins	[0, 1, 2]

Table 1 Stringency Index Policy Indicators

The Stringency Index is calculated using the policy indicators C1 to C8 and H1. SI value on any given day is the average of the nine sub-indices and gets a value between 0 and 100.

$$I = \frac{1}{9}\sum_{j=1}^{9} I_j$$

Fig. 2 Stringency Index Formula

Design Principles

From reference [4}, the SOLID design principles should be taken into consideration for every software product design, as they help the developers to create maintainable, reusable and flexible designs. Each letter in the acronym stands for a specific principle. For this project, the first two principles must be considered as part of the non-functional objective.

The Single Responsibility Principle (SRP) principle states that every class, module, or function in a program should have exactly one purpose. As a commonly used definition, "every class should have only

one reason to change"; as a result, the code can be maintainable and scalable by clearly separating concerns and responsibilities and not by trying to achieve many functionalities in the same entity.

The Open–Closed Principle (OCP) states that software entities should be open for extension, but closed for modification. The program classes, functions and methods functionalities should be capable to be extended by user-demand without the need to alter the program source code. It works with the analogy of adding a block on top of a solid one, without the need to reshape the backbone block.

3 Software Product Development Background

Product and software development efforts require a process to follow. Modern industries model their workflow using a methodological framework, defining the objectives, the scope, the limitations and the product features since the planning phase. The components can be modified through time, as time, human, and budget resources can suffer unexpected changes.

The adoption of a project development framework aims to benefit the product owner, the company, the team members, and the end users or customers. For software development, the introduction of the Agile mindset is a popular choice for planning and executing projects, and two well-known frameworks like Kanban and Scrum are briefly described.

Agile Framework

Using the given definition by [5}, Agile is an iterative approach to project management and software development that helps teams deliver value to their customers, which adapts to tight feedback cycles and continuous improvement. It works by organizing work in small, but consumable, increments; requirements, plans, and results are evaluated continuously so teams can respond to changes quickly.

Agile calls for collaborative cross-functional teams: open communication, collaboration, adaptation, and trust amongst team members. The team is intended to take the lead on deciding how the work is going to be executed, by self-organization of tasks and assignments. The Agile essence is summarized in the following bullet points, which are part of the Agile Manifesto that came out in 2001:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Kanban

Kanban is a popular framework, which dates more than 50 years back, used to implement Agile and software development. It requires real-time communication of capacity and full transparency of work, as it fully visualizes work items on a Kanban board, allowing team members to monitor the state of every work task at any time. Kanban provide teams more flexible planning options, faster output, clearer focus, and transparency throughout the development cycle.

The Kanban board makes use of cards, columns, and continuous improvement to help technology and service teams commit to the right amount of work, and with the intention to get it done. The term “Kanban" is the Japanese word for "visual signal", which accurately describes the main concept of visualizing work, by keeping every team member on the same page through the entire workflow.

Scrum

Scrum is a framework that helps teams work together by learning through experiences, self-organizing while working on a problem and reflecting on their successful progresses and blockers to continuously improve. Where Agile is more seen as an adopted mindset, scrum is the framework which describes a set of planning and retrospective meetings, tools, and defined roles for helping teams to structure and manage the workload.

The scrum framework is heuristic, as it is based on continuous learning and improvement. It acknowledges that the team does not know everything at the start of a project and will evolve through experience. Scrum differs from Kanban, as the former is based on learning from past experiences and has fixed and short planning and execution intervals; the latter relies its concept entirely on full transparency and visualization of the work progress and has a continuous workflow.

Sprint Planning

Scrum framework moves fast, with iterations called sprints that usually last between one to four weeks, which have clear start and finish dates. The short time frame forces complex tasks to be split into smaller stories and help the team to learn quickly and to accept the expected flaws.

The sprint is a set time window where all the work is done; the team must decide on how long the time box is going to be, the sprint goal, and the starting point of the sprint. The sprint planning session kicks off the sprint by setting the agenda and focus. A starting point for the sprint plan is the product backlog - a list of previously agreed tasks that could potentially be part of the current sprint. The team should also look at the existing work done in the increment and have a view to capacity.

The purpose of sprint planning is to define what can be delivered in the sprint and how the work will be achieved, by assigning clear roles, defining user stories and its divisions into tasks and subtasks. Sprint planning is done in collaboration with the whole scrum team. If sprint planning is done correctly, it helps on creating a motivation and challenging environment for the team; on the contrary. bad sprint planning can derail the team by setting unrealistic expectations.

User Stories

Work inside the Agile framework is constantly divided: thus, the smallest unit of work is known as a user story. It is an informal, general explanation of a requested software feature written from the perspective of the end user. The purpose of a user story is to highlight the added value of a particular software feature to the customer.

User stories are made up few sentences in simple language that outline the desired outcome without the need to go further into detail. Requirements are added later, once agreed upon by the team. User stories are part of common Agile frameworks like Kanban and Scrum. In Kanban, teams can pull up user stories into the backlog and run them through the workflow, and so the team learns to manage work-in-progress and to refine the workflow. In scrum, user stories are added to sprints and tackled over the sprint duration. By the performance on story completion is how the team can learn from experience, providing retrospective for better time estimation, more accurate forecasting and greater agility.

User stories are also the building blocks of larger Agile frameworks like epics and initiatives. Epics are large work items broken down into a set of stories, and multiple epics comprise an initiative. These

larger structures ensure that the day-to-day work of the development team contributes to the organizational goals built into epics and initiatives.

4 Available Resources

In the previous chapters, the project motivation, objectives and methodology are introduced. Also, the required background concepts related to software and product design are properly explained, which are followed up by the conducted project. For project proposal execution, there is the need of relying on different sources and tools: they can be literature sources, used for collecting information about unknown definitions, or they can be data sources, which provide the information or measurements needed to fulfill the project goals. In addition, there is a requiring list of software tools that the development team uses for developing the product planning, the product tracking, the product implementation and validation, and the product delivery to the end user.

Literature Sources

Specialized articles, websites and videos about how successful project management workflows are executed is a great starting point to refer. Project management is a complex career path which deals with the application of knowledge, previous experiences, tools, methodologies and proven workflows for achieving a specific goal in the frame of a product, application, or any deliverable planning, implementation and delivery to the end customer. Consulting valuable articles about recommendations for software development takes its time, so the key terminologies and definitions were collected for a proper understanding and application on the conducted project.

Data Sources

The product uses two main data sources: the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE), and the Blavatnik School of Government, a research and academic department of the University of Oxford. Both institutions provide the data in form of CSV tables which are easily readable by any chosen programming language package, along with written reports and user guides for data explanation. The data is available in their dedicated GitHub repositories, for which references [7] and [8] point to the data used on the conducted project.

- JHU CSSE data provides daily and cumulative information about Covid-19 cases, for confirmed, death and recovered patients. Starting from January 22, 2020, and covering 192 countries, the cumulative and daily data can be merged for providing interesting visualizations about how registered cases have been evolving across different countries.
- The Oxford COVID-19 Government Response Tracker (OxCGRT) provides the calculated Stringency Index (SI) across time on 185 countries, starting from January 22, 2020. As mentioned on the Chapter 2, the index is derived from the government responses about implemented regulations and restrictions on different aspects, like public transportation, events organization, workplace closures and travel measurements.

As for any data analysis project, the collected data runs a cleaning and filtering process, where special attention is put on non-available and missing data values, and that every column is independent from each other. Moreover, country names must match on the merging process, so that information does not get duplicated, altering the aggregation process. Data collection, preprocessing, aggregation and visualization steps are executed in a Python environment.

Software Tools

The conducted project relies its software part on the use of Python-oriented applications, software development platforms and version control systems for the code development, code maintenance, code update and tracking. The planning stage is also implemented using a website design workspace combined with a spreadsheet software desktop program.

The Jupyter Notebook computing platform is utilized for the software code development, or product implementation; this is the functional objective of the product. It runs over a Python-based environment, making the code development easy and understandable for any experienced user. The use of the Jupyter Notebook must be compliant with the two design principles mentioned as non-functional objectives. The code development is controlled via Git, software which can track changes and coordinating the non-linear workflow, including the running of parallel branches, each of them fulfilling a product feature or user story, and merging them at the end for product development. The developed product is available at the GitHub storing service, together with its proper user guide, and the product planning framework.

Product planning is achieved by choosing an Agile framework like Kanban or Scrum, as mentioned in the chapter 3. The use of a web-based application can be significant for building a solid product planning workflow: there are several web applications for product design and planning.

Whimsical is a recommended visual workspace ready to be plugged in for thinking and collaboration, for planning the project tasks applying the solutions like dedicated boards, backlog diagrams, flowcharts, and sticky notes. In addition, Microsoft Excel continues to be an optimum tool for any product design process, and it is also used for storing the tasks and subtasks across defined iterations, along with team roles and responsibilities.

All project steps are intended to be run in a local machine, where access to the Git repository is granted to the developing team and to the end user. The following bullet points specify the software tools or libraries involved at different steps of the project.

**Product Planning:* Microsoft Excel, Whimsical

**Product Updates:* Git

**Data Collection:* GitHub

**Data Preparation:* Jupyter Notebook – NumPy and Pandas Python libraries

**Data Statistics and Visualization:* Jupyter Notebook - Matplotlib-Plotly libraries, scikit-learn

**Data Proof of Use Case:* Jupyter Notebook, Web application

5 Product Methodology

Planning

The project planning kicks off with a brainstorming session for the product features. The product functional objective is clearly described in the first chapter, and then the product feature planning should be executed according to the user requirements.

A prioritization matrix containing the product feature ideas is constructed to have an overview of how complete the product features should be. The product has a functional feature which the end user requires, but additional enhancements could be done to enhance the product. The matrix is divided by effort and user value, displaying with features are considered important or not, and which features take more time to be completed.

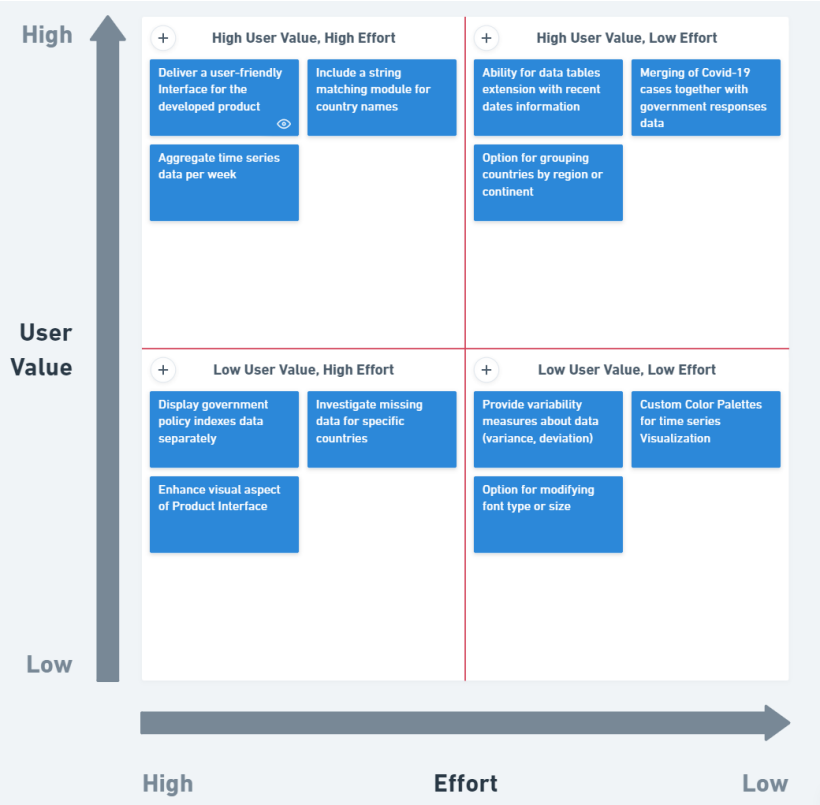

Fig. 3 Product Features Prioritization Matrix

Design

Implementation

---

6 Product Validation and Results

Product Results

Product Validation Tests

---

7 Bibliography

[1] Thomas Hale, Noam Angrist, Rafael Goldszmidt, Beatriz Kira, Anna Petherick, Toby Phillips, Samuel Webster, Emily Cameron-Blake, Laura Hallas, Saptarshi Majumdar, and Helen Tatlow, A global panel database of pandemic policies - Oxford COVID-19 Government Response Tracker. 2020.

[2} Dong E, Du H, Gardner L. An interactive web-based dashboard to track COVID-19 in real time. Lancet Inf Dis. 20(5):533-534. doi: 10.1016/S1473-3099(20)30120-1

[3] "COVID-19 Government Response Tracker." University of Oxford, 2022, https://www.bsg.ox.ac.uk/research/research-projects/covid-19-government-response-tracker

[4] "How to Plan Effective Software Development Projects." Waydev, 14 July 2021, https://waydev.co/software-projects-planning/

[5] "The SOLID Principles of Object-Oriented Design Explained.", 26 Apr 2022, https://www.freecodecamp.org/news/solid-principles-single-responsibility-principle-explained

[6] "What is Agile?", Atlassian, 2022,

https://www.atlassian.com/agile

[7]

https://github.com/CSSEGISandData/COVID-19

[8]

https://github.com/OxCGRT/covid-policy-tracker