## Project Overview

This is a tool to measure changes to Earned Value Management (EVM) based on common changes that may impact the data by either changing the schedule or cost. EVM is measured off a cost and schedule profile that is set as a baseline, and one that is updated as the current plan changes. In this example, we are going to assume it is for a pen shop that makes ballpoint pens in three sizes, fine, medium, and bold. The shop must manufacture a certain number of pens per month and to do so, all parts that are needed to assemble these pens must be bought and assembled in time to support the schedule. With many investors in the company, it is important that Pipa, the owner, can report on financial and schedule data as well as what the impacts will be for various changes. This tool will allow her to upload her data and then simulate various changes and the impacts it will have on the data she reports on.

GitHub Location: <https://github.com/CassieTracks/EVM-Impact-Calculator>

## Stakeholders

The primary stakeholders will be project managers that have to present on financial performance to customers which can be hard to understand. It is also challenging to understand long term impacts of decisions.

Secondary stakeholders are those receiving the analysis after the primary user evaluates changes through the tool and departments that want to integrate this into their operating system

## Why This Tool is Needed

Often changes occur to schedules and financial decisions need to be made as a result. Since funding is granted periodically throughout a project it is important for users to understand when those costs will be incurred. Often an increase in cost can impact a program in a different fiscal period, but it is challenging to know when that may happen. This tool helps users understand how slight changes impact their costs and schedules from a reporting perspective. It also acts as a training tool to help visually demonstrate concepts that are often misunderstood.

## Workflow

#### Upload Cost Profile file

The first file required is the cost and schedule profile. This would traditionally be created based on a demand schedule, the item details, and bill of material, but due to the complexity of the project we will assume that was done through a SQL database and will use a sample dataset as the output. An example of this dataset is below and shows the minimum required columns and specific column names that are required.

A table with numbers and letters

Description automatically generated

In this scenario, Pipa will have a file provided to her that includes her cost profile that she can use to assess different impacts

#### Upload Item Attributes file

The second file is the item details that will contain all the items used to manufacture each of the pens as demonstrated below in the minimum dataset required.

A white grid with black numbers

Description automatically generated

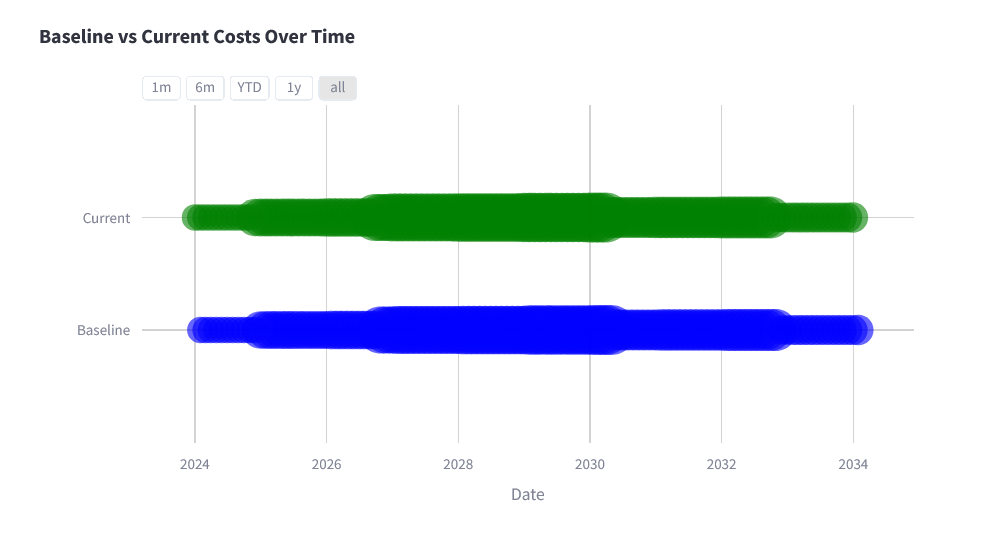
In this example, Pipa would receive this along with her cost profile as the assumptions used to generate the initial file.

#### Plot the data

Once the files are uploaded it will take the user to the next screen that will show two charts in different tabs based on how they prefer to see the data. The first will be a line chart that shows the data cumulatively over time for both datasets. It will also allow the user to hover over each datapoint to get a list of the estimated / calculated EVM metrics for that date.

### A graph with a line Description automatically generated

An additional chart will be provided to help users understand the time phasing of their impacts. This can help them see the distribution of costs over time rather than the cumulative impacts of the chart above. This helps them see when the majority of the costs will be incurred.



#### Modify Item Details

Users will have the option to make changes to the item details in the dashboard through slide bars. These will allow users to adjust the unit cost, lead time, yield, and labor hours for one item at a time. Originally this was going to include changes to demand, but since this is a complex operation managed by ERP systems, it was determined it would be out of scope for this project.

A white background with a blue and white object

Description automatically generated with medium confidence

The sliders will use the Streamlit extension as shown above. The starting value will be set to the default from the item attributes file. It will provide the user feedback by showing what impact it will cause on their data. For example, it would show that cost will increase overall material cost for that item by 20%

As the data is modified in the previous section, the dashboard will be visible and will update at the same time assuming this is not a limitation that is encountered. The amount of data already causes the charts to be slow to render, so it is possible the user will have a button to press that updates the charts instead. The other change is that instead of a dedicated frame for the calculations they will show up as the user overs over the chart. This is a Plotly function that has been proven to work very smoothly.

#### Evaluate impacts of Changes

Once the changes are made, the user will be able to scroll through the chart easily to see what calculations they can expect to see for each of those future months for the proposed change. The reason this is shown in text form is because the data they report on is calculated value and it is important they are able to tie the number value to what is happening in their data. This chart allows the transition from visual data into tabular data that is harder to understand.

### File Comparison

If time permits, users may want to evaluate impacts based on two data files rather than assessing manual changes. This could be added functionality where the user can upload a set of 4 files instead of having slider bars. This may add significant complexities to the code, and will depend on if there is time in the schedule to add this additional feature. The focus should be on creating a smooth interface with the basic functionality before adding nice-to-have features.

### Reset

The way the code is built, it will reset each time a new item is selected. The intent of the tool is to provide context into how impacts change data rather than generating a new data file with the changes.

### Export

The export function is going to be removed for this part of the code since it doesn’t fit with the intent of this being more of a training tool rather than an estimation tool and this will prevent misuse of the data created.

## Logic Flow

### Import data files

#### Import Item Details

This will be the function that takes the item details in and ensures it is in the right format for the code to process it. Will need to check that the correct column headers are there, and the data is formatted correctly

#### Import Schedule and Cost Details

This will be a similar function but will take the schedule and cost details as input instead.

### Filter Data

#### Take a user input of item number

Using the uploaded data, a list of items will be provided in a dropdown for a user to select that will then be used for the rest of the code

#### Filter the dataset

This creates a filtered dataset limited only to the specified part number.

### Calculate EVM Data

The Calculated fields used in the charts will be calculated based on the two comparison files. See the Formulas section for details on these formulas

### Update Dashboard

#### Plot Import Schedule and Cost Details

This uses Plotly Open-Source code to generate a line chart of the original data as well as the modified data set. Initially they will be the same data set. This will include trace annotations with EVM calculations for each time period.

### Take in User Inputs

#### Acknowledge data was updated

Any time any of the values change the code needs to execute on calculating the impacts

#### Calculate changes

For each changed value it needs to determine the original value of the input file, and the new input and calculate a percent change. This percent change will determine the impacts to the schedule and cost data. These impacts are shown below and need to be made for each line of data in the modified dataset.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Need Date | Qty | Monthly Cost |
| Yield | No Change | Qty \* %Change | Cost \* %Change |
| Cost | No Change | No Change | Material Cost \* %Change |
| Lead Time | Date + LeadTime Change | No Change | No Change |
| Hours | No Change | No Change | Labor Cost \* %Change |

#### Update dashboard

Run the same functions to originally update the dashboard each time assuming this is not too slow. If it proves to be too slow, there may have to be a button included to execute the code when the user is ready to see the impacts.

## Assumptions:

* The system will assume backward planning meaning the tasks will start in time to meet need dates for simplicity
* This assumes there are resources available to complete the work based on the needed schedule
* Costs may not be realistic but are used to simulate the data for the dashboard

## Formulas

The following formulas will be used in calculating the Earned Value Management (EVM) metrics. The following two formulas are summative and will not vary across time and will be impacted by changes in user inputs by either increasing and decreasing values or shifting the time phasing. These formulas represent the expected outcome for a project at the end once all costs are incurred and the schedule is completed.

Where:

Where:

Where:

The following formulas vary based on the current time which in this case will be by months that the user can select to understand how these metrics may change over time.

Where:

Where:

Where:

Where: