INEIGHT

Cost Curve Report

Tutorial

REPORT

Changelog

This changelog contains only significant or other notable changes to the document revision. Editorial or minor changes that do not affect the context of the document are not included in the changelog.

Rev	Date	Description
1.0	21-AUG-2026	Initial Release

©2026 InEight Inc. All rights reserved.

Contents

Overview **4**

APIs needed **4**

Power BI steps **6**

Data preparation 6

Data model 6

Canvas 7

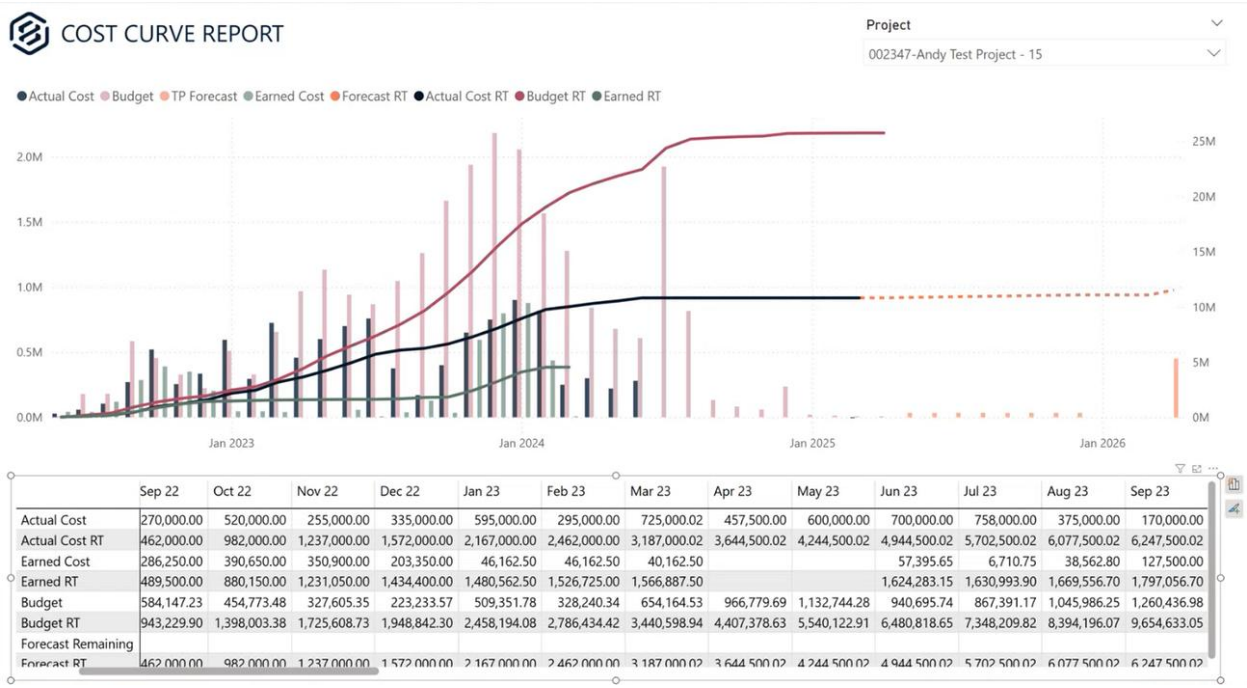
Project selector 7

Line + Clustered Column chart 7

Matrix 10

Overview

This document is intended to be used along with the Cost Curve Report tutorial video. In this document, you will find the DAX required to create the measures described in the video, as well as the logic behind the metrics. You will also find the items to prepare, model, and visualize the data necessary to build a cost curve report using the InEight Explore SelfService APIs.



	Sep 22	Oct 22	Nov 22	Dec 22	Jan 23	Feb 23	Mar 23	Apr 23	May 23	Jun 23	Jul 23	Aug 23	Sep 23
Actual Cost	270,000.00	520,000.00	255,000.00	335,000.00	595,000.00	295,000.00	725,000.02	457,500.00	600,000.00	700,000.00	758,000.00	375,000.00	170,000.00
Actual Cost RT	462,000.00	982,000.00	1,237,000.00	1,572,000.00	2,167,000.00	2,462,000.00	3,187,000.02	3,644,500.02	4,244,500.02	4,944,500.02	5,702,500.02	6,077,500.02	6,247,500.02
Earned Cost	286,250.00	390,650.00	350,900.00	203,350.00	46,162.50	46,162.50	40,162.50			57,395.65	6,710.75	38,562.80	127,500.00
Earned RT	489,500.00	880,150.00	1,231,050.00	1,434,400.00	1,480,562.50	1,526,725.00	1,566,887.50			1,624,283.15	1,630,993.90	1,669,556.70	1,797,056.70
Budget	584,147.23	454,773.48	327,605.35	223,233.57	509,351.78	328,240.34	654,164.53	966,779.69	1,132,744.28	940,695.74	867,391.17	1,045,986.25	1,260,436.98
Budget RT	943,229.90	1,398,003.38	1,725,608.73	1,948,842.30	2,458,194.08	2,786,434.42	3,440,598.94	4,407,378.63	5,540,122.91	6,480,818.65	7,348,209.82	8,394,196.07	9,654,633.05
Forecast Remaining													
Forecast RT	462,000.00	982,000.00	1,237,000.00	1,572,000.00	2,167,000.00	2,462,000.00	3,187,000.02	3,644,500.02	4,244,500.02	4,944,500.02	5,702,500.02	6,077,500.02	6,247,500.02

APIs needed

- Dimension tables (descriptive information)
    - SelfService_Core_Project
    - SelfS_Control_CostItem
    - SelfService_Calendar
        - This pulls from the Fiscal calendar in Core/Platform. Changes to the fiscal calendar will impact your reporting.
- FACT tables (quantitative information)
    - Actuals Cost/MHR/Quantity (calc earned)- MHRs are not needed unless you are building a MHR curve)
        - You will need the Qty table to calc earned value:
            - **Earned value** = (% Complete * CB Total Cost) **OR** (Qty Complete to date * CB Unit Rate)

- **% Complete** = Qty complete to date / Forecast TO Qty
- **CB Unit rate** = CB Total Cost /Forecast TO Qty

- This is used to plot our earned value against a date dimension
- Qty complete to date has the date stamp

- SelfService_Control_CurrentBudget - for unit rate
  - Can use CB or CE - unit rate comes from one of these, you can use these interchangeably based on how you want to display earned as a factor of CB or CE, for this example CB is used.
  - These tables have the MHR unit rate as well

- Explore Timephased *Forecast*
- Explore Timephased *Budget*

- IsTerminal vs IsEnabled
  - Most customers track cost at the superior level. You must be able to determine superior cost items that have been enabled for cost tracking, this methodology will require the IsEnabled field.
    - When organizations are tracking cost on certain branches of their CBS at the superior level, forecasted values are time phased at that superior level.
    - IsEnabled is available in SelfService_Control_CurrentProjectForecastDetail.
    - You will also need forecastMethod when <> Rollup.
  - Otherwise, you would be able to simply use IsTerminal (In the SS Control Cost Item = True built into measures, filters, etc..)

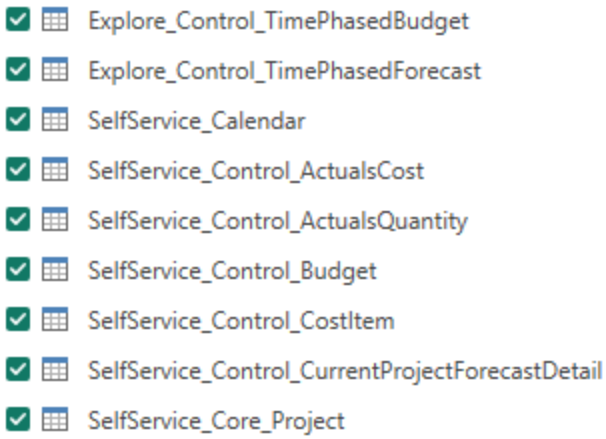


Figure A

Power BI steps

Data preparation

- Performance management - Power Query > Edit mode
    - Uncheck autodetect relationships
    - Remove any columns that you do not need. This will greatly impact performance of your dashboard.
    - Filter all selfservice views to isactive = True
    - SelfService_Calendar, transform the following fields to Date Only, the datetimestamp will cause a mismatch on some of the joins and data will be missing.
        - Calendar_date
        - First_of_month

Data model

- Start with Project; project will filter the cost item
    - Join Project to Cost Item on Project id, one to many single directional filter
- Cost Item to ActualsCost
    - Cost Item ID, one to many single directional filter
- Cost Item to Quantity
    - Cost Item ID, one to many single directional filter
- Cost Item to TimePhasedForecast
    - Cost Item ID, one to many single directional filter
- Cost Item to TimePhasedBudget
    - Cost Item ID, one to many single directional filter
- CurrentProjectForecastDetail to cost item
    - CostItemId, one to many single directional filter
    - This is for performance reasons
- CurrentBudget to Cost Item
    - CostItemId, one to many single directional filter
- ActualCost (manhour and Qty are the same) to calendar

- PostedDate to Calendar_Date, many to one single directional filter

- TimePhasedBudget to Calendar
  - TimephasedPeriod to calendar_date, many to one single directional filter
- TimePhasedForecast to Calendar
  - TimephasedPeriod to calendar_date, many to one single directional filter

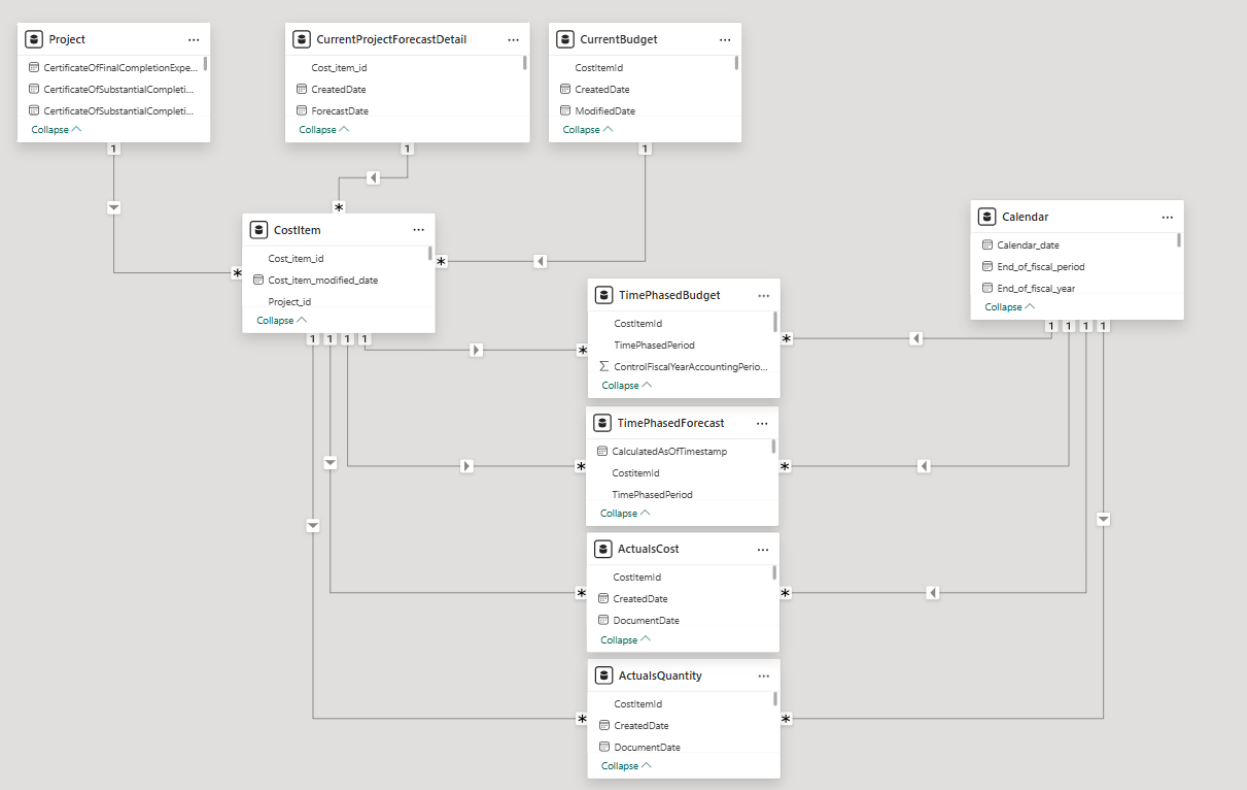


Figure B

Canvas

Project selector

- Slicer on Project.Project
- Make it a dropdown
- Ctrl + F adds search bar in the dropdown

Line + Clustered Column chart

The bars must be grouped, not stacked.

X axis

First_of_month – You can change the format in the column tools ribbon to "mmm-yy" or whatever you visually prefer.

Column y-axis

- Sum of ActualsCost.Cost > Rename "Actual Cost"
- Create Earned Value > Rename " Earned Cost"
    - ActualsQuantity > New measure
    - Earned Cost = SUMX(ActualsQuantity, related(CurrentBudget[CBUnitCost])*ActualsQuantity[Quantity])
        - Every transaction is its own record, therefore SumX is used. Related is the lookup function in DAX.
- Create Budget, but you need to filter on IsTerminal and you only want to filter on this table, so you need to create a measure
    - TimePhasedBudget > New Measure
    - Budget = Calculate(Sum(TimephasedBudget[Cost]), keepfilters(CostItem[Is_Terminal]=True))
        - Calculate is used because you are modifying the filter context.
        - Keepfilters is used because you do not want to filter the entire table.
- Do the same with TP Forecast (OR Forecast Remaining)
    - TP Forecast > New Measure
    - TP Forecast = Calculate(Sum(Timephasedforecast[Cost]), keepfilters(CurrentProjectForecastDetail[IsEnabled]=True), keepfilters(CurrentProjectForecastDetail[Forecast_Method] <> "Rollup"))
        - *Use calculate because you are modifying filter context*
        - *If you are certain that you are only tracking cost items at the terminal level, use the following instead:*
            - TP Forecast = Calculate(Sum(Timephasedforecast[Cost]), Keepfilters(Timephasedforecast[IncludeInSummary] = True))

Line y-axis

You need to create running totals from the measures created above, utilize "..." > New visual calculation > Custom

- Lines need to be plotted on the secondary y-axis (1:05)
    - Format > Visual > Secondary y-axis
        - Values > ON
        - Align Zeroes > ON
- Actual Cost RT = Runningsum([Actual Cost])
- Budget RT = Runningsum([Budget])
- Earned RT = Runningsum([Earned Cost])
- TP Forecast - Because the actuals and the TP forecast will need to meet, and the forecast will continue on with a dashed line, as an industry best practice, they need to be combined.
    - Forecast incl Actuals = [Actual Cost] + [TP Forecast]
    - Add this to the column-y-axis, then hide it because at the column level it is redundant. For visual calculations to work, they need to be used in the canvas, and you will need to use this field to make another field.
    - Then take the running total TP Forecast RT= runningsum([Forecast incl Actuals])
- You need to make sure Actuals/Budget/Earned lines stop where the values do:
    - Edit Actuals RT
        - Actuals RT =

```
IF (
    NOT ISBLANK ( [Actual Cost] )
        && [Actual Cost] <> 0,
    RUNNINGSUM ( [Actual Cost] ),
    BLANK ()
    )
```

    - Edit Budget RT
        - Budget RT =

```
IF (
    NOT ISBLANK ( [Budget] )
        && [Budget] <> 0,
    RUNNINGSUM ( [Budget] ),
    BLANK ()
    )
```

- Edit Earned RT
  - Earned RT =

```
IF (
    NOT ISBLANK ( [Earned Cost] )
        && [Earned Cost] <> 0,
    RUNNINGSUM ( [Earned Cost] ),
    BLANK ()
    )
```

- Make Forecast dashed: Format > Visual > Lines > Select TP Forecast RT > Line Style: select dashed.

Matrix

Add a matrix visual below the chart to display values.

- Copy the graph and make it a Matrix visual
- Move First_of_month to columns
- Format > Values > Values > Options >Switch values to rows "On"
- Rearrange the RT to come after the regular values