EVM Data Visualization:
The Technomics Radar Tool (TRT)

Tyler Staffin and Brian Flynn

Abstract

As the world continues to advance technologically, the sheer quantity, complexity, and availability of raw data threatens to overwhelm the user. The art and science of data synthesis and visualization is therefore as imperative now to a program office as the analysis itself. If data are not synthesized and visualized in a meaningful fashion, they cannot be expected to be treated appropriately – to yield real insights. This research presents an innovation in EVM data analytics and visualization that gives cost analysts and acquisition managers the potential to better understand system level risk and to laser-focus their attention on critical elements of cost. The tool transforms a massive set of disaggregated data into a manageable form, and extracts critical elements by filtering on a set of user-defined, rules-based criteria. The back-end data processing for the Radar Tool is powered by R, with its capability of harvesting, manipulating, and formatting incredibly large datasets efficiently. After initial cleansing, massive quantities of data are pulled into a front-end, user-friendly dashboard wherein critical patterns, insights, and intelligence are gleaned. The tool holistically displays projected growth in hours, percent growth in hours, and percent contract complete in one radar plot. The tool can be further customized by its users to accurately depict program-specific data.

Problem

The Department of Defense (DoD), during Robert McNamara's tenure as Secretary of Defense in the 1960's, adopted Earned Value Management (EVM) based on an evaluation of nascent commercial systems that were showing promise and yielding insights.[1] The first EVM systems had been started a decade earlier by major corporations seeking ways to forecast total funding requirements from project inception to completion, with the intent of identifying problems and preventing unpleasant surprises. Forecasting cost and budget overruns had the salutary effect of gauging project status – progress against tasks – well before the midpoint of a project. DoD adopted the early success/failure predictive capability part of EVM, with Gary Christle and Wayne Abba spearheading the effort in the Pentagon for more than a decade. EVM's goal was to provide timely, reliable management data to senior acquisition officials for defense contractor performance on complex projects.

Today, EVM is a half-century old in the Department and pervasive in its implementation. Mandatory reporting is required for contracts and subcontracts that meet minimum threshold dollar values, as prescribed in DOD Instruction 5000.02, Operation of the Defense Acquisition System.[2]

Nevertheless, problems with EVM remain, as gleaned by the authors in support of several acquisition programs:

- <u>Simplistic Perspective</u>. There's often too much emphasis placed on basic rote calculations and mnemonics as keys to project success. The math is easy. But the resulting metrics are useful only if a valid accounting structure is in place to provide the true status of what is being measured. Unfortunately, the authors found that not all costs are always reported. For whatever reason, EVM data for selected Contract Line Item Numbers (CLINs) are sometimes either missing or nonsensical, at least on selected contracts.

- <u>Inadequate Governance Structure</u>. As a corollary, EVM achieves its ultimate purpose only if a governance structure is in place whereby individuals and organizations accept responsibility for explaining problems and taking corrective action. Too often there's too little emphasis placed on root cause analysis and the critical role played by the Cost Account Manager, and his/her enforcement of the 32 criteria for cost/schedule control.

---

[1] "EVM originated in Department of Defense (DoD) policy as Cost/Schedule Control Systems Criteria (C/SCSC or CS2) in 1967, and was at the core of an emerging concept known as Integrated Program Performance Management (IPPM)." Abba, Wayne, "The Evolution of Earned Value Management," Defense AT&L Magazine, DAU, 1 Mar 2017.

[2] EVM compliance is currently required on cost or incentive contracts, subcontracts, intra-government work agreements, and other agreements valued at or greater than $20 million. An EVMS that has been formally validated by DCMA and accepted by the cognizant contracting officer is required on cost or incentive contracts, subcontracts, intra-government work agreements, and other agreements valued at or greater than $50 million. Source: Performance Assessments and Root Cause Analysis (PARCA), USD(Acquisition and Sustainment), 2019.

- Moving Target. EVM has been likened to Einstein's Theory of Relativity. That is, the speed at which an object travels or progresses is relative to the position of the observer. All projects, especially the big, complex ones, progress at different speeds. It's essential, then, to understand the underlying dynamics of the acquisition. Early on, common EVM metrics such as the Cost Performance Index (CPI), Schedule Performance Index (SPI) and Estimate at Completion (EAC) are often misleading or even meaningless.

- Overwhelming Detail. Reams of detailed EVM data can and do overwhelm the user. Over time in the Department, there's been a growing demand for more "accurate, timely" data, with emphasis on contractor compliance enforced by punitive contractual provisions. Unfortunately, the volume of data now available from modern accounting systems, if not synthesized and understood, can confuse rather than illuminate project performance issues. EVM's original purpose, a timely, reliable management information system, can be compromised to the extent it is redefined as an audit-oriented oversight system with excruciating detail that misses ways to improve performance.

- Poor Visualization. Along the same lines, failure to cogently and clearly present results of EVM analysis inhibits corporate and government decision makers from understanding critical programmatic issues and taking corrective action. This may partly explain why project budgets continue to overrun in too many instances.

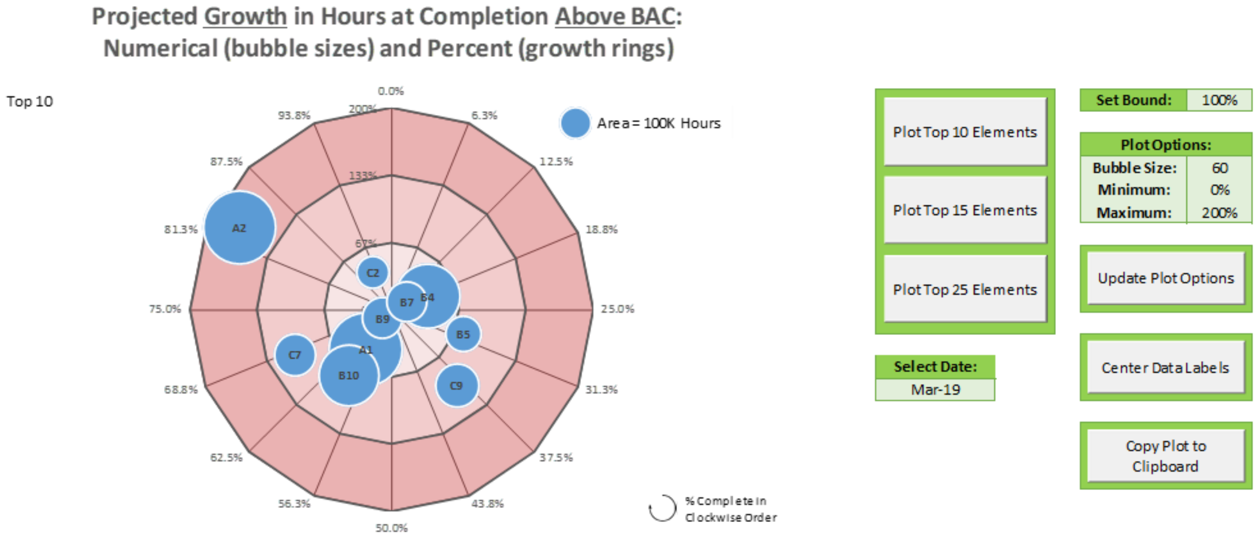
Solution Overview

Technomics' multi-faceted solution to ameliorating some of these problems entails these interrelated steps:

- Developing techniques in R to automatically load hundreds of thousands of raw data elements into a simplified dataset
- Developing a set of user-defined criteria for selecting "problem" elements in the WBS and computing key EVM metrics and trends
- Developing a Radar Plot that visualizes the results of the analysis, as shown in Figure 1, with increasing level of detail available for sub-elements down to the level traceable to a Cost Account Manager

The discussion that follows addresses these steps and the resulting innovation in EVM data analytics and visualization that advances the state-of-the-art and serves as a benchmark for excellence!

Figure 1: The Radar Plot



(Note: BAC denotes Budget at Completion.)

Solution Details

Our solution to some of the problems (principally "overwhelming detail" and "poor visualization") is the Technomics Radar Tool (TRT). The following subsections describe the building blocks of the tool, starting with some important context relative to the data itself.

Hypothetical EVM Dataset

In this paper, the tool is explained using a hypothetical, large EVMS dataset for an acquisition program, "Program Z." The hypothetical dataset nevertheless conforms to the standards and norms of earned value reporting. Program Z, like every defense project, is a combination of organized activities performed under the scope of one or more contracts. Thus, the total amount of work performed across the entire project directly equates to the sum of the work performed under each unique activity. This holds for the duration of the contract. Symbolically, let A, B, and C denote three main Products included in the scope of Program Z, and let a, b, and c denote the amount of work required to complete Products A, B, and C, respectfully.

Thus z, the total amount of work to be performed under the defined scope of Program Z is the sum of all of the work performed across each of the activities that constitute the project. Hence, the total amount of work performed to complete Product A is the sum of the work performed under each Account, A_1, A_2, A_3, etc., with the same conditions applying to Products B and C. Mathematically, this can be expressed as:

$$z = \sum_{p=1}^{k} a_p + \sum_{q=1}^{m} b_q + \sum_{r=1}^{n} c_r$$

where k, m, and n are positive integers corresponding to the number of tasks under each unique Account. For actual Major Defense Acquisition Programs (MDAPs), the magnitude of $k + m + n$ often exceeds 100,000 tasks. This significant volume of tasks becomes the foundation of an extremely large EVM dataset for a program because every EVM report for a given contract includes a variety of important metrics associated with each task and is typically generated on a monthly basis!

The specific reported metrics of interest for our work include:

- ACWP: actual cost of work performed on given task
- BCWP: amount budgeted for the work performed on given task
- BCWS: amount budgeted for the work scheduled for a given task
- BAC: current defined budget of the task (originally determined at program baselining)

These metrics are used to compute a couple other metrics of interest, specifically performance indices:

- CPI: calculated by taking BCWP/ACWP, a value greater than 1 indicates a task is currently under budget and less than 1 indicates a task is currently over budget
- SPI: calculated by taking BCWP/BCWS, a value greater than 1 indicates a task is currently ahead of schedule and less than 1 indicates a task is currently behind schedule

The preceding discussion provides clear evidence of the "overwhelming detail" problem that has been an obstacle to analysts and, perhaps more importantly, managers at different levels of a program. Put simply, in the past, the sheer volume of data posed data management and manipulation challenges that were either beyond the capability of the organizations/individuals involved or deemed too costly relative to the perceived value to address.

Additional evidence of the "overwhelming detail" problem is the fact that without the right data analytics capability it is extremely difficult if not impossible to discern *what's really important* in the data, i.e., what the data is (or not) telling users at different levels of detail in an EVM dataset. Not surprisingly, the lack of effective visualization capability (i.e., the "poor visualization" problem) bears directly on effective identification of cost and/or schedule performance problems that warrant special attention.

The next two subsections detail how we cleverly used the power of R and MS Excel to develop back-end and front-end solutions that have enabled cost analysts and managers to exploit very large datasets and better understand *what's really important*.

TRT Back-end

The back-end data processing for the Radar Tool is powered by R, with its capability of harvesting, manipulating, and formatting incredibly large datasets efficiently. R has the ability to efficiently perform row-wise and column-wise operations on a multitude of data structures. While this type of data processing and manipulation can be executed in Excel, it can become challenging when handling larger datasets – especially in the context of EVM. Excel is limited to slightly over one million rows, so efficient data manipulation becomes a problem as datasets grow larger in size. The hypothetical EVM dataset submitted by contractors contains a fairly extensive amount of data, so R will be a key component in the generation of the task-level breakdown needed to populate the Radar Tool.

- R is a programming language designed for statistical analysis and graphics development
- Additional capabilities have been added to R in the realm of data management and manipulation, dashboard development, database interaction, and machine learning capabilities
- The R language is thoroughly maintained and regularly improved by an international community of programmers and data scientists
- The R syntax is constructed in a simple fashion, yet can handle complex and detailed directions from users

The base R language contains a wide range of functions that can handle detail-oriented manipulation. Contributing members of the R community have developed packages that emphasize more efficient approaches to data management and manipulation than the default methods included with the software. One resource relied upon by data practitioners is the `dplyr` package. A part of the package group known as the `Tidyverse` (developed by Hadley Wickham, Chief Scientist of RStudio), `dplyr` contains a holistic grammar of functions and capabilities that expand the horizons of data manipulation in the R environment. For the TRT EVM dataset, this includes use of some of the basic `dplyr` syntax, such as `%>%` (the forward-piping operator), `filter()`, `mutate()`, `group_by()`, `arrange()`, `summarize()`, and `select()`. The piping operator and other listed functions achieve the following purposes:

- `%>%`: takes the data frame preceding the operator and passes as an argument in the next line of code
- `filter()`: filters data to rows that match provided criteria
- `mutate()`: creates a new data frame column based on provided criteria
- `arrange()`:indicates the row-sorted order of the data in data frame
- `group_by()`: prepares columns of a data frame to be aggregated by grouped factors
- `summarize()`: indicates operations to be performed over grouped factors
- `select()`: filters and orders data frame columns to selected variable

Further, datasets should be imported into R as CSV (comma-separated values) files (which are easily created in Excel, assuming that there exists no special formatting structures such as merged cells, tables, grids, etc.). R includes a variety of packages that are readily available to the user. One of these, the `read.csv()` function, takes a CSV file as input and stores it as a data frame in the R environment.

Data frames are structures defined in R's base syntax documentation as "tightly coupled collections of variables which share many of the properties of matrices and lists, used as the fundamental data structure by most of R's modeling software." To some extent, data frames are similar in structure to tables seen in Excel. In this case, the imported data is assigned to the data frame variable `dataEVM`. Now, the data is aggregated over the `Activity` variable (and creates the input for the Radar Plot) by running the following script:

```
dataEVM <- read.csv("ProjectZ_EVMS_012019.csv")

dataEVM.output <- dataEVM %>%
        filter(Contract = "Phase III") %>%
        arrange(Product, Account, Task, SubTask, Activity) %>%
        group_by(Activity) %>%
        summarize(BAC = sum(BAC), BCWS = sum(BCWS), BCWP = sum(BCWP), ACWP = sum(ACWP)) %>%
        select(ReportMonth, Contract, Account, Task, SubTask, BAC, BCWS, BCWP, ACWP)
```

In simplified terms, the data is filtered down to the observations pertaining to the current contract, and the data is sorted by `Product`, `Account`, `Task`, `SubTask`, and `Activity`. R is told to treat `Activity` as a grouping variable (so following numeric operations are be performed over that variable), and calculate total BAC, BCWS, BCWP, and ACWP accordingly. Finally, `ReportMonth`, `Contract`, `Account`, `Task`, `SubTask`, `BAC`, `BCWS`, `BCWP`, and `ACWP` are selected, and the resulting data frame is assigned to the variable `dataEVM.output`. Since monthly EVM datasets are reported in a consistent structure, future datasets can be imported and processed through the same script to achieve the necessary output. While this is a highly simplified case of data manipulation in R, it provides the general framework for the procedure related to the use of the Radar Tool.

TRT Front-end

The tool transforms a massive set of disaggregated data into a manageable form, and extracts critical elements by filtering on a set of user-defined, rules-based criteria. After initial cleansing, massive quantities of data are pulled into a front-end, user-friendly dashboard wherein critical patterns, insights, and intelligence are gleaned. The tool holistically displays projected growth in hours, percent growth in hours, and percent contract complete in one radar plot. The tool can be further customized by its users to accurately depict program-specific data.

The TRT front-end is an Excel macro-enabled workbook that consists of four components:

- Raw Data entry worksheet
- Data Processing worksheet
- Plot Formatting worksheet
- Radar Plot

The Raw Data worksheet solely consists of the output data from the R-powered back-end. Each monthly dataset is added sequentially to the bottom of the existing table that includes all previous months' data.

The Data Processing worksheet contains formulas to further aggregate the data at the Account Level, as well as EVM performance index metrics, calculated EACs, and percent complete. The principal purpose of this worksheet is to calculate the estimated cost variance at complete (EAC CV) for each Account in the dataset. This value is computed as the difference between the EAC (in this case, using the traditional "Gold Card" CPI x SPI approach) and the BAC for each Account. The EAC CV provides program managers insight into the areas most likely to contribute to cost growth for the duration of the project, assuming current cost and schedule performance remains constant. A corresponding ranking of EAC CV is calculated within the table (in which the worst performers are the highest in the ranking scale).

The Plot Formatting worksheet (Figure 2) bridges the gap between the raw data and the Radar Plot, the latter of which is TRT's centerpiece and users' avenue to actionable insights into contractor cost and schedule performance. This part of the tool consists of two automated

Figure 2: Plot Formatting Worksheet

Rank	Account	% Complete	EAC	|EAC CV|	Relative Growth (Nominal Hours)	Bubble Diameter	Θ (Includes π Radians Rotation)	Cost Growth Factor (CGF)	X-Coordinate (CGFcosΘ)	Y-Coordinate (CGFsinΘ)
1	A1	32.7%	1,593,390	529,663	5.30	0.575	-0.484	0.498	0.441	-0.232
2	B10	32.8%	860,693	411,176	4.11	0.507	-0.491	0.915	0.807	-0.431
3	B4	11.0%	1,433,545	401,922	4.02	0.501	0.879	0.390	0.249	0.300
4	A2	57.5%	635,671	344,560	3.45	0.464	-2.042	1.184	-0.538	-1.054
5	B9	34.7%	1,511,877	229,534	2.30	0.379	-0.607	0.179	0.147	-0.102
6	C9	21.5%	394,394	208,235	2.08	0.361	0.219	1.119	1.092	0.243
7	C7	37.7%	347,012	185,097	1.85	0.340	-0.800	1.143	0.797	-0.820
8	B7	10.0%	1,098,247	166,477	1.66	0.323	0.944	0.179	0.105	0.145
9	B5	16.9%	309,205	142,191	1.42	0.298	0.510	0.851	0.743	0.415
10	C2	51.5%	373,969	117,209	1.17	0.271	-1.664	0.456	-0.043	-0.455
Legend:	Area = 100K Hours		Legend Size:	100,000	1.00	0.25	Legend Coordinates:		1.38	1.38

% Dropdown:
60%
70%
80%
90%
100%

Plot Text:
Top 10
Plot Max %:
60%
Total % Options:
5
Total Dates:
3

Dates Dropdown:
Jan-19
Feb-19
Mar-19

Plot Control Panel

PC Bound:
100.0%

% Labels:
0.0%
6.3%
12.5%
18.8%
25.0%
31.3%
37.5%
43.8%
50.0%
56.3%
62.5%
68.8%
75.0%
81.3%
87.5%
93.8%
100.0%

Styling Control Panel:						
Background 1	Background 2	Background 3	Border 1	Border 2	Border 3	Border 4
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5
1.25	0.75	0.25	0	0.5	1	1.5

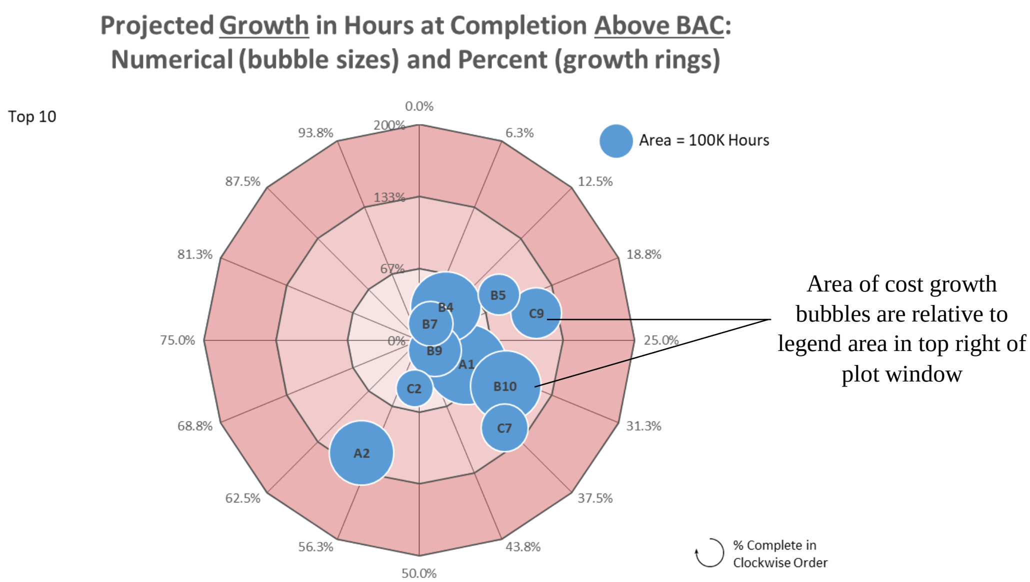
Circle Size Parameters:	
Nominal Hours	100,000
Area	0.049
Diameter	0.25

Styling Control Panel

sections – the plot control panel and the styling control panel. The plot control panel calculates the relative sizing of each of the bubbles on the plot (illustrated in Figure 3) as a factor of both user-set circle sizing parameters and the absolute value of the given Account's EAC CV. The bubbles are used as a visual tool for the user to rapidly identify elements undergoing significant cost growth.

Since the scaling of cost growth is relative to a fixed value, a user has the ability to determine the relative scaling factor for the plot. For example, if a user enters 100K as the legend size, all of the bubbles in the plot are constructed relative to the size of that input. Thus, the relative growth value is calculated by taking the ratio of the projected cost growth to 100K. This value is transformed into a corresponding diameter based on a user-set key diameter.

Figure 3: Radar Plot Cost Growth Bubbles



The location of the bubble in the plot is dependent on the current percent complete of the Account, with an additional rotation of -π/2 radians (to set the 0% mark at the intersection of the first and second quadrants of the Cartesian coordinate axis). Since this plot is constructed in terms of polar coordinates, the precise location of each bubble about the Cartesian axis is determined by the following trigonometric transformation:

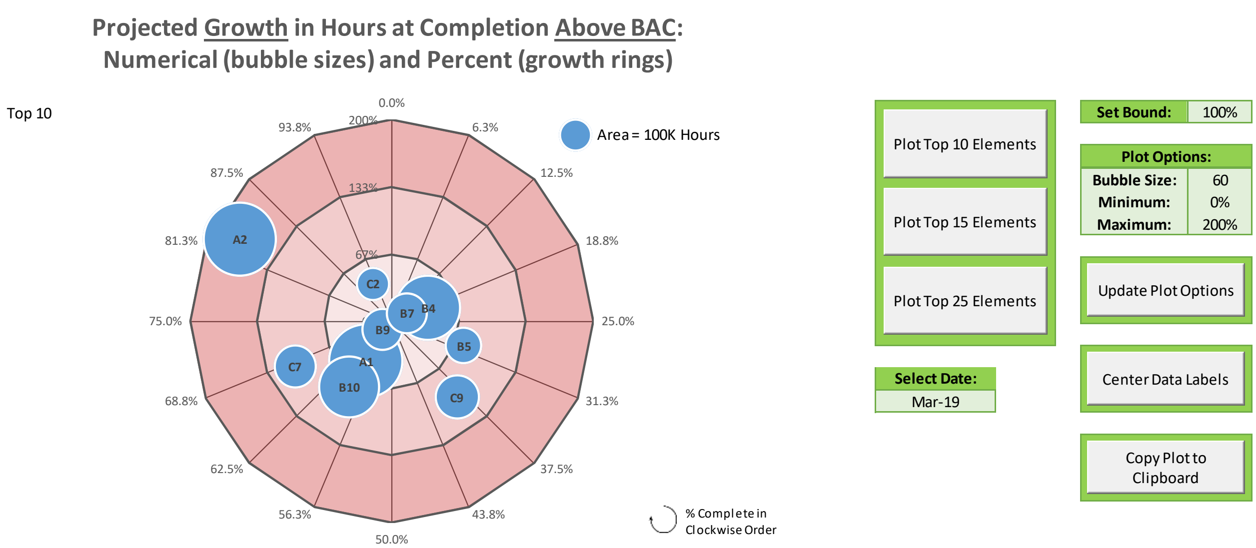
$$\begin{cases} X = r \cos \Theta \\ Y = r \sin \Theta \end{cases}$$

The value of r is determined by the Cost Growth Factor (CGF), and the value Θ is determined by the current percent complete normalized to a user-set percent complete upper bound. These transformations are handled automatically by Excel upon addition of data. There is also a set of coordinates and sizing parameters for the legend bubble. The location of the legend is fixed (regardless of the percent complete boundary or the growth ring scale) and bubble size is scaled automatically by a combination of the other bubbles on the plot and user-determined options. The number of rows seen in the plot control panel (Figure 2) stems from the number of elements the user chooses to include in the plot (based on 'Plot Elements' buttons depicted in Figure 4).

The second section of the Plot Formatting Worksheet is the styling control panel. The values here indicate the coordinates of the background colors and borders that make up the structure of the Radar Plot. If a user decides to modify any component of the axis scaling of the plot, these values are automatically updated.

Additionally, there are a small number of accessory tables and lists in this worksheet. These components mostly pertain to dropdown box options on the Radar Plot worksheet, but also control some other plot formatting components. All of the pieces of this worksheet function seamlessly to display the data on the Radar Plot in the crisp and highly customizable manner shown in Figure 4.

Figure 4: The Radar Plot User Interface



The Radar Plot displays the following key variables of the worst performing Accounts:

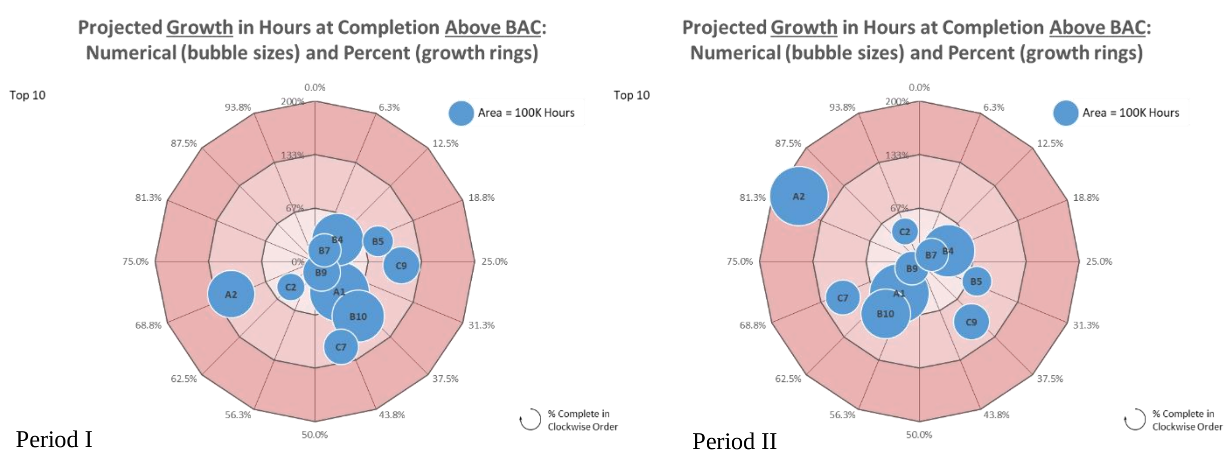
- Percent complete
- Projected growth in hours above completion (EAC CV)
- Expected percentage in growth above the current BAC: determined by the clockwise position around the circle, percent complete is easily identified for each element

The physical size of the bubble corresponds to the calculated EAC CV, and is scaled relative to the size and value of the plot legend. The outward progression from the center depicts the percent growth of the EAC over the current BAC (cost growth factor). The elements with a higher percent complete (i.e., larger bubble size) and greater distance from the center are drivers of poor program performance. Barring any improvements in performance, these are the areas likely to continue pushing the EAC in the wrong direction.

Users are provided flexibility in the number of visible elements. On the right of the plot, a user has the option to plot the top 10, 15, or 25 worst-performing accounts with the push of a button. The date selection drop-down feature is used to rapidly cycle through available performance periods, providing a time-series view of Account performance. A number of additional plot scaling options can also be changed to ensure that the data is appropriately captured (all changes are taken into effect after pressing the 'Update Plot Options' button). If data labels on the plot are manually moved (for visual clarity), this action can be reset by toggling the 'Center Data Labels' button. Lastly, the entire plot is copied by selecting the 'Copy Plot to Clipboard' button.

Figure 5 displays the time-series feature of the Radar Plot. Between these two periods, it is evident that there is not a significant amount of change in bubble size for the majority of the elements, except Account A2. There is approximately a 20% increase in level of completion from the first to second period, as well as a noticeable increase in the EAC CV. Because it is

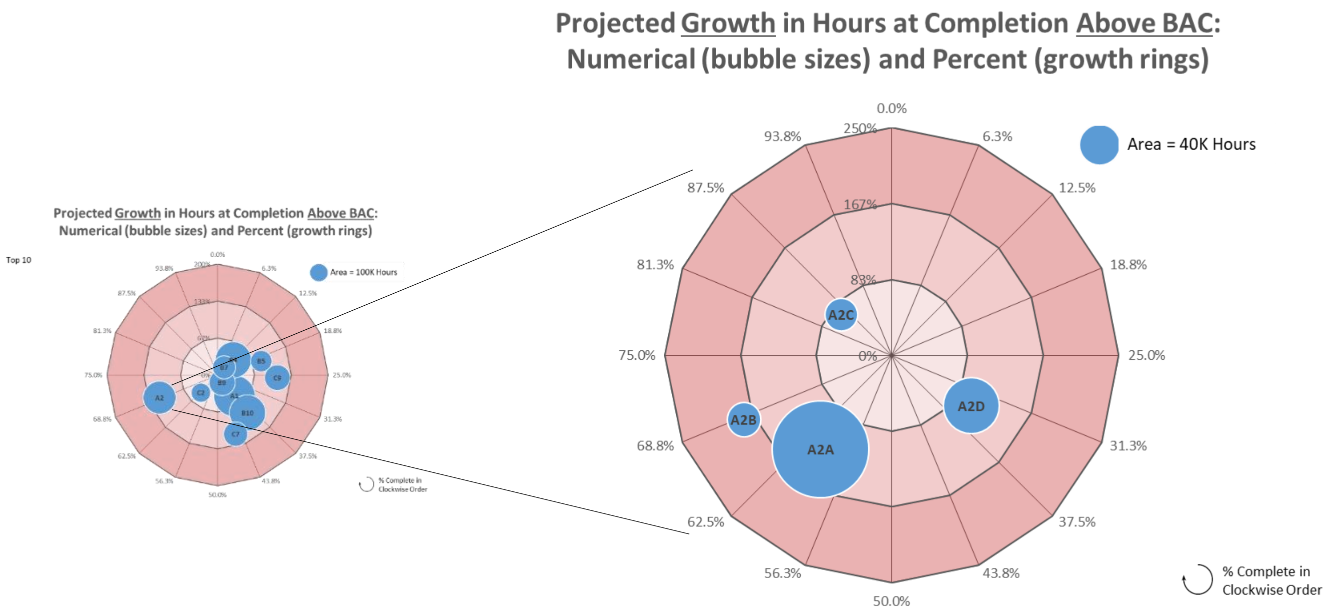
Figure 5: Visualizing Cost Growth over Sequential Periods



demonstrating an increasing amount of expected cost growth at a fairly high percent complete (thus has less opportunity to recover), this account warrants management investigation of the root causes of the poor performance.

A future tool enhancement will include another layer of automated insight, permitting users to look closer at the lower level components underlying each Account. Figure 6 depicts a prototype of this future enhancement.

Figure 6: Account A2 Task Deep-Dive



Conclusion

EVM has traditionally suffered due to several problems: 1) simplistic perspective (e.g., missing or bad data for certain CLINs); 2) inadequate governance structure (e.g., failure to accurately capture all costs in a contract); 3) moving target (e.g., values change quickly); and 4) overwhelming detail and poor visualization.

Program managers and their support staffs are faced with an overwhelming amount of detail. EV datasets often contain thousands of rows of information at the task level. The detail is daunting. Without automated tools, it's difficult to understand what's included and what may be missing in the files. Our current application of the TRT in support of a major defense acquisition program indicates that the sum of details presented in an EVM data dump for a particular contract do not always match summary-level totals reporting in other documents for the same program. This discovery has prompted improvements in the EV datasets and in contractor governance procedures.

Program managers need to quickly identify elements in the WBS that are trending towards an irreversible path of cost growth. This cannot be accomplished by visually examining reams of EVM data; the volume of information is simply overwhelming for human consumption. The TRT automatically sifts through massive datasets. If an element is experiencing significant cost growth at an early stage of the program, it will appear on the Radar Plot as a larger element in the top-right section of the plot. This visibility prompts a manager to look deeper into the lower-level drivers of performance for that Account in an effort to understand and mitigate potential risk factors that could adversely impact cost or schedule performance. Any Account that appears on the Radar Plot implies some level of EAC CV, but it is at the discretion of the manager to determine which areas are the most critical to the success of program. The tool's time-series functionality makes it easy to track the growth, appearance, or disappearance of Accounts as the program progresses, enabling manager visibility into the success of performance recovery efforts.

Program managers bear the fiduciary responsibility of effectively managing risks inherent in the acquisition of highly complex defense systems. Problems inevitably arise. As Kerby and Count claim, "… in many instances there is no forewarning; schedules slip, costs soar, and the project manager is faced with the near impossible task of explaining why each impact occurred."[3] EVM, at its best, tracks program performance in a consistent and manageable fashion. But, it doesn't readily identify root causes and doesn't routinely present information holistically. Therefore, synthesis and visualization of EVM data is an essential requirement for effective program management. The innovative TRT will forewarn managers of potential critical problems and enable a critical shift in behavior – from *reaction* to *prevention* of the show-stoppers that all-too-many acquisition programs have experienced.

[3] Kerby and Counts, Stacy. *The Benefits of Earned Value Management from a Project Manager's Perspective*. https://www.nasa.gov/pdf/293270main_63777main_kerby_counts_forum9.pdf