



SYSTEMS ENGINEERING  
Research Center

# Development and Application of FACT Portfolio Management Capability

Technical Report SERC-2015-TR-047-1

January 20, 2015

Principal Investigator: Daniel C. Browne, Georgia Institute of Technology

Research Team:

Todd Shayler

Melissa Rost

Devraj Mehta

Sponsor: Marine Corps Systems Command – FACT PM

Copyright © 2015 Stevens Institute of Technology, Systems Engineering Research Center

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-13-3-0004 (Task Order 0012).

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

No Warranty.

This Stevens Institute of Technology and Systems Engineering Research Center material is furnished on an “as-is” basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

This material has been approved for public release and unlimited distribution.

TABLE OF CONTENTS

---

**Table of Contents** ..... 3

**Background + Objectives** ..... 5

**Overview + Motivation**..... 5

**Systems Engineering Process for Portfolio Management Problems** ..... 6

    Define the Problem and List Alternatives ..... 6

    Develop the Evaluation Framework ..... 6

    Complete the Framework ..... 7

    Test the Framework ..... 7

**Portfolio Management and Analysis Toolset**..... 7

**Requirements** ..... 8

        Requirements Definition Tool Overview ..... 8

**Data**..... 12

**Analysis** ..... 13

**Future Plans** ..... 15

This page intentionally left blank

## BACKGROUND AND OBJECTIVES

---

This research effort is comprised of three tasks:

1. Develop approach. SERC shall define the portfolio management capabilities needed for FACT, using inputs from the Global Combat Support System – Marine Corps (GCSS-MC) program office. SERC shall assess the current FACT capabilities to meet these needs, identify the gaps to be addressed and evaluate the technologies to be developed to address these gaps. SERC shall also explore concepts for visualizing capability options, such as the concentric spheres methodology.
2. Evaluate existing tools and data. SERC shall work with the GCSS-MC program office and other subject matter experts to develop a components-to-capability correlation matrix, or similar construct, for the GCSS-MC program. This matrix will be used to evaluate the FACT portfolio management capability. SERC shall identify and evaluate existing data, subject matter expert input, and GCSS-MC planning activities needed to support the GCSS program. SERC will keep an inventory of data and databases and will evaluate detailed models to determine how they can be incorporated into the FACT model. SERC shall explore techniques and develop a strategy for including portfolio roadmap planning within the FACT tradespace capability.
3. Develop and apply methods for incorporating portfolio data and models into FACT. SERC shall assist in identifying and developing additional models needed for the portfolio management capability, such as rule-based design models based on historical trends, subject matter experts or heuristic reasoning. SERC shall explore and potentially generate the first set of cost estimating relationships for the new capability. SERC shall incorporate the data and models in an executable, browser-based/metadata model into FACT and provide a GCSS Portfolio Management Capability.

## OVERVIEW AND MOTIVATION

---

Management of a portfolio of capabilities, generally realized in a suite of physical systems, is a common problem faced by the agencies and programs within the Department of Defense (DoD). Standardized methods of tackling these problems have been developed over time, with a focus on sound Systems Engineering (SE) principles and tools. Although a standard process is commonly executed in solving these portfolio management problems, a standardized suite of tools is non-existent to support that process. Most often, new tools are developed by each SE team, sometimes leveraging a past tool utilized by the team, but often time not being reused for future efforts.

This limited reuse of tools was a primary motivator to the SERC for this research: to capture the standard SE process and develop a toolset which allows analysis teams and Systems Engineers to focus time and effort on the analysis rather than management tools and processes. In order to inform the prioritization and development of capabilities, SERC worked closely with an ongoing project team: GCSS-MC. SERC provided SE expertise for the specific GCSS-MC problem while in parallel developing the Portfolio Management and Analysis Toolset (PMAT). When schedule permitted, the developed tools were utilized by GCSS-MC to conduct analysis and inform decisions.

The remainder of this report is organized as follows. First, SERC documents the standardized SE process applied to portfolio management problems. Following, the existing PMAT capabilities are described including instructions on how to utilize the tool. Finally, the capability gaps of the PMAT itself are described with summaries of the tools that need to be developed and integrated into the toolset to offer the comprehensive set of capabilities.

Portfolio management problems are well addressed when following a solid set of systems engineering principles and practices. As such, it is important that a systems engineer works closely with the customer as a facilitator. The overall process is described below to provide context for the discussion of the toolset that follows in the next section.

---

### DEFINE THE PROBLEM AND LIST ALTERNATIVES

The process begins with a careful consideration of the problem. This entails both developing the problem statement (high-level goals) and compiling a list of alternatives that serve as potential solutions. Note that these are created iteratively, not sequentially. Decomposition and revision of the problem statement may highlight capability gaps in the alternatives and necessitate further consideration. Alternatives may also become irrelevant as the problem statement is refined.

Similarly, the inclusion (and exclusion) of options from the list of alternatives may lead to reconsideration of the problem statement. The goal is to make sure that all tacit needs are explicitly stated in the high-level goals. *This option seems relevant; why is it excluded? How do its capabilities differ from the problem as stated? Why are similar options included in the list of alternatives?*

The provenance of both the problem statement and the list of alternatives must be traced to ensure that they are accurate, complete, and current. It is possible that the customer may already have an idea of the high-level goals of the project and potential alternatives. If not, the facilitator is prepared to start from scratch, potentially organizing a workshop or developing a market survey. When developing the list of alternatives, it is important to forecast future alternatives relative to the problem scope, accounting for options that may not be available now, but will be by any implementation deadlines. Portfolio management exercises are often implemented in the not-so-near future.

---

### DEVELOP THE EVALUATION FRAMEWORK

How are alternatives to be evaluated against the high-level goals of the project? The development of the evaluation framework requires extensive interaction with a small group of subject matter experts who possess a detailed understanding of the problem.

The high-level goals must be iteratively decomposed until they can be mapped to tangible alternatives. For example, a GCSS-MC combat scenario (e.g. high demand for bandwidth, high latency) might be decomposed into a series of activities (e.g. make supply request, update inventory). These activities might then be mapped directly to the capabilities of the alternatives. This is an iterative process; there should be just enough levels to make the mappings traceable. Furthermore, the number of items in each level should increase monotonically. That is, a 10-item level should not map to a 5-item level and then up to a 15-item level. This is indicative of information loss.

Next, define the mapping between each level. The map is a general function, but in most cases it takes the form of a 2-dimensional matrix with its rows corresponding to the domain level and its columns corresponding to the range level. The matrix will be populated with values from a scale defined by the facilitator. The scale may be numeric (e.g. 0, 1, 3, 9) or semantic (e.g. None, Little, Some, Much), though a semantic scale generally has an underlying numeric equivalence. Use clear, precise language. Avoid detailed scales with too much granularity; effort and time may be wasted in trying to differentiate between nearly equivalent values. Moreover, it may be

necessary to hide the numeric meaning of the scale from the experts filling out the matrix. This information may affect their ratings either consciously or unconsciously.

---

### **COMPLETE THE FRAMEWORK**

The previous step developed the general framework for evaluating the alternatives against the high-level goals of the project. Now experts fill in the details (e.g. populate the matrices that map between levels). This may be accomplished with a survey or a workshop.

The usual biases that affect survey data apply here as well. It is important to make sure that responses are not influenced by superiors and that the loudest voice does not carry the day unquestioned. If possible, assess the accuracy of the data and explore any patterns.

If a workshop is necessary, be efficient. Distribute individual surveys at least 2 weeks before the workshop. Compile the data and identify disagreements for further discussion. Use anonymous voting methods to minimize biases and intimidation tactics. Furthermore, work with a respected mediator capable of controlling the experts, keeping the discussion moving, and minimizing unproductive conflict.

Pay attention to the proceedings. The validity of the framework may be assessed from the rate of progress. For example, slow progress may indicate that the map is too difficult to complete and further decomposition of the levels is necessary. Alternatively, rapid progress may indicate that the map is too obvious and there are too many levels. The framework may very well require further consideration.

It is not required that subject matter experts complete the framework. These experts provide insight into how alternatives function in the real world in their specific area of expertise, which may not cover the whole framework. The theoretical functionality may be studied with a quantitative model (e.g. physics models, battlefield simulation). The important result is a complete evaluation framework.

---

### **TEST THE FRAMEWORK**

Now, test the framework with various inputs and examine the results. If the customer is surprised by the results, determine the reason. The framework (or its data) may require further iterations. If the framework does not behave as desired, it is important to fix the problem at its source and not just artificially manipulate the results.

### **PORTFOLIO MANAGEMENT AND ANALYSIS TOOLSET**

---

The PMAT applies these Model-Based Systems Engineering standards, a browser-based front-end, and open source software to create a framework for portfolio development and analysis. The PMAT separates the portfolio management process into modules; this allows the user and a facilitator to work together to accomplish the following:

- State the problem and define requirements;
- Manage data;
- Analyze and explore options.

The PMAT is web hosted by SERC to be available to the entire GCSS-MC program team.

---

## REQUIREMENTS

The Requirements Development and Analysis module specifies the problem statement, creates high-level capabilities, and produces a set of well-defined requirements. The module keeps a complete history of the drafts and revisions of the problem statement. Additionally, the user may make notes, refer to external documents, add important terms to a glossary, and assign responsibilities.

Similar functionality is available to define the high-level capabilities necessary to solve the problem. The responsibility to develop a capability is assigned either to a team or an external group. This ensures that all required capabilities are developed.

Ultimately, the product of the Requirements Development and Analysis module is a set of well-defined requirements (see **Error! Reference source not found.**). The module assists the user wherever possible in creating these. For example, the notes and revision history help to trace the origins of each requirement. Requirements are prioritized according to a user-defined scale (e.g. blocker, critical, major, minor, or trivial). The module captures the verification method and a detailed description of the verification plan.

**Table 1: The Requirements Development and Analysis module facilitates the creation of a set of good requirements, the characteristics of which are defined above (Source: Wikipedia).**

Characteristic	Explanation
Unitary (Cohesive)	The requirement addresses one and only one thing.
Complete	The requirement is fully stated in one place with no missing information.
Consistent	The requirement does not contradict any other requirements and is fully consistent with all authoritative external documentation.
Non-Conjugated (Atomic)	The requirement does not contain conjunctions.
Traceable	The requirement meets all or part of a business need as stated by stakeholders and authoritatively documented.
Current	The requirement has not been made obsolete by the passage of time.
Unambiguous	The requirement is concisely stated and expresses objective facts, not subjective opinions.
Specify Importance	The requirement must specify a level of performance.
Verifiable	The implementation of the requirement can be determined through basic possible methods: inspection, demonstration, test (instrumental) or analysis (to include validated modeling and simulation).

---

## REQUIREMENTS DEFINITION TOOL OVERVIEW

The Requirements Definition Tool walks you through the process of creating well defined requirements as described. The tool is set up such that it can support multiple projects from the same tool instance, therefore the first step is to setup a Project and define the User Group that is responsible for it, thereby providing access (see Figure 1).

# Create New Project

Name

Provide a short description name for the project.

Team

You may first need to setup a Group in the [Admin](#) interface.

Figure 1. Create New Project form.

All functions of the tool are provided in a single tabbed dashboard. The initial tab, shown in Figure 2, allows for modifying the project name and team. All tabs offer a means to add time-stamped notes and attachments.

The screenshot shows the 'GCSS Portfolio Management Analysis Tool' dashboard. The top navigation bar includes 'Home', 'RTM Interface', 'Metrics Data', 'Requirements', 'Data Confidence', 'Risk Assessment', and a user profile 'danny'. The main content area is titled 'GCSS-MC' and has several tabs: 'Summary', 'Setup', 'Story', 'Requirements', 'Glossary', and 'Attachment Details'. The 'Summary' tab is active, showing a table with project details and two buttons: 'Edit' and 'Delete'.

Name	GCSS-MC
Team	GCSS-MC <ul style="list-style-type: none"><li>Danny Browne</li></ul>

General Notes section contains a text area with a blue 'Add' button below it. An existing note reads: 'This project will support the Global Combat Support System - Marine Corps in determine the technologies needed to support the warfighter.' — danny @ Nov. 7, 2014, 1:42 p.m.

Attachments section has an 'Upload a file' button and a list item: 'Project-Charter.pdf' — admin @ Oct. 13, 2014, 7:54 a.m.

Figure 2: Summary tab of the Requirements Definition Tool dashboard.

On the Setup tab, users have the ability to define their importance scale, which will be utilized later for qualitatively evaluating requirements. Note the standard icons used throughout the tool which are first displayed here. Table 2 provides the legend for these common icons. Figure 3 provides a view with an example quantitative importance scale similar to one in a Quality Function Deployment.

**Table 2: Legend for common icons used throughout the Requirements Definition Tool.**

	<b>Edit.</b> Edit the standard form that was created with any data object.
	<b>View item with object-specific notes.</b> Display notes only for a single data object due to the context-specific nature, in contrast to the project-wide notes.
	<b>View History.</b> View the revisions that are time stamped and marked by the creator/editor. For most data objects in the tool, revisions are kept to allow for provenance to be determined.
	<b>Delete.</b> Delete the item including all history.

Summary Setup Story Requirements Glossary Attachment Details

## Importance Scale

Define your importance scale which will be used to weight your requirements. ✕

[+ Add Importance Scale](#)

Value	Name	Description	
10.0	Must Have	This is an absolute must have. If the provided system cannot achieve a requirement with this importance, than the system does not provide sufficient value added.	  
5.0	Nice to Have	A requirement with this importance is highly desirable, but a system which does not meet it may still have value added.	  
3.0	Desirable	Like "Nice to Have", but not worth as much in the value added sense.	  
1.0	Pie in the Sky	Would be really nice, but not at all necessary.	  

**Figure 3: On the setup tab, users can define the importance scale that will be used to evaluate the requirements.**

The Story tab contains perhaps the most important step: defining the story. The story is the overarching, high-level description of the problem at hand that will be decomposed into measurable needs by deriving the set of requirements. It is expected that this will be an iterative process that may change even as the first set of

requirements is defined. Note the blue words in the GCSS-MC story shown in Figure 4, as well as the [+ Add Term](#) button. These are indications of the *Glossary* capability. When working on a problem definition and building a set of requirements, it is critical that the team is utilizing the same vocabulary. The use of a glossary provides a way to ensure consistent definitions. Within the story and the requirements, words or phrases from the glossary appear as hyperlinks; the definition is displayed in a tooltip. The Glossary itself is displayed in Figure 5. Note that the remaining icon from Table 2 is provided here, which indicates object-specific notes may be available.

You can iterate on your story here. Don't worry, copies of each change are saved and you can view the history. ✕

[✎ Edit Story](#) [Create Story](#)

[+ Add Term](#)

**Global Combat Support System Marine Corps provides the means for **Marines** to request supplies. This system, however, is not available in **areas** with poor **comms**. The system needs to be improved to be more robust in poor **comms areas**.**

— danny @ Nov. 7, 2014, 2:02 p.m.

[Story History](#)

**Figure 4: The Story tab displays the current version of the story as well as an option to view the history.**

[+ Add Term](#)

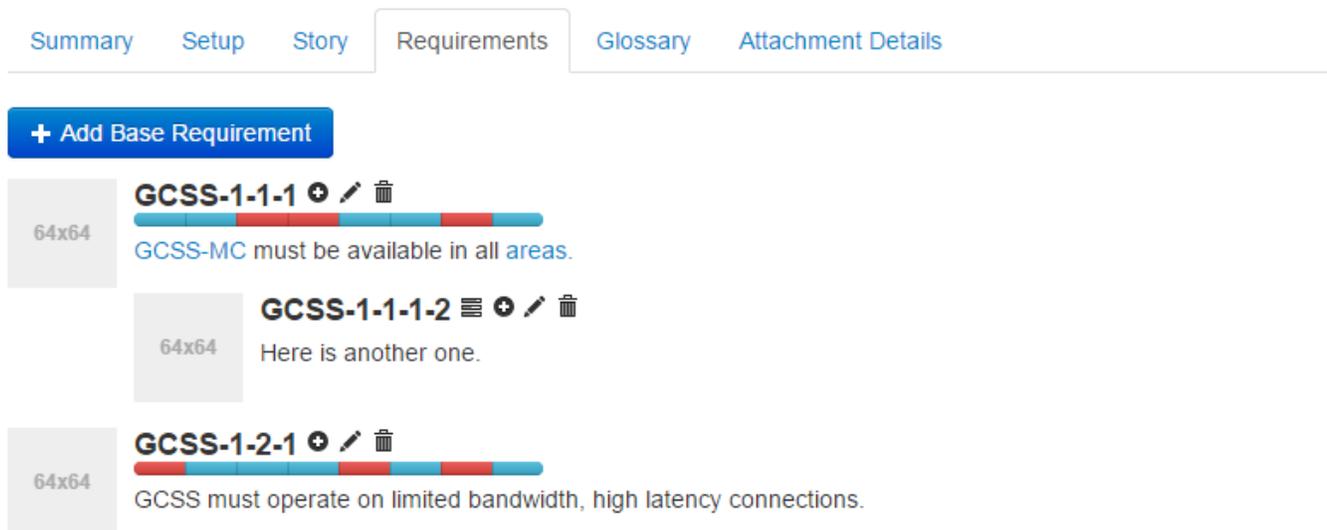
Term	Current Definition	
<b>areas</b>	For this effort, areas are decomposed into Zones. They are numbered 1 thru 5. (add details about what each Zone is)	<a href="#">✎</a> <a href="#">💬</a> <a href="#">🕒</a> <a href="#">🗑️</a>
<b>comms</b>	Communication. In this context, poor comms refers to systems that have high latency and/or low bandwidth.	<a href="#">✎</a> <a href="#">💬</a> <a href="#">🕒</a> <a href="#">🗑️</a>
<b>GCSS-MC</b>	Global Combat Support System - Marine Corps	<a href="#">✎</a> <a href="#">💬</a> <a href="#">🕒</a> <a href="#">🗑️</a>
<b>GTRI</b>	Georgia Tech Research Institute	<a href="#">✎</a> <a href="#">💬</a> <a href="#">🕒</a> <a href="#">🗑️</a>
<b>Marines</b>	The United States Marine Corps (USMC) is a branch of the United States Armed Forces responsible for providing power projection from the sea,[7] using the mobility of the U.S. Navy to rapidly deliver combined-arms task forces.	<a href="#">✎</a> <a href="#">💬</a> <a href="#">🕒</a> <a href="#">🗑️</a>

**Figure 5: The Glossary provides a standard means to define project-specific terms.**

The primary capabilities of the tool exist within the Requirements tab (see Figure 6). Here you can add new base requirements, add child requirements to existing requirements, and add access details about each requirement. Specifically note the blue and red bar, decomposed into eight components, underneath the requirement identifier. This visual cue provides a quick reference to determine if a requirement meets the eight criteria defined in **Error! Reference source not found.**. The regions on the bar are ordered: unitary, complete, consistent, non-conjugated, traceable, current, unambiguous, verifiable. Clicking the bar navigates the user to a checkbox form where they can indicate whether the requirement as-is satisfies each criteria. A solid blue bar indicates that a

“good,” fully decomposed requirement has been achieved. The goal is to have all “good” leaf-node requirements with mid-level requirements assisting in organization and context and assumed to be satisfied once all children requirements are satisfied.

Future work will allow requirements to be marked as satisfied/complete as well as allowing requirement import/export via a comma separated value format.



**Figure 6. The Requirements tab offers tools to build a requirements hierarchy and visual cues to determine if requirements are "good."**

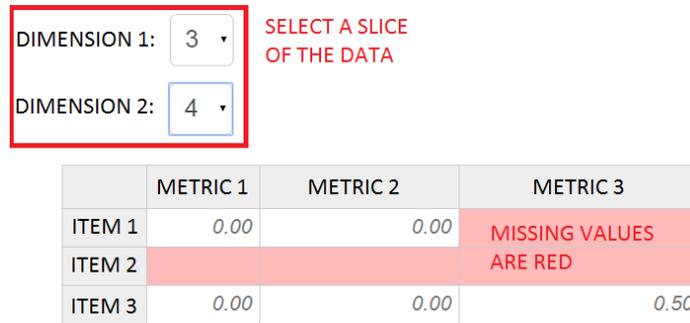
---

## DATA

The flexibility of the software allows external data to be considered in the analysis, even when the format is complex. For example, the GCSS-MC Requirements Traceability Matrix and metrics data are both stored in custom Excel files. The Oracle database reports are human-readable text files, requiring a custom parser to generate a computer-understandable format. The custom parser process conducts basic checks and maintains the processed copy. Parsers may be reused or extended to handle similar data sets. The automation allows the user to focus on data quality rather than simple transformation.

The Solution Set Exploration module is an interface to the GCSS-MC database of options and their corresponding metrics (e.g. power consumption, bandwidth utilized). Metrics vary across the dimensions of bandwidth demanded and network connectivity. The GCSS-MC database is thus four-dimensional: solution options, metrics, bandwidth demand, and network connectivity. Note that the module is general enough to handle any multidimensional data set.

The data are inspected via two-dimensional slices (see Figure 7). Select a slice of the data using the dropdown menus. The cells of missing values are colored red to highlight gaps in the data. The viewer is read-only; once data issues are discovered, they must be remedied in the original data file and uploaded again.



**Figure 7: Inspect a multidimensional data set.**

For some data (e.g. vendor surveys), quality is a question of trust. Is this vendor overselling the product? Is *this vendor* exaggerating the response to *this question*? With many vendors and questions it quickly becomes infeasible to evaluate the confidence in each answer. The Data Confidence module simplifies the process by asking two questions:

1. How trustworthy is each vendor?
2. How specific (unambiguous) is each question?

Suppose there are  $m$  vendors and  $n$  questions on each survey. This methodology reduces the number of necessary measurements from  $mn$  to  $m+n$  (see Figure 8).

$$\begin{array}{ccc}
 \begin{bmatrix} r_{1,1} & \dots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{m,1} & \dots & r_{m,n} \end{bmatrix} & = & \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \times \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix} \\
 m \times n & & m \times 1 \quad \quad \quad 1 \times n \\
 \text{Matrix of} & & \text{Vector of} & & \text{Vector of} \\
 \text{Response} & = & \text{Vendor Trust} & \times & \text{Question} \\
 \text{Confidence} & & & & \text{Specificity}
 \end{array}$$

**Figure 8: Calculate the confidence in each vendor response to a survey.**

---

## ANALYSIS

The PMAT analyzes and compares portfolios. The analysis is limited only by the available data sources.

The Requirements Traceability Matrix (RTM) Analysis module displays the requirements coverage of each portfolio (see Figure 9). The portfolio items are scored against each requirement according to the scale in Table 3 **Error! Reference source not found..**

Show Rows Where: All Show All Rows

Ctrl-click to hide rows, columns, and categories.

FILTER THE TABLE

Overall portfolio scores:

Group01	Group02	Group03
50.00%	75.00%	90.10%

SUMMARIZE  
REQUIREMENT  
COVERAGE

HIDE COLUMNS

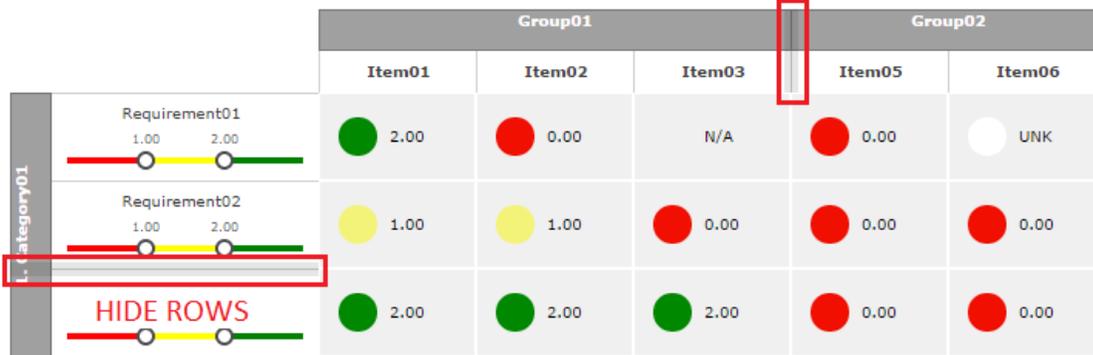


Figure 9: The RTM Analysis module displays requirements coverage.

Table 3: The RTM Analysis module scale.

Icon	Value	Description
2.00	2	The objective value of the requirement is achieved.
1.00	1	The threshold value of the requirement is achieved.
0.00	0	The requirement is not met.
N/A	0	The requirement does not apply.
UNK	0	The status of the requirement is unknown.

Hide rows by ctrl-clicking on a row label or row group label. Columns are hidden in the same way. Use the filter menu to only display rows that satisfy some criteria. For example, display gaps in coverage by only showing requirements that are not satisfied by any portfolio. Or display gaps in data by only showing requirements with unknown status.

Each portfolio is given a summary score according to how well it satisfies the requirements. First, a portfolio is assumed to perform to the level of its best component, so for each requirement

$$Portfolio\ Value = \max_{Components} \{Component\ Value\}.$$

Then, the portfolio values are aggregated across all requirements:

$$Portfolio\ Score\ (\%) = \frac{\sum_{Requirements} Portfolio\ Value}{2 \times Number\ of\ Requirements} \times 100.$$

Note that the RTM Analysis module is capable of visualizing any set of solution options and requirements; its application is not restricted to the GCSS-MC program.

## FUTURE PLANS

---

Additional modules are planned to complete the PMAT. These capabilities are listed below.

- The **Survey Builder** converts the well-defined requirements from the Requirements Development and Analysis module into a market survey form. Market surveys are common in portfolio management. Moreover, this module manages and visualizes the survey responses using interactive histograms.
- The **Integration Risk Assessment** module measures the risk inherent in combining a set of systems into a portfolio. SERC has developed a quantitative methodology for capturing this risk that highly leverages the standard NASA Risk Matrix. SERC built upon the one-to-one risk integration that can be displayed on the standard two-dimensional Risk Matrix and proposes a means to aggregate the information across a portfolio.
- The **Metrics Analysis** module calculates a portfolio score using the metrics for each system. Each metric has two associated functions. The aggregation function calculates the metric value of the portfolio from the metric values of its components. The utility function transforms the metric values to a standard scale. For example, larger metric values might be better than smaller or perhaps it is best that metric values be close to some optimal target.
- The **Document Library** allows for the storage, sharing, and versioning of reference material. The goal is to track authoritative data sources and allow for references to these authoritative data sources within requirements, capability assessments, or captured decisions.
- The **Data Citation** module provides a means to associate application data to its source. Currently, there is no clean capability for citation. This lack confuses the analysis because data may have come from an authoritative source, subject matter expert, or the best guess of a team member.