RT 112: Development and Application of FACT Portfolio Management Capability

SYSTEMS ENGINEERING Research Center

Daniel Browne   Todd Shayler
Melissa Rost   Devraj Mehta

Georgia Tech | Research Institute

Research Task / Overview

1. **Develop approach.** Define portfolio management capabilities needed for FACT, using inputs from the Global Combat Support System – Marine Corps (GCSS-MC) program office. Assess current FACT capabilities to meet these needs, identify the gaps to be addressed and evaluate the technologies to be developed to address these gaps.

2. **Evaluate existing tools and data.** Work with the GCSS-MC program office and other SMEs to develop a components-to-capability correlation matrix. Identify and evaluate existing data, SME input, and GCSS-MC planning activities needed to support the program. Explore techniques and develop a strategy for including portfolio roadmap planning within the FACT capability.

3. **Develop and apply methods for incorporating portfolio data and models into FACT.** Assist in identifying and developing additional models needed for the portfolio management capability, such as rule-based design models based on historical trends, SMEs or heuristic reasoning. Explore and potentially generate the first set of cost estimating relationships for the new capability. Incorporate the data and models in an executable, browser-based/metadata model into FACT and provide a GCSS Portfolio Management Capability.

Portfolio Management Analysis Tool

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:
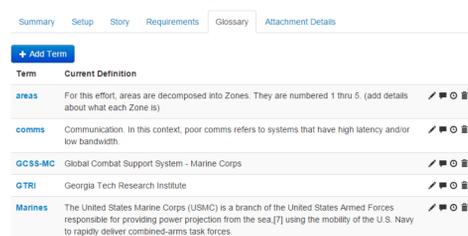
- State the problem and define requirements

- Manage data

- Analyze and explore options.

PMAT first offers a Requirements Decomposition Wizard. It assists users in first defining the high-level goals of the problem at hand and iteratively decomposing that goal into *good* requirements (i.e. requirements which meet the 8 characteristics of good requirements). This tool captures the entire process, including detailed revision history and some configuration management capabilities.

For data analysis, PMAT offers multiple capabilities. First is a tool for evaluating collected data to determine how trustworthy data is. An additional tool offers a means to aggregate high-dimensional data down into 3 dimensions so that it can be visualized and inspected by the SMEs. Finally, a standard stoplight chart is fed by collected data for a quick and commonly used view for seeing how specific solutions match up against requirements.

View of stoplight comparison with filter capabilities for quick inspection of solutions against the requirement set.
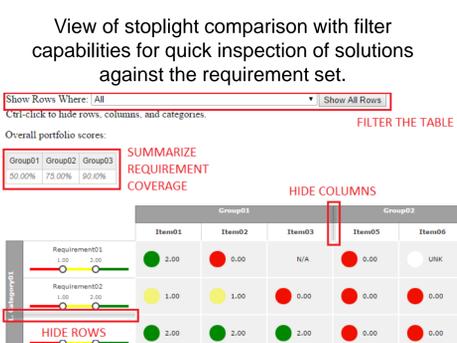
View of one tab in the Requirements Decomposition Wizard tool, specifically the Glossary. The Glossary allows users to define terms which are then hyperlinked throughout the rest of the tool.

Systems Engineering Process for Portfolio Management Process

1. **Define the Problem and List Alternatives.** The process begins with a careful consideration of the problem. Develop the problem statement (high-level goals) and compile a list of alternatives that serve as potential solutions. 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.

2. **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. 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.

3. **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. Be aware of the usual biases that affect survey data. If a workshop is necessary, be efficient. 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.

4. **Test the Framework.** 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.

Contacts

- Daniel Browne
  Georgia Tech Research Institute
  Research Engineer II
  Associate Head, Systems Engineering Software Applications Branch
  daniel.browne@gtri.gatech.edu
  404-407-7264

- Todd Shayler
  Georgia Tech Research Institute
  Research Engineer II
  todd.shayler@gtri.gatech.edu

- Luis E. Velazquez (NH-IV)
  Deputy Director, Modeling & Simulations Division
  Systems Engineering, Interoperability, Architecture & Technology (SIAT)
  MARCORSYSCOM
  luis.velazquez@usmc.mil
  703-432-3791