Supply Chain Planning & Optimization Implementations: Avoiding Armageddon

Most of us are familiar with the impact that a supply chain planning and optimization (SCP&O) application can have on an organization's bottom line. And how strategic network design models, tactical planning and other SCP&O solutions can quickly become an invaluable source of sustainable competitive advantage.

As with many things in life, high returns go hand in hand with high risks, and these projects are no exception. One significant risk is a failed implementation. We rarely read or hear about them, unless it is through the grapevine or if we had the misfortune of experiencing one first hand. But they do happen, much more frequently than they should. With few exceptions they end up being very frustrating, expensive and avoidable missed opportunities.

SCP&O implementations fail for a number of different reasons, some beyond the control of the people directly involved in the project. The two main culprits, however, are preventable: Poor quality solutions and a system too complicated and time consuming to operate.

What should be the first two tasks of any SCP&O project are also the key to their eventual success: **first, define the scope and then select the software solution that best fits this scope.** Get both right and your chances of success are very good. Miss one, or both, and you better load up on aspirins and antacids – your project will have a tough time surviving.

The following is a basic road map that can be applied when defining the scope of SCP&O projects. In a second post we will do the same for the software selection question.

The approach is not infallible (nothing really is when implementing planning & optimization systems!), but it has served us well in more than 200 projects over the past 25 years. It can and should be applied to all SCP&O problems, large or small, from strategic to operational, from relatively simple Excel-based network design models to very complex and fully integrated planning or execution applications.

Photo: www.securityintelligence.com

First and Foremost: Get The Scope Right

A SCP&O project arises from the decision to address a specific need or solve a problem. Its ultimate success will depend on how effective (quality) and efficient (speed) it is in doing so.

Every project should start in the exact same way: With a workshop session to confirm its purpose, clearly understand the problem and, based on both, define its scope.

There are two parts to the scope of any SCP&O application:

1. What the system must do (problem definition and model specification)
2. How it will do it (system and functional specification).

The 'What' drives the 'How', so it must come first. But both need to work well for the project to succeed:

- If the model cannot produce good solutions, it won't matter how smooth the user interface looks, or how impressive the BI module is.
- Alternatively, if it will take hours or days to build and solve a new scenario, it won't really matter how good the solutions may be.

1. Make sure the objective is clear

Start by confirming the main objective of the application (e.g. to identify the optimal network design, generate tactical planning solutions, schedule production on a weekly basis, etc.). This objective will have a major influence on the dimensions and characteristics of the model as well as the mathematical approach that will work best.

2. Do not mix beef chili with creme brulee

I love a thick, spicy beef chili and there will never be a serving of creme brulee I won't enjoy. But I would NEVER mix them in the same plate! You should stick to this principle when scoping a SCP&O application: Beware trying to solve two (or more!) problems with the same application, especially if they target different planning horizons.

3. Do NOT start looking at software alternatives yet!

Now that the objective is clear, it is very tempting to start looking at software alternatives. Please don't. Not yet.

Continue with your efforts defining what you actually need, without external, often biased influences. Otherwise you may end up with an application that will have to adapt to what the technology can offer, when what you should be aiming for is exactly the opposite.

4. Define the business/operational problem

To define the problem to be solved, you will typically need to look at:

The type of solution required (a network flow, a detailed schedule, a monthly distribution plan, a replenishment schedule, etc.)
The objective function (costs, margins, a KPI, etc.)
The physical assets involved
The planning horizon and time buckets
The business rules affecting and driving decisions
Operational restrictions
The interaction between rules, restrictions and physical assets

5. From problem definition to model specification (with a little help from Pareto)

When defining the problem there is always a tendency to include as much detail as possible, for fear of missing something that might be relevant. While the resulting model will describe reality very accurately (at least on paper), it may also be too complex to implement, maintain and use because of the extensive list of business rules and operational restrictions included.

The challenge is to find the right balance between the detail required to produce good quality solutions and the size and complexity of the model. This is one of the critical success factors in these projects. The skills required involve as much art as they do science, and they come from having done this before, and often.

This is where you should start:

Make sure the planning horizon and time buckets truly reflect the objectives (You probably don't need to model 52 weeks in a strategic network analysis)
Remove business rules and operational restrictions that are not compatible with the planning horizon and time buckets
Consider removing those rules and restrictions that, although relevant to the objective, clearly fall under the 'nice to have' category.

6. Functional specifications: Focus on efficiency

Developing the functional specifications for a SCP&O system is no different from any other system. The area that requires special attention relates to the set of features unique to your SCP&O application aimed specifically at maximizing users' productivity. This goes back to one of the main reasons these systems fail: It takes too much time to build, solve and analyze a new scenario.

Following are a few examples of such features:

- Automated procedures and logic that will allow users to build and generate the entire data set for a new scenario with a single click of the mouse
- A data integrity check procedure that flags inconsistencies in both import and scenario data
- Case management and comparison capabilities
- The ability to override both scenario data and the solution generated

7. Do NOT trust others with the data you need!

SCP&O applications are voracious consumers of input data – cost, demand, actuals, forecasts, capacities, you name it, they need it. This data will be imported from various sources and in different formats, units of measure, currencies and levels of aggregation.

This input data usually needs to be manipulated, processed and transformed in order to create new scenarios. An good example would be the allocation of aggregate forecast data to individual customer locations based on historical actuals.

Inaccurate data is one of the key reasons why otherwise sound models produce solutions that don't make too much sense. We have learned (unfortunately, the hard way,,,) that the best way to avoid this problem is to import all the data in its raw state and develop the automated procedures within the application to perform all required data manipulations and transformations.

8. When (not if!) you realize you missed or need to add something....

It doesn't matter how much effort and resources you invest in developing the best possible project scope, you will miss something. It may be a business rule or operational restriction, or a functional feature (section 6 above) that the project team didn't think about.

This will only become apparent later on, when you start validating and calibrating the application or once operational as your business model changes (new rules, restrictions, etc.).

We will discuss the selection of the right software solution in my next post.

One of the key features we will emphasize is the need for flexibility and adaptability. If a software solution can't truly adapt and offer the flexibility to incorporate changes as your business needs evolve, it is not one you want to consider.

At Technologix we have been modeling and solving all sorts of interesting supply chain and related problems since 1993. If you are interested in the subject you will find a well organized (or so I think!) gallery of Projects on our website.