Foundation of Systems Engineering Problem Solving

Dr. Orin E. Marvel
Naval Postgraduate School
Monterey, California, USA, 93940
opainc@nps.edu

ABSTRACT

The fundamental goal of Systems Engineering is problem solving. The breadth of this definition comes from the problems that can vary from simple to impossible. A lot of people have broken the problem solving process into three steps with a variety of names. For this paper, I am choosing to call the steps (1). Problem System, (2) Project System, (3) Delivered System. The problem System encompasses the definition or requirements of the problem as an "Abstraction", the project System defines the design, production, integration, and testing, while the delivered Systems defines the solution as provided to the customer.

This model is then used to evaluate candidate applications. As these applications pass through the model one records the problems, constraints, and what might go wrong leading to a practical definition of "Complexity". Also, one records the successful activities that come from experience or common sense in order to define "Process". In order to better understand complexity and process, we must take a practical example, "The Farmer Exercise". The reader will be asked to participate in personal experiment using the farmer exercise. The analysis of the results of this exercise will help us cement the definition of process and problem solving.

With a practical understanding of these definitions, we finally can understand the practical definition of "Architecture". The above is not an easy road to follow; but is necessary to understand the foundation of Systems Engineering.

INTRODUCTION

What is Systems Engineering (SE)? What is the definition of Systems Engineering? There is no consensus answer to either of those questions, even though we all know it is important. With no definition, how do we test concepts, methodologies, and processes? How do we describe systems engineering to customers, stakeholders and bosses? This paper starts with the hypothesis that SE is used for problem solving. It should be used to solve all problems; but there are simple problems that applying SE would be a waste of time. What is the boundary between simple problems and complex problems? What about when large numbers of people must work on the same problem? How do we insure they are all viewing the same problem? Read on!

PROBLEM SOLVING

The general problem solving process is shown in Figure 1. It consists of three activities:

1. The "problem system" which contains all the customer needs and requirements. It produces an acceptable base line that has been validated with the customer. It also contains the amount of influence that the customer will have over the problem solution. (Think of the pyramids. If people didn't do what the Pharos wanted, they were killed and new people put on the job.)

2. The "project system" which includes all of the development, design, and production of the solution to the problem. For very complex systems, like an aircraft carrier, this could require on the order of 600,000 people working to gather to complete the project.

3. The "Delivered system" includes the testing, integration, verification, certification and delivery of the working solution. Through all this effort, SE is only successful if the customer is smiling.

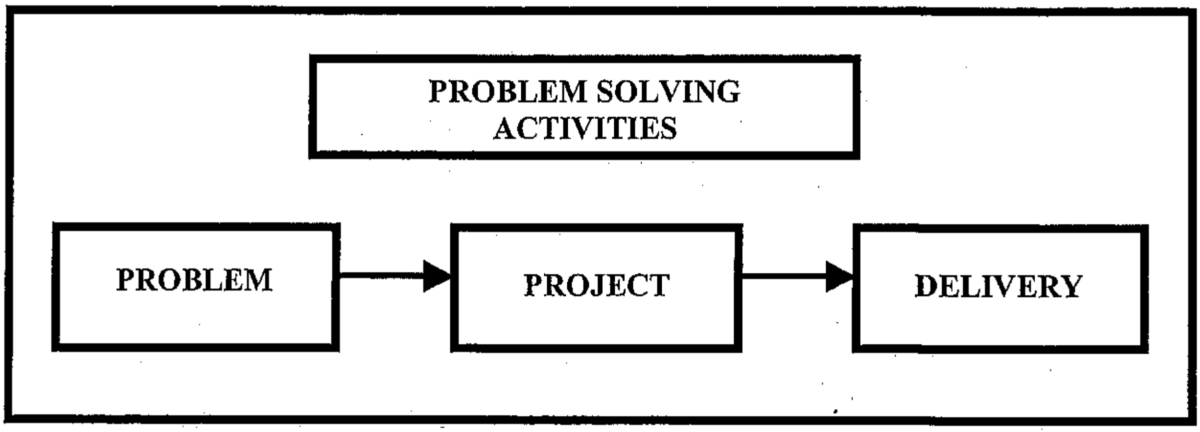

Figure 1 Problem Solving Process

Many models of problem solving have been proposed over the years and most of the ones that are simple to understand use three elements. If you don't like mine, pick another. This model is then used to derive the concepts, methodologies, and processes of SE. We use this model to develop two concepts that are very important to SE and successful product development. (Booch 1996)

The first is "lessons learned". With lessons learned we are trying to develop general rules, methods, and heuristics that always work for problems or applications that we are interested in. We due this by tracking programs through the problem solving process, while recording those things that worked and why they worked. With this database, we should be smarter the next time we have to develop a product that is similar. We must be very cautious that we understand all of the variables or changes from the system that generated the database; so that we don't do something that is really stupid, just because it was done that way before.

The second is "Complexity". Complexity is a measure or list of all the things that could go wrong while performing the problem solving process. After passing many products through the problem solving process, the definition of complexity and its solution has been defined. For example see Figure 2. Here complexity, as a generalized quantitative measure, is the sum of the number of people involved on the project, the number of technologies required to make the system work, and the number of customer needs or requirements. A number of metrics studies have shown that the sum of these three quantities gives a general quantification of complexity. This quantification can then be used to predict cost, resources and schedule from the quantification of similar successful programs.

Figure 2, also, shows how organizing and planning can solve complexity or overcome chaos. The management and SE activities sends its plan into the "chaos" and then after a while evaluates the results. When these results are compared with the system model and requirements, than a new plan is develop and sent into the "chaos". This continues as the product gets closer and closer to the desired solution. It probably could go on forever; but there usually comes a time when it is good enough and the project has run out of money. Then we Stop and deliver the product to the customer.

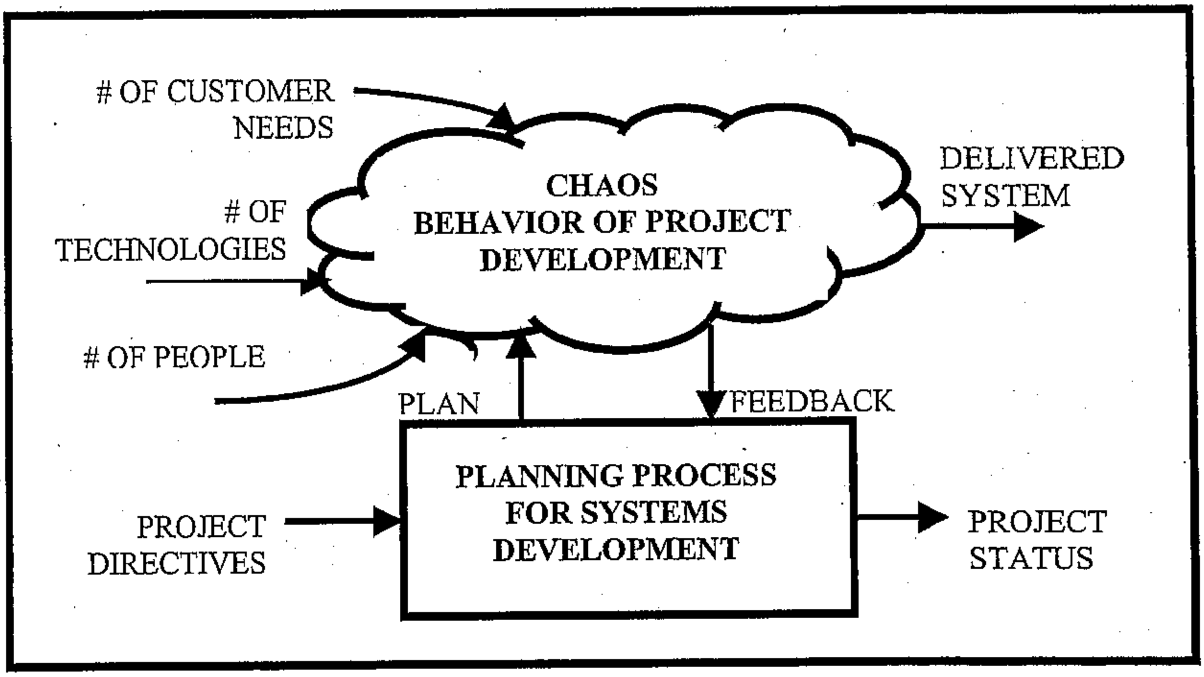

Figure 2 Definition of complexity and how to solve it

PROCESS

A lot of money has been made on SE books that supposedly teach the SE process. All of them are a sham. WOW !! For 15 years I have been running a class exercise called the Farmer exercise. It consists of the 30 clues shown in Figure 3. A group of 5 students is each given 6 of the clues. They can read their clues to the group; but can't write anything down or look at each other's clues. (This is also a proof of short memory; see my paper on "SE Element Building Blocks".) From the clues, the students must determine of the 5 farmers: who has what crop, what animal, what house, and what vehicle. Then they must answer: (Which farmer drives a truck? And which farmer grows apples?)

PLEASE TRY IT, AT LEAST FOR A WHILE, BEFORE YOU READ ON.

Do you see that the teams must develop the process for solving the problem while they are solving the problem. They must do this without the knowledge of any problem solving process. They have never read any of the SE books or heard of the waterfall or spiral process. Do you also see that they are doing Planning, Problem Solving and Management all at the same time?

Usually after 10 to 15 minutes of chaos, they start to see that the clues are based around the positioning of the houses in a semi-circle. At this point they start to make real progress. Then it becomes hard to associate the elements with each farmer. They decide that each person on the team must be in charge of the elements for one farmer. From this point they have the solution within a few minutes. Every team uses the same process, as shown in Figure 4. Where did this process come from? (No one person on the team knows all 25 answers; but each can remember 5 in their short-term memory.)

1. The dog owner lives next door to the house with a plum orchard.
2. Hull raises Albino rats.
3. The farmer who lives in the bungalow raises pigeons.
4. Only one of the village houses is located on the east side.
5. The farmer who lives next to Pavlov drives a station wagon.
6. Pavlov's neighbor raises chimpanzees.
7. The farmer who raises dogs also grows cherries.
8. Skinner lives next to the red brick house.
9. **One of your group's tasks is to decide who drives a truck.**
10. The house of the village are standing in a semi circle, beside each other.
11. Kohler grows pears.
12. There is a limousine in the garage of the ranch house.
13. Each farmer raises a different kind of animal.
14. Farmer Thorndike lives next to farmer Skinner.
15. A motorcycle stands in the back yard of the log cabin.
16. The person who raises cats lives next door, to the east, of the house with the almond trees.
17. Your group has less than three tasks.
18. Every week boxes of dog food are placed at the gate of the log cabin.
19. Only one of the village houses is located on the west side.
20. Each of the five farmers living in the village drives a different kind of vehicle.
21. The ranch house stands next to the cottage.
22. Farmer Thorndike drives a sports car.
23. Farmer Skinner raises pigeons.
24. The log cabin is in the most northern position in the village.
25. Each farmer grows a different kind of fruit.
26. Only farmer Skinner lives at the west end of the village.
27. There are albino rats in the yard of the ranch house.
28. **One of the groups tasks is to decide who grows apples.**
29. Pavlov lives in the log cabin.
30. Each farmer lives in a different type of house.

Figure 3 Farmer exercise clues

None of the SE books or processes that are taught in SE courses did them any good in solving this problem. How then did they solve it? They used human thinking and common sense to develop the process needed as they better understood the requirements and clues as they went along. This result can be extrapolated to all complex development processes. The best people to develop the process are the people that must use it to solve the problems and no process is ever cast in concrete; but evolves as the project progresses. Just look at the DoD "Life cycle" Process. Ask your self, why does it change every 4 to 6 years? It changes because we humans are smarter every 4 to 6 years and the new process is better. Once in a while for example during the Gulf war, new methods are tried even though they violated the process. When they work, we can't chastise the developers for doing good work; so we change the process. Of course this means that we don't have it right yet. And, one can violate the process if one is successful; but if one is unsuccessful one goes to jail.

THE PROJECT TEAM

Having an understanding of problem solving, lessons learned, and complexity, we should be ready to see how all this fits together for systems. The best metaphor for SE or what a System Engineer does is the orchestra leader, Figure 5. In order to be successful, the orchestra leader and SE must do the following three activities correctly. Failure in even a small part of one of these leads to a lousy performance or a bad system that no one wants.

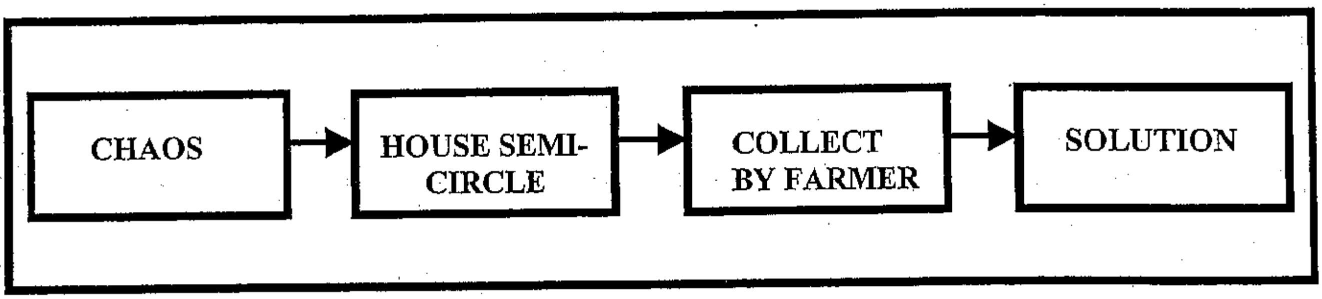

Figure 4 Farmer Exercise Perocess

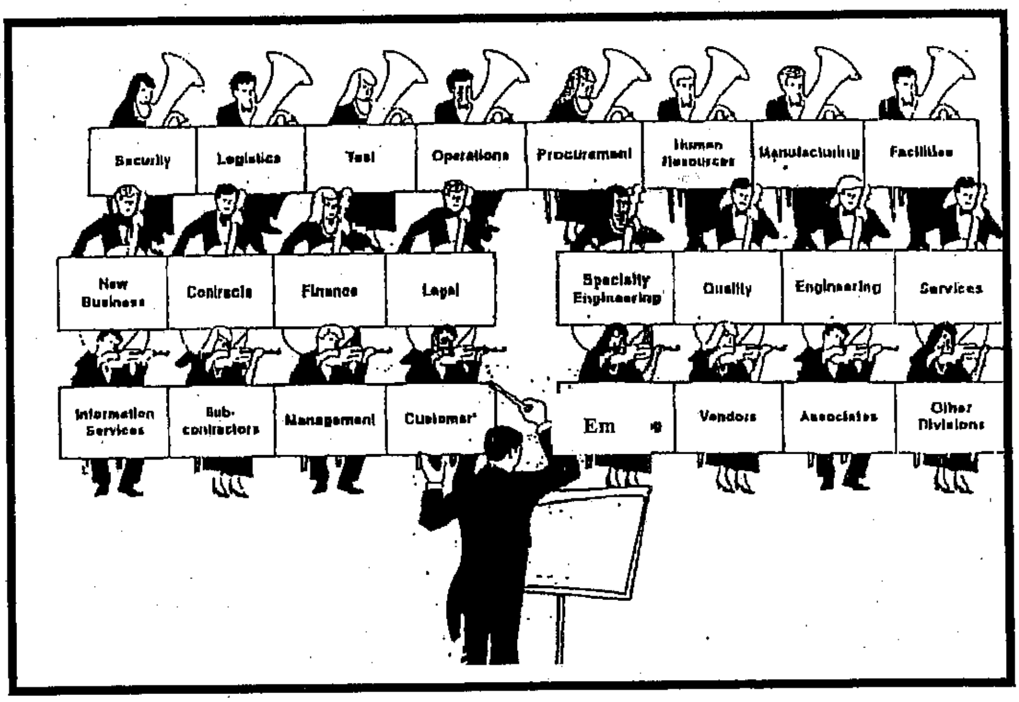

Figure 5 The Project Team Controlled by the Systems Engineer

1. The orchestra leader as SE must pick the best people for the specialty positions. The players or specialists must be arranged by expertise in leader and follower positions. Neither the SE nor the orchestra leader can play all the instruments or understand each specialty. But they can talk to each specialist; in the language the specialists understand and have a sense when they have a good performer.

2. The sheet music is the same as the customer needs or requirements that the SE manages. Each instrument has the music written just for that instrument. Likewise, each specialist on the project must have the requirements written in the correct semantics for his design specialty.

3. The orchestra leader uses the baton to keep everyone in step; while the system engineer uses the progress review and plan to keep all the designers synchronized. Both can change the performance slightly in real time as the activity progresses: The orchestra leader through hand signals and the SE through changes in the plan.

We see that the managing and coordinating activities of the SE is synonymous to that of the orchestra leader. This metaphor should help us understand what a SE does.

ARCHITECTURE

Architecture is one of the most misused terms in SE. Everyone has a different definition and tries to use the metaphor of producing the drawings for a building. But architecture, Figure 6, can be defined using an open-ended definition based on problem solving. The definition requires an open-ended definition, because it depends on the problem or application. Just like process, methodology, and Systems Engineering the real definition depends on the problem, there doesn't seem to be a "one size fits all" definition that makes everyone happy. I believe there never will be.

Starting with problem solving as the reason for SE to exist, we have traveled down a winding road to understand what SE is. In the process we have come to understand what lessons learned, complexity, process, project team, and architecture are. All of these understandings are necessary to produce a successful product in the 21 st Century.

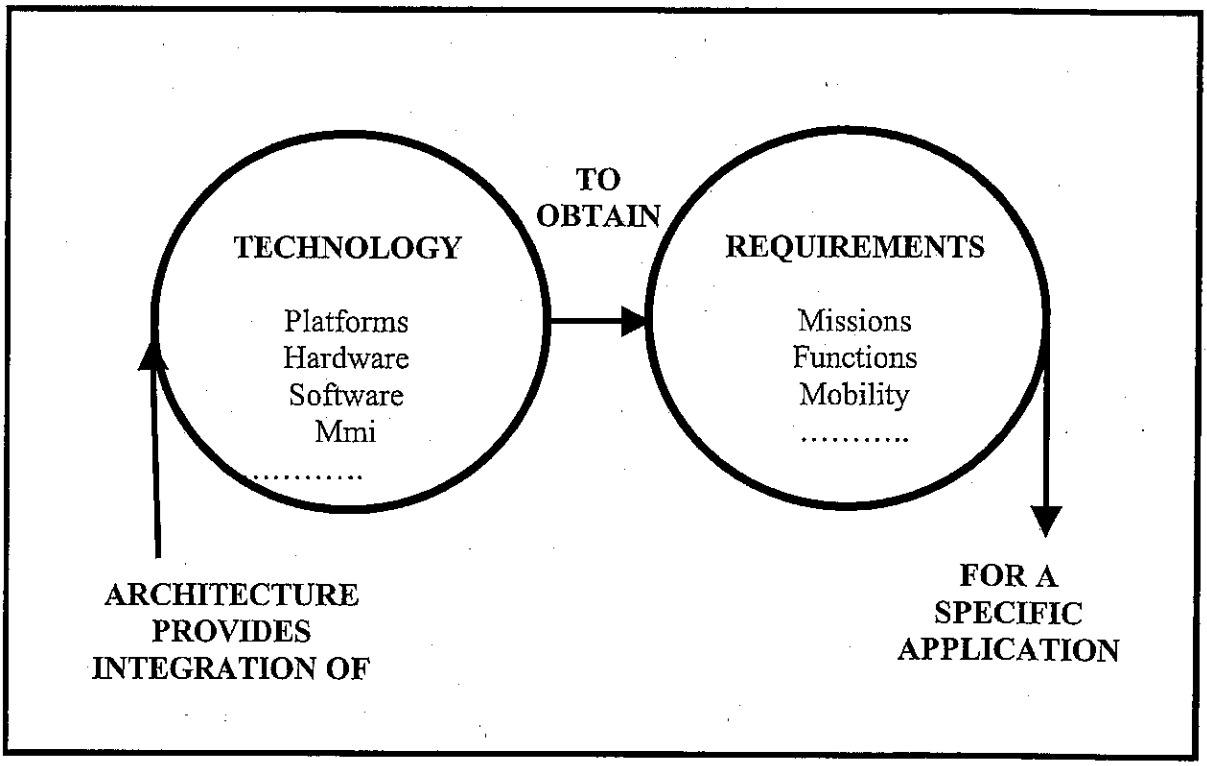

Figure 6 Definition of Architecture

BIOGRAPHY

Dr. Marvel has spent 20 Years in Industry, 8 Years in Government, and 18 years in Academia. He has been at the Naval Postgraduate School for past 12 years, teaching a number of variations of Systems Engineering, Command & Control and Information Systems courses.

REFERENCE LIST

Booch, Grady, Object Solutions, Addison-Wesley, ISBN 0-8053-0594-7, 1996