

## Systems Engineering – An Everyday Process

**Leslie L. Bordelon**

U S Air Force Senior Executive Service Retired  
38470 Shadow Valley Lane  
Yucaipa, California 92399-7010  
USA

[lbordelon@adelphia.net](mailto:lbordelon@adelphia.net)

### **ABSTRACT**

*The Systems Engineering Process evolved out of the need for an engineering approach that could deal with the technical and integration challenges posed by extremely large and complex systems that overwhelmed traditional development approaches. Military and civilian managers faced the daunting tasks of meeting concurrent development schedules, coordinating the work of large numbers of technical personnel, allocating requirements to multiple system elements, and adjusting interfaces for compatibility. Systems engineering emerged as an instrument for turning these issues into assets, but recently it has acquired a reputation as a tool appropriate only to large systems. This couldn't be farther from the truth. The simplicity of the process is explained and an example relating to everyday activities is described. Systems engineering provides both a set of principles and a process for ensuring the development of successful solutions to difficult problems. With an understanding of these principles, one can appropriately tailor the process to the development of any system or solution to any problem. In today's acquisition environment, it is extremely important for military and civilian managers to be thoroughly knowledgeable about processes proposed and used by the performing activity. This knowledge is required to adequately evaluate proposals, and to measure the effectiveness of the effort.*

### **1.0 INTRODUCTION**

Systems engineering traces its roots to the mid-1950s and the early ballistic missile development programs. Since that time it has continued to evolve and flourish in the ballistic missile community while also being employed in numerous other space, defense, and commercial enterprises to varying degrees of success. The inconsistency in its use and varying success is due to a number of causes, but chiefly to a lack of understanding as to how it is practiced and even why it is necessary! The intent of this paper is to contribute to an understanding of what systems engineering is and how it may be utilized. As to why systems engineering should be considered at all, examining its track record shows that it works and also adds value at the same time that it ensures success.

Systems engineering evolved out of the need for an engineering approach that could deal with the technical and integration challenges posed by extremely large and complex systems, such as the ballistic missile forces, that overwhelmed traditional development approaches. Military and civilian managers faced the daunting tasks of meeting concurrent development schedules, coordinating the work of large numbers of technical personnel, allocating requirements to multiple system elements, and adjusting interfaces for compatibility -- all at the forefront of technology.

Systems engineering emerged as an instrument for turning these issues into assets, but recently it has acquired a reputation as a tool appropriate only to large systems. This couldn't be farther from the truth. While specific areas can and should be tailored when applied to smaller systems, the underlying principles

## **Systems Engineering – An Everyday Process**

---

of systems engineering are as crucial to the development of small and medium systems as to large ones. Consider the following benefits that can be derived from the systems engineering process:

- Systems engineering centers around a top-down, requirements-oriented process that ensures that the system requirements and designs track the customer's requirements and programmatic constraints (cost and schedule) which tend to evolve over time.
- Systems engineering provides a means to address technically challenging concepts by permitting the system to be analyzed as a set of smaller problems through the use of functional decomposition.
- Systems engineering requires that the total life cycle of the system be considered from the start of the requirements analysis effort. This ensures that all critical requirements are identified before the design begins, serving to reduce program risk and cost.
- Systems engineering recognizes the importance of considering the contributions of multiple technical disciplines. A proper systems engineering approach ensures specialty engineering areas such as maintainability, human engineering, and safety are considered at the earliest possibility to reduce risk and cost and to provide the best system solution for the customer.
- Systems engineering is focused towards reaching a balanced system solution to satisfy the requirements of the customer. Tradeoff studies and effectiveness analyses are conducted throughout the systems development effort to ensure that the final solution set meets cost, schedule, and performance goals.
- Systems engineering imposes a discipline to ensure requirements traceability and the full justification and documentation of all engineering decisions.

Systems engineering provides both a set of principles and a process for ensuring the development of successful solutions to difficult problems. With an understanding of these principles, one can appropriately tailor the process to support the development of any system.

In today's acquisition environment, it is extremely important for Government (tasking activity) personnel to be thoroughly knowledgeable about processes proposed and used by the contractor (performing activity). This knowledge is required to adequately evaluate proposals, and to measure the effectiveness of the contractor's performance.

## **2.0 THE PROCESS**

A system can refer to a new product or process, an upgrade or modification to an existing system, or an engineering effort to resolve problems in a fielded system. The development approach involves: establishing and implementing a structured, disciplined project incorporating systems engineering principles; multidisciplinary teamwork; and, the simultaneous development of the product, process, and people solutions needed to satisfy customer needs. There is no "cookbook" for doing this. However, the Systems Engineering Process can provide solutions that meet identified needs.

The systems engineering process is an interdisciplinary approach encompassing the entire technical effort to develop and verify an integrated and life-cycle balanced set of solutions (people, product, and process) that satisfy customer needs. Systems engineering encompasses: the technical efforts related to the development, manufacturing, verification, deployment, operations, support, disposal of, and user training for system products and processes; the definition and management of the system configuration; the translation of the system definition into work breakdown structures; and the development of information for management decision making. The process continues until objectives are satisfied.

The main objective of the systems engineering process is to transform validated customer needs and requirements into a (properly documented) life-cycle balanced solution set of system product and process designs. The solution set is typically described by specifications at varying levels of detail (i.e., system/segment, development, or product specifications) depending on the acquisition phase. The problem and success criteria are defined through requirements analysis and functional analysis/allocation. Development of alternative solutions, evaluation of those alternatives, selection of the best life-cycle balanced solution, and the description of the solution through the design package are accomplished via systems synthesis and balance and control.

The systems engineering approach is particularly applicable to projects involving the advancement, development, and application of new technologies. It applies to large and small scale systems, to single or multiple procurements, to hardware intensive and software intensive systems, and to the replacement of current products and processes.

The systems engineering process operates iteratively throughout the system life cycle to translate stated problems into design requirements. Problems are normally expressed in terms of needs for new developments or modifications, or as deficiencies in operating, supporting, or providing training for already-fielded items. Performance-based requirements and alternative solutions to problems are then iteratively defined and refined. Solutions may employ existing, limited development, or emerging technologies transitioned from the technology base to product and process applications. Preferred solutions are selected based on cost, schedule, performance, and risks. Technical risk management is integral to the process and includes the identification, quantification, and impact assessment of risks, and the implementation of mitigation measures throughout the acquisition cycle.

A comprehensive, responsive verification effort is conducted to ensure that designs satisfy requirements. The purpose of this structured and disciplined process is to define and select optimal solutions for the stated system problems from a lifecycle perspective. The result is an integrated system solution consisting of people, products, and processes that satisfies customer needs.

The following should be considered in the implementation of any system development:

- Mission requirements should be expressed initially in broad operational capability terms using an approach for integrating requirements, and planning, and budgeting systems to satisfy operational needs, and organize for efficiency and effectiveness.
- Major decision points should identify mission needs that cannot be satisfied by non-materiel solutions. Threat projections, affordability constraints, cost-performance-schedule trade-offs, life-cycle costs, and risk management should be major considerations at each milestone.
- Acquisition strategies accomplish program objectives, control risk, and provide information for milestone decisions:
  - Event-driven acquisition strategies link contractual commitments and milestone decisions to development and testing demonstrations.
  - Systems engineering allows a balance among system design requirements.
  - Non-Government standards and commercial item descriptions describe commercial and other non-developmental items.
  - Solicitations and contract requirements are streamlined at program initiation and during each acquisition phase.
  - The contracting approach selected for each acquisition phase must permit equitable risk between Government and industry.

- Each milestone decision point establishes objectives for cost, schedule, and performance, and expands them for inclusion in subsequent program baselines.
- Performance objectives must satisfy operational needs, be verifiable by testing, and include critical supportability factors such as reliability, availability, and maintainability.
- User participation in acquisition is important in helping define feasible operational requirements. Without the constant cross-check by the user, needs may not be understood and solutions will not meet the requirements.
- Program risks and risk management plans are assessed at each milestone decision point before proceeding into the next acquisition phase.
  - Critical parameters impacting design cost, readiness capability, or life-cycle costs are identified early and managed intensively.
  - Technology demonstrations and prototyping, are coupled with early operational assessments to reduce risk.
  - Test and evaluation determine system maturity and identify areas of technical risk. Risk areas to be assessed at milestone decision points include: threat, technology, design and engineering, support, manufacturing, cost, and schedule.

Integrating the technical efforts to meet cost, schedule, and performance objectives with an optimal design encompassing manufacturing, test, and support processes is necessary. Systems engineering is applied throughout the system life cycle as a comprehensive, iterative technical management process to:

- Satisfy an operational need with a concurrently integrated system (and related manufacturing, test, and support process) design;
- Integrate the technical inputs of the entire development community to meet established program cost, schedule, and performance objectives;
- And ensure the compatibility of functional and physical interfaces; that the system design reflects the requirements for all its elements; and develop risk abatement approaches, and characterize and reduce technical risk through early test and demonstration of system elements.

A number of key tasks become important as systems engineering process presses forward. When a capability or technology becomes an operational requirement, a requirements collection and translation methodology identifies critical operational characteristics and constraints to convert into detailed design specifications. The design specification is balanced by trading-off requirements to optimize the system, and allocating tasks and resources to transition technology into the specification. Methods to accomplish the technology transition criteria are defined before proceeding into engineering development.

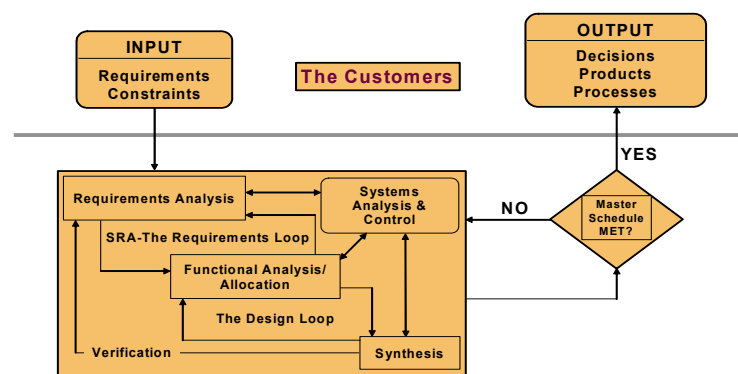
Technical risks are identified and assessed throughout the acquisition cycle. The acquisition strategy should include provisions for reducing the risks to acceptable levels. The effects of technical risk on program cost and schedule, risk reduction measures and rationale, and alternative acquisition strategies are assessed at each milestone decision point.

A verification process integrates design analysis, design simulation, and demonstration and test. All critical characteristics are identified, and required performance levels verified, by demonstration and test. The tests are designed to demonstrate operational effectiveness and suitability and prove the manufacturing processes.

Systems development requires integrating technical disciplines to analyze the design specification, conduct trade-offs, balance system requirements, and establish the final configuration.

Planning and control includes engineering planning, technical performance measures, configuration management, and technical data management. Planning for major SE events is included in the program acquisition strategy. A management plan should be used to document management of the SE process, integration of technical specialties, performance measures reporting, and key milestones and schedules. Performance measures should also be included to assess how well the design meets requirements. Configuration management should be used to control system design throughout the system life cycle. The formal product of the SE process is usable technical data.

A work breakdown structure provides a task-oriented framework relating statements of work, contract line items, configuration items, technical and management reports, and hardware, software, and data elements of the system.



**The Systems Engineering Process**

There are four basic elements in the SE Process.

- **REQUIREMENTS ANALYSIS** defines and refines the problem to determine top-level performance and functional requirements for the system.
- **FUNCTIONAL ANALYSIS AND ALLOCATION** breaks down the top-level functional requirements into the lowest level actions the system must perform and allocates performance to each of those actions.
- **SYNTHESIS** determines alternative product and process solutions--“How Can It Be Done?”--to the functional and performance requirements. Key Point: The design doesn’t begin until here!
- **SYSTEMS ANALYSIS AND CONTROL**--the “Doable” block--balances the requirements and designs with solutions that best satisfy the customer’s need.

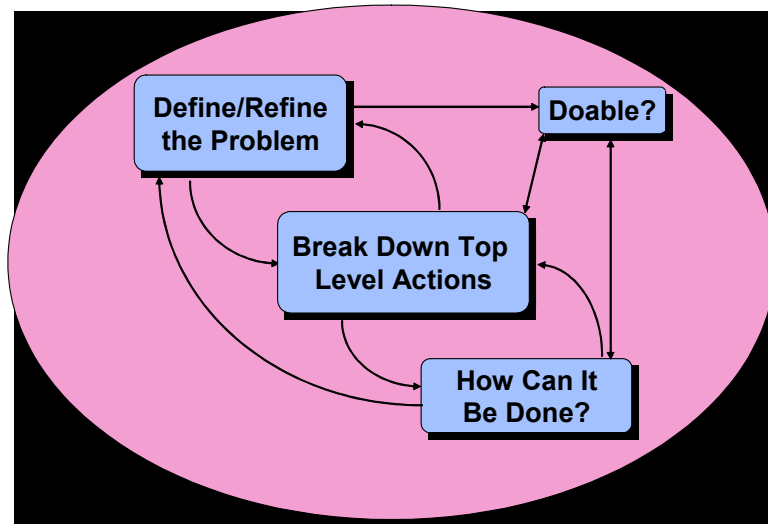
The application of Systems Engineering is very much the same as applying any other problem solving technique. A generic technique which can be found in various books on problem solution employs the following steps in getting to a solution to a problem:

- **Problem Recognition.** This is the first step where we define the problem--just like in the SE Process when the user identifies a need, a threat that cannot be met, and provides us with operational requirements--the input. In this technique this also includes understanding the environment the system operates within, which relates to **INPUT** and **REQUIREMENTS ANALYSIS**.

- Objectives, Requirements, Constraints. The operational requirements become the objective from which we develop performance specifications. The user has also furnished us with some constraints. This encompasses both REQUIREMENTS ANALYSIS and FUNCTIONAL ANALYSIS / ALLOCATIONS.
- Model Construction. In this technique this step is establishing the method for which best alternatives are selected--the decision model. This roughly correlates to our objectives of meeting cost, schedule, and performance requirements. Can it do the job within the specified constraints? We do this in the SE Process as part of the REQUIREMENTS ANALYSIS effort.
- List Alternatives. Here we start listing the solution--SYNTHESIS.
- Evaluate Alternatives. You compare alternatives to the requirements, measure effectiveness, etc. This equates to SYNTHESIS and its FEEDBACK LOOP with the FUNCTIONAL ANALYSIS / ALLOCATION effort. As defined in the reference, this also touches on SYSTEMS ANALYSIS AND CONTROL.
- Select Best Alternative. This correlates to SYNTHESIS and SYSTEMS ANALYSIS AND CONTROL where you ensure you indeed picked the best solution. The best alternative is an OUTPUT.
- Data Collection. This block relates to all the other efforts, just as SYSTEMS ANALYSIS AND CONTROL does within the SE Process. You collect data in order to provide feedback on whether the solution does meet the performance requirements, constraints, etc. Just like in the SE Process, this effort is so critical to all the other steps, it is shown as a supporting function that pervades the entire process and impinges on each of the steps. The collected data is an OUTPUT.

So the SE Process is really just a problem solving technique, except the SE Process is applied to the system acquisition problem throughout the life cycle, including both acquisition AND sustainment.

We'll use a simpler view to further define the SE Process. The Input and Output will just be shown as arrows leading in to and out of the process. As we stated before, Requirements Analysis is defining and refining the problem. Functional Analysis / Allocation further breaks down top-level actions to lower levels and Synthesis is determining how these can be done. Finally, Systems Analysis and Control is asking, "Is it doable?"



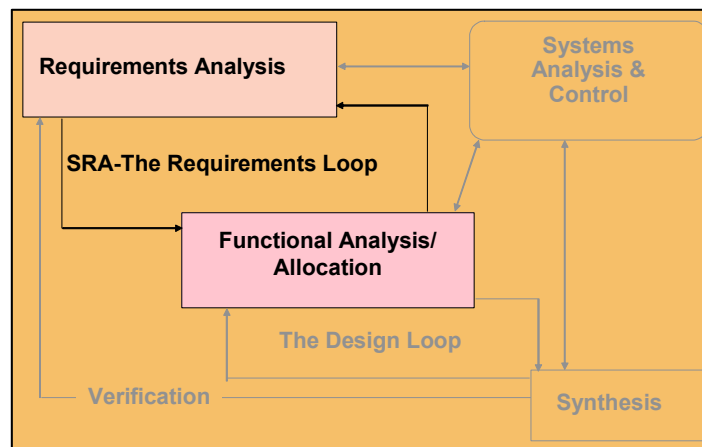
**A Simple Look at Systems Engineering**

Requirements analysis (define/refine the problem) reviews mission requirements, utilization environments, and system characteristics to determine functional and performance requirements (define and refine the problem) for each function.

- Requirements take the form of customer needs:
  - What the customer needs, how well the customer needs it to work,
  - Where the customer needs it,
  - How soon the customer needs it, and
  - How much the customer needs it to cost.
 Requirements also take the form of measurement criteria (how to measure goodness).
- Critical technologies from development programs are transitioned to program efforts. Performance requirements for critical technologies should be monitored via Technical Performance Measurement. Methods to identify technology alternatives, and selection criteria should be defined.
- To support management decisions, performance requirements are developed based on life cycle factors such as degree of certainty or criticality. Information is required to support technical effort, technology base data, outputs from a previous phase, and program constraints.
- Selection of preferred solutions is based on cost, schedule, performance, and risk. Technical risk management uses risk mitigation measures throughout the acquisition cycle. Progressive verification from components up through the total system and across the eight primary system functions yields optimal solutions from a life cycle perspective.

Functional analysis/allocation (break down top level actions) turns big actions into smaller, simpler actions that are solvable. Functional requirements should be used as the top-level functions for functional analysis and allocation. Performance requirements should be successively established for each functional requirement and interface.

- Functional analysis/allocation is conducted iteratively to:
  - Define successively lower-level functions required to satisfy higher level functional requirements,
  - Establish lower level performance requirements and design constraints, and
  - Define performance requirements across all functions based on system life cycle factors.
- Performance requirements should be traceable throughout the functional architecture, through the analysis by which they were allocated, to the higher-level requirement they are intended to fulfil.



### System Requirements Analysis—The Feedback Loops

The key factor is to maintain a consistent, disciplined process for identifying, analyzing, and documenting customer needs and translating them into “system” requirements over the life cycle. This analysis is an evolving definition of performance and functional requirements--all of the requirements, all of the time. The evolving nature is depicted in the loops. As requirements are defined and refined, a life cycle focus is maintained for all products and processes. This requirements loop is depicted as the feedback loops. Feedback loops ensure all actions have been addressed and corrective actions taken when necessary; solutions and alternative solutions to requirements are iteratively defined and refined; and designs satisfy requirements (solutions solve problems) through a comprehensive, responsive verification effort.

Out of the requirements and functional analysis/allocation, a functional architecture is defined that Synthesis (how can it be done) converts into products and processes. Synthesis translates performance and functional requirements (including interfaces) into product and process solution descriptions. Synthesis has two parts: one part characterizes the best solution to requirements, and the other part designs the best solution.

- Synthesis uses a bottom-up approach, where system elements are organized into sets, then candidate system solutions are assembled.
- To increase system definition, the definitions of the elements and internal interfaces of the selected system architecture are integrated into a fully defined system description.



- Synthesis may be iterative, alternating between bottom-up and top-down, to arrive at a set of candidate architectures and a menu of elements which are consistent with each other and reasonably exhaustive of the available range of possibility.
- One product or process may be a solution to satisfy different sets of functions. Synthesis collects a set of functional requirements and develops solution alternatives for them.

A Systems Analysis and Control program is implemented to answer the question of “Doable.” The systems analysis portion includes trade-off studies, effectiveness analyses and assessments, and design analyses to determine progress in satisfying technical requirements and program objectives. The control mechanisms include risk management, configuration management, data management, and performance-based progress measurement including the master schedule, technical performance measurement, and technical reviews.

Determining “Best” Solution(s) requires evaluating each alternative from a cost, schedule, performance and risk perspective. Requirements are sometimes altered, alternative sets of functions developed, and alternative solutions devised. A master schedule provides the events plan (not calendar driven) which includes reviews and other technical events to measure progress before proceeding with technical efforts. For each event in the schedule, critical tasks are identified that must be achieved before initiating the event. It is essential that the success criteria be measurable. The effort should be documented to ensure everyone is working to the same version.

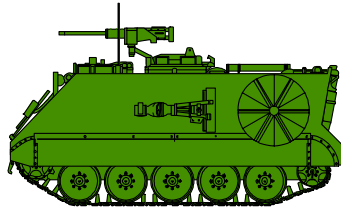
Systems Analysis and Control measures progress, evaluates alternatives, and documents decisions. In addressing all parts of the problem, solution alternatives are evaluated for impact on system effectiveness, life cycle resources, risk, and customer requirements. To ensure continuity from bottom to top, product and process design requirements need to be traceable to the functional and performance requirements they were designed to fulfil. Analysis and Control examines lower level (follow-on) actions to see if they have caused new constraints or modified existing ones. Lower level actions are examined to ensure all constraints have been addressed. Expanded problem definition ensures all actions will be done as well as needed.

This completes the introduction to the SE Process. However, an example is probably the best way to visualize an application. SE requires interdisciplinary decision making from “Lust-to-Dust” of our weapons systems acquisition strategies. “Dust” means the environmental impacts of “retiring” a weapons system are planned for, even before the system is on the design drawing board. In this way, “Lust-to-Dust” transcends traditional life cycle considerations. As an example of SE at work, analyzing a simple task, such as changing the oil in an Armoured Personnel Carrier or “APC,” can minimize the cost and time required, and yet still get the job done.

### 3.0 THE EXAMPLE

**Systems Engineering Process** 

**An Illustrative Example**  
**APC Requires Oil Change**




Let's go to the customer for the requirements.

**Systems Engineering Process** 

**Customer Input**

**Objective:** Change the Oil

**Inputs:**

- Spend Only a "Reasonable" Amount of Money
- **Unskilled Labor Available** 
- Buy Off-the-Shelf Parts
- Don't Spend More Than 4 Hours



First, we'll start off with a few inputs. We want to do the job as cheap as possible. This means using cheap labor and also using off-the-shelf parts to avoid a lengthy and costly design phase. Four hours sounds like a reasonable length of time.

**Systems Engineering Process** 

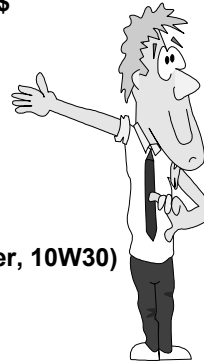
**Define/Refine The Problem?**

**Identify Constraints**

- Take No More Than 4 Hours
- Use Only a “Reasonable” Amount of \$\$
- Use COTS

**Identify Top Level Actions**

- Acquire Materials
  - What Materials? (Define Them)
    - Oil
      - What Amount? (4 Quarts)
      - What Kind? Season? (Winter, 10W30)
    - New Filter? (Yes)
- Remove Old Oil
- Put In New Oil



So far, all of our inputs have become constraints, so we’ll list them as such. Next, we should identify what actions we’ll need to take to get the job done, such as get the correct oil and how to go about replacing the “goo” in our “APC” with it!

**Systems Engineering Process** 

**Break Down Top-Level Actions**

**Acquire Materials**

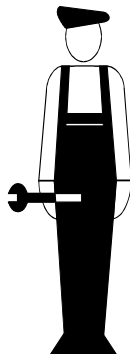
- Buy Oil and Filter

**Remove Old Oil**

- Remove Oil Filter
- Remove Drain Bolt

**Put In New Oil**

- Put On New Oil Filter
- Insert Drain Bolt
- Remove Oil Cap
- Add Oil
- Replace Oil Cap



“Turning Big Ones Into Little Ones”

So far, our list has been pretty simple, but we need to organize our activities so we can apply SE techniques to minimize the parameters of our constraints.

**Systems Engineering Process** 

**Divvy-Up “How Well” - Check Completeness**

Assign Performance (Time/Cost) to the Actions and See How We Are Satisfying the Problem.

- Buy Oil and Filter ↔ Nearest APC parts store - 45 min round trip
  - Remove Oil Filter ↔
  - Remove Drain Bolt
  - Put On New Oil Filter
  - Insert Drain Bolt
  - Remove Oil Cap
  - Add Oil ←————→ 5 min
  - Replace Oil Cap
- 80% of oil drains 0.5 sec after removing oil filter. Drains 1 quart per 30 sec from oil pan drain.

First, we’ll determine how long it takes to do each action in our list of requirements.

**Systems Engineering Process** 

**Divvy-Up “How Well” - Sum “How Well’s”**  
*time ; cost*

Buy Oil and Filter.....	45 min; \$10	<p style="text-align: center;">To Date:</p> <p>Total Time - 4.5 hr</p> <p>Total Cost: \$10.00</p>
Remove Oil Filter .....	? sec	
Remove Drain Bolt .....	? sec	
• Oil Drain Time .....	10 min	
Put On New Oil Filter .....	? sec	
Insert Drain Bolt .....	? sec	
Remove Oil Cap .....	? sec	
Add Oil .....	5 min	
Replace Oil Cap .....	? sec	
Clean-Up Oil Spill .....	2.5 hrs	
Shower with Lava .....	1 hr	

Now let’s make a cost and time chart to see how well we’re minimizing our constraints.

Not bad with the total cost hovering at ten dollars, but the 4.5 hour time requirement is a bit over our 4 hour limit.

**Systems Engineering Process** 

**Is It Doable?**

**“Deconflict & Balance Requirements”**

- “Clean-Ups Take Too Long
- Potential Environmental Problems
  - “New” Requirement - Disposal
    - New & Revised Actions
      - Catch the Oil in a Container
      - Remove Drain Plug & Oil, Then Remove Filter

**“Broke” Constraint:  
4.5 hr > 4 hr**

**Refine Problem and Actions:**

- Obtain 5 Quart Catch Pan (Catch Oil)
- Empty Gallon Milk Carton (Dispose of Oil)
- Plastic Bag (Dispose of Oil Filter)

It appears that we exceeded the time limit because our cleanup time takes too long. Since SE covers “Lust-to-Dust,” we should think about the environment and plan how we are going to dispose of the oil once it’s out of the “APC.” Now we have an additional list of actions we have to take because of environmental considerations.

**Systems Engineering Process** 

**Divvy-Up “How Well”**

- Buy Oil, Filter, Pan (New) ...45 min, \$10.00
- Remove Drain Bolt ..... ? sec [reorder sequence]
  - Oil Drain Time ..... 10 min [reorder sequence]
- Remove Oil Filter ..... ? sec [reorder sequence]
- Put On New Oil Filter ..... ? sec
- Insert Drain Bolt ..... ? sec
- Remove Oil Cap ..... ? sec
- Add Oil ..... 5 min
- Replace Oil Cap ..... ? sec
- Clean-Ups ..... 1 hour [revised]

**To Date:  
Total Time - 2 hours  
Total Cost - \$10.00**

Refining our plan for draining the oil, cleaning up the spill, and cleaning up ourselves has resulted in a time savings, so the total clean-up time is now 1 hour. This gives us a new total time of 2 hours and still an old total cost of 10 dollars. But we still have lots of question marks instead of time estimates, so we need to fill them in, add them up, and see what our total time to do the job will be.

**Systems Engineering Process**



**How Can It Be Done ?**

Develop Solution (s)

Design-to-Time Available ~ 2 hour

Analysis

- Remove Drain Bolt: Bolt Design - 3/4" Hex Nut (Wrench Adequate, Cost = \$4, R&R Time = 4 min)
- Remove Oil Filter: Special Tool Required (Oil Filter Wrench, Cost = \$10, R&R Time = 10 min)
- Remove Oil Cap: Hand Tighten (3" Wrench to remove, Cost = \$4, R&R Time = 1 min)

**Note: An adjustable wrench could work for both the drain bolt and the oil cap (Cost = \$5, Savings = \$3)**

Filling in some of the question marks from our previous slide, it will take a total of 15 minutes to remove and replace the drain bolt, remove and replace the oil filter, and remove and replace the oil cap. We have also figured out a way to save 3 dollars by using an adjustable wrench for both the oil cap and the drain bolt. Also, this gives us one less wrench to misplace. It looks like we're stuck with the oil filter wrench.

**Systems Engineering Process**



**How Can It Be Done ?**

Develop Solution (s)

Problem:

- Mechanic doesn't fit under APC, and can't reach drain bolt & oil filter from the side.

Alternatives:

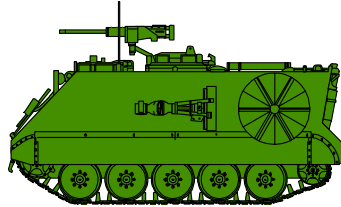
- Get Mechanic with Longer Arms
- Lower the Ground Underneath Oil Pan Area
- Raise the APC
  - Park APC with Front Wheels on the Curb
  - Use Fork Lift Truck to Raise Front End (\$25, 10 min)
  - Obtain APC Ramps (\$50, 30 min Set-Up Time)
- Take APC to Dealership (\$20, Leave APC All Day)

This is my favorite part. We can't get under the APC to get to the drain bolt and oil filter. As you can see, we have lots of ways of attacking this problem, but let's analyze alternative #1 more closely.

Systems Engineering Process



Alternative #1  
Get Mechanic With Longer Arms



ELIMINATE:  
Overqualified for Job

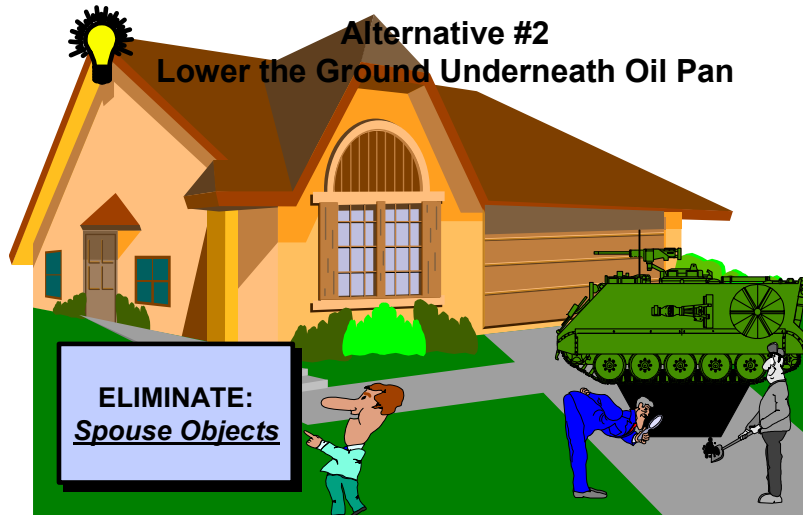


If we really could find a mechanic smart enough and with long enough arms to do the job, he'd probably rather hire on with a circus, or start his own computer software business.

Systems Engineering Process



Alternative #2  
Lower the Ground Underneath Oil Pan



Moving on to alternative #2, if you were really stupid enough to dig a hole under the driveway, you wouldn't have to worry about your spouse objecting. Your spouse would have already left you.

Systems Engineering Process



Alternative #3  
Park APC with Front Wheels on the Curb



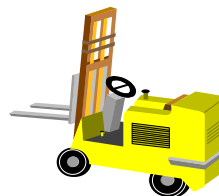
Alternative #3 also has problems. A person could get into trouble hanging his APC out in the street over a curb to change the oil.

Systems Engineering Process

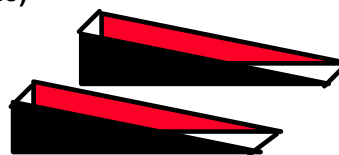


Candidate Alternatives

- Use Fork-Lift Truck to Raise Front End (\$25, 10 min)



- Obtain APC Ramps (\$50, 30 min Set-Up Time)



Moving on to other alternatives, we have a fork lift truck (which only takes us 10 minutes total time to get, raise the front end, lower the front end, and return to wherever we got it), and a set of APC ramps which take 1/2 hour to shove into place. I think we may assume that 50 dollars will BUY the APC ramps, and 25 dollars only RENTS the fork lift truck.



**Systems Engineering Process** 

Are Alternatives Doable?

<u>Tasks</u>	<u>Lift Truck</u>	<u>Ramps</u>	<u>Dealer</u>
• Buy oil, filter	\$10, 45 min	\$10, 45 min	
• Buy pan, wrenches*	\$20, 45 min	\$20, 45 min	
• Lift front end	\$25, 10 min	\$50, 30 min	
• Remove/replace drain bolt	4 min	4 min	
• Drain oil	10 min	10 min	
• Remove/replace oil filter	10 min	10 min	
• Remove/replace oil cap	1 min	1 min	
• Add oil	5 min	5 min	
• <u>Clean-ups</u>	<u>1 hour</u>	<u>1 hour</u>	
<b>Total Time</b>	<b>3 hr 10 min</b>	<b>3 hr, 40 min</b>	<b>1 Day</b>
<b>Total Cost</b>	<b>\$55</b>	<b>\$80</b>	<b>\$20</b>

\* One Time Cost **RETAIN DEALER: TOO LONG, BUT LEAST COST**

Well, I was wrong! We also rented the ramps. The reason it took so long to use them is because we had to push them home-from and back-to the ramp rental dealer. So our oil change under both the ramp and the fork lift truck alternatives is too expensive. Although taking the APC to a dealer takes a whole day, which is too long, at least the price is reasonable.

**Systems Engineering Process** 

**Life Cycle Considerations**

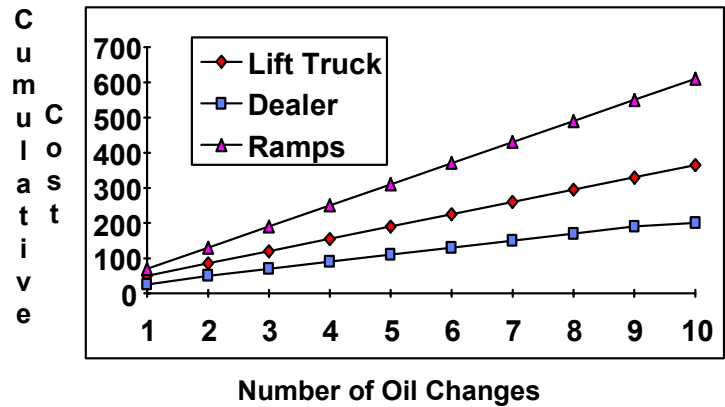
(But How Many Times Will the Oil Be Changed?)

	<u>Lift Truck</u>	<u>Ramps</u>	<u>Dealer</u>
<b>One-Time Purchases</b> (Pan and wrenches)	\$20	\$20	\$ 0
<b>One-Time Delay</b> (Buying materials)	45 min	45 min	0 min
<b>Recurring Costs</b>	\$35	\$60	\$20
<b>Recurring Time Delay</b>	2 hr, 25 min	2 hr, 55 min	1 day

Let's look at this from a life cycle standpoint: the lift truck method has the lowest recurring cost and lowest recurring time, but the recurring cost is still \$35, which is a bit pricey. Using ramps offers no visible advantage, plus it is more expensive and more time consuming. Using the dealer is the cheapest alternative if one can stand having one's APC tied up for a whole day at the garage.

**Systems Engineering Process** 

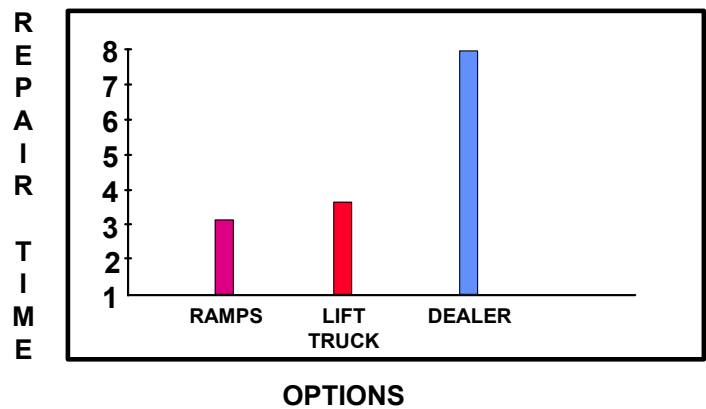
**Life Cycle Assessment**



This chart compares the cumulative costs of the three alternatives. From a purely life cycle cost perspective, we might opt for the dealer alternative. However, judging from the relatively low non-recurring costs of the ramps, and probable ease of maintenance, it would be advisable to investigate the cost of purchasing a set of ramps, and keep them on the premises for successive oil changes. You might understand this rationale if we take a closer look at the time-to-repair comparison.

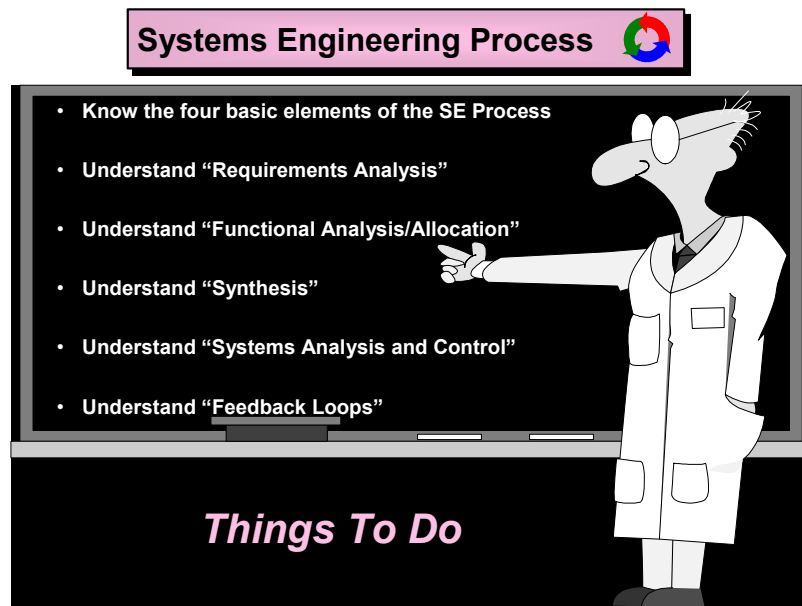
**Systems Engineering Process** 

**Repair Time Assessment**



If you recall, our customer is the warfighter, then you'll appreciate that time-to-repair might just be the real priority. In this crude example time-to-repair equates to mission readiness. We'd probably want to discuss this kind of trade-off with our user. The customer may be willing to trade the higher life cycle cost for higher readiness rates--the mission may dictate it!

Hopefully, this example gives you a clearer perspective that the SE Process really is just a problem solving technique, applicable to any problem.



As a recap: there are four basic elements in the SE Process.

- **REQUIREMENTS ANALYSIS** defines and refines the problem to determine top-level performance and functional requirements for the system.
- **FUNCTIONAL ANALYSIS AND ALLOCATION** breaks down the top-level functional requirements into simpler actions the system must perform and allocates performance to each of those actions.
- **SYNTHESIS** determines alternative product and process solutions to the functional and performance requirements. **Key Point:** The design doesn't begin until here!
- **SYSTEMS ANALYSIS AND CONTROL** balances the requirements and designs with solutions that best satisfy the customer's need.

**FEEDBACK LOOPS** ensure: all actions have been addressed and corrective actions taken when necessary; solutions and alternative solutions to requirements are iteratively defined and refined; and designs satisfy requirements (solutions solve problems) through a comprehensive, responsive verification effort.

### 4.0 REFERENCES

- [1] Bordelon, L., Kordik, J. and Johnson, V. (1995). *Systems Engineering Process Guide (Unpublished)*, U. S. Air Force Space and Missile Systems Center
- [2] Systems Engineering Process Action Team (1992). *Systems Engineering and Configuration Management Life Cycle Application Guide (Draft)*, U. S. Air Force Materiel Command
- [3] Systems Engineering Process Action Team (1992). *Systems Engineering Training Supplement to Life Cycle Application Guide*, U. S. Air Force Materiel Command
- [4] Kordik, J. and Bordelon, L. (1995), *Systems Engineering Training Presentation on Changing Oil in APC*, U. S. Air Force Materiel Command