8th Conference on Systems Engineering Research
March 17-19, 2010, Hoboken, NJ

Paper #1569269915

Unifying the different systems engineering processes

Joseph Kasser
Visiting Associate Professor
Temasek Defence Systems Institute
National University of Singapore
Block E1, #05-05, 1 Engineering Drive 2
Singapore 117576
joseph.kasser@incose.org

Derek Hitchins
ASTEM
Consultant Systems Architect

"It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so." - Mark Twain 1835-1910.

Abstract

Teaching the systems engineering process is difficult because of the contradictory and confusing process information in the literature as well as the overlap between the systems engineering process and the problem solving process as well as the confusion between the systems engineering process and the system lifecycle. This paper resolves the conflict and confusion and documents two separate meta-systems engineering processes; a 'planning' process that produces the planning documents and a 'doing' process in which the plan is implemented. The result of this research is a meta-model of the two systems engineering processes that not only facilitate teaching by showing that all documented systems engineering processes are subsets of the meta-processes, but also show how agile systems engineering, lean systems engineering and evolutionary acquisition all fit together in an integrated manner.

Introduction

In teaching systems engineering the author[1] has observed that students that come into the class knowing some systems engineering come out of the class knowing a little more systems engineering, while students that come into the class not knowing systems engineering, come out of the class not knowing it a little less. Reflection on this situation has indicated that there may be ways to improve the way the systems engineering process (SEP) is taught, including:

1. Pointing out the myth of the single systems engineering process.
2. Explaining the overlap between some versions of the SEP and the problem solving process.
3. The way iteration of/in the SEP is taught.
4. The misuse of functional diagrams to represent processes.

Consider these four points.

[1] J. Kasser.

The myth of the single systems engineering process

According to the (United States Department of Defense 5000 Guidebook 4.1.1), "The successful implementation of proven, disciplined SEPs results in a total system solution that is--

- *Robust to changing technical, production, and operating environments;*
- *Adaptive to the needs of the user; and*
- *Balanced among the multiple requirements, design considerations, design constraints, and program budgets*".

(Arnold, 2000) quotes (MIL-STD-499B, 1993) and (IEEE 1220, 1998) stating "*a single process, standardizing the scope, purpose and a set of development actions, has been traditionally associated with systems engineering*". However, there is no single widely agreed upon SEP since over the years, the SEP has been stated in many different ways, including:

- The (EIA 632, 1994) and (IEEE 1220, 1998) processes shown in Figure 1 and Figure 2;
- Lists of processes in ISO/IEC 15288 (Arnold, 2002);
- The waterfall process (Royce, 1970);
- The V model version of the process;
- The spiral, incremental and evolutionary models;
- System Lifecycle functions (Blanchard and Fabrycky, 1981) shown in Figure 3;
- State, Investigate, Model, Integrate, Launch, Assess and Re-evaluate (SIMILAR) (Bahill and Gissing, 1998) shown in Figure 5;
- The basic core concepts accepted by most systems engineers (Mar, 2009);
- A systems engineering approach to addressing a problem (Hitchins, 2007).

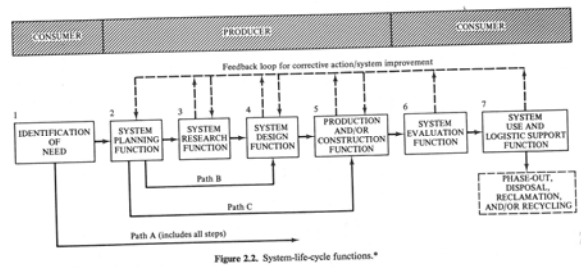

Figure 1 System Lifecycle functions (Blanchard and Fabrycky, 1981)

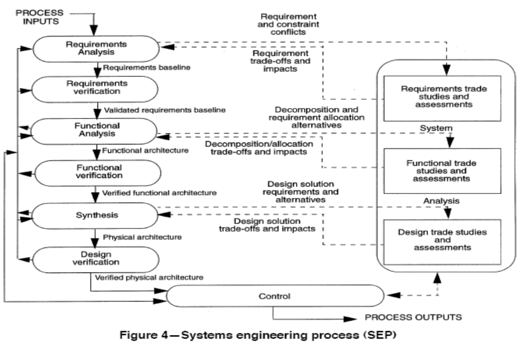

Figure 3 IEEE 1220 Systems Engineering Process

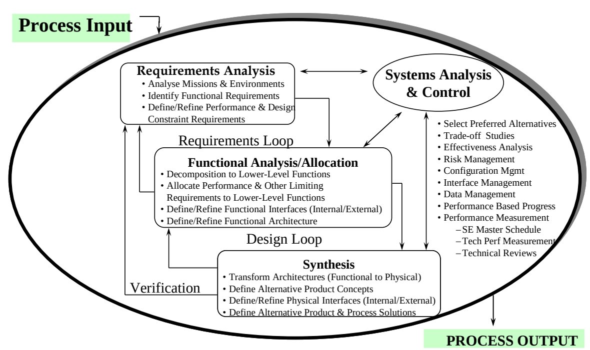

Figure 2 ANSI/EIA-632 Egg diagram

Consequently, given the conflicting and contradictory information in the various versions of the SEPs, the SEP concept is difficult to explain and, teaching has focused on using the waterfall and V models since while not representative of the real world, they are simple to explain (Biemer and Sage, 2009) pages 152 and 153).

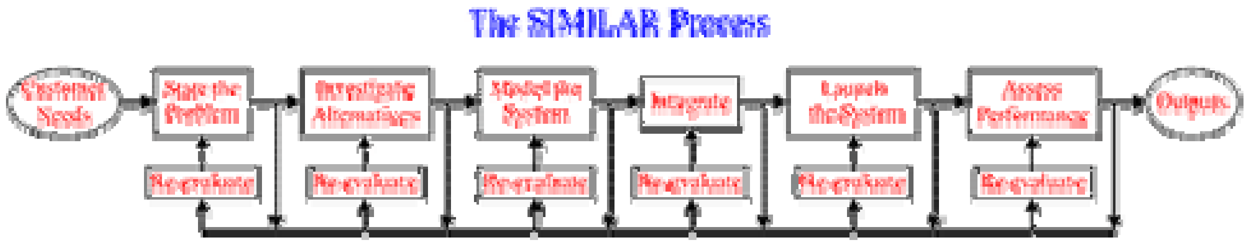

Figure 5 The SIMILAR process (Bahill and Gissing, 1998)

The key insight to understanding the reason for the variety of SEPs may lie with (Biemer and Sage, 2009) page 153) who state that "*the systems engineer creates a unique process for his or her particular development effort*". Consider each published version of the SEP[2] as the *unique process created for their particular development effort* by someone or some group at some point in time, at some point in the system lifecycle, in the context of what they defined as a systems engineering problem and subsequently documented as their SEP.

Looking for patterns in the various versions of the SEP listed above as well as others in the literature, one can identity versions that:

- focus on early stage systems engineering where the problem is explored and conceptual solutions developed;
- focus on engineering the system and realizing the solution;
- focus on both aspects.

A process is a sequence of activities. The SEP takes place in the context of the Hitchins-Kasser-Massie Framework (HKMF) for understanding systems engineering (Kasser, 2007) shown in Figure 4. The HKMF is two dimensional where:

- The vertical dimension of the HKMF contains the five-layers of systems engineering (Hitchins, 2000).
- The horizontal dimension of the framework represents the system lifecycle from conception to disposal.

Each area in the HKMF contains a set of systems engineering and non-systems engineering activities from which a process may be constructed. Column A contains activities that address the initial problem and conceptual solution (Kasser, et al., 2009). Columns B, C, D, E and F contain the activities that realize the solution.

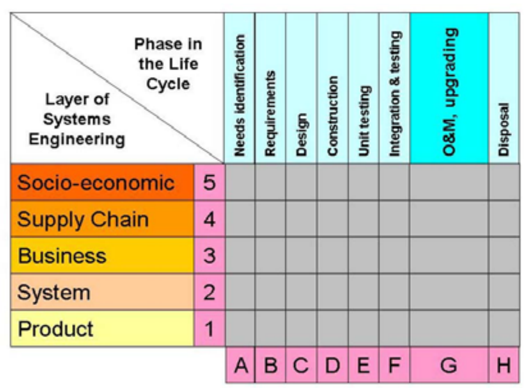

Figure 4 The HKM Framework for understanding systems engineering

From the big picture perspective, there seem to be two interdependent SEPs:

1. The traditional 'doing' SEP in which Layer 2 systems engineering is performed. This is the unique SEP which is constructed for the realization of a specific system. The activities performed in the unique SEP will depend on the problem-identification-solution-realization activities that have and have not been done at the time the unique SEP is con-

[2] In a Standard or in a text book.

Table 1 Two versions of the problem solving process

GDRC, 2009	OVAE, 2005
1. Problem Definition 2. Problem Analysis. 3. Generating possible Solutions. 4. Analyzing the Solutions. 5. Selecting the best Solution(s). 6. Planning the next course of action (Next Steps)	1. Identify and Select the Problem 2. Analyze the Problem 3. Generate Potential Solutions 4. Select and Plan the Solution 5. Implement the Solution 6. Evaluate the Solution

structed.

2. The planning SEP; the process used by the systems engineer to create the unique SEP. When designing/planning the unique SEP for the realization of a system, systems engineers use knowledge based on experience and the activities functions and processes which can be found in the processes and Standards listed above and in the literature. This planning process is a problem solving activity, consequently it ought to, and does, map into the problem solving process.

The overlap between some versions of the systems engineering process and the problem solving process.

(Mar, 2009) stated that *"most systems engineers accept the following basic core concepts*[3]*:*

1. *Understand the whole problem before you try to solve it*
2. *Translate the problem into measurable requirements*
3. *Examine all feasible alternatives before selecting a solution*
4. *Make sure you consider the total system life cycle. The birth to death concept extends to maintenance, replacement and decommission. If these are not considered in the other tasks, major life cycle costs can be ignored.*
5. *Make sure to test the total system before delivering it."*

Two outlines of generic problem solving processes (GDRC, 2009) and (OVAE, 2005) are shown in Table 1. If one compares these examples of the problem solving process with the functions (Blanchard and Fabrycky, 1981) shown in Figure 3, the (SIMILAR) process (Bahill and Gissing, 1998) shown in Figure 5 and assertion by (Mar, 2009), it can be seen that while the aggregation of activities into the steps are different and the level of detail in each step is different, these versions of the SEP seem to be the same as the problem solving process.

The confusion between the SEP and the problem solving process can be resolved by recognizing that from the problem solving perspective, a man-made system is realized as a solution to a problem. As such, the parts of the SEP that takes place in Columns A and C of the HKMF constitute problem solving processes.

The way iteration of/in the systems engineering process is taught

The iterative nature of systems engineering has sometimes been taught using the egg diagram from the (EIA 632, 1994) as being applicable to each phase of the system life cycle in the manner shown in Figure 6.

[3] Of systems engineering, Author's interpretation.

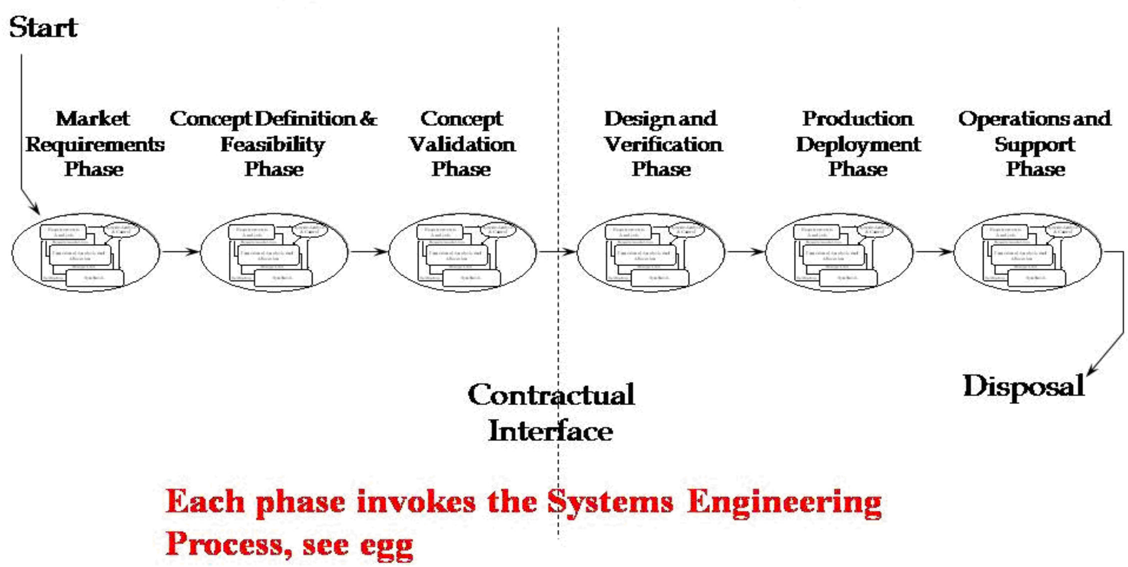

Figure 6 A Typical System Lifecycle (UNiSA, 2006)

Since the contents of the egg are expressed in Layer 2 realization language it is difficult to get the concept of iteration across to the students is because the words have incorrect meanings in the other phases and layers. In addition, this teaching approach only addresses a part of the iterative nature of systems engineering which includes the following four types of iteration.

1. Iteration inside an area of the HKMF.
2. Iteration across a row of the HKMF.
3. Iteration in a column of the HKMF.
4. Iteration of a number of system lifecycles in series.

Consider each of them.

Iteration inside an area of the HKMF. This type of iteration takes place when a process is repeated during the production of a product. It is generally drawn as a circular sequence of activities. Two examples are:

1. The iterative part of the process for producing a document. The system engineer produces a version of a document, circulates it for comment, receives comments, incorporates the comments in the document and circulates the document for further comment (Kasser, 1995) pages 158 – 160). This writing-reviewing-update loop takes place until the criterion for terminating the loop is reached. Examples of such termination criteria include the document is complete, or the scheduled date for delivering the document has been reached.
2. The design activities which take place in Columns A.2 and A.3 of the HKMF (Kasser, et al., 2009).

Iteration across a row of the HKMF. This type of iteration takes place when the same sequence of activities is performed in more than one column of the HKMF. An example is the design process which takes place at the conceptual level[4] in Column A and at the realization level in Column C.

Iteration in a column of the HKMF. Systems engineering is a problem solving discipline and while the types of problems that are found in each column are different, the tools and techniques used to solve them will

---

[4] To complicate the situation, the conceptual design process in Column A also iterates a realization design process to the extent needed to show that the conceptual design is feasible and to identify any risks associated with realizing that design (e.g. technological, schedule, etc.).

be different but the problem solving approach will most likely be the same. For example,

- A Layer 3 situation dealing with human issues may require an adaption of an appropriate methodology such as the Soft Systems Methodology (Checkland and Scholes, 1990) while a Layer 2 situation in the same column applying to a different project may require the use of Quality Function Deployment (QFD) (Clausing, 1994), queuing theory, linear programming or some other mathematical approach.
- In government acquisitions in Layer 2, the preferred implementation option determined in Column A is often to outsource the realization and proving phases to a contractor. In such a situation, Column A contains the activities that would produce:
  - the acquisition plan;
  - the tender or request for proposal;
  - the tender or proposal evaluation and selection of development contractor;
  - the contract for the realization and proving phases in Columns B to F.
- If the disposal method for a system has not been predetermined, Column H may cycle though Columns A to F for the system disposal project.

Iteration of a number of system lifecycles in series. This type of iteration has a number of names including evolutionary acquisition, sequential software Builds and the cataract process (Kasser, 2002). Two ways of thinking about the nature of this type of iteration are

1. To consider the activities in Columns A to F as the first iteration through the SEP. Changes are requested in the performance of the system during Column G. Configuration control allocates a set of changes to an upgraded version and the activities in Columns A to F take place for each new version of the system.
2. To use two HKMFs, one placed below the other such that the end of Column H of the top HKMF is in line with the start of Column G of the lower HKMF.

The misuse of functional diagrams to represent processes

Another teaching difficulty is built into the graphical representations used in teaching, Consider typical examples; the (EIA 632, 1994) version shown in Figure 1, the (IEEE 1220, 1998) version shown in Figure 2, (Bahill and Gissing, 1998)'s SIMILAR process shown in Figure 5 and (Blanchard and Fabrycky, 1981)'s system lifecycle functions shown in Figure 3. These figures do not show processes they show functions and the relationship between the functions. Looping back from the end of a sequence of functions to the start is generally added in functional drawings by adding a feedback arrow from the output of a function to the input of a previous function. Processes take time to implement and time does not flow backwards so there shall not be feedback lines in a process. Systems engineers proc-

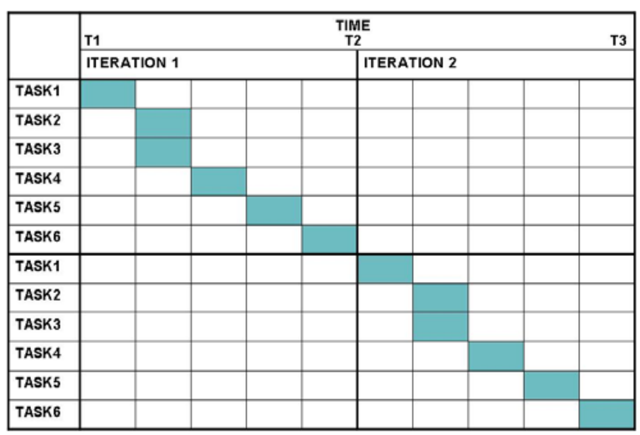

Figure 7 Gantt chart representation of iteration

ess architect (Kasser and Palmer, 2005) or create (Biemer and Sage, 2009) page 153) a unique SEP for realizing a system. The resulting SEP is depicted in the form of Gantt or PERT style charts not in drawings containing feedback loops. For example, when a sequence of six tasks need to be repeated (iterated) the process shall be drawn as a Gantt chart showing a second set of activities right-shifted as shown in Figure 7 instead of as a flow chart with a feedback link from the end of Task 6 to the start of Task 1.

The common systems engineering process

Systems engineering is a problem solving activity as discussed above. Systems engineers process architect (Kasser and Palmer, 2005) or create (Biemer and Sage, 2009) page 153) a unique SEP for realizing a system. The planning process they use to create the unique SEP for realizing a system should be common to all systems engineering activities. If the systems engineering activity is considered as a project, then a common meta-SEP can be created by combining the (Hitchins, 2007) page 173) and (Mar, 2009) approaches into the following 10-step sequence that joins the problem solving process and the solution realization process:

1 Plan the project.
2 Explore/survey the problem space.
3 Conceive at least two feasible ways to tackling the problem by solving, resolving, dissolving or absolving it (Ackoff, 1999) page 115). If the problem is to be absolved, then proceed directly to Step 10.
4 Identify ideal selection criteria for evaluation of the feasible ways of addressing the problem.
5 Perform tradeoffs to determine and select the best way of addressing the problem.
6 Fine tune selected option.
7 Formulate strategies and plans to realize preferred option.
8 Realize preferred option.
9 Verify that preferred option tackled the problem.
10 Terminate the project.

Notes:

a) Step 1 may sometimes take place in less detail before the project begins to determine that there is a need for the project and to allocate an initial set of resources.
b) Step 1 should include a review of best practices lessons learned from previous and similar projects to determine what worked and what did not work (the both the process and product domains) in the context of the similar projects, and the nature of the differences between the similar projects and this one.
c) In practice, Steps 3 and 4 may be conducted in parallel, not sequentially.
d) Iteration may take place as discussed above.
e) Step 10 includes documenting the lessons learned from the project.

Restating this meta-problem-solving-solution-realization process as a 'planning' process to create the unique SEP, the wording would be:

1 Plan the project that will create the required planning documentation for the unique SEP that will realize the solution system.
2 Explore/survey what needs to be done.
3 Conceive at least two feasible SEPs.
4 Identify ideal selection criteria for evaluation of the SEPs.
5 Perform tradeoffs to determine and select the best SEP.
6 Fine tune selected SEP.
7 Formulate strategies and plans to realize preferred SEP.

8 Document preferred SEP using activities as building blocks in the appropriate planning documentation.

9 Obtain stakeholder consensus that the planned unique ‘doing’ SEP can realize the solution system.

10 Terminate the project. This step begins the transition from ‘planning’ to ‘doing’.

Lean and agile systems engineering

Lean and agile systems engineering do not need special processes and treatment when designing the unique SEP to realize a system since:

- **Lean systems engineering** takes place when the unique SEP designed to realize the system does not contain any non-productive activities. Since non-productive activities are wasteful, lean systems engineering should be the norm.
- **Agile systems engineering** takes place when the system lifecycle is short enough to deliver a solution in time to deal with the problem. In a situation where the problem changes during the time the solution is being developed, the SEP should iterate a number of system lifecycles to provide timely solutions to the changing problems. This is the evolutionary paradigm albeit with a shorter than usual lifecycle time and is not a special case of systems engineering.

Summary

Starting with the observation that in teaching systems engineering students that come into the class knowing some systems engineering come out of the class knowing a little more systems engineering, while students that come into the class not knowing systems engineering, come out of the class not knowing it a little less, the paper discussed four problems associated with the way the systems engineering process is taught and clarified some of the confusion and contradictory information associated with current teaching approaches.

Conclusions

This research has shown that the single SEP is a myth in the way the SEP is currently taught. From the big picture perspective, there seem to be two interdependent meta-systems engineering processes, one for ‘planning’ and one for ‘doing’ or realizing the solution system:

The unique ‘doing’ SEP is constructed for the realization of a specific system. When designing the unique SEP for the realization of a system in the areas of the HKMF to be inhabited by the unique SEP, systems engineers use knowledge based on experience and the activities functions and processes which can be found in the processes and Standards listed above and in the literature as building blocks. The activities to be performed in the unique SEP will depend on the work that has and has not been done at the point in the system lifecycle in which the process is constructed.

The second meta-SEP is the ‘planning’ process used by the systems engineer to create the unique SEP. Since this process is a problem solving activity, it ought to, and does, map into the problem solving process.

This research has also shown that agile systems engineering and lean systems engineering are not special cases of systems engineering and should be the norm.

The research has also clarified the conflicting and contradictory information in the various versions of the SEPs by viewing them as different unique subsets of the meta-SEPs appropriate to their situation.

The research has also determined that the columns in the HKMF may need adjusting. The HKMF was developed from the

(EIA 632, 1994) and (IEEE 1220, 1998) perspectives and rolled up the early phases of the system lifecycle into a single phase labeled 'needs identification'. However, during the course of developing a framework for a systems engineering body of knowledge (Kasser, et al., 2009; Kasser and Hitchins, 2009) Column A, the 'needs identification' phase had to be expanded into three sub-phases to properly address the problem exploration and solution determination phases inside Column A.

This research has raised the need for further research to explore aligning the columns (phases of the system lifecycle) with the appropriate steps of the problem solving process perhaps by expanding Column A and rolling up Columns C, D, E and part of F into a single 'realization' column. This change would keep the number of columns manageable and should further facilitate teaching about the SEP.

Since the unique SEP is constructed from 'building blocks' described in the Standards, text books and other literature, further research should be performed to create standard set of building blocks for the systems engineering and non-systems engineering activities in each area of the HKMF. Use of these process building blocks would be similar to the way electronic engineers use digital integrated circuits to create digital circuits. The building blocks would have a standard format such as the task statement described in (Kasser, 1995). Inclusion of the building blocks and a software agent to verify that the blocks are linked together (in a similar manner to the way requirements management tools monitor traceability) would be a useful way of adding intelligence to project management tools.

References

Ackoff, R. L., *Ackoff's Best. His Classic Writings on Management*, John Wiley & Sons, Inc., New York, 1999.

Arnold, S., "*Systems Engineering: From Process towards Profession*", proceedings of The 10th Annual Symposium of the INCOSE, Minneapolis, MN, 2000.

Arnold, S. (Editor), *ISO 15288 Systems engineering — System life cycle processes*, International Standards Organisation, 2002.

Bahill, A. T. and Gissing, B., "*Re-evaluating systems engineering concepts using systems thinking*", IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, Vol. 28 (1998), no. 4, 516-527.

Biemer, S. M. and Sage, A. P., "Systems Engineering: Basic Concepts and Life Cycle," in *Agent-Directed Simulation and Systems Engineering*, L. Yilmaz and T. Oren (Editors), Wiley-VCH, Weinheim, 2009.

Blanchard, B. and Fabrycky, W., *Systems Engineering and Analysis*, Prentice Hall, 1981.

Checkland, P. and Scholes, J., *Soft Systems Methodology in Action*, John Wiley & Sons, 1990.

Clausing, D., *Total Quality Development*, ASME Press, 1994.

EIA 632, "EIA 632 Standard: Processes for engineering a system," 1994.

GDRC, The Problem Solving Process, 2009, http://www.gdrc.org/decision/problem-solve.html, last accessed 11 Jan 2009

Hitchins, D. K., World Class Systems Engineering - the five layer Model, 2000, http://www.hitchins.net/5layer.html, last accessed 3 November 2006

Hitchins, D. K., *Systems Engineering. A 21st Century Systems Methodology*, John Wiley & Sons Ltd., Chichester, England, 2007.

IEEE 1220, "Standard 1220 IEEE Standard for Application and Management of the Systems Engineering Process," 1998.

Kasser, J. E., *Applying Total Quality Management to Systems Engineering*, Artech House, Boston, 1995.

Kasser, J. E., "*The Cataract Methodology for Systems and Software Acquisition"*, proceedings of SETE 2002, Sydney Australia, 2002.

Kasser, J. E., *A Framework for Understanding Systems Engineering*, Booksurge Ltd, 2007.

Kasser, J. E., Hitchins, D. and Huynh, T. V., "*Reengineering Systems Engineering"*, proceedings of the 3rd Annual Asia-Pacific Conference on Systems Engineering (APCOSE), Singapore, 2009.

Kasser, J. E. and Hitchins, D. K., "A framework for a systems engineering body of knowledge, 0.6," Report to the Fellows Committee, International Symposium of the International Council on Systems Engineering, Singapore, 2009.

Kasser, J. E. and Palmer, K., "*Reducing and Managing Complexity by Changing the Boundaries of the System"*, proceedings of the Conference on Systems Engineering Research, Hoboken NJ, 2005.

Mar, B., Commentary on the Consensus of INCOSE Fellows, 2009, http://www.incose.org/practice/fellowsconsensus.aspx, last accessed 8 November 2009

MIL-STD-499B, "Draft MIL-STD-499B Systems Engineering," United States Department of Defense, 1993.

OVAE, Problem-Solving Process, 2005, http://www.c-pal.net/course/module3/pdf/Week3_Lesson21.pdf, last accessed 11 Jan 2009

Royce, W. W., "*Managing the Development of Large Software Systems"*, proceedings of IEEE WESCON, 1970.

UNiSA, "Module 7: Introduction to Traditional Systems Engineering and Life Cycle Modelling," in *Systems Engineering for Complex Problem Solving*, University of South Australia, Adelaide, Australia, 2006.