

A Value-Based Theory of Systems Engineering

Barry Boehm, Apurva Jain
Computer Science Department
University of Southern California
Los Angeles, CA 90089
{boehm, apurvaja}@usc.edu

Abstract. The INCOSE definition of “systems engineering” is “an interdisciplinary approach and means to enable the realization of successful systems.” The Value-Based Theory of Systems Engineering presents necessary and sufficient conditions for realizing a successful system and elaborates them into an executable process. The theory and process are illustrated on a supply-chain system example, and evaluated with respect to criteria for a good theory.

Introduction

A good deal of discussion around the development of the INCOSE Technical Vision [Crisp et. al., 2005] has been concerned with the questions of:

- What is the distinguishing intellectual content of systems engineering as compared to other engineering disciplines?
- What is a theoretical basis for systems engineering? What is its content and structure? How does it satisfy the criteria for a good theory? How does it address emerging challenges such as sociotechnical systems, emergent requirements, rapid change, and complex systems of systems? How can it be used in the practice of systems engineering?

A good place to start is with the INCOSE definition of systems engineering: An interdisciplinary approach and means to enable the realization of successful systems. A good set of objectives for a theory of systems engineering would therefore be to characterize the nature of “successful systems” and the means for realizing them.

The theory below presents a set of necessary and sufficient conditions for a system to be successful, and a set of necessary steps for realizing a successful system. Its results depend largely on dealing with the value propositions of the system’s success critical stakeholders. It thereby advances the proposition that the ability to deal with value considerations is a key factor in the distinguishing intellectual content of systems engineering (Another is the ability to deal with combinations of heterogeneous components and humans).

The intellectual content of most engineering disciplines is component-oriented and value neutral. Ohm’s Law, Hooke’s Law, and Newton’s Laws are excellent for determining how various physical components perform. But they do not address the contributions of this performance to the value of a system involving various possible combinations of heterogeneous physical components and humans.

A Value-Based Theory of Systems Engineering (VBTSE)

Figure 1 summarizes the “4+1” structure of the VBTSE. The engine in the center is the success-critical stakeholder (SCS) win-win Theory W [Boehm and Ross, 1989], which addresses the

questions of “what values are important?” and “how is success assured?” for a given systems engineering enterprise. The four additional theories that it draws upon are dependency theory (how do dependencies affect value realization? On what stakeholders does success depend), utility theory (how important are the values?), decision theory (how do stakeholders’ values determine decisions?), and control theory (how to adapt to change and control value realization?).

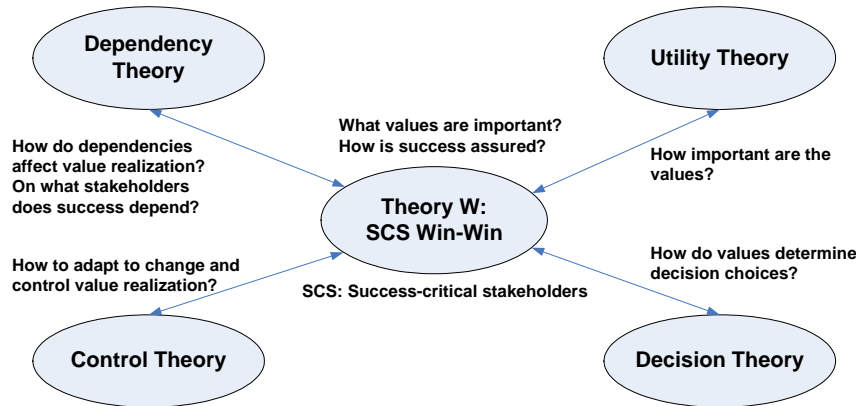


Figure 1. The "4+1" Structure of VBTSE

The Central Engine: Theory W

The core of Theory W is the Fundamental System Success Theorem, which states that: A system will succeed if and only if it makes winners of its success-critical stakeholders.

An informal proof follows. It should be noted that value-based theorems and proofs are less formal than those in such areas as mathematics and physics.

Proof of “if”:

1. Everyone significant is a winner.
2. Nobody significant is left to complain.

Proof of “only if”:

1. Nobody wants to lose.
2. Prospective losers will refuse to participate, or will counterattack.
3. The usual result is lose-lose.

The proof of “if” is reasonably clear. The proof of “only if” may not be so clear, so we illustrate it in three frequently-occurring examples of the primary stakeholders in an enterprise involving a customer contracting with a developer for a software system that will benefit a community of users, as shown in Figure 2.

Proposed Solution	“Winner”	Loser
Quick, Cheap, Sloppy Product	Developer & Customer	User
Lots of “bells and whistles”	Developer & User	Customer
Driving too hard a bargain	Customer & User	Developer

Figure 2. Win-lose Generally Becomes Lose-Lose

In Case 1, the customer and developer attempt to win at the expense of the user by skimping on effort and quality. When presented with the product, the user refuses to use it, leaving everyone a loser with respect to their expectations.

In Case 2, the developer and user attempt to win at the expense of the customer (usually on a cost-plus contract) by adding numerous low-value “bells and whistles” to the product. When the customer’s budget is exhausted without a resulting value-adding product, again everyone is a loser with respect to their expectations.

In Case 3, the user and customer compile an ambitious set of features to be developed and pressure competing developers to bid low or lose the competition. Once on contract, the surviving bidder will usually counterattack by colluding with the user or customer to convert the project into Case 2 (adding user bells and whistles with funded Engineering Change Proposals) or Case 1 (saying, for example, “The contract specifies user-friendly error messages. For my programmers, a memory dump is a user-friendly error message and thus is a contractually compliant deliverable”). Again, everyone is a loser with respect to their expectations.

The System Success Realization Theorem. However, the Fundamental System Success Theorem does not tell us how to realize and maintain a win-win state. This requires the

System Success Realization Theorem: Making winners of your success-critical stakeholders requires:

1. Identifying all of the success-critical stakeholders (SCSs).
2. Understanding how the SCSs want to win.
3. Having the SCSs negotiate a win-win set of product and process plans.
4. Controlling progress toward SCS win-win realization, including adaptation to change.

Identifying all of the SCSs: Dependency Theory

Identifying all of the SCSs is in the province of dependency theory (how do dependencies affect value realization?). Dependency theory covers much of the traditional theory of physical systems engineering such as in [Wymore, 1967], along with its extensions into “soft systems engineering” such as in [Checkland, 1981] and [Booher, 2003].

One aspect of dependency theory addresses theories about stakeholder interdependencies such as sociology and organization theories [Parsons, 1977; March and Simon, 1958; Argyris, 1978; Rifkin, 2004; Daft, 2003]; theories about human systems integration, particularly the integration of macroergonomic and microergonomic concerns [Booher, 2003]; and theories about how people and initiatives combine to realize successful systems, such as in value chains [Baldwin et. al., 2000] and results chains [Thorp, 1998]. Results chains and their extensions are good approaches for identifying a system’s success-critical stakeholders, as illustrated in Figure 4 below.

Other aspects of dependencies cover product interdependencies such as physics, electrical engineering, aerospace engineering, civil engineering, computer science, and architectural theories [Alexander, 1979; Rechtin, 1991; Shaw and Garlan, 1996]; and process interdependencies such as PERT/Critical Path Methods [Wiest and Levy, 1977], system dynamics [Forrester, 1961], and network flow theory [Ford and Fulkerson, 1962]. Dependency

theory also covers general interdependencies among products, processes, and stakeholders from such viewpoints as constraint theory [Friedman, 1976; Goldratt, 1985; Friedman, 2005]; optimization theory [Dantzig, 1963; Bellman 1957; Intriligator, 2002]; engineering economics [Newman, 2004; Marschak and Radner, 1972; Boehm 1981]; operations research [Churchman, 1957]; management theory [Simon, 1969; Cyert and March, 1963; Womack and Jones, 1996; Koskela and Howell, 2002]; and general systems engineering text books and handbooks. The Handbook of Systems Engineering and Management [Sage and Rouse, 1999] is a good source of recent coverage of stakeholder interdependencies (Chapters 16-21), product interdependencies (Chapters 8, 12, 13, 23, 25), process interdependencies (Chapters 1, 2, 14, 15, 29) and general interdependencies.

Understanding how the SCSs want to win: Utility Theory

Understanding how the SCSs want to win (the second predicate in the System Success Realization Theorem) is in the province of utility theory (how important are the values?) [Dupuit, 1844; Debreu 1959; Fishburn, 1982]. Misunderstanding SCS utility functions does not guarantee failure if an enterprise happens to get lucky. But again, understanding how the SCSs want to win is essentially a necessary condition for WinWin achievement. Utility theory also has several branches such as the satisficing theory of bounded rationality [Simon, 1957], multi-attribute utility theory [Keeney-Raiffa, 1976], and its situation-dependent aspects such as the Maslow need hierarchy [Maslow, 1954] stating that lower-level needs (food and drink; safety and security) have dominant utilities when unsatisfied and negligible utilities when satisfied.

Having the SCSs negotiate win-win plans: Decision Theory

Having the SCSs negotiate win-win plans is in the province of decision theory (how do stakeholders' values determine decisions?). Decision theory also has many aspects such as negotiation theory [Raiffa, 1982; Fisher-Ury, 1981], game theory [von Neumann-Morgenstern, 1944; Luce-Raiffa, 1957], multi-attribute decision theory [Keeney-Raiffa, 1976], statistical decision theory and the buying of information to reduce risk [Blackwell-Girshick, 1954], real options theory [Luehrman, 1998] and the Theory of Justice [Rawls, 1971].

Getting to a Win-Win Decision. Navigating through all of these decision options can be rather complex. One aid in the stakeholder win-win negotiation context is the win-win equilibrium theory in [Boehm-Bose, 1994] and [Lee, 1996]. The Win-Win Negotiation Model begins with the success-critical stakeholders (SCSs) identifying their win conditions (or value propositions) about the system to be developed and evolved. The SCSs can include considerably more classes than users, customers, and developers. Additional SCS classes can include maintainers, administrators, interoperators of co-dependent systems, testers, marketers, venture capitalists, and, as in [Rawls, 1971], representatives of the least-advantaged people whose health, lives, or quality of life may be affected by the system.

Besides Win Conditions (a synonym for stakeholder value propositions or utility functions), the Win-Win Negotiation Model involves Agreements (in which all the SCSs agree to adopt a win condition or an option), Issues (in which an SCS can identify a conflict between their and others' win conditions), and Options (proposals for resolving issues by expanding the option space). Agreements can also be reached by having the SCSs agree to adopt an option to resolve an issue.

A WinWin equilibrium state holds when all the win conditions are covered by agreements, and there are no outstanding issues. At the beginning of a negotiation, this is true by default. As soon as a stakeholder enters a win condition, the other stakeholders can all accept it via an agreement, in which case the WinWin equilibrium state still holds, or some stakeholder enters an issue and an associated conflicting win condition. The negotiation then leaves the WinWin equilibrium state, and the stakeholders attempt to formulate options to resolve the issue. The negotiation proceeds until all of the stakeholders' win conditions are entered and the WinWin equilibrium state is achieved, or until the stakeholders agree that the project should be disbanded because some issues are irresolvable. In such situations, it is much preferable to determine this before rather than after developing the system. And in terms of the System Success Realization Theorem, this also makes negotiating win-win plans a necessary condition for WinWin achievement.

Controlling progress toward SCS win-win realization: Control Theory

Controlling progress toward SCS win-win realization (the fourth and final predicate in the System Success Realization Theorem) is in the province of control theory (how to adapt to change and control value realization). As summarized in [Brogan, 1974] the necessary conditions for successful enterprise control are observability (the ability to observe the current enterprise state), predictability (the ability to predict whether the enterprise is heading toward an unacceptable state), controllability (the ability to redirect the enterprise toward an acceptable near-term state and a successful end state), and stability (the avoidance of positive feedback cycles that cause control systems to overcompensate and become unstable).

Particularly for VBTSE, it is more important to apply control theory principles to the expected value being realized by the project rather than just to project progress with respect to plans. Traditional "earned value" systems have their uses, but they need to be complemented by business-value and mission-value achievement monitoring and control systems as discussed in [Boehm-Huang, 2003]. These involve the use of risk management; adaptive control functions such as market watch and plan renegotiation; and multi-criteria control mechanisms such as BTOPP [Scott Morton, 1991; Thorp, 1998] and balanced scorecards [Kaplan-Norton, 1996]. Particularly in an era of increasing rates of change, this makes both traditional and adaptive control [Highsmith, 2000] necessary conditions for system success in terms of the System Success Realization Theorem.

Using and Testing the VBTSE: Process Framework and Example

In this Section, we present in Figure 3 a seven-step process-oriented expansion of the VBTSE framework shown in Figure 1, and will then apply it step-by-step to a supply chain management system development example. In the next section, we will use the results to evaluate how well it addresses a set of criteria for a good theory.

Step 1 of the process starts with a protagonist or change agent who provides the motivating force to get a new project, initiative, or enterprise started. As examples, protagonists can be organization leaders with goals, authority, and resources, entrepreneurs with goals and resources, inventors with goals and ideas, or consortia with shared goals and distributed leadership and resources. (For further process details in a software context, see [Boehm and Jain, 2005].)

Each class of protagonist will take a somewhat different approach in visiting the seven main steps in Figure 3 to create and sustain a win-win combination of SCSs to achieve their goals. In this Section, we will trace the approach taken by a leader whose goals involve a combination of opportunities and problems, who has the authority and resources to address the goals, and who is open to different ideas for addressing them. She is Susan Swanson, an experienced MBA-holding executive, former bicycling champion, and newly-hired CEO of Sierra Mountainbikes, Inc. (a fictitious company representative of two similar companies with less successful projects).

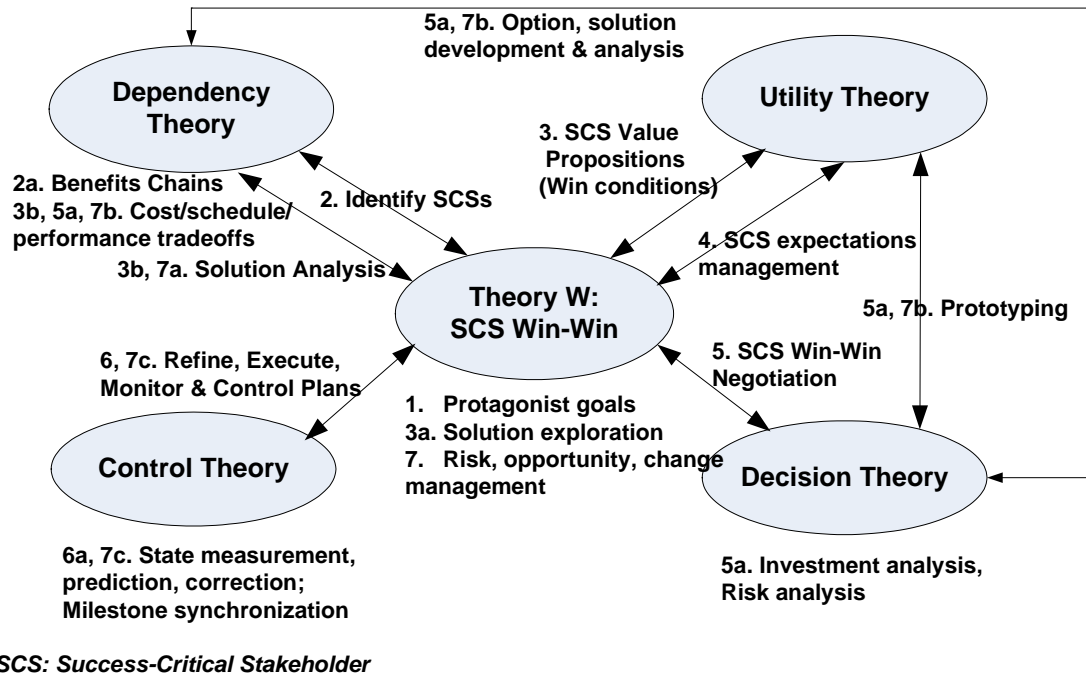


Figure 3. Process-Oriented Expansion of 4+1 VBTSE Framework

Sierra Mountainbikes Opportunities and Problems

Susan began by convening her management and technology leaders, along with a couple of external consultants, to develop a constructive shared vision of Sierra Mountainbikes’ primary opportunities and problems. The results determined a significant opportunity for growth, as Sierra’s bicycles were considered top quality and competitively priced. The major problem area was in Sierra’s old manual order processing system. Distributors, retailers, and customers were very frustrated with the high rates of late or wrong deliveries; poor synchronization between order entry, confirmation, and fulfillment; and disorganized responses to problem situations. As sales volumes increased, the problems and overhead expenses continued to escalate.

In considering solution options, Susan and her Sierra team concluded that since their primary core competence was in bicycles rather than software, their best strategy would be to outsource the development of a new order processing system, but to do it in a way that gave the external developers a share in the system’s success. As a result, to address these problems, Sierra entered into a strategic partnership with eServices Inc. for joint development of a new order processing and fulfillment system. eServices was a growing innovator in the development of supply chain

management systems (an inventor with ideas looking for protagonist leaders with compatible goals and resources to apply their ideas).

Step 2: Identifying the Success-Critical Stakeholders (SCSs)

Step 2 in the process version of the VBTSE shown in Figure 3 involves identifying all of the success-critical stakeholders involved in achieving a project's goals. As seen in Figure 4, the Step 2a Benefits Chain jointly determined by Sierra and eServices, this includes not only the sales personnel, distributors, retailers, and customers involved in order processing, but also the suppliers involved in timely delivery of Sierra's bicycle components (our Benefits Chain extension to the Thorp/DMR Results Chain includes identifying SCSs in parallelograms and unifying Assumptions into a table).

The Benefits Chain includes initiatives to integrate the new system with an upgrade of Sierra's supplier, financial, production, and human resource management information systems. The Sierra-eServices strategic partnership is organized around both the system's benefits chain and business case, so that both parties share in the responsibilities and rewards of realizing the system's benefits. Thus, both parties share a motivation to understand and accommodate each other's value propositions or win conditions and to use value-based feedback control to manage the program of initiatives.

This illustrates the "only if" part of the Fundamental System Success Theorem. If Susan had been a traditional cost-cutting, short-horizon executive, Sierra would have contracted for a lowest-bidder order processing system using Case 3 in Figure 2, and would have ended up with a buggy, unmaintainable stovepipe order processing system and many downstream order-fulfillment and supplier problems to plague its future. In terms of the VBTSE process in Figure 3, however, Sierra and eServices used the Benefits Chain form of Dependency Theory to identify additional SCSs (sales personnel, distributors, retailers, customers, suppliers) who also need to be brought into the SCS WinWin equilibrium state.

Steps 3 and 4: Understanding SCS Value Propositions; Managing Expectations

Step 3 (understanding all of the SCSs' value propositions or win conditions) primarily involves utility theory. But it also involves Theory W in reconciling SCS win conditions with achievable solutions (Step 3a), and various forms of dependency theory in conducting cost/schedule/performance solution tradeoff and sensitivity analyses (Step 3b).

For example, the suppliers and distributors may identify some complex exception reporting, trend analysis, and customer relations management features they would like to have in the system's Initial Operational Capability (IOC) in early 2005. However, the use of forms of dependency theory such as software cost and schedule estimation models may show that there is too much proposed IOC software to try to develop by the IOC date. In such a case, Sierra and eServices will have to revisit the SCSs' utility functions in Step 4 (expectations management) by showing them the cost and schedule model credentials and results, and asking them to recalibrate their utility functions, prioritize their desired features, and participate in further solution exploration (a go-back to Step 3a) to achieve a win-win consensus on the top-priority subset of features to include in the IOC.

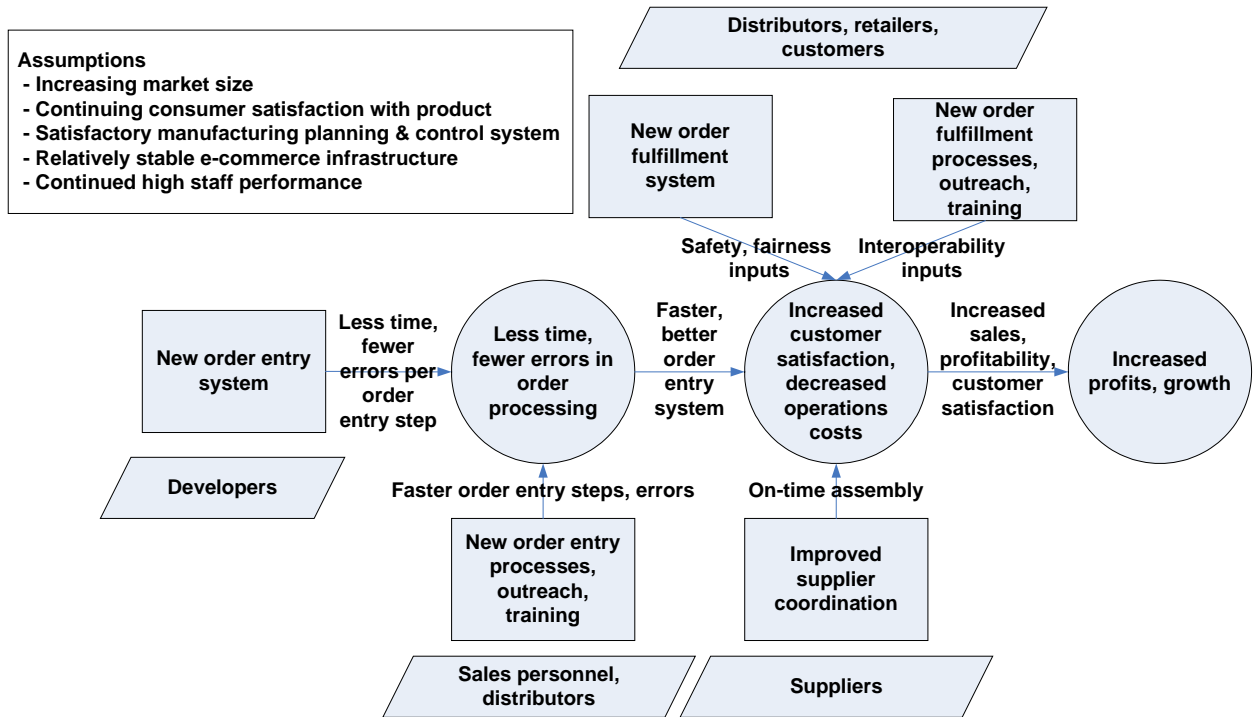


Figure 4. Benefits Chain for Sierra Supply Chain Management

It may be in some cases that the SCSs' IOC needs are irreconcilable with the IOC schedule. If so, the SCSs may need to live with a later IOC, or to declare that a SCS win-win state is unachievable and to abort the project. Again, it is better to do this earlier rather than later.

Step 5: SCSs Negotiate a WinWin Decision

Actually, the previous paragraph anticipates the content of Step 5, in which the SCSs negotiate a win-win decision to commit themselves to go forward. Once the SCSs have identified and calibrated their Win Conditions in Steps 3 and 4, the process of identifying conflicts or Issues among Win Conditions; inventing and exploring Options to resolve Issues; and converging on Agreements to adopt Win Conditions or Options proceeds as described in the WinWin Negotiation Model above.

In a situation such as the Sierra supply chain project, the number of SCSs and the variety of their win conditions (cost, schedule, personnel, functionality, performance, usability, interoperability, etc.) means that multi-attribute decision theory will be involved as well as negotiation theory. Susan will also be concerned with investment theory or business case analysis to assure her stakeholders that the supply chain initiative will generate a strong return on investment. As many of the decisions will involve uncertainties (market trends, COTS product compatibilities, user interface choices), forms of statistical decision theory such as buying information to reduce risk will be involved as well.

User interface prototypes are actually ways of buying information to reduce the risk of misunderstanding SCS utility functions, as indicated in Figure 3 by the arrow between decision theory and utility theory. The other components of Step 5a in Figure 3 involve other aspects of

dependency theory, such as performance analysis, business case analysis, or critical-path schedule analysis. As also shown in Figure 3, these analyses will often proceed at increasing levels of detail in supporting steps 3a, 5a, and 7a as the project proceeds into detailed design, development, integration, and test.

Figure 5 summarizes the business case analysis for the Sierra project. Dollar values are all in millions of 2004 dollars (\$M) for simplicity. The analysis compares the expected sales and profits for the current system (columns 4, 5) and the new system (columns 7, 8) between 2004 and 2008, the cumulative increase in profits, investment cost, and resulting return on investment (columns 11-13), and expected improvements in other dimensions such as late delivery and customer satisfaction (columns 14-17). The bottom line is a strong 2.97 ROI, plus good expected outcomes in the customer satisfaction dimensions. More detail can be found in [Boehm-Huang, 2003].

The negotiations converge on a number of win-win agreements, such as involving the suppliers and distributors in reviews, prototype exercising, and beta-testing; having Sierra provide eServices with two of their staff members to work on the software development team; and agreeing on compatible data definitions for product and financial interchange. At one point in the negotiation, an unfortunate go-back is necessary when an Agreement on a product definition standard is reversed by the management of one of the distributors, who disclose that they are now committed to an emerging international standard. After some renegotiation, the other SCSs agree to this at some additional cost. But it brings up another necessary condition for successful win-win negotiations (and other collaborative ventures such as agile methods): that the stakeholder representatives be CRACK (collaborative, representative, authorized, committed, and knowledgeable) participants [Boehm-Turner, 2004]. Some other perspectives on win-win management are in [Waitley, 1985] and [Covey, 1989].

Steps 6 and 7: Planning, Executing, Monitoring, Adapting, and Controlling

As with the dependency analyses, project planning, executing, monitoring, adapting, and controlling proceed incrementally in increasing amounts of detail, generally following a risk-driven spiral process. Questions such as “how much is enough planning, specifying, prototyping, COTS evaluation, business case analysis, architecting, documenting, verifying, validating etc.?” are best resolved by balancing the risk exposures of doing too little or too much. As $\text{Risk Exposure} = \text{Probability (Loss)} * \text{Value (Loss)}$ is a value-based concept, risk balancing is integral to VBTSE [Boehm, 2005].

Value-based planning and control differs most significantly from traditional project planning and control in its emphasis on monitoring progress toward value realization rather than towards project completion. Particularly in an era of increasing rates of change in market, technology, organizational, and environmental conditions, there is an increasing probability that managing to a fixed initial set of plans and specifications will produce systems that are out of step and non-competitive with projects managing adaptively toward evolving value realization.

		<u>Current System</u>			<u>New System</u>											
					<u>Financial</u>							<u>Customers</u>				
Date	Market Size (\$M)	Market Share %	Sales	Profits	Market Share %	Sales	Profits	Cost Savings	Change in Profits	Cum. Change in Profits	Cum. Cost	ROI	Late Delivery %	Customer Satisfaction (0-5)	In-Transit Visibility (0-5)	Ease of Use (0-5)
12/31/03	360	20	72	7	20	72	7	0	0	0	0	0	12.4	1.7	1.0	1.8
12/31/04	400	20	80	8	20	80	8	0	0	0	4	-1	11.4	3.0	2.5	3.0
12/31/05	440	20	88	9	22	97	10	2.2	3.2	3.2	6	-.47	7.0	4.0	3.5	4.0
12/31/06	480	20	96	10	25	120	13	3.2	6.2	9.4	6.5	.45	4.0	4.3	4.0	4.3
12/31/07	520	20	104	11	28	146	16	4.0	9.0	18.4	7	1.63	3.0	4.5	4.3	4.5
12/31/08	560	20	112	12	30	168	19	4.4	11.4	29.8	7.5	2.97	2.5	4.6	4.6	4.6

Figure 5. Expected Benefits and Business Case

Perhaps the most provocative example is the traditional technique of “earned value management”. It assigns “value” to the completion of project tasks and helps track progress with respect to planned budgets and schedules, but has no way of telling whether completing these tasks will add to or subtract from the business value or mission value of the enterprise. Example failure modes from this approach are systems that had to be 95% redeveloped on delivery because they failed to track evolving requirements [Boehm, 1973], and startup companies that fail to track closure of market windows.

If an organization has used steps 1-5 to identify SCSs, determine their value propositions, and develop business cases, it has developed the framework to monitor expected value realization, adjust plans, and control progress toward real SCS value achievement. Figure 7 shows how this is done for the Sierra project, based on the initial budgets, schedules, and business case in Figure 5. Value-based monitoring and control for Sierra requires additional effort in terms of technology watch and market watch, but these help Sierra to discover early that their in-transit-visibility (ITV) COTS vendor was changing direction away from Sierra’s needs.

This enabled Sierra to adapt by producing a timely fallback plan, and to proactively identify and approach other likely ITV COTS vendors. The results, as shown in the ITV column and explained in the Risks/Opportunities column of Figure 6, was an initial dip in achieved ITV rating relative to plans, but a recovery to close to the originally planned value. The Risks/Opportunities column also shows a “new hardware competitor” opportunity found by market watch activities that results in a \$200K hardware cost savings that mostly compensated for the added software costs of the ITV fallback. The use of prioritized requirements to drive value-based Pareto- and risk-based inspection and testing, as discussed in [Gerrard and Thompson, 2002] and [Lee and Boehm, 2005], is another source of software cost savings.

The bottom-line results are a good example of multi-attribute quantitative/qualitative balanced-scorecard methods of value-based monitoring, adaptation, and control. They are also a good example of use of the necessary conditions for value-based control based on control theory. A traditional value-neutral “earned value” management system would fail on the criteria of business-value observability, predictability, and controllability, because its plans, measurements, and controls deal only with internal-project progress and not with external business-value observables and controllables. They also show the value of adaptive control in changing plans to address new risks and opportunities, along with the associated go-backs to revisit previous analyses and revise previous plans in Steps 7a, 7b, and 7c.

Milestone	Schedule	Cost (\$K)	Op. Cost Savings	Market Share %	Annual Sales (\$M)	Annual Profits (\$M)	Cum. Profits	ROI	Late Delivery %	Customer Satisfaction	ITV	Ease of Use	Risks/Opportunities
Life Cycle Architecture	3/31/04	400		20	72	7.0			12.4	1.7	1.0	1.8	(1)
	3/31/04	427		20	72	7.0			12.4	1.7	1.0	1.8	
Core Capability Demo (CCD)	7/31/04	1050											(2)
	7/20/04	1096								2.4*	1.0*	2.7*	
Software Init. Op. Cap. (IOC)	9/30/04	1400											
	9/30/04	1532								2.7*	1.4*	2.8*	
Hardware IOC	9/30/04	3500											(3)
	10/11/04	3432											
Deployed IOC	12/31/04	4000		20	80	8.0	0.0	-1.0	11.4	3.0	2.5	3.0	(4)
	12/20/04	4041		22	88	8.6	0.6	-0.85	10.8	2.8	1.6	3.2	
Responsive IOC	3/31/05	4500	300						9.0	3.5	3.0	3.5	
	3/30/05	4604	324						7.4	3.3	1.6	3.8	
Full Op. Cap. CCD	7/31/05	5200	1000										(5)
	7/28/05	5328	946							3.5*	2.5*	3.8*	
Full Op. Cap. Beta	9/30/05	5600	1700										
	9/30/05	5689	1851							3.8*	3.1*	4.1*	
Full Op. Cap. Deployed Release 2.1	12/31/05	6000	2200	22	106	12.2	3.2	-0.47	7.0	4.0	3.5	4.0	
	12/20/05	5977	2483	24	115	13.5	5.1	-0.15	4.8	4.1	3.3	4.2	
	6/30/06	6250											

(1) Increased COTS ITV risk, fallback identified. (2) Using COTS ITV fallback; new HW competitor; renegotiating HW. (3) \$200K savings from renegotiated HW. (4) New COTS ITV source identified, being prototyped. (5) New COTS ITV source initially integrated.

* Interim ratings based on trial use

Figure 6. Value-Based Expected/Actual Outcome Tracking

VBSE Theory Evaluation With Respect to Goodness Criteria

The Sierra example in the previous section provides an opportunity to evaluate the VBTSE theory with respect to a set of criteria for a good theory discussed further in [Boehm and Jain, 2005].

Utility: Addressing Critical Success Factors. The Results Chain method in Step 2 identified missing success-critical initiatives and stakeholders that were the downfall of supply chain initiatives at Hershey's and Toys'R'Us [Carr, 2002]. The risk-driven inspection and test approaches in Step 6 avoid wasting inspection and test time on trivial-value aspects of the system.

Generality: Covering procedural, technical, economic, and human concerns; covering small and large systems. The 7-step process and its ability to accommodate parallel activities and go-backs were sufficient to cover the Sierra project's procedural needs. Technical and economic concerns are addressed in the use of dependency theory for cost, schedule, performance, and business case analyses in Steps 3a, 5a, and 7b. Human concerns are the essence of Theory W and utility theory, and of the SCS negotiations in Step 5. The steps in the VBTSE have worked well for several mid-sized supply chain and customer relations management systems similar to Sierra; for over 100 small real-client e-services projects at USC; and as a framework for addressing very large systems of systems in such areas as defense and air traffic control.

Practicality: Supporting practical needs for prediction, diagnosis, solution synthesis, good-practice generation, and explanation. The theory draws on a wide-variety of dependency models (e.g. cost, schedule, performance, quality) to predict outcomes. In a stable, well-understood environment, managing to the predictions usually produces a self-fulfilling prophecy. In less stable and less familiar situations such as the Sierra case study, dependency theory was able to diagnose risks such as missing stakeholders in Step 2, Theory W was able to support synthesis of SCS win-win solutions in Steps 3-5, and adaptive control theory was able to generate good value-achievement monitoring practices to support in-process diagnosis and re-synthesis in Steps 6-7. The control theory necessary conditions of observability, predictability, and controllability were able to explain why traditional earned value systems would not have addressed and resolved these value-domain problems.

Preciseness: Providing situation-specific and accurate guidance. The theory is no more (and no less) accurate than its constituent theories in predicting outcomes of unprecedented situations, but it is able to provide situation-specific guidance, as shown in its application to the Sierra supply-chain project. Also, several examples were provided in the Sierra process discussion of how the theory would have generated different guidance in different situations, such as with the distributor management's reversal of a win-win agreement on a product definition standard in Step 5, and with the ITV COTS vendor's change of direction in Steps 6 and 7.

Parsimony: Avoiding excess complexity; ease of learning and application. The theory's use of risk management to determine "how much is enough" planning, specifying, testing, etc. helps avoid excess complexity and to make "everything as simple as possible, but no simpler" [Einstein, 1879-1955]. Its ease of learning and use has been tested mainly on USC's over 100 e-

services projects. These are developed by teams of 5-6 MS-level students who learn the technologies as they go, and have a 92% success rate of on-time, satisfied-customer delivery [Boehm et al., 1998].

Falsifiability: Ability to be empirically refuted. The case study identified a particular situation in which application of the theory could not produce a win-win solution, which would lead to a timely decision to cancel the project. This involved incompatible and non-negotiable SCS win conditions about Initial Operational Capability content and schedule in Steps 3 and 4. A similar outcome could have resulted from the distributor management change of direction in Step 5.

Actually, there are several other classes of situations in which our experience has shown that the win-win approach may not succeed. One is that people may disguise their true win conditions. In one situation, a stakeholder rejected a COTS product for being too expensive. When the price was lowered, the stakeholder said that some essential features were missing. When the vendor offered to supply the features at no extra cost, the true reason came out: the stakeholder had had bad dealings with the COTS vendor in the past.

Another is that some people like to win by making others losers. It is best to seek other partners when you encounter such people. Another is that you can't make omelets without breaking eggs. Many large-scale dams that benefited millions of people had to drown some other people's homes and villages. Generous payment can reduce the loss, but generally not completely eliminate it. Finally, some situations have only one winner. A good example involves political elections, in which political parties are motivated to discredit and demonize candidates and platforms of other parties.

However, many apparent only-one-winner or zero-sum-game situations can be turned into win-win situations by expanding the option space. A good example is provided in Getting to Yes [Fisher-Ury, 1981], in which a boundary-line location stalemate on ownership of the Sinai Desert between Egypt and Israel was resolved by creating a new option: the land was given back to Egypt, satisfying its territorial win condition, but it was turned into a demilitarized zone, satisfying Israel's security win condition. Other examples are provided in [Boehm-Ross, 1989].

Conclusions and Areas for Further Research

The VBTSE presented above has been shown to apply well to a reasonably complex supply chain application. In other situations, versions of the theory have been successfully applied to over 100 small e-services applications, and to the initial stages of some very large systems of systems. It satisfies the main criteria for a good theory (utility, generality, practicality, preciseness, parsimony, and falsifiability) reasonably well, particularly when compared to other theories involving explanations of human behavior.

The theory identifies several fruitful areas for further research. Some involve elaborations of aspects of utility theory, decision theory, and dependency theory to address particular VBTSE issues. Others are extensions of the theory to cover such areas as manufacturing, lean and agile methods, quality assurance, COTS-based applications, life cycle support, and combinations of these and the other areas covered. Finally, as with all theories, the initial VBSE theory needs

many more tests. The easiest tests to start with are tests of its ability to explain differences between success and failure on completed projects. Other tests that can be done right away are tests of its ability to generate good systems and software engineering practices; an early example is in [Boehm-Ross, 1989].

Further analyses can be performed on its consistency with other theories, such as the chaos-type theories underlying agile and adaptive software development [Highsmith, 2000] or the theories underlying formal software development [Jones, 1980] and generative programming approaches [Czarnecki-Eisenecker, 2000], and mathematical systems engineering theories [Wymore, 1967]. Tests of utility, generality, practicality, preciseness, and parsimony basically involve trying to apply the theory in different situations, observing its successes and shortfalls, and generating improvements in the theory that improve its capability in different situations or uncover unstated assumptions that should be made explicit to limit its domain of dependable applicability. We hope that this initial presentation of the theory will be sufficiently attractive for people to give this option a try.

Acknowledgments

The research on this Chapter has been supported by a National Science Foundation grant, “Value Based Science of Design”, and by the Affiliates of the USC Center for Software Engineering.

References

- Alexander, C., *The Timeless Way of Building*, Oxford University Press, 1979.
- Argyris, C., *Organizational Learning*, Addison-Wesley, 1978.
- Baldwin, C., Clark, K., Magretta, J., Dyer, J., Fisher, M., and Fites, D., *Managing the Value Chain*, Harvard Business School Press, 2000.
- Bellman, R., *Dynamic Programming*, Princeton U. Press, 1957.
- Blackwell, D., Girshick, M., *Theory of Games and Statistical Decisions*, Wiley, 1954.
- Boehm, B., *Software Engineering Economics*, Prentice-Hall, 1981.
- Boehm, B., “Software and Its Impact: A Quantitative Assessment”, *Datamation*, May 1973, pp 48-59.
- Boehm, B., “VBSE: Seven Key Elements and Ethical Considerations”, in S. Biffi, A. Aurum, B. Boehm, H. Erdogmus, P. Gruenbacher (eds.), *Value-Based Software Engineering*, Springer Verlag, 2005.
- Boehm, B., Bose, P., “A Collaborative Spiral Software Process Model Based on Theory W”, *Proceedings, ICSP 3, IEEE*, October 1994.
- Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., and Madachy, R., “Using the WinWin Spiral Model: A Case Study”, *IEEE Computer*, July 1998, pp 33-44.
- Boehm, B., Huang, L., “Value-Based Software Engineering: A Case Study”, *IEEE Computer*, March 2003, pp. 21-29.
- Boehm, B., Turner, R., *Balancing Agility and Discipline*, Addison Wesley, 2004.
- Boehm, B., Ross, R., “Theory-W Software Project Management: Principles and Examples”, *IEEE Trans. SW Engineering*, July 1989, pp 902-916.
- Boehm B., Jain, A., “An Initial Theory of Value-Based Software Engineering”, in S. Biffi, A. Aurum, B. Boehm, H. Erdogmus, P. Gruenbacher (eds.), *Value-Based Software Engineering*, Springer Verlag, 2005, pp 15-37.
- Booher, H. (ed.), *Handbook of Human Systems Integration*, Wiley, 2003.
- Brogan, W., *Modern Control Theory*, Prentice Hall, 1974 (3rd ed., 1991).
- Carr, D., “Sweet Victory”, *Baseline*, December 2002.
- Checkland, P., *Systems Thinking, Systems Practice*, Wiley, 1981.
- Covey, S., *The Seven Habits of Highly Successful People*, Fireside/Simon & Schuster, 1989.
- Crisp, H., et. al., *INCOSE Technical Vision*, Version 1.3, INCOSE, September 2005.
- Cyert, R.M., March, J. G., *A Behavioral Theory of the Firm*, Prentice Hall, 1963.

Czarnecki, K., Eisenecker, U., *Generative Programming: Methods, Tools, and Applications*, Addison-Wesley, 2000.

Daft, R., *Organization Theory and Design*, South-Western College Pub. (8th ed.), 2003.

Dantzig, G., *Linear Programming and Extensions*, Princeton U. Press, 1963.

Debreu, G., *Theory of Value*, Wiley, 1959.

DeMarco, T., *Controlling Software Projects*, Yourdon Press, 1982.

Dupuit, J., "On the Measurement of the Utility of Public Works", Translated by R. H. Barback, *International Economic Papers* 2:83-110, 1844 (1952).

Fishburn, P. C., *The Foundations of Expected Utility*, Dordrecht, 1982.

Fisher, R., Ury, W., *Getting To Yes: Negotiating Agreement Without Giving In*, Houghton Mifflin, 1981.

Ford, L., Fulkerson, D. R., *Flows in Networks*, Princeton U. Press, 1962.

Forrester, J., *Industrial Dynamics*, MIT Press, 1961.

Friedman, G., *Constraint Theory: Multidimensional Mathematical Model Management*, Springer, 2005.

Friedman, G., "Constraint Theory", *Int'l Journal Systems Science*, 1976.

Gerrard, P., Thompson, N., *Risk-Based E-Business Testing*, Artech House, 2002.

Goldratt, E., *The Goal*, North River Press, 1984.

Highsmith, J., *Adaptive Software Development*, Dorset House, 2000.

Holland, J., *Emergence: From Chaos to Order*, Perseus Books, 1998.

Intriligator, M., *Mathematical Optimization and Economic Theory*, SIAM Press, 2002.

Jones, C. B., *Software Development: A Rigorous Approach*, Prentice Hall, 1980.

Kaplan, R., Norton, D., *The Balanced Scorecard: Translating Strategy into Action*, Harvard Business School Press, 1996.

Koskela, L., Howell, G., "The Underlying Theory of Project Management is Obsolete", Proceedings, AMI Research Conference, 2002, pp 293-302.

Kauffman, S., *At Home in the Universe*, Oxford U. Press, 1995.

Keeney R. L., Raiffa, H., *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, 1976.

Lee, M. J., "Foundations of the WinWin Requirements Negotiation System", Ph.D. dissertation, University of Southern California, 1996.

Luce, R. D., Raiffa, H., *Games and Decisions*, John Wiley, 1957.

Luehrman, T. A., "Investment Opportunities as Real Options: Getting Started on the Numbers" *Harvard Business Review*, July/August (1998), pp 51-67.

Marschak, J., Radner, R., *Economic Theory of Teams*, Yale University Press, 1972.

March, J., Simon, H., *Organizations*, Wiley, 1958.

Maslow, A., *Motivation and Personality*, Harper, 1954.

Newnan, D., Lavelle, J., and Eschenbach, *Engineering Economic Analysis*, Oxford U. Press, 9th ed., 2004.

Parsons, T., *Social Systems and the Evolution of Action Theory*, The Free Press, 1977.

Raiffa, H., *The Art and Science of Negotiation*, Belknap/Harvard U. Press, 1982.

Rawls, J., *Theory of Justice*, Belknap/Harvard U. Press, 1971, 1999.

Rechtin, E., *Systems Architecting: Creating and Building Complex Systems*, Prentice-Hall, 1991.

Rifkin, S., "The Parsons Game: The First Simulation of Talcott Parsons' Theory of Action", Ph.D. dissertation, George Washington University, 2004.

Sage, A., Rouse, W., *Handbook of Systems Engineering and Management*, Wiley, 1999.

Scott Morton, M., *The Corporation of the 1990s: Information Technology and Organization Transformation*, Oxford University Press, 1991.

Shaw, M., Garlan, D., *Software Architecture. Perspectives on an emerging discipline*, Prentice Hall, 1996.

Simon, H., *The Science of the Artificial*, MIT Press, 1969.

Simon, H., *Models of Man*, Wiley, 1957.

Thorp, J., DMR's Center for Strategic Leadership, *The Information Paradox: Realizing the Benefits of Information Technology*, McGraw-Hill, 1998.

von Neumann, J., Morgenstern, O., *Theory of Games and Economic Behavior*, Princeton University Press, 1944.

Waitley, D., *The Double Win*, Berkley, 1985.

Wiest, J. D., Levy, F. K., *A Management Guide to PERT/CPM*, Prentice Hall, 1977.

Womack, J., Jones, D., *Lean Thinking*, Simon and Schuster, 1996.