



Explaining success & failure: Value-based software engineering

Stan Rifkin

MASTER SYSTEMS INC.

2604B El Camino Real 244

Carlsbad, California 92008 USA

+1 760 729 3388

sr @ Master-Systems.com

Ver. 1.0 – March 20, 2006 – © Copyright Center for Software Engineering at
the University of Southern California unless otherwise shown.

Master Systems Inc. is an affiliate member of the Center

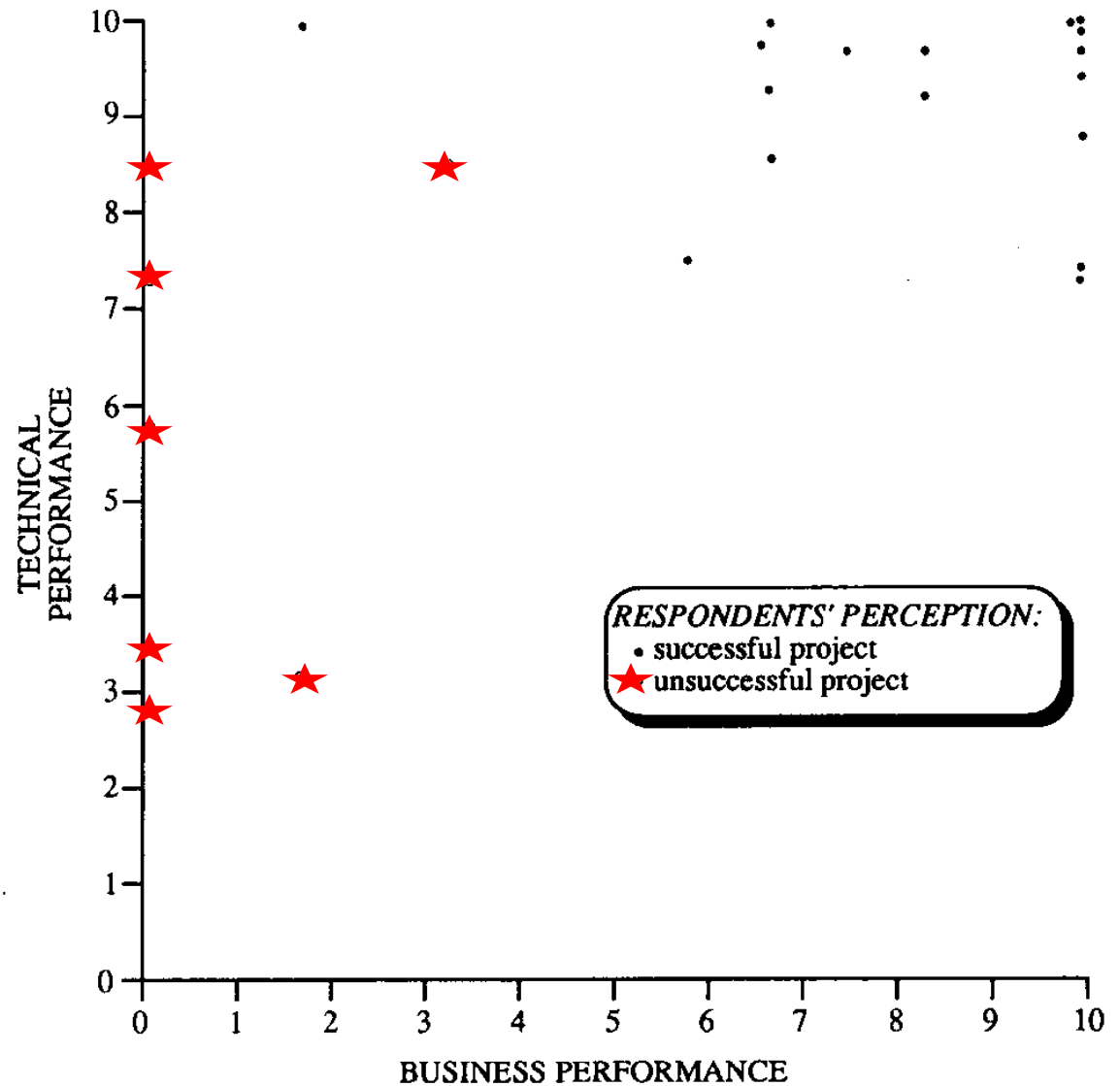


My remarks will be brief

- There is a book.
- Many of the details refer to old sources; the added value is the synthesis, the framework.
- Codification of what some of us already do.
- A promising avenue of research, already with some application.
- A way to think.

- The future!

Successful vs. unsuccessful



Source: An Exploratory Analysis Relating the Software Project Management Process to Project Success, Michael S. Deutsch, *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT*, VOL. 38, NO. 4, 365-375, NOVEMBER 1991



Success & failure

- Completely subjective, even when numeric characteristics are given.
- Depends upon aspirations, goals.
- But these “evolve” during execution.
- Still, everyone has his/her ideas about why software projects succeed or fail.

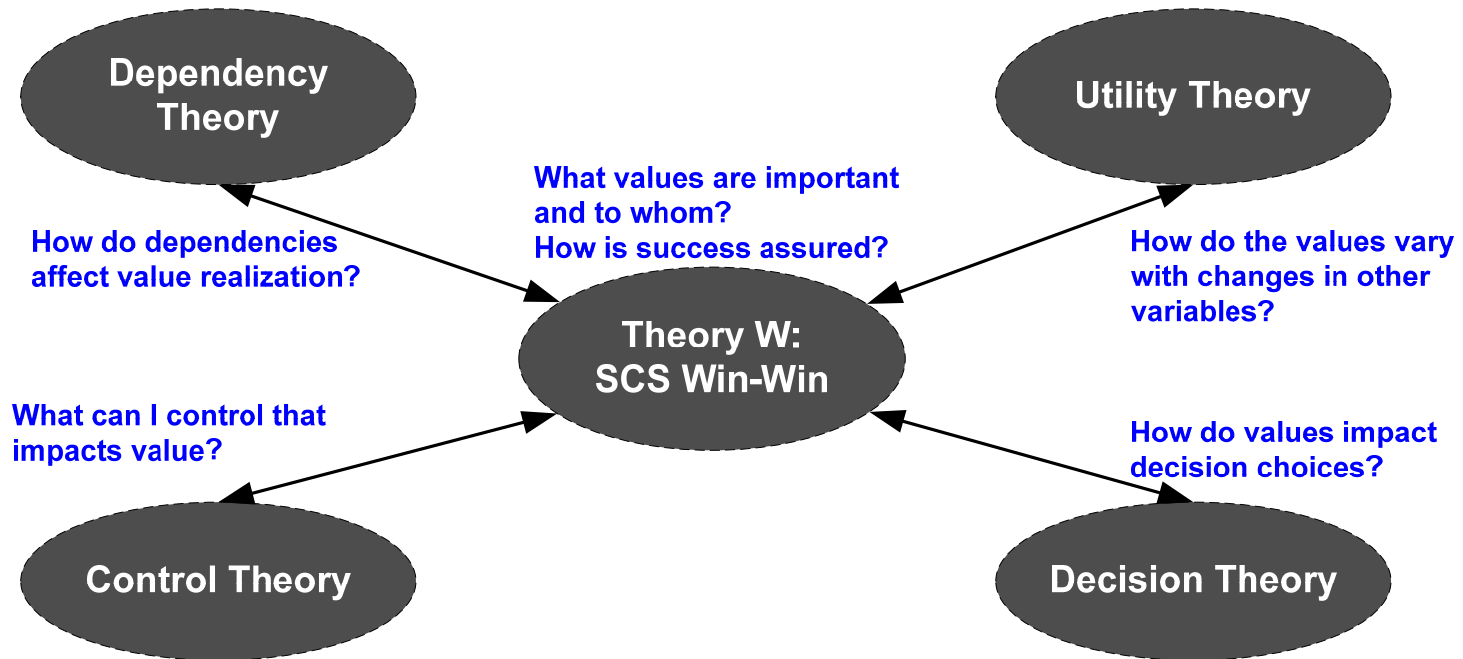
- “Wicked problem”



Enter: Value-based software engineering

- The problems it is trying to solve:
 - Canceled projects – after large investment
 - Inefficient projects
- Limitations:
 - Method independent
 - Cannot solve all problems
 - More notional than detailed today

4 + 1 Framework



7 Step process of VBSE





Theory W -- Barry Boehm

Steps 1-3

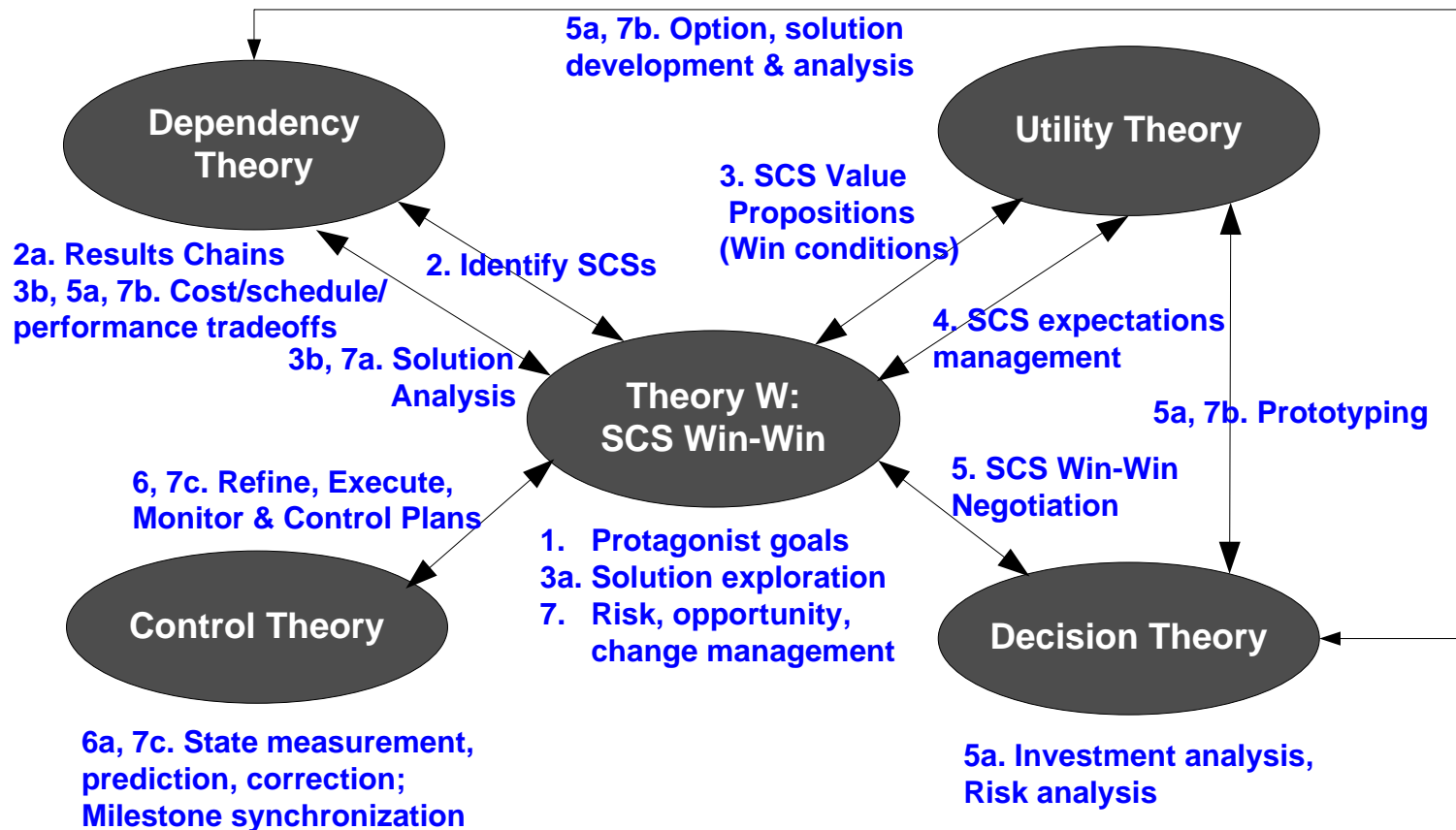
- Seek Win-Win
 - Identify success-critical stakeholders (SCS)
 - Find out SCS win conditions
 - Problem-solve – does not always converge (“Getting to Yes”)
- There are theorems



Theory W

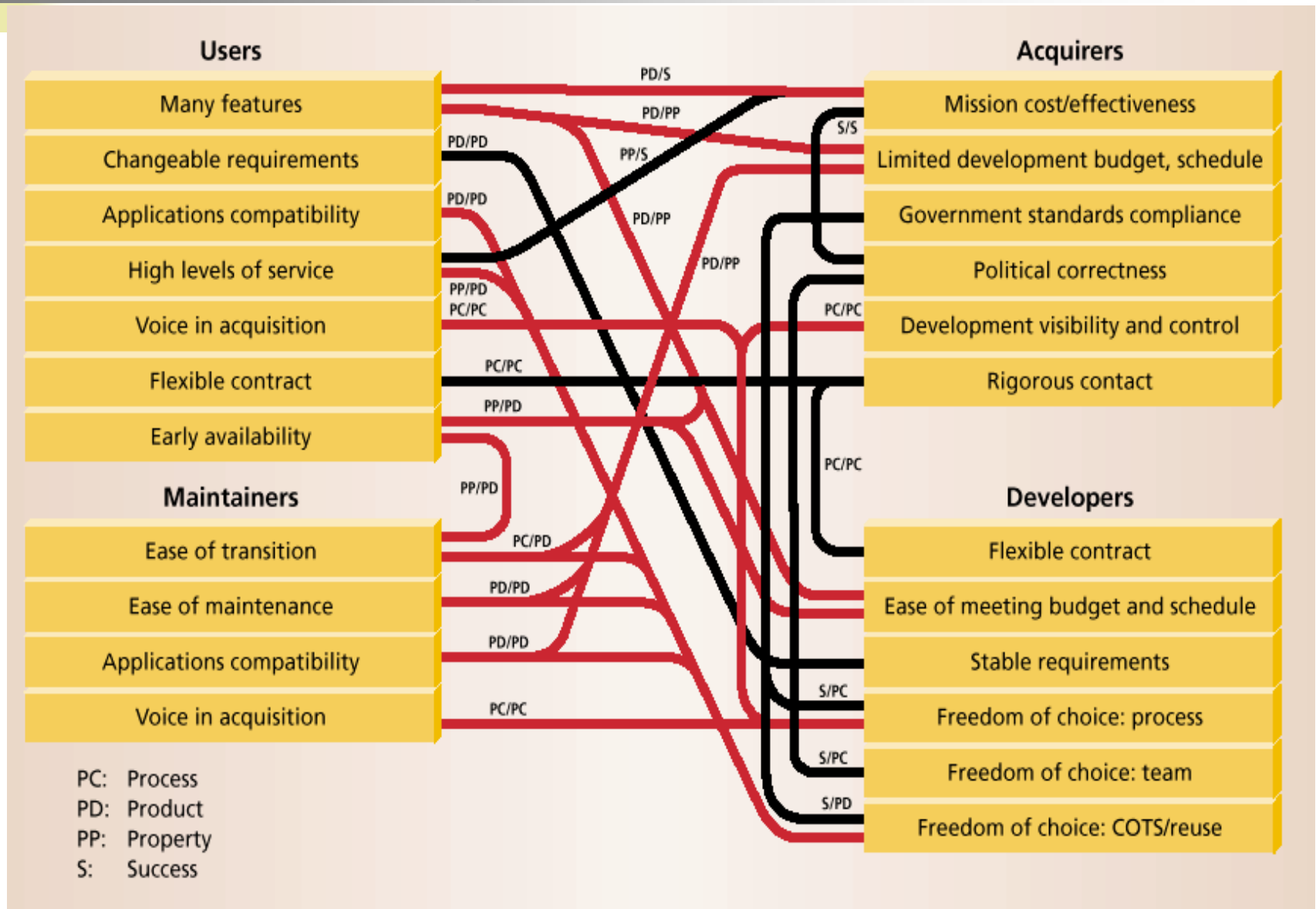
1. Establish a set of win-win preconditions.
 - Understand how people want to win
 - Establish reasonable expectations
 - Match people's tasks to their win conditions
 - Provide a supportive environment
2. Structure a win-win software process.
 - Establish a realistic process
 - Use the plan to control the project
 - Identify and manage your win-lose or lose-lose risks
 - Keep people involved
3. Structure a win-win software product.
 - Match product to users' & maintainers' win conditions.

7 Step process of VBSE

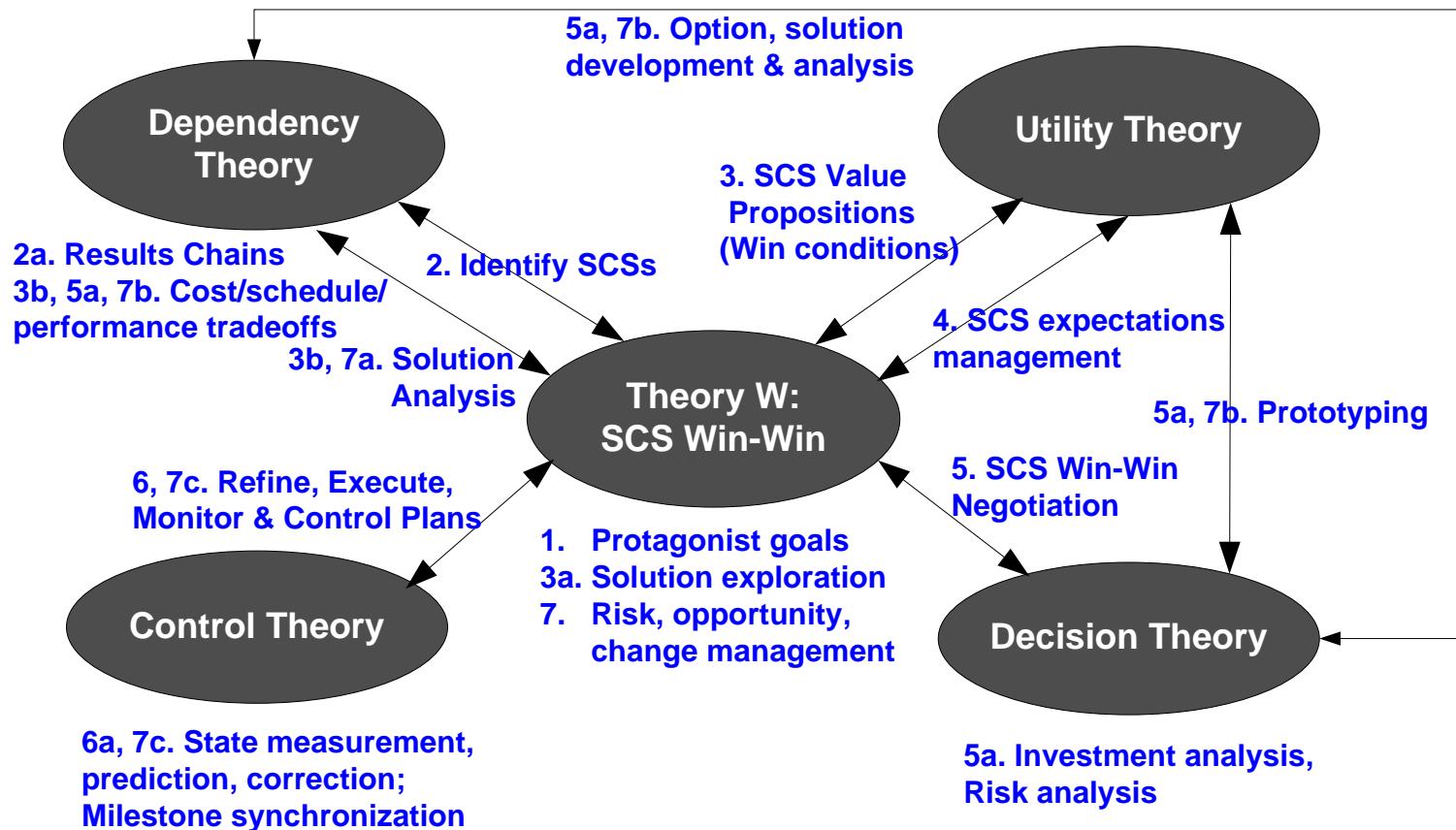


Mismatches = dependency problems

Step 2

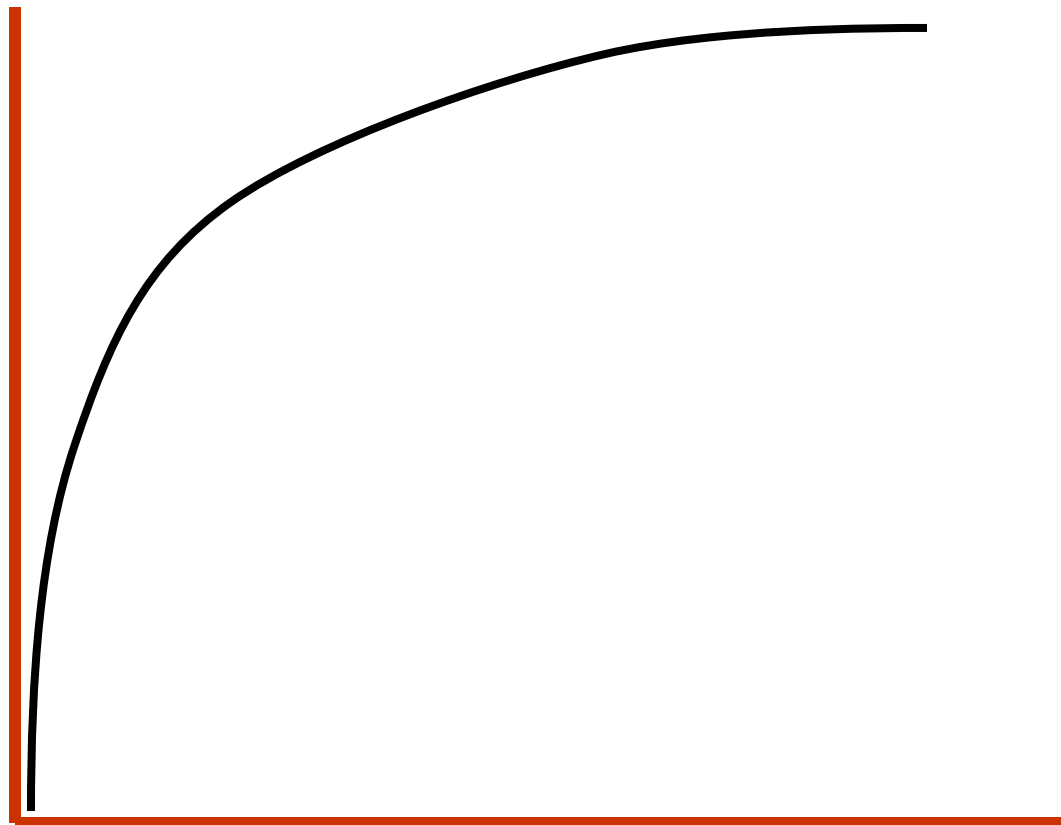


7 Step process of VBSE



Utility theory (for money)

UTILITY



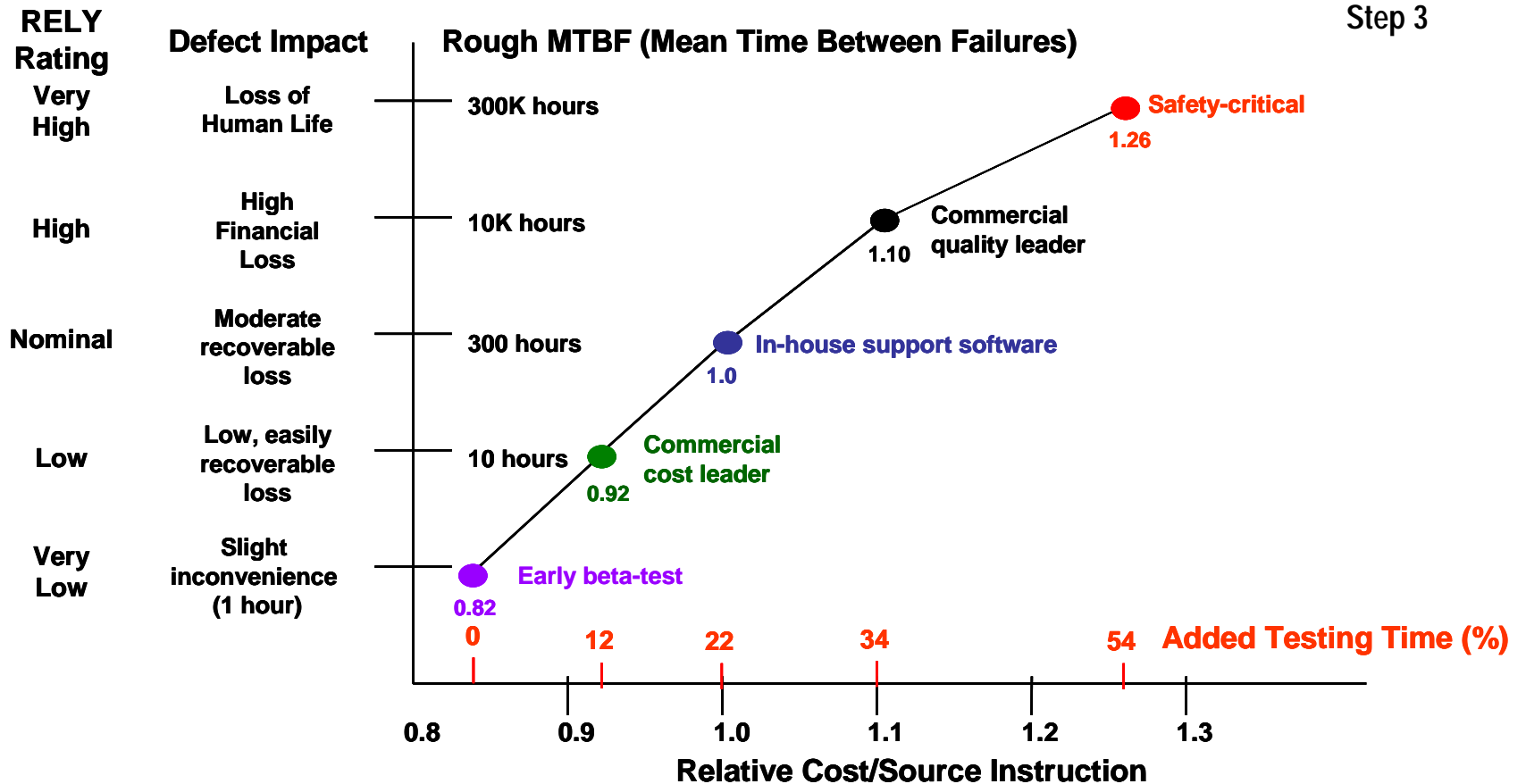
Step 3

CUMULATIVE WEALTH

Diminishing marginal returns

Cost/Reliability/Test Time Tradeoff (from COCOMO)

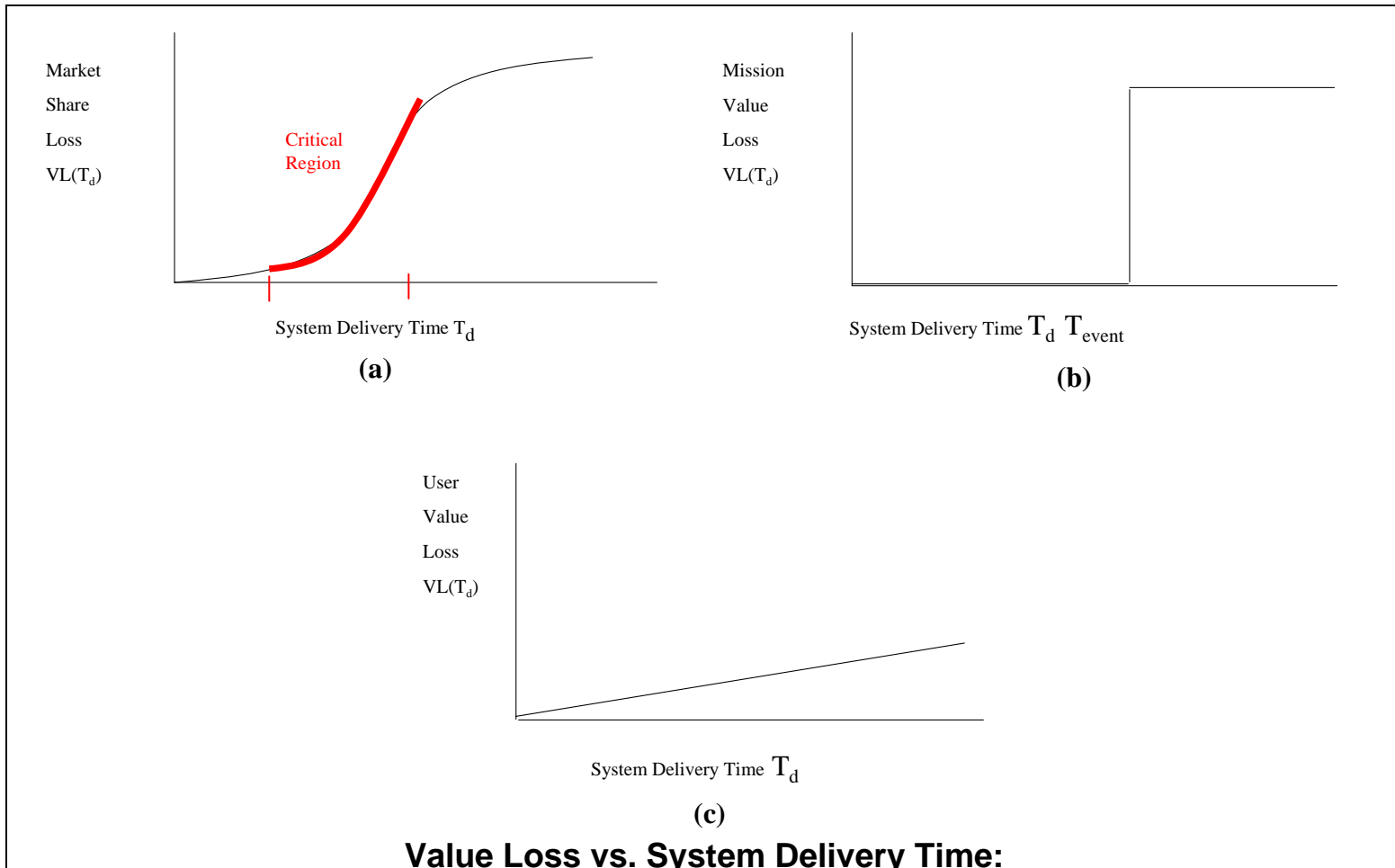
Step 3



Source: How Much Software Quality Investment is Enough: A Value-Based Approach, LiGuo Huang and Barry Boehm, IEEE Software, 2006, to appear.

Other utility curves

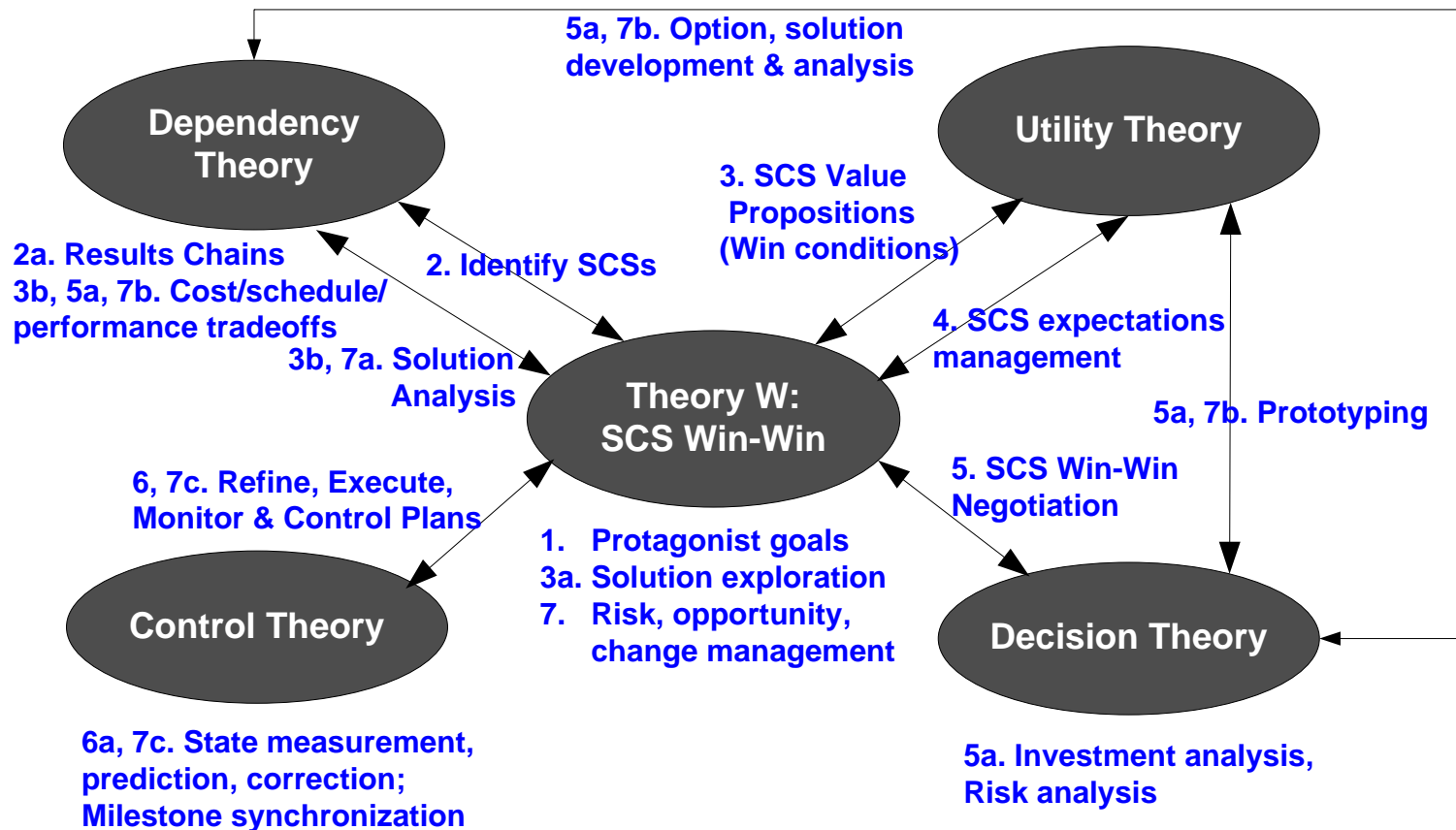
Step 3



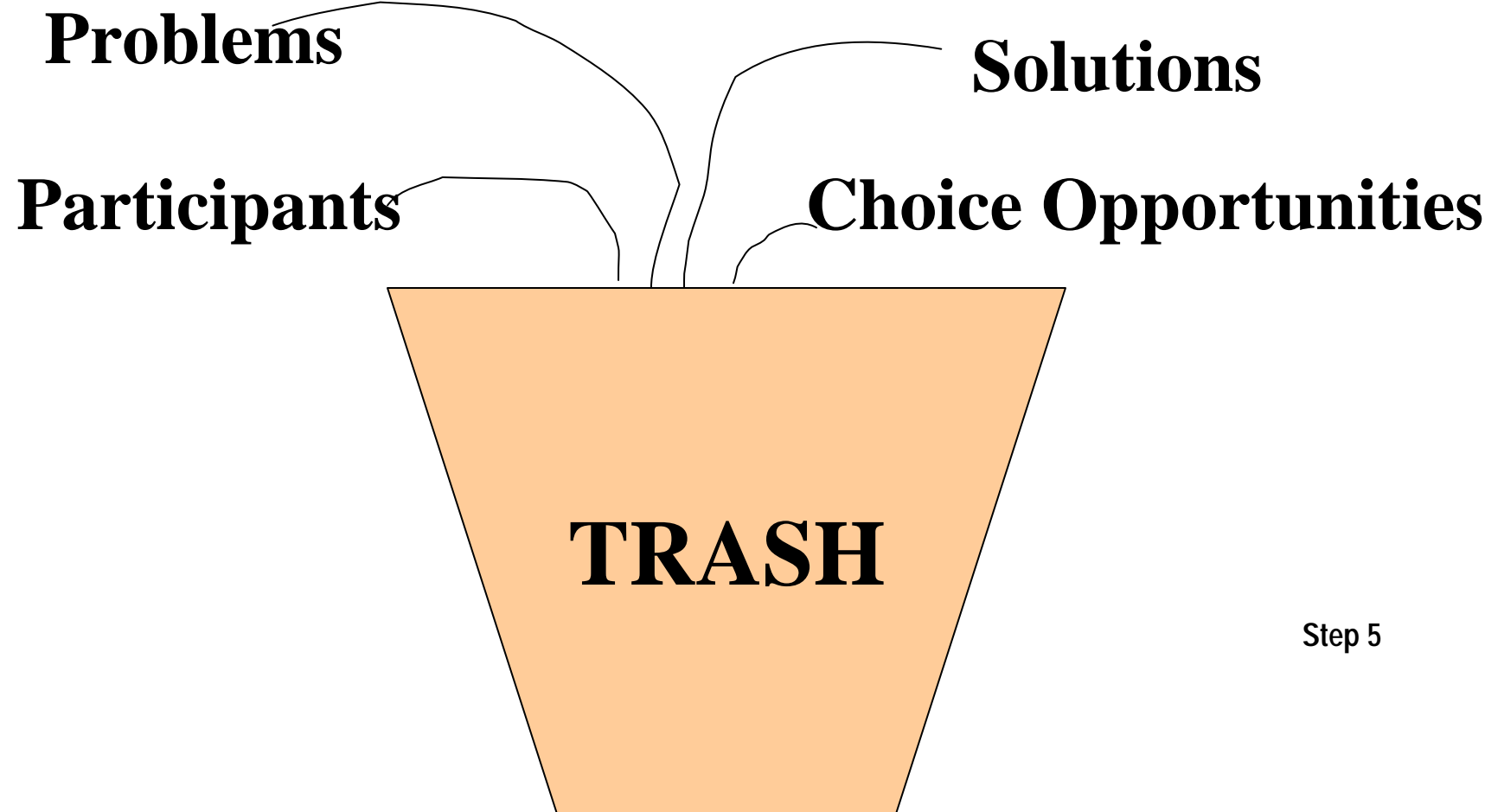
**(a) Marketplace Competition (Internet Services, Wireless Infrastructure);
(b) Fixed-schedule Event Support; (c) Off-line Data Processing**

Source: Huang & Boehm, op. cit.

7 Step process of VBSE



Garbage can model of organizational choice

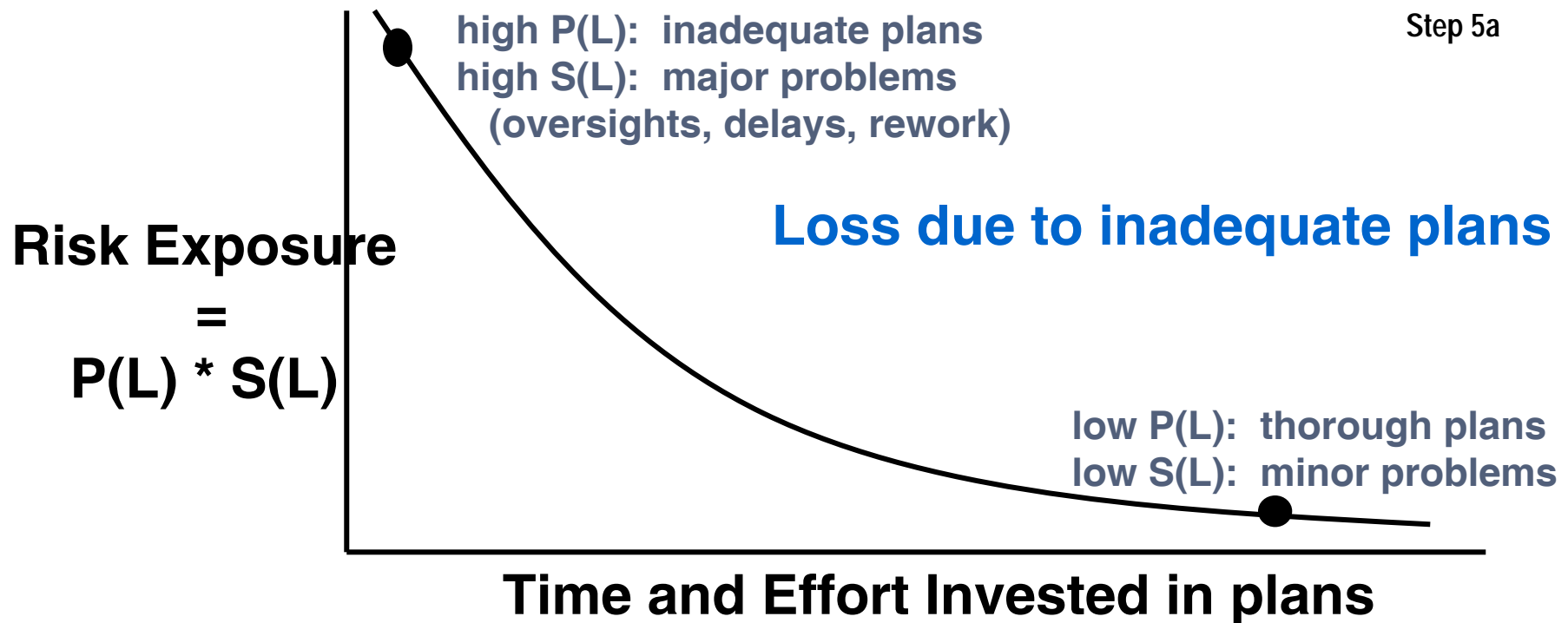


Step 5

Adapted from: A garbage can model of organizational choice, Michael Cohen, James March & Johan Olsen, *Administrative Science Quarterly*, March 1972, vol. 17, no. 1, 1-25

Example RE Profile: Planning Detail

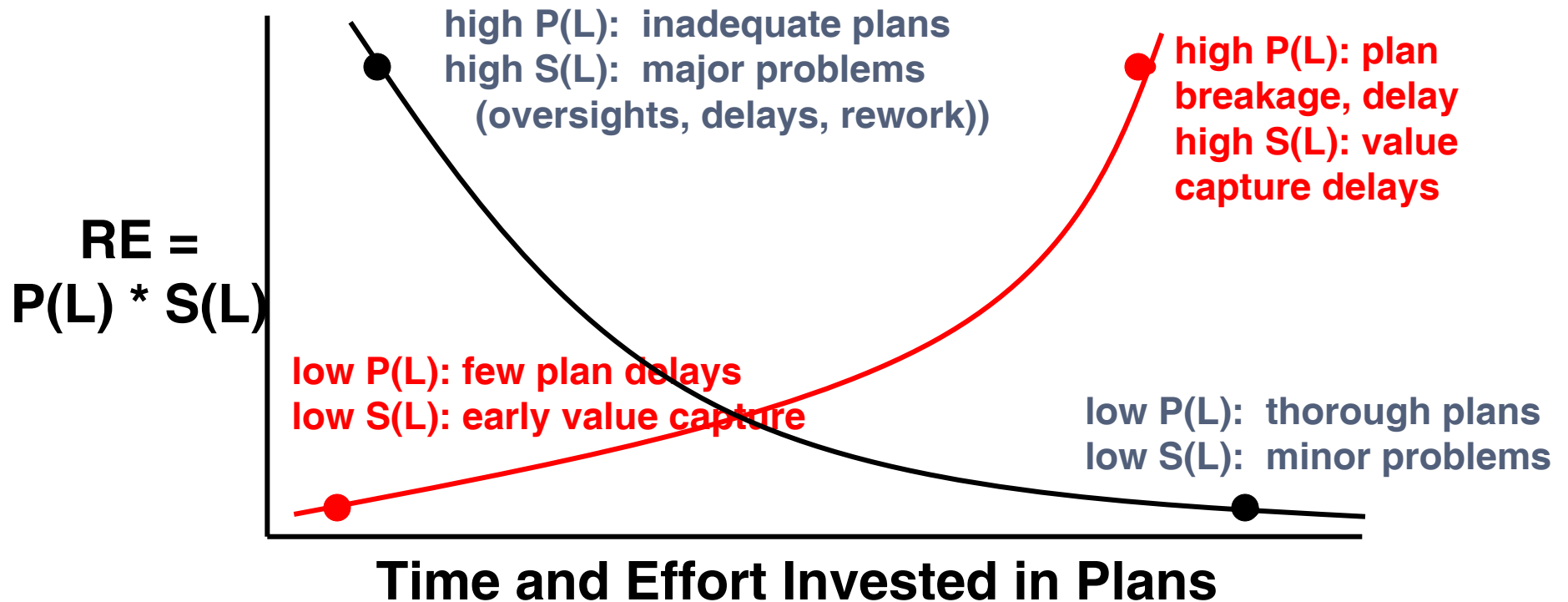
**Risk exposure = Sum over all events of
[Probability of event x size (impact) of event]**



Source for this slide and the following four: Many of Barry Boehm's presentations and last year's SPIN presentation by Stan Rifkin, "What is the best way to develop software? Continuing the conversation about agility and plan-driven methods," June 2005.

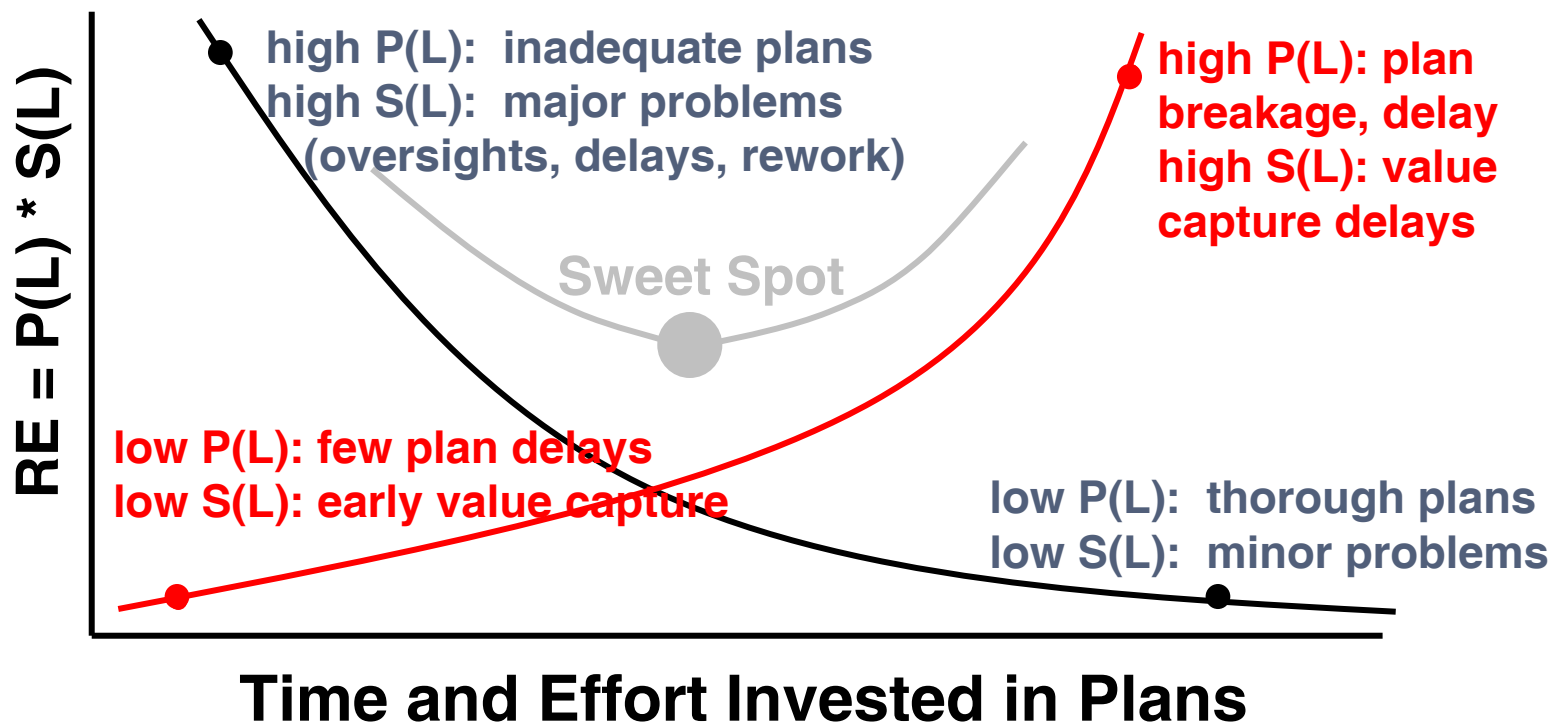
Example (cont.)

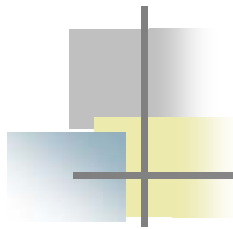
- Loss due to inadequate plans
- Loss due to market share erosion



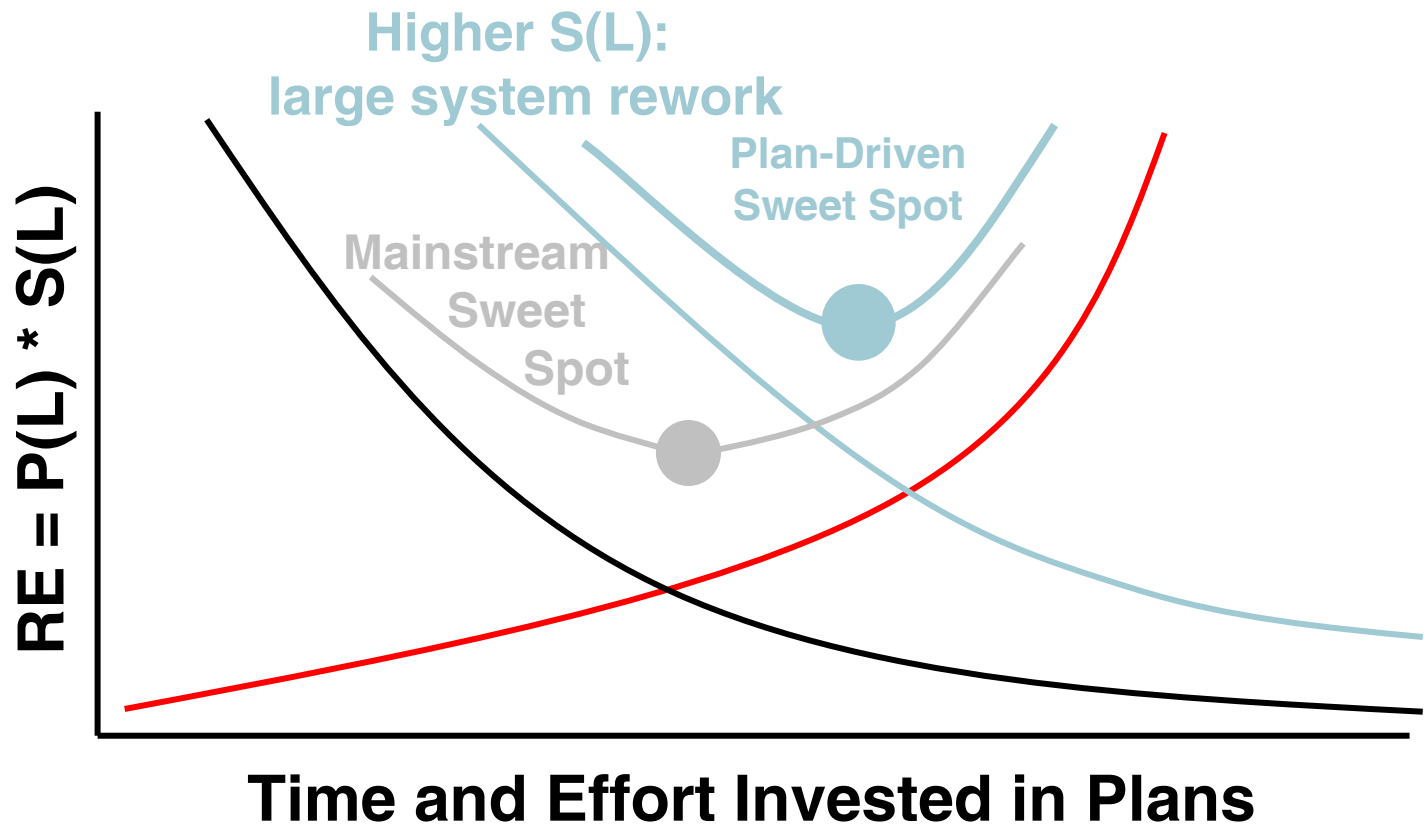
Example RE Profile: When to Ship

- Sum of Risk Exposures

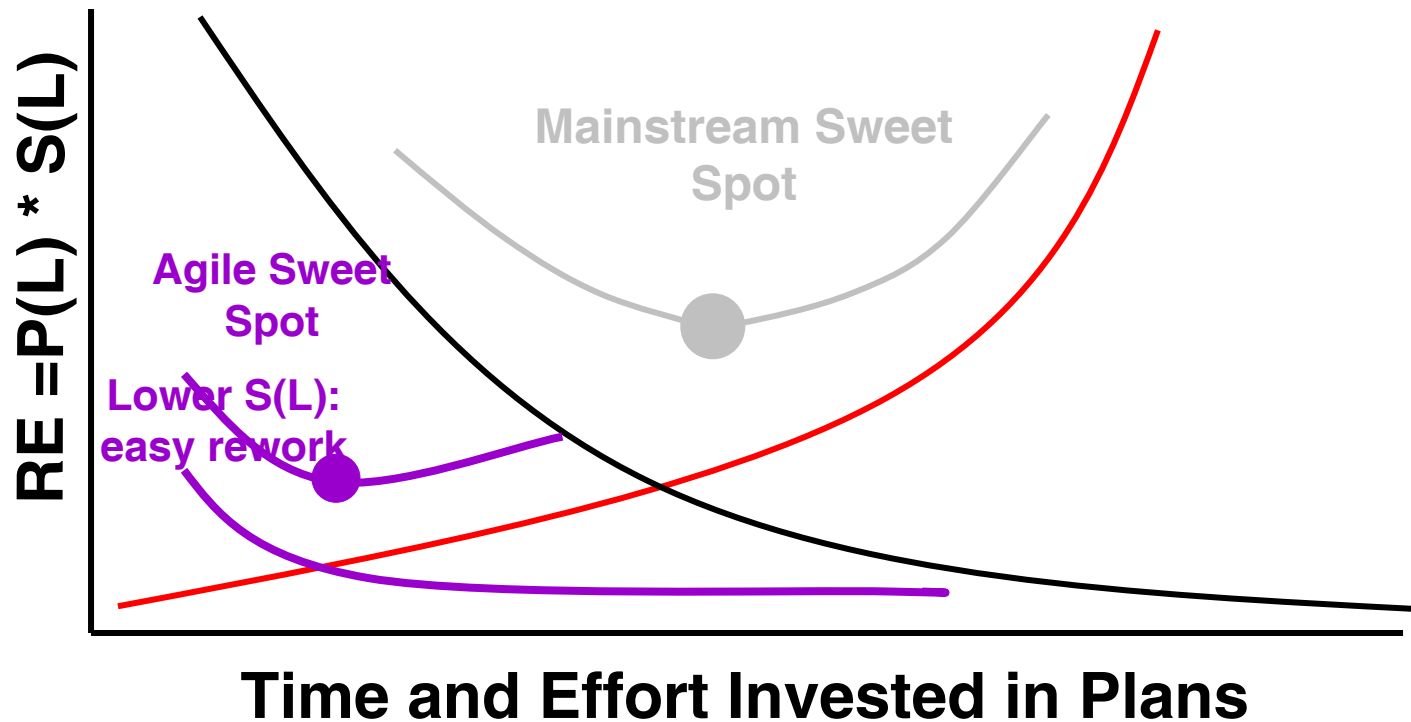




Plan-Driven Home Ground

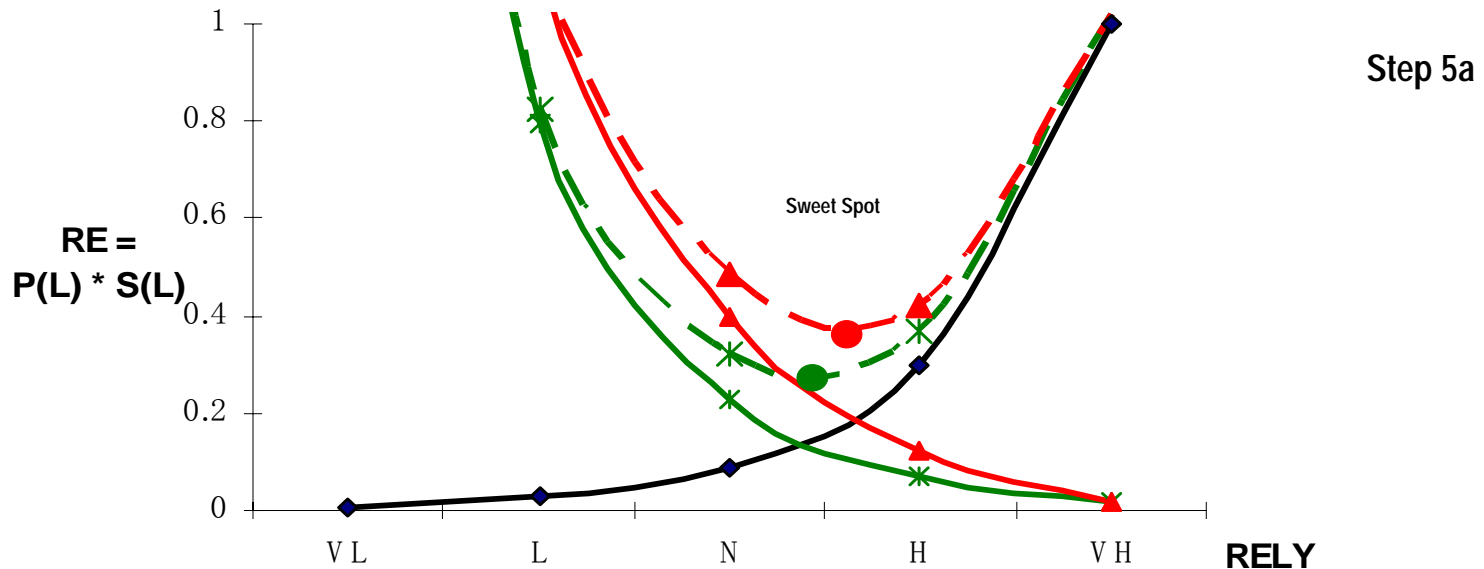


Agile Home Ground



Comparing Value-Based Testing vs. Value-Neutral Testing

◆ Market Share Erosion
 ✱ Value-based Testing
 ▲ Value-neutral Testing



	VL	L	N	H	VH	RELY
COCOMO II:	0	12	22	34	54	Added % test time
COQUALMO:	1.0	.475	.24	.125	.06	$P_a(L)$
Value-Based:	3.0	1.68	.96	.54	.30	$S_a(L)$: Pareto
Value-Neutral:	3.0	2.33	1.65	0.975	.30	$S_a(L)$: Linear
Market Risk:	.008	.027	.09	.30	1.0	RE_m

Source: Huang & Boehm, op. cit.

7 Step process of VBSE



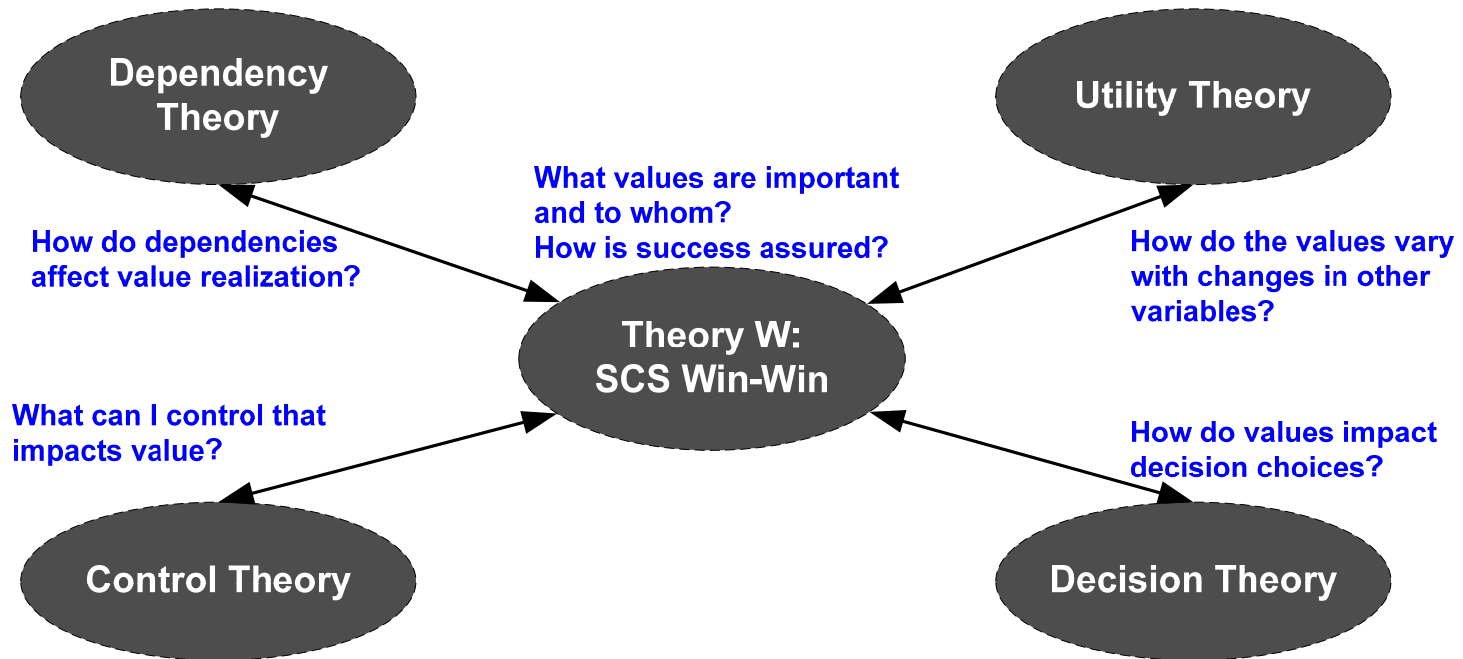


Plan & manage

Steps 6-7

- Finally something we know!
- There are two differences between traditional/agile project management + control and this step:
 - Frequent re-visits to the win-win conditions with the success-critical stakeholders; in other words: NOT passive broadcasting of status.
 - Connection between what we do to exercise control and actual outcome; in other words, we have to understand cause & effect.

4 + 1 Framework





References

■ Papers

- Barry Boehm and Rony Ross, "Theory-W Software Project Management: Principles and Examples," *IEEE Transactions on Software Engineering*, Vol. 15, no. 7, July 1989.
- LiGuo Huang, Barry Boehm, "How Much Software Assurance is Enough: A Value-Based Approach," *IEEE Software*, 2006, to appear.

■ Conferences

- Apurva Jain and B. Boehm, "SimVBSE: Developing a Game for Value-Based Software Engineering," *Proceedings*, CSEET, Hawaii, April 2006 (to appear).
- B. Boehm and A. Jain, "A Value-Based Theory of Systems Engineering" (submitted to INCOSE, to appear)

■ Workshops

- A. Jain and B. Boehm, "Developing a Theory for Value-Based Software Engineering," *Proceedings* of Economics-driven software engineering research (EDSER-7), St. Louis, May 2005.

■ Book Chapters

- B. Boehm and A. Jain, "An Initial Theory of Value-Based Software Engineering," in S. Biffi, A. Aurum, B. Boehm, H. Erdogmus, and P. Gruenbacher (eds), *Value-Based Software Engineering*, Springer, 2005, pp. 15-37.