University of Southern California
Center for Software Engineering

Automating Value-Based Software Engineering (VBSE)

Barry Boehm, USC

ASE 2004 Keynote

September 23, 2004

Outline

- Motivation and definitions
- Seven key practices
  - Examples of automation
- VBSE automation challenges
- Conclusions; references

Software Testing Business Case

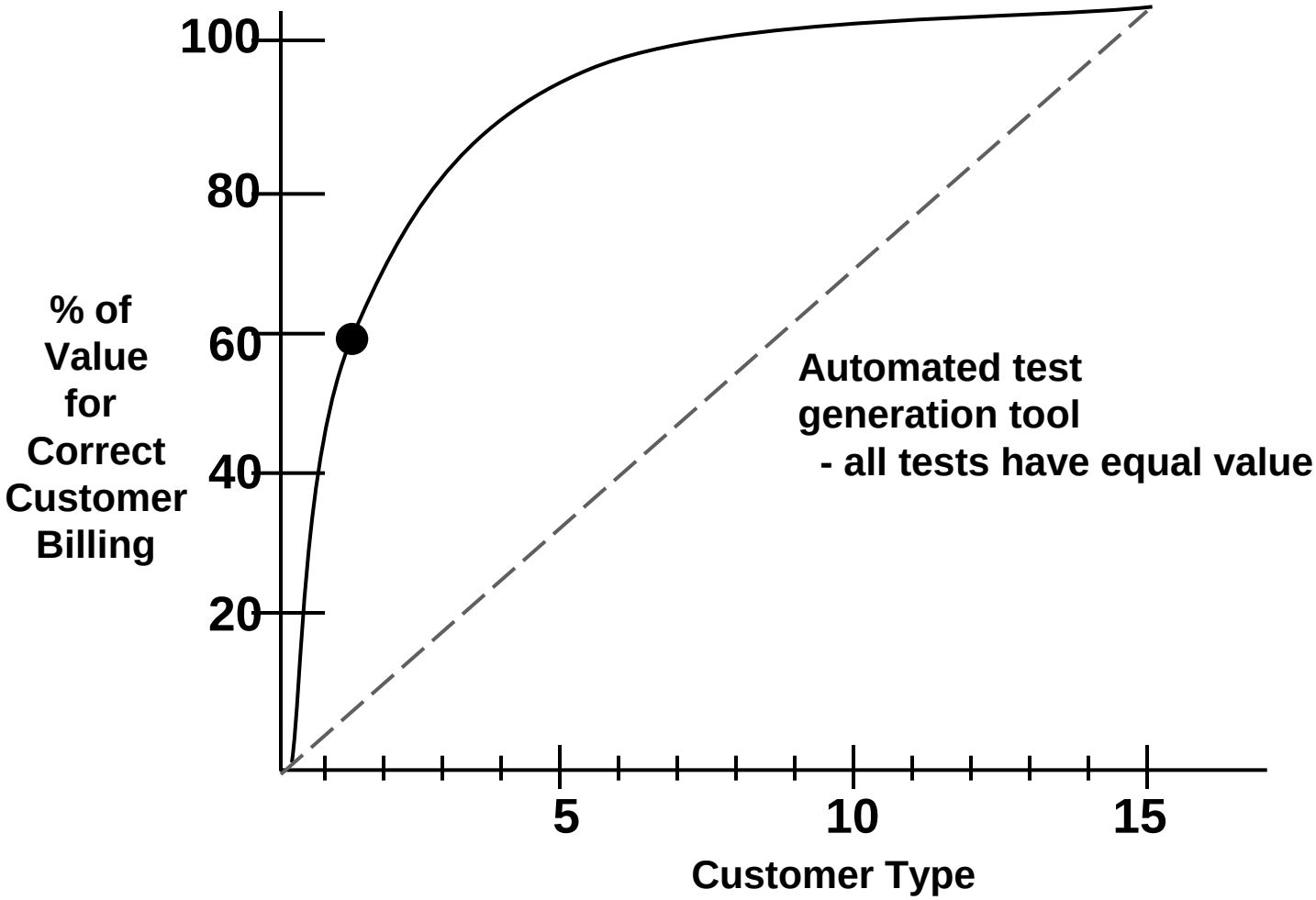
- **Vendor proposition**
  - **Our test data generator will cut your test costs in half**
  - **We'll provide it to you for 30% of your test costs**
  - **After you run all your tests for 50% of your original cost, you are 20% ahead**
- **Any concerns with vendor proposition?**

Software Testing Business Case

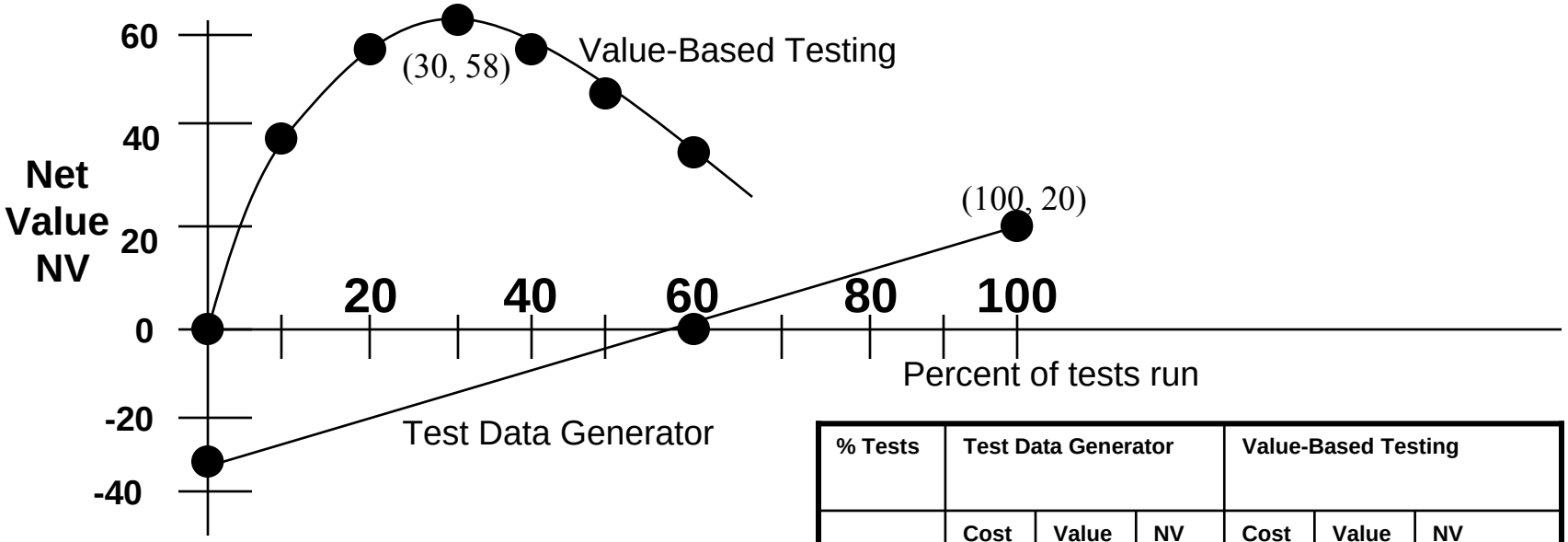
- **Vendor proposition**
  - **Our test data generator will cut your test costs in half**
  - **We'll provide it to you for 30% of your test costs**
  - **After you run all your tests for 50% of your original cost, you are 20% ahead**
- **Any concerns with vendor proposition?**
  - **Test data generator is value-neutral***
  - **Every test case, defect is equally important**
  - **Usually, 20% of test cases cover 80% of business case**

*** As are most current software engineering techniques**

20% of Features Provide 80% of Value: Focus Testing on These (Bullock, 2000)



Value-Based Testing Provides More Net Value



% Tests	Test Data Generator			Value-Based Testing		
	Cost	Value	NV	Cost	Value	NV
0	30	0	-30	0	0	0
10	35	10	-25	10	50	40
20	40	20	-20	20	75	55
30	45	30	-15	30	88	58
40	50	40	-10	40	94	54
….	….	….	….	….	….	….
100	80	100	+20	100	100	0

Motivation for Value-Based SE

- **Current SE methods are basically value-neutral**
  - **Every requirement, use case, object, and defect is equally important**
  - **Object oriented development is a logic exercise**
  - **"Earned Value" Systems don't track business value**
  - **Separation of concerns: SE's job is to turn requirements into verified code**
  - **Ethical concerns separated from daily practices**
- **Value – neutral SE methods are increasingly risky**
  - **Software decisions increasingly drive system value**
  - **Corporate adaptability to change achieved via software decisions**
  - **System value-domain problems are the chief sources of software project failures**

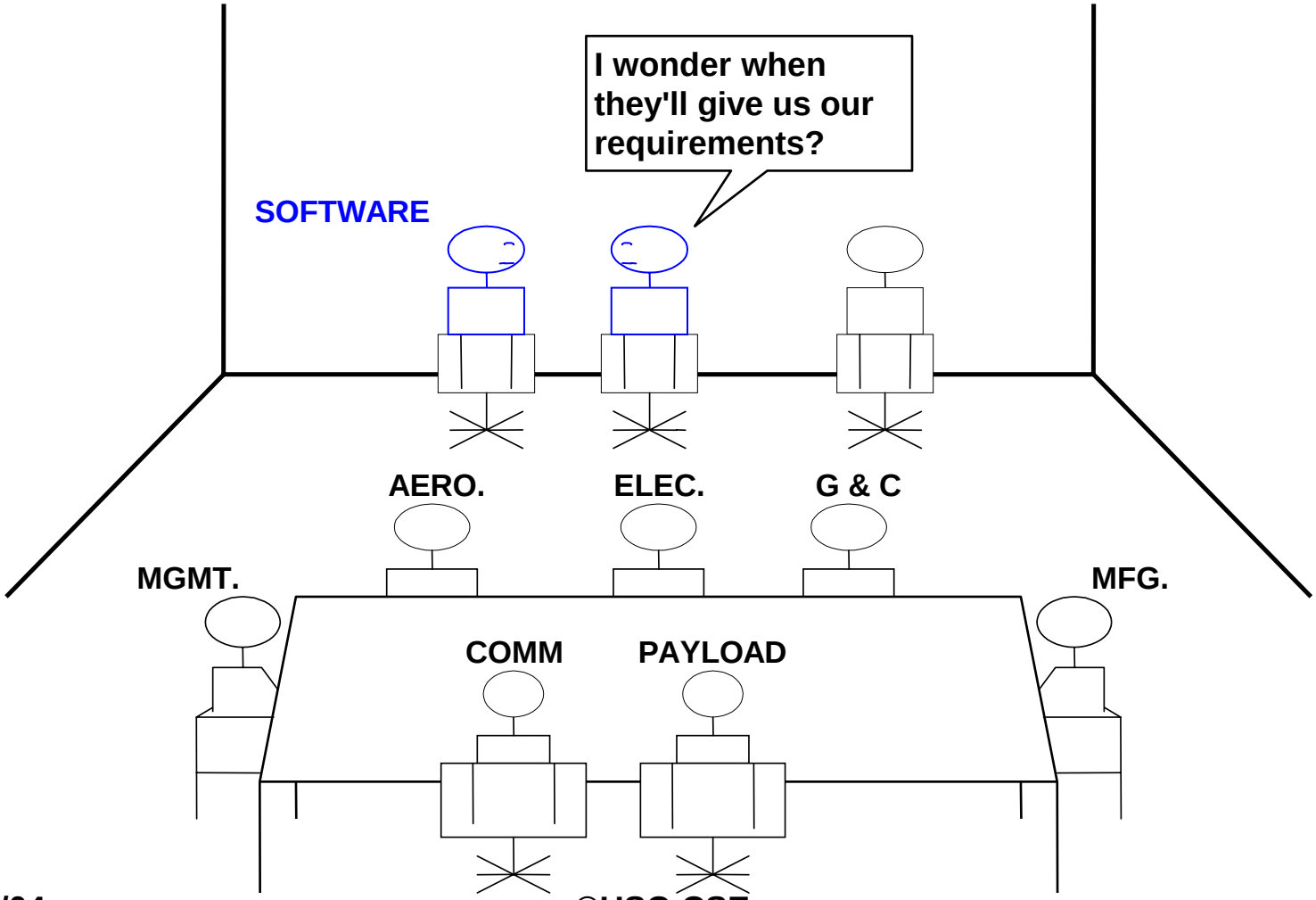
The "Separation of Concerns" Legacy

- **"The notion of 'user' cannot be precisely defined, and therefore has no place in CS or SE."**
  - **Edsger Dijkstra, ICSE 4, 1979**

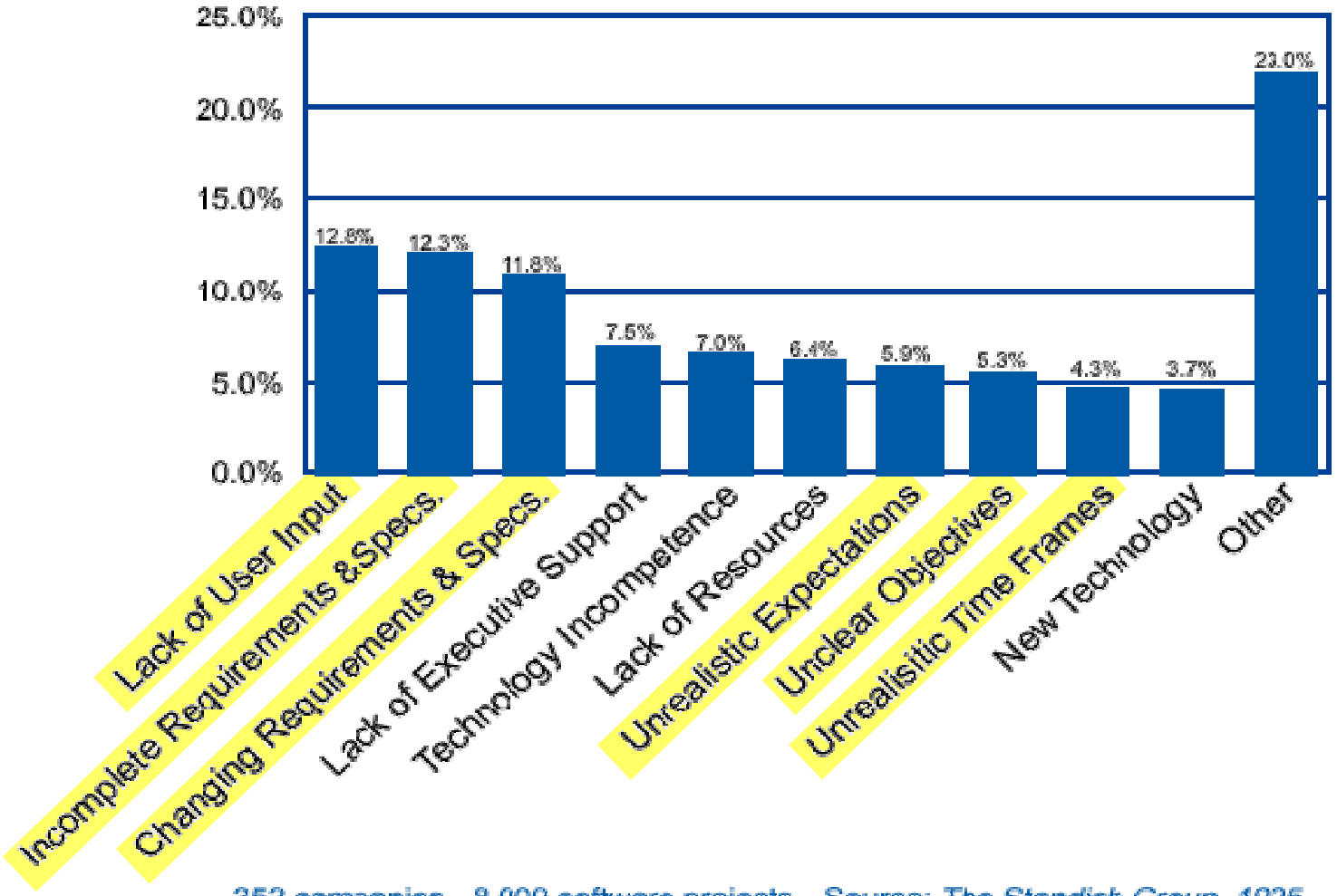
- **"Analysis and allocation of the system requirements is not the responsibility of the SE group but is a prerequisite for their work"**
  - **Mark Paulk at al., SEI Software CMM* v.1.1, 1993**

***Capability Maturity Model**

Resulting Project Social Structure



Why Software Projects Fail



352 companies - 8,000 software projects. Source: *The Standish Group, 1995*

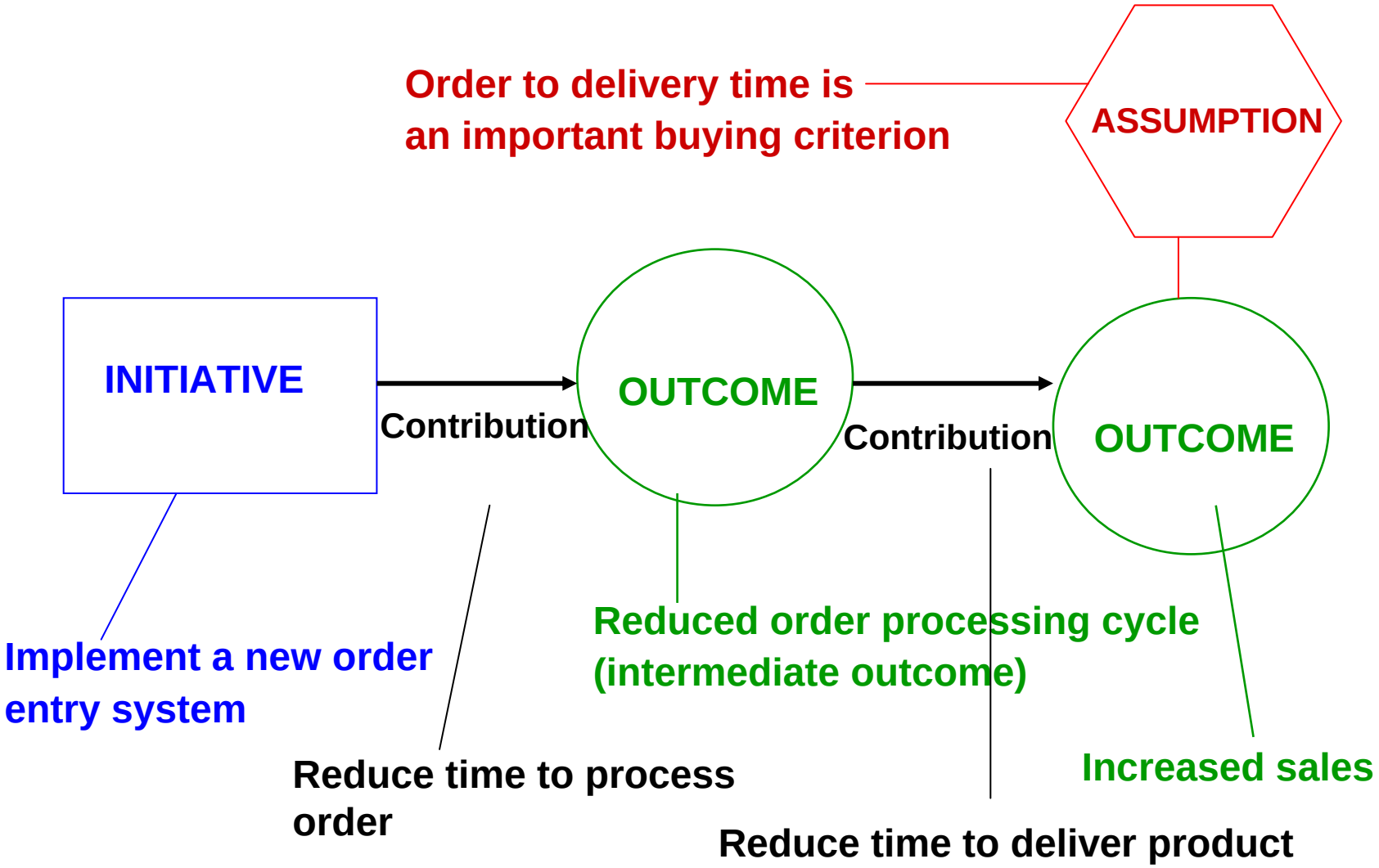
Outline

- Motivation and definitions
- Seven key practices
  - Examples of automation
- VBSE automation challenges
- Conclusions; references

7 Key Elements of VBSE

1. Benefits Realization Analysis
2. Stakeholders' Value Proposition Elicitation and Reconciliation
3. Business Case Analysis
4. Continuous Risk and Opportunity Management
5. Concurrent System and Software Engineering
6. Value-Based Monitoring and Control
7. Change as Opportunity

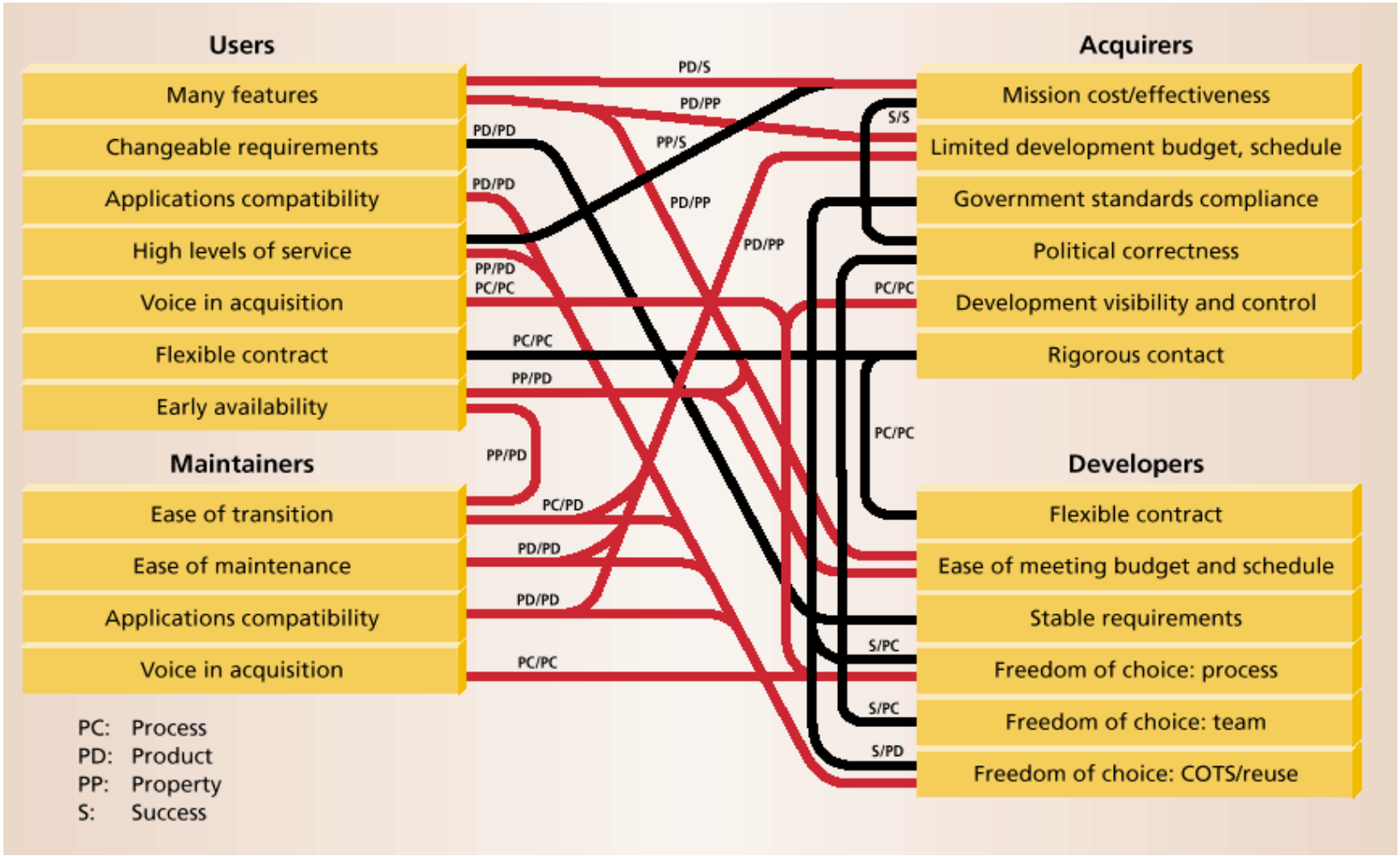
DMR/BRA* Results Chain



*DMR Consulting Group's Benefits Realization Approach

The Model-Clash Spider Web: Master Net

- Stakeholder value propositions (win conditions)



EasyWinWin OnLine Negotiation Steps

Review and Expand Negotiation Topics (Group Outliner)
Jointly review and define the scope of the negotiation. Identify the negotiation topics for your EasyWinWin activity.

Brainstorm Stakeholder Interests (Electronic Brainstorming)
Collect ideas about Win Conditions for your EasyWinWin activity

Converge on Win Conditions (Categorizer)
Jointly craft and organize a succinct list of win conditions.

Capture Glossary of Terms (Topic Commenter)
Define important terms of the domain.

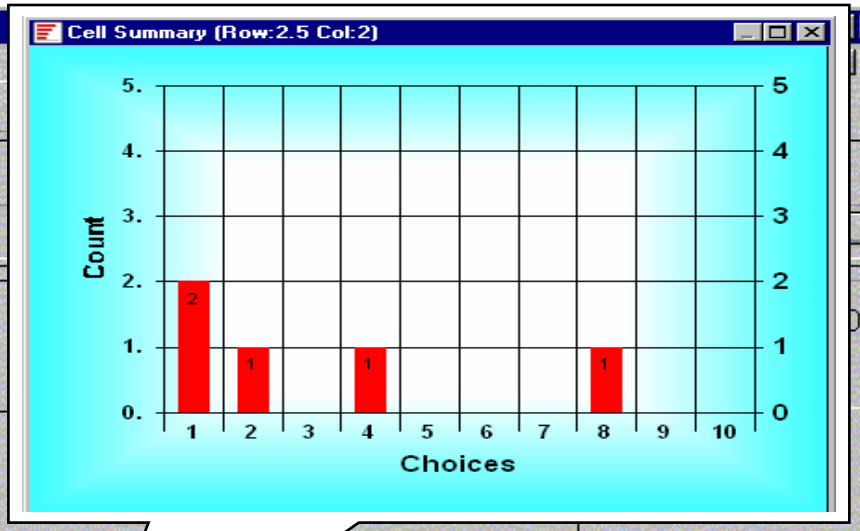
Prioritize Win Conditions (Alternative Analysis)
Determine the business importance and the ease of implementation of all win conditions. Reveal issues and constraints.

WinWin Tree (Group Outliner)
Identify Issues and Options. Negotiate Agreements.

Organize Negotiation Results (Categorizer)
Categorize the results using the negotiation topics.

Red cells indicate lack of consensus.

Oral discussion of cell graph reveals unshared information, unnoticed assumptions, hidden issues, constraints, etc.



	Features	Impo	ase of Implementation	Total	Mean
2.	Application Capabilities				
2.1	W2 Integrate banner ads with email and chat		6.50	16.50	8.25
2.2	W3 The banner will provide a link to the universit	.00	10.00	20.00	10.00
2.3	W4 Interface for advertisers to select their sched	8.67	3.00	11.67	5.83
2.4	W5 Default banner of bookstore if no other events	8.00	10.00	18.00	9.00
2.5	W6 The site management must have a website which	10.00	10.00	20.00	10.00
2.6	W7 Different kinds of advertising, including sales	10.00	10.00	20.00	10.00
2.7	W8 Flexible text on banners	10.00	5.00	15.00	7.50
2.8	W9 Display address of the bookstore, a map of it a	4.00	7.50	11.50	5.75
2.9	W10 Ads must be hyperlinked so that users can clic	7.33	6.00	13.33	6.67
2.10	W11 Link to bookstore site (incl book's prices)	9.33	10.00	19.33	9.67
2.11	W12 Web statistics tracking to determine number of	8.00	4.00	12.00	6.00
2.12	W13 Input of banner contents to admin via email	5.50	10.00	15.50	7.75

Example Project: Sierra Mountainbikes

- Based on what would have worked on a similar project
- Quality leader in specialty area
- Competitively priced
- Major problems with order processing
  - Delivery delays and mistakes
  - Poor synchronization of order entry, confirmation, fulfillment
  - Disorganized responses to problem situations
  - Excess costs; low distributor satisfaction

Order Processing Project GQM

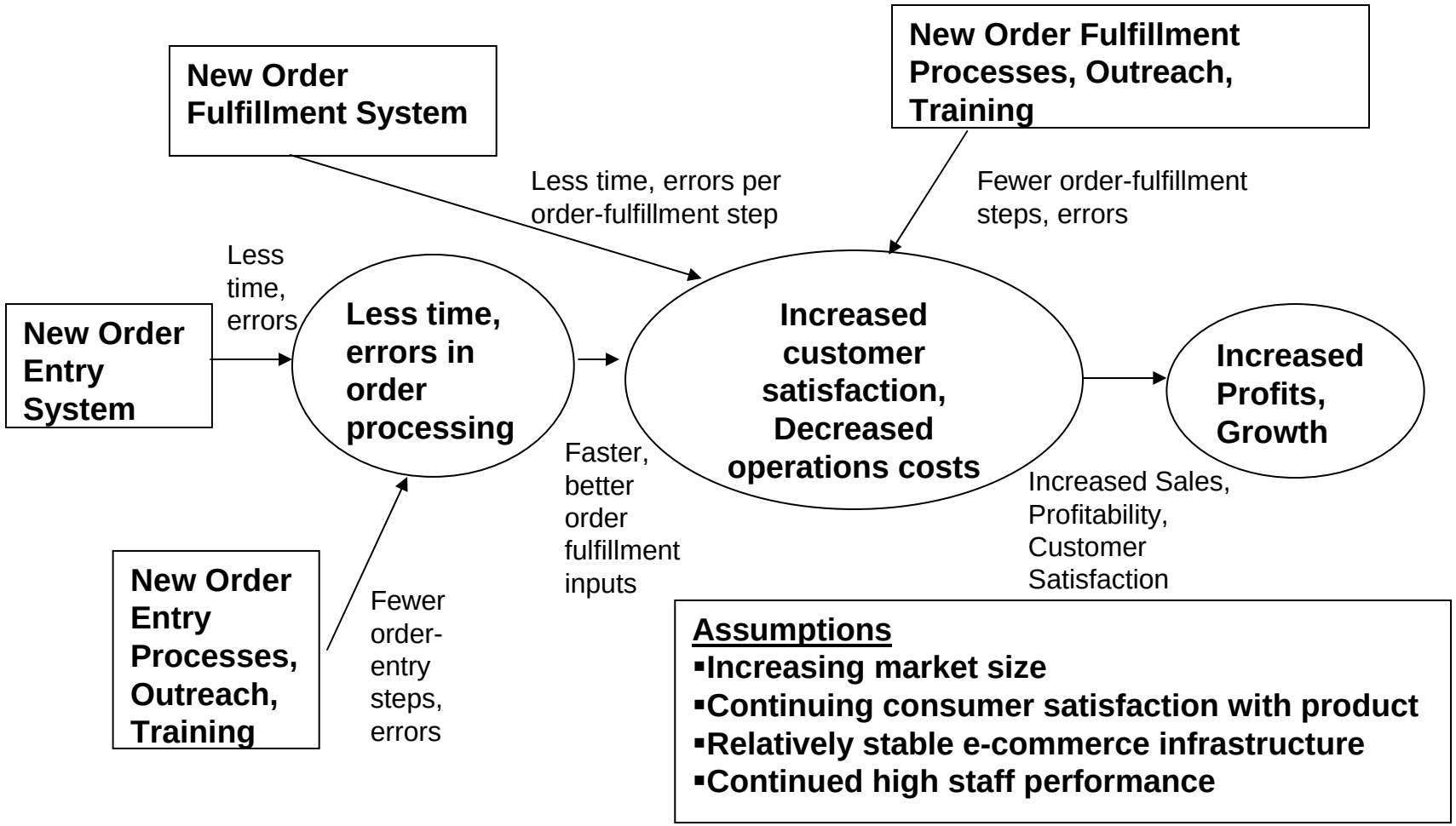
Goals: Improve profits, market share, customer satisfaction via improved order processing

Questions: Current state? Root causes of problems? Keys to improvement?

Metrics: Balanced Scorecard of benefits realized, proxies

- Customer satisfaction ratings; key elements (ITV: in-transit visibility)
- Overhead cost reduction
- Actual vs. expected benefit and cost flows, ROI

Expanded Order Processing System Results Chain



Project Strategy and Partnerships

- Partner with eServices, Inc. for order processing and fulfillment system
  - Profit sharing based on jointly-developed business case
- Partner with key distributors to provide user feedback
  - Evaluate prototypes, beta-test early versions, provide satisfaction ratings
- Incremental development using MBASE/RUP anchor points
  - Life Cycle Objectives; Architecture (LCO; LCA)
  - Core Capability Drivethrough (CCD)
  - Initial; Full Operational Capability (IOC; FOC)

Business Case Analysis

- Estimate costs and schedules
  - COCOMO II and/or alternative
- Estimate financial benefits
  - Increased profits
  - Reduced operating costs
- Compute Return on Investment
  - ROI = (Benefits – Costs) / Costs
  - Normalized to present value
- Identify quantitative metrics for other goals
  - Customer satisfaction ratings
    - Ease of use; In-transit visibility; overall
  - Late delivery percentage

Order Processing System Schedules and Budgets

Milestone	Due Date	Budget ($K)	Cumulative Budget ($K)
Inception Readiness	1/1/2004	0	0
Life Cycle Objectives	1/31/2004	120	120
Life Cycle Architecture	3/31/2004	280	400
Core Capability Drivethrough	7/31/2004	650	1050
Initial Oper. Capability: SW	9/30/2004	350	1400
Initial Oper. Capability: HW	9/30/2004	2100	3500
Developed IOC	12/31/2004	500	4000
Responsive IOC	3/31/2005	500	4500
Full Oper. Cap'y CCD	7/31/2005	700	5200
FOC Beta	9/30/2005	400	5600
FOC Deployed	12/31/2005	400	6000
Annual Oper. & Maintenance		3800	
Annual O&M; Old System		7600	

Order Processing System: Expected Benefits and Business Case

		Current System			New System											
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 %	Cust. Statis. 0-5	In-Tran. Visib. 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

A Real Earned Value System

- Current "earned value" systems monitor cost and schedule, not business value
  - Budgeted cost of work performed ("earned")
  - Budgeted cost of work scheduled ("yearned")
  - Actual costs vs. schedule ("burned")
- A real earned value system monitors benefits realized
  - Financial benefits realized vs. cost (ROI)
  - Benefits realized vs. schedule
    - Including non-financial metrics
  - Actual costs vs. schedule

Value-Based Expected/Actual Outcome Tracking Capability

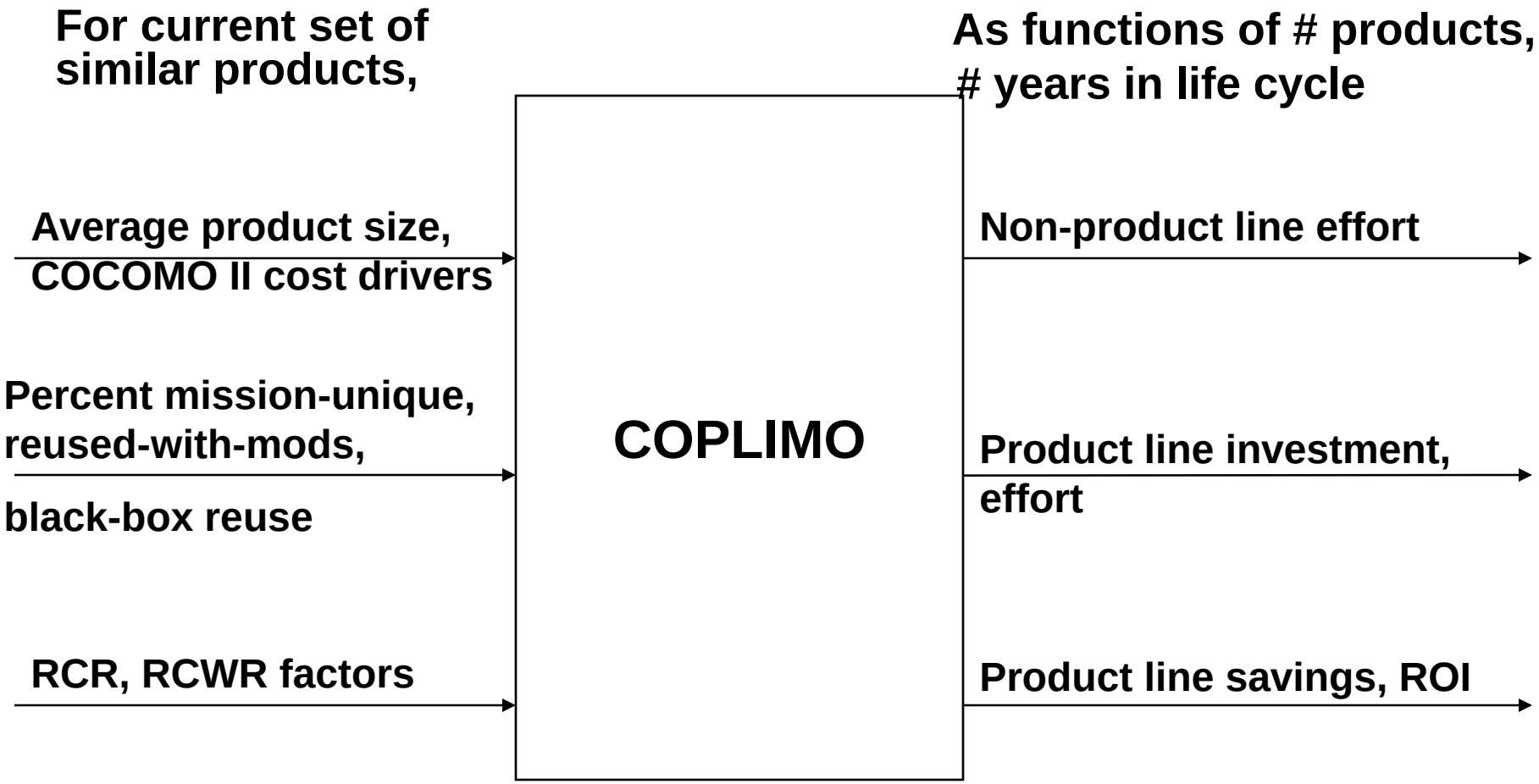
Milestone	Schedule	Cost ($K)	Op-Cost Savings	Market Share %	Annual Sales ($M)	Annual Profits ($M)	CumΔ Profits	ROI	Late Deliv %	Cust. Sat.	ITV	Ease of Use	Risks/Opportunities
Life Cycle Architecture	3/31/04 3/31/04	400 427		20 20	72 72	7.0 7.0			12.4 12.4	1.7 1.7	1.0 1.0	1.8 1.8	Increased COTS ITV Risk. Fallback identified.
Core Capa-bility Demo (CCD)	7/31/04 7/20/04	1050 1096								2.4*	1.0*	2.7*	Using COTS ITV Fallback. New HW Competitor; renegotiating HW
Software Init. Op. Cap'y (IOC)	9/30/04 9/30/04	1400 1532								2.7*	1.4*	2.8*	
Hardware IOC	9/30/04 10/11/04	3500 3432											$200K savings from renegotiated HW
Deployed IOC	12/31/04 12/20/04	4000 4041		20 22	80 88	8.0 8.6	0.0 0.6	-1.0 -.85	11.4 10.8	3.0 2.8	2.5 1.6	3.0 3.2	New COTS ITV source identified, being protoyped
Responsive IOC	3/31/05 3/30/05	4500 4604	300 324						9.0 7.4	3.5 3.3	3.0 1.6	3.5 3.8	
Full Op. Cap'y CCD	7/31/05 7/28/05	5200 5328	1000 946							3.5*	2.5*	3.8*	New COTS ITV source initially integrated
Full Op. Cap'y Beta	9/30/05 9/30/05	5600 5689	1700 1851							3.8*	3.1*	4.1*	
Full Op. Cap'y Deployed Release 2.1	12/31/05 12/20/05 6/30/06	6000 5977 6250	2200 2483	22 24	106 115	12.2 13.5	3.2 5.1	-.47 -.15	7.0 4.8	4.0 4.1	3.5 3.3	4.0 4.2	

The COPLIMO Model

– Constructive Product Line Investment Model

- **Based on COCOMO II software cost model**
  - Statistically calibrated to 161 projects, representing 18 diverse organizations
- **Based on standard software reuse economic terms**
  - RCR: Relative cost of reuse
  - RCWR: Relative cost of writing for reuse
- **Avoids overestimation**
  - Avoids RCWR for non-reused components
  - Adds life cycle cost savings
- **Provides experience-based default parameter values**
- **Simple Excel spreadsheet model**
  - Easy to modify, extend, interoperate

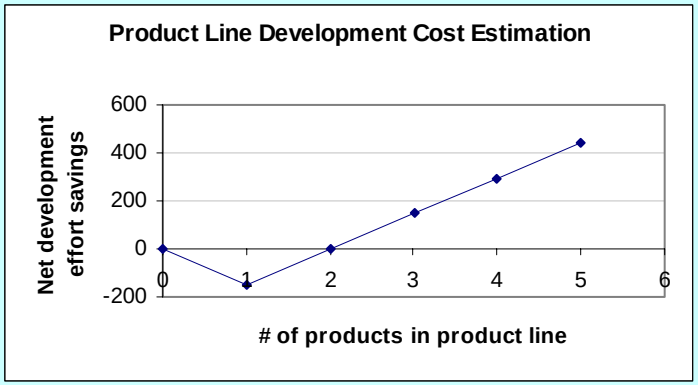
COPLIMO Inputs and Outputs



COPLIMO Estimation Summary

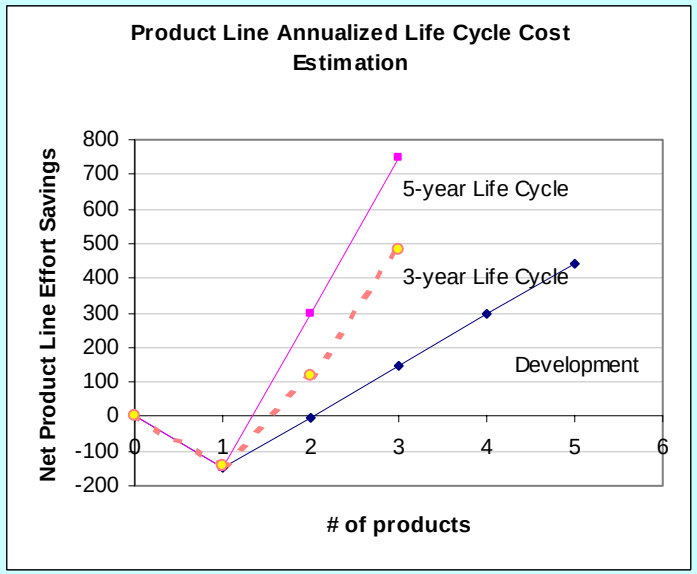
Part I: Product Line Development Cost Estimation Summary:

# of Products Effort (PM)	0	1	2	3	4	5
No Reuse	0	294	588	882	1176	1470
Product Line	0	444	589	735	881	1026
Product Line Savings	0	-150	-1	147	295	444
ROI	0	-1.00	-0.01	0.98	1.97	2.96



Part II: Product Line Annualized Life Cycle Cost Estimation Summary:

# of Products	0	1	2	3	4	5
AMSIZE-P	0	8.1	16.2	24.2	32.3	40.4
AMSIZE-R	0	6.1	6.1	6.1	6.1	6.1
AMSIZE-A	0	6.1	7.7	9.3	11.0	12.6
Total Equiv. KSLOC	0	20.2	29.9	39.6	49.3	59.1
Effort (AM) (*2.94)	0	59.4	88.0	116.5	145.1	173.7
5-year Life Cycle PM	0	296.9	439.8	582.6	725.4	868.3
PM(N, 5)-R (+444)	0	740.9	883.7	1026.5	1169.4	1312.2
PM(N, 5)-NR	0	590.9	1181.9	1772.8	2363.8	2954.7
Product Line Savings (PM)	0	-149.9	298.2	746.3	1194.4	1642.5
ROI	0	-1.00	1.99	4.98	7.97	10.96
Devel. ROI	0	-1.00	-0.01	0.98	1.97	2.96
3-year Life Cycle	0	-142.0	120.0	480.0		

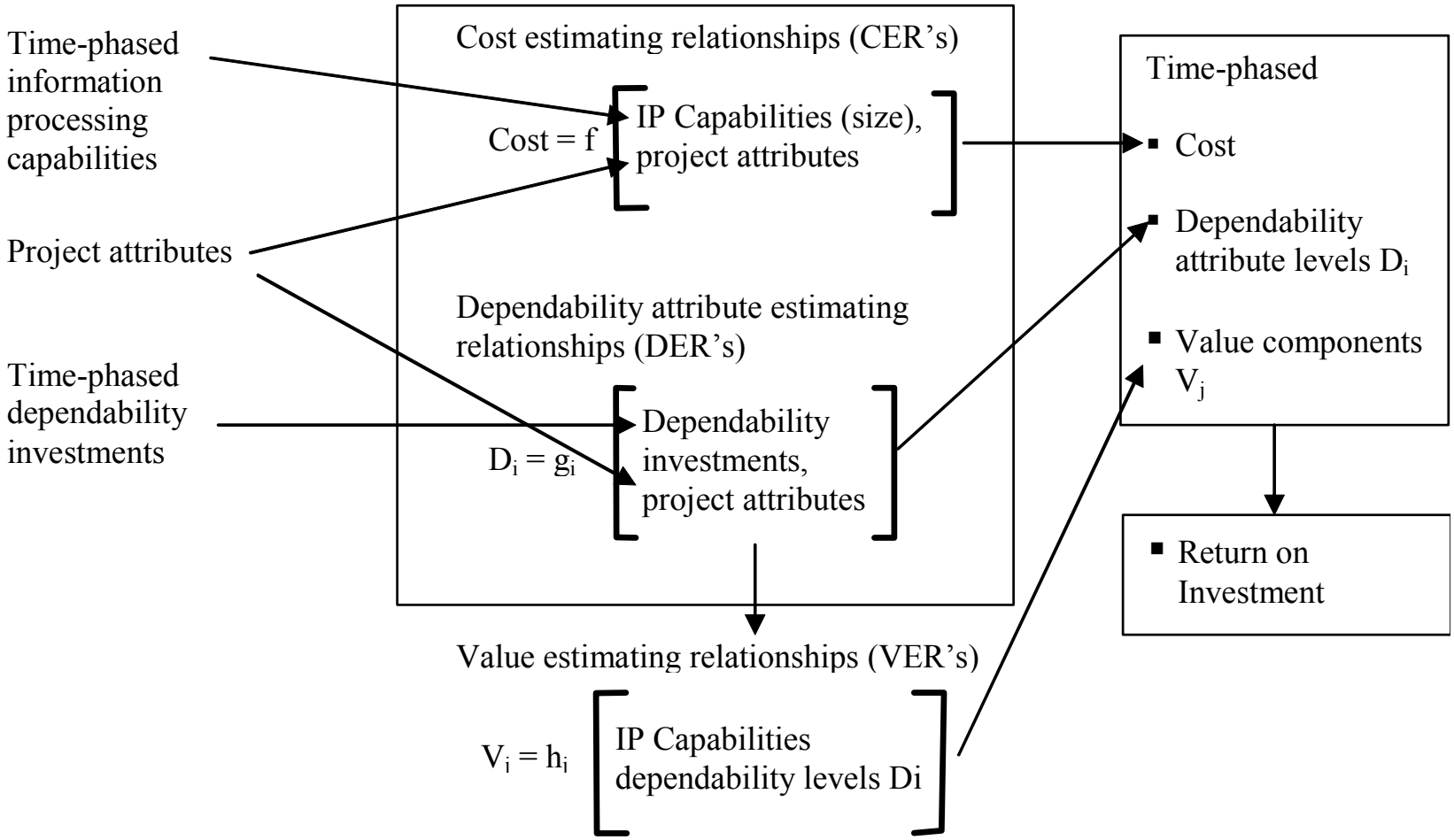


AMSIZE: Annually Maintained Software Size

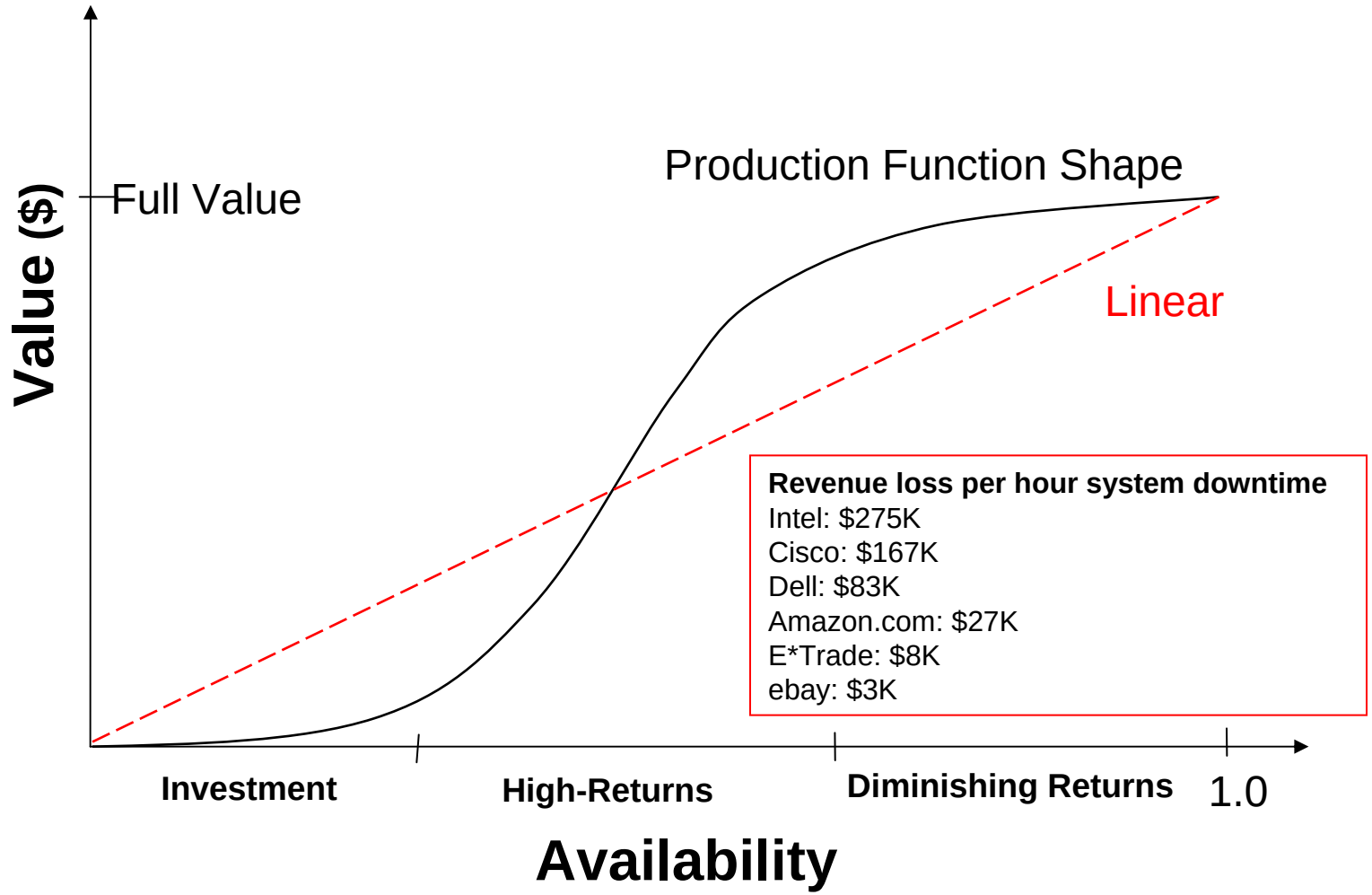
Reasoning about the Value of Dependability – iDAVE

- iDAVE: Information Dependability Attribute Value Estimator
- Use iDAVE model to estimate and track software dependability ROI
  - Help determine how much dependability is enough
  - Help analyze and select the most cost-effective combination of software dependability techniques
  - Use estimates as a basis for tracking performance

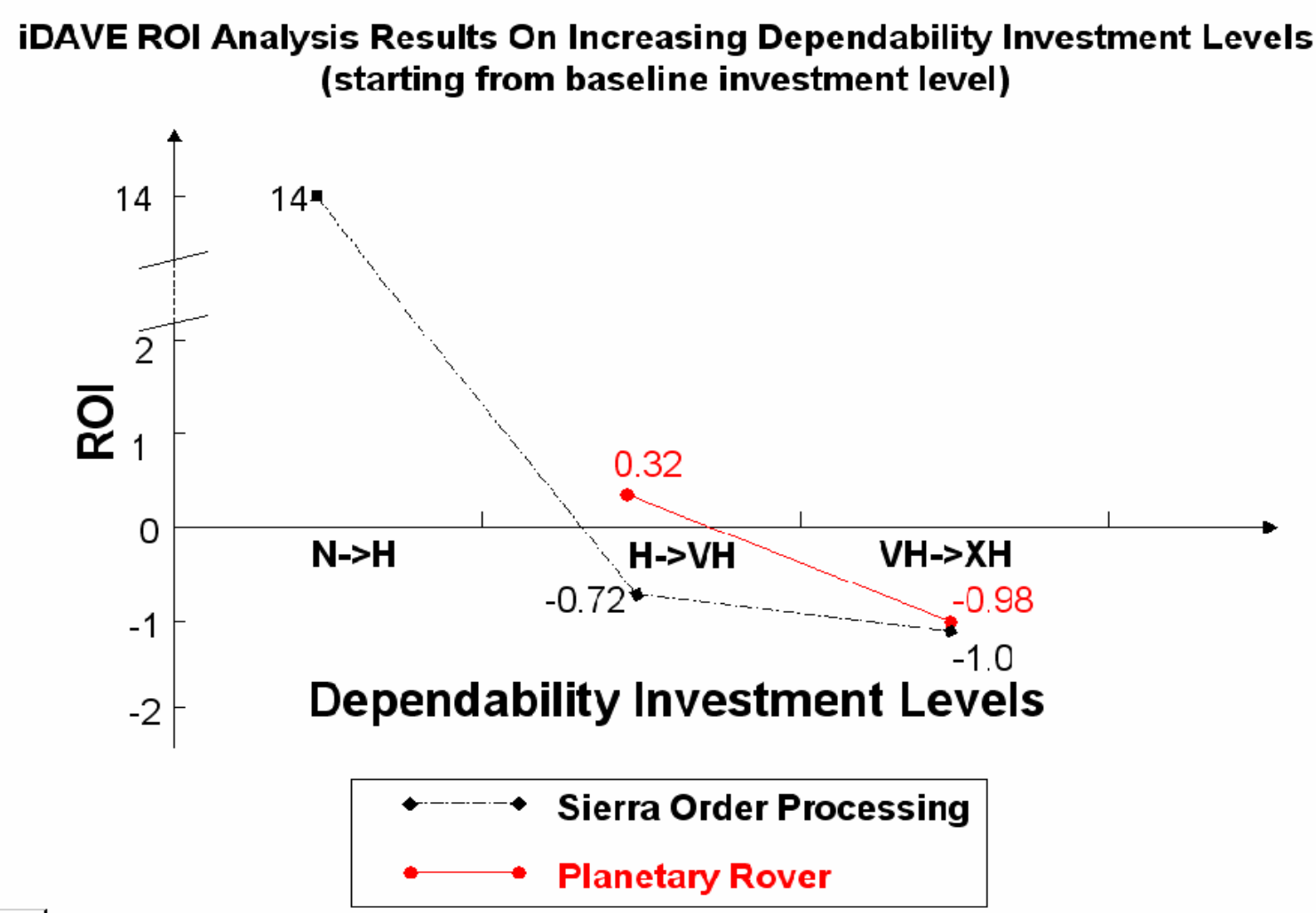
iDAVE Model Framework



Typical Value Estimating Relationships

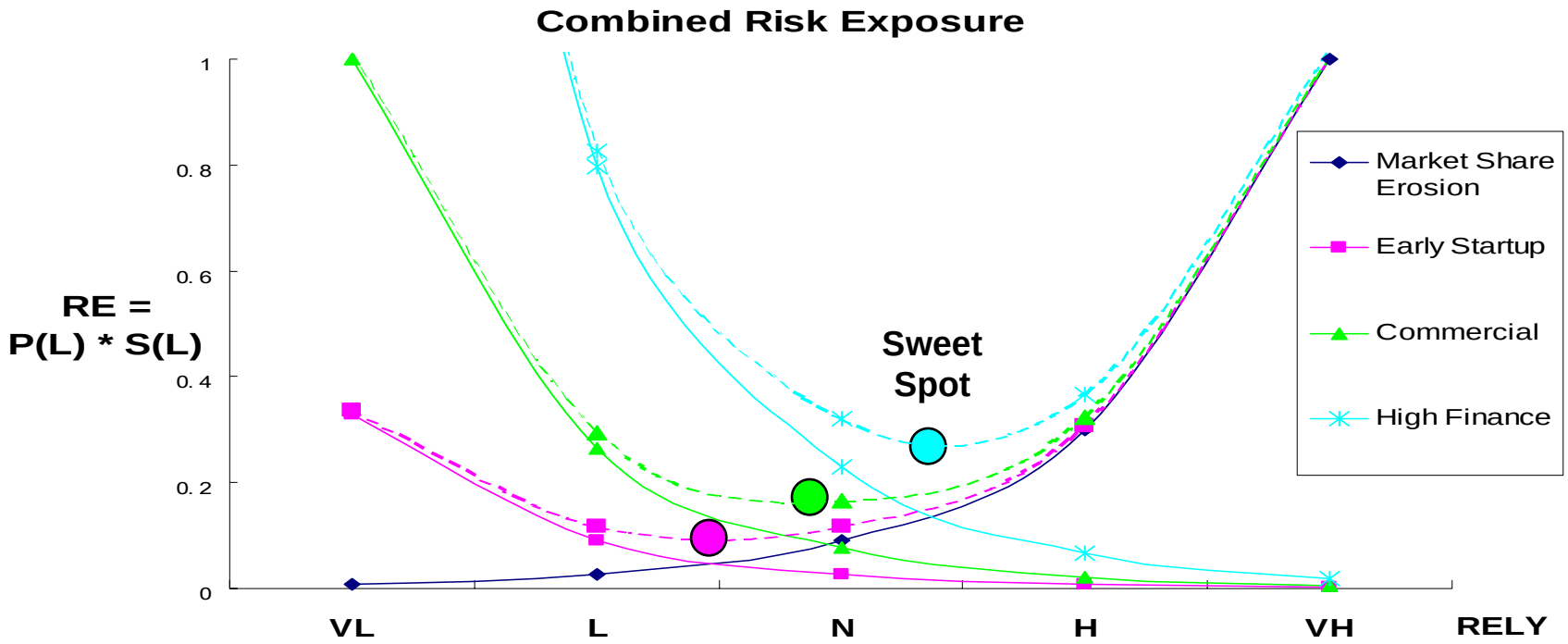


ROI Analysis Results Comparison



How much Dependability is Enough?

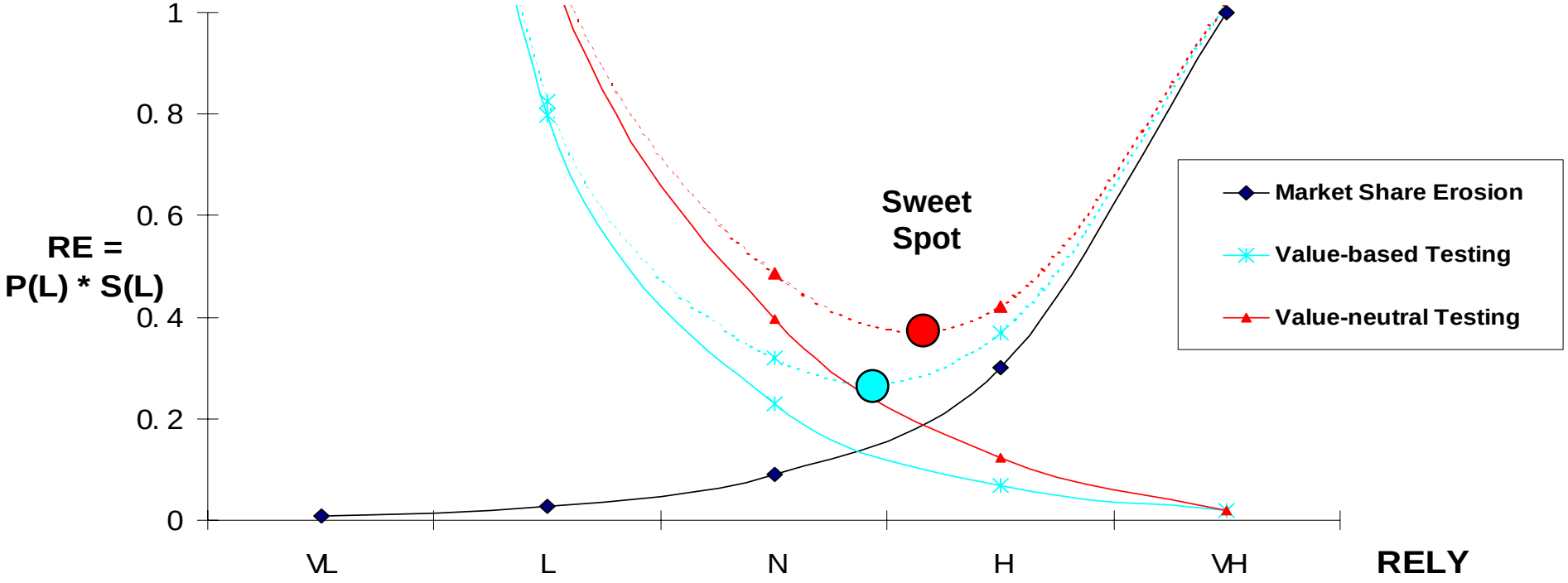
- Nominal Defect Introduction Rate (60 defects/KSLOC)



COCOMO II:	0	12	22	34	54	Added % test time
COQUALMO:	1.0	.475	.24	.125	0.06	P(L)
Early Startup:	.33	.19	.11	.06	.03	S(L)
Commercial:	1.0	.56	.32	.18	.10	S(L)
High Finance:	3.0	1.68	.96	.54	.30	S(L)
Market Risk:	.008	.027	.09	.30	1.0	RE_m

Value-Based vs. Value-Neutral Testing – High Finance

Combined Risk Exposure

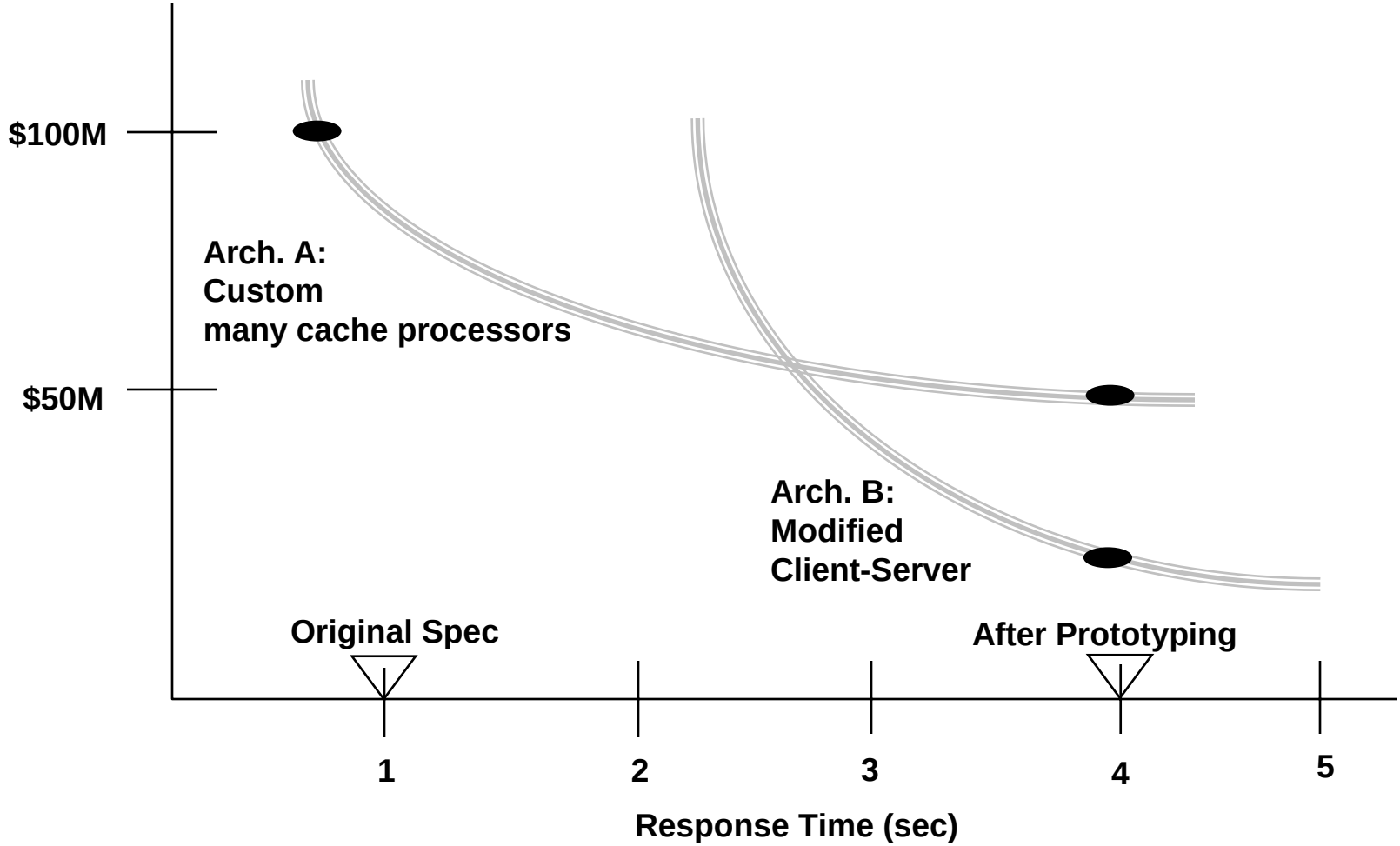


COCOMO II:	0	12	22	34	54	Added % test time
COQUALMO:	1.0	.475	.24	.125	0.06	P(L)
Value-based:	3.0	1.68	.96	.54	.30	S(L): Exponential
Value-Neutral:	3.0	2.33	1.65	.975	.30	S(L): Linear
Market Risk:	.008	.027	.09	.30	1.0	RE_m

7 Key Elements of VBSE

1. Benefits Realization Analysis
2. Stakeholders' Value Proposition Elicitation and Reconciliation
3. Business Case Analysis
4. Continuous Risk and Opportunity Management
5. Concurrent System and Software Engineering
6. Value-Based Monitoring and Control
7. Change as Opportunity

Sequential Engineering Neglects Risk

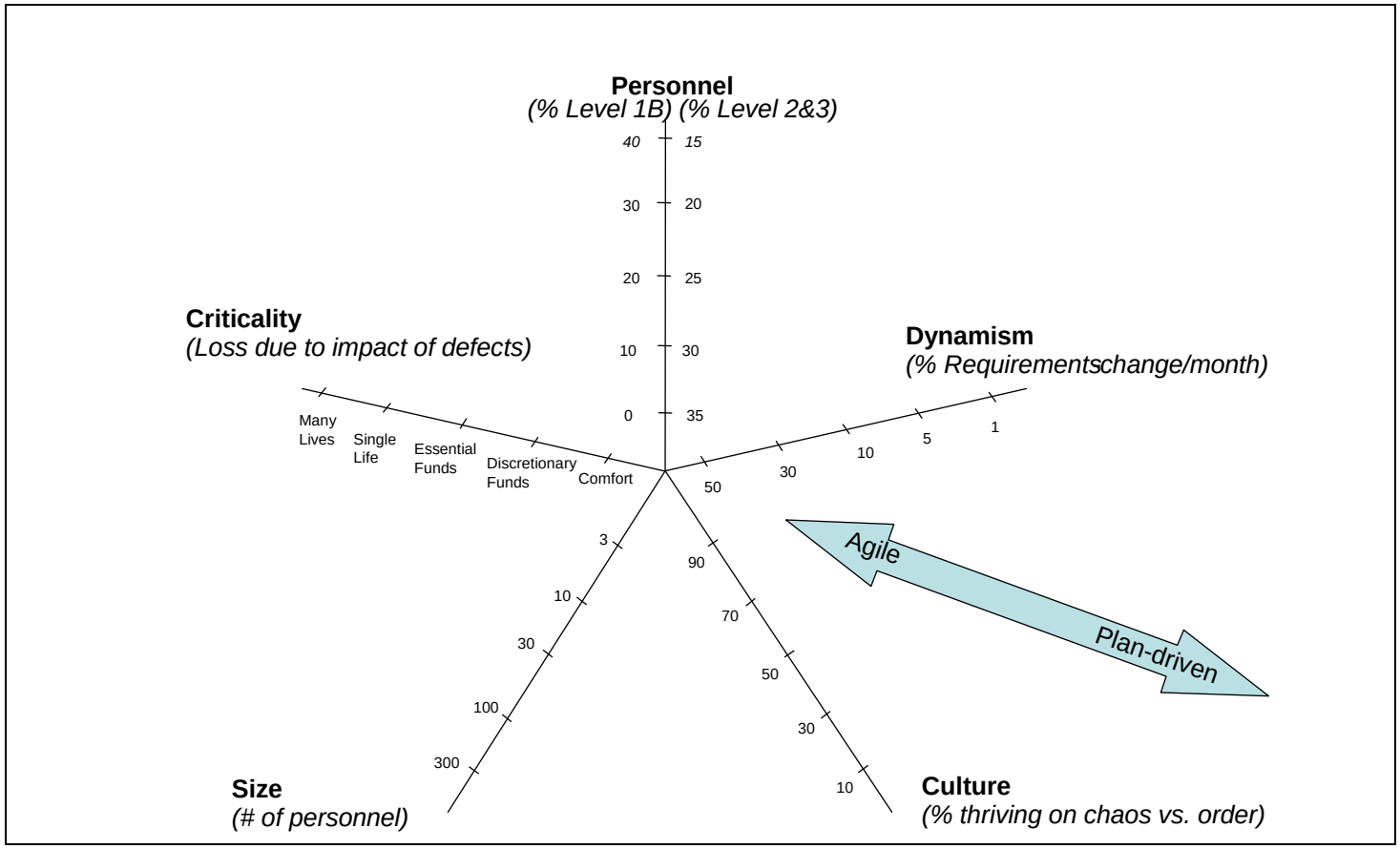


Change As Opportunity: Agile Methods

- Continuous customer interaction
- Short value - adding increments
- Tacit interpersonal knowledge
  - Stories, Planning game, pair programming
  - Explicit documented knowledge expensive to change
- Simple design and refactoring
  - Vs. Big Design Up Front
- Some automation activities
  - Story cards; lightweight earned value; test-first

Five Critical Decision Factors

- **Represent five dimensions**
- **Size, Criticality, Dynamism, Personnel, Culture**



Outline

- Motivation and definitions
- Seven key practices
  - Examples of automation
- VBSE automation challenges
- Conclusions; references

VBSE Automation Challenges – Many opportunities for further research

- Characterizing, incorporating value estimation relationships
- Integrating cost models and benefit models
- Integrating these with collaboration tools, financial tools
- More powerful risk analysis tools (JPL DDP)
- Other value-based dependability attribute analysis
  - Security, safety, performance
- Value-based enhancements of traditional tools
  - Many opportunities for improving cost-effectiveness

Value-Based Enhancements of Traditional Tools

- Real-value earned-value management tools
- Value-based/risk-based test tools, defect closure tracking
- Value-based review checklists and guidelines
- Value-based requirements tools (marketable features)
- Real-options analysis of architecture investments
- Value-based cost/schedule tradeoff tools

Conclusions

- Marketplace trends favor transition to VBSE paradigm
  - Software a/the major source of product value
  - Software the primary enabler of adaptability
- VBSE involves 7 key elements
  1. Benefits Realization Analysis
  2. Stakeholders' Value Proposition Elicitation and Reconciliation
  3. Business Case Analysis
  4. Continuous Risk and Opportunity Management
  5. Concurrent System and Software Engineering
  6. Value-Based Monitoring and Control
  7. Change as Opportunity
- Processes for implementing VBSE emerging
  - CeBASE Method, CMMI, DMR/BRA, Balanced Scorecard, RUP extensions, Strategic Design, Agile Methods

References

C. Baldwin & K. Clark, Design Rules: The Power of Modularity, MIT Press, 1999.

B. Boehm, "Value-Based Software Engineering," ACM Software Engineering Notes, March 2003.

B. Boehm, C. Abts, A.W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, Software Cost Estimation with COCOMO II, Prentice Hall, 2000.

B. Boehm and L. Huang, "Value-Based Software Engineering: A Case Study, Computer, March 2003, pp. 33-41.

B. Boehm & K. Sullivan, "Software Economics: A Roadmap," The Future of Software Economics, A. Finkelstein (ed.), ACM Press, 2000.

B. Boehm and R. Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Addison Wesley, 2003 (to appear).

B. Boehm, L. Huang, A. Jain. R. Madachy, " The ROI of Software Dependability: The iDAVE Model", IEEE Software Special Issue on Return on Investment, May/June 2004.

M. Denne and J. Cleland-Huang, Software by Numbers, Prentice Hall, 2004.

S. Faulk, D. Harmon, and D. Raffo, "Value-Based Software Engineering (VBSE): A Value-Driven Approach to Product-Line Engineering," Proceedings, First Intl. Conf. On SW Product Line Engineering, August 2000.

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

D. Reifer, Making the Software Business Case, Addison Wesley, 2002.

K. Sullivan, Y. Cai, B. Hallen, and W. Griswold, "The Structure and Value of Modularity in Software Design," Proceedings, ESEC/FSE, 2001, ACM Press, pp. 99-108.

J. Thorp and DMR, The Information Paradox, McGraw Hill, 1998.

S. Tockey, Return on Software, Addison Wesley, 2004.

Economics-Driven Software Engineering Research (EDSER) web site: www.edser.org

MBASE web site : sunset.usc.edu/research/MBASE