A Value-Based Orthogonal Framework for Improving Life-Cycle Affordability

Barry Boehm, Jo Ann Lane, Sue Koolmanojwong
http://csse.usc.edu

NDIA Systems Engineering Conference

October 25, 2012

Outline

- Affordability definitions, concepts, issues, strategies
  - Addressing both costs and benefits
  - Using life cycle present value
  - Coping with uncertainty: incrementally; pro-actively
  - Coping with multi-stakeholder value diversity
  - Addressing tradeoffs with other -ilities
- An orthogonal framework for improving affordability costs
  - Cost modeling and other insights
- Conclusions

Affordability Definitions

- **INCOSE: The balance of system performance, cost, and schedule constraints over the system life cycle, while satisfying mission needs in concert with strategic and organizational needs.**

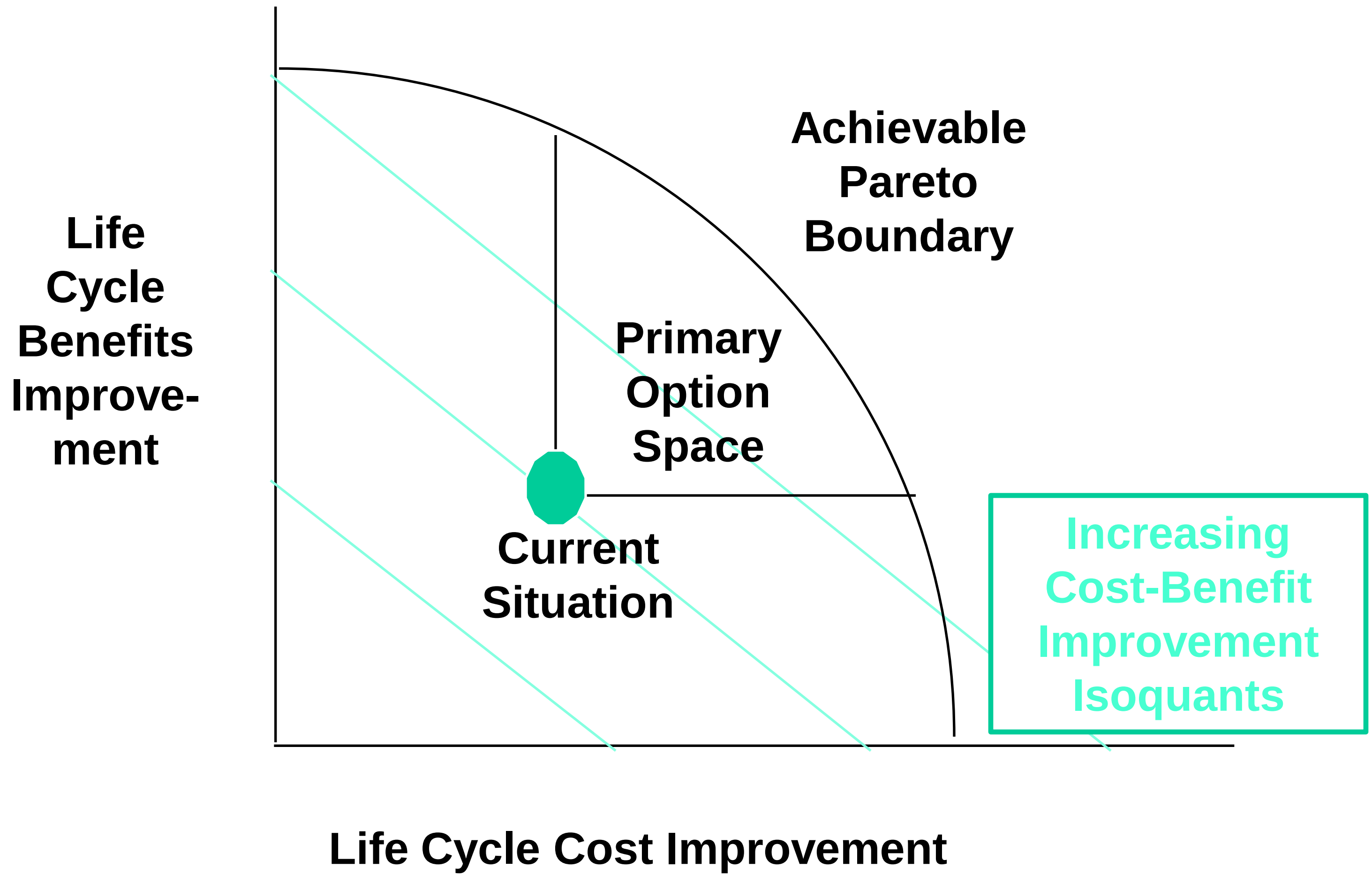
- **MORS: Cost-effective capability (USD/ATL).**

- **NDIA: The practice of assuring program success through the balancing of system performance (KPPs), cost, and schedule constraints, while satisfying mission needs in concert with the long-range investment and force structure plans of the DoD.**

- **Webster: Keeping within your financial means.**

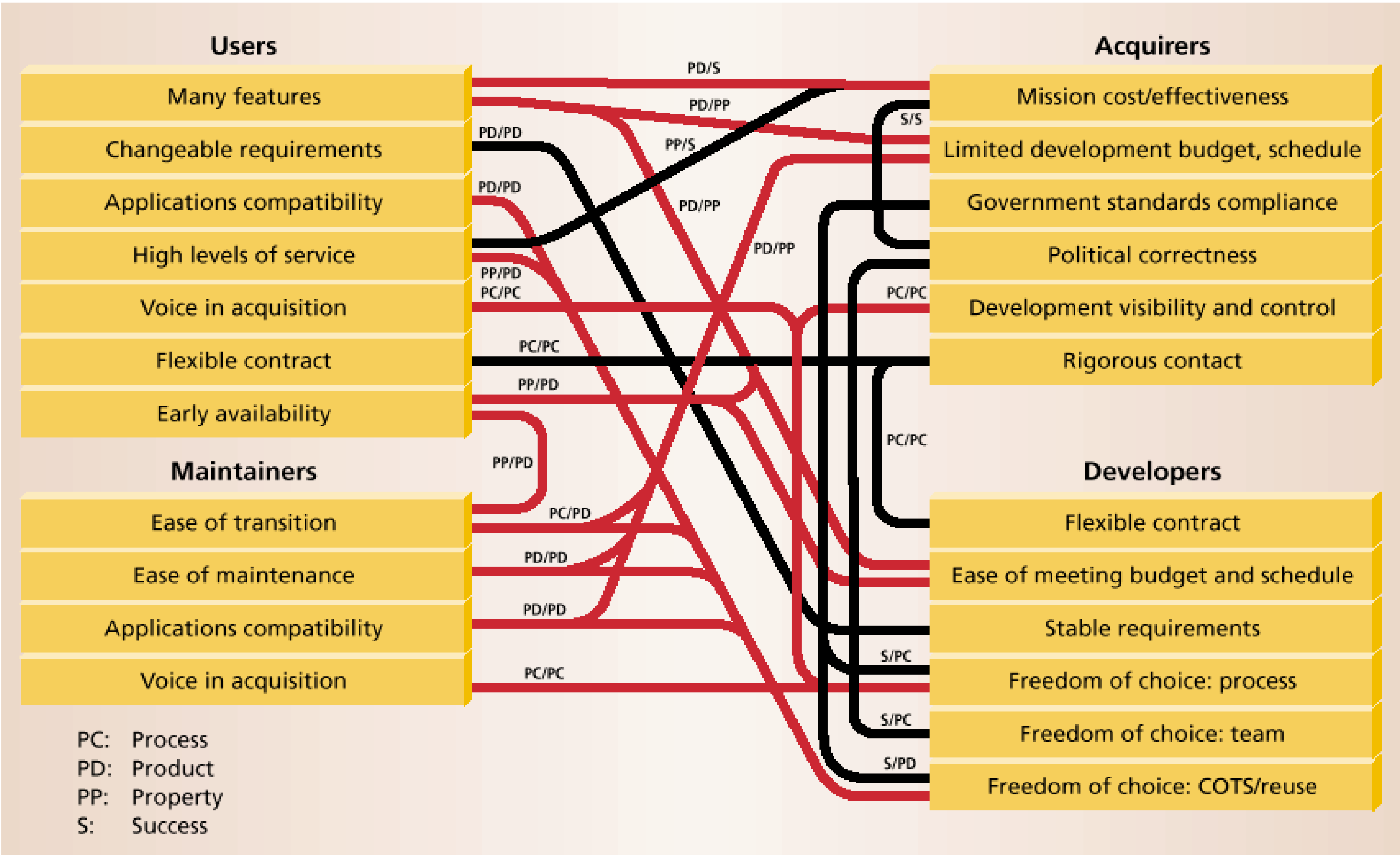
Affordability Concepts

Coping with uncertainty: incrementally; pro-actively
Coping with multi-stakeholder value diversity



Multi-Stakeholder Value Diversity

Bank of America Master Net



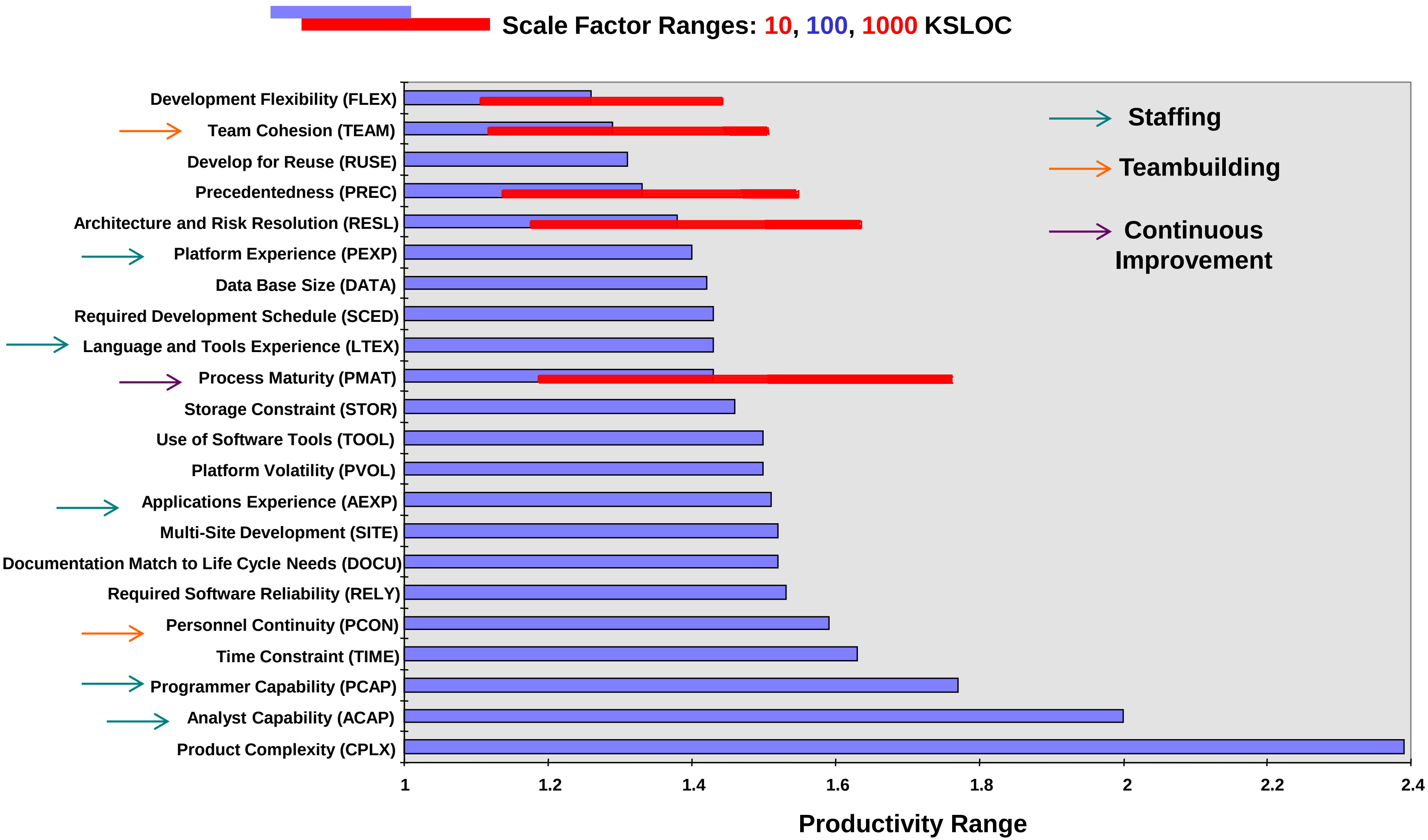
Outline

- Affordability definitions, concepts, issues, strategies
  - Addressing both costs and benefits
  - Using life cycle present value
  - Coping with uncertainty: incrementally; pro-actively
  - Coping with multi-stakeholder value diversity
  - Addressing tradeoffs with other -ilities
- An orthogonal framework for improving affordability costs
  - Cost modeling and other insights
- Conclusions

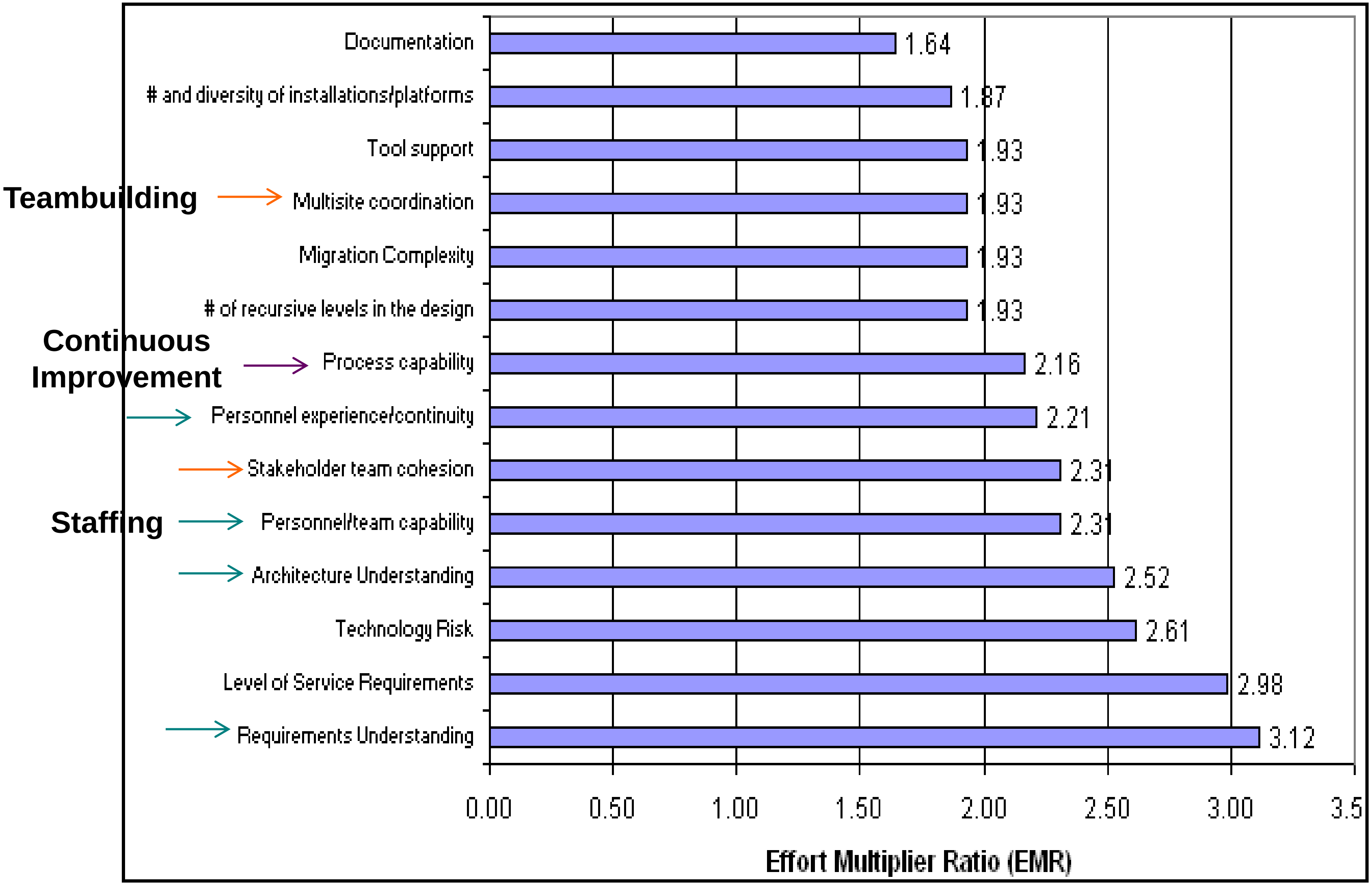
Affordability and Tradespace Framework

- **Affordability Improvements and Tradeoffs**
  - **Get the Best from People**
    - Staffing, Incentivizing, Teambuilding
    - Facilities, Support Services
    - Kaizen (continuous improvement)
  - **Make Tasks More Efficient**
    - Tools and Automation
    - Work and Oversight Streamlining
    - Collaboration Technology
  - **Eliminate Tasks**
    - Lean and Agile Methods
    - Task Automation
    - Model-Based Product Generation
  - **Eliminate Scrap, Rework**
    - Early Risk and Defect Elimination
    - Evidence-Based Decision Gates
    - Modularity Around Sources of Change
    - Incremental, Evolutionary Development
    - Value-Based, Agile Process Maturity
    - Risk-Based Prototyping
  - **Simplify Products (KISS)**
    - Value-Based Capability Prioritization
    - Satisficing vs. Optimizing Performance
  - **Reuse Components**
    - Domain Engineering and Architecture
    - Composable Components,Services, COTS
    - Legacy System Repurposing
  - **Reduce Operations, Support Costs**
    - Automate Operations Elements
    - Design for Maintainability, Evolvability
    - Streamline Supply Chain
    - Anticipate, Prepare for Change
  - **Value- and Architecture-Based Tradeoffs and Balancing**

Costing Insights: COCOMO II Productivity Ranges



COSYSMO Sys Engr Cost Drivers

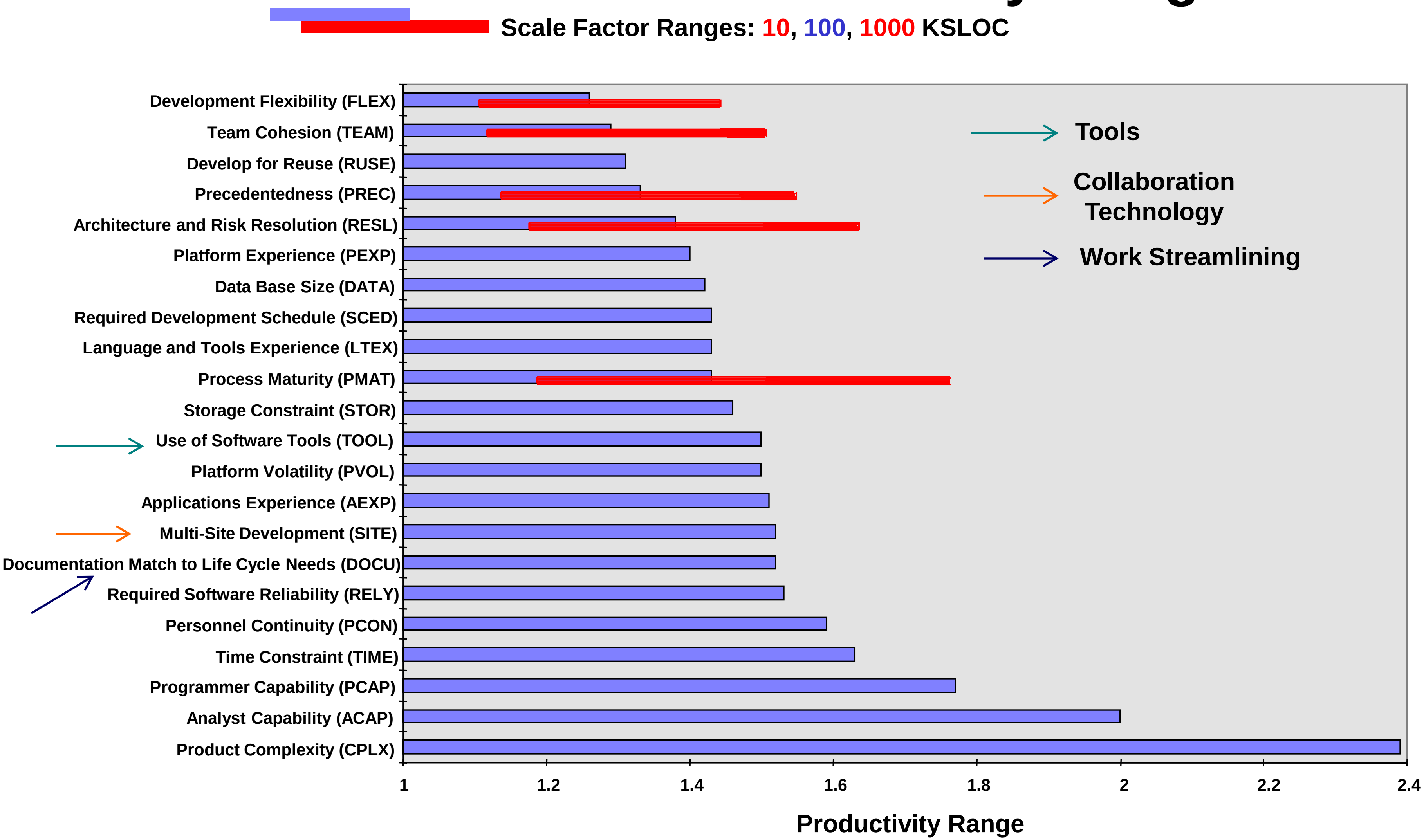


Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

- **Get the Best from People**
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)
- **Make Tasks More Efficient**
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology
- **Eliminate Tasks**
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation
- **Eliminate Scrap, Rework**
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity
  - Risk-Based Prototyping
- **Simplify Products (KISS)**
  - Value-Based Capability Prioritization
  - Satisficing vs. Optimizing Performance
- **Reuse Components**
  - Domain Engineering and Architecture
  - Composable Components,Services, COTS
  - Legacy System Repurposing
- **Reduce Operations, Support Costs**
  - Automate Operations Elements
  - Design for Maintainability, Evolvability
  - Streamline Supply Chain
  - Anticipate, Prepare for Change
- **Value- and Architecture-Based Tradeoffs and Balancing**

COCOMO II. 2000 Productivity Ranges

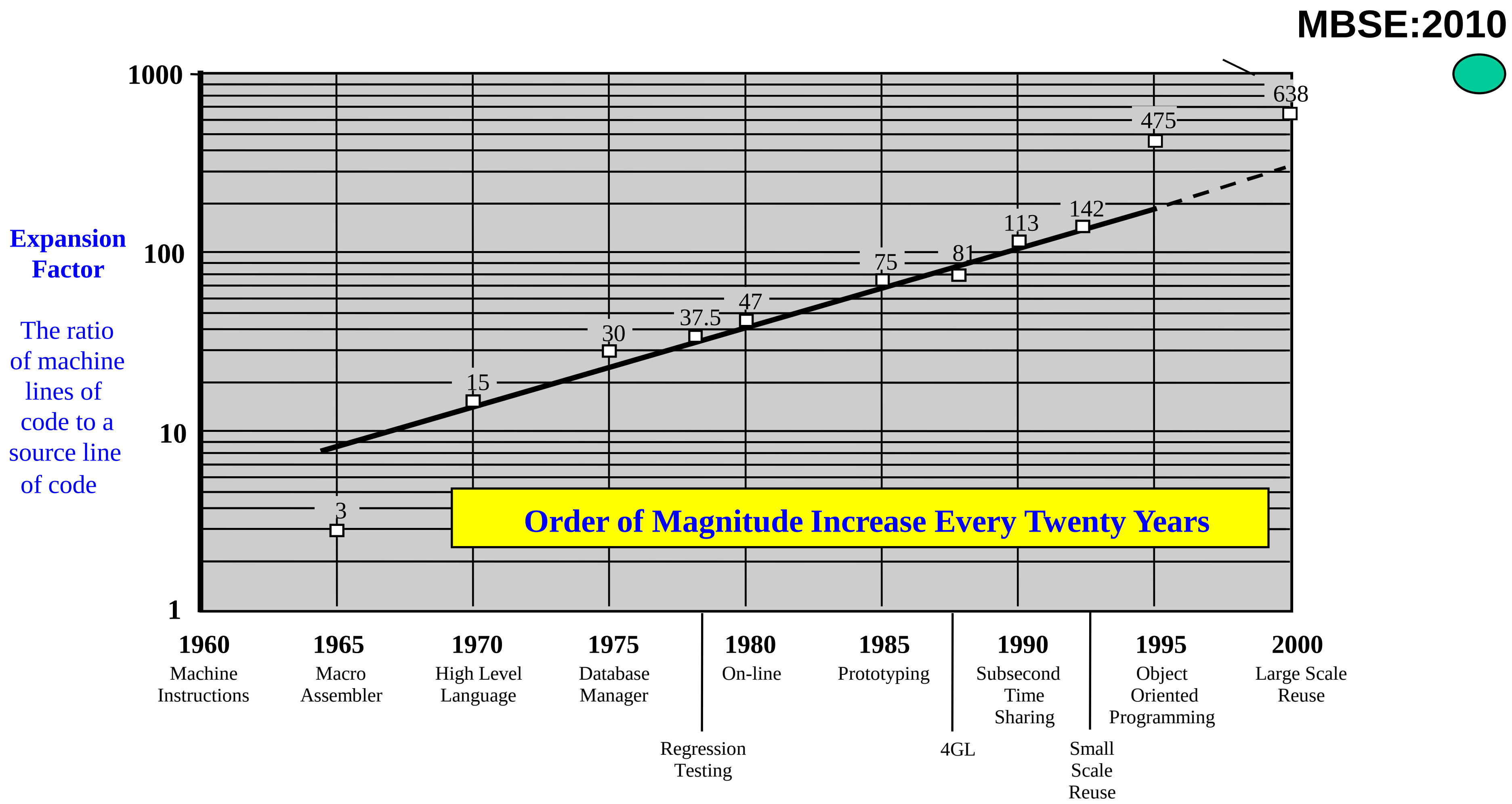


Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

- **Get the Best from People**
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)
- **Make Tasks More Efficient**
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology
- **Eliminate Tasks**
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation
- **Eliminate Scrap, Rework**
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity
  - Risk-Based Prototyping
- **Simplify Products (KISS)**
  - Value-Based Capability Prioritization
  - Satisficing vs. Optimizing Performance
- **Reuse Components**
  - Domain Engineering and Architecture
  - Composable Components,Services, COTS
  - Legacy System Repurposing
- **Reduce Operations, Support Costs**
  - Automate Operations Elements
  - Design for Maintainability, Evolvability
  - Streamline Supply Chain
  - Anticipate, Prepare for Change
- **Value- and Architecture-Based Tradeoffs and Balancing**

Trends in Software Expansion (Bernstein, 1997)



Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

- **Get the Best from People**
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)
- **Make Tasks More Efficient**
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology
- **Eliminate Tasks**
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation
- **Eliminate Scrap, Rework**
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity
- **Simplify Products (KISS)**
  - Risk-Based Prototyping
  - Value-Based Capability Prioritization
  - Satisficing vs. Optimizing Performance
- **Reuse Components**
  - Domain Engineering and Architecture
  - Composable Components,Services, COTS
  - Legacy System Repurposing
- **Reduce Operations, Support Costs**
  - Automate Operations Elements
  - Design for Maintainability, Evolvability
  - Streamline Supply Chain
  - Anticipate, Prepare for Change
- **Value- and Architecture-Based Tradeoffs and Balancing**

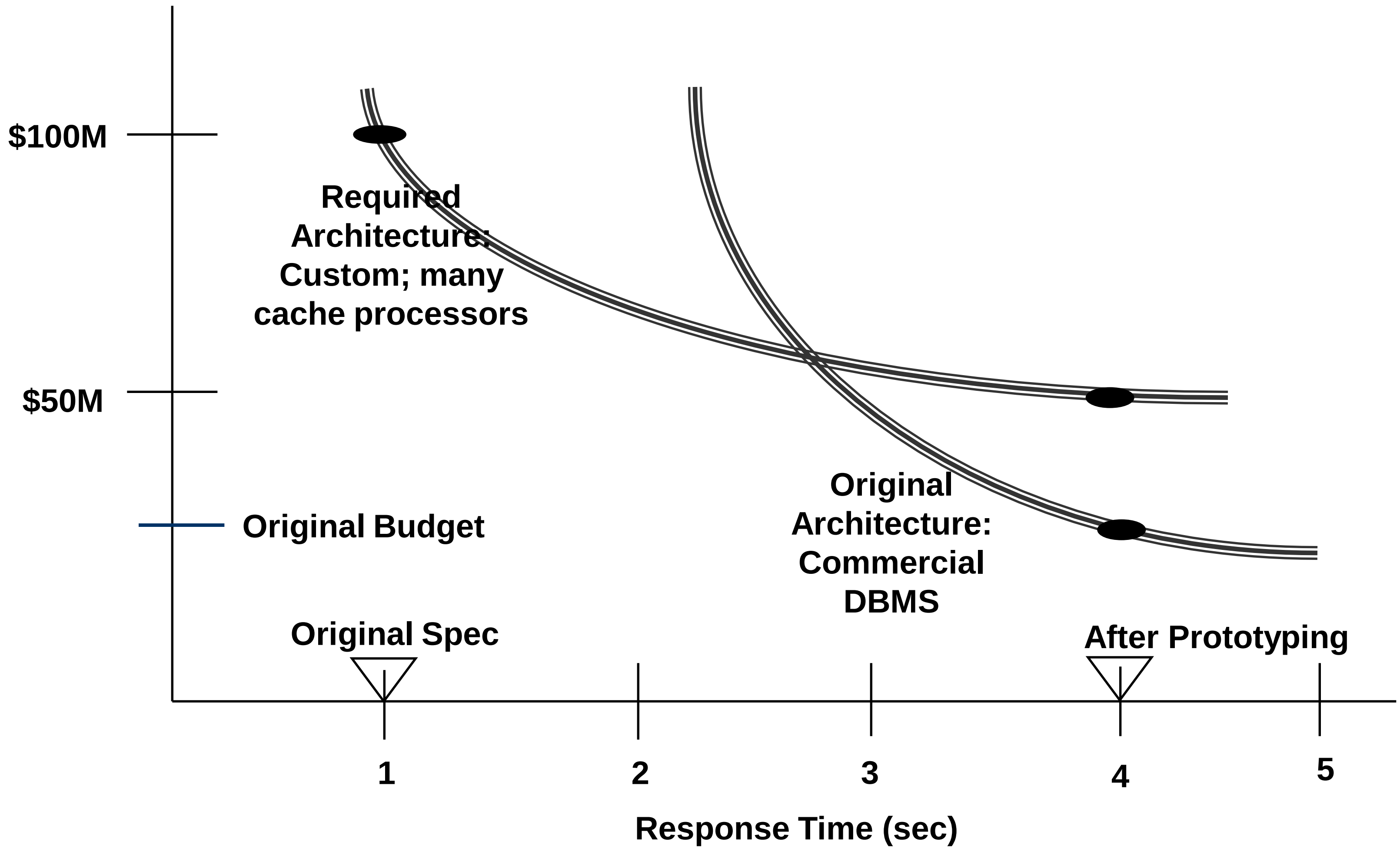
Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

- **Get the Best from People**
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)
- **Make Tasks More Efficient**
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology
- **Eliminate Tasks**
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation
- **Eliminate Scrap, Rework**
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity
  - Risk-Based Prototyping
- **Simplify Products (KISS)**
  - Value-Based Capability Prioritization
  - Satisficing vs. Optimizing Performance
- **Reuse Components**
  - Domain Engineering and Architecture
  - Composable Components,Services, COTS
  - Legacy System Repurposing
- **Reduce Operations, Support Costs**
  - Automate Operations Elements
  - Design for Maintainability, Evolvability
  - Streamline Supply Chain
  - Anticipate, Prepare for Change
- **Value- and Architecture-Based Tradeoffs and Balancing**

Sequential Requirements-First Risks

It's not a requirement if you can't afford it



Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

- **Get the Best from People**
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)
- **Make Tasks More Efficient**
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology
- **Eliminate Tasks**
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation
- **Eliminate Scrap, Rework**
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity
  - Risk-Based Prototyping
- **Simplify Products (KISS)**
  - Value-Based Capability Prioritization
  - Satisficing vs. Optimizing Performance
- **Reuse Components**
  - Domain Engineering and Architecture
  - Composable Components,Services, COTS
  - Legacy System Repurposing
- **Reduce Operations, Support Costs**
  - Automate Operations Elements
  - Design for Maintainability, Evolvability
  - Streamline Supply Chain
  - Anticipate, Prepare for Change
- **Value- and Architecture-Based Tradeoffs and Balancing**

The HMMWV

300,000 Produced, 22 Fielded Versions

Initial draft requirements in 1979, Initial delivery in 1984

Bolt on armor required upgraded suspension, engine, and steering

Additional armor and cupola raise the CG and increase rollovers

Mattracks or wheels

Upgrades:
- **Increased cab space**
- **Increased payload capacity**
- **Strengthened frame**

Upper deck space is always at a premium

Imbalance in cupola required motorized drive

Suspension and steering for CG shift

Base cab & flatbed with mission modules

Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

- **Get the Best from People**
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)
- **Make Tasks More Efficient**
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology
- **Eliminate Tasks**
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation
- **Eliminate Scrap, Rework**
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity
  - Risk-Based Prototyping
- **Simplify Products (KISS)**
  - Value-Based Capability Prioritization
  - Satisficing vs. Optimizing Performance
- **Reuse Components**
  - Domain Engineering and Architecture
  - Composable Components,Services, COTS
  - Legacy System Repurposing
- **Reduce Operations, Support Costs**
  - Automate Operations Elements
  - Design for Maintainability, Evolvability
  - Streamline Supply Chain
  - Anticipate, Prepare for Change
- **Value- and Architecture-Based Tradeoffs and Balancing**

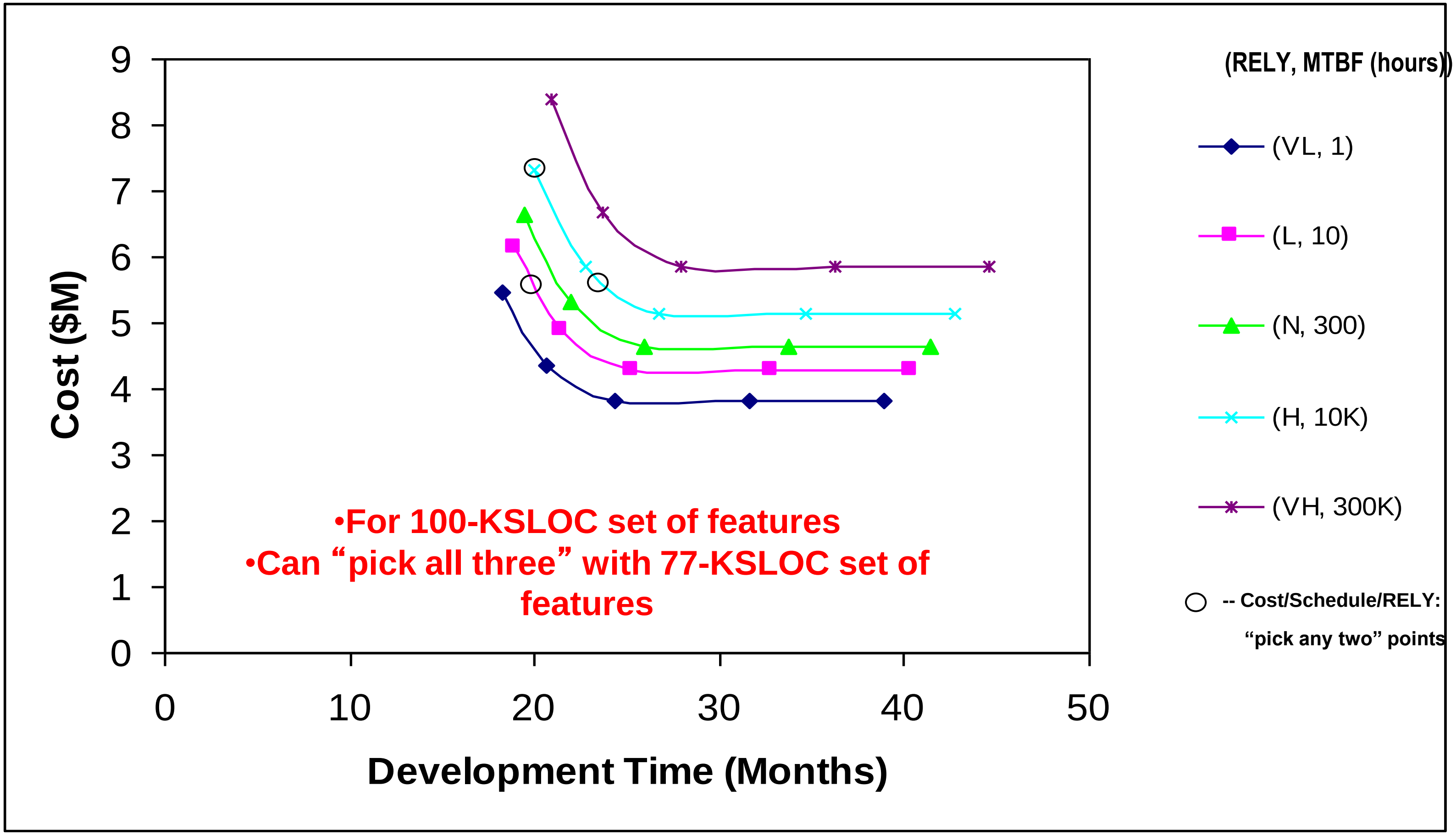
Post-Acquisition Costs Dominate (%O&M)

- Hardware [Redman 2008]
  - 12% -- Missiles (average)
  - 60% -- Ships (average)
  - 78% -- Aircraft (F-16)
  - 84% -- Ground vehicles (Bradley)
- Software [Koskinen 2010]
  - 75-90% -- Business, Command-Control
  - 50-80% -- Complex platforms as above
  - 10-30% -- Simple embedded software
- Apply lack-of-flexibility factor to O&M component

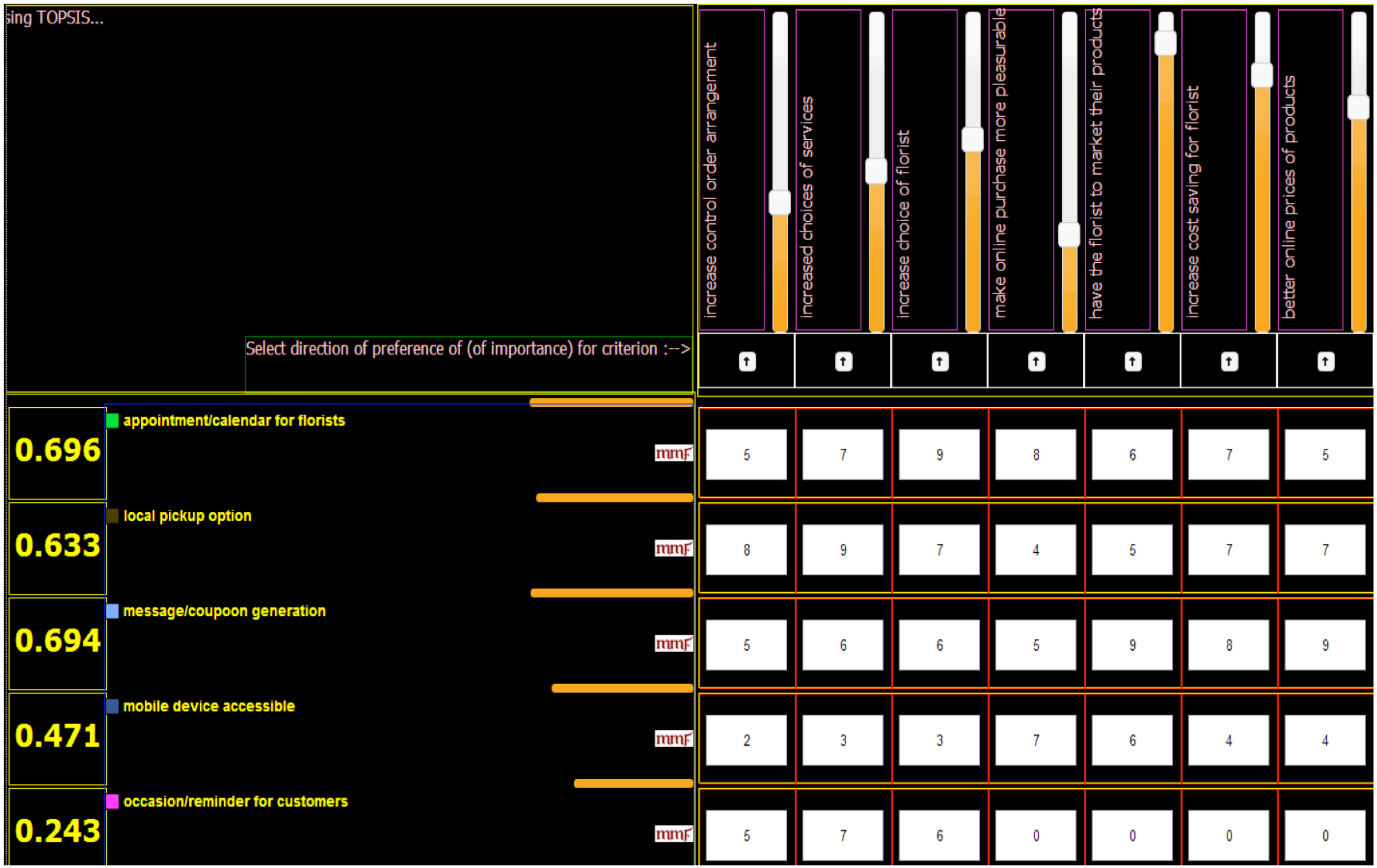
Tradespace and Affordability Framework

- **Affordability Improvements and Tradeoffs**
  - **Get the Best from People**
    - Staffing, Incentivizing, Teambuilding
    - Facilities, Support Services
    - Kaizen (continuous improvement)
  - **Make Tasks More Efficient**
    - Tools and Automation
    - Work and Oversight Streamlining
    - Collaboration Technology
  - **Eliminate Tasks**
    - Lean and Agile Methods
    - Task Automation
    - Model-Based Product Generation
  - **Eliminate Scrap, Rework**
    - Early Risk and Defect Elimination
    - Evidence-Based Decision Gates
    - Modularity Around Sources of Change
    - Incremental, Evolutionary Development
    - Value-Based, Agile Process Maturity
    - Risk-Based Prototyping
  - **Simplify Products (KISS)**
    - Value-Based Capability Prioritization
    - Satisficing vs. Optimizing Performance
  - **Reuse Components**
    - Domain Engineering and Architecture
    - Composable Components,Services, COTS
    - Legacy System Repurposing
  - **Reduce Operations, Support Costs**
    - Automate Operations Elements
    - Design for Maintainability, Evolvability
    - Streamline Supply Chain
    - Anticipate, Prepare for Change
  - **Value- and Architecture-Based Tradeoffs and Balancing**

Tradeoffs Among Cost, Schedule, and Reliability, and Functionality: COCOMO II



A Value-Priority Tradeoff Equalizer



1.1. Figure 1 TOPSIS Prioritization of the MMFs. The priority scores seen on extreme left and the goals along

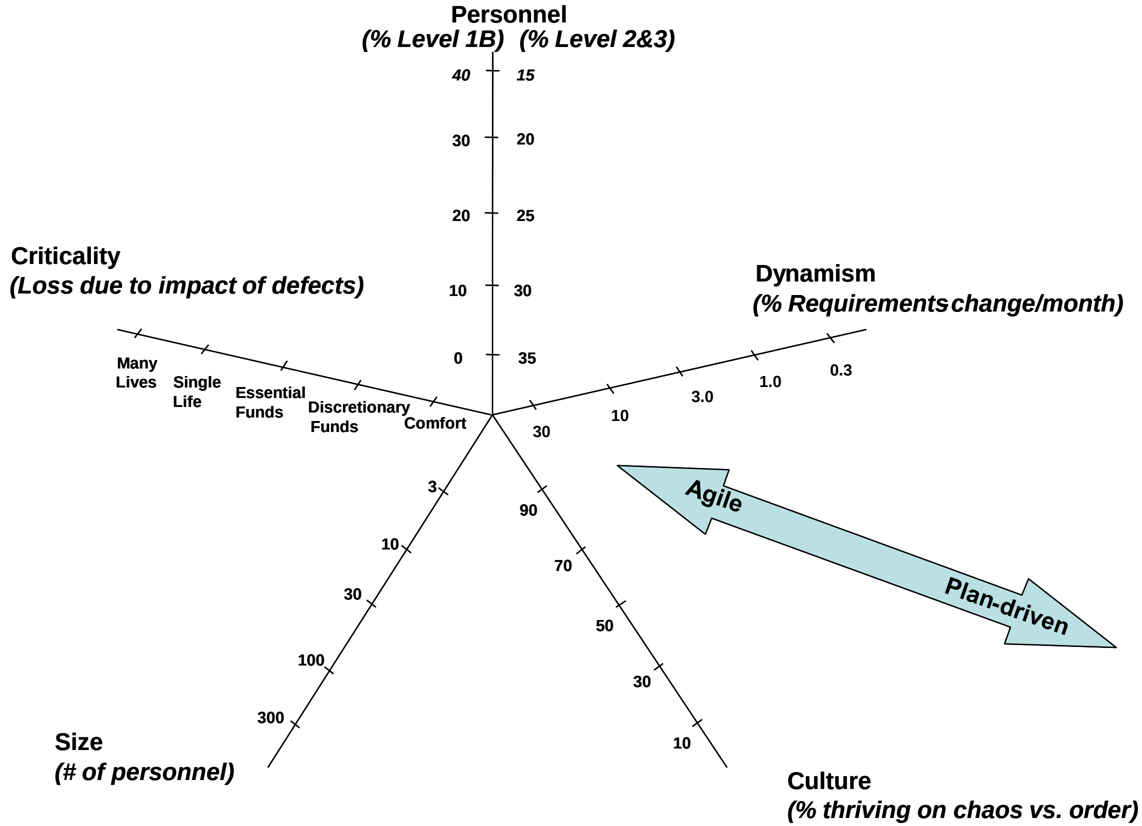
Conclusions

- Affordability increasingly competition-critical
  - Need to balance cost, schedule, performance, functionality
- Orthogonal framework helps tailor improvements
  - Getting the best from people
  - Making tasks more efficient
  - Eliminating tasks
  - Eliminating scrap and rework
  - Simplifying products
  - Reusing assets
  - Reducing operations and support costs
  - Value- and architecture-based tradeoffs and balancing
- No one-size-fits-all solution

Backup Charts

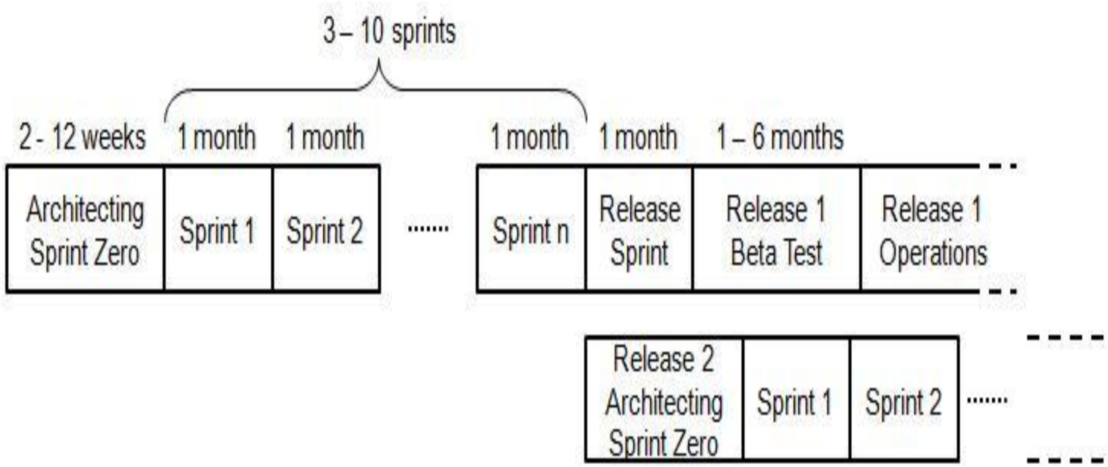
Agile and Plan-Driven Home Grounds: Five Critical Decision Factors

- **Size, Criticality, Dynamism, Personnel, Culture**

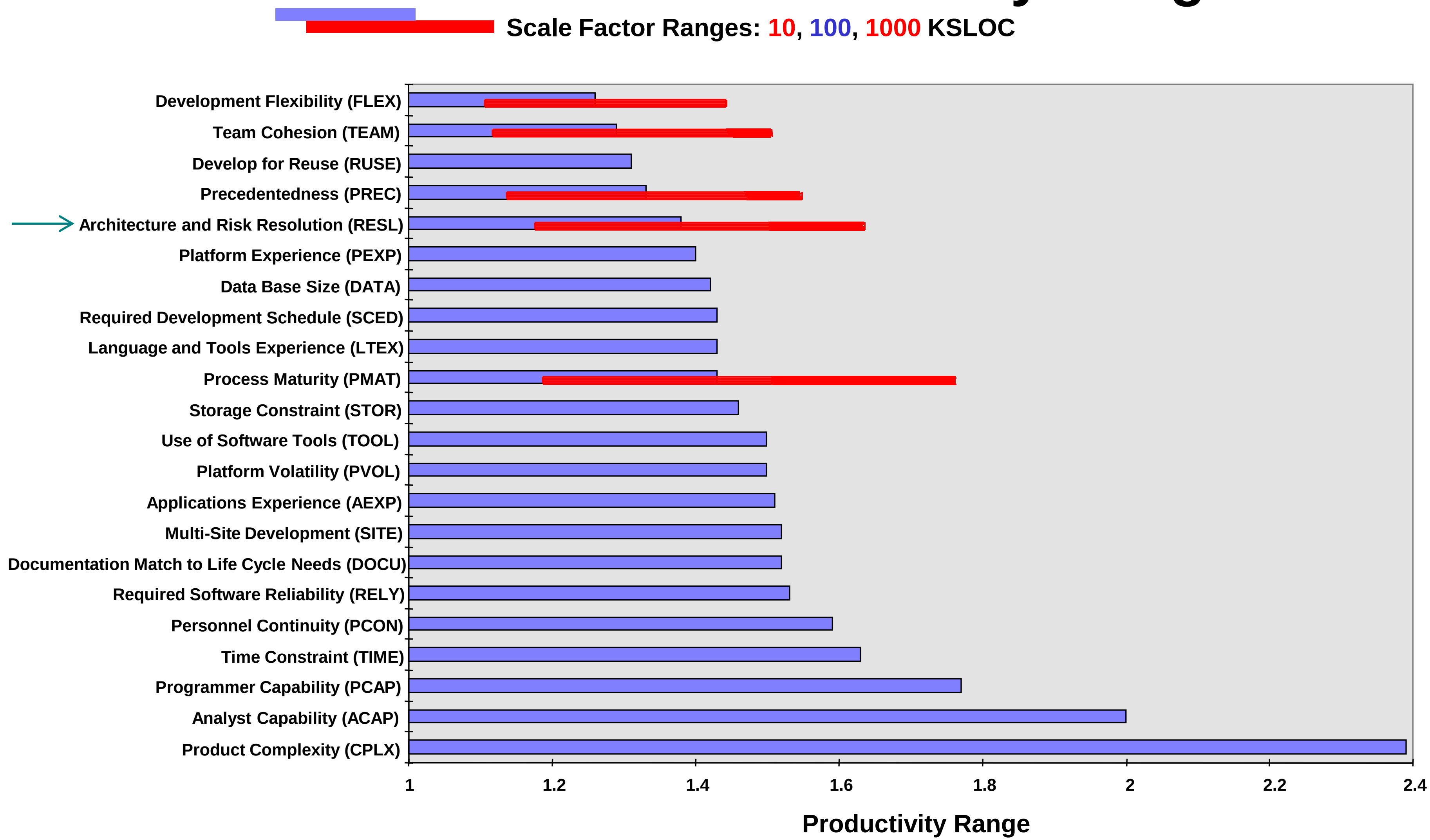


Architected Agile Approach

- Uses Scrum of Scrums approach
  - Up to 10 Scrum teams of 10 people each
  - Has worked for distributed international teams
  - Going to three levels generally infeasible
- General approach shown below
  - Often tailored to special circumstances

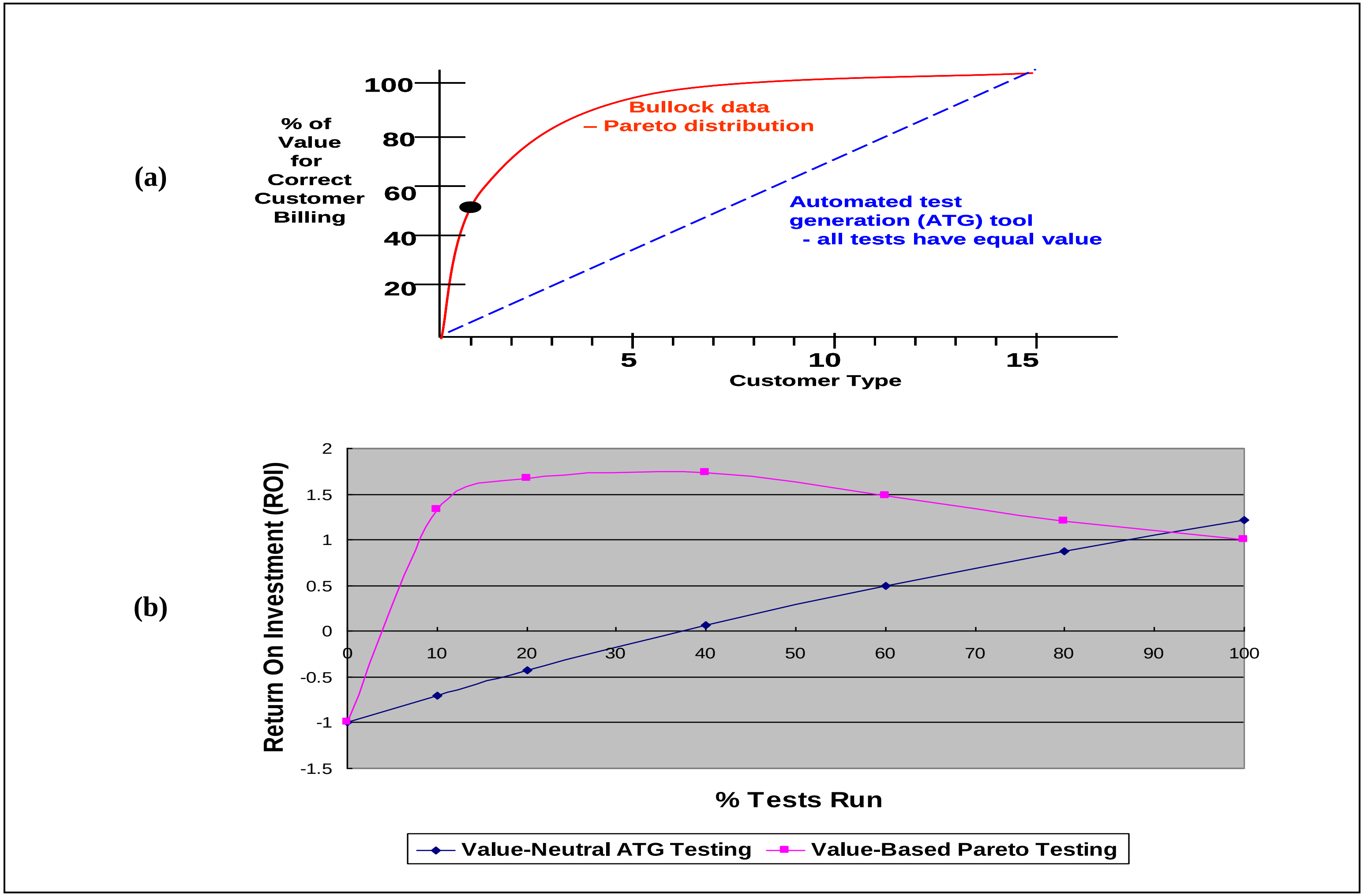


COCOMO II. 2000 Productivity Ranges

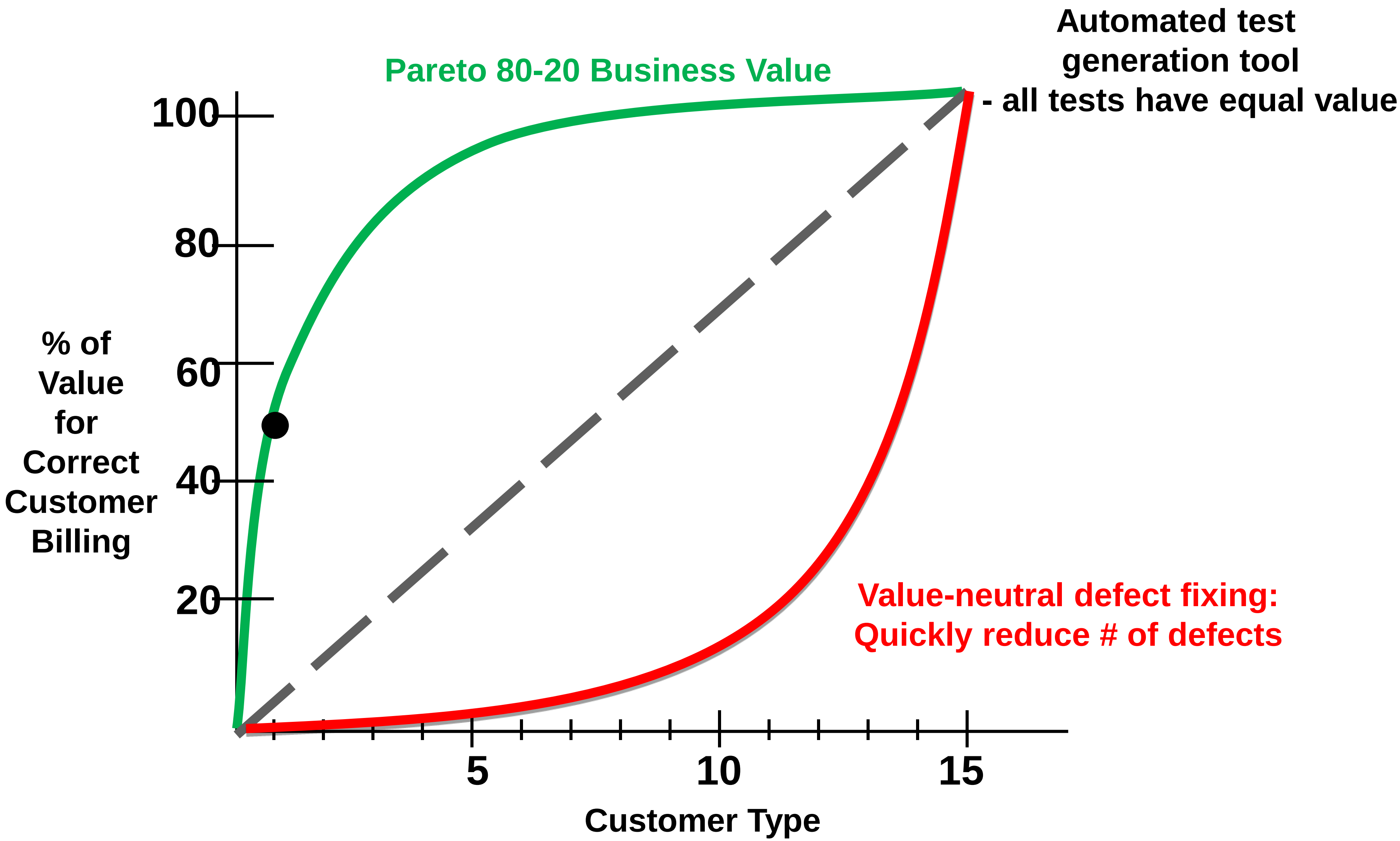


Value-Based Testing: Empirical Data and ROI

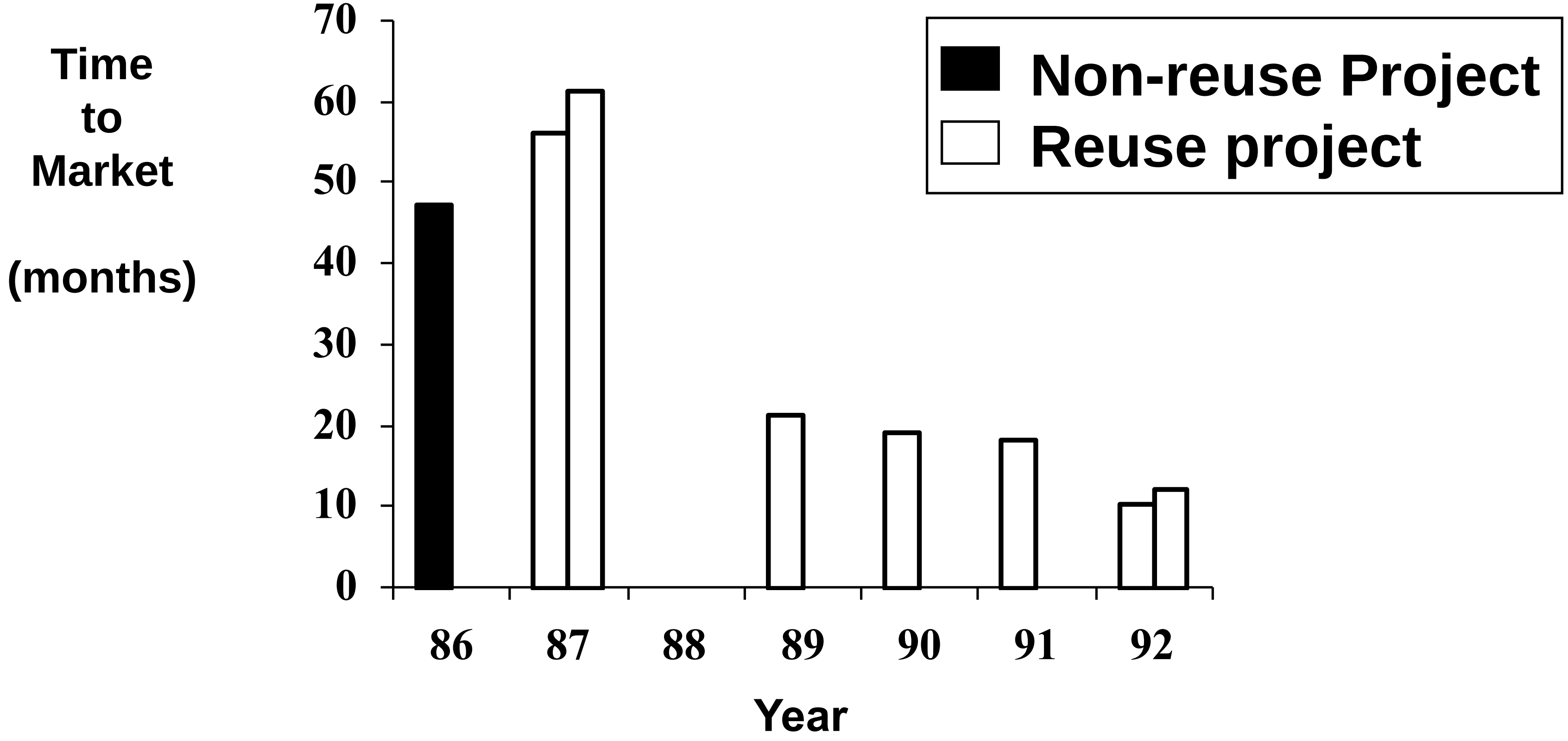
— LiGuo Huang, ISESE 2005



Value-Neutral Defect Fixing Is Even Worse



Reuse at HP’s Queensferry Telecommunication Division



Product Line Engineering and Management

SYSTEMS ENGINEERING Research Center

Preferences

Systems Product Line Flexibility Value Model

Welcome SERC Collaborator

Open | Save | Save As

System Costs

Average Product Development Cost (Burdened $M): 5
Ownership Time (Years): 3
Annual Change Cost (% of Development Cost): 10
Interest Rate (Annual %): 7

Product Line Percentages

Unique %: 40
Adapted %: 30
Reused %: 30

Relative Costs of Reuse (%)

Relative Cost of Reuse for Adapted: 40
Relative Cost of Reuse for Reused: 5

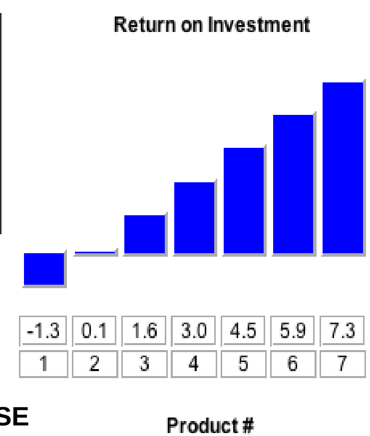
Investment Cost

Relative Cost of Developing for PL Flexibility via Reuse: 1.7

Calculate

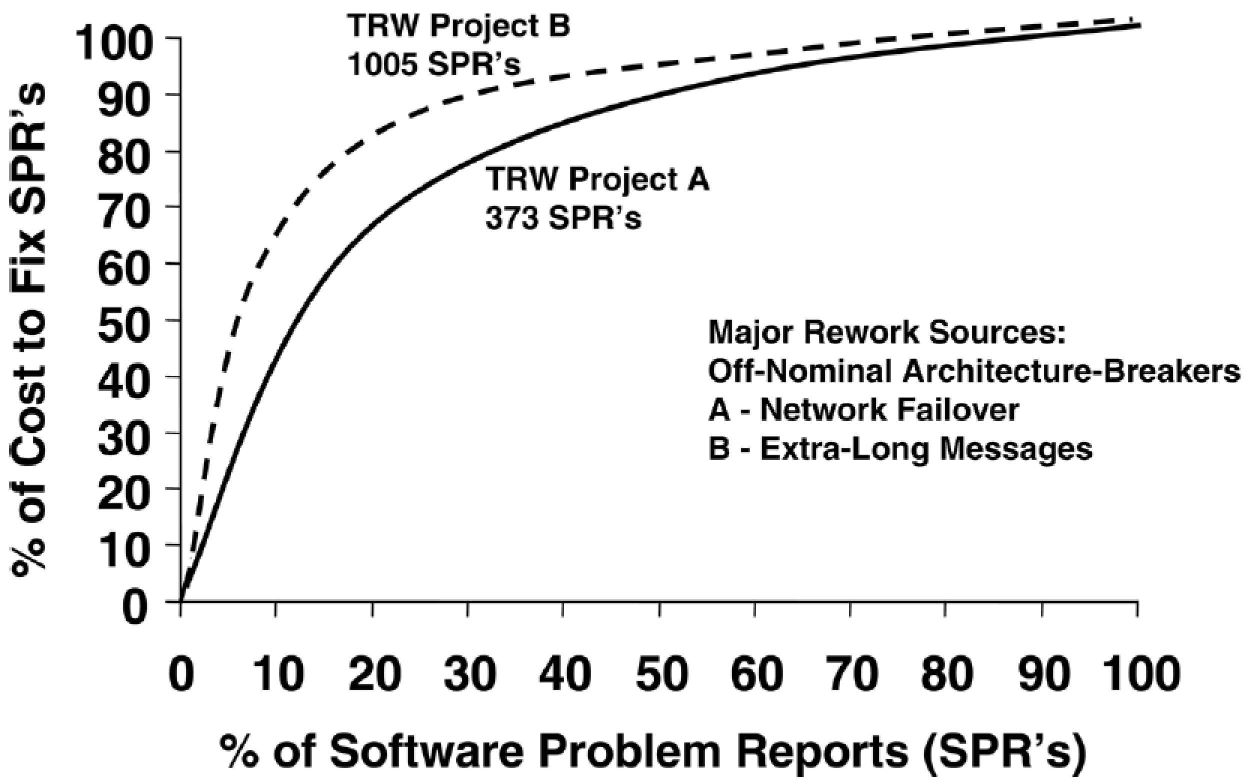
Results

# of Products	1	2	3	4	5	6	7
Development Cost ($M)	$7.1	$2.7	$2.7	$2.7	$2.7	$2.7	$2.7
Ownership Cost ($M)	$2.1	$0.8	$0.8	$0.8	$0.8	$0.8	$0.8
Cum. PL Cost ($M)	$9.2	$12.7	$16.2	$19.7	$23.1	$26.6	$30.1
PL Flexibility Investment ($M)	$2.1	$0	$0	$0	$0	$0	$0
PL Effort Savings	($2.7)	$0.3	$3.3	$6.3	$9.4	$12.4	$15.4
Return on Investment	-1.30	0.14	1.58	3.02	4.46	5.90	7.34



Overfocus on Acquisition Cost

C4ISR Contracts: Nominal-case requirements; 90 days to PDR



C4ISR Project C: Architecting for Change

USAF/ESC-TRW CCPDS-R Project*

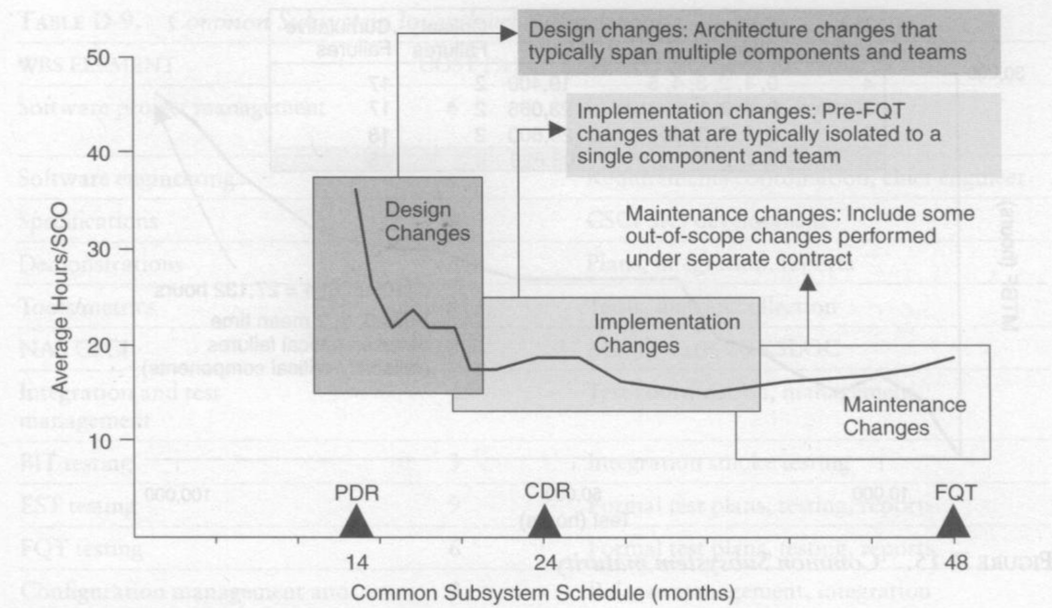
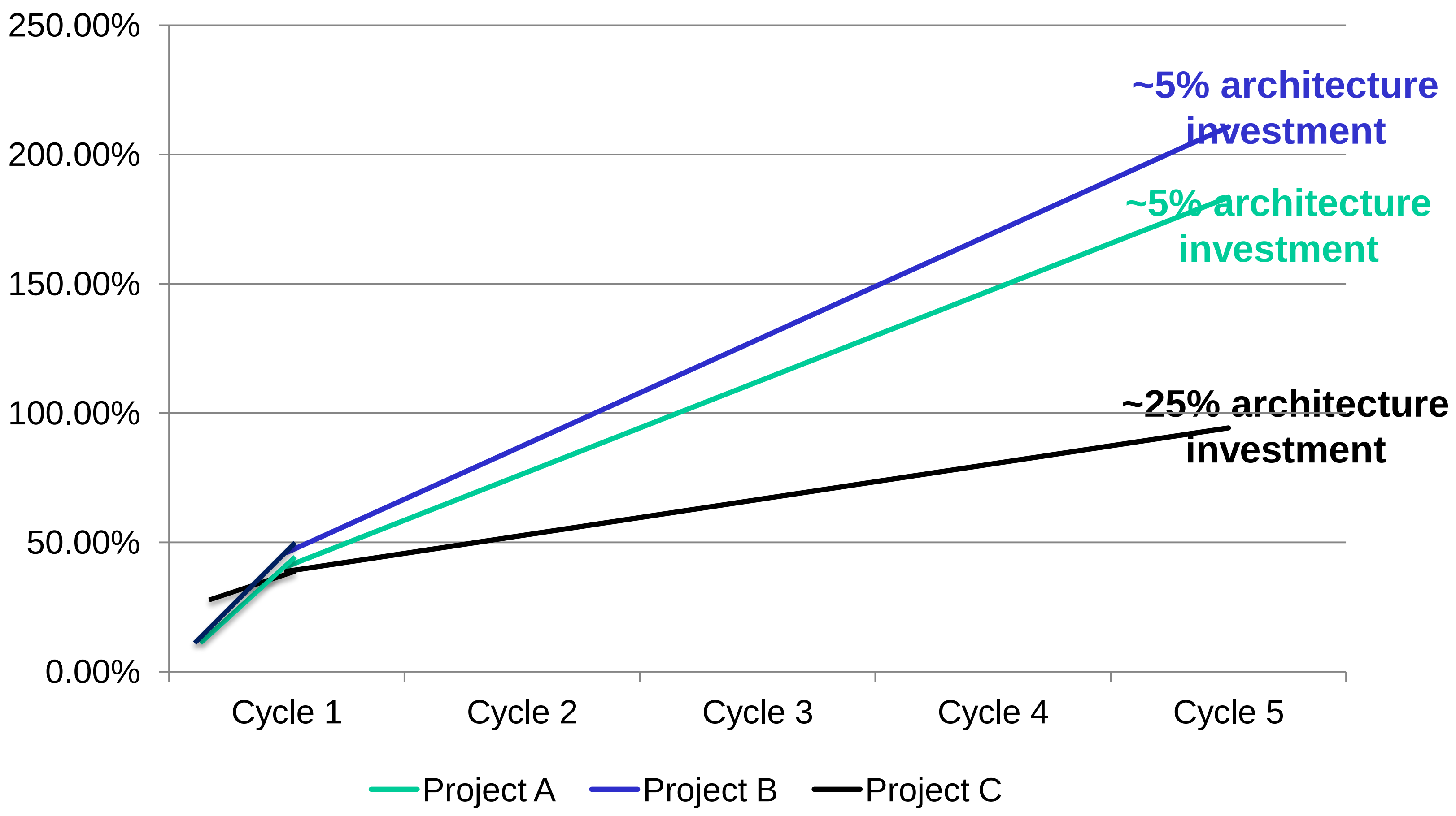


FIGURE D-14. *Common Subsystem adaptability*

When investments made in architecture, average time for change order becomes relatively stable over time…

*** Walker Royce, *Software Project Management: A Unified Framework*. Addison-Wesley, 1998.**

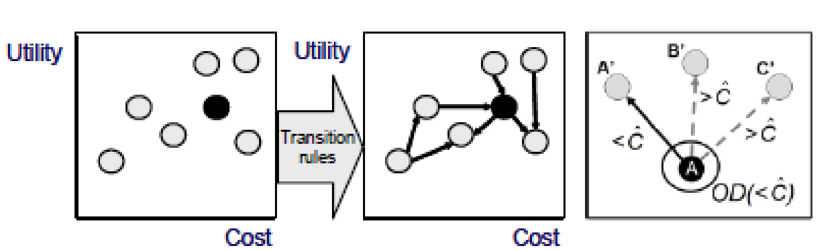
Relative* Total Ownership Cost (TOC)

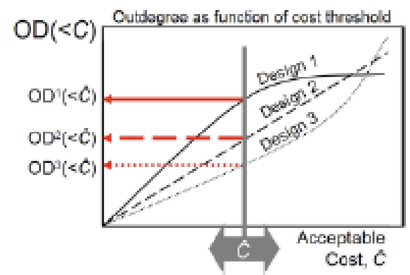


*** Cumulative architecting and rework effort relative to initial development effort**

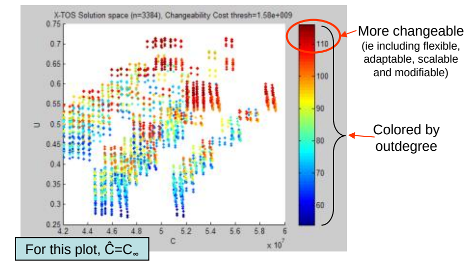
Ilities in Tradespace Exploration: MIT

Enabling Construct: Tradespace Networks

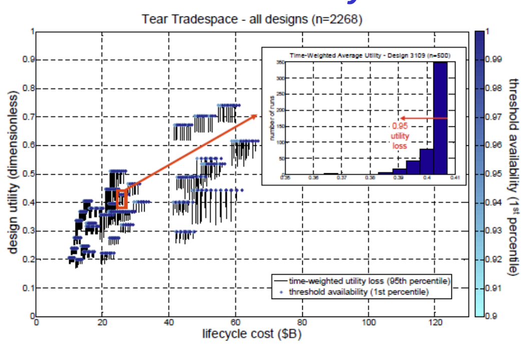




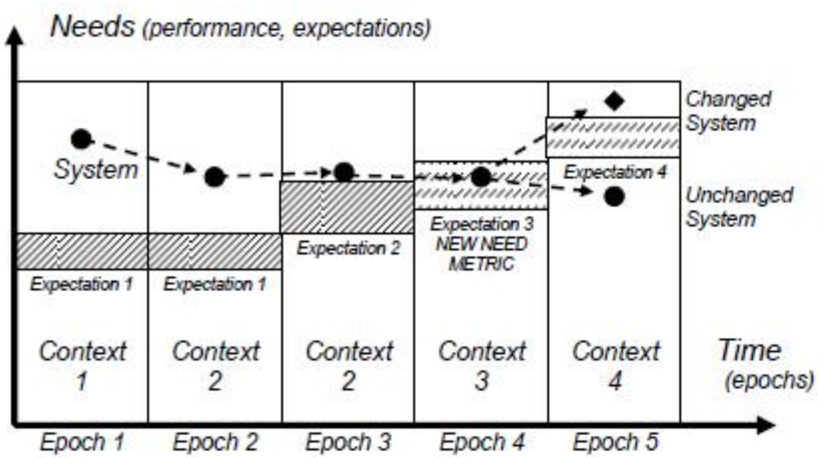
Changeability



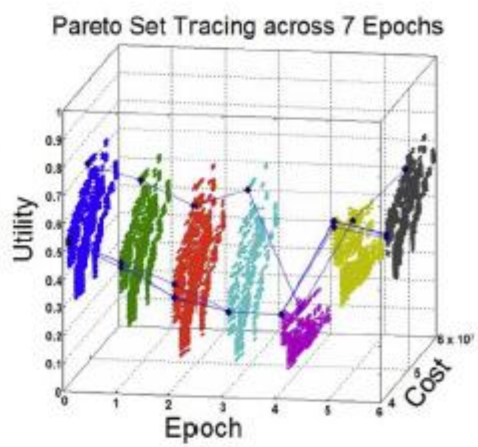
Survivability



Enabling Construct: Epochs and Eras



Value Robustness



Set of Metrics

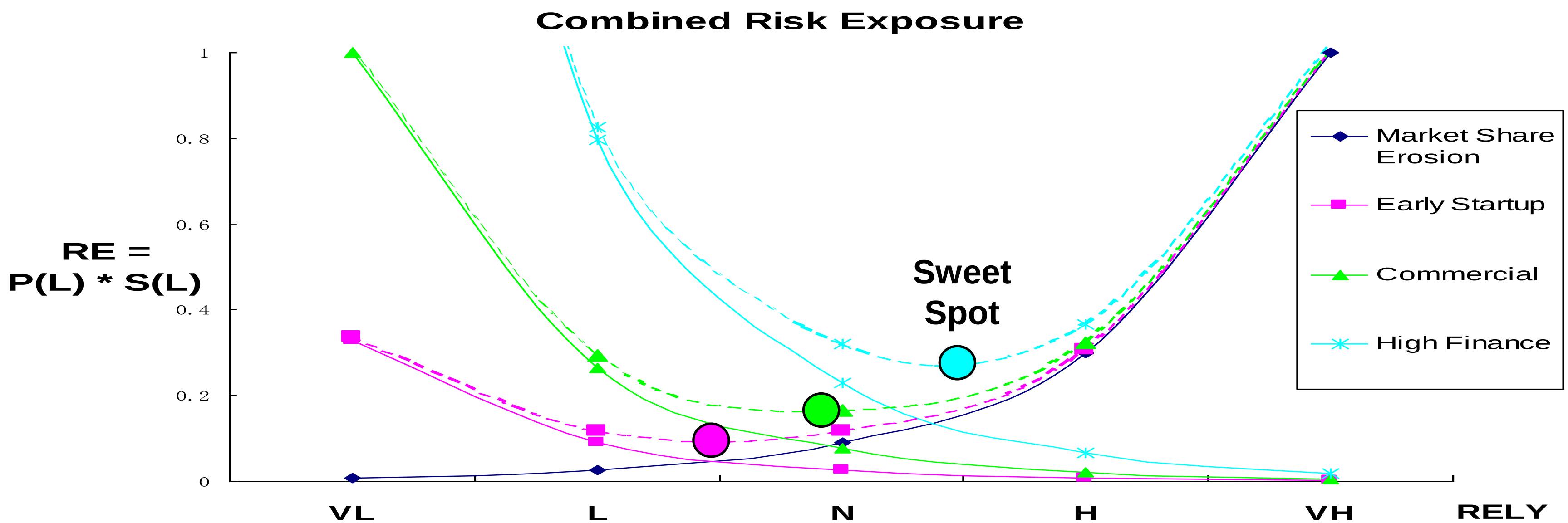
Value Aspect	Acronym	Stands For	Definition
Robustness via "no change"	NPT	Normalized Pareto Trace	% epochs for which design is Pareto efficient in utility/cost
Robustness via "no change"	fNPT	Fuzzy Normalized Pareto Trace	Above, with margin from Pareto front allowed
Robustness via "change"	eNPT, efNPT	Effective (Fuzzy) Normalized Pareto Trace	Above, considering the design's end state after transitioning
"Value" gap	FPN	Fuzzy Pareto Number	% margin needed to include design in the fuzzy Pareto front
"Value" of a change	FPS	Fuzzy Pareto Shift	Difference in FPN before and after transition
"Value" of a change	ARI	Available Rank Increase	# of designs able to be passed in utility via best possible change
Degree of changeability	OD	Outdegree	# outgoing transition arcs from a design
Degree of changeability	FOD	Filtered Outdegree	Above, considering only arcs below a chosen cost threshold
Survivability	TWAUL	Time-weighted Average Utility Loss	Measure of central tendency of value losses over time for a design, as a result of experienced disturbances
Survivability	AT	Threshold Availability	% of lifetime for which design delivers utility above minimum acceptable levels before, during, and after a disturbance

Architecture-Based Attribute Trades:

Flexibility Arch. Strategy	Synergies	Conflicts
High module cohesion; Low module coupling	Interoperability Reliability	High Performance via Tight coupling
Service-oriented architecture	Composability, Usability, Testability	High Performance via Tight coupling
Autonomous adaptive systems	Affordability via task automation; Response time	Excess autonomy reduces human Controllability
Modularization around sources of change	Interoperability, Usability, Reliability, Availability	Extra time on critical path of Rapid Fielding
Multi-layered architecture	Reliability, Availability	Lower Performance due to layer traversal overhead
Many built-in options, entry points	Functionality, Accessibility	Reduced Usability via options proliferation; harder to Secure
User programmability	Usability, Mission Effectiveness	Full programmability causes Reliability, Safety, Security risks
Spare/expandable capacity	Performance, Reliability	Added cost
Product line architecture, reusable components	Cost, Schedule, Reliability	Some loss of performance vs. optimized stovepipes

Value/Risk-Based Tradespace Analysis

- Early Startup: Risk due to low dependability
- Commercial: Risk due to low dependability
- High Finance: Risk due to low dependability
- Risk due to market share erosion



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

Magnitude of Overrun Problem: DoD

Analysis of U.S. Defense Dept. Major Defense Acquisition Program Portfolios

Fiscal 2009 dollars

Portfolio size	2003	2007	2008
Number of programs	77	95	96
Total planned commitments	$1.2 trillion	$1.6 trillion	$1.6 trillion
Commitments outstanding	$724.2 billion	$875.2 billion	$786.3 billion
Portfolio indicators			
Change to total RDT&E* costs from first estimate	37%	40%	42%
Change to total acquisition cost from first estimate	19%	26%	25%
Total acquisition cost growth	$183 billion	$301.3 billion	$296.4 billion
Share of programs with 25% increase in program acquisition unit cost growth	41%	44%	42%
Average schedule delay in delivering initial capabilities	18 months	21 months	22 months

Source: U.S. Government Accountability Office

**Research, Development, Testing & Evaluation*

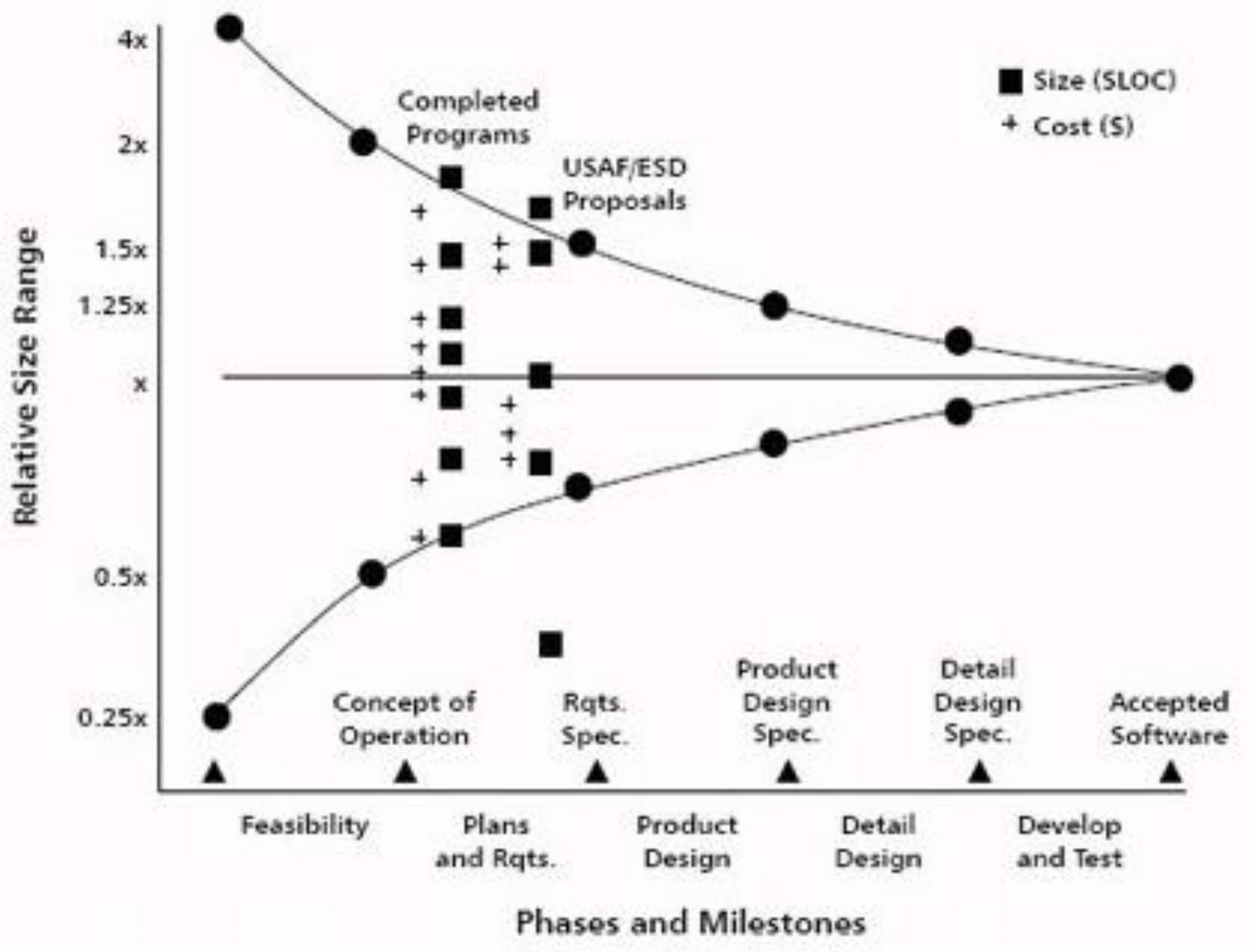
Magnitude of Overrun Problem:
Standish Surveys of Commercial Projects

Year	2000	2002	2004	2006	2008
Within budget and schedule	28	34	29	35	32
Prematurely cancelled	23	15	18	19	24
Budget or schedule overrun	49	51	53	46	44

Some Frequent Overrun Causes

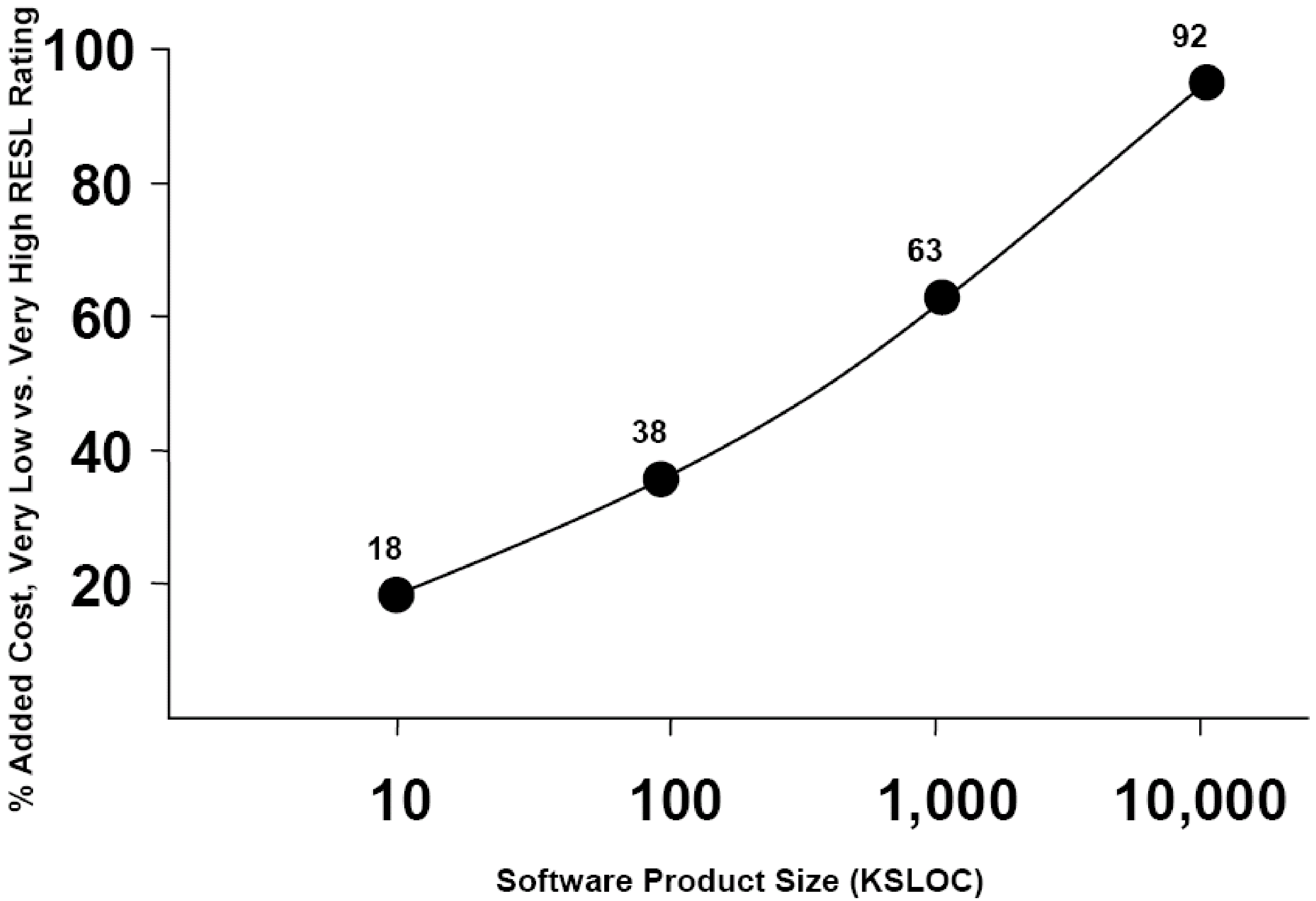
- Conspiracy of Optimism
- Effects of First Budget Shortfall
  - System Engineering
- Decoupling of Technical and Cost Analysis
  - Overfocus on Performance, Security, Functionality
- Overfocus on Acquisition Cost
- Assumption of Stability
- Total vs. Incremental Commitment

The Conspiracy of Optimism and The Cone of Uncertainty



Effects of First Budget Shortfall: Added Cost of Weak System Engineering

Calibration of COCOMO II Architecture and Risk Resolution (RESL) factor to 161 project data points



Assumption of Stability vs. Rapid Change

– Need evolutionary/incremental vs. one-shot development