An Initial Ontology for System Qualities

Barry Boehm, Nupul Kukreja, USC

INCOSE IS 2015
July 14, 2015

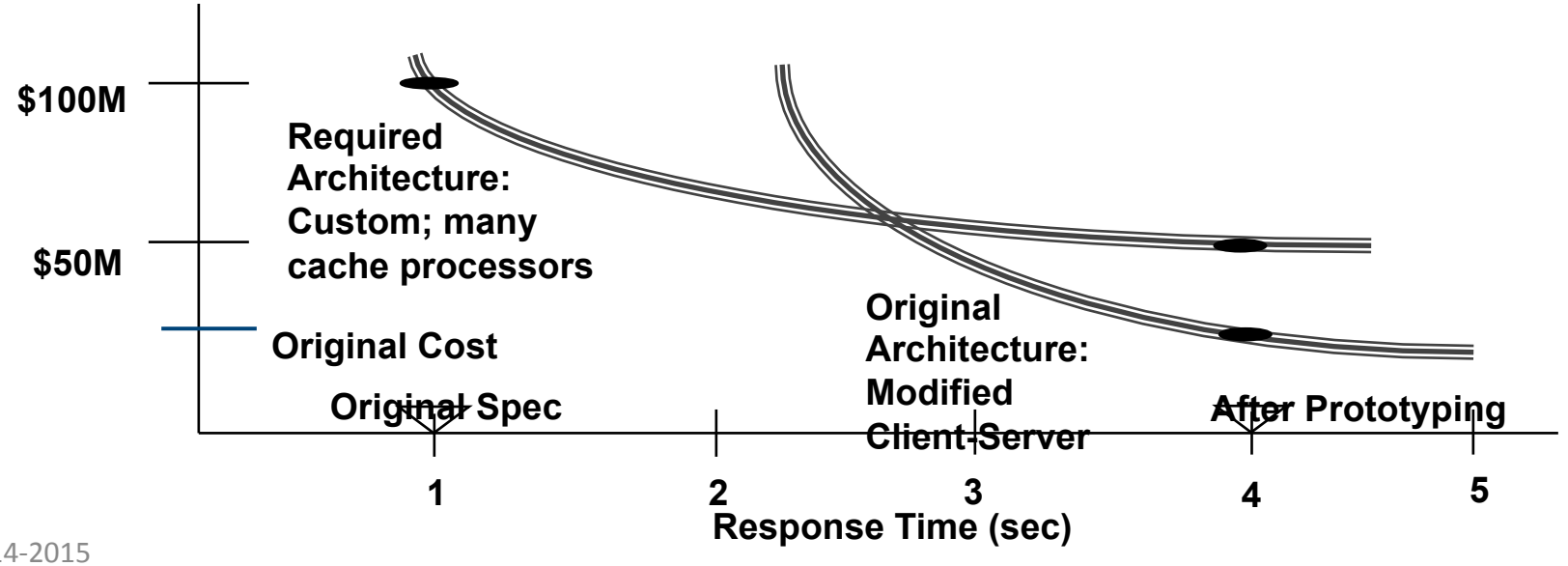
Outline

- **Critical nature of system qualities (SQs)**
  - **Or non-functional requirements (NFRs); ilities**
  - **Major source of project overruns, failures**
  - **Significant source of stakeholder value conflicts**
  - **Poorly defined, understood**
  - **Underemphasized in project management**
- **Need for and nature of SQs ontology**
  - **Nature of an ontology; choice of IDEF5 structure**
  - **Stakeholder value-based, means-ends hierarchy**
  - **Synergies and Conflicts matrix and expansions**
    - **Example means-ends hierarchy: Affordability**

Importance of SQ Tradeoffs

Major source of DoD, other system overruns

- SQs have systemwide impact
  - System elements generally just have local impact
- SQs often exhibit asymptotic behavior
  - Watch out for the knee of the curve
- Best architecture is a discontinuous function of SQ level
  - "Build it quickly, tune or fix it later" highly risky
  - Large system example below



Example of SQ Value Conflicts: Security IPT

- **Single-agent key distribution; single data copy**
  - **Reliability: single points of failure**
- **Elaborate multilayer defense**
  - **Performance: 50% overhead; real-time deadline problems**
- **Elaborate authentication**
  - **Usability: delays, delegation problems; GUI complexity**
- **Everything at highest level**
  - **Modifiability: overly complex changes, recertification**

Proliferation of Definitions: Resilience

- **Wikipedia Resilience variants: Climate, Ecology, Energy Development, Engineering and Construction, Network, Organizational, Psychological, Soil**

- **Ecology and Society Organization Resilience variants: Original-ecological, Extended-ecological, Walker et al. list, Folke et al. list; Systemic-heuristic, Operational, Sociological, Ecological-economic, Social-ecological system, Metaphoric, Sustainabilty-related**

- **Variants in resilience outcomes**
  - **Returning to original state; Restoring or improving original state; Maintaining same relationships among state variables; Maintaining desired services; Maintaining an acceptable level of service; Retaining essentially the same function, structure, and feedbacks; Absorbing disturbances; Coping with disturbances; Self-organizing; Learning and adaptation; Creating lasting value**
  - **Source of serious cross-discipline collaboration problems**

Example of Current Practice

- "The system shall have a Mean Time Between Failures of 10,000 hours"
- What is a "failure?"
  - 10,000 hours on liveness
  - But several dropped or garbled messages per hour?
- What is the operational context?
  - Base operations? Field operations? Conflict operations?
- Most management practices focused on functions
  - Requirements, design reviews; traceability matrices; work breakdown structures; data item descriptions; earned value management
- What are the effects of or on other SQs?
  - Cost, schedule, performance, maintainability?

Outline

- Critical nature of system qualities (SQs)
  - Or non-functional requirements; ilities
  - Major source of project overruns, failures
  - Significant source of stakeholder value conflicts
  - Poorly defined, understood
  - Underemphasized in project management
- ➡ Need for and nature of system SQs ontology
  - Nature of an ontology; choice of IDEF5 structure
  - Stakeholder value-based, means-ends hierarchy
  - Synergies and Conflicts matrix and expansions
    - Example means-ends hierarchy: Affordability

Need for SQs Ontology

- **Oversimplified one-size-fits all definitions**
  - **ISO/IEC 25010, Reliability: the degree to which a system , product, or component performs specified functions under specified conditions for a specified period of time**
  - **OK if specifications are precise, but increasingly "specified conditions" are informal, sunny-day user stories.**
    - **Satisfying just these will pass "ISO/IEC Reliability," even if system fails on rainy-day user stories**
  - **Need to reflect that different stakeholders rely on different capabilities (functions, performance, flexibility, etc.) at different times and in different environments**
- **Proliferation of definitions, as with Resilience**
- **Weak understanding of inter-SQ relationships**
  - **Security Synergies and Conflicts with other qualities**

Nature of an ontology; choice of IDEF5 structure

- An ontology for a collection of elements is a definition of what it means to be a member of the collection
- For "system qualities," this means that an SQ identifies an aspect of "how well" the system performs
  - The ontology also identifies the sources of variability in the value of "how well" the system performs
    - Functional requirements specify "what;" NFRs specify "how well"
- After investigating several ontology frameworks, the IDEF5 framework appeared to best address the nature and sources of variability of system SQs
  - Good fit so far

Initial SERC SQs Ontology

- **Modified version of IDEF5 ontology framework**
  - **Classes, Subclasses, and Individuals**
  - **Referents, States, Processes, and Relations**
- **Top classes cover stakeholder value propositions**
  - **Mission Effectiveness, Resource Utilization, Dependability, Flexibiity**
- **Subclasses identify means for achieving higher-class ends**
  - **Means-ends one-to-many for top classes**
  - **Ideally mutually exclusive and exhaustive, but some exceptions**
  - **Many-to-many for lower-level subclasses**
- **Referents, States, Processes, Relations cover SQ variation**
  - **Referents: Sources of variation by stakeholder value context:**
  - **States: Internal (beta-test); External (rural, temperate, sunny)**
  - **Processes: Operational scenarios (normal vs. crisis; experts vs. novices)**
  - **Relations: Impact of other SQs (security as above, synergies & conflicts)**

Referents: MIT 14-D Semantic Basis

Prescriptive Semantic Basis for Change-type Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	Phase	Agent	Impetus Change	System	Outcome Change	System	Valuable

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the design "parameter" with "destination(s)" in the "aspect" to have an "effect" to the outcome "parameter" with "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Impetus				Outcome				Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Destination	Aspect	Effect	Parameter	Destination	Aspect					
					"parameter"	"state"			"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
disturbance	circumstantia	pre-ops	internal	increase	level	one	form	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	same		any	any	any	any	any	any	any
			any	any				any								

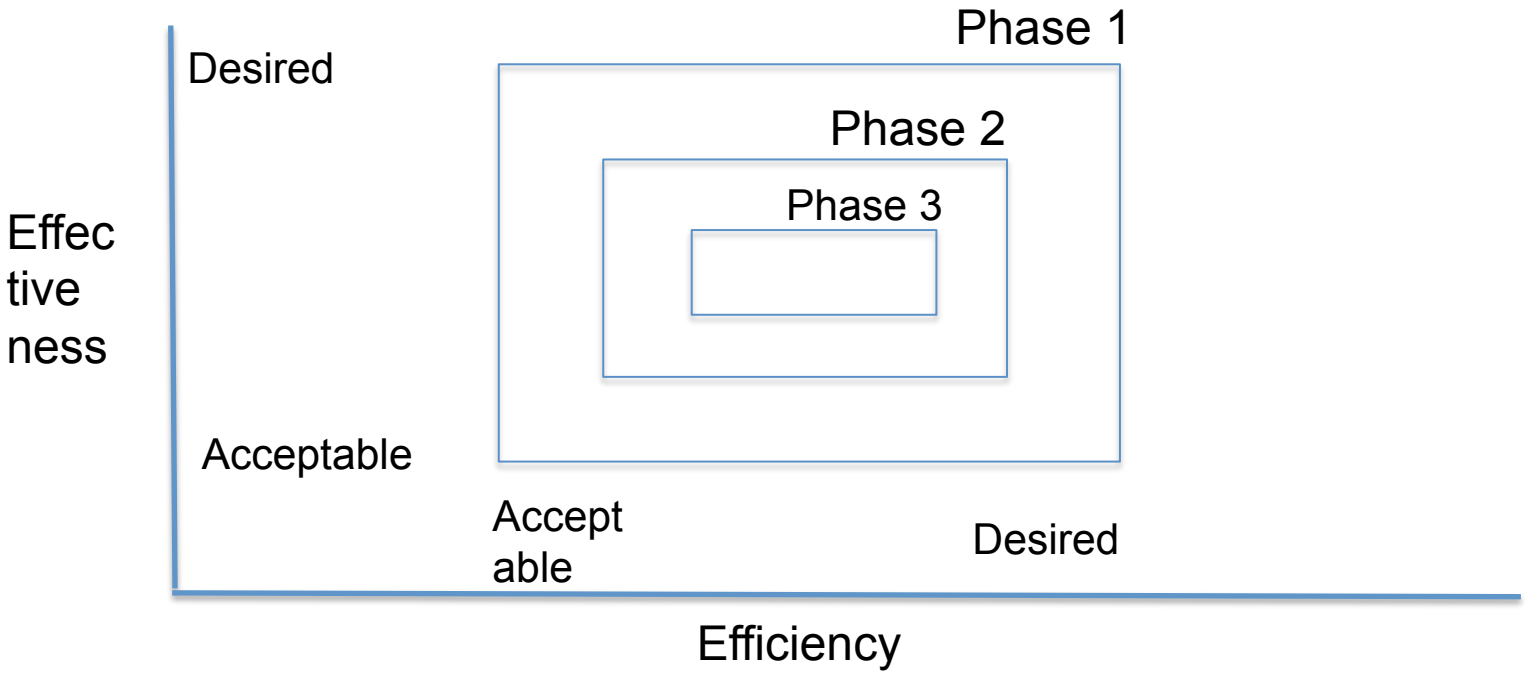
shift		ops						same	"Value"	few						
disturbance		ops						same	"Value"	few						
shift		ops						same		few						
shift		ops		not-same				same		few						
shift		ops		same		few		same		few						
shift		ops	none	same		few		same	level	few	form	system				
disturbance		ops						same		few						
			either	not-same				not-same								
shift	general	inter-LC		not-same				not-same				architecture				
			internal	not-same				not-same								
			external	not-same				not-same								
				not-same				not-same	level							
				not-same				not-same	set							
		ops	either	not-same				increase	set							
				not-same				not-same	any					shorter		
				not-same				not-same	any				sooner			
		ops		same	"Element set"	one	form	not-same	"Link set"		form					
		ops		same	"Element set"	one	operations	not-same	"Order set"		operations					
		ops		same		one	form/ops	not-same	set	few/many						
		ops		same		one	form/ops	not-same	set	few/many	function					
		ops		same		one	form/fnct	not-same	set	few/many	operations					
		ops		same		one	fnct/ops	not-same	set	few/many	form					

Ility Label
Value Robustness
Value Survivability
Robustness
Active Robustness
Passive Robustness
Classical Passive Robustness
Survivability
Changeability
Evolvability
Adaptability
Flexibility
Scalability
Modifiability
Extensibility
Agility
Reactivity
Form Reconfigurability
Operational Reconfigurability
Versatility
Functional Versatility
Operational Versatility
Substitutability

Example: Reliability Revisited

- Reliability is the probability that the system will deliver stakeholder-satisfactory results for a given time period (generally an hour), given specified ranges of:
  - Stakeholders: desired and acceptable ranges of liveness, accuracy, response time, speed, capabilities, etc.
  - System internal and external states: integration test, acceptance test, field test, etc.; weather, terrain, DEFCON, takeoff/flight/landing, etc.
  - System internal and external processes: security thresholds, types of payload/cargo; workload volume, diversity
  - Effects of other SQs: synergies, conflicts

Set-Based SQs Definition Convergence



Phase 1. Rough ConOps, Rqts, Solution Understanding

Phase 2. Improved ConOps, Rqts, Solution Understanding

Phase 3. Good ConOps, Rqts, Solution Understanding

Stakeholder value-based, means-ends hierarchy

- **Mission operators and managers want improved Mission Effectiveness**
  - **Involves Physical Capability, Cyber Capability, Human Usability, Speed, Accuracy, Impact, Endurability, Maneuverability, Scalability, Versatility, Interoperability**
- **Mission investors and system owners want Mission Cost-Effectiveness**
  - **Involves Cost, Duration, Personnel, Scarce Quantities (capacity, weight, energy, …); Manufacturability, Sustainability**
- **All want system Dependability: cost-effective defect-freedom, availability, and safety and security for the communities that they serve**
  - **Involves Reliability, Availablilty, Maintainability, Survivability, Safety, Security, Robustness**
- **In an increasingly dynamic world, all want system Flexibility: to be rapidly and cost-effectively changeable**
  - **Involves Modifiability, Tailorability, Adaptability**

U. Virginia: Coq Formal Reasoning Structure

- Inductive Dependable (s: System): Prop :=
- mk_dependability: Security s -> Safety s -> Reliability s ->
- Maintainability s -> Availability s -> Survivability s ->
- Robustness s -> Dependable s.

- Example aSystemisDependable: Dependable aSystem.
- apply mk_dependability.
- exact (is_secure aSystem).
- exact (is_safe aSystem).
- exact (is_reliable aSystem).
- exact (is_maintainable aSystem).
- exact (is_avaliable aSystem).
- exact (is_survivable aSystem).
- exact (is_robust aSystem).
- Qed.

Outline

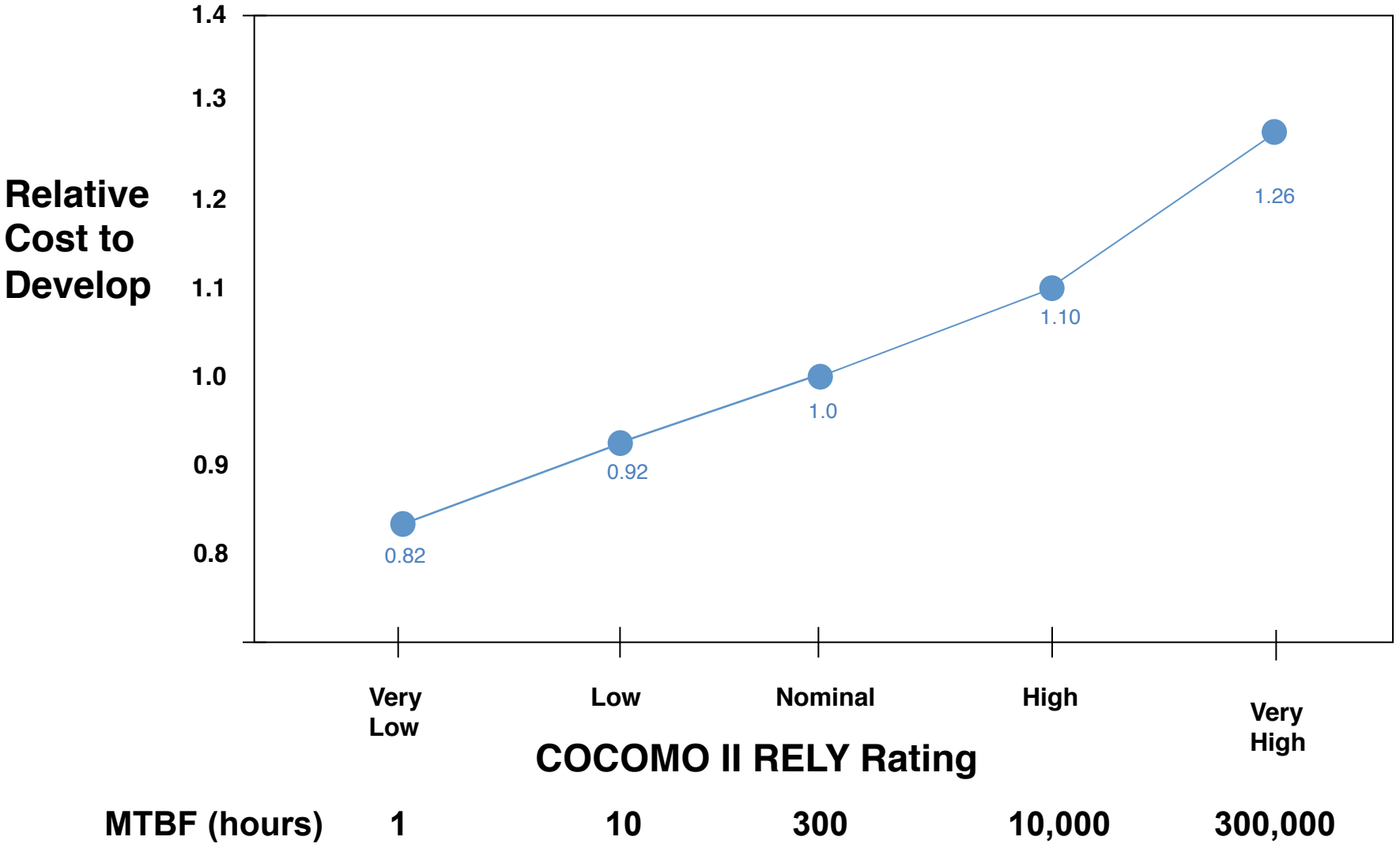
- Critical nature of system qualities (SQs)
  - Or non-functional requirements; ilities
  - Major source of project overruns, failures
  - Significant source of stakeholder value conflicts
  - Poorly defined, understood
  - Underemphasized in project management
- Need for and nature of SQs ontology
  - Nature of an ontology; choice of IDEF5 structure
  - Stakeholder value-based, means-ends hierarchy
  - ➡ Synergies and Conflicts matrix and expansions
    - Example means-ends hierarchy: Affordability

7x7 Synergies and Conflicts Matrix

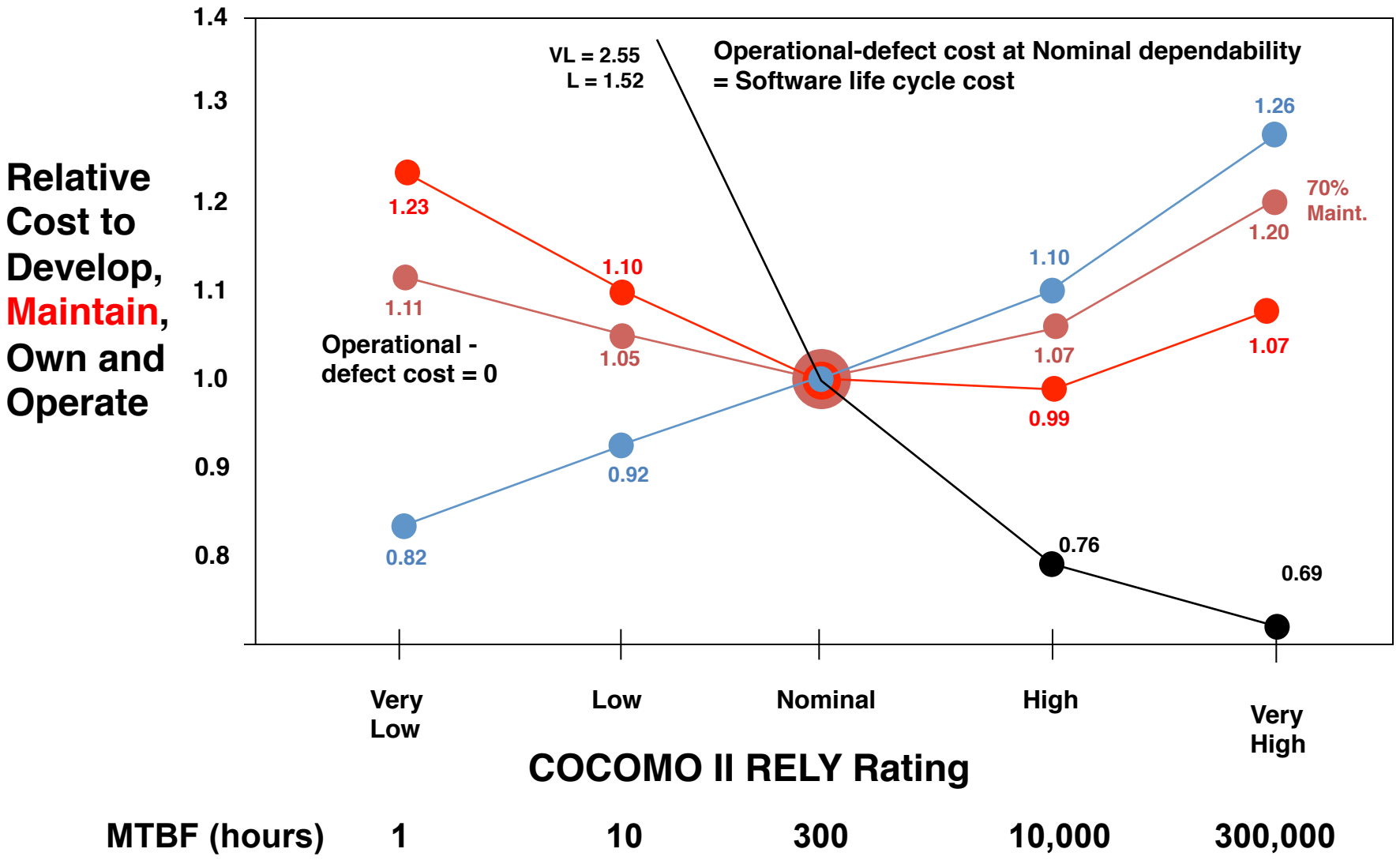
- **Mission Effectiveness expanded to 4 elements**
  - **Physical Capability, Cyber Capability, Interoperability, Other Mission Effectiveness (including Usability as Human Capability)**
- **Synergies and Conflicts among the 7 resulting elements identified in 7x7 matrix**
  - **Synergies above main diagonal, Conflicts below**
- **Work-in-progress tool will enable clicking on an entry and obtaining details about the synergy or conflict**
  - **Ideally quantitative; some examples next**
- **Still need synergies and conflicts within elements**
  - **Example 3x3 Dependability subset provided**

	Flexibility	Dependability	Mission Effectivenss	Resource Utilization	Physical Capability	Cyber Capability	Interoperability
Flexibility		Domain architecting within domain Modularity Self Adaptive Smart monitoring Spare Capacity Use software vs. hardware	Adaptability Many options Service oriented Spare capacity User programmability Versatility	Adaptability Agile methods Automated I/O validation Loose coupling for sustainability Product line architectures Staffing, Empowering	Adaptability Spare capacity	Adaptability Spare capacity	Adaptability Loose coupling Modularity Product line architectures Service-oriented connectors Use software vs. Hardware User programmability
Dependability	Accreditation Agile methods assurance Encryption Many options Multi-domain modifiability Multi-level security Self Adaptive defects User programmability		Accreditation FMEA Multi-level security Survivability Spare capacity	Automated aids Automated I/O validation Domain architecting within domain Product line architectures Staffing, Empowering **Total Ownership Cost** Value prioritizing	Fallbacks Lightweight agility Redundancy Spare capacity Value prioritizing	Fallbacks Redundancy Value prioritizing	Assertion Checking Domain architecting within domain Service oriented
Mission Effectivenss	Autonomy vs. Usability Modularity slowdowns Multi-domain architecture interoperability conflicts Versatility vs. Usability	Anti-tamper Armor vs. Weight Easiest-first development Redundancy Scalability Spare Capacity Usability vs. Security		Automated aids Domain architecting within domain Staffing, Empowering Value prioritizing	Automated aids Domain architecting within domain Staffing, Empowering Value prioritizing	Automated aids Domain architecting within domain Staffing, Empowering Value prioritizing	Automated aids Domain architecting within domain Staffing, Empowering
Resource Utilization	Agile Methods scalability Assertion checking overhead Fixed cost contracts Modularity Multi-domain architecture interoperability conflicts Spare capacity Tight coupling Use software vs. hardware	Accreditation **Acquisition Cost** Certification Easiest-first development Fallbacks Multi-domain architecture interoperability conflicts Redundancy Spare Capacity, tools costs Usability vs. Cost savings	Agile methods scalability Cost of automated aids Many options Multi-domain architecture interoperability conflicts Spare capacity Usability vs. Cost savings Versatility		Automated aids Domain architecting within domain Staffing, Empowering Value prioritizing	Automated aids Domain architecting within domain Staffing, Empowering Value prioritizing	Automated aids Domain architecting within domain Rework cost savings Staffing, Empowering
Physical Capability	Multi-domain architecture interoperability conflicts Over-optimizing Tight coupling Use software vs. hardware	Lightweight agility Multi-domain architecture interoperability conflicts Over-optimizing	Multi-domain architecture interoperability conflicts Over-optimizing	Cost of automated aids Multi-domain architecture interoperability conflicts Over-optimizing		Automated aids Staffing, Empowering Value prioritizing	Automated aids Domain architecting within domain
Cyber Capability	Agile Methods scalability Multi-domain architecture interoperability conflicts Over-optimizing Tight coupling Use software vs. hardware	Multi-domain architecture interoperability conflicts Over-optimizing	Multi-domain architecture interoperability conflicts Over-optimizing	Cost of automated aids Multi-domain architecture interoperability conflicts Over-optimizing	Over-optimizing Physical architecture or cyber architecture		Automated aids Domain architecting within domain
Interoperability	Multi-domain architecture interoperability conflicts User-programmed interoperability	Encryption interoperability Multi-domain architecture interoperability conflicts	Multi-domain architecture interoperability conflicts	Assertion checking Cost, duration of added connectors	Over-optimizing Tight vs. Loose coupling	Reduced speed of Assertion checking Reduced speed of connectors, standards compliance Tight vs. Loose coupling	

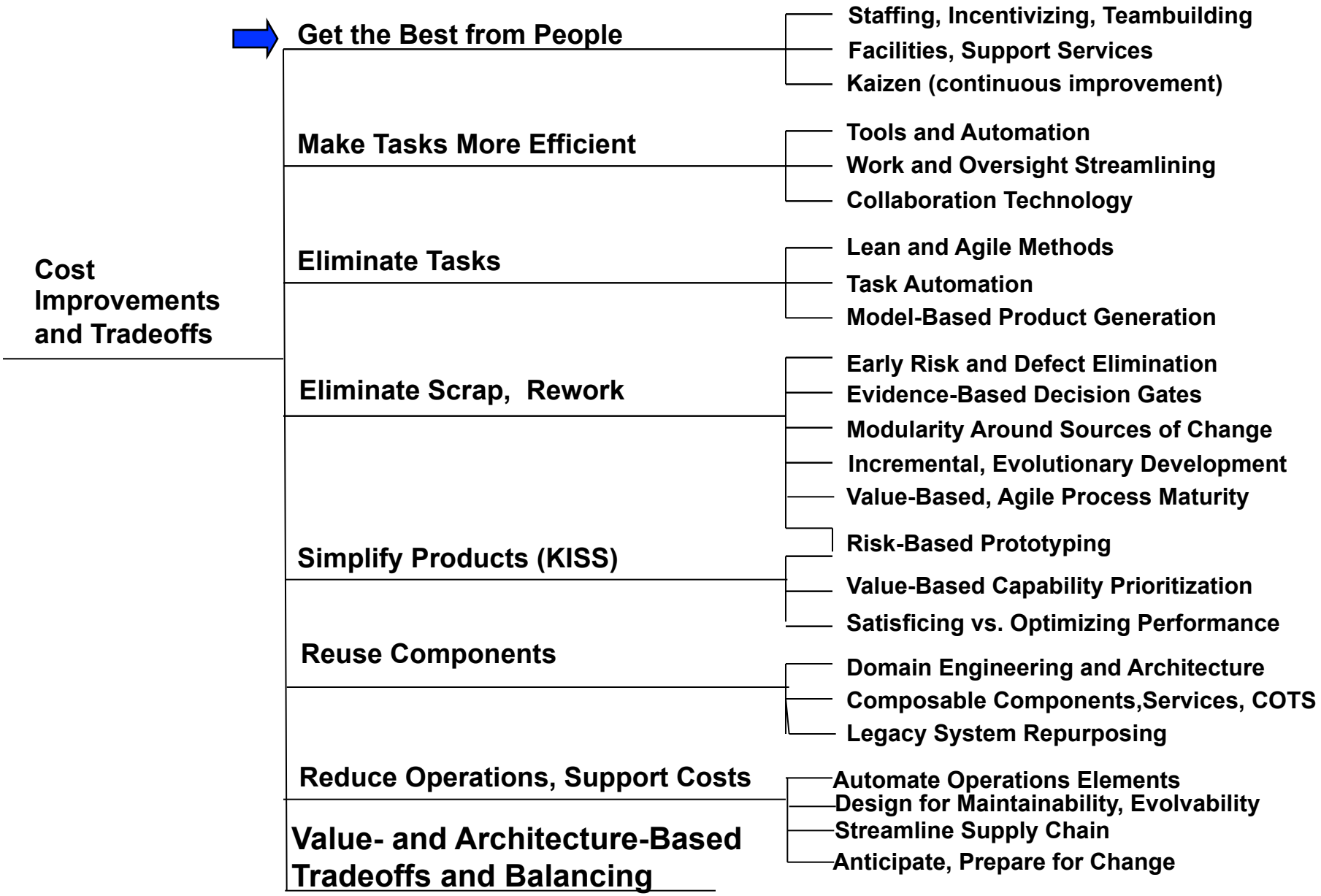
Software Development Cost vs. Reliability



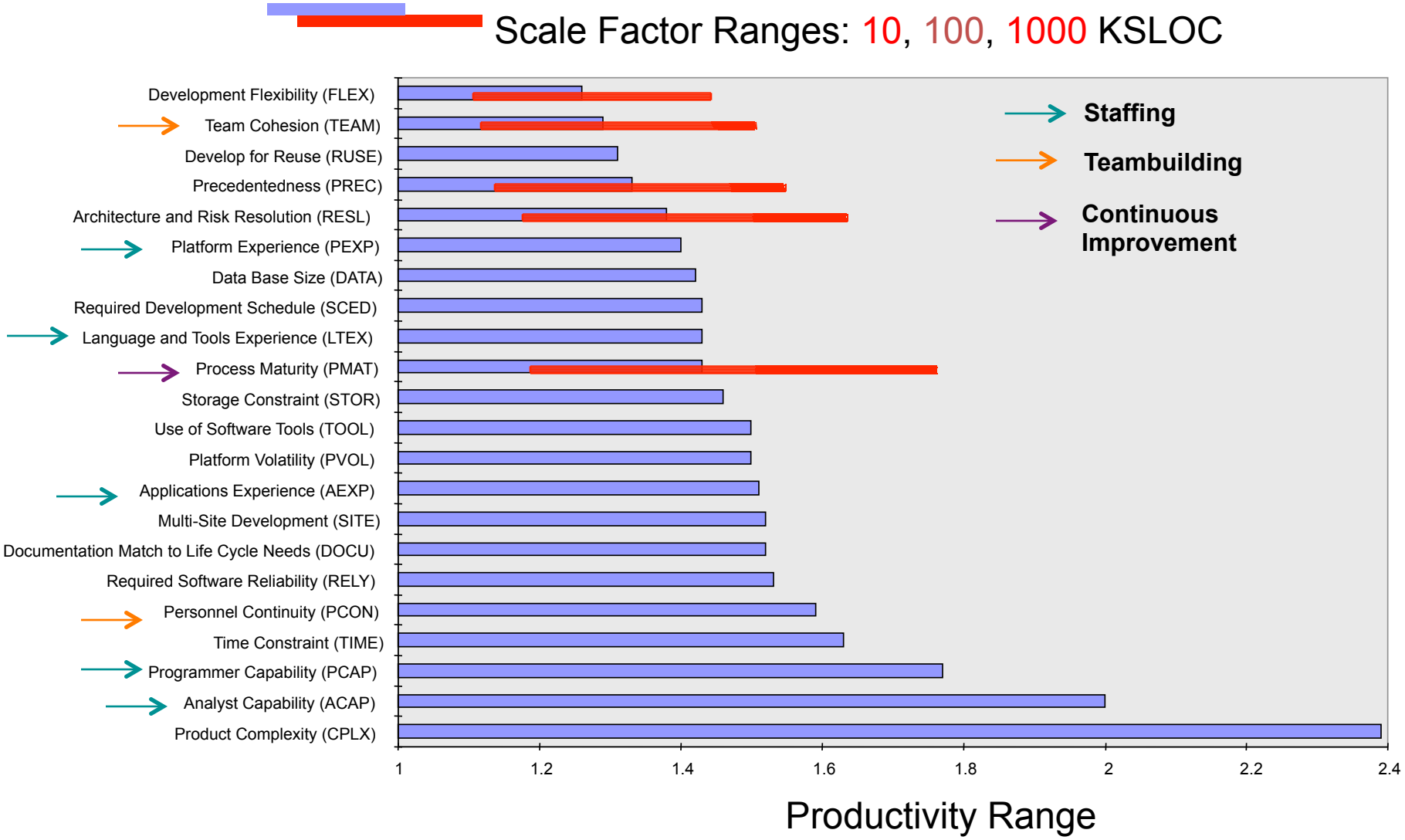
Software Ownership Cost vs. Reliability



Affordability and Tradespace Framework



Costing Insights: COCOMO II Productivity Ranges



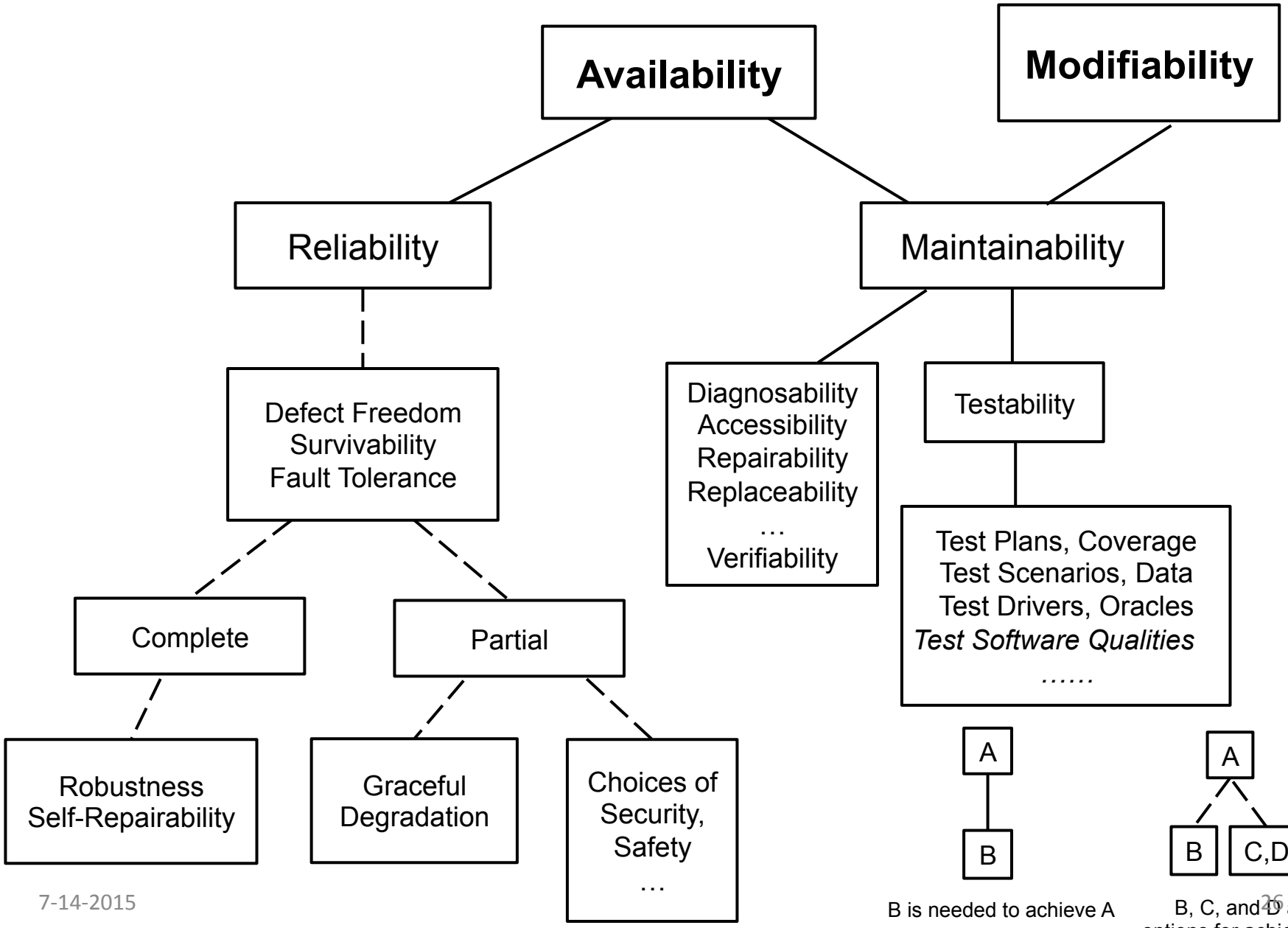
Conclusions

- System qualities (SQs) are success-critical
  - Major source of project overruns, failures
  - Significant source of stakeholder value conflicts
  - Poorly defined, understood
  - Underemphasized in project management
- SQs ontology clarifies nature of system qualities
  - Using value-based, means-ends hierarchy
  - Identifies variation types: referents, states, processes, relations
  - Relations enable SQ synergies and conflicts identification
- Continuing SERC research creating tools, formal definitions

Backup charts

	Flexibility	Dependability	Mission Effectivenss	Resource Utilization	Physical Capability	Cyber Capability	Interoperability
Flexibility		Domain architecting within domain	Adaptability	Adaptability	Adaptability	Adaptability	Adaptability
		Modularity	Many options	Agile methods	Spare capacity	Spare capacity	Loose coupling
		Self Adaptive	Service oriented	Automated I/O validation			Modularity
		Smart monitoring	Spare capacity	Loose coupling for sustainability			Product line architectures
		Spare Capacity	User programmability	Product line architectures			Service-oriented connectors
		Use software vs. hardware	Versatility	Staffing, Empowering			Use software vs. Hardware
							User programmability
Dependability	Accreditation		Accreditation	Automated aids	Fallbacks	Fallbacks	Assertion Checking
	Agile methods assurance		FMEA	Automated I/O validation	Lightweight agility	Redundancy	Domain architecting within domain
	Encryption		Multi-level security	Domain architecting within domain	Redundancy	Value prioritizing	Service oriented
	Many options		Survivability	Product line architectures	Spare capacity		
	Multi-domain modifiability		Spare capacity	Staffing, Empowering	Value prioritizing		
	Multi-level security			**Total Ownership Cost**			
	Self Adaptive defects			Value prioritizing			
	User programmability						
Mission Effectivenss	Autonomy vs. Usability	Anti-tamper		Automated aids	Automated aids	Automated aids	Automated aids
	Modularity slowdowns	Armor vs. Weight		Domain architecting within domain	Domain architecting within domain	Domain architecting within domain	Domain architecting within domain
	Multi-domain architecture interoperability conflicts	Easiest-first development		Staffing, Empowering	Staffing, Empowering	Staffing, Empowering	Staffing, Empowering
	Versatility vs. Usability	Redundancy		Value prioritizing	Value prioritizing	Value prioritizing	
		Scalability					
		Spare Capacity					
		Usability vs. Security					
Resource Utilization	Agile Methods scalability	Accreditation	Agile methods scalability		Automated aids	Automated aids	Automated aids
	Assertion checking overhead	**Acquisition Cost**	Cost of automated aids		Domain architecting within domain	Domain architecting within domain	Domain architecting within domain
	Fixed cost contracts	Certification	Many options		Staffing, Empowering	Staffing, Empowering	Rework cost savings
	Modularity	Easiest-first development	Multi-domain architecture interoperability conflicts		Value prioritizing	Value prioritizing	Staffing, Empowering
	Multi-domain architecture interoperability conflicts	Fallbacks	Spare capacity				
	Spare capacity	Multi-domain architecture interoperability conflicts	Usability vs. Cost savings				
	Tight coupling	Redundancy	Versatility				
	Use software vs. hardware	Spare Capacity, tools costs					
		Usability vs. Cost savings					
Physical Capability	Multi-domain architecture interoperability conflicts	Lightweight agility	Multi-domain architecture interoperability conflicts	Cost of automated aids		Automated aids	Automated aids
	Over-optimizing	Multi-domain architecture interoperability conflicts	Over-optimizing	Multi-domain architecture interoperability conflicts		Staffing, Empowering	Domain architecting within domain
	Tight coupling	Over-optimizing		Over-optimizing		Value prioritizing	
	Use software vs. hardware						
Cyber Capability	Agile Methods scalability	Multi-domain architecture interoperability conflicts	Multi-domain architecture interoperability conflicts	Cost of automated aids	Over-optimizing		Automated aids
	Multi-domain architecture interoperability conflicts	Over-optimizing	Over-optimizing	Multi-domain architecture interoperability conflicts	Physical architecture or cyber architecture		Domain architecting within domain
	Over-optimizing			Over-optimizing			
	Tight coupling						
	Use software vs. hardware						
Interoperability	Multi-domain architecture interoperability conflicts	Encryption interoperability	Multi-domain architecture interoperability conflicts	Assertion checking	Over-optimizing	Reduced speed of Assertion checking	
	User-programmed interoperability	Multi-domain architecture interoperability conflicts		Cost, duration of added connectors	Tight vs. Loose coupling	Reduced speed of connectors, standards compliance	
						Tight vs. Loose coupling	

Dependability Relations



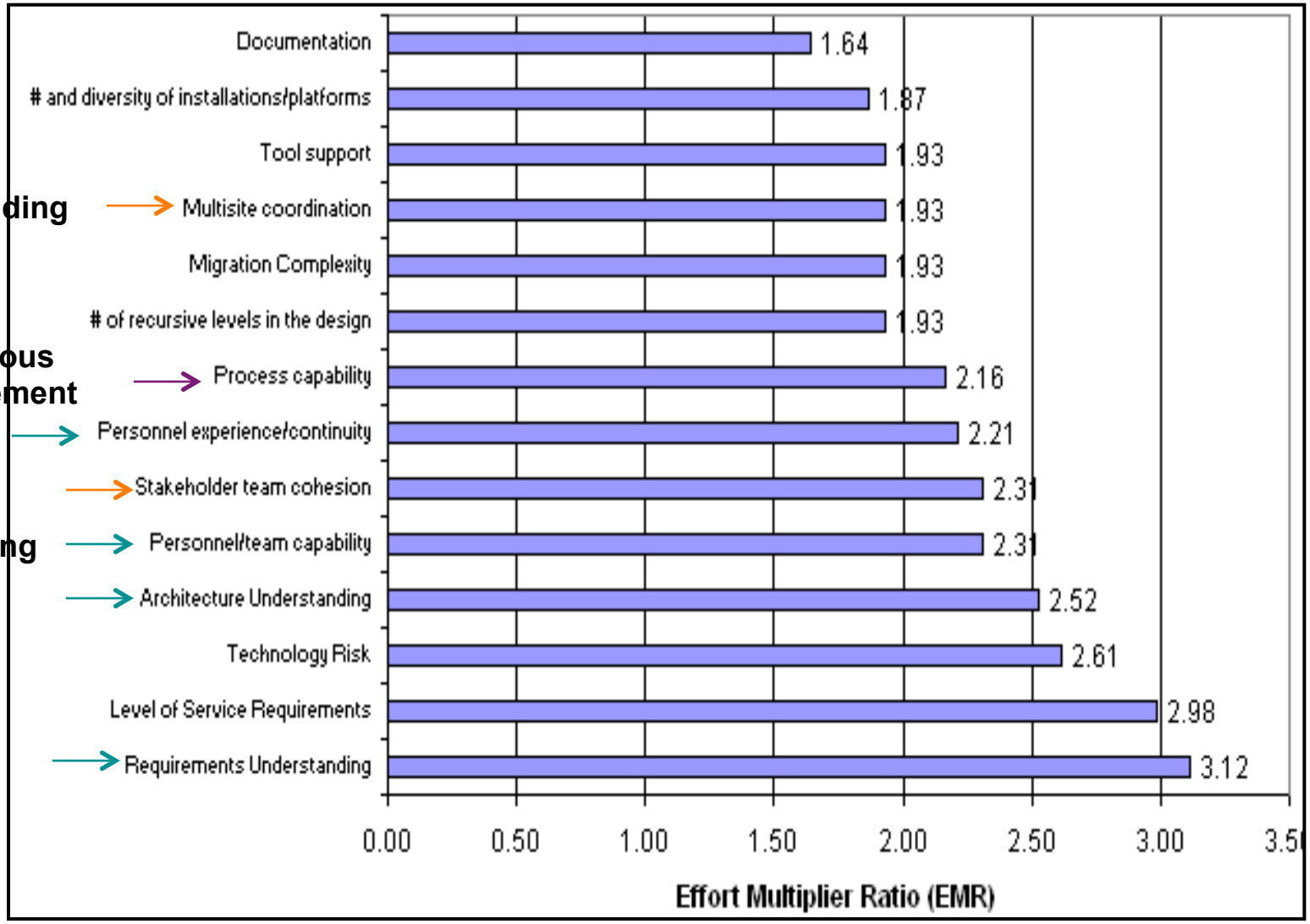
	Security	Reliability	Maintainability
Security		Confidentiality, Integrity, Avalability Assurance Cases Certification Failure Modes and Effects Analysis Fault Tree Analysis Recertification	Certification Diagnosability Integrity, Avalability Repairability Smart Monitoring Spare Capacity
Reliability	Non-redundancy (For Security) Redundancy (For Reliability)		Accessibility Certification Diagnosability Repairability Smart Monitoring Spare Capacity
Maintainability	Accessibility Compartmentalization Encryption Recertification	Armor Recertification	

Metrics Evaluation Criteria

- **Correlation with quality**: How much does the metric relate with our notion of software quality? It implies that nearly all programs with a similar value of the metric will possess a similar level of quality. This is a subjective correlational measure, based on our experience.**I**
- **Importance**: How important is the metric and are low or high values preferable when measuring them? The scales, in descending order of priority are: Extremely Important, Important and Good to have
- **Feasibility of automated evaluation**: Are things fully or partially automable and what kinds of metrics are obtainable?
- **Ease of automated evaluation**: In case of automation how easy is it to compute the metric? Does it involve mammoth effort to set up or can it be plug-and-play or does it need to be developed from scratch? Any OTS tools readily available?
- **Completeness of automated evaluation**: Does the automation completely capture the metric value or is it inconclusive, requiring manual intervention? Do we need to verify things manually or can we directly rely on the metric reported by the tool?
- **Units**: What units/measures are we using to quantify the metric?

SYSTEMS ENGINEERING
Research Center

COSYSMO Sys Engr Cost Drivers



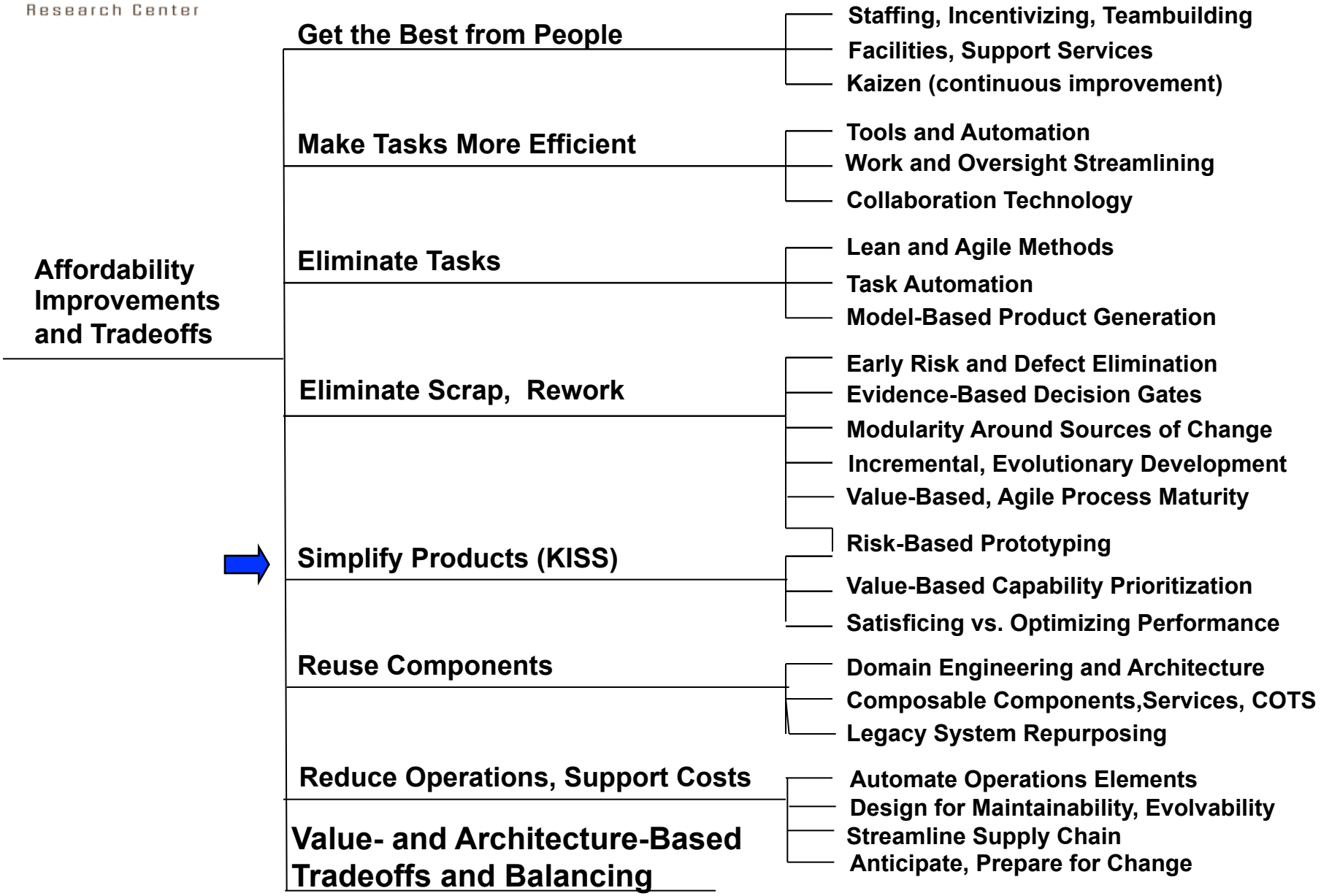
Teambuilding

Continuous Improvement

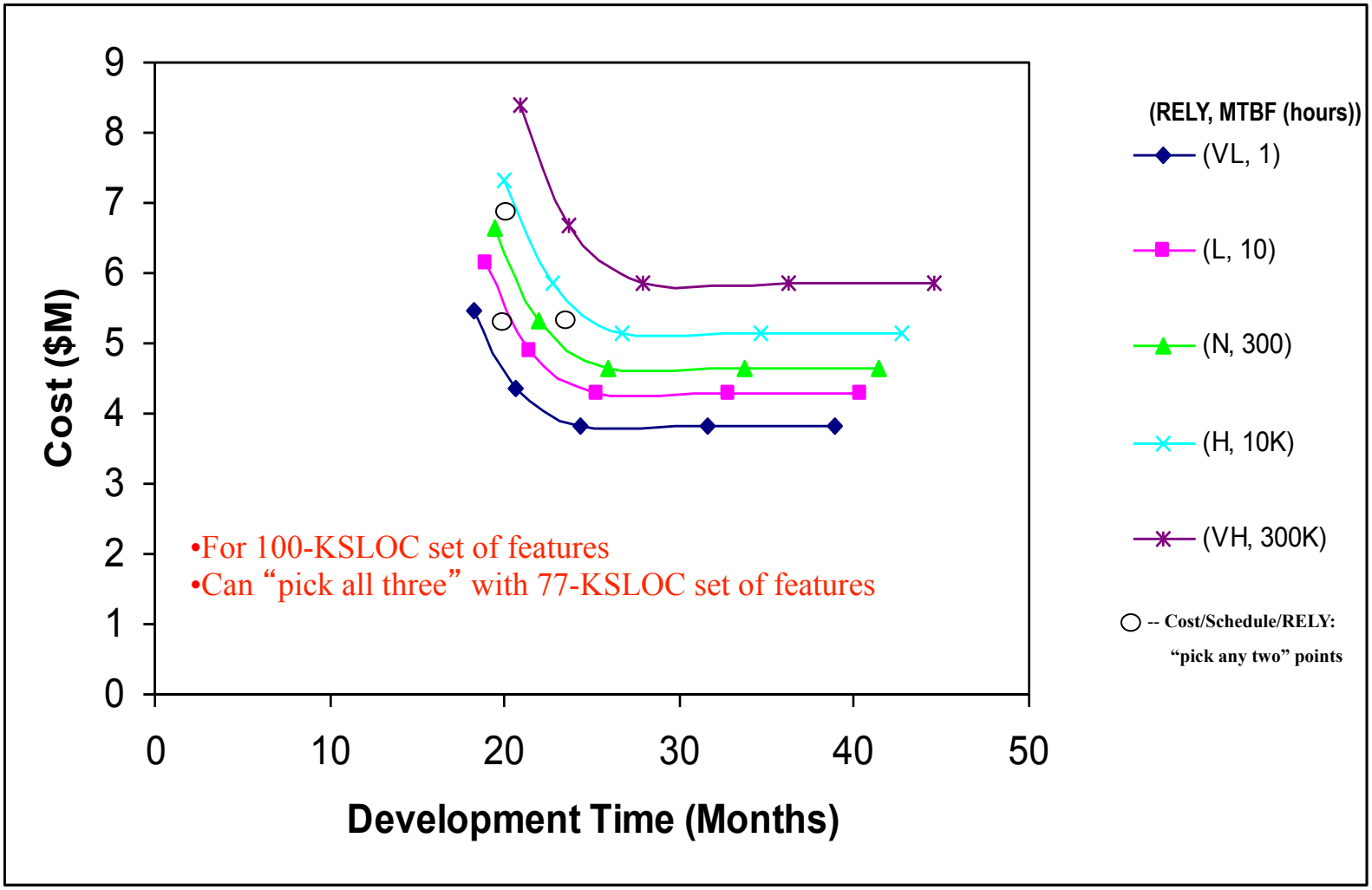
Staffing

SYSTEMS ENGINEERING
Research Center

Tradespace and Affordability Framework



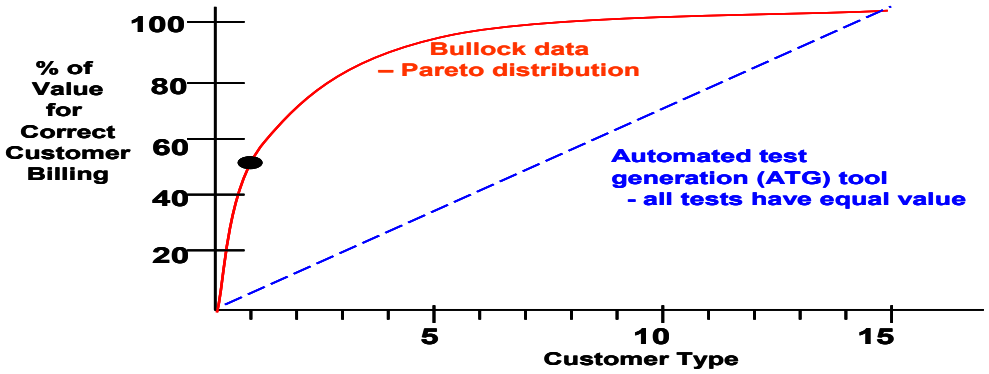
COCOMO II-Based Tradeoff Analysis
Better, Cheaper, Faster: Pick Any Two?



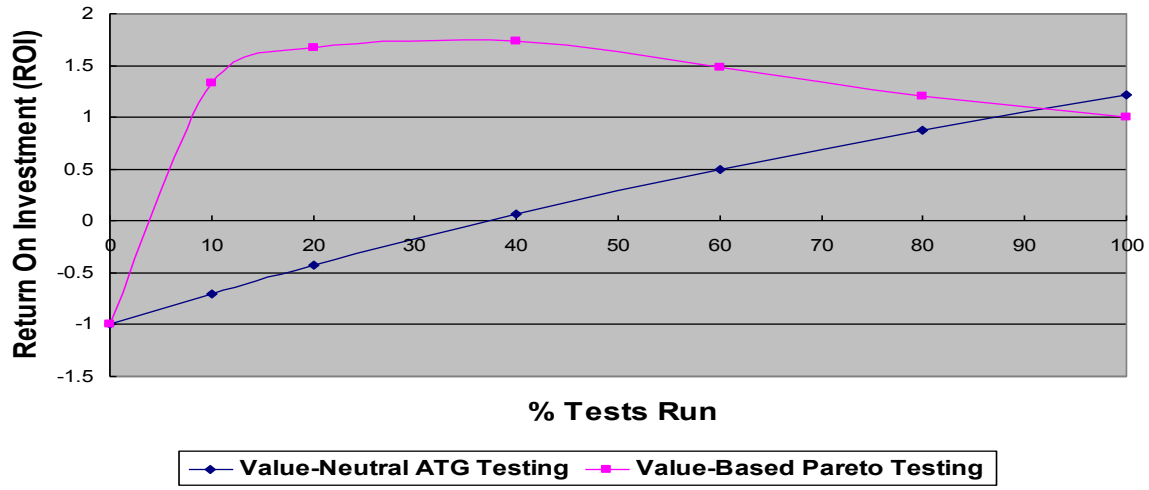
Value-Based Testing: Empirical Data and ROI

— LiGuo Huang, ISESE 2005

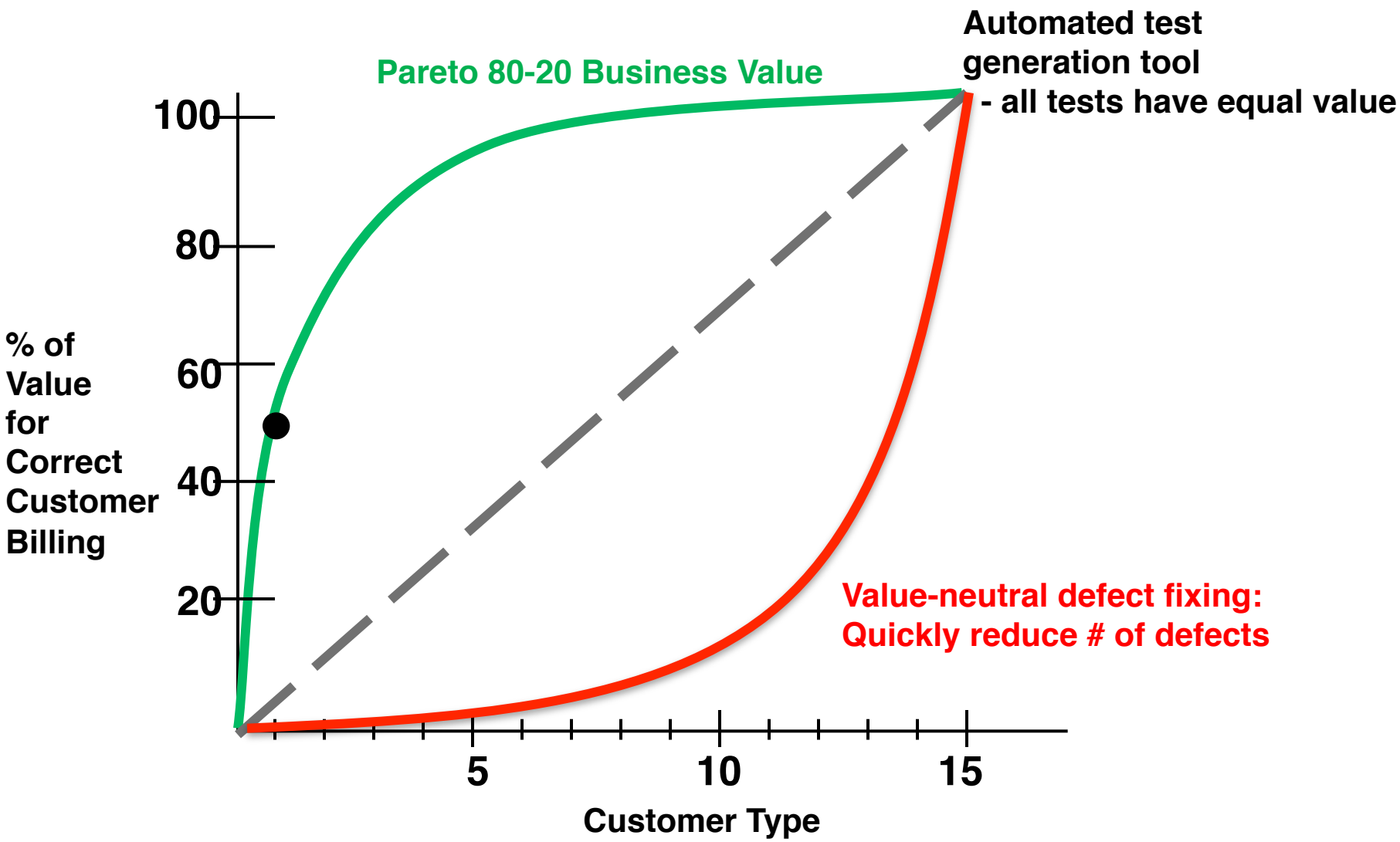
(a)



(b)



Value-Neutral Defect Fixing Is Even Worse



Product Line Engineering and Management: NPS

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

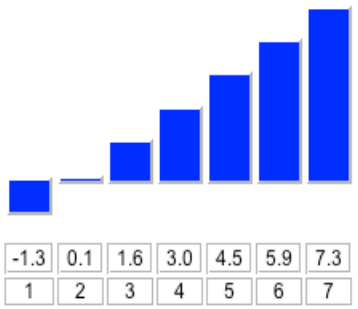
Relative Cost of Developing for PL Flexibility via Reuse: 1.7

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

Return on Investment



Cost-Schedule Tradespace Analysis

- Generally, reducing schedule adds cost
  - Pair programming: 60% schedule * 2 people = 120% cost
- Increasing schedule may or may not add cost
  - Pre-planned smaller team: less communications overhead
  - Mid-course stretchout: pay longer for tech, admin overhead
- Can often decrease both cost and schedule
  - Lean, agile, value-based methods; product-line reuse
- Can optimize on schedule via concurrent vs. sequential processes
  - Sequential; cost-optimized: Schedule = 3 * cube root (effort)
    - 27 person-months: Schedule = 3*3=9 months; 3 personnel
  - Concurrent, schedule-optimized: Schedule = square root (effort)
    - 27 person-months: Schedule = 5.5 months; 5.4 personnel
- Can also accelerate agile square root schedule
  - SERC Expediting SysE study: product, process, people, project, risk

SYSTEMS ENGINEERING Research Center

Context: SERC iTAP Initiative Elements

- Ilities Tradespace and Affordability Project (iTAP) foundations
  - More precise ility definitions and relationships
  - Stakeholder value-based, means-ends relationships
  - Ility strategy effects, synergies, conflicts
  - USC, MIT, U. Virginia
- ➡ Next-generation system cost-schedule estimation models
  - Initially for full-coverage space systems (COSATMO)
  - Extendable to other domains
  - USC, AFIT, GaTech, NPS
- Applied iTAP methods, processes, and tools (MPTs)
  - For concurrent cyber-physical-human systems
  - Experimental MPT piloting, evolution, improvement
  - Wayne State, AFIT, GaTech, NPS, Penn State, USC

COSATMO Concept

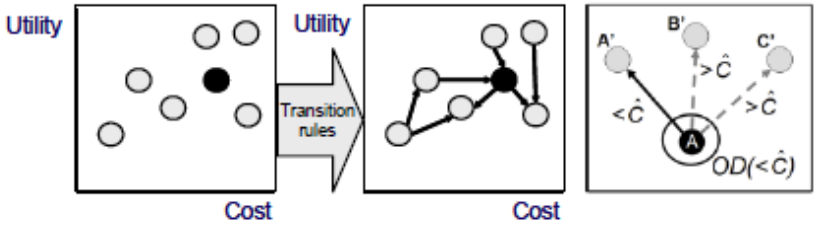
- Co-sponsored by OSD, USAF/SMC
- Focused on current and future satellite systems
  - Accommodating rapid change, evolutionary development, Net-Centric SoSs, families of systems, future security and self-defense needs, microsats, satellite constellations, model-based development
  - Recognizes new draft DoDI 5000.02 process models
    - Hardware-intensive, DoD-unique SW-intensive, Incremental SW-intensive, Accelerated acquisition, 2 Hybrids (HW-, SW-dominant)
  - Covers full life cycle: definition, development, production, operations, support, phaseout
  - Covers full system: satellite(s), ground systems, launch
  - Covers hardware, software, personnel costs
- Extensions to cover systems of systems, families of systems
- Several PhD dissertations involved (as with COSYSMO)
  - Incrementally developed based on priority, data availability
- Upcoming workshop at USC Annual Research Review April 29-May 1

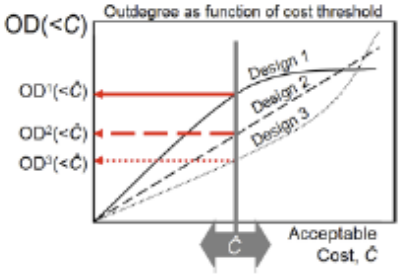
SYSTEMS ENGINEERING Research Center

MIT: ilities in Tradespace Exploration

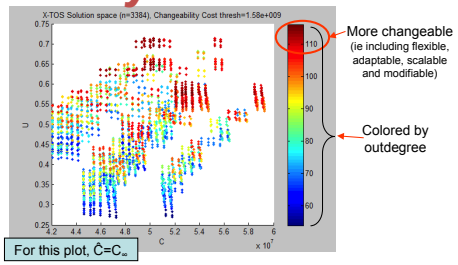
Based on SEAri research

Enabling Construct: Tradespace Networks

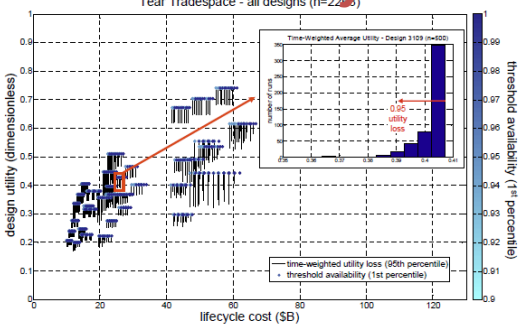




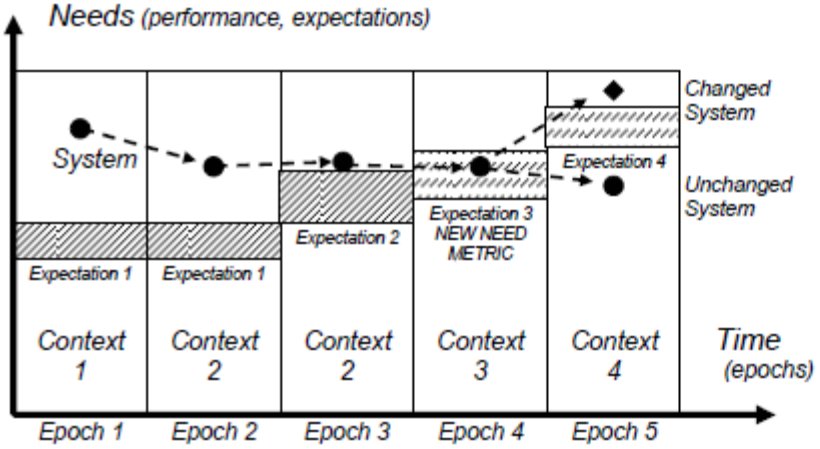
Changeability



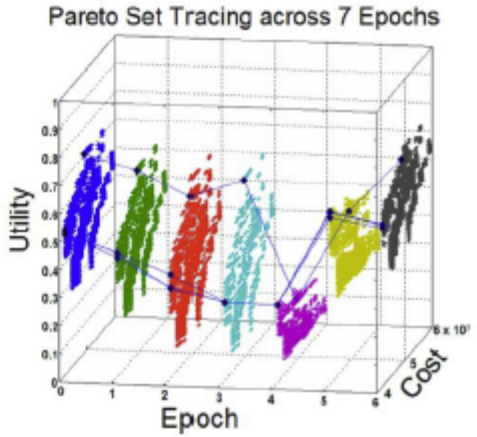
Survivability



Enabling Construct: Epochs and Eras



Value Robustness



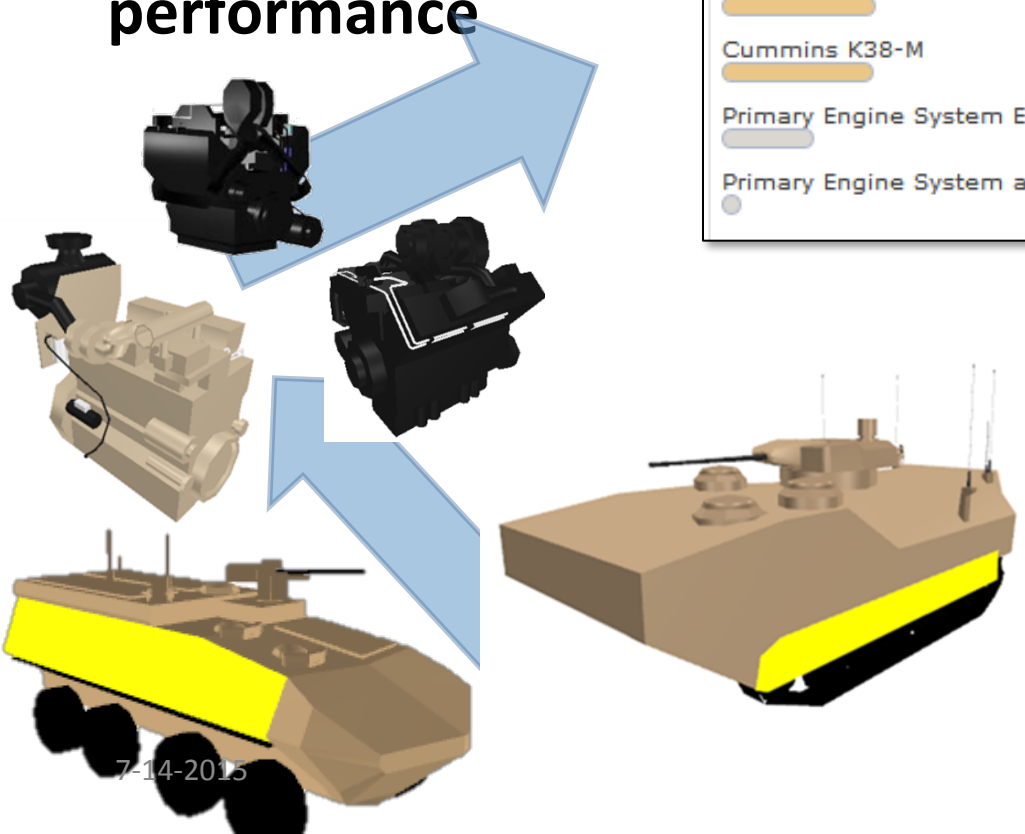
Set of Metrics

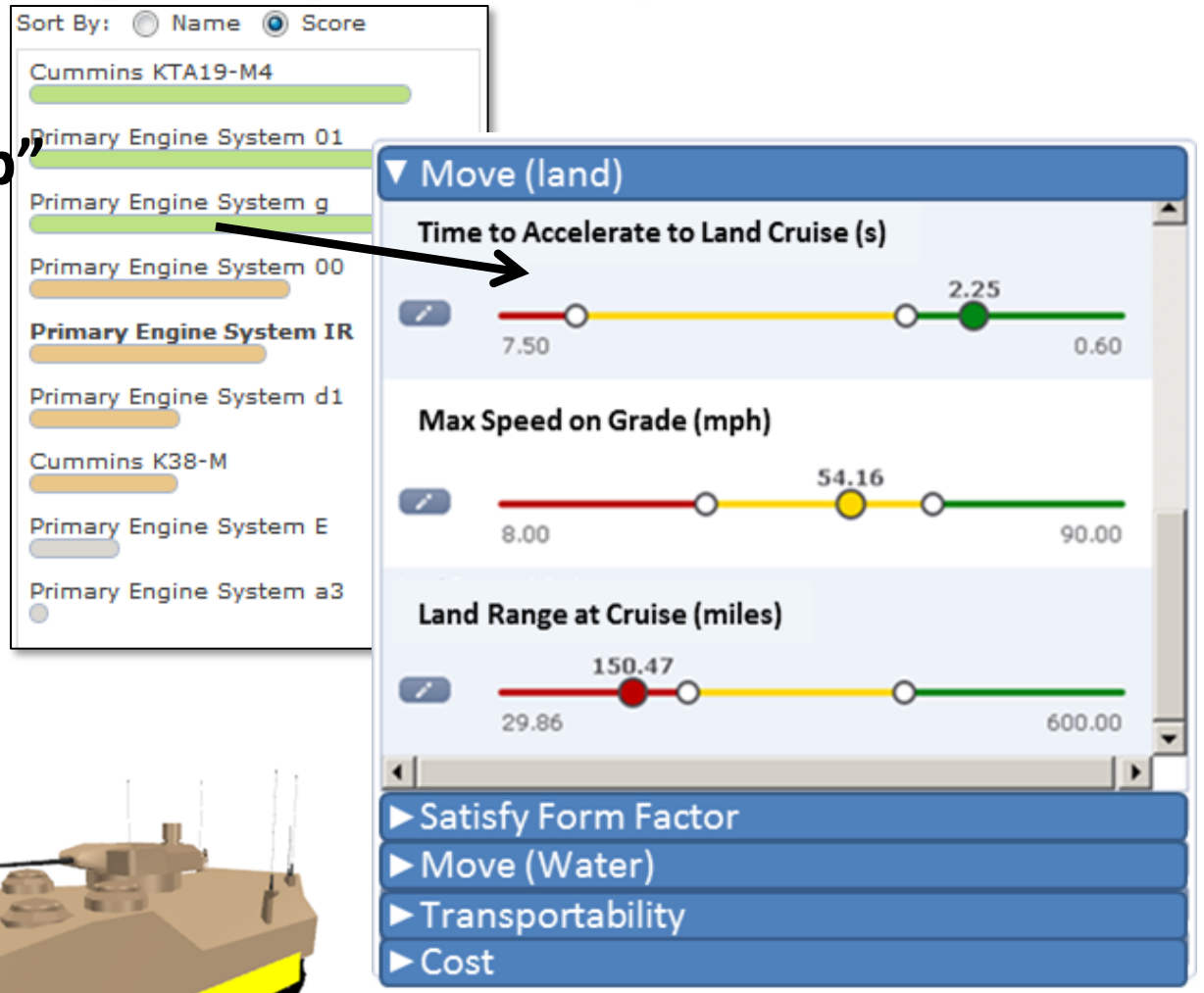
Value Aspect	Acronym	Stands For	Definition
Robustness via "no change"	NPT	Normalized Pareto Trace	% epochs for which design is Pareto efficient in utility/cost
Robustness via "no change"	fNPT	Fuzzy Normalized Pareto Trace	Above, with margin from Pareto front allowed
Robustness via "change"	eNPT, efNPT	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

GaTech – FACT Tradespace Tool

Being used by Marine Corps

- **Configure vehicles from the "bottom up"**
- **Quickly assess impacts on performance**

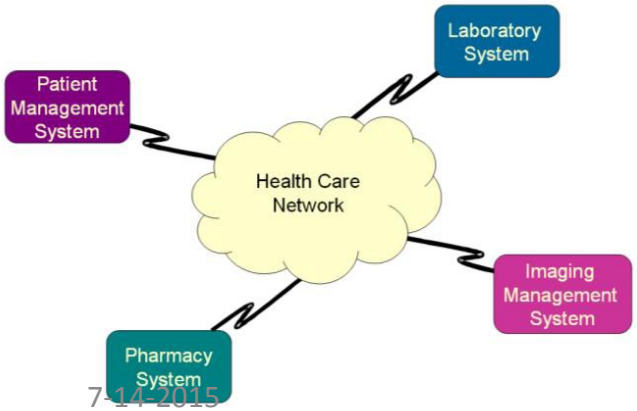


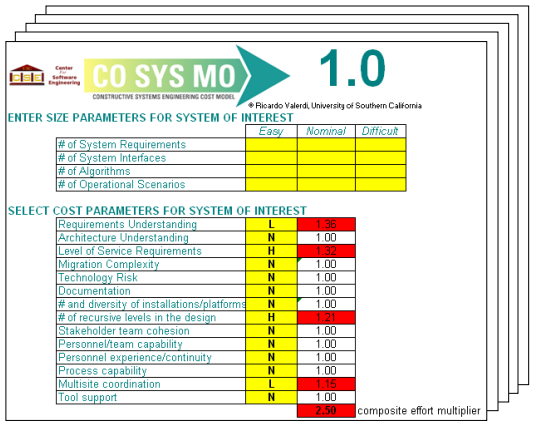


SysML Building Blocks for Cost Modeling

GaTech-USC Work in RT46 Phase 2 (Oct-Dec 2013)

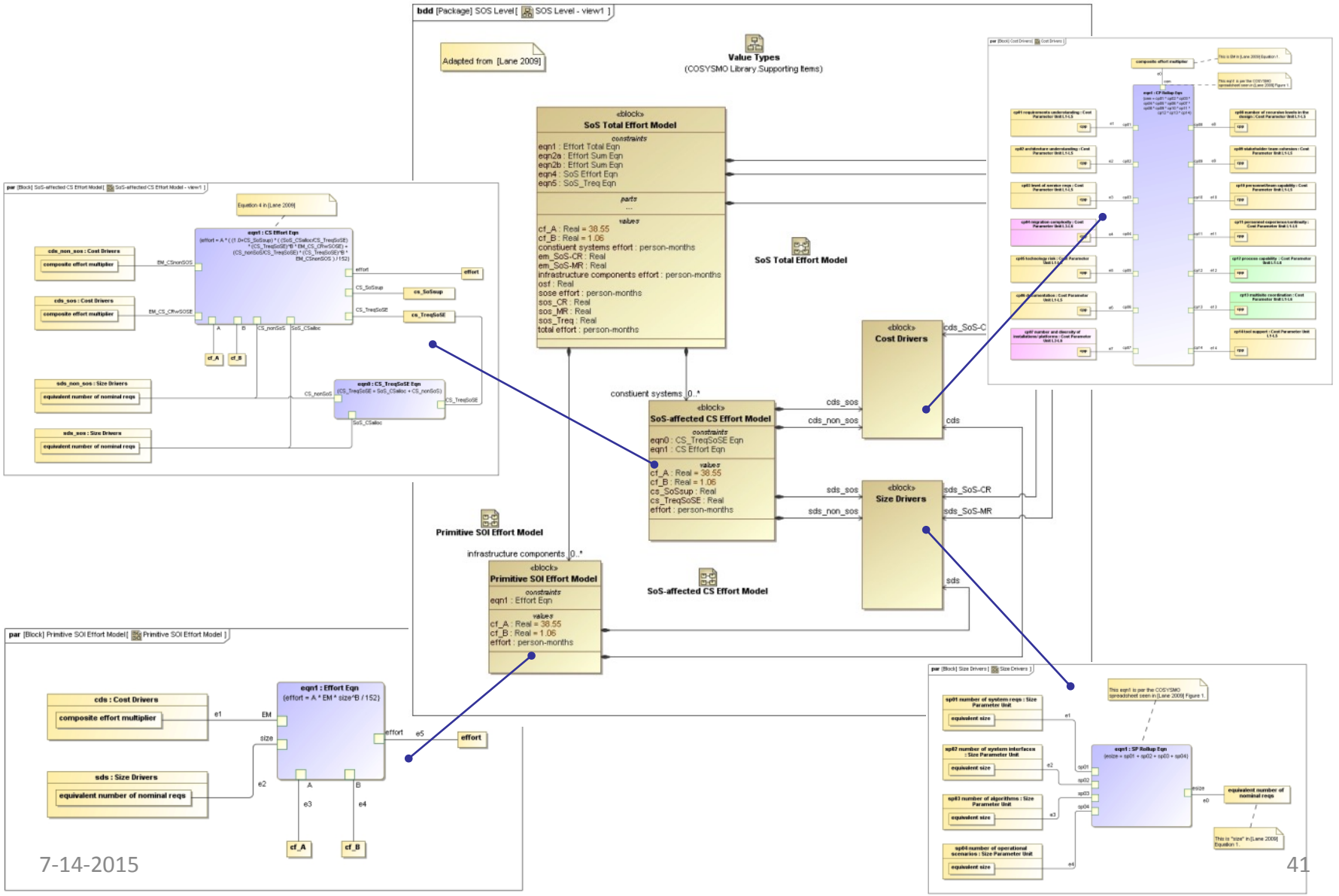
- Implemented reusable SysML building blocks
  - Based on SoS/COSYSMO SE cost (effort) modeling work by Lane, Valerdi, Boehm, et al.
- Successfully applied building blocks to healthcare SoS case study from [Lane 2009]
- Provides key step towards affordability trade studies involving diverse "-ilities" (*see MIM slides*)





Aspect	Formula	Calculated Effort
SoSE effort (Equation 5)	Effort = $38.55*[((SoS_{CR} / SoS_{Treq})* (SoS_{Treq})^{1.06} * EM_{SOS-CR}) + ((SoS_{MR} / SoS_{Treq})* (SoS_{Treq})^{1.06} * EM_{SOS-MR}*OSF)] /152$ = $38.55*[((50 / 52) * (52)^{1.06} * 2.50) + (20/52)*(52)^{1.06}* 0.47 * 10\%)] / 152$	40.41
Pharmacy System effort (Equation 4)	Effort = $38.55*[(1.0+CS_{SoSmsp}) * ((SoS_{CSalloc}/CS_{TreqSoSE})* (CS_{TreqSoSE})^{1.06}* EM_{CS-CRwSOSE}) + (CS_{nonSoS}/CS_{TreqSoSE}) * (CS_{TreqSoSE})^{1.06} * EM_{CSnonSOS}] /152$ = $38.55 *[(1.15) * ((50/70)*(70)^{1.06} * 1.06 + (20/70) * (70)^{1.06}*0.72] / 152$	22.02
Laboratory System effort (Equation 4)	Effort = $38.55*[(1.0+CS_{SoSmsp}) * ((SoS_{CSalloc}/CS_{TreqSoSE})* (CS_{TreqSoSE})^{1.06}* EM_{CS-CRwSOSE}) + (CS_{nonSoS}/CS_{TreqSoSE}) * (CS_{TreqSoSE})^{1.06} * EM_{CSnonSOS}] /152$ = $38.55 *[(1.15) * ((50/50)*(50)^{1.06} * 1.06 + 0] / 152$	19.55
Imaging System effort (Equation 4)	Effort = $38.55*[(1.0+CS_{SoSmsp}) * ((SoS_{CSalloc}/CS_{TreqSoSE})* (CS_{TreqSoSE})^{1.06}* EM_{CS-CRwSOSE}) + (CS_{nonSoS}/CS_{TreqSoSE}) * (CS_{TreqSoSE})^{1.06} * EM_{CSnonSOS}] /152$ = $38.55 *[(1.15) * ((50/50)*(50)^{1.06} * 1.06 + 0] / 152$	19.55
New infrastructure component effort (Equation 1)	Effort = $38.55*EM*(size)^{1.06}/152$ = $38.55 * 1.0 * (100)^{1.06} / 152$	33.43
	Total Effort:	134.96

Healthcare SoS Case Study [Lane 2009] Implemented Using SysML Building Blocks: *Selected SysML Diagrams*



SERC Expediting SysE study:

Product, process, people, project; risk factors

Final Database

Over 30 Interviews with Gov't/ Industry Rapid Development Organizations

Over 23,500 words from interview notes

Product, Process, People … all in a Project Context

SYSTEMS ENGINEERING
Research Center

CORADMO-SE Rating Scales, Schedule Multipliers

Accelerators/Ratings	Very Low	Low	Nominal	High	Very High	Extra High
Product Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Simplicity	Extremely complex	Highly complex	Mod. complex	Moderately simple	Highly simple	Extremely simple
Element Reuse	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Low-Priority Deferrals	Never	Rarely	Sometimes	Often	Usually	Anytime
Models vs Documents	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Key Technology Maturity	>0 TRL 1,2 or >1 TRL 3	1 TRL 3 or > 1 TRL 4	1 TRL 4 or > 2 TRL 5	1-2 TRL 5 or >2 TRL 6	1-2 TRL 6	All > TRL 7
Process Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Concurrent Operational Concept, Requirements, Architecture, V&V	Highly sequential	Mostly sequential	2 artifacts mostly concurrent	3 artifacts mostly concurrent	All artifacts mostly concurrent	Fully concurrent
Process Streamlining	Heavily bureaucratic	Largely bureaucratic	Conservative bureaucratic	Moderate streamline	Mostly streamlined	Fully streamlined
General SE tool support CIM (Coverage, Integration, Maturity)	Simple tools, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Project Factors	**1.08**	**1.04**	**1.0**	**0.96**	**0.93**	**0.9**
Project size (peak # of personnel)	Over 300	Over 100	Over 30	Over 10	Over 3	≤ 3
Collaboration support	Globally distributed weak comm. , data sharing	Nationally distributed, some sharing	Regionally distributed, moderate sharing	Metro-area distributed, good sharing	Simple campus, strong sharing	Largely collocated, Very strong sharing
Single-domain MMPTs (Models, Methods, Processes, Tools)	Simple MMPTs, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Multi-domain MMPTs	Simple; weak integration	Minimal CIM	Some CIM or not needed	Moderate CIM	Considerable CIM	Extensive CIM
People Factors	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
General SE KSAs (Knowledge, Skills, Agility)	Weak KSAs	Some KSAs	Moderate KSAs	Good KSAs	Strong KSAs	Very strong KSAs
Single-Domain KSAs	Weak	Some	Moderate	Good	Strong	Very strong
Multi-Domain KSAs	Weak	Some	Moderate or not needed	Good	Strong	Very strong
Team Compatibility	Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions
Risk Acceptance Factor	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
	Highly risk-averse	Partly risk-averse	Balanced risk aversion, acceptance	Moderately risk-accepting	Considerably risk-accepting	Strongly risk-accepting

CORADMO-SE Calibration Data

Mostly Commercial; Some DoD

Application Type	Technologies	Person Months	Duration (Months)	Duration / √PM	Product	Process	Project	People	Risk	Multi-plier	Error %
Insurance agency system	HTML/VB	34.94	3.82	0.65	VH	VH	XH	VH	N	0.68	5%
Scientific/engineering	C++	18.66	3.72	0.86	L	VH	VH	VH	N	0.80	-7%
Compliance - expert	HTML/VB	17.89	3.36	0.79	VH	VH	XH	VH	N	0.68	-15%
Barter exchange	SQL/VB/ HTML	112.58	9.54	0.90	VH	H	H	VH	N	0.75	-16%
Options exchange site	HTML/SQL	13.94	2.67	0.72	VH	VH	XH	VH	N	0.68	-5%
Commercial HMI	C++	205.27	13.81	0.96	L	N	N	VH	N	0.93	-3%
Options exchange site	HTML	42.41	4.48	0.69	VH	VH	XH	VH	N	0.68	-1%
Time and billing	C++/VB	26.87	4.80	0.93	L	VH	VH	VH	N	0.80	-14%
Hybrid Web/client-server	VB/HTML	70.93	8.62	1.02	L	N	VH	VH	N	0.87	-15%
ASP	HTML/VB/SQL	9.79	1.39	0.44	VH	VH	XH	VH	N	0.68	53%
On-line billing/tracking	VB/HTML	17.20	2.70	0.65	VH	VH	XH	VH	N	0.68	4%
Palm email client	C/HTML	4.53	1.45	0.68	N	VH	VH	VH	N	0.76	12%

SYSTEMS ENGINEERING
Research Center

Case Study: From Plan-Driven to Agile

Initial Project: Focus on Concurrent SE

Accelerators/Ratings	VL	L	N	H	VH	XH
Product Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Simplicity			X			
Element Reuse	X					
Low-Priority Deferrals	X					
Models vs Documents		X				
Key Technology Maturity			X			
Process Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Concurrent Operational Concept, Requirements, Architecture, V&V				X		
Process Streamlining		X				
General SE tool support CIM (Coverage, Integration, Maturity)				X		
Project Factors	**1.08**	**1.04**	**1.0**	**0.96**	**0.93**	**0.9**
Project size (peak # of personnel)				X		
Collaboration support				X		
Single-domain MMPTs (Models, Methods, Processes, Tools)				X		
Multi-domain MMPTs		X				
People Factors	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
General SE KSAs (Knowledge, Skills, Agility)			X			
Single-Domain KSAs				X		
Multi-Domain KSAs		X				
Team Compatibility			X			
Risk Acceptance Factor	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
			X			

Expected schedule reduction of 1.09/0.96 = 0.88 (green arrow)
Actual schedule delay of 15% due to side effects (red arrows)
Model prediction: 0.88*1.09*1.04*1.06*1.06 = 1.13

SYSTEMS ENGINEERING
Research Center

Case Study: From Plan-Driven to Agile

Next Project: Fix Side Effects; Reduce Bureaucracy

Accelerators/Ratings	VL	L	N	H	VH	XH
Product Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Simplicity			X			
Element Reuse	X					
Low-Priority Deferrals	X					
Models vs Documents		X				
Key Technology Maturity					X	
Process Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Concurrent Operational Concept, Requirements, Architecture, V&V					X	
Process Streamlining				X		
General SE tool support CIM (Coverage, Integration, Maturity)				X		
Project Factors	**1.08**	**1.04**	**1.0**	**0.96**	**0.93**	**0.9**
Project size (peak # of personnel)				X		
Collaboration support				X		
Single-domain MMPTs (Models, Methods, Processes, Tools)				X		
Multi-domain MMPTs		X				
People Factors	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
General SE KSAs (Knowledge, Skills, Agility)				X		
Single-Domain KSAs				X		
Multi-Domain KSAs		X				
Team Compatibility				X		
Risk Acceptance Factor	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
			X			

Model estimate: 0.88*(0.92/0.96)*(0.96/1.05) = 0.77 speedup

Project results: 0.8 speedup

Model tracks project status; identifies further speedup potential