SYSTEMS ENGINEERING
Research Center

System Qualities Ontology, Tradespace and Affordability (SQOTA) Project

Barry Boehm, USC

GSAW/SPIN/INCOSE-LA/IEEE-CS Event

March 2, 2016

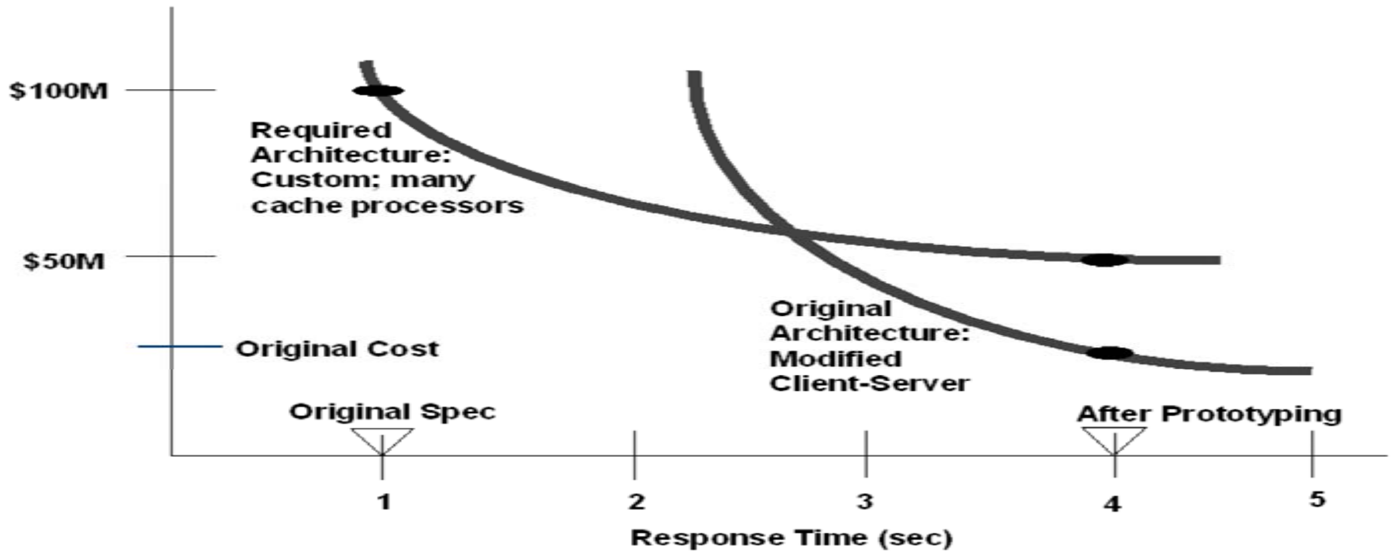
SQs Tradespace and Affordability Analysis

- **Critical nature of system qualities**
  - 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 ilities ontology
- **SQ synergies and conflicts analysis**
  - Stakeholder value-based, means-ends hierarchy
  - Synergies and Conflicts matrix and expansion
- **Affordability: Next-generation cost estimation models**

Importance of SQ Tradeoffs

Major source of DoD system overruns

- **System qualities (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



SYSTEMS ENGINEERING
Research Center

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

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 on other qualities?
  - Cost, schedule, performance, maintainability?

Need for Qualities Ontology

A structural framework for organizing information about a topic of interest

- **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, but fail on rainy-day use cases**
  - **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-quality relationships**
  - **Synergies and Conflicts, as with Security**

Initial SERC Qualities Ontology

- **Modified version of IDEF5 ontology framework**
  - Classes, Subclasses, and Individuals
  - States, Processes, and Relations
- **Top classes cover stakeholder value propositions**
  - Mission Effectiveness, Life Cycle Efficiency, Dependability, Changeabiity
- **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
- **States, Processes, and Relations cover sources of ility variation**
  - 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)

SYSTEMS ENGINEERING Research Center

SQs Tradespace and Affordability Analysis

- **Critical nature of system qualities**
  - **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 ilities ontology**
- ➡ **SQ synergies and conflicts analysis**
  - **Stakeholder value-based, means-ends hierarchy**
  - **Synergies and Conflicts matrix and expansion**
- **Affordability: Next-generation cost estimation models**

SYSTEMS ENGINEERING
Research Center

Stakeholder value-based, means-ends hierarchy

- **Mission operators and managers want improved Mission Effectiveness**
  - **Involves Physical Capability, Cyber Capability, Human Usability, Speed, Accuracy, Impact, Mobility, Scalability, Versatility, Interoperability**
- **Mission investors and system owners want Life Cycle Efficiency**
  - **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, Availability, Maintainability, Survivability, Safety, Security**
- **In an increasingly dynamic world, all want system Changeability: to be rapidly and cost-effectively changeable**
  - **Involves Maintainability, Adaptability**

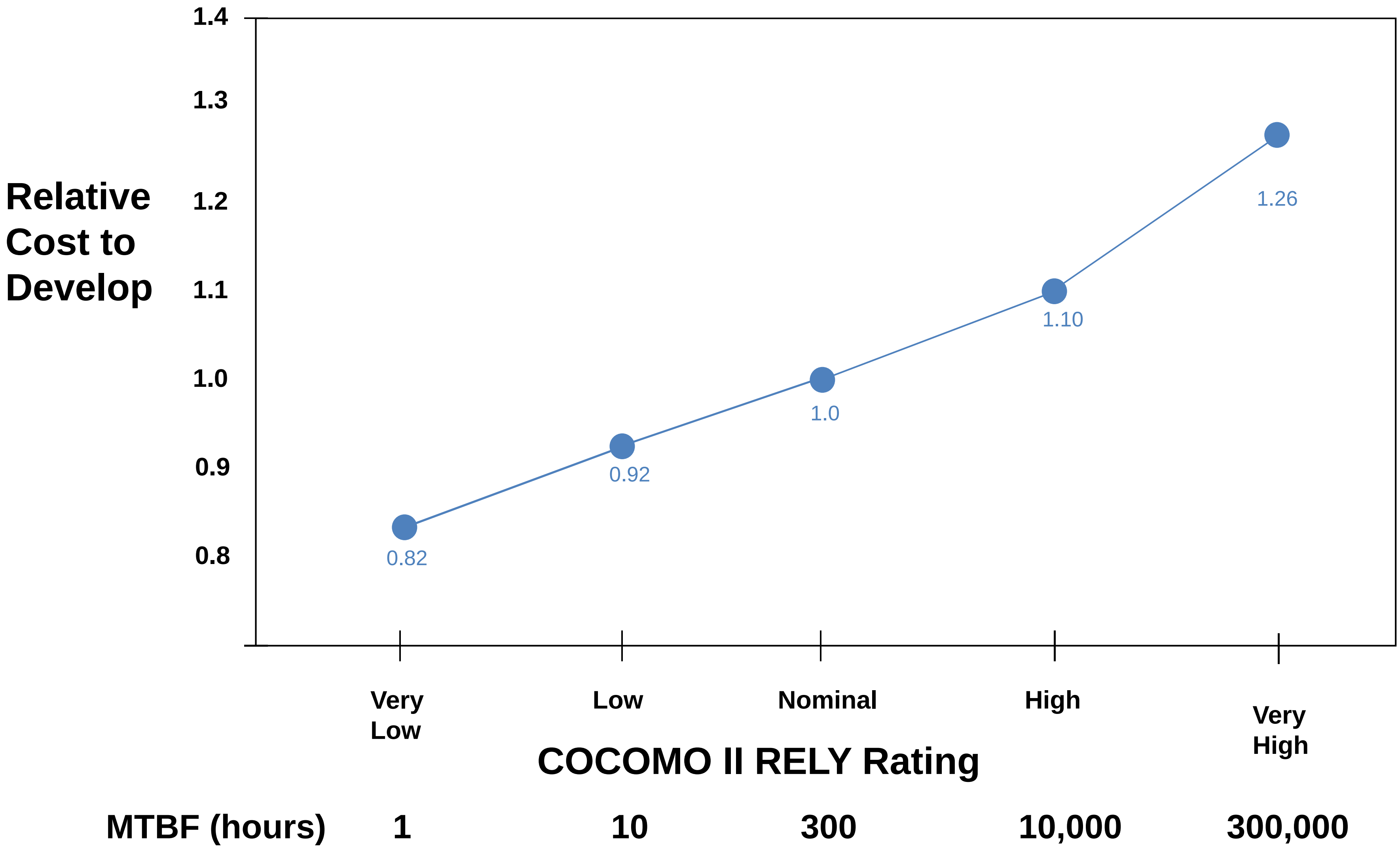
SYSTEMS ENGINEERING
Research Center

7x7 Synergies and Conflicts Matrix

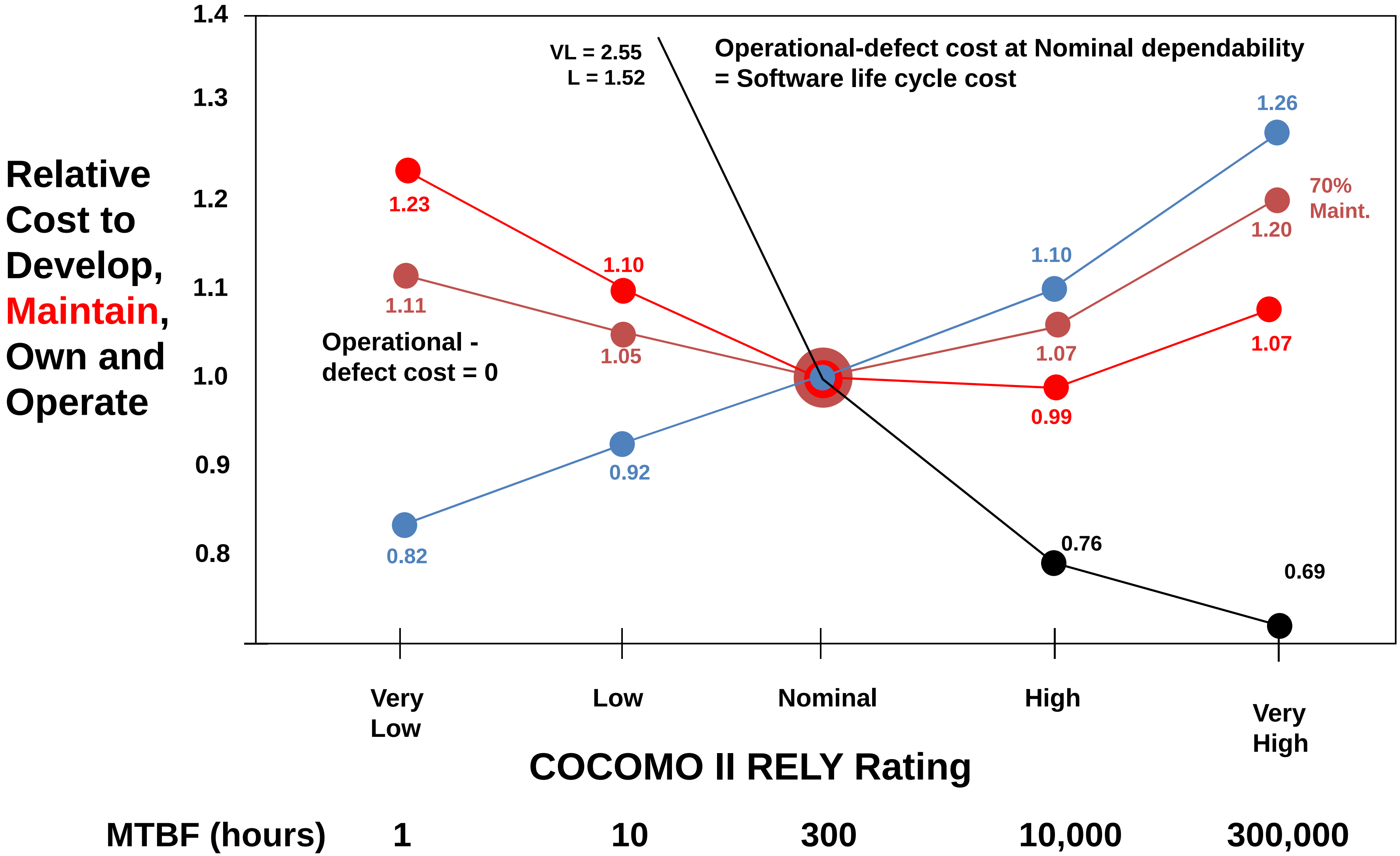
- **Mission Effectiveness expanded to 4 elements**
  - **Physical Capability, Cyber Capability, Interoperability, Other Mission Effectiveness (including Usability as Human Capability)**
- **Synergies and Conflicts among the 7 resulting elements identified in 7x7 matrix**
  - **Synergies above main diagonal, Conflicts below**
- **Exploring Qualipedia approach for obtaining details about the synergy or conflict**
  - **Ideally quantitative; example next**
- **Still need synergies and conflicts within elements**
  - **3x3 Dependability subset developed**

	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



SQs Tradespace and Affordability Analysis

- **Critical nature of system qualities**
  - **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 ilities ontology**
- **SQ synergies and conflicts analysis**
  - **Stakeholder value-based, means-ends hierarchy**
  - **Synergies and Conflicts matrix and expansion**
- **Affordability: Next-generation cost estimation models**

Next-generation cost estimation models:
Future challenges and current initiatives

- **Future Challenges**
  - **Rapid change; Systems of systems; Model-Driven and non-developmental item (NDI)-intensive systems; Ultrahigh software system assurance; Legacy maintenance; Brownfield development; Agile and iterative development.**

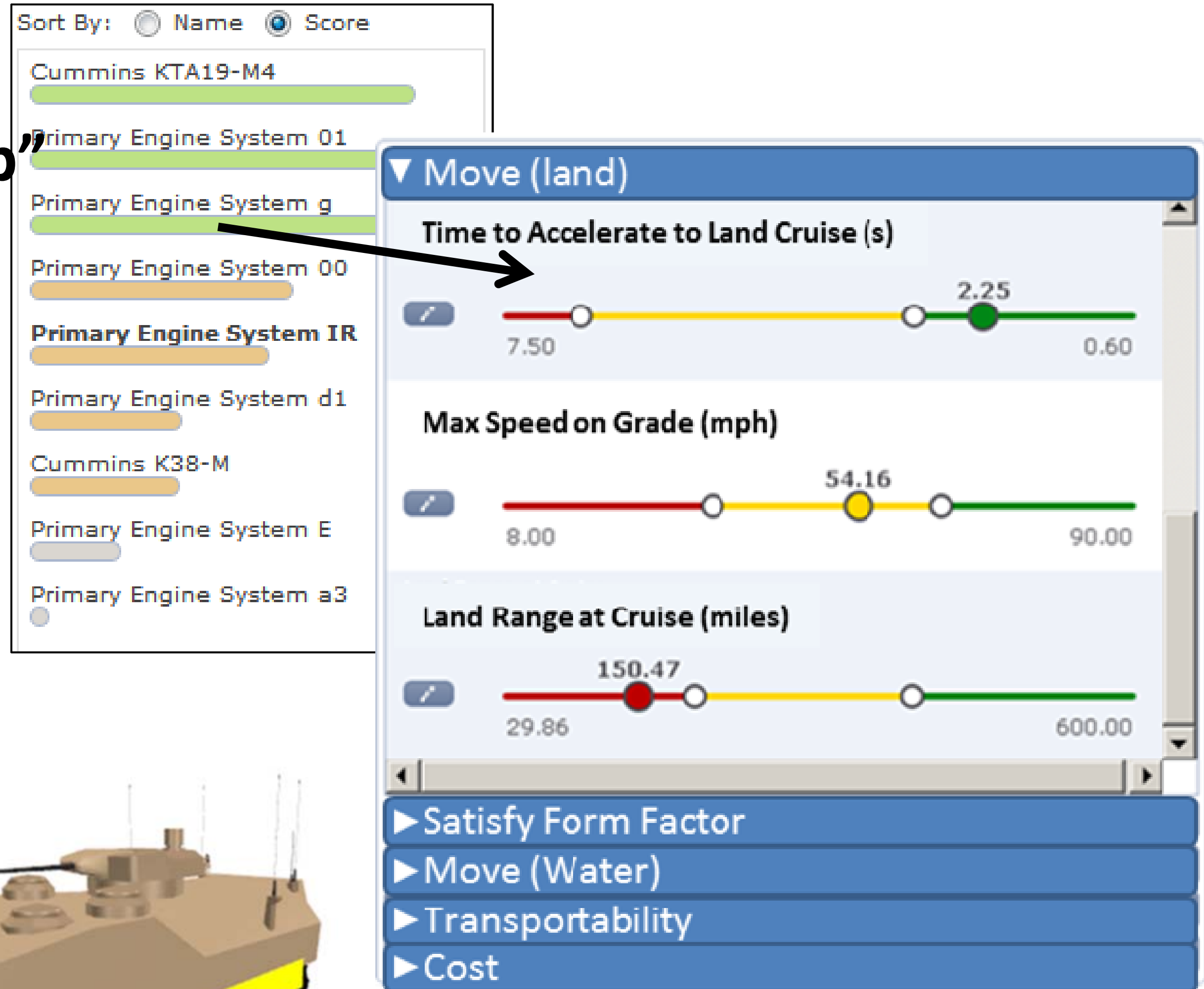
- **Current Initiatives**
  - **COCOMO III (Brad Clark lead)**
    - **Preserve most of current COCOMO II parameters**
    - **Different sizing, cost drivers for different domains**
  - **COSYSMO 3.0 (Jim Alstad lead)**
    - **Bring together reuse, requirements volatility extensions**
    - **Address future challenges above**

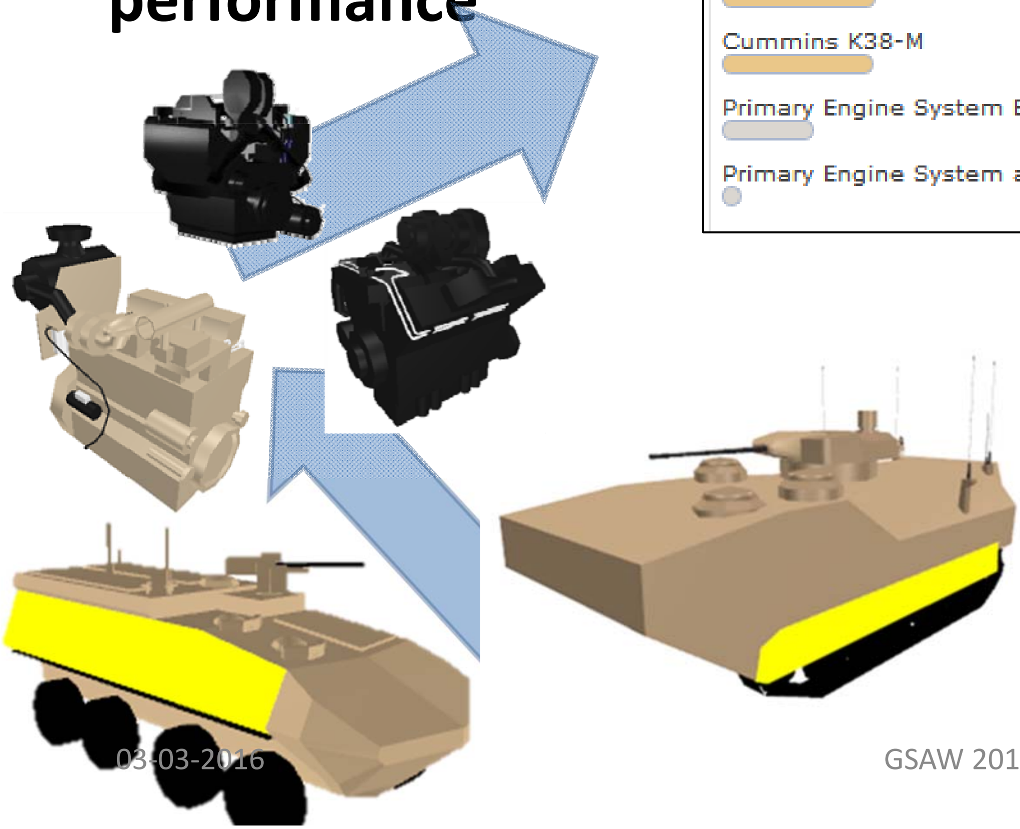
Backup charts

GaTech – FACT Tradespace Tool

Being used by Marine Corps; Army, Navy extensions

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

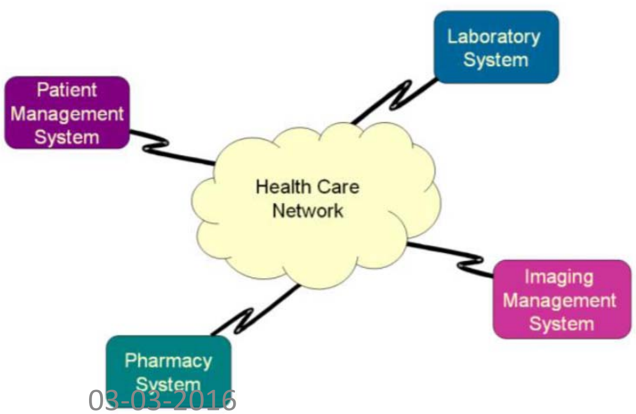


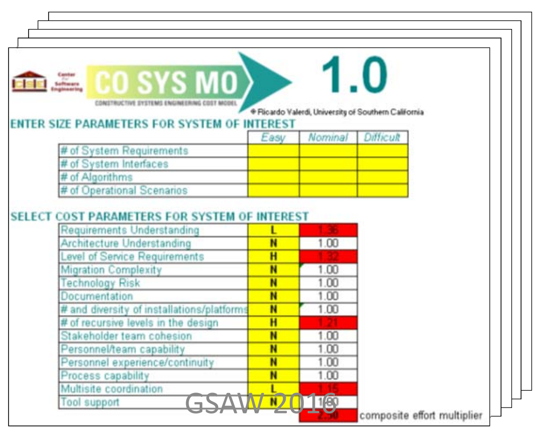


SysML Building Blocks for Cost Modeling

Russell Peak, GTRI; Jo Ann Lane, USC

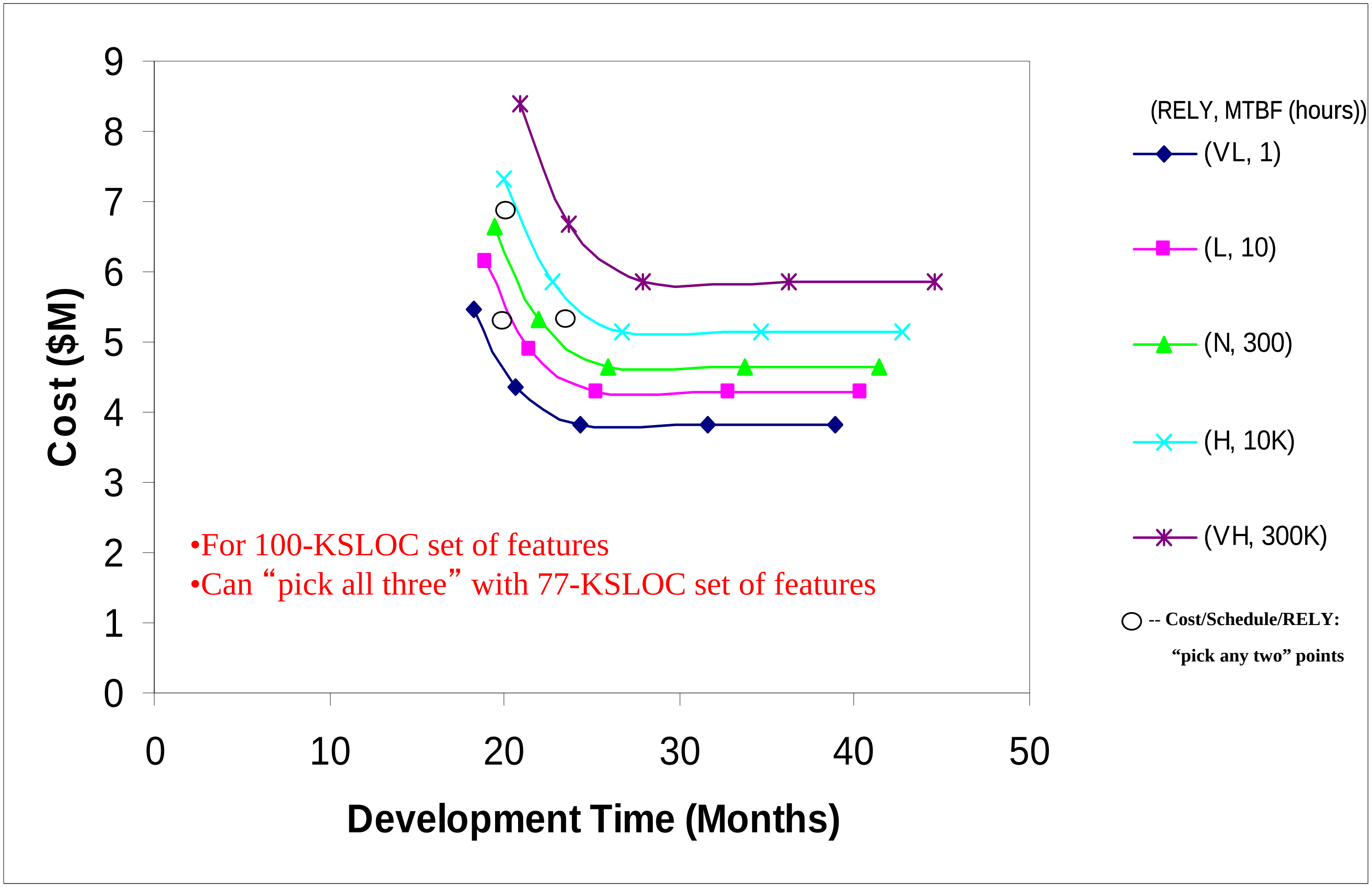
- Implemented reusable SysML building blocks
  - Based on SoS/COSYSMO SE cost (effort) modeling work by Lane, Valerdi, Boehm, et al.
- Successfully applied building blocks to healthcare SoS case study from [Lane 2009]
- Provides key step towards affordability trade studies involving diverse "-ilities" (*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_{SoSsup}) * (($SoS_{CSalloc}$/$CS_{TreqSoSE}$)* ($CS_{TreqSoSE}$)$^{1.06}$* $EM_{CS-CRwSOSE}$) + (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_{SoSsup}) * (($SoS_{CSalloc}$/$CS_{TreqSoSE}$)* ($CS_{TreqSoSE}$)$^{1.06}$* $EM_{CS-CRwSOSE}$) + (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_{SoSsup}) * (($SoS_{CSalloc}$/$CS_{TreqSoSE}$)* ($CS_{TreqSoSE}$)$^{1.06}$* $EM_{CS-CRwSOSE}$) + (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

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

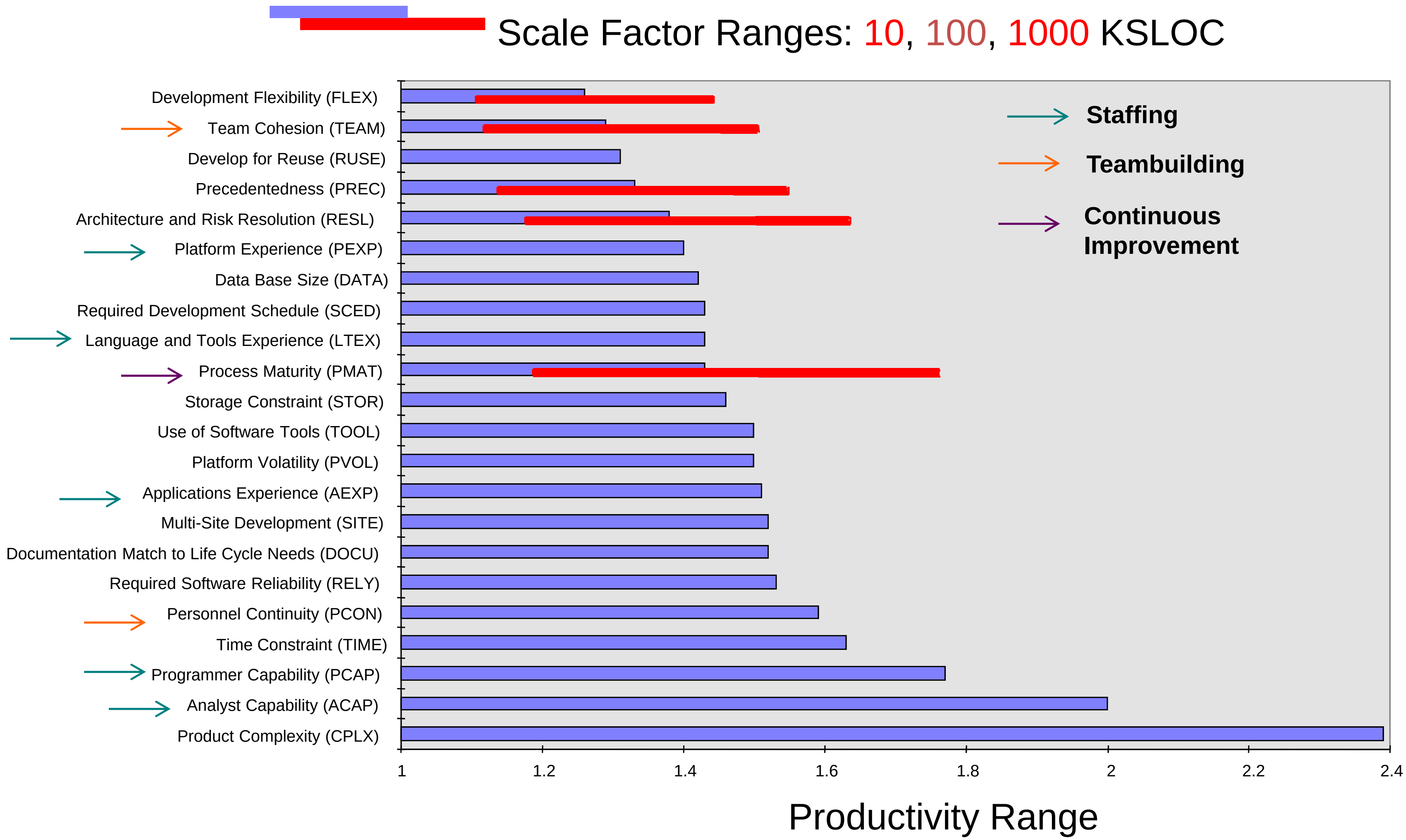


Affordability and Tradespace Framework

Cost 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