SYSTEMS ENGINEERING
Research Center

System Qualities (SQs) Tradespace and Affordability

Barry Boehm, USC

SERC SSRR 2016
November 17, 2016

SYSTEMS ENGINEERING Research Center

Project Overview - I

PROJECT DESCRIPTION:

- Develop and build upon firm scientific foundations for reasoning about tradespaces among System Qualities, particularly for Life Cycle Affordability, using its MORS and INCOSE definition as Cost-Effectiveness. Develop, pilot, refine, and transition improved SQ tradespace methods, processes, and tools (MPTs), using set-based design tradespaces, versus current point-solution designs.

VALUE:

- Being able to quickly and rigorously analyze the tradespace of complex systems, especially with regard to DoD-critical SQs such as security, resilience, adaptability usability, interoperability, and affordability, will aid decision makers early in the life cycle in a project when alternative solutions are all under consideration.

CONTRIBUTOR(S):

- University of Southern California, Georgia Institute of Technology, Massachusetts Institute of Technology, Stevens Institute of Technology, University of Virginia, Wayne State University, Air Force Institute of Technology, Naval Post-graduate School, Pennsylvania State University

SYSTEMS ENGINEERING Research Center

Project Overview - II

RESEARCH AREAS:

Research divided into three areas

- Foundations and Frameworks: Developing a formally-based, systems engineering and management-based ontology for the SQs, their sources of variation, and their tradespace interactions
- Full-Lifecycle, Set-Based SQ Tradespace Methods, Processes, and Tools Extension and Demonstration
- Next Generation, Full Life Cycle Cost, Schedule, and Quality Estimation Models for System Engineering and Software-Intensive Systems

Non-ASD(R&E) Sponsors

- USAF ASC, SMC, AFCAA; USA ERDC, TARDEC; USMC; USN NAVSEA, NSWC, NCCA

SYSTEMS ENGINEERING Research Center

Outline

- ➡ Critical nature of system qualities (SQs)
  - Or non-functional requirements (NFRs); ilities
  - Major source of project overruns, failures
  - Underemphasized in project management
  - Poorly defined, understood
- Foundations: An initial SQs ontology
  - Nature of an ontology; choice of IDEF5 structure
  - Stakeholder value-based, means-ends hierarchy
  - Synergies and Conflicts matrix and expansions
  - Maintainability deep dive results to date
- Future plans

Importance of SQ Tradeoffs

Major source of system overruns

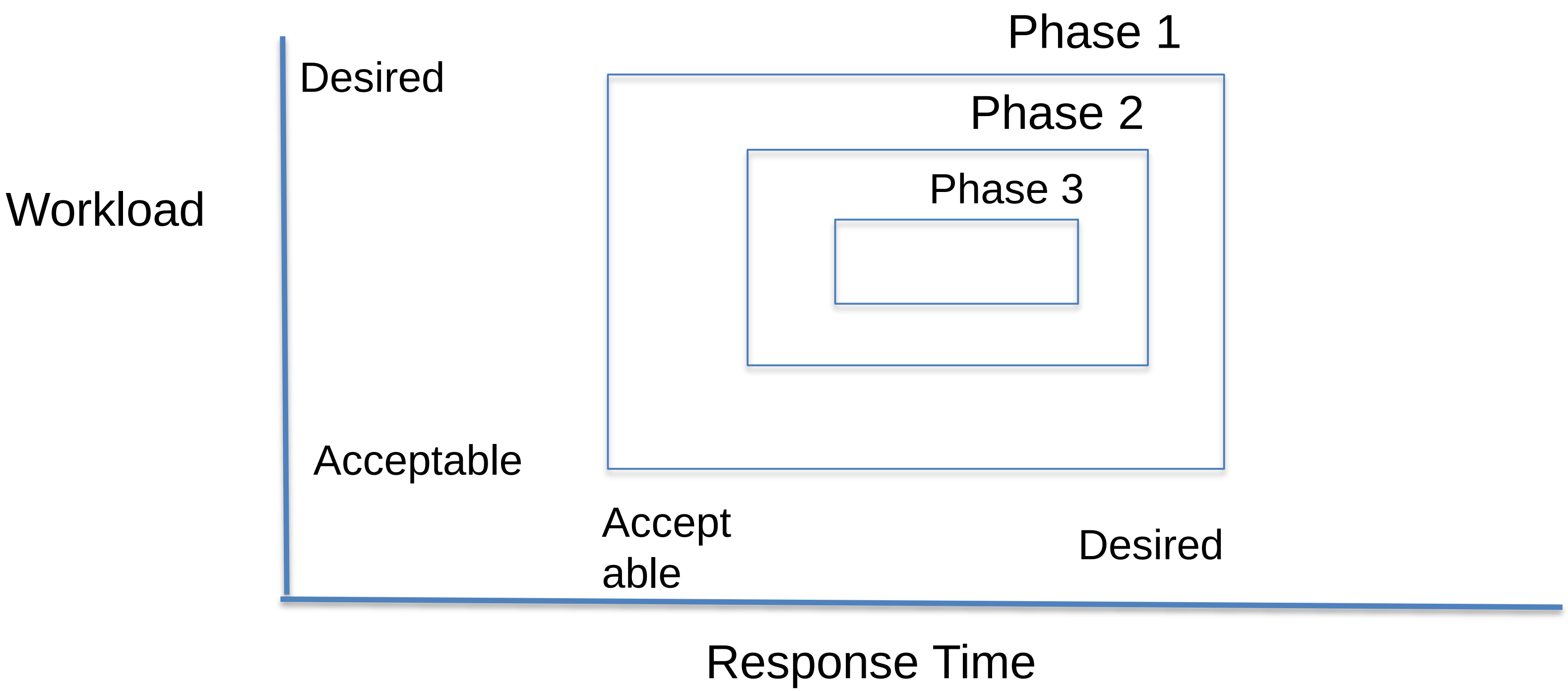
- SQs have systemwide impact
  - System elements generally just have local impact
- SQs often exhibit asymptotic behavior
  - Watch out for the knee of the curve
- Best architecture is a discontinuous function of SQ level
  - "Build it quickly, tune or fix it later" highly risky
- Large system example
  - COTS-based Dept-level system successful with 1-second response
  - Contract to extend to full organization with 1-second response
  - COTS–based system unscalable; custom solution developed
    - Cost: $100 million vs. $30 million budget
    - Prototyping found 4-second response OK 90% of time
    - COTS-based system workable; changing 1 character in SQ requirement changed cost from $100M to $30M

SYSTEMS ENGINEERING Research Center

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**

Set-Based SQs Definition Convergence
Enables Systems Engineering Tradespace



Phase 1. Rough ConOps, Rqts, Solution Understanding
Phase 2. Improved ConOps, Rqts, Solution Understanding
Phase 3. Good ConOps, Rqts, Solution Understanding

SYSTEMS ENGINEERING Research Center

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?

SYSTEMS ENGINEERING Research Center

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**

SYSTEMS ENGINEERING Research Center

Example Concern with Quality Standards

Among several with ISO/IEC 25010

- **Example: Definition of 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 (bad data, communications, users)
    - Similarly for unspecified quality requirements, e.g., security
- **Need to reflect diversity**
  - Different stakeholders rely on different capabilities (functions, performance, flexibility, etc.) at different times and in different environments
- **Quality definitions need a more precise ontology**

Outline

- Critical nature of system qualities (SQs)
  - Or non-functional requirements (NFRs); ilities
  - Major source of project overruns, failures
  - Underemphasized in project management
  - Poorly defined, understood
- Foundations: An initial SQs ontology
  - Nature of an ontology; choice of IDEF5 structure
  - Stakeholder value-based, means-ends hierarchy
  - Synergies and Conflicts matrix and expansions
  - Maintainability deep dive results to date
- Future plans

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

SYSTEMS ENGINEERING
Research Center

Current 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, 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
- **Referents, States, Processes, Relations cover SQ variation**
  - Referents: Sources of variation by stakeholder value context
  - States: Internal (beta-test); External (infrastructure, interoperators)
  - Processes: Operational scenarios (normal vs. crisis; experts vs. novices)
  - Relations: Impact of other SQs (security as above, synergies & conflicts)

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, Endurabilty, Maneuverability, Accuracy, Impact, Scalability, Versatility, Interoperability
- Mission investors and system owners want Life Cycle Efficiency
  - Involves Development and Maintenance Cost, Duration, Key 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, Maintainability, Availability, Survivability, Robustness, Graceful Degredation, Safety, Security
- In an increasingly dynamic world, all want system Changeability: to be rapidly and cost-effectively evolvable
  - Involves Maintainability, Modifiability, Repairability, Adaptability

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:**
  - **Stakeholder value propositions: 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: status checking frequency; types of missions supported; workload volume, interoperability with independently evolving external systems**
  - **Effects via relations with other SQs: synergies improving other SQs; conflicts degrading other SQs**

Current 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, 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
- **Referents, States, Processes, Relations cover SQ variation**
  - Referents: Sources of variation by stakeholder value context
  - States: Internal (test vs. use); External (infrastructure, interoperators)
  - Processes: Operational scenarios (normal vs. crisis workload)
  - Relations: Impact of other SQs (security as above, synergies & conflicts)

Relations: 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
  - Ideally quantitative; example 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	

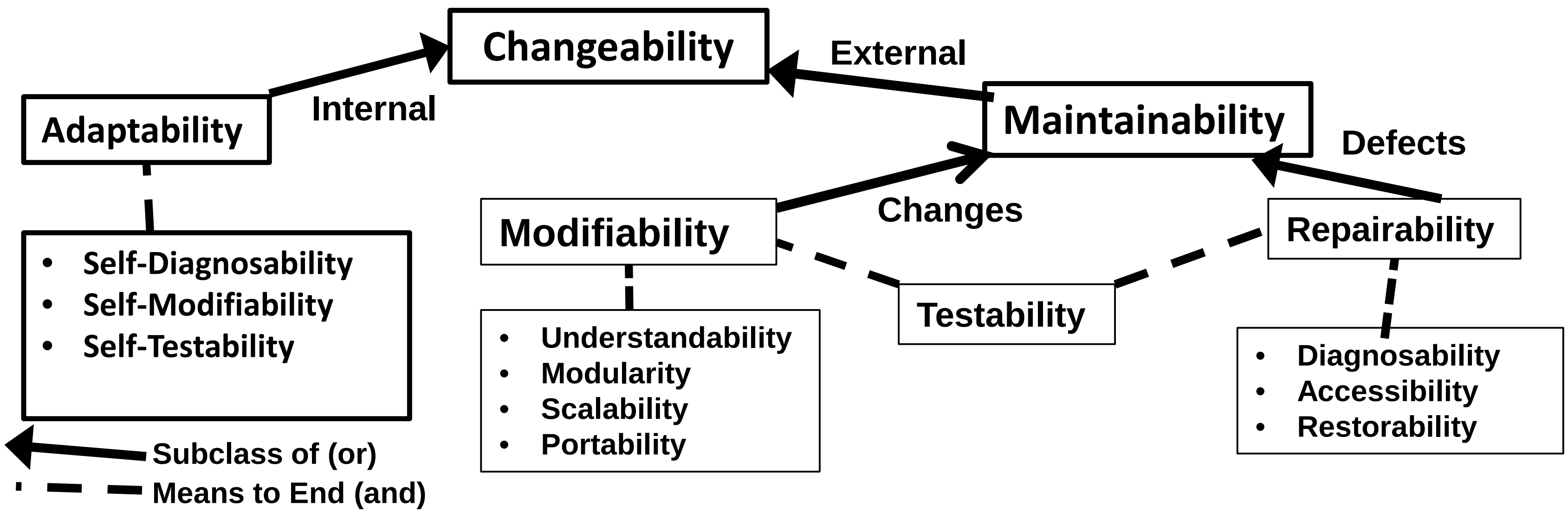
Outline

- Critical nature of system qualities (SQs)
  – Or non-functional requirements (NFRs); ilities
  – Major source of project overruns, failures
  – Underemphasized in project management
  – Poorly defined, understood
- Foundations: An initial SQs ontology
  – Nature of an ontology; choice of IDEF5 structure
  – Stakeholder value-based, means-ends hierarchy
  – Synergies and Conflicts matrix and expansions
  ➡ Maintainability deep dive results to date
- Future plans

Product Quality View of Changeability

MIT Quality in Use View also valuable

- **Changeability (PQ): Ability to become different product**
  - **Swiss Army Knife Versatile but not Changeable**
- **Changeability (Q in Use): Ability to accommodate changes in use**
  - **Swiss Army Knife doesn't change as a product but is Changeable in use**



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"

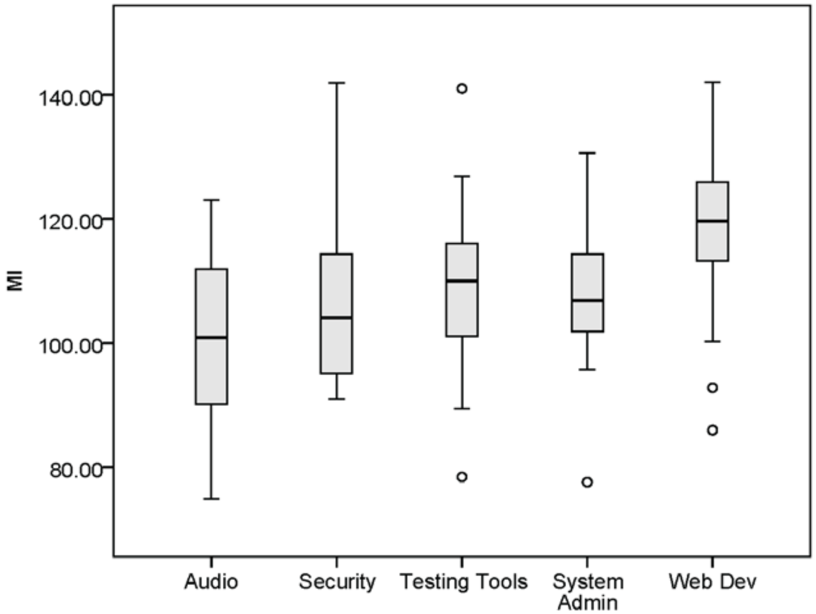
Perturbation	Context	Phase	Agent	Impetus: Nature	Impetus: Parameter	Impetus: Destination	Impetus: Aspect	Outcome: Effect	Outcome: Parameter	Outcome: Destination	Outcome: Aspect	Abstraction	Reaction	Span	Cost	Benefit
					"parameter"	"state"			"parameter"	"state"			"threshold"	"threshold"	"threshold"	"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								

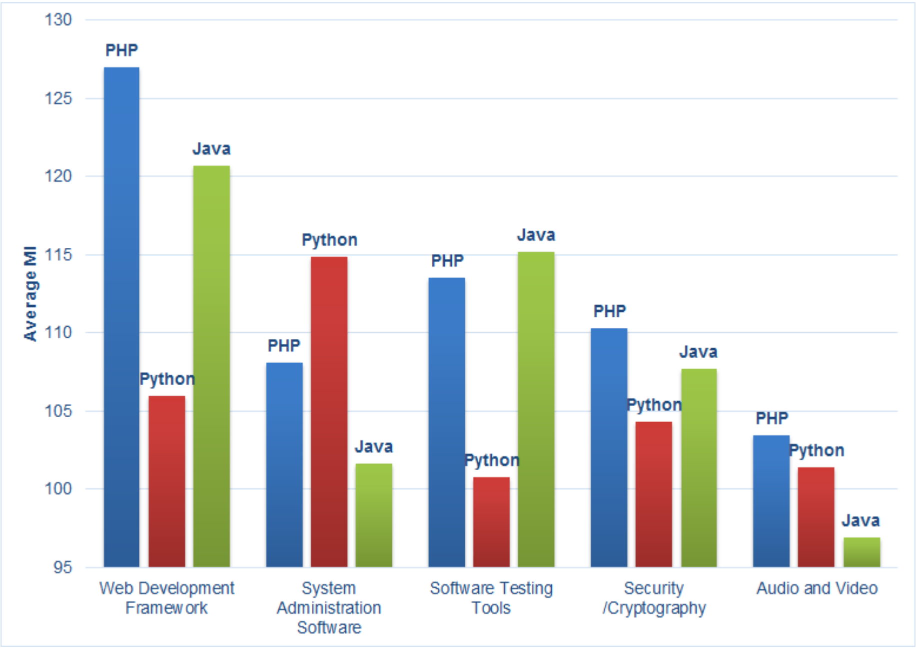
Perturbation	Context	Phase	Agent	Impetus: Nature	Impetus: Parameter	Impetus: Destination	Impetus: Aspect	Outcome: Effect	Outcome: Parameter	Outcome: Destination	Outcome: Aspect	Abstraction	Reaction	Span	Cost	Benefit	Ility Label
shift		ops						same	"Value"	few							Value Robustness
disturbance		ops						same	"Value"	few							Value Survivability
shift		ops						same		few							Robustness
shift		ops		not-same				same		few							Active Robustness
shift		ops		same		few		same		few							Passive Robustness
shift		ops	none	same		few		same	level	few	form	system					Classical Passive Robustness
disturbance		ops						same		few							Survivability
			either	not-same				not-same									Changeability
shift	general	inter-LC		not-same				not-same				architecture					Evolvability
			internal	not-same				not-same									Adaptability
			external	not-same				not-same									Flexibility
				not-same				not-same	level								Scalability
				not-same				not-same	set								Modifiability
		ops	either	not-same				increase	set								Extensibility
				not-same				not-same	any					shorter			Agility
				not-same				not-same	any				sooner				Reactivity
		ops		same	"Element set"	one	form	not-same	"Link set"		form						Form Reconfigurability
		ops		same	"Element set"	one	operations	not-same	"Order set"		operations						Operational Reconfigurability
		ops		same		one	form/ops	not-same	set	few/many							Versatility
		ops		same		one	form/ops	not-same	set	few/many	function						Functional Versatility
		ops		same		one	form/fnct	not-same	set	few/many	operations						Operational Versatility
		ops		same		one	fnct/ops	not-same	set	few/many	form						Substitutability

Initial Empirical Study:
Evaluate SW Maintainability Index on Open Source Projects

- **Evaluate MI across 97 open source projects**
  - **3 programming languages: Java, PHP, Python**
  - **5 domains: Web development framework, System administration, Test tools, Security/Encryption, Audio-Video**
- **Test MI invariance across languages, domains**
- **Evaluate completeness of MI vs. other sources**
  - **COCOMO II Software Understandability factors**
    - **Structuredness (cohesion, coupling)**
    - **Self-descriptiveness (documentation quality)**
    - **Application clarity (software reflects application content)**
  - **Other maintainability enablers (architecture, V&V support)**
    - **Repairability: Diagnosability, Accessibility, Testability, Tool support**
    - **Search for similar defects; root cause analysis**

MI Variation among domains





- Web Development Framework has shown the highest medians and the highest maximum value.
- Audio and Video has both the lowest maximum value and the lowest median value
- **PHP** may be a good option for projects that desires higher maintainability within Web Development Framework, Security/Cryptography and Audio and Video domain,
- **Python** may be a good option for System Administrative Software
- **Java** may be a good option for Software Testing Tools**.**

What is Technical Debt (TD)?

- TD: Delayed technical work or rework that is incurred when short-cuts are taken or short-term needs are addressed first
  - The later you pay for it, the more it costs (interest on debt)
- Global Information Technology Technical Debt [Gartner 2010]
  - 2010: Over $500 Billion; By 2015: Over $1 Trillion
- TD as Investment
  - Competing for first-to-market
  - Risk assessment: Build-upon prototype of key elements
  - Rapid fielding of defenses from terrorist threats
- TD as Lack of Foresight
  - Overfocus on Development vs. Life Cycle
  - Skimping on Systems Engineering
  - Hyper-Agile Development: Easiest-First increments
    - Neglecting Rainy-Day Use Cases, Non-Functional Requirements

SYSTEMS ENGINEERING Research Center

Top-10 Non-Technical Sources of TD

1. Separate organizations and budgets for systems and software acquisition and maintenance
2. Overconcern with the Voice of the Customer
3. The Conspiracy of Optimism
4. Inadequate system engineering resources
5. Hasty contracting that focuses on fixed operational requirements
6. CAIV-limited system requirements
7. Brittle, point-solution architectures
8. The Vicious Circle
9. Stovepipe systems
10. Over-extreme forms of agile development

SIS Maintainability Readiness Framework (SMRF)

SYSTEMS ENGINEERING Research Center

Software-Intensive Systems Maintainability Readiness Levels			
SMR Level	OpCon, Contracting: Missions, Scenarios, Resources, Incentives	Personnel Capabilities and Participation	Enabling Methods, Processes, and Tools (MPTs)
9	5 years of successful maintenance operations,including outcome-based incentives, adaptation to new technologies, missions, and stakeholders	In addition, creating incentives for continuing effective maintainability. performance on long-duration projects	Evidence of improvements in innovative O&M MPTs based on ongoing O&M experience
8	One year of successful maintenance operations, including outcome-based incentives, refinements of OpCon.	Stimulating and applying People CMM Level 5 maintainability practices in continuous improvement and innovation in such technology areas as smart systems, use of multicore processors, and 3-D printing	Evidence of MPT improvements based on ongoing refinement, and extensions of ongoing evaluation, initial O&M MPTs.
7	System passes Maintainability Readiness Review with evidence of viable OpCon, Contracting, Logistics, Resources, Incentives, personnel capabilities, enabling MPTs	Achieving advanced People CMM Level 4 maintainability capabilities such as empowered work groups, mentoring, quantitative performance management and competency-based assets, particularly across key domains.	Advanced, integrated, tested, and exercised full-LC MBS&SE MPTs and Maintainability-other-SQ tradespace analysis
6	Mostly-elaborated maintainability OpCon. with roles, responsibilities, workflows, logistics management plans with budgets, schedules, resources, staffing, infrastructure and enabling MPT choices, V&V and review procedures.	Achieving basic People CMM levels 2 and 3 maintainability practices such as maintainability work environment, competency and career development, and performance management especially in such key areas such as V&V, identification & reduction of technical debt.	Advanced, integrated, tested full-LC Model-Based Software & Systems (MBS&SE) MPTs and Maintainability-other-SQ tradespace analysis tools identified for use, and being individually used and integrated.
5	Convergence, involvement of main maintainability success-critical stakeholders. Some maintainability use cases defined. Rough maintainability OpCon, other success-critical stakeholders, staffing, resource estimates. Preparation for NDI and outsource selections.	In addition, independent maintainability experts participate in project evidence-based decision reviews, identify potential maintainability conflicts with other SQs	Advanced full-lifecycle (full-LC) O&M MPTs and SW/SE MPTs identified for use. Basic MPTs for tradespace analysis among maintainability & other SQs, including TCO being used.
4	Artifacts focused on missions. Primary maintenance options determined, Early involvement of maintainability success-critical stakeholders in elaborating and evaluating maintenance options.	Critical mass of maintainability SysEs with mission SysE capability, coverage of full M-SysE.skills areas, representation of maintainability success-critical-stakeholder organizations.	Advanced O&M MPT capabilities identified for use: Model-Based SW/SE, TCO analysis support. Basic O&M MPT capabilities for modification, repair and V&V: some initial use.
3	Elaboration of mission OpCon, Arch views, lifecycle cost estimation. Key mission, O&M, success-critical stakeholders (SCSHs) identified, some maintainability options explored.	O&M success-critical stakeholders's provide critical mass of maintainability-capable Sys. engrs. Identification of additional. M-critical success-critical stakeholders.	Basic O&M MPT capabilities identified for use, particularly for OpCon, Arch, and Total cost of ownership (TCO) analysis: some initial use.
2	Mission evolution directions and maintainability implications explored. Some mission use cases defined, some O&M options explored.	Highly maintainability-capable SysEs included in Early SysE team.	Initial exploration of O&M MPT options
1	Focus on mission opportunities, needs. Maintainability not yet considered	Awareness of needs for early expertise for maintainability. concurrent engr'g, O&M integration, Life Cycle cost estimation	Focus on O&M MPT options considered

SYSTEMS ENGINEERING Research Center

SIS Maintainability Readiness Levels 3-5

Software-Intensive Systems Maintainability Readiness Framework (SMRF)			
SMR Level	OpCon, Contracting: Missions, Scenarios, Resources, Incentives	Personnel Capabilities and Participation	Enabling Methods, Processes, and Tools (MPTs)
5	Convergence, involvement of main maintainability success-critical stakeholders. Some maintainability use cases defined. Rough maintainability OpCon, other success-critical stakeholders, staffing, resource estimates. Preparation for NDI and outsource selections.	In addition, independent maintainability experts participate in project evidence-based decision reviews, identify potential maintainability conflicts with other SQs	Advanced full-lifecycle (full-LC) O&M MPTs and SW/SE MPTs identified for use. Basic MPTs for tradespace analysis among maintainability & other SQs, including TCO being used.
4	Artifacts focused on missions. Primary maintenance options determined, Early involvement of maintainability success-critical stakeholders in elaborating and evaluating maintenance options.	Critical mass of maintainability SysEs with mission SysE capability, coverage of full M-SysE.skills areas, representation of maintainability success-critical-stakeholder organizations.	Advanced O&M MPT capabilities identified for use: Model-Based SW/SE, TCO analysis support. Basic O&M MPT capabilities for modification, repair and V&V: some initial use.
3	Elaboration of mission OpCon, Arch views, lifecycle cost estimation. Key mission, O&M, success-critical stakeholders (SCSHs) identified, some maintainability options explored.	O&M success-critical stakeholders's provide critical mass of maintainability-capable Sys. engrs. Identification of additional. M-critical success-critical stakeholders.	Basic O&M MPT capabilities identified for use, particularly for OpCon, Arch, and Total cost of ownership (TCO) analysis: some initial use.

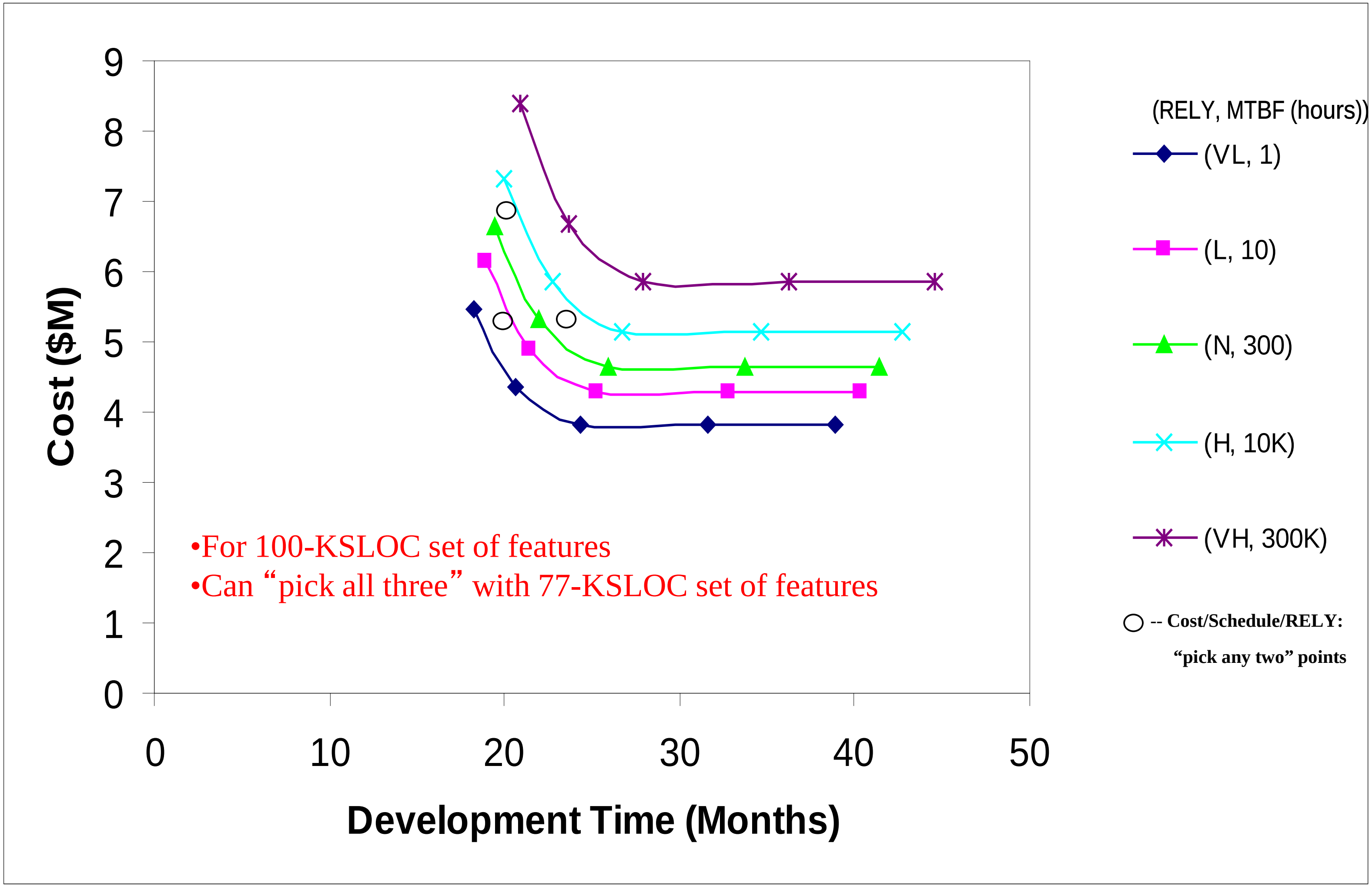
Future Plans

- Explore applications of SQ tradespace analysis to help ensure balanced solutions to new initiatives in cyber security, autonomy, modular open-systems acquisition, internets of things, learning systems, other Third Offset initiatives
- Develop full-lifecycle set-based design MPTs, analyze areas of requirement uncertainty and evolution; life cycle readiness MPTs; extension of Maintainability data analytics
- Continue satellite cost modeling efforts with DoD and Services centers for cost analysis; industry via INCOSE, MORS, and NDIA
- Continue trial application of MPTs with NSWC Carderock, Army ERDC and TARDEC, USAF ASC and SMC, USMC
  Extend transition collaborations with FFRDCs Aerospace, Mitre, SEI

SYSTEMS ENGINEERING
Research Center

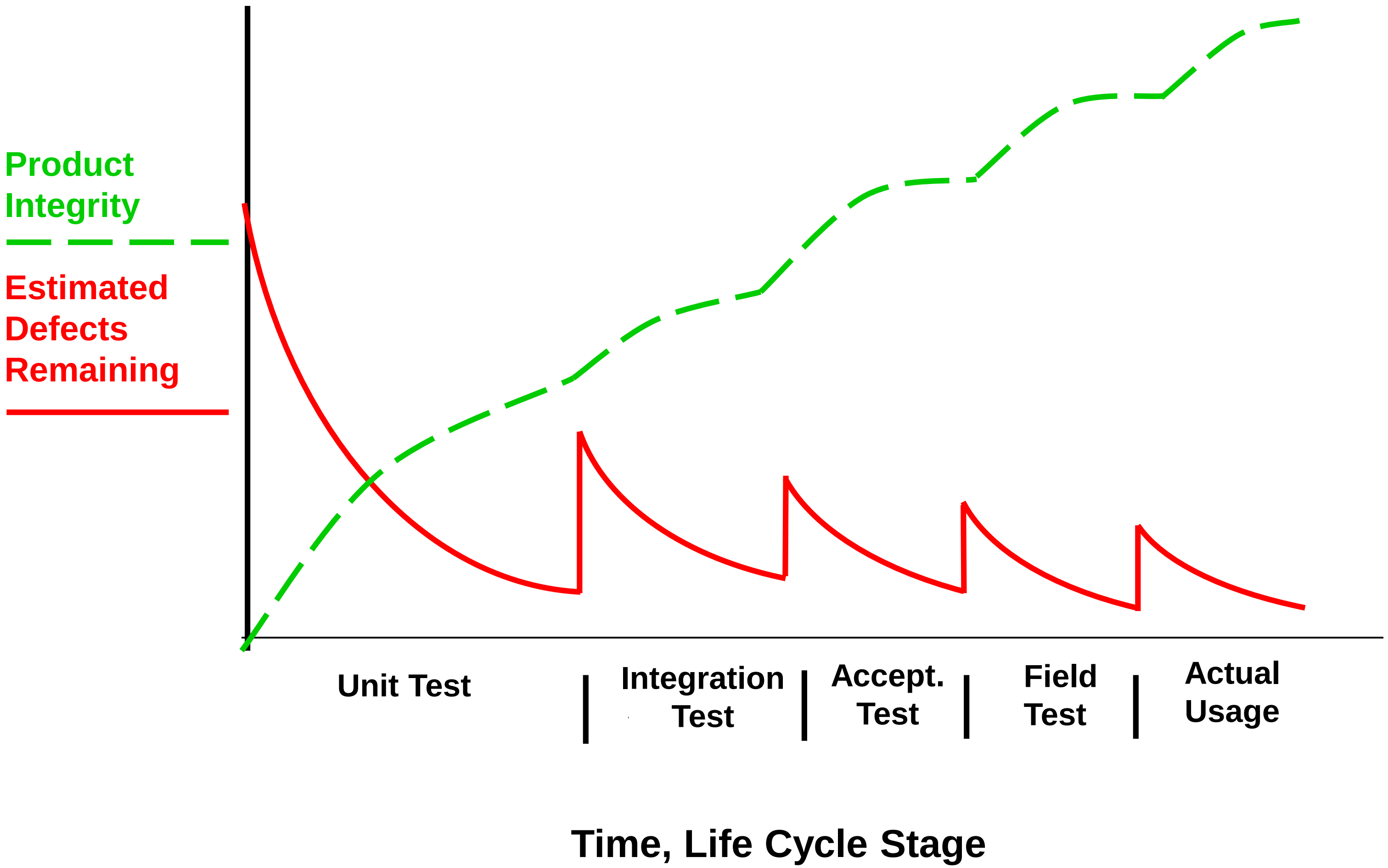
Backup Charts

Referents: Stakeholder Priorities
Cost, Schedule, Reliability, Functionality
COCOMO II Model Results



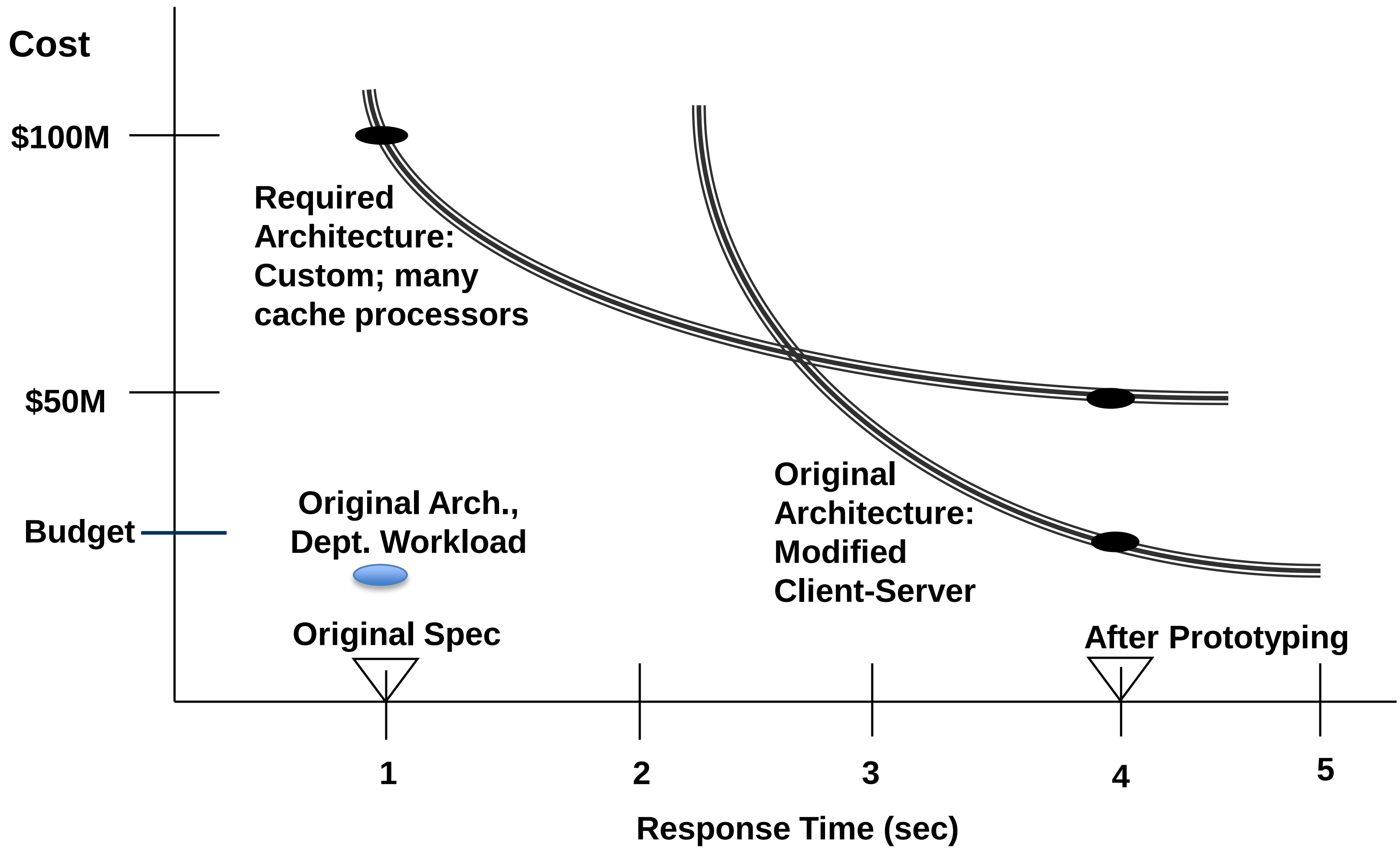
States: Variation by Life Cycle Stage

TRW project defect estimates



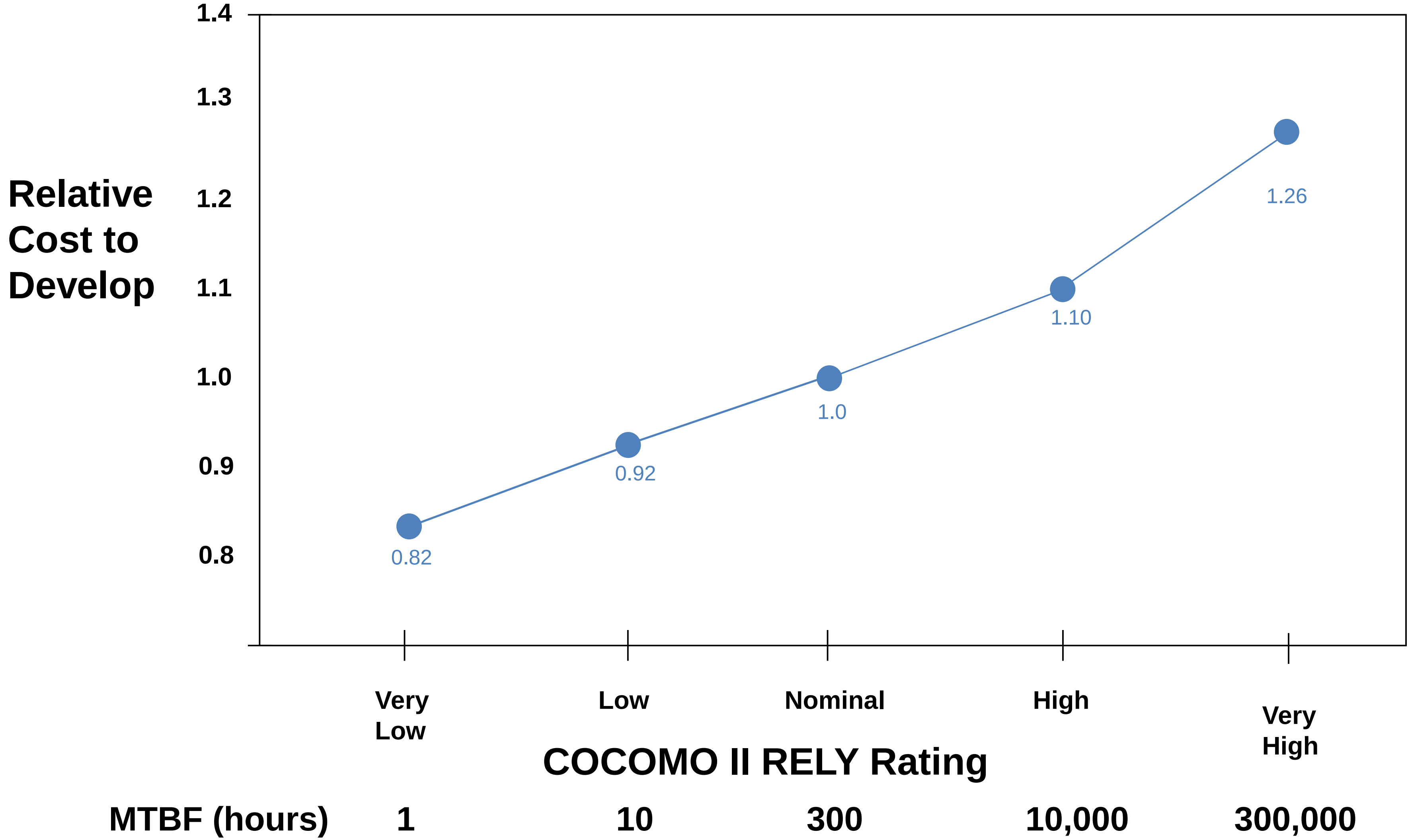
Processes: Cost, Speed Variation by Workload Level

Cost to Process Enterprise Workload vs. Response Time



Relations: SW Development Cost vs. Reliability

Quality is Free: Did Crosby Get it Wrong?



Software Ownership Cost vs. Reliability