System Qualities Ontology, Tradespace and Affordability (SQOTA) Project

Barry Boehm, USC

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

March 2, 2016

© 2016 by USC CSSE.
Published by The Aerospace Corporation with permission.
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
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, Changeability

• 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 illiy 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)
• 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
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
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
<table>
<thead>
<tr>
<th>Flexibility</th>
<th>Dependability</th>
<th>Mission Effectiveness</th>
<th>Resource Utilization</th>
<th>Physical Capability</th>
<th>Cyber Capability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain architecting within domain</td>
<td>Adaptability</td>
<td>Adaptability</td>
<td>Adaptability</td>
<td>Adaptability</td>
<td>Adaptability</td>
<td>Adaptability</td>
</tr>
<tr>
<td>Modularity</td>
<td>Agile methods</td>
<td>Automated I/O validation</td>
<td>Loose coupling</td>
<td></td>
<td>Loose coupling</td>
<td></td>
</tr>
<tr>
<td>Self Adaptive</td>
<td>Service oriented</td>
<td>Automated I/O validation</td>
<td>Sustainability</td>
<td></td>
<td>Modularity</td>
<td></td>
</tr>
<tr>
<td>Smart monitoring</td>
<td>Spare capacity</td>
<td>Product line architectures</td>
<td></td>
<td></td>
<td></td>
<td>Product line architectures</td>
</tr>
<tr>
<td>Spare Capacity</td>
<td>User programmability</td>
<td></td>
<td></td>
<td></td>
<td>Value prioritizing</td>
<td>Service-oriented connectors</td>
</tr>
<tr>
<td>Use software vs. hardware</td>
<td>Versatility</td>
<td></td>
<td></td>
<td></td>
<td>Use software vs. hardware</td>
<td>Use software vs. hardware</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependability</th>
<th>Mission Effectiveness</th>
<th>Resource Utilization</th>
<th>Physical Capability</th>
<th>Cyber Capability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accreditation</td>
<td>Automated aids</td>
<td>Automated aids</td>
<td>Automated aids</td>
<td>Automated aids</td>
<td>Automated aids</td>
</tr>
<tr>
<td>Agile methods assurance</td>
<td>FMEA</td>
<td>Automated I/O validation</td>
<td>Domain architecting within domain</td>
<td>Domain architecting within domain</td>
<td>Automated aids</td>
</tr>
<tr>
<td>Encryption</td>
<td>Multi-level security</td>
<td>Domain architecting within domain</td>
<td>Staffing, Empowering</td>
<td>Staffing, Empowering</td>
<td>Automated aids</td>
</tr>
<tr>
<td>Many options</td>
<td>Survivability</td>
<td>Product line architectures</td>
<td>Staffing, Empowering</td>
<td>Staffing, Empowering</td>
<td>Automated aids</td>
</tr>
<tr>
<td>Multi-domain modifiability</td>
<td>Spare capacity</td>
<td></td>
<td></td>
<td>Value prioritizing</td>
<td>Value prioritizing</td>
</tr>
<tr>
<td>Multi-level security</td>
<td>Total Ownership Cost</td>
<td></td>
<td></td>
<td>Value prioritizing</td>
<td>Value prioritizing</td>
</tr>
<tr>
<td>Self Adaptive defects</td>
<td>Value prioritizing</td>
<td></td>
<td></td>
<td>Value prioritizing</td>
<td>Value prioritizing</td>
</tr>
<tr>
<td>User programmability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mission Effectiveness</th>
<th>Resource Utilization</th>
<th>Physical Capability</th>
<th>Cyber Capability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anti-tamper</td>
<td>Automated aids</td>
<td>Automated aids</td>
<td>Automated aids</td>
<td>Automated aids</td>
</tr>
<tr>
<td>Modularity slowdowns</td>
<td>Armor vs. Weight</td>
<td>Domain architecting within domain</td>
<td>Domain architecting within domain</td>
<td>Automated aids</td>
</tr>
<tr>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Easiest-first development</td>
<td>Staffing, Empowering</td>
<td>Staffing, Empowering</td>
<td>Automated aids</td>
</tr>
<tr>
<td>Versatility vs. Usability</td>
<td>Redundancy</td>
<td>Staffing, Empowering</td>
<td>Staffing, Empowering</td>
<td>Staffing, Empowering</td>
</tr>
<tr>
<td></td>
<td>Scalability</td>
<td>Value prioritizing</td>
<td>Value prioritizing</td>
<td>Value prioritizing</td>
</tr>
<tr>
<td></td>
<td>Spare Capacity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Usability vs. Security</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource Utilization</th>
<th>Physical Capability</th>
<th>Cyber Capability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Methods scalability</td>
<td>Lightweight agility</td>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Multi-domain architecture interoperability conflicts</td>
</tr>
<tr>
<td>Assertion checking overhead</td>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Over-optimizing</td>
<td>Over-optimizing</td>
</tr>
<tr>
<td>Fixed cost contracts</td>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Over-optimizing</td>
<td>Over-optimizing</td>
</tr>
<tr>
<td>Modularity</td>
<td>Over-optimizing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Over-optimizing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spare capacity</td>
<td>Use software vs. hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tight coupling</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical Capability</th>
<th>Cyber Capability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Over-optimizing</td>
<td>Reduced speed of Assertion checking</td>
</tr>
<tr>
<td>Over-optimizing</td>
<td></td>
<td>Reduced speed of connectors, standards compliance</td>
</tr>
<tr>
<td>Tight coupling</td>
<td></td>
<td>Tight vs. Loose coupling</td>
</tr>
<tr>
<td>Use software vs. hardware</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cyber Capability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Encryption interoperability</td>
</tr>
<tr>
<td>Over-optimizing</td>
<td></td>
</tr>
<tr>
<td>Tight coupling</td>
<td></td>
</tr>
<tr>
<td>Use software vs. hardware</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interoperability</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-domain architecture interoperability conflicts</td>
<td>Encryption interoperability</td>
</tr>
<tr>
<td>Over-optimizing</td>
<td></td>
</tr>
<tr>
<td>Tight vs. Loose coupling</td>
<td></td>
</tr>
</tbody>
</table>
Software Development Cost vs. Reliability

<table>
<thead>
<tr>
<th>MTBF (hours)</th>
<th>COCOMO II RELY Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Very Low</td>
</tr>
<tr>
<td>10</td>
<td>Low</td>
</tr>
<tr>
<td>300</td>
<td>Nominal</td>
</tr>
<tr>
<td>10,000</td>
<td>High</td>
</tr>
<tr>
<td>300,000</td>
<td>Very High</td>
</tr>
</tbody>
</table>

Relative Cost to Develop

0.82 0.92 1.0 1.10 1.26

Very Low Low Nominal High Very High
Software Ownership Cost vs. Reliability

Relative Cost to Develop, Maintain, Own and Operate

Operational defect cost at Nominal dependability = Software life cycle cost

Operational defect cost = 0

VL = 2.55
L = 1.52

COCOMO II RELY Rating

MTBF (hours) 1 10 300 10,000 300,000

03-03-2016

GSAW 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
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

---

Move (land)

- Time to Accelerate to Land Cruise (s)
  - 7.50
  - 2.25
  - 0.60

- Max Speed on Grade (mph)
  - 8.00
  - 54.16
  - 90.00

- Land Range at Cruise (miles)
  - 150.47
  - 29.86
  - 600.00

- Satisfy Form Factor
- Move (Water)
- Transportability
- Cost

GSAW 2016
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)
COCOMO II-Based Tradeoff Analysis
Better, Cheaper, Faster: Pick Any Two?

For 100-KSLOC set of features
Can “pick all three” with 77-KSLOC set of features
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

- 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
Costing Insights: COCOMO II Productivity Ranges

Scale Factor Ranges: 10, 100, 1000 KSLOC

Productivity Range

Product Complexity (CPLX)

Analyst Capability (ACAP)

Programmer Capability (PCAP)

Time Constraint (TIME)

Personnel Continuity (PCON)

Required Software Reliability (RELY)

Documentation Match to Life Cycle Needs (DOCU)

Multi-Site Development (SITE)

Applications Experience (AEXP)

Architecture and Risk Resolution (RESL)

Platform Experience (PEXP)

Data Base Size (DATA)

Required Development Schedule (SCED)

Language and Tools Experience (LTEX)

Process Maturity (PMAT)

Storage Constraint (STOR)

Use of Software Tools (TOOL)

Platform Volatility (PVOL)

Development Flexibility (FLEX)

Team Cohesion (TEAM)

Develop for Reuse (RUSE)

Precedentedness (PREC)

Continuous Improvement

Staffing

Teambuilding

GSAW 2016

03-03-2016
COSYSMO Sys Engr Cost Drivers

Teambuilding
- Multisite coordination
- Process capability

Continuous Improvement
- Personnel experience/continuity
- Stakeholder team cohesion
- Personnel/team capability
- Architecture Understanding
- Technology Risk
- Level of Service Requirements
- Requirements Understanding

Staffing
- Documentation
- # and diversity of installations/platforms
- Tool support
- Migration Complexity
- # of recursive levels in the design

Effort Multiplier Ratio (EMR)