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

Barry Boehm, Jo Ann Lane, Sue Koolmanojwong
http://csse.usc.edu
NDIA Systems Engineering Conference
October 25, 2012
Outline

• Affordability definitions, concepts, issues, strategies
  – Addressing both costs and benefits
  – Using life cycle present value
  – Coping with uncertainty: incrementally; pro-actively
  – Coping with multi-stakeholder value diversity
  – Addressing tradeoffs with other -ilities

• An orthogonal framework for improving affordability costs
  – Cost modeling and other insights

• Conclusions
Affordability Definitions

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

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

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

• **Webster:** Keeping within your financial means.
Affordability Concepts

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

Life Cycle Cost Improvement

Achievable Pareto Boundary

Primary Option Space

Current Situation

Increasing Cost-Benefit Improvement Isoquants

Life Cycle Benefits Improvement
## Multi-Stakeholder Value Diversity

### Bank of America Master Net

<table>
<thead>
<tr>
<th>Users</th>
<th>Acquirers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many features</td>
<td>Mission cost/effectiveness</td>
</tr>
<tr>
<td>Changeable requirements</td>
<td>Limited development budget, schedule</td>
</tr>
<tr>
<td>Applications compatibility</td>
<td>Government standards compliance</td>
</tr>
<tr>
<td>High levels of service</td>
<td>Political correctness</td>
</tr>
<tr>
<td>Voice in acquisition</td>
<td>Development visibility and control</td>
</tr>
<tr>
<td>Flexible contract</td>
<td>Rigorous contact</td>
</tr>
<tr>
<td>Early availability</td>
<td>Developers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maintainers</th>
<th>Developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of transition</td>
<td>Flexible contract</td>
</tr>
<tr>
<td>Ease of maintenance</td>
<td>Ease of meeting budget and schedule</td>
</tr>
<tr>
<td>Applications compatibility</td>
<td>Stable requirements</td>
</tr>
<tr>
<td>Voice in acquisition</td>
<td>Freedom of choice: process</td>
</tr>
</tbody>
</table>

PC: Process  
PD: Product  
PP: Property  
S: Success
Outline

• Affordability definitions, concepts, issues, strategies
  – Addressing both costs and benefits
  – Using life cycle present value
  – Coping with uncertainty: incrementally; pro-actively
  – Coping with multi-stakeholder value diversity
  – Addressing tradeoffs with other -ilities

An orthogonal framework for improving affordability costs
  – Cost modeling and other insights

• Conclusions
Affordability and Tradespace Framework

Affordability Improvements and Tradeoffs

- Get the Best from People
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)
- Make Tasks More Efficient
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology
- Eliminate Tasks
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation
- Eliminate Scrap, Rework
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity
- 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

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)
Platform Volatility (PVOL)
Use of Software Tools (TOOL)
Data Base Size (DATA)
Required Development Schedule (SCED)
Platform Experience (PEXP)
Language and Tools Experience (LTEX)
Architecture and Risk Resolution (RESL)
Precedenteness (PREC)
Develop for Reuse (RUSE)
Team Cohesion (TEAM)
Development Flexibility (FLEX)

Scale Factor Ranges: 10, 100, 1000 KSLOC

Staffing
Teambuilding
Continuous Improvement

Productivity Range

October 16, 2012
Copyright © USC-CSSE
COSYSMO Sys Engr Cost Drivers

Teambuilding
- Multisite coordination
- Migration Complexity
- # of recursive levels in the design

Continuous Improvement
- Process capability
- Personnel experience/continuity
- Stakeholder team cohesion

Staffing
- Personnel/team capability
- Architecture Understanding
- Technology Risk
- Level of Service Requirements
- Requirements Understanding

Effort Multiplier Ratio (EMR)
## Tradespace and Affordability Framework

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

The ratio of machine lines of code to a source line of code

Order of Magnitude Increase Every Twenty Years

Trends in Software Expansion (Bernstein, 1997)

Expansion Factor

Machine Instructions
1960
Macro Assembler
1965
High Level Language
1970
Database Manager
1975
On-line
1980
Prototyping
1985
Subsecond Time Sharing
1990
Object Oriented Programming
1995
Large Scale Reuse
2000
Regression Testing
4GL
Small Scale Reuse
75
37.5
30
15
638
475
142
113
81
47
37.5
30
15
3
100
1000
10
1
1960
1965
1970
1975
1980
1985
1990
1995
2000
MBSE:2010

Copyright © USC-CSSE
Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

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

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

- Required Architecture: Custom; many cache processors
- Original Architecture: Commercial DBMS

Original Budget

Original Spec

After Prototyping

Response Time (sec)

1 2 3 4 5
Affordability Improvements and Tradeoffs

Get the Best from People
- Staffing, Incentivizing, Teambuilding
- Facilities, Support Services
- Kaizen (continuous improvement)

Make Tasks More Efficient
- Tools and Automation
- Work and Oversight Streamlining
- Collaboration Technology

Eliminate Tasks
- Lean and Agile Methods
- Task Automation
- Model-Based Product Generation

Eliminate Scrap, Rework
- Early Risk and Defect Elimination
- Evidence-Based Decision Gates
- Modularity Around Sources of Change
- Incremental, Evolutionary Development
- Value-Based, Agile Process Maturity
- Risk-Based Prototyping
- Value-Based Capability Prioritization
- Satisficing vs. Optimizing Performance

Simplify Products (KISS)
- Domain Engineering and Architecture
- Composable Components, Services, COTS
- Legacy System Repurposing

Reuse Components
- Automate Operations Elements
- Design for Maintainability, Evolvability
- Streamline Supply Chain
- Anticipate, Prepare for Change

Reduce Operations, Support Costs

Value- and Architecture-Based Tradeoffs and Balancing
The HMMWV

300,000 Produced, 22 Fielded Versions
Initial draft requirements in 1979, Initial delivery in 1984

Bolt on armor required upgraded suspension, engine, and steering

Additional armor and cupola raise the CG and increase rollovers

Mattracks or wheels

Upper deck space is always at a premium

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

Imbalance in cupola required motorized drive

Suspension and steering for CG shift

Base cab & flatbed with mission modules
Tradespace and Affordability Framework

Affordability Improvements and Tradeoffs

Get the Best from People
- Staffing, Incentivizing, Teambuilding
- Facilities, Support Services
- Kaizen (continuous improvement)

Make Tasks More Efficient
- Tools and Automation
- Work and Oversight Streamlining
- Collaboration Technology

Eliminate Tasks
- Lean and Agile Methods
- Task Automation
- Model-Based Product Generation

Eliminate Scrap, Rework
- Early Risk and Defect Elimination
- Evidence-Based Decision Gates
- Modularity Around Sources of Change
- Incremental, Evolutionary Development
- Value-Based, Agile Process Maturity

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
Post-Acquisition Costs Dominate (%O&M)

• Hardware [Redman 2008]
  – 12% -- Missiles (average)
  – 60% -- Ships (average)
  – 78% -- Aircraft (F-16)
  – 84% -- Ground vehicles (Bradley)

• Software [Koskinen 2010]
  – 75-90% -- Business, Command-Control
  – 50-80% -- Complex platforms as above
  – 10-30% -- Simple embedded software

• Apply lack-of-flexibility factor to O&M component
Tradespace and Affordability Framework

**Affordability Improvements and Tradeoffs**

- Get the Best from People
  - Staffing, Incentivizing, Teambuilding
  - Facilities, Support Services
  - Kaizen (continuous improvement)

- Make Tasks More Efficient
  - Tools and Automation
  - Work and Oversight Streamlining
  - Collaboration Technology

- Eliminate Tasks
  - Lean and Agile Methods
  - Task Automation
  - Model-Based Product Generation

- Eliminate Scrap, Rework
  - Early Risk and Defect Elimination
  - Evidence-Based Decision Gates
  - Modularity Around Sources of Change
  - Incremental, Evolutionary Development
  - Value-Based, Agile Process Maturity

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

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

(RELY, MTBF (hours))

- (VL, 1)
- (L, 10)
- (N, 300)
- (H, 10K)
- (VH, 300K)

Cost/Schedule/RELY: “pick any two” points
A Value-Priority Tradeoff Equalizer

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

• Affordability increasingly competition-critical
  – Need to balance cost, schedule, performance, functionality

• Orthogonal framework helps tailor improvements
  – Getting the best from people
  – Making tasks more efficient
  – Eliminating tasks
  – Eliminating scrap and rework
  – Simplifying products
  – Reusing assets
  – Reducing operations and support costs
  – Value- and architecture-based tradeoffs and balancing

• No one-size-fits-all solution
Backup Charts
Agile and Plan-Driven Home Grounds: Five Critical Decision Factors

- Size, Criticality, Dynamism, Personnel, Culture
Architected Agile Approach

- Uses Scrum of Scrums approach
  - Up to 10 Scrum teams of 10 people each
  - Has worked for distributed international teams
  - Going to three levels generally infeasible

- General approach shown below
  - Often tailored to special circumstances
COCOMO II. 2000 Productivity Ranges

Scale Factor Ranges: 10, 100, 1000 KSLOC

- Development Flexibility (FLEX)
- Team Cohesion (TEAM)
- Develop for Reuse (RUSE)
- Precedentedness (PREC)
- 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)
- Applications Experience (AEXP)
- Multi-Site Development (SITE)
- Documentation Match to Life Cycle Needs (DOCU)
- Required Software Reliability (RELY)
- Personnel Continuity (PCON)
- Time Constraint (TIME)
- Programmer Capability (PCAP)
- Analyst Capability (ACAP)
- Product Complexity (CPLX)

Scale Factor Ranges: 10, 100, 1000 KSLOC

October 16, 2012

Copyright © USC-CSSE
Value-Based Testing: Empirical Data and ROI

— LiGuo Huang, ISESE 2005

(a)

% of Value for Correct Customer Billing

(b)

Return On Investment (ROI)

 automated test generation (ATG) tool
- all tests have equal value

Automated test generation (ATG) tool

Copyright © USC-CSSE
Value-Neutral Defect Fixing Is Even Worse

Automated test generation tool
- all tests have equal value

Pareto 80-20 Business Value

Value-neutral defect fixing:
Quickly reduce # of defects

% of Value for Correct Customer Billing

Customer Type

Copyright © USC-CSSE 2012
Reuse at HP’s Queensferry Telecommunication Division

Time to Market (months)

<table>
<thead>
<tr>
<th>Year</th>
<th>Non-reuse Project</th>
<th>Reuse project</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>60</td>
<td>50</td>
</tr>
<tr>
<td>88</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>89</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>90</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>91</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>92</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Product Line Engineering and Management

Systems Product Line Flexibility Value Model

Welcome SERC Collaborator

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

Product Line Percentages Relative Costs of Reuse (%)
Unique % 40 Relative Cost of Reuse for Adapted 40
Adapted % 30 Relative Cost of Reuse for Reused 5
Reused % 30

Investment Cost
Relative Cost of Developing for PL Flexibility via Reuse 1.7

Results

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

Return on Investment

| Return on Investment | -1.3 | 0.1 | 1.6 | 3.0 | 4.5 | 5.9 | 7.3 |
| Product # | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
Overfocus on Acquisition Cost
C4ISR Contracts: Nominal-case requirements; 90 days to PDR

TRW Project B
1005 SPR’s

TRW Project A
373 SPR’s

Major Rework Sources:
- Off-Nominal Architecture-Breakers
- A - Network Failover
- B - Extra-Long Messages

% of Cost to Fix SPR’s vs % of Software Problem Reports (SPR’s)
When investments made in architecture, average time for change order becomes relatively stable over time...

Relative* Total Ownership Cost (TOC)

* Cumulative architecting and rework effort relative to initial development effort
Utilities in Tradespace Exploration: MIT

Enabling Construct: Tradespace Networks

More changeable (i.e., including flexible, adaptable, scalable, and modifiable)
Colored by outdegree

Enabling Construct: Epochs and Eras

Changeability

Survivability

Value Robustness

Set of Metrics

Enabling Construct: Tradespace Networks

More changeable (i.e., including flexible, adaptable, scalable, and modifiable)
Colored by outdegree

Enabling Construct: Epochs and Eras

Changeability

Survivability

Value Robustness

Set of Metrics

<table>
<thead>
<tr>
<th>Value Aspect</th>
<th>Acronym</th>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness to change</td>
<td>NPI</td>
<td>Normalized Pareto Index</td>
<td>% of designs that are Pareto efficient after modification</td>
</tr>
<tr>
<td>Robustness to</td>
<td>FNP</td>
<td>Fuzzy Normalized Pareto Index</td>
<td>Above, with change from Pareto front allowed</td>
</tr>
<tr>
<td>&quot;change&quot;</td>
<td>FNP</td>
<td>Fuzzy Normalized Parity</td>
<td>Above, with change from Pareto front allowed</td>
</tr>
<tr>
<td>&quot;Value&quot; of change</td>
<td>IFP</td>
<td>Fuzzy Parity Shift</td>
<td>Difference in FNP before and after transition</td>
</tr>
<tr>
<td>&quot;Value&quot; of change</td>
<td>AR</td>
<td>Available Range</td>
<td># of designs that are feasible in a certain range after transition</td>
</tr>
<tr>
<td>Degree of change</td>
<td>OD</td>
<td>Outdegree</td>
<td># of outgoing transition arcs from a design</td>
</tr>
<tr>
<td>Degree of change</td>
<td>PD</td>
<td>Path Degree</td>
<td>Above, considering only arcs below a certain threshold</td>
</tr>
<tr>
<td>Survivability</td>
<td>TWAUL</td>
<td>Time-weighted Average Utility Loss</td>
<td>Measure of central tendency of raise of design over time for a design as a result of exponential distribution</td>
</tr>
<tr>
<td>Survivability</td>
<td>AT</td>
<td>Threshold Availability</td>
<td>% of lifetime for which range delivers value above assumed acceptable level before redesign and after a distribution</td>
</tr>
</tbody>
</table>
# Architecture-Based Attribute Trades:

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

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

COCOMO II:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>12</th>
<th>22</th>
<th>34</th>
<th>54</th>
<th>Added % test time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COQUALMO:

<table>
<thead>
<tr>
<th></th>
<th>1.0</th>
<th>.475</th>
<th>.24</th>
<th>.125</th>
<th>0.06</th>
<th>P(L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early Startup:</td>
<td>.33</td>
<td>.19</td>
<td>.11</td>
<td>.06</td>
<td>.03</td>
<td>S(L)</td>
</tr>
<tr>
<td>Commercial:</td>
<td>1.0</td>
<td>.56</td>
<td>.32</td>
<td>.18</td>
<td>.10</td>
<td>S(L)</td>
</tr>
<tr>
<td>High Finance:</td>
<td>3.0</td>
<td>1.68</td>
<td>.96</td>
<td>.54</td>
<td>.30</td>
<td>S(L)</td>
</tr>
<tr>
<td>Market Risk:</td>
<td>.008</td>
<td>.027</td>
<td>.09</td>
<td>.30</td>
<td>1.0</td>
<td>RE</td>
</tr>
</tbody>
</table>

Combined Risk Exposure

RE = P(L) * S(L)
## Magnitude of Overrun Problem: DoD

### Analysis of U.S. Defense Dept.
**Major Defense Acquisition Program Portfolios**  
*Fiscal 2009 dollars*

<table>
<thead>
<tr>
<th>Portfolio size</th>
<th>2003</th>
<th>2007</th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of programs</td>
<td>77</td>
<td>95</td>
<td>96</td>
</tr>
<tr>
<td>Total planned commitments</td>
<td>$1.2 trillion</td>
<td>$1.6 trillion</td>
<td>$1.6 trillion</td>
</tr>
<tr>
<td>Commitments outstanding</td>
<td>$724.2 billion</td>
<td>$875.2 billion</td>
<td>$786.3 billion</td>
</tr>
</tbody>
</table>

### Portfolio indicators

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
<th>2007</th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change to total RDT&amp;E* costs from first estimate</td>
<td>37%</td>
<td>40%</td>
<td>42%</td>
</tr>
<tr>
<td>Change to total acquisition cost from first estimate</td>
<td>19%</td>
<td>26%</td>
<td>25%</td>
</tr>
<tr>
<td>Total acquisition cost growth</td>
<td>$183 billion</td>
<td>$301.3 billion</td>
<td>$296.4 billion</td>
</tr>
<tr>
<td>Share of programs with 25% increase in program acquisition unit cost growth</td>
<td>41%</td>
<td>44%</td>
<td>42%</td>
</tr>
<tr>
<td>Average schedule delay in delivering initial capabilities</td>
<td>18 months</td>
<td>21 months</td>
<td>22 months</td>
</tr>
</tbody>
</table>

# Magnitude of Overrun Problem: Standish Surveys of Commercial Projects

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Within budget and schedule</td>
<td>28</td>
<td>34</td>
<td>29</td>
<td>35</td>
<td>32</td>
</tr>
<tr>
<td>Prematurely cancelled</td>
<td>23</td>
<td>15</td>
<td>18</td>
<td>19</td>
<td>24</td>
</tr>
<tr>
<td>Budget or schedule overrun</td>
<td>49</td>
<td>51</td>
<td>53</td>
<td>46</td>
<td>44</td>
</tr>
</tbody>
</table>
Some Frequent Overrun Causes

- Conspiracy of Optimism
- Effects of First Budget Shortfall
  - System Engineering
- Decoupling of Technical and Cost Analysis
  - Overfocus on Performance, Security, Functionality
- Overfocus on Acquisition Cost
- Assumption of Stability
- Total vs. Incremental Commitment
The Conspiracy of Optimism and The Cone of Uncertainty
Effects of First Budget Shortfall: Added Cost of Weak System Engineering
Calibration of COCOMO II Architecture and Risk Resolution (RESL) factor to 161 project data points

Graph showing the relationship between software product size (in KSLOC) and the percentage of added cost, with points indicating increased costs as product size increases.
Assumption of Stability vs. Rapid Change
– Need evolutionary/incremental vs. one-shot development

Uncertainties in competition, technology, organizations, mission priorities

Phases and Milestones

Relative Cost Range

Feasibility
Plans and Rqts.
Product Design
Detail Design
Devel. and Test
Accepted Software

Copyright © USC-CSSE