



iTAP Methods, Processes and Tools

Ray Madachy
6th Annual SERC Sponsor Research Review
December 4, 2014
Georgetown University
School of Continuing Studies
640 Massachusetts Ave NW,
Washington, DC

www.sercuarc.org

- End Vision
- Status
- Phase 4 Plans
 - Task 2: iTAP Methods and Tools Piloting and Refinement
 - Task 3: Next-Generation, Full-Coverage Cost Estimation Model Ensembles

- Total Ownership Cost (TOC) modeling to enable affordability tradeoffs with integrated software-hardware-human factors
- Current shortfalls forilities tradespace analysis
 - Models/tools are incomplete wrt/ TOC phases, activities, disciplines, SoS aspects
 - No integration with physical design space analysis tools, system modeling, or each other
- New aspects
 - Integrated costing of systems, software, hardware and human factors across full lifecycle operations
 - Extensions and consolidations for DoD application domains
 - Tool interoperability and tailorability (service-oriented)
- Can improve affordability-related decisions across all joint services

- Research and development
 - Create new ensemble of cost models with DoD stakeholders for broader coverage.
 - Enable interoperability with other toolsets and researchers (plug and play)
- Piloting and refinement
 - Engaging with DoD organizations to pilot the TOC methods, process and tools (MPTs); then refine them based on the results of the pilot applications

- Extended parametric cost models for breadth of engineering disciplines to include systems engineering, software engineering and hardware.
- Improved TOC capabilities by adding lifecycle maintenance models.
- Added Monte Carlo risk analysis for subset of cost parameters in integrated SE/SW/HW cost model.
- Initial extensions of general cost models for DoD system types starting with space systems and ships.
- Developed web service for Orthogonal Defect Classification Constructive Quality Model (ODC COQUALMO) supporting tool interoperability (costing in the cloud).
- Successful piloting and follow-on extensions of product line model at NAVAIR.

- Extending models and tools to analyze TOC for a family of systems. The value of investing in product-line flexibility using Return-On-Investment (ROI) and TOC is assessed with parametric models adapted from the Constructive Product Line Investment Model (COPLIMO).
- Models are implemented in separate tools for 1) System-level product line flexibility investment model and 2) Software product line flexibility investment model. The detailed software model includes schedule time with NPV calculations.

Example Product Line TOC and ROI



Open Save

System Type: Aircraft, Ground Vehicle, **Missile**, Ship

System Costs

Average Product Development Cost (Burdened \$M) Ownership Time (Years)

Annual Change Cost (% of Development Cost) Interest Rate (Annual %)

Product Line Percentages

Unique % Relative Cost of Reuse for Adapted

Adapted % Relative Cost of Reuse for Reused

Reused %

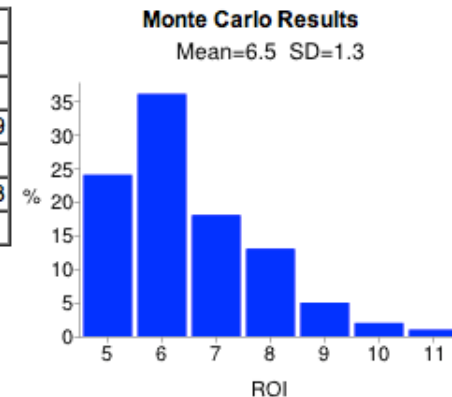
Investment Cost

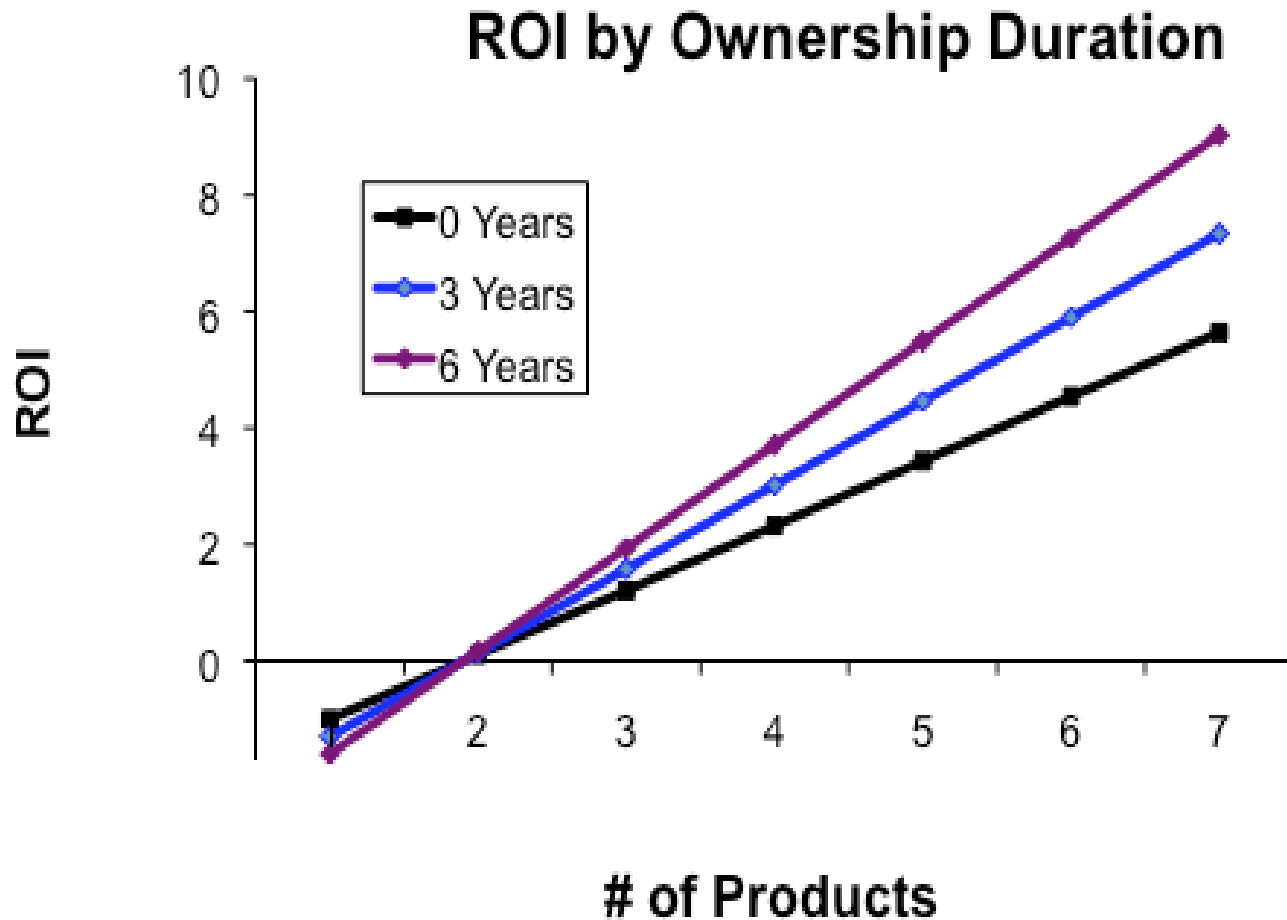
Relative Cost of Developing for PL Flexibility via Reuse

Calculate Monte Carlo

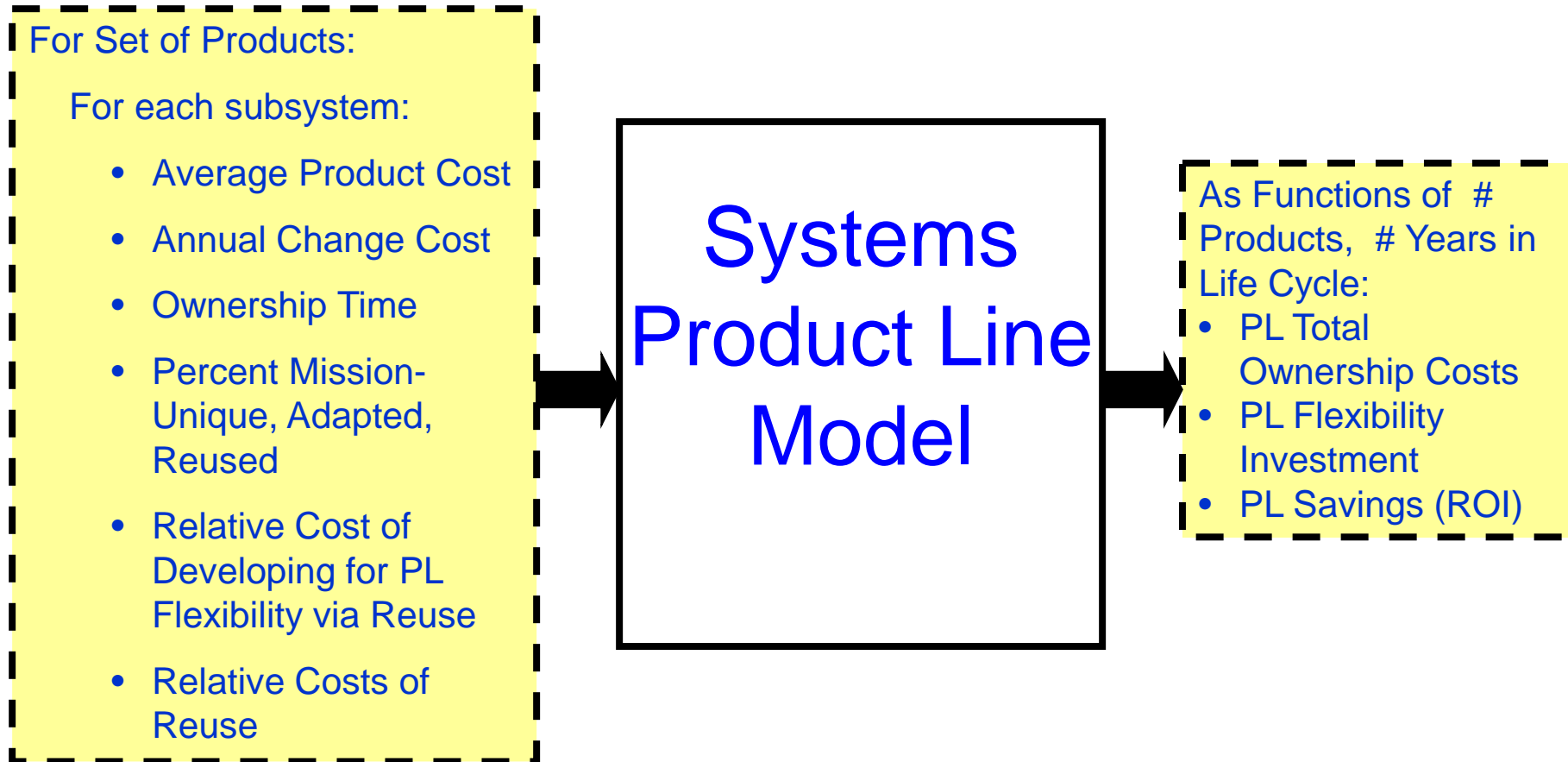
Means

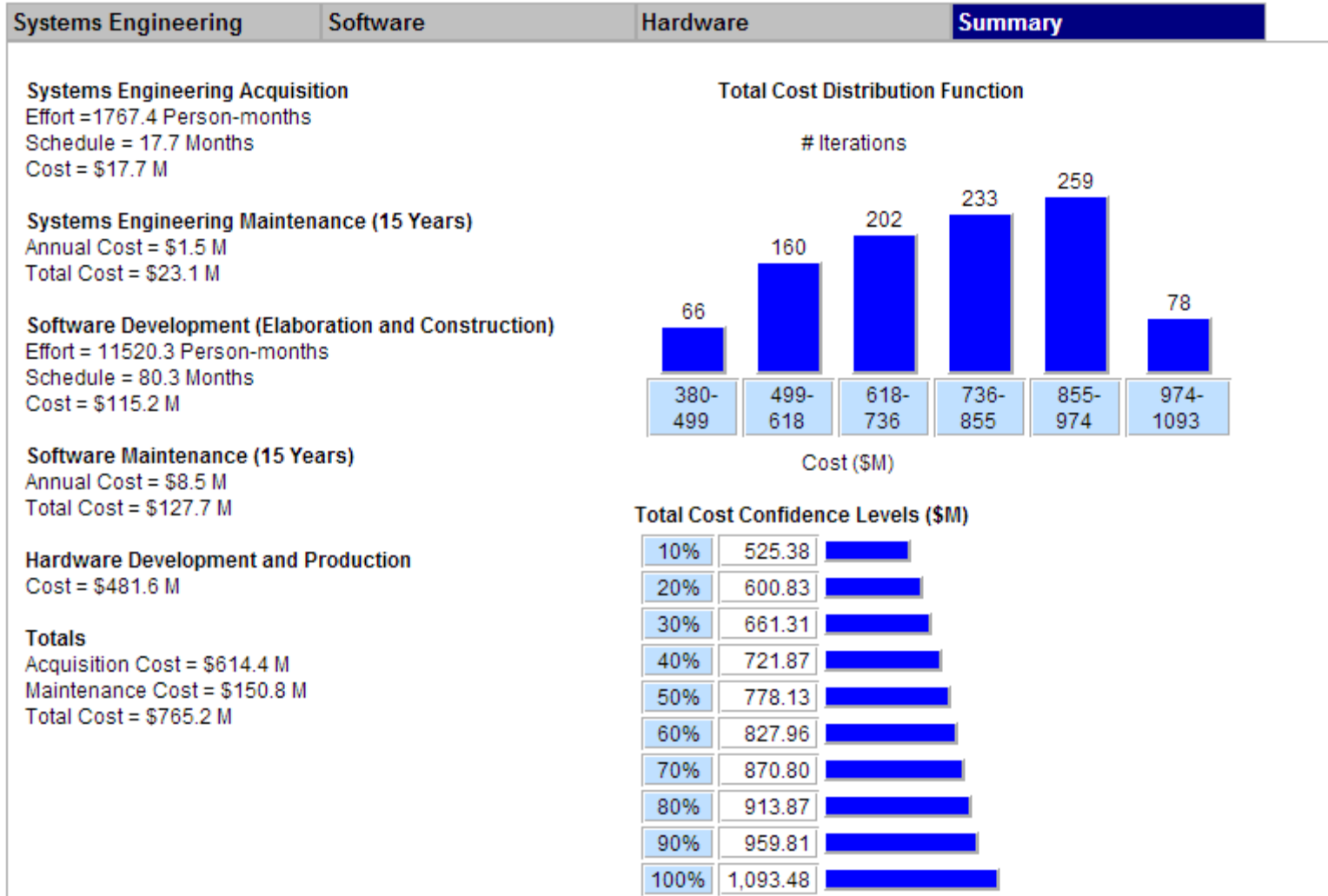
# of Products	1	2	3	4	5	6	7
Development Cost (\$M)	\$7.1	\$2.7	\$2.7	\$2.7	\$2.7	\$2.7	\$2.7
Ownership Cost (\$M)	\$0.9	\$0.3	\$0.3	\$0.3	\$0.3	\$0.3	\$0.3
Cum. PL Cost (\$M)	\$8.0	\$10.9	\$13.9	\$16.9	\$19.9	\$22.9	\$25.9
PL Flexibility Investment (\$M)	\$2.1	\$0	\$0	\$0	\$0	\$0	\$0
PL Effort Savings	(\$2.4)	\$0.3	\$2.9	\$5.5	\$8.1	\$10.7	\$13.3
Return on Investment	-1.12	0.12	1.36	2.60	3.84	5.08	6.32





- Multi-mission, multi-platform needs call for extension of the top-level COPLIMO model to handle subsystems. Immediate pilot applications include:
 - NAVAIR avionics software product line modeling
 - USAF/SMC spacecraft and ground systems cost model development
- Each subsystem has respective cost factors and product line characteristics including
 - Fractions of system fully reusable, partially reusable and cost of developing them for reuse
 - Fraction of system variabilities and cost of development
 - System lifetime and rates of change





- The ODC COQUALMO and Product Line TOC models/tools have been enhanced for interoperability and tailorability.
- Adaptations to the web-based tools enable other toolsets to plug-in, so their analyses can be cross-pollinated with cost, schedule and quality dimensions.
 - External applications can automatically send input parameters and/or files and receive results in lieu of manual user sessions.
 - Ability to modify or add internal model parameters for different scenarios (e.g. effort, schedule and quality calibration parameters; phase/activity distributions for effort and schedule; defect type distributions, etc.).
 - APIs demonstrated for multiple languages and platforms.

- Collaboration with AFIT for a joint Intelligence, Surveillance and Reconnaissance (ISR) mission application involving heterogenous teams of autonomous and cooperative agents.
- NPS will provide cost modeling expertise, tools and Monterey Phoenix (MP) modeling support. A focus will be on translations between models/tools in MBSE, specifically mapping architectural elements into cost model inputs.
- Approach
 - Develop a baseline operational and system architecture to capture a set of military scenarios.
 - Transition the baseline architecture to the MP environment.
 - Utilize the executable architecture modeling framework of MP to perform automated assertion checking and find counterexamples of behavior that violate the expected system's correctness.
 - Operational scenarios will be cycled through the MP modeling process, whereby alternate events are captured for each actor in each scenario. This will produce a superset of scenario variants from the behavior models, suitable for input to tradespace analysis and cost models.
 - Design and demonstrate an ISR UAV tradespace.
 - Develop cost model interfaces for components of the architecture in order to evaluate cost effectiveness in an uncertain future environment.

- Continue extending the scope and tradespace interoperability of cost models and tools from previous phases.
- Cost modeling will engage domain experts for Delphi estimates, evolve baseline definitions of cost driver parameters and rating scales for use in data collection, gather empirical data and determine areas needing further research to account for differences between estimated and actual costs.
 - Prototype cost models and tools will be extended accordingly for piloting and refinement.
- For tool interoperability we will integrate cost models in different ways with MBSE architectural modeling approaches and as web services. We will also automate systems and software risk advisors that operate in conjunction with the cost models.
- NPS will provide domain expertise for SysML cost model integration with Georgia Tech and USC to add software cost model formulas and the risk assessment capabilities.
 - This is also allied with Task 2 where we will assess Monterey Phoenix (MP) for automatically providing cost information from architectural models. MP will extract software sizing cost model inputs to compute costs, and we will assess mapping MP architectural elements into systems engineering cost model inputs.

- Monterey Phoenix (MP) is approach to formal software and system specification based on behavior models
- A view on the architecture model as a high level description of possible behaviors of subsystems and interactions between subsystems
- The emphasis on specifying the interaction between the system and its environment
- The behavior composition operations support architecture reuse and refinement toward design and implementation models
- Executable architecture models provide for system architecture testing and verification with tools
- See <http://wiki.nps.edu/display/MP>

MP Pipe/Filter Architecture Pattern Example



SCHEMA simple_message_flow

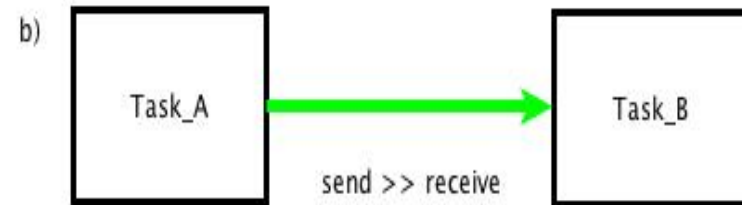
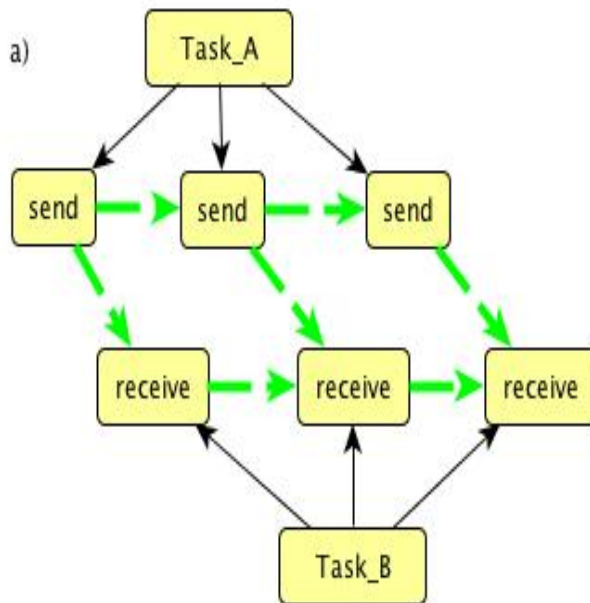
ROOT Task_A: (* send *);

ROOT Task_B: (* receive *);

COORDINATE \$x: send FROM Task_A,

\$y: receive FROM Task_B

DO ADD \$x PRECEDES \$y OD;



a) Example of composed event trace

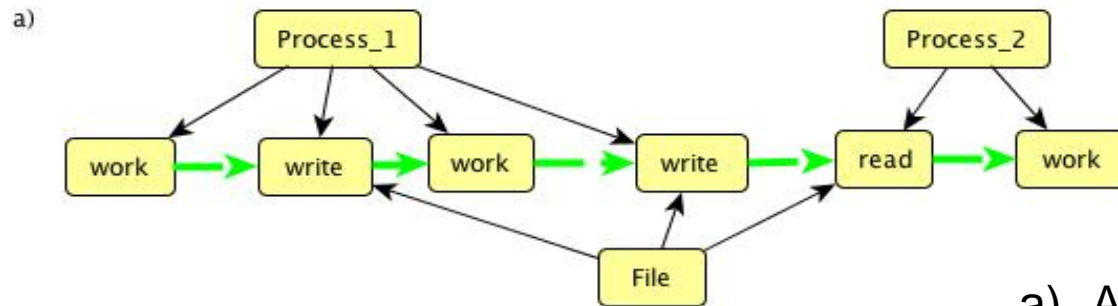
b) An architecture view for the schema

Data items are represented by actions that may be performed on that data

SCHEMA Data_flow

```

ROOT Process_1:  (* work write *);
ROOT Process_2:  (* ( read | work ) *);
ROOT File:       (* write *) (* read *);
Process_1, File  SHARE ALL write;
Process_2, File  SHARE ALL read;
    
```



a) An example of composed event trace



b) An architecture view

- K. Giammarco and M. Auguston, “Well, You didn’t Say not to! A Formal Systems Engineering Approach to Teaching an Unruly Architecture Good Behavior”, Complex Adaptive Systems Conference, 2013
- M. Auguston and C. Whitcomb, "Behavior Models and Composition for Software and Systems Architecture", ICSSEA 2012, 24th International Conference on Software & Systems Engineering and their Applications, 2012
- R. Madachy and R. Valerdi, “Automating Systems Engineering Risk Assessment”, Proceedings of the 8th Annual Conference on Systems Engineering Research, 2010
- R. Madachy, *Heuristic Risk Assessment Using Cost Factors*, IEEE Software, May 1997





System Cost Model Suite

Options

Monte Carlo Risk ▾

Systems Engineering

Software

Hardware

Summary

Constructive Systems Engineering Cost Model (COSYSMO)

System Size

	Easy	Nominal	Difficult
# of System Requirements	<input type="text" value="120"/>	<input type="text" value="185"/>	<input type="text" value="48"/>
# of System Interfaces	<input type="text" value="12"/>	<input type="text" value="67"/>	<input type="text" value="45"/>
# of Algorithms	<input type="text" value="19"/>	<input type="text" value="125"/>	<input type="text" value="58"/>
# of Operational Scenarios	<input type="text" value="3"/>	<input type="text" value="14"/>	<input type="text" value="8"/>

System Cost Drivers

Requirements Understanding	<input type="button" value="High"/> ▾	Documentation	<input type="button" value="Nominal"/> ▾	Personnel Experience/Continuity	<input type="button" value="Nominal"/> ▾
Architecture Understanding	<input type="button" value="High"/> ▾	# and Diversity of Installations/Platforms	<input type="button" value="Very High"/> ▾	Process Capability	<input type="button" value="Nominal"/> ▾
Level of Service Requirements	<input type="button" value="Very High"/> ▾	# of Recursive Levels in the Design	<input type="button" value="Nominal"/> ▾	Multisite Coordination	<input type="button" value="Nominal"/> ▾
Migration Complexity	<input type="button" value="Nominal"/> ▾	Stakeholder Team Cohesion	<input type="button" value="Nominal"/> ▾	Tool Support	<input type="button" value="Nominal"/> ▾
Technology Risk	<input type="button" value="Nominal"/> ▾	Personnel/Team Capability	<input type="button" value="Nominal"/> ▾		

Maintenance ▾

System Labor Rates

Cost per Person-Month (Dollars)

Results

Systems Engineering

Effort = 1767.9 Person-months

Schedule = 17.7 Months

Cost = \$17679187

Total Size = 2650 Equivalent Nominal Requirements

Acquisition Effort Distribution (Person-Months)

Phase / Activity	Conceptualize	Develop	Operational Test and Evaluation	Transition to Operation
Acquisition and Supply	34.7	63.1	16.1	9.9
Technical Management	66.1	114.2	75.1	45.1
System Design	180.3	212.2	90.2	47.7
Product Realization	34.5	79.6	84.9	66.3
Product Evaluation	98.6	148.0	219.2	82.2

Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_07_42_42_618469.txt

Advanced Missions Cost Model (AMCM)

[Quantity](#)

[Dry Weight \(lb.\)](#)

[Mission Type](#) ▼

[IOC Year](#)

[Block Number](#)

[Difficulty](#) ▼

Results

Hardware Development and Production
Total Cost = \$608 M

This is a simple advanced missions cost model (AMCM) for quick turnaround, rough-order-of-magnitude estimating. The model can be used for estimating the development and production cost of spacecraft, space transportation systems, aircraft, missiles, ships, and land vehicles. Initial model provided courtesy of NASA with extensions by NPS.

Systems Engineering	Software	Hardware	Summary
Systems Engineering Acquisition Effort = 1767.9 Person-months Schedule = 17.7 Months Cost = \$17.7 M			
Software Development (Elaboration and Construction) Effort = 10344.6 Person-months Schedule = 77.5 Months Cost = \$103.4 M			
Hardware Development and Production Cost = \$608 M			
Total System Cost = \$744.4 M			
Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_07_42_42_618469.txt			

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Results

Systems Engineering

Effort = 1767.4 Person-months

Schedule = 17.7 Months

Cost = \$17673532

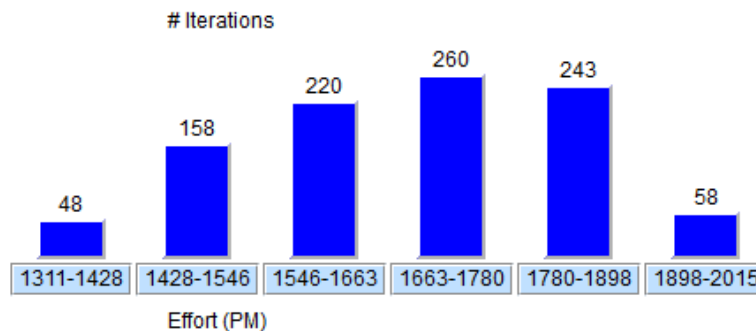
Total Size = 2650 Equivalent Nominal Requirements

Acquisition Effort Distribution (Person-Months)

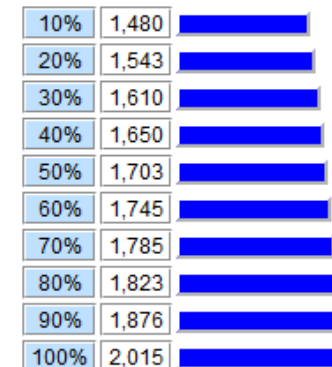
Phase / Activity	Conceptualize	Develop	Operational Test and Evaluation	Transition to Operation
Acquisition and Supply	34.6	63.1	16.1	9.9
Technical Management	66.1	114.2	75.1	45.1
System Design	180.3	212.1	90.1	47.7
Product Realization	34.5	79.5	84.8	66.3
Product Evaluation	98.6	147.9	219.2	82.2

Monte Carlo Results

Systems Engineering Effort Distribution Function



Systems Engineering Effort Confidence Levels



Your output file is http://diana.nps.edu/~madachv/tools/data/cost_model_suiteSeptember_17_2013_07_55_02_565284.txt

Systems Engineering	Software	Hardware	Summary
Advanced Missions Cost Model (AMCM)			
Quantity	<input type="text" value="1"/>		
Dry Weight (lb.)	Distribution <input type="text" value="Uniform"/> ▼	Min <input type="text" value="900000"/>	Max <input type="text" value="1500000"/>
Mission Type	<input type="text" value="Ship - Amphib Assault"/> ▼		
IOC Year	<input type="text" value="2013"/>		
Block Number	<input type="text" value="1"/>		
Difficulty	<input type="text" value="Average"/> ▼		
<input type="button" value="Calculate"/>			

Systems Engineering
Software
Hardware
Summary

Systems Engineering Acquisition

Effort = 1767.4 Person-months
 Schedule = 17.7 Months
 Cost = \$17.7 M

Software Development (Elaboration and Construction)

Effort = 10344.6 Person-months
 Schedule = 77.5 Months
 Cost = \$103.4 M

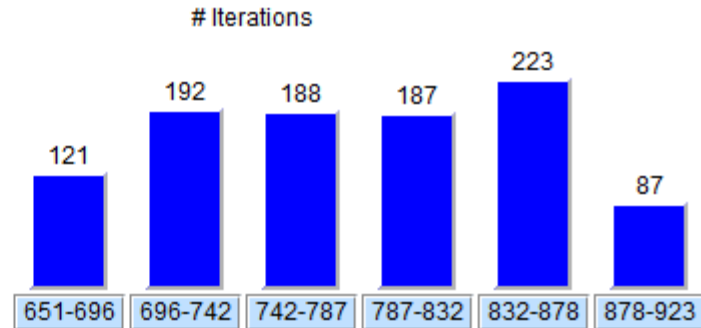
Hardware Development and Production

Cost = \$685 M

Total

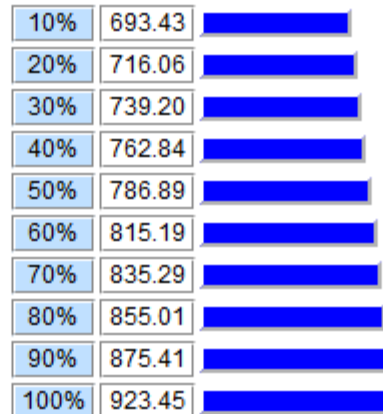
System Cost = \$806.1 M

Total Cost Distribution Function



Cost (\$M)

Total Cost Confidence Levels (\$M)



Systems Engineering Software Hardware Summary

Constructive Systems Engineering Cost Model (COSYSMO)

System Size

	Easy	Nominal	Difficult
# of System Requirements	120	185	48
# of System Interfaces	12	67	45
# of Algorithms	19	125	58
# of Operational Scenarios	3	14	8

System Cost Drivers

Requirements Understanding	High	Documentation	Nominal	Personnel Experience/Continuity	Nominal
Architecture Understanding	High	# and Diversity of Installations/Platforms	Very High	Process Capability	Nominal
Level of Service Requirements	Very High	# of Recursive Levels in the Design	Nominal	Multisite Coordination	Nominal
Migration Complexity	Nominal	Stakeholder Team Cohesion	Nominal	Tool Support	Nominal
Technology Risk	Nominal	Personnel/Team Capability	Nominal		

Maintenance On Annual Change % Maintenance Duration (Years)

System Labor Rates

Cost per Person-Month (Dollars)

Results

Systems Engineering

Effort = 1767.9 Person-months

Schedule = 17.7 Months

Cost = \$17679187

Total Size = 2650 Equivalent Nominal Requirements

Acquisition Effort Distribution (Person-Months)

Phase / Activity	Conceptualize	Develop	Operational Test and Evaluation	Transition to Operation
Acquisition and Supply	34.7	63.1	16.1	9.9
Technical Management	66.1	114.2	75.1	45.1
System Design	180.3	212.2	90.2	47.7
Product Realization	34.5	79.6	84.9	66.3
Product Evaluation	98.6	148.0	219.2	82.2

Maintenance

Annual Maintenance Effort = 154.0 Person-Months

Annual Maintenance Cost = \$1539792

Total Maintenance Cost = \$23096893

Your output file is [http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember 17 2013 08 06 07 160035.txt](http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_08_06_07_160035.txt)

Systems Engineering **Software** Hardware Summary

Constructive Cost Model (COCOMO II)

Software Size Sizing Method

	<u>SLOC</u>	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	<input type="text" value="850000"/>						
Reused	<input type="text" value="225000"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="4"/>		
Modified	<input type="text" value="400000"/>	<input type="text" value="10"/>	<input type="text" value="15"/>	<input type="text" value="60"/>	<input type="text" value="4"/>	<input type="text" value="20"/>	<input type="text" value=".4"/>

Software Scale Drivers

Precedentedness	<input type="text" value="Nominal"/>	Architecture / Risk Resolution	<input type="text" value="Nominal"/>	Process Maturity	<input type="text" value="Nominal"/>
Development Flexibility	<input type="text" value="Low"/>	Team Cohesion	<input type="text" value="High"/>		

Software Cost Drivers

Product

Required Software Reliability	<input type="text" value="Very High"/>
Data Base Size	<input type="text" value="Nominal"/>
Product Complexity	<input type="text" value="High"/>
Developed for Reusability	<input type="text" value="Nominal"/>
Documentation Match to Lifecycle Needs	<input type="text" value="Nominal"/>

Personnel

Analyst Capability	<input type="text" value="Nominal"/>
Programmer Capability	<input type="text" value="Nominal"/>
Personnel Continuity	<input type="text" value="Nominal"/>
Application Experience	<input type="text" value="Nominal"/>
Platform Experience	<input type="text" value="Nominal"/>
Language and Toolset Experience	<input type="text" value="Nominal"/>

Platform

Time Constraint	<input type="text" value="High"/>
Storage Constraint	<input type="text" value="High"/>
Platform Volatility	<input type="text" value="Nominal"/>

Project

Use of Software Tools	<input type="text" value="Nominal"/>
Multisite Development	<input type="text" value="Nominal"/>
Required Development Schedule	<input type="text" value="Nominal"/>

Maintenance

Annual Change Size (ESLOC)	<input type="text" value="80000"/>	Maintenance Duration (Years)	<input type="text" value="15"/>
Software Understanding (0%-50%)	<input type="text" value="25"/>	Unfamiliarity (0-1)	<input type="text" value=".4"/>

Software Labor Rates

Cost per Person-Month (Dollars)

Systems Engineering	Software	Hardware	Summary
Systems Engineering Acquisition Effort = 1767.9 Person-months Schedule = 17.7 Months Cost = \$17.7 M			
Systems Engineering Maintenance Cost = \$23.1 M			
Software Development (Elaboration and Construction) Effort = 10344.6 Person-months Schedule = 77.5 Months Cost = \$103.4 M			
Software Maintenance Cost = \$103.8 M			
Hardware Development and Production Cost = \$608 M			
Total System Cost = \$856.0 M			
Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_08_08_09_196913.txt			

Acquisition and Maintenance Monte Carlo Risk Results (1/3)



Systems Engineering
Software
Hardware
Summary

Systems Engineering Acquisition

Effort = 1767.4 Person-months
 Schedule = 17.7 Months
 Cost = \$17.7 M

Systems Engineering Maintenance (15 Years)

Annual Cost = \$1.5 M
 Total Cost = \$23.1 M

Software Development (Elaboration and Construction)

Effort = 11520.3 Person-months
 Schedule = 80.3 Months
 Cost = \$115.2 M

Software Maintenance (15 Years)

Annual Cost = \$8.5 M
 Total Cost = \$127.7 M

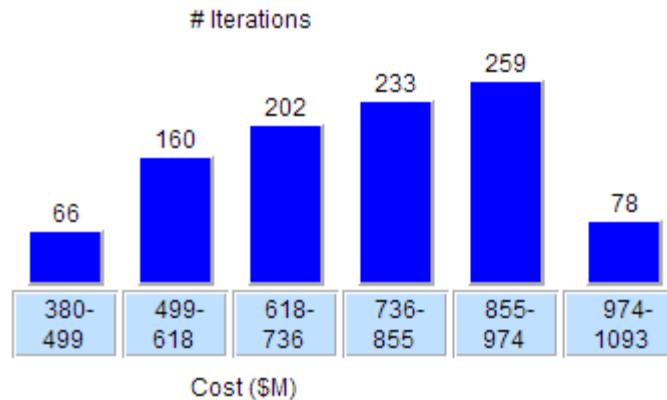
Hardware Development and Production

Cost = \$481.6 M

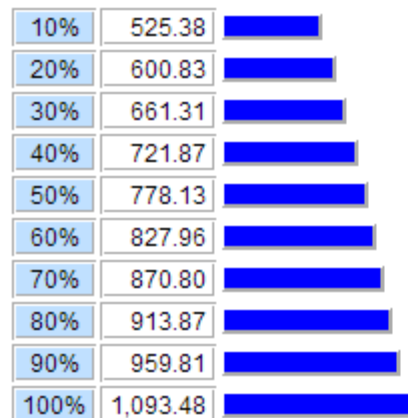
Totals

Acquisition Cost = \$614.4 M
 Maintenance Cost = \$150.8 M
 Total Cost = \$765.2 M

Total Cost Distribution Function

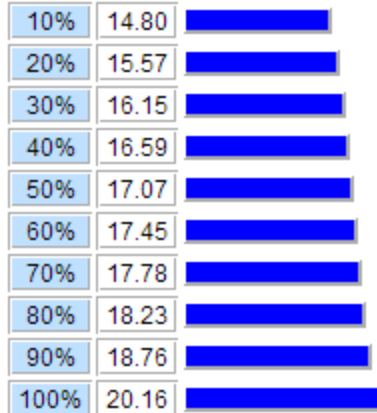


Total Cost Confidence Levels (\$M)

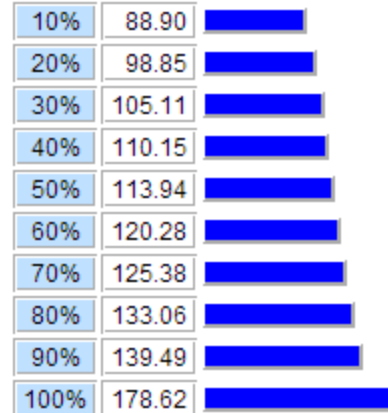


Acquisition Monte Carlo Results

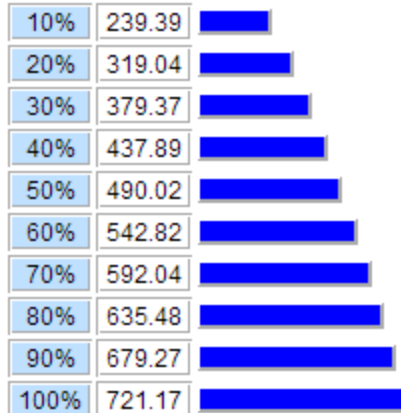
Systems Engineering Cost Confidence Levels (\$M)



Software Engineering Cost Confidence Levels (\$M)













Hardware Cost Confidence Levels (\$M)













Maintenance Monte Carlo Results

Systems Engineering Maintenance Cost Confidence Levels (\$M)

10%	19.34	
20%	20.34	
30%	21.10	
40%	21.68	
50%	22.30	
60%	22.80	
70%	23.23	
80%	23.82	
90%	24.51	
100%	26.33	

Software Engineering Maintenance Cost Confidence Levels (\$M)

10%	98.56	
20%	109.59	
30%	116.53	
40%	122.12	
50%	126.32	
60%	133.35	
70%	139.00	
80%	147.52	
90%	154.65	
100%	198.03	



System Cost Model Suite

Project Name:

System Type:

Options

Monte Carlo Risk

Systems Engineering

Software

Hardware

Summary

Satellite Hardware RDT&E

Element	Parameter		Cost
1. Payload			
1.1 Visible Light Sensor	Aperture diameter (m)	<input type="text" value="1.0"/>	\$128,827
2. Spacecraft			
2.1 Structure	Structure weight (kg)	<input type="text" value="245"/>	\$15,098
2.2 Thermal	Thermal weight (kg)	<input type="text" value="40"/>	\$4,100
2.3 Electrical Power System	X1 = EPS weight (kg)	<input type="text" value="420"/>	\$26,528
	X2 = BOL power (wt)	<input type="text" value="100"/>	
2.4 Telemetry, Tracking and Command	TT&C weight (kg)	<input type="text" value="50"/>	\$10,698
2.5 Attitude Determination and Control	ADCS weight (kg)	<input type="text" value="110"/>	\$27,315

Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_17_2013_08_46_43_265739.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

Results

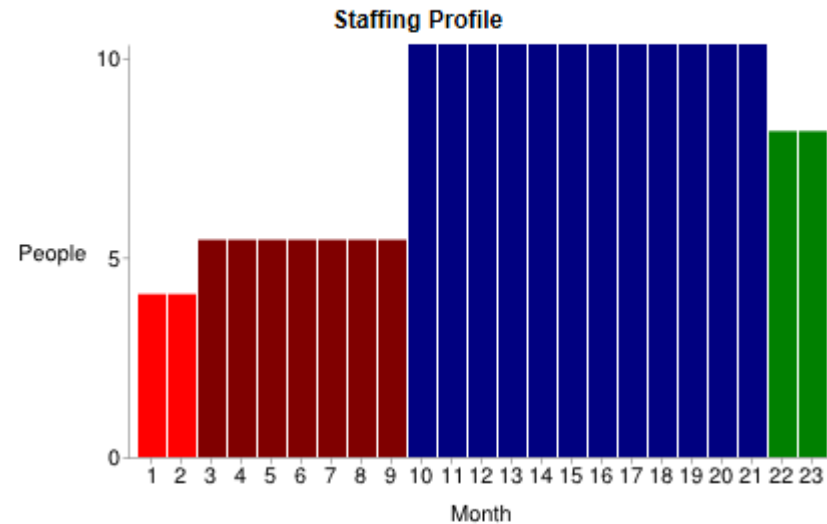
Software Development (Elaboration and Construction)

Effort = 169.9 Person-months
Schedule = 20.0 Months
Cost = \$1698762

Total Equivalent Size = 40000 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	10.2	2.5	4.1	\$101926
Elaboration	40.8	7.5	5.4	\$407703
Construction	129.1	12.5	10.3	\$1291060
Transition	20.4	2.5	8.2	\$203852



Software Activity Distribution (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	1.4	4.9	12.9	2.9
Environment/CM	1.0	3.3	6.5	1.0
Requirements	3.9	7.3	10.3	0.8
Design	1.9	14.7	20.7	0.8
Implementation	0.8	5.3	43.9	3.9
Assessment	0.8	4.1	31.0	4.9
Deployment	0.3	1.2	3.9	6.1

Your output file is http://diana.nps.edu/~madachy/tools/data/cost_model_suiteSeptember_16_2013_11_23_24_410790.txt

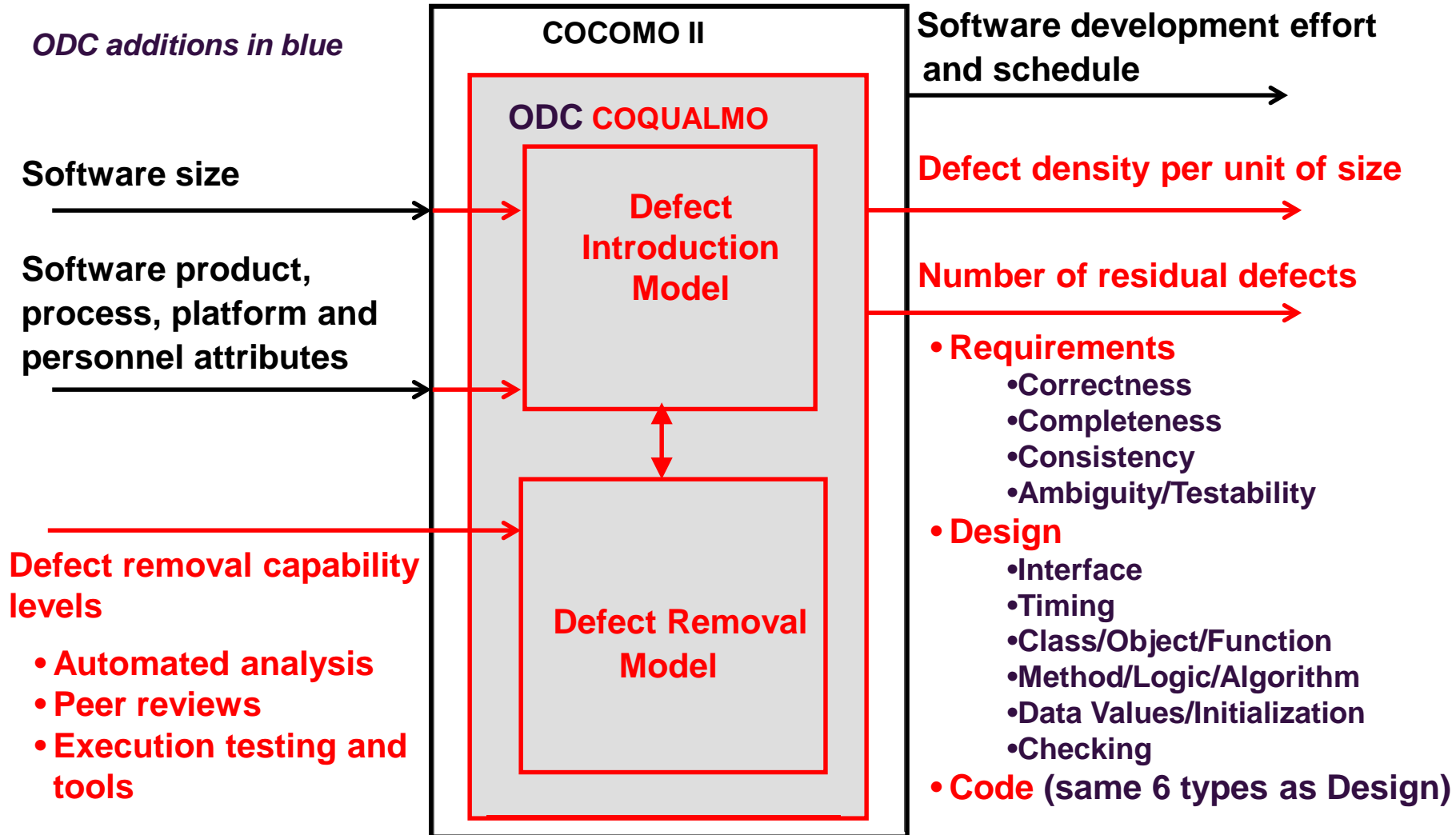
- **The Advance Missions Cost Model (AMCM)** predicts development, recurring and mission operations costs of ground vehicles, ships, aircraft, helicopters, missiles, launch vehicles, spacecraft, and human explorations missions.
- It is a system level cost model appropriate for large scale programs requiring many different systems that will be integrated to perform a complex mission. The model is most useful in the pre-conceptual and conceptual design phases of a program when the actual design of the systems is not known and many factors are being traded off.

- The AMCM is a two equation, multi-variable cost estimating relationship. The first equation predicts the development and production cost of the system based of various technical and programmatic factors. The second equation predicts the basic mission operations cost of the system.
- Both equations are fitted to a large historical database that spans 50 years of systems development. Most of the systems are US Government developed, but some commercial and European systems are included.
 - Database of land, water, air and space systems. The data includes 54 spacecraft, 22 space transportation systems, 61 aircraft, 86 missiles, 29 ships, and 18 ground vehicles. All of the data points are from programs that were completed through IOC.

- Cost, schedule and quality are highly correlated factors in software processes
- Thus the COConstructive QUALity MOdel (COQUALMO) was created to predict defects as an extension of the COCOMO II software cost model [Chulani, Boehm 1999]
 - Uses COCOMO II cost estimation inputs with defect removal parameters to predict the numbers of generated, detected and remaining defects for requirements, design and code
- Provides insights into cost/schedule/quality tradeoff analyses, quality investment payoffs, interactions amongst quality strategies, and likely schedule
- Enables what-if analyses that demonstrate the impact of
 - Defect removal techniques for automated analysis, peer reviews, and execution testing on defect types
 - Effects of **product**, personnel, project, and platform characteristics on software quality
- ODC COQUALMO is a further extension that predicts software defects introduced and removed classifying them with Orthogonal Defect Classification (ODC) defect types

COQUALMO extensions to COCOMO II in red

ODC additions in blue



- ODC COQUALMO decomposes defects using ODC categories [Chillarege et al. 1992]
 - Enables tradeoffs of different detection efficiencies for the removal practices per type of defect
- The ODC taxonomy provides well-defined criteria for the defect types and has been successfully applied on NASA projects and others
- With more granular defect definitions, ODC COQUALMO enables tradeoffs of different detection efficiencies for the removal practices per type of defect.
 - V&V techniques have different detection efficiencies for different types of defects, and may have overlapping capabilities between them
- ODC defect types can be mapped to technical performance parameters for trade analysis

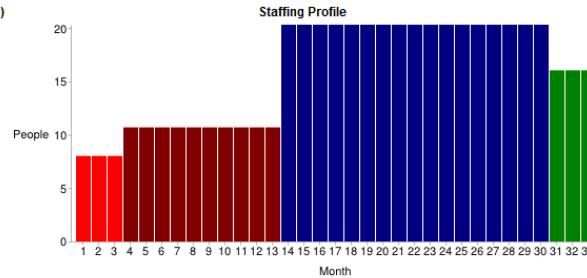
Software Development (Elaboration and Construction)

Effort = 465.3 Person-months
Schedule = 27.9 Months
Cost = \$4653152

Total Equivalent Size = 100000 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	27.9	3.5	8.0	\$279189
Elaboration	111.7	10.4	10.7	\$1116757
Construction	353.6	17.4	20.3	\$3536396
Transition	55.8	3.5	16.0	\$558378



Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	3.9	13.4	35.4	7.8
Environment/CM	2.8	8.9	17.7	2.8
Requirements	10.6	20.1	28.3	2.2
Design	5.3	40.2	56.6	2.2
Implementation	2.2	14.5	120.2	10.6
Assessment	2.2	11.2	84.9	13.4
Deployment	0.8	3.4	10.6	16.8

Requirements Defects

Introduced

Ambiguity/Testability	70	
Completeness	230	
Consistency	170	
Correctness	530	

Removed

Ambiguity/Testability	28	
Completeness	92	
Consistency	76	
Correctness	204	

Remaining

Ambiguity/Testability	42	
Completeness	138	
Consistency	93	
Correctness	325	

Design Defects

Introduced

Checking	146	
Class/Object/Function	254	
Data Values/Initialization	510	
Interface	510	
Method/Logic/Algorithm	580	
Timing	0	

Removed

Checking	62	
Class/Object/Function	43	
Data Values/Initialization	239	
Interface	134	
Method/Logic/Algorithm	309	
Timing	0	

Remaining

Checking	83	
Class/Object/Function	210	
Data Values/Initialization	270	
Interface	375	
Method/Logic/Algorithm	270	
Timing	0	

Code Defects

Introduced

Checking	219	
Class/Object/Function	381	
Data Values/Initialization	765	
Interface	765	
Method/Logic/Algorithm	869	
Timing	0	

Removed

Checking	94	
Class/Object/Function	65	
Data Values/Initialization	359	
Interface	201	
Method/Logic/Algorithm	463	
Timing	0	

Remaining

Checking	124	
Class/Object/Function	315	
Data Values/Initialization	405	
Interface	563	
Method/Logic/Algorithm	406	
Timing	0	

Your output file is https://diana.nps.edu/MSAcc/tools/data/COQUALMO_July_16_2012_21_08_56_333443.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

- General cost modeling tool currently available at
http://diana.nps.edu/~madachy/tools/cost_model_suite.php
http://csse.usc.edu/tools/cost_model_suite.php