University of Southern California
Center for Software Engineering

COINCOMO Software Cost and Schedule Estimation Models

COCOTS Integration Challenges

A. Winsor Brown
Ed Colbert
AWBrown@CSSE.USC.edu
EColbert@CSSE.USC.edu

Goals

Present the new, extended, combined COCOMO Model: COINCOMO (Incremental Development)

- COCOTS for COTS software with applications
- COSECMO for software systems with security

Show how RUP/MBASE and COINCOMO fit with the Incremental Commitment Model for systems development

Outline

COCOMO + COPSEMO (COINCOMO Base) Models

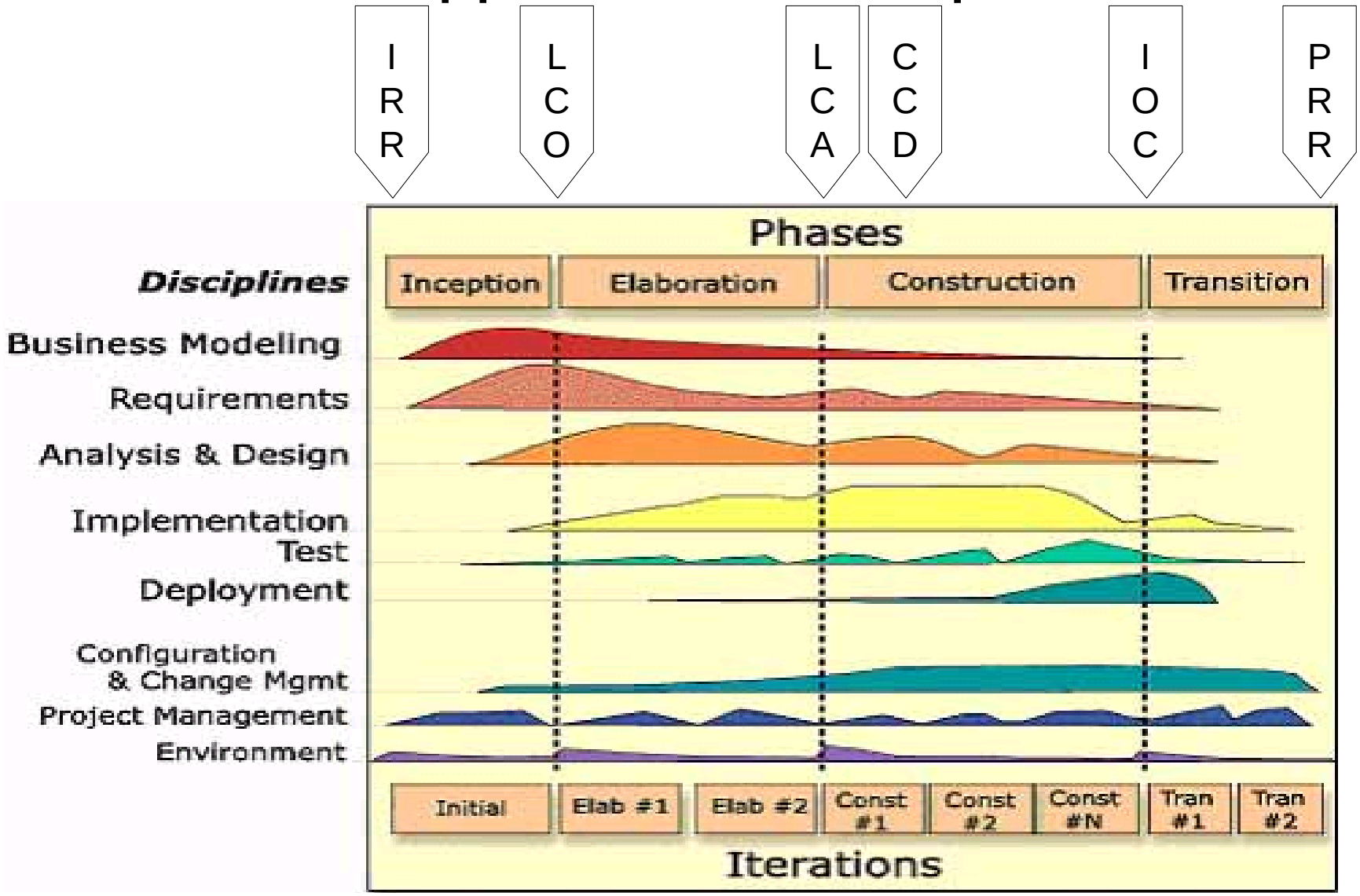
COTS and (some) Open Source Models

COSECMO with COCOTS and COINCOMO = the New [or Extended] COINCOMO

Incremental Commitment Models (ICMs)

ICM for Software with Extended COINCOMO

RUP/MBASE Application Development Model[1]



1 (efforts not to scale)

Building on the COCOMO II (CII) Base

COPSEMO (COngruent Phase Schedule and Effort MOdel)
[nee COnstructive Phase Schedule and Effort MOdel]

$$E_P\% = X_P\% \text{ of } E_{CIIbase}$$

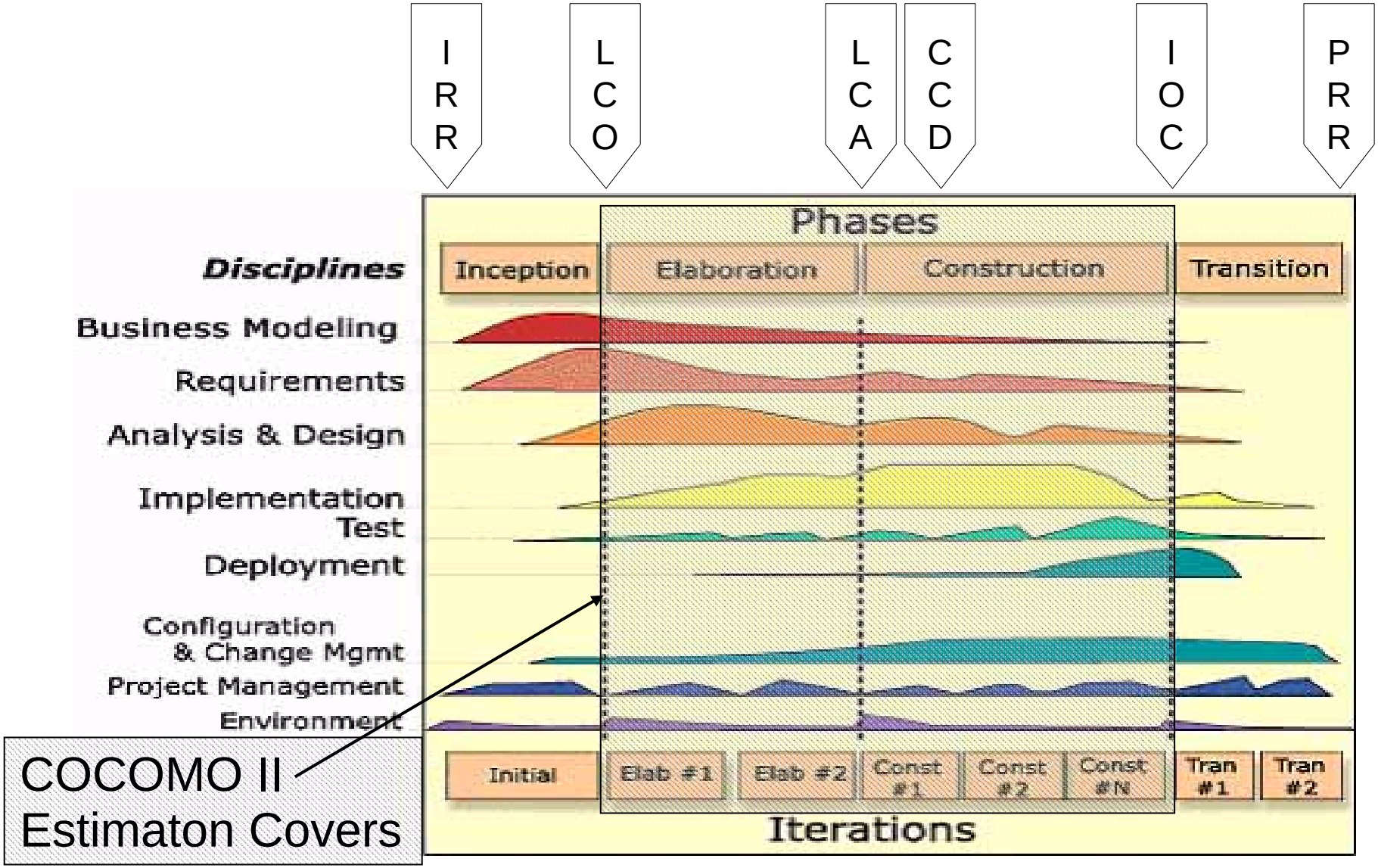
$$S_P\% = Y_P\% \text{ of } S_{CIIbase}$$

- Extrapolation of Effort (E) and Schedule (S) from CII's
  - E_I% & S_I% of CII to Inception E and S
  - E_T% & S_T% of CII to Transition E and S
- Interpolation of Effort (E) and Schedule (S) from CII's
  - E_E% & S_E% of CII to Elaboration E and S
  - 100-E_E% & 100-S_E% (the rest) of CII's to Construction

NOTE: Percentages currently all̃ based on experience

Recalculate Persons needed per phase: P=PM/M

MBASE/RUP Concurrent Activities



COCOMO II with COPSEMO (for IECT)

USC-COCOMO II.2000.3 - Untitled

File Edit View Parameters Calibrate Phase Maintenance Help

Project Name: Buildl=Core

Scale Factor: 16.14 Schedule

Project Notes

Development Model: Post Architecture

X	Module Name	Module Size	LABOR Rate ($/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Core Buildl	S:229999	0.00	1.68	Non-Specified	997.0	1675.1	137.3	0.00	0.0	44.9	1.4

Total Lines of Code: 229999

Hours/PM: 152.00

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	1340.1	34.8	171.6	0.00	0.0	38.5	
Most Likely	1675.1	37.3	137.3	0.00	0.0	44.9	1.4
Pessimistic	2093.9	40.0	109.8	0.00	0.0	52.4	

Project Is Saved To File : C:\COCOMOII.2000.3\MultiBuildExample\3Builds=200+150+100K\Build1\Build1-200K.est

COPSEMO

Step NOTE: **BOLD** implies a required value. ***Italic*** implies an optional value

1.0 . . . . . Currently implemented only for projects (not modules)

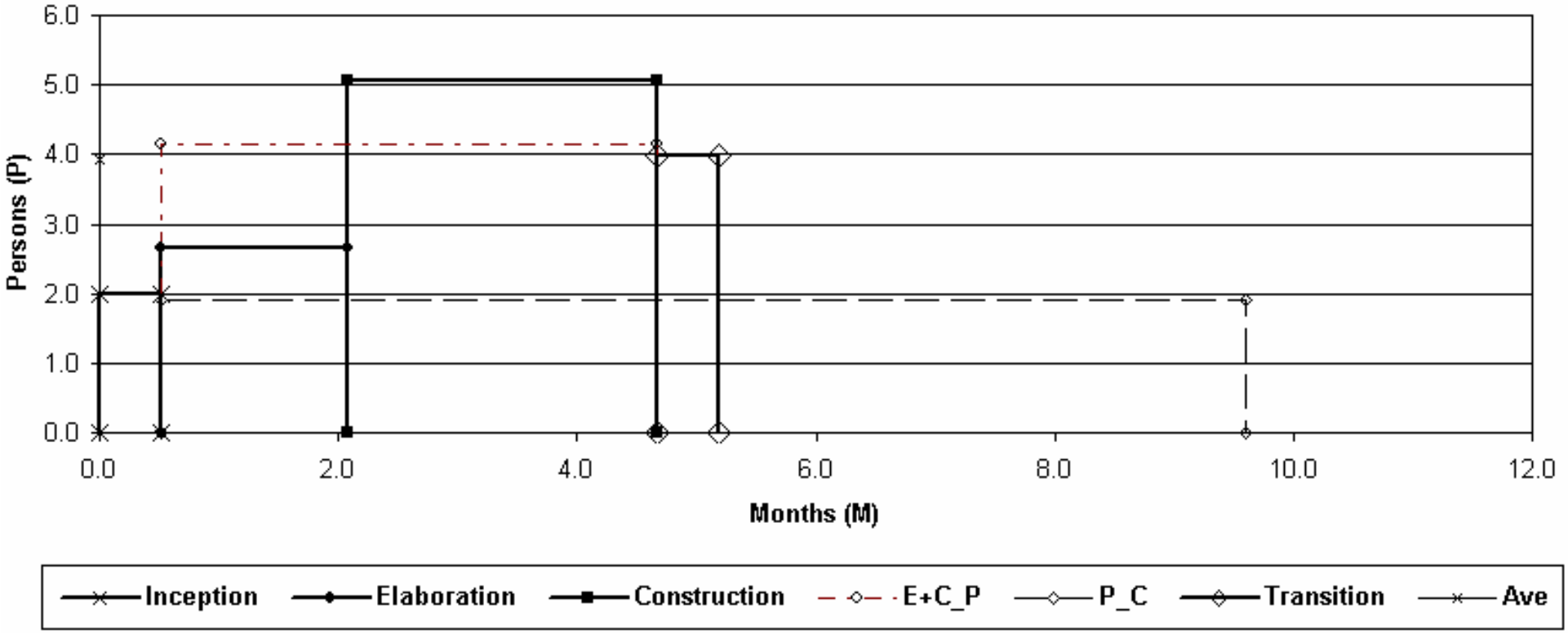
Get COCOMO II.2000 data and adjustments

2.0 ***Project:*** ***Build1=Core*** **TotalSize=** 229,999 **PM_C=** 1675.1 **M_C=** 37.28

(automatically from COCOMO_charts.xls) **SCED_R=** N **SCEDinc:** 0% **SCEDV=** 1.00 P_C= 44.93

Work hours per months (non-overhead or "billable"; COCOMO-II.2000 default=152)= 152

COPSEMO: Phased Schedule and Effort Dist.



Default values: *Italic*

	Inception		Elaboration		Construction		Transition		Total E&C	Total
Effort %	*6.0*	**6.0**	*24.0*	**24.0**	*76.0*	**76.0**	*12*	**12.0**	100.0	118.0
Schedule %	*12.5*	**12.5**	*37.5*	**37.5**	*62.5*	**62.5**	*12.5*	**12.5**	100.0	125.0
P/Ave(P)	0.48		0.64		1.22		0.96		1.00	Does not apply
	PM / M = P		PM / M = P		PM / M = P		PM / M = P		PM / M = P	PM / M = P-ave
PSE Distributed	1.04 / 0.52 = 1.99		4.14 / 1.56 = 2.66		13.12 / 2.60 = 5.05		2.07 / 0.52 = 3.99		**17.3** / **4.15** = 4.15	20.4 / 5.19 = 3.92

Ave(P) refers to the average number of persons on the project; it is the same as PM_BS/M_BS for the entire project, and each stage's P/Ave(P) is the same as stage's Effort%/Schedule%.

COCOMO Models Represented in UML

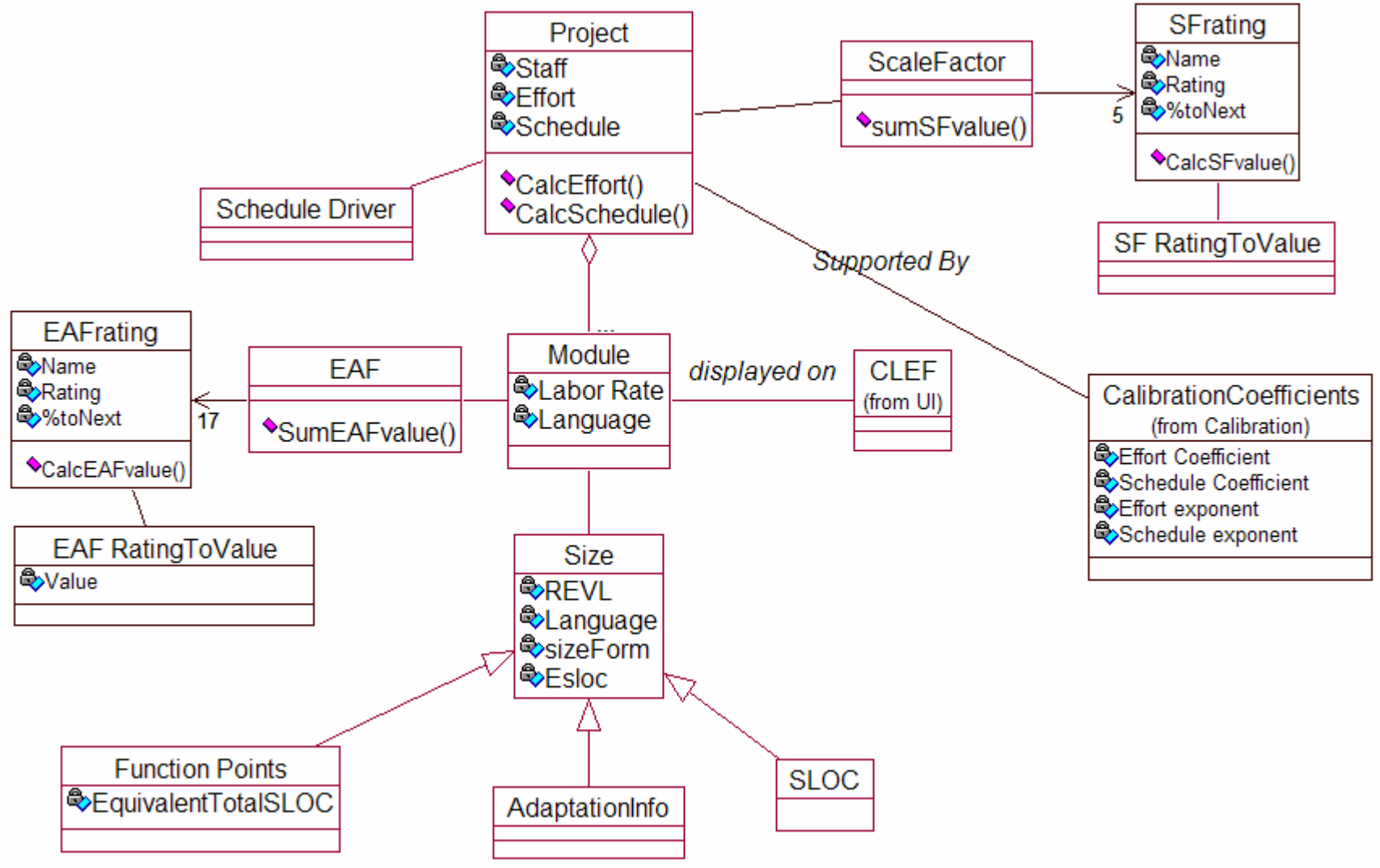
COCOMO.II.2003 (the software package)

COINCOMO.2006 version of COCOMO.II.2003

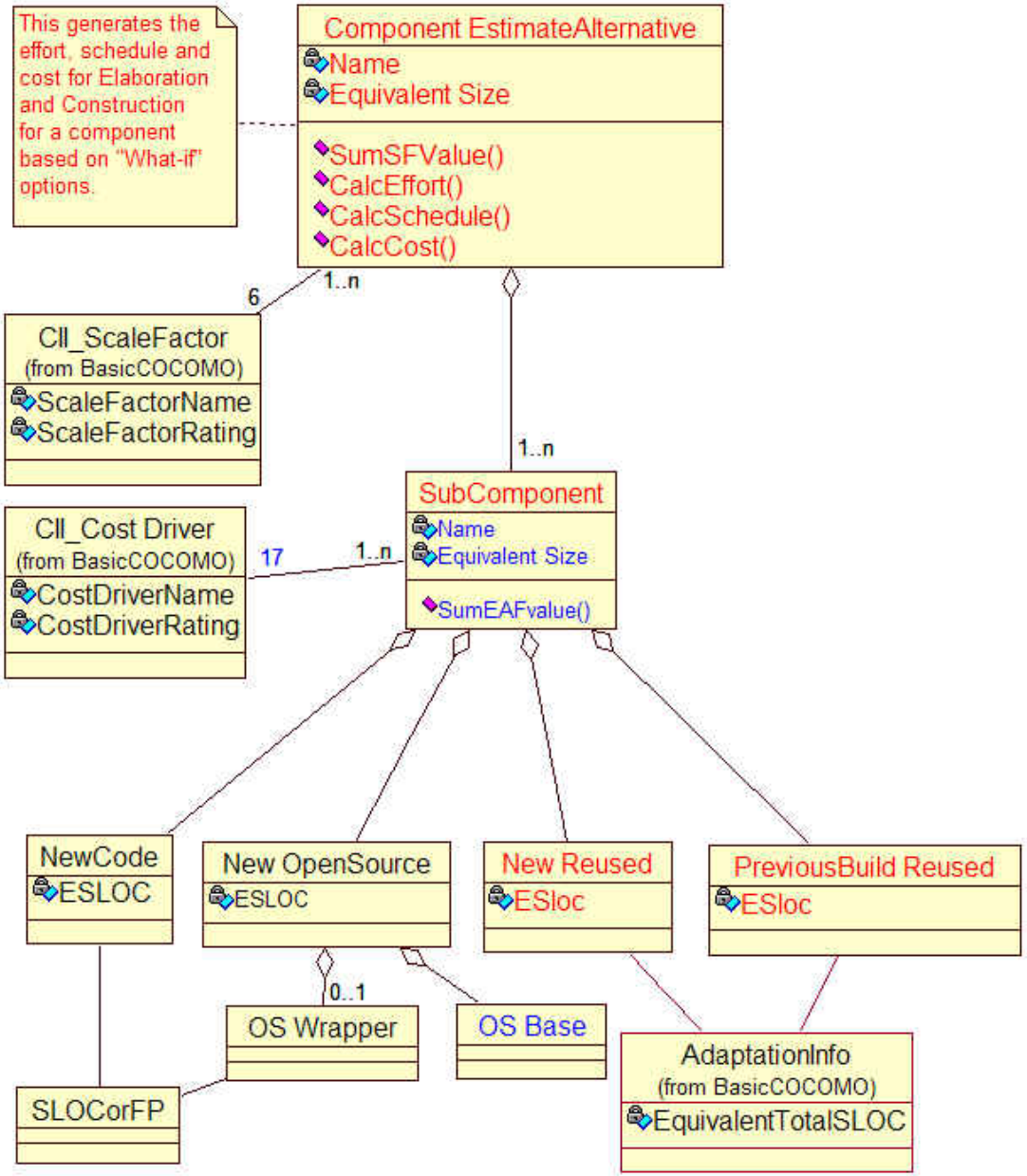
- Has concept of stored alternatives for
  - Components (AKA Project) [shown]
  - EAF and SF Driver sets [not shown]
  - Sub-Components (AKA Modules) [not shown]
- Has four SLOC sources:
  - SLOC or FP converted to SLOC: New & New Open Source
  - Adaptation Only: New Reused & Previous Build Reused

COINCOMO version of COCOMO.II.2003 with COPSEMO

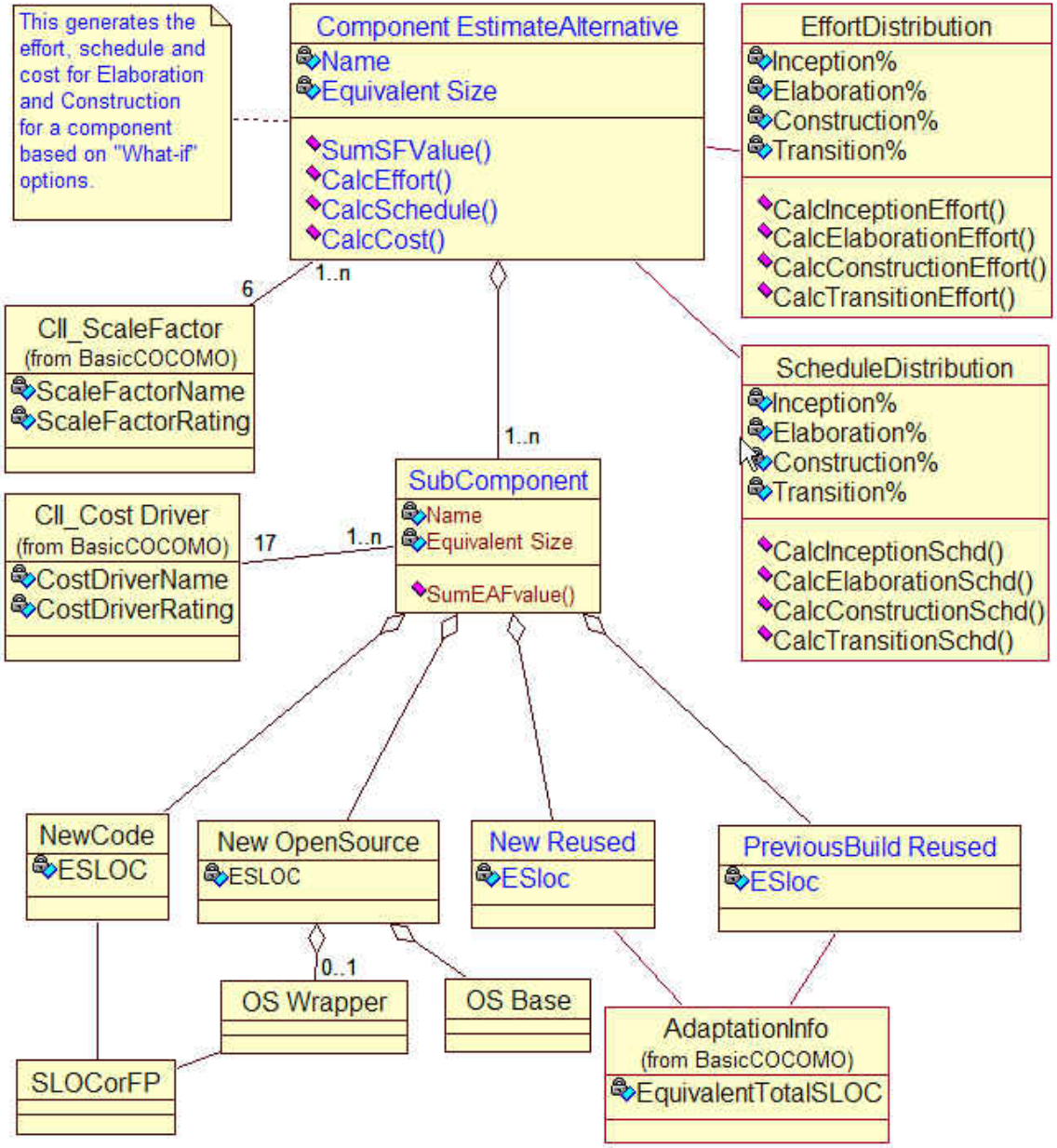
COCOMO.II.2003 in UML



COINCOMO's COCOMO.II.2003 in UML



COINCOMO's COCOMO + COPSEMO in UML



COTS and Open Source Software

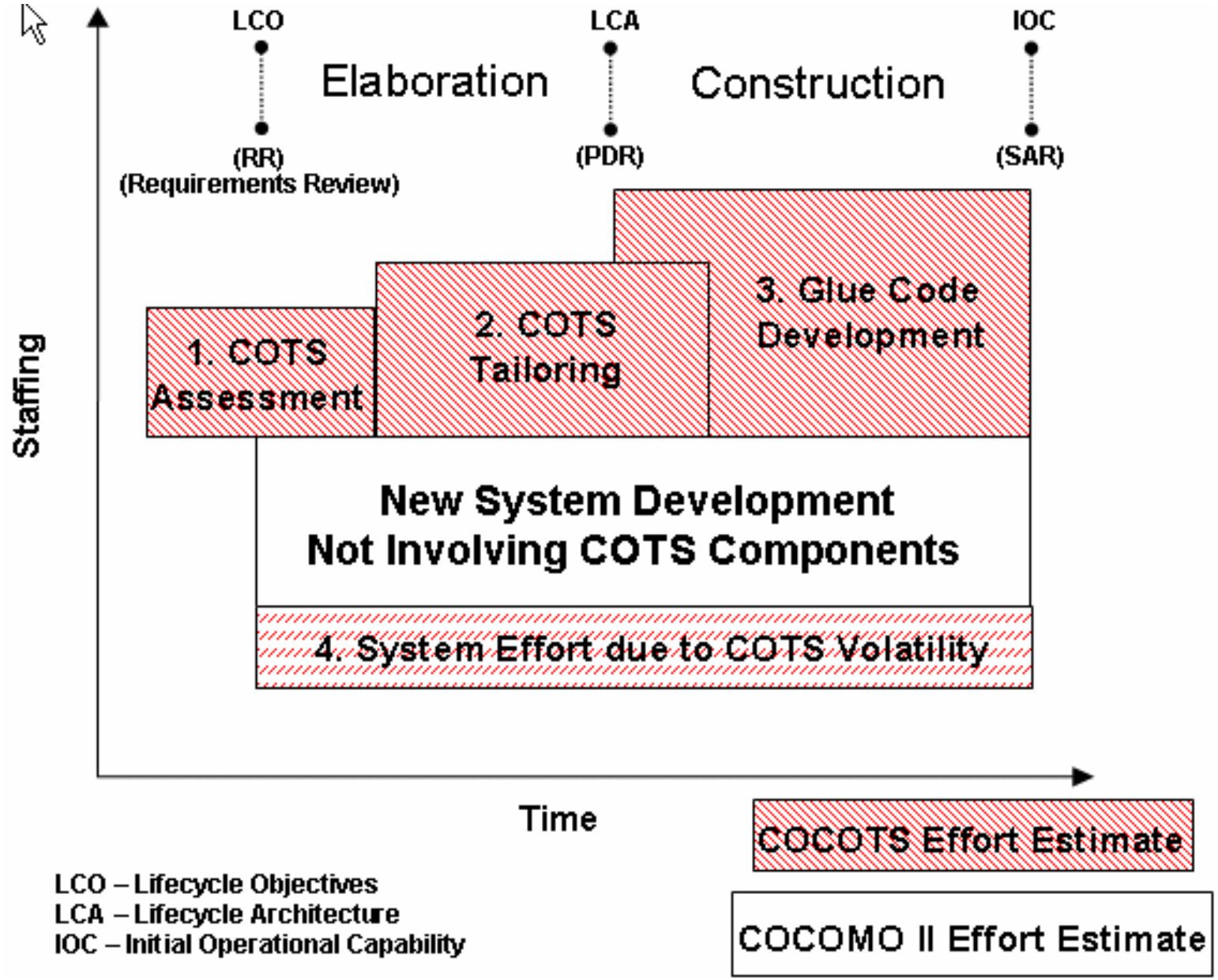
Today's Realities:

- Many, many systems have COTS components
- Many systems use Open Source components which are treated like COTS (AKA pseudo-COTS):
  - Might put wrappers around it
  - Don't look at internals
  - Let the Open Source developers control its evolution

COCOTS calculates effort and schedule for systems with COTS (or pseudo-COTS)

- Activity based estimates for Assessment and Tailoring
- Special "Glue Code" model for COTS to rest of system
- NOTE: COTS volatility effects not calculated!

COCOTS: COTS with Assessment, Tailoring and Glue Code



COCOTS + COCOMO + COPSEMO

Issues:

- How to determine relative anchor point dates?
  - Apply maximum calculated LCA date (adjust COPSEMO S&E percentages to fit)
  - Reality is that they are probably politically based
    - But don't do LCO or LCA too early;
    - If "scheduled" IOC before calculated IOC, adjust Sched!
- How to allocate Assessment to phases?
  - Initial Assessment (or screening) to Inception?
  - Detailed Assessment and selection in Elaboration?
- How to allocate Tailoring to phases?
  - All to Elaboration: COTS selection required(?) for LCA?
  - Need another percentage factor, derived from experience, to allocate parts to Elaboration and Construction.

COSECMO Extension to COCOMO

COSECMO, the COCOMO SECurity MOdel, Focus: Cost & Schedule Estimates (C&SE)

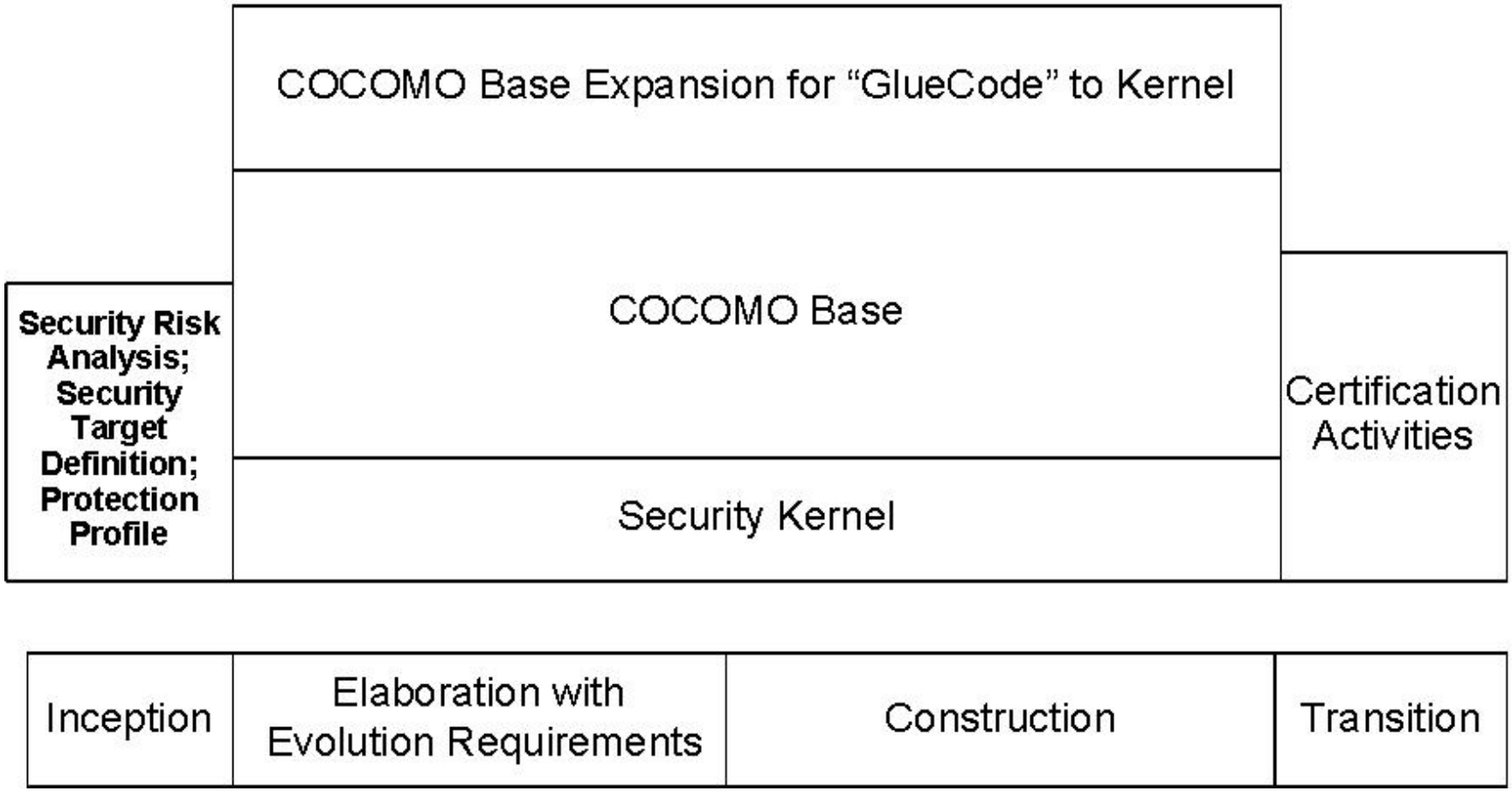
- COCOMO covers Elaboration (E) & Construction (C)
- COSECMO covers the ***increased*** costs ***and*** schedule for security, spread over
  - Inception (I)
  - Elaboration (E)
  - Construction (C)
  - Transition (T)

COSECMO to help government and industry, using modern practices (e.g., the Common Criteria), predict cost of developing or acquiring secure systems.

COSECMO model still evolving, implementation based on COINCOMO available to Affiliates since the fall of 2005

COSECMO

COSECMO Conceptual Model



COSECMO Impacts on Schedule

COPSEMO ranges of percentages[2]

	MBASE	
Phase (endpoints)	Effort%	Schedule%
Inception (IRR to LCO)	6 (2-15)	12.5 (2-30)
Elaboration (LCO to LCA)	24 (20-28)	37.5 (33-42)
Construction (LCA to IOC)	76 (72-80)	62.5 (58-67)
Transition (IOC to RRR)	12 (0-20)	12.5 (0-20)
Totals:	118	125

COSECMO *increases in* schedule depend on EAL – suggestions for initial selections (no experience)

- EALs 1 and 2: use normal
- EALs 3 and 4: use high end of ranges
- EALs 5, 6 and 7: beyond the ranges shown above

[2] from Table A.5 of

COSECMO *increases in* schedule for security

S_I% of CII with COSECMO (for Inception)

- EAL 3 and 4: S_I% = 25. Higher because of Security Risk Analysis, Security Target Definition and Protection Profile

S_E% of CII with COSECMO (for Elaboration)

- EAL 3 and 4: S_E% = 28+. Higher because of need for architecture completeness with evaluation of security.

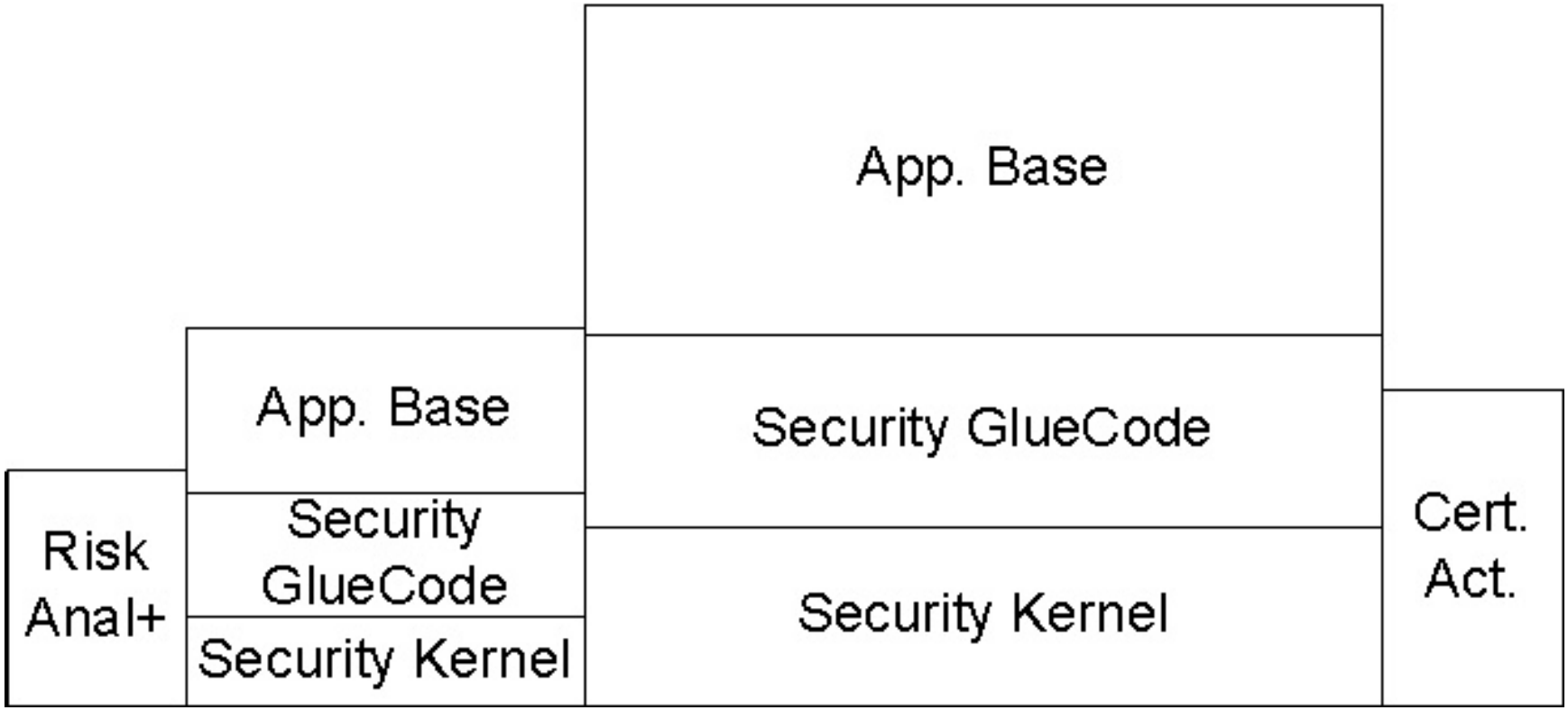
S_C% of CII with COSECMO (for Construction)

- EAL 3 and 4: 100-S_E% = 72- (72 or lower)

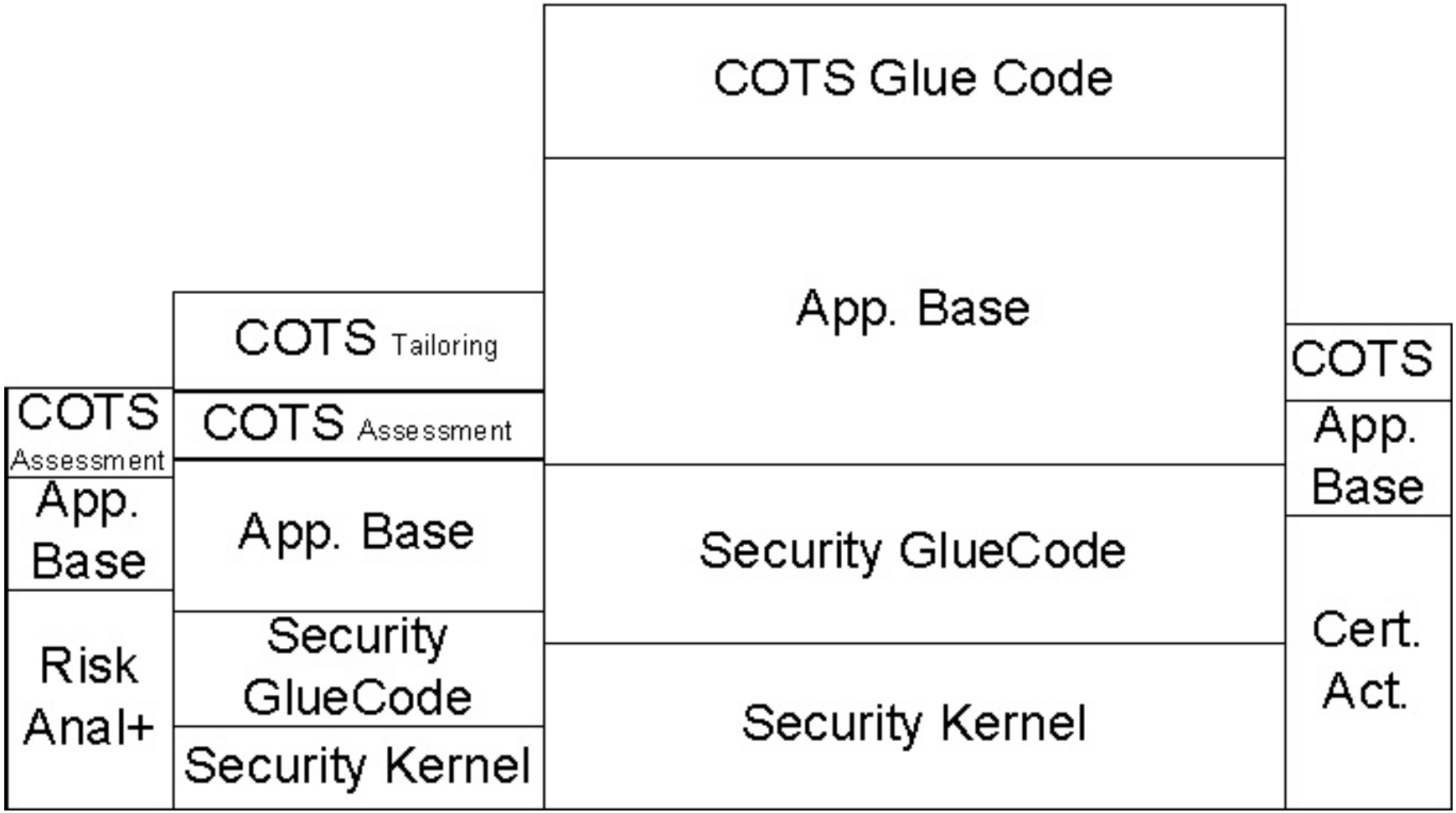
S_T% of CII with COSECMO (for Transition)

- EAL 3 and 4: S_T% = 20+. Higher because of certification time (on top of increase related to higher cost).

COSECMO with COPSEMO distributions



COINCOMO=COSECMO+COPSEMO+COCOTS



Incremental Commitment Models

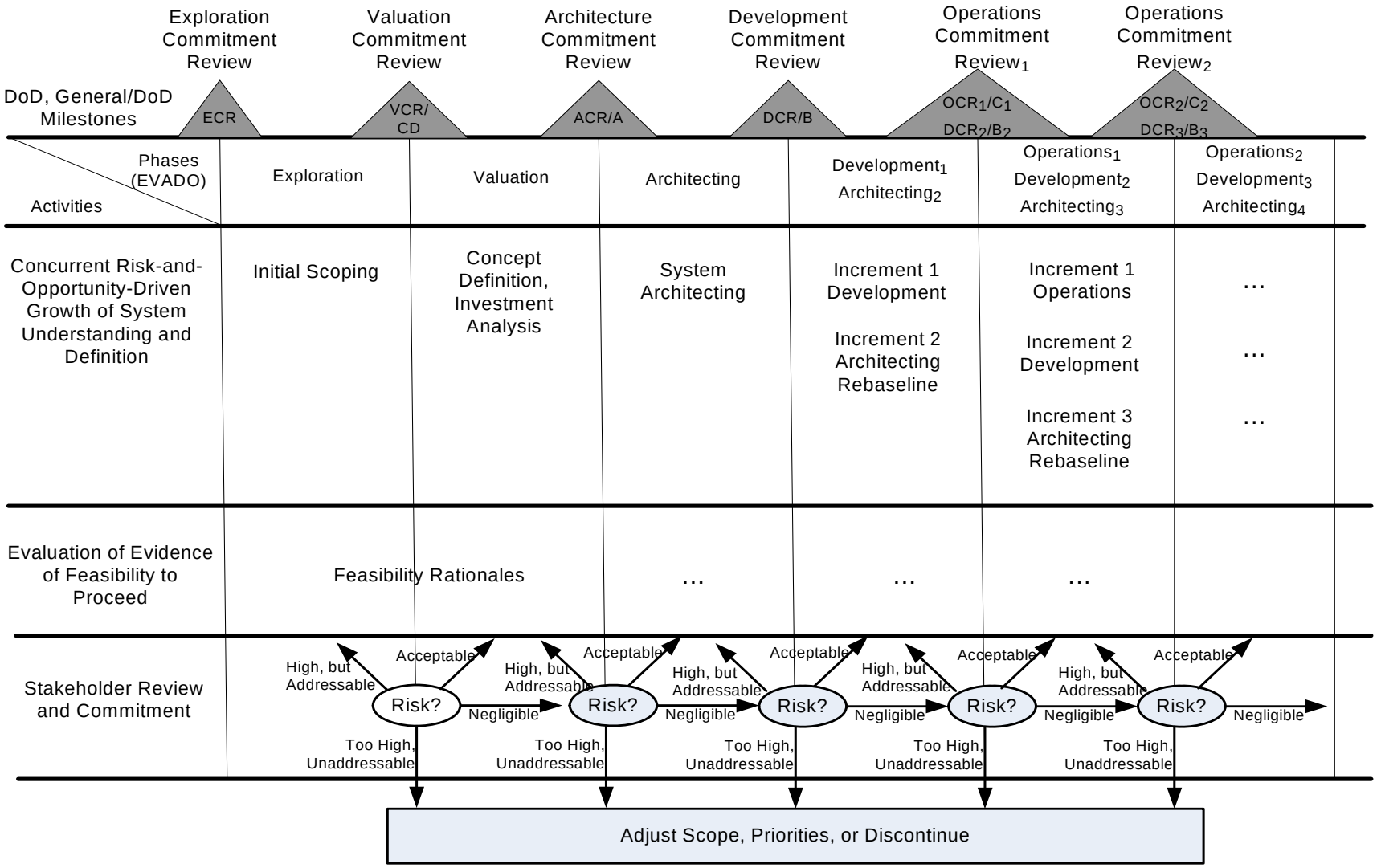
The Incremental Commitment Models (ICMs) for Life Cycle Processes

- For Systems (Human, Hardware and Software)
- ICM for Software Intensive Systems (ubiquitous hardware)
- ICM for Software (only) Systems

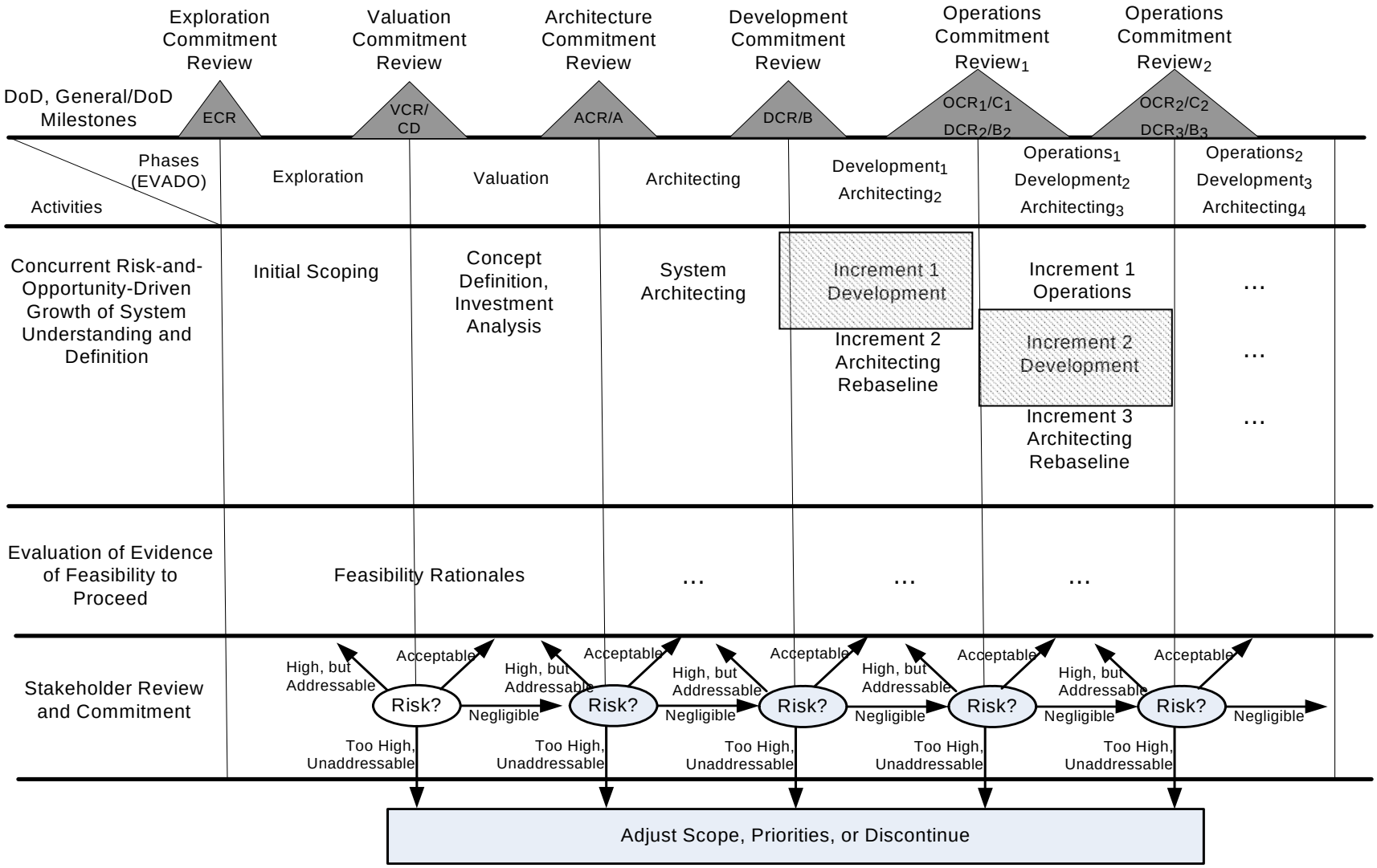
ICMs solve Spiral Model problems

- Use spiral principles vs. diagram
- Relate to stakeholder commitments and values
- Make concurrency explicit
- Use risk to explicitly show go-backs and skips
- Provide view for handling mini-spirals

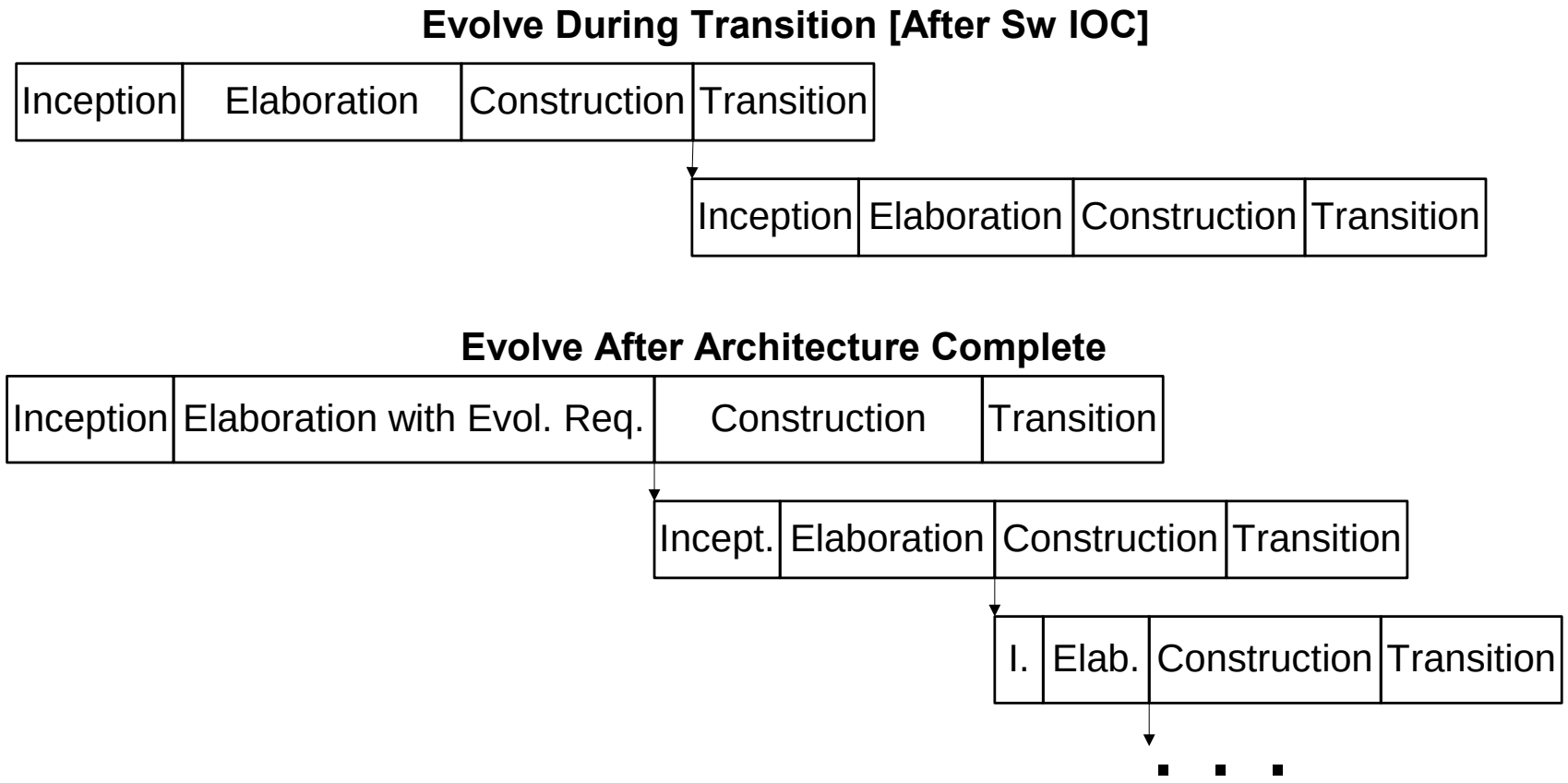
ICM LC Processes For Systems [of Systems]



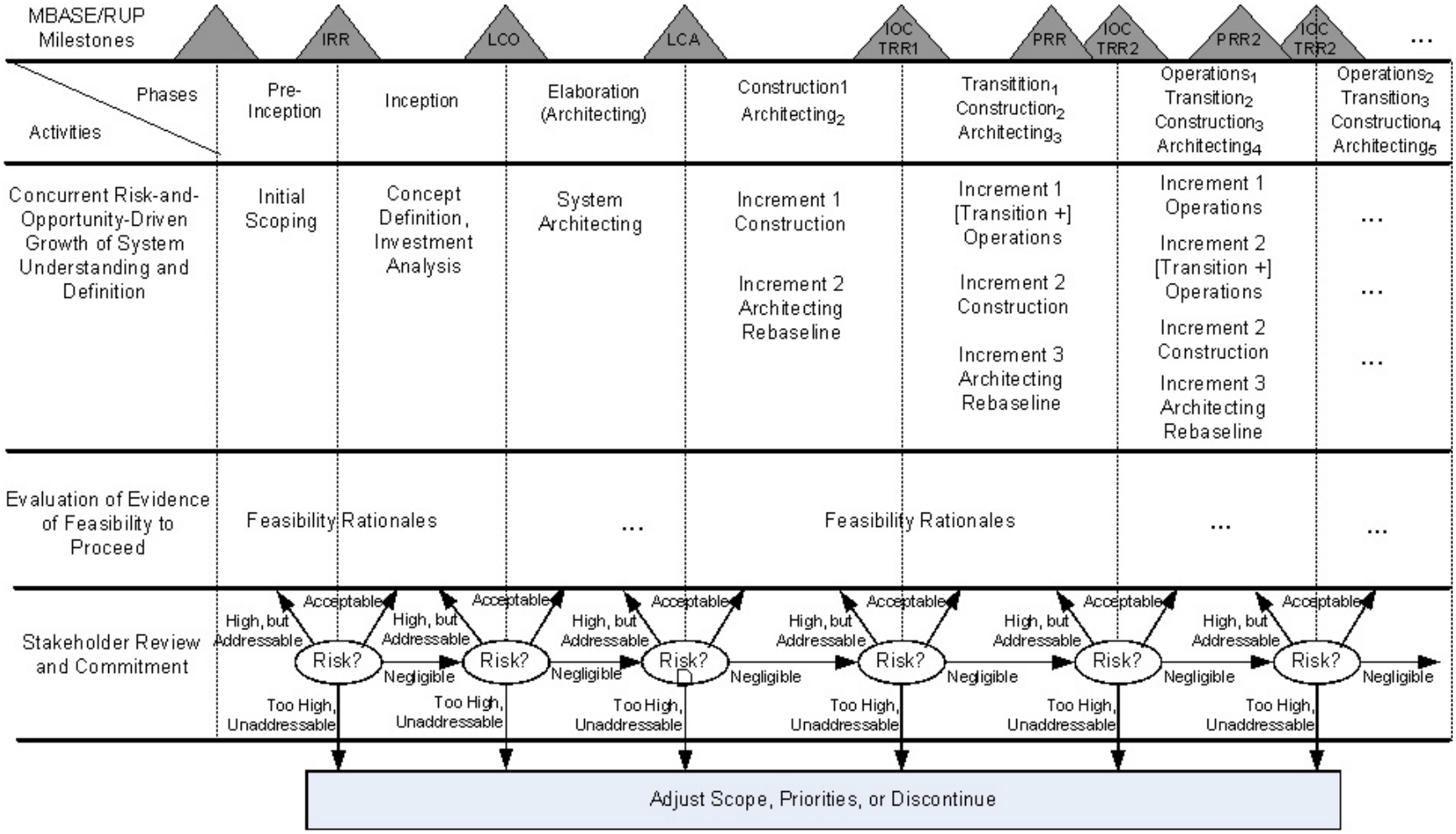
ICM Showing Software in Systems



Overlaps Across Software Builds

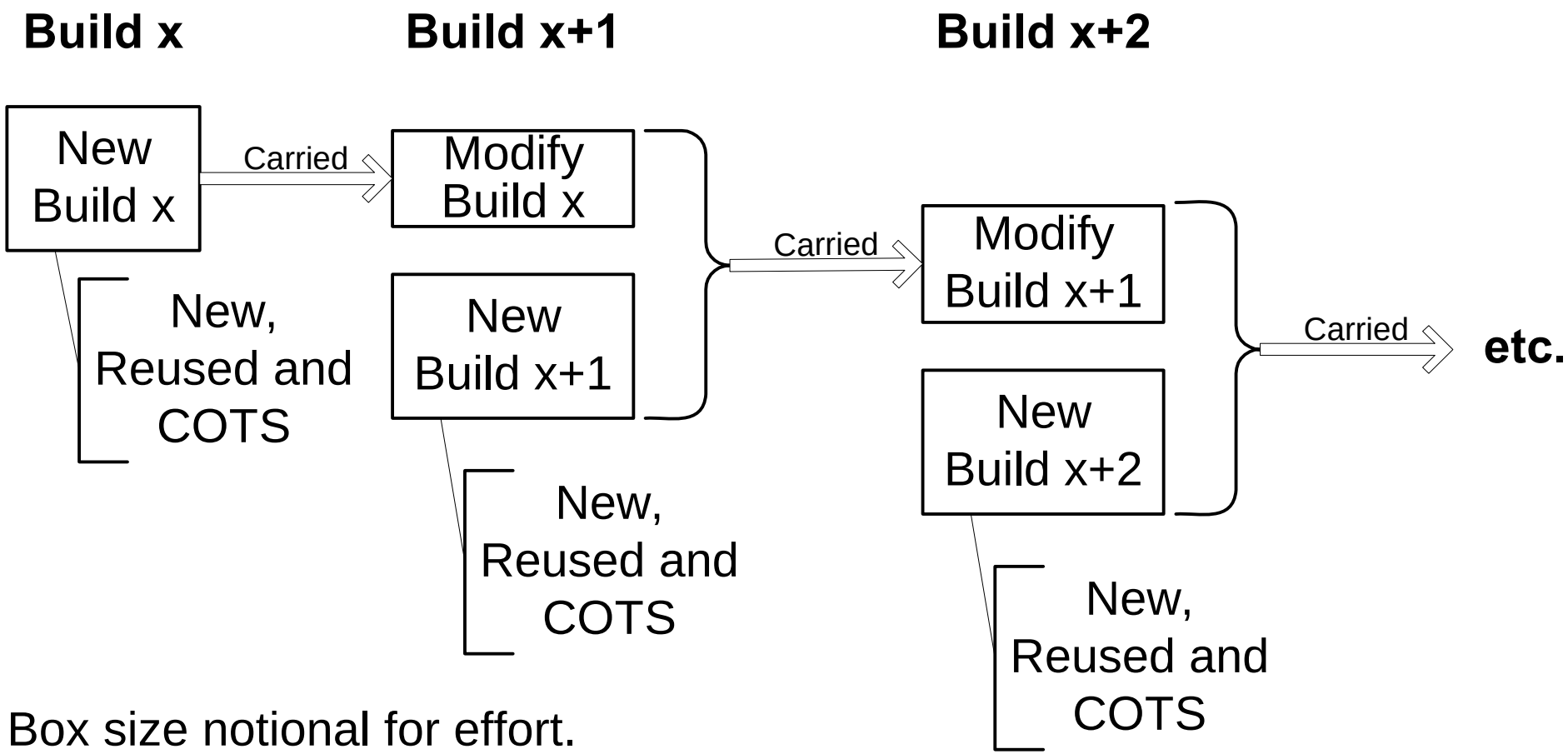


ICM for Software



Multi-Build COCOMO II

COINCOMO Sums Across Builds



Box size notional for effort.

COINCOMO = Multi-Build COCOMO II

Guidance about how to "carry" forward

Table 2 Modified Code Adaptation Parameters

Modified Component	REVL %	% Design Modified (DM)	% Code Modified (CM)	% Integration Required (IM)	Software Under-standing (SU) (%)	Assessment and Assimil-ation (AA) 0-1 *(10%)	Programmer Unfamiliarity (UNFM) (0-1)
1. default	15	20	40	100	30	4	.4
2. Carried Build assumed designed for reuse (CB) 1st time	15	10	20	50	20	0	.1
3. CB 2nd time	10	5	10	40	10	0	0
4. CB 3rd or more times	5	5	5	30	5	0	0

COINCOMO = Multi-Build COCOMO II

Guidance about how to "carry" forward (cont.)

Table 3 Reused Code Adaptation Parameters

Reused Component Type	REVL (%)	% Integration Required (IM)	Assessment & Assimilation (AA) 0-10 * (10%)
1. COTS	0	100 – 25 – 10 based on mission criticality	4
2. COTS from previous build (re-test only)	0	25 – 10 – 5 based on mission criticality	0
3. Which has been (Re)Designed for reuse, based on new (or modified) code by vendor.	15	100	6
4. 1st Carry from Previous build of code (Re)Designed for reuse, based on new (or modified) code by vendor.	15	100	2
5. 2nd Carry from Previous build of code (Re)Designed for reuse, based on new (or modified) code by vendor.	10	100	1
6. 3rd ..nth Carry from Previous build of code (Re)Designed for reuse, based on new (or modified) code by vendor.	5	100	0.5

Background

WinWin Spiral Model (WWSM)

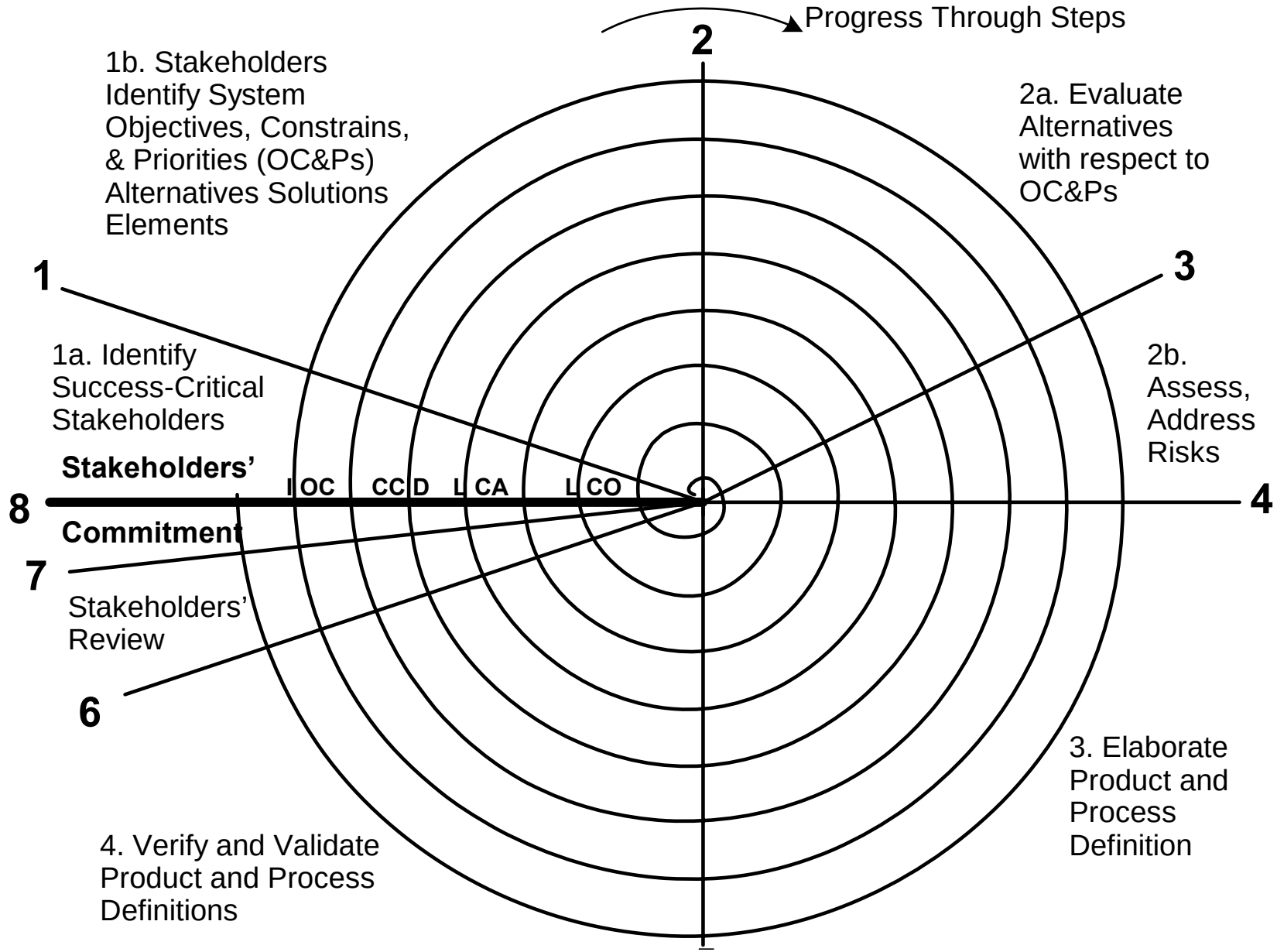
WWSM: CSCI577 Unrolled with Repeated Cycles

Background and Definitions (cont.)

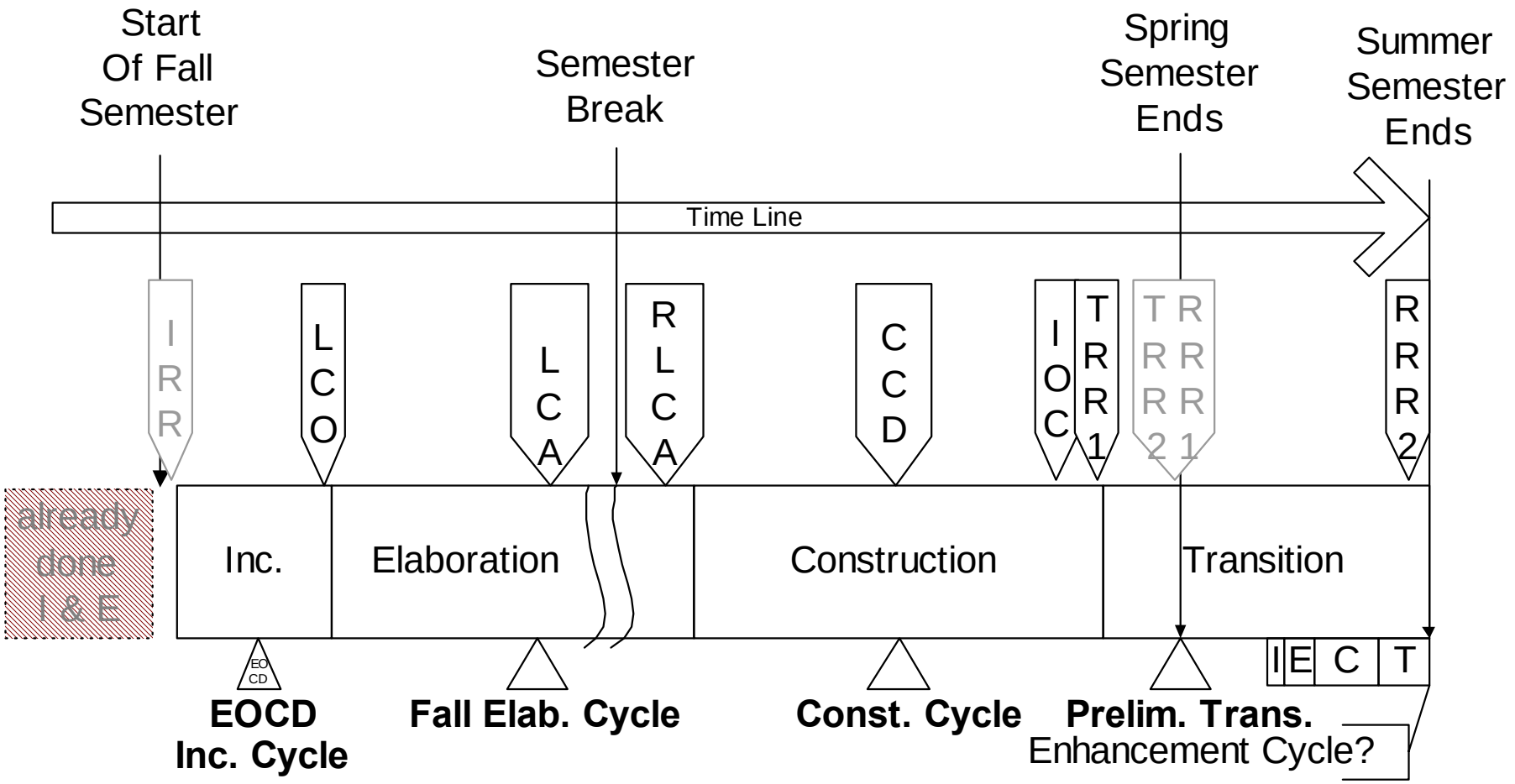
WinWin Spiral Model

- Risk Driven Selection, Execution and Validitaion of Activities and Products
- Feasibility "demonstration" needed to proceed
- Stakeholder concurrence to proceed at major milestones
- Life Cycle Process(s) Model Generator:
  Select and document/plan for next "rounds"
  a specific "Development Process Model".

Stylized WinWin Spiral with Activities Mapped to Original Spiral
[radial dimension (cost) not to scale]



WWSM: CSCI577 Unrolled With Repeated Cycles



Key: IRR - Inception Readiness Review
LCO - Life Cycle Objective
LCA - Life Cycle Architecture
RLCA- Rebaselined LCA
CCD- Core Capability Demo.
IOC- Initial Operational Capability
TRR- Transition Readiness Review
PRR - Product Release Review