Software Cost Estimation Meets Software Diversity

Barry Boehm, USC

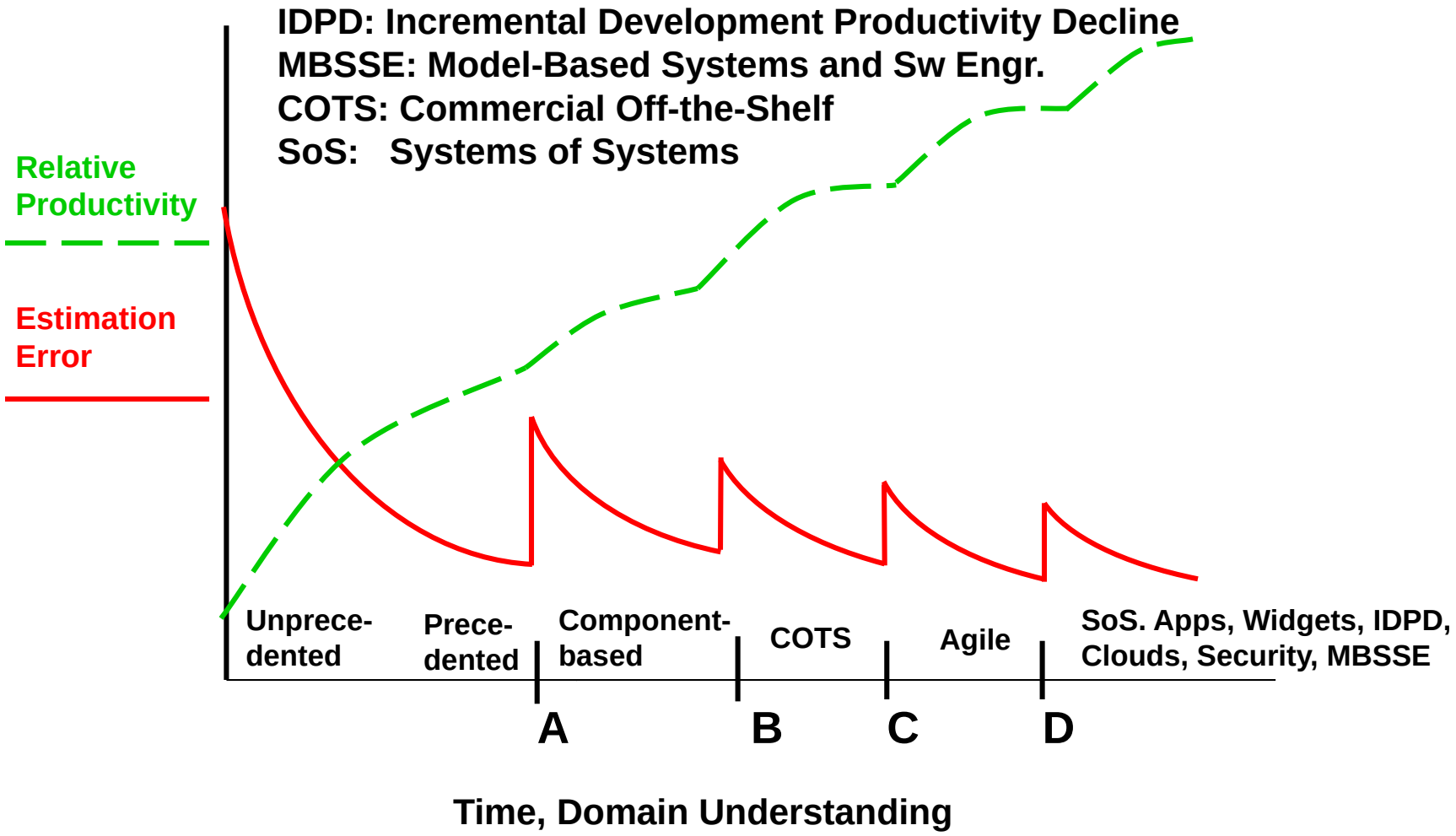
STC 2017 Keynote

September 26, 2017

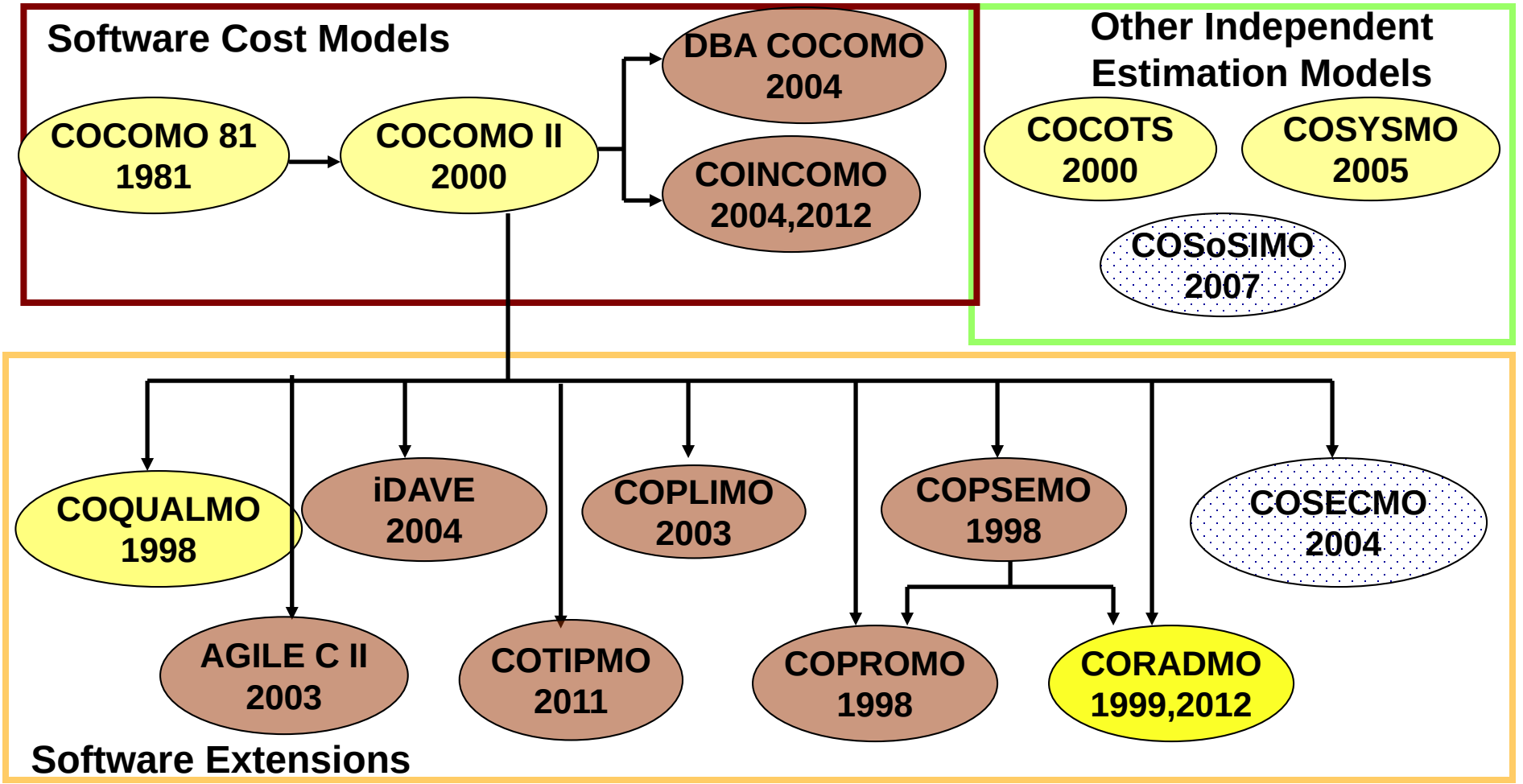
Outline

- **Sources of Software Diversity**
  - A Short History of Software Estimation Accuracy
  - Process, Product, Property, and People Drivers
- **Options for Software Cost Estimation**
  - Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites
- **Best Fits of Estimation-Types to Diversity-Types**
  - Extensions of ICSM Common Cases
- **Charting Your Path to Improved Estimates**

A Short History of Software Estimation Accuracy



COCOMO Family of Cost Models



Legend:
Model has been calibrated with historical project data and expert (Delphi) data
Model is derived from COCOMO II
Model has been calibrated with expert (Delphi) data

Dates indicate the time that the first paper was published for the model

Future Software Process Diversity

- Sequential Phases
  - Waterfall, V-Model
- Sequential Increments
  - Most agile methods: XP, Scrum, Crystal , SAFE
  - Pre-Planned Product Improvement (P3I)
- Continuous reprioritization
  - Kanban, DevOps
- Evolutionary Definition and Development
  - Incremental Commitment Spiral, Rational Unified Process
- Fully concurrent: Open Source

ICSM Common Case Examples

Accounting Application

Size/Complexity: Small/low
Typical Change Rate/Month: Low
Criticality: High
NDI Support: NDI-driven architecture
Organizational Personnel Capability: NDI-experienced, medium to high

Software Strategy: COTS

Simple Customer Business App

Size/Complexity: Small/low
Typical Change Rate/Month: Medium to high
Criticality: Medium
NDI Support: No COTS, development and target environment well-defined
Organizational Personnel Capability: Agile-ready, domain experience high

Software Strategy: Architected agile

Cellphone Feature

Size/Complexity: Medium/medium
Typical Change Rate/Month: Medium to high
Criticality: Low
NDI Support: No COTS, development and target environment well-defined
Organizational Personnel Capability: Agile-ready, domain experience high

Software Strategy: Agile

Security Kernel

Size/Complexity: Small/low
Typical Change Rate/Month: Low
Criticality: Extra high
NDI Support: No COTS, development and target environment well-defined
Organizational Personnel Capability: Strong formal methods experience

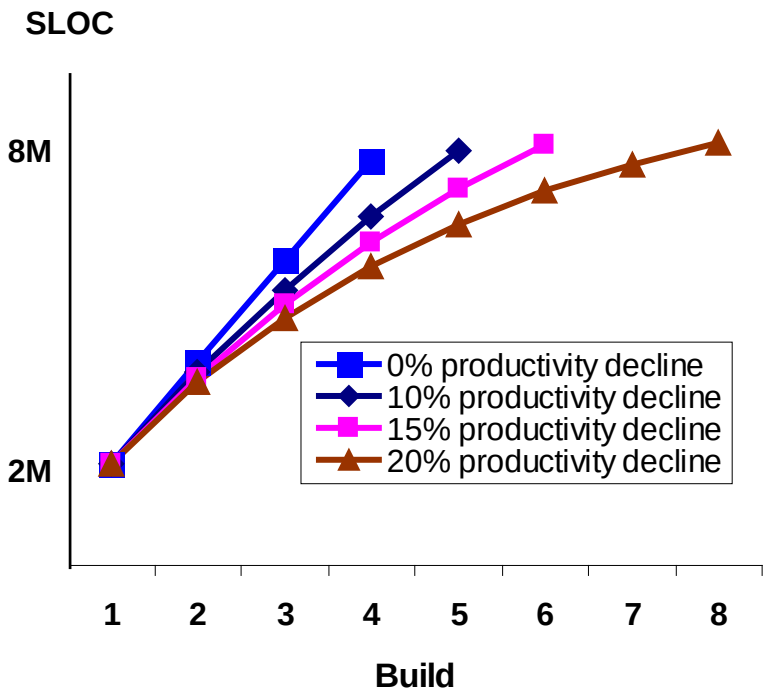
Software Strategy: Formal methods

Incremental Development Productivity Decline (IDPD)

- **Example: Site Defense BMD Software**
  - **5 builds, 7 years, $100M; operational and support software**
  - **Build 1 productivity over 300 LOC/person month**
  - **Build 5 productivity under 150 LOC/PM**
    - **Including Build 1-4 breakage, integration, rework**
    - **318% change in requirements across all builds**
    - **IDPD factor = 20% productivity decrease per build**
  - **Similar trends in later unprecedented systems**
  - **Not unique to DoD: key source of Windows Vista delays**
- **Maintenance of full non-COTS SLOC, not ESLOC**
  - **Build 1: 200 KSLOC new; 200K reused@20% = 240K ESLOC**
  - **Build 2: 400 KSLOC of Build 1 software to maintain, integrate**

Effects of IDPD on Number of Increments

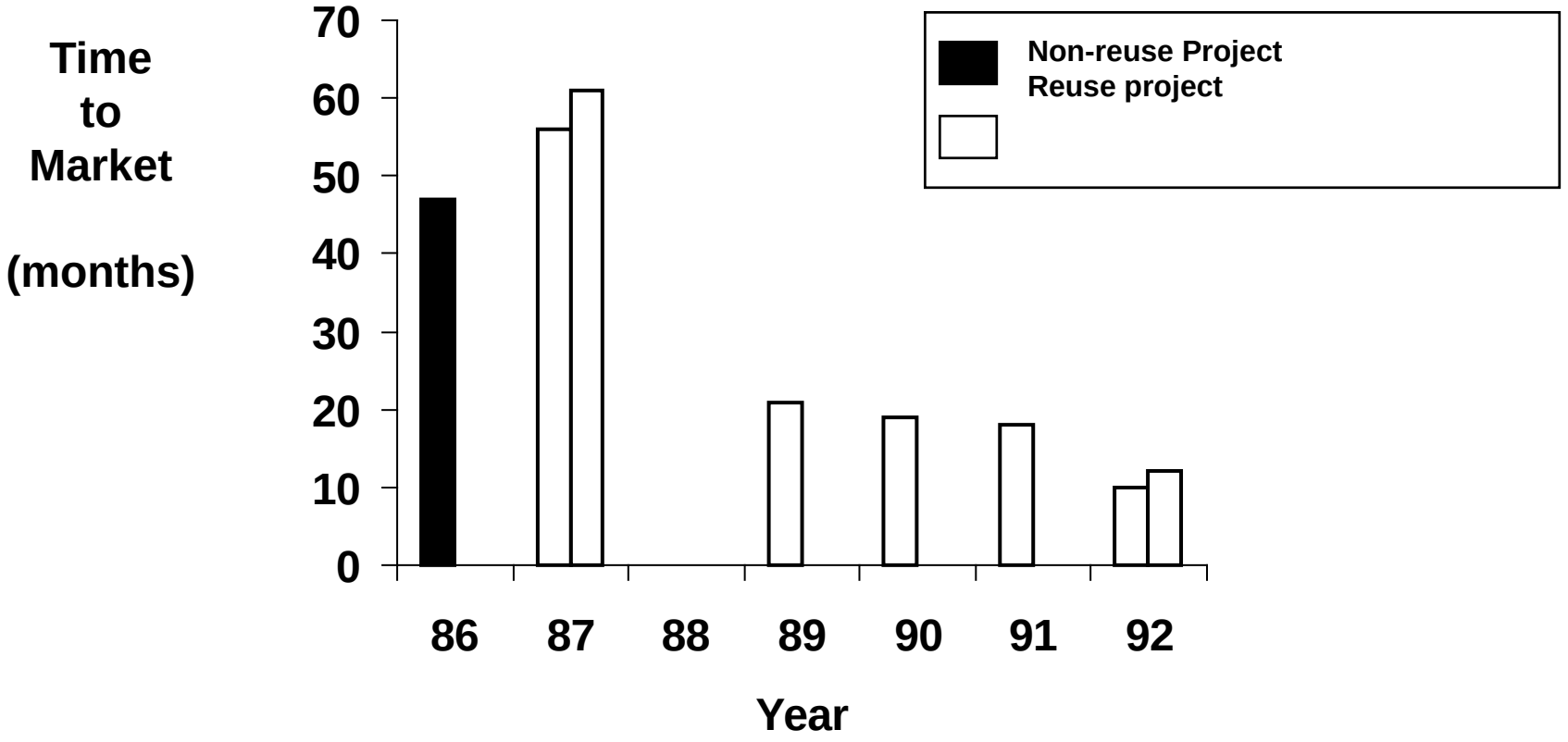
- Model relating productivity decline to number of builds needed to reach 8M SLOC Full Operational Capability
- Assumes Build 1 production of 2M SLOC @ 100 SLOC/PM
  - 20000 PM/ 24 mo. = 833 developers
  - Constant staff size for all builds
- Analysis varies the productivity decline per build
  - Extremely important to determine the incremental development productivity decline (IDPD) factor per build



Future Software Product Diversity

- Developed, Reused, Generated Software
  - Source Lines of Code (SLOC), Function Points (FP)
  - Reused: Equivalent SLOC
  - Generated: Model Directives
- Product Line Definition and Development
  - Reused, Modified, Generated SLOC or FP
- Non-Developmental Items (NDI), Cloud Services
  - NDI: Commercial Off-the-Shelf (COTS), Open Source
  - Costing: Assessment, Tailoring, Glue Code, New-Release Adaptation
- Domain Languages: Business, Supply Chain, Space
- Datasource-Driven: Selection Criteria

Reuse at HP's Queensferry Telecommunication Division



Multi-Mission Support Systems Costing

- **Product Line Engineering**
  - **Identify multi-mission commonalities and variabilities**
  - **Identify fully, partially sharable commonalities**
  - **Develop plug-compatible interfaces for variabilities**
- **Product Line Costing (COPLIMO) Parameters**
  - **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**
- **Product Line Life Cycle Challenges**
  - **Layered services vs. functional hierarchy**
  - **Modularization around sources of change**
  - **Version control, COTS refresh, and change prioritization**
  - **Balancing agilty, assurance, and affordability**

The Basic COPLIMO Model
- Constructive Product Line Investment Model

- Based on COCOMO II software cost model
  - Statistically calibrated to 161 projects, representing 18 diverse organizations
- Based on standard software reuse economic terms
  - RCR: Relative cost of reuse
  - RCWR: Relative cost of writing for reuse
- Avoids overestimation
  - Avoids RCWR for non-reused components
- Provides experience-based default parameter values
- Simple Excel spreadsheet model
  - Easy to modify, extend, interoperate

Basic COPLIMO Output Summary

Summary of Inputs:

AVPROD	300	
AVSIZE	50000	(SLOC)
UNIQ%	40	(%)
ADAP%	30	(%)
RUSE%	30	(%)
RCR-UNIQ	100	(%)
RCR-ADAP	40	(%)
RCR-RUSE	5	(%)
RCWR	1.7	

(Note: Do not change above values!)
(Change from "Input" sheet.)

7 year Product Line Effort Savings:

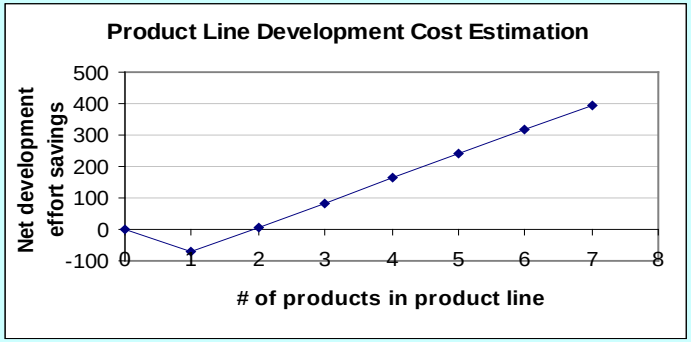


Table of Results:

# of Products	0	1	2	3	4	5	6	7
Unique SLOC	0	20000	40000	60000	80000	100000	120000	140000
Adapted SLOC	0	15000	30000	45000	60000	75000	90000	105000
Reused SLOC	0	15000	30000	45000	60000	75000	90000	105000
Total Non-PL SLOC	0	50000	100000	150000	200000	250000	300000	350000
Non-PL Effort (PM)	0	166.667	333.333	500	666.667	833.333	1000	1166.667
1-Product Equiv. SLOC	0	71000	26750	26750	26750	26750	26750	26750
1-Product Equiv. Effort	0	236.667	89.1667	89.1667	89.1667	89.1667	89.1667	89.16667
Cum. Equiv. PL SLOC	0	71000	97750	124500	151250	178000	204750	231500
Cum. PL Effort	0	236.667	325.833	415	504.167	593.333	682.5	771.6667
PL Effort Savings	0	-70	7.5	85	162.5	240	317.5	395
PL Reuse Investment	0	70						
Return on Investment	N/A	-1	0.10714	1.21429	2.32143	3.42857	4.53571	5.642857

Persistence of Legacy Systems

- **Before establishing new-system increments**
  - **Determine how to undo legacy system**

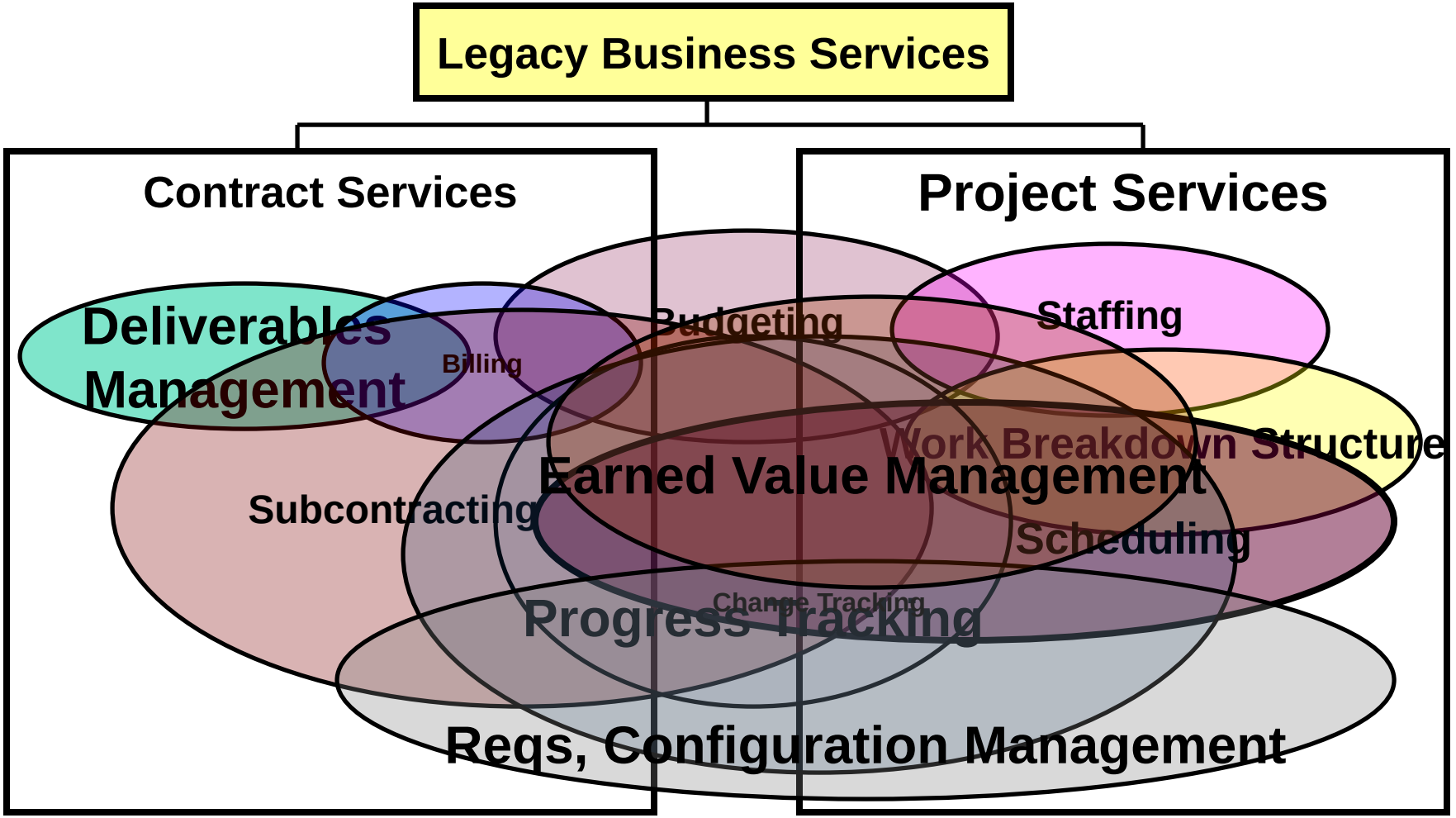
1939's Science Fiction World of 2000

Actual World of 2000

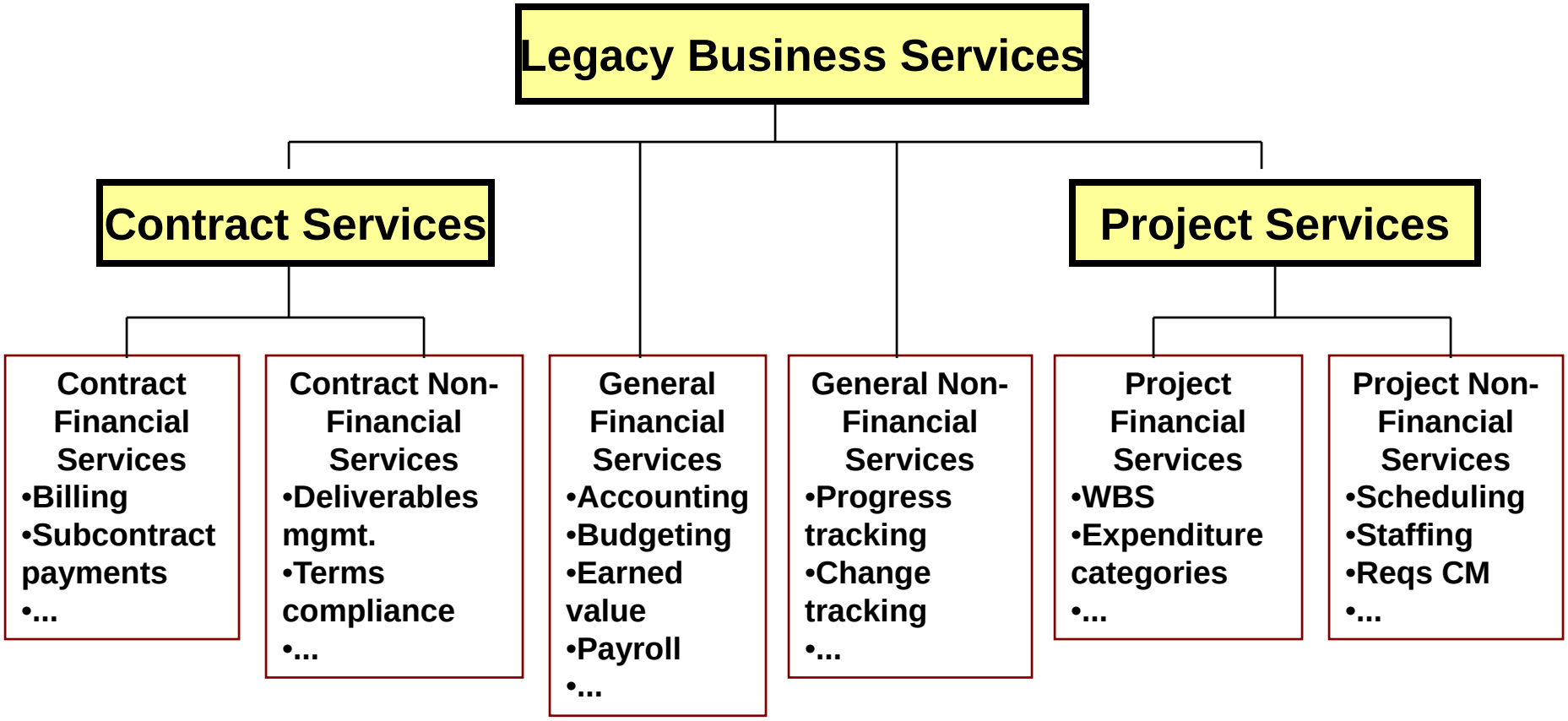
Failed Greenfield Corporate Financial System

- Used waterfall approach
  - Gathered requirements
  - Chose best-fit ERP system
  - Provided remaining enhancements
- Needed to ensure continuity of service
  - Planned incremental phase-in of new services
- Failed due to inability to selectively phase out legacy services
  - Dropped after 2 failed tries at cost of $40M

Legacy Systems Patched, Highly Coupled Financial and Non-Financial Services



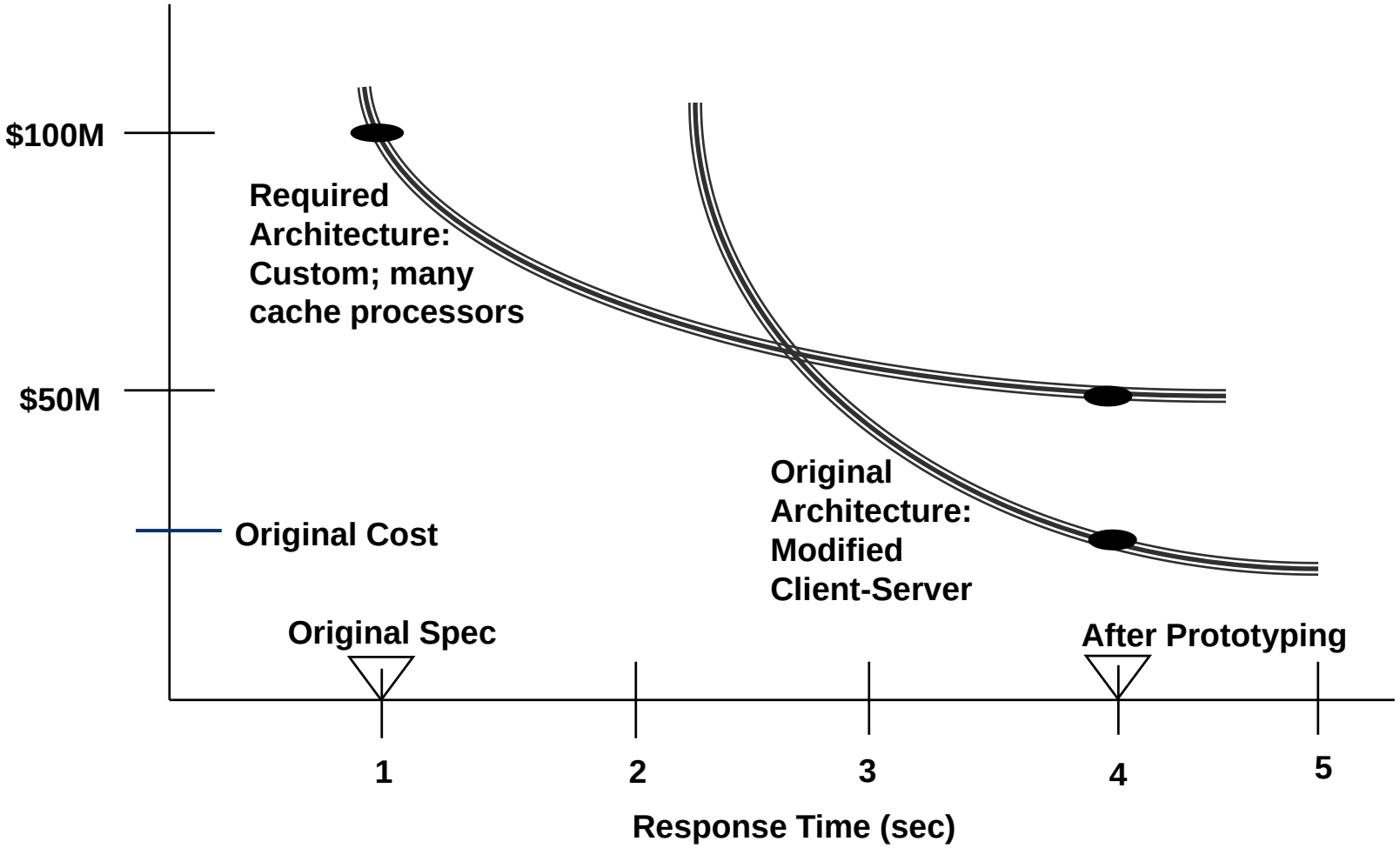
Result of Legacy Re-engineering



Future Software Properties Diversity

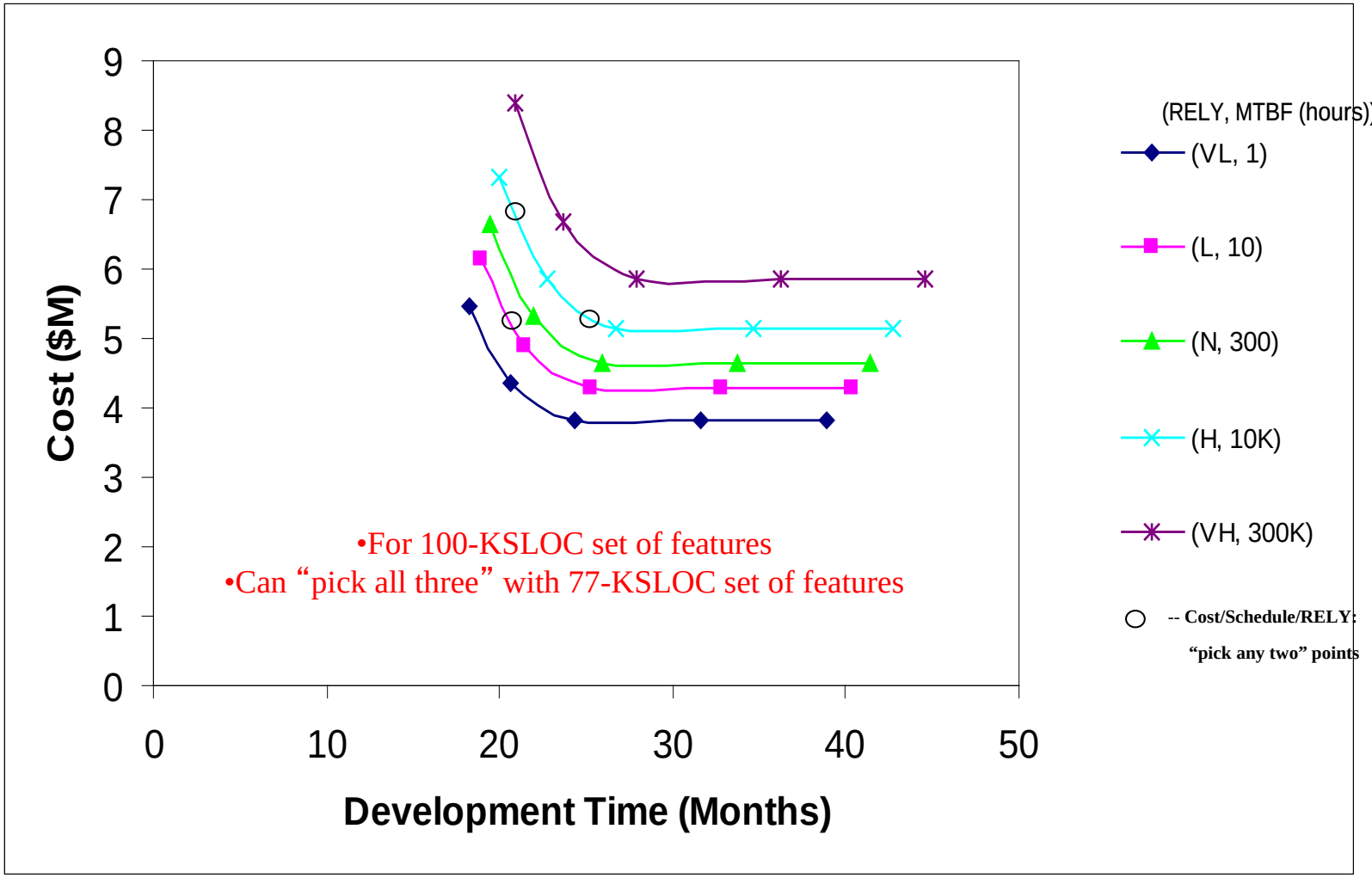
- **Dependability**
  - **Reliability, Availability, Safety, Security**
- **Changeability**
  - **Adaptability, Maintainability, Modifiability, Repairability**
- **Mission Effectiveness**
  - **Response Time, Throughput, Accuracy, Usability, Scalability, Interoperability**
- **Life Cycle Efficiency (Cost-Effectiveness)**
  - **Development and Maintenance Cost, Schedule; Reusability**

Response Time Rqt. Impact on Cost



Better, Cheaper, Faster: Pick Any Two

COCOMO II Model Results



Future Software People Diversity

- **Desired Software People Capabilities**
  - **Software System Analysis**
  - **Software System Development**
  - **Application Domain Experience**
  - **Software Languages and Tools Experience**
  - **Software Process Maturity**
  - **Team Cohesion**
  - **Low Personnel Turnover**
  - **Familiarity with Apps, Widgets, Social Media, Data Analytics, Multimedia, Virtual Reality**

Outline

- **Sources of Software Diversity**
  - **A Short History of Software Estimation Accuracy**
  - **Process, Product, Property, and People Drivers**

➡ **Options for Software Cost Estimation**
  - **Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites**

- **Best Fits of Estimation-Types to Diversity-Types**
  - **Extensions of ICSM Common Cases**

- **Charting Your Path to Improved Estimates**

Estimation-Type Options

- Expert-Judgement; Stakeholder Consensus
  - Planning Poker, Wideband Delphi, Bottom-Up
- Analogy: Previous Projects; Yesterday's Weather
  - Agile COCOMO II, Case-Based Reasoning, Causal Modeling
- Parametric Models
  - COCOMO/COSTAR, Knowledge Plan, SEER, SLIM, True-S
- Resource-Limited
  - Cost or Schedule as Independent Variable (CAIV, SAIV)
- Reuse-Driven: Equivalent Size
  - Adjusted for %Design,Code,Test Modified, Understandability
- Product Line
  - % Development for Reuse; % Development with Reuse

Outline

- **Sources of Software Diversity**
  - **A Short History of Software Estimation Accuracy**
  - **Process, Product, Property, and People Drivers**
- **Options for Software Cost Estimation**
  - **Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites**
- **Best Fits of Estimation-Types to Diversity-Types**
  - **Extensions of ICSM Common Cases**
- **Charting Your Path to Improved Estimates**

Best Fits of Estimation-Types to Diversity-Types

- Pure Agile: Planning Poker, Agile COCOMO II
- Architected Agile
  - COSYSMO for architecting; Planning Poker, CAIV-SAIV for sprints, releases; IDPD for large systems
- Formal Methods: $/SLOC by Evaluated Assurance Level
- NDI/Services-Intensive: Oracle, SAP, other ERP
  - RICE Objects: (R)eports, (I)nterfaces, (C)onversions, (E)nhancements
  - COCOTS, Value-Added Function Points, Agile for portions
- Hybrid Agile/Plan-Driven
  - Expert Delphi, Parametric Models, Agile for portions; IDPD
- Systems of Systems
  - COSYSMO for Integrator; Hybrid Agile/Plan-Driven for component systems
- Family of Systems: COPLIMO
- Brownfield: Experiment for refactoring; above for rebuilding

Proliferation of Estimation Types

Thanks to Capers Jones

- **Source Lines of Code (SLOC)**
  - **Physical/Logical; Executable/nonexecutable; New/reused; Programmmed/generated/translated; Added/modified/deleted**
- **Function points (FP)**
  - **Original IBM; IFPUG 2,3,4; Fast; COSMIC; Mark II, FISMA, NESMA; Unadjusted/adjusted; RICE Objects**
- **SLOC/FP backfire ratios**
  - **SPR, QSM, DAVIDS, Gartner Group**
- **Agile sizing**
  - **Story points (Planning Poker, T-shirt size); ideal person-weeks**
- **Risky: high variability**
  - **Number of requirements/shalls; nonfunctional requirements (SNAP points); UML diagram counts**

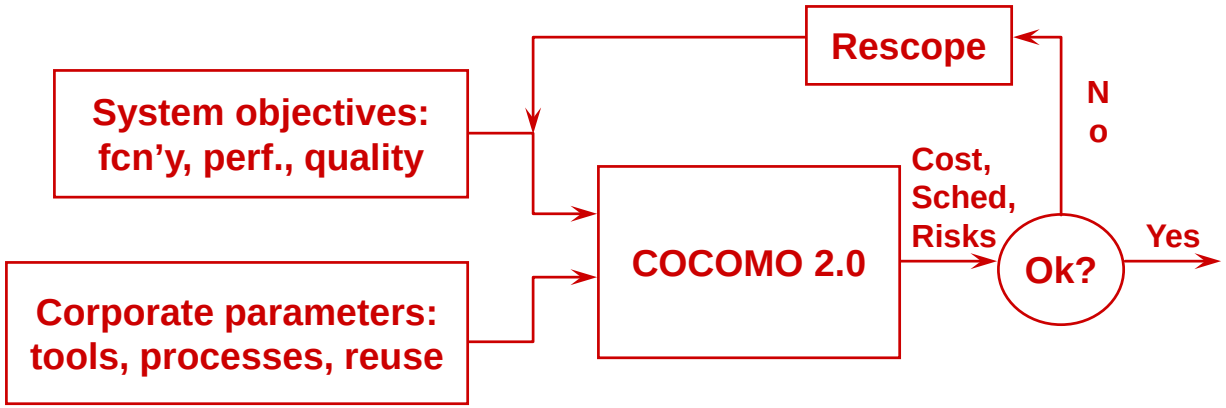
Outline

- **Sources of Software Diversity**
  - **A Short History of Software Estimation Accuracy**
  - **Process, Product, Property, and People Drivers**
- **Options for Software Cost Estimation**
  - **Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites**
- **Best Fits of Estimation-Types to Diversity-Types**
  - **Extensions of ICSM Common Cases**
- **Charting Your Path to Improved Estimates**

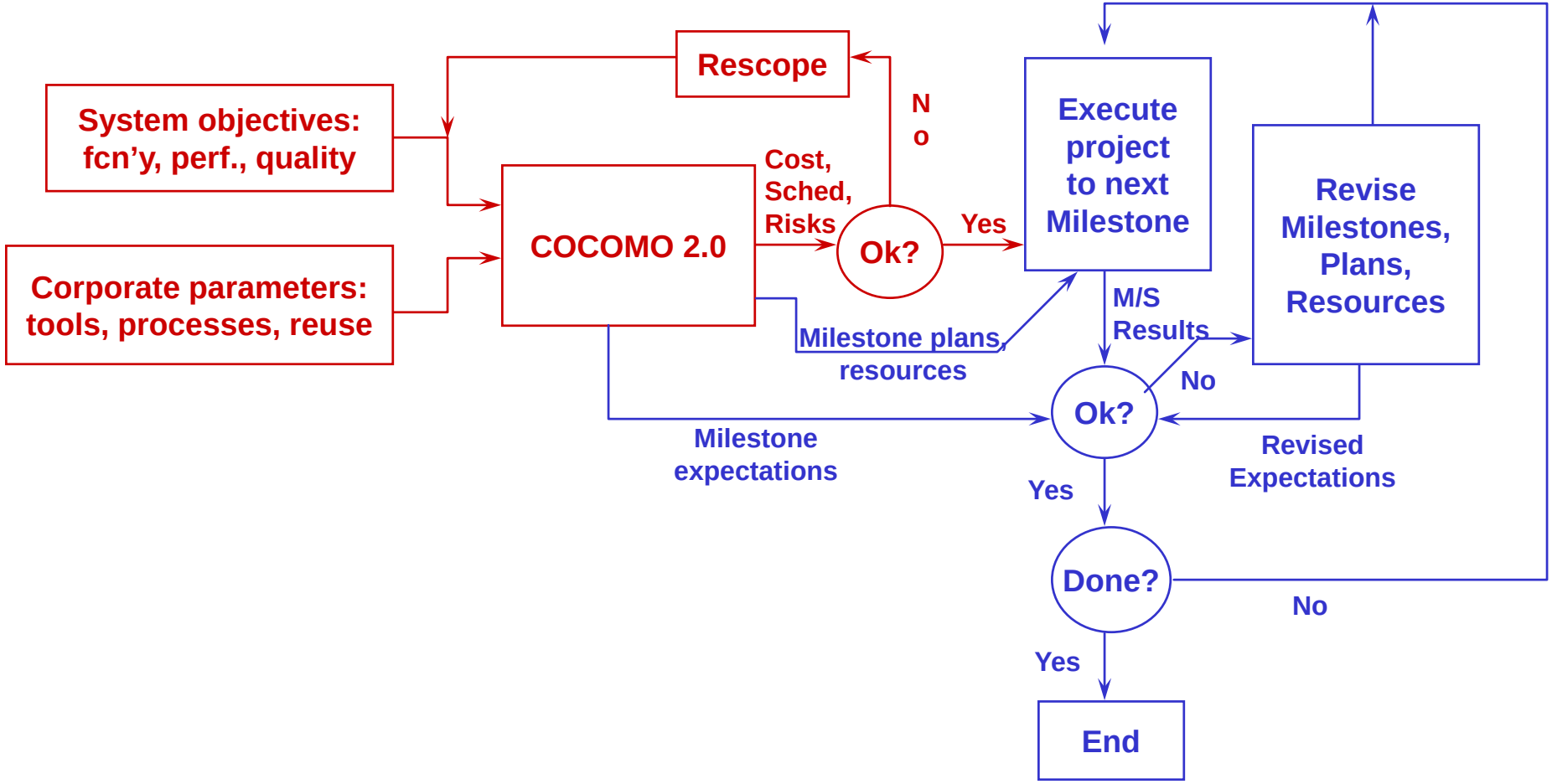
Charting Your Path to Improved Estimates

- Identify your most critical future improvement areas
- Identify, experiment with best candidate estimation methods in most critical areas
- Experiment with available methods for others; evaluate further improvement needs
- Build up, analyze experience base, use to steer path

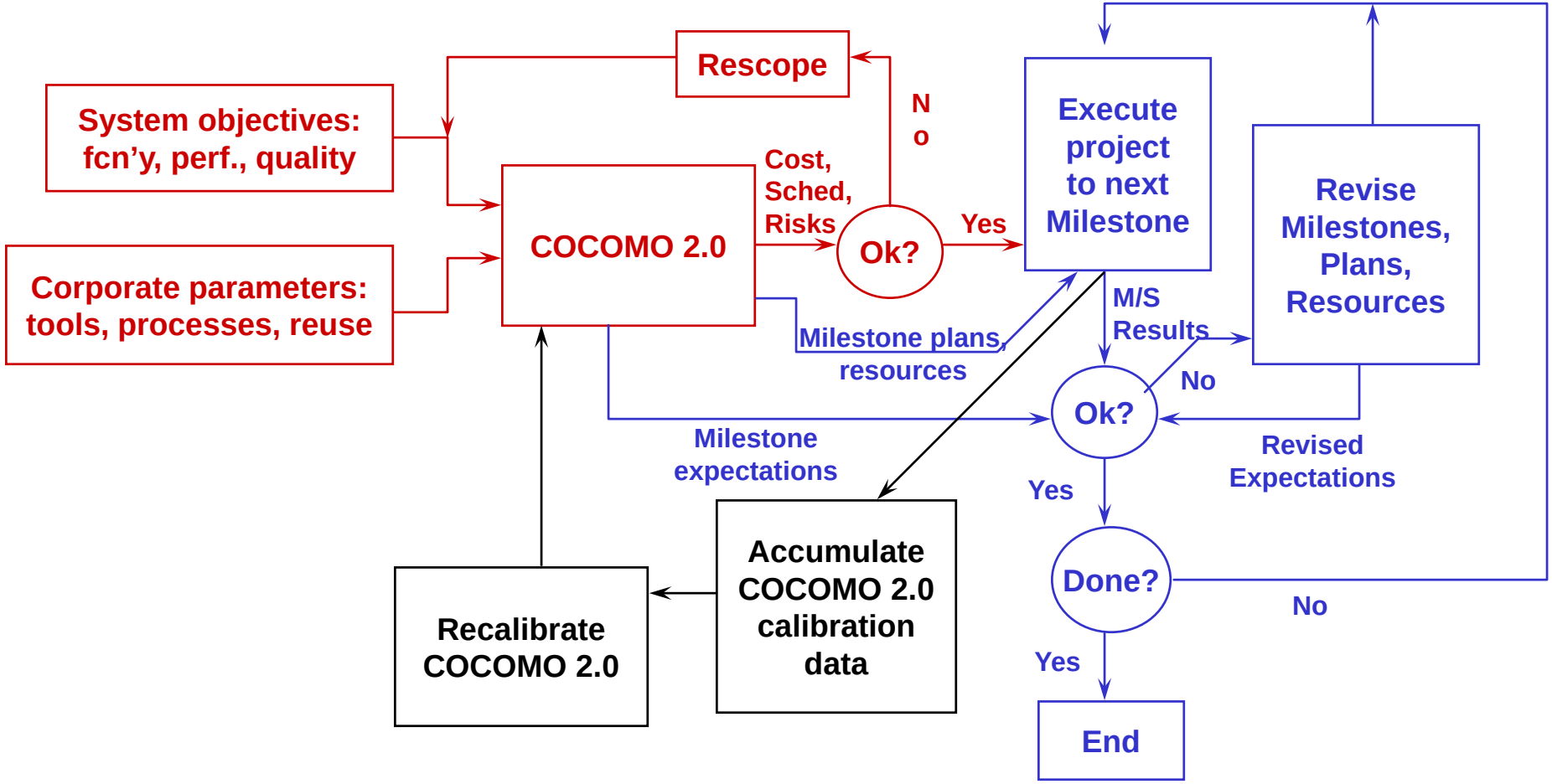
COCOMO II Experience Factory: I



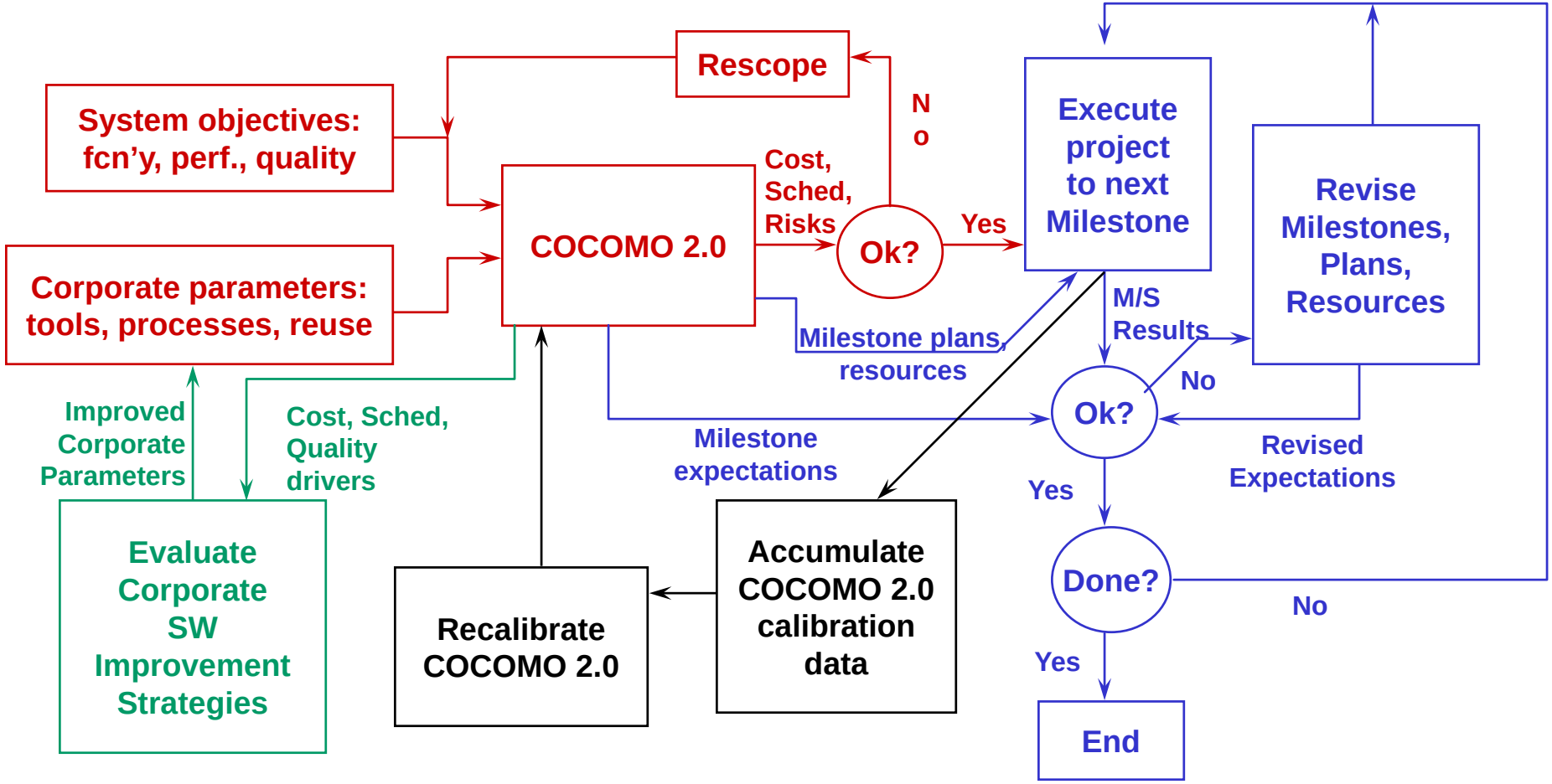
COCOMO II Experience Factory: II



COCOMO II Experience Factory: III



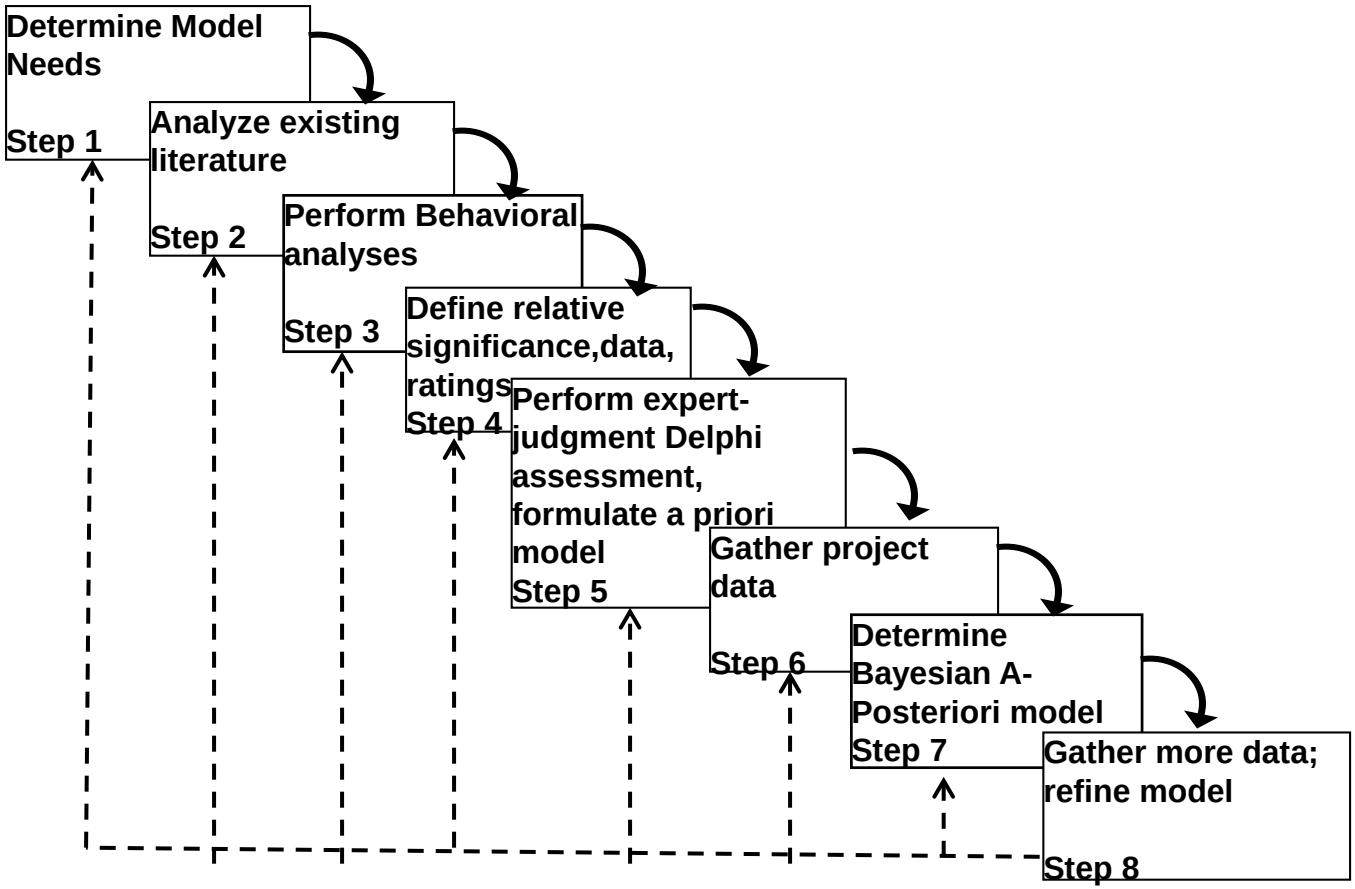
COCOMO II Experience Factory: IV



Backup Charts

USC-CSSE Modeling Methodology

- concurrency and feedback implied



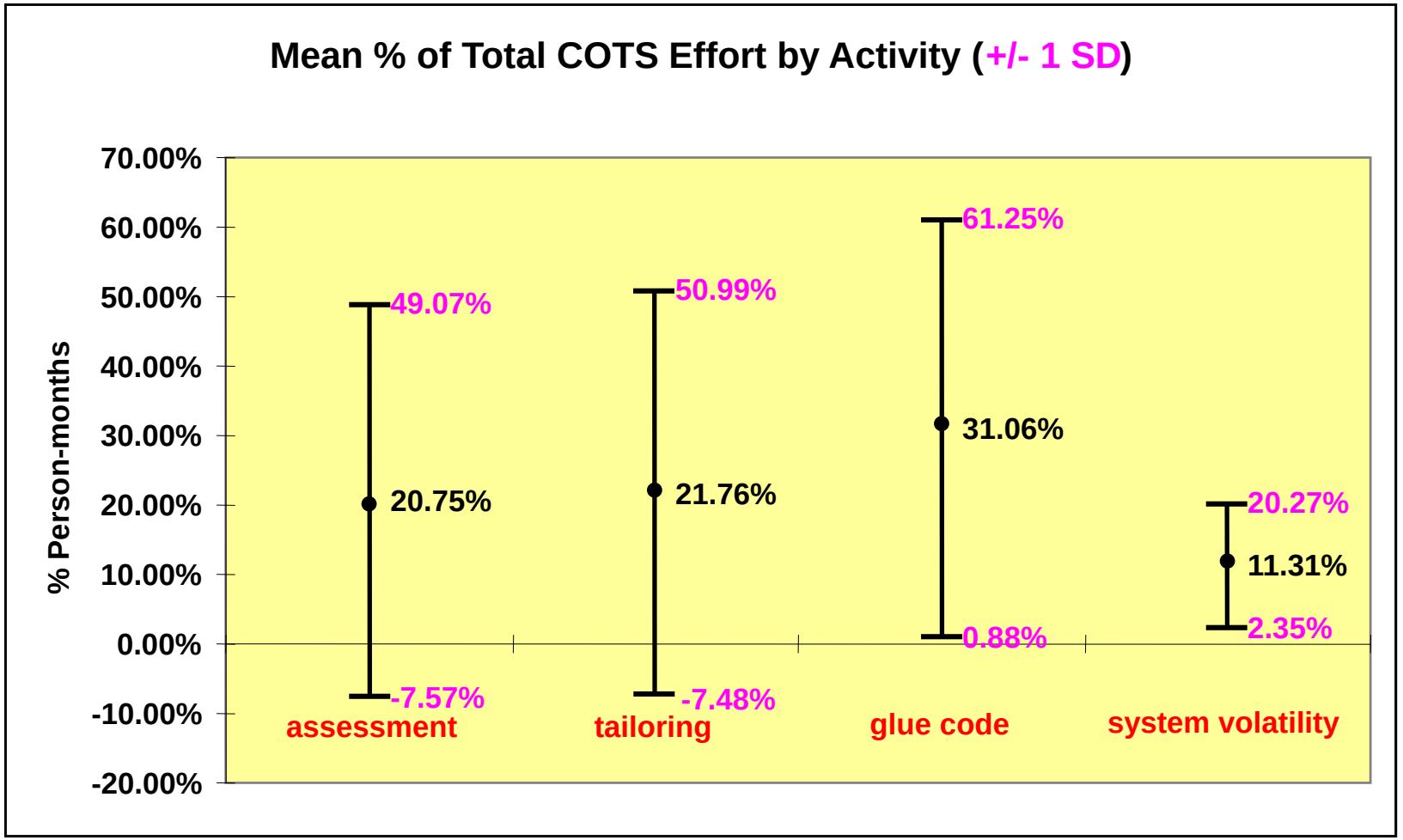
Step 6: Gather, Analyze Project Data

- Best to pilot data collection with early adopters
  - Identifies data definition ambiguities
  - Identifies data availability problems
  - Identifies need for data conditioning
- Best to collect initial data via interviews
  - Avoids misinterpretations
    - Endpoint milestones; activities included/excluded; size definitions
  - Uncovers hidden assumptions
    - Schedule vs. cost minimization; overtime effort reported

Initial Data Analysis May Require Model Revision

- Initial COCOTS model adapted from COCOMO II, with different parameters
  - Effort = A* (Size)^{B*} $\prod$(Effort Multipliers)
- Amount of COTS integration glue code used for Size
- Data analysis showed some projects with no glue code, much effort
  - Effort devoted to COTS assessment, tailoring

COCOTS Effort Distribution: 20 Projects



Revised COCOTS Model

- COCOMO-like model for glue code effort
- Unit cost approach for COTS assessment effort
  - Number of COTS products to assess
  - Number of attributes to assess, weighted by complexity
- Activity-based approach for COTS tailoring effort
  - COTS parameters setting, script writing, reports layout, GUI tailoring, protocol definitions

New Glue Code Submodel Results

- New calibration results
  - Excluding projects with very large, very small amounts of glue code
    - [0.5 - 100 KLOC]: Pred (.30) = 9/17 = 53%
    - [2 - 100 KLOC]: Pred (.30) = 8/13 = 62%
  - Previous calibration results:
    - [0.1 - 390 KLOC]: Pred (.30) = 4/13 = 31%
- Pred(.30) = percent of projects with estimates within 30% of actuals