CODEFMO
(COnstructive DEFect MOdel):
Defect Model Extension to COCOMO II

Sunita Chulani

Research Assistant

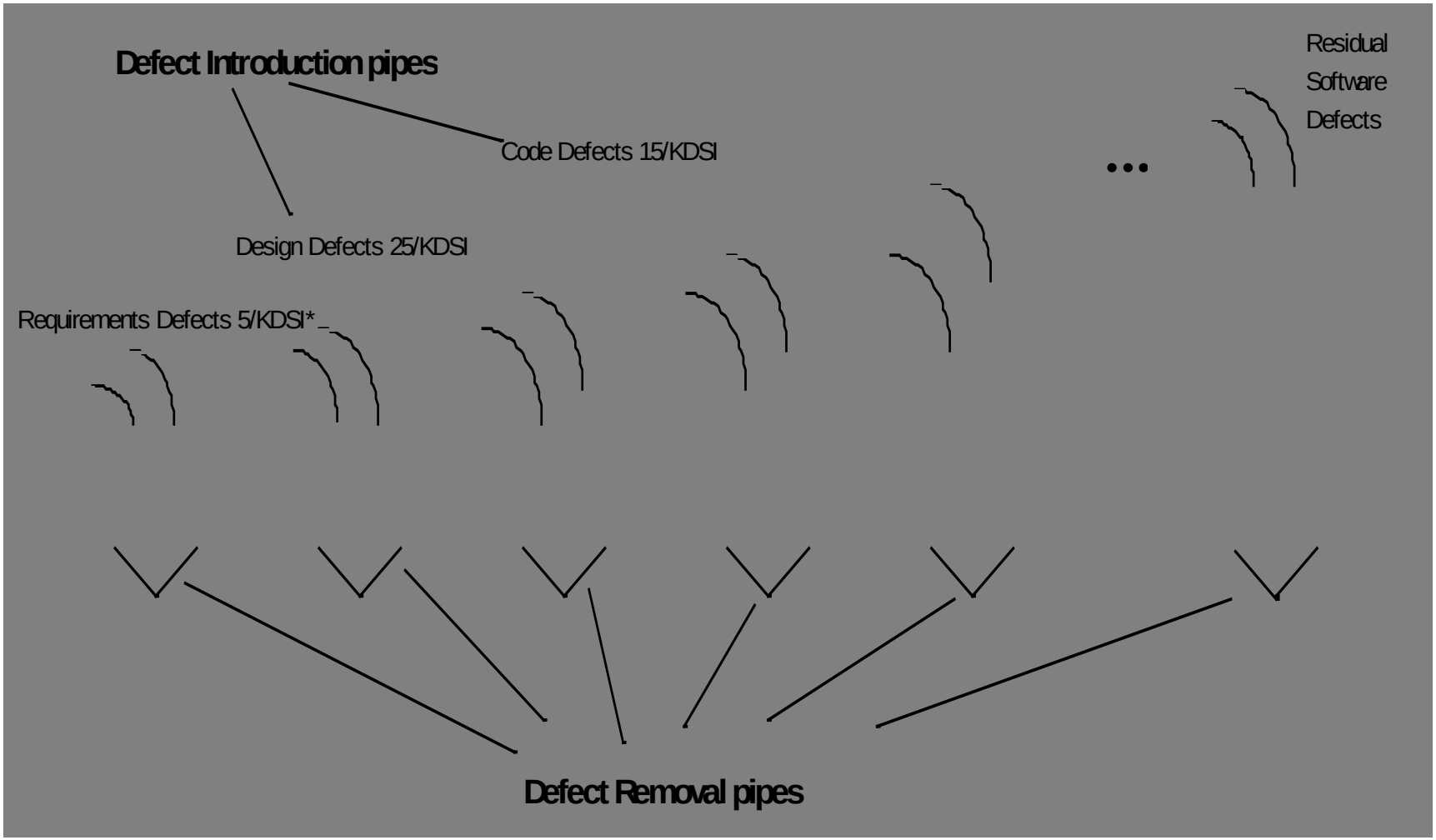
University of Southern California

Center for Software Engineering

Outline

- **Model Framework**
- **The Defect Introduction Sub-Model**
  - **Initial Data Results**
- **The Defect Removal Sub-Model**
- **CODEFMO Integrated with COCOMO II**
- **Calibration Plans**

Model Framework

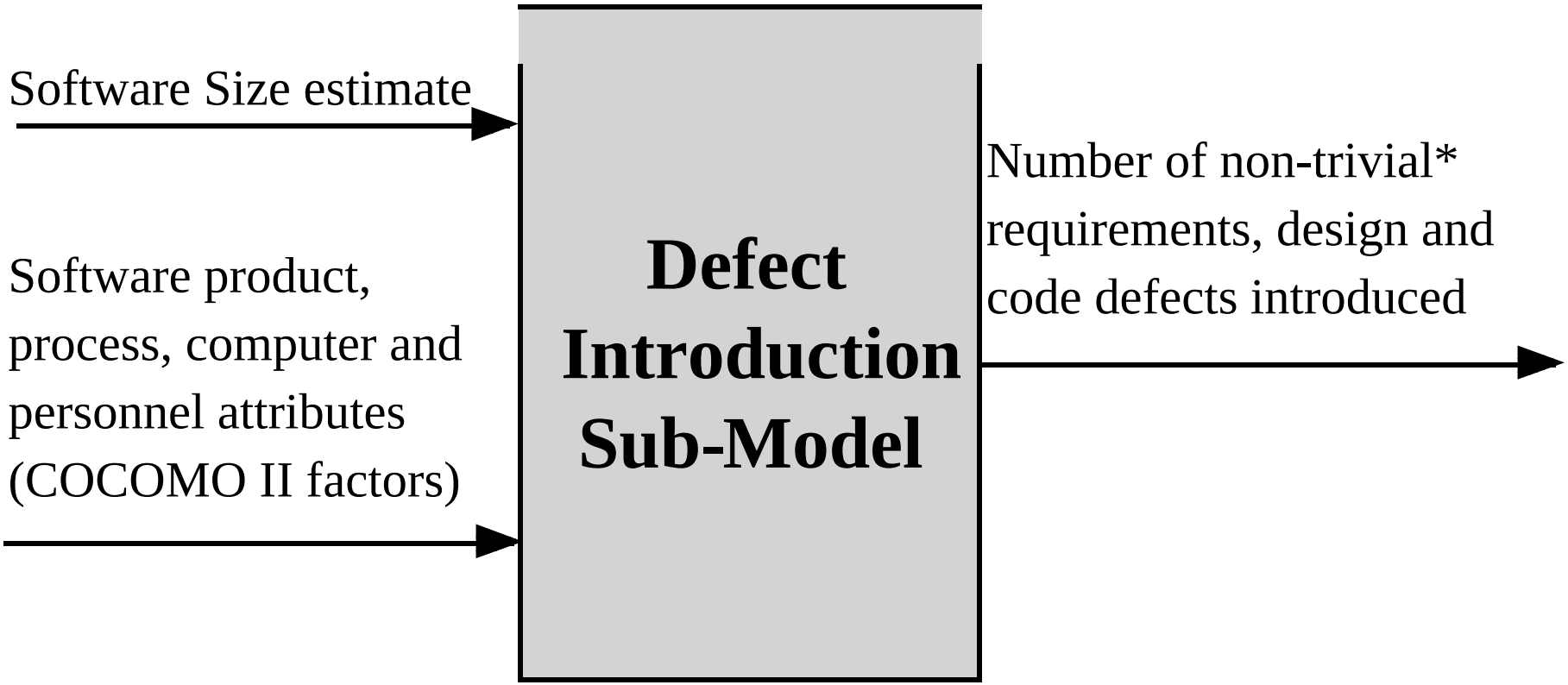


***Baseline rates of the late 1970s based on studies done by Jones, Thayer et al and Boehm**

Outline

- **Model Framework**
- ☛**The Defect Introduction Sub-Model**
  - **Initial Data Results**
- **The Defect Removal Sub-Model**
- **CODEFMO Integrated with COCOMO II**
- **Calibration Plans**

The Defect Introduction (DI) Sub-Model



*Non-trivial defects include

- **Critical** (causes a system crash or unrecoverable data loss or jeopardizes personnel
- **High** (causes impairment of critical system functions and no workaround solution exists)
- **Medium** (causes impairment of critical system function, though a workaround solution does exist).

Modeling Effects of COCOMO II Cost Drivers on DI

Defects Inserted/ KDSI or 10FPS	Requirements	Design	Code
Baseline	5	25	15

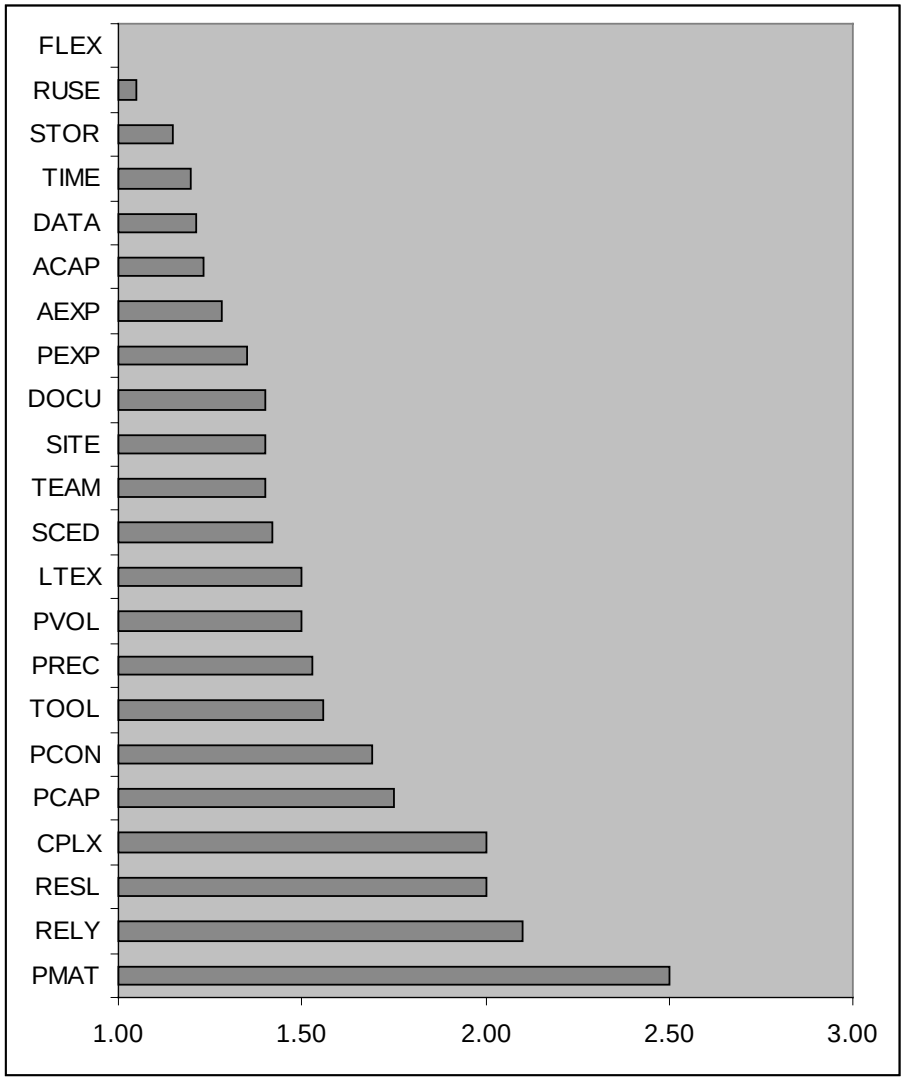
Now,

If ACAP is VH & RELY is VH	How does baseline change?	As compared to ACAP-VL & RELY-VL

An Example DI Rate Driver

AEXP (Applications Experience) level		Requirements	Design	Code
VH		Fewer Requirements defects due to less learning and fewer false starts Fewer Requirements understanding defects 0.81	Fewer Design defects due to less learning and fewer false starts Fewer Requirements traceability defects Fewer defects introduced in fixing requirements, preliminary design fixes 0.82	Fewer Coding defects due to less learning Fewer Coding defects due to requirements, design shortfalls 0.88
Nominal		Nominal level of defect introduction 1.0		
VL		More Requirements defects due to extensive learning and more false starts More Requirements understanding defects 1.24	More Design defects due to less learning and fewer false starts More Requirements traceability defects More defects introduced in fixing requirements, preliminary design fixes 1.22	More Coding defects due to extensive learning More Coding defects due to requirements, design shortfalls 1.13
Behavioural analysis; relative significance		1.56	1.56	1.32
Expert-judgment	Range	1.4-1.65	1.3-1.56	1.05-1.4
Delphi Round 1	Median	1.5	1.5	1.26
Expert-judgment	Range	1.5-1.6	1.4-1.56	1.2-1.32
Delphi Round 2	Median	1.53	1.5	1.28

A-Priori Expert-Judgment Based Code DI Ranges



DI Model Equations

- **Baseline Defect Introduction Rate - 1970s**

Type of Artifact	$DIR_{Baseline}$
Requirements Defects	5/KDSI
Design Defects	25/KDSI
Coding Defects	15/KDSI

- **Nominal Defect Introduction (DI) (like Basic COCOMO**

$$DI_{Nom\ j} = DIR_{Baseline\ j}\ X\ (Size)^{B}$$

- **Estimated Defect Introduction Rate (like Intermediate COCOMO)**

$$DI_{Est\ j} = A_j\ X \sum_{i=1}^{22} DI_{Nom\ j}\ X\ QAF_j$$

$$QAF_j = \prod_{i=1} DIR\text{-}driver_{ij}$$

Initial Data Analysis on the DI Model

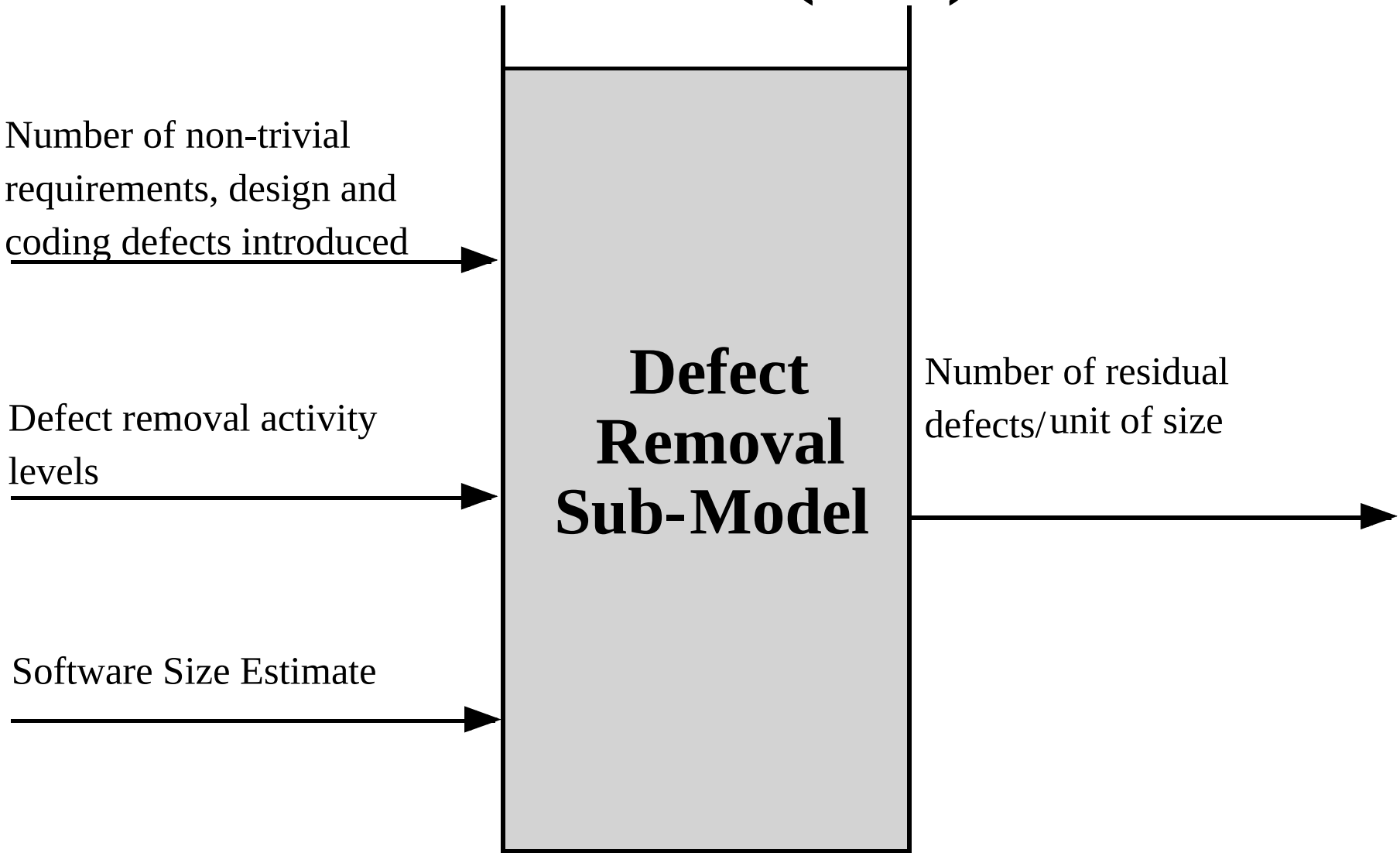
Type of Artifact	1970's Baseline DIRs	Quality Adjustment Factor	Predicted DIR	Actual DIR	Calibrated Constant (A)	1990's Baseline DIRs
Requirements	5	0.5	2.5	4.5	1.8	9
Design	25	0.44	11	8.4	0.77	19
Code	15	0.5	7.5	16.6	2.21	33

DIR = Defect Introduction Rate

Outline

- **Model Framework**
- **The Defect Introduction Sub-Model**
  - **Initial Data Results**
- **☛The Defect Removal Sub-Model**
- **CODEFMO Integrated with COCOMO II**
- **Calibration Plans**

The Defect Removal (DR) Sub-Model



DR Activity Profiles

- 3 relatively orthogonal profiles
  - Automated Analysis
  - People Reviews
  - Execution Testing and Tools
- Each profile has 6 levels
  - Very Low, Low, Nominal, High, Very High, Extra High
    - Very Low level -- removes the least number of defects
    - Extra High level -- removes the most number of defects

DR Activity Profiles

Rating	Automated Analysis	People Reviews	Execution Testing and Tools
Very Low	Simple compiler syntax checking	No people review	No testing
Low	Basic compiler capabilities for static module-level code analysis, syntax, type-checking.	Ad-hoc informal walkthroughs Minimal preparation, no follow-up	Ad-hoc testing and debugging. Basic text-based debugger
Nominal	Some compiler extensions for static module and inter-module level code analysis, syntax, type-checking. Basic requirements and design consistency, traceability checking	Well-defined sequence of preparation, review, minimal follow-up. Informal review roles and procedures	Basic unit test, integration test, system test process Basic test data management, problem tracking support Test criteria based on checklists
High	Intermediate-level module and inter-module code syntax and semantic analysis. Simple requirements/design view consistency checking	Formal review roles and procedures applied to all products using basic checklists, follow up	Well-defined test sequence tailored to organization (acceptance / alpha / beta / flight / etc.) test Basic test coverage tools, test support system Basic test process management
Very High	More elaborate requirements/design view consistency checking. Basic distributed-processing and temporal analysis, model checking, symbolic execution	Formal review roles and procedures applied to all product artifacts & changes(formal change control boards) Basic review checklists, root cause analysis. Use of historical data on inspection rate, preparation rate, fault density.	More advanced test tools, test data preparation, basic test oracle support, distributed monitoring and analysis, assertion checking Metrics-based test process management
Extra High	Formalized* specification and verification. Advanced distributed processing and temporal analysis, model checking, symbolic execution *Consistency-checkable pre-conditions and post-conditions, but not mathematical theorems.	Formal review roles and procedures for fixes, change control. Extensive review checklists, root cause analysis. Continuous review process improvement User/Customer involvement, Statistical Process Control	Highly advanced tools for test oracles, distributed monitoring and analysis, assertion checking Integration of automated analysis and test tools Model-based test process management

Residual Defects Equation

- **Estimated Number of Residual Defects**

$$DRes_{Est,j} = B_j \times DI_{Est,j} \times \prod_i (1 - DRF_{ij})$$

$DRes_{Est,j}$ = Estimated No. of Residual Defects for the j_{th} artifact
B_j = Calibration Constant for the j^{th} artifact
$DI_{Est,j}$ = Estimated No. of Defects Introduced for the j_{th} artifact (output of DI Sub-Model)
i = Defect Removal profile
DRF_{ij}= Defect Removal Fraction

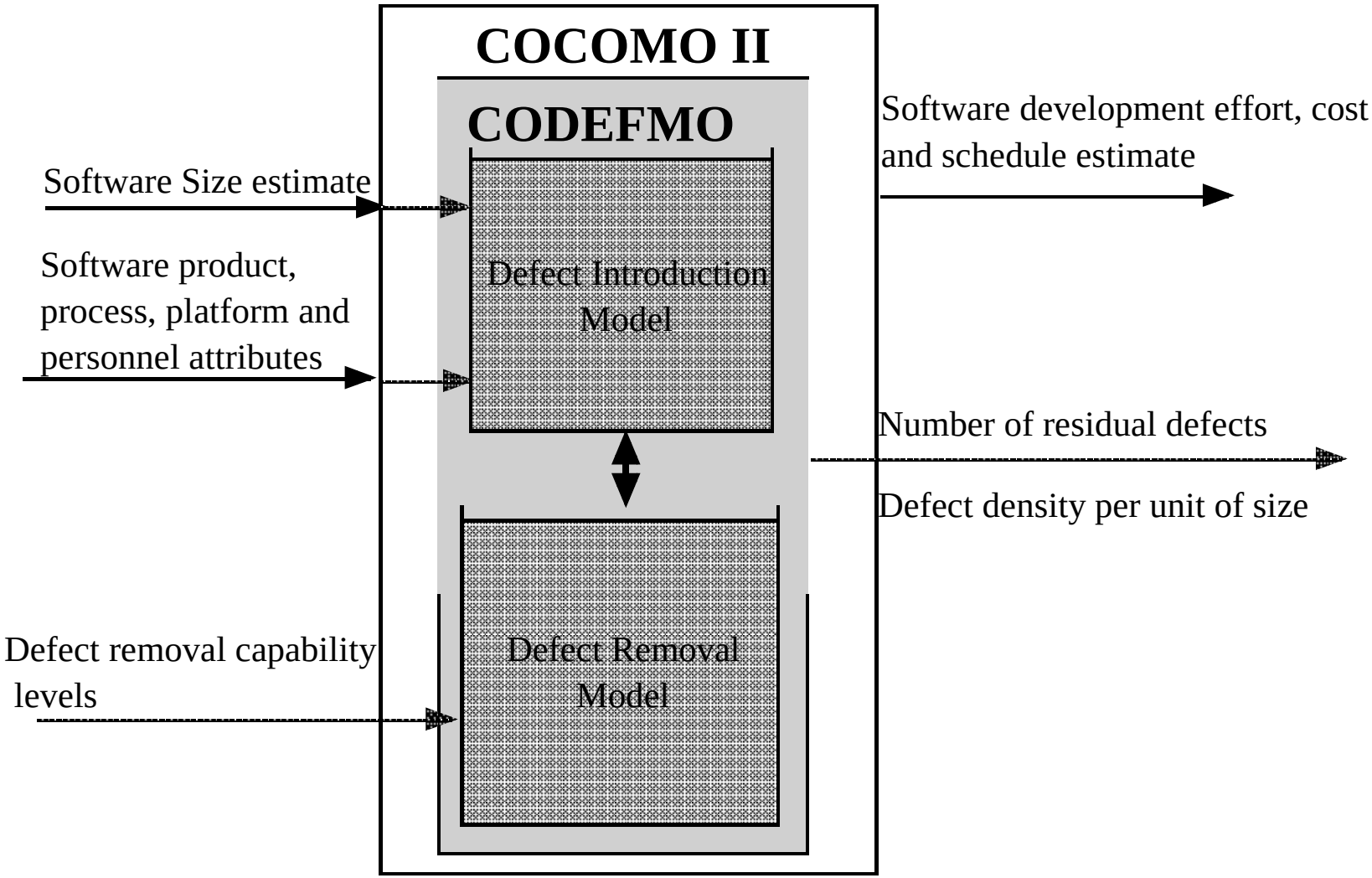
Defect Densities from Initial DRF Values (these will get updated by a Delphi process)

	Automated Analysis DRF	People Reviews DRF	Execution Testing and Tools DRF	Product ($1-DRF_{ij}$)	DI/KSLOC	DRes/KSLOC
Very Low	0	0.25	0.50	0.38	10	3.8
	0	0.30	0.60	0.28	20	5.6
	0	0.35	0.70	0.195	30	5.8
					Total:	15.2
Low	0.40	0.40	0.75	0.09	10	0.9
	0.40	0.50	0.80	0.06	20	1.2
	0	0.50	0.90	0.05	30	1.5
					Total:	3.6
Nom	0.60	0.60	0.90	0.016	10	0.16
	0.60	0.65	0.93	0.0098	20	0.20
	0.40	0.65	0.96	0.0084	30	0.25
					Total:	0.61
High	0.70	0.55	0.97	0.004	10	0.04
	0.70	0.75	0.98	0.0015	20	0.03
	0.60	0.75	0.99	0.001	30	0.03
					Total:	0.1
Very High	0.80	0.70	0.99	0.0006	10	0.006
	0.80	0.80	0.992	0.00032	20	0.0064
	0.80	0.85	0.995	0.00015	30	0.0045
					Total:	0.017

Outline

- **Model Framework**
- **The Defect Introduction Sub-Model**
  - **Initial Data Results**
- **The Defect Removal Sub-Model**
- ☛**CODEFMO Integrated with COCOMO II**
- **Calibration Plans**

Integrated CODEFMO



Outline

- **Model Framework**
- **The Defect Introduction Sub-Model**
  - **Initial Data Results**
- **The Defect Removal Sub-Model**
- **CODEFMO Integrated with COCOMO II**

☛**Calibration Plans**

Calibration Plans

- **A-Priori CODEFMO**
  - **DI Sub-Model A-Priori Model defined**
  - **DR Sub-Model A-Priori Model will be refined at breakout group**
- **Collect data on actual completed projects**
- **Adjust a-priori values based on the data-determined results using the Bayesian framework**