SOFTWARE QUALITY

CPTS 583

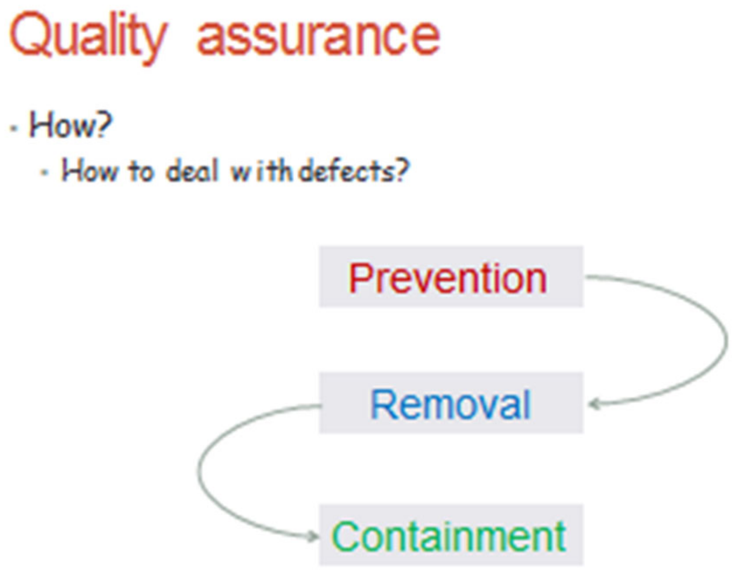
Quality Models (II): Reliability (Rayleigh) Model

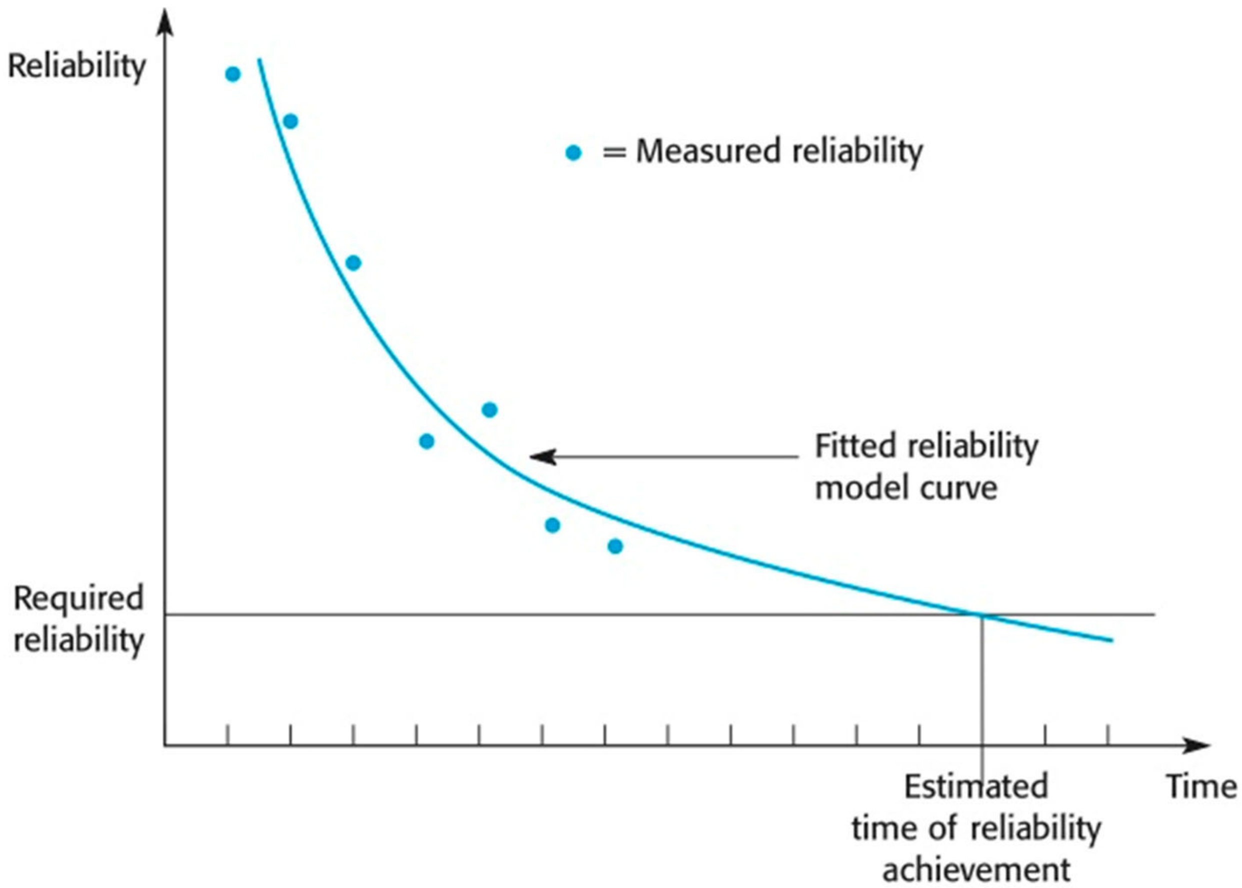
Outline

- Reliability Models
  - What & Why
- Rayleigh Model
  - Internals
  - Applications

Software Reliability Models

- Defects in software





Software Reliability Models

- What are they?

models are used to assess a software product's reliability or to estimate the number of latent defects when it is available to the customers.

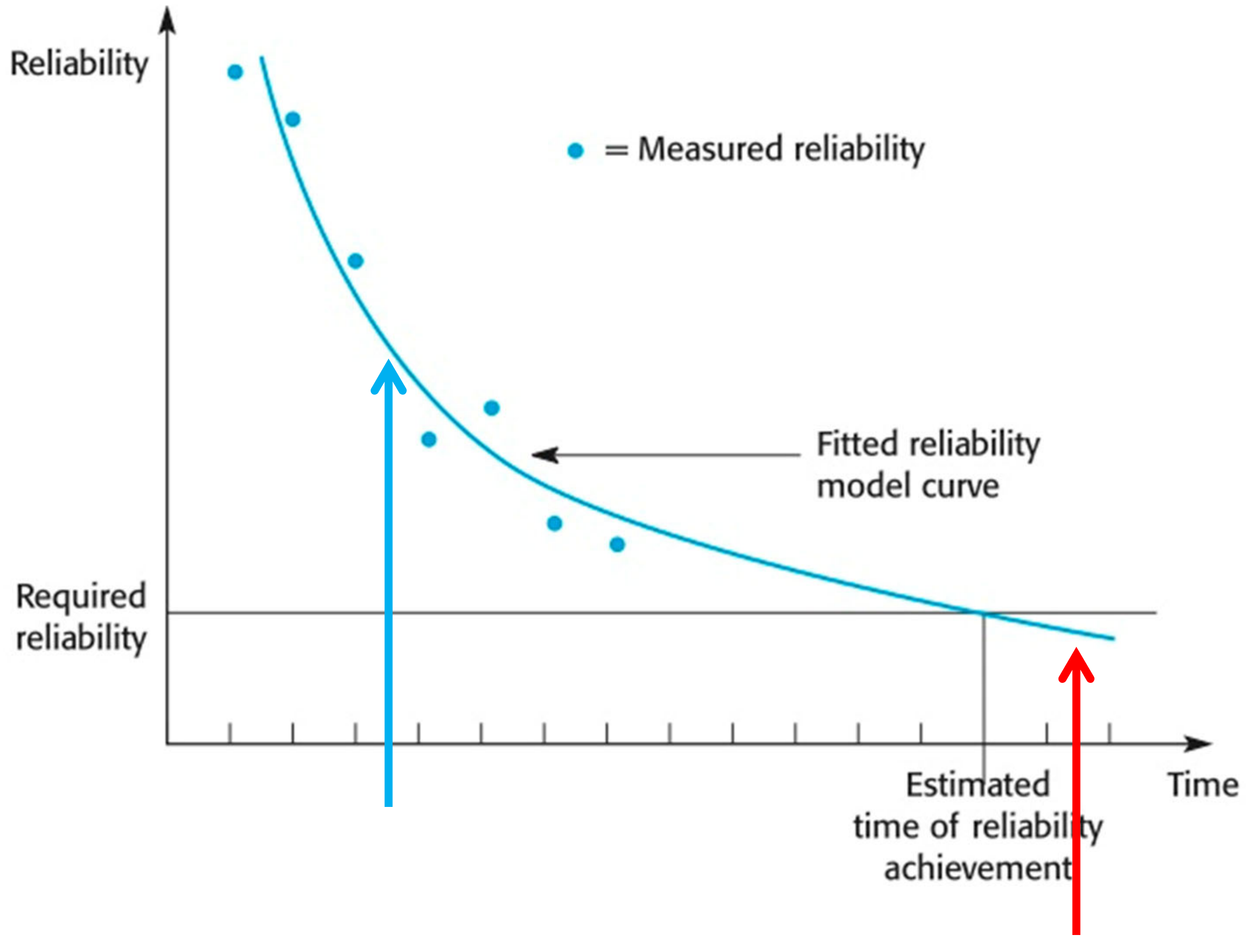
- Why do we need them?

(1) as an objective statement of the quality of the product

(2) for resource planning for the software maintenance phase

Software Reliability Models

- Prediction



Software Reliability Models

- Static models

#defects / defect rate

Product/project/process attributes

Error term

$$y = f(x_1, x_2, \ldots\ldots, xi) + e$$

- Dynamic models

Probability density function (PDF)

time

$$y = f(t)$$

Software Reliability Models

- Static versus Dynamic models

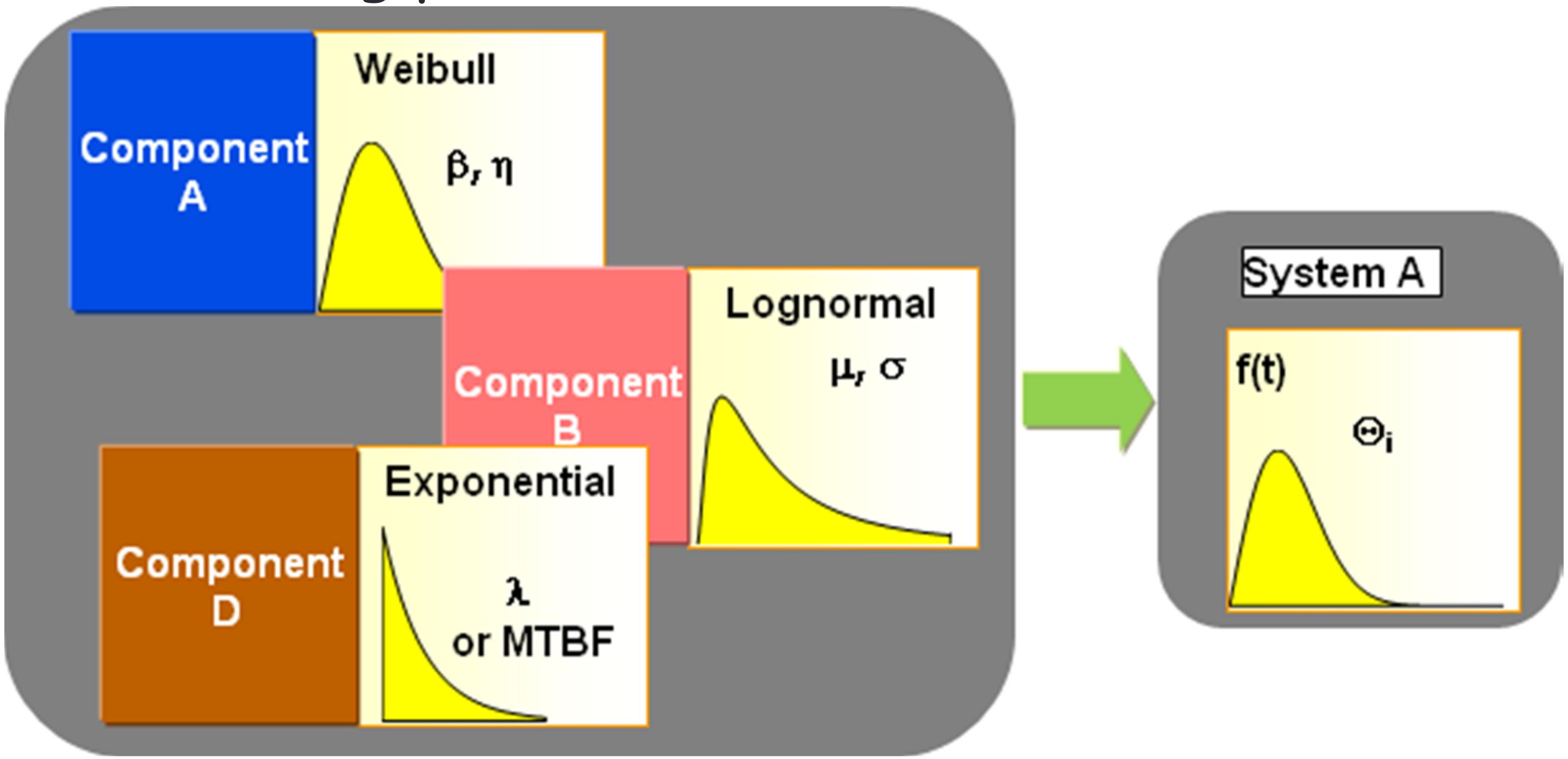
	Static	**Dynamic**
Data source for model parameters	Attributes of previous projects	Defect patterns of current project
Application scope	Process/product line level: providing clues on improving design and implementation quality	Product level: resulting model specific to the current project/product

Software Reliability Models

- Dynamic models

- For the entire development process 
  - Rayleigh Model

- For the testing phase



Rayleigh Model

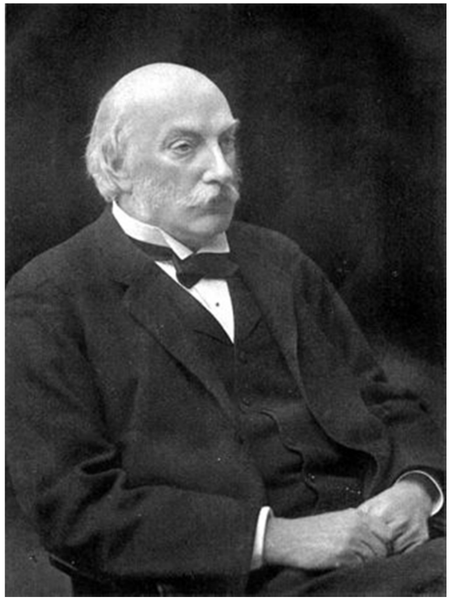
- A dynamic reliability model
- Based on Weibull distribution with m=2

PDF: $\frac{m}{t}\left(\frac{t}{c}\right)^{m} e^{-\left(\frac{t}{c}\right)^{m}}$

CDF: $1 - e^{-\left(\frac{t}{c}\right)^{m}}$

t: time
m: shape parameter
c: scale parameter

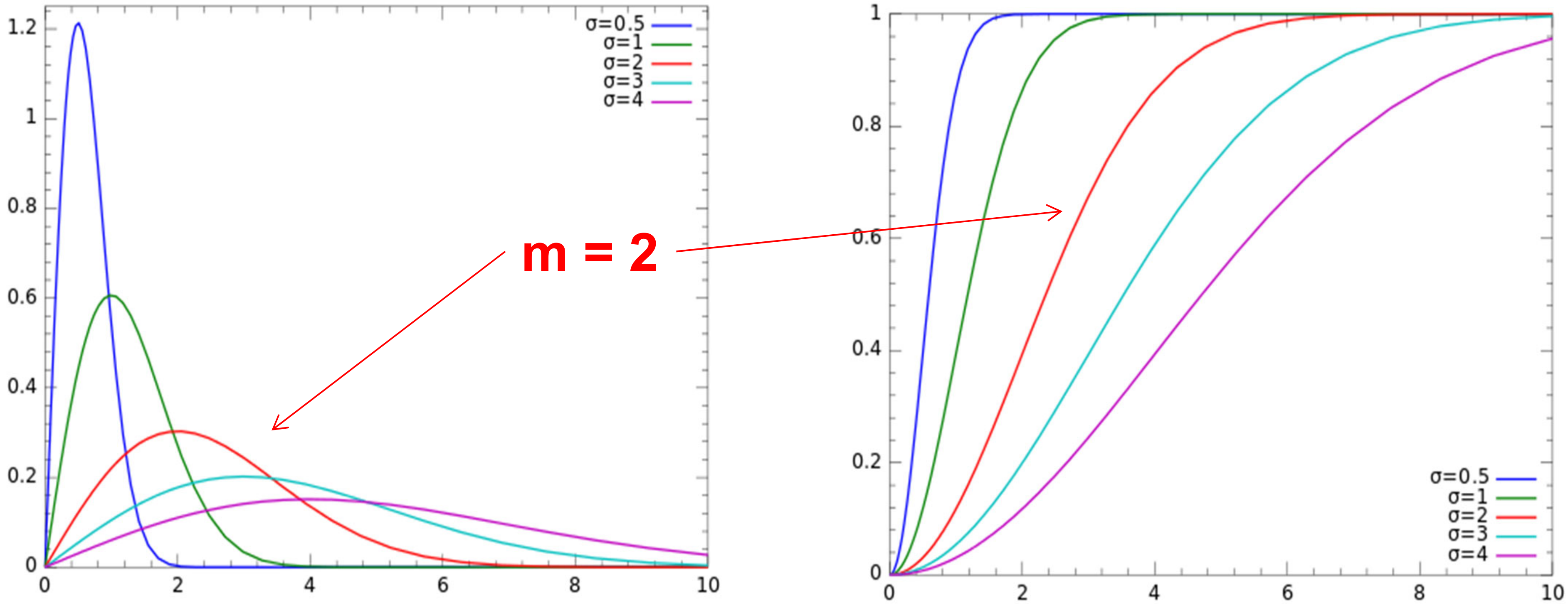
Lord Rayleigh



Physicist, discovered Argon and explained *why the sky was blue*

Rayleigh Model

- Wide applications



Probability density function (PDF)

Cumulative distribution function (CDF)

Rayleigh Model

- For software reliability

Defect density (rate) over time / defect arrival pattern

PDF: $\frac{2}{t}\left(\frac{t}{c}\right)^2 e^{-\left(\frac{t}{c}\right)^2}$

CDF: $1 - e^{-\left(\frac{t}{c}\right)^2}$

Cumulative defect arrival pattern

Rayleigh Model

- For software reliability

$$\text{PDF: } f(t) = \frac{2}{t}\left(\frac{t}{c}\right)^2 e^{-\left(\frac{t}{c}\right)^2}$$

$$f'(t) = 0$$

$$t_m = c/\sqrt{2}$$

Time at which the PDF curve reaches its peak

$$c = t_m\sqrt{2}$$

Rayleigh Model for software

- Projects follow a life-cycle pattern described by the Rayleigh density curve.

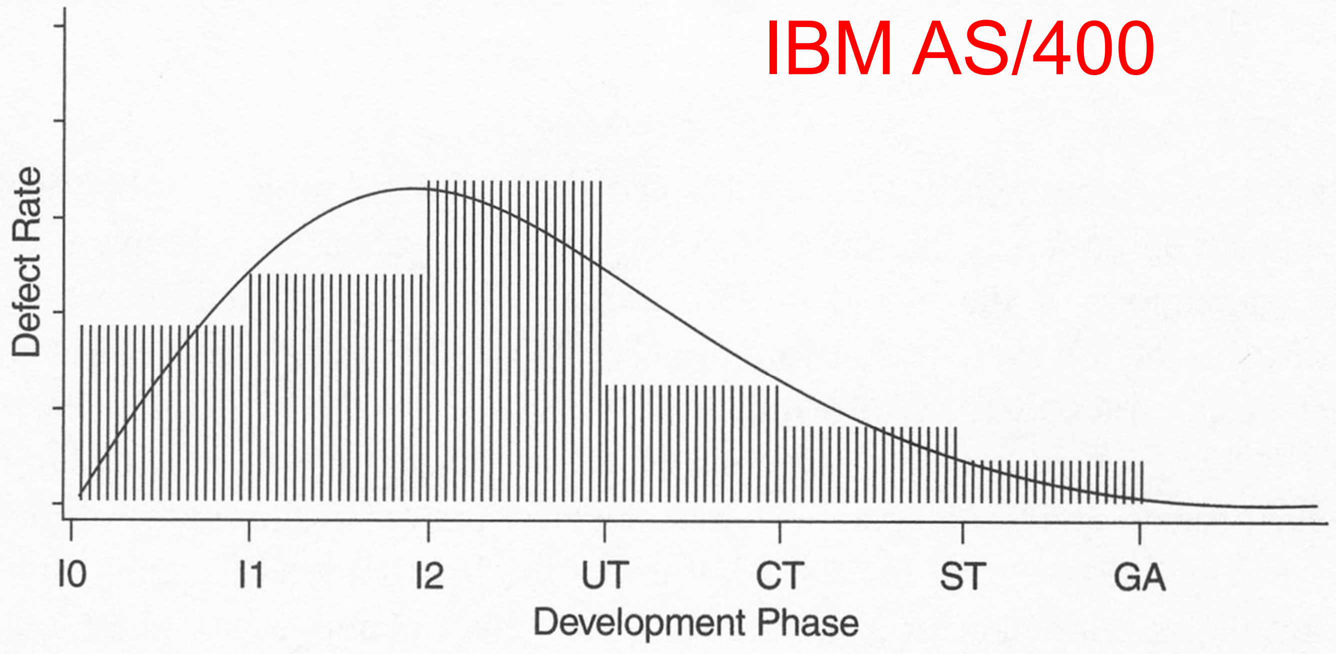
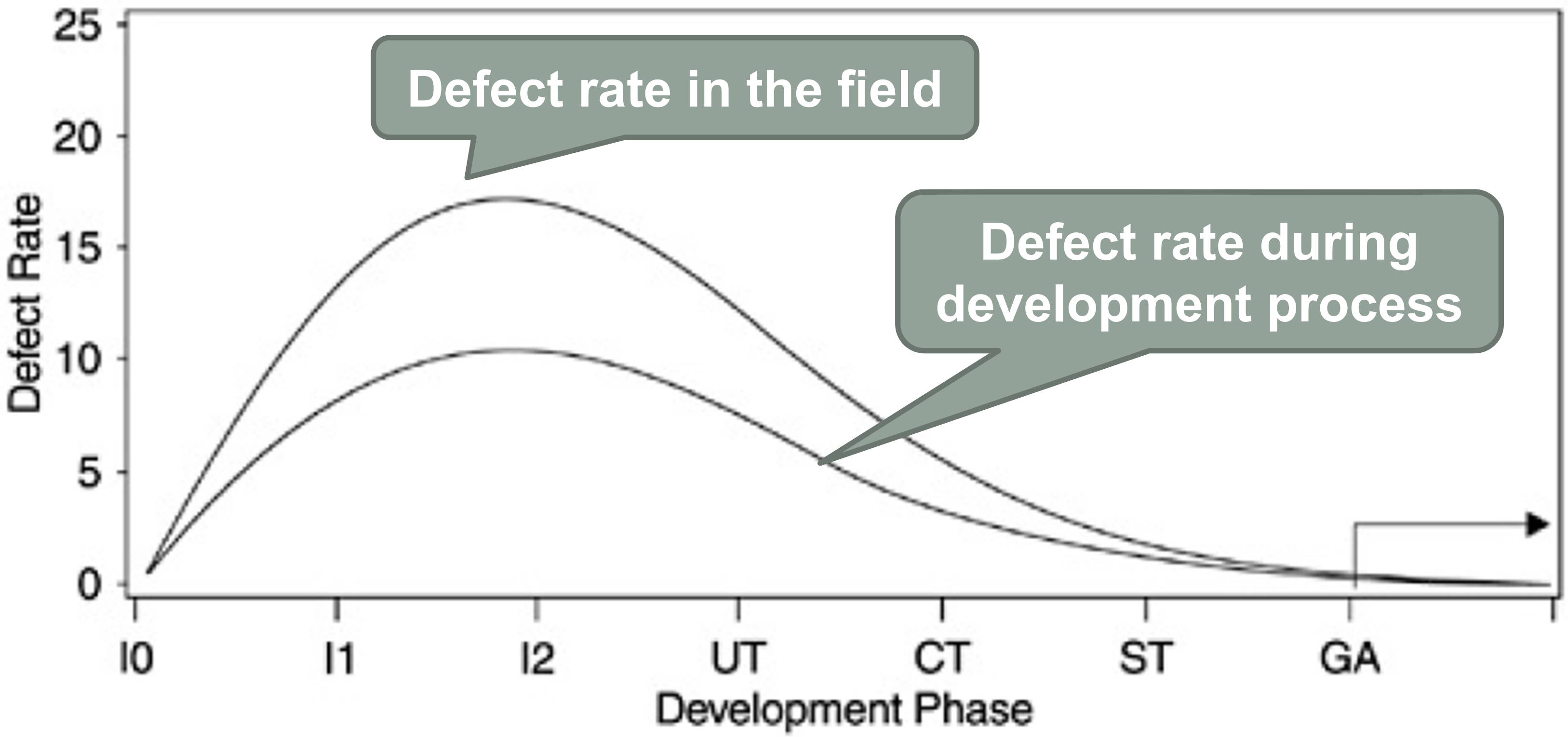


FIGURE 7.2
Rayleigh Model

Applying Rayleigh Model

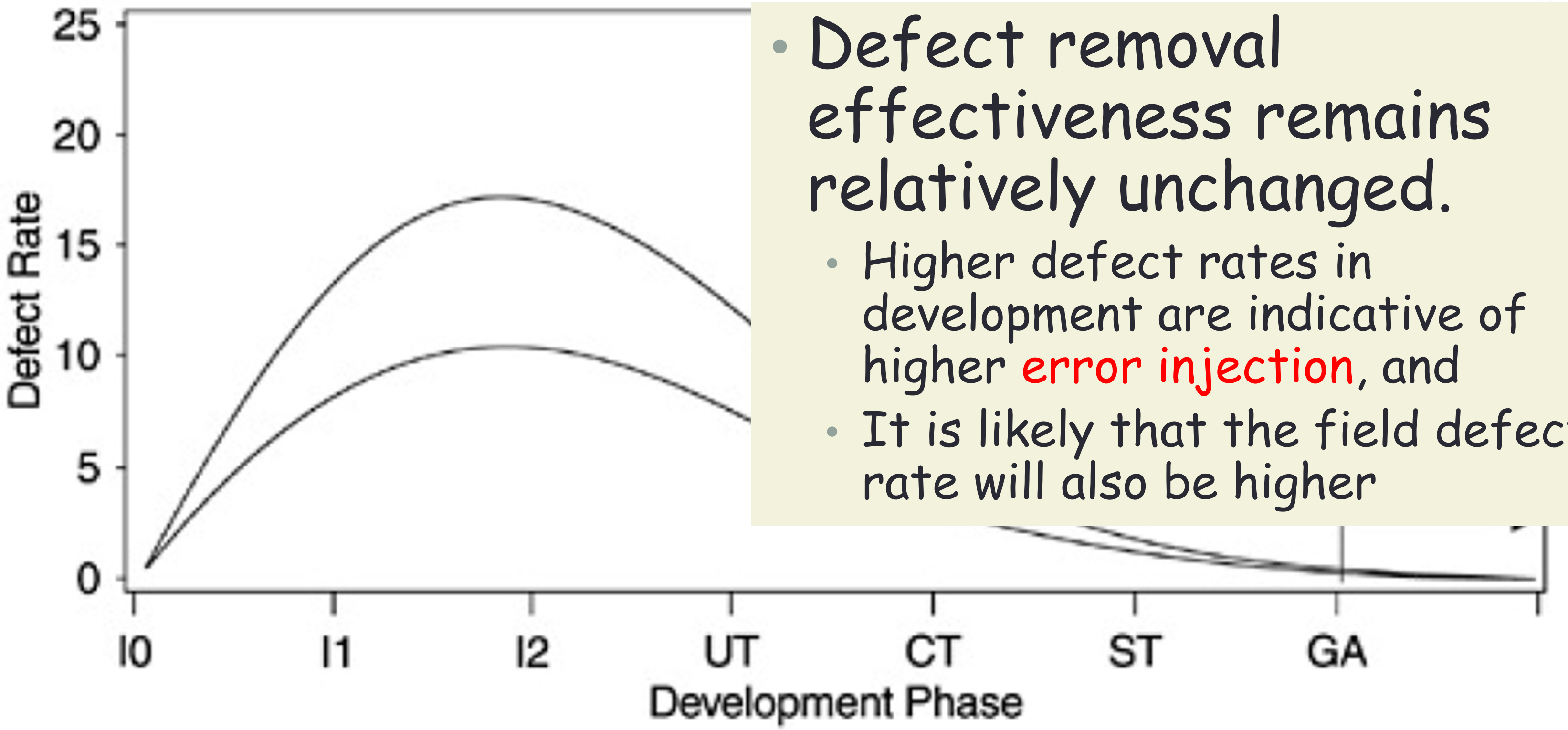
- Assumptions

- Defect rate during development positively correlated with Defect rate in the field



Applying Rayleigh Model

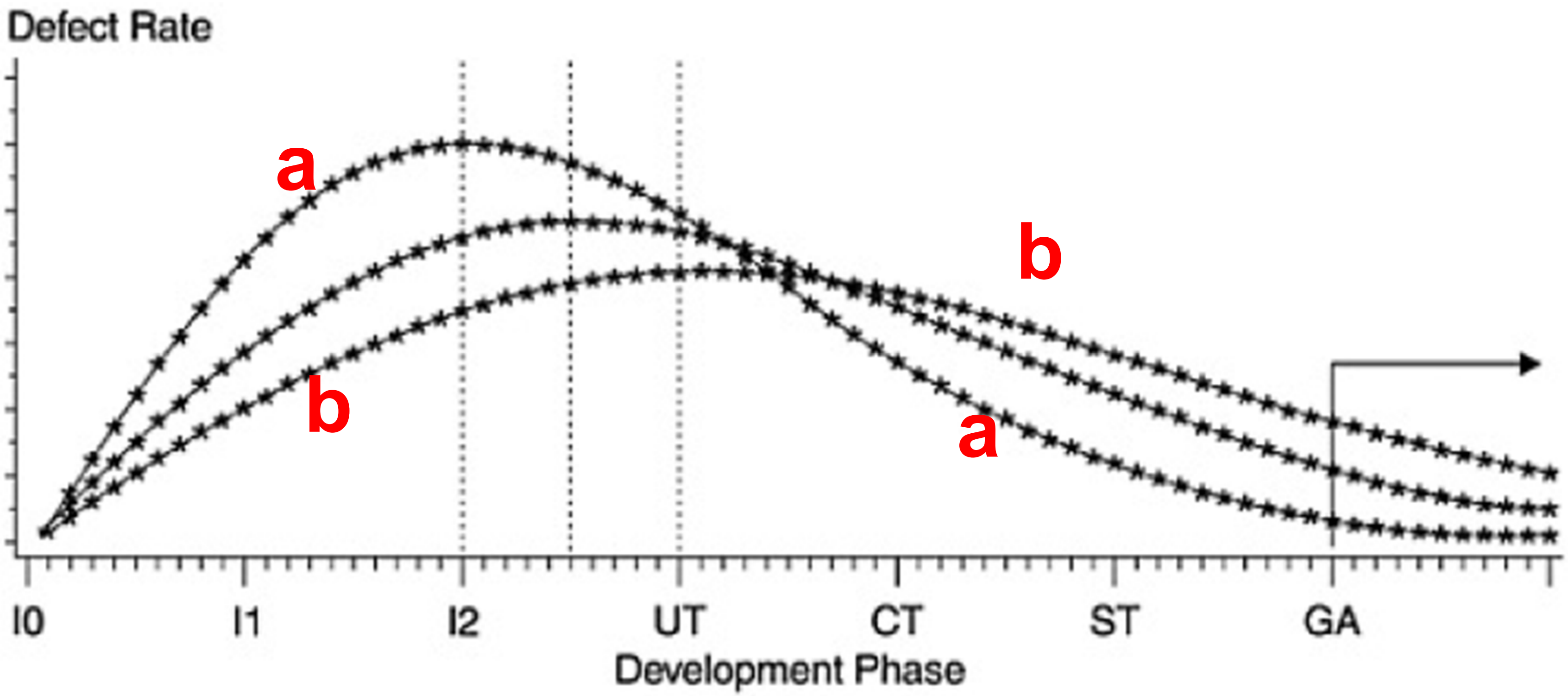
- Assumptions
  - Defect rate during development positively correlated with Defect rate in the field



- Defect removal effectiveness remains relatively unchanged.
  - Higher defect rates in development are indicative of higher error injection, and
  - It is likely that the field defect rate will also be higher

Applying Rayleigh Model

- Assumptions
  - Given the same error injection rate, if more defects are discovered and removed earlier, fewer will remain in later stages



Applying Rayleigh Model

- IBM AS/400

- Defect data
  - for 65 components
  - for different stages of each component
    - I0, I1, I2, UT, CT, ST, GA
- Correlation analysis
  - Spearman Rank Order Correlation
  - Between development-time defects and field defects

Applying Rayleigh Model

- IBM AS/400

TABLE 7.1
Spearman Rank Order Correlations

Phase	Rank-Order Correlation	*n*	Significance Level
I0	.11	65	Not significant
I1	.01	65	Not significant
I2	.28	65	.02
CT	.48	65	.0001
ST	.49	65	.0001
All (I0, I1, I2, CT, ST)	.31	65	.01

Strongly supported the first assumption.

Applying Rayleigh Model

- IBM AS/400

- Second assumption: also validated
- Significance even stronger at coarser levels

the more granular the unit of analysis, the less chance it will obtain statistical significance

Applying Rayleigh Model

1. Collect defect data (#defects / defect rate)
2. Derive model parameters from the data
   - E.g., the scale parameter
   - Using a statistical software package (e.g., SAS)
3. Estimate end-product reliability by plugging data values into model

Applying Rayleigh Model

- Example implementation: STEER @IBM

TABLE 7.2
Defect Removal Patterns and STEER Projections

			Defects Per KLOC								
Project	LOC	Language	High-Level Design	Low-Level Design	Code	Unit Test	Integration Test	System Test	First-Year Field Defect	LOP Field Defect	STEER Estimate
A	680K	Jovial	4	—	13	5	4	2	0.3	0.6	0.6
B	30K	PL/1	2	7	14	9	7	—	3.0	6.0	6.0
C	70K	BAL	6	25	6	3	2	0.5	0.2	0.4	0.3
D	1700K	Jovial	4	10	15	4	3	3	0.4	0.8	0.9
E	290K	ADA	4	8	13	—	8	0.1	0.3	0.6	0.7
F	70K	—	1	2	4	6	5	0.9	1.1	2.2	2.1
G	540K	ADA	2	5	12	12	4	1.8	0.6	1.2	1.1
H	700K	ADA	6	7	14	3	1	0.4	0.2	0.4	0.4

Validity

- Depends heavily on "data quality"
  - Back-end data tends to have better quality than front end data
- Model estimates and actual outcomes must be compared and empirical validity must be established.
  - Empirical validity is essential, and *context-specific*

Summary

- Reliability model
  - As a particular quality model
  - Static vs Dynamic
  - Two kinds of Dynamic model
- Rayleigh model
  - Weibull distribution, m=2
  - Derive the model from past stage defect data
  - Apply the model to projects for reliability estimation
  - Validate the predictive accuracy of the model