
An Introduction to the Quantitative, Rational and Scientific Process of Software Development (Part 2)

Zenya Koono (Creation Project),
Hui Chen (Kokushikan University) and
Hassan Abolhassani (Shariff University of Technology)
SOMET 07 (Rome) in 7 November, 2007.

Outline of Part 2

Object of Part 2: Use of “Process”

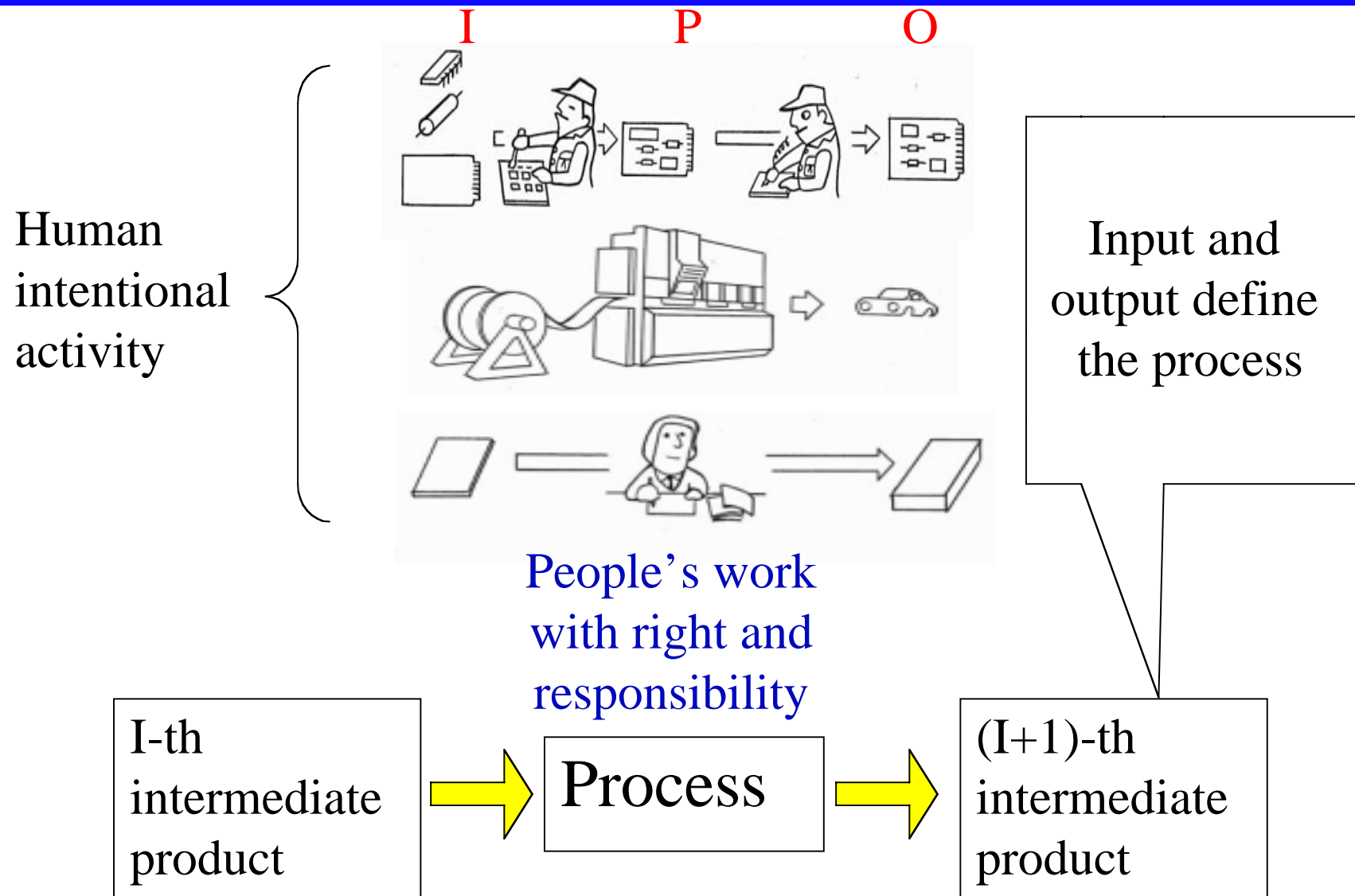
1. What is “process”

2. Quantitative Characteristics

3. Management Issues

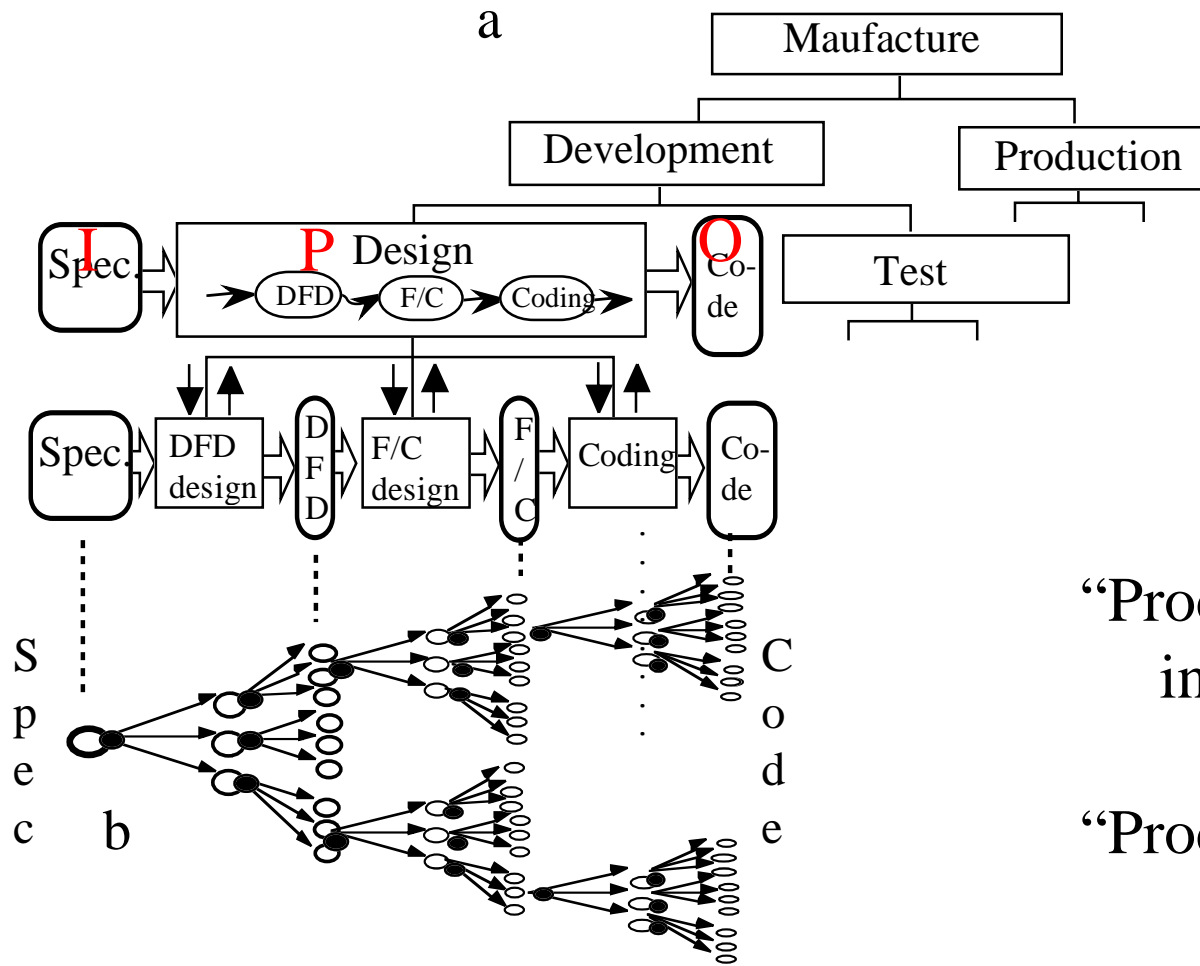
1. What is “Process”

What is “Process”



Structure of “Process”

Hierarchical “Process” is orthogonal to Hierarchical “Product”



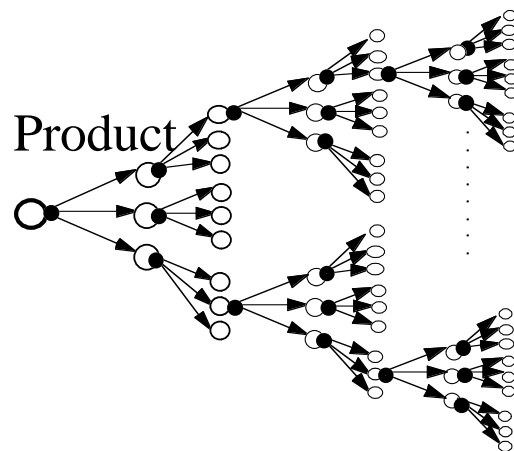
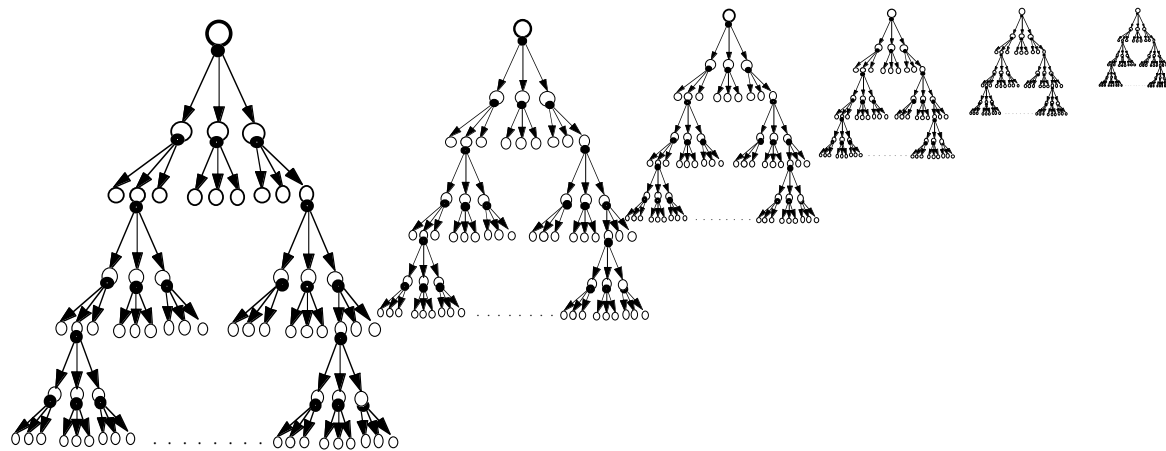
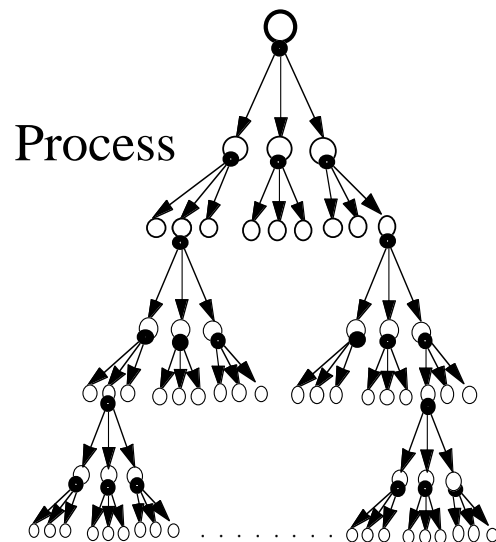
“Process” is another hierarchical decomposition of object.
It is another human intentional activity.

“Process” is commonly used, independent of “Product”.

“Process” is control technique.

Product’s “hierarchical objective”

Hierarchical Knowledge Tower

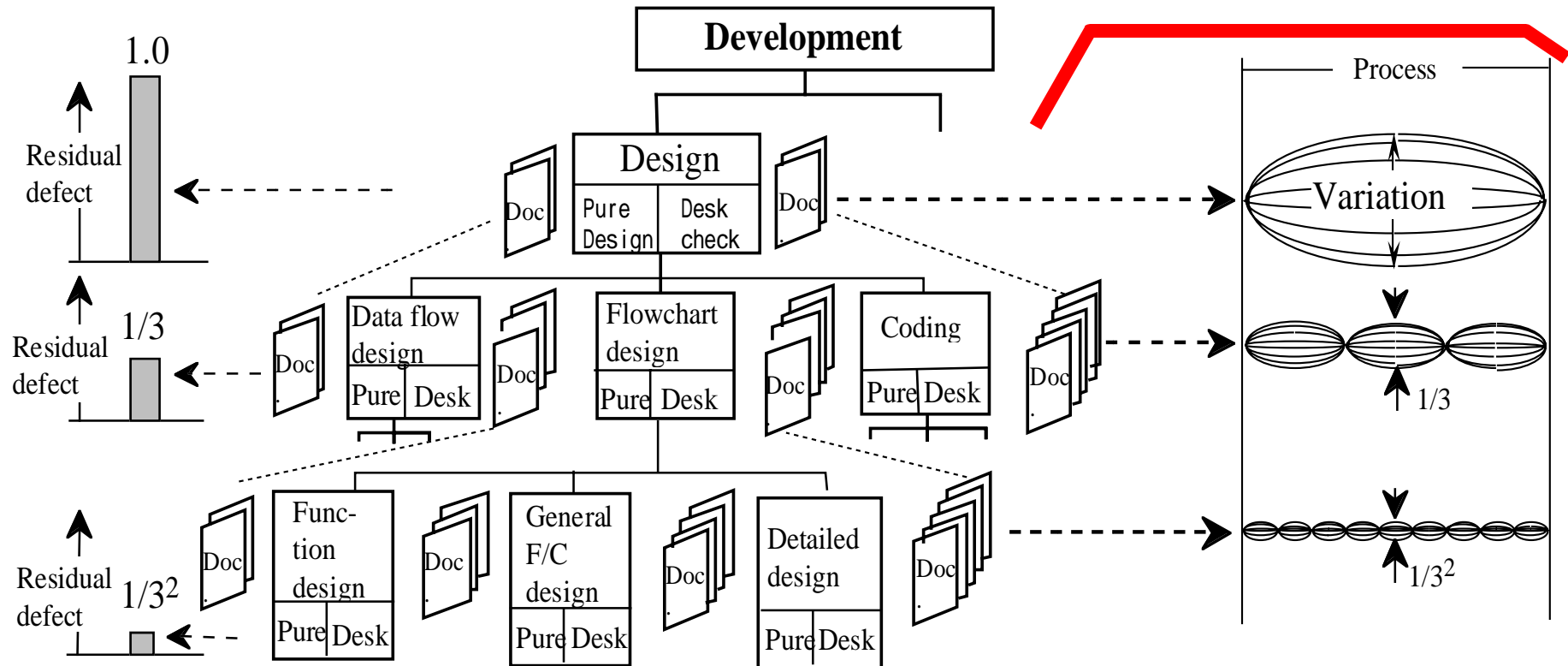


- “Process” is also knowledge.
- Human society may be regarded as knowledge society, consisting of many knowledge towers.

Minsky’s Hierarchical Agents

Process by “Divide and Conquer”

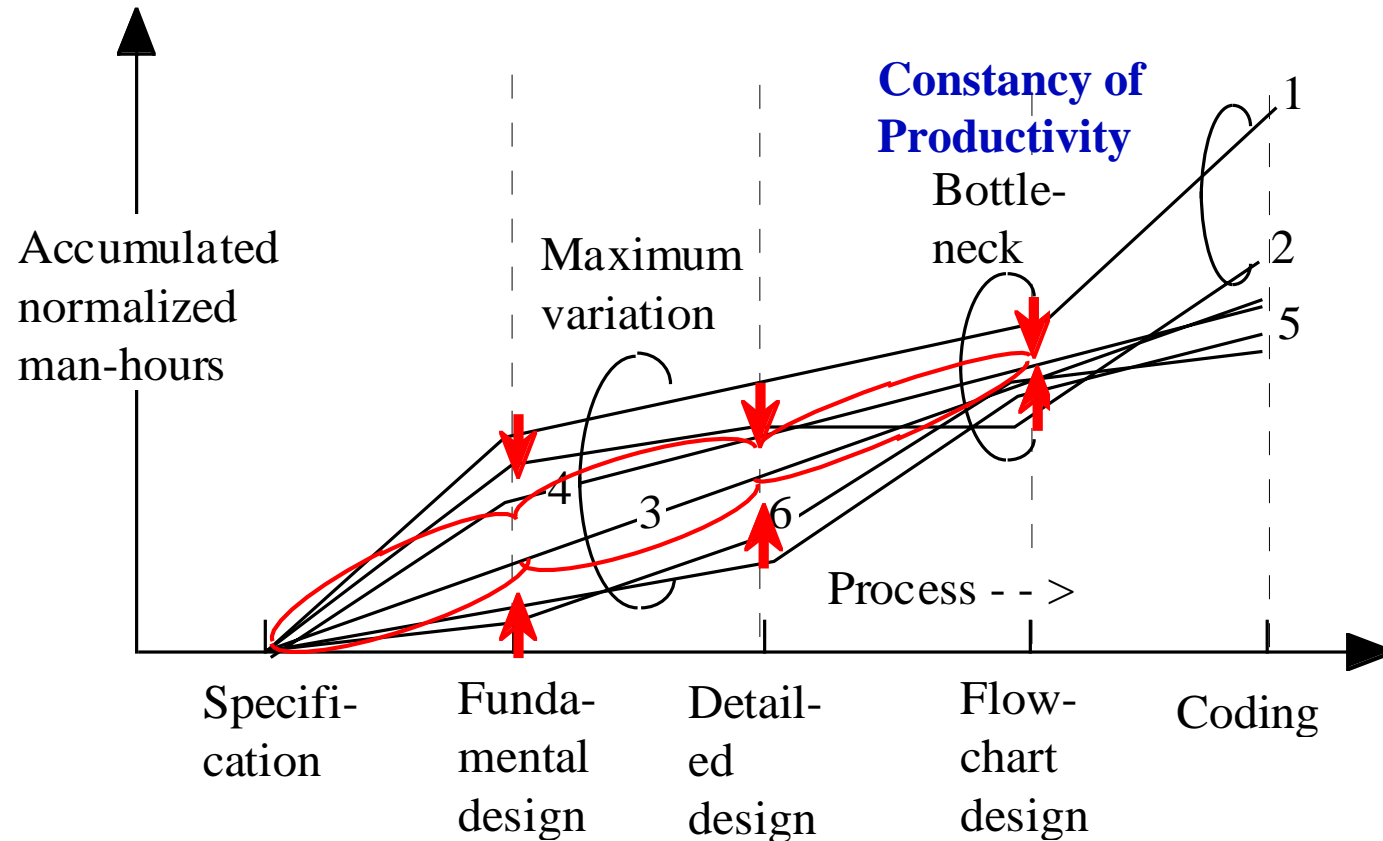
“Process” is a mean for managing development



When variation of process characteristics is large,
constrain some intermediate interfaces.

If not enough, do the same on poor processes.

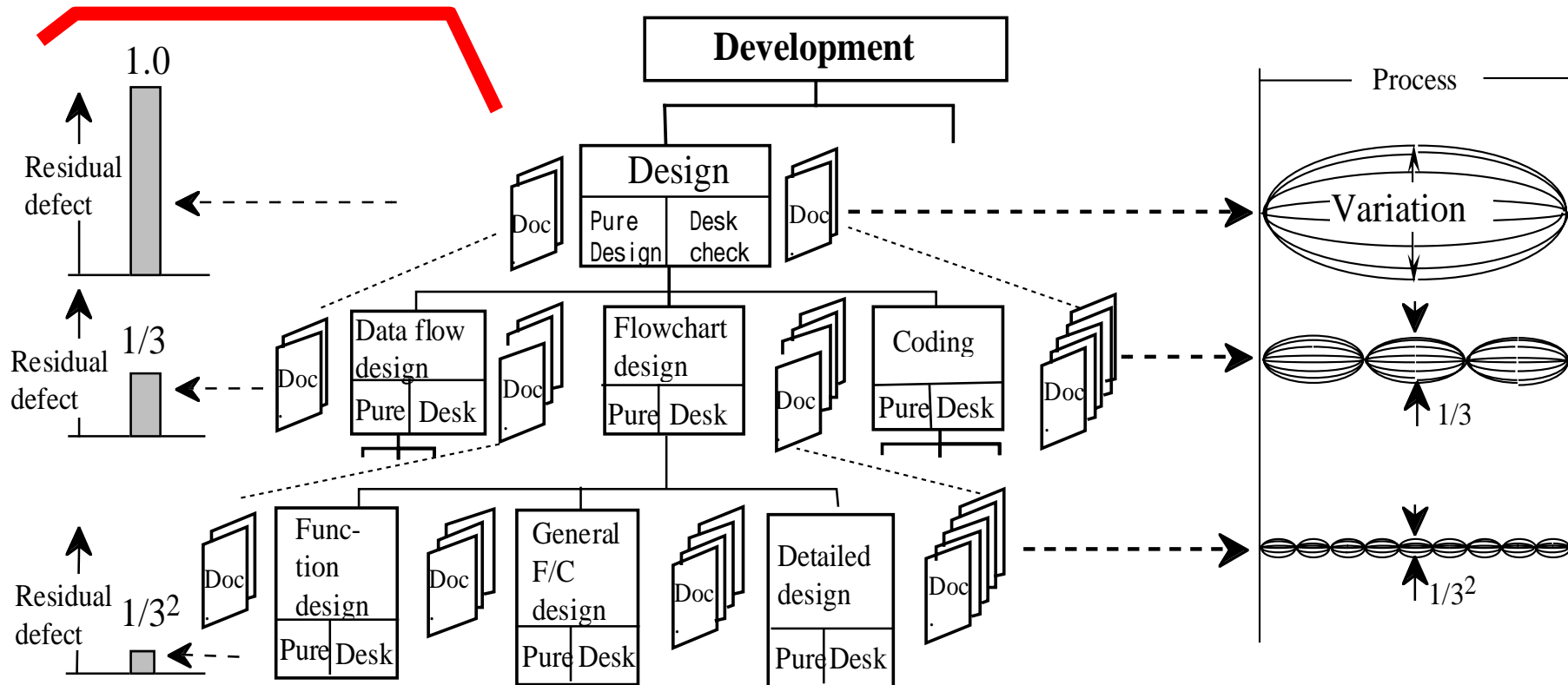
Field Experience



- Insufficient doc. specification/practice
- Enrich description with sample documents

Process by “Divide and Conquer”

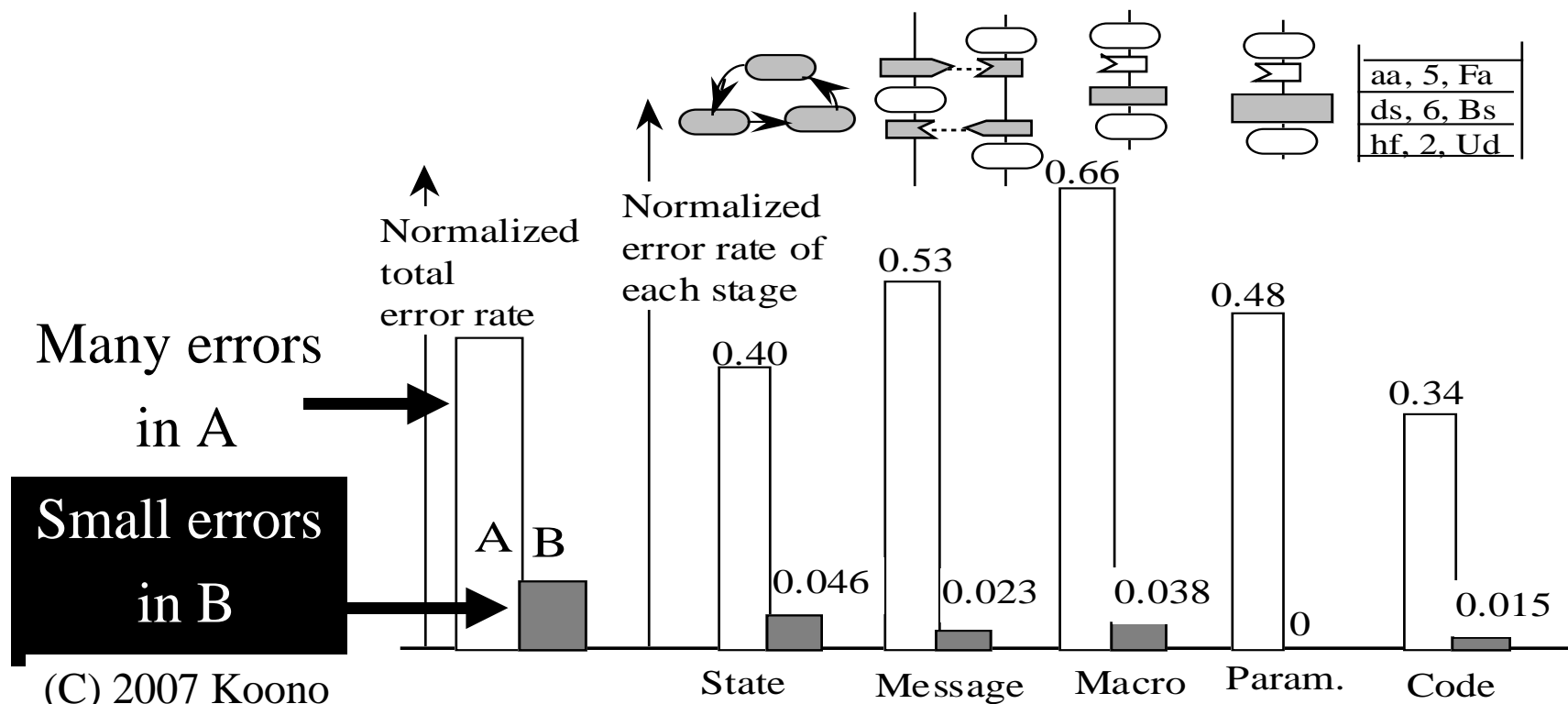
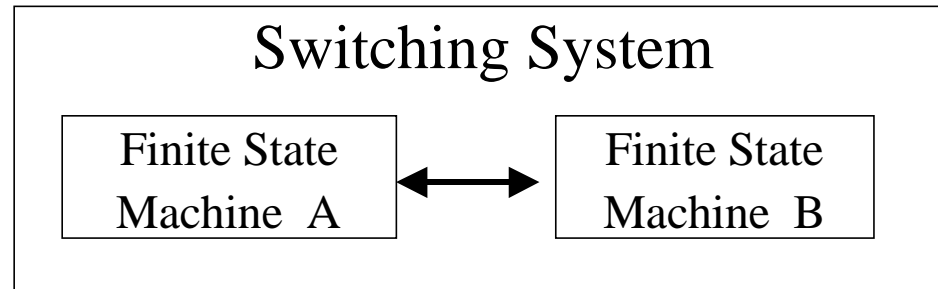
“Process” is a mean for managing development



- When errors are many, then divide the process
- When M-divided, the residual is decreased to $1/M$
- Divide until they are enough small

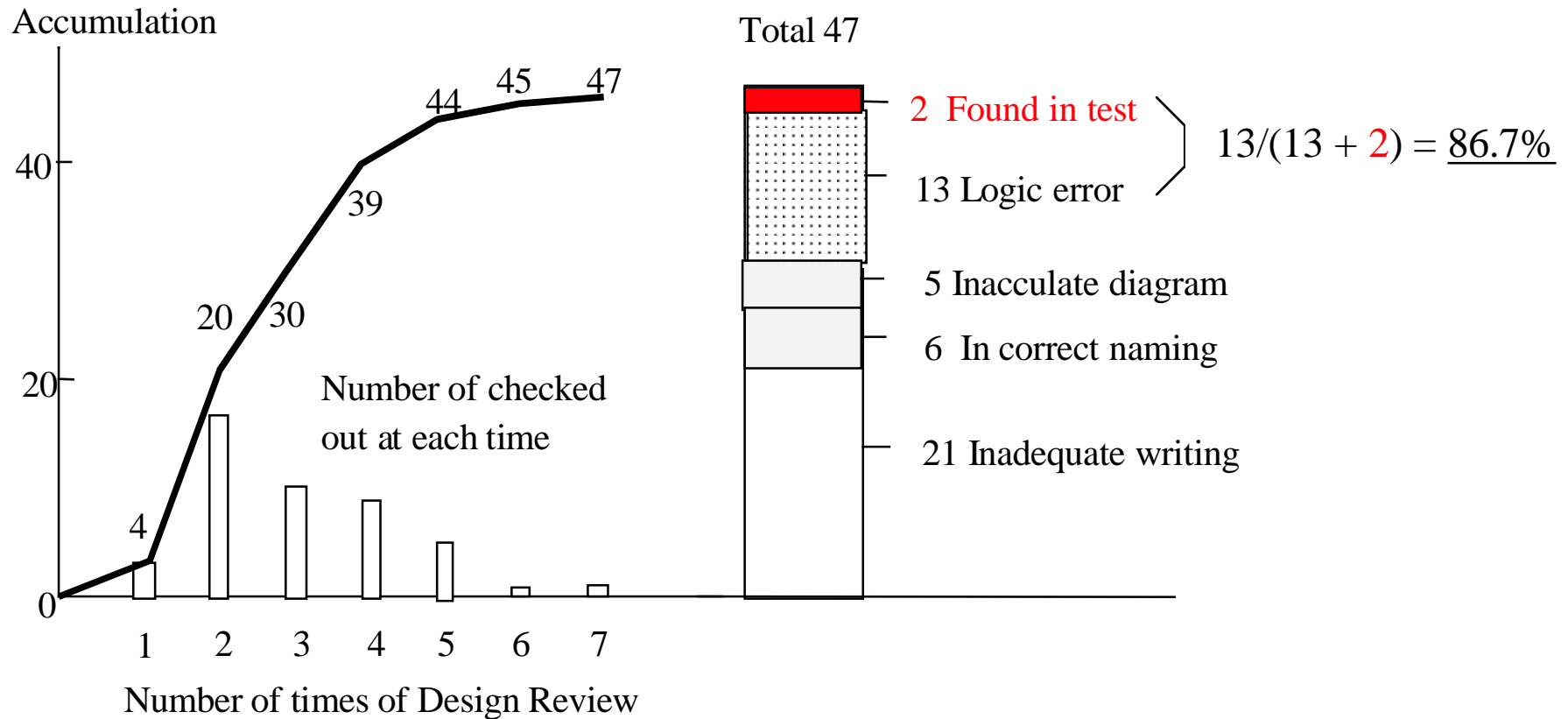
Difference of Two Projects

Desk check after design in small step of design progress is vital



An Example of Desk Check

Desk check after design in small step of design progress is vital



- A program for pasting a child DFD to the parent DFD
- Hierarchical data flows form HIPO 147 C code lines
- Error rate 14 E/kL (Built-in 102 E/kL, student average)

(C) 2007 Koono

2. Quantitative Characteristics

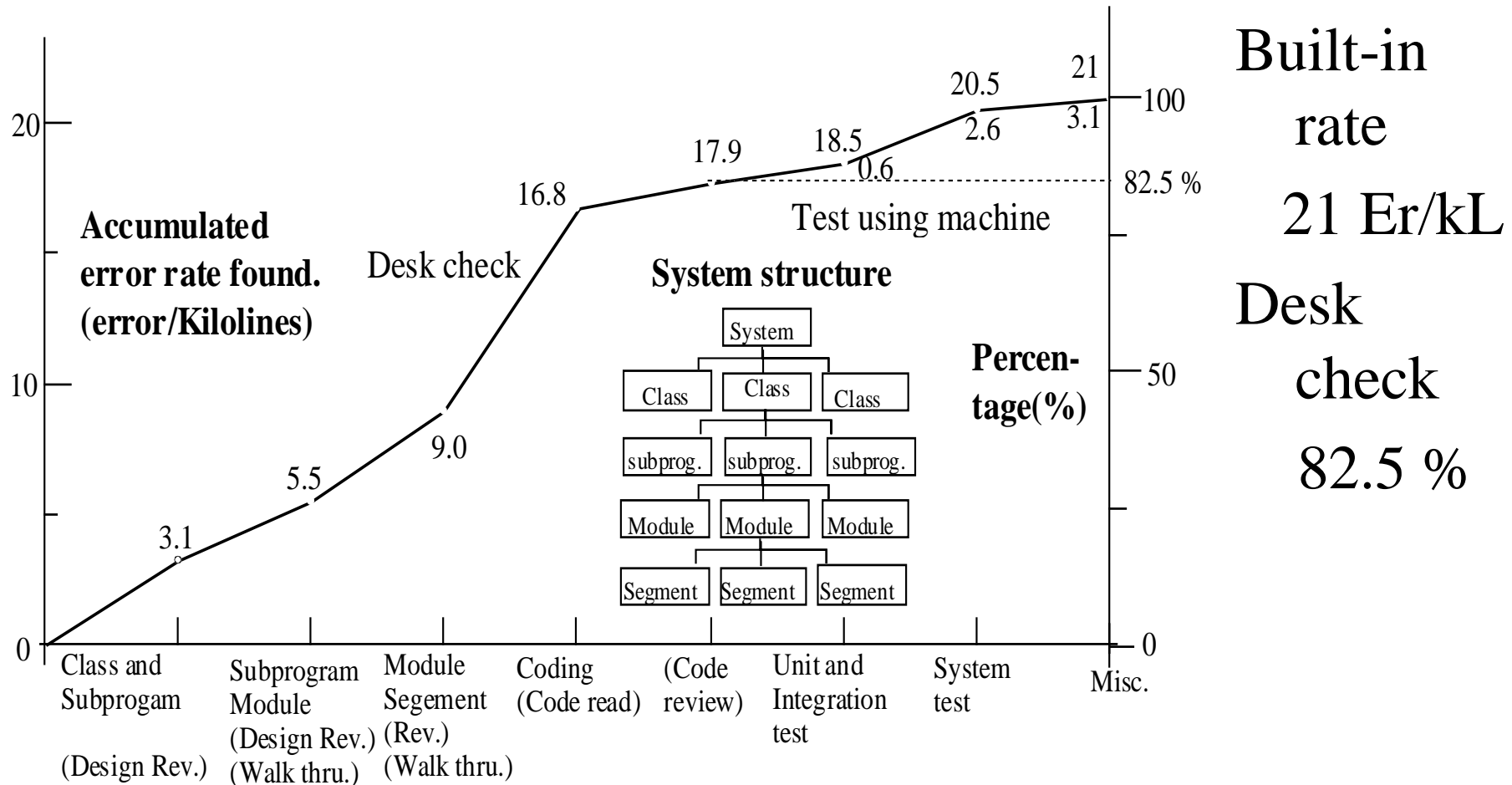
2. 1 An Excellent Development in GTE

Excellent paper in Intl Switching Symposium 1979

Visited and discussed in the next ISS 81

Based on the paper and GTE internal documents,
authors reproduced their characteristics
at their responsibility

GTE's Excellent Development

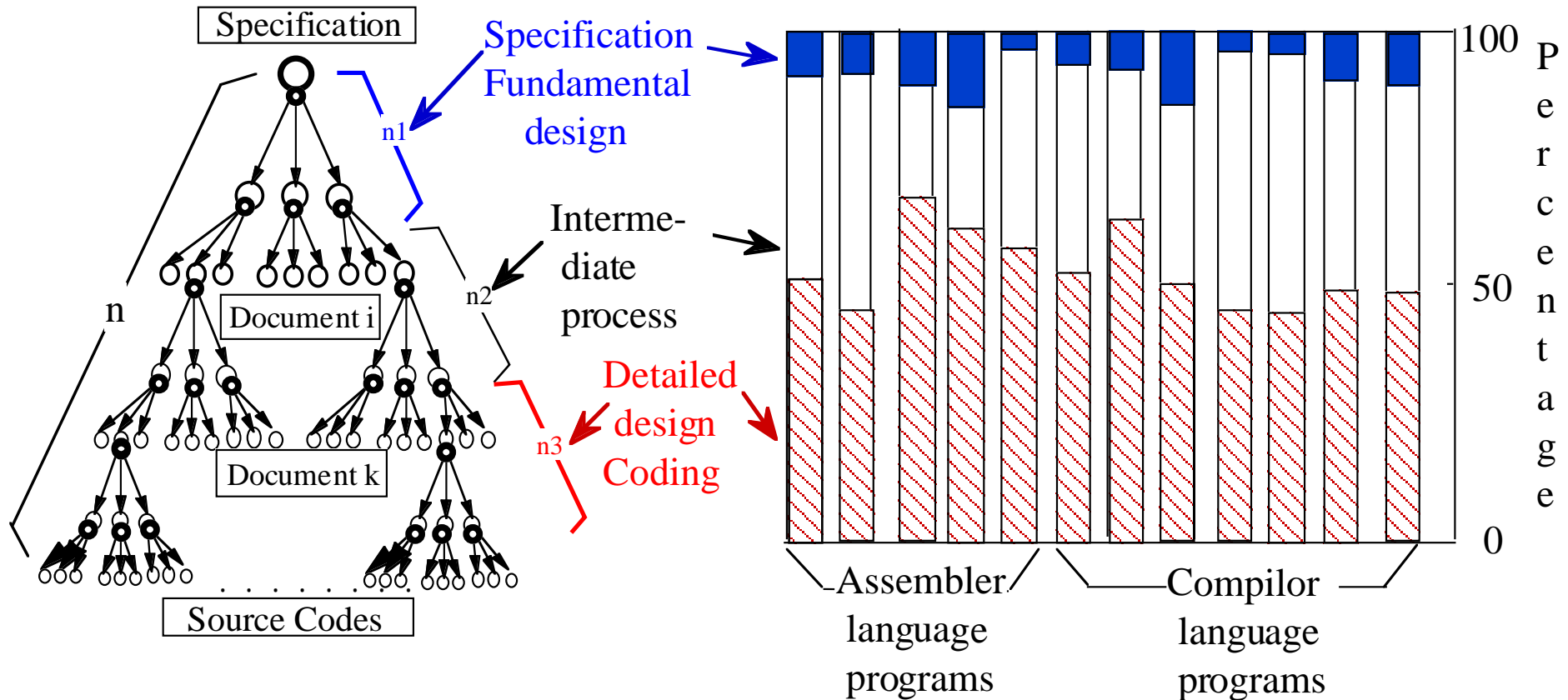


Excellent leaders: Exec. B.S.E.E. + MBA, management s/w.

Mgr.: Rational, quantitative & scientific. experience based.

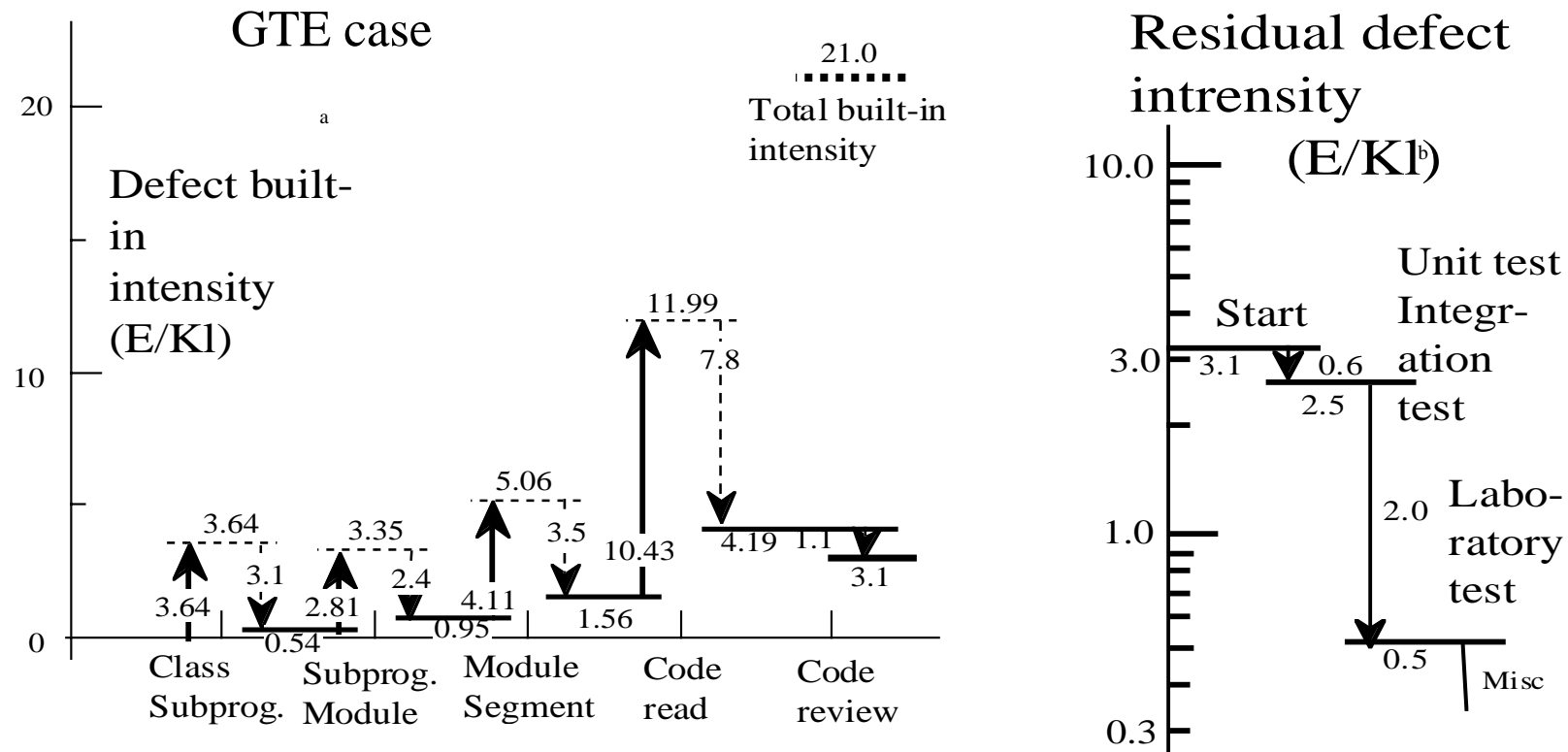
Young people: Eagerly watching new technologies (C) 2007 Koono

Linear Nature of Process



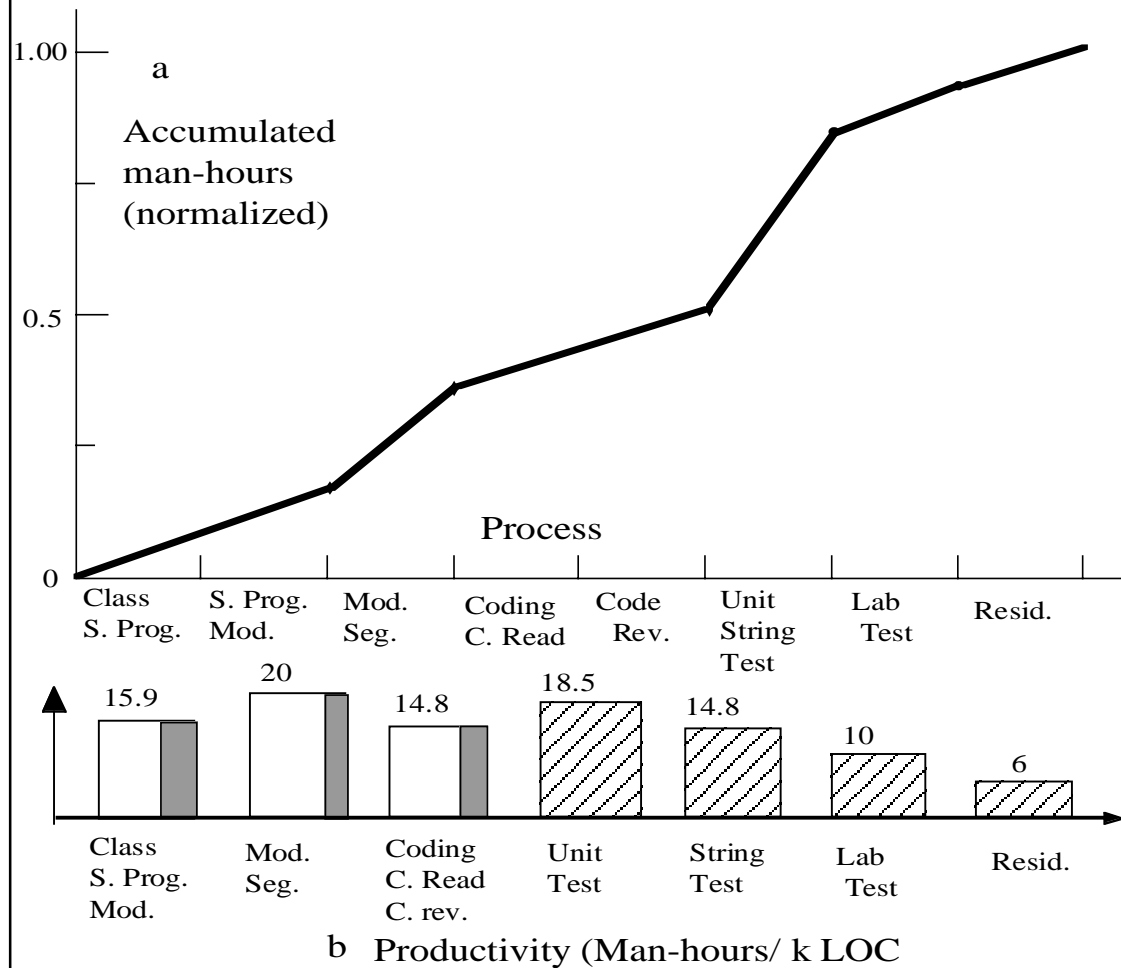
- (Desk checked + test detected)/(Total of built-in)
- Similar trend in man-hour data

Error/Defect Level Diagram



- Built-in and check-out during design
- Attenuation by each test

Man-hour Data



Pure design
 Desk check
 Test

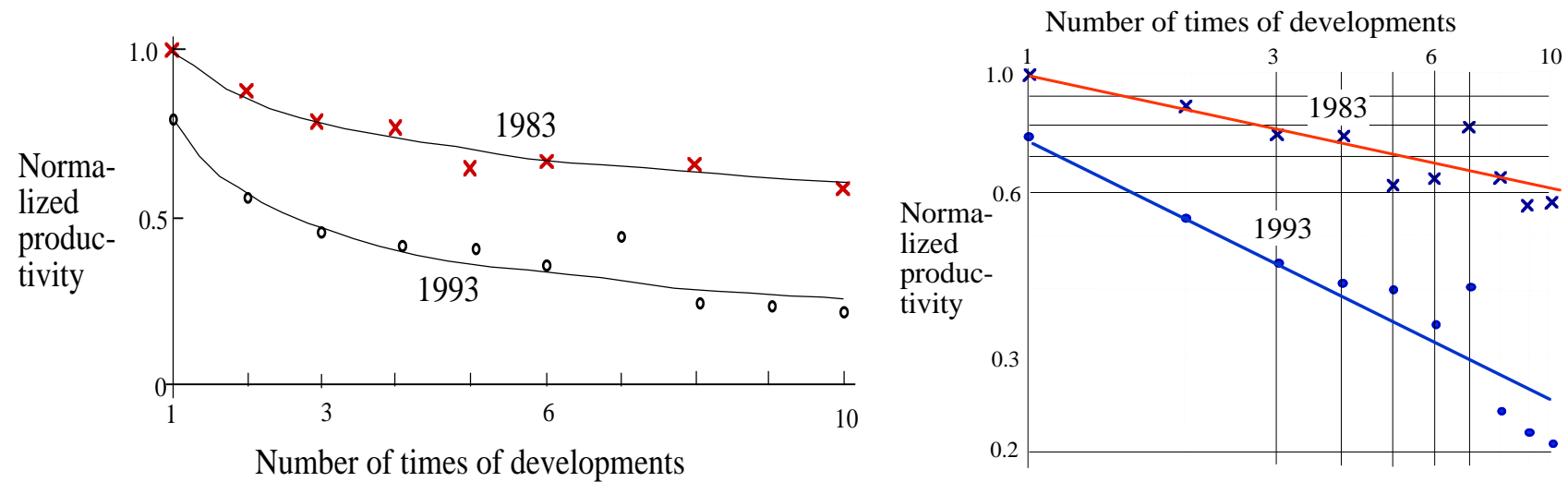
- Total man-hour or no. of output (mod./sheet/line)
- Budget curve based on past data and actual data
- Overall data for project leader group data for team leader a process data for each designer

2. Quantitative Characteristics

2. 2 Learning Effect

Accumulation of Knowledge

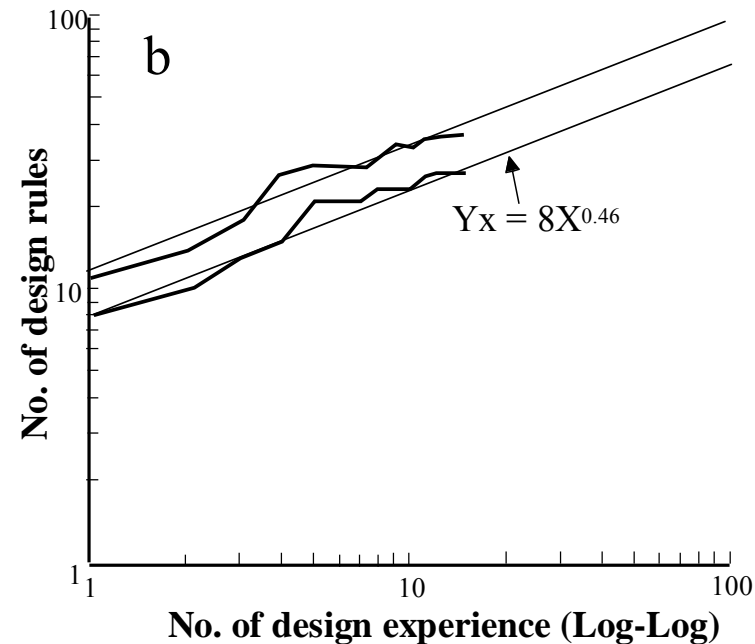
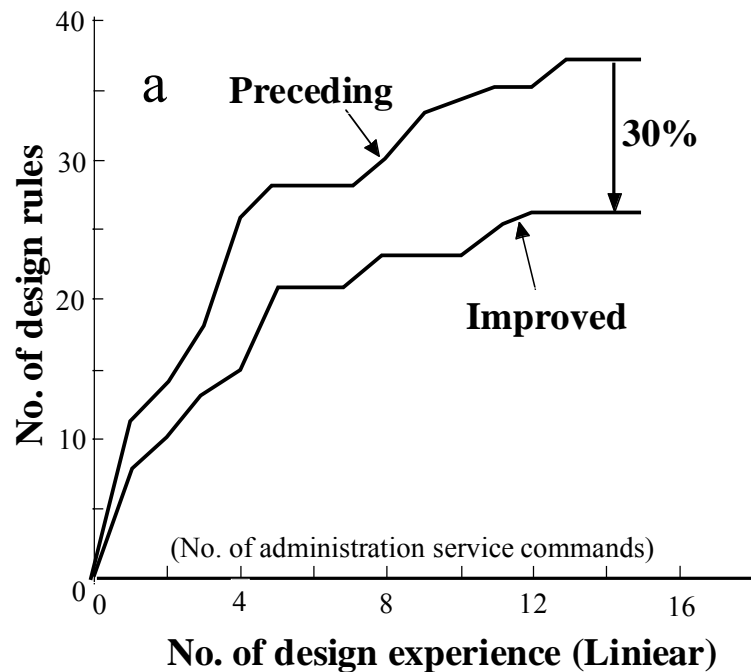
Learning Effect



- When a person begins a sport or a game, the person show a rapid growth at first, then, the gradient gradually decreases, but the similar growth continues.
- When plotted on both-logarithmic chart, plots show a linear trend line. It is called a Logarithmic Learning Effect. (First official report 1936 from Boeing)
- Most human characteristics show Logarithmic Learning Effect. Used in many predictions.

Logarithmic Accumulation

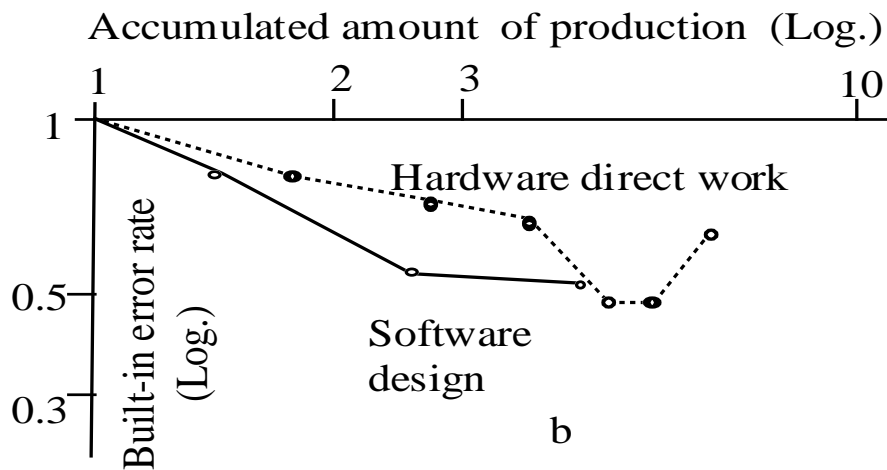
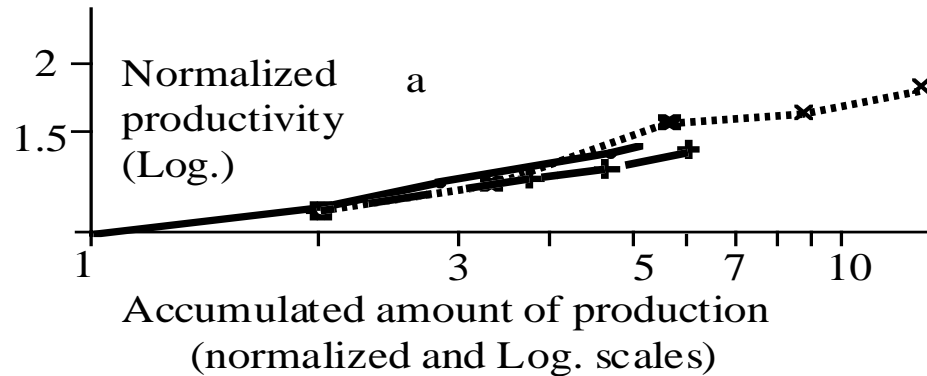
The improvement is achieved by Knowledge



- First rapid progress then the gradient gradually decreases.
The logarithmic growth of knowledge supports the learning effect.
It arises from “Zipf’s law of least effort”
- Learning effect is clarified 70 years after the first report in 1936

Productivity and Error Rate

The improvement is achieved by Knowledge



Productivities of various software in 1970's - 1980's

Built-in error rates of H/W production and S/W development in 1970's - 1980's

Both results by efforts of each leaders and people (Typical TQM)

Management issues

3. Management Issues

3.1 Quality Assurance Organization

3.2 Both “Process” and “Product”

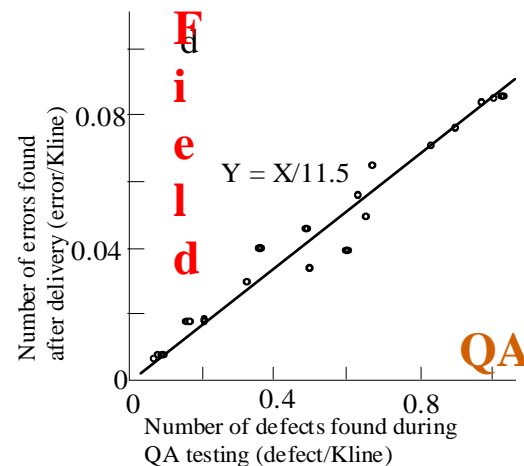
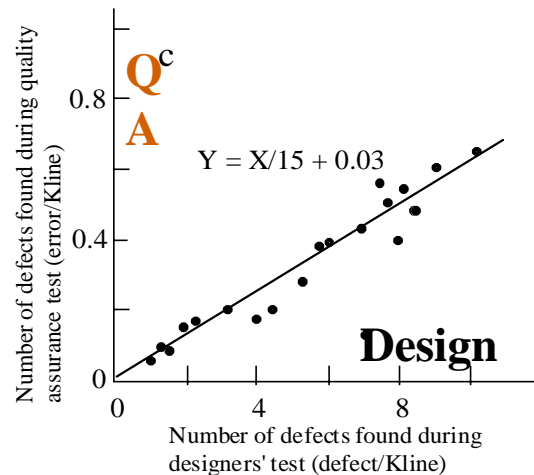
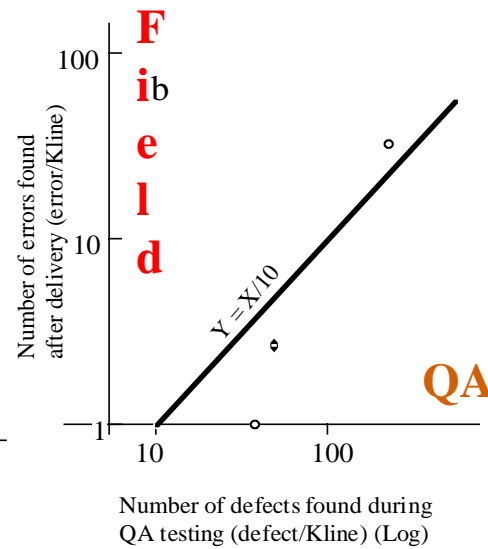
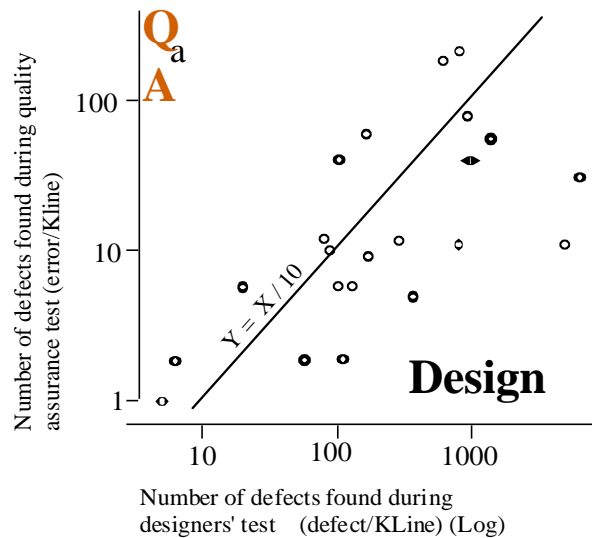
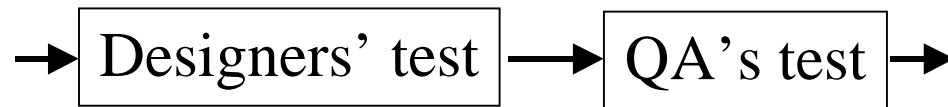
3.3 Repeat PDCA

3.4 Approach for Improvement

3.5 Not to Repeat The Same (Error) Again

3.1 Quality Assurance Organization

Quality Assurance Organization improves corporate quality



Koono

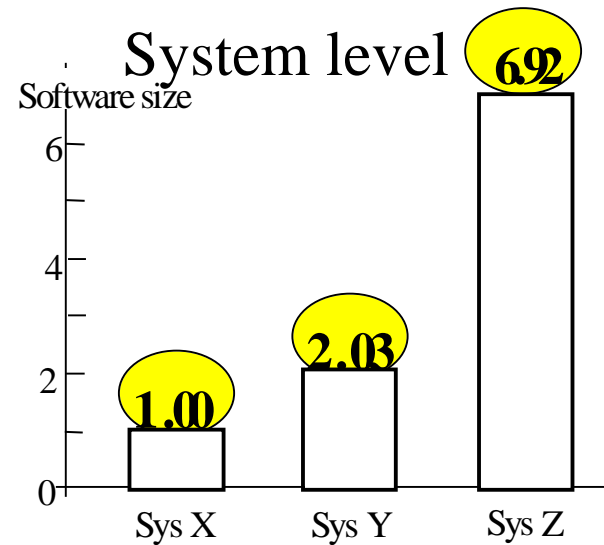
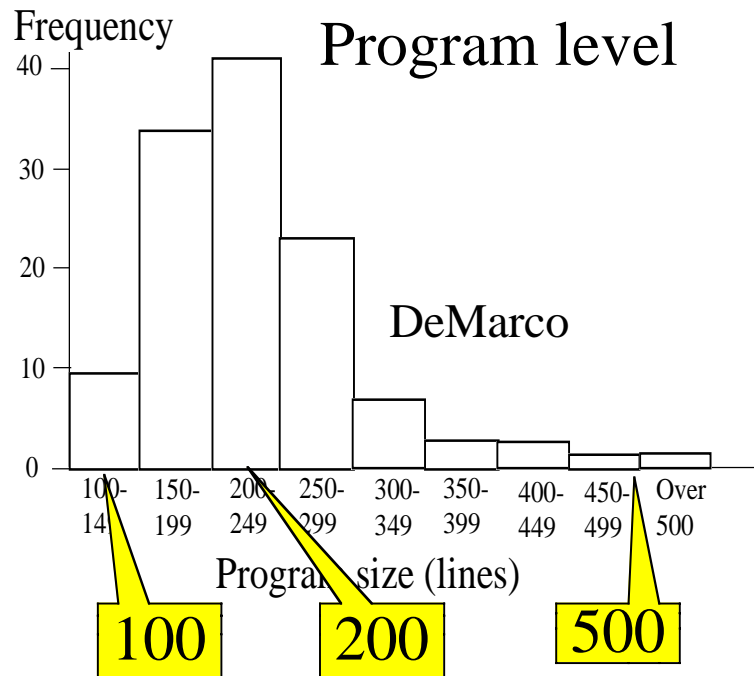
Dr. Watanabe

- Designers' test: White box
- QA's test: Black box test
- Equal level people
- Equal number of tests
- Around 1/10 attenuation = Second kind of error rate

(C) 2007 Koono

3.2 Both “Process” and “Product”

Both must be strictly controlled



Software size follows to lognormal distribution

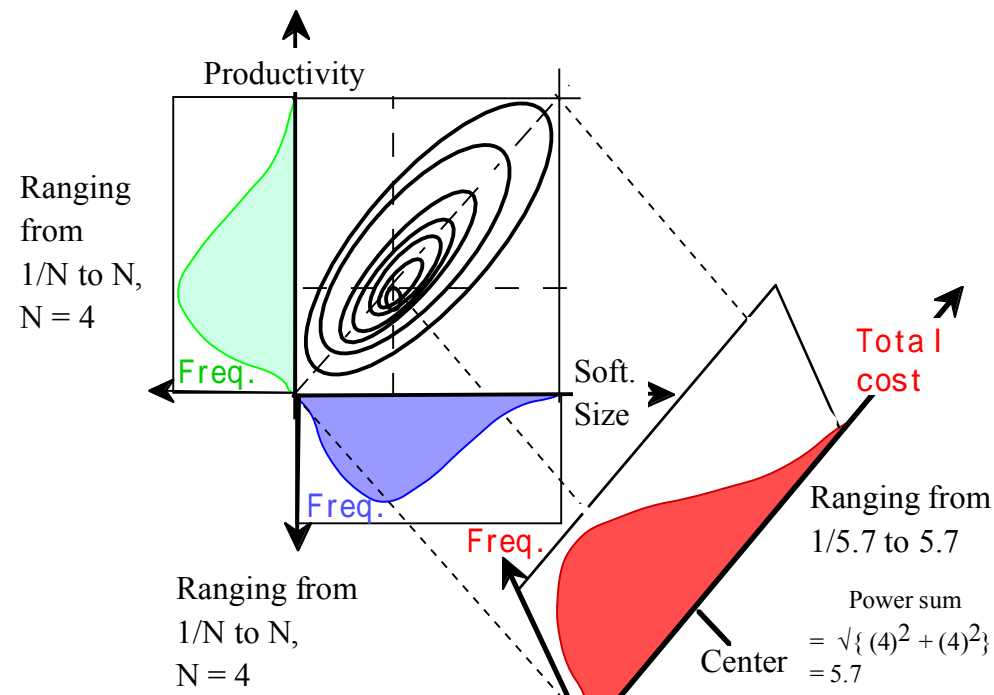
3.2 Both “Process” and “Product”

Both must be quantitatively evaluated and improved.

Development cost is proportional to

$$\frac{\text{(Software size)}}{\text{Function}} \times \frac{\text{(Man-hours)}}{\text{(Software size)}}$$

Quality of Design Productivity



(C) 2007 Koono

3.3 Repeat PDCA

Any human intentional activities may be managed in the same way

So-called “*specialty of software*”

- It is difficult to manage invisible software
- When it ended, the cost was twice the budget
- It is the delivery day today, still bugs spring up

Not specialty but “Lack of management”

Solution from preceding industrial experiences

1. Quantitative measure
2. “Management by Object” based on budget system

Example: Plan Do Check Action

3. Accumulate improvements

Plan, Do, Check and Action

• 1. PLAN Ex. Proven productivity x Proven man-month

• (1 + margin)
2. DO Execute the plan

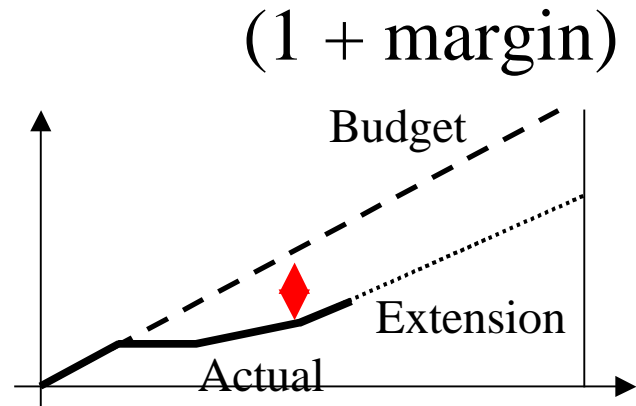
• 3. CHECK Quantitative measure

4. ACTION

5.

•

•



Overrun is not basically allowed

Hierarchical organization's rescue

While recovery is possible,

Recover as planned already

After project Analysis (Check and Action)

(C) 2007 Koono

After Project Analysis

Feedback through experiences strengthen an organization

Product analysis

<u>Term</u>	<u>Grade</u>	<u>Cause</u>
• Spec. 1	O	-----
• Spec. 2	X	-----
• Spec. 3	Δ	-----

Project analysis

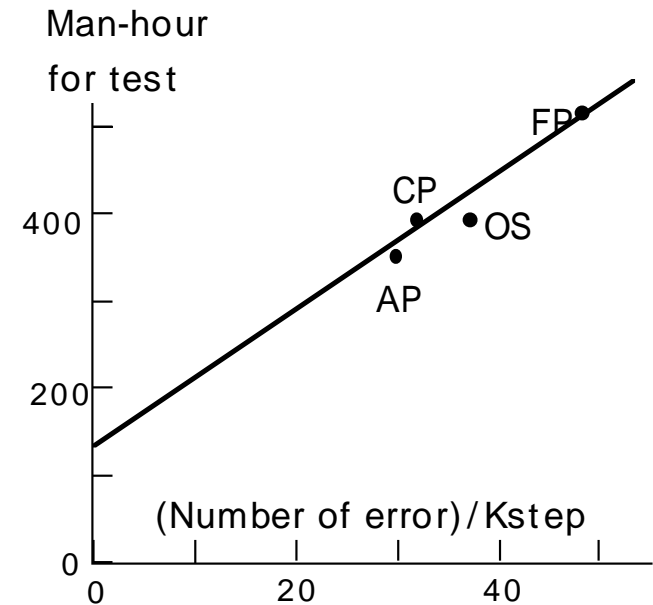
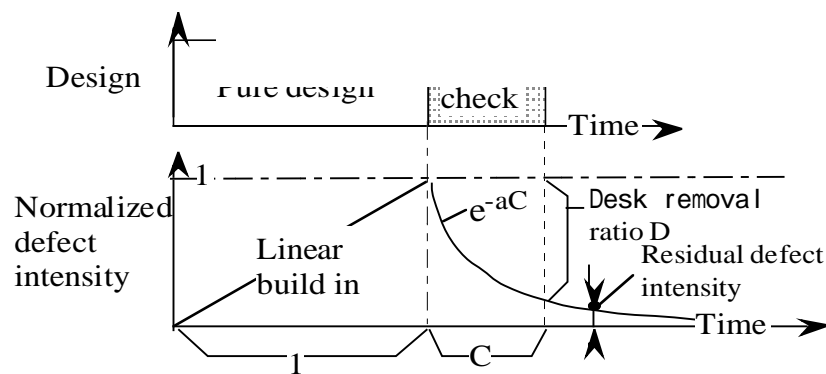
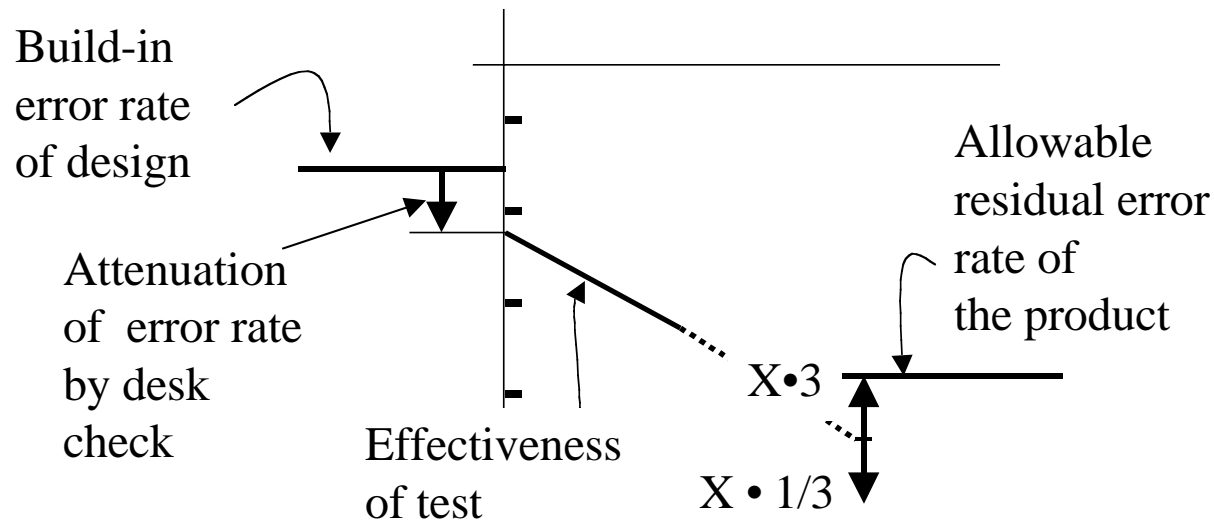
<u>No</u>	<u>Tbl</u>	<u>Grade</u>	<u>Cause</u>
1	adh	O	-----
2	fru	Δ	-----
3	mgp	X	-----

- A problem is a result of Cause and Effect network.
- From tracing back the network, primary / direct cause then root cause, in the network ,are known.
- A preventive mean not to repeat is to change primary / direct cause or root cause.
- The feedback loop works as a preventive mean.

Quantitative Project Planning

- Quantitative evaluations after project enable advances

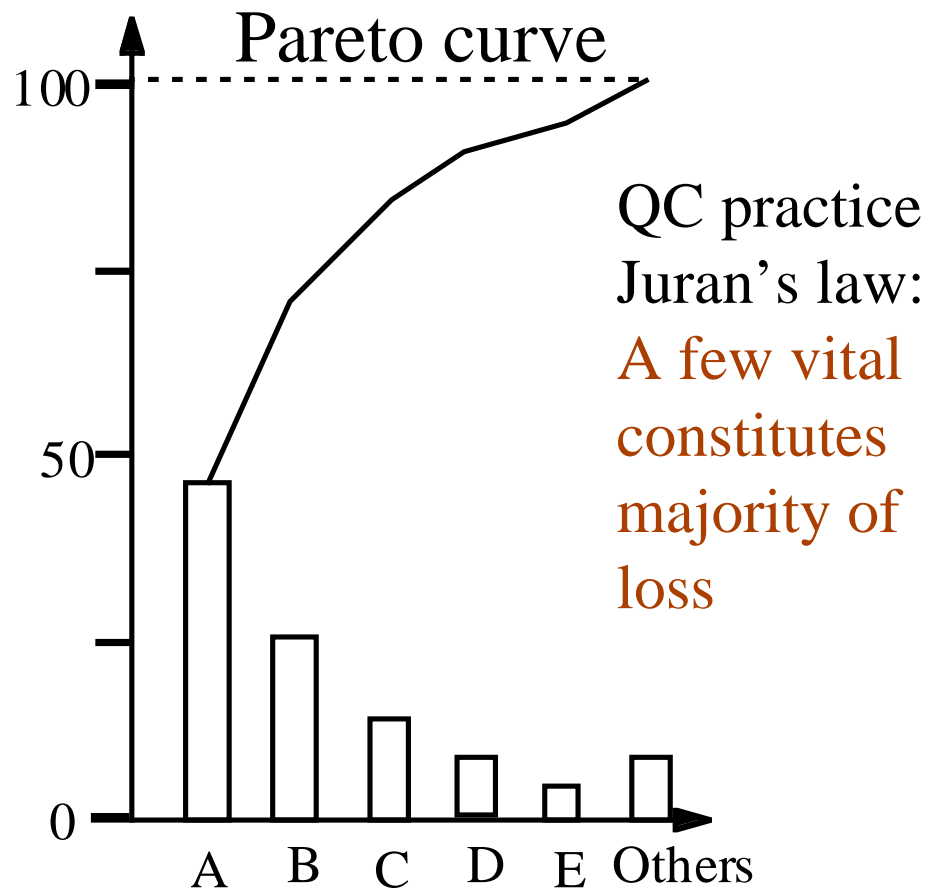
Test intensity



b. Medium quality case

Purge “a Few” Vitals

A good practice in other field is effective also in S/W.

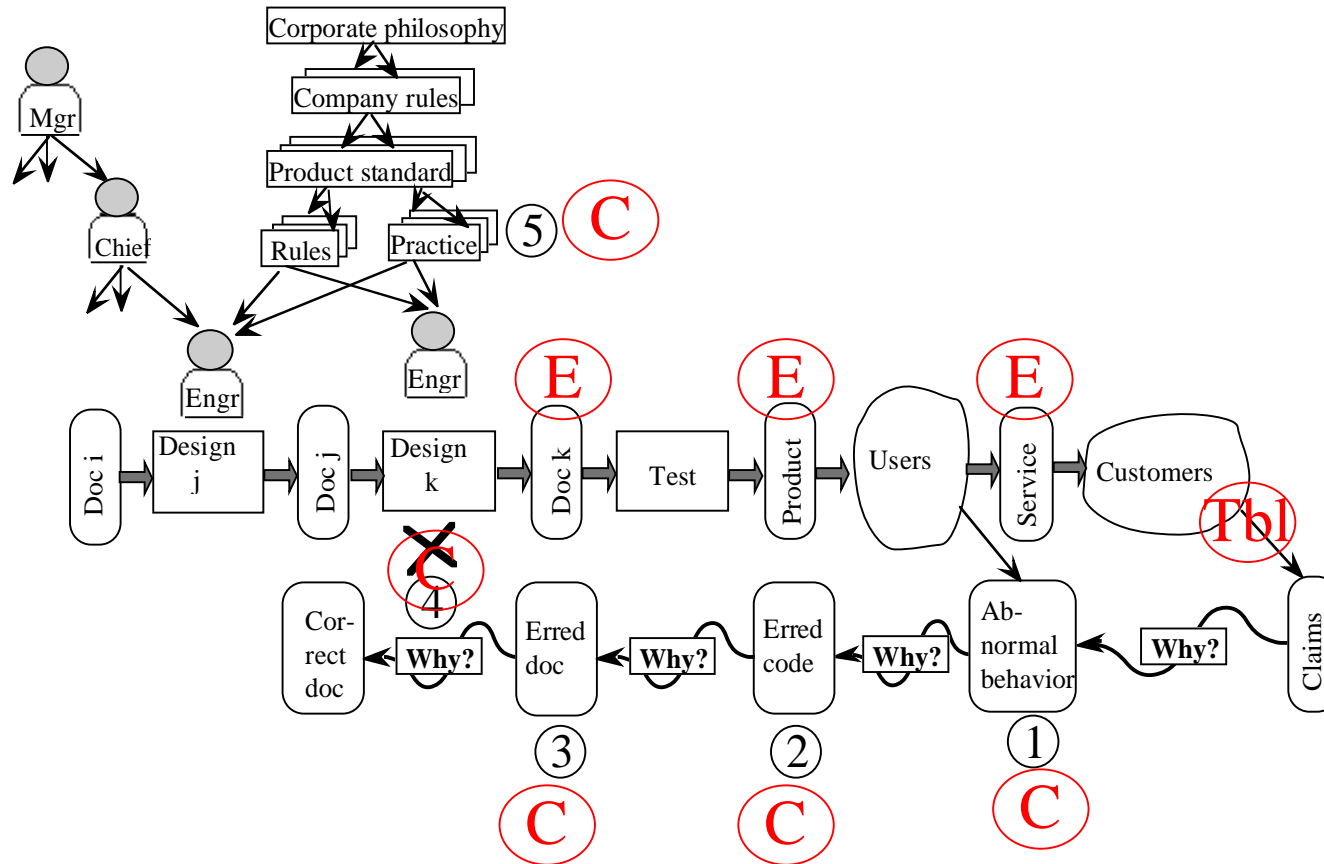


This practice may be applicable in all human related fields.

It may be proved by Zipf's law.

Not to Repeat The Same Again

Hunting on a cause and effect network



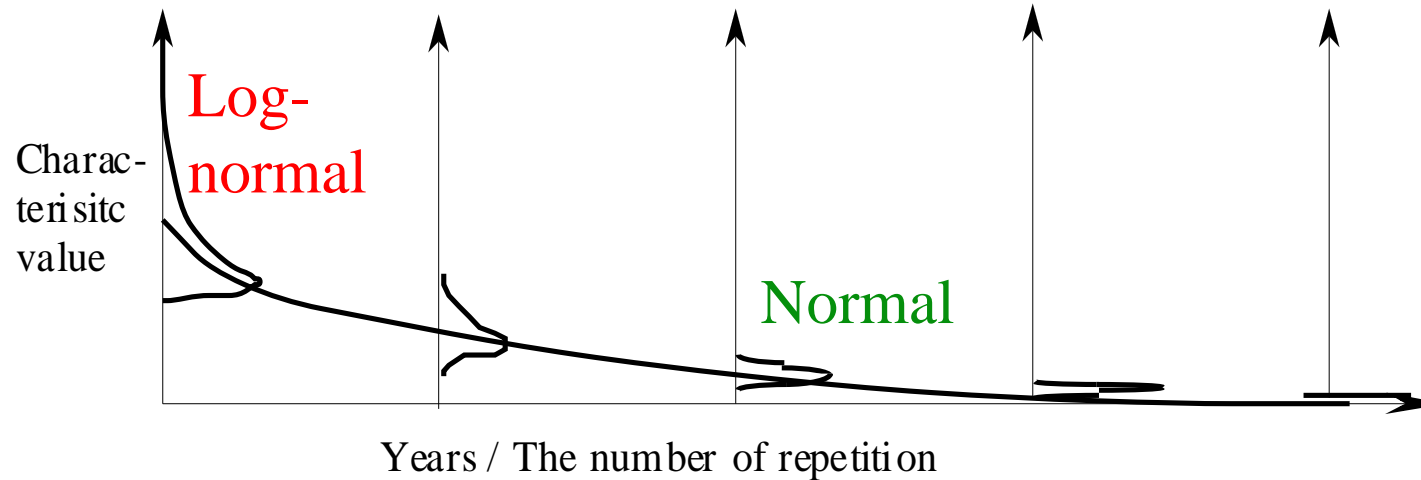
Approach for Improvement

H/W
production

End of
19th C

Beginning
20th C

Beginning
21st C



S/W
Design

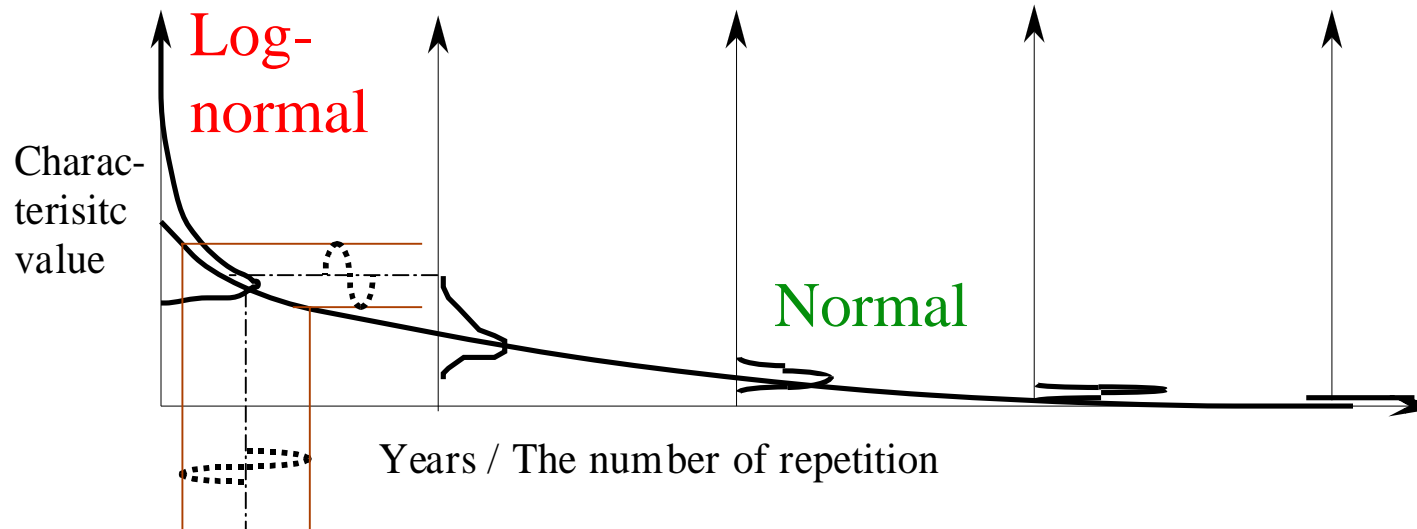
Purge
“Waste”,
“Strain”,
“Imbalances”

Quantitative control

Problem in Software

H/W industries have accumulated all the efforts and succeeded

Example: TOYOTA



Some S/W companies pour people as demanded and withdraw them as their direct works ends, and thus do not allow people to analyse the development result and to accumulate the knowledge.
No feedback and thus no accumulation, no advance.

Conclusion

- Management, design, physical works are human intentional activities.
- From a viewpoint of knowledge, they show the same characteristics.
- Thus the same Quantitative, Rational and Scientific management / control is possible.
(Reported here is abstracted knowledge)
- The problem left is how to penetrate them

Thank you very much for
your kind attention.

Your any questions, comments as
well as objections are wellcome.