Empowered by Innovation NEC

Improvement of The Fault-Prone Class Prediction Precision by The Process Metrics Use

Nobuko Koketsu, N.Honda, S.Kawamura,J.Nomura
NEC Corporation
Makoto Nonaka
Toyo Univ.

Business Domains and Our Chief products

IT Services

Cloud-Oriented Service Platform Solutions

MegaOak

GPRIME

PanelDirector

Platform

Server

Super Computer

WebSAM

Integrated Operation/ Management Middleware

UNIVERGE

Unified Communication

Personal Solutions

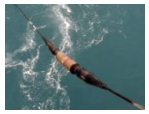
Personal Computers

Mobile Terminals

BIGLOBE

Carrier Network

Long Term Evolution Network Systems



Unity Cable Systems

WiMAX Network Systems

Compact Microwave Communications Systems

Social Infrastructure

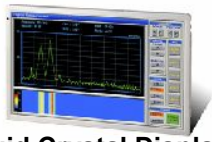
Digital Terrestrial TV Transmitters

Asteroid Explorer "HAYABUSA" (provided by Japan Aerospace Exploration Agency)

Others

Electron Devices

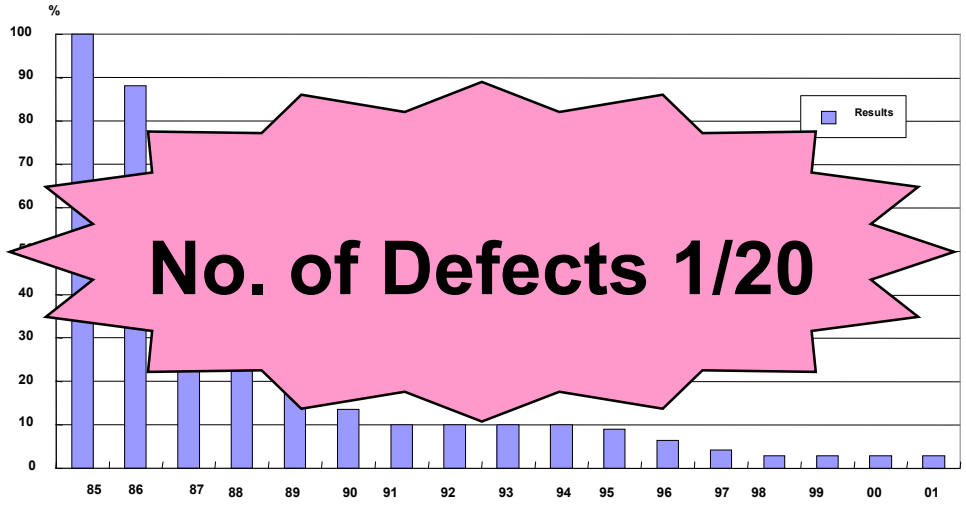
Lithium-ion Batteries



Liquid Crystal Displays

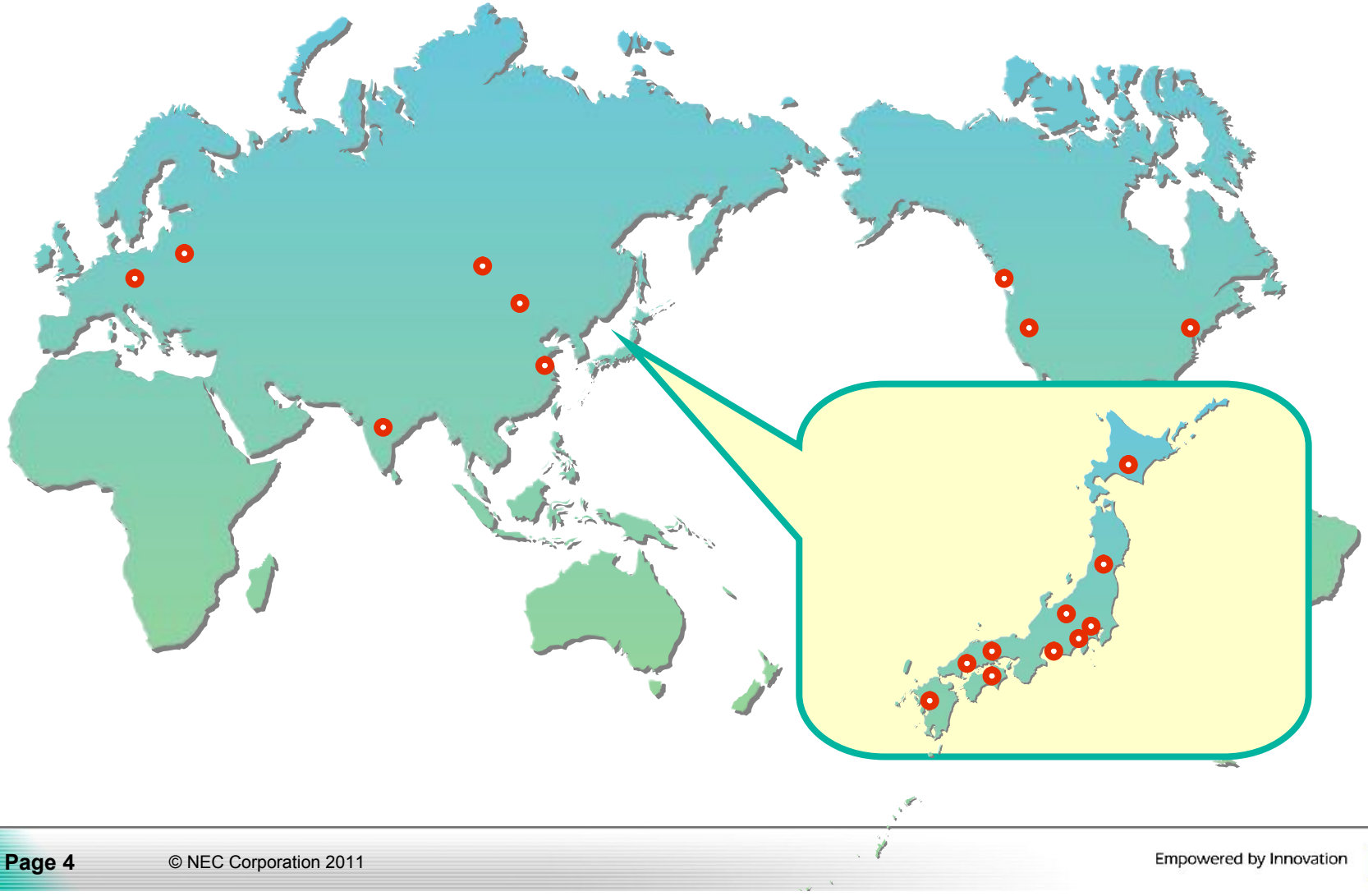
Characteristics of organization

- Continuous software quality improvement activity for more than 20 years
- Quality management with highly matured software life cycle process with technique know as “Quality accounting”
- Attained CMMI level 5 in 2004.



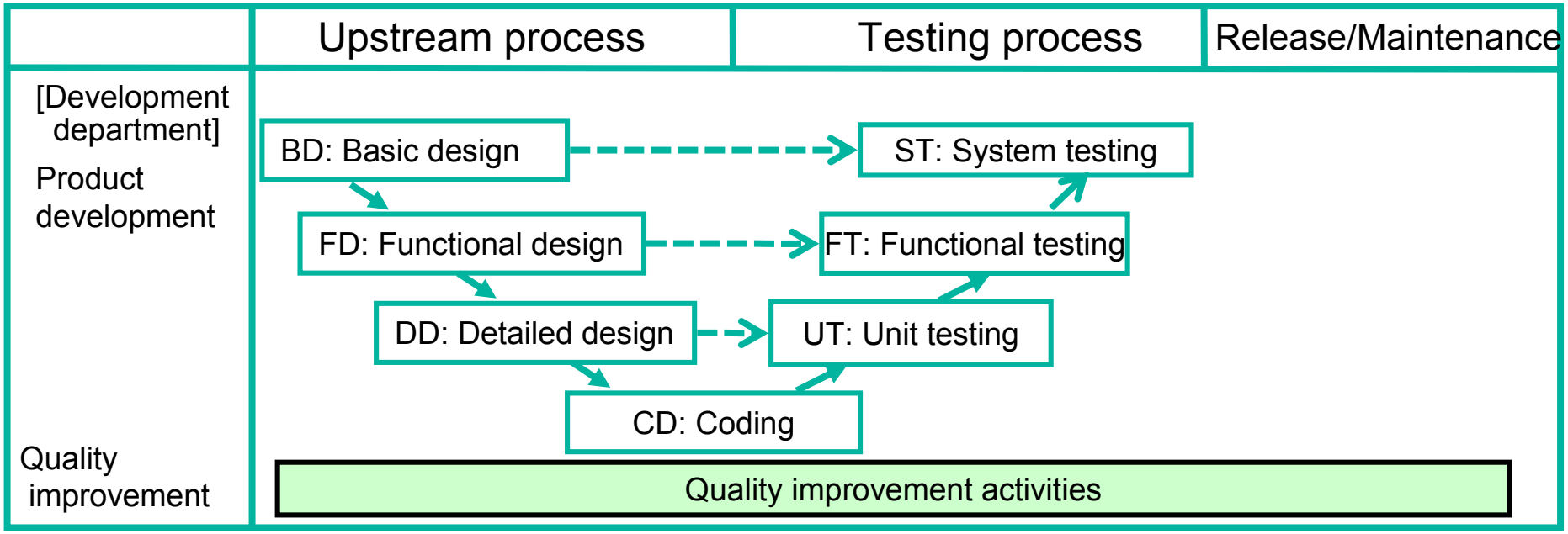
Co-operative Development

Products are made by local development companies and overseas operations



The Process Phase of Software Development

- The process phase : defined as the product development and quality improvement process.
- Each phase defines deliverables, implementation tasks and measurement items



Standardized development process in organized manner

Quality Assurance Process

- Performed independently by the quality assurance department
  - The data is collected at every phase of the development process
  - Analyzed quality objectively and from various angles to identify any issues.

Typical Process Metrics

Phased to be measured	Metrics
Entire development process	Starting date delay (days), completion date delay (days), number of defect/KL, and rate of defect detection
Only upstream process	Rate of work progress, effort/KL, review effort/KL, and defect count/review effort
Only testing process	Execution ratio of test items, number of test items/KL, and testing effort /KL

Established quality assurance activity based on collected metrics

Issues in Current Quality Assurance

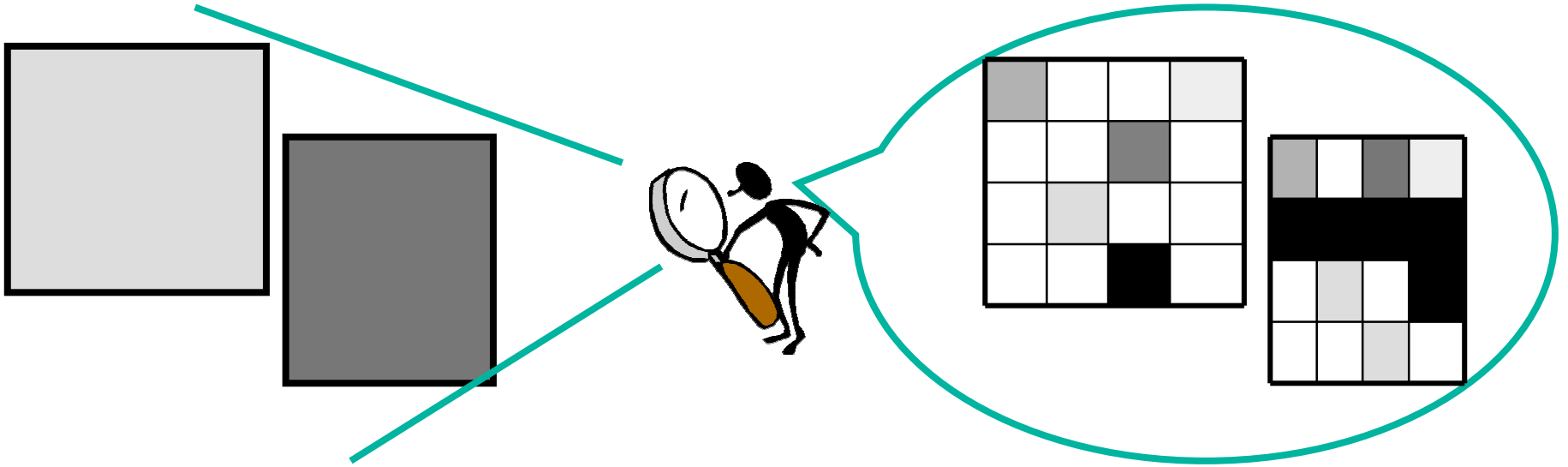
- Product is composed of sub systems
- Quantitative quality management is based on the collected data reported from developer
- Control with higher precision is left to individual quality analysis done by each developer.

- If developer's analysis skill is low, quality problems occur until the last phase
- Analysis skill can not be assessed objectively

Solution

Need to identify quality status
in higher precision

- No dependency on developer's analysis
- In the quantitative value
- More precision that sub-system level



What is the Fault-Prone Class Prediction?

What is Fault-Prone(FP) Module Prediction?

- Method to identify any module that is more likely to contain a fault, from among all the modules constituting the software

Unit of fault-prone module*

- A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.
- A logically separable part of a program

*(reference) Definition of “module” in IEEE Std 610

  - File, Class, Function, Class Method etc…

Could be applied to solve our issues

The Situation of FP Module Prediction

The FP module prediction studies are becoming widespread.

- Initial research started in the 1980s
- Use CK metrics as an explanatory variable for object-oriented software in to the 1990s.
- Study for OSS or excluding the dominant multiple logistic regression model used in the past.

Explanatory variables of FP module prediction

- Metrics collected from source code（most popular）
  - Scale … LOC
  - Complexity level … Cyclomatic complexity
  - Design … CK object-oriented metrics, Fan-in/out
- Information for design documents（less study）
  - Describe a design element graphically and extract it automatically
- Factors related to the process（less study）

Issues in The Application of FP Prediction Study

Our organization and data does not meet the precondition of the analysis of existing study of FP prediction

- Fault number in module to be predicted with multiple regression model of product metrics and process metrics (Shen 1985)
  →Development language, used metrics and fault density in test phase and post-release
- Fault-Prone model prediction with automatic extraction of design information from the specification（Ohlsson 1996）
  → It is difficult to change existing design methodology only for FP prediction
- Predict Fault-Prone Class with CK metrics (Dr.Basili 1996）
  → Target is software development project by students
- Usage of data of Open Source Software Development（2000 ～）
  → It is difficult to apply open source data to the development of commercial software under established quality assurance process
- Fault number to be predicted with process data（qualitative data ）as variant of Bayesian network (Fenton 2007)
  → Different nature from quantitative data measured in the organization

Application to Actual Development

- We built the environment that can automatically measure scale and complexity of source code under development
  It is possible to collect metrics used for FP module predication during development

- The modification of source code that happened after functional testing phase are controlled
  - It is possible to apply model based on modification after functional test and collected metrics

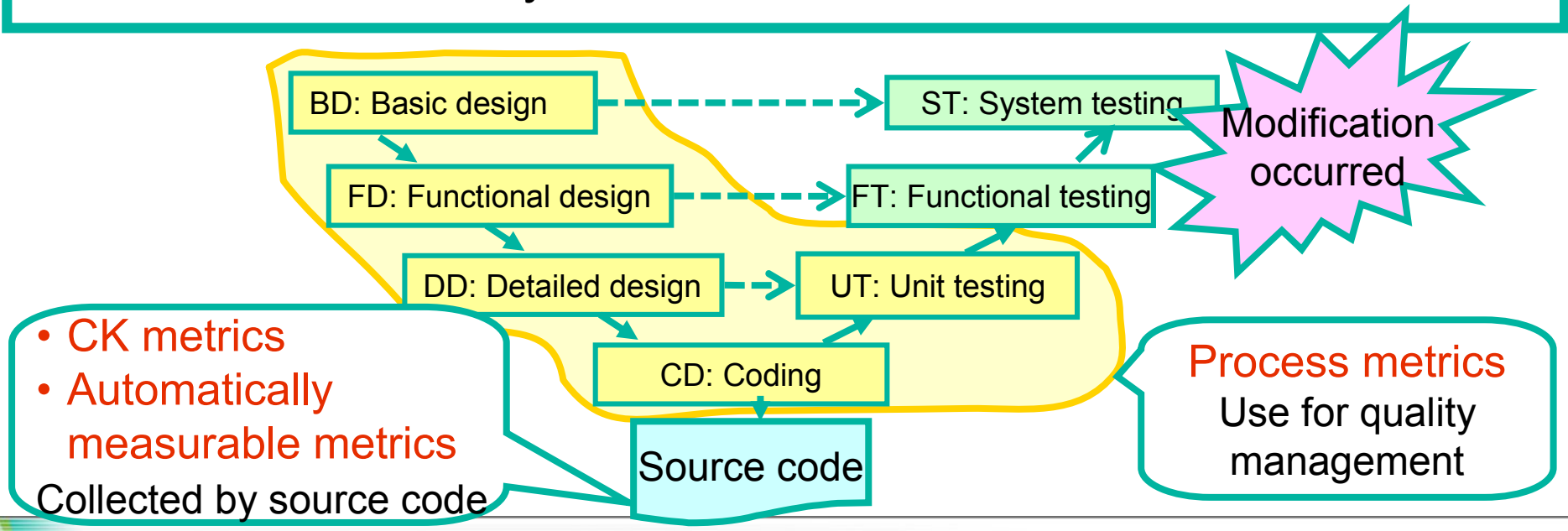
- Effective in improving precision of quality assurance than doing it per subsystem
  c.f. 105KLoc source code　　subsystem **7**：class**215** -> nearly **30** times

Application of FP module prediction to improve the precision of existing quality assurance method

Approach to Apply FP Class Prediction

Using metrics which collected by coding and unit testing phase, we predict the class with the possibility of the modification after functional testing

- Use as candidate of explanatory variables
  - CK metrics
  - Process metrics
  - Automatically measurable metrics



Candidate for Explanatory Variable 1

Process metrics

- Collected until unit testing

Phased to be measured	Metrics
Entire development process	number of defect/KL, and rate of defect detection
Only upstream process	effort/KL, review effort/KL, and defect count/review effort
Only testing process	execution ratio of test items, number of test items/KL, and testing effort /KL

Candidate for Explanatory Variable 2

Automatically measurable metrics
collected from source code

Metrics		Outline
Scale		
	Number of effective lines	Value derived by subtracting the comment and blank lines from the total number of lines (total summation per class)
	Number of methods	Number of methods contained in class
Complexity level		
	Cyclomatic complexity	Value representing route complexity by branching command (total summation per class)
	Number of branch conditions	Number of conditional equations for branching command (total summation per class)
	nesting levels	Class average of maximum number of nesting levels for each method

Candidate for Explanatory Variable 3

CK metrics: Reported as effective for FP module prediction

Metrics	Outline
WMC	Weighted Methods per Class
DIT	Number of hierarchies up to root class in inheritance tree (Depth of Inheritance Tree)
NOC	Number of direct subclasses (Number Of Children)
CBO	A count of the number of non-inheritance related couples with other classes (Coupling Between Objects)
RFC	Total of methods in which an object is executed in response to received messages (Response For a Class)
LCOM	Number of methods in which common attributes are manipulated, which represents a lack of cohesion (Lack of Cohesion in Methods)

Outline of Analysis

- Collected from the initial development of a new product and from the first version-up development
- When the first version-up developed, method for separating the phases was divided into two ways according to its subsystems.
- We use version-up data is divided into two

Characteristics of software used for analysis

Reference symbol	Development content	Development size	Analysis size	Number of classes	Number of subsystems
A	New development	105KL	57KL	215	7
B	Version-up development1	177KL	165KL	611	9
C	Version-up development2	124KL	91KL	450	6

FP Class Predication Evaluation Patterns

- 1 (with fault) and 0 (without fault) were assigned to classes
- 9 evaluation patterns, depending on which data is used for the creation and evaluation of expression
  - P1,P2,P3: creation and evaluation by using the same data
  - P4,P5 : really used pattern
  - P6 to P9 : no use

FP class prediction evaluation patterns

Reference	P1	P2	P3	P4	P5	P6	P7	P8	P9
Data for model creation	A	B	C	A	A	B	B	C	C
Data for model evaluation	A	B	C	B	C	A	C	A	B

Evaluation Indices and threshold for FP Predictions

Evaluation Indices

Evaluation indices	definition
Recall ratio	Ratio of modules correctly determined to be FP among those modules that were actually faulty
Precision ratio	Ratio of modules that were actually faulty among those modules determined to be FP
F value	Harmonic mean of recall and precision ratios. A larger harmonic mean represents a higher-precision determination
Early fault class test ratio	Original index. Average of the rate determined not to be FP among all classes, and a recall ratio

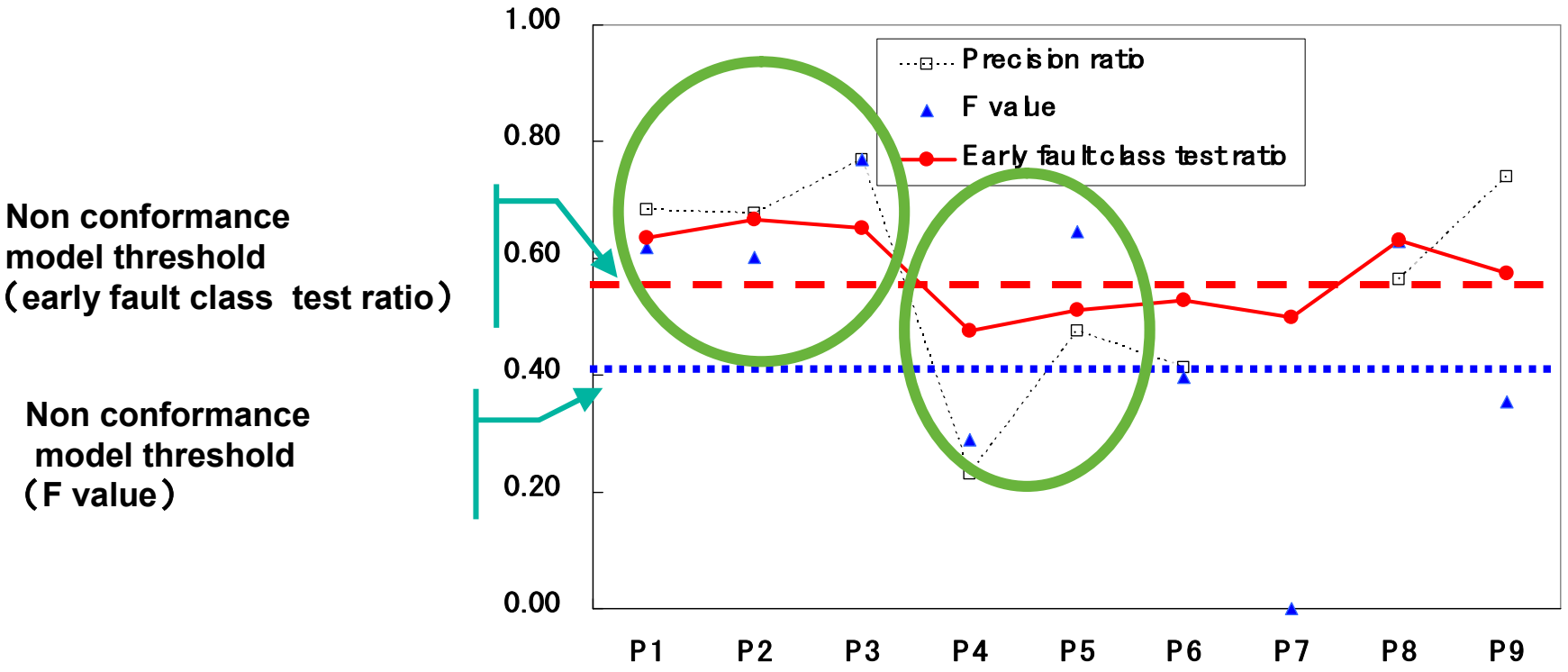
$$F\ value = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

Recall,precision and F value was defined by Kaur,A. and Malhotra,R.

”Early fault class test ratio” 0.55 and F value 0.4 are set as a threshold of model equation based on the experience and knowledge

Evaluation of Logistic Regression Model

- P1 to P3 : Results are high
- P4,P5 : Both results are lower than threshold



Application of Bayesian Network Model

Linear Regression Model assume the absence of a mutually dependent relationship between explanatory variables

Try to predict with a Bayesian Network Model

What is the Bayesian Network Model?

- One type of network model that stochastically describes cause and effect relationships
- A probabilistic inference model that expresses inferences for relationships based on a directed graph
- Individual variable relationships based on a conditional probability

Evaluation method of the network model

Characteristics of Bayesian network models to be evaluated

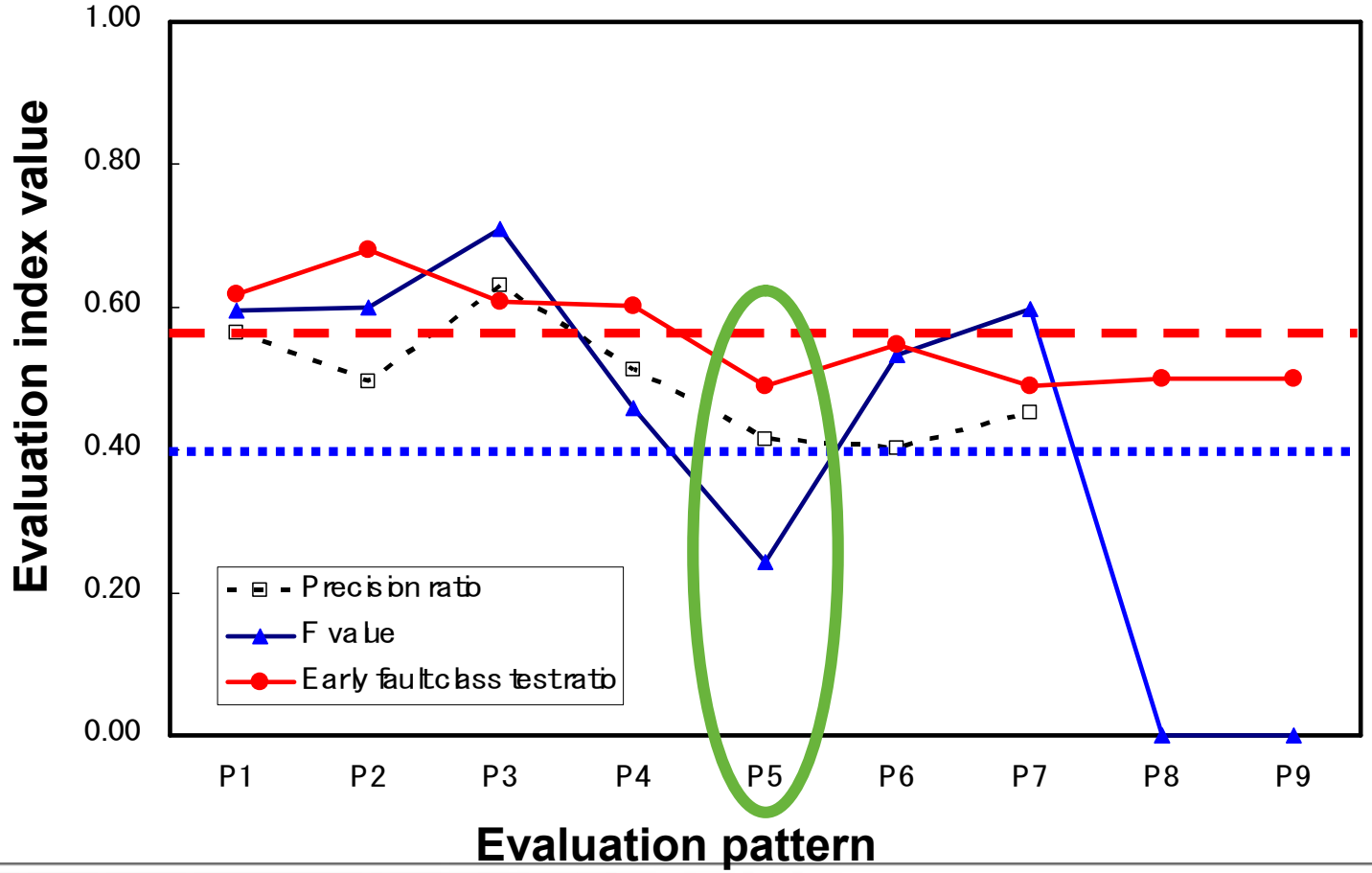
Reference symbol	Model type name	Mutually dependent relationship between objective and explanatory variables	Mutually dependent relationship between explanatory variables
BN1	NaiveBayes	**Required**	**Absence**
BN2	TAN	**Required**	**Presence (Max 1)**
BN3	Bayes Net	**Optional**	**Presence (Max 3)**

TAN: Tree Augmented NaiveBaye TAN: Tree Augmented NaiveBaye

- To evaluate the precision of the prediction of the Bayesian network models
  - We used a bootstrap method
  - 80% of all the samples were randomly selected from data to create a expression.
  - Then, 4000 iterations were made to evaluate the created expression
  - The average obtained from these iterations was treated as the final predictive value.

The Results of Naive Bays Evaluation

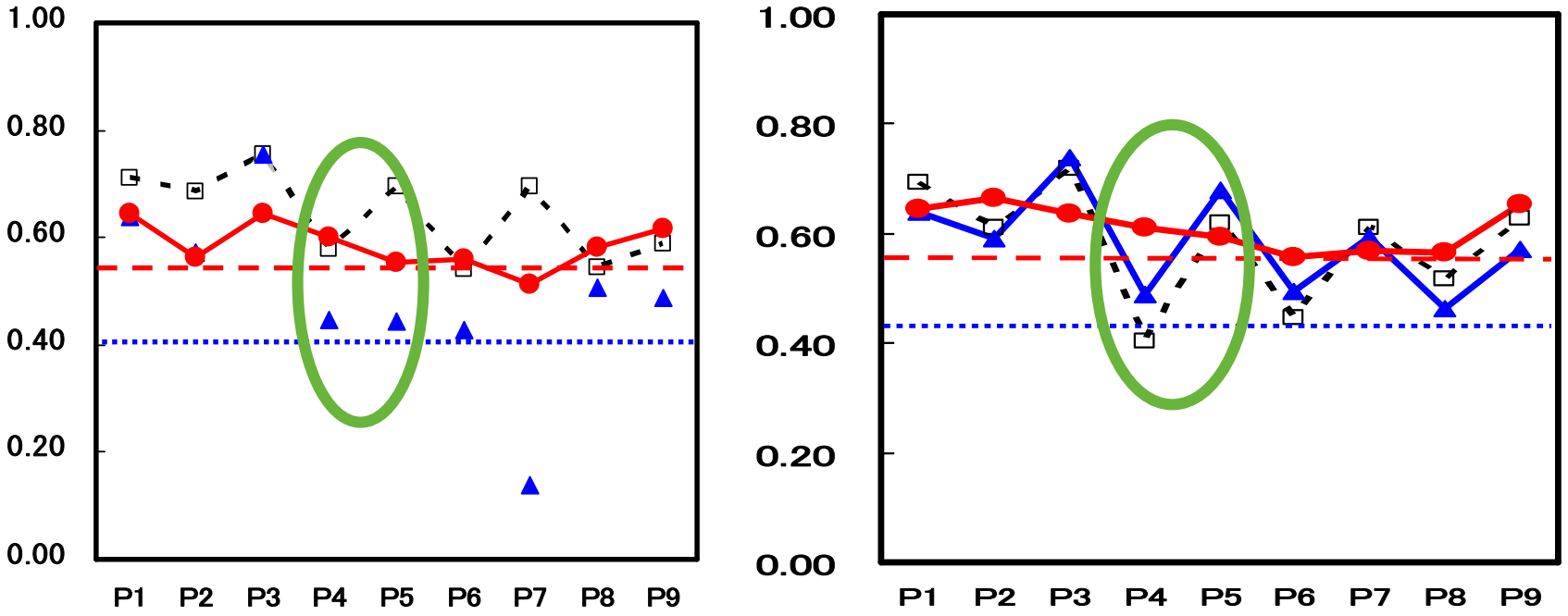
- P1 to P3 :Results are high
- P4's results are higher than threshold
  but P5's results are lower than threshold



The Results of Evaluation(Bayes Net,TAN)

- P4,P5 : Both model's results are higher than threshold
- In particular, all TAN model's evaluation results are higer than threshold

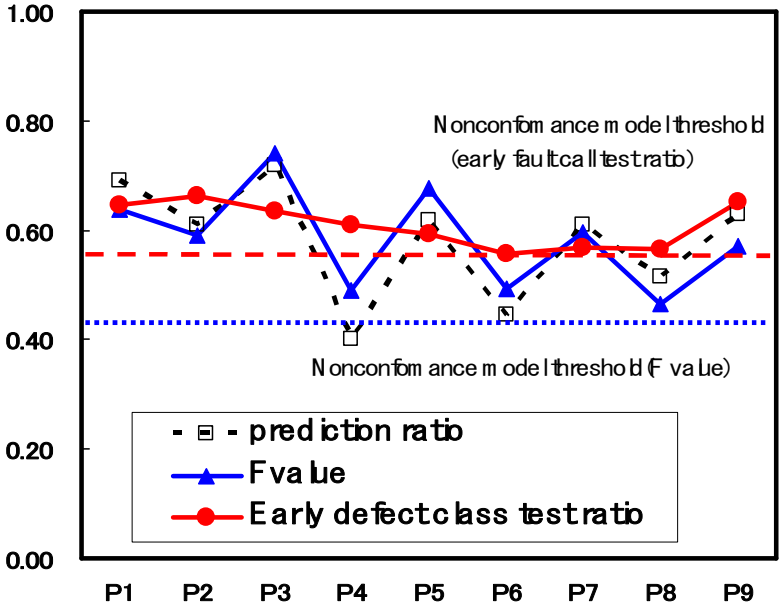
Certain level of robustness could be ensured with TAN model



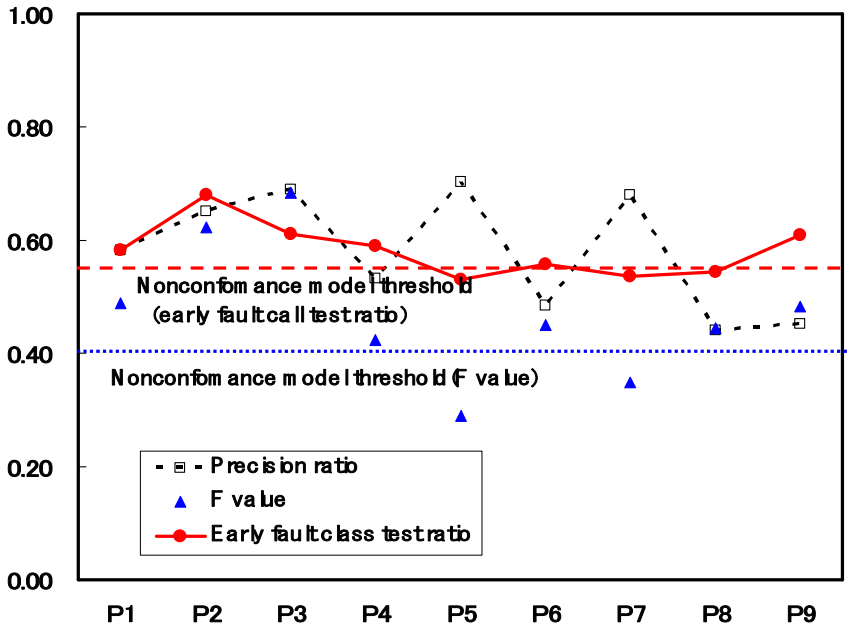
Contribution of Process Metrics

- Without process metrics, the number of nonconformance expression in is larger, and the early fault class test ratios and F values are lower.

Process metrics contribute to the increase in the TAN model prediction precision.



Results of evaluation of TAN



Results of evaluation of TAN (no process metrics)

Creation of the expression to Actual Development

FP Class prediction upon the completion of coding phase.

Accomplishment (subsequent to the functionality testing)		Results of FP class prediction		Total
		the absence of any modification	the presence of any modification	
	Absence of any modification in the steps	303	149	452
	Presence of any modification in the steps	59	100	159
Total		362	249	611

To detect faults in 100 classes

- Without FP class prediction : 63% (100/159) of test items to be executed
- With FP class prediction : 41% (249/611) of test items to be executed

Possible to improve the quality at early stage of the testing phase

Future study

- To increase the precision of the prediction with the accumulation of data
  - Analyze the cause of generated modification well as the classification of those modification

    (A modification in specification level or a fault in coding is treated equally as 1 modification)

- To improve the precision of existing quality assurance method
  - Impact of Complexity to maintainability
  - Identify “Quality” that is measurable with product metrics

Conclusion

- Predict modification classes that occur after functional testing using the FP class prediction
- By applying the network model for prediction, we can construct an expression that ensures a certain robustness for actual products
- Improve the robustness of the expression by adding process metrics to collect metrics from the source code
- Fixed explanatory variable could not be identified, but can identify quality status in higher precision than subsystems
- Proposed a method for improving the quality at early stage of the test phase By using the FP class prediction

NEC Group Vision 2017

To be a leading global company leveraging the power of innovation to realize an information society friendly to humans and the earth

Empowered by Innovation

NEC