

Analyses Based on the Data Collected by EPM

1st International Software Project Analysis Workshop (2006)

Tomoko Matsumura

Postdoctoral Researcher of EASE Project
(<http://empirical.jp>),

Nara Institute of Science and Technology

Analysis Menu

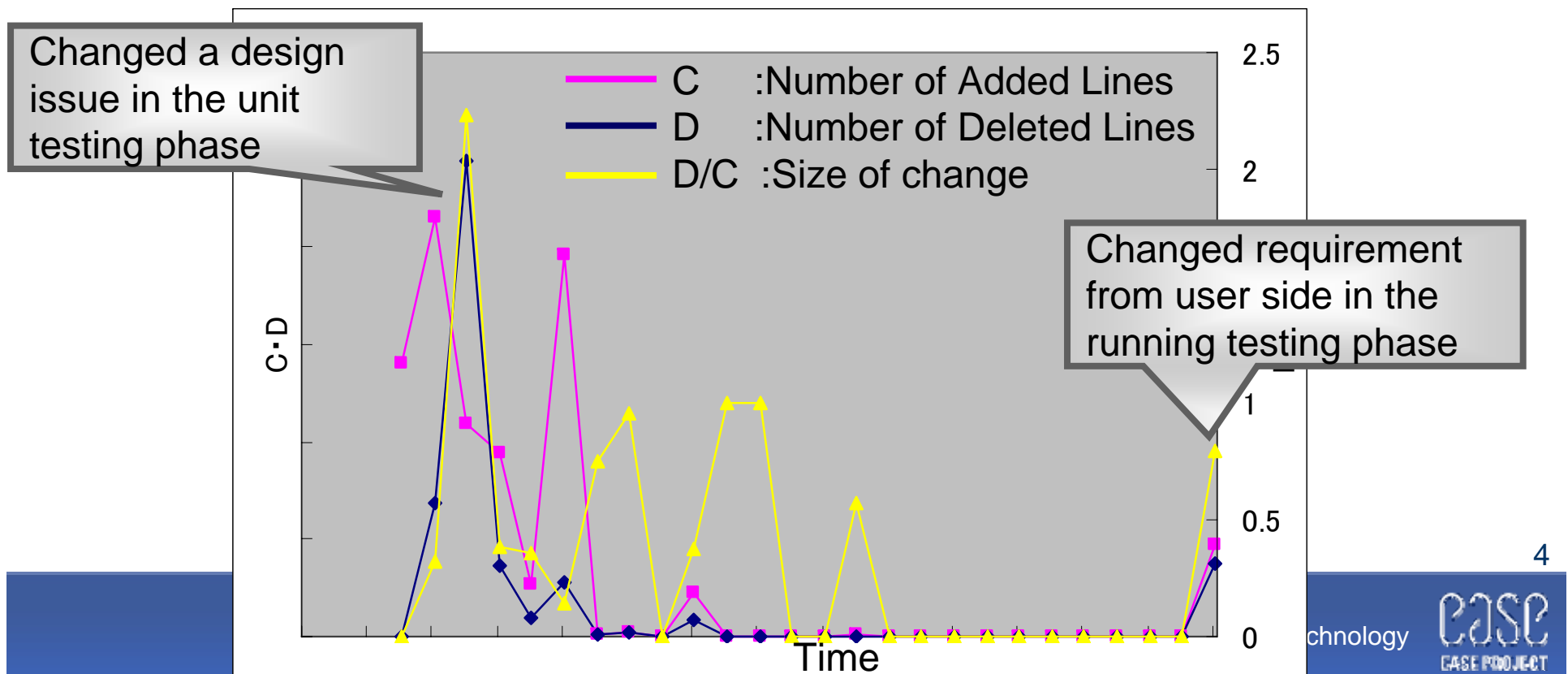
- Risk Detection Preventing Project Delay
 - Detect risks of project delay (e.g. Unstable Requirements, Incomplete Designs, Low Quality Program or Inappropriate Resource Planning) by monitoring the program change history
- Logical Coupling Analysis
 - Discover implicit knowledge for system maintenance to reduce mistakes or lack of needed changes
 - Relationships among files (modules) frequently changed at the same time
- Factor Analysis of Defect Correction Effort
 - Improve software process taking into account return on investment

Data & Project Outline

- COSE : Consortium for Software Engineering
 - Includes 6 Companies : Japanese Big Software Vendor
 - Fujitsu
 - Denso
 - NEC
 - Matsushita
 - Hitachi
 - NTTData (Project Management)
- Probe Information System Project (Phase1)
 - 2005/4 – 2006/1
 - Phase2 is continuing now
- We applied the analyses to this project

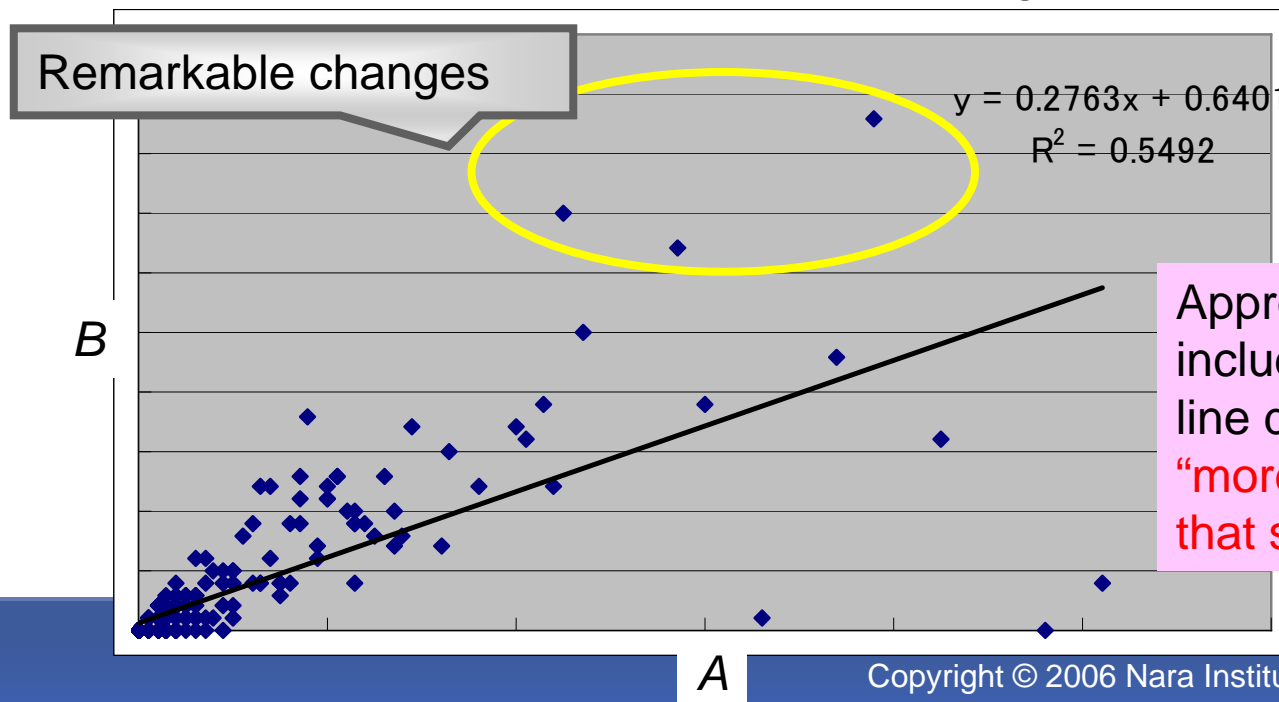
Risk Detection Preventing Project Delay(1)

- Visualizing changed volume of metrics weekly or in a regular period for each module
 - E.g. size of change weekly for Module A



Risk Detection Preventing Project Delay(2)

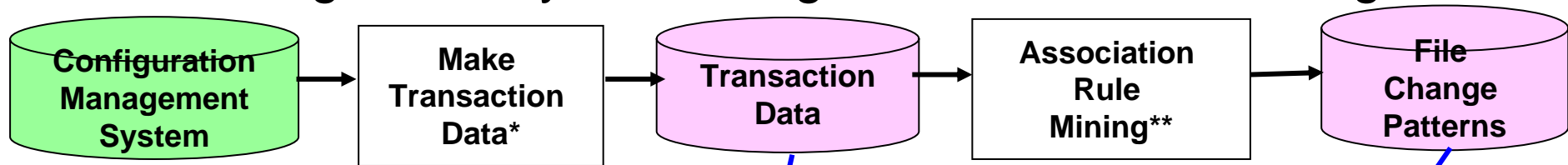
- Give the alert for project managers depending on a certain threshold
 - E.g. change frequency weekly for each module
 - A: Number of update weekly
 - B: Number of updates including a certain number of line deletion



Approximately 30% of changes include a certain amount of line deletion normally -> "more than 30%" implies that some problems happened

Logical Coupling Analysis(1)

- Analyze change history from Configuration Management System using association rule mining



Trans. No.	Date of Update	Author	Number of Files	File1	File2	File3	File4	File5	File6	...
1	2000/2/26 9:28:42	tomoko	21	FileA	FileB	FileC	FileD	FileE	FileF	
2	2000/2/27 9:42:12	tomoko	263	FileG	FileH	FileB	FileI	FileJ	FileK	
3	2000/2/27 11:03:35	noriko	4	FileK	FileL	FileM	FileN	FileO	FileP	
4	2000/2/28 1:30:28	kohei	5	FileF	FileG	FileH	FileI	FileJ	FileK	
5	2000/2/28 1:36:06	noriko	2	FileP	FileQ	FileR	FileS	FileT	FileU	
6	2000/2/28 1:37:14	tomoko	2	FileF	FileG	FileH	FileI	FileJ	FileK	
7	2000/2/28 1:38:07	kohei	1	FileT	FileU	FileV	FileW	FileX	FileY	
...										

No. of Patterns	Count of changed together	Support	Number of files	File 1	File 2	File 3	File 4
1	9	0.00438	4	FileL	FileM	FileN	FileP
2	9	0.00438	4	FileF	FileG	FileH	FileI
3	9	0.00438	4	FileC	FileJ	FileE	FileK
4	9	0.00438	4	FileC	FileD	FileE	FileK
5	9	0.00438	4	FileA	FileB	FileC	FileK
6	10	0.00486	3	FileN	FileO	FileP	
7	9	0.00438	3	FileM	FileN	FileP	
.....							

*Thomas Zimmermann, Peter Weißgerber: "Preprocessing CVS Data for Fine-Grained Analysis", Proc. International Workshop on Mining Software Repositories (MSR), Edinburgh, Scotland, UK, May 2004

**Apriori Algorithm : R. Agrawal, R. Srikant: "Fast algorithm for mining association rules", Proc. 20th Very Large Data Bases Conference(VLDB), pp.487-499. Morgan Kaufmann, 1994

Logical Coupling Analysis(2)

- Analyze each pattern by confidence
 - Association Rule : X (Antecedent) \Rightarrow Y (Consequent)
 - Confidence (X \Rightarrow Y) : # of Update (XuY) / # of Update (X)

Confidence	# of Update (Antecedent)	# of Update (Consequent)	Antecedent		Consequent
0.8	5	4	FileA.cpp FileB.h FileC.cpp	->	FileD.h
1	4	4	FileA.cpp FileB.h FileD.h	->	FileC.cpp
1	4	4	FileA.cpp FileC.cpp FileD.h	->	FileB.h
1	4	4	FileB.h FileC.cpp FileD.h	->	FileA.cpp

Has "FileD.h" missed a change?

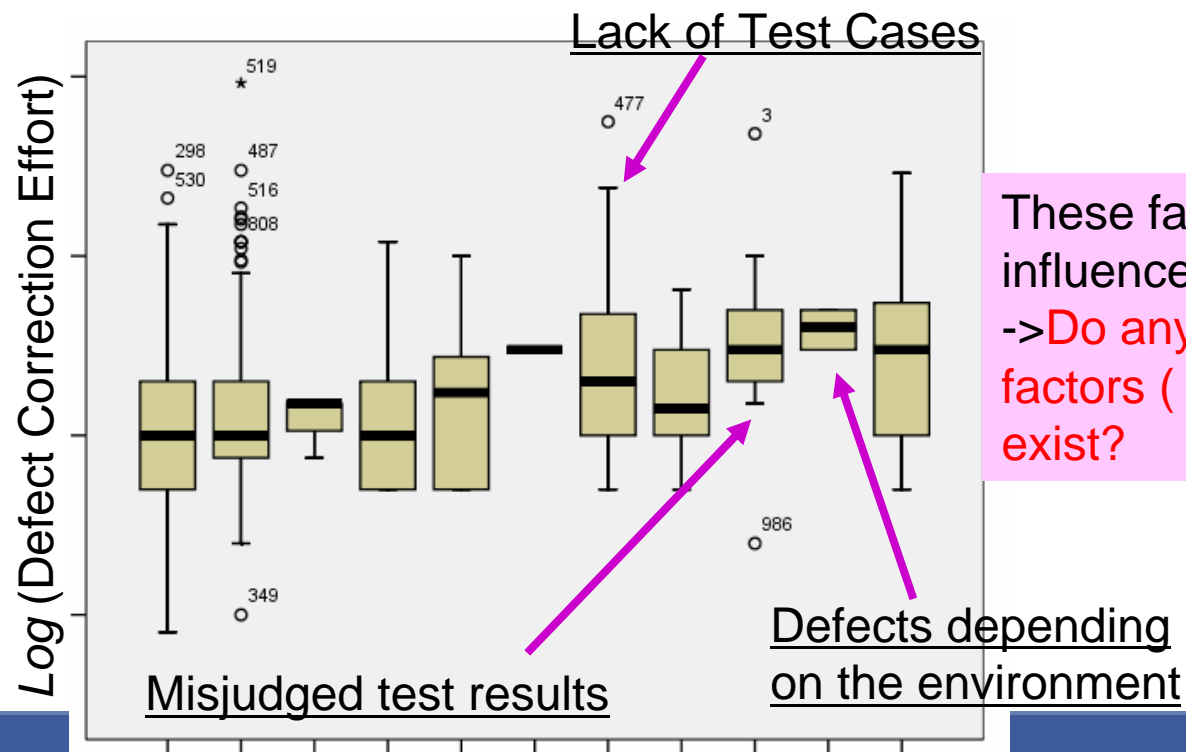
Was "FileY.cpp" copied from "FileX.cpp"?
Or does "FileY.cpp" strongly depend on the "FileX.cpp"?

Confidence	# of Update (Antecedent)	# of Update (Consequent)	Antecedent		Consequent
0.5	8	4	FileX.cpp	->	FileY.cpp
1	4	4	FileY.cpp	->	FileX.cpp

7

Factor Analysis of Defect Correction Effort(1)

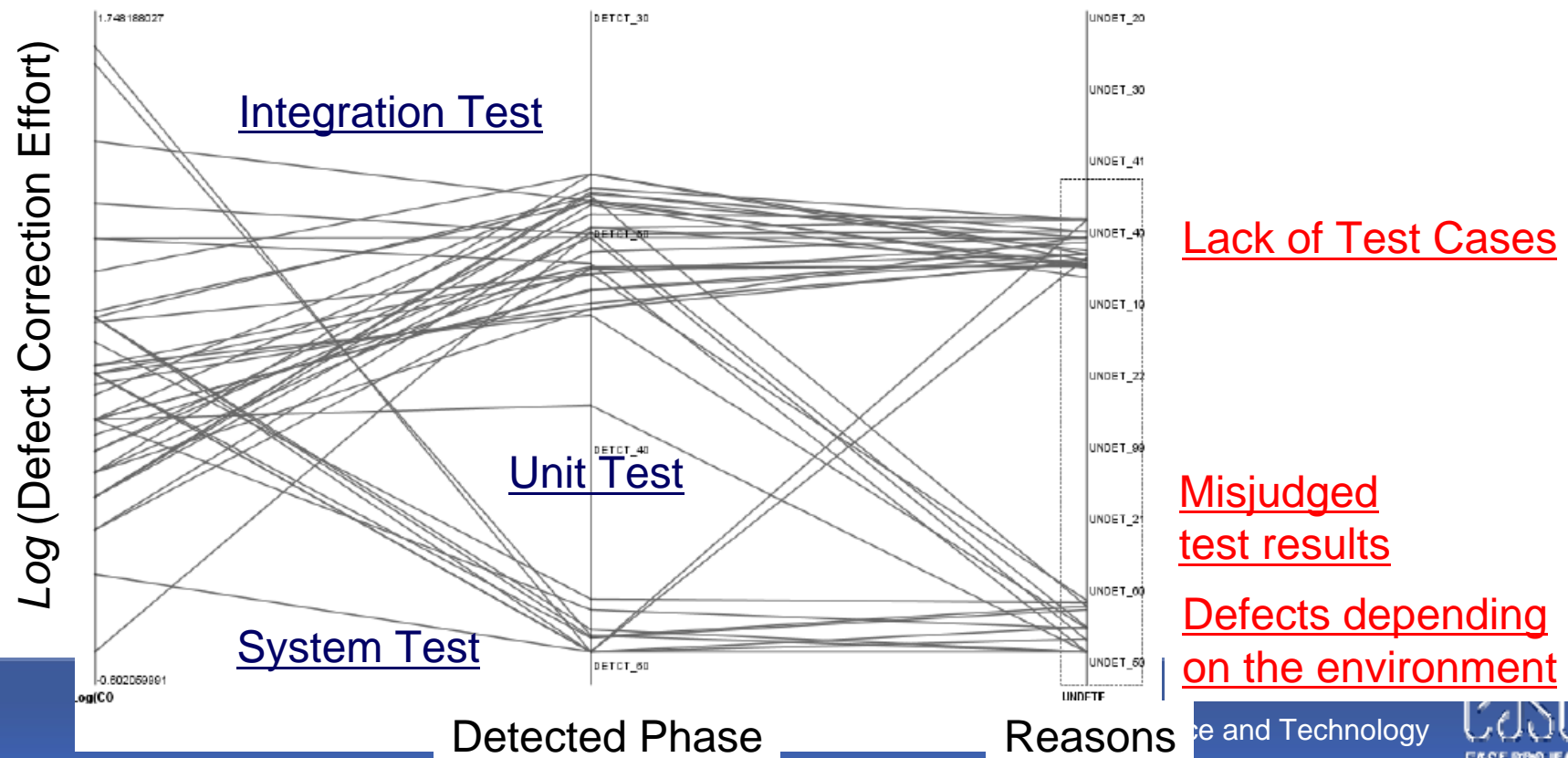
- E.g.) Defect Correction Effort classified by “Reason why the defect was not detected in the preferable phase”



These factors seem to influence the correction effort
->Do any relations to another factors (e.g. detected phase) exist?

Factor Analysis of Defect Correction Effort(2)

- E.g.) Relation Analysis between “Reason why the defect was not detected in the preferable phase” and “Defect detected phase” using Parallel Coordinate Plot



Lack of Test Cases

Misjudged test results

Defects depending on the environment

END of Slide

What is the EPM tool?

- Automatic data collection tool from the CASE tools
 - Configuration Management System(e.g. CVS)
 - Bug Tracking Tool(e.g. GNATS)
 - Mailing List Management Tool(e.g. Mailman)
- Data Collected or Calculated by EPM
 - LOC, # of Files, # of Changes, # of added, Deleted or Changed lines, # of author for each file etc.
 - Sets of files updated at the same time
 - # of Detected Bugs, # of Unsolved Bugs, Detected and Fixed Date, Resolution Time, Cause, Detected Phase etc.
 - # of mails for each topic, Author and Receiver of mails etc.