



Mikko Rönkkö, Antero Järvi, Markus M. Mäkelä

Measuring and Comparing the Adoption of Software Process Practices in the Software Product Industry

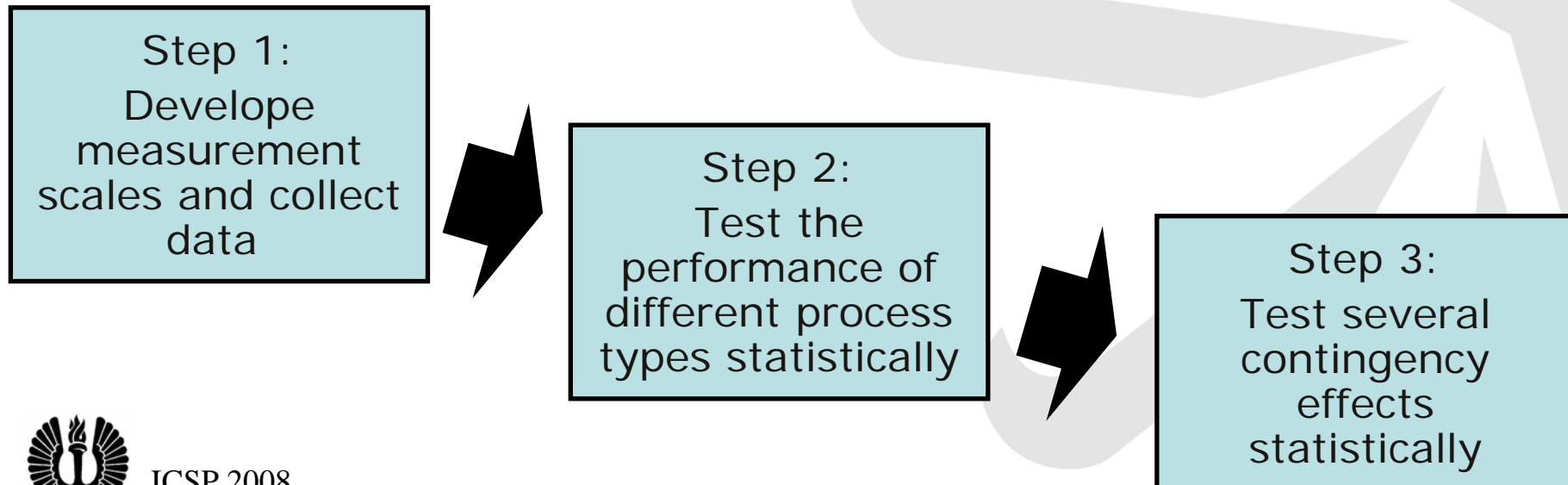
Content of the Presentation

- 1) Introduction
- 2) Research approach
- 3) Results
- 4) Discussion and future research



Sample survey of Finnish software product industry

- Three linked surveys during spring and summer 2007
- Enables us to statistically test several common beliefs that current lack larger scale empirical support (e.g. Agile Methods work best in turbulent environments)



Why this study?

Sample surveys in software engineering

- Most empirical research is case studies, which cannot establish full generalizability of the results
- Studies using researcher intervention need to be complemented with "neutral observation"
- Lack of validated measurement scales mentioned as one reason for of survey studies

Sample surveys in other disciplines

- Used extensively in e.g. psychology, sociology and medicine
- Common procedure is to first create a model with small sample studies and then establish the validity with a statistical study

Our aim is to provide validating (or refuting) evidence on the performance and compatibility of mature processes and agile methods



Development of the survey instrument

- 21 CMMI 1.2 process areas (excluding IPPD additions), 12 XP practices and 5 Scrum practices
- Indicative Likert-type question developed for each practice

Data collection

- Mail and internet survey of all Finnish software product firms
- Follow-up study to software development managers of all product firms with more than 5 employees (123 firms)

Data analysis

- Rasch model
- Correlations and partial correlations



Development of the questionnaire

- 1) CMMI 1.2 Process areas, XP Practices and Scrum practices, adopted verbatim from the sources
- 2) Items were iteratively reduced in length by focusing on what is the core of the process area
 - E.g. CM = "We use a configuration management system",
DAR = "We make the more important decisions concerning software development by comparing alternatives by using predefined criteria."
- 3) All process areas were presented as Likert questions (1 = never used, 7 = always used)
- 4) Two academics not related to scale development reviewed the questions
- 5) The form was sent for commenting to several companies

Altogether, 37 questions were generated



Sample and data collection

Data collection

- First survey was sent to CEOs of all Finnish software product firms (2616) resulting in 287 responses
- The CEO nominated one person as the manager of software development
- Second survey sent out to 123 software development managers resulting in 86 responses

Sample characteristics

- Mean size 33 persons, only 7 firms employing more than 50 persons
- Average firm age 11 years
- Software product firms located in Finland



Survey resulted in 86 responses

- 37 questions indicating use of practices or process areas
- Self assesment (1 = never used, 7 = always used)

Aims of data analysis

- Check if there is a "natural order of adoption" of different practices
- Calculate scores for the following variables
 - Process maturity (=how far with the CMMI the firm is)
 - Use of XP practices
 - Use of Scrum practices
 - Use of Agile methods (combination of the two previous)
 - All items



Each seven point scales is converted to seven binary variables

- E.g. "Response to CM-question is at least 3"

Five scalings were conducted

- CMMI items
- XP Items
- Scrum items
- All agile items (Scrum & XP)
- All items (Scrum, XP & CMMI)

Scaling establishes a score for each firm

- E.g. "How far in the CMMI model has the firm proceeded"

Scaling establishes a score for each item

- E.g. "How far in the CMMI model firms need to have proceeded before they implement this item with 50% probability"



Rasch analysis, statistical principle

Establishes measures for item difficulty and respondent capability

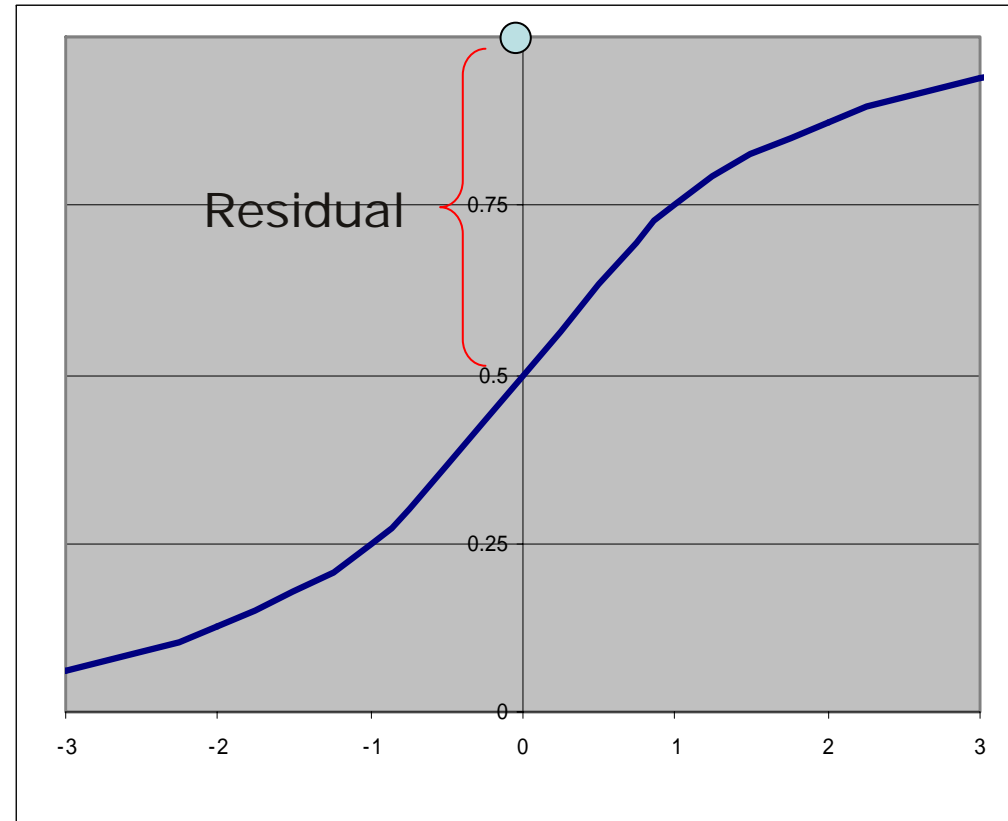
Iterative algorithm fits data to logit-curves and works to minimize the sum of squared residuals

X-axis is the difference of item difficulty and respondent capability.

1 on Y-axis indicates that the respondent has implemented the practice, 0 that it has not been implemented

All items have initially zero difficulty and all firms zero ability

If item or firm has positive sum of residuals, the difficulty/ability score is increased for the next iteration



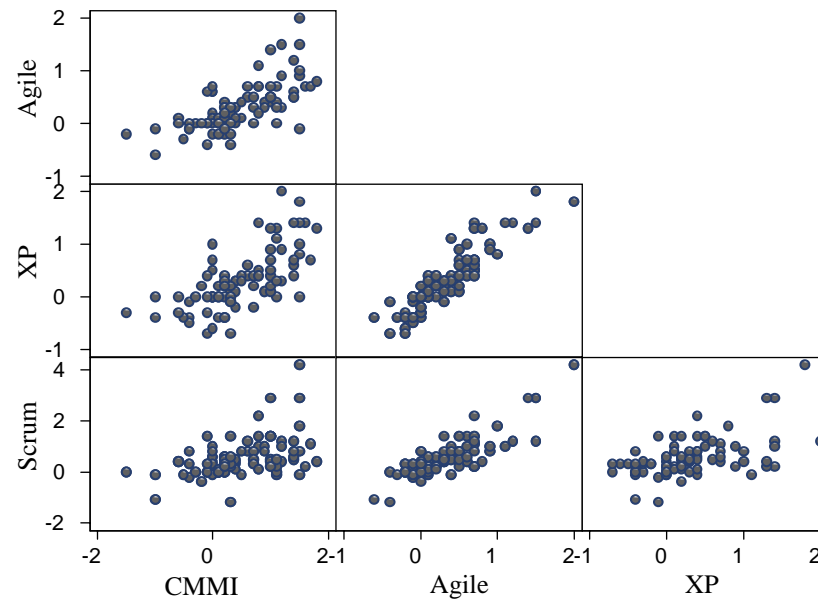
Results, Big Picture

Scale reliability

- CMMI Scale and a scale containing all agile items seem reliable
- Scrum scale would need more items to be reliable

Big picture of the results

- Agile methods and process maturity scale are linked in the data
- Correlations can be caused by common method variance, but this is cannot fully explain the high correlations



Variable	1	2	3	4	5	6	7
1 CMMI		-.08	-.33 **	-.27 *	.43 **	.22 *	.12
2 XP	.67 **		.26 *	.85 **	.68 **	-.05	-.23 *
3 Scrum	.47 **	.54 **		.69 **	.40 **	.06	.01
4 Agile	.67 **	.91 **	.83 **		.69 **	.01	-.15
5 Combined	.94 **	.83 **	.66 **	.87 **		.12	-.08
6 Age	.16	.01	.10	.05	.13		.23 *
7 Size	.16	-.14	.09	-.05	.09	.23 *	

Lower diagonal shows correlations, upper diagonal partial correlations.

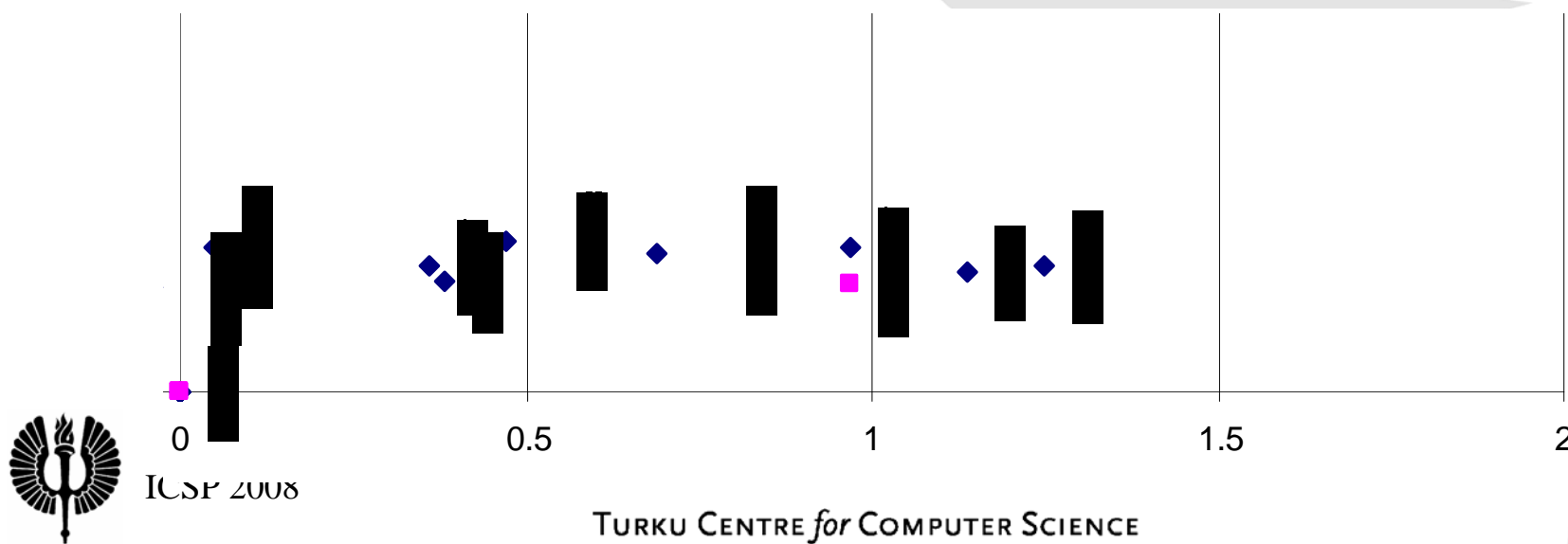
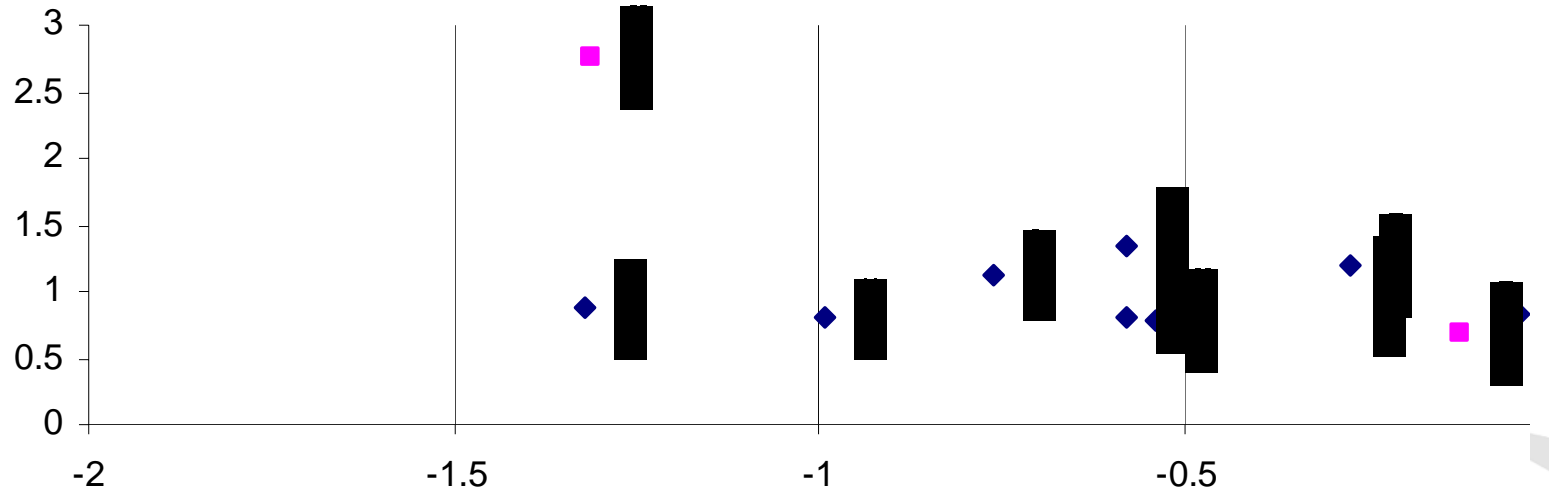
Two-tailed tests of significance. † = p < .10, ★ = p < .05, ★★ = p < .01

Example of the Statistical Results

Item	Descriptives		Rasch results			
	Mean	S.D.	Mea- sure	S.E.	Infit M.S.	Outfit M.S.
We deal with programming errors by looking for their fundamental causes from our own ways of working. (CAR)	4.13	1.24	.47	.09	1.19	1.19
We use a version management system. (CM)	6.19	1.58	-1.31 ^a	.12	2.75	2.43
We make the more important decisions concerning software development by comparing alternatives by using predefined criteria. (DAR)	4.24	1.43	.38	.09	.87	.95
We use metrics when monitoring the progress and efficiency of our software development. (MA)	3.33	1.50	.94 ^b	.09	.92	.94
We continuously innovate and develop our software development processes to better support our business. (OID)	4.88	1.40	-.19 ^b	.10	.71	.72
We maintain a collection of the methods and tools of software development that we have used. (OPD)	4.97	1.70	-.27	.10	1.20	1.18
We develop our software process by analyzing strengths, weaknesses, and development opportunities. (OPF)	4.28	1.45	.36	.09	1.01	1.02
We monitor our software development process and product quality by metrics. (OPP)	3.14	1.50	1.14	.09	.94	.94
We train software developers according to an existing training plan. (OT)	3.31	1.65	.97	.09	1.14	1.10
During software development, we continuously focus on the integration of our software components. (PI)	5.38	1.36	-.58	.11	.80	.79



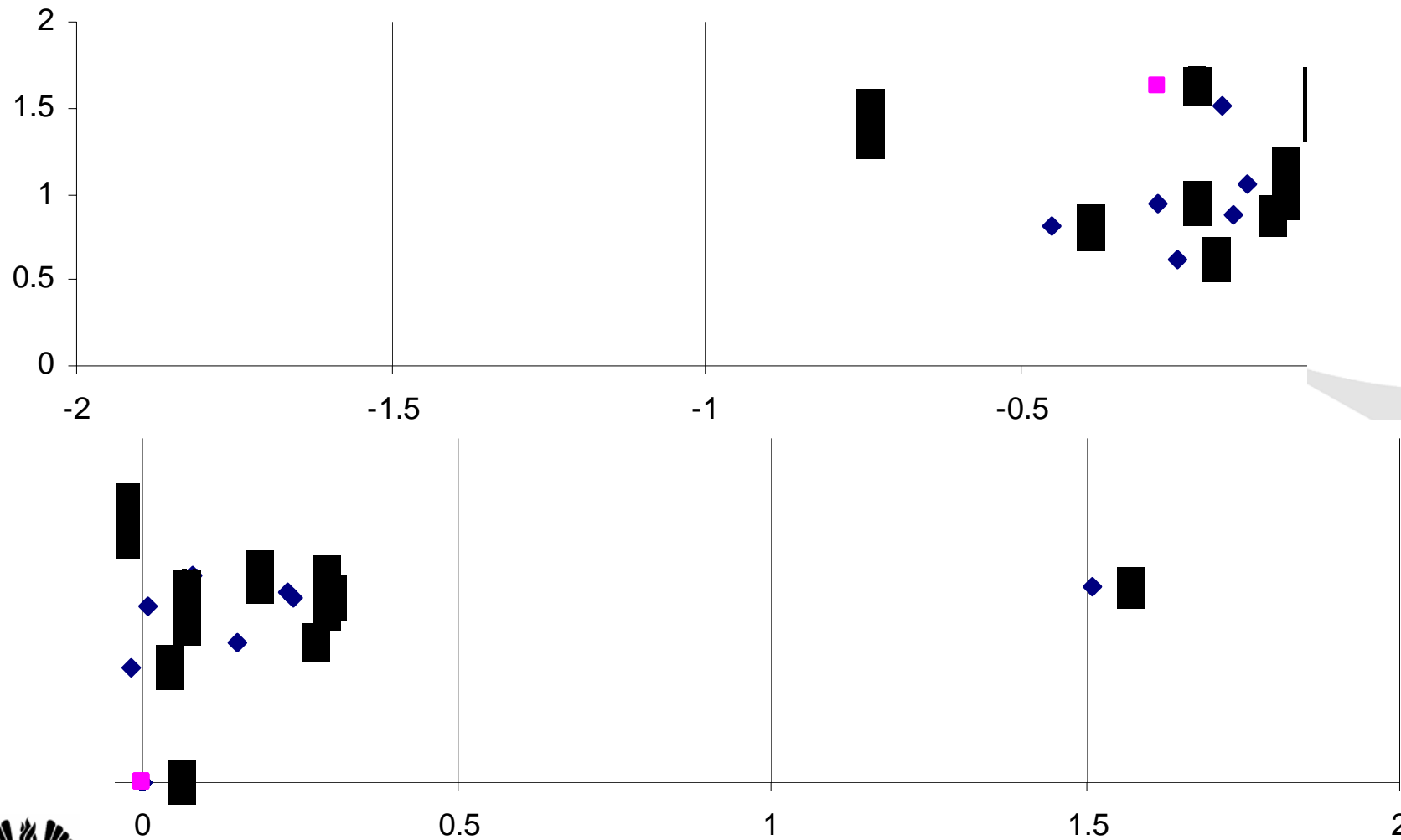
Graphical Illustration, CMMI items



ICSP 2008

TURKU CENTRE for COMPUTER SCIENCE

Graphical Illustration, Agile Items



Process Area Level Considerations

Our data show that the practices indicative to CMMI process areas seem to have a natural order of adoption that aligns with the intended order

Process areas belonging to levels two and three seem to be adopted in parallel.

XP practice “pair programming” is implemented rarely



Limitations of the Study

Specific issues

- The population consists mainly of small firms, and the sample is biased toward smaller firms
- Quality control problems with some of the items
- Some of the items are poorly balanced
- Common method bias possible

The implications of the issues

- Any single practice level conclusions are probably not stable if the study is to be repeated
- However, the reliability and validity of the scales as measuring process maturity and use of agile methods is at an acceptable level





TUUS

Discussion and Future Research (1)

Contributions of this study

- Development of a relatively compact measurement scale
- One of the first larger scale investigation of the relationship between process maturity and use of agile methods
- Provides empirical evidence for CMMI model without the possible bias caused by researcher intervention

Tasks for further research

- Improve the scale by analyzing and improving the poorly performing items
- Replication studies with larger samples



ICSP 2008

TURKU CENTRE *for* COMPUTER SCIENCE

Preliminary results of further analyses

- Process maturity is a strong and significant determinant of product development capability
- Agile methods further improve product development efficiency and effectiveness
- Firms that have a more aggressive strategy benefit more from process maturity

Parts 2 and 3 of this study will be reported in later publications

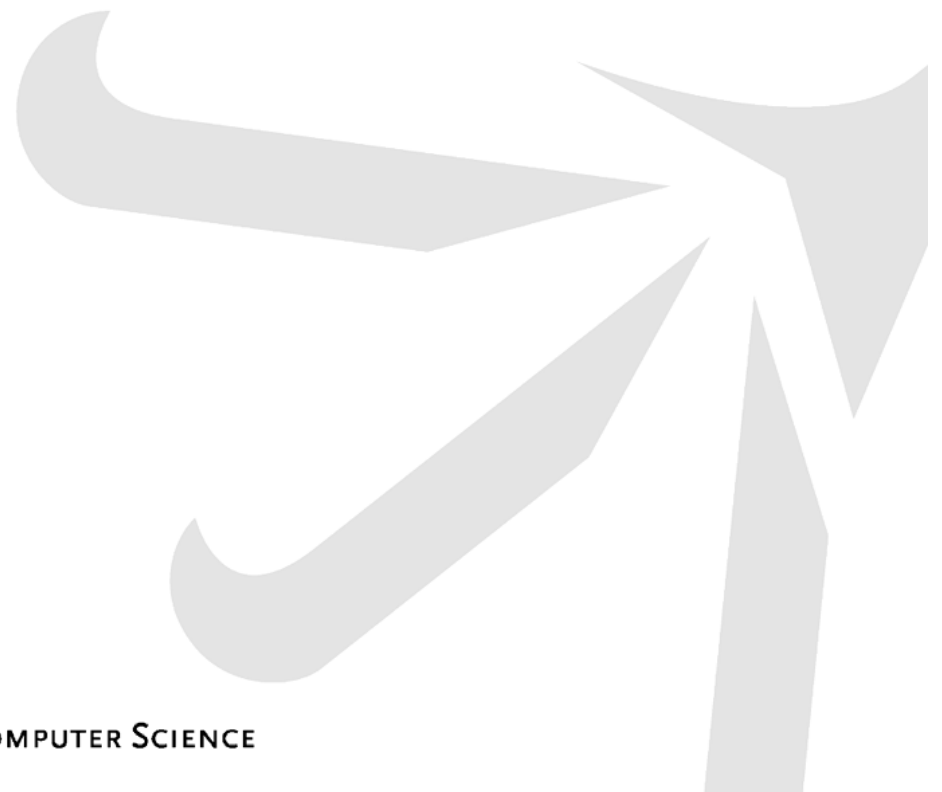
- Rönkkö & Mäkelä: Contingency Effects of Strategy on Software Product Development Capabilities, to be presented at Academy of Management Meeting 2008, August 8-13, Anaheim, California
- Rönkkö & Mäkelä: The Effects of Development Process Maturity and Agility on Software Product Development Performance, submitted for review to Twenty Ninth International Conference on Information Systems, Dec 14-17 2008, Paris





Thank you

Questions and comments are welcome!



ICSP 2008

TURKU CENTRE *for* COMPUTER SCIENCE