Empirical Assessment of Cost Factors and Productivity during Software Evolution through the Analysis of Software Change Effort

Hans Christian Benestad

June 30, 2009

Presentation of thesis for the degree of Ph.D.

Supervisors
Bente Anda
Erik Arisholm

UNIVERSITETET I OSLO

[simula . research laboratory]
· by thinking constantly about it ·

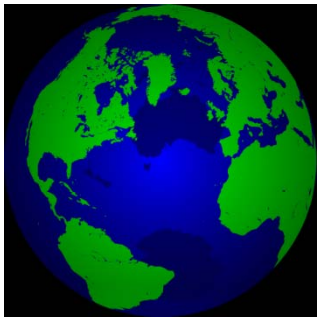
Changes to operational software are inevitable*

Corrections:
Developers will commit errors

Adaptations:
Technological environments will change

Perfective/Enhancive:
Users will require more functionality

Dimensions of maintenance**

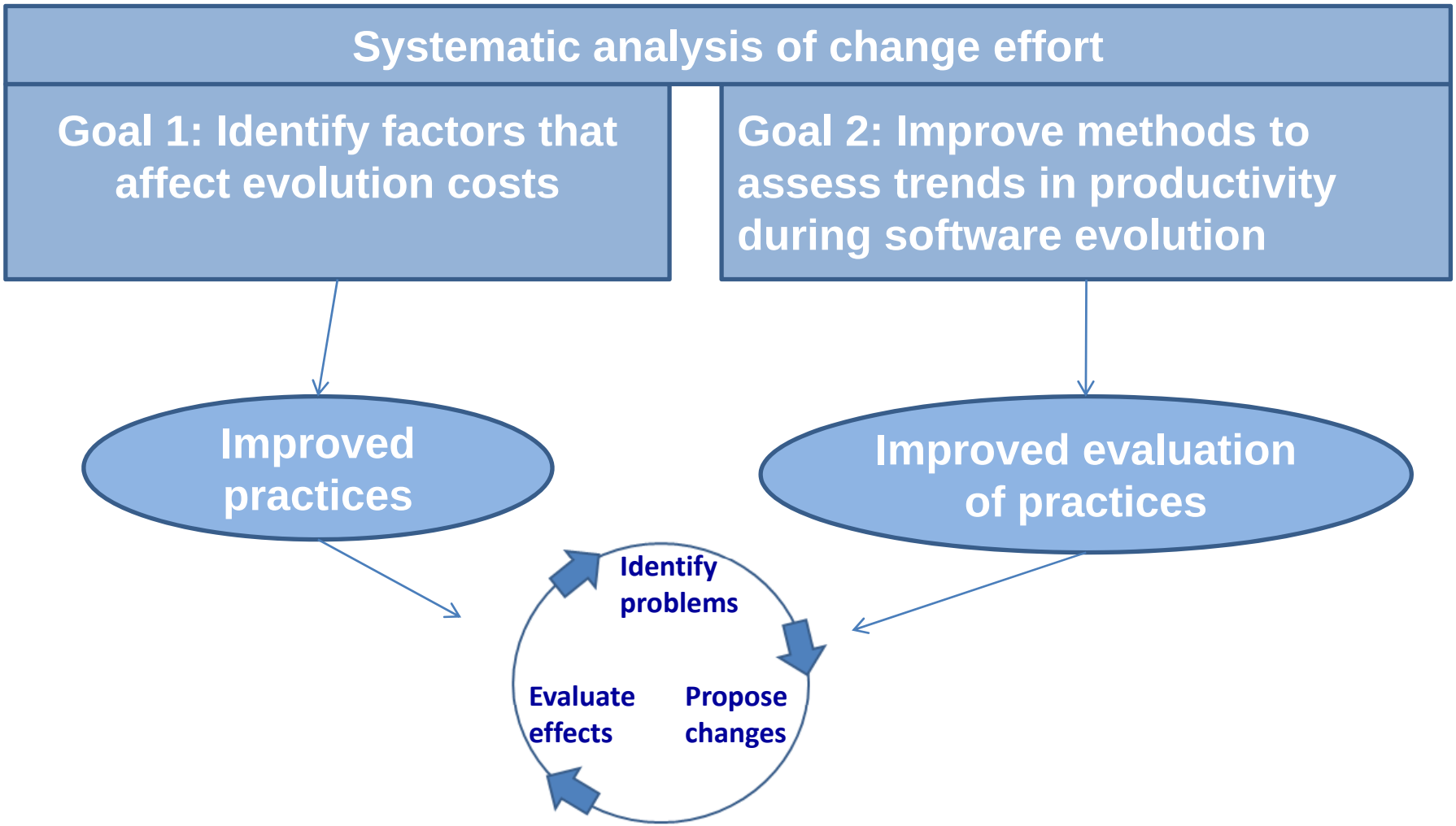


50 billion USD worth of evolution costs, annually

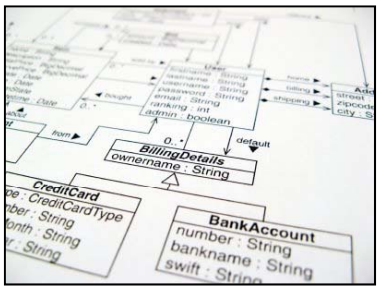
*Essence of Lehman's "first law of software evolution" (1976)

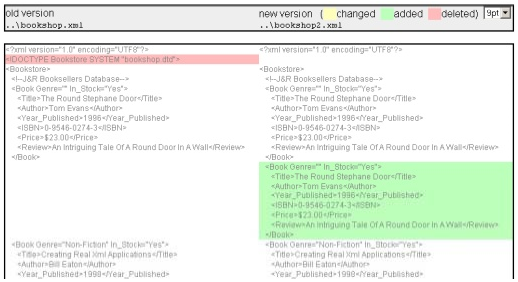
** As proposed by Swanson (1976)

The overall aim was to better understand development costs involved in making changes to software



Costs of software evolution can be assessed by analyzing drivers of change effort





People

Experience

Product

Structural attributes

Performed changes

Size, type

Practices

Collaboration

Change effort

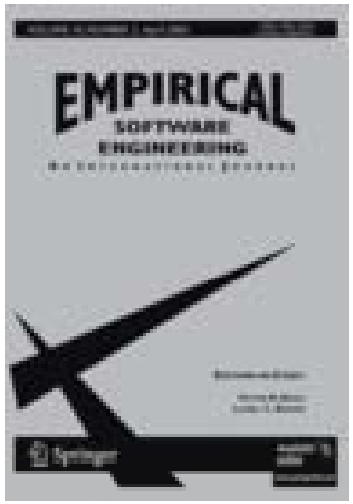
*

Software evolution cost

Systematic analysis of change effort

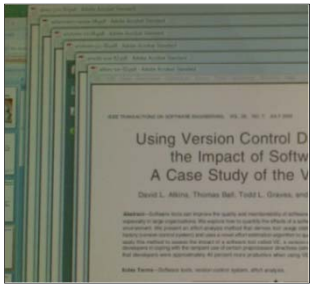
Goal 1: Identify factors that affect evolution costs

Goal 2: Improve methods to assess trends in productivity during software evolution



A framework for change-based studies was established using systematic literature procedures

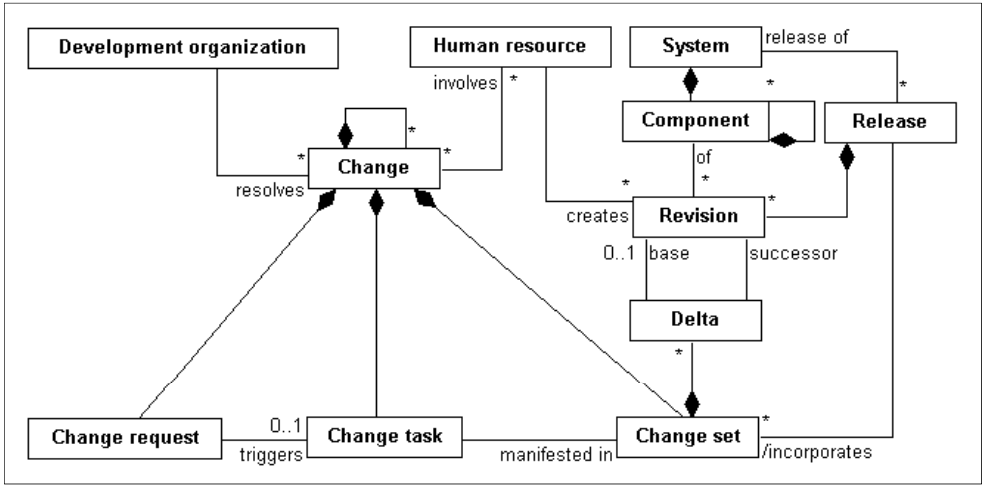
Peer reviewed change-based studies



Systematic literature review

"A rigorous methodology to ensure a fair evaluation and interpretation of all research relevant to a phenomenon"

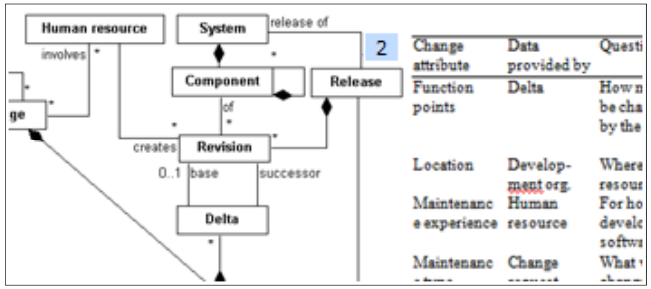
A conceptual model



Current evidence

Change attribute	Data provided by	Question asked	Typical values	Goal 1 studies
Function points	Delta	How many logical units will be changed, added or deleted by the change?	Count of changed, added and deleted units, weighted by complexity	-
Location	Develop-ment org.	Where were human resources located physically?	Distributed/not distributed	-
Maintenance experience	Human resource	For how long had the developers performed software maintenance work?	Number of years	-
Maintenance type	Change request	What was the purpose of the change?	Fix/enhance/ adapt	[21, 37, 41-47, 49-51]

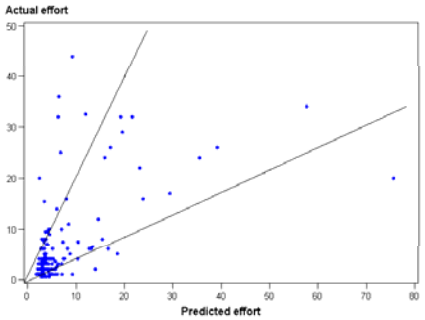
A case study investigated costs factors in two commercial software organizations



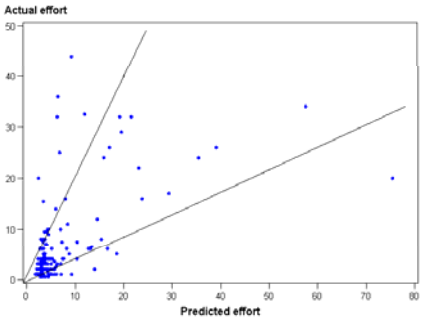
Research model and hypotheses based on the establised framework

c.jira	changespan	techspan	devspan	words	ceffort
MONA-1555	6	1	2	44	18000
MONA-2817	10	1	1	107	36000
MONA-3152	5	1	1	34	14400
MONA-3168	24	2	2	375	345600
MONA-3169	7	3	1	33	14400
MONA-3209	1	1	1	114	3600

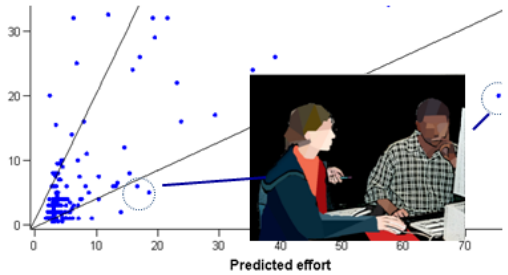
Collected 31 measures for 336 changes



Evidence-driven statistical analysis



Data-driven statistical analysis



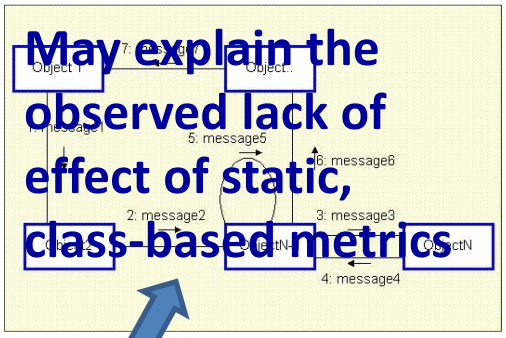
Interview analysis to explain large residuals

Developers' effort to comprehend and change dispersed code was an important cost driver

Strong correlation

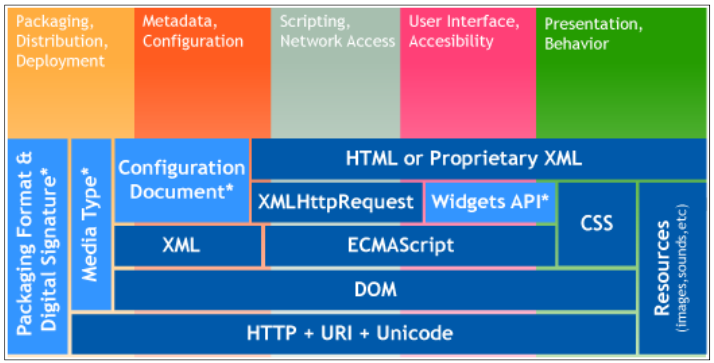
Number of components changed → ***Change effort***

Comprehension of dispersed "relevant" code



May explain the observed lack of effect of static, class-based metrics

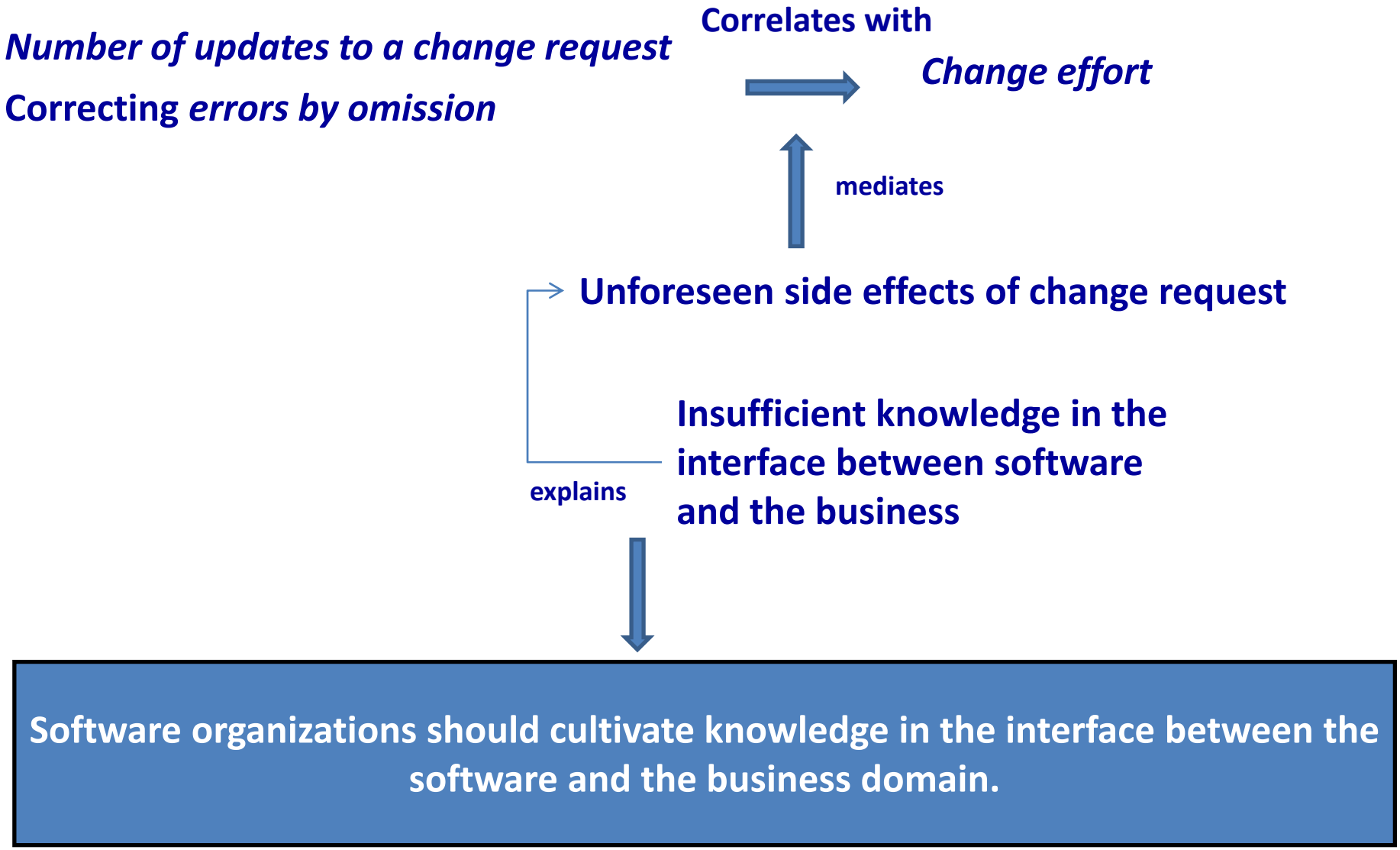
Comprehension occurred along object interactions within user scenarios, rather than architectural units



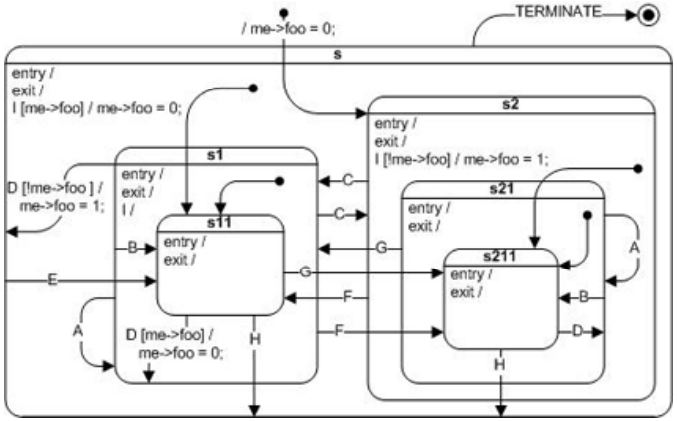
An additional effect occurred when comprehension and change spanned several technologies

Design practices and tools should recognize developers' need to comprehend functional crosscuts of the software, in particular when several technologies are involved

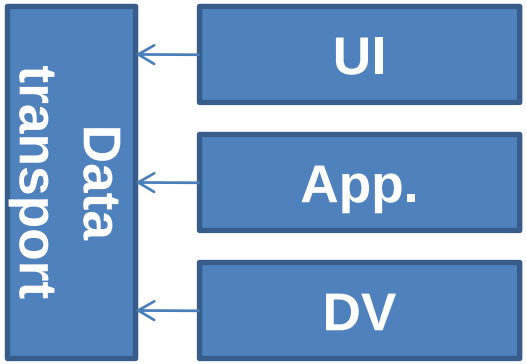
Volatility of change request was an important cost driver



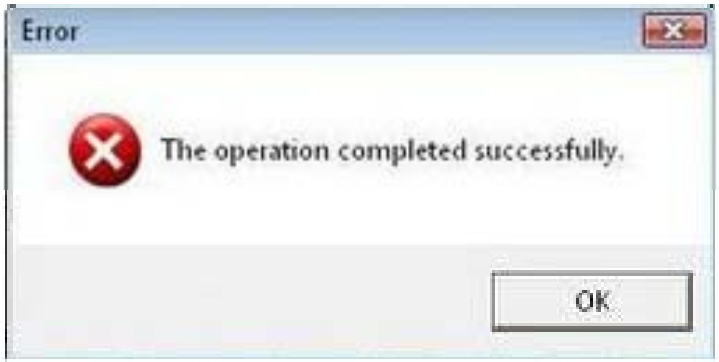
The qualitative analyses proposed a number of cost drivers not captured by statistical models



Understand a complex underlying state-model



Develop reusable mechanisms



Circumvent technology flaws

Systematic analysis of change effort

Goal 1: Identify factors that affect evolution costs

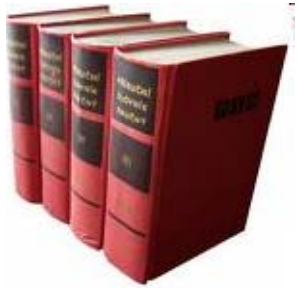
Goal 2: Improve methods to assess trends in productivity during software evolution

Practical and trustworthy measures of productivity are needed to support process improvement

$$\text{productivity} = \frac{\text{output production}}{\text{input effort}}$$

SE

$$productivity = \frac{developed\ size}{developer\ effort}$$



Measures of developed size:

- ***Lines of code*** **[1]**
- ***Developed components***
- ***Function points*** **[2]**
- ***Specification weight metrics*** **[3]**

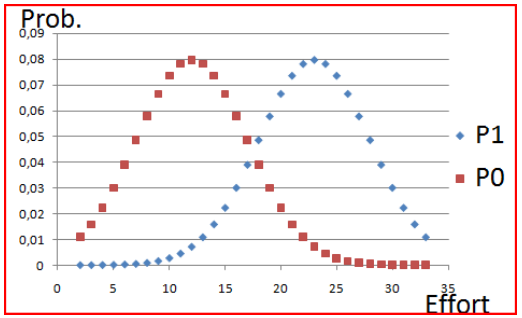
Such measures can be adopted to software evolution [4, 5, 6]

1. Fenton&Pfleeger-1997
2. Albrecht-1983
3. DeMarco m-1984
4. Ramil&Lehman – 2000
5. Maya&Abran -1996
6. Abran&Maya - 1995

Claim: Current measures are either difficult to collect or have questionable validity

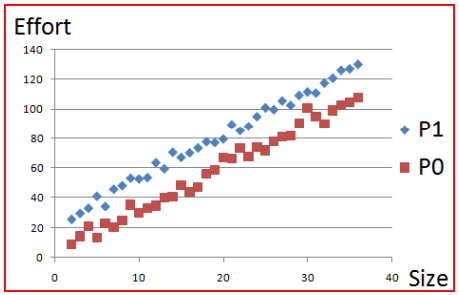
Compare the time needed to complete change tasks between two time periods: Four variants

Compare mean effort



$$ICPR_1 = \frac{mean(p1)}{mean(p0)}$$

Control for change characteristics



$$effort = \beta_1 \cdot words + \beta_2 \cdot loc + \beta_3 \cdot inP1.$$

$$ICPR_2$$

Compare actual with predicted effort

	Actual	Predicted	Act/Pred
45	4	3	1,333333
46	5	3	1,666667
47	6	4	1,5
48	...	..	...
49	7	4	1,75

$$ICPR_3 = median\{\frac{act}{pred}\}$$

Estimate the same tasks in P0 and P1

Jun 08 P1Pred	Jun 07 P0Pred	P1Pred/ P0Pred
6	7	0,85714
4	6	0,66667
5	8	0,625
...	..	...
7	8	0,875

$$ICPR_4 = median\{\frac{P1pred}{P0pred}\}$$

Insipired by:

[Trad.]

[Eick *et al.*, 2001]

[Kitchenham&Mendes, 2000]

[Arisholm&Sjøberg, 2001]

First application: Assessment showed opposite productivity trends, consistent with major project events

Performed a major restructuring effort of the software

Switched from fixed price to time and material contracts

		
$ICPR_1$	0.81	1.50
$ICPR_2$	0.90	1.50*
$ICPR_3$	0.78***	1.18
$ICPR_4$	1.00	1.33*

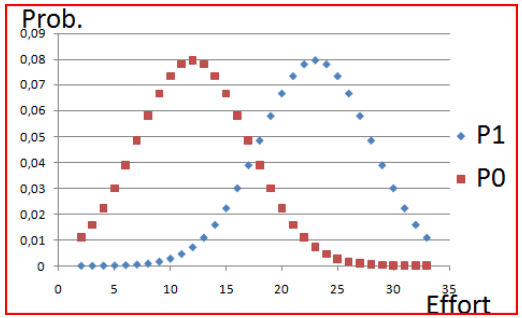
First cut evaluation: Statistical tests to determine statistical significance

Indicates higher productivity

Indicates lower productivity

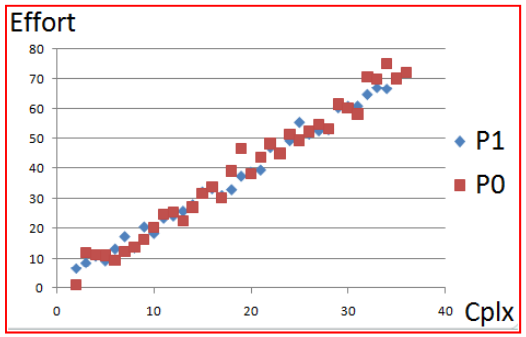
Second application:
Assessing the effect of a new estimation practice

Compare mean effort



Indication of lower productivity using P1

Control for change characteristics



No difference in productivity when controlling for differences in complexity of changes

Structured interviews: P1 helped in identifying subtasks/side effects

Surprisingly, the estimation method seemed to affect change effort, rather than estimation accuracy

Indicators help in discovering, and understanding causes for, productivity trends

In summary, the systematic change-based analysis proved effective to understand development costs during evolution

A framework for measuring and analyzing changes that combines quantitative and qualitative methods

A promising method for assessing productivity trends during software evolution

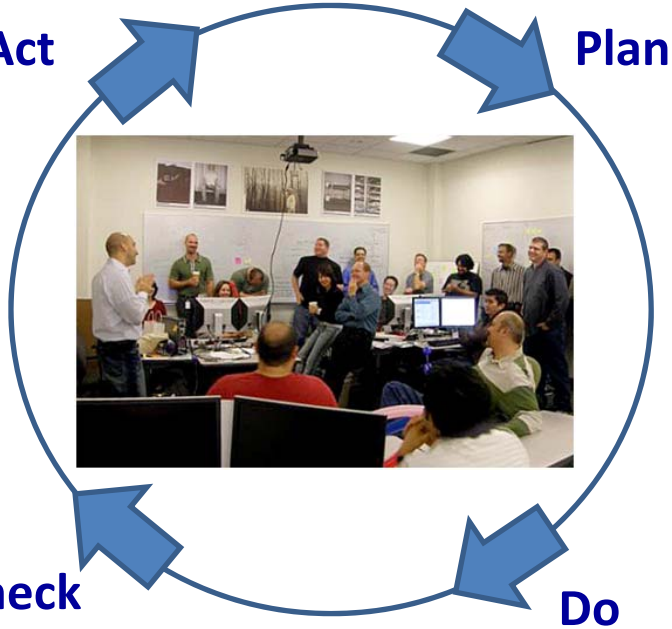
Empirical evidence and understanding of important cost drivers in software evolution

Thank you for listening

UNIVERSITETET I OSLO

[simula . research laboratory]
· by thinking constantly about it ·

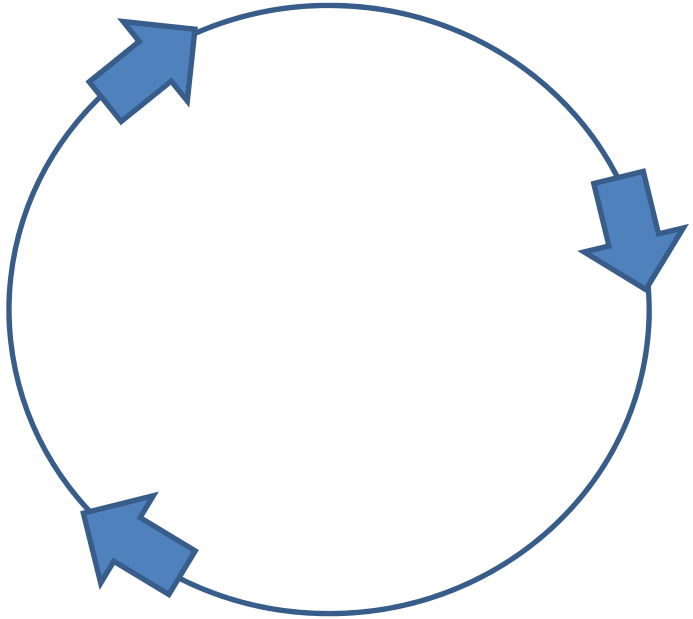
The overall aim was to better understand development costs during software evolution



Identify factors affecting costs

Improve software process and product

Assess productivity trends



Comparison of change task properties helps in validating the indicators' assumptions

Table 3. Properties of change tasks in RCN

Variable	*P0*	*P1*	p-value
chLoc (mean)	26	104	0.0004
crWords (mean)	107	88	0.89
filetypes (mean)	2.7	2.9	0.50
isCorrective (%)	38	39	0.90

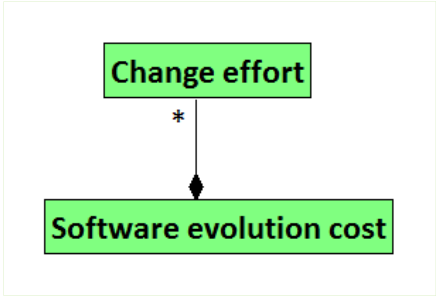
Table 4. Properties of change tasks in MT

Variable	*P0*	*P1*	p-value
addCC (mean)	8.7	44	0.06
components (mean)	3.6	7	0.09
crTracks (mean)	4.8	2.5	<0.0001
systExp (mean)	1870	2140	0.43

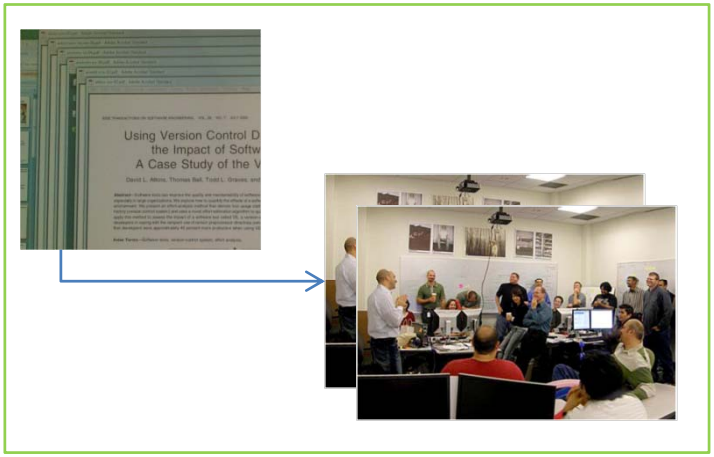
Change tasks were indeed different between the periods

Indicator controlling for the differences was justified

This presentation describes the motivation, research approach and key results



Analysis of individual changes to understand software evolution costs



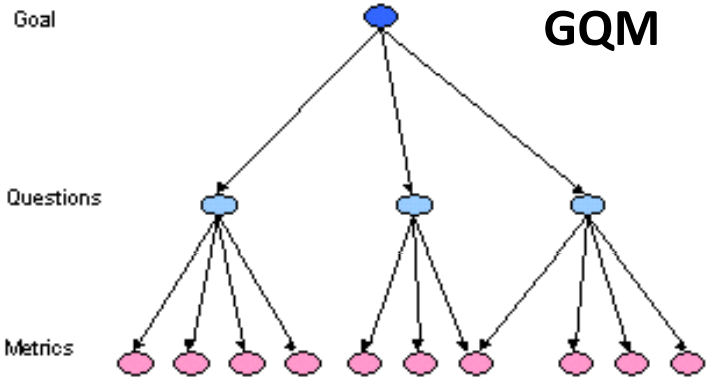
Systematic review and multiple case study as key research methods

Evidence on cost factors
Method for measuring productivity

Frameworks for improving software processes and products presume that productivity can be measured

Deming

GQM



Productity measures are essential

QIP