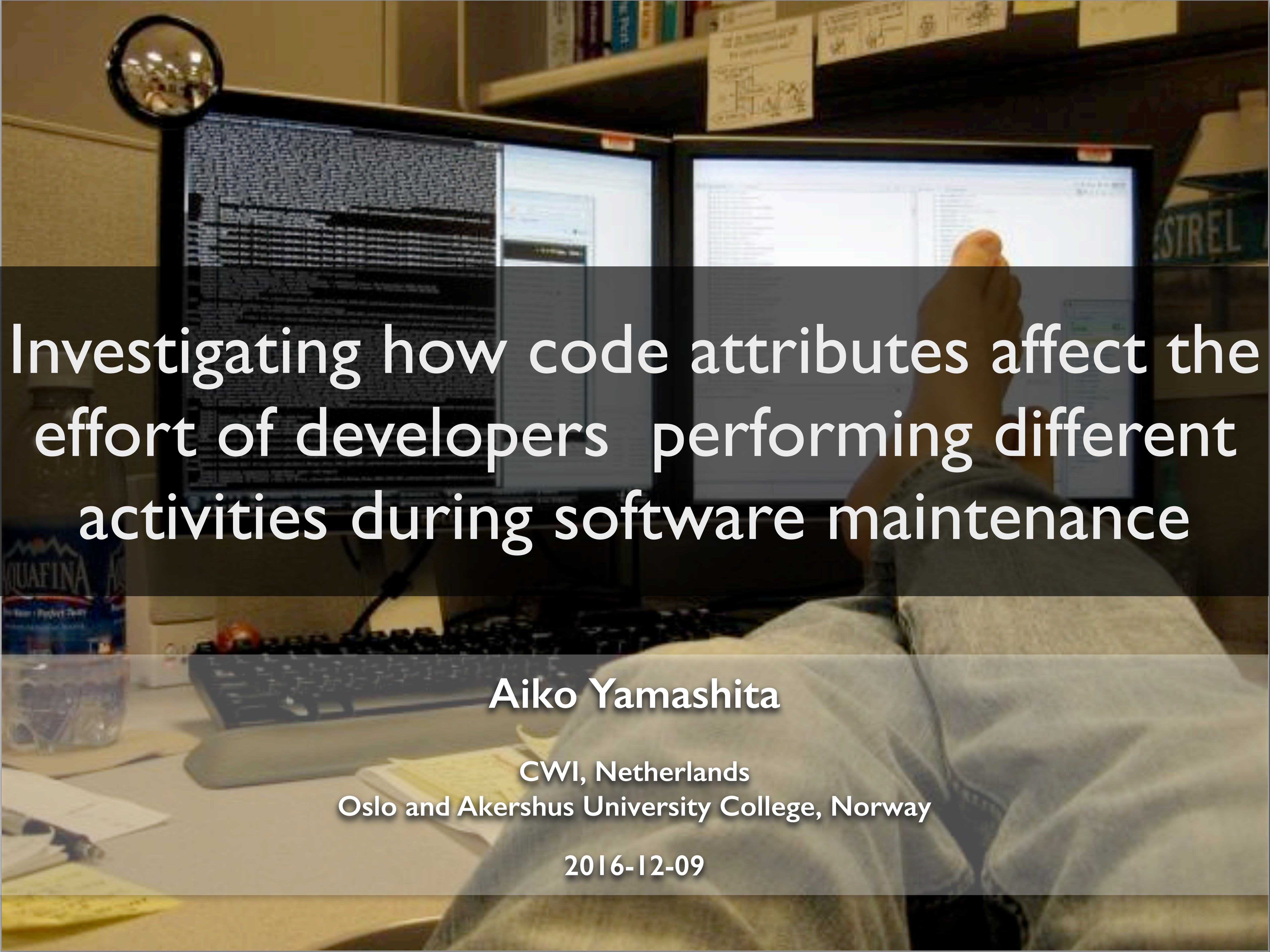


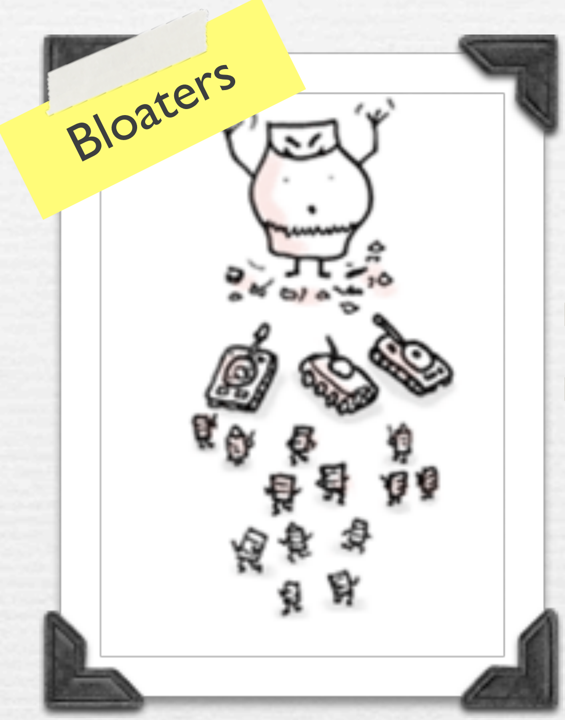
Investigating how code attributes affect the effort of developers performing different activities during software maintenance

Aiko Yamashita

CWI, Netherlands
Oslo and Akershus University College, Norway

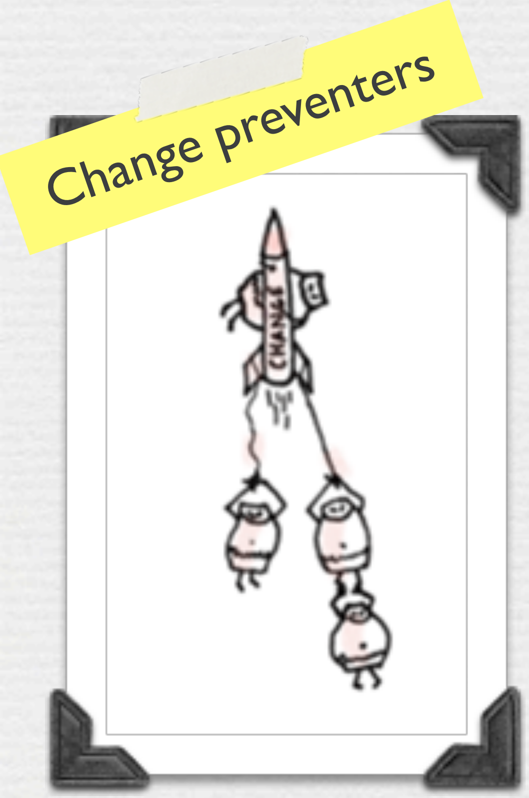
2016-12-09

STRUCTURAL PROPERTIES AFFECT SOFTWARE



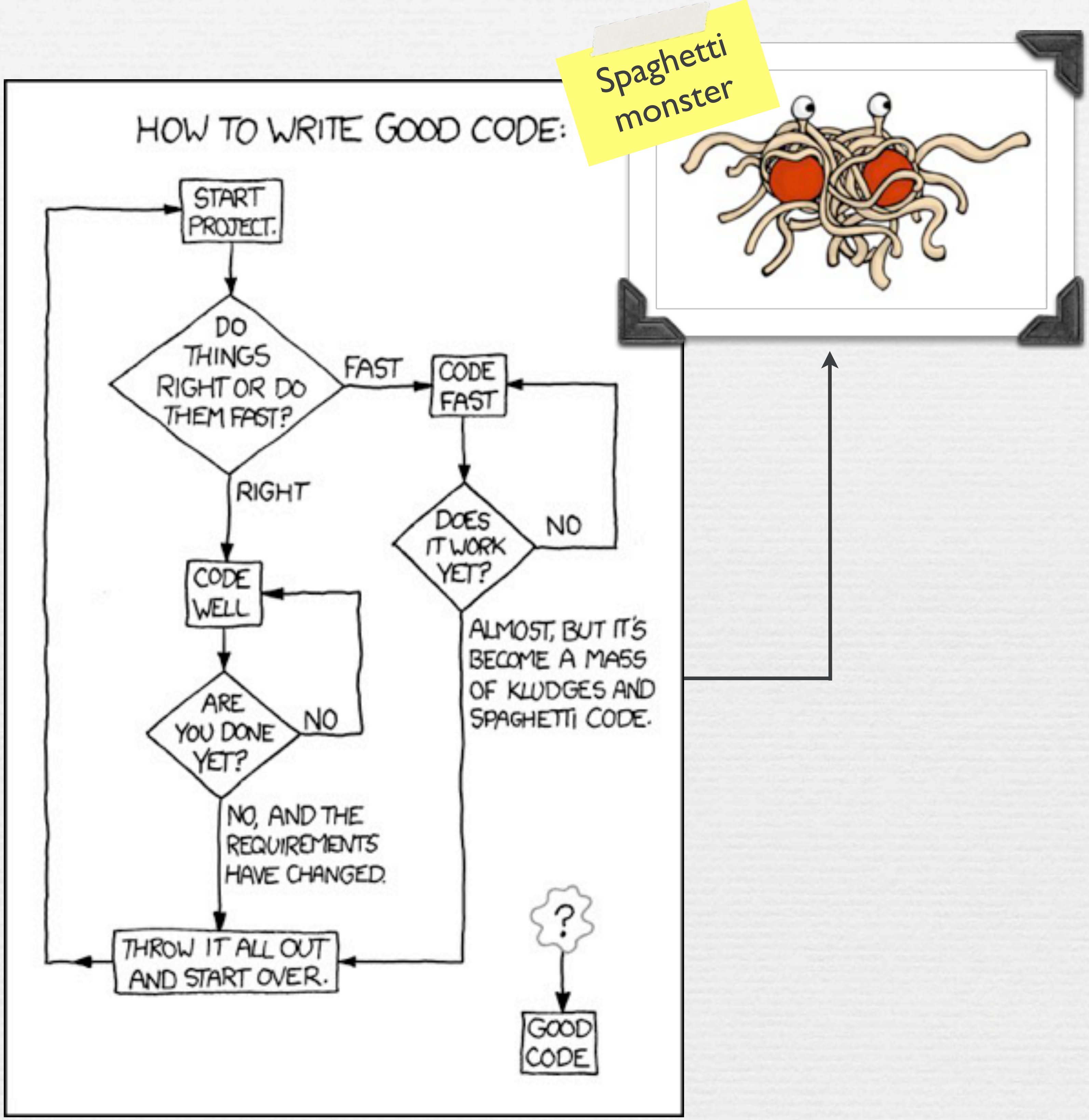
Bloaters

- Long Method
- Large Class



Change preventers

- Divergent Change
- Shotgun Surgery



Spaghetti monster

HOW TO WRITE GOOD CODE:
START PROJECT.
DO THINGS RIGHT OR DO THEM FAST?
FAST
CODE FAST
DOES IT WORK YET?
NO
ALMOST, BUT IT'S BECOME A MASS OF KLUDGES AND SPAGHETTI CODE.
RIGHT
CODE WELL
ARE YOU DONE YET?
NO
NO, AND THE REQUIREMENTS HAVE CHANGED.
THROW IT ALL OUT AND START OVER.
?
GOOD CODE

https://sourcemaking.com/refactoring/smells

https://xkcd.com/844/

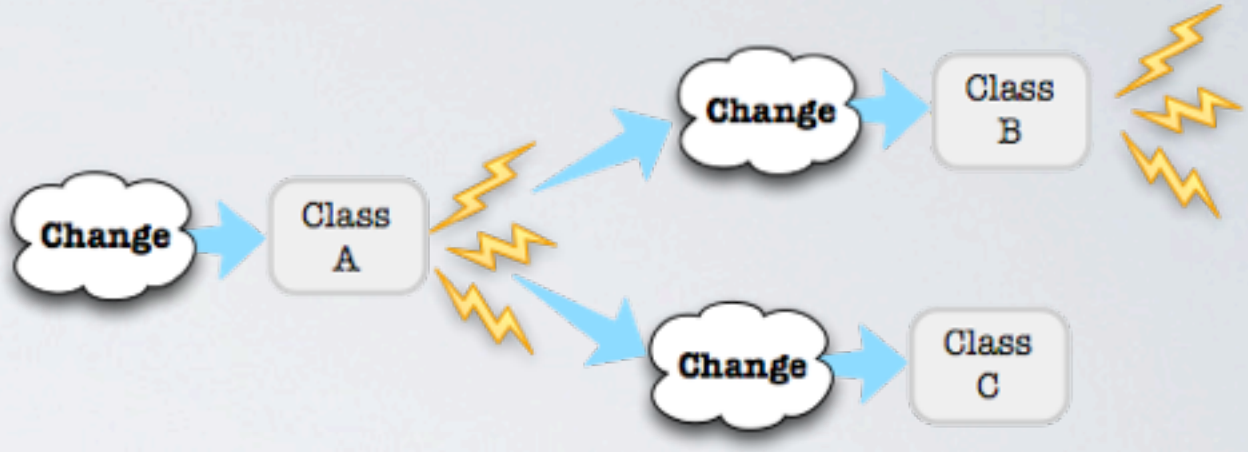
CODE SMELLS: ONE FORM OF STRUCTURAL ANOMALY

A hint about suboptimal implementation choices that can affect negatively future maintenance and evolution.

EXAMPLE OF CODE SMELL AND REFACTORING

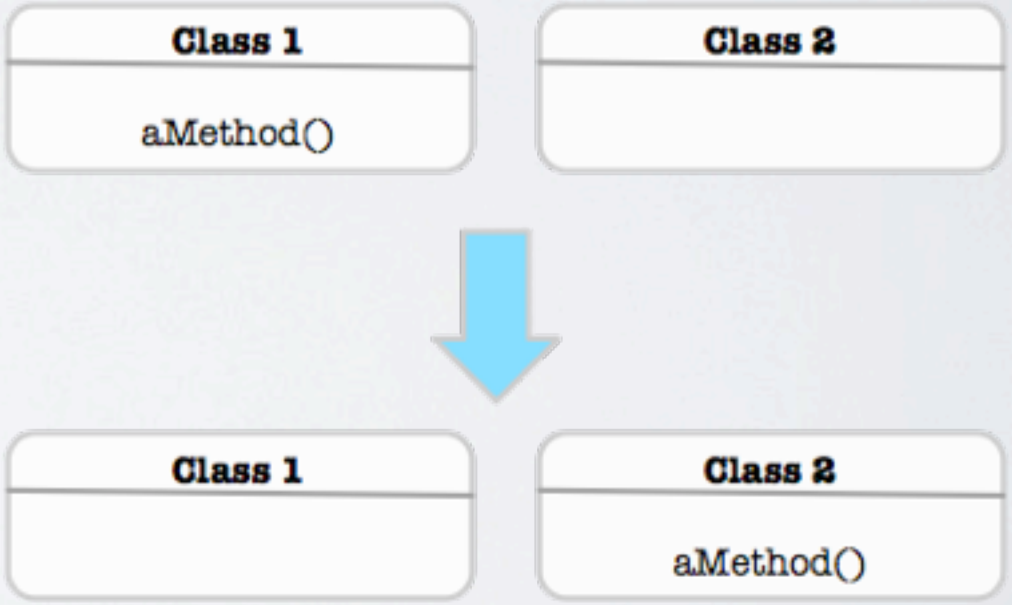
Shotgun Surgery

A change leads to another change, to another, to another..



Move method refactoring

Reduce the coupling between components



STATE OF ART ON CODE SMELLS

Empirical studies

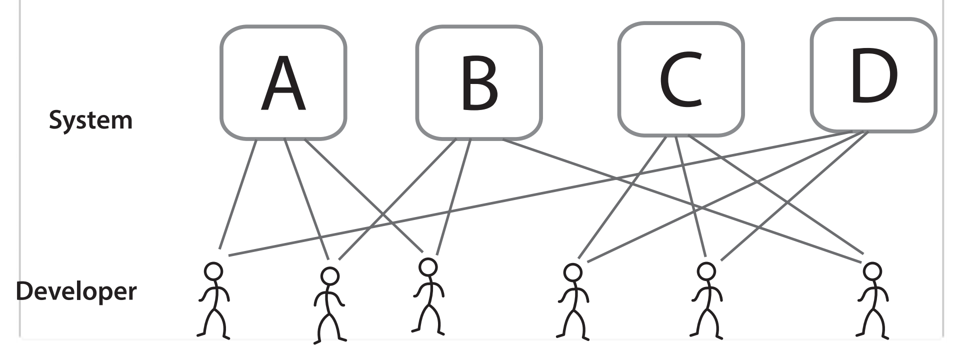
- Code smells have deterring effects on the introduction of defects
    - Monden (2002), Li &Shatnawi (2007), Kapser (2006), Juergens (2009), Rahman (2011)
- Larger maintenance effort
    - Deligiannis (2004), Abbes (2011)
- Larger and more frequent changes in the code
    - Olbrich (2009), Khomh (2009)
- The overall capacity of code smell analysis to explain or predict maintenance problems or maintenance effort is rather modest
    - Yamashita (2012), Sjøberg (2013)

Previous work: Multiple, controlled case study (Yamashita 2012, Sjøberg et al., 2013)

Context

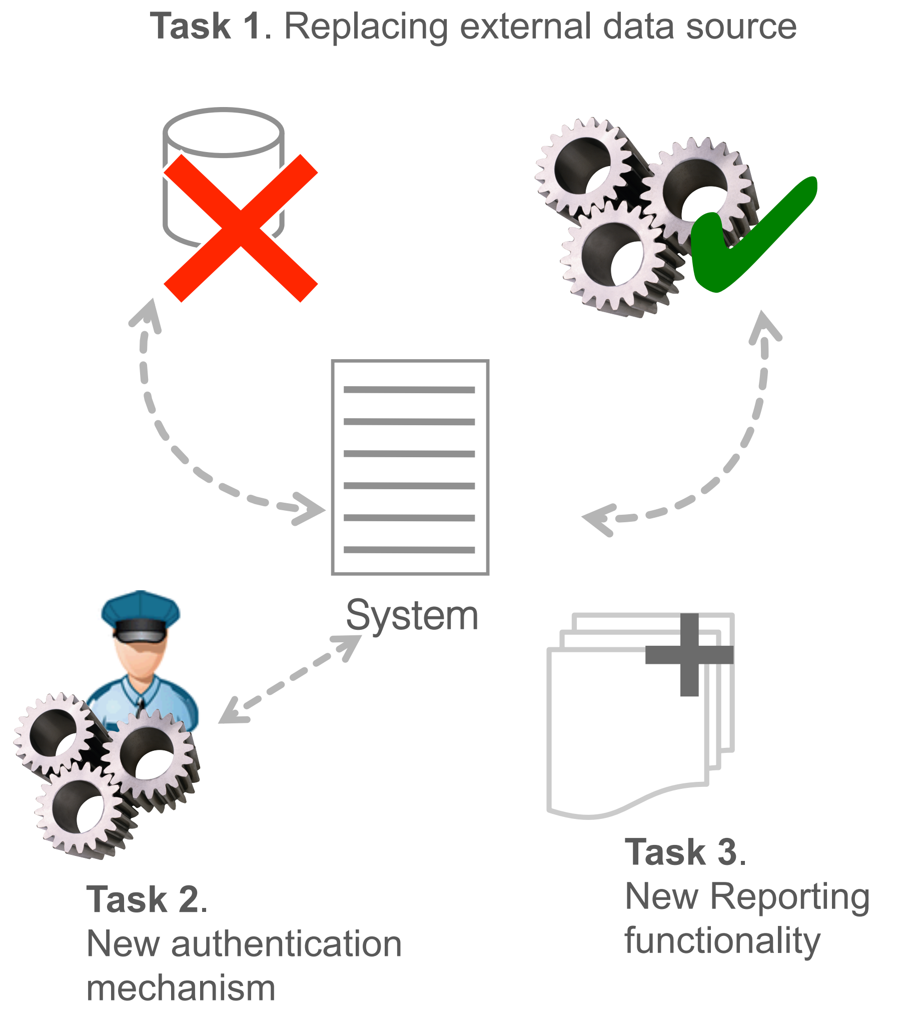
- 4 Java Applications
- Same functionality
- Different design/code
- Size: 7KLOC to 14KLOC

Study Design



		Developer					
		1	2	3	4	5	6
Round	1	A	B	C	D	C	A
	2	D	A	D	C	B	B

Maintenance Tasks



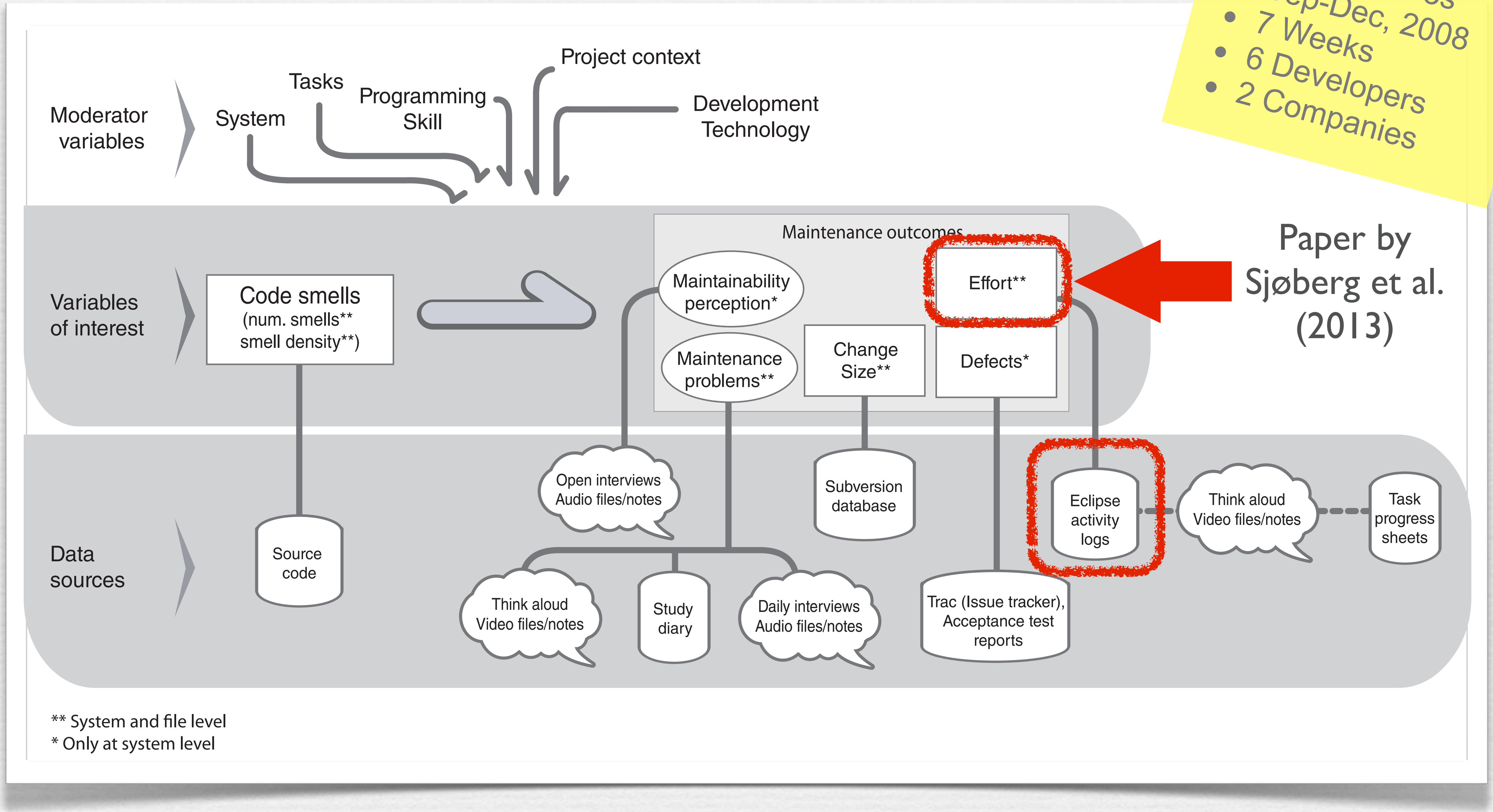
Task 1. Replacing external data source

Task 2.
New authentication mechanism

Task 3.
New Reporting functionality

Previous work: Multiple, controlled case study (Yamashita 2012, Sjøberg et al., 2013)

- 50,000 Euros
- Sep-Dec, 2008
- 7 Weeks
- 6 Developers
- 2 Companies



** System and file level
* Only at system level

Paper by Sjøberg et al. (2013)

Analysis done in previous work (Sjøberg et al., 2013)

Explanatory model for Effort

Dependent variable: Effort (time)

Independent variables: 12 smells

Control variables:

- File size (LOC)
- Number of revisions on a file
- System
- Developer
- Round

Analysis: Multiple Linear Regression

Results

- A model that includes file size and number of changes and code smells displayed a fit of R2 = 0.58
- Removing the smells from that model did not decrease the fit!! (R2 = 0.58)
- Only smell that remained significant was Refused Bequest, which registered a decrease in effort ($\alpha < 0.01$)
- File size and number of changes remain the most significant predictors of effort ($\alpha < 0.001$)

Analysis done in previous work (Sjøberg et al., 2013)

Explanatory model for Effort

Dependent variable: Effort (time)

Independent variables: 12 smells

Control variables:

- File size (LOC)
- Number of revisions on a file
- System
- Developer
- Round

Analysis: Multiple Linear Regression

Results

- A model that includes file size and number of changes and code smells displayed a fit of R2 = 0.58
- [illegible] del d[illegible]
- [illegible] nt was Refused Bequest, which registered a decrease in effort ($\alpha < 0.01$)
- File size and number of changes remain the most significant predictors of effort ($\alpha < 0.001$)

Code smells are not better at explaining sheer-effort at file level, than size and number of revisions.

Previous analysis considers sheer effort

Reading

Programming (code-related) activities during Maintenance

Navigating

Searching

Others

Editing

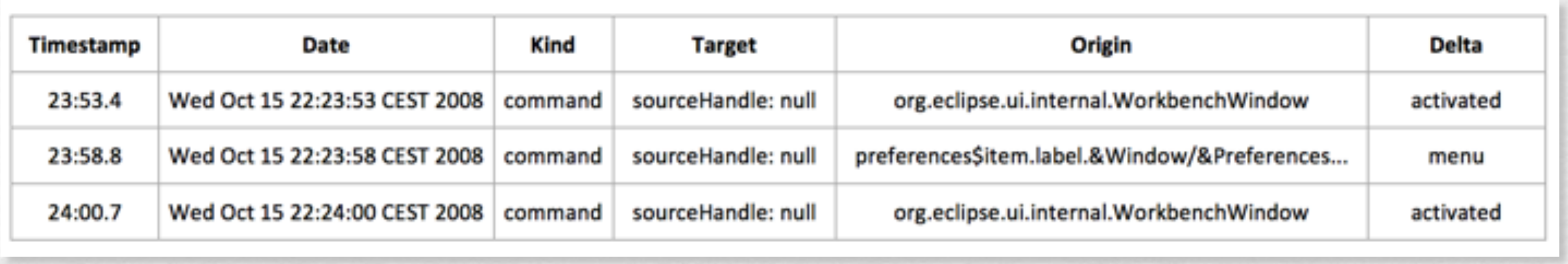
Procedure for extracting activity effort

MimEc: Intelligent User Notification of Faults in the Eclipse IDE

Lucas M. Layman, Laurie A. Williams, Robert St. Amant
Department of Computer Science
North Carolina State University
Campus Box 8206
Raleigh, NC 27695

- Selection of artifacts in the package explorer
- Selection of Java elements in the editor window
- Selecting Java elements in the file outline
- Editing source files (Java files)
- Scrolling the source code window
- Switching between open files
- Running Eclipse “commands” (copy, paste, go to line)

Activity logs



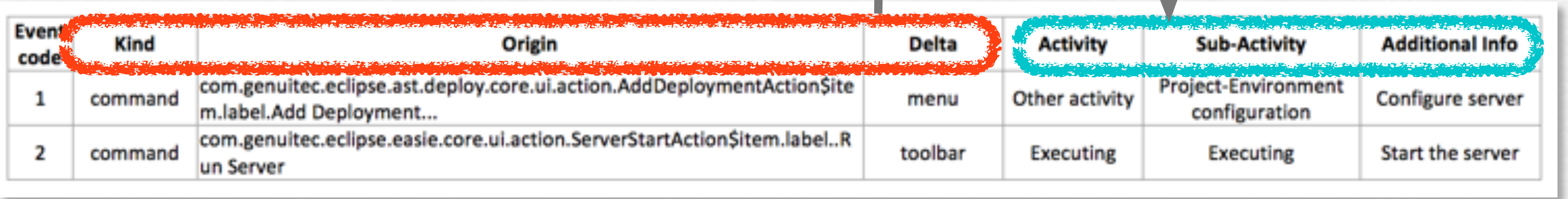
Timestamp	Date	Kind	Target	Origin	Delta
23:53.4	Wed Oct 15 22:23:53 CEST 2008	command	sourceHandle: null	org.eclipse.ui.internal.WorkbenchWindow	activated
23:58.8	Wed Oct 15 22:23:58 CEST 2008	command	sourceHandle: null	preferences$item.label.&Window/&Preferences...	menu
24:00.7	Wed Oct 15 22:24:00 CEST 2008	command	sourceHandle: null	org.eclipse.ui.internal.WorkbenchWindow	activated

Procedure for extracting activity effort

Eclipse activity logs

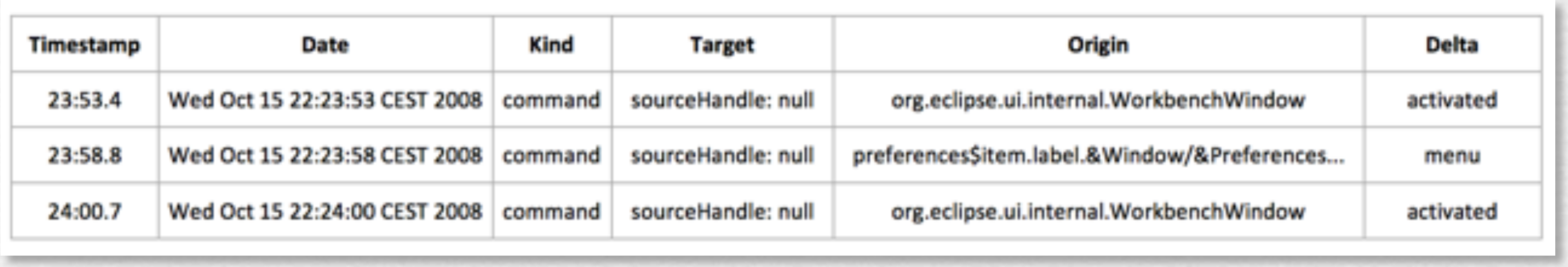
Think aloud Video files/notes

Annotation schema



Event code	Kind	Origin	Delta	Activity	Sub-Activity	Additional Info
1	command	com.genuitec.eclipse.ast.deploy.core.ui.action.AddDeploymentAction$item.label.Add Deployment...	menu	Other activity	Project-Environment configuration	Configure server
2	command	com.genuitec.eclipse.easie.core.ui.action.ServerStartAction$item.label..Run Server	toolbar	Executing	Executing	Start the server

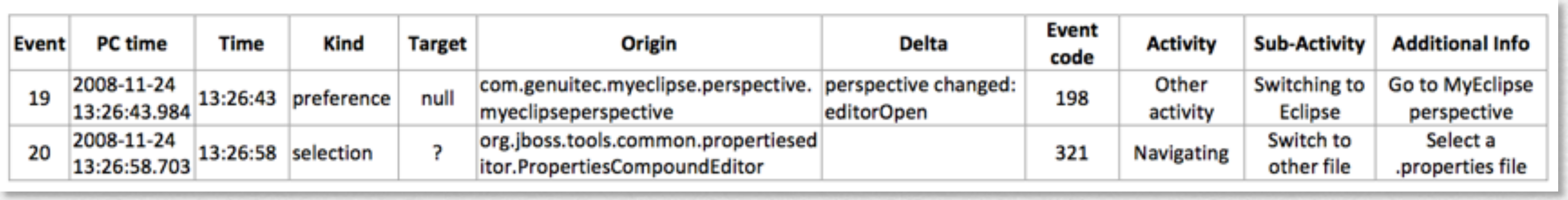
Activity logs



Timestamp	Date	Kind	Target	Origin	Delta
23:53.4	Wed Oct 15 22:23:53 CEST 2008	command	sourceHandle: null	org.eclipse.ui.internal.WorkbenchWindow	activated
23:58.8	Wed Oct 15 22:23:58 CEST 2008	command	sourceHandle: null	preferences$item.label.&Window/&Preferences...	menu
24:00.7	Wed Oct 15 22:24:00 CEST 2008	command	sourceHandle: null	org.eclipse.ui.internal.WorkbenchWindow	activated

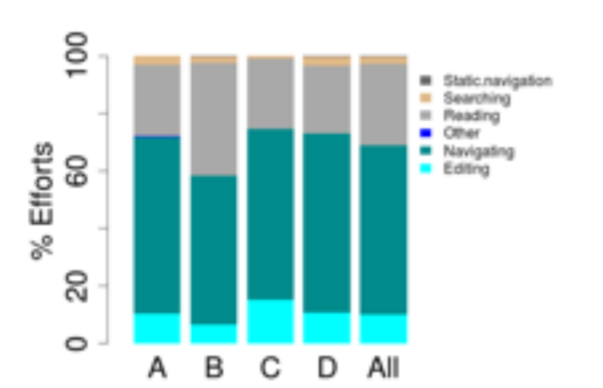
Automated annotation

Annotated activity logs



Event	PC time	Time	Kind	Target	Origin	Delta	Event code	Activity	Sub-Activity	Additional Info
19	2008-11-24 13:26:43.984	13:26:43	preference	null	com.genuitec.myeclipse.perspective.myeclipseperspective	perspective changed: editorOpen	198	Other activity	Switching to Eclipse	Go to MyEclipse perspective
20	2008-11-24 13:26:58.703	13:26:58	selection	?	org.jboss.tools.common.propertieseditor.PropertiesCompoundEditor		321	Navigating	Switch to other file	Select a .properties file

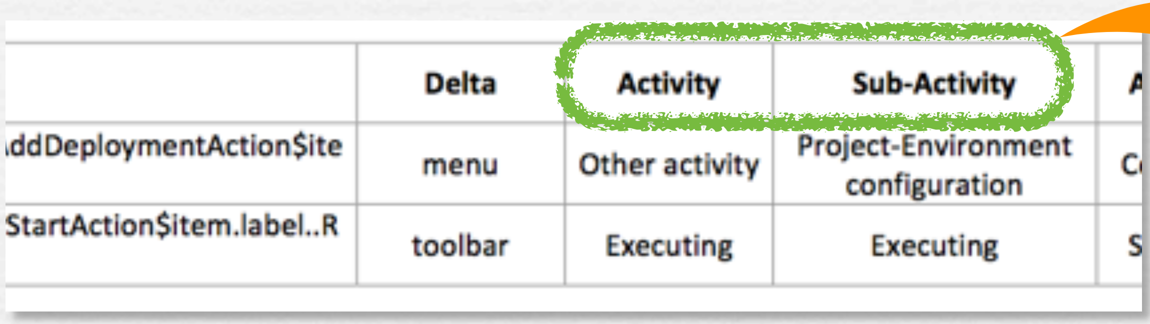
Truncate consecutive events with same activity and calculate elapsing time



Effort per activity

Eclipse activity logs

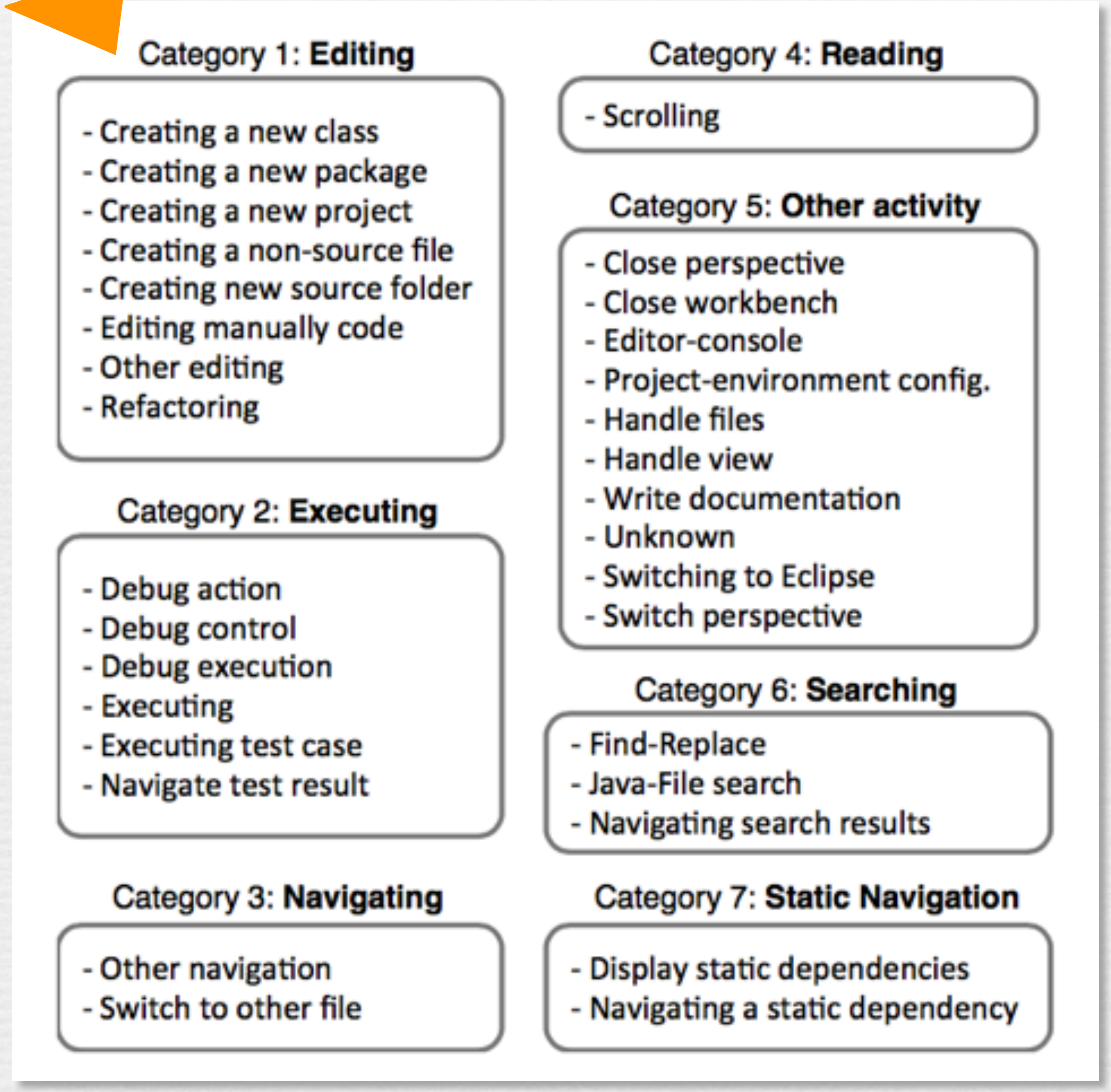
Procedure for extracting activity effort



	Delta	Activity	Sub-Activity	A
ddDeploymentAction$ite	menu	Other activity	Project-Environment configuration	C
StartAction$item.label..R	toolbar	Executing	Executing	S

Categorization of developers' activities and sub-activities was adapted from previous work (Layman 2008)

Categorization



Category 1: **Editing**

- Creating a new class
- Creating a new package
- Creating a new project
- Creating a non-source file
- Creating new source folder
- Editing manually code
- Other editing
- Refactoring

Category 2: **Executing**

- Debug action
- Debug control
- Debug execution
- Executing
- Executing test case
- Navigate test result

Category 3: **Navigating**

- Other navigation
- Switch to other file

Category 4: **Reading**

- Scrolling

Category 5: **Other activity**

- Close perspective
- Close workbench
- Editor-console
- Project-environment config.
- Handle files
- Handle view
- Write documentation
- Unknown
- Switching to Eclipse
- Switch perspective

Category 6: **Searching**

- Find-Replace
- Java-File search
- Navigating search results

Category 7: **Static Navigation**

- Display static dependencies
- Navigating a static dependency

Analysis performed

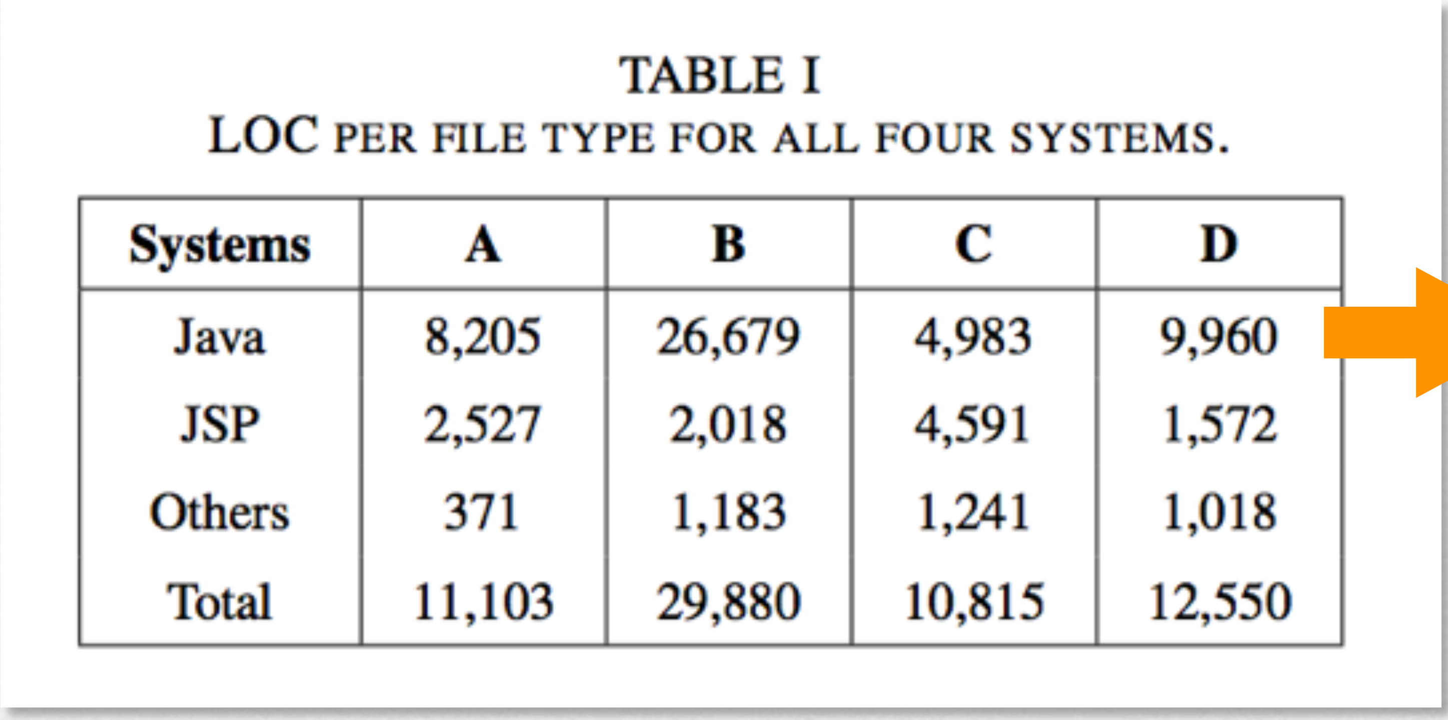
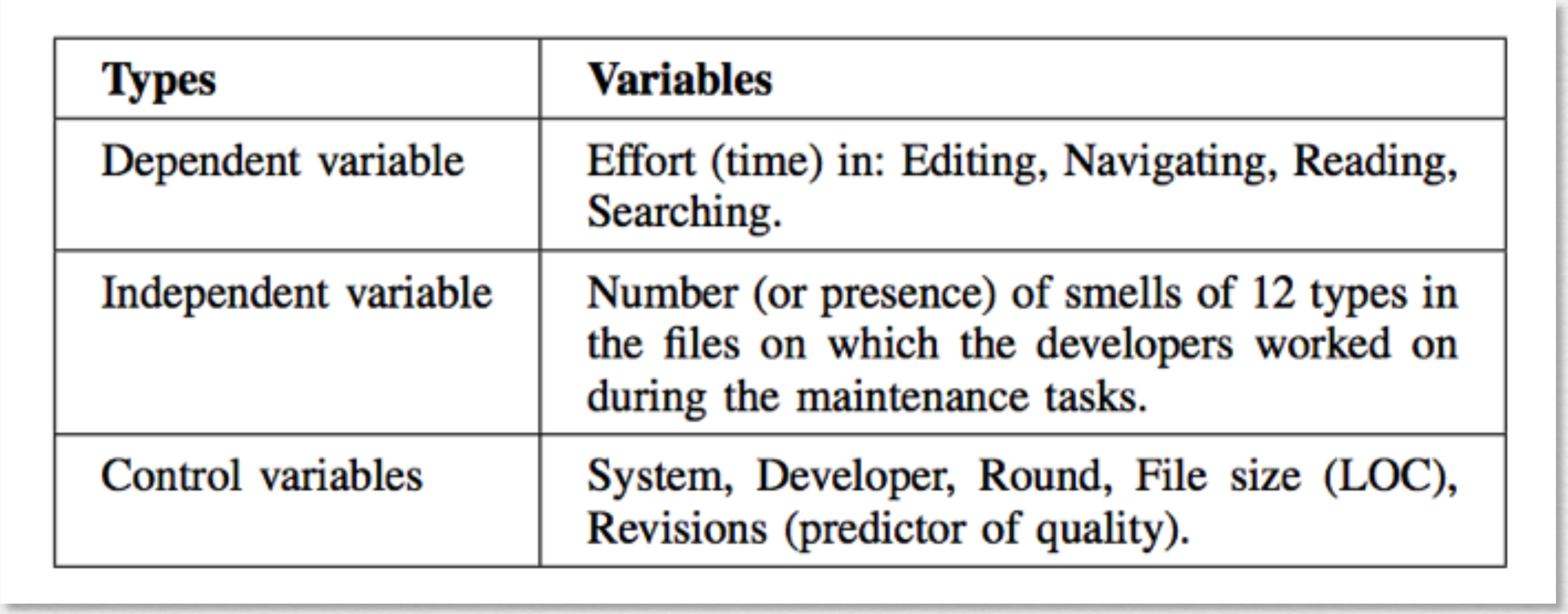


TABLE I
LOC PER FILE TYPE FOR ALL FOUR SYSTEMS.

Systems	A	B	C	D
Java	8,205	26,679	4,983	9,960
JSP	2,527	2,018	4,591	1,572
Others	371	1,183	1,241	1,018
Total	11,103	29,880	10,815	12,550

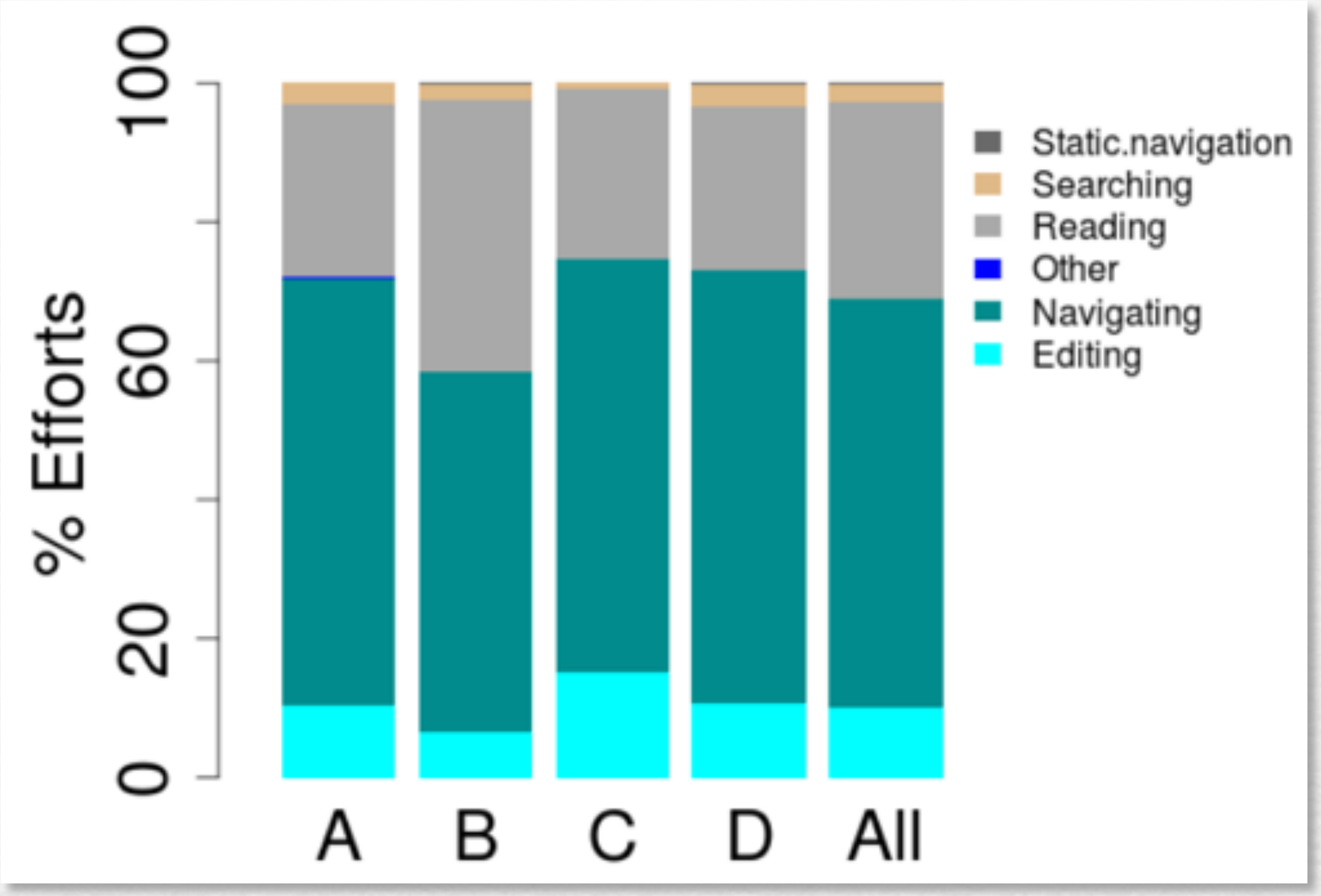
Only Java files considered

Multiple linear regression
- Forward stepwise -



Types	Variables
Dependent variable	Effort (time) in: Editing, Navigating, Reading, Searching.
Independent variable	Number (or presence) of smells of 12 types in the files on which the developers worked on during the maintenance tasks.
Control variables	System, Developer, Round, File size (LOC), Revisions (predictor of quality).

Results: Distribution of activity effort



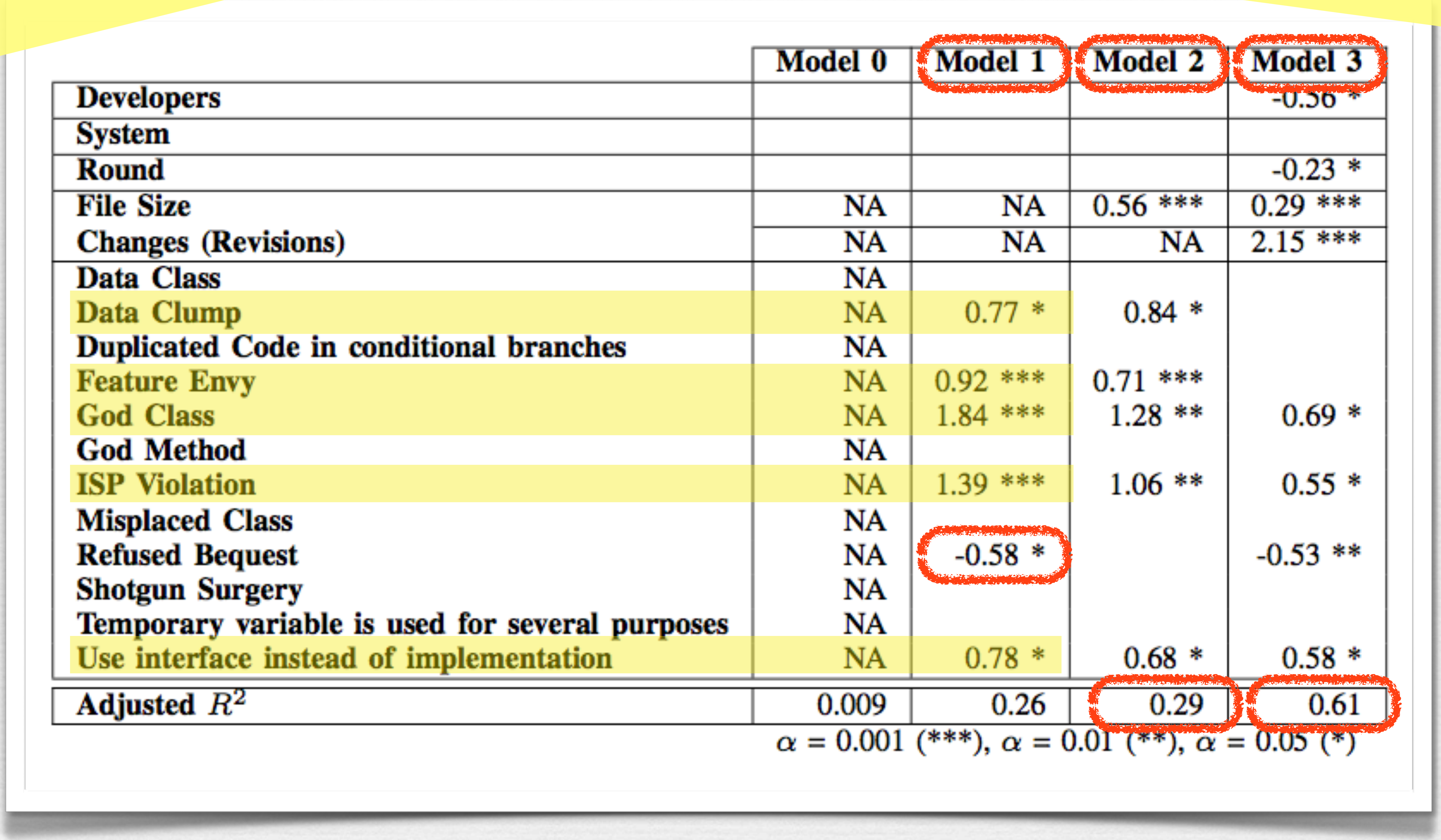
- ➡ Mostly performed activities: Navigating (58.72%), Reading (28.27%), Editing (10.18%) and searching (2.47%)
- ➡ Distribution is consistent with Ko et al. 2006 (top four)
- ➡ Reading as most consuming activity in Ko et al. 2006.
  - ▸ Definition of event/action belonging to an activity

For our analysis, we only consider:
Editing, Navigating, Searching and Reading

Model 0 + file size
R2 = 0.11

Results: Editing Effort

Model 3 - smells
R2 = 0.59



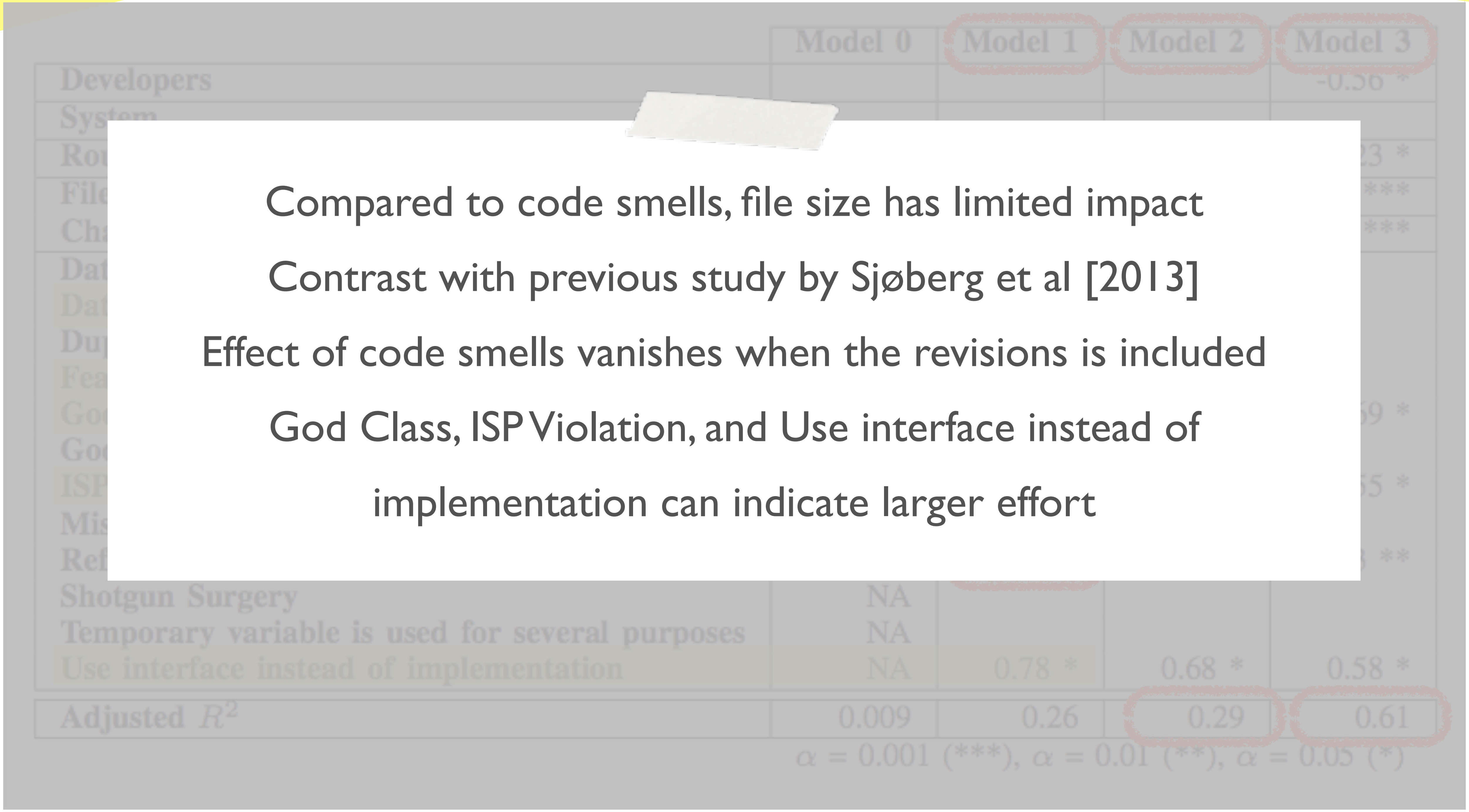
	Model 0	Model 1	Model 2	Model 3
Developers				-0.56 *
System				
Round				-0.23 *
File Size	NA	NA	0.56 ***	0.29 ***
Changes (Revisions)	NA	NA	NA	2.15 ***
Data Class	NA			
Data Clump	NA	0.77 *	0.84 *	
Duplicated Code in conditional branches	NA			
Feature Envy	NA	0.92 ***	0.71 ***	
God Class	NA	1.84 ***	1.28 **	0.69 *
God Method	NA			
ISP Violation	NA	1.39 ***	1.06 **	0.55 *
Misplaced Class	NA			
Refused Bequest	NA	-0.58 *		-0.53 **
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			
Use interface instead of implementation	NA	0.78 *	0.68 *	0.58 *
Adjusted R^2	0.009	0.26	0.29	0.61

$\alpha = 0.001$ (***), $\alpha = 0.01$ (**), $\alpha = 0.05$ (*)

Model 0 + file size
R2 = 0.11

Results: Editing Effort

Model 3 - smells
R2 = 0.59



	Model 0	Model 1	Model 2	Model 3
Developers				-0.56 *
Sys				
Ro				23 *
Fil				***
Ch				***
Dat				
Dat				
Du				
Fea				
Go				59 *
Go				
ISP				55 *
Mis				
Ref				3 **
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			
Use interface instead of implementation	NA	0.78 *	0.68 *	0.58 *
Adjusted R^2	0.009	0.26	0.29	0.61

$\alpha = 0.001$ (***), $\alpha = 0.01$ (**), $\alpha = 0.05$ (*)

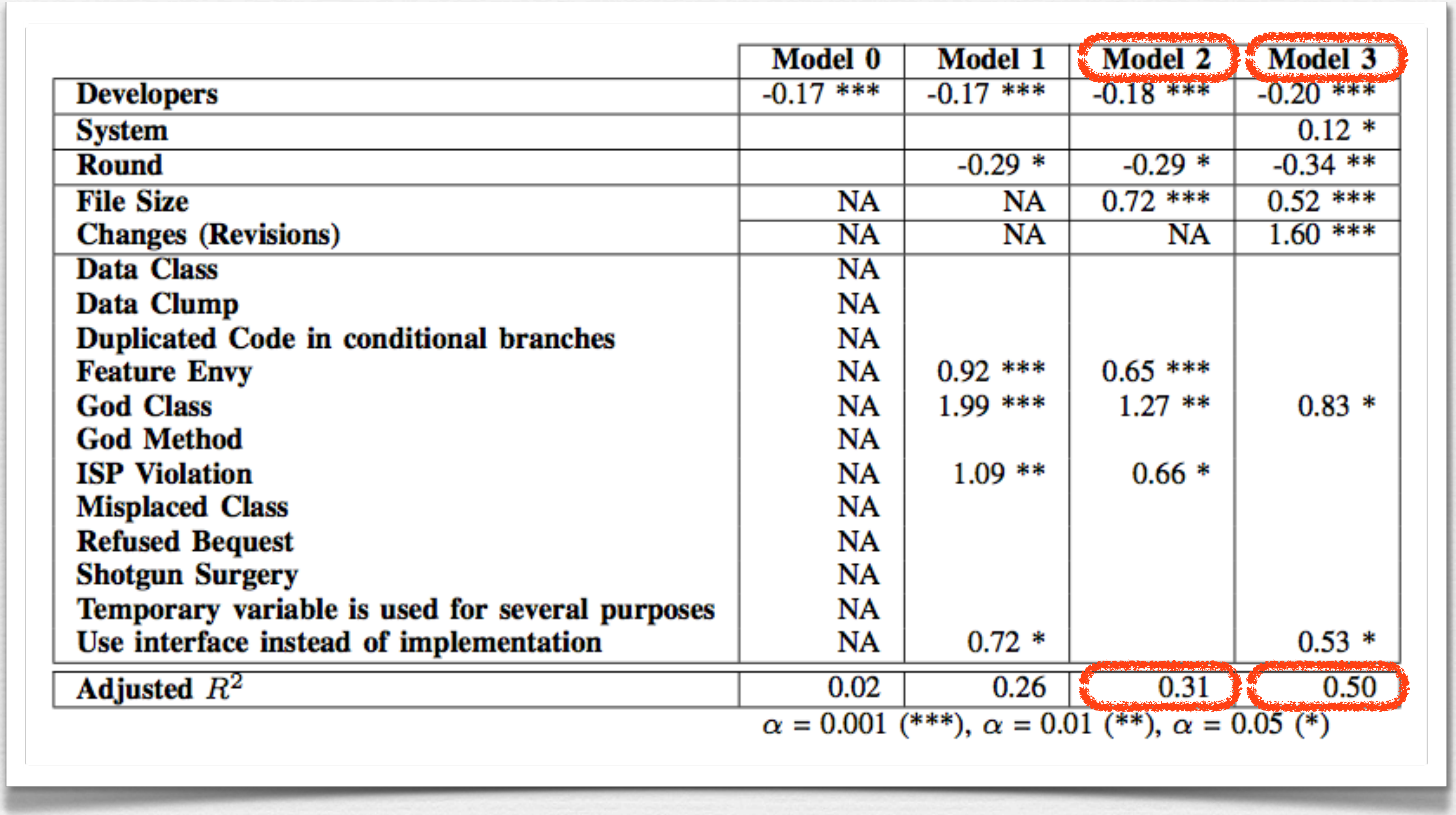
Compared to code smells, file size has limited impact

Contrast with previous study by Sjøberg et al [2013]

Effect of code smells vanishes when the revisions is included

God Class, ISP Violation, and Use interface instead of implementation can indicate larger effort

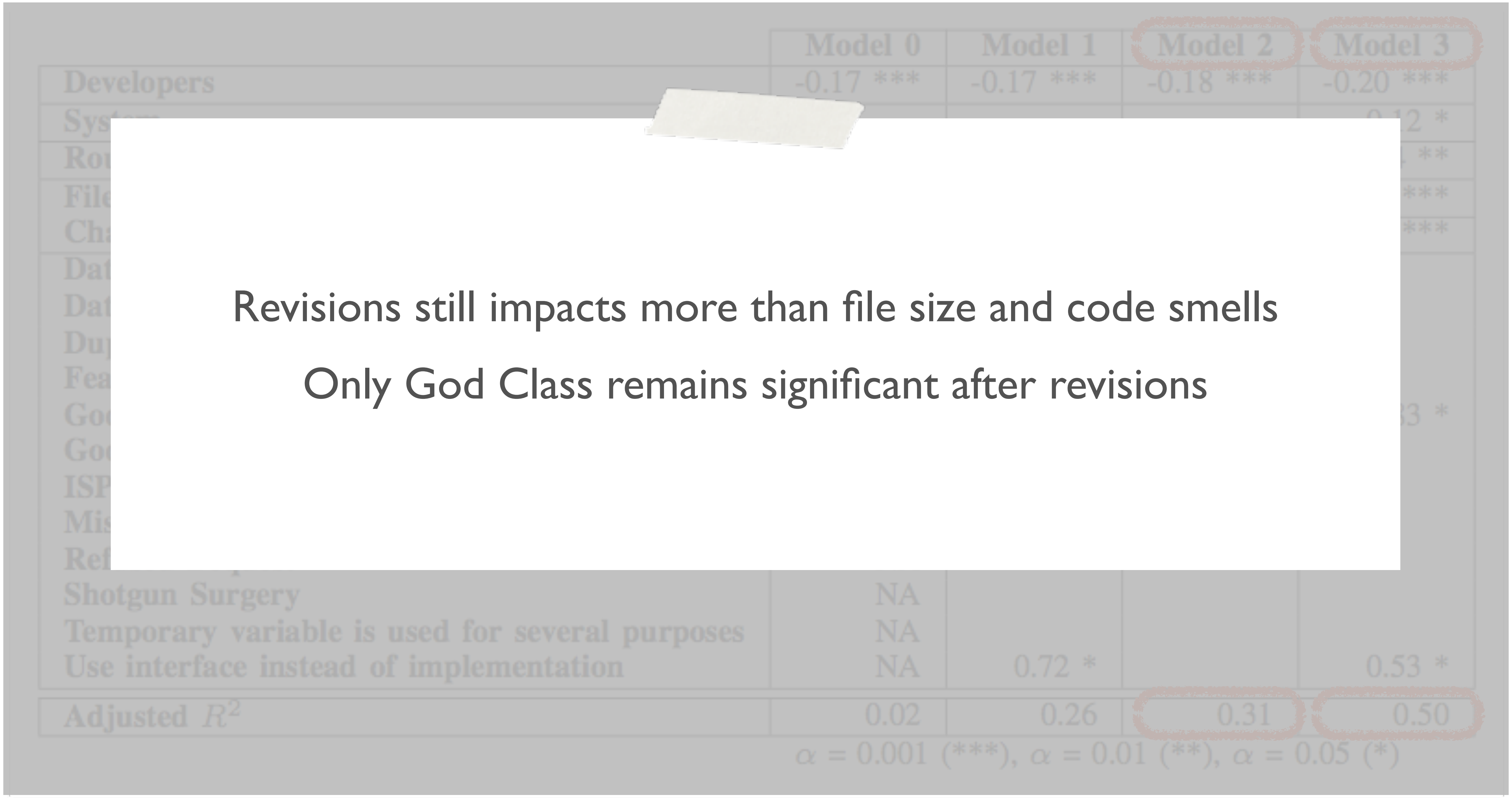
Results: Navigating Effort



	Model 0	Model 1	Model 2	Model 3
Developers	-0.17 ***	-0.17 ***	-0.18 ***	-0.20 ***
System				0.12 *
Round		-0.29 *	-0.29 *	-0.34 **
File Size	NA	NA	0.72 ***	0.52 ***
Changes (Revisions)	NA	NA	NA	1.60 ***
Data Class	NA			
Data Clump	NA			
Duplicated Code in conditional branches	NA			
Feature Envy	NA	0.92 ***	0.65 ***	
God Class	NA	1.99 ***	1.27 **	0.83 *
God Method	NA			
ISP Violation	NA	1.09 **	0.66 *	
Misplaced Class	NA			
Refused Bequest	NA			
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			
Use interface instead of implementation	NA	0.72 *		0.53 *
Adjusted R^2	0.02	0.26	0.31	0.50

α = 0.001 (***), α = 0.01 (**), α = 0.05 (*)

Results: Navigating Effort



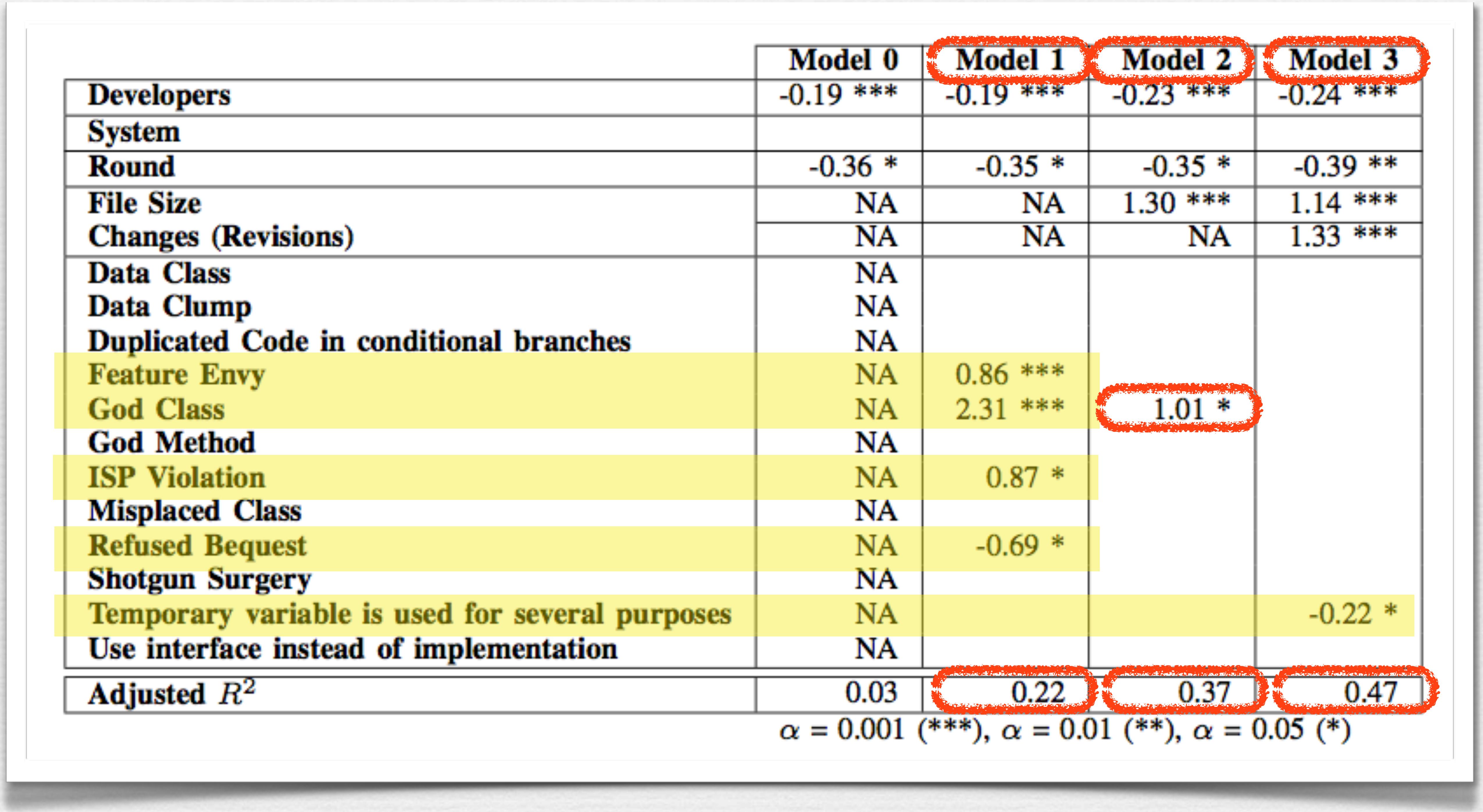
	Model 0	Model 1	Model 2	Model 3
Developers	-0.17 ***	-0.17 ***	-0.18 ***	-0.20 ***
Sys				0.12 *
Rou				**
File				***
Cha				***
Dat				
Dat				
Dup				
Fea				
Go				3 *
Go				
ISP				
Mis				
Ref				
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			
Use interface instead of implementation	NA	0.72 *		0.53 *
Adjusted R^2	0.02	0.26	0.31	0.50

α = 0.001 (***), α = 0.01 (**), α = 0.05 (*)

Revisions still impacts more than file size and code smells

Only God Class remains significant after revisions

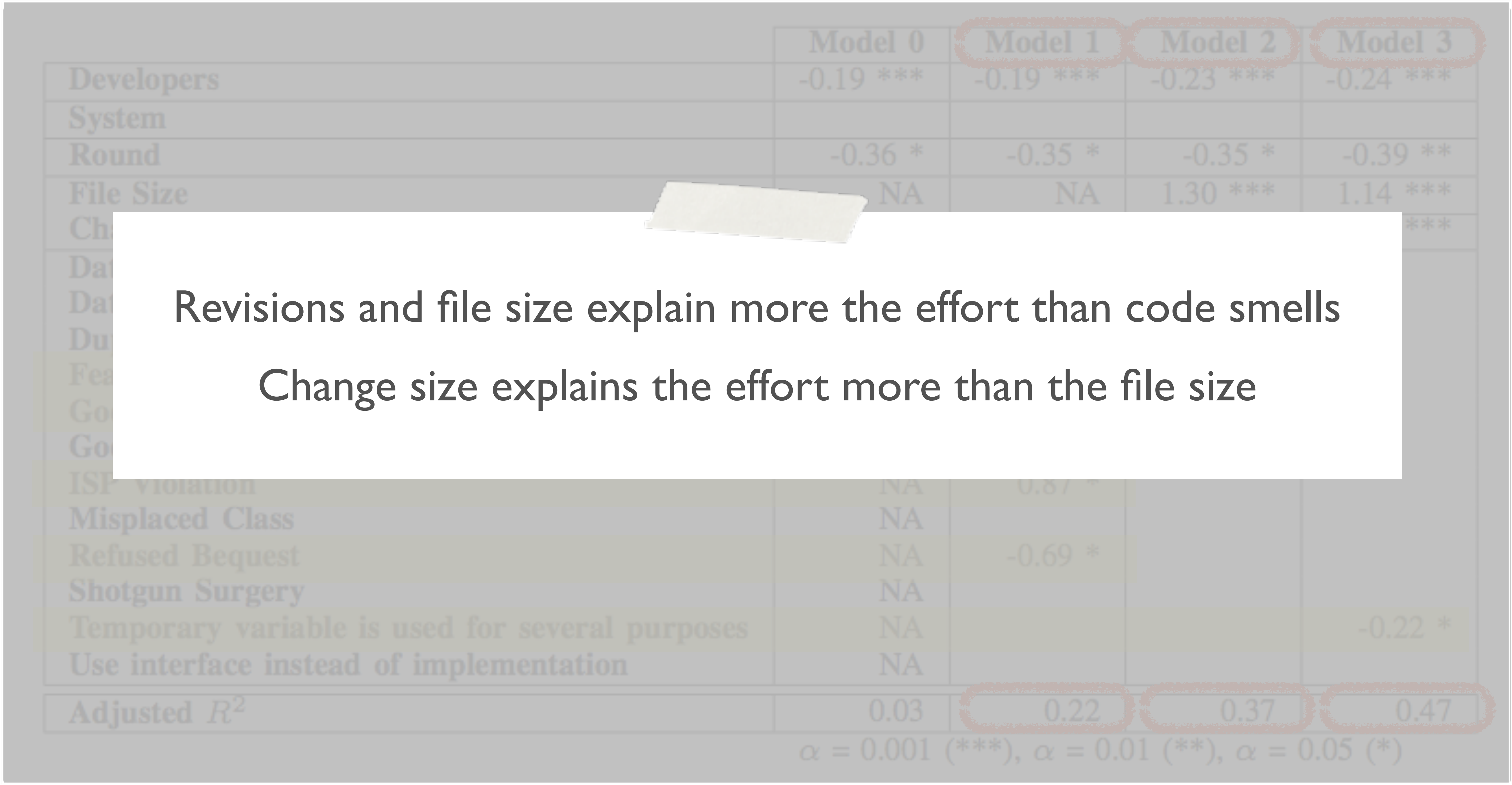
Results: Reading Effort



	Model 0	Model 1	Model 2	Model 3
Developers	-0.19 ***	-0.19 ***	-0.23 ***	-0.24 ***
System				
Round	-0.36 *	-0.35 *	-0.35 *	-0.39 **
File Size	NA	NA	1.30 ***	1.14 ***
Changes (Revisions)	NA	NA	NA	1.33 ***
Data Class	NA			
Data Clump	NA			
Duplicated Code in conditional branches	NA			
Feature Envy	NA	0.86 ***		
God Class	NA	2.31 ***	1.01 *	
God Method	NA			
ISP Violation	NA	0.87 *		
Misplaced Class	NA			
Refused Bequest	NA	-0.69 *		
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			-0.22 *
Use interface instead of implementation	NA			
Adjusted R^2	0.03	0.22	0.37	0.47

α = 0.001 (***), α = 0.01 (**), α = 0.05 (*)

Results: Reading Effort



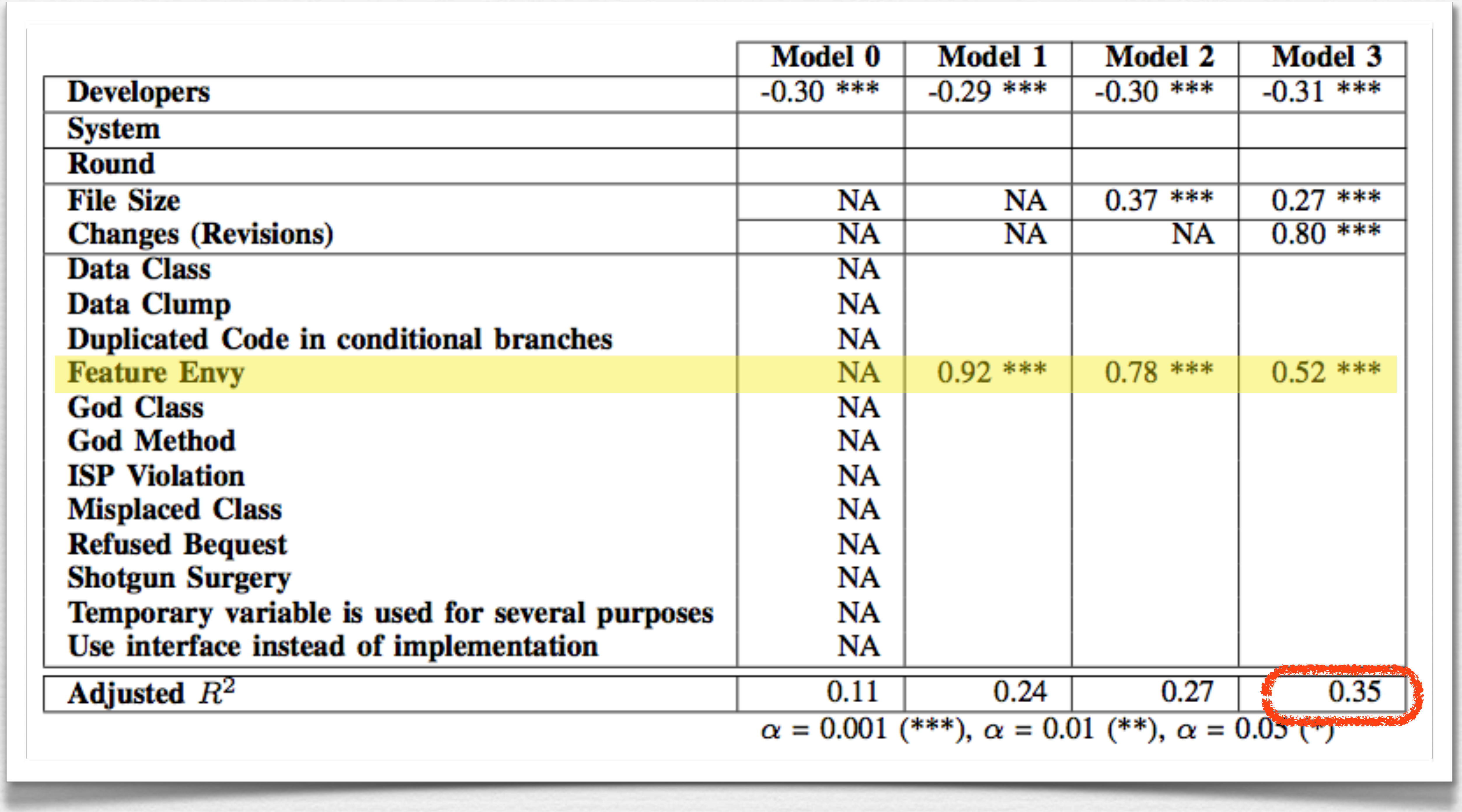
	Model 0	Model 1	Model 2	Model 3
Developers	-0.19 ***	-0.19 ***	-0.23 ***	-0.24 ***
System				
Round	-0.36 *	-0.35 *	-0.35 *	-0.39 **
File Size	NA	NA	1.30 ***	1.14 ***
Ch				***
Dat				
Dat				
Du				
Fea				
Go				
Go				
ISP Violation	NA	0.87 *		
Misplaced Class	NA			
Refused Bequest	NA	-0.69 *		
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			-0.22 *
Use interface instead of implementation	NA			
Adjusted R^2	0.03	0.22	0.37	0.47

α = 0.001 (***), α = 0.01 (**), α = 0.05 (*)

Revisions and file size explain more the effort than code smells

Change size explains the effort more than the file size

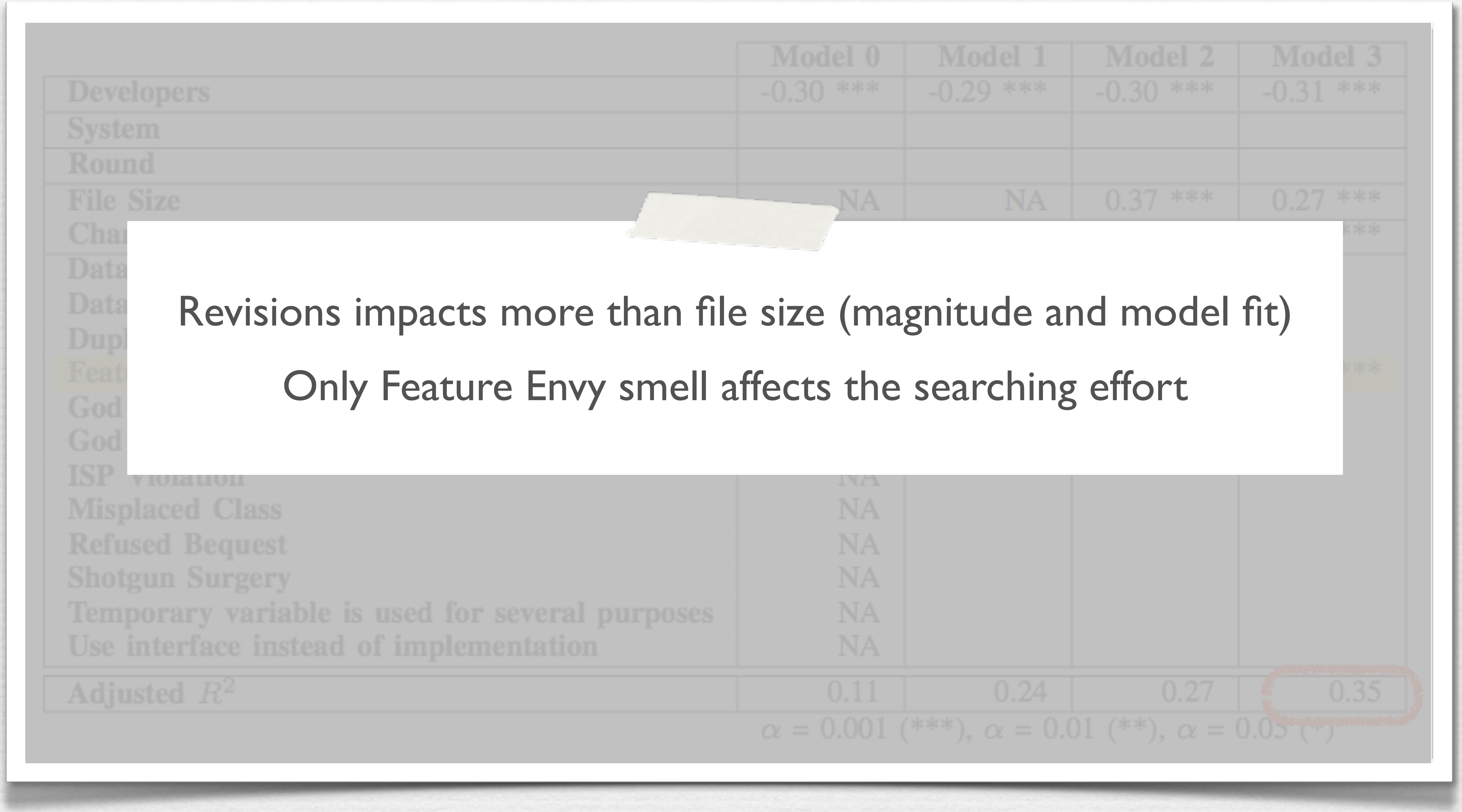
Results: Searching Effort



	Model 0	Model 1	Model 2	Model 3
Developers	-0.30 ***	-0.29 ***	-0.30 ***	-0.31 ***
System				
Round				
File Size	NA	NA	0.37 ***	0.27 ***
Changes (Revisions)	NA	NA	NA	0.80 ***
Data Class	NA			
Data Clump	NA			
Duplicated Code in conditional branches	NA			
Feature Envy	NA	0.92 ***	0.78 ***	0.52 ***
God Class	NA			
God Method	NA			
ISP Violation	NA			
Misplaced Class	NA			
Refused Bequest	NA			
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			
Use interface instead of implementation	NA			
Adjusted R^2	0.11	0.24	0.27	0.35

α = 0.001 (***), α = 0.01 (**), α = 0.05 (*)

Results: Searching Effort



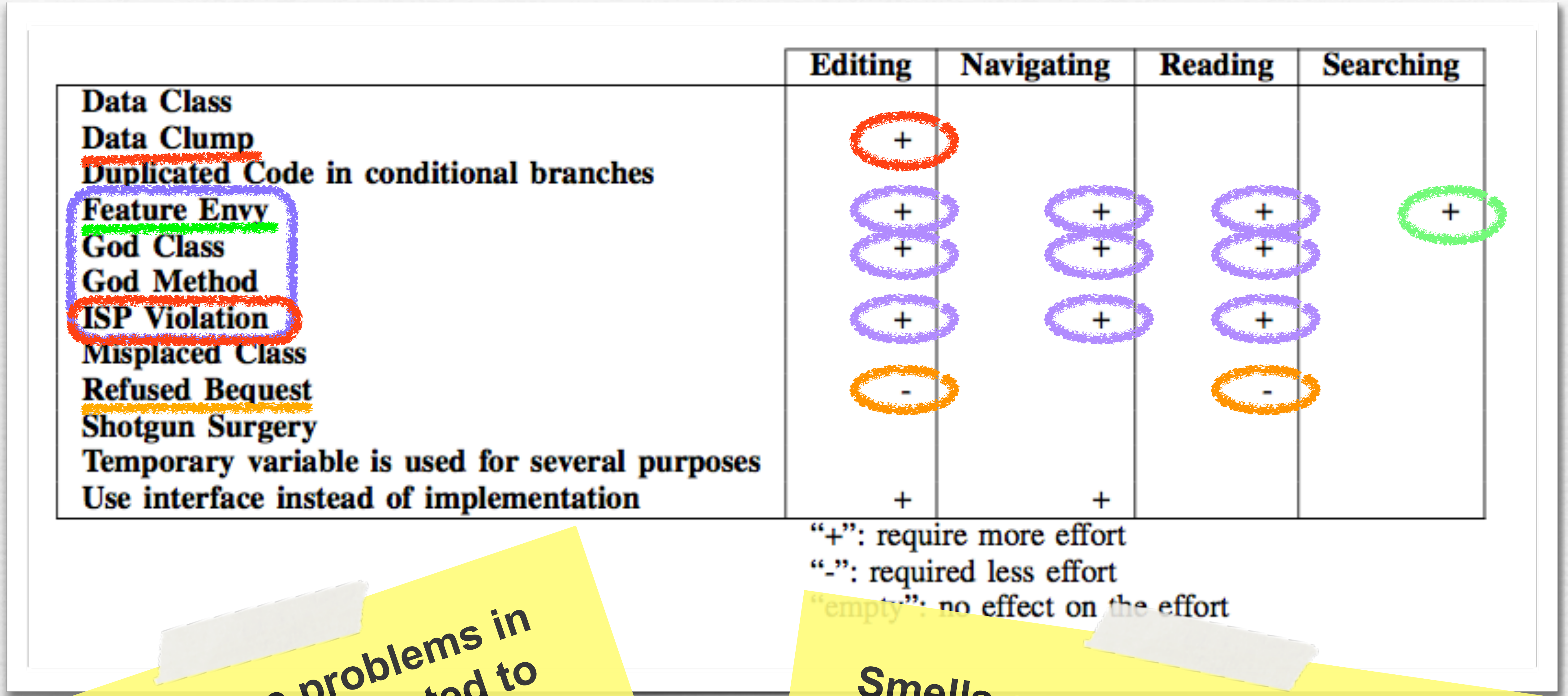
	Model 0	Model 1	Model 2	Model 3
Developers	-0.30 ***	-0.29 ***	-0.30 ***	-0.31 ***
System				
Round				
File Size	NA	NA	0.37 ***	0.27 ***
Char				***
Data				
Data				
Dupl				
Feat				***
God				
God				
ISP Violation	NA			
Misplaced Class	NA			
Refused Bequest	NA			
Shotgun Surgery	NA			
Temporary variable is used for several purposes	NA			
Use interface instead of implementation	NA			
Adjusted R^2	0.11	0.24	0.27	0.35

α = 0.001 (***), α = 0.01 (**), α = 0.05 (*)

Revisions impacts more than file size (magnitude and model fit)

Only Feature Envy smell affects the searching effort

Summary of Results



	Editing	Navigating	Reading	Searching
Data Class				
Data Clump	+			
Duplicated Code in conditional branches				
Feature Envy	+	+	+	+
God Class	+	+	+	
God Method				
ISP Violation	+	+	+	
Misplaced Class				
Refused Bequest	-		-	
Shotgun Surgery				
Temporary variable is used for several purposes				
Use interface instead of implementation	+	+		

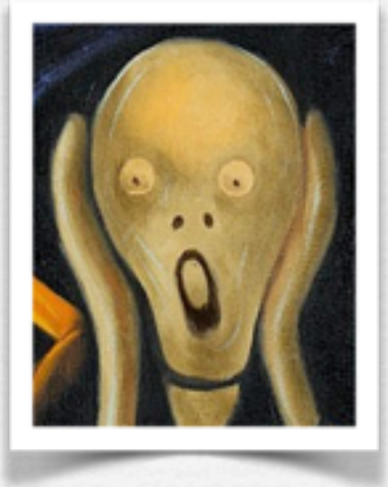
"+": require more effort

"-": required less effort

"empty": no effect on the effort

Maintenance problems in previous work related to increased effort for editing, navigating and reading

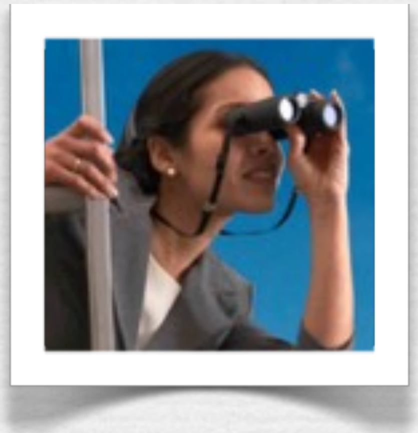
Smells explain better Editing and Navigating effort than file size, but not for Reading and Searching



Threats to validity

- Learning effect (accounted with rounds)
- Instrumentation and log processing accuracy
- Choice of tools for code smells (inCode and Together)
- Generalization is limited to context

Future work



- Think-aloud + Log analysis = how smells affect the activities
- Taxonomies on programming problems during maintenance
- Explore GLM for better explanatory models
- More replications!

What to take home today...

- Structural attributes represented in the form of different code smells do indeed have an effect on the developers' effort for certain kinds of activities.

- Different code smells significantly impact the effort of different activities. For example, we found that "Feature Envy" affects searching effort while "Data Clumps" affects editing effort.

- The effect of code smells on editing and navigating effort is, in fact, larger than file size, whiles the opposite is the true for reading and searching effort

- If the effect of code smells is contingent on the type of activity, this may mean that is contingent on the task at hand (e.g., some tasks may require more reading than others)

Thank you!

Contact: aiko.yamashita@cwi.nl