lab-soft
software engineering laboratory

On The Effectiveness of Concern Metrics to Detect Code Smells: An Empirical Study

Juliana Padilha, Juliana Pereira, Eduardo Figueiredo, Jussara Almeida, Alessandro Garcia, Cláudio Sant'Anna

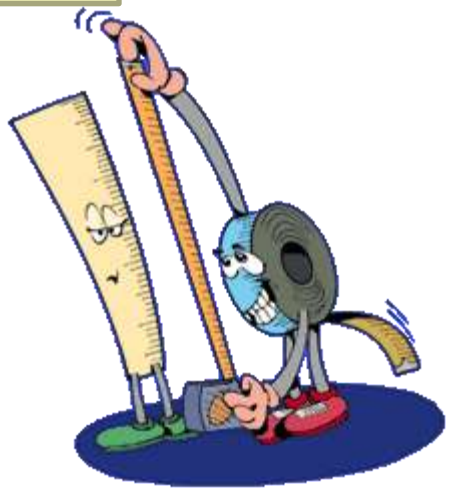
{juliana.padilha,juliana.pereira,figueiredo, jussara}@dcc.ufmg.br, afgarcia@inf.puc-rio.br, santanna@dcc.ufba.br

Software Engineering Lab (LabSoft)
http://labsoft.dcc.ufmg.br/

Motivation

- Software metrics are used to

  - Analyze a particular aspect of quality
  - Measure a quality attribute



Motivation

- The lack of quality in software can take you at least two problems

→ **Low Separation of Concern**

→ **Design flaws, such as Code Smells**

Separation of Concern

- **A concern is**
  - something you want to treat as conceptual unit modular

- Example of mechanisms used to separate concerns
  - **Class**
  - **Methods**

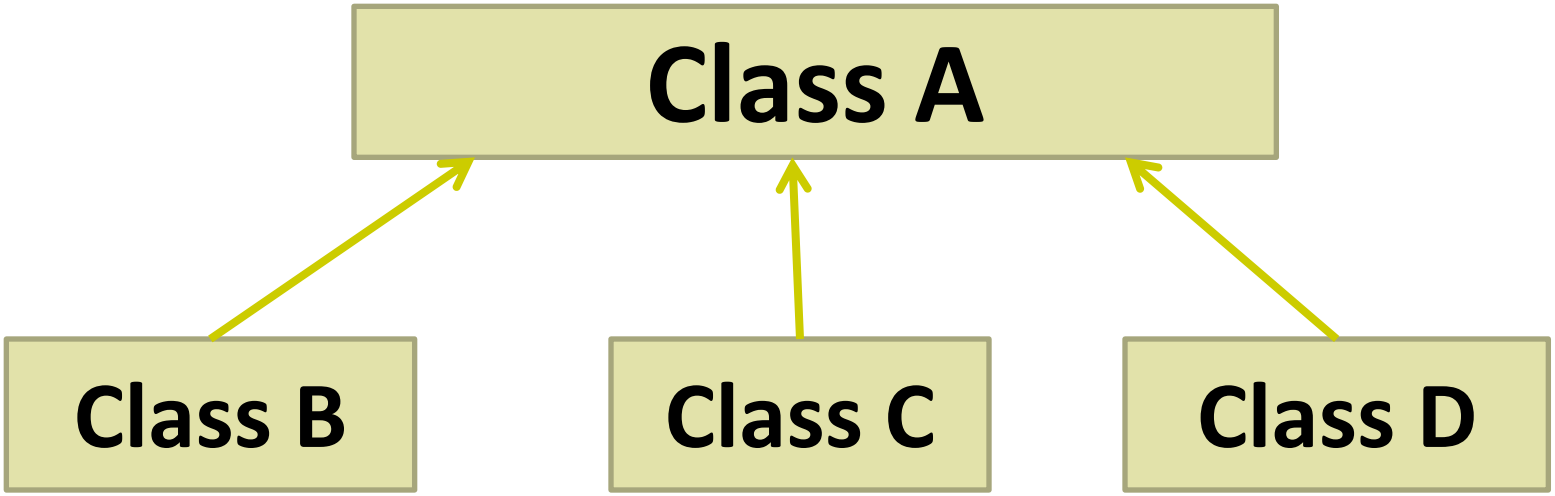
Code Smells

- **Divergent Change**
- **Shotgun Surgery**
- **God Class**

```
public static void download(String address, String localFileName, String host, int porta) {

    Proxy proxy = new Proxy(Proxy.Type.HTTP, new InetSocketAddress(host, porta));

    OutputStream out = null;
    URLConnection conn = null;
    InputStream in = null;

    try {
        URL url = new URL(address);
        out = new BufferedOutputStream(
        conn = url.openConnection(proxy
        in = conn.getInputStream();
        byte[] buffer = new byte[1024];
        int numRead;
        long numWritten = 0;
        while ((numRead = in.read(buffe
            out.write(buffer, 0, nu
            numWritten += numRead;
        }
        System.out.println(localFileNam
    } catch (Exception exception) {
        exception.printStackTrace();
    } finally {
        try {
            if (in != null) {
                in.close();
            }
            if (out != null) {
                out.close();
            }
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }
}
```

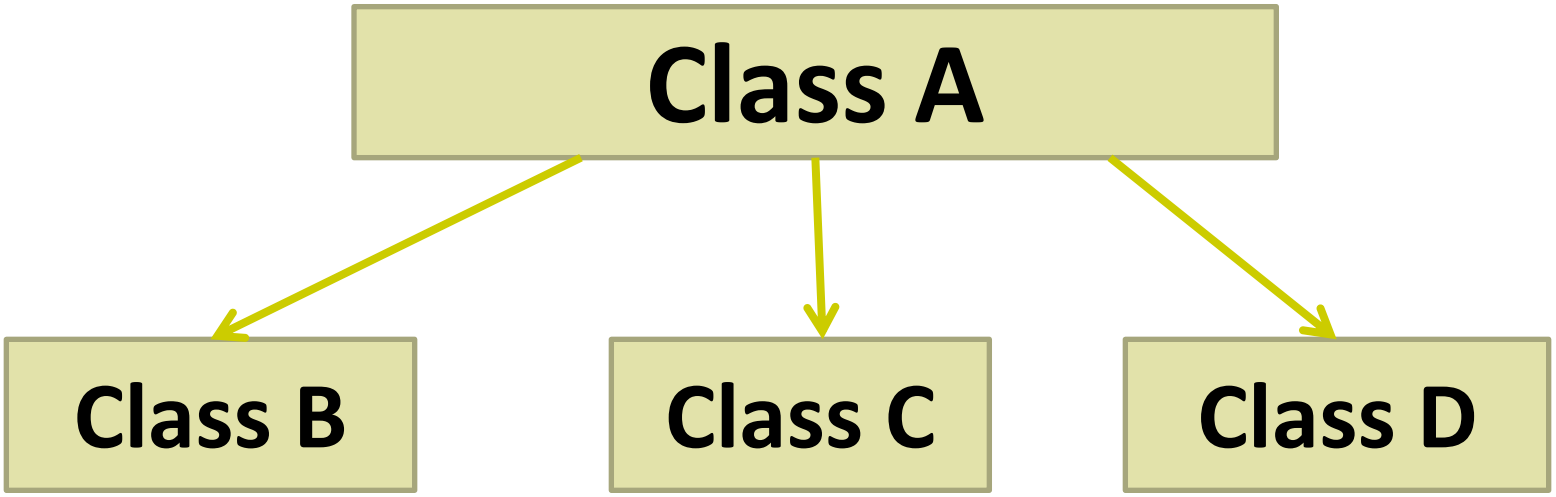
Divergent Change

- Occurs when a class is often modified
- **Symptom:** an scattering of concerns



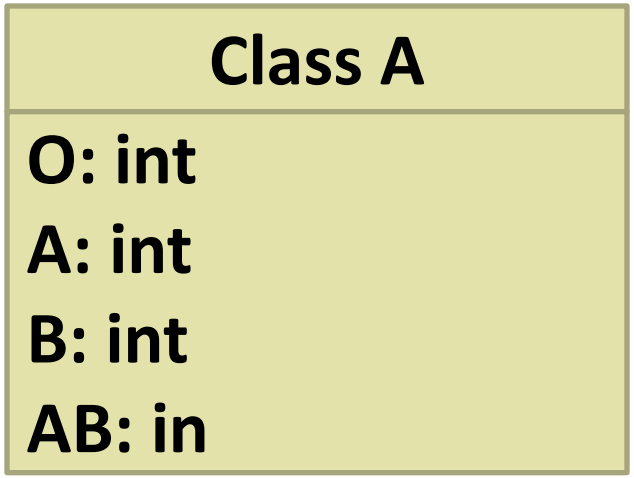
Shotgun Surgery

- Occurs when a class is changed and minor changes should be made
- **Symptom:** a tangled concerns



God Class

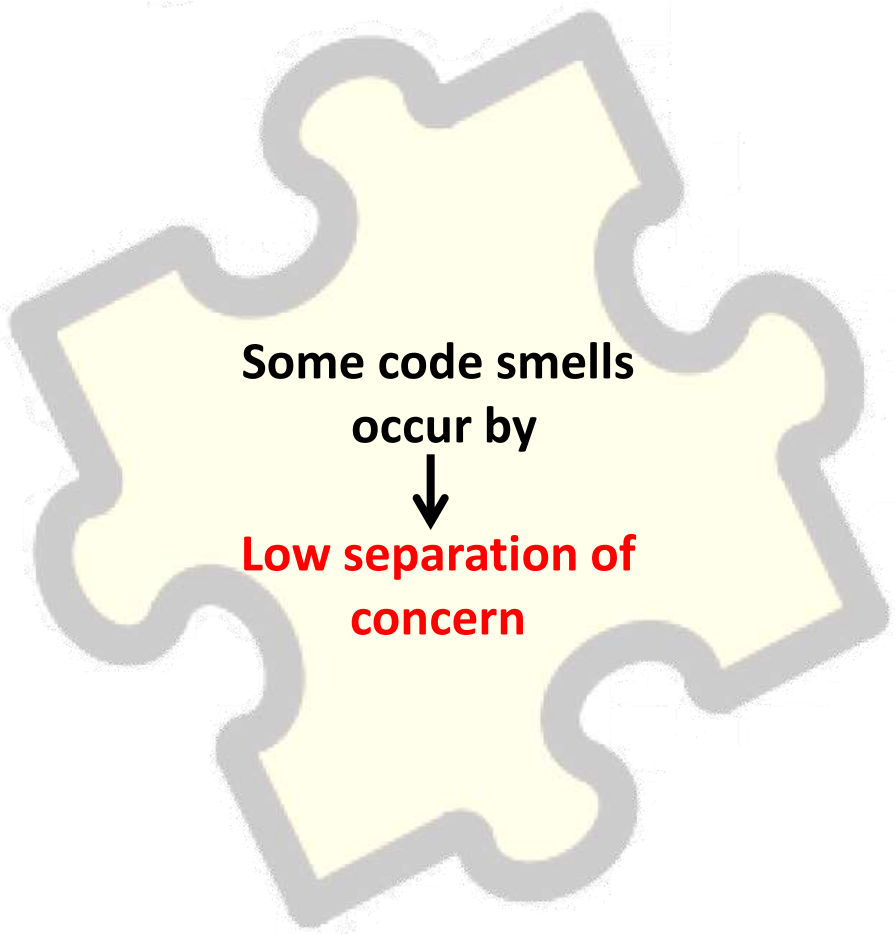
- Class that does almost everything in the system
- **Symptom:** implementation of many concerns



Traditional Metrics

- Metrics used in study
  - Coupling between Objects **(CBO)**
  - Lack of Cohesion in Methods **(LCOM)**
  - Lines of Code **(LOC)**
  - Number of Attributes **(NOA)**
  - Number of Methods **(NOM)**
  - Weighted Methods per Class **(WMC)**

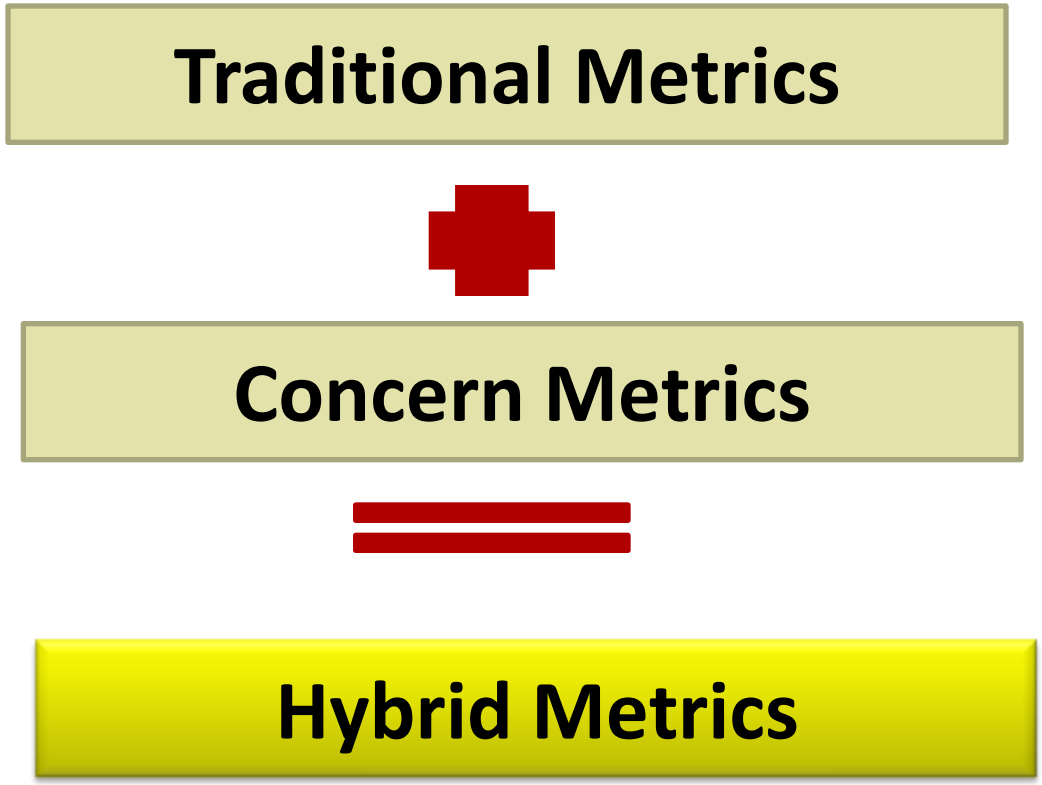
Limitations of Traditional Metrics



Concern Metrics

- Metrics used in study
  - Concern Diffusion per Components **(CDC)**
  - Concern Diffusion over Operations **(CDO)**
  - Number Concerns per Component **(NCC)**

Hybrid Metrics



Target Systems

- **Health Watcher**
  - About 6KLOC
  - Some concerns implemented:
    - *Business*, *Concurrency*, *Distribution*, *Exception Handling*, *Persistence*, and *View*
- **MobileMedia**
  - About 4KLOC
  - Some concerns implemented:
    - *Sorting*, *Favorites*, *Exception Handling*, *Security,* and *Persistence*

Background of Subjects

Divergent Change Code Smell					
Metrics		**Traditional**	**Concern**	**Hybrid**	**No Answer**
Knowledge	Class Diagram	S4 - S6	S9 - S11	S14 - S24	S1, S2, S3, S7, S8, S12, S13, S18
	Java	S4 - S6	S9 - S11	S14 - S24	
	Work Experience	S5	S10,S11	S14 - S17, S20	
	Measurement	S4 - S6	S9 - S11	S14 - S24	
Shotgun Surgery Code Smell					
		Traditional	**Concern**	**Hybrid**	**No Answer**
Knowledge	Class Diagram	S27 - S29	S31 - S32	S34 - S37, S39 - S44	S25, S26, S30, S33, S38
	Java	S27 - S29	S31 - S32	S34 - S37, S39 - S45	
	Work Experience	S28	S32, S32	S34 - S37, S40	
	Measurement	S27 - S29	S31 - S32	S34 - S37, S39 - S44	
God Class Code Smell					
		Traditional	**Concern**	**Hybrid**	**No Answer**
Knowledge	Class Diagram	S45, S46	S48 - S50	S51- S54	-
	Java	S45, S46	S48 - S50	S51- S54	
	Work Experience	S45	S48	S51	
	Measurement	S45, S46	S49 - S50	S51 -S54	

Experimental Tasks

- **Experimental Tasks**
  - Training on the metrics and code smells
  - The participant had access the metrics they were assigned them

- **Each participant**
  - Read the description of the system (HW or MM)
  - Identify the classes with code smells (DC, SS and GC)
  - Register the time spent on each task
  - Document which metrics they found useful

Reference List

System	Smell	Classes in the Reference List
Health Watcher	Divergent Change (12 classes)	EmployeeRecord, HealthWatcherFacade, HealthUnitRecord, PersistenceMechanism, IFacade, HealthWatcherFacadeInit, IPersistenceMechanism, ServletInsertEmployee, ComplaintRecord, ServletSearchComplaintData, ServletUpdateComplaintData, ServletUpdateHealthUnitData
	Shotgun Surgery (8 classes)	PersistenceMechanism, ComplaintRecordRDB, EmployeeRepositoryRDB, IComplaintRepository, HealthUnitRepositoryRDB, IPersistenceMechanism, IHealthUnitRepository, IEmployeeRepository
	God Class (3 classes)	HealthWatcherFacade, HealthWatcherFacadeInit, PersistenceMechanism
Mobile Media	Divergent Change (4 classes)	ImageMediaAccessor, MediaController, MediaAcessor, MediaListController
	Shotgun Surgery (3 classes)	ControllerInterface, MediaAccessor, ScreenSingleton

Recall and Precision

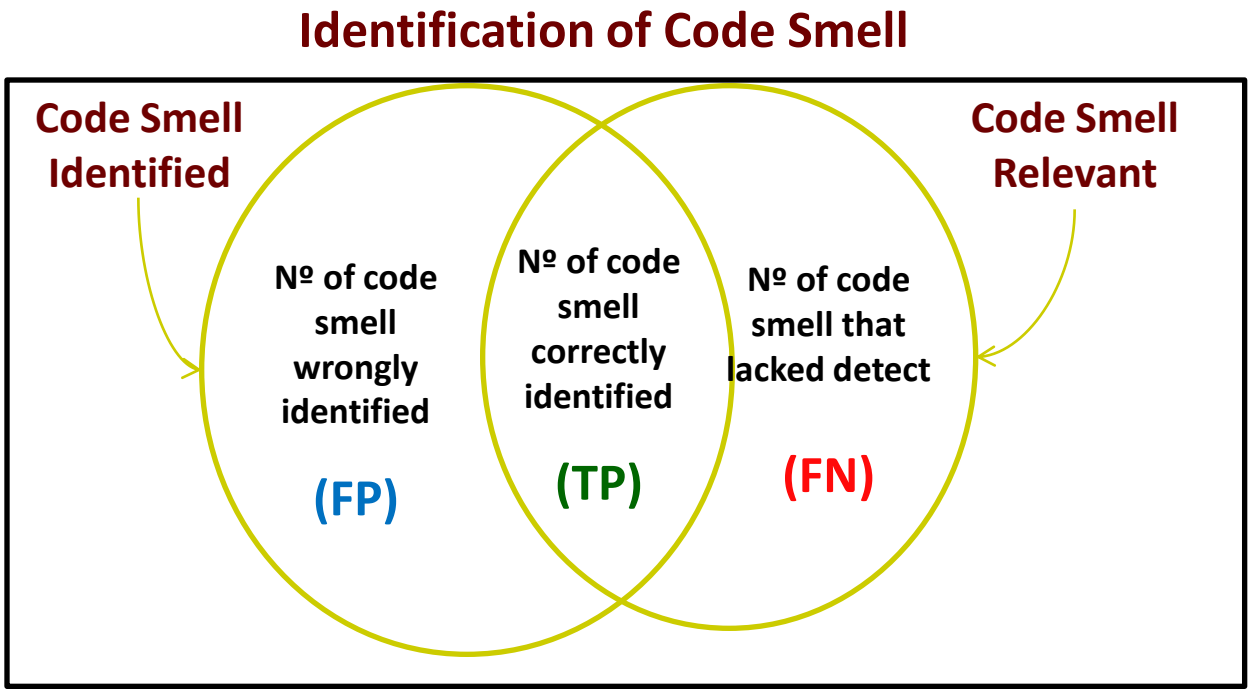
Hits

- True Positive (TP)

Error

- False Positive (FP)
- False Negative (FN)

Recall and Precision



$$\text{Recall:} \quad \frac{TP}{TP + FN}$$

$$\text{Precision:} \quad \frac{TP}{TP + FP}$$

Divergent Change

Concern metrics support detection this code smell

Group	Traditional						Concern				
Subject	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
R(%)	17	17	17	33	25	25	100	100	33	25	50
P(%)	67	50	40	50	17	25	63	100	100	25	29
T(min)	15	15	40	38	41	36	26	29	29	15	33

Group	Hybrid												
Subject	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
R(%)	75	8	25	50	100	25	50	0	50	25	50	25	50
P(%)	100	50	75	25	67	33	40	0	67	17	40	17	50
T(min)	40	31	23	36	27	39	24	11	18	19	13	13	12

Shotgun Surgery

Hard to detect this code smell with metrics

Group	Traditional					Concern		
Subject	S25	S26	S27	S28	S29	S30	S31	S32
R(%)	13	13	0	67	33	75	25	33
P(%)	25	33	0	25	25	35	40	25
T(min)	6	10	27	12	14	13	28	14

Group	Hybrid										
Subject	S33	S34	S35	S36	S37	S38	S39	S40	S41	S42	S43	S44
R(%)	13	50	67	33	33	33	0	0	33	0	0	0
P(%)	25	80	6	33	25	33	0	0	20	0	0	0
T(min)	35	14	19	15	4	10	14	9	21	3	7	5

God Class

Joint Data Analysis

Group	Traditional			Concern			Hybrid			
Subject	S45	S46	S47	S48	S49	S50	S51	S52	S53	S54
R(%)	33	33	67	100	67	100	33	100	100	100
P(%)	33	33	67	75	100	75	50	100	60	75
T(min)	18	25	27	37	66	43	22	53	51	35

Statistical Analysis and Discussions

- Analyzes the recall of concern metrics compared to the traditional metrics
- Discusses to which extent the background of subjects and the time spent impact the recall of code smell
- Analyzes possible combinations of metrics that increases the recall of indentifying each code smell

Comparing Metrics

RQ 1- How accurate do concern metrics perform in comparison with traditional metrics to detect code smells?

- We apply an Unpaired t-test
  - with 90% confidence level

Comparing Metrics

Systems	Health Watcher			MobileMedia		
Groups	**Traditional (T)**	**Concern (C)**	**Hybrid (H)**	**Traditional (T)**	**Concern (C)**	**Hybrid (H)**
All	(13.26, 35.34)	(54.59, 95.40)	(20.98, 91.02)	(18.70, 46.49)	(22.94, 49.32)	(19.08, 43.32)
DC	(11.6, 30.4)	(12.5, 142.9)	(-22.7, 94.8)	(23.9, 26.5)	(-41.4, 116.4)	(27.2, 57.9)
SS	(-4.0, 21.3)	(-107.9, 207.9)	(-85.3, 148.3)	(-57.3, 157.3)	(28.9, 37.9)	(6.4, 33.4)
GC	(11.2, 77.4)	(56.9, 121.1)	(43.8, 123)	-	-	-

This Table shows Confidence Intervals (CI) for the average recall in Health Watcher and MobileMedia

Comparing Metrics

Systems	Health Watcher			MobileMedia		
Groups	Traditional (T)	Concern (C)	Hybrid (H)	Traditional (T)	Concern (C)	Hybrid (H)
All	(13.26, 35.34)	(54.59, 95.40)	(20.98, 91.02)	(18.70, 46.49)	(22.94, 49.32)	(19.08, 43.32)
DC	(11.6, 30.4)	(12.5, 142.9)	(-22.7, 94.8)	(23.9, 26.5)	(-41.4, 116.4)	(27.2, 57.9)
SS	(-4.0, 21.3)	(-107.9, 207.9)	(-85.3, 148.3)	(-57.3, 157.3)	(28.9, 37.9)	(6.4, 33.4)
GC	(11.2, 77.4)	(56.9, 121.1)	(43.8, 123)	-	-	-

This Table shows Confidence Intervals (CI) for the average recall in Health Watcher and MobileMedia

Are systems different?

Metric	Traditional		Concern		Hybrid	
System	HW	MM	HW	MM	HW	MM
Average	24.300	37.600	75	36.133	56	31.2
S^2	339.567	397.707	1032.267	161.853	1510.476	723.747
n	10	4	8	3	5	20
v	7		11		5	
CI	(13.26,35.34)	(18.70, 46.49)	(54.59, 95.40)	(22.94, 49.32)	(20.98, 91.02)	(19.08, 43.32)

Are systems different?

Metric	Traditional		Concern		Hybrid	
System	HW	MM	HW	MM	HW	MM
Average	24.300	37.600	75	36.133	56	31.2
S^2	339.567	397.707	1032.267	161.853	1510.476	723.747
n	10	4	8	3	5	20
v	7		11		5	
CI	(13.26,35.34)	(18.70, 46.49)	(54.59, 95.40)	(22.94, 49.32)	(20.98, 91.02)	(19.08, 43.32)

- **Analyze about Concern Metrics**
  - for two systems **do not overlap** (IC)
  - systems **are different at the 90% confidence**
  - **HW is better** – Concern metrics leads to higher recall

Are systems different?

Metric	Traditional		Concern		Hybrid	
System	HW	MM	HW	MM	HW	MM
Average	24.300	37.600	75	36.133	56	31.2
S^2	339.567	397.707	1032.267	161.853	1510.476	723.747
n	10	4	8	3	5	20
v	7		11		5	
CI	(13.26,35.34)	(18.70, 46.49)	(54.59, 95.40)	(22.94, 49.32)	(20.98, 91.02)	(19.08, 43.32)

- **Analyze about Traditional and Hybrid Metrics**
  - for two systems **do overlap**

Are systems different?

Metric	Traditional		Concern		Hybrid	
System	HW	MM	HW	MM	HW	MM
Average	24.300	37.600	75	36.133	56	31.2
S^2	339.567	397.707	1032.267	161.853	1510.476	723.747
n	10	4	8	3	5	20
v	7		11		5	
CI	(13.26,35.34)	(18.70, 46.49)	(54.59, 95.40)	(22.94, 49.32)	(20.98, 91.02)	(19.08, 43.32)

- **Conclusion:**
  - **Concern metrics: systems** used does **impact the detection** of code smells
  - **Traditional and Hybrid metrics: is not** significantly **influenced by it**

Concern metrics lead to significantly different results?

Metric	Traditional		Concern		Hybrid	
System	HW	MM	HW	MM	HW	MM
Average	24.300	37.600	75	36.133	56	31.2
S^2	339.567	397.707	1032.267	161.853	1510.476	723.747
n	10	4	8	3	5	20
v	7		11		5	
CI	(13.26,35.34)	(18.70, 46.49)	(54.59, 95.40	(22.94, 49.32)	(20.98, 91.02	(19.08, 43.32)

- Concern metrics produce higher recall, compared to traditional for HW
- There is a statistical tie (90% IC) for MM
  - Though average results are better for the concern metrics

Concern metrics lead to significantly different results?

Metric	Traditional		Concern		Hybrid	
System	HW	MM	HW	MM	HW	MM
Average	24.300	37.600	75	36.133	56	31.2
S^2	339.567	397.707	1032.267	161.853	1510.476	723.747
n	10	4	8	3	5	20
v	7		11		5	
CI	(13.26,35.34)	(18.70, 46.49)	(54.59, 95.40)	(22.94, 49.32)	(20.98, 91.02)	(19.08, 43.32)

- **Conclusion:**
  - Concern metrics are the best ones, among those analyzed, for the detection of 3 types of code smells studied

Type of code smell detected influences the recall

- In comparison concern metrics with traditional ones

Metric	Divergent Change		Shotgun Surgery	
System	HW	MM	HW	MM
CI	(35.87, 49,12)	(28.34, 50.22)	(9.38, 44.66)	(13.64, 37.42)

- **Analysis to two code smells (DC and SS)**
- **God Class was not analyzed on MobileMedia**

Type of code smell detected influences the recall

- **Conclusion:**
  - **There is no significant difference** between the two systems in terms of recall, **for any code smell**
  - Subjects had
    - same rates of code smells
    - Independent of the analyzed system

Concern xTraditional x Hybrid

Analysis	Concern and Traditional			Concern and Hibrid		
Code Smell	DC	SS	GC	DC	SS	GC
x_C-$x_{T;H}$	39,267	19,133	44,667	20,6	22,500	5,75
S_C-$S_{T;H}$	16,4064	19,42895	15,7938	17,85219	16,8622	20,0390
v	4,32605	6,103933	5,99288	6,699767	3,53686	6,310562
CI	(4.288, 74.25)	(-18.6, 56.88)	(13.98, 75.35)	(-13.2, 54.43)	(-17.2, 62.18)	(-33.2, 44.69)

Legend

x_C-$x_{T;H}$: Mean difference

S_C-$S_{T;H}$: Standard Deviation of the mean difference

V : Degrees of freedom

Concern xTraditional x Hybrid

- **Conclusion:**

- The superiority of the concern metrics
  - varied according to the type of code smell
- Concern metrics was consistently better
  - in comparison with traditional metrics in the DC and GC detection cases
- SS is not statistically significant (with 90% confidence).

Background of Subjects

RQ2- Does background of subjects impact the efficiency of the detected code smell?

- We apply a 2^k full Factorial Design
  - with k=2 factors

Background of Subjects

Time	Work Experience	
	No (-1)	Some (+1)
Short (-1)	29.000	36.000
Long (+ 1)	83.500	100.000

- Work experience:
  - **No experience** = who never worked, or worked for fewer than 6 months
  - **Some experience** = who worked for at least 6 months in software development industry
- Time spent
  - **Short time** = who took less than 33 minutes (overall average)
  - **Long time** = who took at least 33 minutes

Background of Subjects

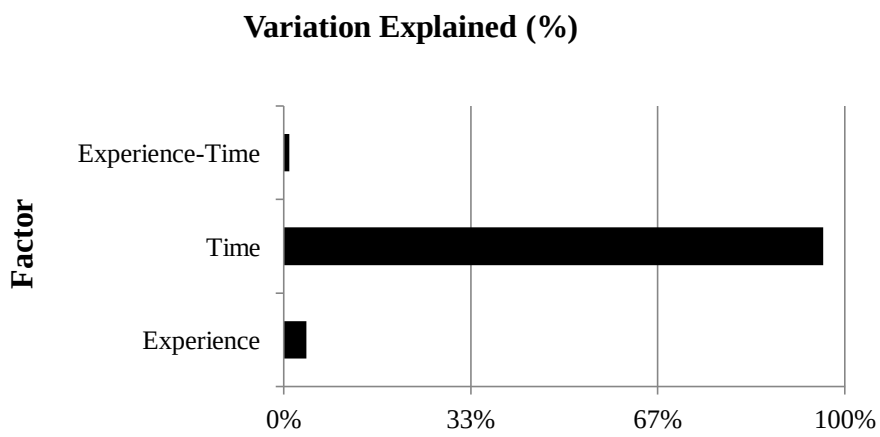
Time	Work Experience	
	No (-1)	Some (+1)
Short (-1)	29.000	36.000
Long (+ 1)	83.500	100.000

$$X_A = \begin{cases} -1, if\ no\ experience \\ 1, if\ some\ experience \end{cases}$$

$$X_B = \begin{cases} -1, if\ short\ time \\ 1, if\ long\ time \end{cases}$$

Background of Subjects

Factor	Effect	Variation explained (%)
A (Experience)	138.063	4%
B (Time)	3510.563	96%
AB (Time and Experience)	22.563	1%
Total Variation	3671.188	-



Conclusions:

- The experience factor working no influence
- The time factor influence on code smell detection
- The longer the time, the highest rate of Recall
  i.e. God Class

Combinations of Metrics

RQ3- Is there a combination of metrics that increases recall of code smell detection?

- Analysis of data collected
  - consider metrics reported by more than 3 subjects

Combinations of Metrics

- **Divergent Change:** NCC e LCOM
- i.e.

Subjects	S7	S12	S16
Recall	100%	75%	100%
Precision	63%	100%	67%
Time	26min	40min	27min

Combinations of Metrics

- **Shotgun Surgery:** NCC, CDC e CBO
- i.e.

Subjects	S28	S30	S37
Recall	67%	75%	33%
Precision	25%	35%	25%
Time	12min	13min	4min

Combinations of Metrics

- **God Class:** CBO e LCOM, WMC e LOC, CDO e CDLOC
- i.e.

Subjects	S52	S53	S54
Recall	100%	100%	100%
Precision	100%	60%	75%
Time	53min	51min	35min

Related Work

- Many studies involving both types of metrics [1,2, 3]
  - **Marinescu et al [1]** proposed the use of detection strategies consist of traditional metrics to detect code smells.
  - **Eaddy et al [2]** performed experiments involving the metrics of interest. They used the CDC and CDO metrics.
  - **Sant'Anna et al [3]** was the first study to assess how the metrics of interest can improve the process-related defects failures modular design.

[1] Marinescu *et al*. (2004). "Detection Strategies: Metrics-based Rules for Detecting Design Flaws." (ICSM)
[2] Eaddy, M. *et al*. (2008). "Do Crosscutting Concerns Cause Defects?" IEEE Transactions on Software Engineering. 34:497-515.
[3] Sant'Anna *et al.* (2008). "Evaluating the Efficacy of Concern Driven Metrics: A Comparative Study." (ACOM)

Conclusions

- Our results revealed that
  - concern metrics are useful to detect
    - Divergent Change
    - God Class
  - Number of Concern per Components (NCC)
    - efficient to detect Divergent Change

lab-soft
software engineering laboratory

Thanks!

FAPEMIG
Fundação de Amparo à Pesquisa do Estado de Minas Gerais

CAPES