Impact of Linguistic Antipatterns on Code Comprehension

MASTER THESIS DEFENSE

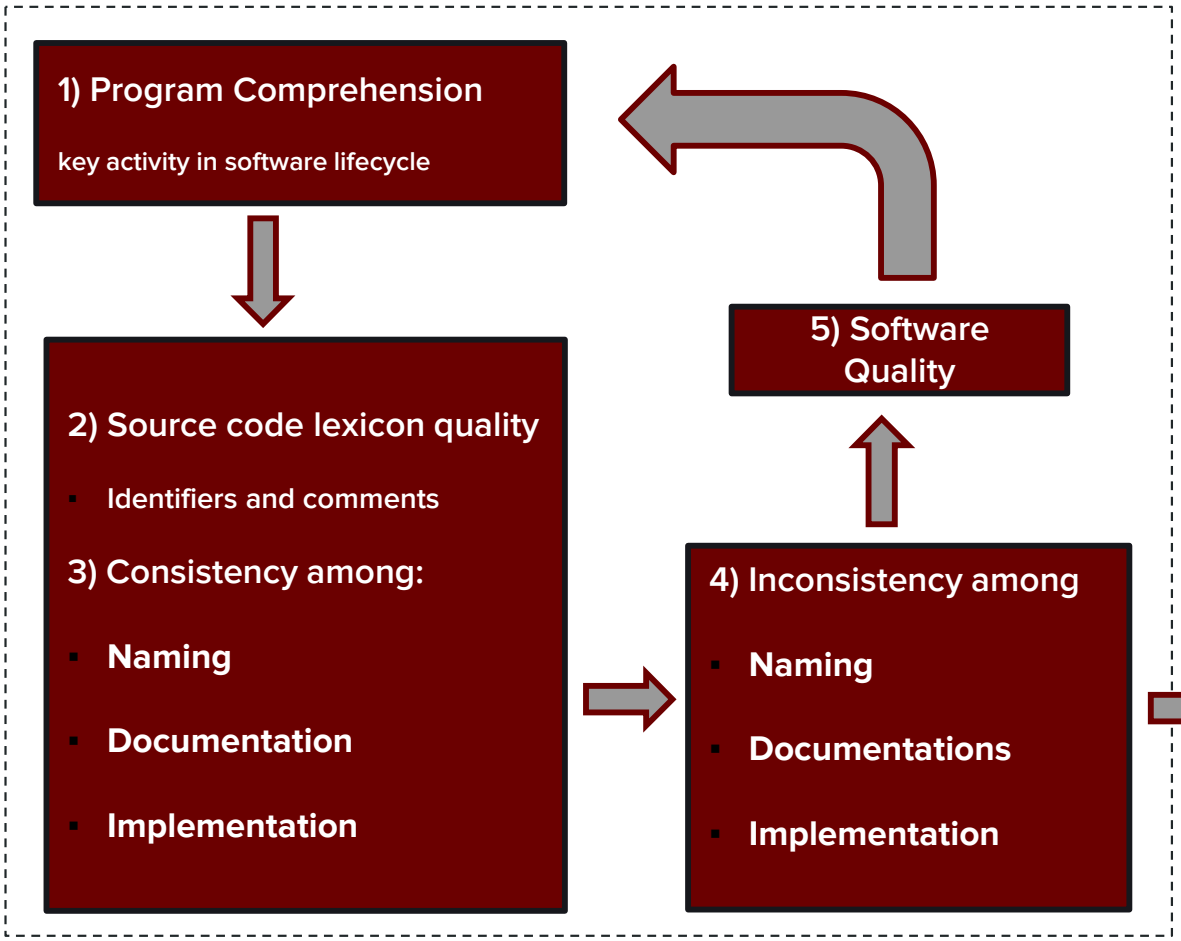
Farideh Sanei

Summer 2024

Supervised by: Dr. Yann-Gaël Guéhéneuc

Department of Computer Science
and Software Engineering
Faculty of Engineering and Computer Science
Concordia University

Software anti-patterns



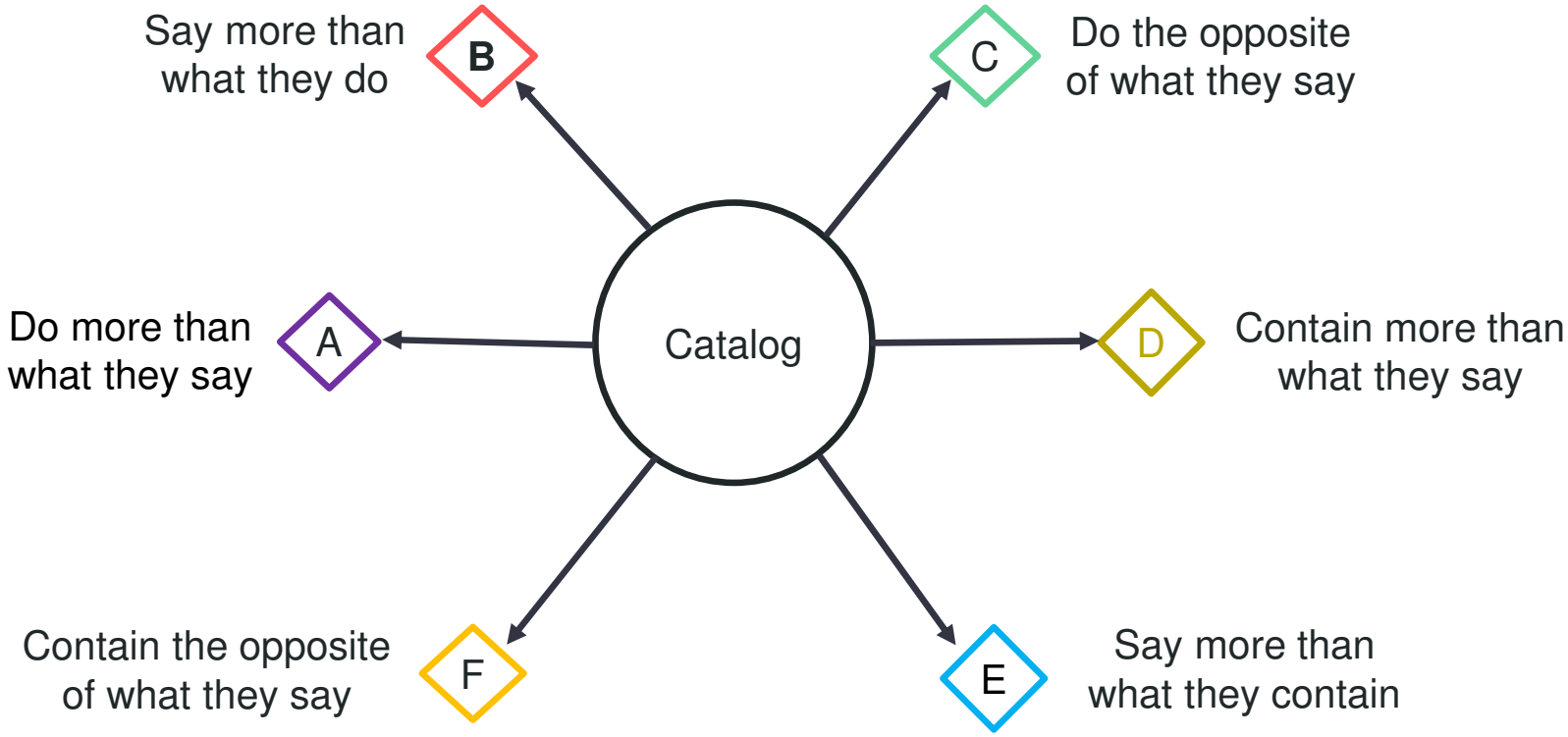
Problems

- Developers get confused and make wrong assumptions about code behavior
- Developers spend unnecessary time and effort to comprehend the source code

Linguistic Antipatterns (LA)

Arnaoudova, V., Towards Improving the Code Lexicon and its Consistency, 2014.

Linguistic Anti-Patterns Catalog



V. Arnaoudova, A new family of software antipatterns: Linguistic anti-patterns, in 17th European Conference on Software Maintenance and Reengineering (CSMR), 2013.

Linguistic Antipatterns - Examples

- **Set method returns**

  A. Do more than they say

```
public Dimension setBreadth
        (Dimension target, int source) {
  if (orientation == VERTICAL)
    return new Dimension(source,
        (int)target.getHeight());
  else
    return new Dimension(
                (int)target.getWidth(), source);
}
```

- **Attribute name and type are opposite**

  F. Contain the opposite

```
MAssociationEnd start = null;
```

- **Not answered question**

  B. Say more than they do

```
public void isValid(
        Object[] selection, StatusInfo res) {
  // only single selection
  if (selection.length == 1
        && (selection[0] instanceof IFile))
    res.setOK();
  else res.setError(""); //$NON-NLS-1$
}
```

Linguistic Anti-Patterns – Related Work

Researches on

- **Importance of Source Code lexicon quality**
- Code Comment Quality Assessment
- Definition of LAs
- Detection of LAs
- Impact of LAs

- Choice of identifiers directly influences software understandability
- Programs with comments and full-word identifiers are more understandable

F. Palma et al., Are restful apis welldesigned? detection of their linguistic (anti) patterns, ICSOC 2015.
V. Arnaoudova et al., Linguistic antipatterns: What they are and how developers perceive them, Empirical Software Engineering, 2016.

Linguistic Anti-Patterns – Related Work

Researches on

- Importance of Source Code lexicon quality
- **Code Comment Quality Assessment** → Importance of maintaining consistency between comments and code
- Definition of LAs
- Detection of LAs
- Impact of LAs

P. Rani et al., A decade of code comment quality assessment: A systematic literature review, Journal of Systems and Software, 2023.

Linguistic Anti-Patterns – Related Work

Researches on

- Importance of Source Code lexicon quality
- Code Comment Quality Assessment
- **Definition of LAs** → poor practices in naming, documentation, and implementation of an entity
- Detection of LAs
- Impact of LAs

V. Arnaoudova, A new family of software antipatterns: Linguistic anti-patterns, in Software Maintenance and Reengineering (CSMR), 2013

Linguistic Anti-Patterns – Related Work

Researches on

- Importance of Source Code lexicon quality
- Code Comment Quality Assessment
- Definition of LAs
- **Detection of LAs**
- Impact of LAs

Tools to find inconsistencies

- iComment: NLP, ML, Statistics,
- LAPD: NLP
- FINDICI: ML, word embedding, classification

Proper identifiers suggestions

- Namesake
- RepliComment

S. H. Tan et al., Testing javadoc comments to detect comment-code inconsistencies, in Software Testing, 2012 IEEE Fifth International Conference.
A. Blasi wt al., Replicomment: identifying clones in code comments, in Proceedings of the 26th Conference on Program Comprehension, 2018.

Linguistic Anti-Patterns – Related Work

Researches on

- Importance of Source Code lexicon quality
- Code Comment Quality Assessment
- Definition of LAs
- Detection of LAs
- **Impact of LAs**

- Negative impact of poor source code lexicon on developers' cognitive loads
- Developers' perception: LAs should be refactored

S. Fakhoury et al., The effect of poor source code lexicon and readability on developers' cognitive load, in Proceedings of the 26th Conference on Program Comprehension, 2018.
E. Aghajani et al A large-scale empirical study on linguistic antipatterns affecting apis, in 2018 IEEE International conference on software maintenance and evolution, 2018.

Linguistic Anti-Patterns – Related Work

Research works on

- Importance of Source Code lexicon quality
- Code Comment Quality Assessment
- Definition of LAs
- Detection of LAs
- Impact of LAs

They introduced LAs, and their detection algorithms and established a relation between LAs and program comprehension.

GAP

- **Limitations like participants' number**
- **No strong negative impact of LAs on program comprehension**
- **No study on Impact of prior knowledge of LAs on developers' code comprehension**

Main Hypothesis

- H0: LAs have a small impact on program comprehension because they can be easily recognised by developers, who are not negatively affected by their presence in the code.

- To test H0:
  - 2 Experiments (BeforeLecture and AfterLecture)
  - 4 Research Questions

Research Questions

Impact of LAs and LA Knowledge

- **RQ1: How do LAs affect developers' program comprehension?**
- **RQ2: How does knowledge of LAs affect developers' program comprehension?**

LAs Recognition rate

- **RQ3: What is the recognition rate of LAs by unknowledgeable participants?**
- **RQ4: What is the recognition rate of LAs by knowledgeable participants?**

Study Background

Studied LAs

A2 "Is" returns more than a Boolean

A3 "Set" returns a value

B4: Not answered question

F1: Attributes name and type are opposite

F2: Attributes signature and comments are opposite

D1: Says one but contains many

E1: Says many but contains one

- **Subject Systems**

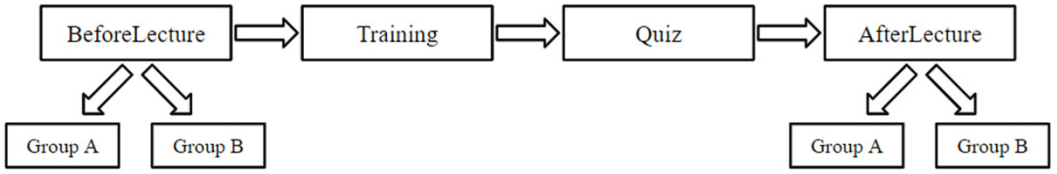
	Systems	Release date
System 1	ArgoUML0.34 ArgoUML0.14	2011-12-15 2003-12-05
System 2	Cocoon2.2.0	2013-03-14
System 3	JFreeChart1.0.19	2014-07-31
System 4	JHotDraw7.0.6	2011-09-06
System 5	Rhino1.7.7.2	2017-09-27
System 6	Xerces2-j2-11-0	2010-11-26
System 7	Apache Ant1.10.1	2017-02-06
System 8	Hibernate5.2.12.Final	2017-10-19
System 9	Apache commons-lang-3.7	2017-11-08
System 10	Apache Hadoop3.0.0	2017-12-13

- **Detection Algorithm**

→ PMD

Study Design

Experiments process



Questionnaires

Question	BeforeLecture		AfterLecture	
	Group A	Group B	Group A	Group B
Q1	A2 (with)	A2 (without)	A3 (with)	A3 (without)
Q2	E1 (without)	E1 (with)	B4 (without)	B4 (with)
Q3	F1	F1	D1	D1
Q4.a	F2	D1	F2	F2
Q4.b	Correct	Correct	F1	F1
Q4.c	Correct	F2	A3	A3

Variables

Independent Variables

- LA presence/absence
- Knowledge of LA

Dependent Variables

- Correctness
- Time
- Effort (TLX)
  - Mental demands
  - Effort
  - Frustration

Mitigating Factors

Participants' Age, Gender, Degree, Knowledge level of programming, Work experience, and Level of English fluency

Participants

229 Participants

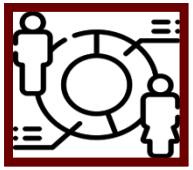
- 100 Concordia
- 129 Polytechnique

Concordia

Software Quality Engineering

Polytechnique Montreal

- Software Architecture and Advanced Design
- Software Quality Engineering



Demographics

- Good English
- Good programming skills
- Men
- Bachelor's degree
- Aged 18 to 25
- Work experience 1 to 3 years

Hypotheses

RQ1: How do LAs affect unknowledgeable developers' code understanding?

Question	BeforeLecture Group A	BeforeLecture Group B	AfterLecture Group A	AfterLecture Group B
Q1	A2 (with) ✓	A2 (without) ✓	A3 (with) ✓	A3 (without) ✓
Q2	E1 (without) ✓	E1 (with) ✓	B4 (without) ✓	B4 (with) ✓

Hypothesis	Experiment	Question
H1	BeforeLecture	Q1
H2	BeforeLecture	Q2
H3	BeforeLecture	Q1&Q2
H4	Before&AfterLecture	Q1&Q2
H5	AfterLecture	Q1
H6	AfterLecture	Q2
H7	AfterLecture	Q1&Q2

HN: {Before learning, After learning, Combining both experiments}, given {Q1/Q2/Q1 & Q2}, there is no difference in terms of correctness, time, and effort between participants working on code with LAs and participants working on code without LAs.

Hypotheses

RQ2: How does knowledge of LAs affect developers' code understanding?

Question	BeforeLecture Group A	BeforeLecture Group B	AfterLecture Group A	AfterLecture Group B
Q1	A2 (with) ✓	A2 (without)	A3 (with) ✓	A3 (without)
Q2	E1 (without)	E1 (with) ✓	B4 (without)	B4 (with) ✓

Hypothesis	Experiment	Question
H8	Before&AfterLecture	Q1&Q2

H8: Given Q1 & Q2, there is no difference in terms of correctness, time, and effort between participants having knowledge of LA and participants not having knowledge of LA

Data Analysis

- Datasets: PM, CU, and Combined dataset
- Normality testing (Shapiro-Wilks' test)
  - ☐ Non-normal: non-parametric
  - ☑ Normal: parametric
- Unequal Sample sizes
- Statistical significance testing (Mann Whitney U test)
- Effect size (Cliff's delta)
- Direction of significance

Results

First Set of RQs

RQ1: How do LAs affect developers' program comprehension?

Negatively Positively

Correctness

Time

		Correctness	Time	Effort	No. Responses
H1	Concordia	0.12	0.30	0.16	100
	Poly	0.37	**0.000002**	0.527	129
	Effect Size	-	0.49	-	-
	Combined	**0.017**	**0.005**	0.29	229
	Effect Size	-0.15	0.21	-	-
		Correctness	**Time**	**Effort**	**No. Responses**
H2	Concordia	0.70	0.45	**0.042**	100
	Effect Size	-	-	0.22	-
	Poly	**0.01**	0.63	0.92	129
	Effect Size	-0.17	-	-	-
	Combined	**0.048**	0.138	**0.058**	229
	Effect Size	-0.11	-	0.13	-
		Correctness	**Time**	**Effort**	**No. Responses**
H3	Concordia	0.392	0.8431	0.6615	200
	Poly	**0.0004**	**0.000009**	0.73	258
	Effect Size	-0.21	0.32	-	-
	Combined	**0.001**	**0.002**	0.60	458
	Effect Size	-0.48	0.16	-	-
		Correctness	**Time**	**Effort**	**No. Responses**
H4	Concordia	0.63	0.92	0.41	418
	Poly	**0.056**	**3.72e-07**	0.14	394
	Effect Size	-0.082	0.28	-	-
	Combined	**0.01**	**0.02**	0.5733	812
	Effect Size	-0.07	0.09	-	-
		Correctness	**Time**	**Effort**	**No. Responses**
H5	Concordia	0.827	0.7376	0.29	109
	Poly	0.22	0.27	0.11	68
	Combined	0.36	0.29	0.29	177
		Correctness	**Time**	**Effort**	**No. Responses**
H6	Concordia	0.45	0.39	0.96	109
	Poly	0.17	0.68	0.19	68
	Combined	0.90	0.16	0.13	177
		Correctness	**Time**	**Effort**	**No. Responses**
H7	Concordia	0.47	0.95	0.48	218
	Poly	1	0.82	0.84	136
	Combined	0.55	0.88	0.76	354

First Set of RQs

RQ1: How do LAs affect developers' program comprehension?

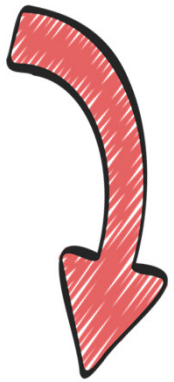
Negatively Positively

Correctness

Time

		Correctness	Time	Effort	No. Responses
H1	Concordia	0.12	0.30	0.16	100
	Poly	0.37	**0.000002**	0.527	129
	Effect Size	-	0.49	-	-
	Combined	**0.017**	**0.005**	0.29	229
	Effect Size	-0.15	0.21	-	-
		Correctness	**Time**	**Effort**	**No. Responses**
H2	Concordia	0.70	0.45	**0.042**	100
	Effect Size	-	-	0.22	-
	Poly	**0.01**	0.63	0.92	129
	Effect Size	-0.17	-	-	-
	Combined	**0.048**	0.138	**0.058**	229
	Effect Size	-0.11	-	0.13	-
		Correctness	**Time**	**Effort**	**No. Responses**
H3	Concordia	0.392	0.8431	0.6615	200
	Poly	**0.0004**	**0.000009**	0.73	258
	Effect Size	-0.21	0.32	-	-
	Combined	**0.001**	**0.002**	0.60	458
	Effect Size	-0.48	0.16	-	-
		Correctness	**Time**	**Effort**	**No. Responses**
H4	Concordia	0.63	0.92	0.41	418
	Poly	**0.056**	**3.72e-07**	0.14	394
	Effect Size	-0.082	0.28	-	-
	Combined	**0.01**	**0.02**	0.5733	812
	Effect Size	-0.07	0.09	-	-
		Correctness	**Time**	**Effort**	**No. Responses**
H5	Concordia	0.827	0.7376	0.29	109
	Poly	0.22	0.27	0.11	68
	Combined	0.36	0.29	0.29	177
		Correctness	**Time**	**Effort**	**No. Responses**
H6	Concordia	0.45	0.39	0.96	109
	Poly	0.17	0.68	0.19	68
	Combined	0.90	0.16	0.13	177
		Correctness	**Time**	**Effort**	**No. Responses**
H7	Concordia	0.47	0.95	0.48	218
	Poly	1	0.82	0.84	136
	Combined	0.55	0.88	0.76	354

Significant



Insignificant

First Set of RQs

RQ2: How does knowledge of LAs affect developers' program comprehension?

Negatively Positively

Correctness

		Correctness	Time	Effort	No. Responses
H8	Concordia	**4.67e-13**	**1.23e-04**	**0.04698**	209
	Effect Size	-0.42	-0.30	-0.15	-
	Poly	**3.73e-13**	**0.003**	0.4131	197
	Effect Size	-0.52	0.25	-	-
	Combined	**< 2.2e-16**	0.1704	**0.04**	406
	Effect Size	-0.48	-	-0.10	-

Second Set of RQs

RQ3: What is the recognition rate of LAs by unknowledgeable participants?

LAs	Impact		Recognition
	Significance	Effect size	Correctness
BeforeLecture			
A2	Yes	Small	35.2%
E1	Yes	Negligible	63.63%
AfterLecture			
A3	No	∅	94.8%
B4	No	∅	97.59%

Second Set of RQs

RQ4: What is the recognition rate of LAs by knowledgeable participants?

LAs	Impact		Recognition
	Significance	Effect size	Correctness
BeforeLecture			
A2	Yes	Small	35.2%
E1	Yes	Negligible	63.63%
AfterLecture			
A3	No	∅	94.8%
B4	No	∅	97.59%

First and Second Sets of RQs - Knowledge of LAs

LAs	Impact		Recognition
	Significance	Effect size	Correctness
	BeforeLecture		
A2	Yes	Small	35.2%
E1	Yes	Negligible	63.63%
	AfterLecture		
A3	No	∅	94.8%
B4	No	∅	97.59%

First and Second Sets of RQs - Conclusion

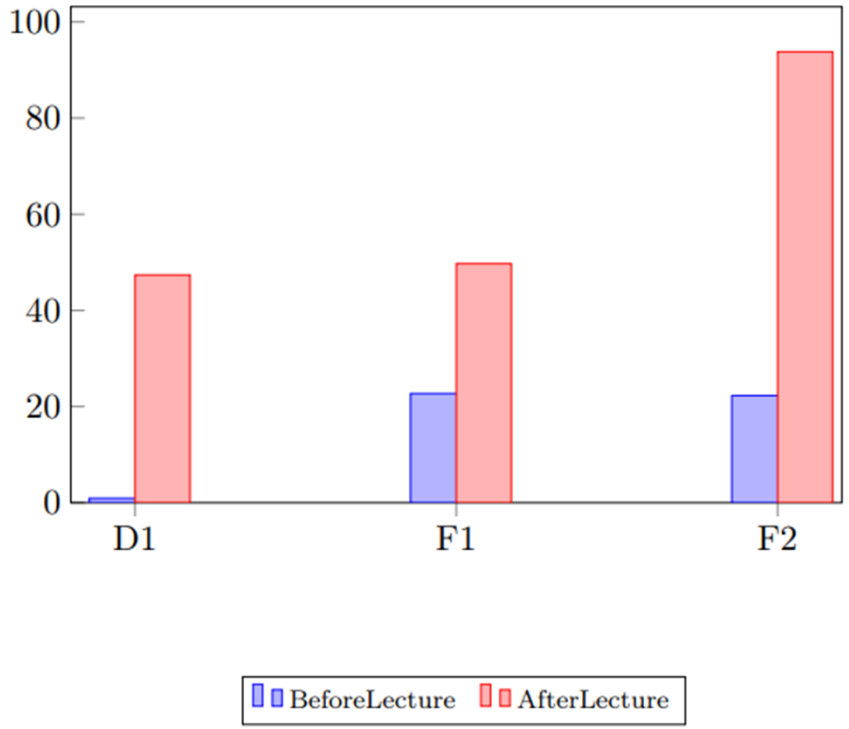
- H0: LAs have a small impact on program comprehension because they are easily recognisable by developers
- Results support H0
- LAs may not truly function as antipatterns
- Learning LAs mitigates the impact on Program Comprehension by improving their recognition rate

Discussion

Extended Analysis

- Recognition rates of D1, F1, and F2

Type	Correctness_Before (%)	Correctness_After (%)
D1	0.9%	47.36%
F1	22.7%	49.7%
F2	22.2%	93.7%



Mitigating Factors

- Age (18-25, 26-35, and 36-45)
- Geder (male, female)
- English Proficiency (high and low level)
- Programming Knowledge (good and bad)
- Degree (Bachelors, Masters, PhDs, and others)
- Work Experience (0-5 , 5-10 , and more than 10)

Mitigating Factors

- **Age (18-25, 26-35, and 36-45)**
- Geder (male, female)
- English Proficiency (high and low level)
- Programming Knowledge (good and bad)
- **Degree (Bachelors, Masters, PhDs, and others)**
- **Work Experience (0-5 , 5-10 , and more than 10)**

→ **Bonferroni Correction**

Mitigating Factors

- **Age (18-25, 26-35, and 36-45)**
- **Geder (male, female)**
- **English Proficiency (high and low level)**
- Programming Knowledge (good and bad)
- Degree (Bachelors, Masters, PhDs, and others)
- Work Experience (0-5 , 5-10 , and more than 10)

→ **No Significant result**

Mitigating Factors

- Age (18-25, 26-35, and 36-45)
- Geder (male, female)
- English Proficiency (high and low level)
- **Programming Knowledge (good and bad)**
- Degree (Bachelors, Masters, PhDs, and others)
- Work Experience (0-5 , 5-10 , and more than 10)

Correctness

- Q3 with D1

Time

- Q1 with A2, A3

Effort

- All questions

Mitigating Factors

- Age (18-25, 26-35, and 36-45)
- Geder (male, female)
- English Proficiency (high and low level)
- Programming Knowledge (good and bad)
- **Degree (Bachelors, Masters, PhDs, and others)**
- Work Experience (0-5 , 5-10 , and more than 10)

Correctness

- Q3 Bachelors & others
- Q3 Masters & others
- Q3 with D1

Time

- Q2 Bachelors & others
- Q2 Masters & others
- Q2 with E1, B4

Effort

- All questions, Bachelors & others
- All questions, Masters & others

Mitigating Factors

- Age (18-25, 26-35, and 36-45)
- Geder (male, female)
- English Proficiency (high and low level)
- Programming Knowledge (good and bad)
- Degree (Bachelors, Masters, PhDs, and others)
- **Work Experience (0-5 , 5-10 , and more than 10)**

Correctness

- No significant result

Time

- No significant result

Effort

- Q1, 0 to 5 and >10

Threats to Validity

Construct Validity

- Subjectivity in TLX
- Participant's experience
- Time recording Accuracy
- Missed objective in Quiz
- PMD precision
- Correctness accuracy

Internal Validity

- Practical Knowledge
- Diffusion
- Participants selection
- Learning with experiment
- Participants Variability
- Participants Performance

Conclusion Validity

- Correctness of results due to limited code access

Reliability Threat

- possibility of replicating this study

External Validity

- generalizability of the results

Conclusion

- LAs impact program comprehension but with small or negligible effect size when the LAs are recognised
- Learning LAs can reduce their negative impact by increasing their recognition rate
- Beneficial to increase the awareness of LAs among students
- LAs may not effectively function as antipatterns

Future Work

- Investigate the impact of other types of LAs on program comprehension
- Other participants specially from industry
- Other programming languages