Cognitive Biases in Architectural Decision-Making: Impact and debiasing strategies

AUTHOR : KLARA BOROWA

SUPERVISOR: ANDRZEJ ZALEWSKI

WARSAW UNIVERSITY OF TECHNOLOGY, INSTITUTE OF CONTROL AND COMPUTATION ENGINEERING

Agenda

Introduction

- Software architecture
- Cognitive biases
- Main Research Questions
- Research outline (published papers)

What rationales drive architectural decisions?

Cognitive biases in architectural decision-making

The impact of cognitive biases on architectural technical debt

Debiasing

- Pilot study
- Debiasing students
- Debiasing experts

Thesis contributions

Introduction

Software architecture can be defined as:

A set of the following elements (Perry and Wolf, 1992):

- Processing elements;
- Data elements, and
- Connecting elements

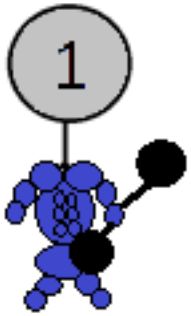
A set of structures needed to reason about the system (Bass et al., 2003)

A set of design decisions (Jansen and Bosch, 2005).

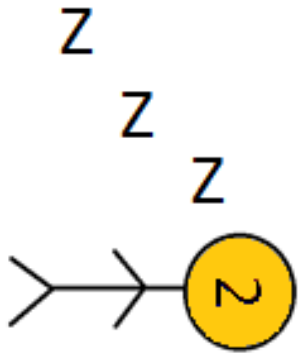
Cognitive biases: definition

Predictable errors in the ways that individuals interpret information and make decisions (Kahneman and Renshon, 2009).

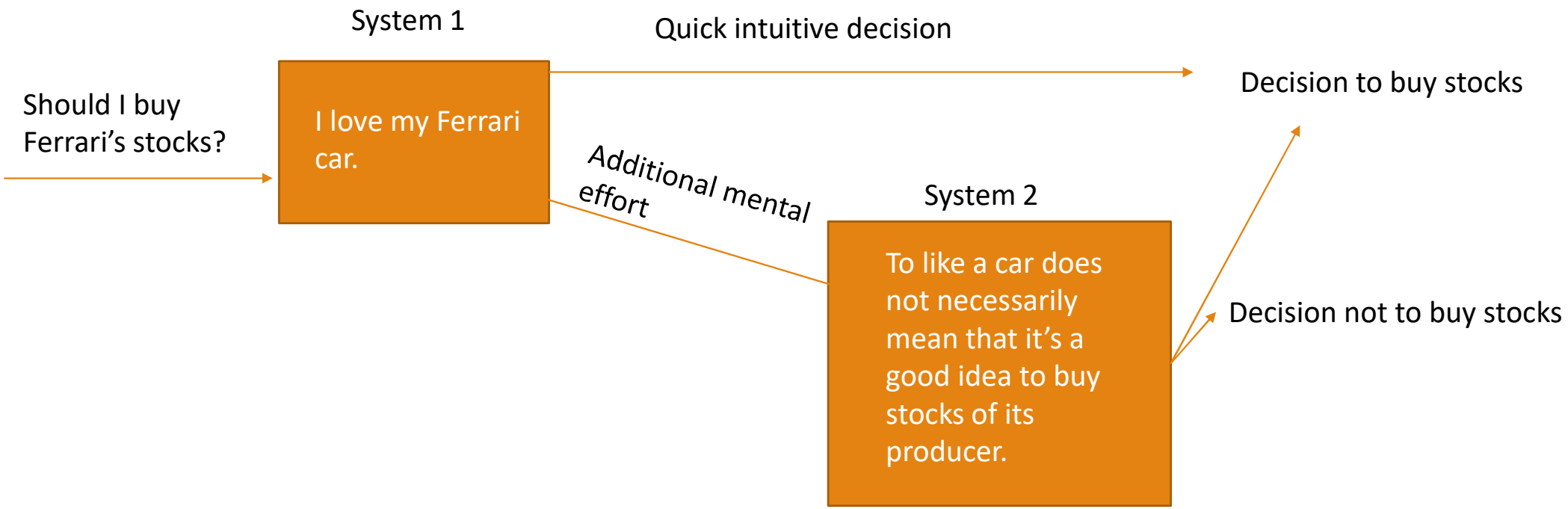
The dual process theory



System 1 (Fast)	System 2 (Slow)
Unconscious Reasoning	Conscious Reasoning
Low Effort	High Effort
Large Capacity	Small Capacity
Fast	Slow
Associative	Rule-Based
Non-Logical	Logical



How cognitive biases happen?



Main Research Questions

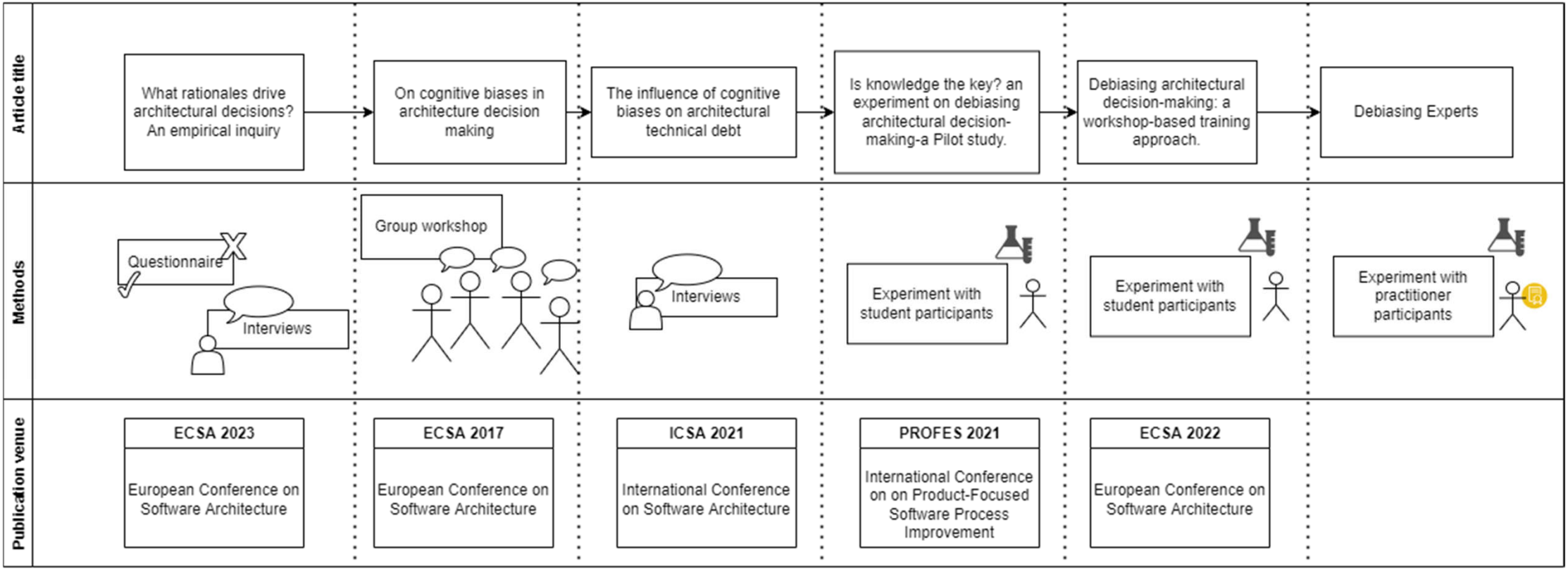
MRQ1: What rationales are the main reasons behind decisions impacting software practitioner's architectural decision-making?

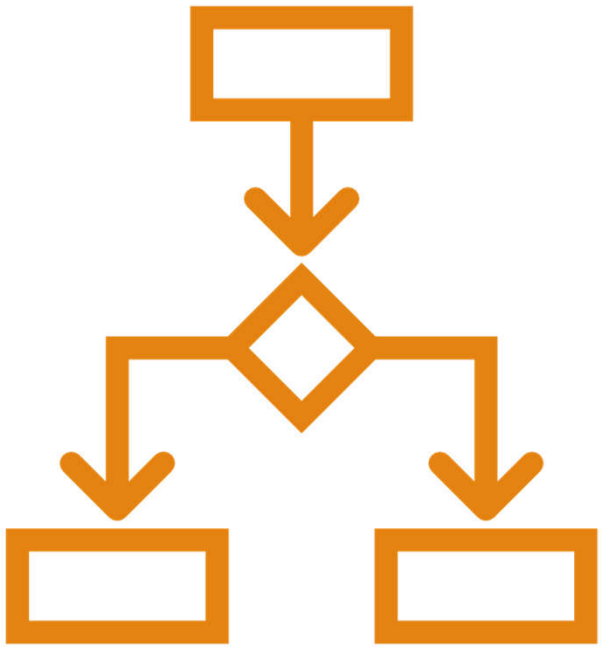
MRQ2: How do cognitive biases impact architectural decision-making?

MRQ3: Does cognitive biases' impact on architectural decision-making cause architectural technical debt?

MRQ4: How can the negative impact of cognitive biases on architectural decision-making be alleviated?

Research outline (published papers)





What rationales drive architectural decisions? An empirical inquiry

Based on:

Borowa, K., Lewanczyk, R., Stpiczyńska, K., Stradomski, P., & Zalewski, A. (2023, September).
What rationales drive architectural decisions? An empirical inquiry.
In *European Conference on Software Architecture* (pp. 303-318). Cham: Springer Nature Switzerland

Motivation

Software architecture is a set of design decisions

- Jansen, A., & Bosch, J. (2005, November). Software architecture as a set of architectural design decisions. *WICSA'05*

Design rationale = knowledge and reasoning justifying design decisions

- Tang, A., Babar, M. A., Gorton, I., & Han, J. (2006). A survey of architecture design rationale. *Journal of systems and software*

There are numerous guidelines on how to make architectural decisions

- Tang, A., & Kazman, R. (2021). Decision-making principles for better software design decisions. *IEEE Software*

Research on factors impacting architectural decision making is limited

- Tang, A., Babar, M. A., Gorton, I., & Han, J. (2006). A survey of architecture design rationale. *Journal of systems and software*
- Miesbauer, C., & Weinreich, R. (2013). Classification of design decisions–an expert survey in practice. *ECSA 2013*
- Weinreich, R., Groher, I., & Miesbauer, C. (2015). An expert survey on kinds, influence factors and documentation of design decisions in practice. *Future Generation Computer Systems*

Sub-Research Questions

What rationales?

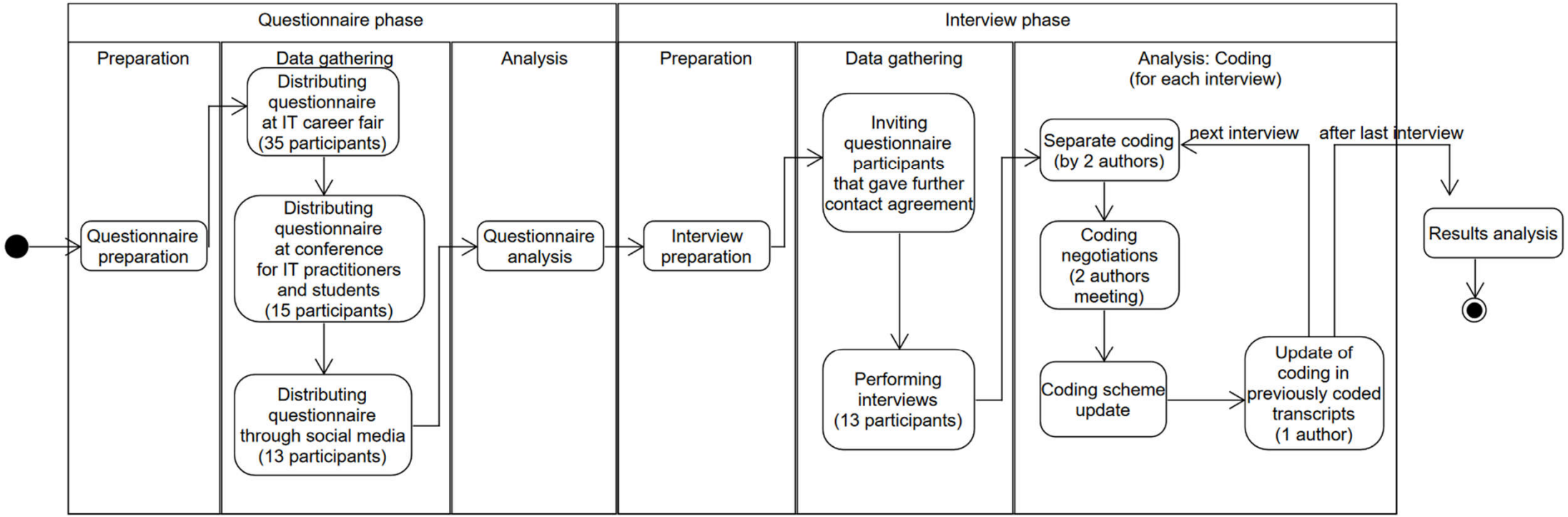
SRQ1: What rationales most frequently influence architectural decisions?

SRQ2: Which software quality attributes are usually prioritised during architectural decision-making?

Why?

SRQ3: Why do practitioners prioritise these rationales?

Methods


Questionnaire phase
Interview phase
Preparation
Data gathering
Analysis
Preparation
Data gathering
Analysis: Coding
(for each interview)
Questionnaire preparation
Distributing questionnaire at IT career fair (35 participants)
Distributing questionnaire at conference for IT practitioners and students (15 participants)
Distributing questionnaire through social media (13 participants)
Questionnaire analysis
Interview preparation
Inviting questionnaire participants that gave further contact agreement
Performing interviews (13 participants)
Separate coding (by 2 authors)
next interview
after last interview
Coding negotiations (2 authors meeting)
Coding scheme update
Update of coding in previously coded transcripts (1 author)
Results analysis

Questionnaire

- Participant data
  - (age, gender, education, years of experience etc.)

- **What 3 factors do you consider** when making architectural decisions?
- **What 3 factors do your colleagues** consider when making architectural decisions?

- Optionaly provide email for follow-up interview

Interviews

We contacted **all 13 participants** that agreed to a follow-up interview

Presented them the main findings from the questionnaire

Asked them about their interpretation of the findings

Results

Questionnaire results

				Beginners (<=4 years of experience)		Mid-career(between 5 and 14 years)		Experienced (over 15 years)	
No.	Rationale category	Sum (participants)	Sum (colleagues)	Participants	Colleagues	Participants	Colleagues	Participants	Colleagues
1	Ease of use/development	23	11	16	7	2	0	5	4
2	**Maintainability**	15	2	12	1	2	1	1	0
3	**Performance**	14	6	13	6	0	0	1	0
4	Prior knowledge/experience	14	14	11	9	1	2	2	3
5	Time/deadline	12	8	10	6	1	0	1	2
6	**Reliability**	10	4	6	3	2	1	2	0
7	Development Project Environment	9	2	4	1	3	1	2	0
..	..	...	...	...	...	...	...	...	...
13	**Usability**	5	0	3	0	2	0	0	0
14	**Security**	5	2	3	2	2	0	0	0
...	..	...	...	...	...	...	...	...	...
21	**Functional Suitability**	3	1	2	1	0	0	1	0
...	..	...	...	...	...	...	...	...	...
24	**Portability**	2	2	2	1	0	0	0	1
...	..	...	...	...	...	...	...	...	...
27	**Compatibility**	1	2	0	0	0	0	1	2
...	..	...	...	...	...	...	...	...	...
33	My colleagues use the same rationales as me	0	19	0	13	0	4	0	2

Questionnaire results

Most frequent rationales (>20% sum of participants) overall:

- "**Ease of use for development**" – mainly beginners and experienced
- Maintainability – mainly beginners
- Performance – mainly beginners
- "**Prior knowledge/experience**" - mainly beginners and experienced

Important only for the particular experience group (>20% of group but <20% of participants):

- "**Time/deadline**" – beginners
- "Development Project Environment" – mid-career
- "Decision-making methodology" – experts

Software quality attributes not considered as important factors (<5% of participants):

- Compatibility
- Portability
- Functional stability

“My colleagues have the same rationales as me”

- Beginners believe this.
- Experienced practitioners do not.

Rationales' origins

Practitioner's experience

- Beginners – afraid of the unknown: new technologies, missing deadlines
- Experienced – comfortable with their current environment, not afraid of deadlines, only ones able to use a decision-making methodology
- Mid-career – want a challenge/to learn new things, not afraid of deadlines

Client focus

- "Ease of use for development" + "Time/deadline" = fast time to market
- "Development Project Environment" – making bespoke solutions that fit the client's needs
- "Performance" – badly running software = unhappy customer

Rationales' origins

Making one's life "easy"

- Ease of use for development – easy learning
- Prior knowledge – knowledge about easy solutions
- Maintainability – easier work later

Thinking of the project's future

- Maintainability
- Ease of use for development – finding future employees

Rationales' origins

Fear of deadlines

- Mainly beginners

Familiarity with a particular solution

- Main source of architectural knowledge

"Obviousness"

- "[Functional Stability] is so mundane and part of such day-to-day work that maybe we don't tie it to the architecture."

Rationales' origins

Empathy

- "Ease of use for development" + "Maintainability" = easier work for colleagues

Personal growth

- Mainly mid-career
- "(...) **resume driven development**, i.e. we choose those technologies that will look nice in the CV, or that will make us learn something."

New technology handles the problem

- Compatibility – widespread standards
- Portability – containerization and virtualization

Rationales' origins

Practitioner's education

- Beginners care about Performance because they were trained so at college

Perception of the QA as unimportant

- "Portability" and "Compaibility" not important when targeting specific platforms

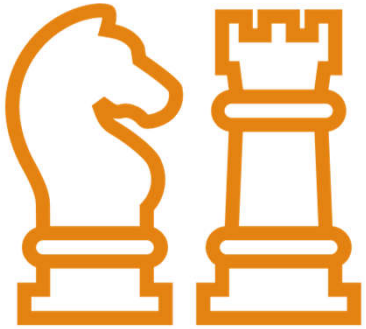
Conclusion

Main findings:

- List of rationales that drive architectural decision-making
- An exploration of the rationales' origins
- Identifying impact of experience level on rationale behind decisions

Implications:

- Understanding how practitioners make decisions
- The major role of experience on decision-making
- **Three common rationales are possibly cognitive bias antecedents:**
  - **Ease of use for development**
  - **"Prior knowledge/experience"**
  - **"Time/deadline"**



On cognitive biases in architecture decision making

Based on:

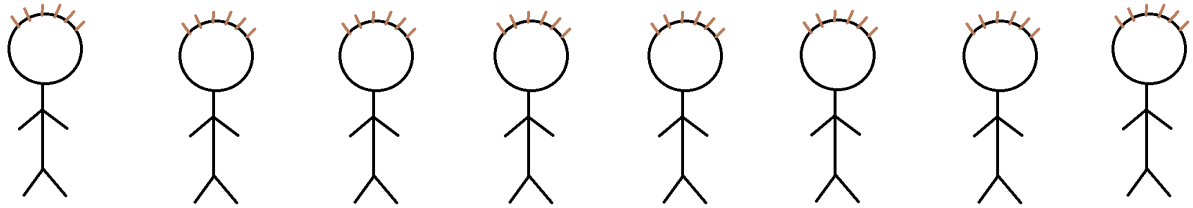
Zalewski, A., Borowa, K., & Ratkowski, A. (2017).
On cognitive biases in architecture decision making.
In Software Architecture: 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings 11 (pp. 123-137). Springer International Publishing.

Sub-Research Questions

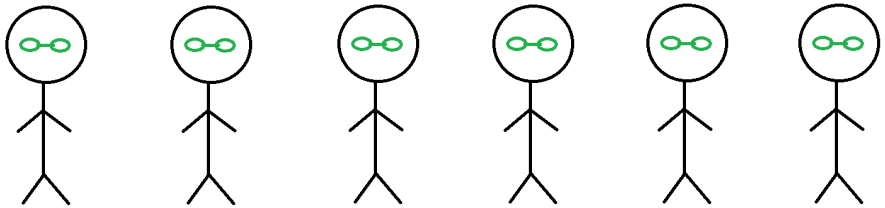
- SRQ1: Are biases in architecture decision making common?
- SRQ2: Which biases are the most significant?
- SRQ3: What exactly can bias architects' decisions?

The Workshop - Participants

8 novices (1-2 years of experience)



6 experts (10+ years of experience)



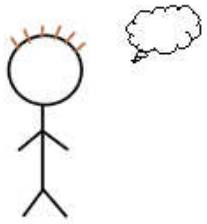
The Workshop - Agenda

- Short presentation on cognitive biases
- Writing down examples of biases that the participants encountered
- Open discussion on the biases indicated by the participants
- Rating the biases

Workshop Summary

Are biases in architecture decision making common?

- Both novices and experts in software engineering noticed biases
- Novices indicated on average 1 bias each
- Experts indicated about 4 biases each

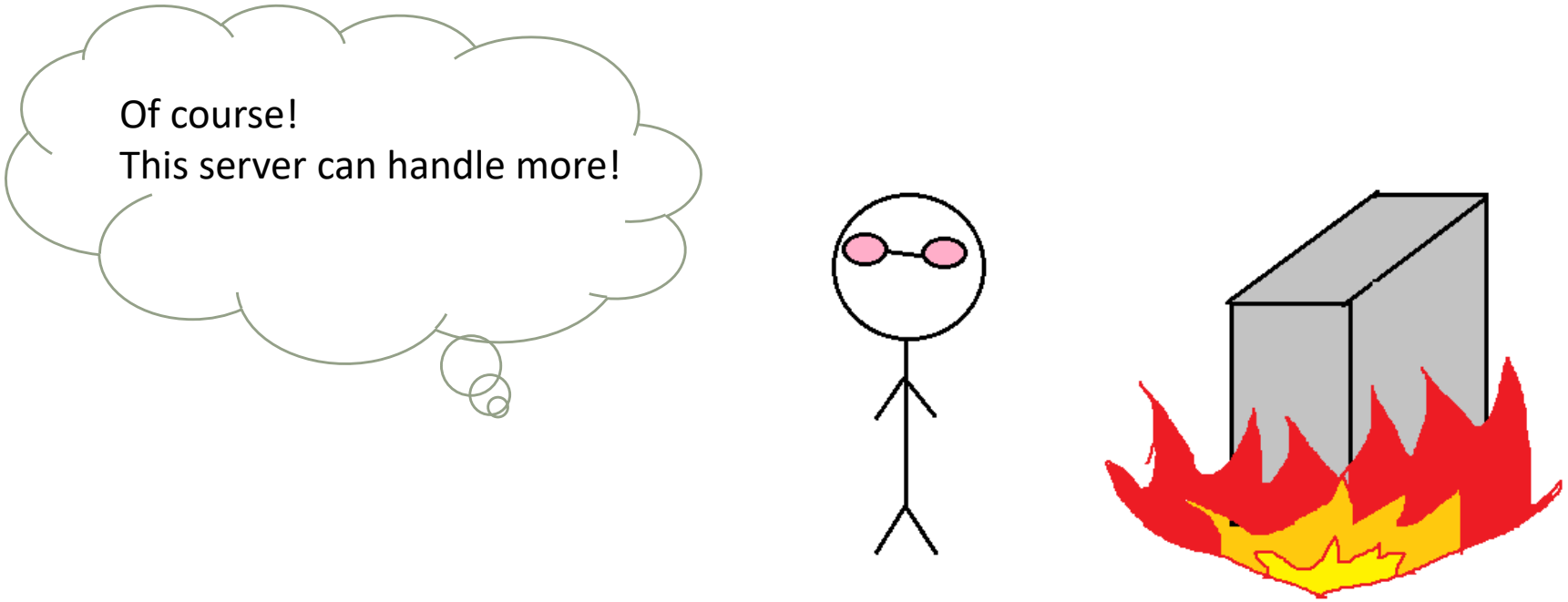


Workshop Summary

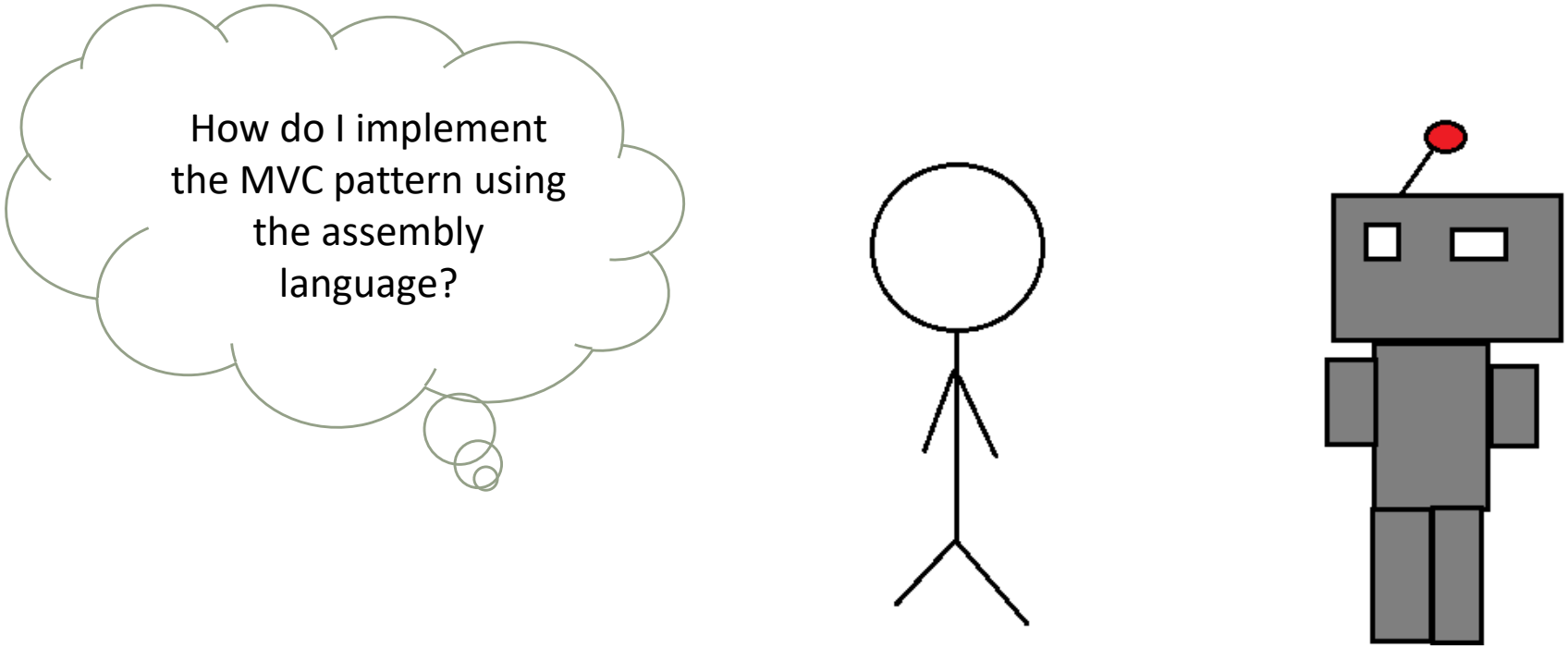
Which biases are the most significant?

Twelve biases were identified from the workshop results.

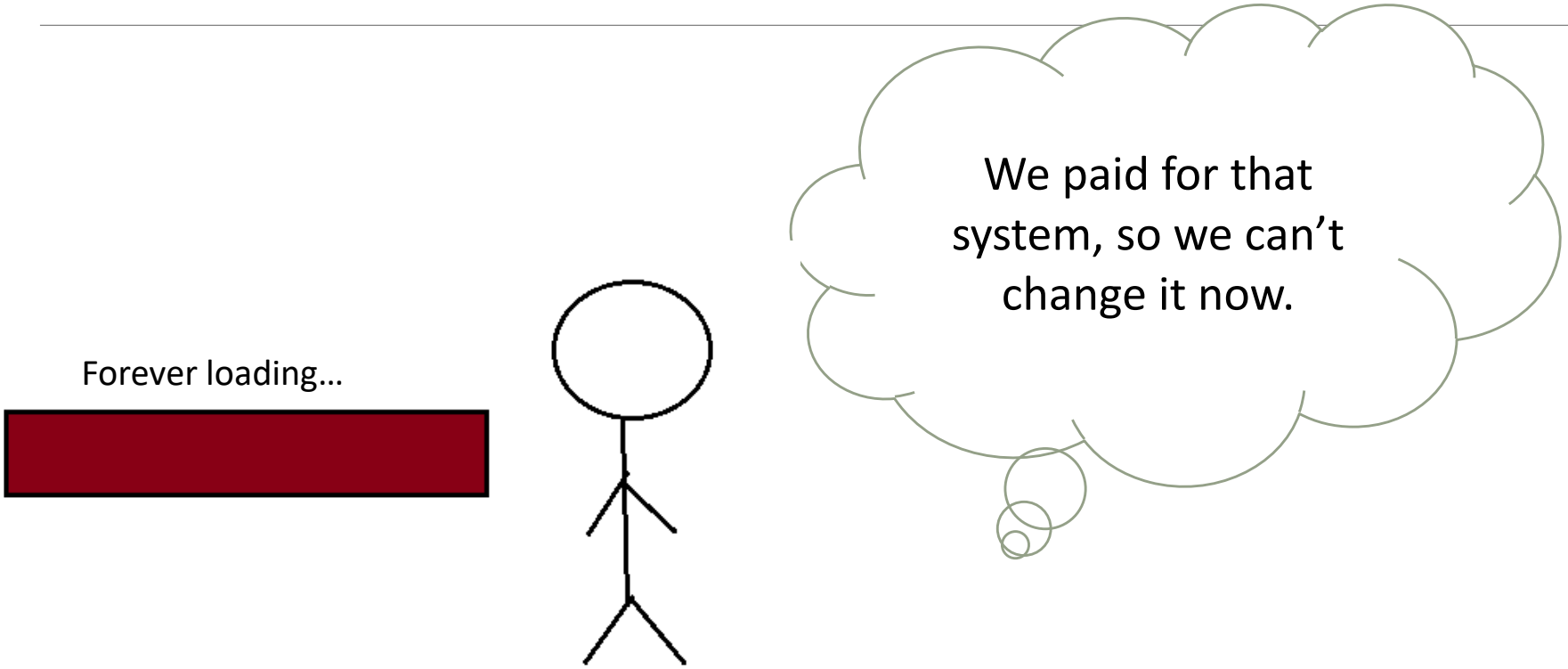
12. Optimism bias



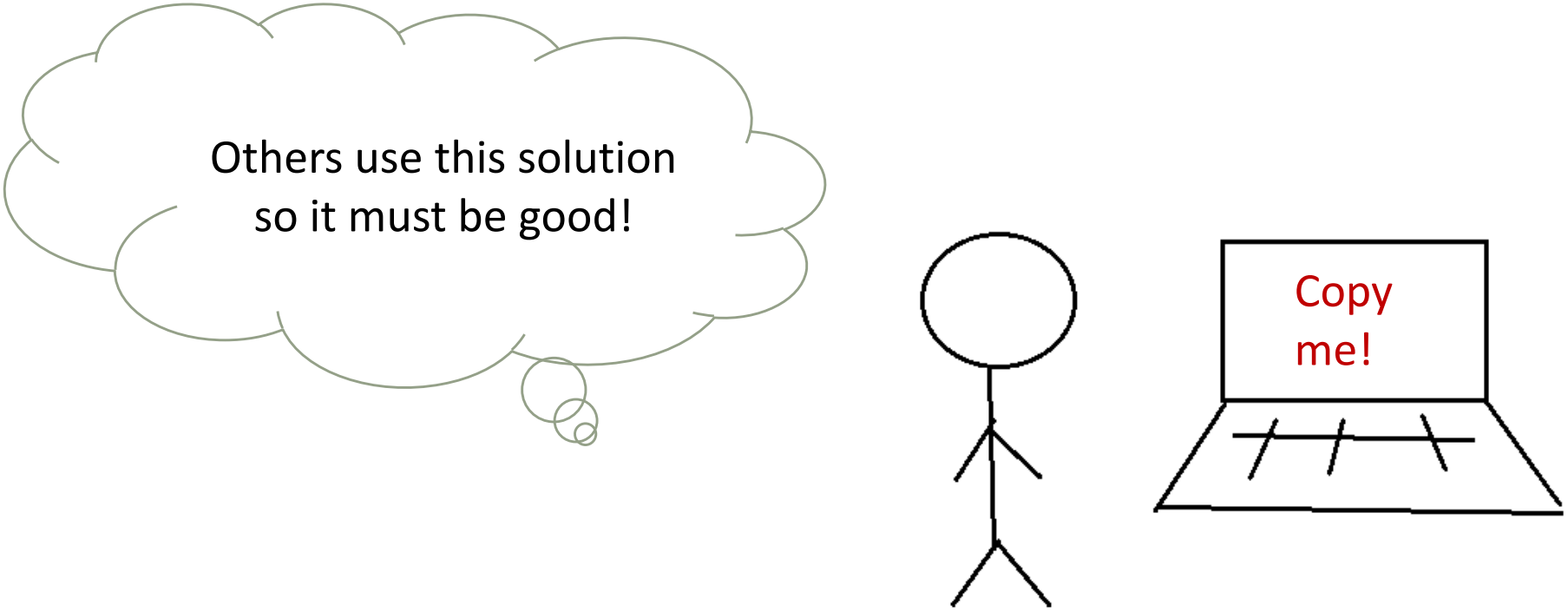
11. Law of the instrument



10. Irrational escalation



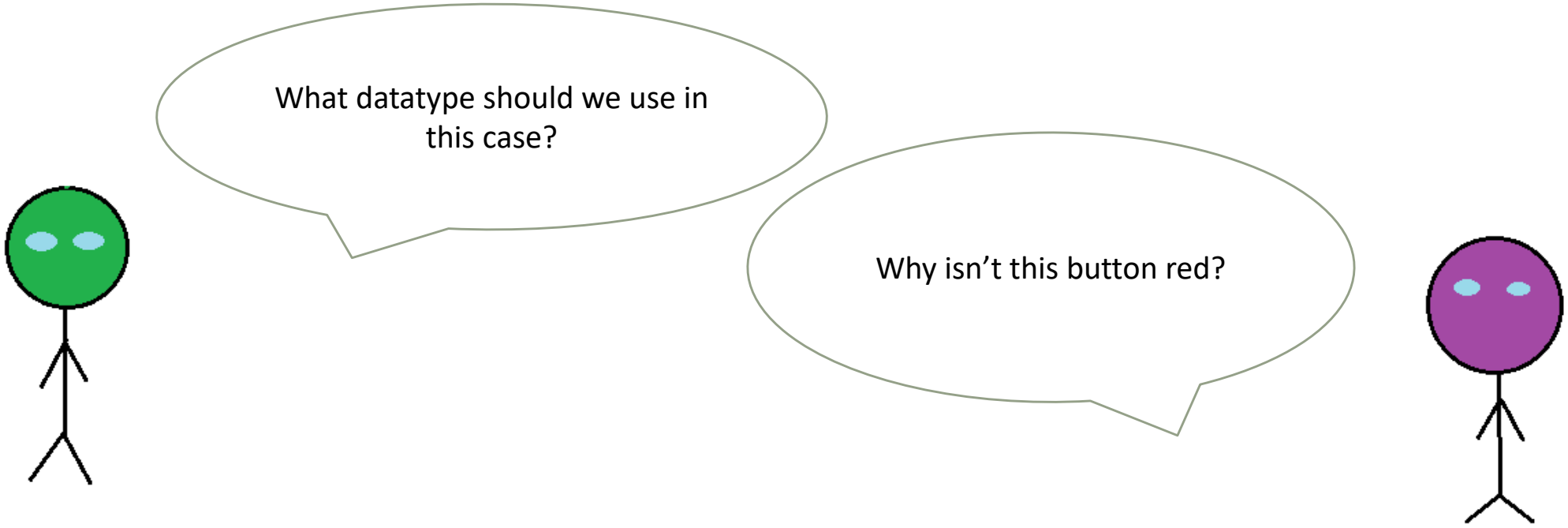
9. Bandwagon effect



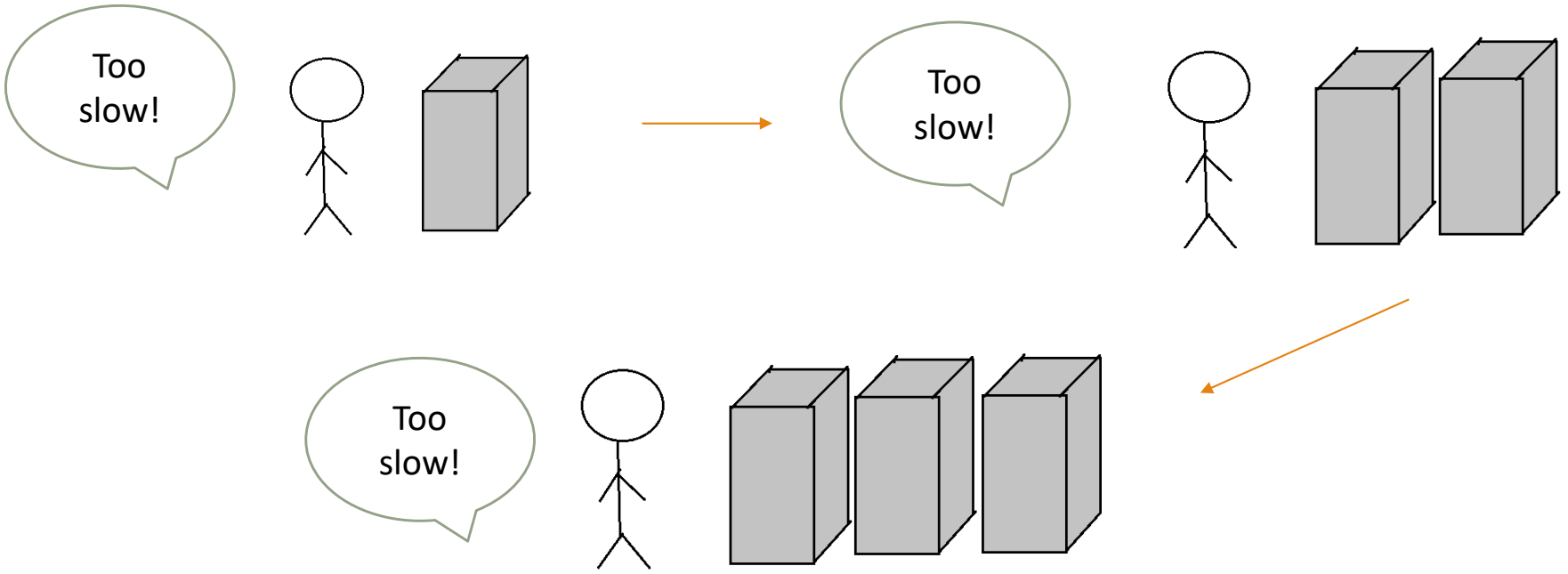
8. Planning fallacy

7. Pro-innovation bias

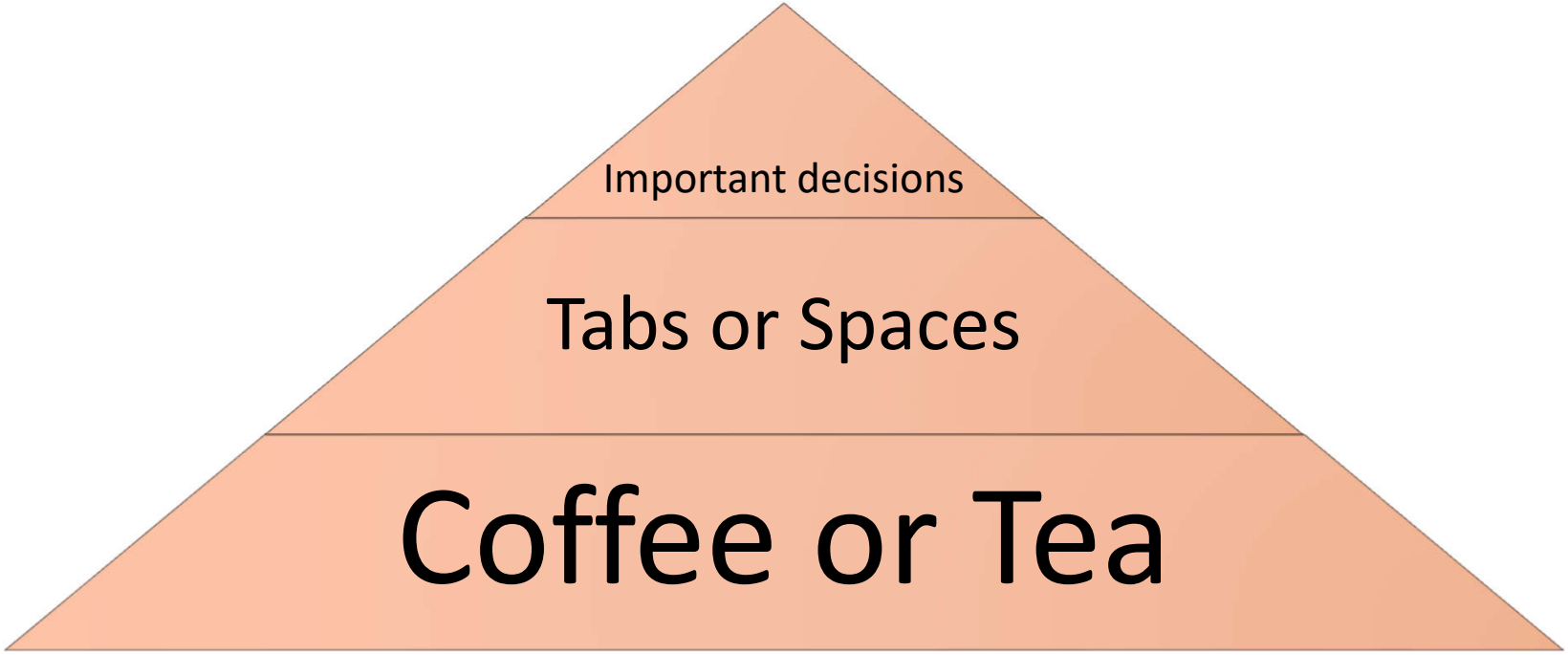
6. Curse of knowledge



5. Anchoring



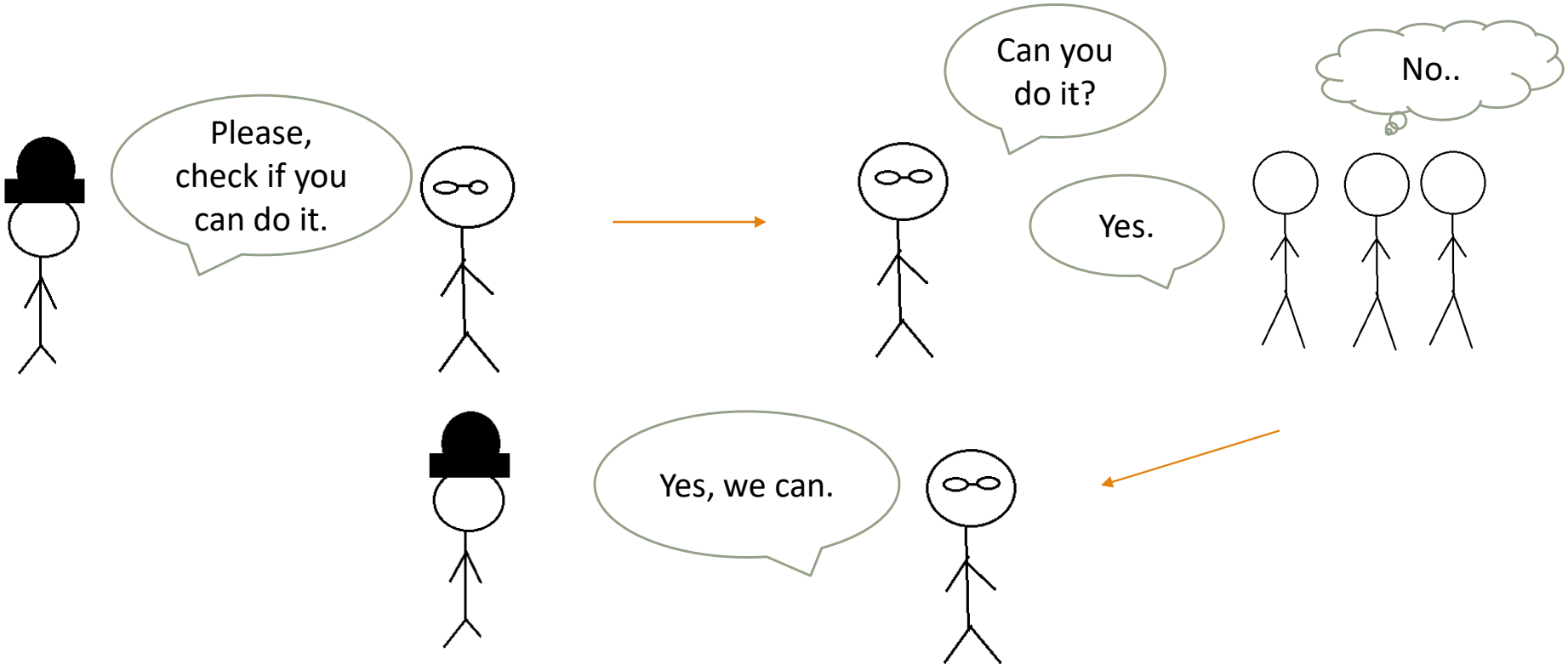
4. Parkinson's Law of triviality



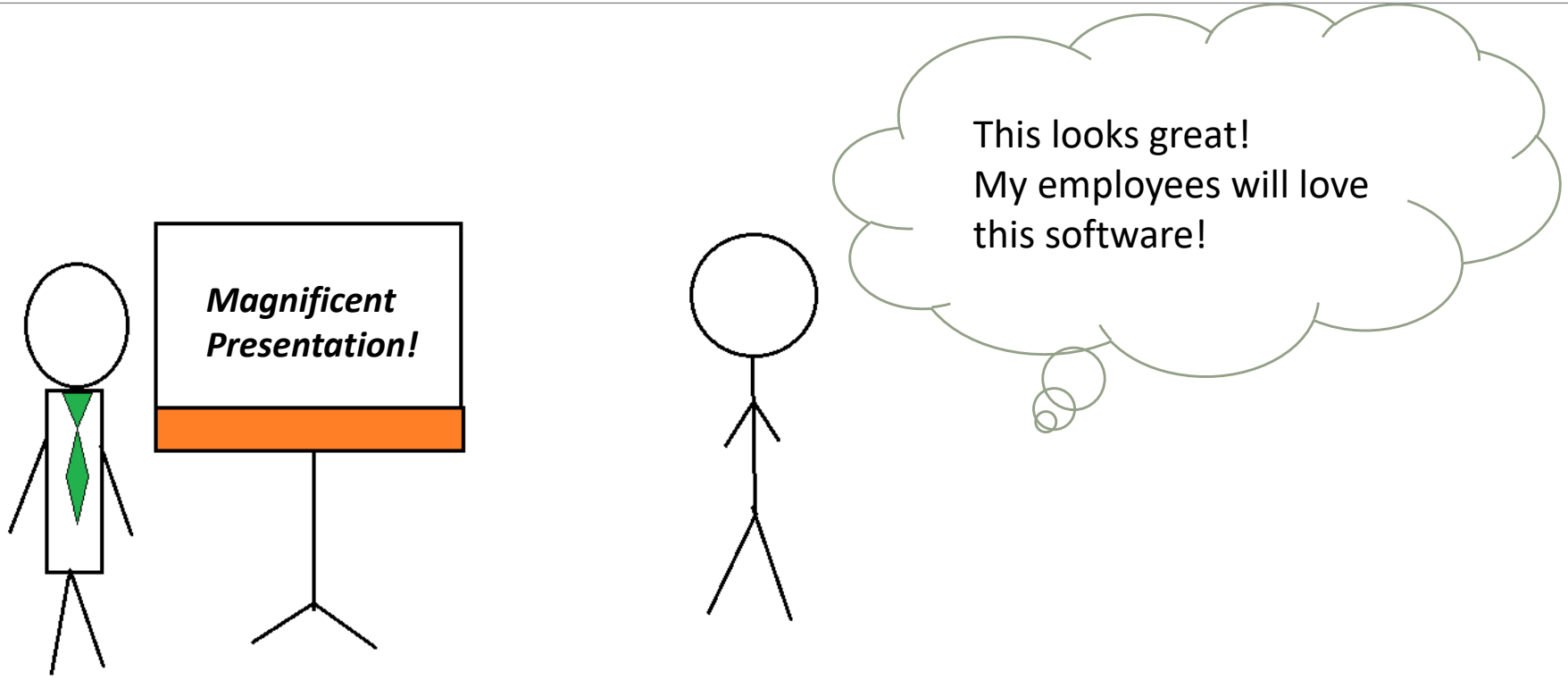
3. IKEA effect

My great program	Your awful program
Slower	Faster
Bigger memory usage	Smaller memory usage
Expensive	Cheap

2. Confirmation bias



1. Framing effect



What can bias architects' decision?

- Form of presentation
- Who was the author of a given design
- The time spent on a given design
- The order of obtaining information
- The experience and background of the stakeholders
- The architect's state of mind
- The complexity of the problem
- The existing widely-accepted solutions
- The course of action contradicting the use of an initial solution
- Architectural solutions focal for the architect

How biases influence Arch. Decision-Making (examples)

Anchoring influences:

- Scope of considered requirements
- Perception of requirements' importance
- Architect's preferences.

IKEA effect inluences:

- Scope of considered alternatives

Contributions

Approach to the analysis of biases' influence on architectural decision-making.

Factors that could make architectural decisions biased.

Key for future research: List of common biases applicable to architectural decision-making and their influence on architectural decision-making.

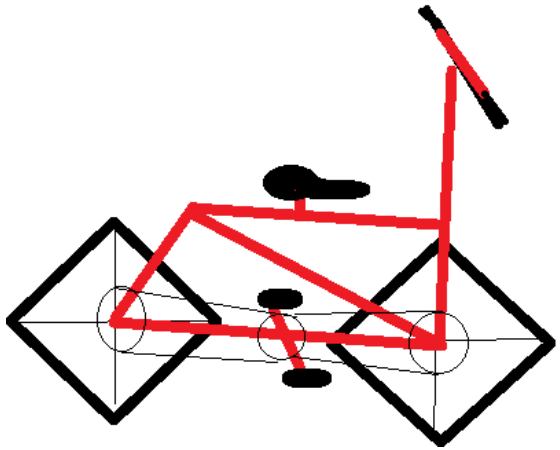
The Influence of Cognitive Biases on Architectural Technical Debt

Based on:

Borowa, K., Zalewski, A., & Kijas, S. (2021, March).
The influence of cognitive biases on architectural technical debt.
In 2021 IEEE 18th International Conference on Software Architecture (ICSA) (pp. 115-125). IEEE.

Architectural Technical Debt

- **Technical debt (TD)** is a metaphor first introduced by Cunningham in 1995 to illustrate the situation when software **quality is forfeited** as a **compromise** between quality and time-to-market.
- **Architectural technical debt (ATD)**, is the type of TD that occurs as a result of sub-optimal **architectural decisions** (Martini et al., 2014).
- The debt has to be paid back someday
  - with **interests**!



Sub-Research Questions

SRQ1: **Do cognitive biases influence** the occurrence of architectural technical debt?

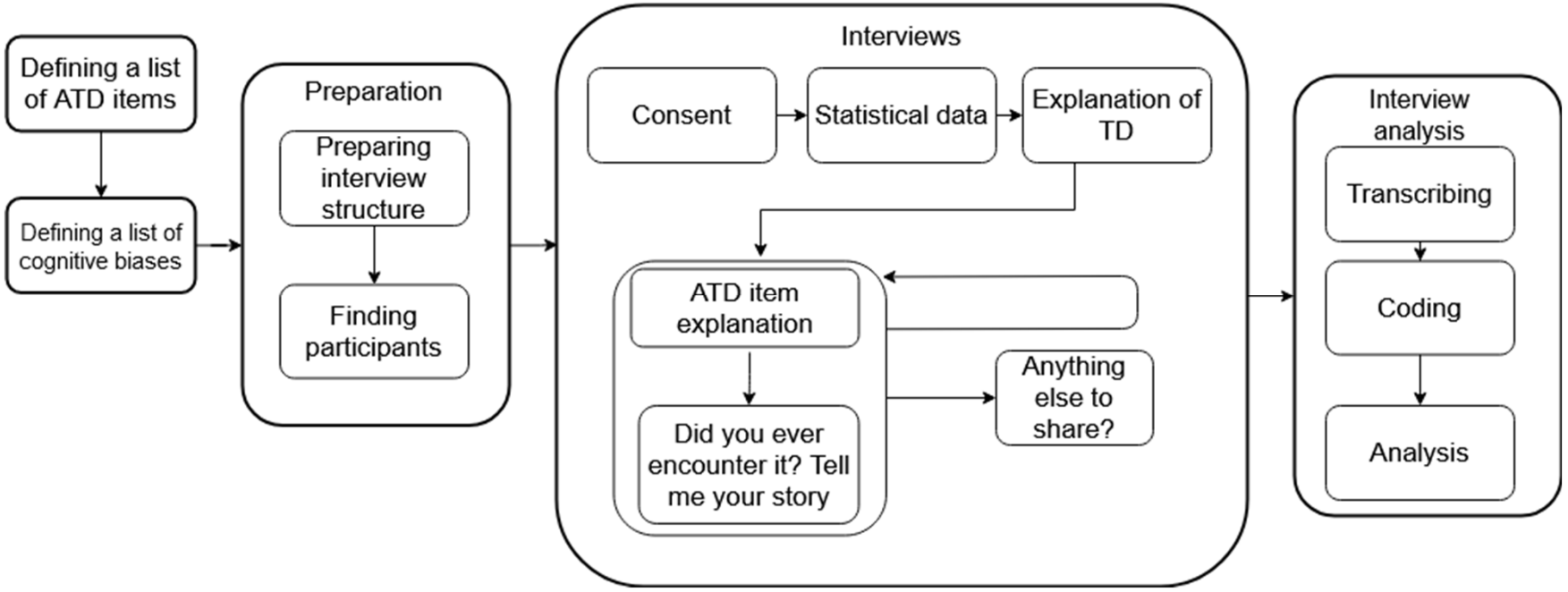
SRQ2: **Which cognitive biases** have an impact on architectural technical debt?

SRQ3: **Which architectural technical debt items** are most frequently affected by cognitive biases?

SRQ4: **What are the antecedents** of a harmful influence of cognitive biases on architectural technical debt?

SRQ5: **What debiasing techniques** can be used to minimise the negative effects of cognitive biases?

Research method



Architectural technical debt items

Types of ATD items (Verdecchia et. al. 2020):

- New Context, Old Architecture
- The Workaround that stayed
- Architectural Lock-in
- Re-inventing the Wheel
- The Minimum Viable Product that stuck
- Source Code ATD

List of cognitive biases

List based on our previous research (Zalewski et al., 2017):

- Anchoring
- Bandwagon effect
- Confirmation bias
- Curse of knowledge
- IKEA effect
- Irrational escalation
- Law of the instrument
- Optimism bias
- Parkinson's Law of triviality
- Planning fallacy
- Pro-innovation bias
- The framing effect

Participants

No.	Age	Gender	Experience (years)	Position	Company size (employees)	Company domain
1	29	M	5	Software Developer	over 10 000	Electronics
2	31	M	10	Architect	around 2 000	E-commerce
3	54	M	35	Chief Operating Officer	around 1 500	High tech
4	37	M	13	Executive consultant	around 50	Systems integrator
5	39	M	17	Head of Architects	around 350	Finance
6	49	M	26	Architect	around 350	Finance
7	37	M	16	Consultant	over 10 000	Enterprise Software
8	45	M	21	Chief of Architects	around 250	Systems integrator
9	36	M	15	Founder and Chief Technology Officer	around 35	Software
10	37	F	15	Architect	around 5 000	Telecom
11	40	M	15	Senior Solution Architect	over 10 000	Enterprise Software
12	37	M	12	Team Leader	over 10 000	Electronics

Results – ATD items

Architectural technical debt item	Appearances
New Context, Old Architecture	17
Source Code ATD	13
The Workaround that stayed	12
Architectural Lock-in	10
Re-inventing the Wheel	8
The Minimum Viable Product that stuck	6
Other (4 different types of ATD)	4

Results – Cognitive biases

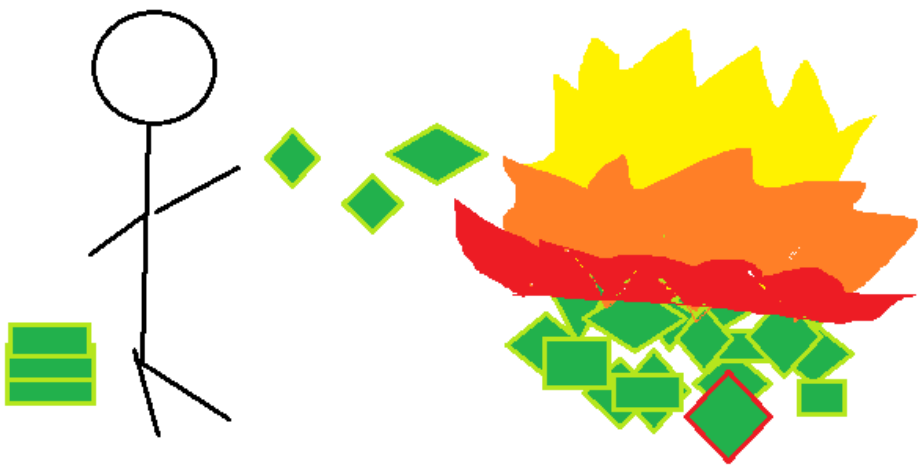
Cognitive Bias	Appearances
Anchoring	**24**
Optimism bias	**20**
Confirmation bias	**19**
Curse of knowledge	14
IKEA effect	14
Pro-innovation bias	13
Irrational escalation	11
Law of the instrument	10
Planning fallacy	10
The framing effect	10
Bandwagon effect	8
Parkinson's Law of triviality	2

Results – Biases influencing ATD

Cognitive Bias	New Context, Old Architecture	Source Code ATD	The Workaround that stayed	Architectural Lock-in	Re-inventing the Wheel	Minimum Viable Product that stuck	Other
Anchoring	**7**	**5**	4	**6**	4	1	0
Bandwagon effect	0	1	1	1	1	0	0
Confirmation bias	2	2	**5**	4	**5**	1	1
Curse of knowledge	2	2	2	4	2	0	0
IKEA effect	3	3	1	2	3	1	0
Irrational escalation	**7**	1	2	0	1	1	0
Law of the instrument	1	3	2	3	0	0	0
Optimism bias	3	3	3	**5**	2	4	1
Parkinson's Law of triviality	0	1	2	0	0	0	0
Planning fallacy	3	4	4	3	1	1	1
Pro-innovation bias	1	1	2	4	4	2	1
The framing effect	1	2	2	3	1	1	0

Frequent influence 1

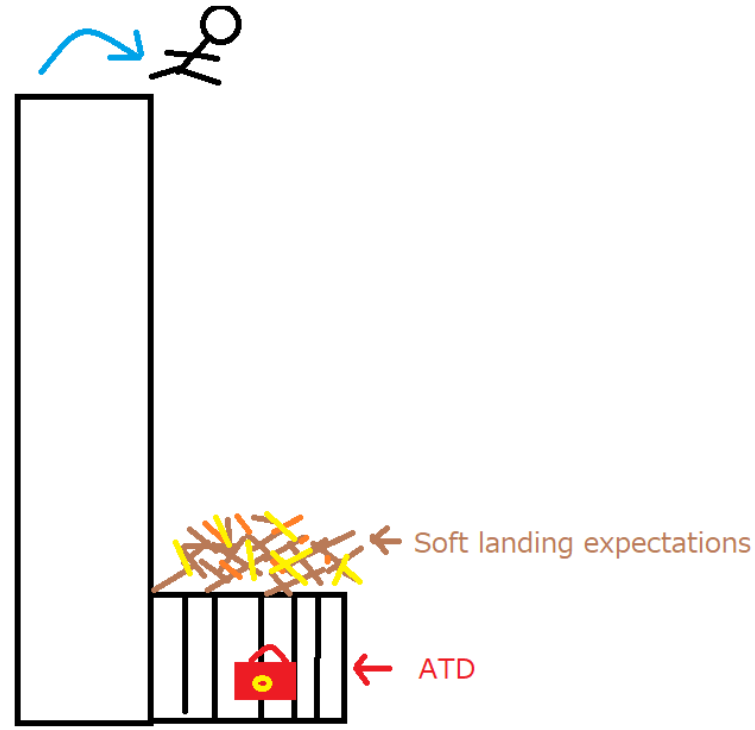
It's cold -> I have money -> Light a fire
(Anchoring on first idea)



Still cold -> burn more money
(Irrational escalation)

Frequent influence 2

The leap of faith (anchoring & optimism bias)



Cognitive bias antecedents

- Individual's emotional state (e.g. fear, shame, haste)
- Individual's personality traits (e.g. overconfident, ambitious)
- Individual's mistakes (e.g. lack of basic knowledge, not searching for alternatives)
- Organisational antecedents (e.g. overly harsh or lax organisational culture, frequent management changes, short-sighted cost/profit optimisation)
- Communicational antecedents (e.g. between specialists from different domains)
- Knowledge vaporisation (e.g. lack of documentation, employee rotation)
- External (e.g. popularity)

Debiasing methods

- **Double-check and challenge** decisions and their underlying ideas.
- Create a working environment based on **trust** – let employees can voice their opinions and admit to their mistakes.
- If something is not written down, it “does not exist”:
  - Explicitly register all accounts of TD.
  - Explicitly search for alternative solutions.
  - Document and pass on knowledge.
- Periodically check for new TD occurrences and whether old TD needs to be paid.
- Clearly define responsibilities – if everyone is responsible, then no one actually feels responsible.

Contributions

List of most frequently appearing biases.

List of most commonly occurring ATD items.

Identification of biases that most frequently influenced ATD

- Mainly but not only -> **anchoring, optimism and confirmation bias**

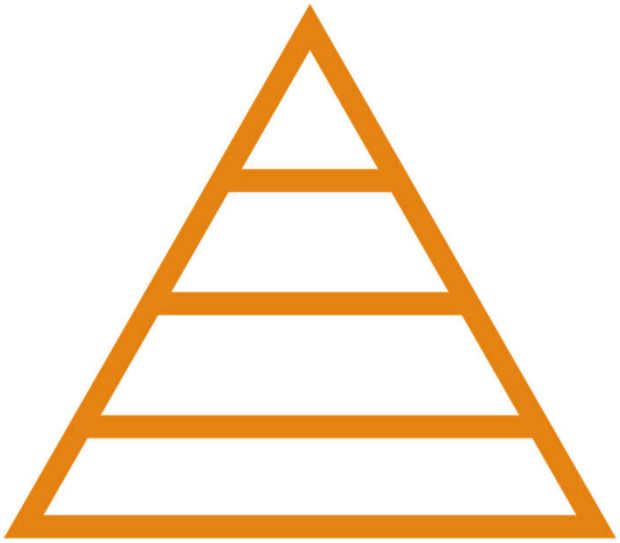
A set of common bias antecedents

Set of possible debiasing methods

Is knowledge the key? An experiment on debiasing architectural decision-making - a pilot study

Based on:

Borowa, K., Dwornik, R., & Zalewski, A. (2021).
Is knowledge the key? an experiment on debiasing architectural decision-making-a Pilot study.
In Product-Focused Software Process Improvement: 22nd International Conference, PROFES 2021, Turin, Italy, November 26, 2021, Proceedings 22 (pp. 207-214). Springer International Publishing.



Debiasing levels

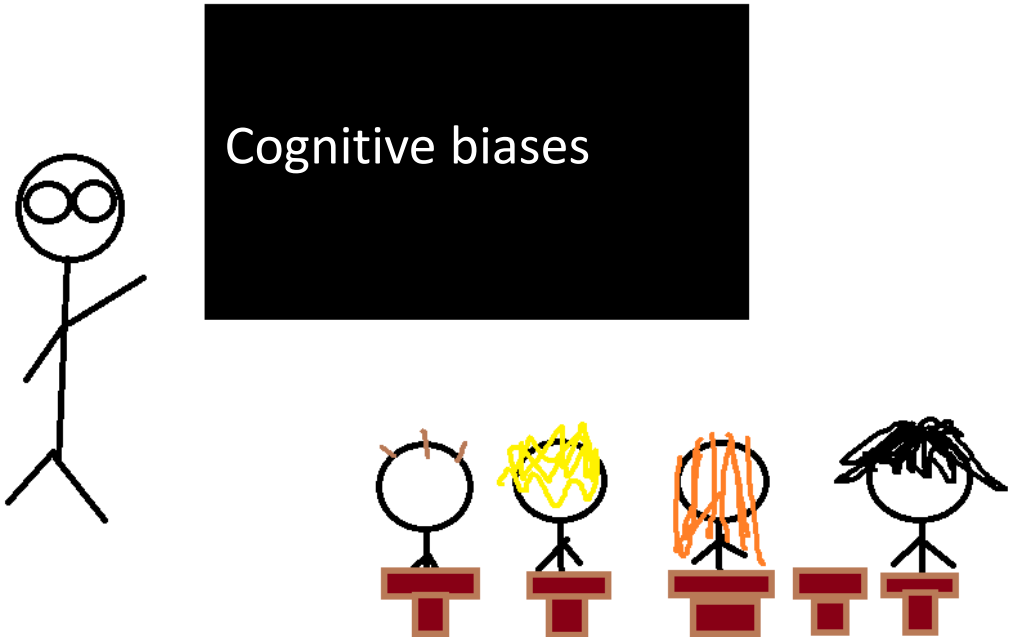
Four levels of debiasing treatments include (Fischhoff, 1982):

A. **Warning about the biases**
B. **Describing typical biases**
C. Providing personalised feedback about the biases
D. An extended programme of debiasing training

Sub-Research question

Does educating* software architects about cognitive biases provide a beneficial debiasing effect, which increases the rationality of decision-making?

***level B debiasing treatment**



Researched biases

We chose 3 biases that previous research [6] shows to be have the greatest influence on architectural technical debt.

Anchoring - when an individual over-relies on a particular solution, estimate, information or item, usually, the first one that they discovered or came up with.

Optimism bias - when baseless, overly positive estimates, assumptions and attributions are made

Confirmation bias - the tendency to avoid the search for information that may contradict one's beliefs.

Data gathering

- We took part in **four meetings** with **two groups** of graduate (masters level) students that were working on a group project during their coursework
- Topic for the project – chosen by the students (the only hard requirement being the use of Kubernetes)
- The students were supposed to gather requirements, design, implement and test their project during the semester.

The meetings proceeded as follows:

- We asked the participants for their consent to record the meeting and to use their data for the purpose of our research.
- In the case of the debiased group (Team 2), **we showed them our presentation about cognitive biases** in architectural decision-making. We did not perform this action with the other group (Team 1).
- The meeting continued naturally, without our participation, although a researcher was present and made notes when necessary.

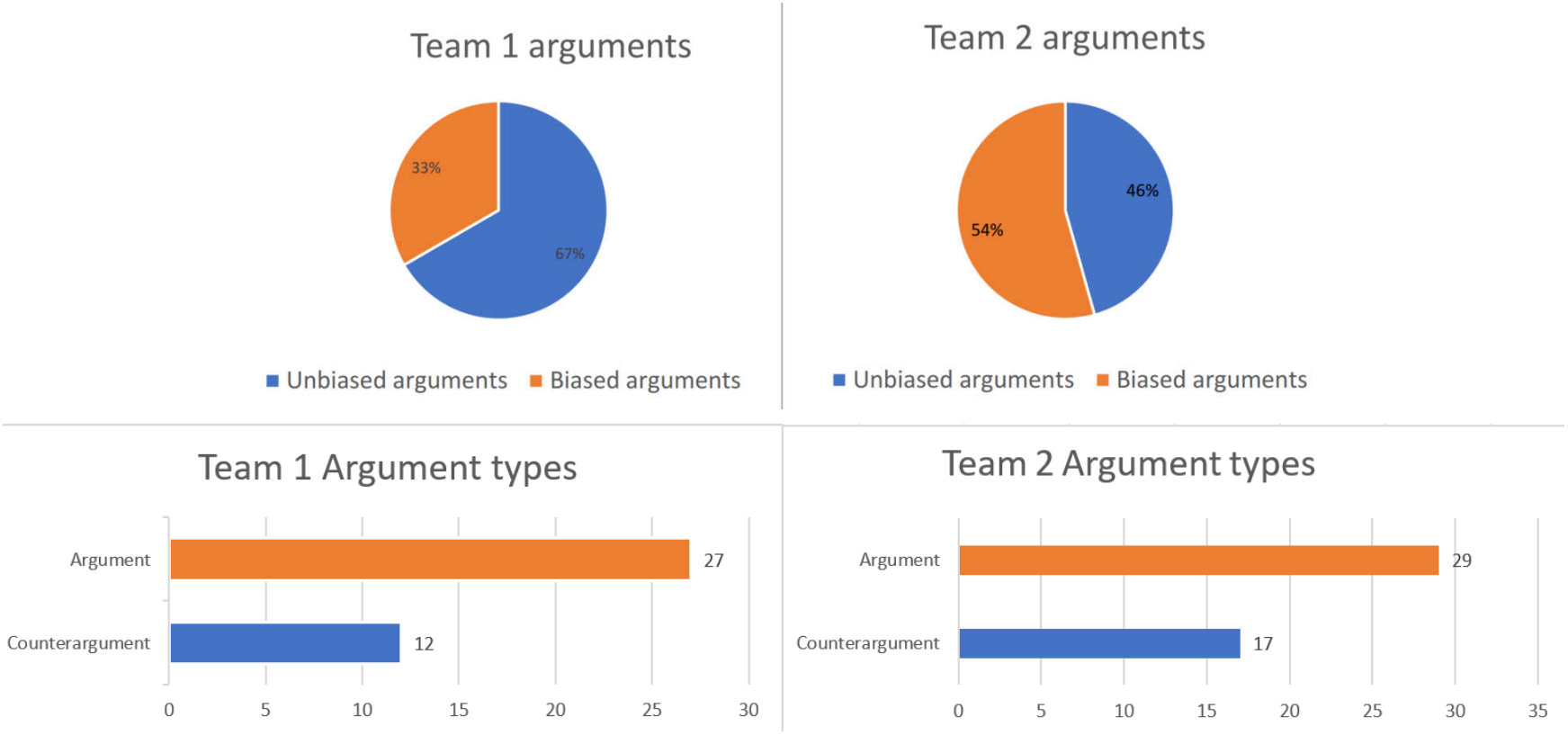
Code category	Code	Definition
Bias - anchoring	KOT	Putting too much emphasis on the first piece of information or idea that was heard/proposed/invented.
Bias - confirmation bias	POT	Not accepting and not seeking information that is inconsistent with our current beliefs.
Bias - optimism bias	OPT	Naive faith that the unpleasant consequences of our decisions will not happen.
Arguments for the decision	ARG	An argument that was in favour of choosing a particular solution.
Arguments against the decision	PARG	A counterargument, against choosing a particular solution.

Data analysis

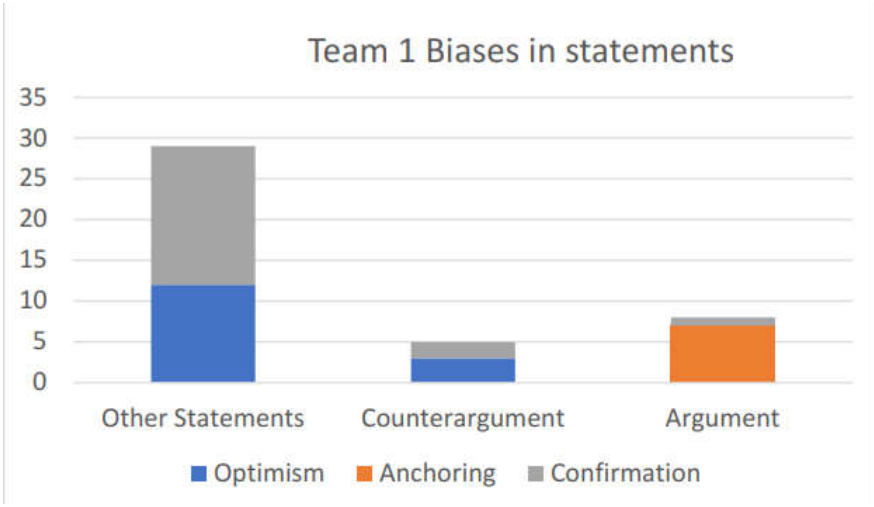
- Transcription of the meeting recordings
- Independent coding
- Negotiated coding
- Counting the occurrence of each code
- Counting the amount of instances when biases influenced arguments/counterarguments/the overall discussion

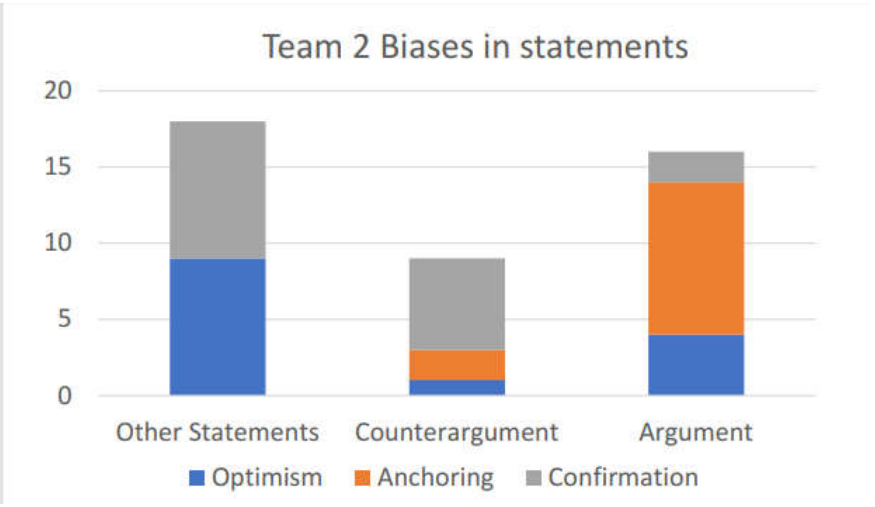
Results

Teams' biased arguments



Biases in statements





Key observations

- Most biased **arguments in favour of a solution were influenced by anchoring**
- **Participants were overall less likely to use counterarguments** (find any faults of proposed solutions, or pointing out any risks)
- Most biased **counterarguments were influenced by confirmation bias**, due to the teams' reluctance to change a previously made decision
- **Optimism bias and confirmation bias influenced the overall atmosphere of the meetings** – both teams had a strong need to reassure themselves that their course of action was correct.

Proposed debiasing interventions

Three practices that could lead to debiasing these kind of meetings:

1. **Against anchoring** - the person presenting a solution, should also present at least one drawback
2. **Against confirmation bias** - one of the team members should monitor the discussion and point out the occurrence of such biased argumentation
3. **Against optimism bias** - at the end of the meeting, after making the initial decisions, teams should explicitly list the risks associted with them.

Debiasing architectural decision-making: a workshop-based training approach

Based on:

Borowa, K., Jarek, M., Mystkowska, G., Paszko, W., & Zalewski, A. (2022, September).
Debiasing architectural decision-making: a workshop-based training approach.
In European Conference on Software Architecture (pp. 159-166). Cham: Springer International Publishing.

Debiasing levels

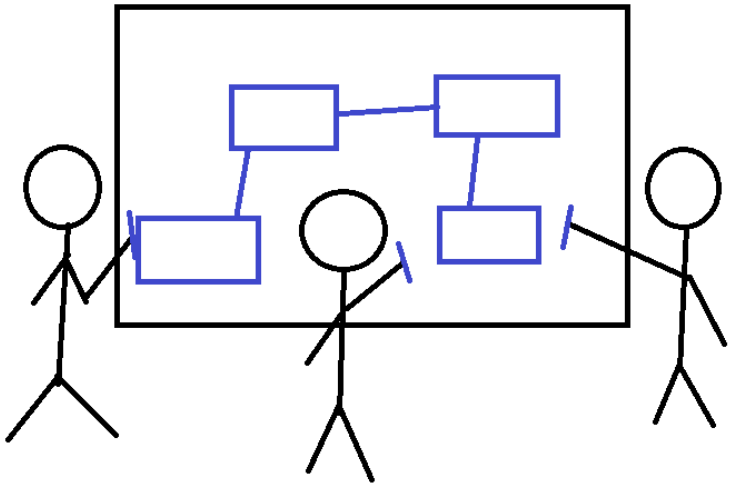
Four levels of debiasing treatments include (Fischhoff, 1982):

A. **Warning about the biases**
B. **Describing typical biases**
C. **Providing personalised feedback about the biases**
D. An extended programme of debiasing training

Sub-Research Question

SRQ. Is a training workshop* an effective method of reducing the impact of cognitive biases on architectural decision-making?

***level C debiasing treatment**



Method

Preparing the debiasing workshop based on the pilot study's findings

Gathering participants.

A series of three-hour long meetings during which we conducted the experiment, which consisted of three steps:

- **Task 1 -- a 1 hour-long ADM task.**
- **The debiasing workshop.**
- **Task 2 -- a 1 hour-long ADM task.**

Analysing the teams' performance during the first and second tasks.

The architectural decision-making task

- Done in 3-4 person teams
- On the MS Teams platform
- The discussion was recorded
- **The task was to design an architecture that could be used as a solution to a given theme.**
- **The architecture was to be recorded using the C4 model notation.**
- The task itself was known to the participants before they took part in the experiment
- This themes were unknown before the workshop
- Different themes for Task 1 and 2 for a single team

The debiasing workshop

- 1 hour long
- Designed to teach about cognitive biases in architectural decision-making
- 3 debiasing techniques were taught, with practical exercises:
  - **The anti-anchoring technique**: having proposed an architectural solution, the individual that presents it must explicitly list one disadvantage of the solution
  - **The anti-confirmation bias technique**: one team member has to monitor the discussion for unjustified statements that dismiss new information and ideas. Such as ``We already decided that''.
  - **The anti-optimism bias technique**: the team must explicitly mention the risks associated with the design decisions.

Sample

- 12 teams consisting of 3-4 participants
- Master's level graduate students majoring in Computer Science
- Participants were taking a Software Architecture course
- Participation was voluntary.
  - The tasks were graded but there was an alternative way to obtain a grade.
  - Students were given additional time after the experiment to polish their designs
- Overall, 61% of the participants had prior experience in software development, ranging from 0.3 to 3 years

Analysis

Recordings from the meetings were transcribed

Each recording was coded by 2 authors independently

The coders met to discuss the differences and negotiate what codes should be used, until they reached full agreement

We coded the following:

Architectural decisions	**Arguments/counterarguments for/against a particular decision**	**Biased statements**	**Uses of the debiasing techniques**

We used the non-parametric Wilcoxon Signed Rank Test to evaluate whether the changes in specific measured values were statistically different (when the p-value was less than 0.05).

Results - Arguments

- Overall: 1470 arguments and 487 counterarguments
- 54% of the statements before the workshop were biased, compared to 36% after
- **The percentage of biased arguments decreased after the workshop in the cases of all teams except one.**
- Significant (p-value < 0.05) changes:
  - Increased number of non-biased arguments
  - Increased number of non-biased counterarguments
  - The decrease of the percentage of biased statements
- Not significant changes:
  - Change in the number of biased arguments and counterarguments

Results - Decisions

- 3 types of decisions depending on the amount of biased arguments/counterarguments impacting them: biased, non-biased, neutral
- Overall: 641 decisions - 266 biased, 281 non-biased and 94 neutral
- **52% of decisions before the workshop were biased, compared to 31% after**
- Only one team had a larger percentage of biased decisions after the workshop. In the case of all the other teams, the percentage of biased decisions decreased.
- Significant (p-value < 0.05) changes:
  - Increase in the number of non-biased decisions
  - Decreased percentage of biased decisions
- Not significant changes:
  - The number of biased decisions

Results – Cognitive biases

- Overall: 1110 bias occurrences - 558 before and 552 after the workshop.
- There was no significant change in the overall number of biases between Task 1 and Task 2 (p-value = 0.8647).
- The debiasing effect was **not achieved by decreasing the number of bias but from increasing the number of non-biased arguments.**

Results - Debiasing techniques.

- Overall: 133 uses of the proposed techniques - 26 techniques before and 107 after the workshop.
- Significant increase in the number of uses of the practices (p-value = 0.0005)
- The **anti-optimism** technique was used **most often** (15 before and 57 after workshop)
- The **anti-anchoring** technique was used **less often** (3 before and 30 after workshop)
- The **anti-confirmation bias** technique **rarely being used at all** (8 before and 20 after workshop).

Conclusion

We explored whether debiasing through a training workshop is an effective method of reducing the impact of cognitive biases on architectural decision making.

The debiasing treatment through the debiasing workshop we designed was successful

Both improving the quality of argumentation and design decisions.

We did not decrease the number of biases impacting the teams

We managed to increase the overall amount and percentage of rational arguments and decisions

Debiasing experts

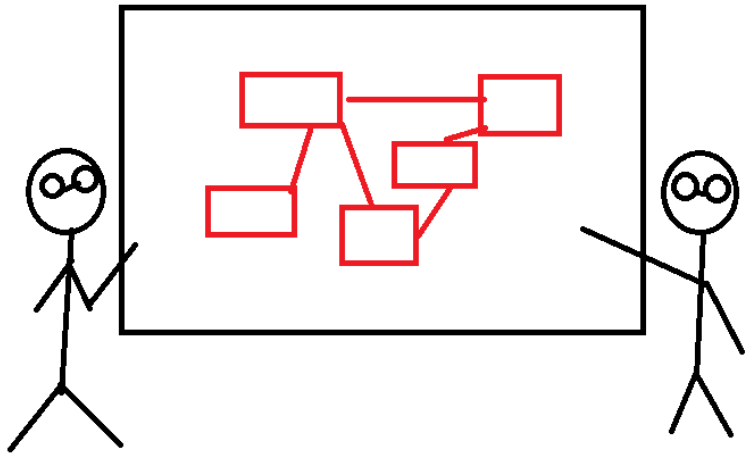
Yet unpublished

Research by:
Klara Borowa, Rodrigo Rebouças de Almeida, and Marion Wiese.

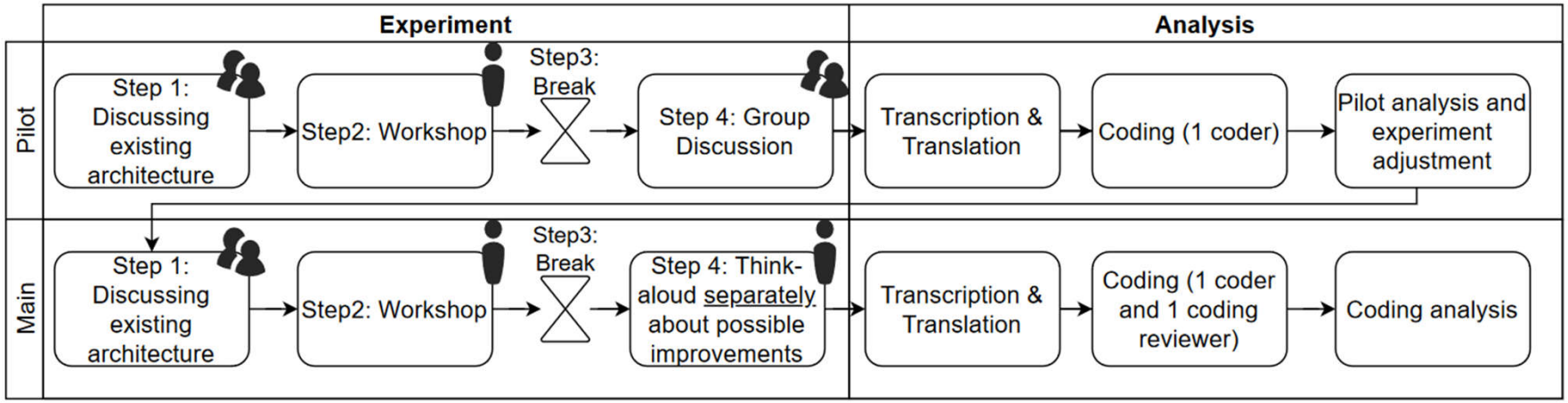
Sub-Research Question

SRQ. How are experienced practitioners influenced by the proposed architectural decision-making debiasing workshop?

- **Would the workshop decrease the number of cognitive bias occurrences?**
- **Would the workshop increase the participant's use of debiasing techniques?**



Method


Experiment
Analysis
Pilot
Main
Step 1: Discussing existing architecture
Step2: Workshop
Step3: Break
Step 4: Group Discussion
Transcription & Translation
Coding (1 coder)
Pilot analysis and experiment adjustment
Step 1: Discussing existing architecture
Step2: Workshop
Step3: Break
Step 4: Think-aloud separately about possible improvements
Transcription & Translation
Coding (1 coder and 1 coding reviewer)
Coding analysis

Debiasing techniques

- **The anti-anchoring technique**: having proposed an architectural solution, the individual that presents it must explicitly list one disadvantage of the solution
- **(Changed)**~~**The anti-confirmation bias technique**: one team member has to monitor the discussion for unjustified statements that dismiss new information and ideas. Such as ``We already decided that''.~~
- **(New)**<u>**Anti confirmation and optimism bias technique:** Explicitly listing multiple solution options.</u>
- **The anti-optimism bias technique**: the team must explicitly mention the risks associated with the design decisions.

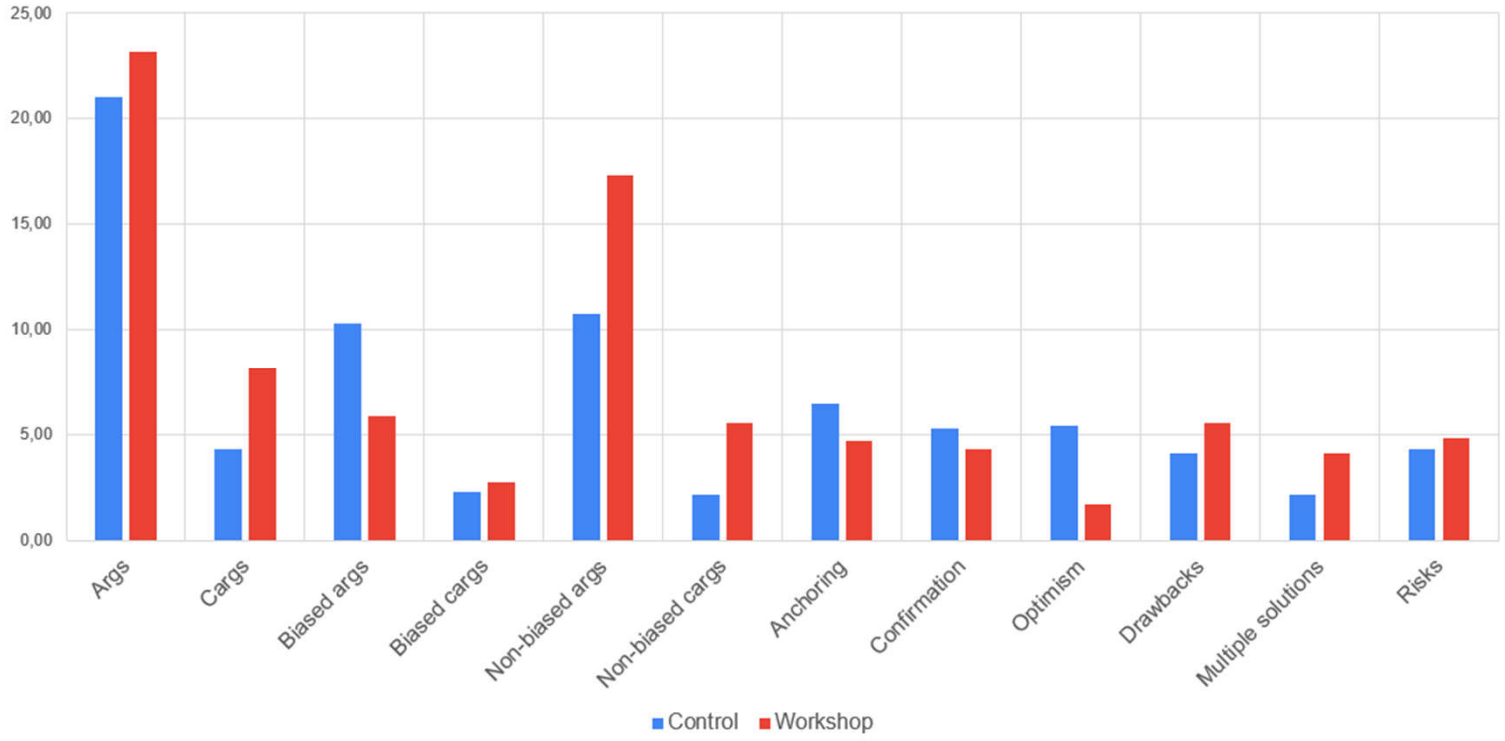
Participants

	No.	Pair No.	Group	Age (years)	IT exp. (years)	Role	Company domain	Company size (employees)
Pilot	1	P1	W	28	7	Developer	Digital payment	21-100
	2	P1	C	28	2	Developer	Digital payment	21-100
	3	P2	W	40	18	Analyst	Digital payment	21 - 100
	4	P2	C	37	5	Product Manager	Digital payment	21 - 100
Main	5	P3	W	52	20	Developer	Media	over 8000
	6	P3	C	59	38	Developer	Media	over 8000
	7	P4	W	42	20	Developer	Marketing Services	101-500
	8	P4	C	40	20	Developer	Marketing Services	101-500
	9	P5	W	42	20	CTO	Finance	101-500
	10	P5	C	52	30	Architect	Finance	101-500
	11	P6	W	39	7	Architect	Signal processing	500-5000
	12	P6	C	59	38	Systems Engineer	Signal processing	500-5000
	13	P7	W	46	20	Architect	Retail	over 5000
	14	P7	C	41	19	Architect	Retail	over 5000
	15	P8	W	29	5	Developer	Education	500-5000
	16	P8	C	30	4	Developer	Education	500-5000
	17	P9	W	39	20	Project Manager	Government	101-500
	18	P9	C	42	19	Systems Development Coordinator	Government	101-500

Analysis - Coding table

Code	Code meaning	Description
Arg	**Argument**	A statement in **support** of a possible solution alternative.
Carg	**Counterargument**	A statement in **opposition** to choosing a particular solution alternative.
Anch	**Anchoring**	A statement suggesting that the participant is impacted by **anchoring**.
Conf	**Confirmation bias**	A Statement suggesting that the participant is impacted by a **confirmation bias**.
Opt	**Optimism bias**	A statement suggesting that the participant is impacted by an **optimism bias**.
Ddraw	**Decision's draw-back**	Use of the **anti-anchoring technique**, i.e., a statement where the participant discusses a drawback of the solution alternative.
Dmulti	**Decision with multiple alternatives**	Use of the **anti-confirmation bias and anti-anchoring technique** , i.e., a statement where the participant mentions more than one solution alternative.
Drisk	**Decision's risk**	Use of the **anti-optimism bias technique**, i.e. a statement where the participant discusses a risk associated with a solution alternative.

Results - overall



Results: p-values

Measurement	Control group average of code sums	Workshop group average of code sums	p-value	Research hypothesis (H_R)
Arg	21	23.14	0.3997	Workshop>Control
Carg	4.29	8.14	0.0574	Workshop>Control
Anch	6.43	4.71	0.1721	Workshop<Control
Conf	5.29	4.29	0.4063	Workshop<Control
Opt	5.43	1.71	0.0862	Workshop<Control
Biases sum	17.15	10.71	0.1094	Workshop<Control
Biased Args	10.29	5.86	0.0739	Workshop<Control
Biased Cargs	2.29	2.71	0.4461	Workshop<Control
Not biased Args	10.71	17.29	0.1342	Workshop>Control
Not biased Cargs	2.14	5.57	**0.0449**	Workshop>Control
% Biased statements	49,18%	28,09%	0.0781	Workshop<Control
Ddraw	4.14	5.57	0.1114	Workshop>Control
Dmulti	2.14	4.14	**0.029**	Workshop>Control
Drisk	4.29	4.86	0.5	Workshop>Control
Techniques use sum	10.57	14.57	0.07813	Workshop>Control

Main findings

- The workshop decreased the occurrence of all three researched cognitive biases.
- The amounts of uses of each debiasing technique that we taught increased.
- Two statistically significant changes:
  - The increase of non-biased counterarguments.
  - The increased use of the debiasing technique of “listing multiple solution options”.

Additional takeaways

Experienced practitioners have fewer problems in specifying non-biased arguments in support of decisions – compared to students.

Debiasing might only be effective when teaching team members of all seniority levels.

A practitioner's high confidence level makes them more susceptible to cognitive biases.

Discussing too many decisions makes biases more likely to occur.

Thesis Contributions

Contributions

Set of rationales which motivate architectural decisions made by software practitioners.

Set of cognitive biases that impact architectural decision-making.

Cognitive biases may make practitioners incur unnecessary architectural technical debt.

The wicked triad – anchoring, optimism, and confirmation bias.

Empirically validated debiasing workshop.

Students and experienced practitioners react differently to a debiasing workshop.

Thank you

References

[1] Boehm, “Software engineering”, IEEE Transactions on Computers, vol. C-25, no. 12, pp. 1226–1241, 1976. DOI: 10.1109/TC.1976.1674590.
[2] P. Naur and B. Randell, Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO. 1969.
[3] P. Bourque, R. E. Fairley, and I. C. Society, Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0, 3rd. Washington, DC, USA: IEEE Computer Society Press, 2014, ISBN: 0769551661.
[4] J. Buxton and B. Randell, Software Engineering Techniques: Report on a conference sponsored by the Science Committee, Rome, Italy, 27th to 31st October 1969. 1960.
[5] D. E. Perry and A. L. Wolf, “Foundations for the study of software architecture”, ACM SIGSOFT Software engineering notes, vol. 17, no. 4, pp. 40–52, 1992.
[6] A. Jansen and J. Bosch, “Software architecture as a set of architectural design decisions”, in 5th Working IEEE/IFIP Conference on Software Architecture (WICSA’05), IEEE, 2005, pp. 109–120.
[7] P. B. Kruchten, “The 4+ 1 view model of architecture”, IEEE software, vol. 12, no. 6, pp. 42–50, 1995.
[8] ISO/IEC/IEEE, “Systems and software engineering – architecture description”, ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), pp. 1–46, Jan. 2011.
[9] S. Brown, “The c4 model for software architecture”, Updated August, vol. 1, 2018.
[10] A. Tversky and D. Kahneman, Judgment under uncertainty: Heuristics and biases. utility, probability, and human decision making, 185 (4157), 141–162, 1975.
[11] D. Kahneman, Thinking, fast and slow. Macmillan, 2011.
[12] D. Kahneman, “Think fast, think slow”, Farrar, Straus and Giroux, New York, 2011.
[13] NobelPrize.org, Daniel kahneman – facts, https://www.nobelprize.org/ prizes/economic-sciences/2002/kahneman/facts/, Accessed: 03.01.2024. [14] R. Mohanani, I. Salman, B. Turhan, P. Rodriguez, and P. Ralph, “Cognitive Biases in Software Engineering: A Systematic Mapping Study”, IEEE Transactions on Software Engineering, vol. 5589, no. c, 2018, ISSN: 19393520. DOI: 10.1109/TSE.2018.
[15] R. Mohanani, P. Ralph, and B. Shreeve, “Requirements fixation”, Proceedings - International Conference on Software Engineering, pp. 895–906, 2014, ISSN: 02705257. DOI: 10.1145/2568225.2568235.
[16] T. Halkjelsvik and M. Jørgensen, Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life. Springer Nature, 2018.
[17] K. Borowa, S. Kamoda, P. Ogrodnik, and A. Zalewski, “Fixations in agile software development teams”, Foundations of Computing and Decision Sciences, vol. 48, no. 1, pp. 3–18, 2023.
[18] I. Salman, P. Rodriguez, B. Turhan, A. Tosun, and A. Güreller, “What leads to a confirmatory or disconfirmatory behavior of software testers?”, IEEE Transactions on Software Engineering, vol. 48, no. 4, pp. 1351–1368, 2020.
[19] H. Van Vliet and A. Tang, “Decision making in software architecture”, Journal of Systems and Software, vol. 117, pp. 638–644, 2016.
[20] S. Chattopadhyay, N. Nelson, A. Au, et al., “A Tale from the Trenches : Cognitive Biases and Software Development”, in International Conference on Software Engineering (ICSE), 2020, pp. 654–665, ISBN: 9781450371216. DOI: 10.1145/3377811.
[21] M. Razavian, B. Paech, and A. Tang, “Empirical research for software architecture decision making: An analysis”, Journal of Systems and Software, vol. 149, pp. 360–381, 2019.
[22] A. Tang, “Software designers, are you biased?”, Proceedings - International Conference on Software Engineering, no. January 2011, pp. 1–8, 2011, ISSN: 02705257. DOI: 10.1145/1988676.1988678.
[23] A. Manjunath, M. Bhat, K. Shumaiev, A. Biesdorf, and F. Matthes, “Decision Making and Cognitive Biases in Designing Software Architectures”, Proceedings - 2018 IEEE 15th International Conference on Software Architecture Companion, ICSA-C 2018, pp. 52–55, 2018. DOI: 10.1109/ICSA-C.2018.00022.
[24] A. Martini and J. Bosch, “The danger of architectural technical debt: Contagious debt and vicious circles”, in 2015 12th Working IEEE/IFIP Conference on Software Architecture, IEEE, 2015, pp. 1–10.
[25] M. Soliman, P. Avgeriou, and Y. Li, “Architectural design decisions that incur technical debt—an industrial case study”, Information and Software Technology, vol. 139, p. 106669, 2021.
[26] A. Zalewski, K. Borowa, and A. Ratkowski, “On cognitive biases in architecture decision making”, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10475 LNCS, 2017, pp. 123–137, ISBN: 9783319658308.
[27] P. Avgeriou, P. Kruchten, I. Ozkaya, and C. Seaman, “Managing technical debt in software engineering (dagstuhl seminar 16162)”, in Dagstuhl reports, Schloss Dagstuhl Leibniz - Zentrum fuer Informatik, vol. 6, 2016.
[28] T. Besker, A. Martini, and J. Bosch, “Managing architectural technical debt: A unified model and systematic literature review”, Journal of Systems and Software, vol. 135, pp. 1–16, 2018, ISSN: 01641212. DOI: 10.1016/j.jss.2017.09.025.
[29] M. Shepperd, C. Mair, and M. Jørgensen, “An Experimental Evaluation of a De-biasing Intervention for Professional Software Developers”, in Proceedings of the 33rd Annual ACM Symposium on Applied Computing, 2018. DOI: 10.1145/3167132.
[30] K. M. Borowa, “On cognitive biases in software engineering”, Instytut Automatyki iInformatyki Stosowanej, 2019.
[31] W. James, Principles of Psychology (1980). Henry Holt and Company, 1931.
[32] M. I. Posner, C. R. Snyder, and R. Solso, “Attention and cognitive control”, Cognitive psychology: Key readings, vol. 205, pp. 55–85, 2004.
[33] A. Tversky and D. Kahneman, “Judgment under uncertainty: Heuristics and biases”, science, vol. 185, no. 4157, pp. 1124–1131, 1974.
[34] D. Kahneman and A. Tversky, “Subjective probability: A judgment of representativeness”, Cognitive psychology, vol. 3, no. 3, pp. 430–454, 1972.
[35] T. Pachur, R. Hertwig, and F. Steinmann, “How do people judge risks: Availability heuristic, affect heuristic, or both?”, Journal of Experimental Psychology: Applied, vol. 18, no. 3, p. 314, 2012.
[36] A. Furnham and H. C. Boo, “A literature review of the anchoring effect”, The journal of socio-economics, vol. 40, no. 1, pp. 35–42, 2011.
[37] W. Wattanacharoensil and D. La-ornual, “A systematic review of cognitive biases in tourist decisions”, Tourism Management, vol. 75, pp. 353–369, 2019.
[38] G. Saposnik, D. Redelmeier, C. C. Ruff, and P. N. Tobler, “Cognitive biases associated with medical decisions: A systematic review”, BMC medical informatics and decision making, vol. 16, no. 1, pp. 1–14, 2016.
[39] M. E. Oswald and S. Grosjean, “Confirmation bias”, in Cognitive illusions: A handbook on fallacies and biases in thinking, judgement and memory, R. Pohl, Ed., Psychology Press, 2004, pp. 97–114.
[40] L. M. Leventhal, B. M. Teasley, D. S. Rohlman, and K. Instone, “Positive test bias in software testing among professionals: A review”, in Human-Computer Interaction: Third International Conference, EWHCI’93 Moscow, Russia, August 3–7, 1993 Selected Papers 3, Springer, 1993, pp. 210–218.
[41] A. Bracha and D. J. Brown, “Affective decision making: A theory of optimism bias”, Games and Economic Behavior, vol. 75, no. 1, pp. 67–80, 2012.
[42] R. Buehler, D. Griffin, and M. Ross, “Exploring the" planning fallacy": Why people underestimate their task completion times.”, Journal of personality and social psychology, vol. 67, no. 3, p. 366, 1994.
[43] M. Ross and F. Sicoly, “Egocentric biases in availability and attribution.”, Journal of personality and social psychology, vol. 37, no. 3, p. 322, 1979.
[44] S. Freud, Three essays on the theory of sexuality: The 1905 edition. Verso Books, 2017.
[45] P. Ralph, “Toward a theory of debiasing software development”, Lecture Notes in Business Information Processing, vol. 93 LNBIP, pp. 92–105, 2011, ISSN: 18651348. DOI: 10.1007/978-3-642-25676-9_8.
[46] D. J. Malenka, J. A. Baron, S. Johansen, J. W. Wahrenberger, and J. M. Ross, “The framing effect of relative and absolute risk”, Journal of general internal medicine, vol. 8, pp. 543–548, 1993.
[47] J. L. Nicolau, J. P. Mellinas, and E. Martín-Fuentes, “The halo effect: A longitudinal approach”, Annals of Tourism Research, vol. 83, p. 102938, 2020.
[48] G. J. Browne and V. Ramesh, “Improving information requirements determination: A cognitive perspective”, Information & Management, vol. 39, no. 8, pp. 625–645, 2002.
[49] M. G. Pitts and G. J. Browne, “Improving requirements elicitation: An empirical investigation of procedural prompts”, Information systems journal, vol. 17, no. 1, pp. 89–110, 2007.
[50] S. Chakraborty, S. Sarker, and S. Sarker, “An exploration into the process of requirements elicitation: A grounded approach”, Journal of the association for information systems, vol. 11, no. 4, p. 1, 2010.

References

[51] A. Zalewski, K. Borowa, and D. Kowalski, “On cognitive biases in requirements elicitation”, in Integrating Research and Practice in Software Engineering, Springer, 2020, pp. 111–123.
[52] W. Stacy and J. MacMillan, “Cognitive bias in software engineering”, Communications of the ACM, vol. 38, no. 6, pp. 57–63, 1995.
[53] J. Parsons and C. Saunders, “Cognitive heuristics in software engineering applying and extending anchoring and adjustment to artifact reuse”, IEEE Transactions on Software Engineering, vol. 30, no. 12, pp. 873–888, 2004.
[54] G. Allen and J. Parsons, “Is query reuse potentially harmful? anchoring and adjustment in adapting existing database queries”, Information Systems Research, vol. 21, no. 1, pp. 56–77, 2010.
[55] G. Calikli and A. Bener, “Empirical analyses of the factors affecting confirmation bias and the effects of confirmation bias on software developer/tester performance”, in Proceedings of the 6th International Conference on Predictive Models in Software Engineering, 2010, pp. 1–11.
[56] G. Çalıklı and A. B. Bener, “Influence of confirmation biases of developers on software quality: An empirical study”, Software Quality Journal, vol. 21, pp. 377–416, 2013.
[57] G. Calikli, A. Bener, and B. Arslan, “An analysis of the effects of company culture, education and experience on confirmation bias levels of software developers and testers”, in Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2, 2010, pp. 187–190.
[58] G. Calikli, B. Aslan, and A. Bener, “Confirmation bias in software development and testing: An analysis of the effects of company size, experience and reasoning skills”, 2010.
[59] M. Jørgensen and D. I. Sjøberg, “Software process improvement and human judgement heuristics”, Scandinavian Journal of Information Systems, vol. 13, no. 1, p. 2, 2001.
[60] K. Moløkken and M. Jørgensen, “Software effort estimation: Unstructured group discussion as a method to reduce individual biases.”, in PPIG, 2003, p. 4.
[61] K. Moløkken-Østvold and M. Jørgensen, “Group processes in software effort estimation”, Empirical Software Engineering, vol. 9, no. 4, pp. 315–334, 2004.
[62] K. Molokken-Ostvold and N. C. Haugen, “Combining estimates with planning poker–an empirical study”, in 2007 Australian Software Engineering Conference (ASWEC’07), IEEE, 2007, pp. 349–358.
[63] K. Moløkken and M. Jørgensen, “Expert estimation of the effort of web-development projects: Why are software professionals in technical roles more optimistic than those in nontechnical roles”, Journal of Empirical Software Engineering, 2004.
[64] M. Jørgensen, K. H. Teigen, and K. Moløkken, “Better sure than safe? over-confidence in judgement based software development effort prediction intervals”, Journal of systems and software, vol. 70, no. 1-2, pp. 79–93, 2004.
[65] M. Jorgensen and S. Grimstad, “Over-optimism in software development projects:" the winner’s curse"”, in 15th International Conference on Electronics, Communications and Computers (CONIELECOMP’05), IEEE, 2005, pp. 280–285.
[66] M. Jørgensen and B. Faugli, “Prediction of overoptimistic predictions”, in 10th International Conference on Evaluation and Assessment in Software Engineering (EASE) 10, 2006, pp. 1–10.
[67] M. Jørgensen, “Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates”, Information and Software Technology, vol. 52, no. 5, pp. 506–516, 2010.
[68] O. Shmueli, N. Pliskin, and L. Fink, “Can the outside-view approach improve planning decisions in software development projects?”, Information Systems Journal, vol. 26, no. 4, pp. 395–418, 2016.
[69] K. M. Lui and K. C. Chan, “A cognitive model for solo programming and pair programming”, in Proceedings of the Third IEEE International Conference on Cognitive Informatics, 2004., IEEE, 2004, pp. 94–102.
[70] F. Ramin, “The role of egocentric bias in undergraduate agile software development teams”, in Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings, 2020, pp. 122–124.
[71] P. Tobias and D. S. Spiegel, Is design the preeminent protagonist in user experience?, 2009.
[72] C. Gacek, A. Abd-Allah, B. Clark, and B. Boehm, “On the definition of software system architecture”, in Proceedings of the First International Workshop on Architectures for Software Systems, Seattle, Wa, 1995, pp. 85–94.
[73] L. Bass, P. Clements, and R. Kazman, Software architecture in practice. Addison-Wesley Professional, 2003.
[74] A. Tang and R. Kazman, “Decision-making principles for better software design decisions”, IEEE Software, vol. 38, no. 6, pp. 98–102, 2021.
[75] A. Tang, M. Razavian, B. Paech, and T. M. Hesse,
[76] A. Tang and H. van Vliet, “Design strategy and software design effectiveness”, IEEE software, vol. 29, no. 1, pp. 51–55, 2011.
[77] H. van Vliet and A. Tang, “Decision making in software architecture”, Journal of Systems and Software, vol. 117, pp. 638–644, 2016, ISSN: 01641212. DOI: 10.1016/j.jss.2016.01.017.
[78] B. Fischhoff, “Debiasing. judgment under uncertainty: Heuristics and biases”, Judgment under uncertainty: Heuristics and biases, pp. 422–444, 1982.
[79] M. Razavian, A. Tang, R. Capilla, and P. Lago, “In two minds: How reflections influence software design thinking”, Journal of Software: Evolution and Process, vol. 28, no. 6, pp. 394–426, 2016.
[80] A. Tang, F. Bex, C. Schriek, and J. M. E. van der Werf, “Improving software design reasoning–a reminder card approach”, Journal of Systems and Software, vol. 144, pp. 22– 40, 2018.
[81] O. Zimmermann, J. Koehler, F. Leymann, R. Polley, and N. Schuster, “Managing architectural decision models with dependency relations, integrity constraints, and production rules”, Journal of Systems and Software, vol. 82, no. 8, pp. 1249–1267, 2009.
[82] A. Tang, M. A. Babar, I. Gorton, and J. Han, “A survey of architecture design rationale”, Journal of Systems and Software, vol. 79, no. 12, pp. 1792–1804, 2006, ISSN: 01641212.
[83] M. Razavian, B. Paech, and A. Tang, “Empirical research for software architecture decision making: An analysis”, Journal of Systems and Software, vol. 149, pp. 360–381, 2019, ISSN: 01641212.
[84] R. Weinreich, I. Groher, and C. Miesbauer, “An expert survey on kinds, influence factors and documentation of design decisions in practice”, Future Generation Computer Systems, vol. 47, pp. 145–160, 2015, ISSN: 0167739X.
[85] C. Miesbauer and R. Weinreich, “Classification of design decisions–an expert survey in practice”, in Software Architecture: 7th European Conference, ECSA 2013, Montpellier, France, July 1-5, 2013. Proceedings 7, Springer, 2013, pp. 130–145.
[86] T. Bi, P. Liang, and A. Tang, “Architecture Patterns, Quality Attributes, and Design Contexts: How Developers Design with Them”, Proceedings - Asia-Pacific Software Engineering Conference, APSEC, vol. 2018-Decem, no. 61472286, pp. 49–58, 2018, ISSN: 15301362.
[87] M. Soliman, M. Riebisch, and U. Zdun, “Enriching Architecture Knowledge with Technology Design Decisions”, Proceedings - 12th Working IEEE/IFIP Conference on Software Architecture, WICSA 2015, pp. 135–144, 2015.
[88] M. Bhat, K. Shumaiev, U. Hohenstein, A. Biesdorf, and F. Matthes, “The evolution of architectural decision making as a key focus area of software architecture research: A semi-systematic literature study”, in 2020 IEEE International Conference on Software Architecture (ICSA), IEEE, 2020, pp. 69–80.
[89] T. Bi, P. Liang, A. Tang, and X. Xia, “Mining Architecture Tactics and Quality Attributes Knowledge in Stack Overflow”, Journal of Systems and Software, no. May, 2021.
[90] ISO/IEC 25010, ISO/IEC 25010:2011, systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models, 2011.
[91] M. Bhat, C. Tinnes, K. Shumaiev, A. Biesdorf, U. Hohenstein, and F. Matthes, “Adex: A tool for automatic curation of design decision knowledge for architectural decision recommendations”, in 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), IEEE, 2019, pp. 158–161.
[92] M. X. Liu, J. Hsieh, N. Hahn, et al., “Unakite: Scaffolding developers’ decision-making using the web”, in Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, 2019, pp. 67–80.
[93] M. Razavian, A. Tang, R. Capilla, and P. Lago, “Reflective approach for software design decision making”, Proceedings - 1st Workshop on Qualitative Reasoning about Software Architectures, QRASA 2016, pp. 19–26, 2016.
[94] A. Tang and R. Kazman, “Decision-Making Principles for Better Software Design Decisions”, IEEE Software, vol. 38, no. 6, pp. 98–102, 2021, ISSN: 19374194.
[95] J. E. Burge, “Design rationale: Researching under uncertainty”, Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, vol. 22, no. 4, pp. 311–324, 2008, ISSN: 08900604.
[96] M. J. De Dieu, P. Liang, and M. Shahin, “How Do Developers Search for Architectural Information? An Industrial Survey”, Proceedings - IEEE 19th International Conference on Software Architecture, ICSA 2022, no. December 2021, pp. 58–68, 2022.
[97] D. Tofan, M. Galster, and P. Avgeriou, “Difficulty of architectural decisions – a survey with professional architects”, in Software Architecture, K. Drira, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 192–199, ISBN: 978-3-642-39031-9.
[98] P. Runeson, M. Höst, A. Rainer, and B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples. 2012, ISBN: 9781118104354. [Online]. Available: www.wiley.com..
[99] K. Borowa, R. Lewanczyk, K. Stpiczynska, P. Stradomski, and A. Zalewski, What rationales drive architectural decisions? An empirical inquiry - Additional material, version 1, Zenodo, May 2023. DOI: 10.5281/zenodo.7946764. [Online]. Available: https://doi.org/10.5281/zenodo.7946764.
[100] J. Saldaña, “The coding manual for qualitative researchers”, The coding manual for qualitative researchers, pp. 1–440, 2021.

References

[101] D. R. Garrison, M. Cleveland-Innes, M. Koole, and J. Kappelman, "Revisiting methodological issues in transcript analysis: Negotiated coding and reliability", Internet and Higher Education, vol. 9, no. 1, pp. 1–8, 2006, ISSN: 10967516.
[102] K. Beck, M. Beedle, A. van Bennekum, et al. "Principles behind the Agile Manifesto". (2001), [Online]. Available: https://agilemanifesto.org/principles. html (visited on 07/05/2021).
[103] P. Kruchten, R. Nord, and I. Ozkaya, Managing Technical Debt. Addison-Wesley Professional, 2019.
[104] A. Zalewski, K. Borowa, and A. Ratkowski, "On cognitive biases in architecture decision making", in Software Architecture, A. Lopes and R. de Lemos, Eds., Cham: Springer International Publishing, 2017, pp. 123–137, ISBN: 978-3-319-65831-5.
[105] K. Daniel, Thinking, fast and slow. 2017.
[106] P. Naur and B. Randell, "Software engineering: Report of a conference sponsored by the nato science committee, garmisch, germany, 7th-11th october 1968", 1969.
[107] J. N. Buxton and B. Randell, Software Engineering Techniques: Report on a Conference Sponsored by the NATO Science Committee. NATO Science Committee; available from Scientific Affairs Division, NATO, 1970.
[108] A. Tversky and D. Kahneman, "Rational choice and the framing of decisions", Decision making: Descriptive, normative, and prescriptive interactions, pp. 167–192, 1988.
[109] H. Leibenstein, "Bandwagon, snob, and veblen effects in the theory of consumers' demand", The quarterly journal of economics, vol. 64, no. 2, pp. 183–207, 1950.
[110] S. A. Birch and P. Bloom, "The curse of knowledge in reasoning about false beliefs", Psychological Science, vol. 18, no. 5, pp. 382–386, 2007.
[111] D. Kahneman and J. Renshon, "Hawkish biases", American Foreign Policy and the Politics of Fear: Threat Inflation Since, vol. 9, no. 11, pp. 79–96, 2009.
[112] C. Zannier, M. Chiasson, and F. Maurer, "A model of design decision making based on empirical results of interviews with software designers", Information and Software Technology, vol. 49, no. 6, pp. 637–653, 2007.
[113] A. Tang and H. van Vliet, "Software designers satisfice", in Software Architecture: 9th European Conference, ECSA 2015, Dubrovnik/Cavtat, Croatia, September 7-11, 2015. Proceedings 9, Springer, 2015, pp. 105–120.
[114] J. Kruger and D. Dunning, "Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments.", Journal of personality and social psychology, vol. 77, no. 6, p. 1121, 1999.
[115] W. H. Brown, R. C. Malveau, H. W. S. McCormick, and T. J. Mowbray, AntiPatterns: refactoring software, architectures, and projects in crisis. John Wiley & Sons, Inc., 1998.
[116] R. S. Nickerson, "Confirmation bias: A ubiquitous phenomenon in many guises", Review of general psychology, vol. 2, no. 2, pp. 175–220, 1998.
[117] M. I. Norton, D. Mochon, and D. Ariely, "The ikea effect: When labor leads to love", Journal of consumer psychology, vol. 22, no. 3, pp. 453–460, 2012.
[118] C. N. Parkinson, Parkinson's Law, or the Pursuit of Progress. 1958.
[119] E. M. Rogers, Diffusion of innovations. Simon and Schuster, 2010.
[120] M. H. Bazerman and M. A. Neale, Negotiating rationally. Simon and Schuster, 1993. [121] T. Sharot, A. M. Riccardi, C. M. Raio, and E. A. Phelps, "Neural mechanisms mediating optimism bias", Nature, vol. 450, no. 7166, pp. 102–105, 2007.
[122] N. Rios, M. G. de Mendonça Neto, and R. O. Spínola, "A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners", Information and Software Technology, vol. 102, pp. 117–145, 2018.
[123] W. Cunningham, "The WyCash portfolio management system", in Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA, vol. Part F1296, 1992, pp. 29–30, ISBN: 0897916107. DOI: 10.1145/157709.157715.
[124] E. Tom, A. Aurum, and R. Vidgen, "An exploration of technical debt", Journal of Systems and Software, vol. 86, no. 6, pp. 1498–1516, 2013.
[125] T. Amanatidis, N. Mittas, A. Chatzigeorgiou, A. Ampatzoglou, and L. Angelis, "The developer's dilemma: Factors affecting the decision to repay code debt", in 2018 IEEE/ACM International Conference on Technical Debt (TechDebt), vol. 5, 2018, pp. 62–66, ISBN:9781450357135. DOI: 10.1145/3194164.3194174
[126] A. Martini and J. Bosch, "The Danger of Architectural Technical Debt: Contagious Debt and Vicious Circles", in Proceedings - 12th Working IEEE/IFIP Conference on Software Architecture, WICSA 2015, 2015, pp. 1–10, ISBN: 9781479919222. DOI: 10.1109/WICSA.2015.31
[127] R. Verdecchia, P. Kruchten, and P. Lago, "Architectural Technical Debt : A Grounded Theory", European Conference on Software Architecture (ECSA), 2020.
[128] R. Brenner, "Balancing resources and load: Eleven nontechnical phenomena that contribute to formation or persistence of technical debt", in Proceedings - 2019 IEEE/ACM International Conference on Technical Debt, TechDebt 2019, 2019, pp. 38–47, ISBN: 9781728133713. DOI: 10.1109/TechDebt.2019.00013.
[129] R. Mohanani, P. Ralph, and B. Shreeve, "Requirements fixation", in Proceedings of the 36th International Conference on Software Engineering, 2014, pp. 895–906.
[130] J. Bosch, "Software architecture: The next step", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3047, pp. 194–199, 2004, ISSN: 03029743. DOI: 10.1007/978-3-540-24769-214.
[131] R. Alfayez, W. Alwehaibi, R. Winn, E. Venson, and B. Boehm, "A systematic literature review of technical debt prioritization", in Proceedings of the 3rd International Conference on Technical Debt, vol. 10, ACM, 2020, pp. 1–10, ISBN: 9781450379601. DOI: 10.1145/3387906.3388630
[132] C. Becker, R. Chitchyan, S. Betz, and C. McCord, "Trade-off decisions across time in technical debt management: A systematic literature review", in 2018 IEEE/ACM International Conference on Technical Debt (TechDebt), ACM, 2018, pp. 85–94, ISBN: 9781450357135. DOI: 10.1145/3194164.3194171
[133] N. Rios, M. Mendonça, and R. Spínola, "A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners", Information and Software Technology, vol. 102, Jun. 2018. DOI: 10.1016/j.infsof.2018.
[134] A. Martini, J. Bosch, and M. Chaudron, "Architecture technical debt: Understanding causes and a qualitative model", in 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE, 2014, pp. 85–92.
[135] N. A. Ernst, S. Bellomo, I. Ozkaya, R. L. Nord, and I. Gorton, "Measure it? Manage it? Ignore it? Software practitioners and technical debt", in 2015 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015 - Proceedings, 2015, pp. 50–60, ISBN: 9781450336758. DOI: 10.1145/2786805.2786848. [Online]. Available: http://github.com/neilernst/td-survey.
[136] A. Tversky and D. Kahneman, "The framing of decisions and the psychology of choice", Science, 1981, ISSN: 00368075. DOI: 10.1126/science.7455683.
[137] G. B. Chapman and B. H. Bornstein, "The more you ask for, the more you get: Anchoring in personal injury verdicts", Applied Cognitive Psychology, 1996, ISSN: 08884080.
[138] J. Kennedy, "Debiasing in the Audit Curse of Knowledge Judgment", The Accounting Review, 1995, ISSN: 00014826.
[139] M. I. Norton, D. Mochon, and D. Ariely, "The IKEA effect: When labor leads to love", Journal of Consumer Psychology, 2012, ISSN: 10577408. DOI: 10.1016/j.jcps. 2011.08.002.
[140] C. Northcote Parkinson, "Parkinson's law: Or the pursuit of progress", 1961.
[141] E. M. Rogers, A. Singhal, and M. M. Quinlan, "Diffusion of innovations", in An Integrated Approach to Communication Theory and Research, Third Edition, 2019, ISBN: 9781351358712. DOI: 10.4324/9780203710753-35.
[142] M. V. Pezzo, J. A. Litman, and S. P. Pezzo, "On the distinction between yuppies and hippies: Individual differences in prediction biases for planning future tasks", Personality and Individual Differences, 2006, ISSN: 01918869. DOI: 10.1016/j.paid.2006.03.029.
[143] H. Leibenstein, "Bandwagon, snob, and veblen effects in the theory of consumers' demand", Quarterly Journal of Economics, 1950, ISSN: 15314650. DOI: 10.2307/1882692.
[144] B. M. Staw, "The escalation of commitment: An update and appraisal", in Organizational Decision Making, 2010. DOI: 10.1017/cbo9780511584169.011.
[145] A. H. Maslow, The psychology of science; a reconnaissance. 1966, ISBN: National Library: 0354146 LCCN: 66-11479.
[146] O. P. O'Sulliivan, "The Neural Basis of Always Looking on the Bright Side", Dialogues in Philosophy, Mental and Neuro, 2015.
[147] T. Besker, A. Martini, and J. Bosch, "Technical debt cripples software developer productivity: A longitudinal study on developers' daily software development work", in 2018 IEEE/ACM International Conference on Technical Debt (TechDebt), vol. 10, 2018, pp. 105–114, ISBN: 9781450357135
[148] R. Verdecchia, "Architectural Technical Debt Identification: Moving Forward", Proceedings - 2018 IEEE 15th International Conference on Software Architecture Companion, ICSA-C 2018, pp. 43–44, 2018. DOI: 10.1109/ICSA-C.2018.00018.
[149] M. M. Lehman, "Programs, Life Cycles, and Laws of Software Evolution", Proceedings of the IEEE, vol. 68, no. 9, pp. 1060–1076, 1980, ISSN: 15582256. DOI: 10.1109/PROC.1980.11805.
[150] T. Stablein, D. Berndt, and M. Mullarkey, "Technical debt-related information asymmetry between finance and IT", in 2018 IEEE/ACM International Conference on Technical Debt (TechDebt), 2018, pp. 134–137, ISBN: 9781450357135. DOI: 10.1145/3194164.3194180. [Online]. Available: https://doi.org/10.1145/3194164.3194180.

References

[151] T. Besker, A. Martini, R. Edirisooriya Lokuge, K. Blincoe, and J. Bosch, "Embracing technical debt, from a startup company perspective", Proceedings - 2018 IEEE International Conference on Software Maintenance and Evolution, ICSME 2018, pp. 415–425,2018. DOI: 10.1109/ICSME.2018.00051.
[152] T. Besker, A. Martini, and J. Bosch, "Carrot and stick approaches when managing technical debt", in Proceedings of the 3rd International Conference on Technical Debt, 2020, pp. 21–30, ISBN: 9781450379601. DOI: 10.1145/3387906.3388619. [Online]. Available: https://doi.org/10.1145/3387906.3388619.
[153] M. E. Fonteyn, B. Kuipers, and S. J. Grobe, "A description of think aloud method and protocol analysis", Qualitative health research, vol. 3, no. 4, pp. 430–441, 1993.
[154] G. Çalikli and A. B. Bener, "Influence of confirmation biases of developers on software quality: An empirical study", Software Quality Journal, vol. 21, no. 2, pp. 377–416, 2013, ISSN: 09639314. DOI: 10.1007/s11219-012-9180-0.
[155] A. Jansen and J. Bosch, "Software architecture as a set of architectural design decisions", in Proceedings - 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005, vol. 2005, 2005, pp. 109–120, ISBN: 0769525482.
[156] K. Borowa, A. Zalewski, and S. Kijas, "The Influence of Cognitive Biases on Architectural Technical Debt", in International Conference on Software Architecture (ICSA), 2021.
[157] A. Tang, F. Bex, C. Schriek, and J. M. E. van der Werf, "Improving software design reasoning–A reminder card approach", Journal of Systems and Software, vol. 144, no. April 2017, pp. 22–40, 2018, ISSN: 01641212. DOI: 10.1016/j.jss.2018.05.019.
[158] W. Stacy and J. Macmillan, "Cognitive Bias in Software Engineering", Communications of the ACM, vol. 38, no. 6, pp. 57–63, 1995, ISSN: 15577317. DOI: 10.1145/203241
[159] K. Borowa, R. Dwornik, and A. Zalewski, "Is knowledge the key? an experiment on debiasing architectural decision-making-a pilot study", in International Conference on Product-Focused Software Process Improvement, Springer, 2021, pp. 207–214.
[160] A. Tversky and Kahneman Daniel, "Judgment under uncertainty: Heuristics and biases",Science, 1974, ISSN: 15206882.
[161] K. Borowa, M. Jarek, G. Mystkowska, W. Paszko, and A. Zalewski, Additional Material for Debiasing architectural decision-making: a workshop-based training approach, Zenodo, Jun. 2022. DOI: 10.5281/zenodo.6751990.
[162] K. Borowa, A. Zalewski, and S. Kijas, "The influence of cognitive biases on architectural technical debt", in 2021 IEEE 18th International Conference on Software Architecture (ICSA), IEEE, 2021, pp. 115–125.
[163] S. Brown, The c4 model for software architecture, Jun. 2018. [Online]. Available: https://www.infoq.com/articles/C4-architecture-model/.
[164] J. Saldaña, "The coding manual for qualitative researchers", The coding manual for qualitative researchers, pp. 1–440, 2021.
[165] G. R. Norman, S. D. Monteiro, J. Sherbino, J. S. Ilgen, H. G. Schmidt, and S. Mamede, "The causes of errors in clinical reasoning: Cognitive biases, knowledge deficits, and dual process thinking", Academic Medicine, vol. 92, no. 1, pp. 23–30, 2017.
[166] K. Borowa, M. Jarek, G. Mystkowska, W. Paszko, and A. Zalewski, "Debiasing architectural decision-making: a workshop-based training approach", in European Conference on Software Architecture (ECSA), 2022, pp. 1–8. arXiv: 2206.14701.
[167] M. Galster and D. Weyns, "Empirical research in software architecture—perceptions of the community", Journal of Systems and Software, vol. 202, p. 111684, 2023.
[168] S. Baltes and P. Ralph, "Sampling in software engineering research: A critical review and guidelines", Empirical Software Engineering, vol. 27, no. 4, p. 94, 2022.
[169] U. Van Heesch, P. Avgeriou, and R. Hilliard, "A documentation framework for architecture decisions", Journal of Systems and Software, vol. 85, no. 4, pp. 795–820, 2012.
[170] A. Manjunath, M. Bhat, K. Shumaiev, A. Biesdorf, and F. Matthes, "Decision Making and Cognitive Biases in Designing Software Architectures", Proceedings - 2018 IEEE 15th International Conference on Software Architecture Companion, ICSA-C 2018, pp. 52–55, 2018. DOI: 10.1109/ICSA-C.2018.00022.
[171] K. Borowa, A. Zalewski, and S. Kijas, "The Influence of Cognitive Biases on Architectural Technical Debt", in International Conference on Software Architecture (ICSA), 2021.
[172] E. Løhre and M. Jørgensen, "Numerical anchors and their strong effects on software development effort estimates", Journal of Systems and Software, vol. 116, pp. 49–56, 2016, ISSN: 01641212. DOI: 10.1016/j.jss.2015.03.015.
[173] T. Sharot, "The optimism bias", Current biology, vol. 21, no. 23, R941–R945, 2011.
[174] E. Shalev, M. Keil, J. S. Lee, and Y. Ganzach, "Optimism Bias in Managing It Project Risks: a Construal Level", European Conference on Information Systems, 2014.
[175] C. W. Turner, J. R. Lewis, and J. Nielsen, "Determining usability test sample size", International encyclopedia of ergonomics and human factors, vol. 3, no. 2, pp. 3084– 3088, 2006.
[176] F. Wilcoxon, "Individual comparisons by ranking methods", in Breakthroughs in statistics: Methodology and distribution, Springer, 1992, pp. 196–202.
[177] K. A. Ericsson and H. A. Simon, Protocol analysis: Verbal reports as data, Rev. ed. Cambridge, MA, US: The MIT Press, 1993, pp. liii, 443–liii, 443, ISBN: 0-262-05047-1 (Hardcover); 0-262-55023-7 (Paperback).
[178] Anonymous, Additional Material for Debiasing Architectural Decision-Making: Teaching Software Practitioners, Zenodo, Apr. 2024. DOI: 10.5281/zenodo.11047872.
[179] M. Soliman, M. Wiese, Y. Li, M. Riebisch, and P. Avgeriou, "Exploring web search engines to find architectural knowledge", in 2021 IEEE 18th International Conference on Software Architecture (ICSA), IEEE, 2021, pp. 162–172.
[180] H. Cervantes and R. Kazman, Designing software architectures: a practical approach. Addison-Wesley Professional, 2016.
[181] H. Muccini et al., "Group decision-making in software architecture: A study on industrial practices", Information and software technology, vol. 101, pp. 51–63, 2018.
[182] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in Software Engineering: An Introduction (International Series in Software Engineering). Springer Science & Business Media, 2000, ISBN: 978-1-4615-4625-2.