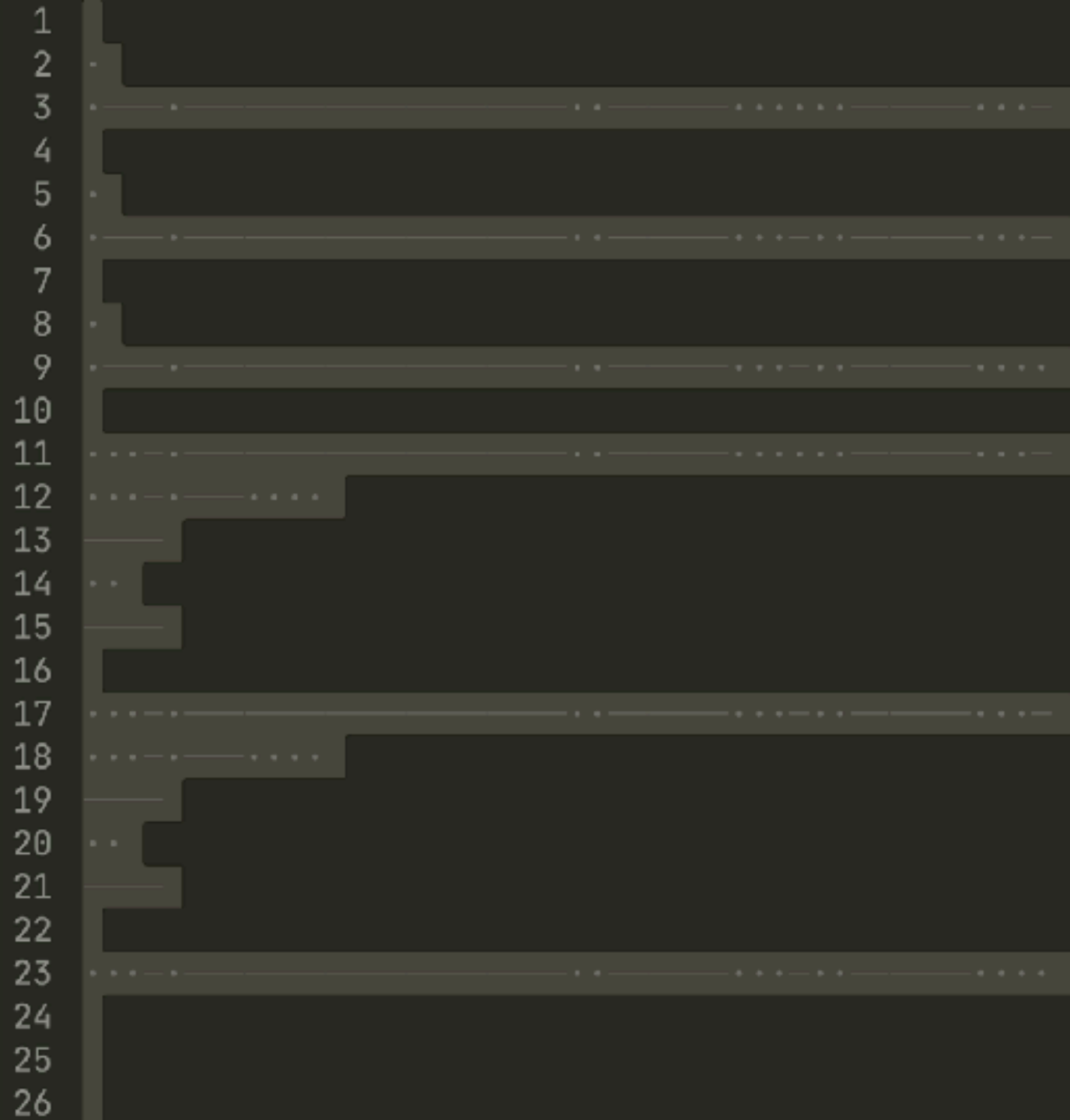


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26



Introduction

- Vadim Zaytsev aka `@grammarware`
 - software evolution ()
 - research ()
 - teaching ()
 - industry ()
- Interests:
 - software language (re)engineering
 - not just Java | Haskell | COBOL



Improving Nothingness. Refactoring Whitespace.

SATToSE, 12 June 2023

Rutger Witmans aka @rwitmans

Dr. Vadim Zaytsev aka @grammarware

Whitespace

- `_ _` = `push`
- `_ \n _` = `dup`
- `_ \n \t` = `swap`
- `_ \n \n` = `drop`
- `\n _ _` = `label`
- `\n _ \n` = `jump`
- `\n _ \t` = `call`
- `\n \t \n` = `return`
- `\n \n \n` = `stop`



Edwin Brady
(U of St Andrews)



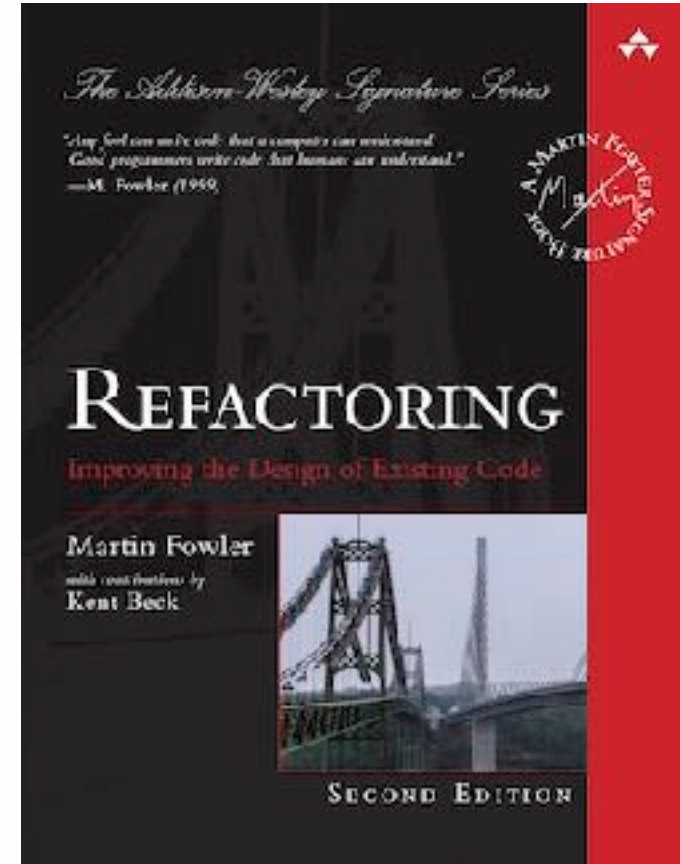
Refactoring

- inspired by *factoring* in mathematics
- William Opdyke's PhD thesis?
- Ralph Johnson's paper?
- Dan Ingalls?
- Forth?



Refactoring

- improving the design
 - of existing code
- **preserve** the behaviour
- **improve** the design
- **minimise** the risk of errors
- each is "*too small to be worth doing*"
 - cumulatively significant



Candidate Refactorings

- Composing Methods
- Moving Features between Objects
- Organising Data
- Simplifying Conditional Expressions
- Simplifying Method Calls
- Dealing with Generalisation
- Code Smells



Candidate Refactorings

- Composing Methods
- Moving Features between Objects
- Organising Data
- Simplifying Conditional Expressions
- Simplifying Method Calls
- Dealing with Generalisation
- Code Smells



Candidate Refactorings

- Composing **Labels**
- Moving Features between Objects
- Organising Data
- Simplifying Conditional Expressions
- Simplifying **Label Jumps**
- Dealing with Generalisation
- Code Smells



Refactorings List

- Extract Method
- Inline Method
- Rename Method
- Consolidate Conditional Expression
- Consolidate Duplicate Conditional Fragments
- Remove Dead Code
- Remove Duplicate Methods



Refactorings List

- Extract Method
- Inline Method
- **Rename Method**
- Consolidate Conditional Expression
- Consolidate Duplicate Conditional Fragments
- **Remove Dead Code**
- **Remove Duplicate Methods**



Details in the paper

- Rust
- <https://github.com/CensoredUsername/whitespace-rs>
- Intermediate Representation
- Focus on size optimisation
- Evaluation is non-trivial
 - in general
 - but ok for Whitespace



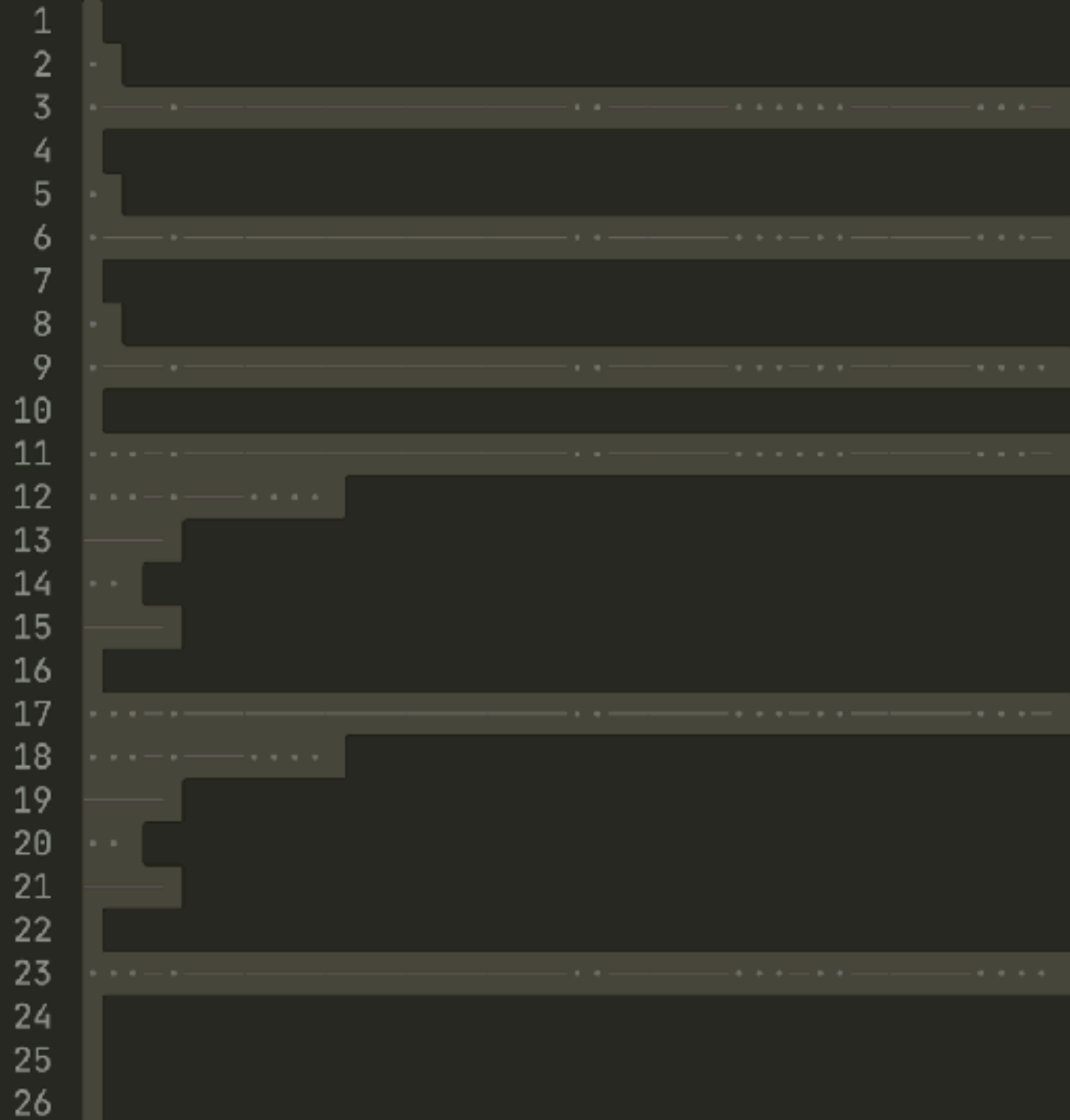
Conclusion

- There is ~~life~~ **research** beyond Java
- Even **Whitespace** can be refactored
- We need more **IR/ASM**-level refactorings
- Conceptually related:
 - **BabyCobol** [ICSME'23] [[SLE'20](#)] [[BENEVOL'20](#)]
 - **Assembler** [[ECMFA'20](#)] [[BENEVOL'20](#)] [[MoreVMs'17](#)] [[SLE'16](#)]
 - **4GLs** [SLE'23] [[ICTD'19](#)] [[PX/17.2](#)]
 - **CSS** [[SLE'16](#)] [[SATToSE'16](#)] [[SATToSE'16](#)] [[ICSME'16](#)]
 - **Python** [[SLE'21](#)] [[<Programming>'17](#)]
- Let's talk about refactoring **detection!**



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21





Thank you!

- Takeaways:
 - There is ~~life~~ **research** beyond Java
 - Even **Whitespace** can be refactored
 - We need more **IR/ASM**-level refactorings
 - Conceptually related:
 - **BabyCobol** [[ICSME'23](#)] [[SLE'20](#)]
 - **Assembler** [[ECMFA'20](#)] [[BENEVOL'20](#)] [[MoreVMs'17](#)] [[SLE'16](#)]
 - **4GLs** [[SLE'23](#)] [[ICTD'19](#)] [[PX/17.2](#)]
 - **CSS** [[SLE'16](#)] [[SATToSE'16](#)] [[SATToSE'16](#)] [[ICSME'16](#)]
 - **Python** [[SLE'21](#)] [[<Programming>'17](#)]

