Coding Traditions

Software Systems — Design — L6T4

Dr. Vadim Zaytsev aka @grammarware, November 2020

UNIVERSITY OF TWENTE.

Formal Methods & Tools

Coding Traditions

- Idioms
  - [x*x for x in X if x < 10]
- Implementation patterns
  - caching / memoisation
- Calling conventions
  - push/pop
- Naming conventions
  - CamelCase, L6T4
- Formatting conventions
  - {}
- Code snippets
  - System.out.println();
- Micropatterns
  - Box
- Templates
- . . .

Anti-patterns

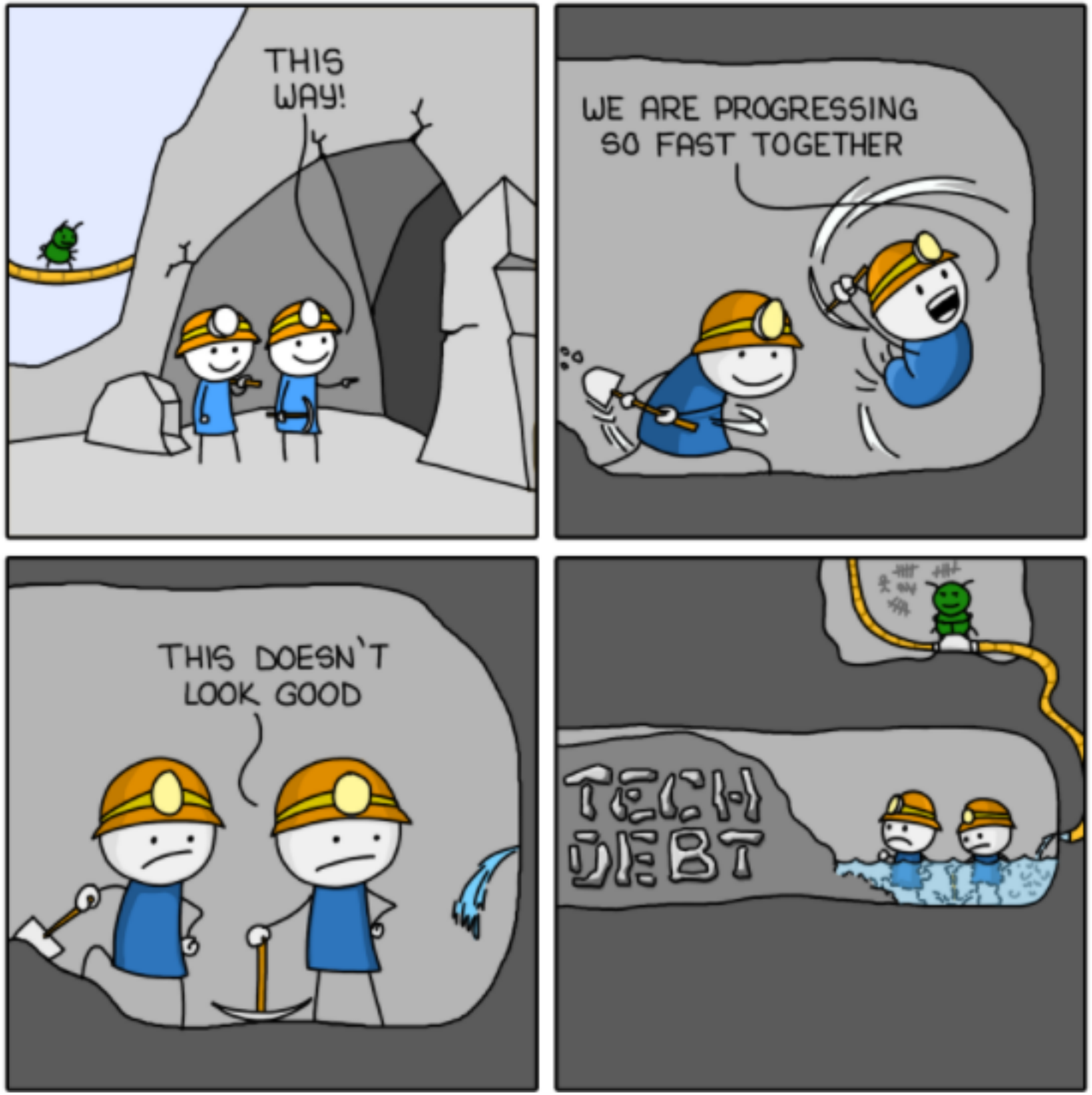
- Copy-paste programming
- Cargo cult programming
- Death march
- Shotgun debugging
- Premature optimisation = $\sqrt{\text{evil}}$

Code Smells

- Vendor Lock-in
- Cyclic Hierarchy
- Useless Control Flow
- Duplicate Code
- More is Less
- Eager Test
- Compound Attribute
- 1SidedRecursion

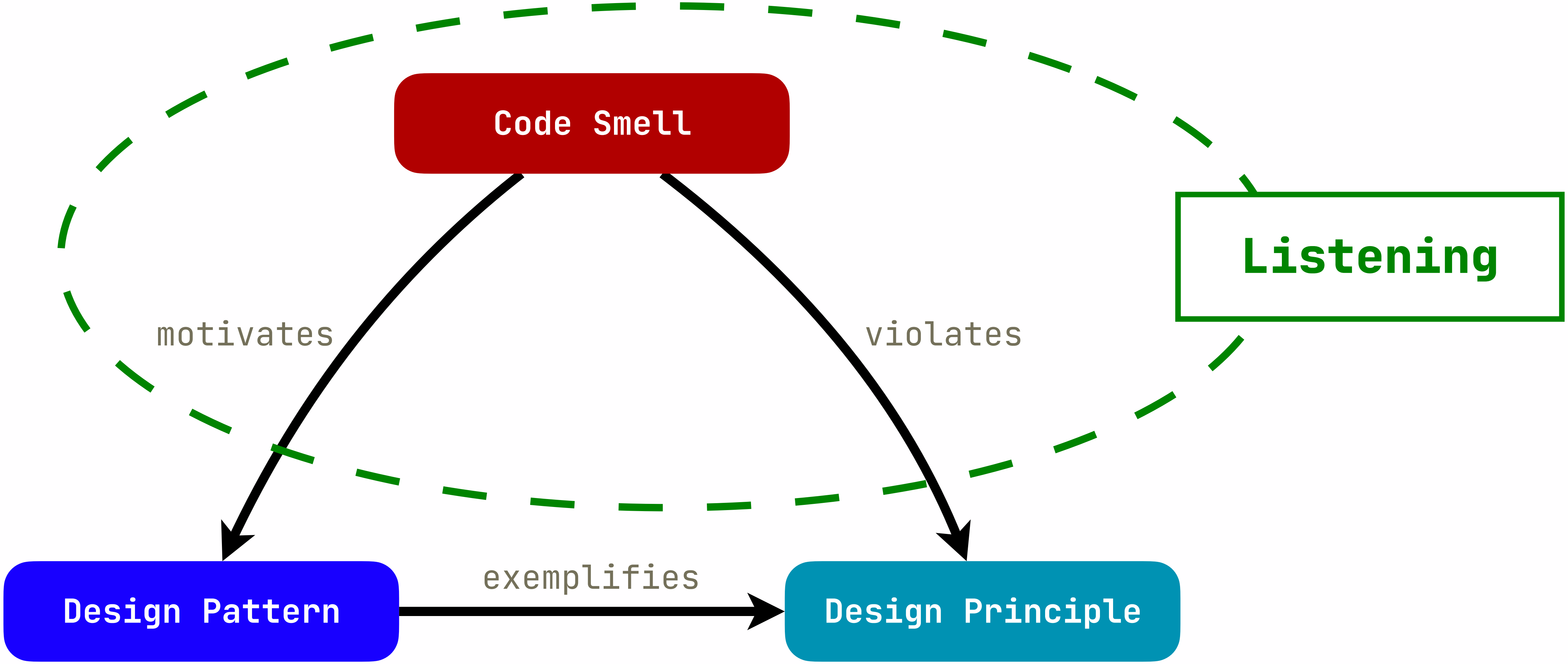
Tushar Sharma, http://www.tusharma.in/smells, 2020

Technical Debt



by MonkeyUser, https://www.monkeyuser.com

Listen to the Smells



E. Baniassad et al, *Learning to Listen for Design*, Onward! 2018

Conclusion

- Coding traditions
  - will return in [P3M]
- Anti-patterns
  - bad practices
- Smells
  - energy, design, grammars, …
- Listen to smells
- Avoid technical debt
- Checkstyle!

Topics/Slides Disclaimer

- Good ✅
  - watch before Q&A
  - embrace reality
  - try out at labs
  - ask for feedback
  - apply to project
  - dig deeper
  - recall from slides
- Bad ❌
  - slides over videos
  - assumptions
  - blanks
  - timing

UML®

Unified
Modeling
Language