

Why Bad Code Quality Is Not (Only) Your Fault

Annelore Egger



OPENVALUE
Better software, faster

Ask Questions on Slido



- slido.com
- [#JCON2025](https://twitter.com/JCON2025)
- Cinema Black Box
- Upvote questions you are interested in! 👍

About Me

- BSc in Computer Science
- 8 years of programming experience
- Software Engineer, OpenValue Switzerland since May 2024

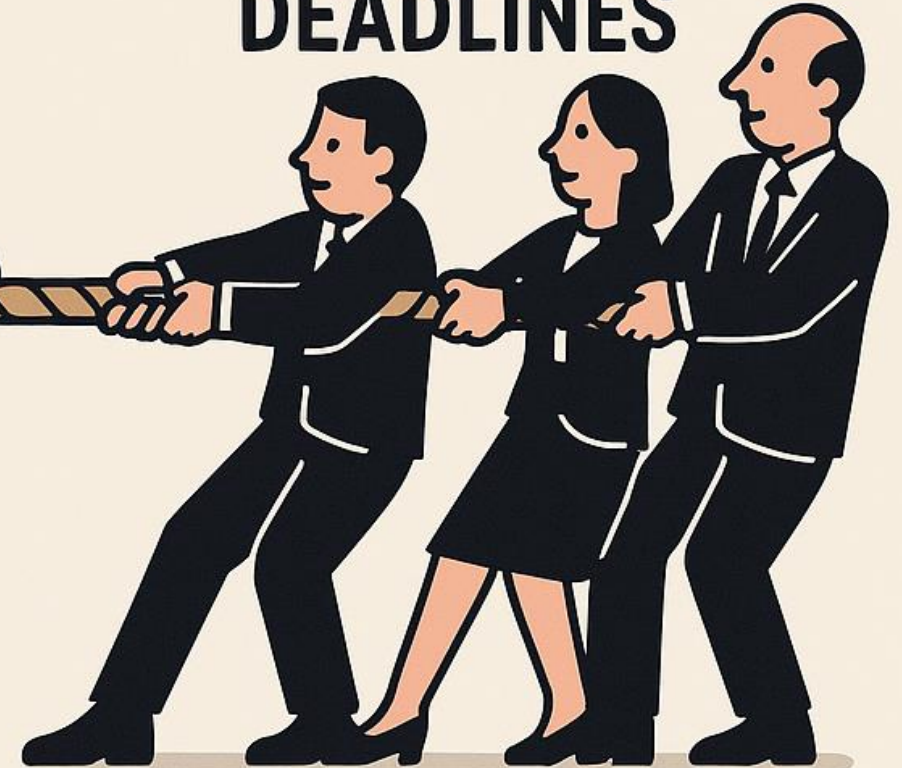


THE TUG-OF-WAR ON CODE QUALITY

DEVELOPERS

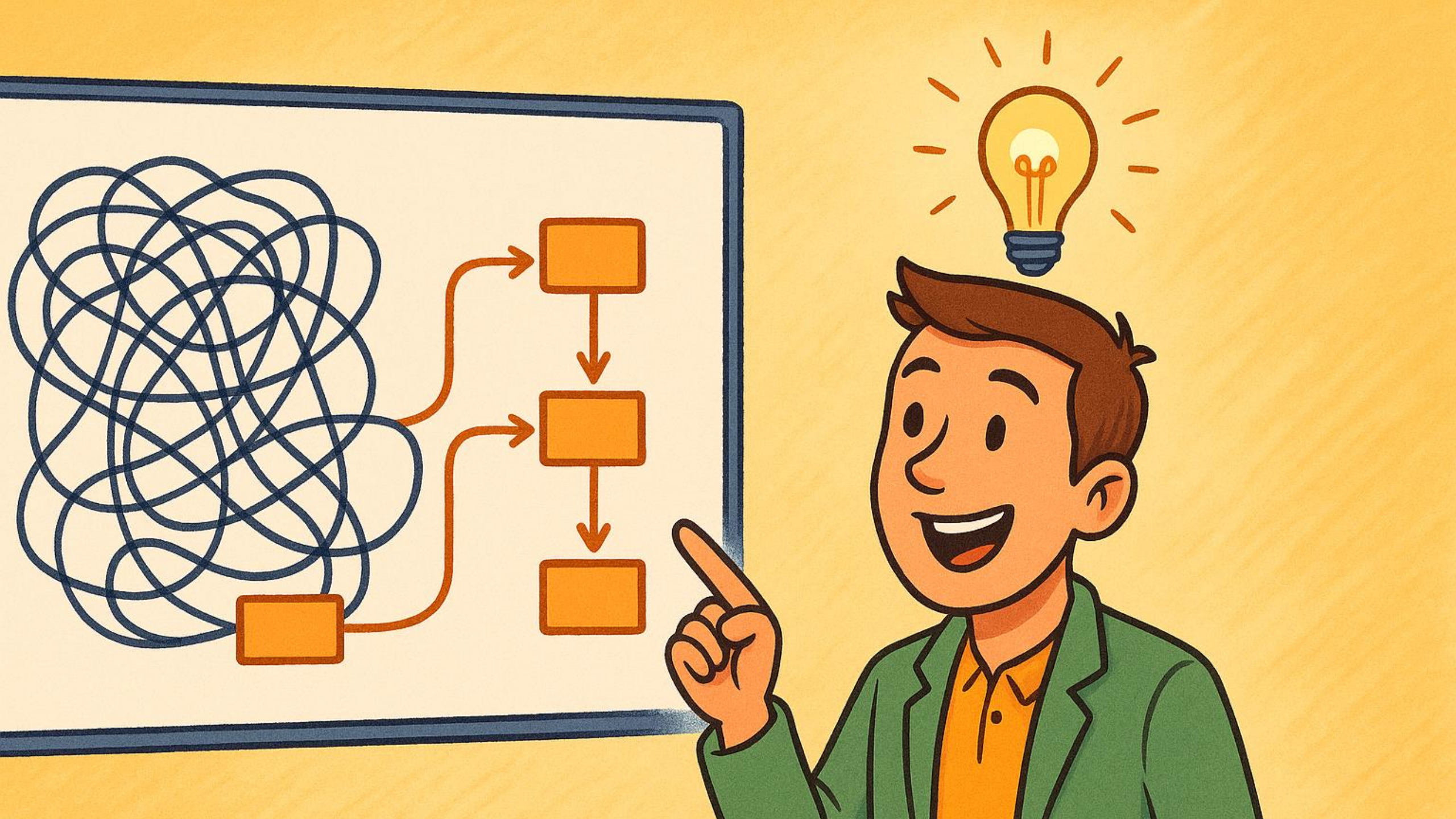


**BUSINESS
MANAGERS
DEADLINES**

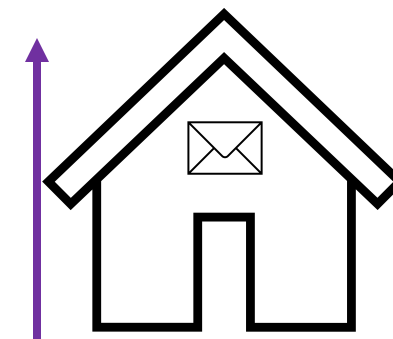
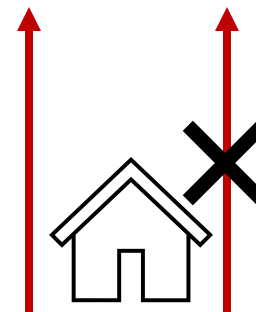
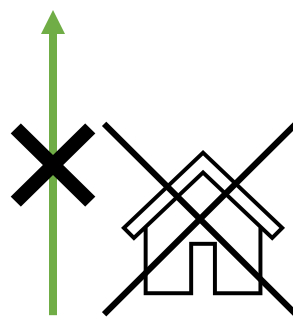
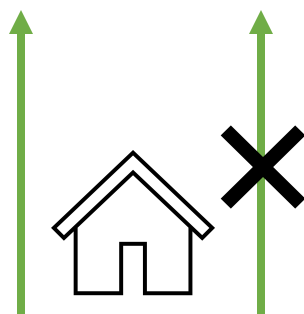
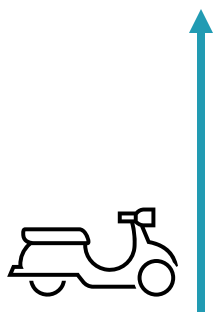


Code Quality

code that is easy to understand, easy to
change, and less likely to break
unexpectedly



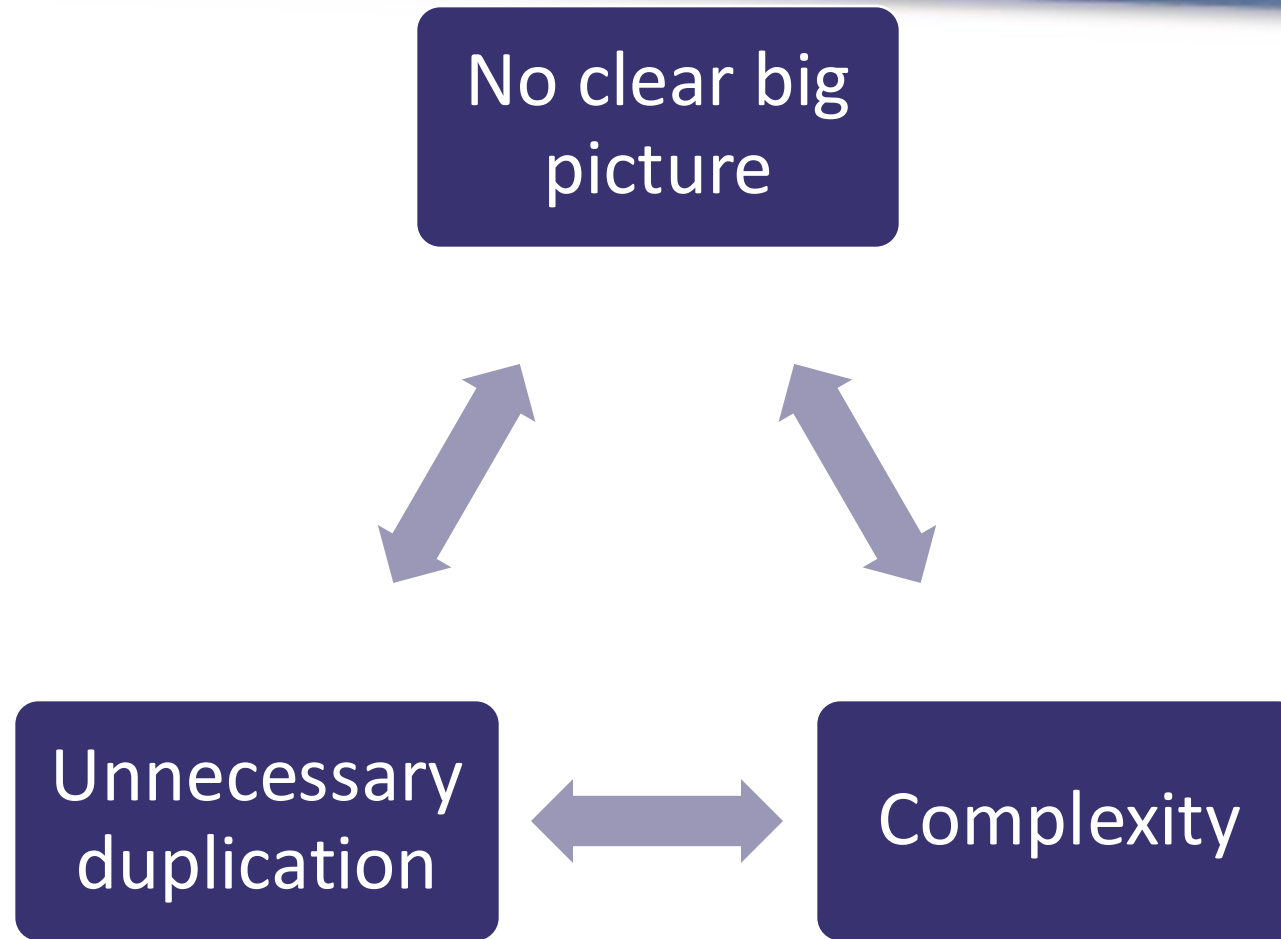
Example



OPENVALUE
Better software, faster

Factors That Impact Code Quality (and What We Can Do About Them)

Underlying Issues



Miscommunication and Misunderstandings

Misunderstanding Between Business and Development

- Business and developers do not speak the same language
- Try to get rid of assumptions
 - Ask questions
 - Give background/context-information

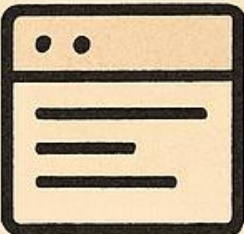
Quote: Business-Developer Misalignment

“What is deployed to production is your
developer’s misunderstanding”
- Jacqui Read

Mitigating Misunderstandings

- [OpenValue courses](#)
 - IT development for non-IT people
 - DevOps for Managers
 - Effective Communication for software developers
- Accompany users when they are using your application

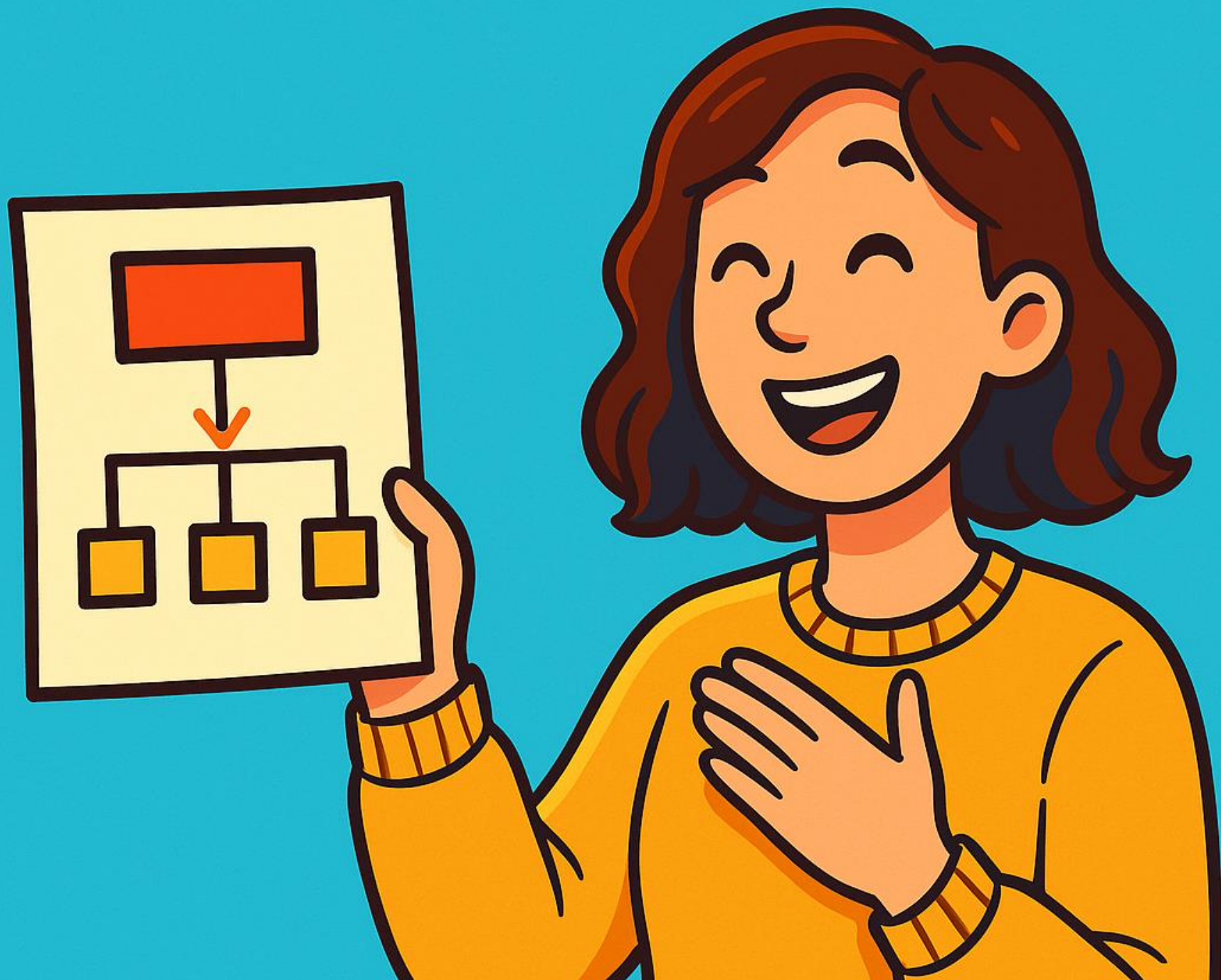
INTERFACE



```
interface  
  Foo  
{  
  bar();  
}
```



Diagrams



Using Diagrams to See the Big Picture

- Existing diagrams may not be enough
- Use the “right” diagram
 - should help to understand a specific concept
- Use a legend, so everyone knows how to use the diagram

Recommendation



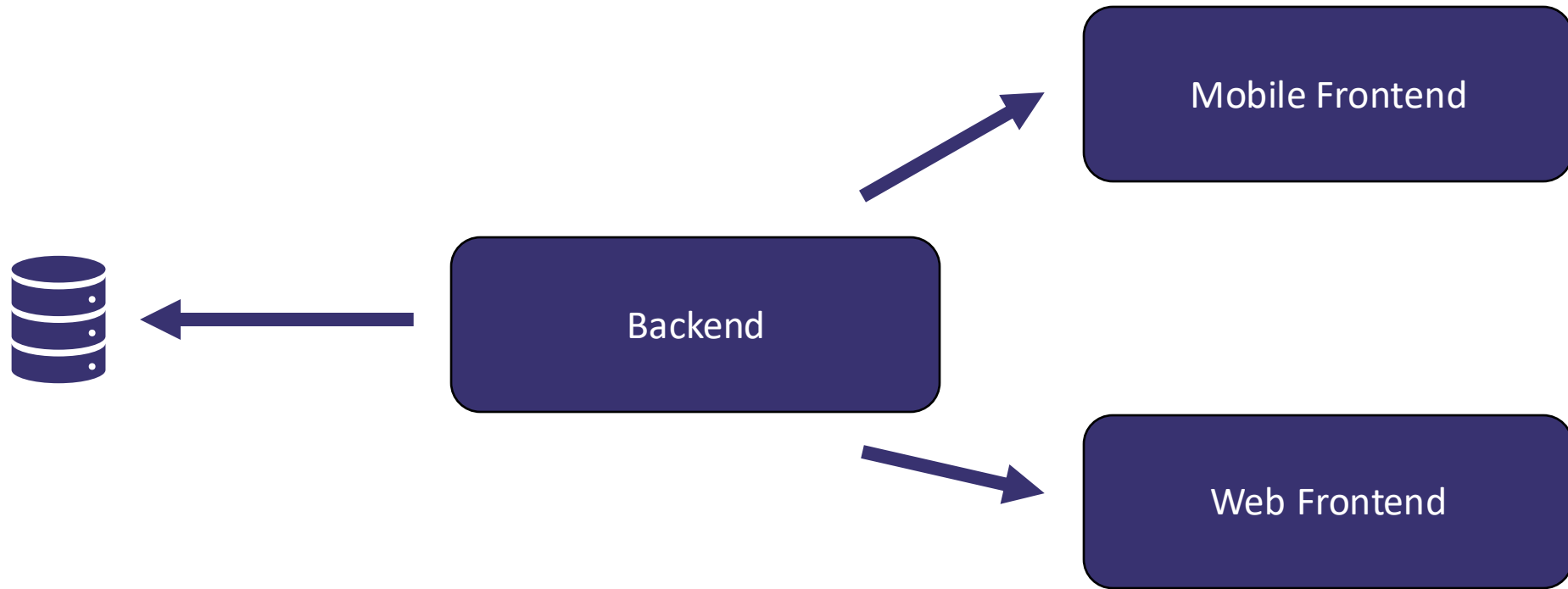
Jacqui Read - Mastering Visual Communication for Software Development - NewCrafts 2024

OPENVALUE
Better software, faster

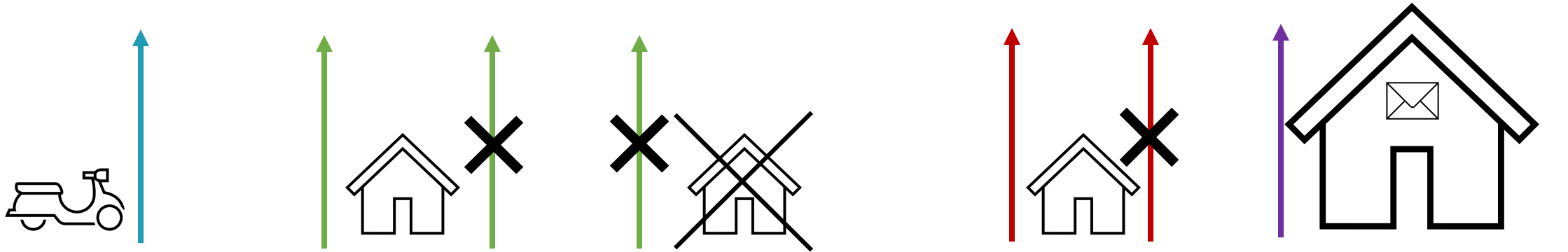
Mastering Visual Communication for Software Development (Jacqui Read)

- Different Audiences need different levels of detail and different types of diagram
- Consider color-blindness: at least check in greyscale
- High-level diagrams stay up to date for the longest time

Big Picture



Intermediate Big Picture



OPENVALUE
Better software, faster

Organizational Scale and Knowledge Silos

Knowledge Fragmentation

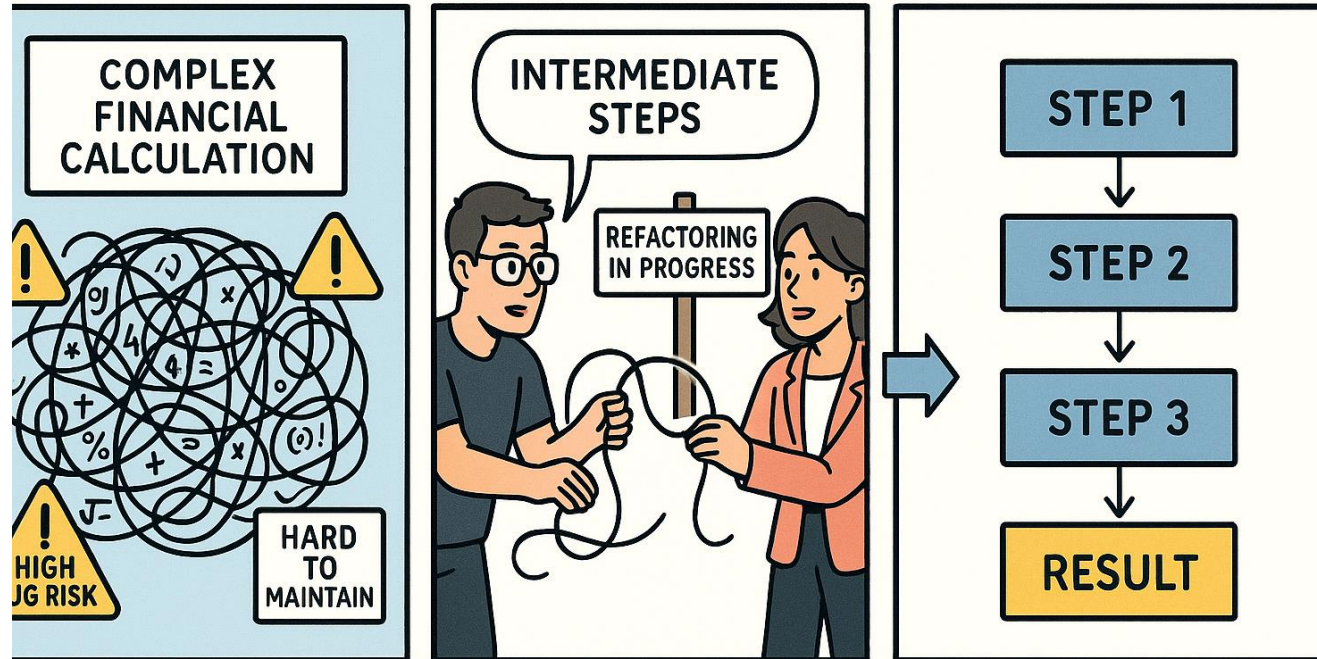
- Most developers do not know enough about the entire project
- Knowledge mostly only of one part of the project
- Combining different modules: one of the most difficult aspects
- Knowledge is not shared enough → collaborate more
 - Pair-Programming
 - [Bytesize Architecture Sessions](#)

Improving Documentation

- Diagrams & documentation should be understandable on its own
 - No one can remember everything
 - A big help when on-boarding employees
- Decisions should be documented: e.g. [decision logs](#)

Inherent Domain Complexity

Some Complexity Cannot Be Simplified



**YOU CAN'T ALWAYS REDUCE COMPLEXITY –
BUT YOU CAN MAKE IT MANAGEABLE.**

Developer responsibility

- Static code analysis
- Code reviews
- Sticking to guidelines

Conclusion

- Code Quality is a shared responsibility
- Work on continuously improving the communication
 - Be aware of your assumptions
 - Use diagrams

Thank you for your attention



✉ annelore.egger@openvalue.ch

🐙 [anneloreegger](https://github.com/anneloreegger)

in [/in/anneloredev](https://www.linkedin.com/company/anneloredev)

✂ [@AnneloreDev](https://twitter.com/AnneloreDev)

▶ [@anneloredev](https://www.youtube.com/channel/UC...)

Further Resources

- [Devovx Greece 2024 - Knowledge Management for the Technically Inclined by Jacqui Read](#)