To Inherit, Or Not To Inherit?

Leanne Matuszyk

A long time ago...

I moved from:

- Australia to Switzerland.

- a job as a researcher to a software engineer.

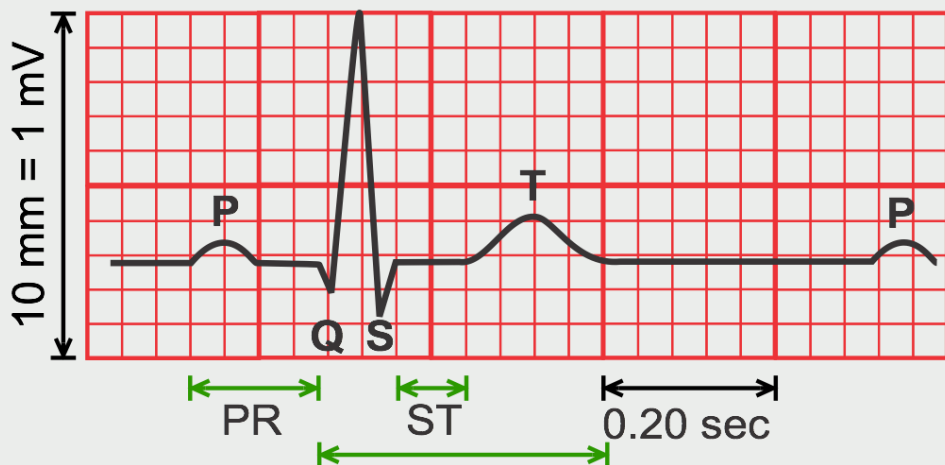- an English-speaking environment to a Swiss-German one.

https://adigaskell.org/2016/02/24/teaching-machines-to-appreciate-images/seeing-machines/

https://cvphysiology.com/Arrhythmias/A009

A Domain Change

- Programming language remained C++

- From head and eye-gaze tracking using computer vision.

- To medical devices for diagnosing and monitoring the heart and lungs.

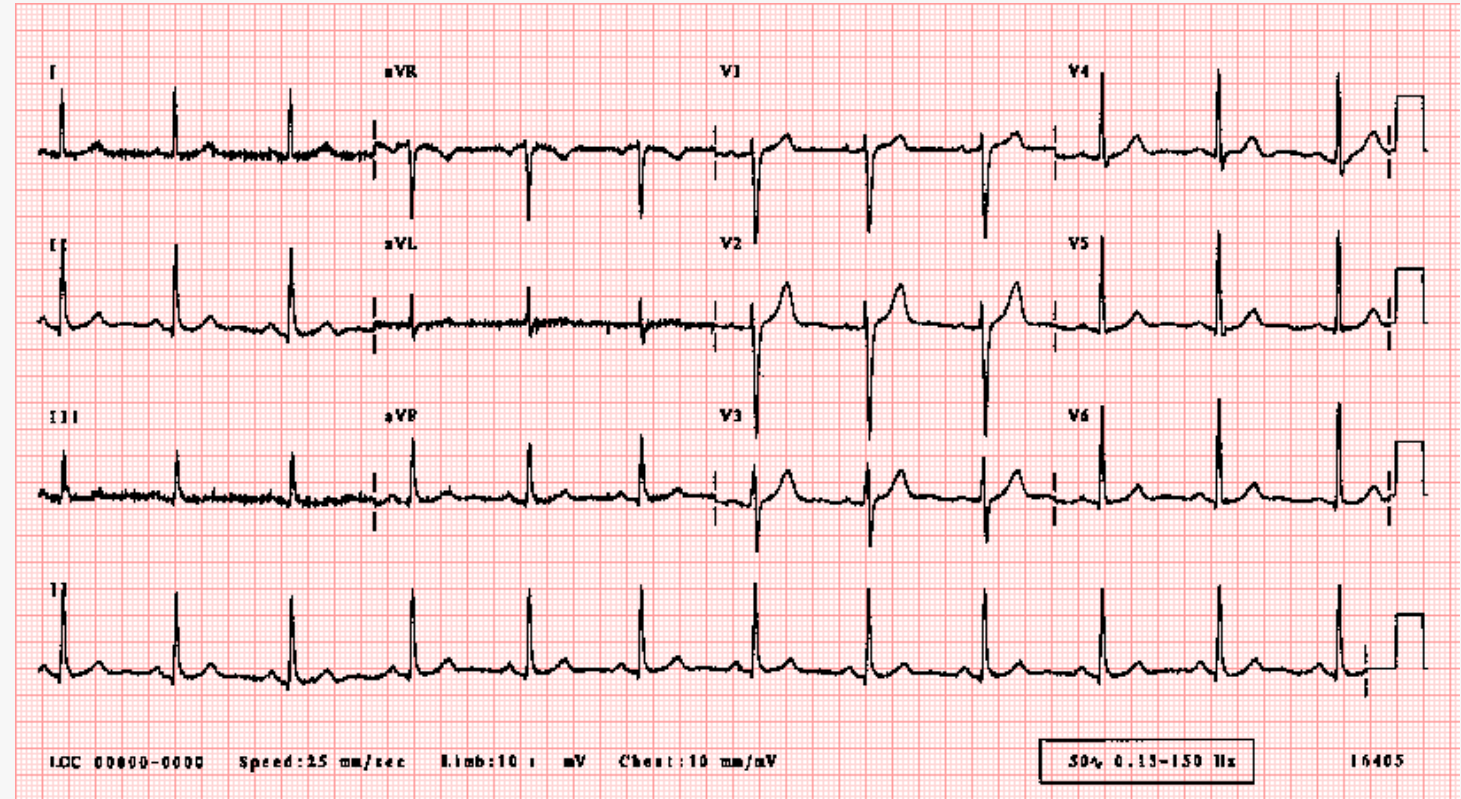- First project was looking into diagnostic recordings.

https://urgentmednetwork.com/three-things-ekg-can-tell-hearts-health/

ECG Recording Types

- Resting ECG (10s)

- Exercise ECG (~20 minutes)

- Holter (hours)

- ...

https://www.indiamart.com/proddetail/computerized-treadmill-exercise-test-service-7714876748.html

ECG Recordings

Data:

- Heart signals
  (up to 12 leads, 60 Hz)

- Blood oxygen levels (10 Hz)

- Heart rate (1 Hz)

- Events

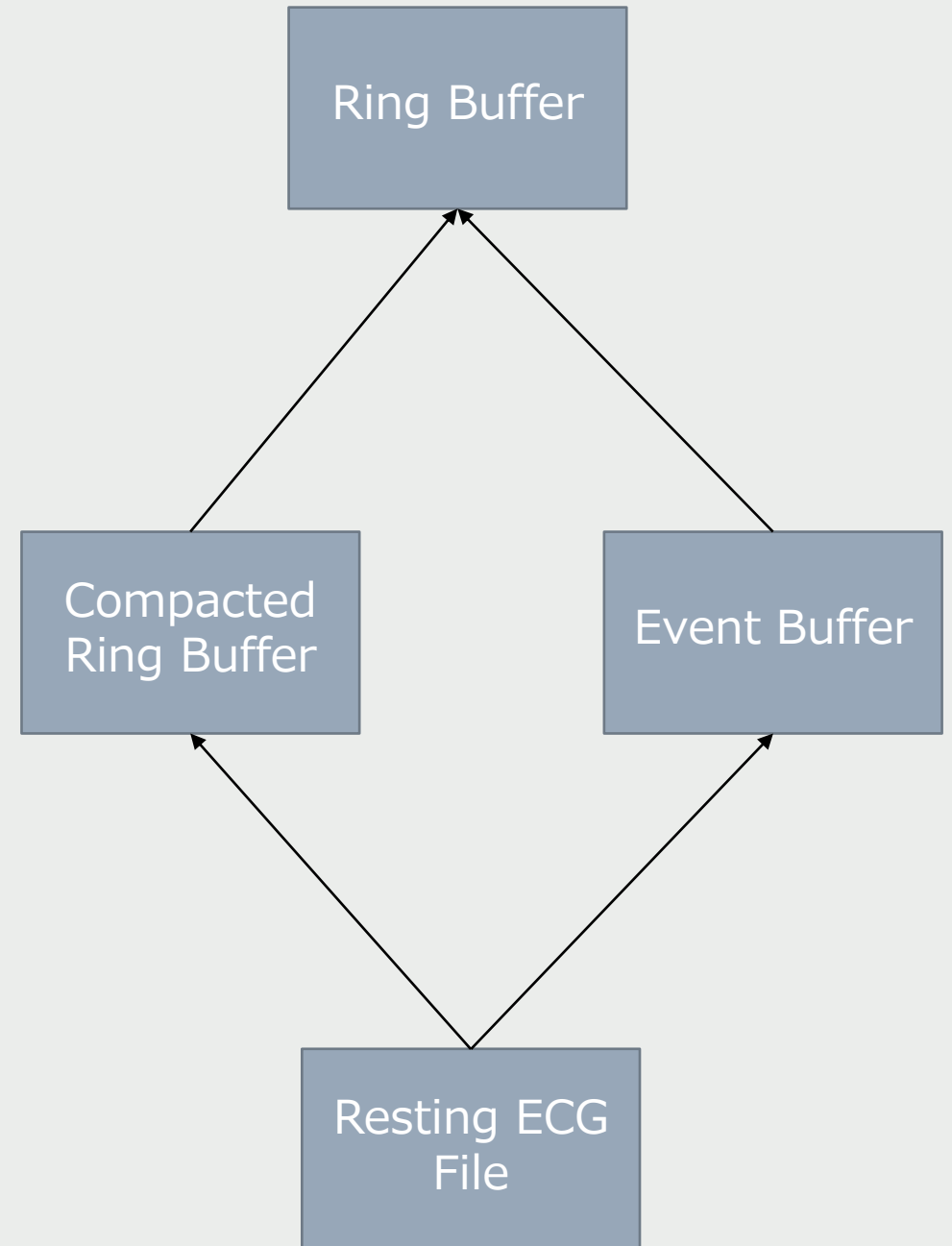– *Something happening at a particular point in time*

- ...

https://ecglibrary.com/norm.php

Understanding the Code

- Interesting domain.
- A huge struggle for me.
- Existential crisis.
- Eventually calmed down.

Breaking Down the Problem

- Systematically analyse the code.

- Tried to write test cases.

- UML class diagrams.

- Something like this:

Ring Buffer

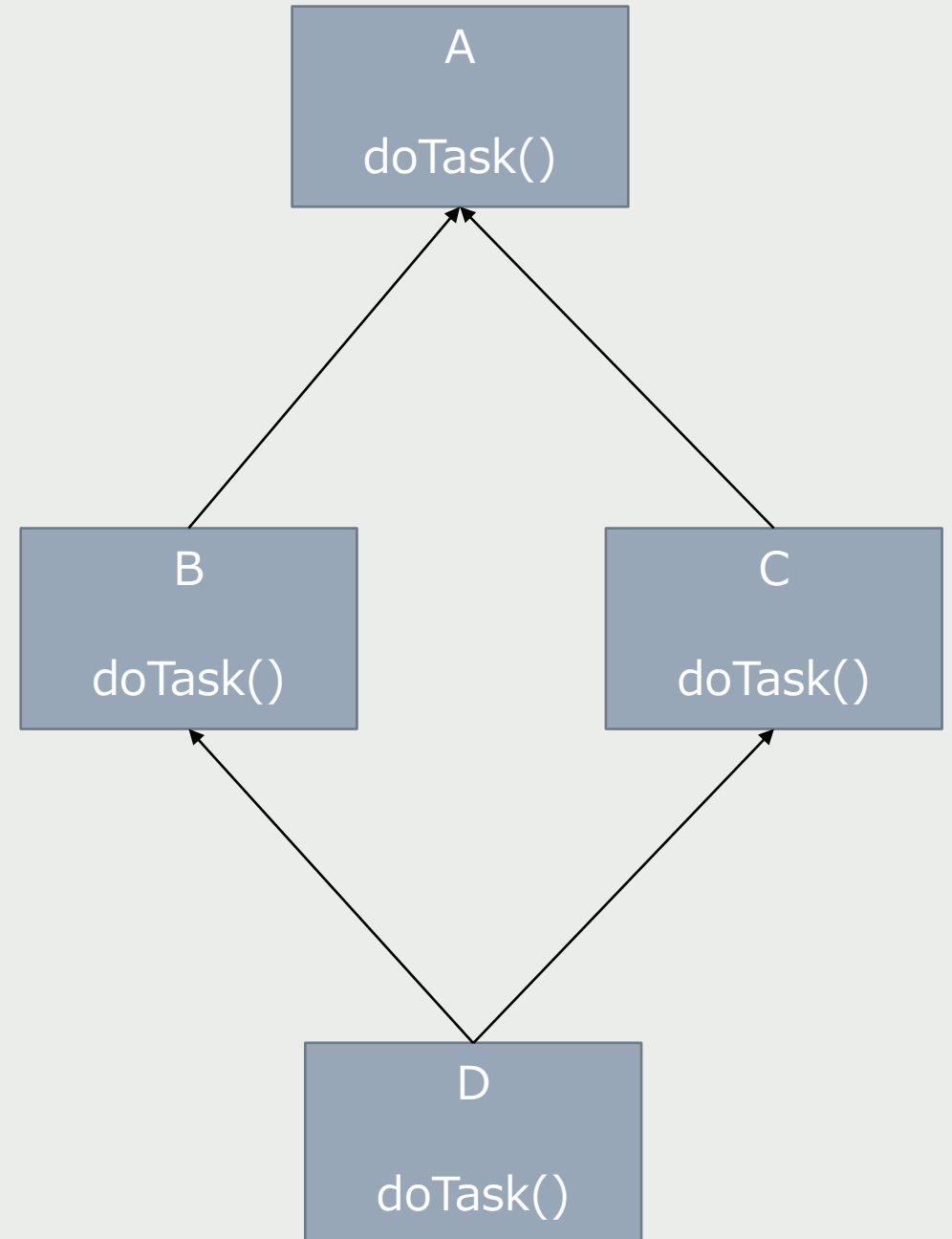Compacted Ring Buffer

Event Buffer

Resting ECG File

Diamonds are a girl's best friend
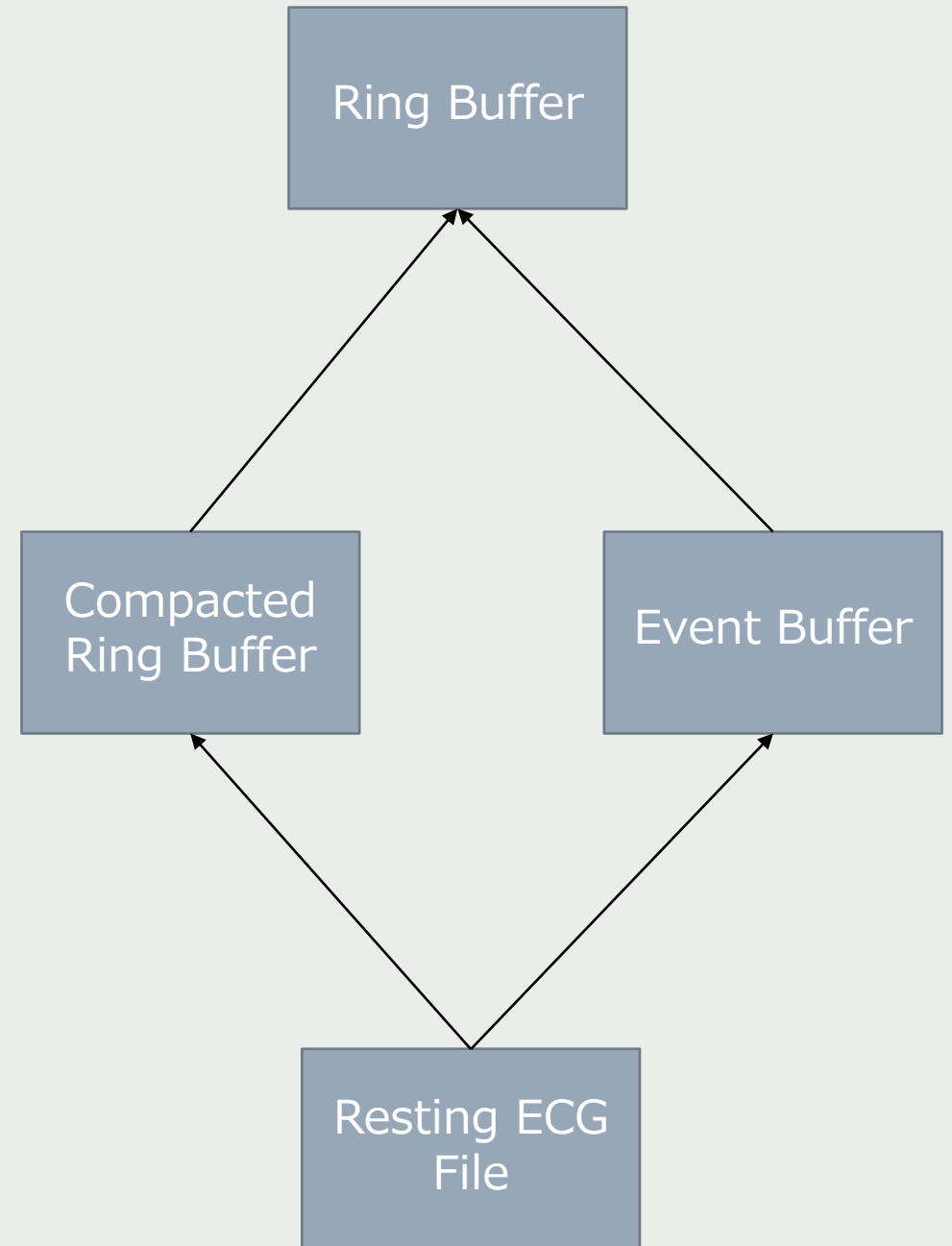
Or are they?

Deadly Diamond of Death

- A common argument against multiple inheritance.

- High complexity and ambiguity

- Which version of the method doTask() does D inherit? From A, B or C?

- Always possible in C++

- From Java 8 with default methods on interfaces.
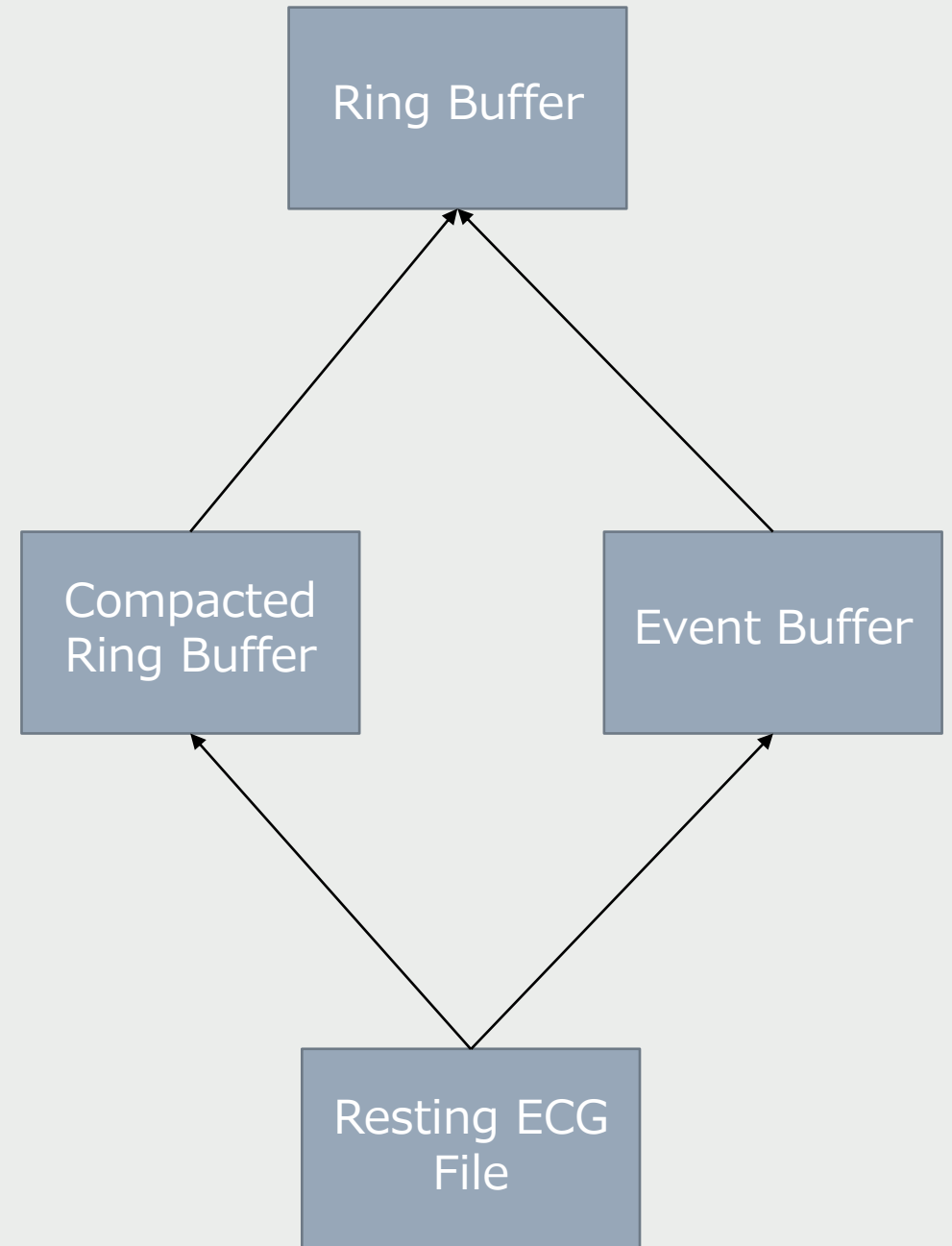
Why is the Diamond a bad thing?

Core Principles:

- KISS
  - *not simple!*
- YAGNI
  - *file compaction?*

Ring Buffer

Compacted Ring Buffer

Event Buffer

Resting ECG File

Why is the Diamond a bad thing?
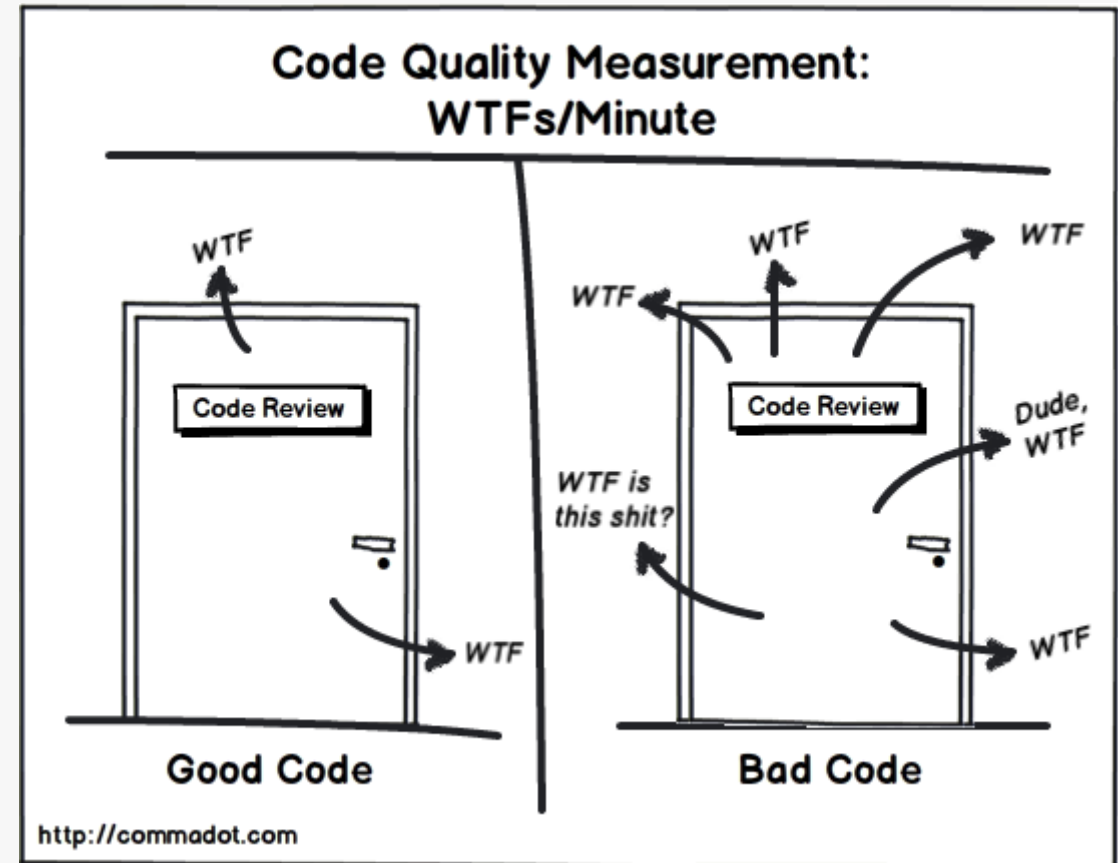
Code Smells:

- Shotgun surgery.

- Refuse bequest.

- Inappropriate intimacy.

Why is the Diamond a bad thing?
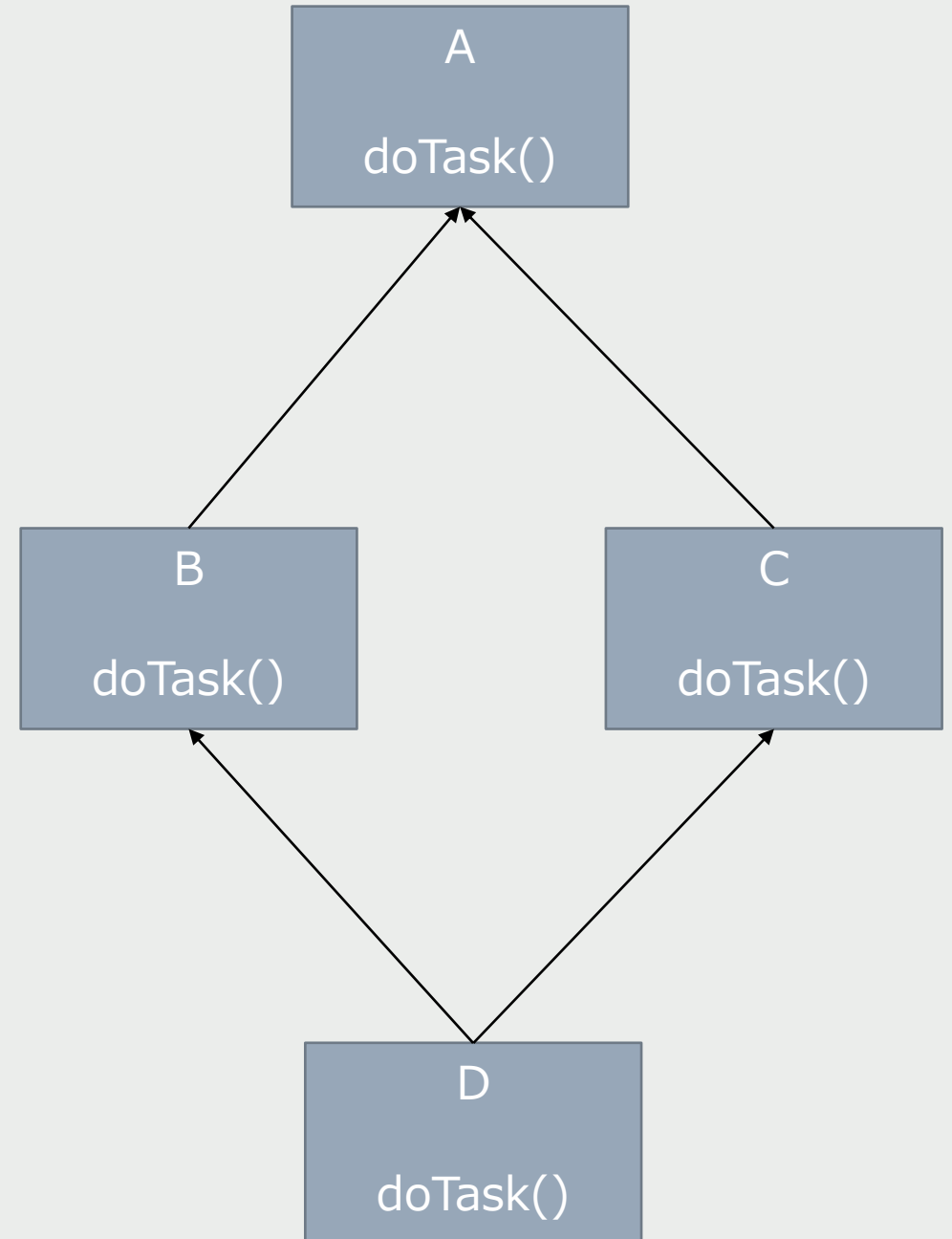
SOLID++

- Single responsibility

- Open/closed principle

- Liskov substitution fail!

- Interface segregation

- Least astonishment (WTF)

Code Quality Measurement: WTFs/Minute
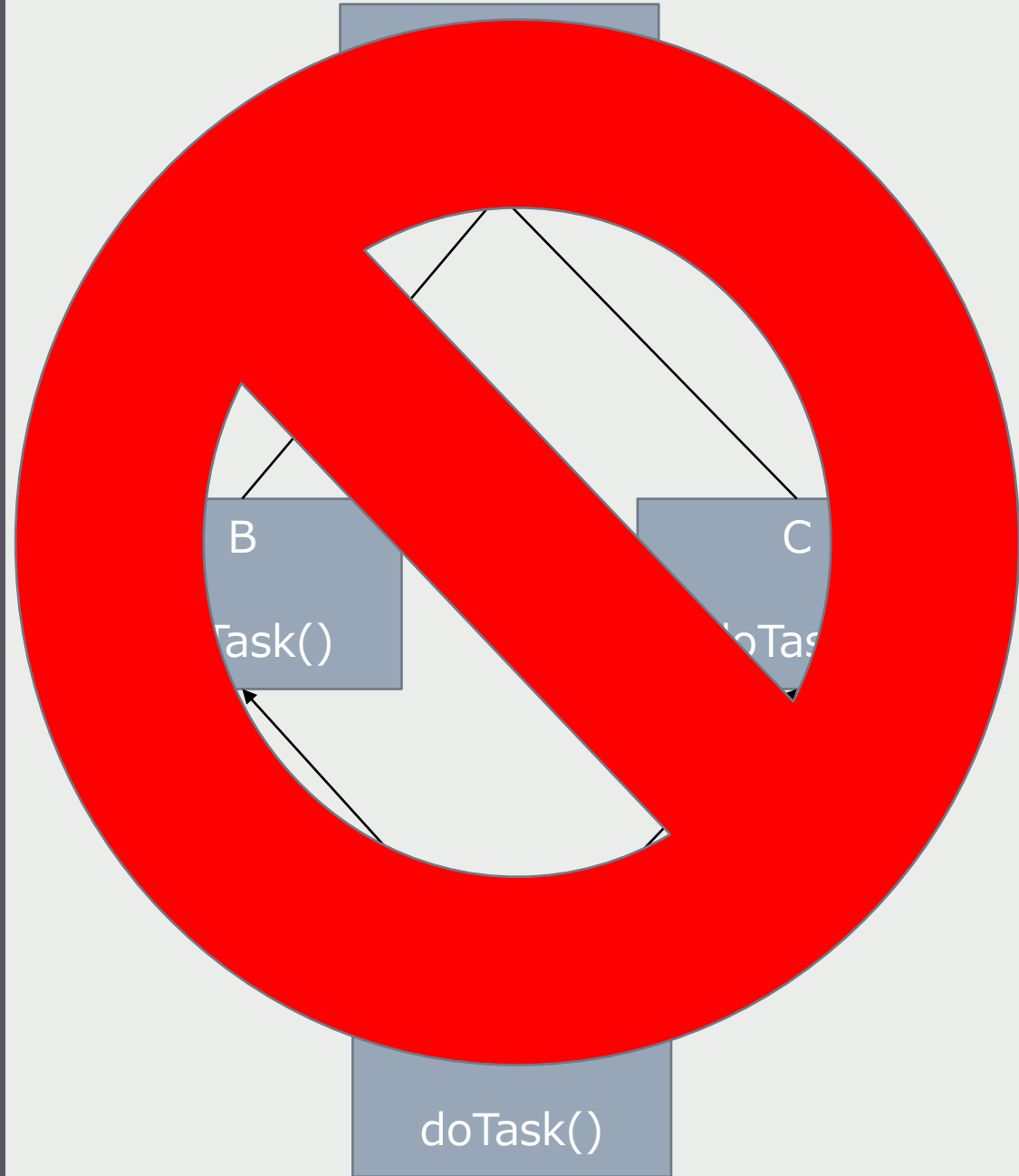
Other Problems

- Implementation was buggy.

- Files were regularly corrupted.

- Not acceptable for medical device.

- Testing/debugging by hand using application for visualizing data in files.

A
doTask()

B
doTask()

C
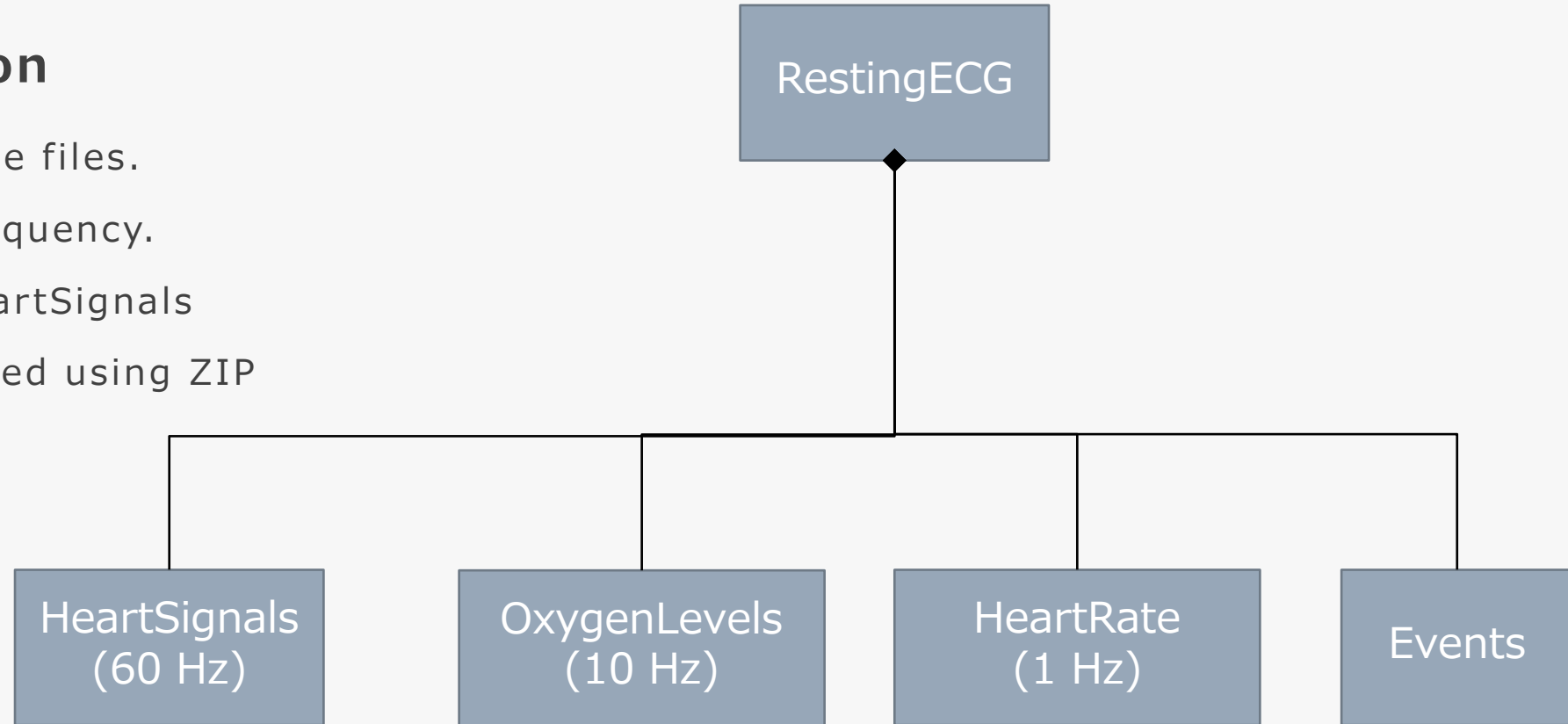doTask()

D
doTask()

Solution

Redesign:

- New implementation.

- Comprehensive unit tests.

Solution

Mostly Composition

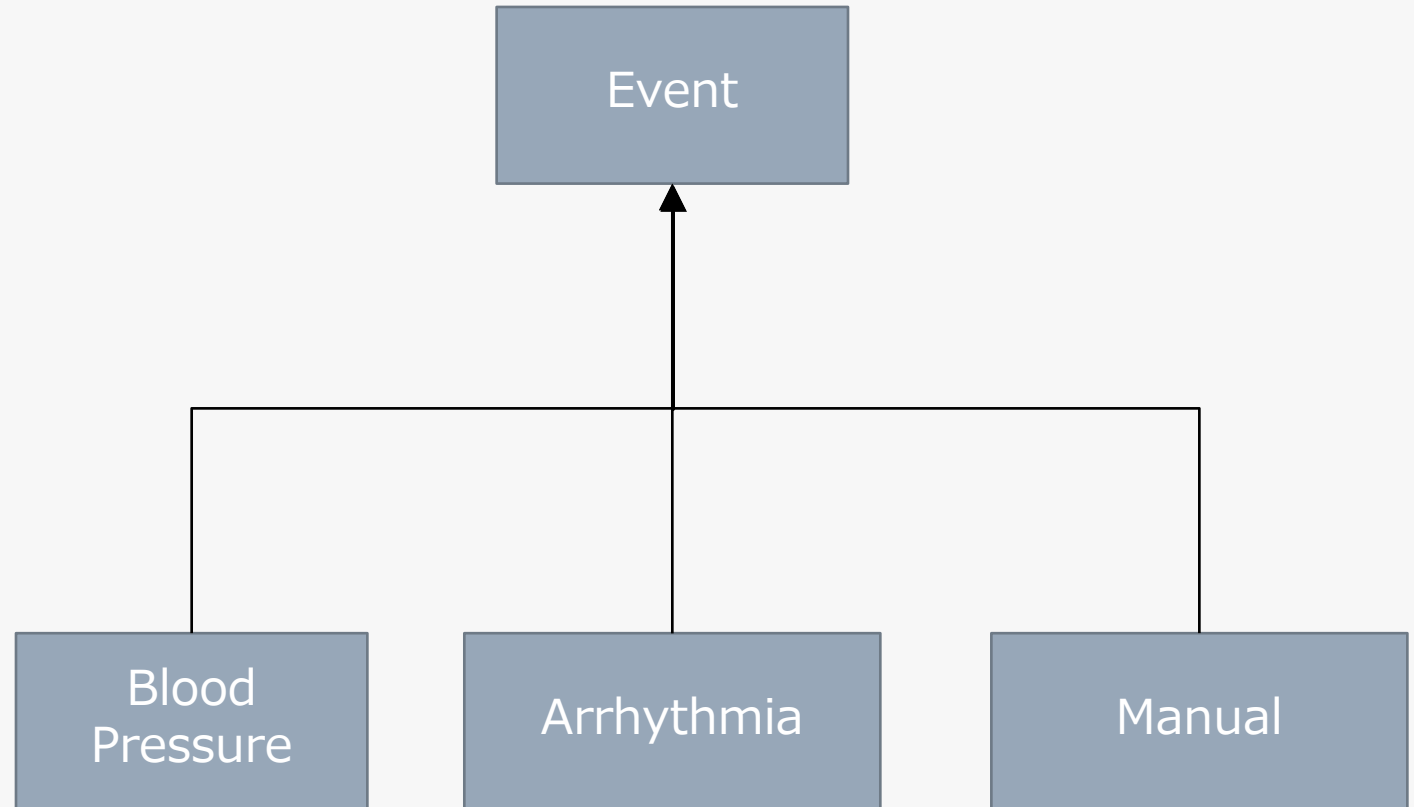- Data stored in multiple files.

- Data separated by frequency.

- A RestingECG **has** HeartSignals

- Multiple files compacted using ZIP (YAGNI).

```
                              RestingECG
                                  |
        ┌──────────────┬──────────┴──────────┬──────────────┐
   HeartSignals    OxygenLevels         HeartRate        Events
    (60 Hz)          (10 Hz)             (1 Hz)
```

Solution

Some Inheritance

- Event:

- *Something happening at a particular point in time.*

- A blood pressure measurement
  is a kind of Event.

- Passes the Liskov substitution test.

- Each event knows how to read and write itself
  to/from an event file.

Conclusion

To inherit, or not to inherit?

- Inheritance coupling is the strongest.

- Test design with **LSP**.

- Only for **same-kind-of** relationship.

- Otherwise, use **composition**.

COMPOSITION OVER INHERITANCE

PREFER YOU MUST

https://medium.com/aviabird/ruby-composition-over-inheritance-3ff786ad9e5d

QUESTIONS?

THANK YOU

Leanne Matuszyk

leanne.matuszyk@css.ch

References

1. https://en.wikipedia.org/wiki/Multiple_inheritance#The_diamond_problem

2. https://www.makeuseof.com/what-is-diamond-problem-in-cpp/

3. https://www.javatpoint.com/what-is-diamond-problem-in-java