RETROSPECTIVE

HOW CLEAN IS A 2+ YEAR OLD RELEASE UNIT

Tamás Günther, 25.05.2021

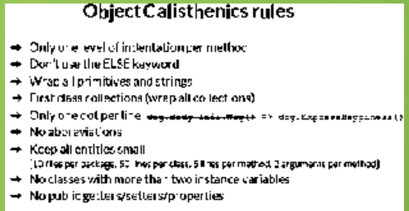
PARTNER-PARTNER KEY FACTS

- Green-field project - no legacy code
- Built with Clean Architecture and DDD in mind
- First iteration: no mob

- Core domain functionality for 8-10 business aggregates
  - CRUD of timeline
  - event-notifications

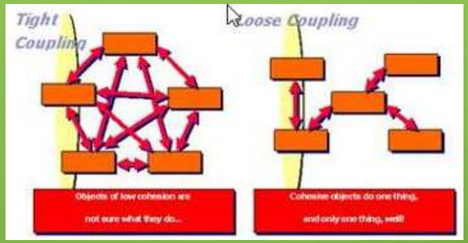
MAIN FOCUS

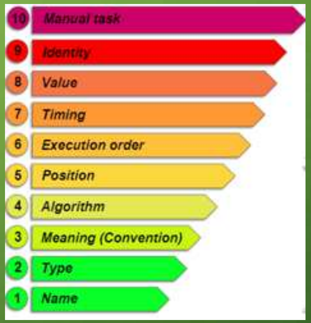
- Object calisthenics
- Coupling/cohesion
- Connascences
- Test
- Onion architecture

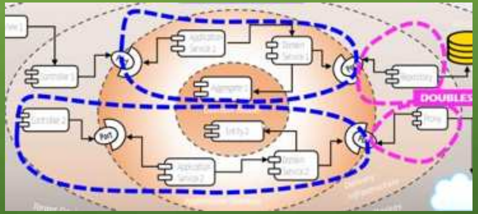


Object Calisthenics rules

- → Only one level of indentation per method
- → Don't use the ELSE keyword
- → Wrap all primitives and strings
- → First class collections (wrap all collections)
- → Only one dot per line ~~dog.Body.Tail.Wag()~~ => dog.ExpressHappiness()
- → No abbreviations
- → Keep all entities small
  (10 files per package, 50 lines per class, 5 lines per method, 2 arguments per method)
- → No classes with more than two instance variables
- → No public getters/setters/properties


Tight Coupling
Loose Coupling
Objects of low cohesion are not sure what they do...
Cohesive objects do one thing, and only one thing, well!





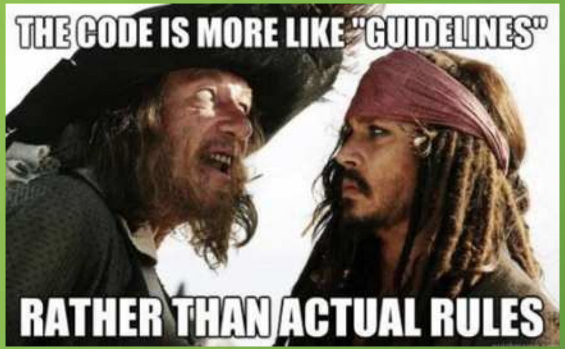
THE DIFFICULT ONES

- Cohesion: timeline-functionality is not closed for all its operations

- Execution order - not enforced on timeline-builder

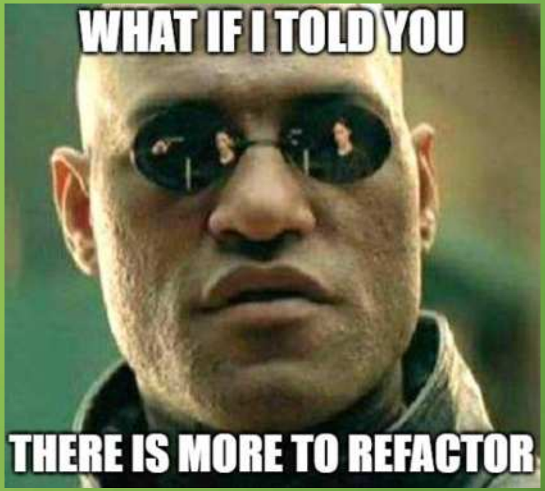
- Timing – related to JPA/JEE Transactions

THE PIRATE CODEX ONES



- Many dependencies – in a few classes
- Sometimes more than two (business) parameters

THE PROBABLY NOT PERCEIVED



- Many params → some cross cutting issues are scattered overall
- Avoidable coupling → use composition
- Setters…

THE GOOD ONES

- Test Mocks/Fakes are used correctly
- Wrapping primitives: custom types everywhere
- Naming: sometimes long, but good
- Connascences of Meaning/Value
- And many others

BUT ALSO SOME WTFS…

- Using optionals sometimes: wtf?
- Unnecessary using / over-complicating of functional interfaces

CONCLUSIONS

- Object calisthenics → fulfilled, with few exceptions
- Coupling/cohesion → potential of improvement
- Connascence → 8 of 10 OK
- Test → only some naming issues
- Onion architecture → fulfilled

MY TAKEAWAYS – FOR ALL OF US

- Built in java features (optionals, streams): more practice needed
- Beyond lombok → Builder(-api)s help accomplish cohesion
- Big packages with loose coupled classes → some code-reorg needed

- Onion: struggle with extra DTOs between proxy and domain

MY TAKEAWAYS – THE NOT SO OBVIOUS

My second attempt at making a salad... I don't know what I'm doing wrong

- simplest rules → complex ones

- Onion: often looks like a pass-through → deceptive!

- TDD vs. BDD

- Beyond BDD/acceptance Tests: do the tests give me high confidence?

THE MANDATORY SLIDE – WITH BAD COHESION

Credits

- 9 (10) weeks with Marco & Alex
- git…../partner-partner.git
- bdd-vs-tdd

Thanx for your attention!

in tamas-guenther