WALKING
Software design
(My thoughts)

Eirik Dyrli

The brain has some challenges (atleast mine has)

- Unorganized
- Forgetting stuff
- Loosing focus
- Making mistakes
- Can only keep focus on a limited amount of items

How to overcome?

- Mob programming (Or pair programming if no mob is available)
- Test Driven Design
- Transformation Priority Premise
- Object Calisthenics

Mob (pair) programming

- More brains remember and knows more stuff
- Discussing a problem results in a better design
- Focus is distributed and each person can forcus on a smaller part (Driver, Navigator, (Mob (IF (available))))
- Helping eachother back on track

Test Driven Design

- By creating test before code we ensure the code does what it is supposed to.
- We ensure customer requirements are met
- Chopping the code into smaller testable parts helps keeping focus and not trying to grasp everything at the same time
- Making debugging A LOT easier. We dont need to run a large application to test parts of the code.

Transformation Priority Premise

- This may seem unessesary (until you tried).
- Using the transformation prority makes the code less complex.
- By praticing the Fake impementation, obvious implementation and triangulation together with the transformation priority table you are forced into making babysteps and are being able to think through every step.

Object Calisthenics

- By using the object calisthenics rules the code becomes
  - A lot
    - More reable
    - More understandable
    - More SOLID
  - Prevents the developer from doing mistakes.
  - Keeps the methods (And developer) more focused

Final words

- It is liberating to keep focus on want to do instead of how to do it.
- By using TDD, Transformation Priority Premise and Object calisthenics the «how to do it» part becomes at lot more obvious.
- By practicing TDD, Pair/mob programming and applying the rules we learned we become more confident that we are making good code.
- Remeber to be lazy and make the IDE do your coding.
- I need to exercise TDD.