

# GRUPPE II INI010 – UKE 13

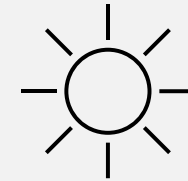
Sivert Fjeldstad Madsen

# I DAG

- Design og testbarhet
- Hva er godt design?
- Hvordan skrive en god test?
- Oppgaver

Coworker: How do you even understand your own code?

Me:



20 CREATIVE WAYS  
TO ENJOY SPAGHETTI



# DESIGN OG TESTBARHET

- Hovedpoenget denne uken er å bli litt mer oppmerksom på hva vi skal ta hensyn til når vi koder
- **Design** handler om hvordan vi skal strukturere et program
  - Hva trenger programmet for å fungere som det skal?
  - Hvordan skal interaksjon fungere?
- **Testbarhet** handler i stor grad om vedlikehold av koden
  - Passe på at ting ikke går i stykker når man legger til noe nytt
  - Gjøre det enklere å oppdage feil

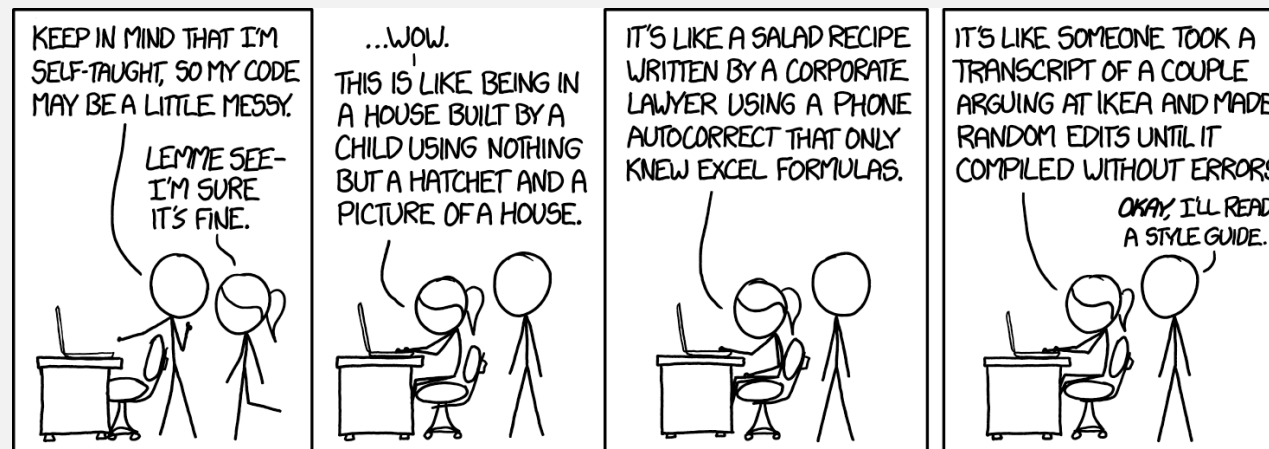


# HVA ER GODT DESIGN?

- Det finnes (dessverre) ikke noe fasitsvar, men her er noen tips:

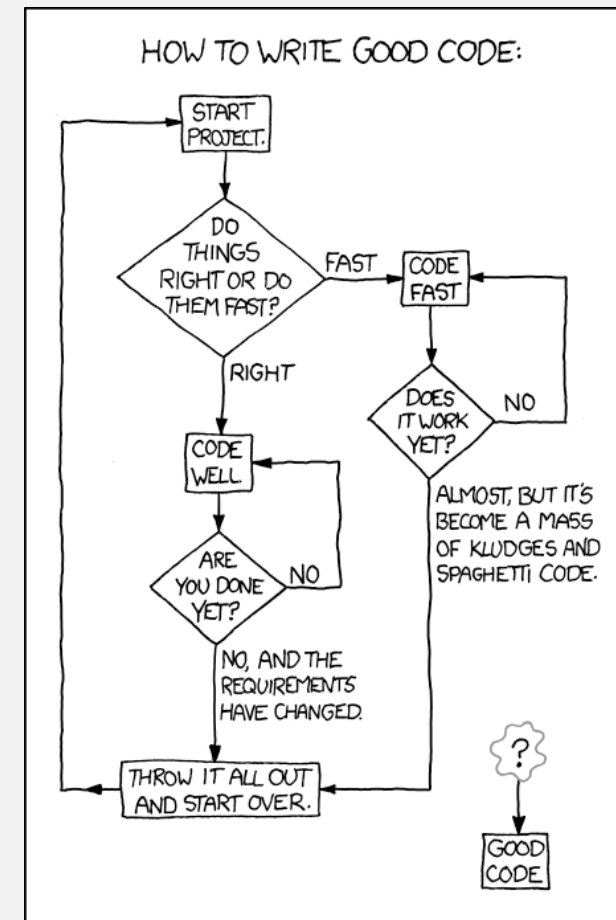


- Del opp koden i logiske deler
- **Innkapsling** – kode som bare skal brukes ett sted bør bare kunne brukes ett sted
- **Høy kohesjon** – hver del av koden har få oppgaver
- **Lav kobling** – deler av koden er avhengige av få andre deler



# FOR MYE DESIGN?

- Pass på at du ikke setter deg fast i planleggingsfasen
- Man må alltid finne en balansegang mellom å designe for mye og for lite
- Lag en plan som er «god nok», og endre den heller underveis
- Å planlegge for nøye kan gjøre koden lite fleksibel
- Å planlegge for lite kan gi mye **teknisk gjeld**
- Tenk på hva dere lærer i INI030 om arbeidsmetoder



# HVA ER EN GOD TEST?

- Tester én funksjon (enhetstest)
- Skriv tester som feiler
- Tester typiske verdier
- Tester ekstremverdier
- Gir gode og tydelige tilbakemeldinger
- Kan kjøres raskt
- Kan kjøres ofte

## ⚠ ERROR

IF YOU'RE SEEING THIS, THE CODE IS IN WHAT I THOUGHT WAS AN UNREACHABLE STATE.

I COULD GIVE YOU ADVICE FOR WHAT TO DO. BUT HONESTLY, WHY SHOULD YOU TRUST ME? I CLEARLY SCREWED THIS UP. I'M WRITING A MESSAGE THAT SHOULD NEVER APPEAR, YET I KNOW IT WILL PROBABLY APPEAR SOMEDAY.

ON A DEEP LEVEL, I KNOW I'M NOT UP TO THIS TASK. I'M SO SORRY.

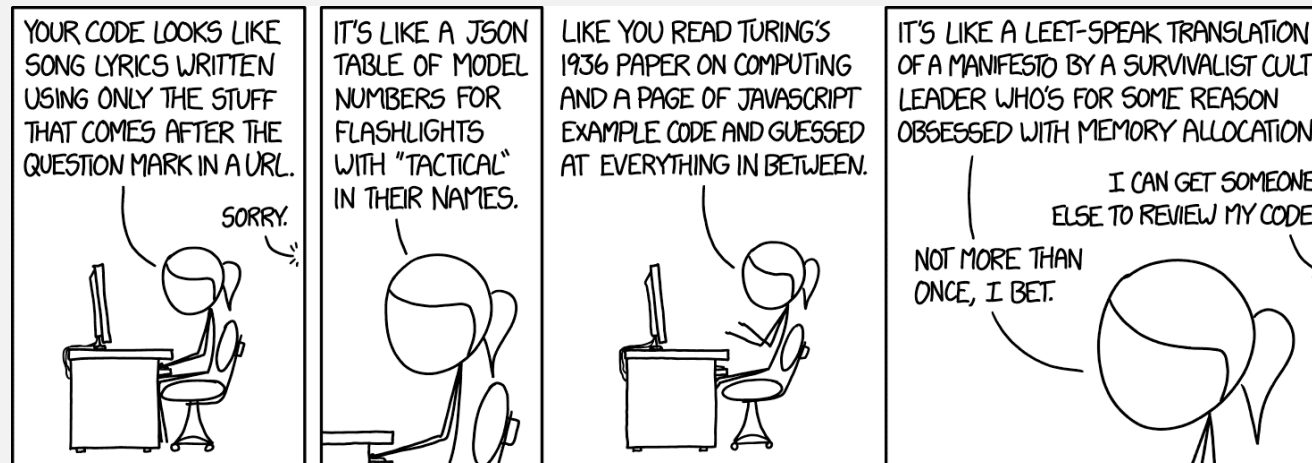
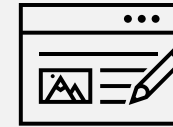


NEVER WRITE ERROR MESSAGES TIRED.



# PROGRAMMERE FOR TESTBARHET

- Jo mer oversiktlig programmet vårt er, desto enklere er det å skrive tester for det
- Prøv å identifisere invarianter
- Skriv interfacer for klasser, slik at de enkelt kan byttes ut med andre klasser
- Tenk på design og arkitektur



# OPPSUMMERING

- Tenk på design og testbarhet for å unngå spagettikode
- Prøv å ikke over-engineere for mye
- «God kode» er et relativt begrep
- Ukas tips:
  - [xkcd.com](http://xkcd.com)

