

# Mobbprogrammering

Lars Jørgen Tvedt  
Heidi Mork

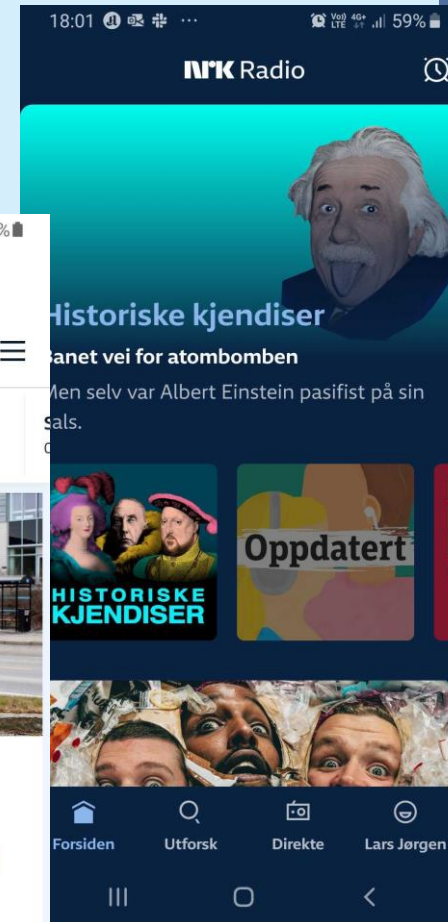
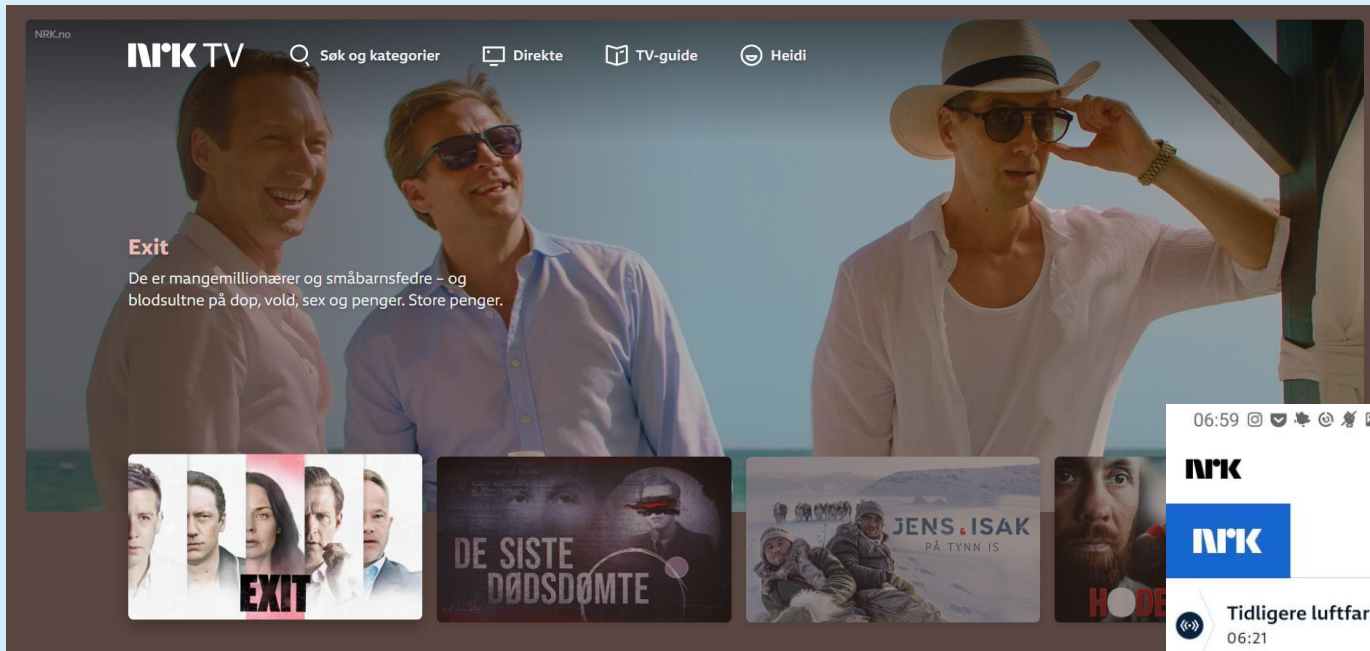


## Lars Jørgen Tvedt

- lars.jorgen.tvedt@nrk.no
- Sivilingeniør i matematikk, Universitetet i Trondheim, NTH
- Har jobbet som utvikler siden 1990
  - SuperOffice
  - Norsk Rikstoto
  - Universitetet i Oslo
  - Computas
  - NRK

## Heidi Mork

- heidi.mork@nrk.no
- PhD i matematikk, Universitetet i Oslo
- Har jobbet som utvikler siden 2009
  - Computas
  - Norsk regnesentral
  - Kantega
  - NRK



# Utgangspunktet

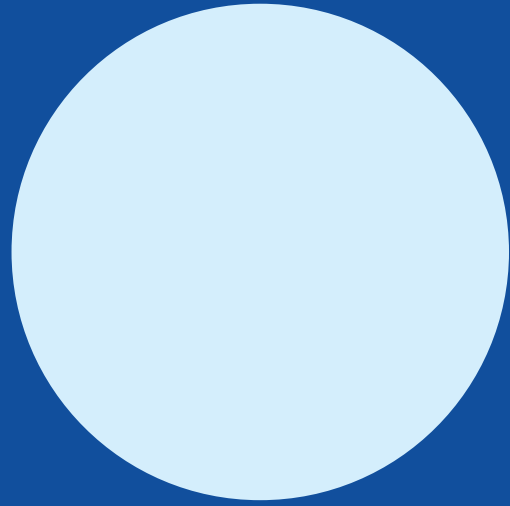
- I 2018 skulle vi erstatte en 10 år gammel databaseintegrasjon
- Vi forsøkte å dele problemet i tre
- Hver for oss klarte vi ikke å komme videre
  
- Vi måtte finne en bedre måte å jobbe på!

# Mobbprogrammering

- Hele teamet jobber sammen om å løse
  - Ett problem
  - I samme rom
  - På samme tid
  - På samme datamaskin



Men vi  
begynte mer  
tradisjonelt...

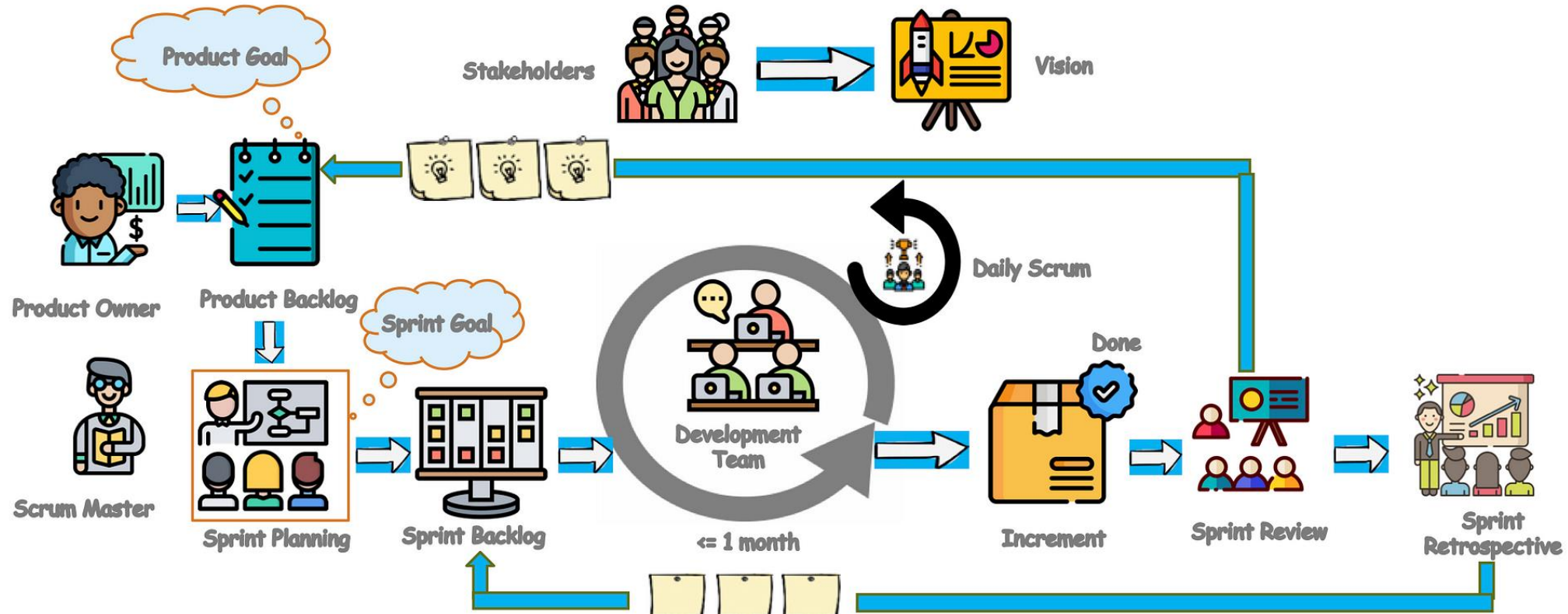


# Scrum

- Parallellisering av arbeidet for å jobbe mest mulig effektivt
- Liste over små og veldefinerte oppgaver som team-medlemmer plukker fra
- Iterasjoner (sprint) for å begrense mengden endringer som skal produksjonsettes
- Daglige standupmøter
- Planleggingsmøte, demo og retro for at alle skal vite hva som skjer

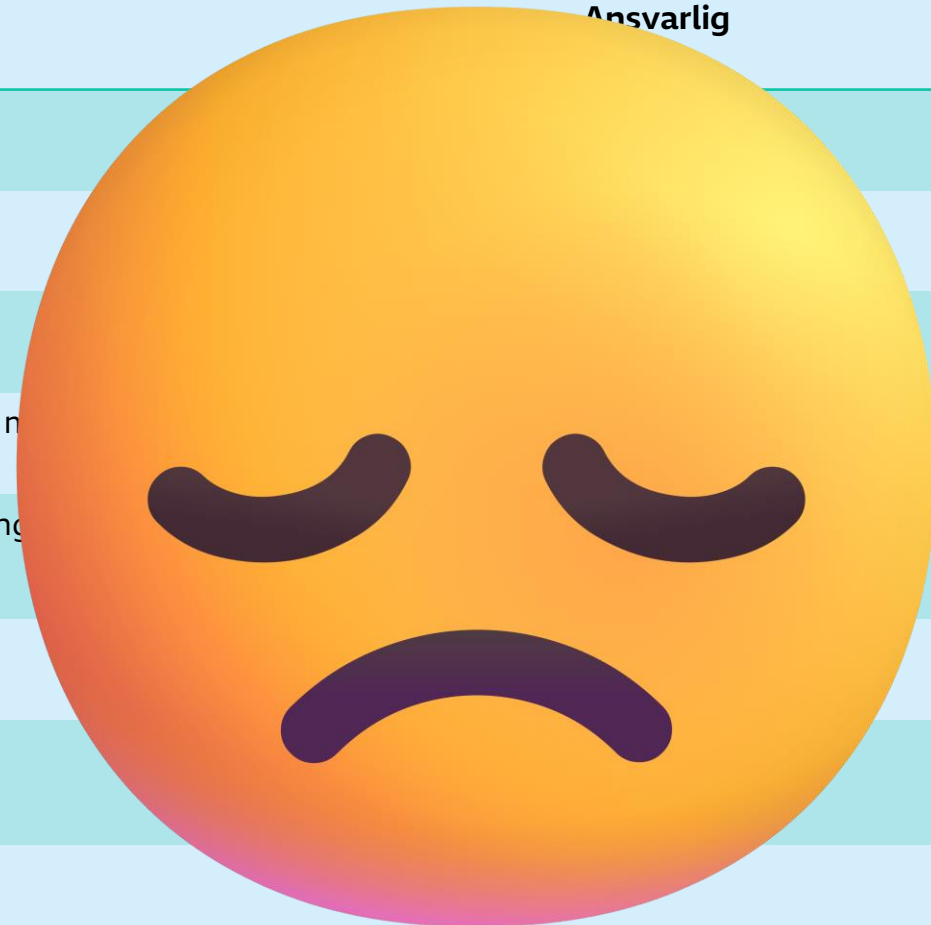


# Scrum Framework




Oppgave	Ansvarlig	Status
Forstå gammel logikk	Lars Jørgen	Under arbeid
Opprette ny database	Lars Jørgen	Under arbeid
Definere tabellstruktur	Lars Jørgen	Ikke påbegynt
Sette opp Service for mottak av meldinger	Steinar	Under arbeid
Sette opp Engine for prosessering av meldinger	Heidi	Under arbeid
Håndtere «sendinger»	Heidi	Ikke påbegynt
Håndtere «program»	Steinar	Ikke påbegynt
Lage elektronisk programguide	???	Ikke påbegynt

Oppgave	Ansvarlig	Status
Forstå gammel logikk		Under arbeid
Opprette ny database		Under arbeid
Definere tabellstruktur		Ikke påbegynt
Sette opp Service for mottak av n		Under arbeid
Sette opp Engine for prosessering		Under arbeid
Håndtere «sendinger»		Ikke påbegynt
Håndtere «program»		Ikke påbegynt
Lage elektronisk programguide		Ikke påbegynt



# Problemer som oppstod

- Vanskelig å lage isolerte oppgaver
- Kunstig oppdeling av oppgaver
- Oppbygging av «spesialister» i teamet
- Prosesser og koordinering førte til
  - Flaskehalser
  - Venting
  - Forstyrrelser
  - Opphopning av uferdige oppgaver



Kan vi jobbe på en bedre  
måte?



# Vi vil jobbe smidigere

- Rett produkt, til rett tid, med høy kvalitet
- Fokus
- Raske avklaringer
- Kort feedback-loop



# Parprogrammierung



# Parprogrammering

- To teammedlemmer jobber sammen
- En maskin, ett tastatur, en mus
- Den som skriver kalles «sjåfør»
- Den som ikke skriver er «navigatør»
- Rollene byttes ofte
- Navigatøren involverer seg aktivt i koden som blir skrevet av sjåføren



"For an idea to go from your head into the computer it **MUST** go through someone else's hands"

Llewellyn Falco



# Fordeler

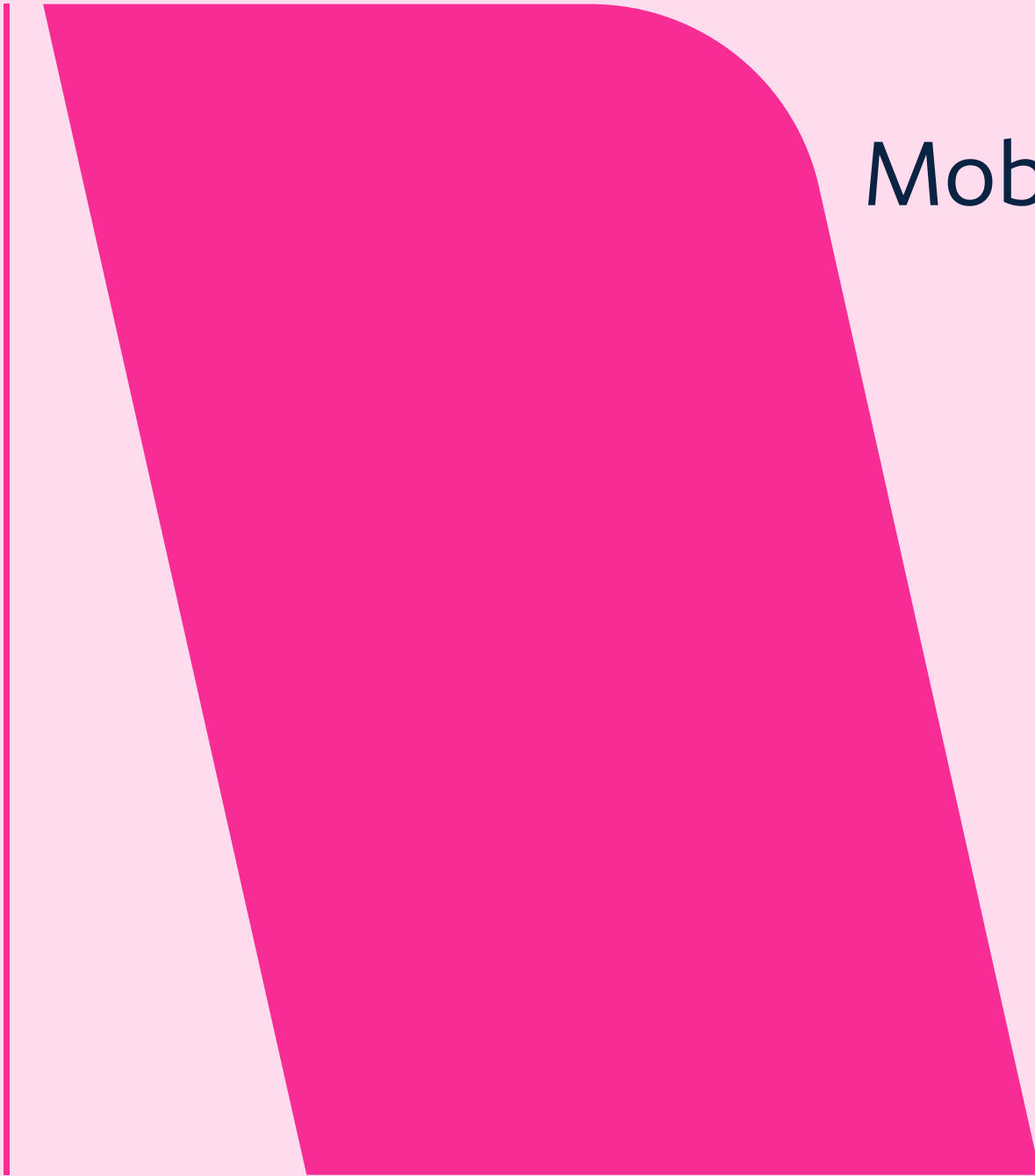
- Kontinuerlig kvalitetssikringsprosess
  - Kan droppe egne steg med PR og test
  - Kortere tid til koden er produksjonsklar
- Kontinuerlig kompetansebygging i paret
- Bygger sosiale relasjoner
- Tåler bedre at enkeltpersoner kan bli borte fra jobben i korte eller lengre perioder

# Parprogrammering i team

- Flere par i et team
- Blandet kompetanse i parene
- Roterer hvem som er med i parene, for å sikre kunnskapsdeling

# Ikke alle Scrum-problemer løst

- Må definerer oppgaver, eller områder for hvert par
- Må sørge for at koden som parene skriver, fungerer sammen
- Spesialister, men nå som par



# Mobbprogrammering

# Mobbprogrammering

- Hele teamet jobber sammen rundt én maskin, ett tastatur, én mus
- Teamet jobber med én ting om gangen
- Den som skriver er sjåfør
- De som ikke skriver er navigatører
- Rollene roteres ofte, f.eks. hvert femte minutt

# Andre begreper

- Software teaming
- Ensemble programming

# Teamet i mobbprogrammering

- Teamet består av utviklere, designere, produkteiere, testere, etc.
- Hele teamet jobber sammen i alle faser av utviklingsarbeidet



# Law of two feet (reversert)

- Du bør være en del av mobben dersom du har noe å bidra med, eller du kan lære noe av å være til stede

# Dynamikk

- Utviklingsfasene (design, utvikling, testing, kvalitetssikring) flyter sømløst over i hverandre i korte iterasjoner
- Teamet tar raske beslutninger når det trengs
- Mobben kan fortsette arbeidet selv om ikke alle er til stede til enhver tid
- Oppgaven er ferdig når teamet er fornøyd

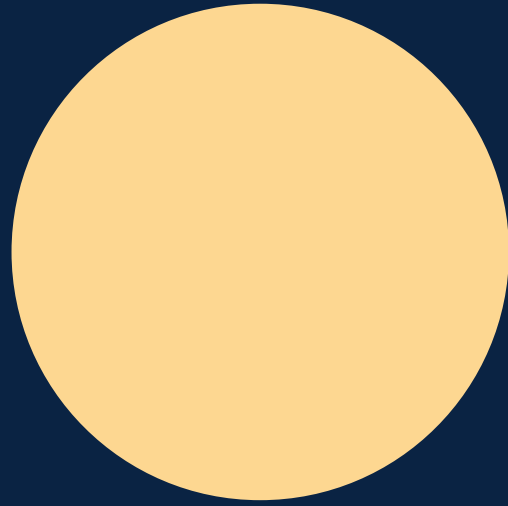
# Fordeler

- Mindre behov for detaljplanlegging
- Mindre venting (positiv type venting der du kan lære)
- Mindre forstyrrelser
- Kontinuerlig kunnskapsdeling
- Felles forståelse av problem og løsning
- Fellesskap gir trygghet
- Teamet tåler at enkeltmedlemmer slutter

# Utfordringer

- Stor overgang fra å sitte alene med et problem, ikke alle synes overgangen er enkel
- Ulike personligheter og behov
- Mister personlig eierskap til koden (kan være viktig for noen)
- Kan være vanskelig å finne plassen sin i en «erfaren» mobb
- Noen vil kanskje aldri trives i en mobb

Våre  
erfaringer



# Historien



**Høst 2018:** Oppstart av et nytt prosjekt



**Vinter 2019:** Vi oppdager at fremdriften er svært dårlig



**Vinter 2019:** Eksperimenterer med mobb



**Våren 2019-mars 2020:** Mobben er etablert med egen permanent mobbstasjon



**Mars 2020:** Mobben blir sendt på hjemmekontor



**Januar 2021:** Hovedprosjektet levert, går over i vedlikeholdsfase, og får ansvaret for flere titalls system



**Februar 2022:** Mobben reetablert på Marienlyst



**November 2022:** Teamet tar en pause fra mobb som primærarbeidsform



**Februar 2024:** Teamet jobber i mobb med de aller fleste oppgavene

# Hva erfarte vi?

- Kontinuerlig «brainstorming»
- Lett å teste ut nye idéer
- Kompetansen på plass når vi trengte den
- Sammen løste vi oppgaver vi ikke klarte å løse alene
- Sammen om oppturer og nedturer
- Fokus på en ting av gangen

```
Bookmarks
- README.md
- MainActivity.kt
- CatFacts
- ...

MainActivity.kt
setContentView(R.layout.activity_main)
// A surface container using the 'background' color from the theme
Surface(modifier = Modifier.fillMaxSize(), color = MaterialTheme.colors.background)
GreetingPreview()
Column(modifier = Modifier.fillMaxSize()) {
    Text(text = "")
}

@Composable
fun ListOfCats() {
}

@Preview
@Composable
private fun ListOfCatsPreview() = ListOfCats()

ListOfCatsPreview
GreetingPreview
@Composable
@Preview
private fun ListOfCatsPreview() = ListOfCats()

Run
2023-10-03 13:06:53: Launching GreetingPreview on "Pixel 6 API 33".
adb shell am start -n com.example.catfacts.android.compose.ui.tooling.PreviewsActivity -a android.intent.action.MAIN -c android.intent.category.LAUNCHER --es composabl
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.catfacts.android.compose.ui.tooling.PreviewsActivity (has extras) }
Open Logcat panel for emulator Pixel 6 API 33
Connected to process 11703 on device 'Pixel_6_API_33 [emulator-10854]'.
```





# God kvalitet

- Vilje til å bruke tid når kvaliteten ikke var god
- Hyppig refaktorering
- Høy testdekning
- Snarveier ble fort avslørt
- Lesbar kode

# Effektivitet

- Løser bare det vi skal løse
- Mindre fare for «scope creep»
- God nok kvalitet
- Fokus på å gjøre en ting om gangen
- Ingen «venteoppgaver»
- Når all kunnskap er tilgjengelig, går man fort i rett retning

# Robust team

- Hele teamet har eierskap
- Alle kan løsningen
- Kunnskapsdeling
- Ferie/fravær et lite problem
- Lite avhengig av enkeltpersoner

# Hjemmekontor-mobb (covid-mobb)

- Mobb via Teams
  - Hver utvikler kodet på sin PC
  - Brukte mob.sh for å flytte koden
  - Sjøføren delte skjerm
- 
- Mobba hele dagen, hele uka, hele pandemien

# Hva vi sluttet med

- Møter/standUp/...
- Lese QA på Pull Requests
- Vente på at andre har lest QA på egen PR
- Forstyrre andre i teamet
- Vente på svar fra andre i teamet
- Bli forstyrret av andre team
- Bli forstyrret av slack og e-post



# Ulike typer mobber

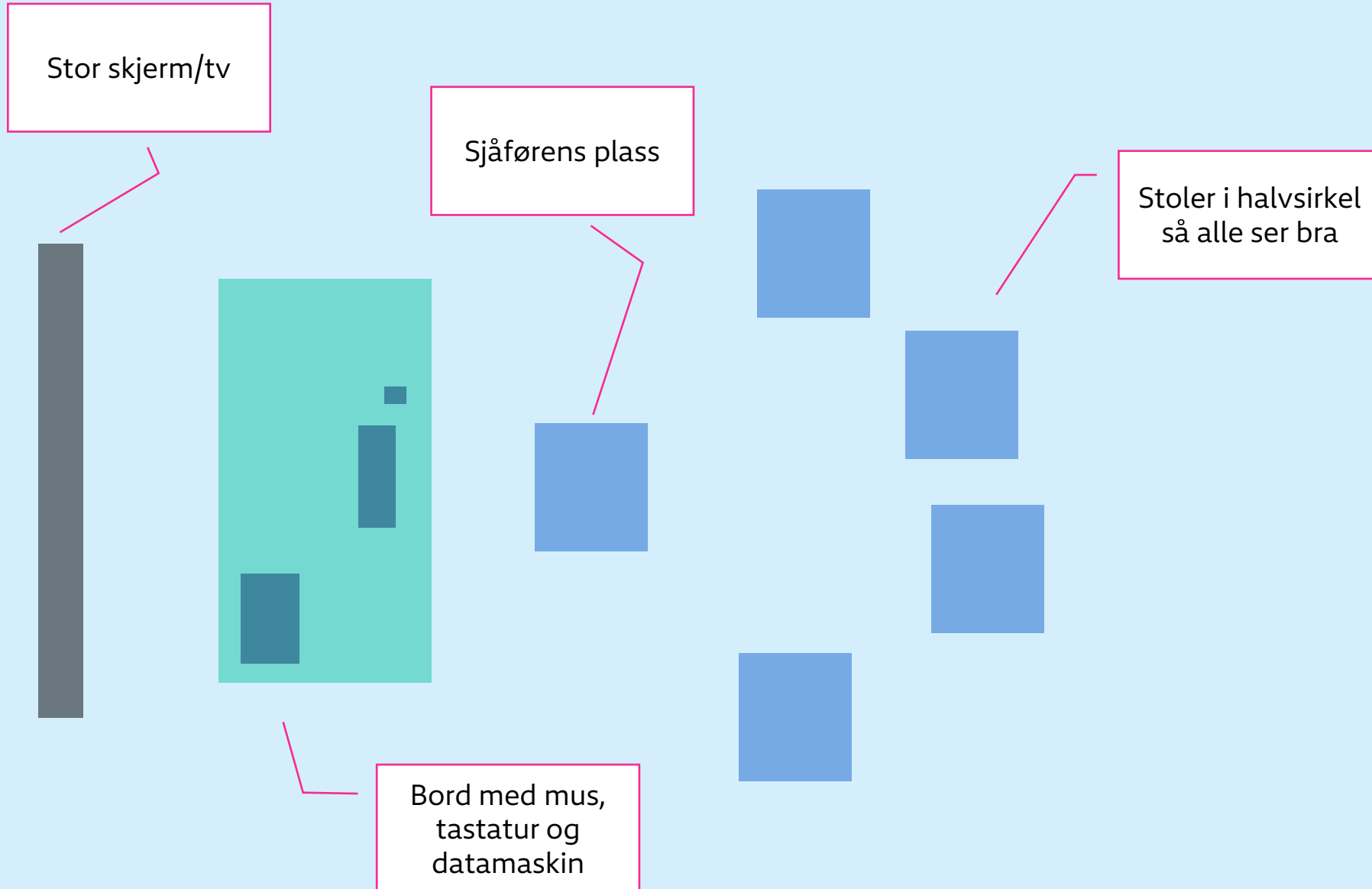
- Team-mobb
- Plattform-mobb
- På-tvers-av-team-mobb
- Lesesirkelmobb
- Innovasjonsdagsmobb
- Kompetansehevingsmobb

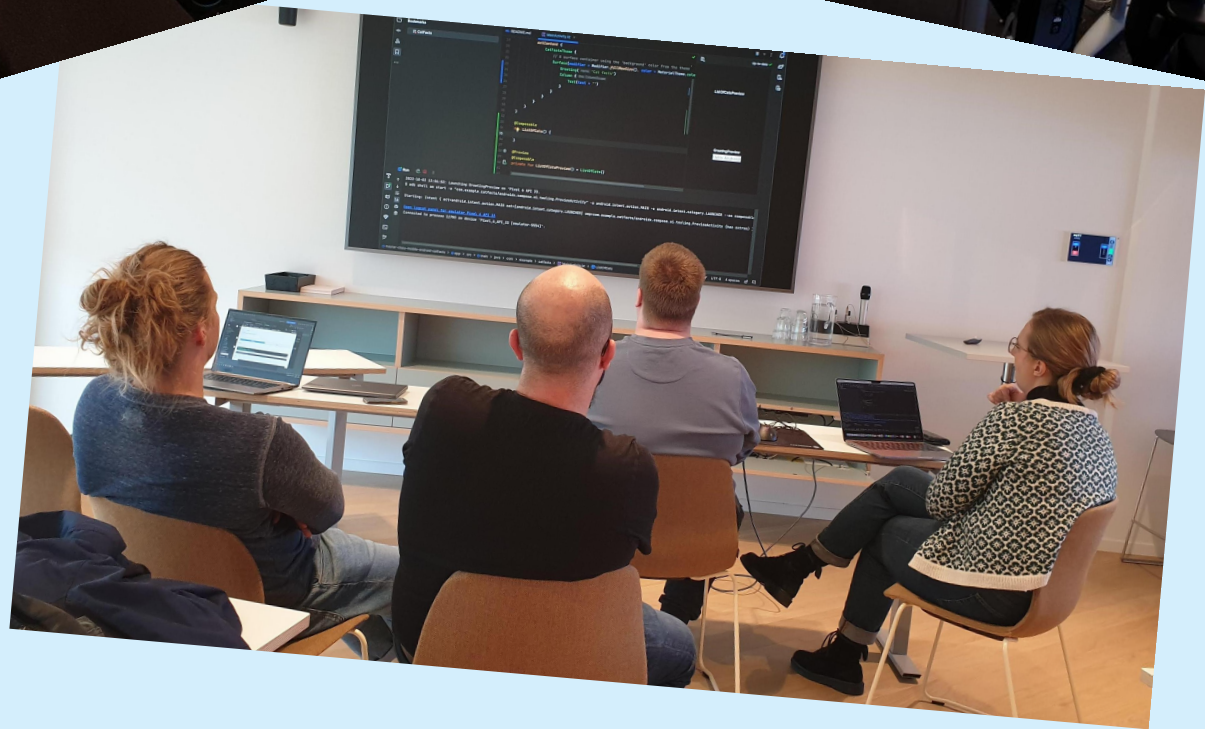
Hvordan  
komme  
i gang





# Grunnoppsett





# Hvordan komme i gang

- Finn et sted med minst en pult, en litt stor skjerm og gjerne stoler til alle
- Ta med en datamaskin med løst tastatur og mus
- Legg bort alle mobiltelefoner og personlige datamaskiner
- Bestem hvor lang tid sjåføren har tastaturet (f.eks 5 minutt)
  - Jo flere i mobben, jo kortere tid på tastaturet
- Bruk timer!

# Hvordan jobbe sammen

- Sjåføren skal bare gjøre det navigatørene sier
- Navigatørene må snakke sammen, og gi sjåføren gode instruksjoner på det nivået som er nødvendig
- Sjåføren må gi fra seg tastaturet umiddelbart når tiden går ut, gjerne midt i skrivingen av en kodelinje
- Ta pauser ofte, og minst en gang i timen

# Kjøreregler

- Vær hyggelig og omtenkksom
- Prøv å forstå hva andre foreslår
- La alle slippe til
- Bruk tid på å forklare slik at alle forstår

# Rollene er snudd

- Å være sjåfør er lett – du skal bare skrive
- Å navigere er vanskelig



# Når du får en ide!

- Har du en god ide, skal du **ikke** sitte ved tastaturet
- Ideen må forklares slik at sjåføren kan implementere den



"For an idea to go from your head into the computer it **MUST** go through someone else's hands"

Llewellyn Falco





# Når mange får en ide!

- Er det mange ulike ideer, prøv alle, ikke diskuter masse for og imot
- Prøv gjerne den minst erfarne sin ide først

# Øvelse gjør mester

- Alle bør være motivert for å prøve, men lov å være skeptisk
- Ikke mobb for lenge i starten (kan være slitsomt)
- Finn ut hva som går bra, og gjør mer av det
- Det er vanskelig i starten, øvelse gjør mester

## STRATEGI

## Egenrådig sjåfør

Teamet bryr seg ikke om å bruke timer eller roller i mobben. Alle kan bidra med ideer, også sjåføren. Sjåføren må gjerne implementere egne ideer, men må uttrykke dem klart først.

## STRATEGI

## Hold munn

En navigatør lar midlertidig være å snakke, så andre navigatører kan slippe til. Kan brukes for å få i gang en treg mobb eller for å forhindre at en person dominerer i mobben.

## STRATEGI

## Nå skal jeg lære

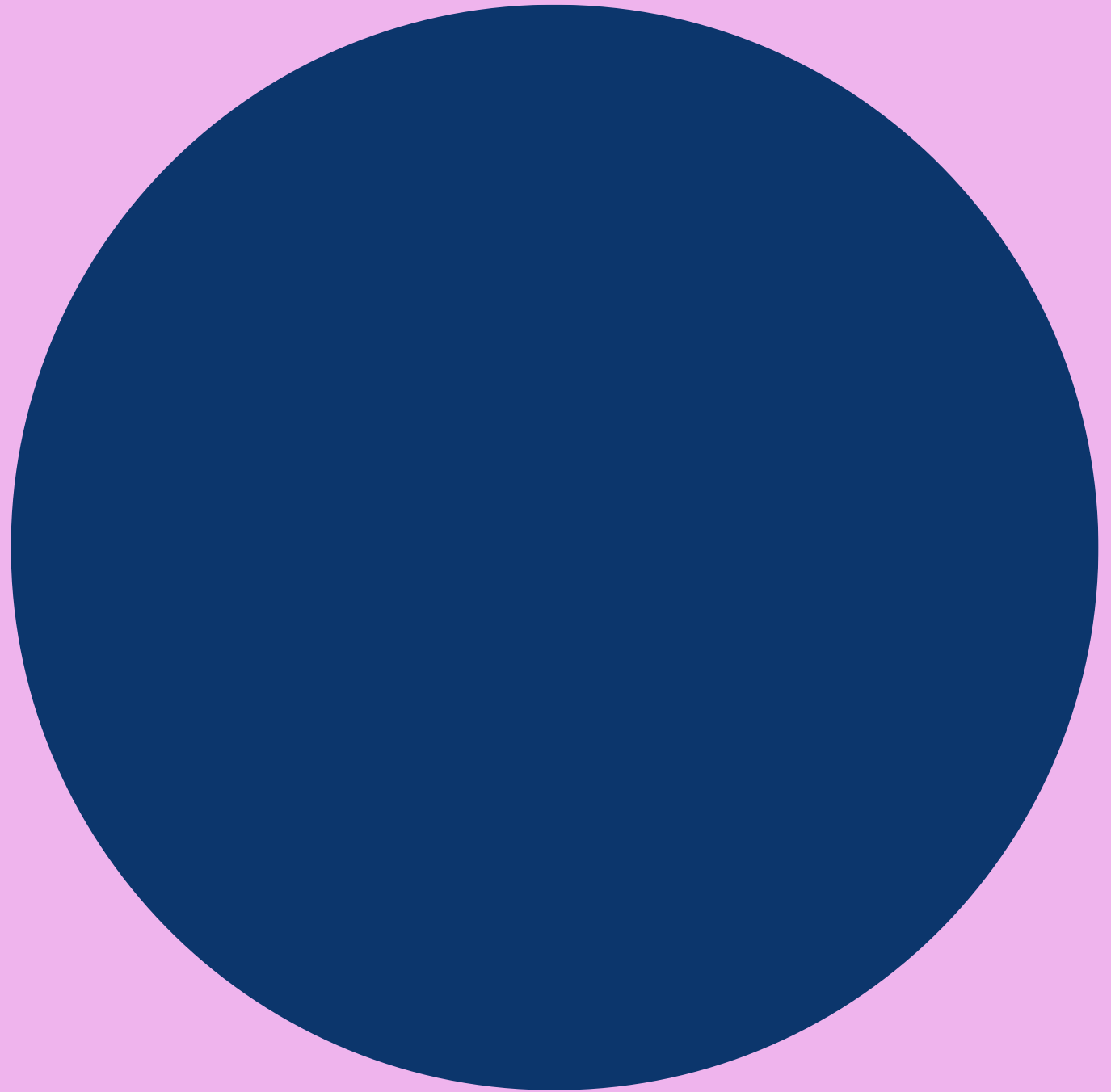
Når en i mobben er usikker eller ny, bruk av vedkommende sin navigatørtid til opplæring i stedet for koding.

# Retro

- Ofte, hver uke eller hver dag!
- Gjør mer av det som fungerer
- Let etter løsninger på det som ikke fungerer



Verktøy  
og  
inspirasjon

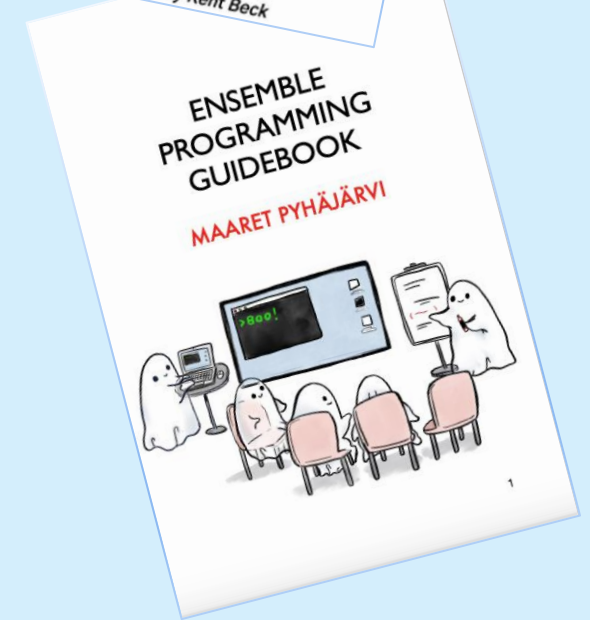
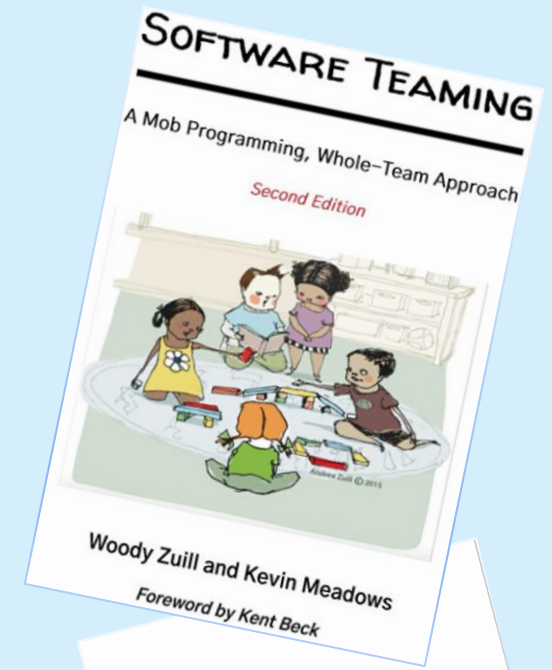


# Verktøy

- <https://mob.sh/>
- <https://timer.mob.sh/>
- <http://mobster.cc/>
- <https://github.com/nrkno/den-mangfoldige-mobben>

# Les mer

- <https://softwareteaming.com/>
- <https://ensembleprogramming.xyz/>
- <https://www.remotemobprogramming.org/>
- <https://www.infoq.com/articles/co-creation-patterns-software-development/>
- <https://www.agilealliance.org/resources/experience-reports/mob-programming-agile2014/>



Se mer



<https://www.mobmentalityshow.com/Mob Programming - Woody Zuill A day of Mob Programming 2016 A day of Mob Programming Fire år i mobb>



Ta  
kontakt,  
vi snakker  
 gjerne om  
 mobb!

