

Mobbprogrammering

Heidi Mork



Heidi Mork

- heidi.mork@nrk.no
- <https://www.linkedin.com/in/heidi-mork-7bb4b031/>
- PhD i matematikk, Universitetet i Oslo
- Har jobbet som utvikler siden 2009
 - Computas
 - Norsk regnesentral
 - Kantega
 - NRK siden 2018



Utgangspunktet

Det var en gang tre utviklere som skulle erstatte en 10 år gammel databaseintegrasjon.

De delte oppgaven i tre deler og startet på hver sin bit...

Mobbprogrammering

Hele teamet jobber sammen om å løse

- ett problem
- i samme rom
- på samme tid
- på samme datamaskin



Smidig
utvikling





Smidigmanifestet

- **Personer og samspill** fremfor prosesser og verktøy
- **Programvare som virker** fremfor omfattende dokumentasjon
- **Samarbeid med kunden** fremfor kontraktsforhandlinger
- **Å reagere på endringer** fremfor å følge en plan



<https://agilemanifesto.org/iso/no/manifesto.html>

12 prinsipp

Our highest priority is to satisfy the customer through **early and continuous delivery** of valuable software.

Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must **work together daily** throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.

Working software is the primary measure of progress.

Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

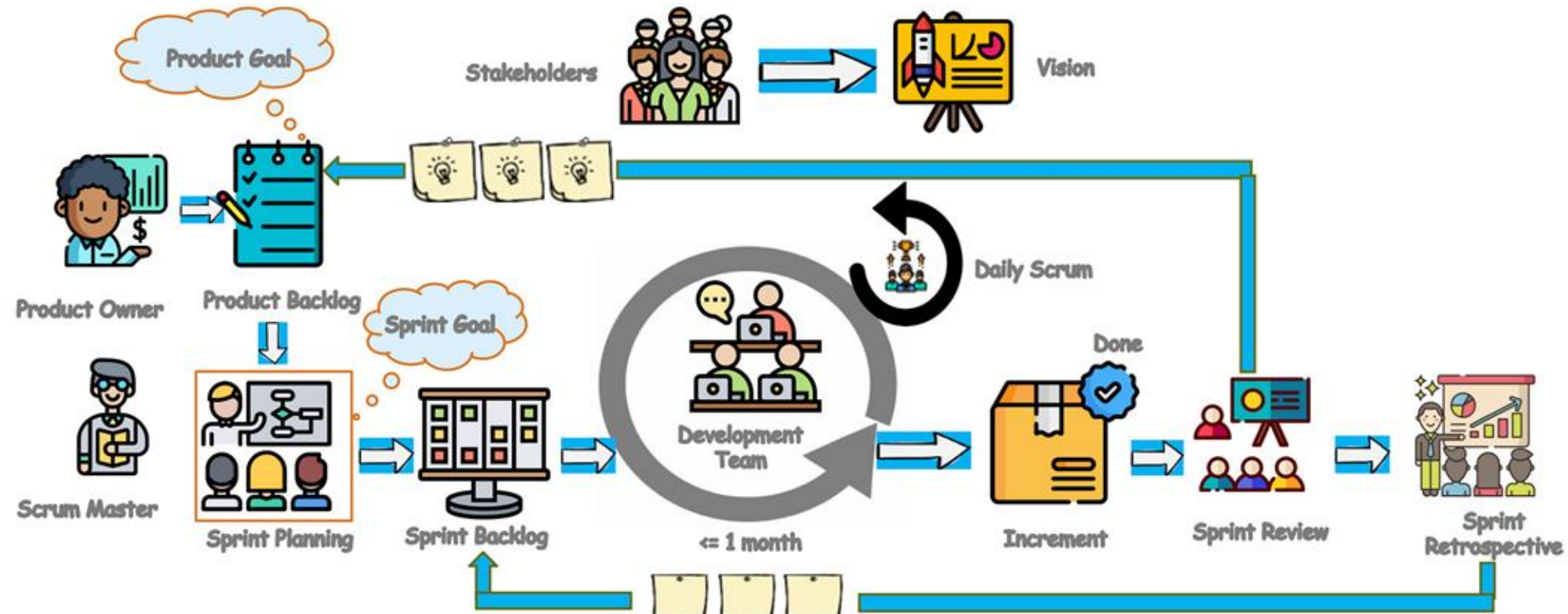
Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from **self-organizing teams**.

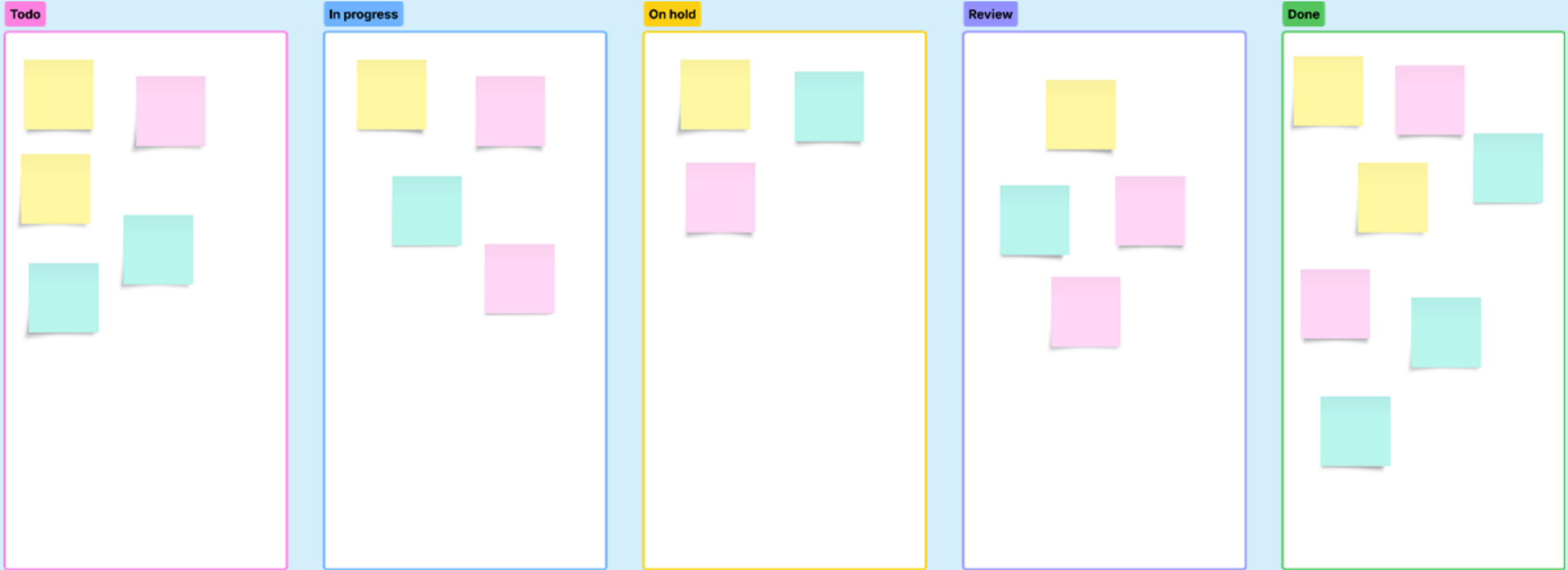
At regular intervals, **the team reflects on how to become more effective**, then tunes and adjusts its behavior accordingly.

Scrum Framework



Scrumish

- Iterasjoner (sprinter) der teamet jobber mot et konkret mål.
- Liste av små og veldefinerte oppgaver som team-medlemmer plukker fra.
- Daglig standup, planleggingsmøte, demo og retro for å holde folk i synk.



Todo



In progress



Done



Vanlige utfordringer

- Kunstig oppdeling i mindre oppgaver
- Bygger «spesialister» i teamet
- Prosesser og koordinering fører til
 - flaskehalser
 - venting
 - forstyrrelser
- Opphopning av uferdige oppgaver

The top of the slide features a dark blue background with three large, light blue geometric shapes: a trapezoid on the left, a circle in the center, and a hexagon on the right. Two thin vertical lines separate these shapes. Below this header is a large light blue rectangular area.

Kan vi jobbe på en bedre
måte?

A small light blue circle is positioned in the bottom right corner of the slide.

Vi vil jobbe smidigere

- Fokus
 - Raske avklaringer
 - Kort feedback-loop
- => Lage rett ting til rett tid, med god kvalitet

The header features a dark blue background with three large, light blue geometric shapes: a rounded rectangle on the left, a circle in the center, and a hexagon on the right. Two thin vertical white lines separate these shapes.

Parprogrammering



Parprogrammering

- To teammedlemmer jobber sammen
- Én maskin, ett tastatur, én 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

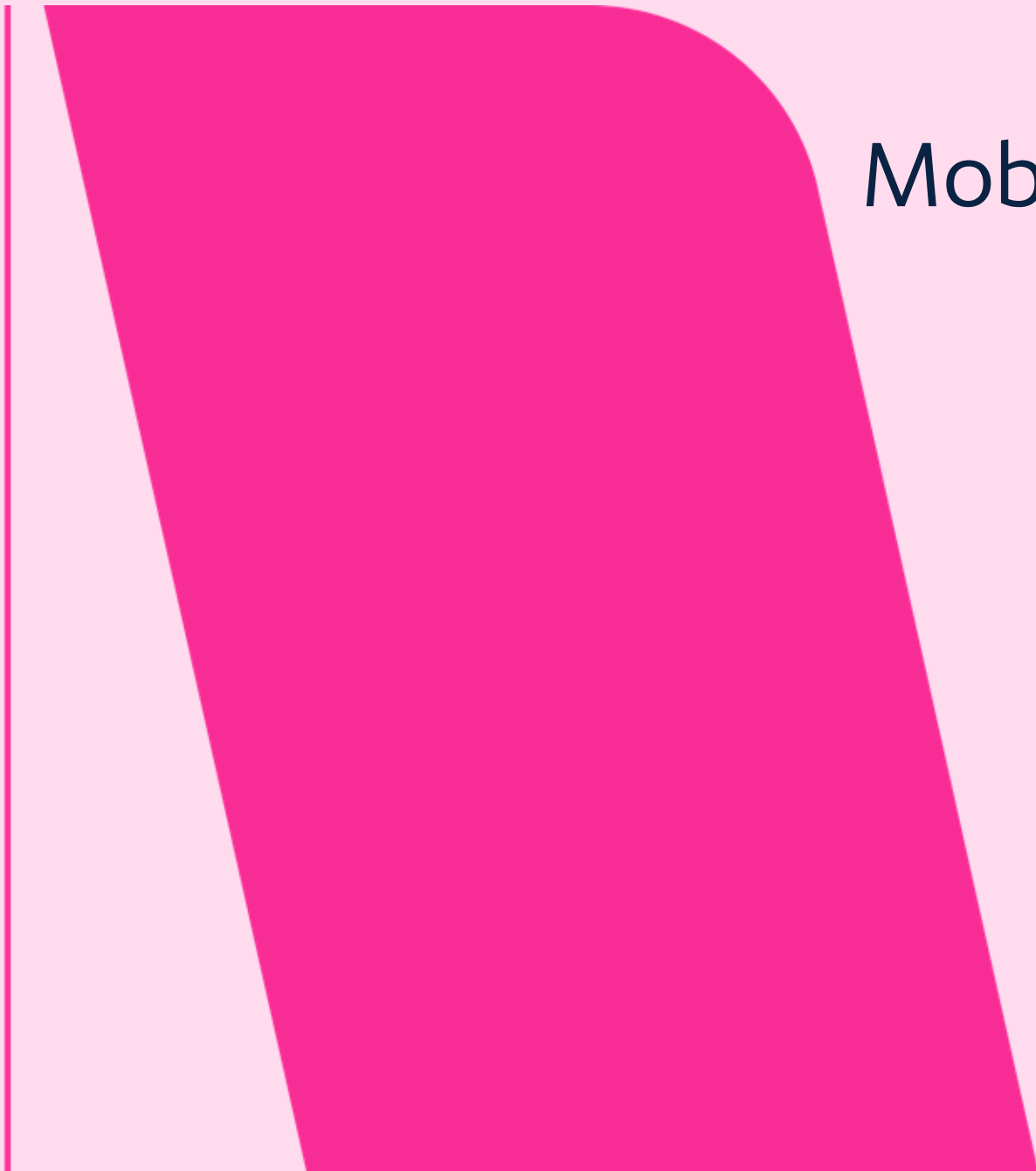
- Kontinuerlig kvalitetssikringsprosess
 - To par øyne på alt som skjer
 - Kan droppe egne steg med kodegjennomgang og test
- Kontinuerlig kompetansebygging i paret
- Bygger sosiale relasjoner
- Tåler bedre at enkeltpersoner kan bli borte fra jobben i korte eller lengre perioder

Parprogrammering i team

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

Utfordringer

- Må definerer oppgaver, eller områder, for hvert par
- Må sørge for at koden som parene skriver, fungerer sammen
- Risikerer spesialistpar, små team i teamet
- Personkjemi og kompetanse har en del å si
- Et par er avhengig av at begge er til stede



Mobbprogrammering

Andre begreper

- Software teaming
- Ensemble programmering
- Flokkprogrammering
- Samkoding
- Lagprogrammering
- Samarbeidsprogrammering
- Teamprogrammering

Mobbprogrammering

- Hele teamet jobber sammen rundt én maskin, ett tastatur, og 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

Sjåføren

- Har den letteste oppgaven
- Skal bare skrive det noen andre forklarer
- Om man ikke vet hva man skal gjøre er det navigatørene som ikke gjør jobben sin



Navigatørene

- Navigatørene må finne ut av hva som skal lages.
- Navigatørene forteller på høyest mulig nivå hva som skal lages – formidler intensjonen



Rotasjon

- Det er rotasjon som gir fokus og fart
- Jo flere som er med i mobben, jo oftere må man rotere
- Det fins ulike måter å rotere på, begynn med timer!
- Når man «glemmer» å rotere:
 - Sjåføren jobber på egenhånd uten kunnskapsdeling
 - Navigatørene mister fokus og begynner å gjøre andre ting



Teamet i mobbprogrammering

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

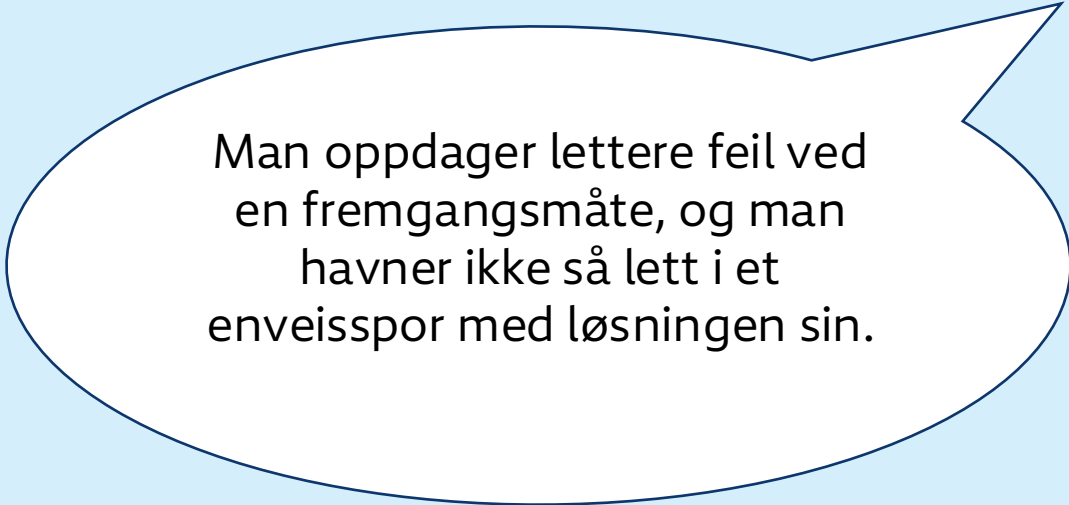
Dynamikk

- 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



God kvalitet

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

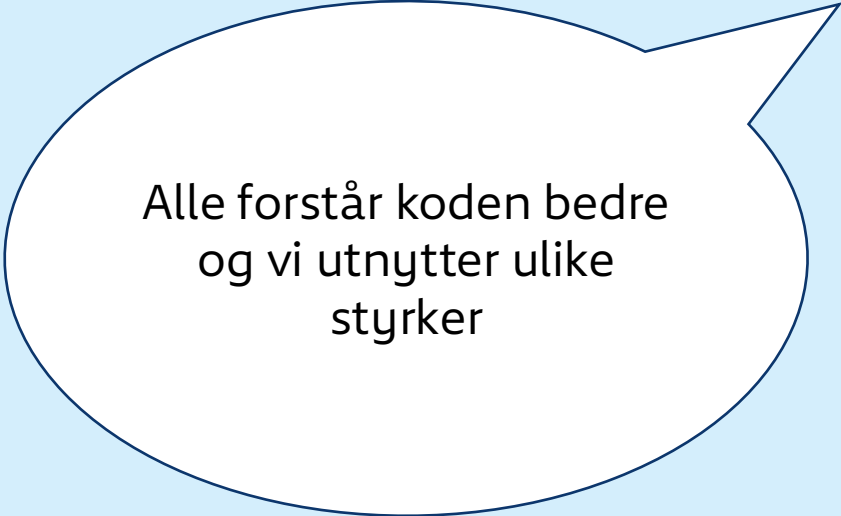


Man oppdager lettere feil ved en fremgangsmåte, og man havner ikke så lett i et enveisspor med løsningen sin.

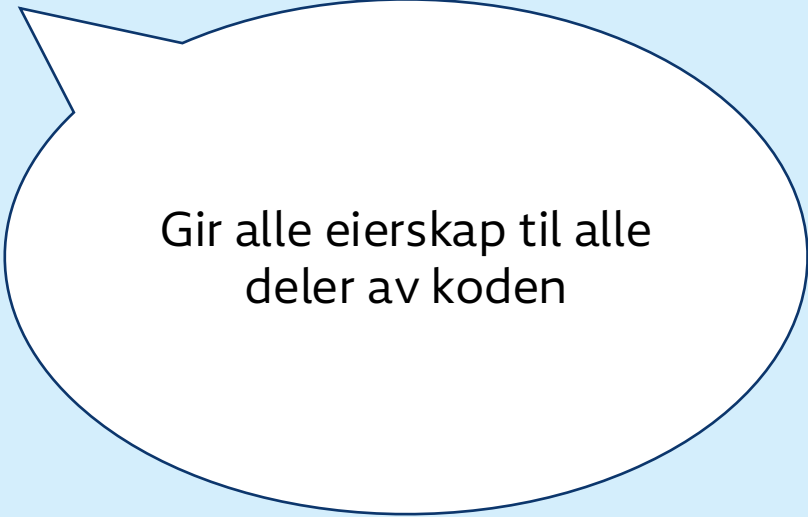
Snakkebobler er sitater fra intern spørreundersøkelse

Robust team

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



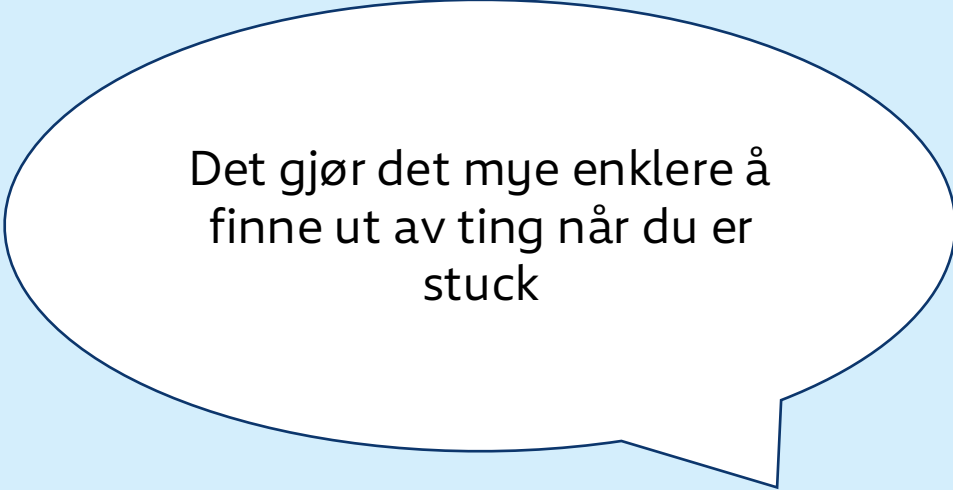
Alle forstår koden bedre
og vi utnytter ulike
styrker



Gir alle eierskap til alle
deler av koden

Effektivitet

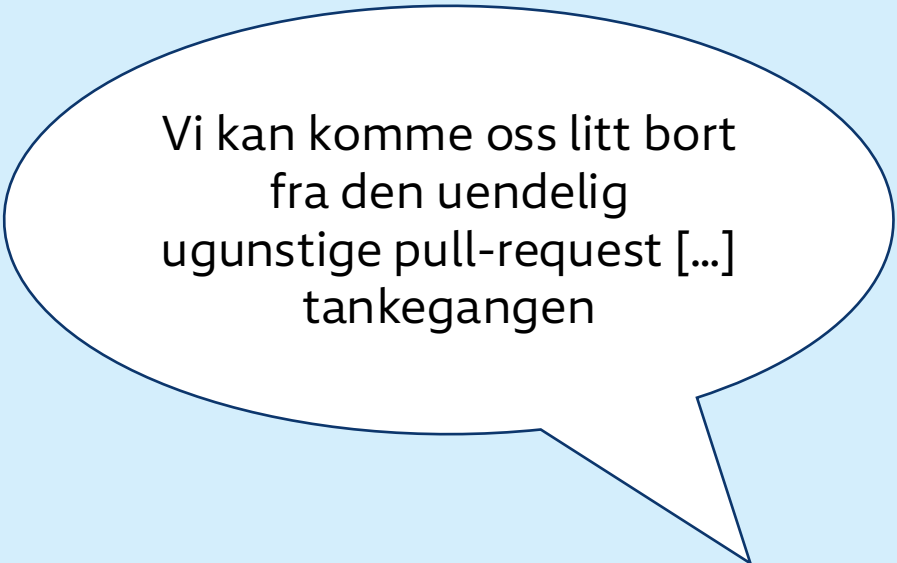
- Løser bare det vi skal løse
- Mindre fare for «scope creep»
- Fokus på å gjøre en ting om gangen
- Når all kunnskap er tilgjengelig går man fort i riktig retning
- Mindre venting (positiv type venting der du kan lære)
- Mindre forstyrrelser
- Mindre behov for detaljplanlegging (men vi planlegger)



Det gjør det mye enklere å
finne ut av ting når du er
stuck

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



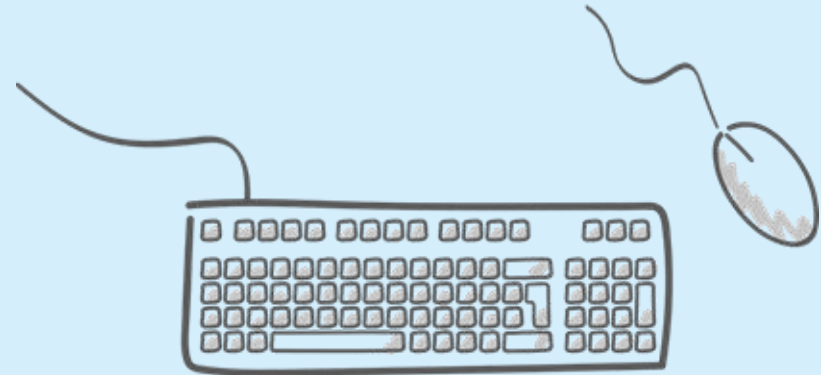
Vi kan komme oss litt bort
fra den uendelig
ugunstige pull-request [...] tankegangen



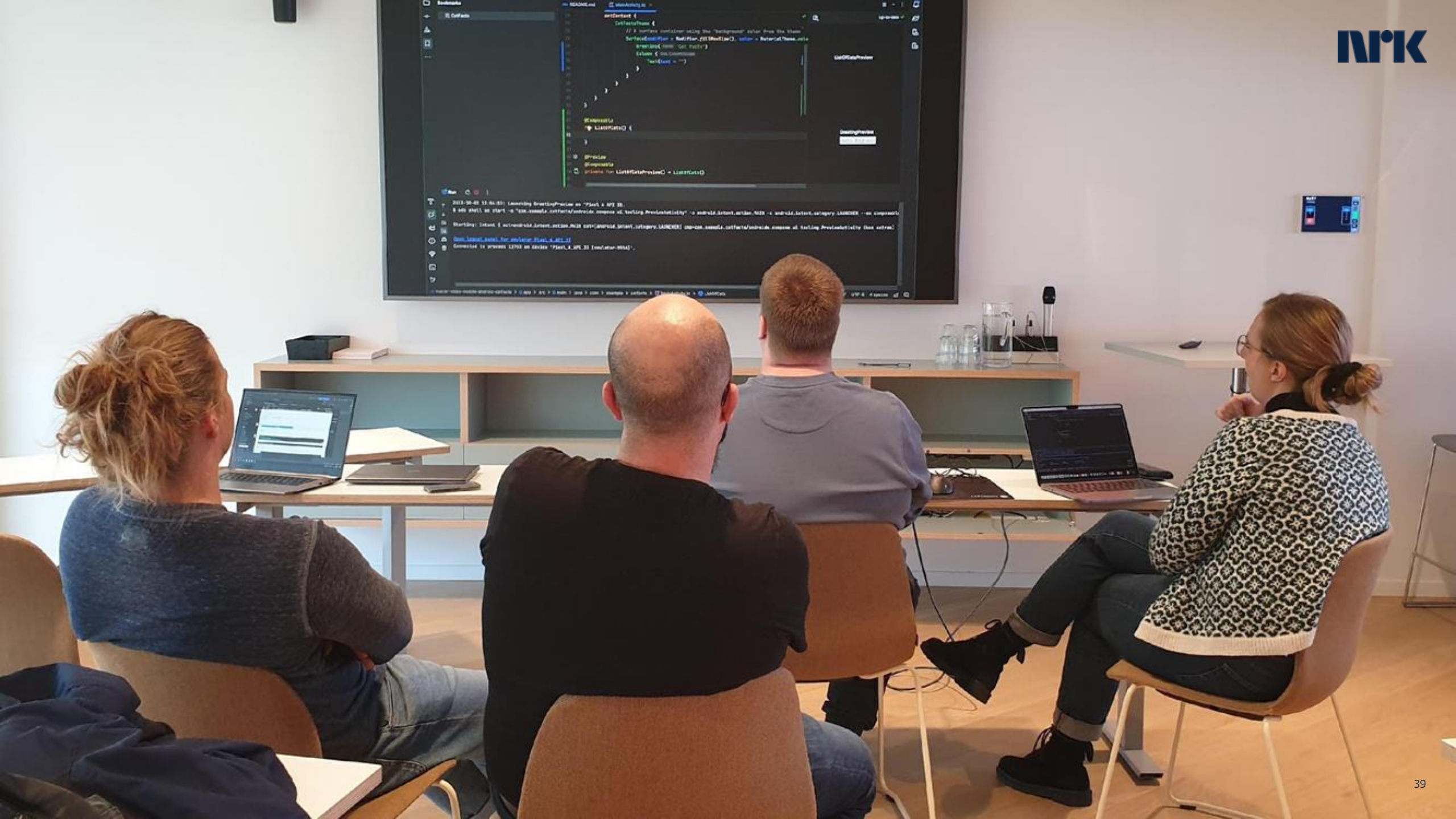
Hva er vanskeligst?



Finne ut hva som skal lages



Skrive kode



Utfordringer

- Stor overgang fra å jobbe alene med et problem
- Ulike personligheter og behov
- Mister personlig eierskap til koden
- Tenke sammen – finne felles retning

Kan være vanskelig når
det er mye som er
ukjent, og du trenger å
knote litt for å komme
fram til noe som helst

Slitsomt over tid
for en introvert

Mange meninger og det
kan bli unødige
diskusjoner om valg der
flere løsninger er like
gode

Ulike typer mobber



- Team
- Plattform
- På tvers av team
- Lesesirkel
- Innovasjonsdag



- Fulltid
- Fast tid i uka
- For bestemte typer oppgaver
- Som teambuilding



- Remote
- Hybrid
- Alle rundt samme PC
- Alle med hver sin PC

Lærer å
problemløse av de
andre

Lærer nye måter å
gjøre ting på

Man lærer bort
hverandres styrker

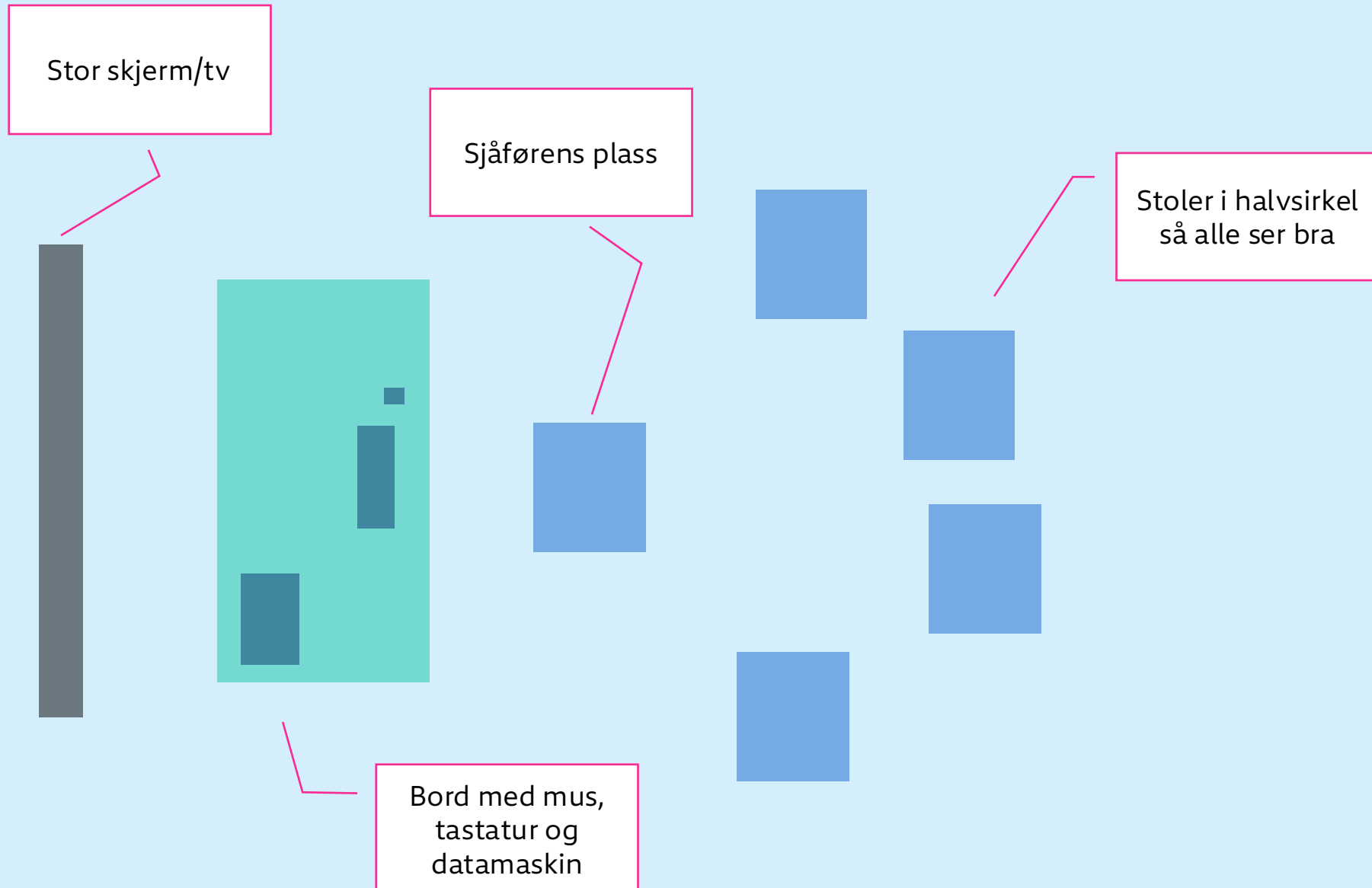
Læring



Hvordan
komme
i gang



Grunnoppsett





Oppstart

- Finn et sted med arbeidsbord, en litt stor skjerm og stoler til alle
- Ta fram én datamaskin med løst tastatur og mus
- Legg bort mobiltelefoner og personlige datamaskiner
- Bruk gjerne tavle for drodling og designarbeid
- 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

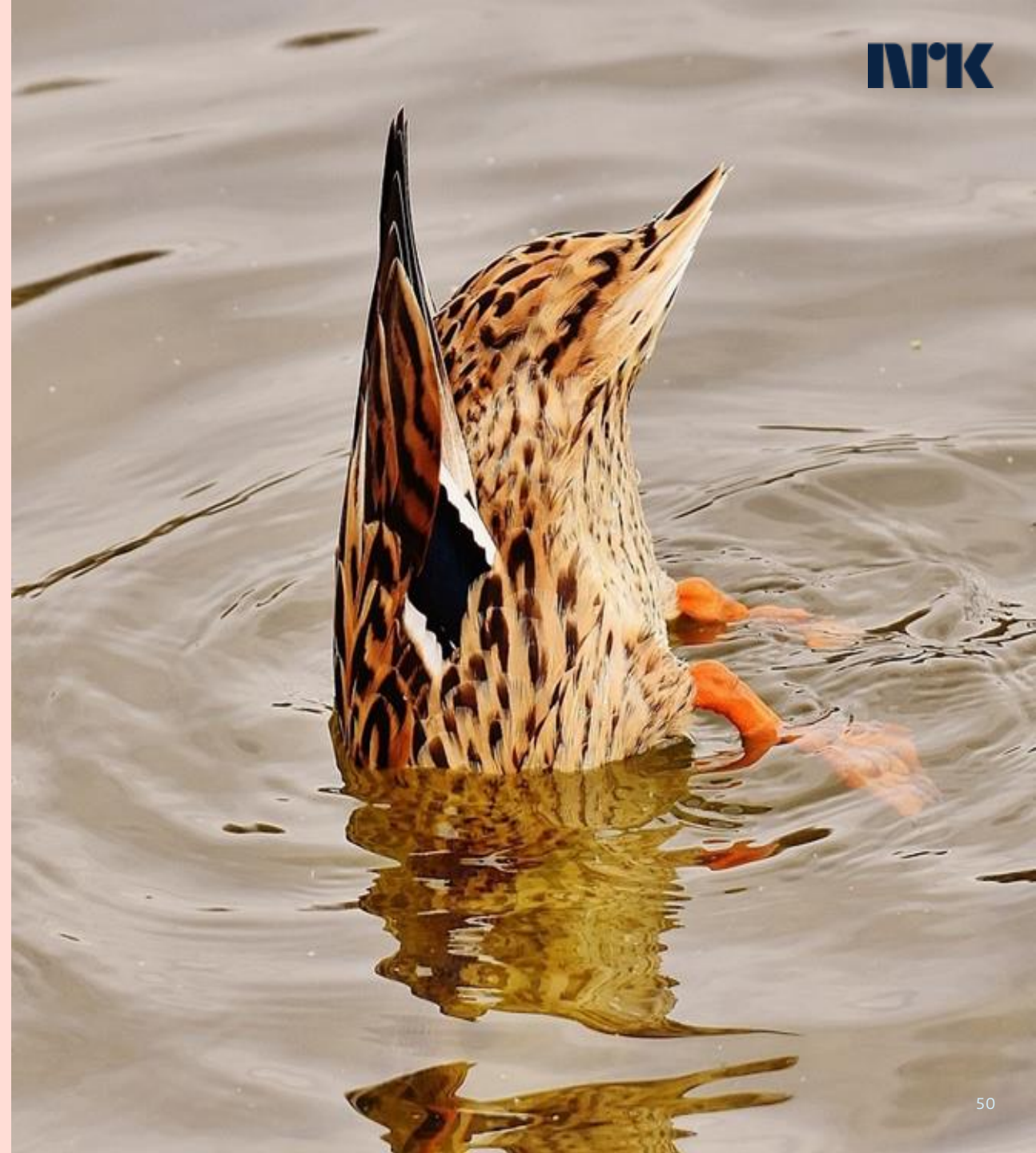
- Sjåføren skal bare gjøre det navigatørene sier
- Navigatørene må snakke sammen, og gi sjåføren gode instruksjoner på det nivået som er nødvendig
- Sjåføren må gi fra seg tastaturet umiddelbart når tiden går ut, gjerne midt i skrivingen av en kodelinje
- Ta pauser ofte, minst en gang i timen

Kjøreregler

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

Rollene er snudd

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



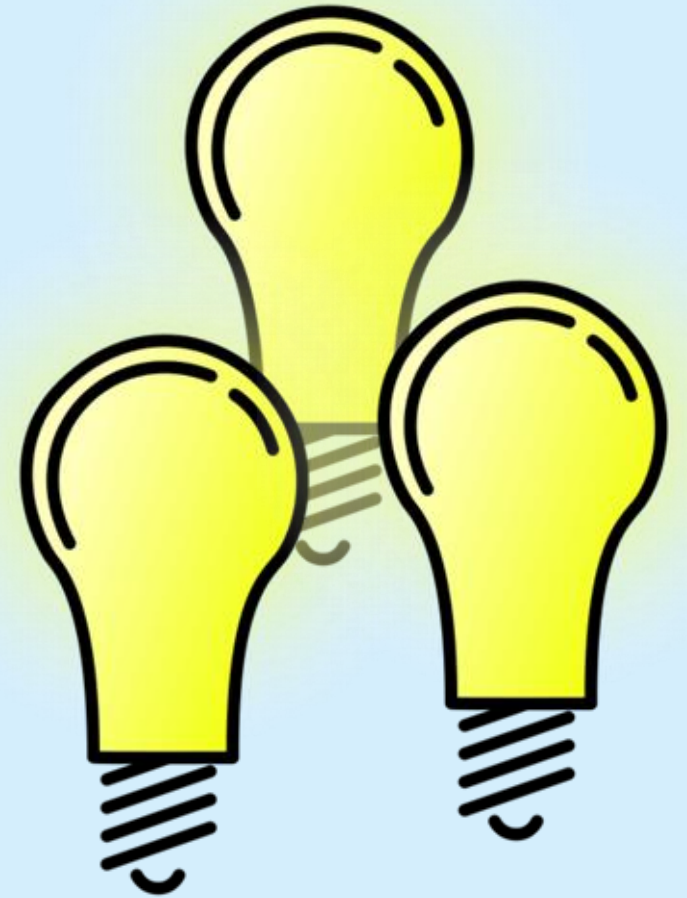
Når du får en ide!

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



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



Vi må ikke være enige

«Samarbeid er å være villig til å prøve andre ideer, uten å måtte være enig i hvilken ide man vil teste eller å måtte se ting på samme måte»

Woody Zuill



Øvelse gjør mester

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

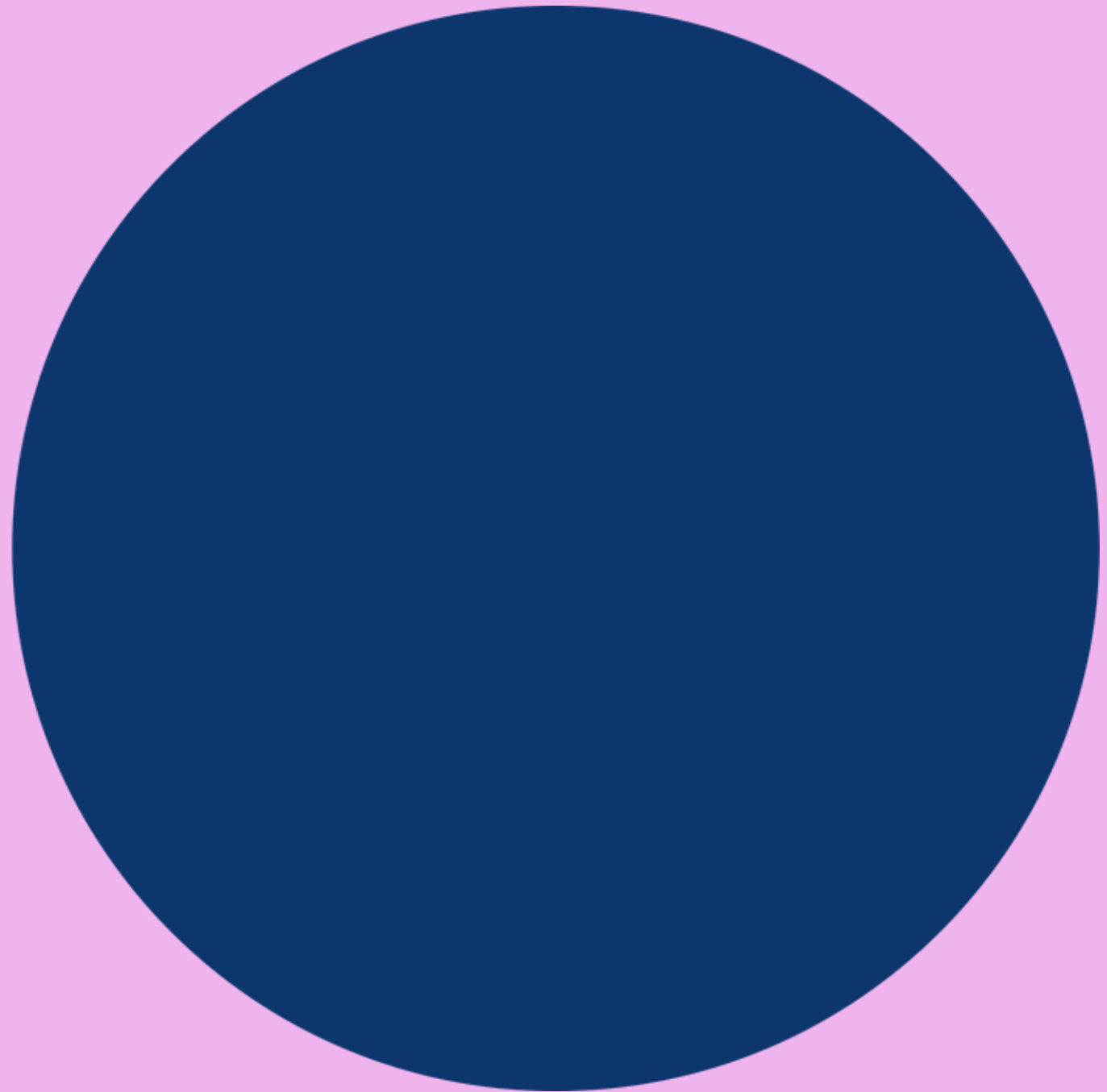


Retro

- Ofte, hver uke eller hver dag!
- Snakk om hva som fungerer
- Let etter løsninger på det som ikke fungerer



Verktøy
og
inspirasjon



Verktøy

Tidsstyring

- <http://mobster.cc/>
- <https://timer.mob.sh/>

Gitkontroll

- <https://mob.sh/>

Mobb-strategier

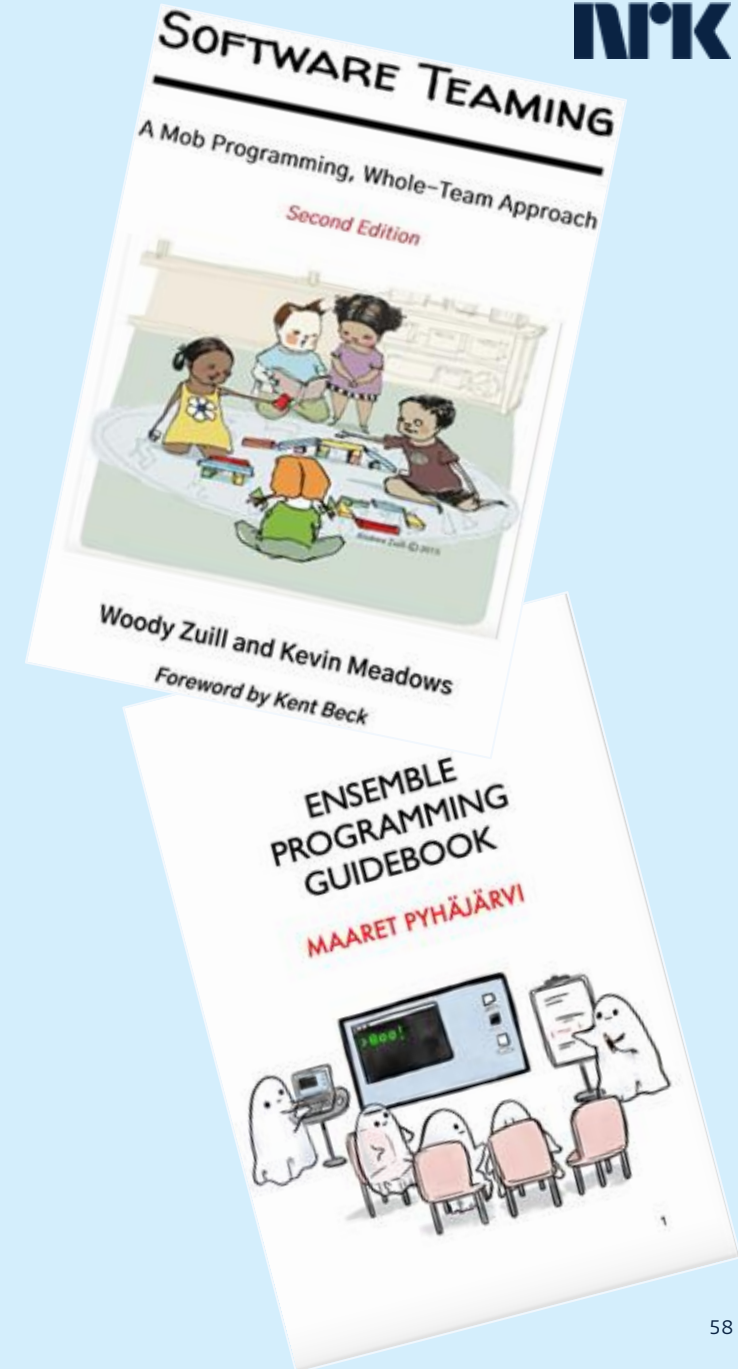
- <https://github.com/nrkno/den-mangfoldige-mobben>

Super samling av ressurser

- <https://trello.com/b/1lfMkCOh/software-profession-resources>

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.youtube.com/watch?v=dVqUcNKVbYg>



<https://www.youtube.com/watch?v=SHOVVnRB4h0>



<https://www.youtube.com/watch?v=T50bk46TbZ0>



<https://vimeo.com/814011581>



Mob Mentality Show
 @mobmentalityshow • 2,79k abonnenter • 328 videoer
 Chris Lucian and Austin Chadwick discuss all things agile and product development from a...mer
twitter.com/mob_mentality og 4 linker til

Abonnerer ▾

Start Videoer Shorts-videoer Podcaster Spillelister Innlegg 🔍

Nyeste Populære Eldste

Open Space Technology for Engineering Leaders: Real Problems, Real Conversations
 with Amy Briggs, Rob Harris, and Peter Christner
 47:30
 24 avspillinger • for 1 dag siden

Building Better Products Together: Henrik Ståhl on Mob Programming, MVPs, and Agile Leadership
 46:55
 70 avspillinger • for 7 dager siden

Can Control Without Competence Cause Chaos? Agile Principle #11 Discussed
 10:22
 88 avspillinger • for 2 uker siden

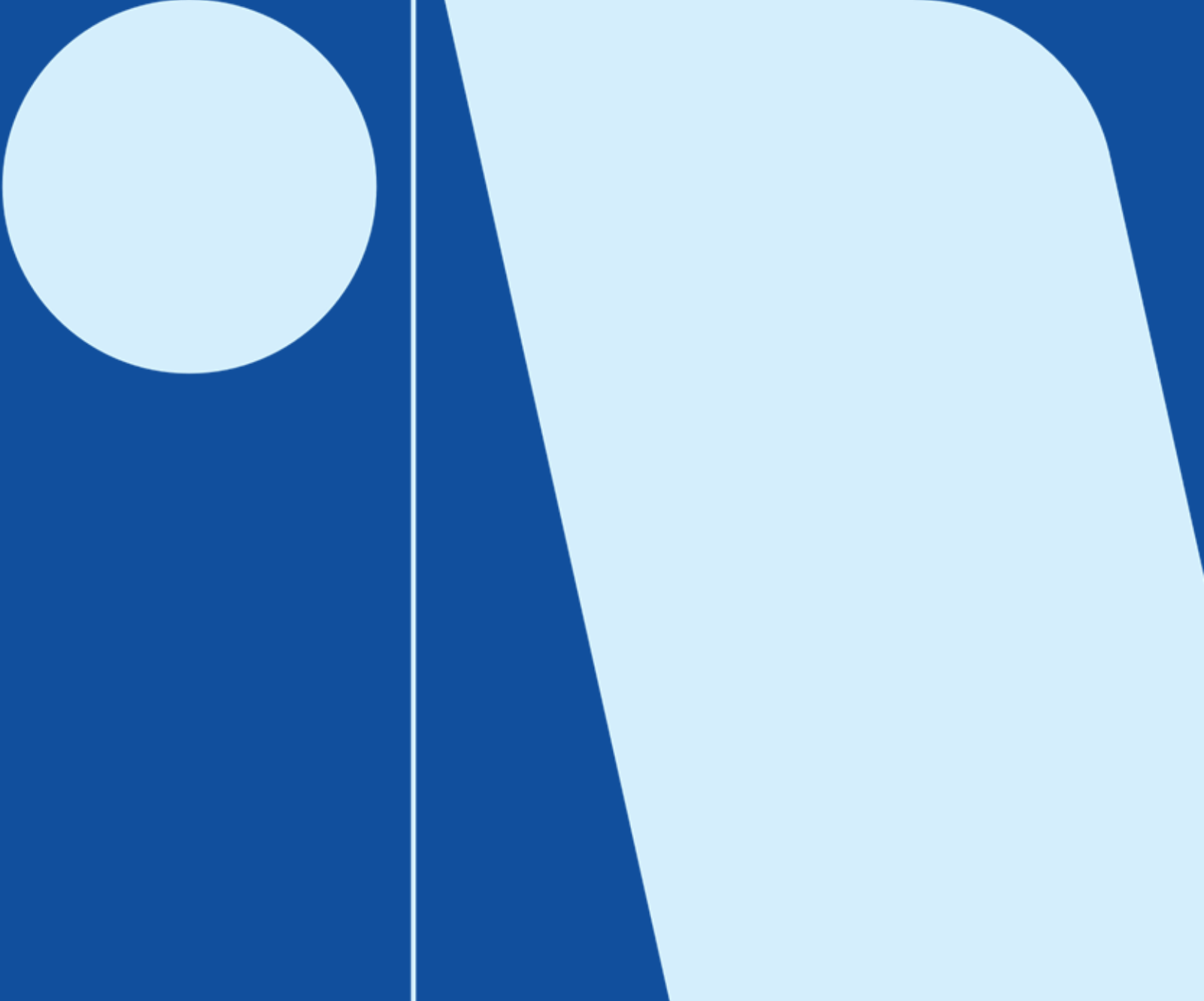
growing the mob
 Chris Lucian
 Austin Chadwick
 35:11
 114 avspillinger • for 4 uker siden

Agentic AI in Action
 Real Stories from the Frontlines of Workflow Automation with...
 40:31
 134 avspillinger • for 1 måned siden

Scaling Agile Teams via Mob Meiosis
 with Brice Ruth
 50:43
 181 avspillinger • for 1 måned siden

Why Team Fit Trumps Resume Skills – Mob Interviewing Stories With William Berning
 43:56
 113 avspillinger • for 1 måned siden

Mobbing at a Startup: Mistakes Made and Lessons Learned
 49:09
 260 avspillinger • for 2 måneder siden



Takk for
meg!

Spørsmål?
heidi.mork@nrk.no

