

Smidig og design

Johanne Thunes

stipendiat ved Institutt for informatikk

Johanne Thunes

- Doktorgradsstipendiat i Informatikk
- Bakgrunn fra IN2000: sjefsgruppelærer (2022) og gruppelærer (2021)
- Forsker blant annet på på digital strategi, outsourcing, low-code/no-code/vibe-coding
- Programleder i PhD podden



Plan for i dag

Del 1:

Smidig metodikk i tverrfaglige team
Intro til "Sprint 0"

PAUSE



Del 2:

Å jobbe tverrfaglig i smidige team
Planlegge "sprint 0"
Tips og triks fra gruppelærerne!

Hva lærer du?

Etter å ha fullført IN2000:

- har du kunnskap om de viktigste systemutviklingsmetodene, inkludert deres styrker og svakheter
- har du kunnskap om sentrale prosesser og aktører i prosjekt- og teamarbeid som anvender smidige prinsipper
- har du kunnskap om følgende aktiviteter innenfor systemutvikling: kravinnsamling og -analyse, design, programmering, testing, samt vedlikehold og videreutvikling
- kan du anvende profesjonelle systemutviklingsmetoder, -teknikker og -verktøy
- har du kompetanse for å arbeide i team og evne til å reflektere over eget og teamets arbeid i systemutviklingsprosjekter
- kjenner du til metoder og prinsipper for innebygget sikkerhet og universell utforming

Etter å ha fullført IN2000:

- har du kunnskap om de viktigste systemutviklingsmetodene, inkludert deres styrker og svakheter
- har du kunnskap om sentrale prosesser og aktører i prosjekt- og teamarbeid som anvender smidige prinsipper
- har du kunnskap om følgende aktiviteter innenfor systemutvikling: kravinnsamling og -analyse, design, programmering, testing, samt vedlikehold og videreutvikling
- kan du anvende profesjonelle systemutviklingsmetoder, -teknikker og -verktøy
- har du kompetanse for å arbeide i team og evne til å reflektere over eget og teamets arbeid i systemutviklingsprosjekter
- kjenner du til metoder og prinsipper for innebygget sikkerhet og universell utforming

Mål med forelesningen

1. Reflektere over hvordan man kan jobbe godt sammen i smidige og tverrfaglige IN2000 team og få utbytte av ulike ferdigheter
2. Gi en inspirasjon til hvordan man kan samle et team (sprint o)
3. En kort innføring i design og designmetoder

Case for context:

Et team bestående av **tre utviklere** og **to designere** skal levere en app i mai. Appen skal bruke API-data og handle om bærekraft.

Teamet kjenner hverandre ikke så godt fra før og de har ikke så mye erfaring med å jobbe i team. Utviklerne vet ikke så mye om hva design er og designeren føler seg ikke som eksperter på koding.

I leveransen i mai skal teamet levere en brukbar og fungerende app, en rapport som beskriver prosessen - teamet skal også argumentere for at appen løser et problem for noen.

Diskusjon: Hvilke utfordringer kan teamet møte?

Snakk sammen med de rundt deg!

Noen utfordringer man kan møte i tverrfaglige team

- Mye ventetid
- Usikkerhet rundt hva de andre holder på med
- Lite forståelse for hva de andre holder på med
- Man jobber bare med det en selv kan fra før

Tilbake til caset

Teamet er **stressa**.

Med en gang begynte designerne å finne behov hos brukeren og et problem å løse. Designeren er stressa. Utviklerne venter på designerne, uten beskjed om hva de skal gjøre vet de ikke hva de skal begynne å kode. Utviklerne kjeder seg litt.

Designerne gjennomfører en lang innsiktsfase, med intervju og survey og finner til slutt ut at teamet skal lage en app som har oversikt over utgått mat i kollektivet. Designerne puster lettet ut.

Utviklerne er stressa. De må begynne å utvikle. Fristen nærmer seg og tiden er knapp, nå er de også stressa! I siste liten får de testet appen. Så leverer de. Ferdig!

Var dette smidig?

Var dette smidig?

Oppstart

Designfase

Utviklingsfase

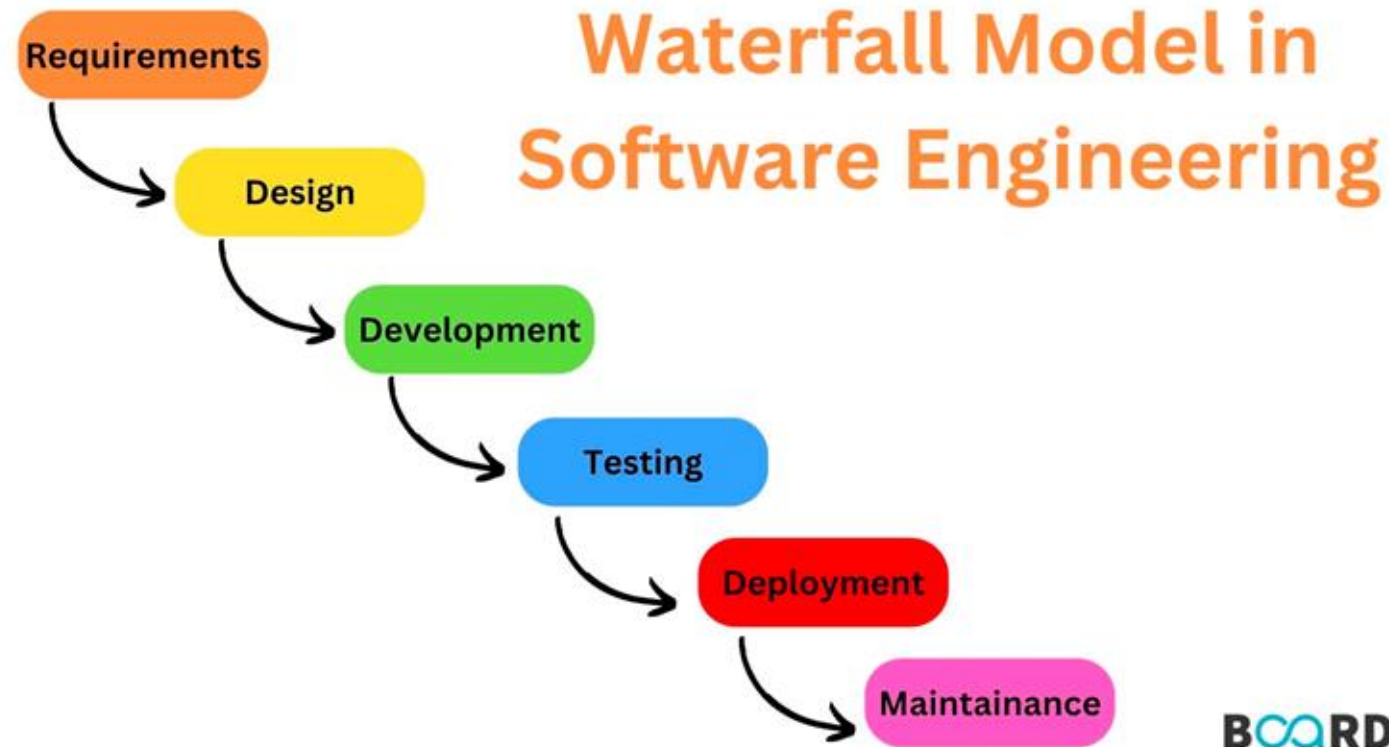
Testfase

Eksamensfrist!

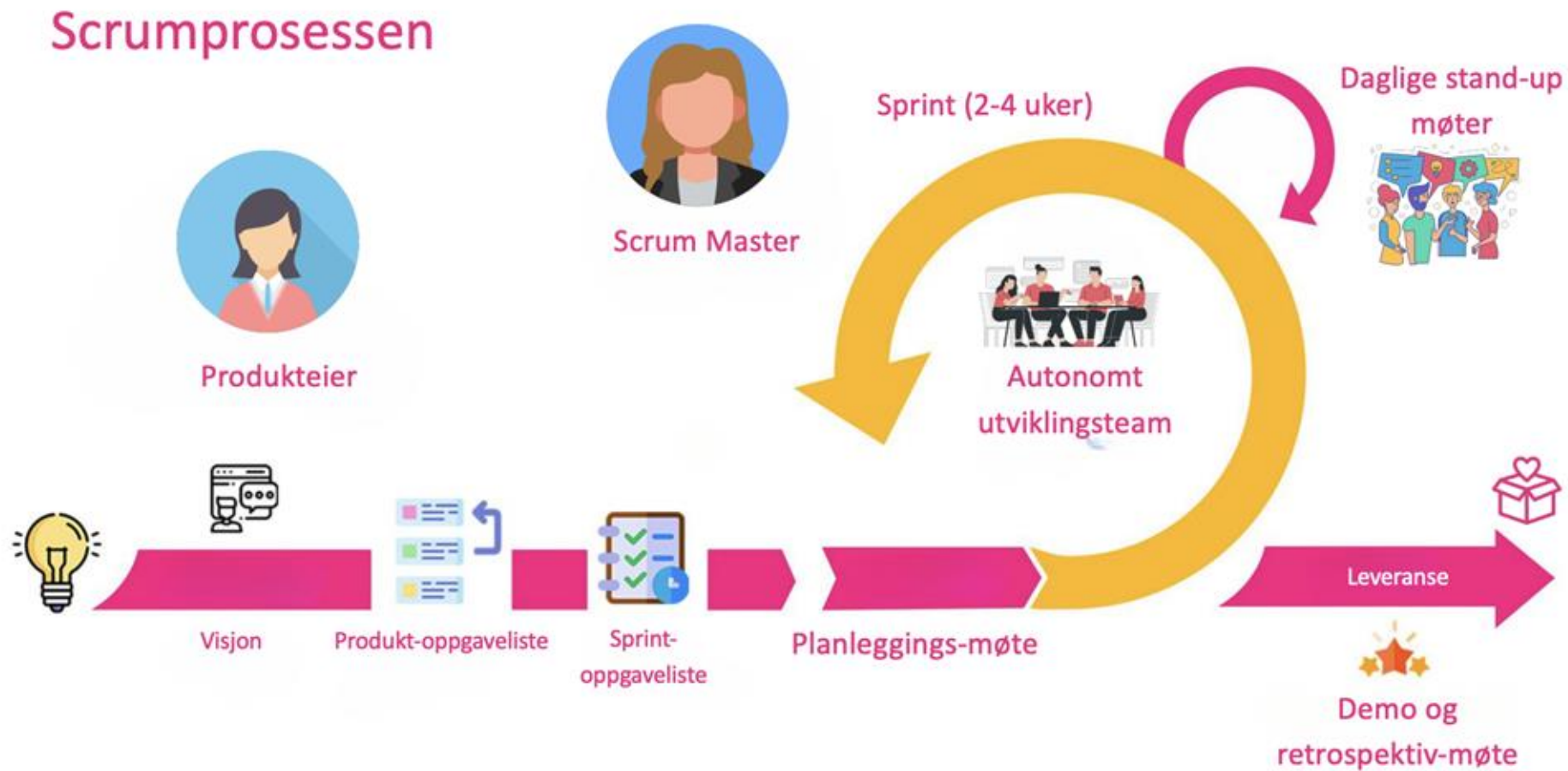
Noen utfordringer fra caset:

- Mye ventetid (utviklerne venter på 'svar' fra deignerne)
- Lite tverrfaglighet - et todelt team
- Igjen: Smidig?

Fossefallsmodellene - plandreven prosess



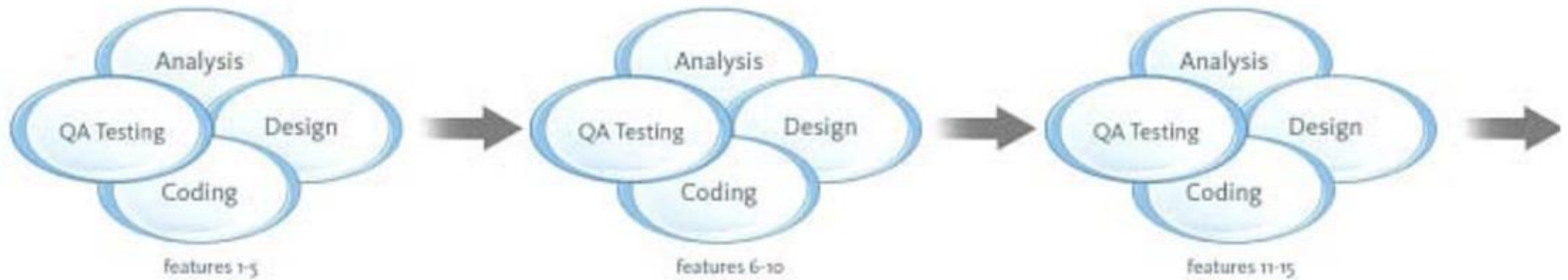
En smidig prosess



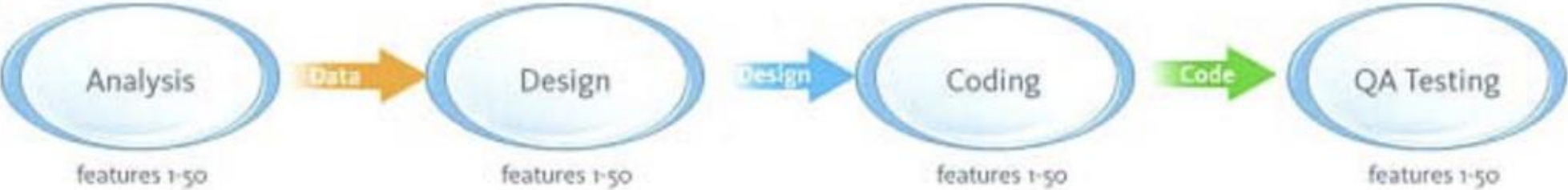
Waterfall



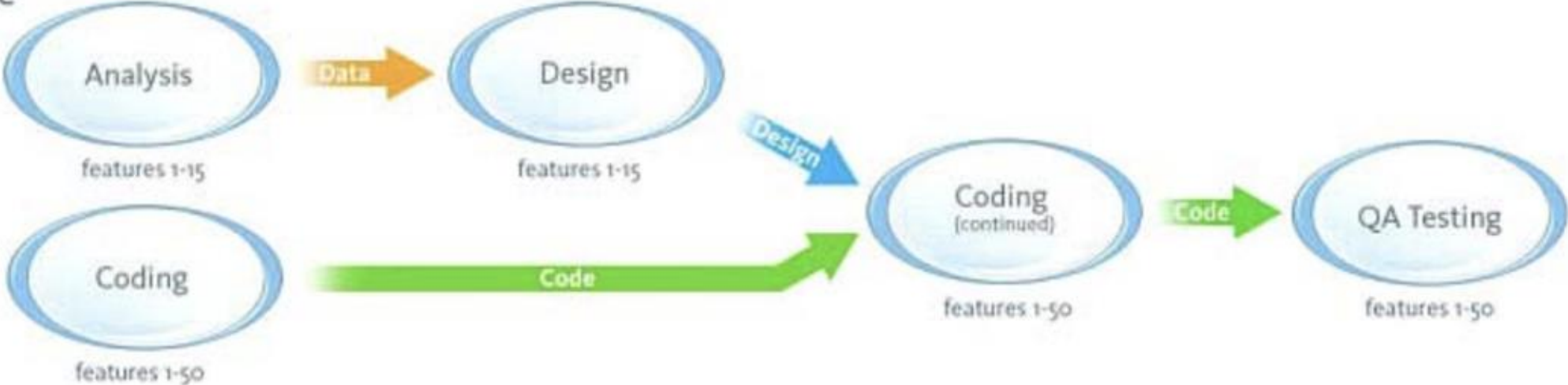
Agile



Waterfall, in theory



Waterfall, in practice



Smidig?

Oppstart

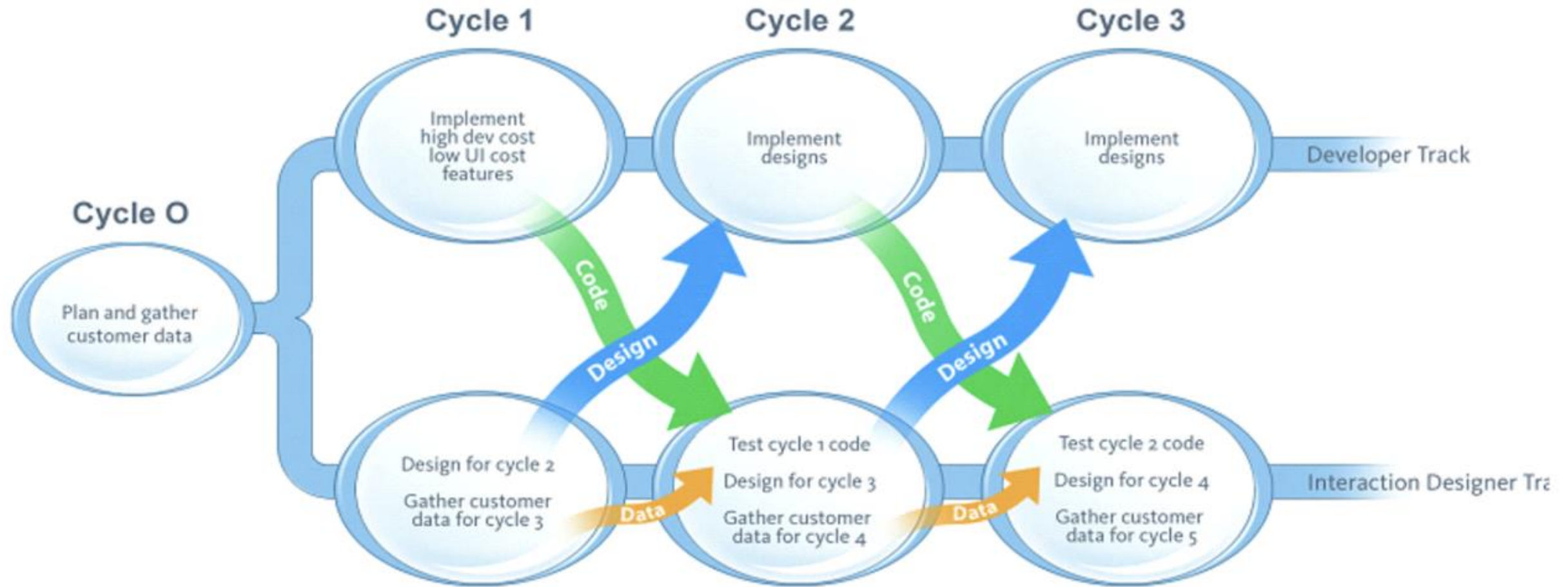
Designfase

Utviklingsfase

Testfase

Eksamensfrist!

TIL inspirasjon!



“Sprint o”

- Teamet jobber **sammen** om starten av prosjektet
 - Hva er problemet?
 - Hva skal vi gjøre?
 - Hvordan skal vi gjøre det?
- Noen fordeler
 - Dere rekker å gå tilbake og gjøre endringer: Fail fast!
 - Felles forståelse for hva problemet **og målet** er. Bedre kommunikasjon
 - Større forståelse for hverandres arbeid og oppgave
 - Mindre ventetid
 - Samme *mentale modell*
- Noen aktiviteter dere kan ha med i spring o:
 - Felles datainnsamling (Dette skal vi se mer på etterpå!)
 - Analyse av data
 - Jevnlig justering av utvikling og UX

Derfor



Hva trykker man på her egentlig?

til Min side forsikring

Har vi rett kontaktinformasjon?



Mobilnummer: [redacted]

E-postadresse: [redacted]

Vennligst oppdater ved å trykke nei nedenfor

Velkommen!

Dette er en app som vil gjøre det lettere å komme seg ut på tur!

Vi ønsker å tilgjengeliggjøre skred- og værdata for enhver turgåer, som et verktøy for å velge trygge turer



NESTE

Fjellvettregler

1. Planlegg turen og meld fra hvor du går
2. Tilpass turen etter evne og forhold
3. Ta hensyn til vær- og skredvarsel
4. Vær forberedt på uvær og kulde, selv på korte turer
5. Ta med nødvendig utstyr for å kunne hjelpe deg selv og andre
6. Ta trygge veivalg. Gjenkjenn skredfarlig terreng og usikker is
7. Bruk kart og kompass. Vit alltid hvor du er
8. Vend i tide, det er ingen skam å snu
9. Spar på kreftene og søk ly om nødvendig



NESTE

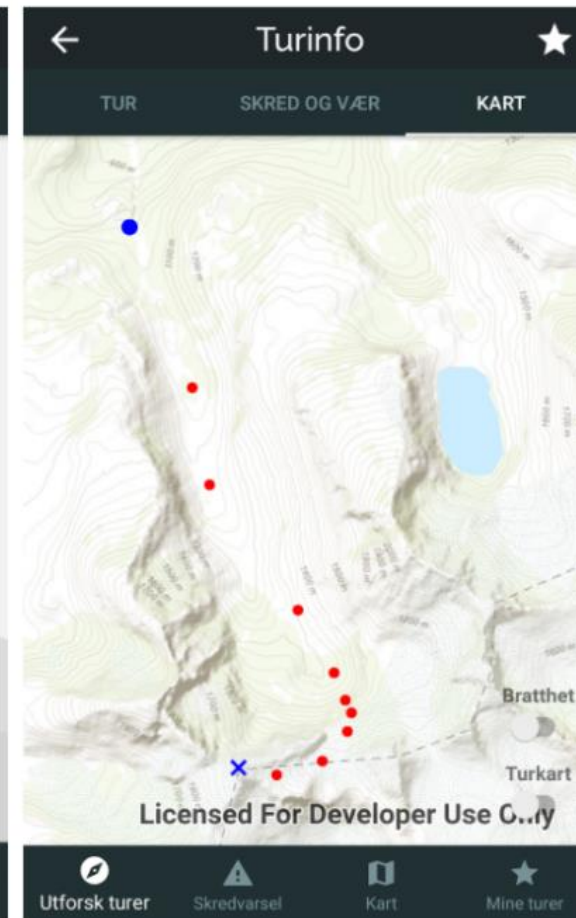
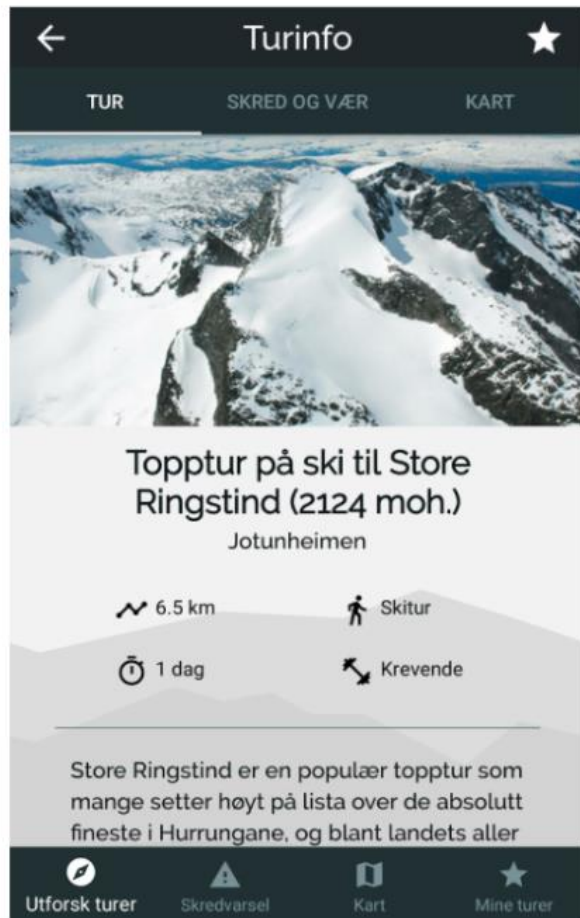
Husk!

Det bes om at enhver tar hensyn og vurderinger på tur. Dette er ikke en fasit, da regionale forskjeller kan føre til ustabile forhold.
Husk å si ifra til dine nærmeste når du skal ut.

God tur!



FERDIG



PAUSE 

Organisere “sprint 0”

Design og designmetoder

Design

- Overordnet forklart kan design defineres som systematiske forsøk på å endre uønskede situasjoner til ønskede situasjoner (Simon, 1979)
- Hva er problemet man skal løse?
 - Et spørsmål mange av dere vil ha i IN2000: Hva skal appen vår gjøre egentlig?
- Snakker ofte om brukerens *behov*

Obs! Det er mange typer design og mange ulike typer designere.
Akkurat som det er ulike typer utviklere.

Å komme frem til hva og hvordan du skal lage

- Hva skal man lage? Flere veier til mål
 - Du begynner med en genial ide - Genius design (F.eks Apples Iphone)
 - Du begynner med å fokusere på behovet til brukeren - UCD - user centered design
Hvem er brukerne og hvilke problemer har de?

Ofte har man rammer i et IT-prosjekt og kan ikke begynne helt fra scratch:

- **IN2000** - lage app en Android Studio app, i kotlin basert på data fra API
 - Det er lurt å stille seg spørsmålet: Hvem skal bruke appen vår?

Noen metoder for å **forstå problemet, brukeren og bruk**

Intervju

Brukertesting

Survey

Geriljatesting



- **Hvem** er brukeren?
- **Hvor** skal systemet brukes?
- Hvilken **rolle** har brukeren i den gitte konteksten?
- Hvilke **behov** har brukeren i den gitte konteksten?

Aka...

Hvilke **krav** stilles til **systemet** basert på disse **behovene** hos **brukeren** i sin **rolle** i den gitte **konteksten**?

Metode: **Intervju**

- Vi snakker ofte om tre typer intervju:
 - Strukturert, **semi-strukturert** og ustrukturert
- En fordel med intervju er at du kommer i **dybden**
- Kan være tidkrevende

Fortsettelse: **Intervju**

- Før intervjuet:
 - Hva er målet med intervjuet?
Eksempel på mål: Finne ut hvordan folk bestemmer hvor de skal bade
 - Lage en *Intervjuplan* - Hvor skal intervjuet være? Hvordan skal du ta opptak? Skal du notere for hånd? Skal den du intervjuer få en liten gave som takk?
 - Lag en *Intervjuguide* - hvilke spørsmål skal du stille og når?
5 steg: **Introduksjon, oppvarming, hoveddel, avrunding, avslutning** (se IN1050-boka, Sharp et al., 2019 (og evt nyere utgave))
- Etter intervjuet:
 - Hva lærte vi egentlig?
 - Transkribere: Her kan et nyttig verktøy være [UiO Autotekst](#)

Metode: **Brukertesting**

- Hvordan bruker egentlig brukeren appen din?
Brukes den som du hadde tenkt?
Er det du har planlagt å lage/ laget intuitivt?

Man kan teste ut fra metrikker

(f.eks Don Normans designprinsipper, eller noen fokusområder dere selv har bestemt)

Metode: Geriljatesting

- Spør "5 tilfeldige på gata" hva de synes om ideen din
- Når er geriljatesting nyttig?
 - Når du vil ha tilbakemelding! Er du helt på jorde eller har du truffet blink?
 - Passer gjerne i et tidlig stadium og på en ide som ikke er helt gjennomarbeidet, men kan benyttes på ulike stadium og flere ganger i løpet av prosessen
- En fordel: Krever relativt "lite ressurser"
- En ulempe: Kan være litt utfordrende å finne igjen de samme folkene senere!

Metode: **Survey**

- Kort fortalt er survey er en samling spørsmål som noen kan svare på
 - Tenk nøye igjennom spørsmålene du stiller og formålet med undersøkelsen
- En fordel: Du kan få mye data på kort tid!
- En ulempe. Det kan være utfordrende å stille oppfølgingsspørsmål og nå samme gruppe to ganger. En mulighet er å f.eks spørre om de vil legge igjen e-posten sin sånn at du kan kontakte dem ved en senere anledning

Analyse av dataen: Å se sammenhenger

- Hva har vi funnet ut av egentlig? Er brukerne enig? Uenige?
- Det er mange måter å analysere data på her er to:
 - Tematisk analyse
 - Affinity diagram (Post-IT lapper, Miro ++)
- Et verktøy/teknikk/metode man kan bruke i kvalitativ analyse
- Innsikt og ideer organiseres i et **hierarki**
- Lager overordnede **kategorier** som man grupperer funnene under
- Kan brukes til **brainstorming** og andre verbale aktiviteter



Hva er målet med alt dette i IN2000 (og alt teamarbeid)?

Forstå problemet du skal løse og hvordan du skal løse det

Erfaringer fra gruppelærerne!

SPØRSMÅL?
johanthu@uio.no