11 Planlegging og dokumentasjon

	Karakteren 2	Karakteren 3 og 4	Karakteren 5 og 6
Planlegging	**Eleven kan;** - lese og forstå krav til it-løsninger - lese og forstå brukerveiledninger for IT-løsninger - lage enkle brukerveiledninger for IT-løsninger	**Eleven kan;** - bruke noen relevante teknikker og verktøy for planlegging og utvikling av IT-løsninger - lage enkle men til dels utfyllende brukerveiledninger for IT-løsninger - gjøre rede for, til en viss grad, hvordan IT-løsninger utvikles og hvilke krav det settes til planleggings - og utviklingsprosessen	**Eleven kan;** - spesifisere krav til IT-løsninger - bruke relevante teknikker og verktøy for planlegging og utvikling av IT-løsninger - lage brukerveiledninger for IT-løsninger - gjøre rede for hvordan IT-løsninger utvikles og hvilke krav det settes til planleggings- og utviklingsprosessen - forklare hensikten med teknisk dokumentasjon og lage slik dokumentasjon
Programmering	**Eleven kan;** - lese og bruke enkel dokumentasjon og kode - definere enkle variabler og velge noen få hensiktsmessige datatyper - tilordne uttrykk til noen få variabler - programmere med noen få enkle variabler	**Eleven kan;** - lese og bruke dokumentasjon og kode - definere enkle variabler og velge noen hensiktsmessige datatyper - tilordne uttrykk til noen variabler - programmere med enkle variabler - lage egne og bruke egne og andres funksjoner eller metoder med parametere til en viss grad - programmere noen funksjoner eller metoder som blir aktivert av hendelser	**Eleven kan;** - lese, bruke og forstå all dokumentasjon og kode - definere enkle variabler og velge hensiktsmessige datatyper - tilordne uttrykk til variabler - programmere med variabler - lage egne og bruke egne og andres funksjoner eller metoder med parametere - programmere funksjoner eller metoder som blir aktivert av hendelser - utvikle og sette sammen del programmer - gjøre rede for hensikten med objektorientert utvikling og begrepene klasse, objekt og arv
Multimedia	**Eleven kan;** - bruke verktøy for å skape og tilpasse digitale objekter av type tekst, tall, bilder, grafikk, animasjon, film og lyd - utvikle enkle multimedieapplikasjoner etter gitte spesifikasjoner	**Eleven kan;** - planlegge å utvikle enkle multimedie-applikasjoner - bruke programmeringsspråk i multimedie-applikasjoner til en viss grad	**Eleven kan;** - planlegge å utvikle multimedie-applikasjoner - bruke programmeringsspråk i multimedie-applikasjoner - vurdere å bruke relevante filformater - vurdere multimedieprodukter med hensyn til grensesnitt og funksjonalitet

Karakteren 1 uttrykker at eleven har svært lav kompetanse i faget.

Ulike arbeidsmetoder

Systemutvikling

Som systemutvikler er du i stand til å omsette din innsikt i brukerbehov til praktiske programbaserte løsninger.

Samarbeid: Programmerer – designer - bruker

Systemarbeid

A: Markedsbehov – hva har samfunnet behov for?

(Apper, programmer ++)

B: Brukerbehov

C: Systemutvikling

D: Arbeidsmetoder

E: Dokumentasjon av funksjonelle krav

F: Teknisk dokumentasjon av programkoden

G: Pseudokode og flytdiagram

A: Skolearena – et brukerverkstøy

Hvilke behov fantes «ute i den store verden» som var grunnlaget for at Skolearena ble skapt?

Hva opplever du som bruker fungerer bra/mindre bra med Skolearena?

Hva tror du avgjør kvaliteten til funksjonene i Skolearena?

B: Brukerbehov

Eksamen H2016 - Matkasser

Oppgave 2: Bestillingsverktøy for matkasse

Bestillingen av matkassene skal foregå på nettsiden. Du skal lage en applikasjon der kunden (brukeren) kan velge en matkasse som inneholder to eller tre middager, og hvor mange personer middagene skal lages til.

Når brukeren har valgt matkasse, skal det komme opp et bilde som illustrerer om det er to eller tre middager som er valgt. Applikasjonen bør sjekke om antall personer som er registrert, er innenfor lovlige/sannsynlige verdier.

Prisen beregnes ut fra valgene, og du kan regne med 80 kroner per person per måltid. Bestilles det matkasser til fem personer eller mer, vil prisen per person være 70 kroner.

Applikasjonen skal bekrefte valgene som er gjort, og navnet og adressen til den som bestiller, skal legges inn og vises sammen med de andre opplysningene.

(Bestillingen vil på en virkelig nettside måtte sendes/lagres ved hjelp av for eksempel e-post eller en fil, men på eksamen trenger du ikke gjøre noe med dette.)

Matkasser

Hva er det en matkasseapplikasjonen skal tilby?

Hva ville du som bruker kreve fungerte bra med denne applikasjonen?

Hva vil avgjøre kvaliteten til denne applikasjonen?

Hvorfor vil folk ønske å bruke det vi har tenkt å lage?

Er det noen som faktisk trenger produktet vi skal lage?

Oppgaven «Matkasser» gir en beskrivelse av brukerbehovene.

Sjekk disse sidene. Hvilke brukerbehov blir dekket/ikke dekket?

Spisriktig.no
Just-eat.no
Easyfood.no

C: Systemutvikling

En systemutviklers viktigste arbeidsoppgaver

1. Lage og forbedre funksjonaliteten til programvaren.

2. Teste at programvaren fungerer som den skal, og rette feil når de oppstår.

3. Drifte programvaren. Føre kontroll med servere, klienter, versjoner, backup, oppetid osv.

4. Holde seg oppdatert på ekstern teknologi som kan benyttes i produktet, for eksempel HTML5, JavaScript, Wordpress, Github, Objective C /App Store, Swift, Java, .Net, Google Analytics, Postgres, Heroku, Elastic Search.

D: Arbeidsmetoder

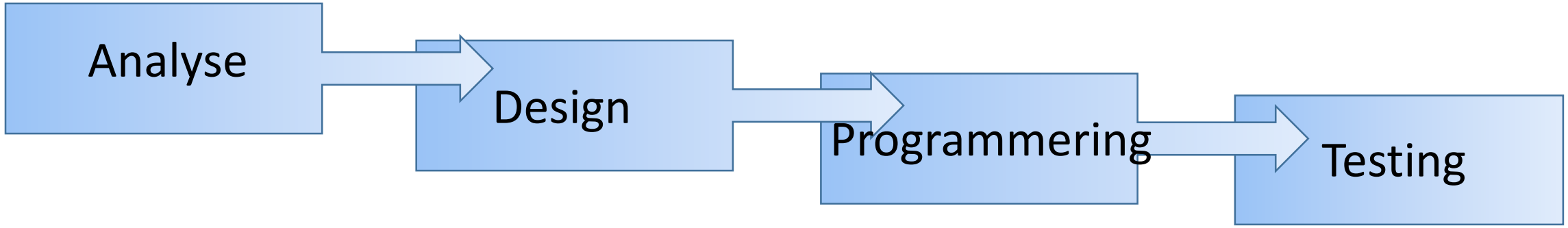
1. **Vannfallsmetoden/fossefallsmetoden** (nå litt utrangert)

2. **Smidig utvikling** (populært, eXtreme programmering (XP) og Scrum)

3. **Lean startup** (det neste store)

1: **Vannfallsmetoden**

I vannfallsmetoden (fossefall) er arbeidet inndelt i oppgaver som skal gjøres i en bestemt rekkefølge:



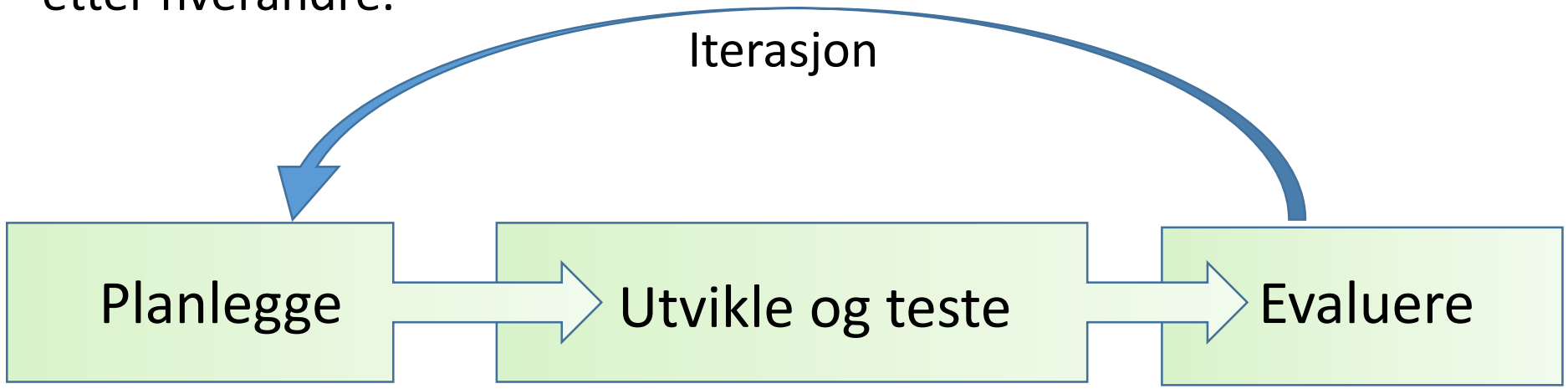
Vannfallsmetoden la ikke til rette for
tilbakemeldinger mellom arbeidsoppgavene, og
dette bremset opp læringen til alle som arbeidet
med IT-løsningen

Testingen startet etter at arbeidet var ferdig.
(Etter at de var ferdige skjønte de hvordan de egentlig skulle ha gjort det…)

2: Smidig utvikling og scrum

Smidig utvikling

I smidig utvikling blir vannfallsmetoden gjennomført mange ganger etter hverandre.



Smidig systemutvikling går ut på at vi arbeider i kortere intervaller som vi kaller **iterasjoner** eller **sprinter**. Smidig utvikling legger dermed til rette for at vi kontinuerlig forbedrer vår egen arbeidsmetode.

Scrum

Scrum har et **scrum-team**, som er et selvorganiserende team.
Ingen leder som forteller hva som skal gjøres, bare hva som er det felles målet.

Teamet selv finner ut hva arbeidet går ut på, og fordeling av arbeidet.

Alle scrum-team har én produkteier, én scrum-master og mange utviklere.

Produkteier: Kundene av produktet vi utvikler. Representerer brukerne.

Scrum-master: Passer arbeidsprosessen og sørger for at forholdene ligger til rette for at alle på scrum-teamet trives og kan jobbe effektivt og målrettet.

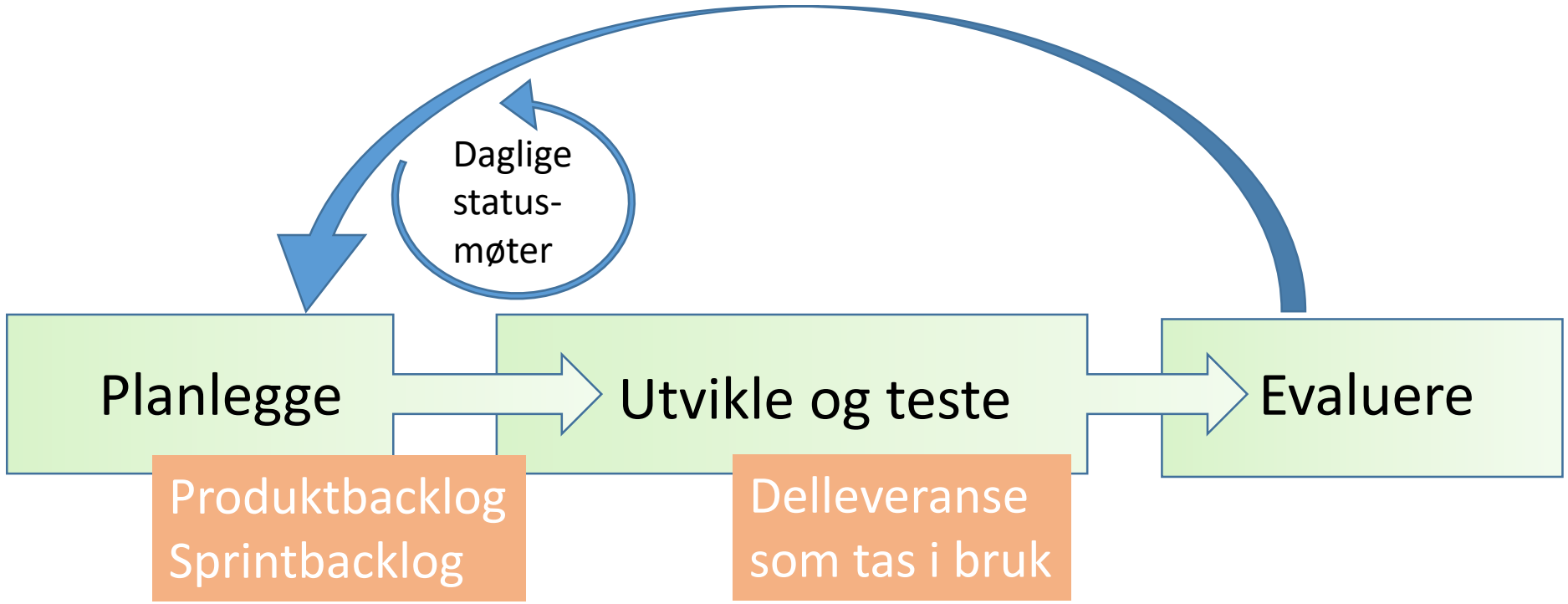
Utviklere: De øvrige team-medlemmene er utviklerne, som løser oppgavene som planlagt.

Sprinter

I **scrum** kalles iterasjonene en **sprinter**.

Målet med en sprint er en **delleveranse** til brukerne.
Vi sier da at vi setter den programvaren vi har utviklet, i produksjon.
Får tilbakemelding fra brukeren.

Sprint, 1 – 4 uker



Vi: Har alle rollene i prosessen. Vi både planlegger, utvikler, tester og evaluerer produktet! ☺

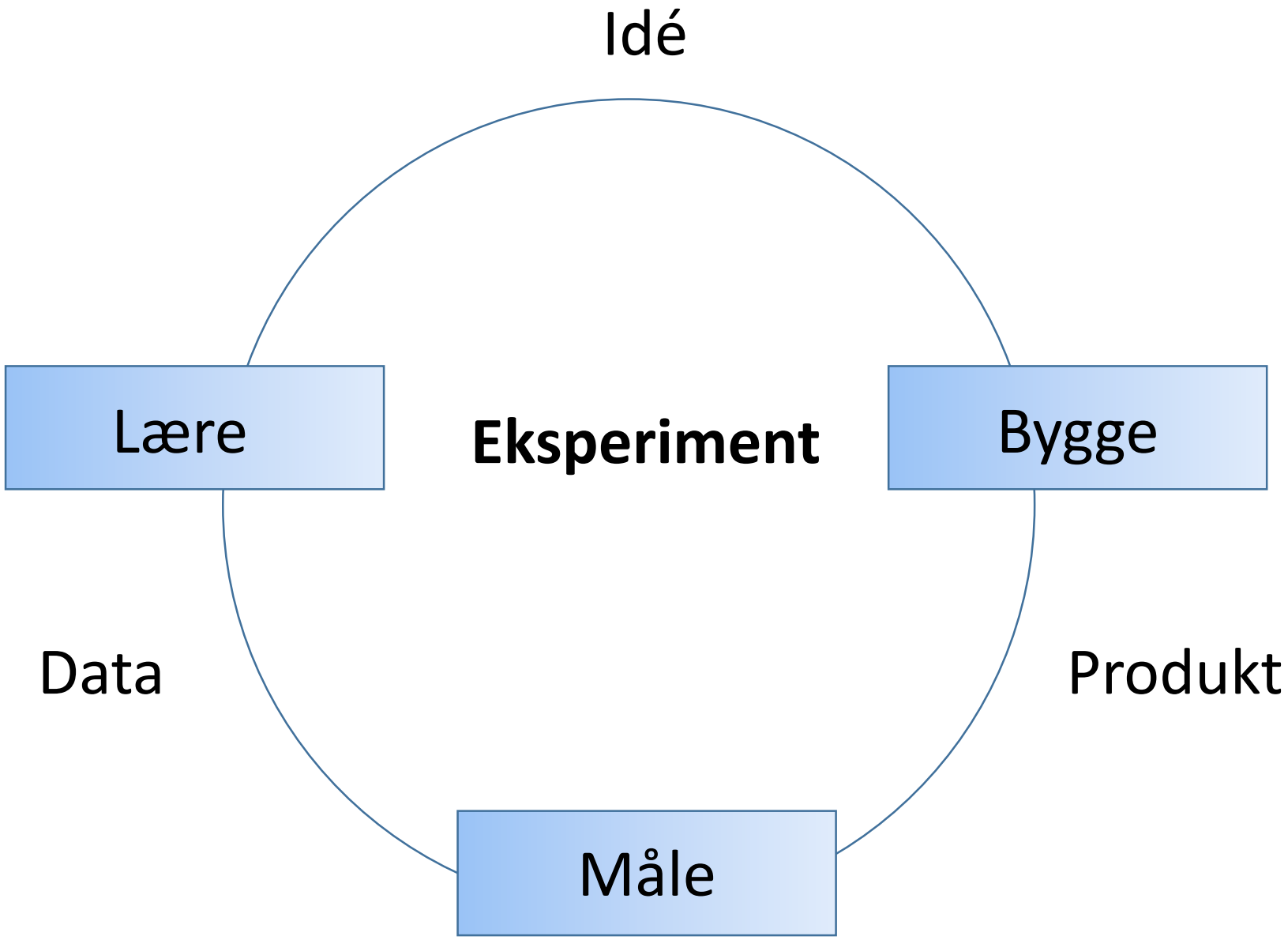
3: Lean startup

lean startup = smart begynnelse

En startidé går over til å bli noe annet (Facebook, hotmail)

En lean startup består av en serie med eksperimenter som kan ta oss stadig nærmere en god produktidé. Eksperimentene må derfor involvere ekte kunder, så de kan gi oss den læringen vi trenger.

Iterasjonene i lean startup dreier seg mer om hvilket produkt vi skal ende opp med og ikke så mye om arbeidsprosessen.



Målet er å komme stadig nærmere en god produktidé

Valg av arbeidsmetode

Problem: Vet vi hvilke behov kunden har?

Løsning: Vet vi hvordan vi skal lage produktet?

Vi må velge riktig arbeidsmetode basert på hva vi vet,
og hva vi ikke vet, før vi begynner.

E: Dokumentasjon av funksjonelle krav

Flere systemutviklere arbeider med den samme metoden

Dokumentasjon deles inn i to hoveddeler:

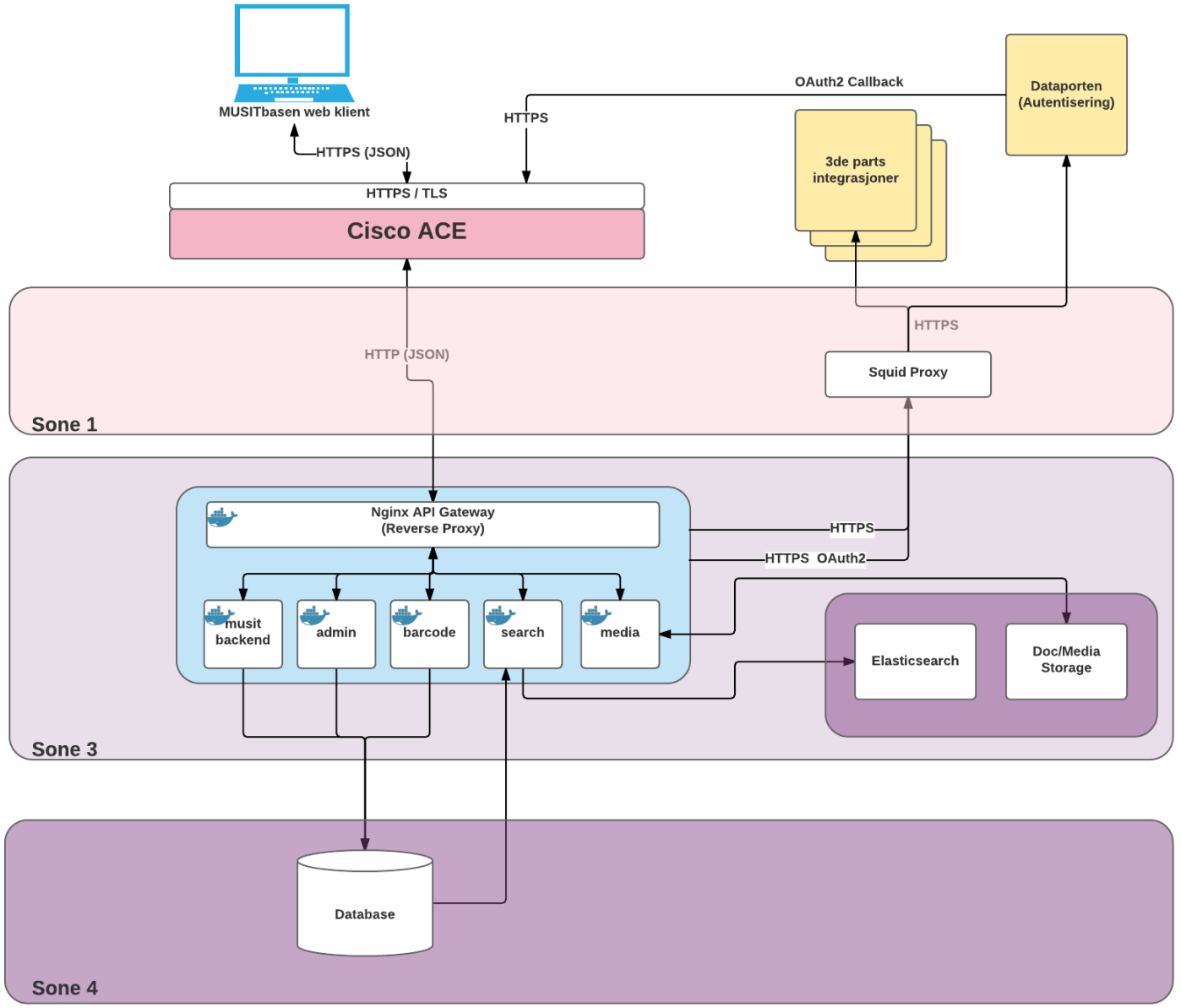
1. **Produktdokumentasjon**: Dokumentasjon av produktet.
2. **Teknisk dokumentasjon**: Dokumentasjon av programkoden.

Funksjonelle krav, skisser og wireframes/flytskjema, use case (arbeidsflyt), UI (user interface/brukergrensesnitt)

Produktdokumentasjon

CE-merkede byggevarer	Ikke CE-merkede byggevarer
CE-merking	**Merking og/eller produktdokumentasjon** *(norsk, svensk, dansk)*
Ytelseserklæring *(norsk, svensk, dansk)*	**Montasjeveiledning** *(norsk, svensk, dansk)*
Montasjeveiledning *(norsk, svensk, dansk)*	**Opplysninger om farlige stoffer** *(norsk, svensk, dansk)*
Opplysninger om farlige stoffer *(norsk)*	**Sertifikater fra tredjepartsorganer** *(skal lett kunne forstås)*

Teknisk dokumentasjon



F: Teknisk dokumentasjon av programkoden

<!-- -->

//

/* */

http://it2cha.com/Kap 11 Planlegging og dokumentasjon/Kap 11 Teknisk dokumentasjon av programkode s 333.html

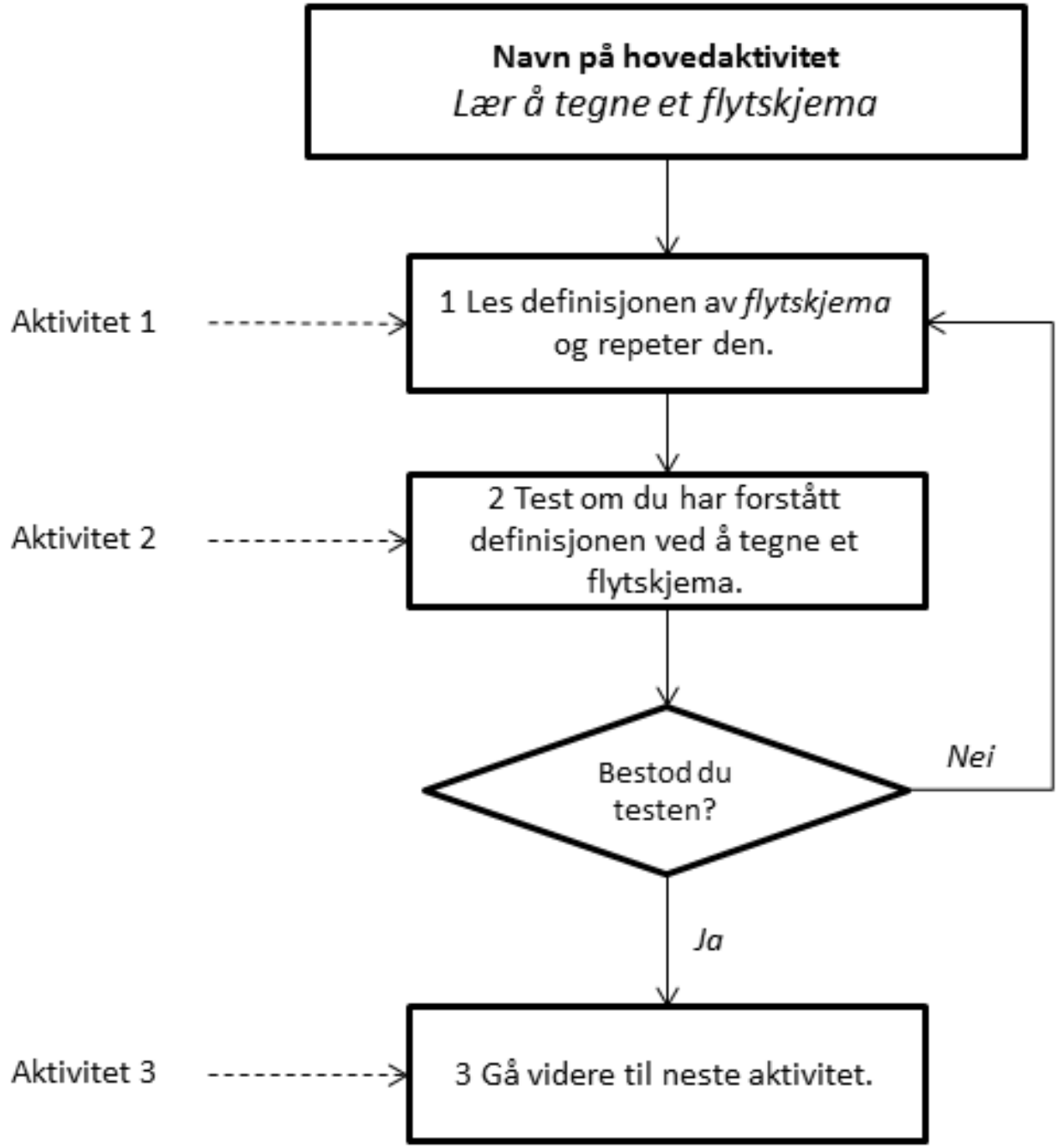
G: Pseudokode og flytdiagram

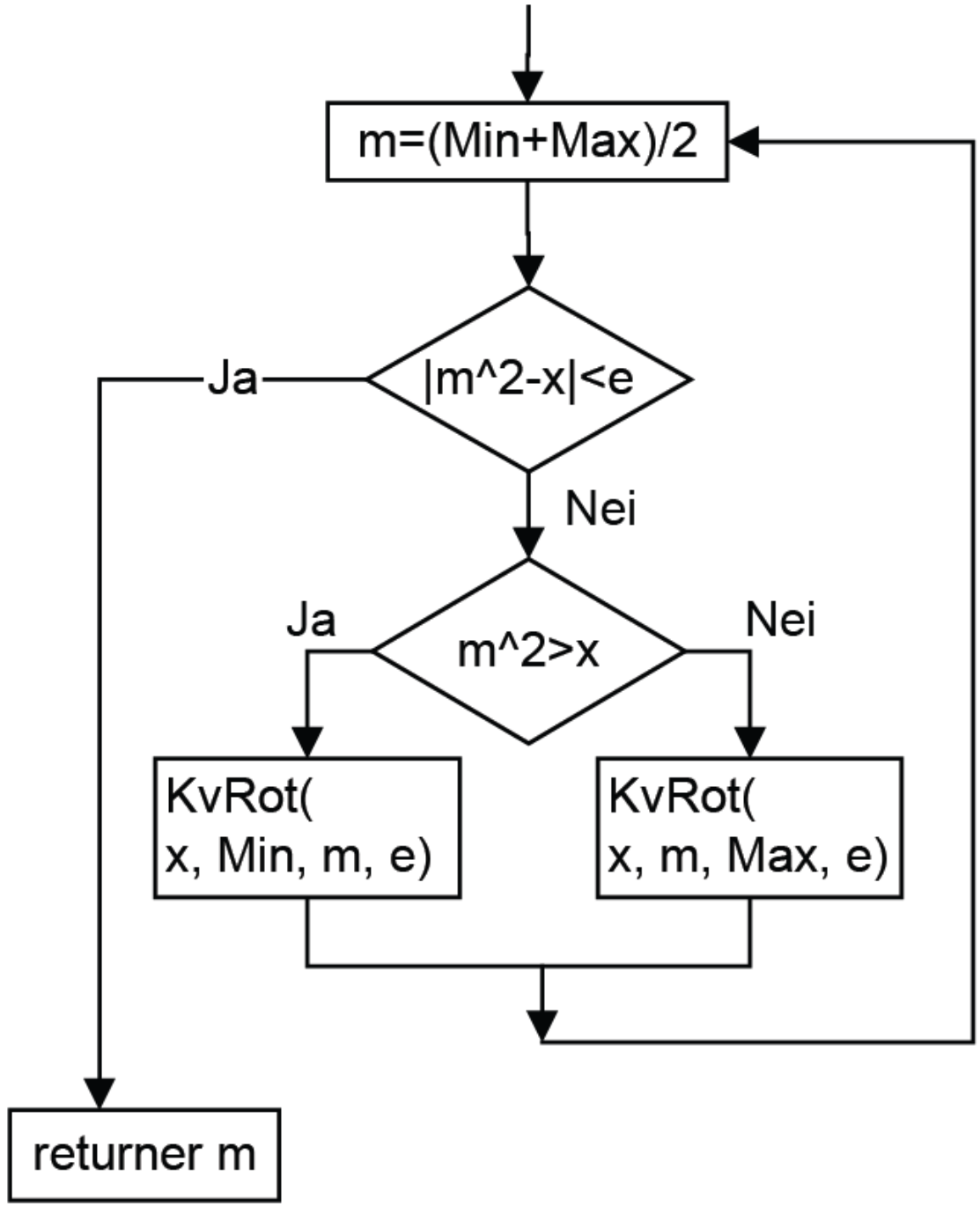
Planlegging av programkoden kan skrives som

- Lister
- Pseudokode
- Flytdiagram

Lister

Jeg er sulten
Tar et valg om jeg vil spise
Sjekker kjøleskapet for mat
Hvis det er nok mat – lager mat
Hvis ikke nok mat – går til butikken
Legger mat i handlekurven
Sjekker om nok penger
Hvis nok penger – kjøper mat
Hvis ikke – reduserer mat i handlekurven
Hvis nok mat – lager mat
Hvis ikke nok mat – går sulten til sengs…


Navn på hovedaktivitet
Lær å tegne et flytskjema
Aktivitet 1
1 Les definisjonen av flytskjema og repeter den.
Aktivitet 2
2 Test om du har forstått definisjonen ved å tegne et flytskjema.
Bestod du testen?
Nei
Ja
Aktivitet 3
3 Gå videre til neste aktivitet.



KvRot (x, Min, Max, e)

sett m lik (Min+Max)/2

IF |m^2-x|<e
    returner m

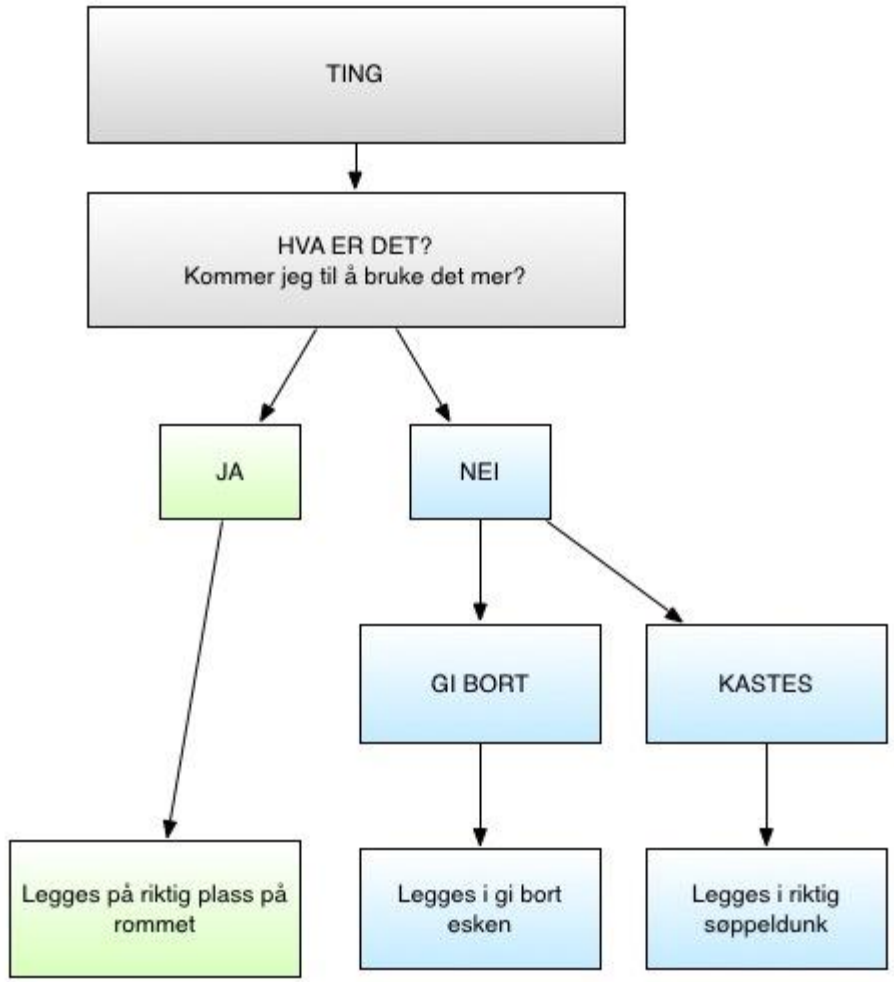
ELSE
    IF m^2>x
        m=KvRot(x, Min, m, e)

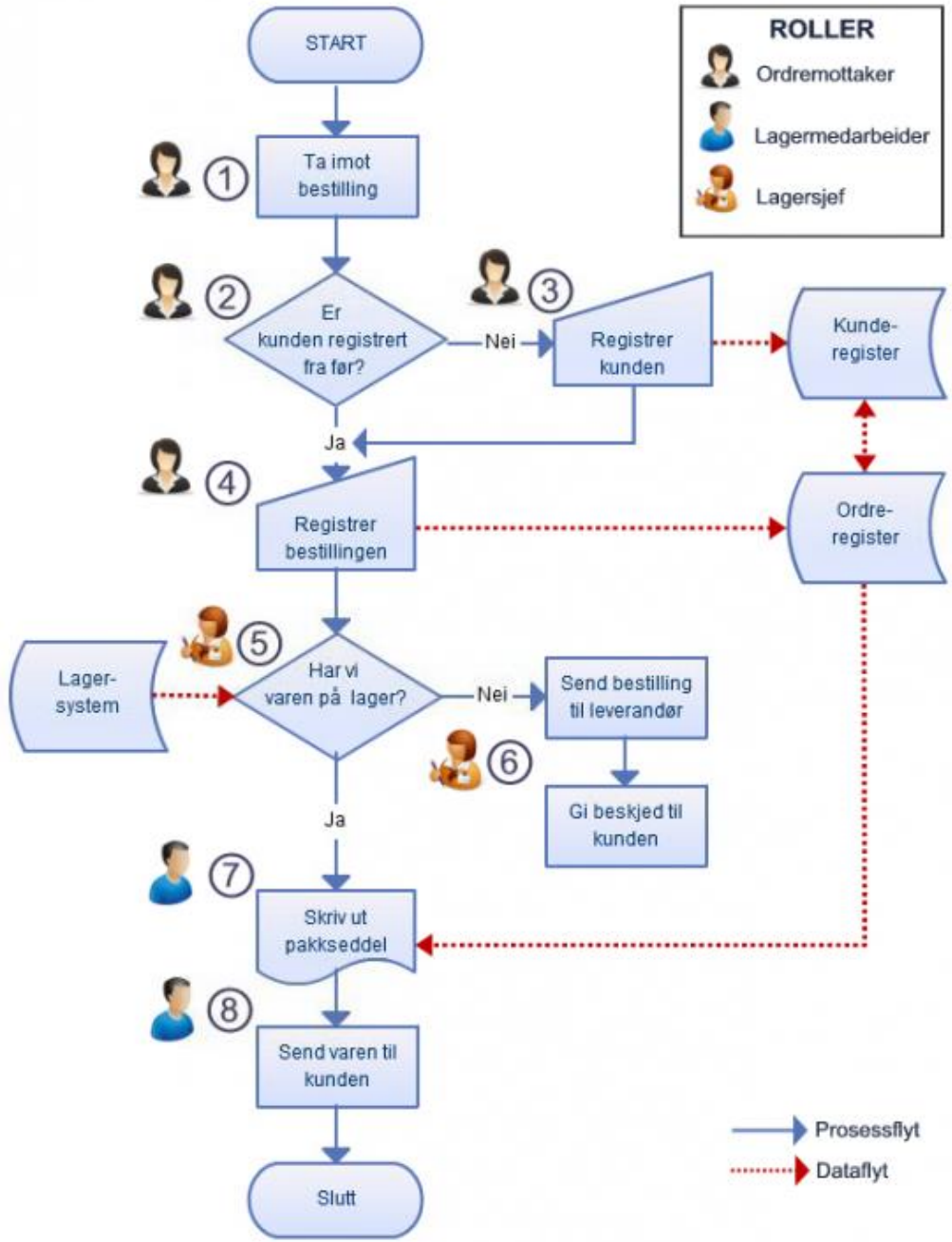
    ELSE
        m=KvRot(x, m, Max, e)

    END
END

RYDDEFLYT

for _____

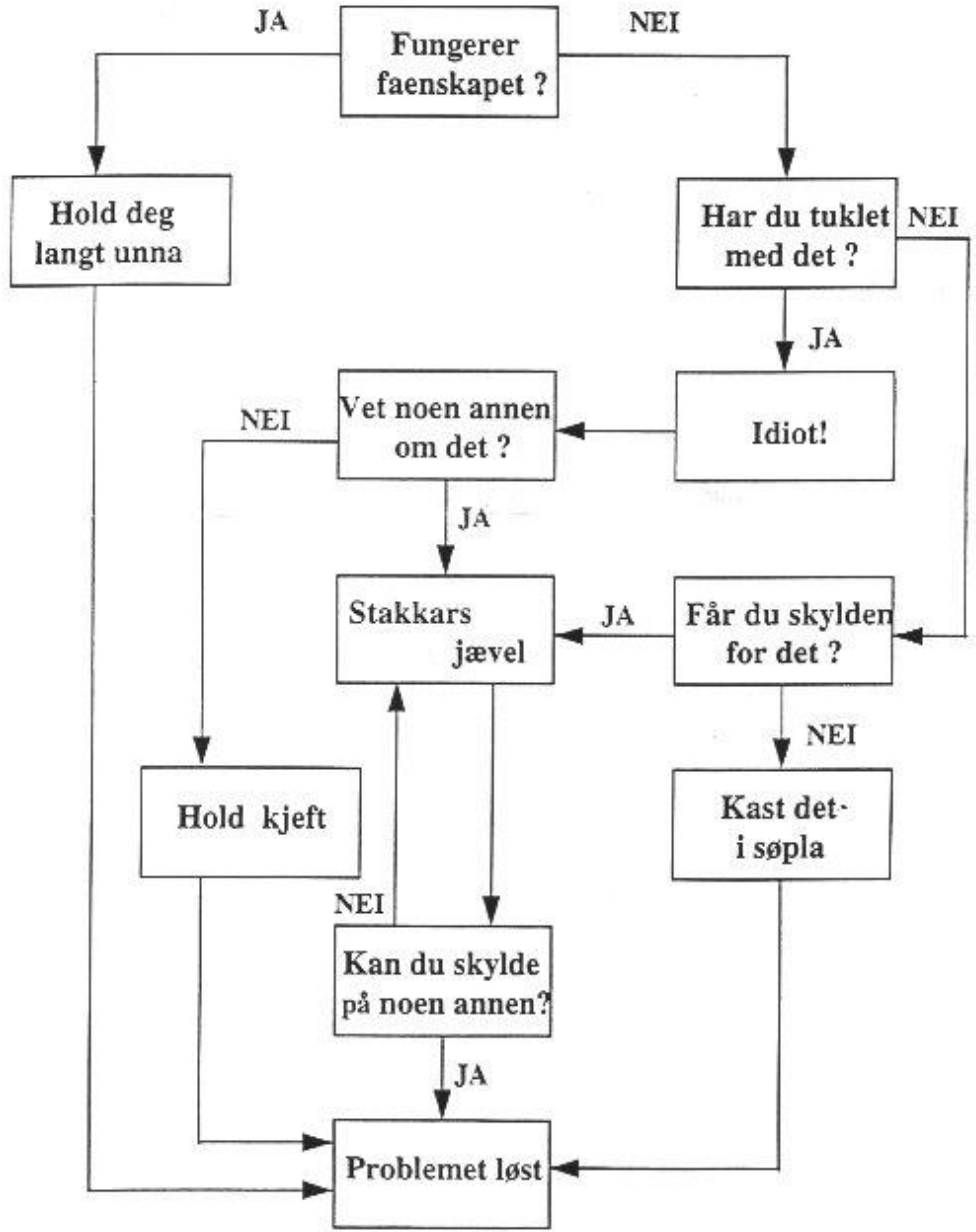



START
ROLLER
Ordremottaker
Lagermedarbeider
Lagersjef
Ta imot bestilling
1
2
Er kunden registrert fra før?
Nei
3
Registrer kunden
Kunde-register
Ja
4
Registrer bestillingen
Ordre-register
5
Lager-system
Har vi varen på lager?
Nei
Send bestilling til leverandør
6
Gi beskjed til kunden
Ja
7
Skriv ut pakkseddel
8
Send varen til kunden
Slutt
Prosessflyt
Dataflyt

ISO 9000

Kontrollskjema

"Problemløsning"



1. Finn en partner og øv dere på dokumentasjon ved å gjøre disse oppgavene sammen:

https://docs.google.com/document/d/16XvOQxXkt22rX5dhwLntYQGhuW1OX_9NdJXAHEe4g0w/edit?usp=sharing

Kopier innholdet og opprett et dokument dere samskriver i ☺

2.
Bli kjent med begrepene

- Kravspesifikasjon
- Pseudokode
- Wireframes (skisser /flytskjema)
- Koder
- Feilsøking
- Funksjonelle krav
- Use case (arbeidsflyt)
- UI (user interface/brukergrensesnitt)

Gjør oppgave 2 Stein, saks og papir