Vurderingsoppgave enkel

Karakter maks 3

# Decompositon

## Hva er decomposition?

1. Dele opp et komplekst problem eller system i mindre, mer håndterbare deler
2. Legge til detaljer for å gjøre et problem mer komplekst
3. Når du ignorerer unødvendige detaljer i et problem

## Hvorfor bruker vi decomposition på komplekse problem?

1. For å gjøre det vanskeligere å løse
2. For å endre problemet har vi
3. For å gjøre det enklere å løse

## Hvilket av disse er et eksempel på decomposition?

1. Se en mekaniker reparere en sykkel
2. Ser på forskjellige sykler for likheter mellom dem
3. Finne ut hvordan en sykkel fungerer ved å se i detalj på de forskjellige delene som utgjør sykkelen

## Hvilket av disse er et eksempel på decomposition?

1. Bryte problemet med å organisere et kakesalg i mindre deler, for eksempel hvem som skal bake kakene og når du skal holde kakesalget
2. Ta problemet med å bake en kake og tenke på hvordan vi kan gjøre det best mulig kake
3. Ser på hva forskjellige typer kake kan gjøres

## Hvor ofte bruker vi decomposition på problemer?

1. Vi dekomponerer dem aldri, datamaskiner gjør dette for oss
2. På daglig basis, ofte uten å tenke på det
3. Av og til, men vi trenger egentlig ikke å

## Hvilke av disse vil IKKE være involvert i å bryte ned et problem?

1. Tenker på hvordan problemet kan deles inn i mindre deler
2. Finne ut hvem som kan hjelpe deg med å løse en del av problemet
3. Legge til flere deler i problemet slik at det blir mer komplekst

## Hvilket av disse er et eksempel på decomposition?

1. Gjett hvem som har løst en forbrytelse fra å se på mønstre som har skjedd før
2. Løse det komplekse problemet med en forbrytelse ved å bryte det ned i når forbrytelsen ble begått og var det noen vitner
3. Ser på hvilke forskjellige forbrytelser som kan begås

# Pattern recognition

## Hva er Pattern Recognition?

1. Bryte ned et komplekst problem i mindre problemer
2. Bygge modeller fra mønstre
3. Leter du etter likheter blant og innenfor problemer

## Hvorfor trenger vi å se etter mønstre i problemer?

1. Mønstre gjør det lettere for oss å løse komplekse problemer
2. Mønstre gjør det vanskeligere å løse komplekse problemer
3. Vi trenger ikke å lete etter mønstre

## Hvilket av følgende inneholder et mønster?

1. Alle biler har hjul
2. Bilen min er blå
3. Min venns bil har en MP3-spiller

## Hvilket av følgende inneholder et mønster?

1. Denne papegøyen er blå og grønn
2. Alle papegøyer har vinger
3. Papegøyen min kan si "Hallo, gutter!"

## Hvilket av følgende inneholder et mønster?

1. Huset mitt har en hage
2. Huset mitt og min venns hus har hager
3. Min venns hus har en hage

## Hvilket av følgende mønstre inneholder IKKE et mønster?

1. Alle hunder har haler
2. Våre hunder liker turer
3. Hunden min liker å svømme

## Hvilket av følgende mønstre inneholder IKKE et mønster?

1. Noen klokker har alarmer
2. Noen klokker er digitale
3. Denne klokken har hender

## Hva kan skje hvis vi ikke ser etter mønstre?

1. Vår løsning kan være ineffektiv
2. Vi løser kanskje ikke problemet på riktig måte
3. Vi kan lage en feil eller ineffektiv løsning

# Abstraction

## Hva er abstraksjon?

1. Prosessen med å filtrere ut unødvendige detaljer
2. Prosessen med å filtrere ut irrelevante egenskaper
3. Prosessen med å filtrere ut irrelevante egenskaper og unødvendige detaljer

## Hva er en modell?

1. En modell er en representasjon av et problem
2. En modell er et dataprogram
3. En modell er et regneark

## Hvilket av følgende kjennetegn er en generell egenskap?

1. Hunder løper raskt
2. Denne hunden har en våt nese
3. Denne hunden har en brun frakk

## Hvilket av følgende kjennetegn er en generell egenskap?

1. Denne båten har en motor
2. Noen båter har seil
3. Denne båten er blå

## Hvilket av følgende er IKKE en generell egenskap?

1. Hunder løper raskt
2. Hunder har nese
3. Denne hunden har en brun frakk

## Hvilket av følgende er IKKE en generell egenskap?

1. Bøker er morsomme å lese
2. Denne boken er morsom å lese
3. Bøkene mine er morsomme å lese

## Når du tegner en hund, hvilke av følgende egenskaper kan ignoreres?

1. Hunder løper raskt
2. Hunder har poter
3. Hunder har nese

## For å kunne tegne biler, hvilken av følgende egenskaper er det nødvendig å vite om?

1. Hvem eier dem
2. Hvem selger dem
3. At de har hjul

## Hus har vegger, dører og folk bor i dem. Hvor mange av disse egenskapene trenger vi å vite for å tegne et hus?

1. Én
2. To
3. Tre

# Algorithms

## Hva er en algoritme?

1. Mønstre og trender som brukes til å løse et problem
2. Et sett med trinnvise instruksjoner for å løse et problem
3. Et programmeringsspråk

## Hva brukes algoritmer til?

1. Slik planlegger du løsningen på et problem
2. Som en plattform for å programmere en løsning
3. Slik tester du en løsning på et problem

## Hvordan kan en algoritme representeres?

1. Som et flytskjema
2. Som pseudokode
3. Som flytskjema eller pseudokode

## Hva er et flytskjema?

1. Et diagram som representerer et sett med instruksjoner
2. Et språk på høyt nivå som har spesifikk syntaks
3. En måte å beskrive et sett med instruksjoner på som ikke bruker bestemt syntaks

## Hva er det riktige symbolet for en prosessinstruksjon i et flytskjema?

1. Et rektangel
2. Et parallellogram
3. En firkant

## Hva er riktig symbol for inndata i et flytskjema?

1. Et parallellogram
2. Et rektangel
3. En firkant

## Hva er det riktige symbolet for utdata i et flytskjema?

1. Et parallellogram
2. Et rektangel
3. En rombe

## Hviordan kobles instruksjoner sammen i et flytskjema?

1. En linje
2. En dobbel linje
3. En pil

## Hva er det riktige symbolet for en beslutning i et flytskjema?

1. Et rektangel
2. En rombe
3. En firkant

## Hva er pseudokode?

1. Et språk på høyt nivå som har spesifikk syntaks
2. En måte å beskrive et sett med instruksjoner på som ikke bruker bestemt syntaks
3. Et diagram som representerer et sett med instruksjoner

# Systemutvikling

## Kravinnsamling:

* Hva er en forstudie

Forstudie er en fase innenfor kravinnsamlig, den består av å analysere risiko, kost og nytte. Du må finne ut om det er en effektiv løsning og planlegge hvordan og når du skal gjøre oppgaver.

* Hvordan samle inn data

Sjekke eksisterende dokumentasjon og god kommunikasjon med klienter. Internett kan også ha mye nyttig data. Det er også viktig å tolke og forstå data riktig

* Hva vil det si å analysere innsamlede data

Du må forstå dataen og tolke den, dele den opp i mer forståelige deler for å forenkle den sånn at det blir lettere å løse problemet

* Hvordan spesifiserer vi krav

Når du spesifiserer kravene skal de stå i et kravdokument kalt kravspesifikasjon, kravene skal skrives forståelig, presist og skal ikke inneholde en løsning. Det beskrives også hvor mye arbeid som legges i utvikling av produkt.

* Hvordan kan vi validere innsamlede krav

Det skal vurderes om kravene er det klienten faktisk vil ha, derfor må man vurdere kravene, når man gjør dette bør man sette krav til kravene.

Har alle samme forståelse av kravene?

Er det noen av kravene som konflikter med hverandre?

Kravene må kunne sjekkes om de er løst eller ikke,

Hvor kommer kravet fra?/ Hvorfor er det kravet her?

Kan man endre litt på kravene?

Dekker kravlisten alle krav? Vil den løse problemet?

Er det noen krav som ikke er så viktige?

Har vi tid og penger til kravet?

Krav skal ikke ha spesielle løsninger.

* Hva er en kravdefinisjon, og hva bruker vi den til

Skal være forstås bart av alle, oversiktig og uformell. Krav definisjon deles opp i to deler, Funksjonelle krav og ikke-funksjonelle krav. Funksjonelle krav beskriver hva systemet skal gjøre og hva den ikke skal gjøre, mens ikke-funksjonelle krav er info om systemet, som responstid, brukere og ressursforbruk. Det er mer tall krav.

Kravdefinisjon er viktig data og trengs når du skal formulere løsningen din.

* Hvorfor kan det være at vi må vurdere og gjennomføre endringer på krav etter at utviklingen har startet

Det kan hende at noe skjer som gjør at krav endres, kanskje klient plutselig skal ha 10 ganger så mange aktive brukere eller at budsjettet deres ble betydelig lavere. Kanskje noen av kravene ikke kan løses likevel eller at det viser seg at noen av kravene var feil eller konflikter med andre krav

## Funksjonelle krav

Beskriv hva som kjennetegner funksjonelle krav

Funksjonelle krav er hva system skal kunne gjøre og hva det ikke skal kunne gjøre

## Ikke-funksjonelle krav

Beskriv hva som kjennetegner ikke-funksjonelle krav

Ikke-Funksjonelle krav er om hvordan systemet skal være effektivt og lett og bruke

## Design

Hva vil det si å designe en løsning.

Å designe en løsning betyr og begynne og planlegge hvordan systemet skal bygges, ikke hvordan det skal gjennomføres, men hva selve løsningen skal være og se ut.

Beskriv noen metoder for å designe løsninger og forskjellen mellom de.

Fossefall. Du planlegger først også utfører du det. endringer blir vanskelig og kommunikasjon er vanskelig.

Tidsboksing. Du planlegger hvor mye tid du skal jobbe på en oppgave istedenfor å jobbe på det til det blir ferdig. Høy prioritet på og bli ferdig innen tid gir dårligere produkt.

Flytbaserte. Samme som tidsboks men når noen blir ferdig går de videre. Prosjekt blir raskere ferdig, kommunikasjon og endringer blir vanskeligere.

# Dokumentasjon

## Hvilke tre dokumentasjonstyper bruker vi til IKT systemer

System dokmentasjon, Bruker dokumentasjon og drift dokumentasjon

## Når brukes de

**System dokumentasjon.** Primært for utviklere med også en viss grad for drift, det blir brukt til forsvarlig vedlikehold og for å utvikle det videre. Det blir også enklere og bruke det.

**Bruker dokumentasjon.** Lar utviklere gjøre det enklere for brukere og bruke systemet. Brukere kan også benytte det

**Drift dokumentasjon**. Gjør vedlikehold av system enklere, lar it drift personell vedlikeholde systemene bedre

## Hva skiller de fra hverandre

**System dokumentasjon** er dokumentasjon av systemet mens **bruker dokumentasjon** er dokumentasjon om brukeren osv.

## Hvilken informasjon inneholder de

**System dokumentasjon.** Dokumentasjon av systemet og hvordan det fungerer. Hvem som eier det og er ansvarlig

**Bruker dokumentasjon.** Dokumentasjon av brukerene og hva de gjør.

**Drift dokumentasjon**. Dokumentasjon av drifting av systemet og vedlikehold

# Databaser

## Hva er forskjellen på relasjons og dokumentdatabaser

Relasjons databaser lagrer informasjon i kolonner, tabeller og rader. Du kan søke opp navnet til en person og du kan finne relatert info om den personen i tillegg.

Dokument databaser putter all informasjon i dokumenter som da blir puttet i mapper. Mappene kan fortelle deg om informasjon inni, men infoen og spesifik informasjon har man ikke tilgang til før man åpner det opp.

## Hva brukes de to best til

Relasjons databaser er veldig bra når dataen er relatert med andre data og når du trenger tilgang til spesifikk informasjon enkelt fra forskjellige tabeller.

Dokument databaser er bra hvis du trenger å finne dokumenter som inneholder informasjonen du trenger.

**Relasjonsdatabaser**

## Hva er en tabell

En tabell er en oversikt av informasjon og data delt opp i rader og kolonner.

## Hva er en post

En rad kan bli kalt mange forskjellige ting, et av dem er en post. En post er noe du da finner i tabeller som organiserer informasjon og inneholder data.

## Hva brukes relasjoner i en relasjonsdatabase til

Til å finne relevant informasjon, hvis du f,eks vil ha tlf nummeret til ThorArne kan man søke opp ThorArne og også finne telefon nummeret hans siden det er relatert info

**Dokumentdatabaser**

## Hva brukes type/ key feltet til i en dokumentdatabase

Det er en data lagrings metode. Det funker som dictionaries hvor du har en type også har du keyen til den.

# Testing og Debugging

## Hva er testing og hva er debugging

Når du tester, finner du ut om programmet utfører funksjonen sin og at det ikke er noen feil.

Å fikse feilene du finner i testing er å debugge, du analyserer feilen også fikser du det.

## Beskriv de viktigste forskjellene mellom testing og debugging

Under testing så finner du feilene, ikke fikser det. Når du starter debugging så finner du ut hva som er feil hvordan du kan fikse det

# Versjonshåndtering

## Beskriv hva GIT er og hva det brukes til

GIT er et type version control system, du får en historie over hva som har skjedd med systemet og du kan når som helst gå tilbake til en gamlere versjon før noen gjorde en endring. Det gjør det også enklere for store og små prosjekter sånn at alle kan jobbe på samme nyeste versjon av koden istedenfor å måtte sende den nyeste koden frem og tilbake.

## Forklar hvorfor vi bruker GitHub

På Github kan passe på og håndtere GIT.

## Hva er et repository

Git repositorien er hvor alle endringene blir lagret, det er historien til prosjektet.

# Informasjonssikkerhet

## Beskriv de tre sikkerhetsmålene i praksis (bruk egne ord)

Konfidensialitet, det må sikres så bare de som trenger å bruke informasjonen får tilgang

Integritet, ressurser skal ikke endres eller slettes med uhell

Tilgjengelighet, det skal ikke være lett tilgang, men når noen har behov må de kunne ha tilgang

## Innenfor sikkerhet i IKT systemer, beskriv forskjellen på sikkerhet, informasjonssikkerhet og cyber sikkerhet

Sikkerhet å beskytte mot skade, ødeleggelse og tenke på problemer som kan oppstå som kan true verdier.

Mens Cyber sikkerhet handler om å holde unna hackere og virus som truet sårbare IKT systemer.

Informasjonssikkerhet er at du må passe på og beskytte informasjon, f,eks kryptere informasjon og være forsiktig med hvem du sender til og via hva. IT utstyr må renses helt når det resirkuleres osv.

## Når du designer og koder et ikt system, hva er det grunnleggende må du tenke på når det gjelder sikkerhet

Tenke på hvilke implikasjoner løsninger kan ha på sikkerhet.

Du må følge de grunnleggende retningslinjene og reglene for programmeringsspråk og programvare typer.

Kode som er sikker kan gjenbrukes

God kvalitets kode som er lesbar og vedlikehold bar. (bra dokumentasjon er viktig)

# Secure Coding

## Hva betyr Prinsippet om minste privilegium

Det er viktig at moduler bare har tilgang til informasjon og ressurser til det som er nødvendig. Brukerstøtte trenger for eksempel ikke tilgang til server derfor gir vi det ikke.

## Hvorfor programmerer vi sikkerhet i dybden?

Siden sikkerhet er viktig og det blir utrolig mye mer sikkert da.

## Hva er Penetration testing og hva bruker vi det til

Du prøver og bryte deg inn i ditt eget system via hacking. Da får du vite svakhetene og kan fikse det

## Hva betyr det å planlegge for sikkerhetsbrudd (fail secure)

Hvis et sikkerhets brudd skjer så har du midler klart for å sikre informasjonen

## Hva betyr det å bruke åpen design

Å designe åpent er å gjøre prosjektet og dokumentasjon offentlig tilgjengelig, du jobber sammen med andre og bruker andre offentlige ting for å gjøre koden din bedre