

Grundbog til informatik-faget

Lavet i samarbejde med Center for Computational Thinking and Design (CCTD)

I denne grundbog, vil der være en række undervisningsforløb samt supplerende materiale til Informatik-faget. Den første del af grundbogen, omhandler de følgende fire emner:

- Brainstorm.....2
- Prototyping.....6
- Brugervenlighedstest - Tænke Højt Test.....9
- Innovation.....13

Herefter komme en række emner og temaer der kan anvendes som støtte til informatikundervisningen:

Domænekendskab - vidensindsamling i forhold til et nyt produkt

- Domænekendskab.....18
- Arbejdsopgaver, domænekendskab.....21
- Et problem, et dilemma, et behov (behovskonflikt, problemløsning).....24
- Opbygning af domænekendskab.....26
- Rige billeder - en metode til opbygning af domænekendskab.....30
- Målgruppe.....32

Dokumenter i udviklingsprocessen

- Kravspecifikation.....34
- Testspecifikation.....36
- Use Case.....38

Interaktionsdesign

- Gestaltlove (fra leksikon).....40
- Arkitektur, applikationsarkitektur.....41
- Teoretisk note, interaktionsdesign.....Bilag 1

Datasamlinger

- Dataanalyse og simpel MySQL database lavet i phpMyAdmin.....44

Brainstorm

Introduktion til Brainstorm - Hvad er brainstorm?

Brainstorm er en metode til idégenerering. Brainstorm gennemføres som pararbejde eller i grupper. Brainstormmetoden anvendes til at udvikle og fastholde ideer og teknikken anvendes mange steder i udviklingsprocessen i informatik - både når der udvikles nye produkter og når eksisterende produkter/systemer udvides. Der er mange variationer af brainstorming. I det efterfølgende anvendes en metode med post-it notes og gruppering af disse. En metode der fungerer fint i undervisningssammenhæng. Undervisningstimen indledes med en kort præsentation af et dilemma, hvorefter eleverne laver en individuel brainstorm. Herefter præsenterer eleven sine post-it notes for en gruppe. En session der gentages to, måske tre gange.

Her er lidt baggrundsmateriale om klassisk brainstorm: <https://innovation.sites.ku.dk/metode/klassisk-brainstormi/>

Relevans

Brainstorm er en af grundelementerne for innovation.

Forudsætninger

Alle kan deltage i en brainstorm session - uden forudsætninger. Inden man går i gang skal der være et dilemma eller et spørgsmål der brainstormes over. Fx: "Hvordan laves en spændende hjemmeside der tiltrækker kunder?"

Relevans for forløb

Brainstorm anvendes i forløbet Malkebedriften Kløvergården og i Pixi-forløbet Hobbylandbruget Kløverhytten

Relevans for kernestof

Brainstorm går fint i spænd med mange af de faglige mål, idet der ofte foretages et idégenererings-forløb inden produkterne/systemerne udvikles. Faglige mål:

- give eksempler på hvordan it-systemer har betydning for og påvirker menneskelig aktivitet
- redegøre for udvalgte elementer i et interaktionsdesign, samt realisere udvalgte interaktionsdesign i et konkret it-system og tilpasse eksisterende design og systemer
- redegøre for innovative it-systemer sammenholdt med egne udviklede it-systemer

Eksempel på eksamensspørgsmål

Et eksamensspørgsmål kan fint indeholde en passus som:

- Din præsentation skal indeholde ... idéudvikling med brainstorm-metoden.
- ...forklar hvorledes projektgruppen anvendte brainstorm-metoden til idéudvikling.

Refleksion of tips fra andre undervisere

I en undervisningsituation hvor eleverne brainstormer, skal man som underviser give sig god tid og "go with the flow". Indtil eleverne er fortrolige med metoden, er det processen der er vigtig - ikke produktet.

Er der opstået en god gruppedynamik i en gruppe, er det vigtigere at lade eleverne "køre der ud af" end at følge modellen for brainstormforløbet. Formålet er ide-generering. Det er ofte en fordel at begynde den efterfølgende lektion med at reflektere over processen (med eleverne) og i den forbindelse kan man samle processen op.

Det er svært at måle om eleverne har lært teknikken. Det er et praktisk emne og det er først til eksamen eleverne giver sig tid til (ordentligt) at reflektere over metoden. Så mind eleverne om at tage billeder af processen undervejs (til deres portfolio-mappe).

Lektionsplan (én 60 min lektion)

15 min. Introduktion til brainstorm

Eleverne skal forstå hvad processen i en brainstorm-session er. De fleste elever har prøvet brainstorm før, så ofte er det nok lige at gentage trinnene i et brainstormforløb for eleverne. Det kan gøres ved at læse nedenstående højt for klassen. Alternativt kan man lave en Power Point med nøgleord og gennemgå den med eleverne.

Et typisk brainstorm forløb:

- En brainstormsession begynder med at dilemma eller et problem præsenteres for gruppen.
- Alle deltagere gives en 3M post-it blok som forskellige idéer skrives ned på - én idé pr. seddel.
- Der begyndes med en individuel brainstorm. Man må ikke snakke sammen. Hver deltager tegner eller skriver idéer til et produkt der løser det opstillede dilemma/problem. Produktet kan være en app, en hjemmeside eller noget helt tredje.
- Den individuelle brainstorm foregår i tre minutter, hvilket føles som lang tid - smartphone og PC er forbudt imens man arbejder.
- Når de tre minutter er gået, præsenterer hver enkelt deltager sine tre bedste idéer for de øvrige gruppemedlemmer. Hver idé forklares kort og klistres på et bord eller en tavle. Det er vigtigt at ingen kritiserer den idé der fremlægges - det kan være at en umiddelbart vandvittig idé, inspirerer en anden deltager.



Idéer skitseres eller skrives ned (Foto: Ken Mathiasen)



Ideerne grupperes løbende - hver gruppe kan navngives (Foto: Ken Mathiasen)

- Hvis en idé minder om nogle af de andre idéer der er gennemgået, klistres den på tavlen i nærheden af dem den minder om. Formålet med at præsentere ideerne for gruppen er, at trigge nye idéer på baggrund af de eksisterende forslag.
- Til sidst navngives de enkelte grupper af idéer.

20 min. Første runde (eleverne skal i gang)

Efter brainstorm-konceptet er forklaret, skal eleverne selv i gang. Det er vigtigt at alle forstår hvilket dilemma/problem der arbejdes med at finde en løsning til. Eleverne gennemfører nu en individuel brainstorm (ca. tre minutter).

Herefter præsenterer eleverne deres idéer for de andre i gruppen. En efter en. Når en idé er præsenteret, klistres den sammen med en tilsvarende idé.

Her er man som underviser passiv og går rundt og iagttager elevernes arbejde.

- Hvis det går trægt, kan man hjælpe ved at gentage dilemmaet/problem.
- Det kan også hjælpe med let provokerende spørgsmål eller "Rasmus modsat"-spørgsmål.

20 min. Anden runde af Brainstorm session

Nu har hver projektgruppe et antal idéer på tavlen - forhåbentlig er idéerne grupperet og navngivet.

Underviseren forklarer det videre forløb for eleverne:

- Nu skal projektgruppen diskutere produktmuligheder i de forskellige idé-grupper.
- Når eleverne har snakket om de produkter der kan laves, udvælges en idé-gruppe man arbejder videre med.
- Der gennemføres endnu en runde individuel brainstorm - i tre minutter. Denne gang med produktidéer til den udvalgte idé-gruppe.
- Efter den individuelle brainstorm præsenterer hver elev sine sedler på en tavle og igen samles ens produktkoncepter og idéer i grupper.

Klassen sættes i gang med anden runde. Igen er det underviserens rolle at liste rundt og lytte. Er der en god gruppedynamik går man stille videre. Er en gruppe gået i stå, hjælper man dem lige i gang - her kan man som underviser stille provokerende spørgsmål eller tage et åbenlyst fejlagtigt standpunkt og få eleverne til at argumentere imod det.

Under projektgruppens gennemgangen af idéerne fra anden runde, opstår der ofte præference for en enkelt gruppe af idéer. Man fornemmer at der er et eller andet spændende/smart ved idéerne i gruppen og dermed er projektgruppen færdig med deres brainstorm session og klar til næste fase i produktudviklingen. Nogle gange er der behov for yderligere en brainstorm-session.

Brainstorm-sessionen afsluttes, når der er konsensus i projektgruppen om en skitse til en løsning af dilemmaet/problem.



5 min. Afslutning

Opsamling på klassen. Er der tid til det, kan man spørge til de produkt-idéer der er fremkommet. Ellers er det altid godt at slutte med at repetere lektionens forløb. Det kan gøres ved at bruge et minut eller to på at høre hvad eleverne synes var det vanskeligste og det bedste ved Brainstorming.

Ros ros ros - husk at rose eleverne for deres indsats.

- Man må ikke lyve - find noget de kan roses for ("Det var nogle kreative forslag I kom frem til" - "Sikke mange idéer I fandt på" - "I sidste halvdel af timen arbejdede I godt sammen" - osv).

Prototyping

Introduktion til Prototyping - Hvad er prototyping?

Prototyping bruges til at sikre udvikleren en forståelse af brugerens krav og forventninger til systemet. Der er som oftest tale om relativt simple prototyper, f.eks. i form af skitser, rumlige modeller, eller click-through prototyper baseret på skitser. Fælles for prototyperne er, at selvom egenskaber ved systemet, designelementer, koncepter ofte går igen i det færdige system, så indgår prototyperne ikke som en del af det færdige system.

Her er noget materiale om dels motivation for rapid prototyping, dels om click-through prototyping: <https://www.smashingmagazine.com/2014/03/building-clickthrough-prototypes-to-support-participatory-design/>

På dansk kan man læse flg. mere generelle artikel om interaktionsdesign: <http://informationsteknologi.wdfiles.com/local-files/interaktionsdesign/Interaktionsdesign.pdf>

Relevans

Prototyping er relevant for innovation.

Forudsætninger

Alle kan medvirke til prototyping - uden forudsætninger. Inden man går i gang skal der være en ide om et program eller et system, eller dele heraf, som der skal laves en (eller flere) prototyper til.

Relevans for forløb

Prototyping anvendes i forløbet

- [Malkebedriften Kløvergården](#)
- [Anderledes Interaktion](#)

Relevans for kernestof

Prototyping går fint i spænd med flere af de faglige mål, idet der ofte arbejdes med interaktionsdesign inden produkterne/systemerne endeligt udvikles. Faglige mål (den relevante del er skrevet med kursiv):

- give eksempler på, hvordan it-systemer har betydning for og påvirker menneskelige aktiviteter samt identificere, hvordan konkrete it-systemer påvirker brugeres adfærd
- beskrive og analysere udvalgte elementer i et interaktionsdesign, udarbejde it-produkter i form af simple interaktionsdesign og tilpasse eksisterende design

Eksempel på eksamensspørgsmål

Et eksamensspørgsmål kan fint indeholde en passus som:

- Din præsentation skal indeholde ...skitser eller andre prototyper.
- ...forklar hvorledes projektgruppen anvendte prototyping til udvikling.

Refleksion og tips fra andre undervisere

Konceptet med at "smide ud" (throw away prototyping) er ofte svært for eleverne. Vær parat til at hjælpe dem med at forstå, at de hver gang beholder det bedste fra deres prototyper, eller i hvert fald lærer noget værdifuldt om deres ide.

For nogle elever er det svært at holde sig til simple prototyper (rapid prototyping), det kan hjælpe at bede dem f.eks. tegne skitser med 6 pr A4 side, så der ikke er plads til så mange detaljer. Eller at køre nogle meget korte sekvenser, f.eks. diskuter og tegn i 5 min og præsenter så for klassen.

Tænk over materialevalg på forhånd - skal eleverne "bare" tegne, eller skal der bygges fysiske prototyper, hvem medbringer hvad? De første prototyper er måske tegninger, som så kan ligge til grund for mere avancerede fysiske prototyper, med materialer, der skal medbringes hjemmefra, eller findes andet sted på skolen.

Lektionsplan

Prototyping læres bedst gennem et konkret projekt. Det betyder også, at ikke vi ofte ikke kommer omkring alle former for prototyping i ét projekt. Præsenter de metoder, der giver mening i jeres projekt. F.eks. vil fysiske prototyper ofte være en del af slutproduktet i et projekt med Makey Makey.

Lektion 1(-2): Interaktionsdesign

Gennemgå noten om Interaktionsdesign. Det virker bedst, hvis (1) eleverne har læst noten på forhånd og (2) den relateres til et igangværende projekt (f.eks. hjemmeside, app eller spil). Lad eleverne arbejde med øvelserne. Hvis I har arbejdet med interaktionsdesign før, så lav kun de øvelser, der passer til jeres nuværende projekt.

Lektion 2: Papir prototyper

Arbejd med throw away prototyping - brug korte sekvenser med præsentationer/feedback for klassen ind i mellem.

Simple skitser kan give et hurtigt overblik over f.eks. hvordan et program bruges, og specielt hvordan de tilgås. Det er vigtigt netop at have fokus på at holde skitserne simple – for et spil tegner man måske tændstikmænd, selvom man senere regner med at have billeder af mennesker i selve spillet. Man kan bruge skitserne interaktivt – her har man et billede pr side og skifter til andre sider afhængigt af brugerens input. Man kan også bruge storyboards, som er en slags tegneserie over gangen i et system – her har man flere skærbilleder på en side, så de læses sammen som en tegneserie.

Sekvenseringen kan f.eks. for et spil være (efter man har ideen til spillet): 4 min forside 1 min (pr gruppe) til præsentation og hurtig feedback 1 min ret til på forsiden (evt. større ændringer gemmes til senere) 7 min overordnet layout og spillets gang 1 min (pr gruppe) til præsentation og hurtig feedback 1 min ret til 3 min den direkte interaktion (taster, mus, særlige tilfælde) 1 min (pr gruppe) til præsentation og hurtig feedback.

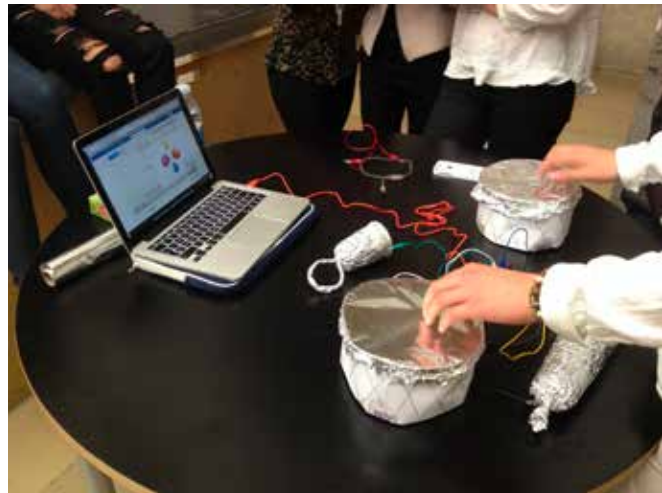


Eksempel på storyboard (Foto: HG)

Lektion 3: Click-through prototyper/fysiske prototyper

Nu kan man enten bygge fysiske prototyper eller videreudvikle sine papirprototyper.

Nogle systemer egner sig bedre til andre inputformer end tastatur og/eller mus. Eller brugeren kræver af andre grunde dette. Man kan arbejde med design af sådanne alternative inputmetoder vha. mockups. En mockup er en simpel fysisk model af, typisk, et interface. Det kan bruges til at eksperimentere med og teste den fysiske opbygning, udseende og taktile egenskaber for systemet, og kan bygges af simple materialer som pap, tape, stanniol, skumgummi osv.



Mockups ad inputmetoder, der ikke er tastatur- eller musebaserede.

En Click-through prototype er en prototype, der kører på en computer/anden device. Den er stadig skitse-baseret, men skitserne gøres eksekverbare, i hvert fald på et overordnet plan. Der er overordnet to metoder til at lave click-through prototyper: med billeder af skitser lave i hånden (på papir, whiteboard eller lign) eller skitser lave direkte på deviceen. I begge tilfælde fungerer click-through prototypen ved at man har en række skærbilleder, og forskellige områder af et billede defineres som "knapper", der skifter videre til et af den andre skærbilleder. Click-through prototyper er gode til at undersøge om arbejdsgangen i systemet (i prototypen) er passende iflg. brugeren, idet man hurtigt vil opdage, hvis brugeren finder konsekvenserne af sine valg uventede. Man kan nemt lave device-baserede prototyper med f.eks. POP (app store, google play eller windows store: <https://popapp.in/>).

Lektion 4: Testing

Lad eleverne teste hinandens prototyper og give feedback. Det kan f.eks. gøres ved at en elev fra hver gruppe besøger en anden gruppe og udfører en såkaldt "tænke-højt-test", hvortil der tages noter. Afhængig af projektet kan det også være en ide med testere/brugere ude fra.

Brugervenlighed - Tænke højt test

Introduktion til Tænke højt test

Resumé af Tænke Højt Testen

“Tænke Højt Testen” er en kvalitativ brugervenlighedstest. Tænke Højt Testen kan anvendes til at afprøve et produkt, fx en hjemmeside eller et brugerinterface i en app.

Henvisning til yderligere materiale om Tænke Højt Testen

Kommunikationsforum.dk beskriver Tænke Højt Testen: [Tænke Højt Testen forklaret for dummes](#)
I bogen [Informationsteknologi fra Systime](#) står der en masse godt om Tænke Højt Testen
På EMU'en er der en god side om Tænke Højt Testen - inklusiv en opvarmningsopgave
“kuglepensøvelsen” [EMU'en: Introduktion til brugertest](#)

Relevans

Tænke-højt-testen er naturlig at anvende i forbindelse med interaktionsdesign.

Forudsætninger

Testlederen bør være en person, der tidligere har deltaget i en tænke-højt-test eller i det mindste har sat sig godt ind i rollen som testdeltager. Yderligere kan følgende afsnit fra Informationsteknologi, Systime læses.

Relevans for forløb

Tænke-højt-testen er velintegreret i forløbet [Brugervenlighedstest](#)

Relevans for faglige mål

Tænke-højt-testen anvendes primært i forhold til design og test af brugerinterface og indgår derfor i følgende faglige mål:

- Konstruktion af it-system som løsning til et problemområde
 - Eleven skal kunne løse et problem ved at beskrive problemet, samt designe, realisere og afprøve it-systemet gennem brugerorienterede teknikker
- It-systemers og menneskelig aktivitets gensidige påvirkning
 - Eleven skal kunne give eksempel på, hvordan it-systemer har betydning for og påvirker menneskelige aktiviteter
- It-sikkerhed, netværk og arkitektur
 - Eleven skal kunne redegøre for beskyttelse af egen digital identitet og egne data på internettet samt redegøre for tekniske og menneskelige aspekter af it-sikkerhed
- Interaktionsdesign
 - Eleven skal kunne redegøre for udvalgte elementer i et interaktionsdesign, samt realisere udvalgte interaktionsdesign i et konkret it-system og tilpasse eksisterende design og systemer.

Eksempel på eksamensspørgsmål

Et eksamensspørgsmål kan fint indeholde en passus som:

- Der skal gennemføres en brugervenlighedstest med metoden Tænke-højt-test
- ...forklar hvorledes en Tænke-højt-test kan anvendes til at forbedre brugerfladen og hvilke fordele/ulemper der er forbundet hermed.

Refleksion of tips fra andre undervisere

Afhængigt af, hvor meget man vil lægge vægt på begreberne, kan man teste elevernes viden i en Kahoot-test. Denne kan også anvendes formativt, således at der testes gentagne gange.

Lektionsplan (to lektioner á 60 min.)

Første Lektion

10 min Introduktion til Brugervenlighed

Underviseren indleder lektionen med at fortælle om nogle gode og dårlige oplevelser han/hun har haft med brugervenlighed. Det kan være der er en dør der altid lukker op den forkerte vej eller en kande der spilder når man hælder med den. Instrumentpanelet i en bil har ofte en god brugervenlighed (vi kan anvende det uden instruktion), imens det modsat ofte er vanskeligt, at finde en bestemt kanal på bilradioen.

Snak med eleverne om deres oplevelser med ting der bøvler.

Lad eleverne søge på nettet og se om de kan finde nogle interessante eksempler (fx. kan de Google "user interface fail")

5 min. Introduktion til Tænke Højt Testen

- Fortæl kort fortæller om brugertest og om Tænke Højt Testen. I den forbindelse virker eksempler godt, så efter introduktionen kan man tænde sin PC imens man sætter ord på sine handlinger.
- *Jeg tager computeren op af min taske - jeg vender den så låget kan åbnes - jeg åbner låget - jeg identificerer tænd knappen - jeg trykker på tænd knappen - jeg kigger på skærmen (den er sort) - nu kommer der billede på - jeg venter - nu kommer log-on billedet op - osv.*

20 min. Opstartsøvelse

- Gruppearbejde 2-3 elever i gruppen (kan også laves som par-arbejde)

Tips: Underviseren medbringer kuglepenne til hele klassen - det skal være en type der kan skilles ad.

- En kuglepen per gruppe
- Brugeren får udleveret kuglepennen
- Testlederen instruerer brugeren i at adskille kuglepennen og samle den igen.
- Brugeren skiller nu kuglepennen ad, imens han/hun hele tiden fortæller om de tanker han/hun har



- Undervejs nedskriver testleder de handlinger brugeren foretager
- Testlederen må gerne stille guidende spørgsmål til brugeren
- Rollerne byttes og øvelsen gennemføres på ny

5 min. opsamling på klassen

- erfaringsudvikling og refleksion
- eksempel på spørgsmål der kan diskuteres i plenum (det er godt at skrive stikord på tavlen imens): hvad var det vanskelige ved at være testleder - hvorfor var det svært for testdeltageren at sætte ord på sine handlinger - hvilke muligheder er der for at forbedre testen - hvad var forskellen til "tænd lærerens PC øvelse" - hvad kan der være af problemer hvis det er en hjemmeside eller en app der skal testes

20 min. Eleverne arbejder med brugeroplevelser

Nu skal eleverne i gang med hjemmesider. De skal undersøge [webshoppen bodylab.dk](http://webshoppen.bodylab.dk)
Dette "arbejde selv" dokument udleveres [til eleverne](#) ([her er opgaven med løsningsforslag](#)).

Anden Lektion

5 min. Opsamling på sidste gang

- Snak med eleverne om deres oplevelser og erkendelser fra sidste time

10 min. Introduktion til Tænk Højt Testen når den anvendes til hjemmesider/apps/spil

- Efter eleverne sidste gang arbejdede med bodylab.dk, gennemgås nu denne præsentation på storskærm

15 min. Eleverne forbereder en Tænk Højt Test

- Eleverne arbejder med en hjemmeside der fungerer dårligt.

Tips: Det er svært at finde en dårlig side (en side med dårlig funktionalitet) - derfor er det en fordel at læreren inden timen har fundet et par, fx:

- [Irish Wreck Online](#)
- [Flensborg havnerundfart](#)
- Eleverne skal udarbejde en spørgeguide til en hjemmeside. Guiden skal indeholde:
 - Kerneområder der skal testes
 - Testscenarier, der sætter testdeltageren ind i en situation, som får testen til at virke plausibel
 - Testspørgsmål, som kommer i forlængelse af hvert testscenarie (fx "hvad koster en billet til Flensburg-Sonwik færge-overfarten" (fra havnerundfarthjemmesiden) - eller "Find kortet over Irland" (fra Wreck Online))

20 min. Eleverne gennemfører en Tænk Højt Test.

- Hver gruppe udleverer et medlem til nabogruppen (og modtager til gengæld et medlem fra en



anden gruppe)

- Det er en fordel at bruge en skæroptager (med lyd) til at dokumentere testene (fx [Screencast-o-matic](#))

5 min. Afslutning og opsamling på klassen

- For at fastholde den viden eleverne har opbygget er det godt at foretage en fælles refleksion
- Her er et par eksempler på spørgsmål der kan diskuteres i plenum: hvad var det vanskelige ved at være gennemføre testen - hvad var testleders største udfordring - hvad skulle man ændre for at gøre testen bedre for testdeltageren

Ros ros ros - husk at rose eleverne for deres indsats.

Innovation

Introduktion til Innovation - Hvad er Innovation?

Der findes mange beskrivelser af innovative produkter og IT-systemer, f.eks.

- "Noget nyt og brugbart, der kan implementeres"
- "Processen, hvormed en ide eller opfindelse overføres til et product eller en service, der giver værdi, som en kunde er villig til at betale for"
- "Innovation er en idé, der omsættes til praksis og skaber værdi"
- "Innovation er en succesfuld implementering af en ny idé"

For at et produkt eller et IT-system kan anses for innovativt, skal det altså være brugbart og nyt - det kan godt være en ny anvendelse af noget, der eksisterer i forvejen. Det skal være noget, der tilføjer værdi for nogen.

Relevans

Innovation er relevant for både udvikling af nye systemer og diskussion af eksisterende systemer.

Forudsætninger

Alle kan medvirke til innovation - uden forudsætninger, samt bruge simple modeller til at diskutere hvorvidt og hvordan systemer er innovative.

Relevans for forløb

Innovation anvendes i forløbet

App-udvikling <https://library.ct-denmark.org/app/>

Reklamespil <https://library.ct-denmark.org/reklamespil/>

Relevans for kernestof

Innovation er beskrevet i selvstændige faglige mål i læreplanerne for informatik:

- C/B: eksempler på og kategorisering af innovative it-systemer.

Desuden skal elevernes "innovative kompetencer" udvikles (se "arbejdsmetoder" i læreplanerne for både C- og B-niveau).

Eksempel på eksamensspørgsmål

Et eksamensspørgsmål kan fint indeholde en passus som:

- Din præsentation skal indeholde en sammenligning af dit udkast med et innovativt IT-system.
- Forklar hvorledes projektgruppen anvendte innovative arbejdsmetoder til at udvikle ...

Refleksion og tips fra andre undervisere

4p-modellen: I enkelte illustrationer vender inkrementel vs radikal omvendt af hvordan det er vist her. Det er dog mest intuitivt, at den stærkeste innovation er yderst i cirklen, og det er også den originale illustration af 4p-modellen.

SAMR-modellen: Modellen er udviklet specielt mht. IT-systemer, og er derfor måske specielt interessant her.

Innovative Arbejdsmetoder: Vi kender alle sammen Georg Gearløs og hans tænkehat. Nu har vi jo ikke selv nogen tænkehat, men der er forskellige metoder, vi kan bruge, når vi gerne vil forsøge at finde på noget, der kan beskrives som innovativt. Det er meget forskelligt, hvordan man får ideer: Nogle har brug for hjælp til at skabe rum for gode ideer, andre har brug for hjælp til at strukturere iderigdommen lidt. Det er vigtigt at vide, at kreativitet ikke er et krav - man kan arbejde systematisk med at skabe nye ideer. Der er mange, mange muligheder her – vælg noget, der passer til jer og det, I arbejder med.

Lektionsplan

Man kan godt arbejde med innovation som et selvstændigt (kort) forløb, men det er nok mest effektivt, hvis det hægtes op på et konkret projekt, eleverne skal lave. Det kan fint gennemarbejdes ud fra et rent design projekt, man behøver således ikke implementere noget for at kunne gennemgå materialet om innovation til fulde.

Lektion 1-2:

Hvad er innovation?

- "Noget, der er nyt, en forbedring og er brugbart"
- "Noget nyt og brugbart, der kan implementeres"
- "Processen, hvormed en ide eller opfindelse overføres til et produkt eller en service, der giver værdi, som en kunde er villig til at betale for"
- "Innovation er en idé, der omsættes til praksis og skaber værdi"

For at et produkt eller et IT-system kan anses for innovativt, skal det altså være brugbart og nyt - det kan godt være en ny anvendelse af noget, der eksisterer i forvejen. Det skal være noget, der tilfører værdi for nogen.

Hvordan kategoriseres innovative systemer?

Der er flere forskellige modeller for, hvordan man kan kategorisere innovative it-systemer. Vi vil her se på flg.:

- Inkrementel vs radikal innovation
- 4p-modellen
- SAMR-modellen

Man taler også om "disruptive" innovation, dvs. innovation, der forstyrrer konkurrenter.

Når man har lært at programmere, ved man noget om forskellige programmeringsstrukturer, f.eks.

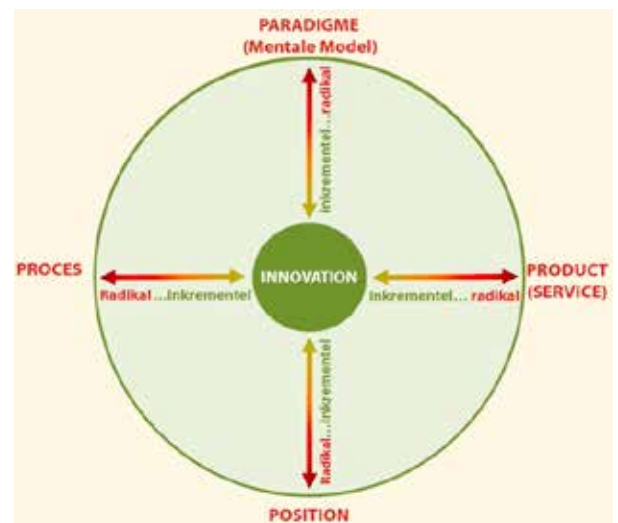
løkker og forgreninger. Man kan tale om programmer generelt. Vi har brug for at kunne snakke om, hvorvidt og på hvilken måde systemer er innovative.

Inkrementel vs radikal innovation

- Inkrementel innovation – ”gør noget bedre”. Som eksempel kan nævnes et samlet elektronisk ugebrev i stedet for at sende mange korte beskeder i løbet af ugen.
- Radikal innovation – ”gør noget anderledes”. Som eksempel kan nævnes Skype som alternativ til fastnet telefoni.

4p-modellen

4p-modellen blev introduceret af Bessant og Tidd. Modellen siger, at innovation kan pege i 4 retninger: Produkt, Proces, Position og Paradigme. Modellen illustreres som vist i figur 1. Bemærk, at inkrementel innovation er tættest på centrum, og radikal innovation længst væk fra centrum. Vi ser, at innovation ikke typisk er enten inkrementel eller radikal, men et sted derimellem. En videopræsentation af modellen kan ses her: <http://www.forklarmiglige.dk/de-4-p-i-innovation-john-bessant-og-joe-tidd> (ca 6 min).



Figuren viser 4p-modellens runde skive

Eksempler på brug af 4p-modellen:

- Produkt - forandring af produkt/service: bannerreklamer (forskellige produkter) vs dedikeret spil, som reklamerer for ét produkt.
- Proces - forandring i proces og/eller levering: fysisk levering (disk) vs download vs app store.
- Position - forandring af hvordan og til hvem produktet introduceres/målrettes: dedikerede reklamespil (børn/unge) vs reklamer i spil (unge/voksne).
- Paradigme - forandring af mentale model af egen organisation: radikalt - reklamespil kommercielt interessant som middel til brugertracking, mere end som reklame.

Øvelse: giv konkrete eksempler på innovation, og placer systemerne/produkterne i 4p-modellen. Kan du ramme alle 4 p'er?

SAMR

SAMR står for

- Substitution - et IT-system erstatter et eksisterende system, men uden at ændre arbejdsgangen, eller tilføje ny værdi.
- Augmentation - et IT-system erstatter et eksisterende system, og tilføjer noget nyt/forbedrer arbejdsgangen.
- Modification - et IT-system giver adgang til væsentlige ændringer i arbejdsgangen.
- Redefinition - et IT-system leder til nye anvendelser, der tidligere var utænkelige.

Hvis vi ser på IT i undervisningen som et eksempel, så kunne det f.eks. se således ud:

- Substitution - et arbejdsark distribueres som en word-fil, eleverne udfylder arket. Arbejdsgangen er ikke væsentligt ændret i forhold til at have arbejdsarket på papir.
- Augmentation - arbejdsarket indeholder (for eleverne ukendte) links til materiale, der er relevant for opgaven. Her er arbejdsgangen ændret, da læreren ellers skulle kopiere større mængde materiale, eller eleverne måske ikke ville have adgang til det.
- Modification - arbejdsarket er interaktivt, med links til materiale, opgavesessioner, osv. Læreren har direkte adgang til at se den enkelte elevs progression, også under arbejdet.
- Redefinition - arbejdsarket indeholder selv-tests, så eleven selv har mulighed for at måle sin egen forståelse undervejs i arbejdet.

Et sæt slides, der præsenterer modellen fås i <http://www.hippasus.com/rrpweblog/archives/2013/10/25/SAMRAContextualizedIntroduction.pdf> Og flere detaljer kan ses på hjemmesiden <http://www.hippasus.com/rrpweblog/>

Øvelse: giv konkrete eksempler på innovation, og placer systemerne/produkterne i SAMR-modellen. Kan du ramme alle 4 niveauer?

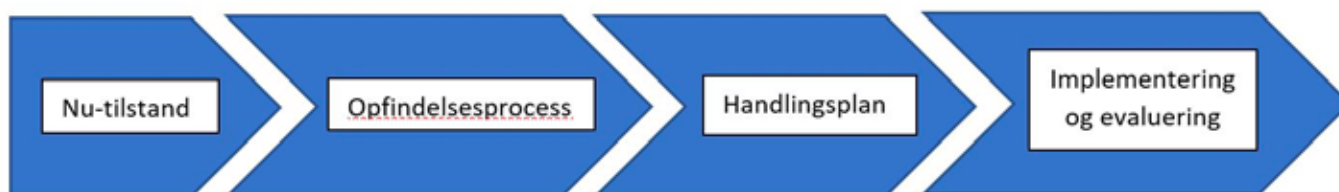
Lektion 3-4 (-5):

Arbejdsmetoder, der opmuntrer til innovation

Vi kender alle sammen Georg Gearløs og hans tænkehat. Nu har vi jo ikke selv nogen tænkehat, men der er forskellige metoder, vi kan bruge, når vi gerne vil forsøge at finde på noget, der kan beskrives som innovativt. Der er mange, mange muligheder her – vælg noget, der passer til jer og det, I arbejder med.

Vi kan arbejde innovativt i fire faser, illustreret i Figur 2:

- "Nu-tilstand" - afdække behov og muligheder. Man skal have baggrundsviden i orden for at kunne forbedre/forny/finde på noget helt nyt.
- Opfindelsesproces



- Handlingsplan
- Implementering og evaluering

Hvad gør vi konkret?

Her er nogle forslag til specifikke øvelser, man kan lave for et givet projekt. Forslag til tidsforbrug er i parentes, men det vil selvfølgelig afhænge både af projektets beskaffenhed og den tid, der er til rådighed.

- Individuel brainstorm (én ide pr post-it) - hver deltager skriver så mange ideer/udfordringer som muligt på hver sin post-it. (5 min)
- Informationssøgning (vær specifik) - sæt klare mål for, hvad der skal undersøges. (30+ min)
- Ændre perspektiv (hvad ville statsministeren gerne have i systemet?) - kan fremprovokere helt andre tankegange end "det sædvanlige" ved at vende udfordringen lidt på hovedet. (10 min - saml op, virker det - 20 min mere)
- Vidensvæg (del med hinanden/skab overblik/skab kaos) - vidensvæggen er holdeplads for ideer, viden, nye spørgsmål, alt det, man har fundet ud af/fundet på indtil videre, giver muligheder for at se nye sammenhænge. (løbende i et projekt - kunne også gøres elektronisk, men det er vigtigt, at det er nemt at omstrukturere og udvide).
- Prioriteringsspil (placer ideer/problemer på en skydeskive – hvad har størst betydning?) (15+ min)
- Prototyper (skitser, fysiske, pop, ...) Se afsnittet om rapid prototyping ([link](#)) - det er muligt at lave adskillige iterationer på prototyping af design på 1 time, også med korte fremvisninger af skitser osv. (1+ time)

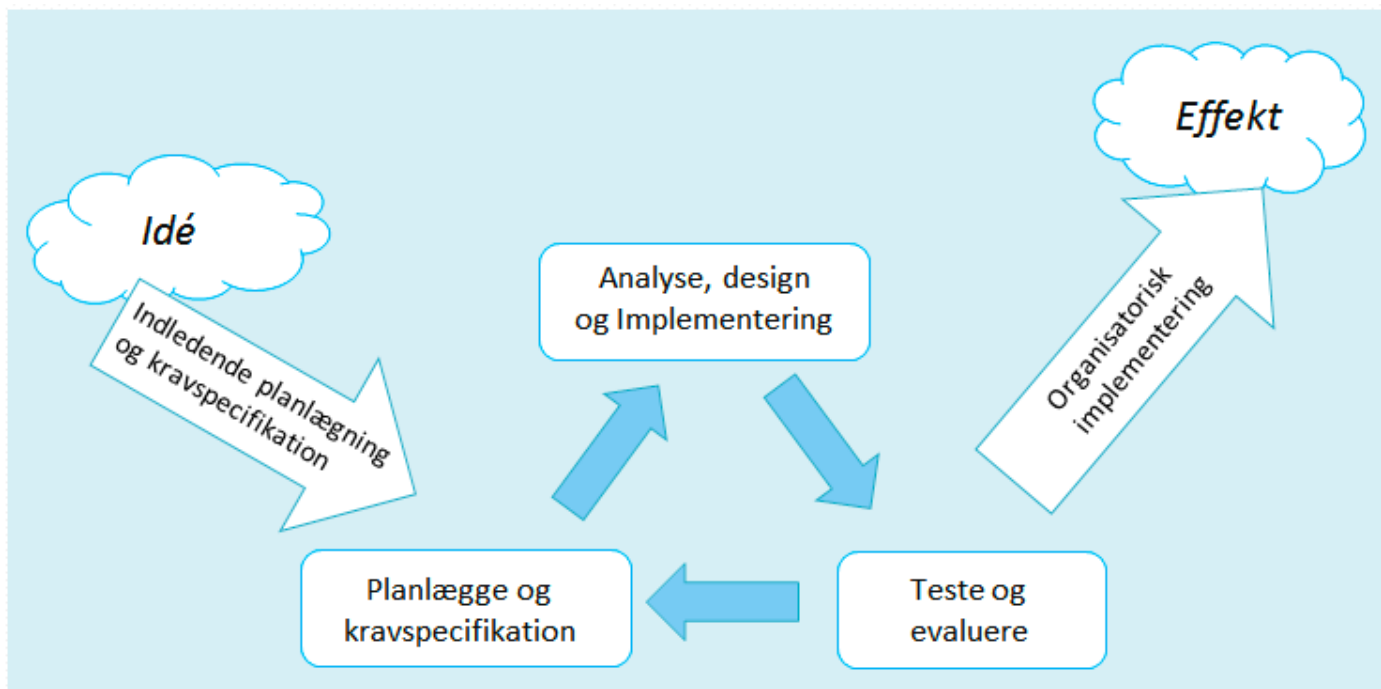
Selvom man prøver øvelserne ovenfor, kan det nogle gange være svært at komme i gang eller komme videre - måske har man en ide, som man synes er god, og så kommer man ikke rigtigt ud over den ide (selvom en variation måske kunne være endnu bedre) - man låser sig fast. I de tilfælde kan man prøve en eller flere af disse ideer:

- Negativ brainstorm (kom med alle de dårlige ideer)
- Provotyper (med vilje dårlige/urealistiske prototyper)
- "Ja, og..." (diskuter, men man skal starte hver sætning med "ja, og..." – benspænd: skal tale efter tur)

Pointen med disse øvelser er, at man ser projektet i et nyt lys, og at man nemmere kommer ud over evt præstationsangst, da det netop handler om de dårlige forslag. (i) handler specielt om at være anerkendende overfor ideer fra de andre i gruppen og tale med det som udgangspunkt.

Domænekendskab

I dette kapitel arbejdes med domæne-kendskab - hvordan man kan indhente viden om en behovskonflikt eller et problem. Til sidst i kapitlet beskrives emnet målgrupper. Domæne-kendskab er vigtig i den første del af udviklingsprocessen. Formålet med kapitlet er at give eleverne en introduktion til emnet. Intentionen er at gøre eleverne bevidste om, at de skal have specifik viden om et domæne, inden de kan udvikle et produkt til det.



Det er første del af udviklingsprocessen der opbygges domænekendskab (idé til kravspecifikation)

Når der skal udvikles en ny app eller et nyt IT-produkt, kigges der på det område/system produktet indgår i. I litteraturen ses området opdelt i to:

- problemområdet (formaliseres i forhold til produktet i maskinen)
- genstandsområdet (hvilken kontekst anvendes IT-produktet i - kontekstområdet)

Kapitlet afsluttes med en række arbejdsopgave til de emner der er gennemgået.

- Domæne-kendskab indgår i mange af de faglige mål - på et overordnet niveau. Domæne-kendskab er en forudsætning for at:
 - anvende konkrete arkitekturer ved udarbejdelse af simple it-produkter
 - anvende programmeringsteknologier til udvikling af it-produkter
 - realisere udvalgte modeller i et konkret it-produkt
 - realisere udvalgte interaktionsdesign i et konkret it-produkt
 - innovative udviklingsprocesser og skitsere idéer til innovative it-produkter.

Emnet vil typisk blive introduceret i Informatik C, men er desuden relevant for Informatik B.

Lektionsplan

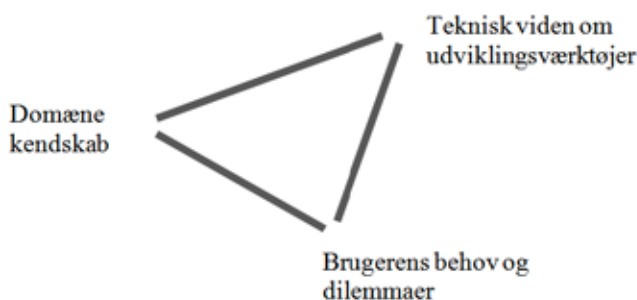
- Der kræves ingen elevforudsætninger for at begynde med emnet.
- Emnet afsluttes med fælles opgaver og en traditionel sekventiel tilgang til emnet vil gradvist opbygge elevens forståelse
 - Undervis i emnerne et ad gangen - begynd med "Problem", så "Opbygning af domænekendskab", osv
- De enkelte emner er selvstændig beskrevet, så det er muligt at plukke i dem.
 - Hvis der plukkes i emnerne, kan det være nødvendig (i forhold til elevernes forforståelse) kort at ridse indholdet af foregående kapitel op

Materiale

Før man kan udvikle et nyt produkt, skal projektgruppen først finde frem til hvilket produkt der skal udvikles. Det kræver indsigt i brugerens dilemma, domænekendskab og det kræver viden om de udviklingsværktøjer der er til rådighed. Løsningsrummet ligger i samspillet mellem de tre.

Materialet er opdelt i under-emnerne (klik på "grundbog" til venstre og så kan emnerne ses i vinduet til højre under "domænekendskab"):

- Et problem, et dilemma, et behov (behovskonflikt, problemløsning)
- Opbygning af domænekendskab
- Målgruppe
- Arbejdsopgaver, domænekendskab



Udviklingen af et nyt produkt sker på baggrund af viden fra tre områder (Ken Mathiasen)

Refleksioner og tips til underviseren

- Domænekendskab er fundamental for at kunne udvikle et godt produkt/system. Et godt domænekendskab opnås ved at gå ydmygt og nysgerrig ind i et område og undersøge det - ofte i dialog med brugerne. For unge udødelige, fordømmende fremstormende og dejlige unge mennesker (vores elever), kan det være svært. Det er godt at få en snak med eleverne om det forhold. Er tiden til det, er der forskellige eye-opener metoder der kan anvendes:
 - Inviter en frisk pensionist ind i klassen, som kan fortælle om sværheden ved at åbne pilleglas, om det avancerede øreapparat, eller om problemerne med NEM-id.
 - Sæt eleverne til at lave et video-portræt af kantinemedarbejderens arbejdsdag, af pedellens, rengøringsassistenten, skemalæggerens osv.

I den daglige undervisning kan det være en fordel at udvikle produkter til en målgruppe som eleverne umiddelbart kan identificere sig med (typisk; "lav en app der hjælper en 1.g'er med at smøre madpakke" eller "hjælp elev i 9.kl med at vælge studieretning" osv). Det omvendte fungerer også - en målgruppe der ligger langt fra eleverne ("lav en app til som gadebørn som viser hvor der er madspild").

Hvordan måles målopfyldelsen? Domænekendskab er en metode som ikke umiddelbart kan måles. Efter et projekt, kan man bede eleverne lave en klassepræsentation af deres arbejdsprocess og på den måde få eleverne til at reflektere over metoderne.



Eksamensspørgsmål kan indeholde en passus som:

- “Din præsentation skal komme ind på målgruppens indvirkning på appens design”
- “Beskriv den anvendte udviklingsproces, herunder: opbygning af domænekendskab og identificering af målgruppe”

Arbejdsopgaver, domænekendskab

I det følgende er der små opgaver til hele emnet domænekendskab.

Formålet er at øge elevernes forståelse for et emne ved at arbejde med det. Intensionen er at bibringe underviseren materiale så noget af forberedelsestiden kan reduceres.

Opgaverne kan både anvendes i Informatik C og Informatik B.

Forudsætninger

Det er tanken at opgaverne laves undervejs, samtidig med at emnerne gennemgås. Er det en presset periode, kan opgaverne også gemmes og laves på et senere tidspunkt (fx den dag netværket ikke virker og man skal have en halv time til at gå)

Materiale

1. Summeøvelse - Problem (behov, dilemma)

a) Et udækket behov er fx "jeg er sulten". Snak med sidemanden og beskriv tre personlige udækkede behov.

b) Der er et dilemma at "jeg vil reducere CO2 udledningen, men at jeg skal bruge energi til at opvarme min bolig"

Snak med sidemanden og beskriv tre dilemmaer fra jeres hverdag.

c) Behovskonflikt. Når to behov skal opfyldes samtidig, bliver man nogle gange tvunget til at vælge det ene behov frem for det andet.

Snak med sidemanden og beskriv en behovskonflikt fra din hverdag.

d) Yderligere læsning. Find et billede af "Maslows behovspyramide".

For hvert niveau af behovspyramiden skal der findes et hverdags eksempel på et konkret behov. Læg mærke til prioriteringen af de forskellige behov - hvilke tanker lægger man til grund for at "Behov for at få sikkerhed" ligger lavt?

2. Summeøvelse - Domænekendskab

a) Der er RAVE fest på skolen.

Forestil dig at du skal lave en app der kan hjælpe med at skaffe tøj til denne fest. En byttebørs-app. Der skal opstilles nogle regler for denne byttebørs, så man er sikker på at tøjstørrelsen passer og at det kun er nyvasket tøj der byttes.

- Brug fem minutter og lav en liste med ti forhold/regler en programmør skal overveje når denne app udvikles.

b) På plejehjemmet Solvang er der en banko-klub.

Bankoklubben lave et IT-produkt som kan hjælpe med at holde styr på antal tilmeldte, og om de

tilmeldte har forudbetalt deres bankoplader. Der skal opstilles nogle regler for dette IT-produkt, så man er sikker på at spillerne får det antal bankoplader de har bestilt. Og at man får pengene tilbage, hvis man er syg og ikke kan komme, selvom man var tilmeldt.

- Brug fem minutter og lav en liste med ti forhold/regler en programmør skal overveje inden han/hun går i gang med at udvikle denne app.

c) Opsamling - RAVE domænet kontra banko domænet.

Diskuter hvor nemt det var at opstille listen for RAVE-byttebørs-appen i forhold til banko-appen. Hvor svært bliver det at lave de to apps - hvor omfangsrigt er det ene problem sammenlignet med det andet?

- Virker det umiddelbart som om at banko-appen er nemmest at lave - er banko-appen et simplere problem som man "bare kan gå i gang med"?

d) Refleksion over egen forforståelse

Tænk på arbejdet med de to lister (banko-app til plejehjemmet kontra RAVE-byttebørsen). Snak sammen om følgende

- Det viser sig at mange plejehjemsbeboere bruger fastnet eller en trykknop-mobiler - fordi det er nemmere når man ryster på fingrene. Da gruppen lavede Banko-app-listen, fik I da snakket om at plejehjemsbeboere ikke har smartphone?
- Snakkede I om, at det skal være muligt at bestille øl og snaps samtidig med, at man bestiller banko-pladerne (det er populært mange steder)?

Når man har et stort domænekendskab (RAVE-appen), kan det virke som der er mange problemer forbundet med at udvikle et produkt. Omvendt, hvis man har et lille domænekendskab, kan det virke nemt at skulle udvikle et produkt. Det kan være svært at vide hvor godt domænekendskab man selv har.

- snak sammen om; hvordan man kan undersøge hvor godt ens eget domænekendskab er?

3) Summeøvelse - målgruppe

a) Målgruppen for spørgeprogrammet "Spørg Charlie" på TV2 er personer over 50 år - det er en bred målgruppe. Det betyder ikke at personer på 20 år ikke får noget ud af at se programmet, men programmet er ikke primært sendt til dem. Målgruppen for "Bamse og Kylling" på DR Ramasjang, er børn i alderen fra 3 til 6 år - det er en forholdsvis smal målgruppe. Snak sammen med din sidekammerat og diskuter følgende:

- Hvad er målgruppen for DR2-nyhedsprogrammet "Deadline" - og hvad er målgruppen for TV-Syds nyhedsudsendelse "Nyheder fra TV Syd"?
- Er målgruppen for "Nyheder fra TV-Syd" bredere eller smallere end målgruppen for "Deadline"?

b) Find tre eksempler på en bred målgruppe - det kunne være en målgruppe der indeholder klassens elever, din mor og Caroline Wozniacki.



c) Opstil parametrene for en smal målgruppe - en målgruppe der måske kun indeholder 1 ud af 10 af eleverne i din klasse.

d) Find en fælles parameter for en målgruppe der indeholder Kim Larsen (sangeren), HKH Dronning Magrethe (monarken), Uffe Ellemand Jensen (tidligere udenrigsminister) og Anders Breinholt (TV-værten).

Ekstra til "d" - målgruppen må ikke indeholde Benjamin Lasnier (YouTuberen) og Kasper Schmeichel (målmænd).

e) Beskriv din egen målgruppe med parametre som alder, postnummer, cyklist osv. Prøv at finde den parameter (eller de parametre) der adskiller dig, fra sin sidemand.

Refleksioner og tips til underviseren

Det er processen der er vigtig - ikke resultatet. Det fungerer fint at lade eleverne arbejde parvis med opgaverne og så samle op i plenum med efterfølgende diskussion. PS. Tips til målgruppen i opgave 3d - rygere.

Et problem, et dilemma, et behov (behovskonflikt, problemløsning)

I dette kapitel arbejdes der kort med emnet "et problem", behovskonflikt og problemløsning. Det er et lille kapitel og formålet er at lægger op til emnet "Opbygning af Domænekendskab".

Kapitlet er baggrundstof, primært for lærerplanens pinde om nye produkter (1 og 7)

- brugerorienterede teknikker til konstruktion af it-produkter
- innovation

Problem og behovskonflikt er primært relevant for udviklingsprocessens første del.

Der er opgaver til emnet (sammen med de omliggende kapitler) i afsnittet "Arbejdsopgaver, domænekendskab"

Materiale

Et problem, et dilemma, et behov

Et nyt produkt udvikles ofte på baggrund af et problem. Et problem er en utilfredsstillende situation, et dilemma. Et problem afdækker et dilemma. Her er en række dilemmaer:

- Jeg skal i byen, men jeg har ikke så mange penge.
- Fiskeriet udsulter havet
- Jeg gider ikke gå i bad, men jeg vil ikke lugte
- Jeg elsker mad, men jeg vil gerne tabe mig

Bag et dilemma lægger der ofte et udækket behov. De menneskelige behov blev i 1954 prioriteret af den amerikanske psykolog Abraham Maslow. Her er forskellige eksempler på behov:

- behov for at sove trygt (et fysiologisk behov)
- behov for at spise sig mæt
- behov for at have gruppetilhørsforhold (et socialt behov)
- behov for at skabe (et selvrealiseringsbehov)

Et udækket behov er fx:

- jeg er tørstig
- jeg er ensom
- jeg fryser

Behovskonflikt

Der er flere behov som samtidig skal opfyldes. Når det ikke kan lade sig gøre, opstår en behovskonflikt. Vi har et behov for tryghed og vi har et behov for mad. Et eksempel på en personlig



behovskonflikt er fx. "jeg er sulten, men jeg kan kun skaffe penge til mad, hvis jeg gør noget farligt". Det tvinger personen til at vælge mellem de to behov.

Hvis man stræber efter at opnå tilfredsstillelse af et behov, uden det lykkes, opstår **frustration**. Når man som app-udvikler støder på frustration hos brugere, peger det sandsynligvis et uløst behov - et problem.

Eksempel på frustration:

På en skole er der en elev som udtrykker et behov for at være i klassen til tiden, når timen begynder (det er et tænkt eksempel:-) Eleven er sulten og går ned i kantinen for at købe en sandwich. Men køen er lang. Eleven bliver frustreret over, at det tager lang tid at købe sandwichen.

Eksemplet viser at en frustration ikke altid afslører hvilket behov der reelt ligger til grund for frustrationen. Er eleven frustreret fordi han/hun er sulten eller fordi han/hun kommer for sent til timen. Dermed er det ikke altid nemt at se hvilke løsninger der er i spil for at løse det udækkede behov - for at løse problemet.

Problemløsning

I en teknologirelateret kontekst, løser vi et problem med en ny teknologi.

- en ny dims (et nyt produkt) løser problemet
- et nyt system løser problemet
- en ny proces løser problemet

I informatik løses et problem ofte med udviklingen af en hjemmeside eller med en ny app. Jo bedre vi forstår det bagvedliggende problem der skal løses, jo bedre og mere succesfuldt bliver den nye app. Det kræver et godt domæne kendskab at lave et godt produkt.

Refleksioner og tips til underviseren

Emnet kan præsenteres som traditionel tavleundervisning med samtidig dialog med klassen. Der fås en bedre dialog, hvis eleverne inden timen har læst kapitlet som lektie.

En variant er at lade eleverne sidde i grupper af to og sammen læse og prøve at forstå teksten.

Opbygning af domænekendskab

I dette kapitel arbejdes der med emnet domænekendskab.

Formålet er at anvise metoder til at undersøge den kontekst et nyt produkt skal indgå i. Intensionen med kapitlet er at beskrivelsen vil give eleverne en forståelse for, at udviklingen af et godt produkt forudsætter et godt domænekendskab.

Domæne-kendskab indgår i mange af de faglige mål - på et overordnet niveau. Domæne-kendskab er en forudsætning for:

- at anvende konkrete arkitekturer ved udarbejdelse af simple it-produkter
- udvikling af it-produkter
- at kunne realisere udvalgte interaktionsdesign i et konkret it-produkt
- innovative udviklingsprocesser og for at kunne skitsere idéer til innovative it-produkter.

Emnet vil typisk blive introduceret i Informatik C, men er desuden relevant for Informatik B.

Domænekendskab er primært relevant for udviklingsprocessens første del. Domænekendskab er oplagt baggrunden for brugerdreven innovation.

Der er opgaver til emnet (sammen med de omliggende kapitler) i afsnittet "Arbejdsopgaver, domænekendskab"

Forudsætninger

Det er en fordel at have gennemgået kapitlet om "Et problem, et dilemma, et behov (behovskonflikt, problemløsning)" inden man tager hul på dette kapitel.

Materiale

Domænekendskab er vigtig - et eksempel

Preben kan ikke svømme. Det vil være svært for Preben at skrive en vejledning i at svømme.

Preben mangler viden om at opholde sig på dybt vand; hvor hurtig synker man, hvad sker der hvis man har tøj på i vandet, hvor hårdt skal man sparke bagud når man svømmer osv. Han mangler domænekendskab.

Preben kan øge hans domænekendskab ved fx at interviewe folk der kan svømme, ved at læse om svømning og ved at se videoer af folk der svømmer.

Dorotina kan svømme. Dorotinas forståelse af genstandsområdet "svømning" er bedre end Prebens. Hvis Dorotina skulle skrive en vejledning i at svømme, vil hendes domænekendskab gøre at hendes vejledning sandsynligvis bliver bedre end Prebens.

Det første der sker i udviklingsprojekter i informatik, er at programmøren opbygger domænekendskab. Man kan ikke lave et succesfuldt produkt eller en ny app der bliver god, uden domænekendskab.

Opbygning af domænekendskab

I SRP, SRO og SSO sammenhæng opstilles en problemformulering der fokuserer på nøgleproblemet. Nøgleproblemet sættes i kontekst ved at beskrive tre/fire problemstillinger der belyser nøgleproblemet.

I informatik opbygges domænekendskab inden forskellige krav til et nyt produkt kendes. I større projekter fastholdes kravene i en kravspecifikation.

Som gymnasieelev har man stor domænekendskab i "det at være gymnasieelev" og vil umiddelbart kunne gennemskue de problemer og muligheder der er i domænet "en gymnasieelevs hverdag". Skal der laves en ny overblik-app til gymnasieelever, vil det være forholdsvis enkelt at belyse og beskrive de krav, der skal opstilles. Det vil være svært for en 50-årig ejendomsmægler at opstille de samme krav. Ejendomsmægleren har ingen domænekendskab.

Omvendt, hvis vi skal udvikle en app til en ejendomsmægler, der kan hjælpe med at planlægge han/hendes hverdag, er vi helt på bar bund. Der opstår straks 1000 spørgsmål.

- Hvad laver en ejendomsmægler ud over at sælge huse?
- Hvordan sælger man huse?
- Snakker man meget i telefon eller skrivers der mest e-mail - skype's der mest?
- Brugers der PC, laptop, tablet eller smartphone?
- osv

Før vi kan programmere en god app til en ejendomsmægler, skal vi vide mere om ejendomsmægleren og de ting han/hun går og laver. Vi skal have svar på de 1000 spørgsmål. Det bedste er at tage ud og snakke med ham/hende, så de kan fortælle om de ting der driller i hverdagen.

Forholdet til den viden vi har

Det tager tid at opbygge et godt domænekendskab. En af farerne ved domænekendskab er vores forhold til den viden vi har. Det er forskel på hvad vi tror og på hvad vi ved.

Pas på. Fordi ofte tror vi, at vi ved meget om en sag, uden at vi reelt gør det. Og vores fordomme kan forstyrrer billedet.

Det er fx kun en jæger der ved hvad der foregår i hans hverdag. Hvad der optager ham/hende og hvad der er af problemer. Mange fordomme om jægere handler om våben og dræbe - og vi ved at jægere har våben og skyder dyr i naturen.

Hvis vi skal lave et reklamespil for jægerforbundet til deres nye hjemmeside, kunne man fristes til at lave skydespil med sniper-rifler og hjorte.

Men det viser sig at der foregår mange andre ting i en jægers domæne. Det være sig forholdet til naturen, datoer for jagtsæson, ulve-diskussioner, hvilket klokkeslet står solen op, diskussion om landbrug kontra vildtpleje, knivlov osv.

Hvis vi umiddelbart går i gang med at løse et problem inden vi forstår domænet (hvis vi udvikler et reklamespil på baggrund af den viden vi tror vi har - og på baggrund vores fordomme), bliver det sandsynligvis ikke godt produkt som brugeren tager til sig. Et skydespil vil nok ikke falde i jægerforbundsts smag, fordi det kun rammer en lille delmængde af deres aktiviteter (og så vil et skydespil puste til negative fordommene om dem).

Metoder til opbygning af domænekendskab

Der er mange metoder til at opbygge viden om et domæne. Det tager tid at opbygge domænekendskab. I informatik arbejdes der med internetsøgninger, litteratur, video og brugerinterview.

Ved at lave en Google-søgning kan der gives et hurtigt overblik. En søgning kan opsætte rammerne for den videre informationssøgning. Pas på du ikke kommer til at bruge fordomme i din søgning. Google svarer på det der spørges om. I tilfældet med jægerforbundet vil en søgning på "skydespil jagt" give almindelige internet skydespil med en jægervinkel.

Her skal indsættes et billede med teksten: Jægere er vilde med naturen og bruger mindre tid foran computeren end gennemsnittet - så en almindelig internetsøgning vil give links og spil som rammer almindelige brugere (ikke jægere).

Ændres søgningen til "jagtforbundet træning" eller "jagtforbund øvelse", kommer der langt flere jæggerrelevante links (formodentlig kedelige links, hvis man ikke er jæger).

Brugerinterview kan foretages ved at invitere en bruger som kommer og fortæller projektgruppen om sine erfaringer. En anden mulighed er at lave et telefoninterview. Fordelen ved et brugerinterview er at der kan opbygges forståelse.

En alternativ metode er at lave et **spørgeskema** der gives til en række brugere. En fordel ved spørgeskema er at en informationsindsamling/undersøgelse kan ramme flere brugere, men ulempen er, at nuancer forsvinder.

Besøg i domænet. Det er altid godt at komme ud og snakke med brugerne på deres hjemmebane - i deres domæne. Det giver brugeren mulighed (fysisk) for at vise eksempler på problemstillinger. Et eksempel kendes fra klimadebatten hvor alle ved, at der smelter meget indlandsis fra Grønland. For nogle år siden inviterede den danske regering en række statsoverhoveder til Grønland, så de ved selvsyn kunne se det. Derved fik statsoverhovederne et lidt bedre domænekendskab og kunne bedre forholde sig til problemets størrelse.

Litteratur. Det kan give god forståelse for et domæne, at læse en bog eller artikler om emnet.

Video-kigging. Googlesøgningen vil ofte give et antal videoer som kan give en god forståelse for domænet. Det kan tage tid at finde en omtale af en specifik problemstilling, så ordentlig videokigging kræver tålmodighed .

Konkurrent analyse. Skal der udvikles en hjemmeside til en web-shop, giver det god mening at undersøge virksomhedens konkurrenter for at se hvorledes deres web-shop fungerer. Det samme gør sig gældende, hvis vi bliver bedt om at udvikle en "find mit barn app" til fx Tivoli. I det tilfælde ville der være fornuftigt at kigge på hvad man gør i Liseberg eller i Legoland - og om der er andre



produkter på markedet der kan bruges til at løse problemet.

Måske kan man købe et allerede udviklet produkt - en løsning der ofte vil være billigere for kunden end et egenudviklet produkt.

En web-butik anvender normalt ikke et egenudviklet betalingsmodul, men implementerer et allerede eksisterende produkt i sin web-shop.

Hver metode har sine fordele og ulemper og det er projektet eller produktet der vil diktere de metoder der passer bedst i den givne situation.

Refleksioner og tips til underviseren

Didaktiske pointer/tips

Dette emne er fleksibelt i forhold til hvor grundigt det gennemgås. I et presset forløb kan kapitlet gennemgås på 30 minutter (hvis eleverne har læst det som lektie inden). Emnet åbner op for at man kan bruge et modul eller to og dykke ned i de forskellige metoder. Det fungerer bedst hvis eleverne har en konkret opgave at arbejde med. Lad eleverne undersøge et ukendt domæne - brug fx. den svenske cykelklub "Cykelklubben Avanti" som cykler imod nazisme. Klubbens hjemmeside er her: [\[1\]](#) og deres facebookside er her: [\[2\]](#)

Der udarbejdeten række opgaver til dette kapitel. Det kan være en fordel at lade eleverne lave opgaverne undervejs (det kan laves som pararbejde). Det gør at præsentationen bliver brudt (variation i undervisningen).

I forhold til eksamensspørgsmål, er emnet med i den overordnede beskrivelse af Domænekendskab

Rige billeder

Rige Billeder er en metode til opbygning af domænekendskab. Rige Billeder gennemføres som pararbejde eller i grupper - der skal være en bruger eller anden person med stort kendskab til domænet. Formålet med dette kapitel er at beskrive metoden.

I forhold til lærerplanens faglige mål berører Rige Billeder følgende pinde:

- analysere og vurdere, hvordan it-systemer har betydning for og påvirker menneskelige aktiviteter, samt anvende brugerorienterede teknikker til konstruktion af it-produkter
- realisere udvalgte interaktionsdesign i et konkret it-produkt og tilpasse eksisterende design og systemer i konsekvens heraf
- redegøre for innovative udviklingsprocesser og skitsere idéer til innovative it-produkter.

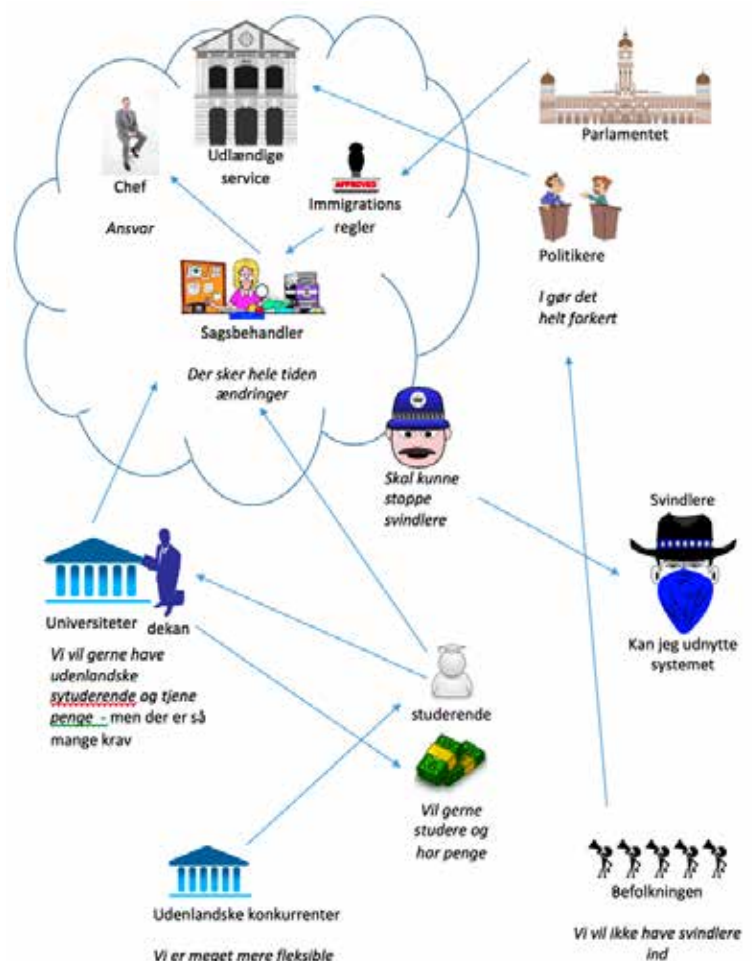
Emnet vil typisk blive introduceret i Informatik B hvor man har tid til at gå i dybden med Domænekendskab, men der er inge problemer i at undervise i emnet på Informatik C. Rige Billeder er primært relevant for udviklingsprocessens første del.

Forudsætninger

Alle kan deltage i en session med Rige Billeder - uden forudsætninger. Husk den ene part skal være en person med domænekendskab.

Materiale

En metode til at opnå et domæne kendskab er tegne et rigt billede i samarbejde med en person med kendskab til domænet. Et rigt billede er en tegning uden formelle symboler eller egentlig syntaks og det bruges til at skabe et visuelt billede overblik over hvilke enheder, processer, problemer og strukturer der er i domænet. Det rige billede kan tegnes i hånden eller på computeren. Når der tales enheder menes der mennesker, fysiske objekter, steder, organisationer, roller og opgaver, mens processerne er de dynamiske sammenhænge der mellem enhederne f.eks. arbejdsgange, informationsstrømme, kontrol osv. I arbejdet med at tegne det rige billede får man oftest øje på de problemstillinger eller misforhold der er i domænet og det er der, den IT-system der skal udvikles, skal sætte ind og afhjælpe situationen. Det rige billede giver også et godt billede på hvilke strukturer der er i domæne, da man kan se ejerforhold og hvem der anvender de fysiske objekter.



Her er et eksempel på Rige Billeder

Det rige billede i ovenstående figur beskriver domænet omkring universiteter der gerne vil have udenlandske studerende ind for at tjene penge. Der er dog nogle regler der er under konstant forandring på grund af pres fra befolkningen og derved politikerne, der er svindlere der forsøger at udnytte systemet og konkurrenter der også gerne vil have de studerende. Den studerende og universitetet har svært ved at håndtere den situation. Billedet viser de enheder, processer og strukturer der er indenfor domænet på en visuel måde, der går domænet lettere anskueligt.

Det rige billede kan fungere som udgangspunkt for den specifikation af IT-systemet man skal foretage sig efter man har opnået det rette domænekendskab, da man har et overblik over de arbejdsgange og eventuelle udfordringer som IT-systemet skal håndtere.

Refleksioner og tips til underviseren

Didaktiske pointer/tips

- I en undervisningssituation hvor eleverne arbejder med opbygningen af et Rigt Billede, skal der være god tid. Indtil eleverne er fortrolige med metoden, er det processen er der vigtig - ikke produktet.
- Det er svært at måle om eleverne har lært teknikken. Det er et praktisk emne og det er først til eksamen eleverne får tid til at reflektere over metoden - så husk at udlever materiale til eleverne, og opfordre dem til at tage billeder af processen undervejs - til deres portfolio-mappe.

Eksamensspørgsmål: overvejelser & eksempler

Et eksamensspørgsmål kan fint indeholde en pasus som:

- Din præsentation skal indeholde ...opbygge domænekandskab med metoden Rige Billeder.
- ..forklar hvorledes projektgruppen anvendte Rige Billeder-metoden til opbygning af domænekendskab.

Målgruppe

I dette kapitel arbejdes der med emnet "målgruppe". Formålet er at give en kort introduktion til emnet. Intentionen er at eleverne bliver bevidste om at der er forskellige målgrupper.

Kapitlet er baggrundstof, primært for de pinde i lærerplanen der involverer brugere og brugerforhold.

- it-systemers betydning for og påvirkning af menneskelig aktivitet
- udvikling af it-produkter og tilpasning af eksisterende it-systemer
- interaktionsdesign
- skitsere idéer til innovative it-produkter.'

Der er opgaver til emnet (sammen med de omliggende kapitler) i afsnittet "Arbejdsopgaver, domænekendskab"

Materiale

Målgruppe

Samtidig med at der opbygges domænekendskab, bliver det tydeligere hvem brugeren af produktet er. En forståelse for brugeren og dennes problemer er lige så vigtig som domænekendskabet. En målgruppe er betegnelsen for den gruppe brugere, et produkt henvender sig til. Jo bedre en målgruppe kan specificeres, jo bedre kan domæne-problemerne forstås og jo bedre kan produktet laves.

Det tales om smalle eller brede målgrupper. Et eksempel på en bred målgruppe er "brugere af Facebook". Næsten alle danskere har en facebookprofil. Det er lige meget om man er barn, ung eller gammel. Om man bor i byen eller på landet. Om man er almindelig spisende eller vegetar. En bred målgruppe indkapsler mange forskellige mennesker.

Jo flere specifikke informationer vi har om en målgruppe, jo smallere er den. En målgruppe der består af "Facebookbrugere på en ungdomsuddannelse", er smallere end bare "brugere af Facebook". Nu indeholder målgruppen primært unge (ikke børn, ældre og gamle). Målgruppen kan gøres smallere ved igen at blive mere specifik. Fx vil en smal målgruppe kunne være kvinder i 3.g på HHX gymnasier i Vejle der spiller håndbold.

Persona

Et produkt udvikles normalt til en målgruppe. En alternativ tilgang er at tage udgangspunkt i én bruger og se denne som en slags rollemodel for en hel brugergruppe. Begrebet kaldes Persona - ordet stammer fra latin og betyder; maske, rolle, karakter. Persona beskriver en slags arketyper der fx i en projektgruppediskussion, gør det nemmere at relatere et produkt til en bruger. Her er et eksempel på en persona fra <http://blog.brandmovers.dk/guide-content-marketing-personaer>:

For Morten handler det ALTID om at komme først. Forpustet og glad smider han sig derfor i sin kontorstol, imens han i samme bevægelse tænder computeren. Mandag morgen er startet som alle dage efter hans mening burde starte for en marketingchef i et større dansk konsulenthus. Farvelkysset til fruene lovede mere. Turen fra hjemmet på Frederiksberg til børnenes institution og derefter til kontoret i indre København blev klaret på rekordtid. Kaffen på kontoret er dampende frisk. Og humøret bliver kun bedre, da tallene fra den seneste kampagne tikker ind på Mortens iPhone og afslører, at virksomhedens nye strategi giver pote, og de fleste kampagner giver et rigtig



pænt ROI - hvilket efter hans bedste overbevisning vil sikre hans virksomhed en god position i branchens førerfelt.

Foruden beskrivelser af hverdagssituationer (kaldet scenarier) indeholder personaen ofte en beskrivelse af rollemodellens alder, køn, uddannelse, hverdag, interesser og så videre.

I afsnittet om interaktionsdesign kan man finde et fyldigt eksempel på en persona-beskrivelse.

Refleksioner og tips til underviseren

Emnet kan præsenteres som traditionel tavleundervisning med samtidig dialog med klassen.

En variant er at lade eleverne sidde i grupper af to og sammen læse og prøve at forstå teksten.

I afsnittet "Arbejdsopgaver, domænekendskab" er der opgaver til emnet.

I forhold til eksamensspørgsmål, er emnet med i den overordnede beskrivelse af Domænekendskab

Kravspecifikation

En kravspecifikation er et dokument der fastholder et produkts nøgleparametre. Formålet med dette kapitel er at give en kort introduktion til en kravspecifikation. Intentionen er at afmystificere dokumentation ved at give nogle praktiske tips. En Kravspecifikation hører typisk hjemme i den første del af udviklingsprocessen (men skal løbene opdateres). Faglige mål dette emne berører:

- anvende konkrete arkitekturer ved udarbejdelse af simple it-produkter
- realisere udvalgte modeller i et konkret it-produkt
- realisere udvalgte interaktionsdesign i et konkret it-produkt

Kravspecifikationen vil typisk blive introduceret i Informatik C (orienterende), og vil være en fast del af større projekter i Informatik B.

Forudsætninger

Det er en fordel (men ikke et krav) at have gennemført følgende grundbogs-emner inden kapitlet: "Et problem, et dilemma..." og "Målgruppe" og "Brainstorm"

Materiale

En kravspecifikation er et dokument der fastholder et produkts nøgleparametre. En kravspecifikation er et Worddokument som beskriver de ting en app skal kunne. Der findes kravspecifikationer på flere hundrede sider - i informatik i gymnasiet er en halv A4 side ofte nok.

I eksemplet med en app til Postmand Per (i brainstorm-forløbet) ser kravspecifikationen således ud:

App til hjemvesforebyggelse - projekt Postmand Pat km 13/8-2016

Opgave (her skrives på almindelig dansk, hvad det er der skal laves):

Der skal laves en lille app til en smartphone som viser et billede af Emil (Postmand Pers kat) når der trykkes på en knap.

Hvis brugeren ær Emil (stryger billedet - flung) skal Emil sig miav.

Input:	tryk på en knap flung på billede	skal vise et Emil-billede skal afspille miav-lyd
Formler:	-ingen	
Output:	visning af billede afspilning af lyd	aktiveres af knappen aktiveres af flung på billede
Kommentarer:	-ingen	

Et krav skal være entydig og målbart.

Der er mange måder at opskrive kravene og strukturere et krav-specifikations-dokument på. Fordelen ved at lave en kravspecifikation er, at alle projektdeltagere bliver enige om, hvilket produkt der skal udvikles og hvordan det skal virke. Samtidig har man et dokument der kan vise opgavestilleren (kunden) hvad det er projektgruppen vil udvikle til ham/hende.

Foruden at være et godt oversigtsdokument for projektgruppen, anvendes kravspecifikationen i industrien som en slags afleveringsdokument. Populær sagt, så betaler kunden for at få leveret det der står i kravspecifikationen. Omvendt har projektgruppen leveret produktet og skal derfor have sine penge når produktet/appen har bestået testspecifikationen.

Refleksioner og tips til underviseren

- Det kan være en fordel at gemme dette kapitel indtil det er relevant for eleverne. Fx umiddelbart efter en brainstorm session (når eleverne har en klar holdning til hvilket produkt de vil udvikle).
- Når emnet "Kravspecifikation" gennemgås, kan det gøres med udgangspunkt i et pseudo-produkt (fx Postmand Per eksemplet) og komme med et par eksempler på, hvordan et krav skal formuleres. Derved gøres emnet mere konkret for eleverne. En god test på om et krav er ordentlig formuleret, er, om kravet er målbart.
- Der er god mulighed for at evaluere elevernes forståelse af emnet, ved at lade dem udarbejde en kravspecifikation i forbindelse med et projekt. Kravspecifikationen (med tilhørende testspecifikation) kan normalt være på én A4 side.
- Kravspecifikation kan indgå som et delement af et eksamensspørgsmål. Her er et eksempel:
 - Dokumentation skal indeholde elementer af:
 - Kravspecifikation.
 - Beskrivelse af App'ens virkemåde (set med en brugers øjne).
 - App'ens opbygning - beskrivelse af hovedfunktionerne (set med en programmørs øjne).
 - Testspecifikation.
- Det er et stort emne, så det er vigtigt at holde fast i de overordnede linjer for en kravspecifikation. Det er godt at lade eleverne arbejde 20 minutter med at udarbejde (formulere) tre specifikke (og testbare) krav til et produkt.

Testspecifikation

En testspecifikation er et dokument forklarer kunden og projektgruppe hvilke test der skal foretages i forbindelse med et produkt afleveres. Formålet med dette kapitel er at give en kort introduktion til en testspecifikation. Intensionen er at forklare hvor simpelt en testspecifikation laves - at afmystificerer dokumentation ved at give nogle praktiske tips. En testspecifikation hører uløseligt sammen med Kravspecifikationen og anvende mest i den første del af udviklingsprocessen (men skal løbene opdateres) Faglige mål dette emne berører:

- anvende konkrete arkitekturer ved udarbejdelse af simple it-produkter
- realisere udvalgte modeller i et konkret it-produkt
- realisere udvalgte interaktionsdesign i et konkret it-produkt

Kravspecifikationen vil typisk blive introduceret i Informatik C (orienterende), og vil være en fast del af større projekter i Informatik B.

Forudsætninger

Det er en fordel at have gennemført kapitlet om Kravspecifikation inden dette kapitlet.

Materiale

I større projekter anvendes kravspecifikationen som udgangspunkt for en test af produktet. Det er en fordel at lave et testdokument inden man går i gang med at udvikle appen.

Et krav som fx "Byg et hus så højt som muligt" er ikke entydig og det er ikke målbart. Fordi, hvor højt er "så højt som muligt"?

Fordelen ved at lave testspecifikationen inden projektudviklingen begynder er, at man derved fanger dårligt formulerede krav. Man kan ikke teste "så højt som muligt". Så i det øjeblik man beskriver en test som skal det, vil man opdage at det ikke er muligt. Det er nemt at teste om et hus er over eller under 16 meter. Så kravet skal ændres til "Byg et hus der er mindst 16 m højt"



Et krav skal være entydig og målbart

Hvis kravspecifikationen ændres, kan det være at kunden ikke vil godkende den. Kunden siger; "16 meter er ikke højt nok - huset skal være mindst 50 meter højt". Men dette forhold er umådelig vigtigt. Hvis projektgruppen havde bygget et 16 m højt hus, men kunden forventede et 50 meter hus, opstår der problemer. Kunden regnede med et 50 m højt hus og vil ikke betale for et 16 meters hus.

Det kan være projektgruppen ikke kan lave et hus der er 50 meter højt. Æv, men sådan er det. Der spares mange kræfter og ærgrelser ved at få det fastlagt inden projektarbejdet påbegyndes. Pointen er at ved at skrive testspecifikationen med det samme afprøves kravspecifikationen og alle bliver mere skarpe på hvad det er der udvikles - også kunden.

I industrien anvendes kravspecifikationen som en slags afleveringsdokument. Populær sagt, så betaler kunden for at få leveret det der står i kravspecifikationen. Omvendt har projektgruppen leveret produktet og skal derfor have sine penge når produktet/appen har bestået de test der er listet i testspecifikationen.

Refleksioner og tips til underviseren

Dette kapitel kan med fordel gennemgås umiddelbart efter "Kravspecifikation" - well, faktisk samtidig. Der ligger en god pointe i, at et krav ikke er målbart hvis det ikke kan testes.

Der er god mulighed for at evaluere elevernes forståelse af emnet, ved at lade dem udarbejde en testspecifikation i forbindelse med et projekt. I et C-projekt bør Kravspecifikationen (med tilhørende testspecifikation) kunne klemmes ned på én A4 side.

Testspecifikation kan indgå som et delelement af et eksamensspørgsmål. Her er et eksempel: Dokumentationen skal indeholde elementer af:

- Kravspecifikation.
- Beskrivelse af App'ens virkemåde (set med en brugers øjne).
- App'ens opbygning - beskrivelse af hovedfunktionerne (set med en programmørs øjne).
- Testspecifikation.

En god måde at lave et review af testespecifikationen er at lade grupperne læse/kommentere de andre gruppers kravspecifikation og testspecifikation.

Use Case

I dette kapitel arbejdes med Use Case - en metode til at undersøge hvilke funktioner et system skal have. Formålet med kapitlet er at give eleverne en introduktion til emnet. Intentionen er at gøre eleverne bevidste om, at der er flere metoder til opbygning af domænekendskab og til informationsindsamling og produkt specifikation.

Emnet vil typisk blive introduceret i Informatik C, men er desuden relevant for Informatik B.

Materiale

En use case kaldes også for et brugstilfælde og det bruges til at beskrive hvilke funktioner et system skal have, hvad der skal være til stede for at funktionen kan fungere og hvilke brugere der anvender funktionerne.

Dette er et eksempel på et use case diagram der viser de helt grundlæggende funktioner, services og brugere der skal være til stede i en online webshop.:

Der er 2 forskellige brugere af systemet, den registrerede og ikke-registrerede, og mens de begge kan se et produkt, er det kun den registrerede bruger der kan købe det. Selve købs use casen indeholder både vis produkt og check ud use cases, da det er funktioner der er nødvendige for at kunne købe et produkt. På højre side af diagrammet er der en service der godkender de data som brugeren indtaster i systemet i forbindelse med udcheckning og registrering, så der ikke sker fejl. Til sidst er det enten paypal eller kreditkort service der tager sig af at validere betalingen i check ud use casen.



Når man har overblikket over hovedfunktionerne i systemet kan man begynde at udspecificere dem, så man får beskrevet hvad

der skal være til stede for at use casen kan fungere (include) og hvilke andre der bruger (extends) use casen. For use casen se produkt kunne det se således ud:

På denne måde kan man bygge sit system op så programmøren ved hvilke funktioner der skal programmeres, hvordan sammenhængen er mellem dem og hvilke brugere der skal kunne bruge funktionerne.

Refleksioner og tips til underviseren

Det kan være en fordel at arbejde med use cases som det næste trin efter man f.eks. har lavet et rigt billede, da det kan være svært at gennemskue, hvilke use cases der skal være i et system udelukkende på baggrund af en tekstbeskrivelse af en proces eller et problemfelt.

Sammen med et rigt billede er use case diagrammet de diagrammer der udgør grundlaget for foranalysen i en systemudviklingsmodel.

Det kan være en fordel at gemme dette kapitel indtil det er relevant for eleverne. Fx umiddelbart efter et bruger interview eller inden eleverne påbegynder et nyt projektforløb.



Hvordan måles målopfyldelsen? Det er muligt at evaluere elevernes forståelse, ved at lade dem en Use Case beskrivelse i forbindelse med et projekt. Beskrivelsen kan eventuelt som afleveres som hjemmeopgave.

Eksamensspørgsmål kan indeholde en passus som:

- "Beskriv den anvendte udviklingsproces, herunder: opbygning af Use Case diagram"
- "På baggrund af problembeskrivelsen udarbejdes et rigt billede og et udse case diagram"

Gestaltlove

I gestaltpsykologien er der opstillet en række love (gestaltlovene), som giver god mening at anvende til interaktionsdesign. Gestaltlovene handler om den måde, hvorpå hjernen grupperer former og figurer, og prøver at uddrage et helhedsbillede. Her er de grundlæggende principper (i parentes er de engelske betegnelser nævnt).

- 1) Nærhed (proximity). Figurer der er placeret tæt på hinanden ses som en gruppe.
- 2) Lighed (similarity invariants). Ens figurer opfattes som en gruppe.
- 3) Lukkethed (closure). Delelementer af et billede stykkes sammen til at skabe helheden.
- 4) Kontinuitet og symmetri. Optræder to figurer symmetriske omkring en linje, ses de som en gruppe.
- 5) Prægnans (Prägnanz) - figur/baggrund. Hjernen leder den efter mønstre, kontinuitet, ensartethed.
- 6) Erfaring (Past experience, "Common Fate"). Dækker bl.a. over brugen af ikoner
- 7) Forbundethed - Forbindes figurer med en streg, opfattes de umiddelbart som sammenhørende (forbundenhed er et design tips - ikke en gestaltlov).

Interaktionsdesign - Se Bilag 1

Arkitektur

Læreplanen om kernestoffet knyttet til It-systemers arkitektur:

- (STX, C-niveau) tre-lagsarkitektur (eller tilsvarende) som generel ramme for at forstå en meget stor klasse af it-systemer, deres komponenter og samspillet mellem disse.
- (STX, B-niveau) "cloud-computing", "client-server"-arkitektur og model-view-controller; konkrete systemer baseret på disse arkitekturer.

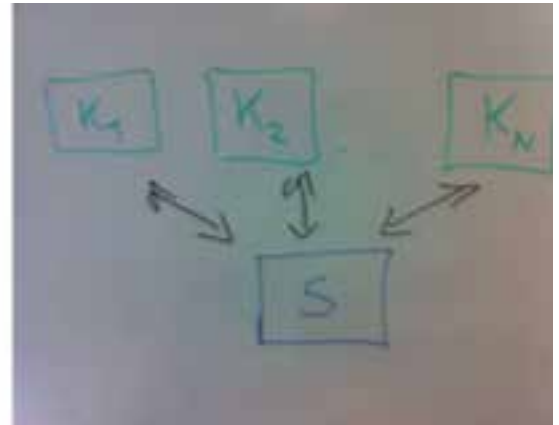
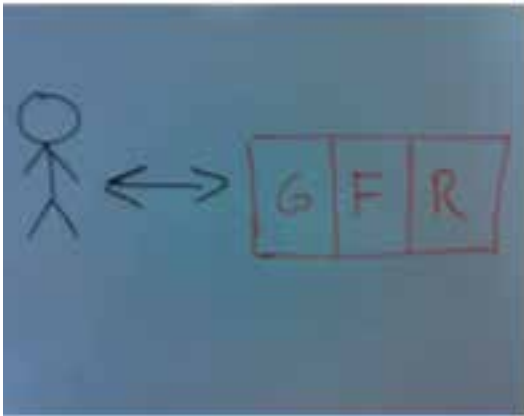
Arkitektur handler om strukturen af it-systemer, og der findes mange forskellige varianter og forståelser af arkitekturbegrebet: man kan tale om applikations-arkitektur, som er strukturen af individuelle systemer, om informations-arkitektur, der handler om strukturen af data (repræsentationer), om system-arkitektur, der handler om strukturen ml. en familie af sammenhængende systemer, osv. Fælles for dem alle er, at når man anvender et arkitektursyn på it, så fokuserer man på strukturen og interaktionen mellem delene (og ikke på selve komponenterne, modulerne, programmerne). Grunden til at dette er vigtigt, er at man erfaringsmæssigt har erkendt, at når et system bliver sat i drift, så kommer der uværgeligt rettelser og tilføjelser til systemerne. Hvis systemet har en uhensigtsmæssigt struktur, så bliver det ofte vanskeligt at vedligeholde disse systemer. Nogle gange så vanskeligt, at man må skrotte et system, og lave et helt nyt, hvilket er dyrt. Omvendt, hvis systemet har en god arkitektur - typisk beskrevet som løs kobling mellem delene - så er det nemmere at vedligeholde systemet.

Naturligvis har UML et omfattende beskrivelsesapparat til at modellere arkitektur - faktisk fra en hel række (typisk 4 eller 5) forskellige synsvinkler (ofte kaldet views). I denne note vil vi dog gå lidt mere praktisk og pædagogisk til værks, idet vi vil beskæftige os med en helt generel model for arkitektur, som kan være vældig nyttig, når man skal forstå, bruge, ændre, eller skabe systemer. Modellen er således ikke specifik for et bestemt konkret system, men er generel anvendelig som mental model (forståelse) af alle systemer. Modellen opdeler et system i 3 dele: grænseflade, funktionalitet, repræsentation (Repræsentationen kaldes ofte for "modellen", idet den er en model af systemets problemområde, men det forvirrer begreberne i forhold til det, vi indtil videre har sagt om modeller og modellering.), i det følgende benævnt G, F, og R.

- Grænsefladen er den del af systemet (af koden), som brugeren (eller andre eksterne systemer) kommunikerer med. Tænk på knapper, vinduer, scroll-barer, indtastningsfelter - alt det som du kan navigere rundt i med mus og tastatur på en pc eller med fingrene på en smartphone.
- Repræsentationen er den del af systemet (koden), som indeholder repræsentationerne af det som systemet handler om - dataene. Hvis det er en musik-app vi kigger på, så drejer det sig om selve musiknumrene, som de er gemt i systemet, og hvad der ellers er gemt af yderligere information i forbindelse med disse musiknumre: coverbillede, titel, albumtitel, kunstner, længde, kvalitet, genre, antal afspilninger, osv.
- Funktionaliteten er den del af systemet (koden), som typisk indeholder algoritmerne for det som vi ønsker at gøre ved vores data. Hvis det f.eks. er et program til at lave ringetoner med, vi kigger på, så er der typisk funktioner til at udvælge, redigere og gemme lyden: vælge en del ud, ændre lydstyrken, lave ekko, ændre pitch, sætte delen ind på et andet tidspunkt i nummeret, lave loops, osv.

Øvelse: Hvorfor tror du disse 3 aspekter typisk er adskilt fra hinanden i et system?

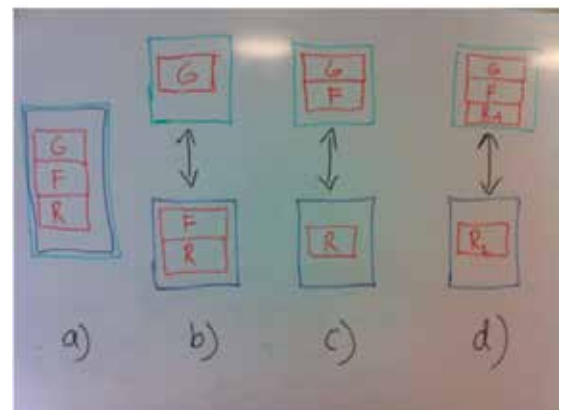
Ovenfor har vi ikke taget stilling til, hvor systemdelen (koden) rent fysisk befinder sig: om det hele er placeret på samme computer eller fordelt på flere forskellige maskiner. Der er nemlig flere muligheder. Betragt figuren nedenfor.



To arkitekturmønstre: GFR og Klient-Server

Figuren viser to klassiske mønstre i arkitektur. Til venstre har vi ideen om lagdeling, hvor vi har opdelt et system i grænseflade, funktionalitet, og repræsentation, som forklaret tidligere. Til højre har vi ideen om klienter og servere: pointen i dette mønster er, at en række forskellige brugere af systemet kan benytte hver deres klient til at forbinde sig til og kommunikere med en fælles server. Dels betyder det, at flere brugere kan bruge deres klienter samtidigt (uden at skulle slås om tastatur og skærm), dels betyder det, at brugerne og klienterne kan være geografisk distribuerede. Forskellige kombinationer af disse to simple mønstre giver en god forståelse for mange af de systemer, vi bruger i det daglige. Betragt figuren nedenfor. De grønne kasser er klienter, de blå er servere, og de røde er vores forskellige logiske systemdele (G, F, R).

- I a-varianten har vi alle tre systemdele kørende på samme computer. Et eksempel er applikationerne fra officepakken, hvor vi kan bruge henholdsvis Word, Powerpoint, og Excel isoleret på vores egen maskine uden at være på nettet.
- I b-varianten har vi vores grænsefladekomponent placeret på en klient, hvorimod funktionaliteten og repræsentationen er placerede på en server. Dette er typisk for mange web-baserede systemer, at vi har vores browser kørende på vores egen computer, men at vi så kan koble op på en webside, der tilbyder noget funktionalitet og noget data vi kan arbejde på. F.eks. Facebook.
- I c-varianten har vi både grænsefladen og funktionaliteten placeret på vores egen klient. Hvis du har prøvet at arbejde med et "fællesdrev", som bruges til at dele dokumenter og filer på en skole eller en arbejdsplads, så kender du også denne variant. Dette gælder f.eks. hvis du bruger Word til at rette i et dokument som ligger på et fællesdrev. Eller hvis du har adgang til en database, som du kan bruge eller opdatere. Hvis du har prøvet dette, så kender du også fornemmelsen, når netværksforbindelsen mistes - enten pga. en fejl, eller fordi du sidder i sommerhuset uden internet og skulle have lagt sidste hånd på en opgave, der skulle afleveres i morgen.
- Endelig er der d-varianten, som skal forstås på den måde, at repræsentationen (vores data)



4 forskellige applikationsarkitekturer

er fordelt på både klient og server. Dette kender du sikkert fra de mange såkaldte cloud-baserede services, som er dukket op de senere år: Google Drive, Microsoft Skydrive, Apple's iCloud, eller DropBox. Her er idéen, at man kan arbejde videre på sine data, selv om man ikke har netforbindelse. Klient-computeren (f.eks. din bærbare) opretholder nemlig en lokal kopi af dine data fra serveren, som du altid kan arbejde på. Når computeren så igen får netforbindelse, så opdaterer den automatisk både din lokale kopi (hvis der er ændringer på serveren, f.eks. fra andre brugere af DropBox, som har uploadet nye filer til jeres delte dropbox) og serveren (hvis du har ændret i eller oprettet nye dokumenter i din lokale kopi af jeres dropbox). Dropbox synkroniserer således klienter og serveren.

Øvelse:

- Vælg en håndfuld applikationseksempler, som du kender fra pc/mac, ipads, og smartphones.
- Overvej for hver enkelt applikation, hvilken af varianterne a-d du tror, der bliver anvendt.
- For hver variant a-d, overvej fordele og ulemper ved denne arkitektur.
- Overvej om der er flere varianter end de nævnte a-d.

Dataanalyse og databaser

En database er lang række data, der er blevet struktureret således, at der er relationer mellem tabellerne og det er muligt at indsætte og udtrække den ønskede information fra databasen.

Når der skal udvikles en database, er det vigtigt at alle de data man ønsker kan udtrækkes fra databasen er til stede eller er muligt at udregne på baggrund det data, der findes i databasen. Det sikres ved at lave en dataanalyse.

Du kan lave en dataanalyse på baggrund af flowdiagrammer eller skitser af skærbilleder fra det system databasen skal understøtte. Det foregår ved at du skriver alle data ned fra hvert skærbillede eller flowchart som enten skal vises til brugeren eller gemmes i databasen.



Eksempel på skærbilleder

Samme proces kan bruges ved flowchart- eller aktivitetsdiagrammer, der skriver man blot ned hvilke data der hentes/skrives i hvert element i diagrammet.

Det giver følgende liste:

Pizzanr	Pizzanavn	Pris	Beskrivelse
Billede	Ordrelinjenr	Pizzanavn	Navn
Adresse	Telefon	Leveringsdato	Leveringstidspunkt
Udbringning/afhent selv			

Dubletter fjernes:

Pizzanr	Pizzanavn	Pris	Beskrivelse
Billede	Ordrelinjenr	Pizzanavn	Navn
Adresse	Telefon	Leveringsdato	Leveringstidspunkt
Udbringning/afhent selv			

Data grupperes ud fra det de siger noget om, hvilket bliver navnet på tabellen.

Pizza	Ordrelinje	Ordre
Pizzanummer Pizzanavn Pris Beskrivelse Billede	Ordrelinjenummer	Navn Adresse Telefon Leveringsdato Leveringstidspunkt Udbringning/afhent selv

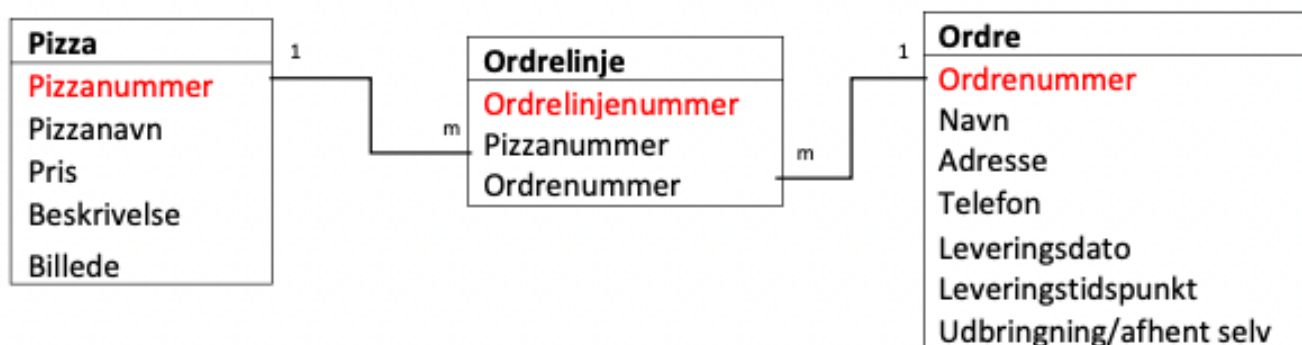
* Note: i dette system gemmes kundens profil ikke, hvilket er grunden til at der ikke er en kundetabel

Hver tabel skal have en primærnøgle, der er unik. Hvis der ikke er data til stede, der egner sig som primærnøgle, skal der automatisk genereres en unik primærnøgle, typisk et løbenummer. Herunder er alle 3 primærnøgle autogeneratede løbenumre.

Pizza	Ordrelinje	Ordre
Pizzanummer Pizzanavn Pris Beskrivelse Billede	Ordrelinjenummer	Ordrenummer Navn Adresse Telefon Leveringsdato Leveringstidspunkt Udbringning/afhent selv

Så skal relationerne på plads: En ordre har mange pizzaer og en pizza kan findes på mange ordrer. Det kaldes en mange-tilmange relation. De skal splittes op for at opfylde den første normalform om at alle rækker i en tabel skal være lige lange. Derfor laver man ordrelinjetabellen, der gør at en pizza kan findes på mange ordrelinjer, men en ordrelinje kan kun have en pizza. På samme måde kan en ordre have mange ordrelinjer, men en ordrelinje kan kun passe med en bestemt ordre.

Det giver følgende relationer:



Før databasen kan oprettes skal man lægge sig fast på hvilken datatype de forskellige data skal have. Datatypen fortæller hvilken type data der er tale om f.eks. om det er heltal, tekst, dato osv.

Der findes en lang række datatyper og de mest gængse er følgende:

Datatype	Beskrivelse	Datatype	Beskrivelse
INT	Heltal	Double	Decimaltal
VARCHAR	Alle former for tegn, både bogstaver og tal. Der skal angive en længde for feltet, som er max antal tegn tilladt.	DATE	Dato
TIME	Tidspunkt	DATETIME	Dato og tidspunkt i et felt
TIMESTAMP	Tidsstempel med dato og tid for oprettelse af rækken	Boolean	Sandt eller falsk 1 eller 0

Følgende datatyper passer til den database vi er ved at designe:

Pizza	Ordrelinje	Ordre
Pizzanummer (INT) Pizzanavn (VARCHAR(20)) Pris (INT) Beskrivelse (VARCHAR(100)) Billede (VARCHAR(50))*	Ordrelinjenummer (INT) Pizzanummer (INT) Ordrenummer (INT)	Ordrenummer (INT) Navn (VARCHAR (50)) Adresse (VARCHAR (50)) Telefon (VARCHAR (10)) Leveringsdato (DATE) Leveringstidspunkt (TIME) Udbringning (BOOLEAN)

* Note: Billede er af typen varchar da feltet skal indeholde filnavnet på det billede der skal vises.

Nu er databasen designet og klar til at blive implementeret i det program i bruger til databaser.

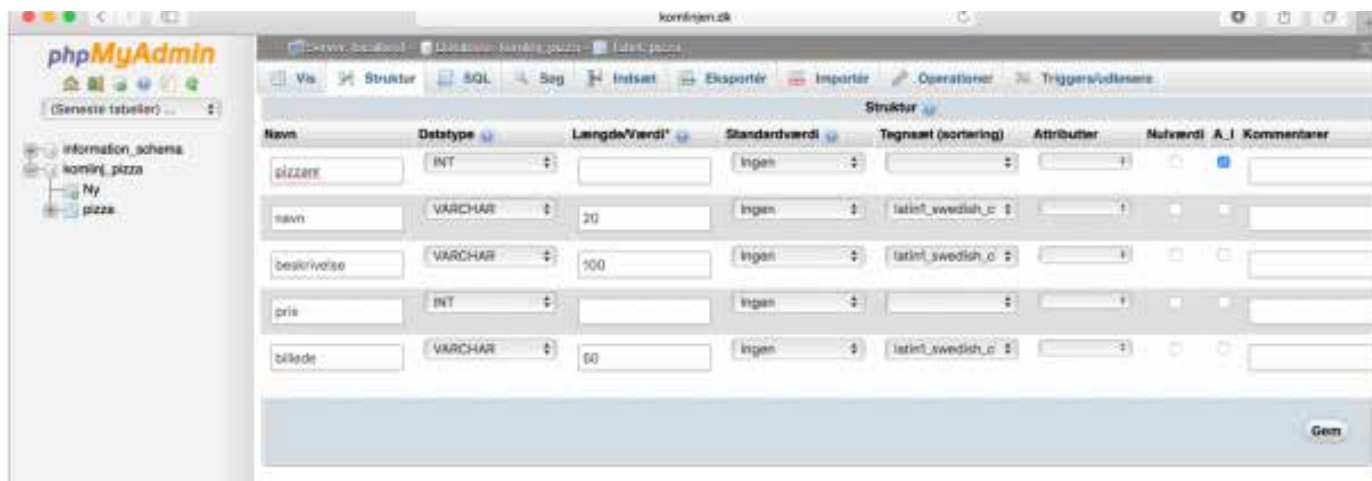
Følgende sider beskriver hvordan tabellerne oprettes i en MySQL database vha. phpMyAdmin.

Opret tabel:

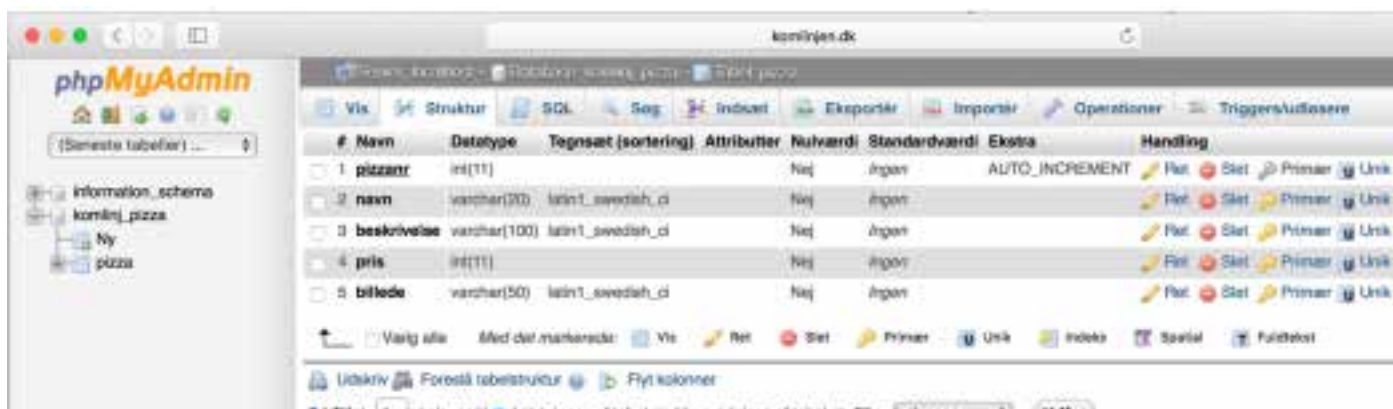
I en hel tom database skriver du navnet på tabellen og antallet af kolonner og klikker udfør. Hvis der allerede er tabeller i databasen klikker du på ny i databasetræet til venstre.



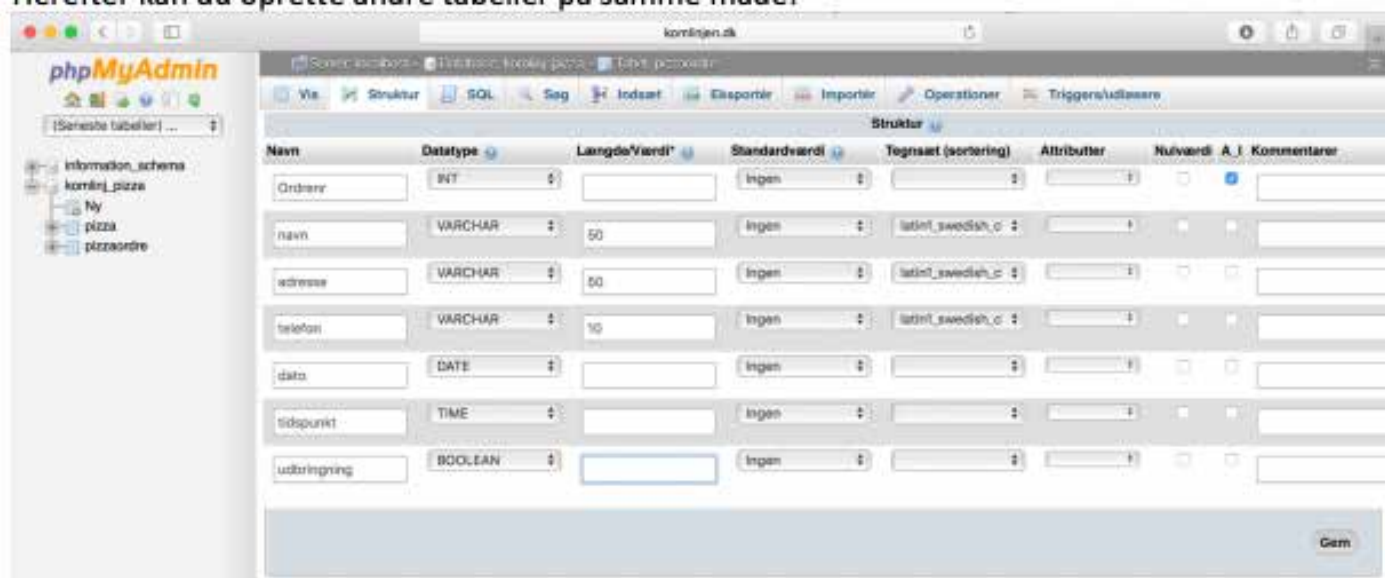
Herefter udfylder du felterne med udgangspunkt i de data og datatyper du har analyseret dig frem til der skal være. For at pizzanr skal være et autogenerated løbenummer skal der sættes et flueben i A_I. Klik til sidst på gem.



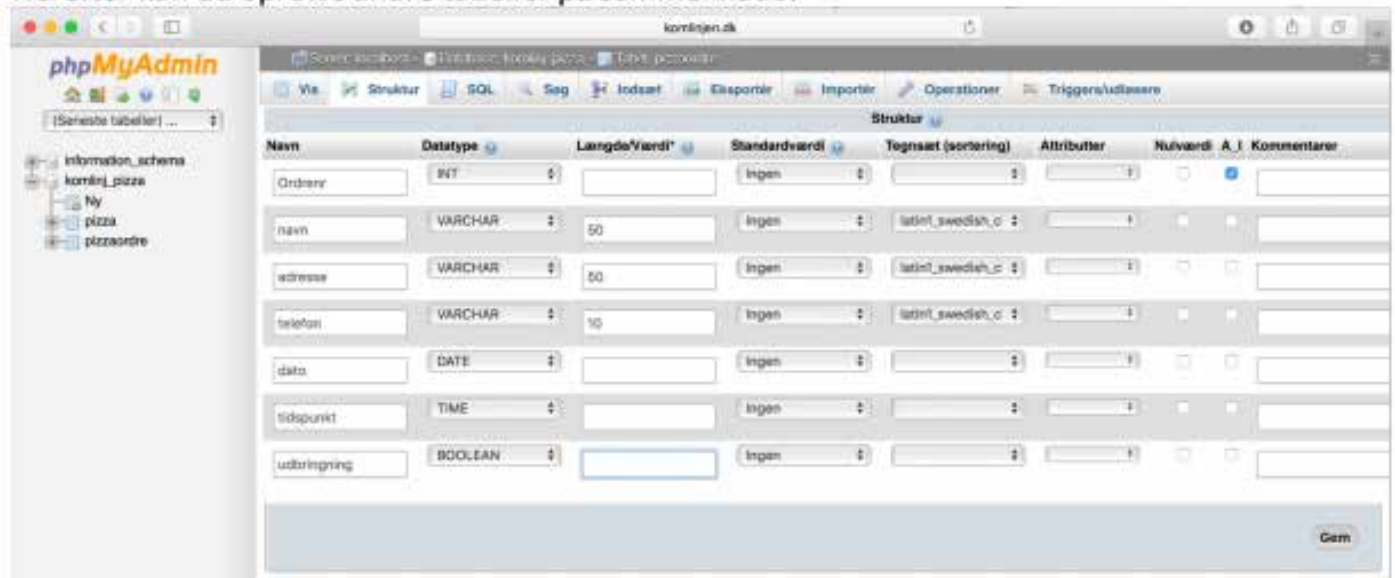
Herefter ser tabellens struktur således ud:



Herefter kan du oprette andre tabeller på samme måde:



Her kan man se oprettelse af tabeller på samme måde.



Nu er tabellerne oprettet og der kan indsættes data i dem ved at klikke på Indsæt i menuen øverst.

Bilag:

1) Note om interaktionsdesign vedlagt.