



BØRNE- OG
UNDERVISNINGSMINISTERIET
STYRELSEN FOR
UNDERVISNING OG KVALITET



Vejledning til Informatik C - forsøgslæreplan, hhx, htx, stx, hf

Januar 2026

Vejledning til Informatik C - forsøgslæreplan, hhx, htx, stx, hf
Januar 2026

2026

ISBN nr. [xxx xxx xxx] (web udgave)

Design: Center for Kommunikation og Presse

Denne publikation kan ikke bestilles.

Der henvises til webudgaven.

Publikationen kan hentes på:

www.uvm.dk

Børne- og Undervisningsministeriet

Departementet

Frederiksholms Kanal 21

1220 København K

Indhold

Indledning.....	5
1 Identitet og formål	6
1.1 Identitet.....	6
1.2 Formål.....	6
2 Faglige mål og fagligt indhold	8
2.1 Faglige mål	8
2.1.1 Digital myndiggørelse.....	9
2.1.2 Digital design og designprocesser.....	9
2.1.3 Computational tankegang	10
2.1.4 Teknologisk handleevne.....	10
2.2 Kerne stof	11
2.2.1 Digital myndiggørelse.....	11
2.2.2 Digital design og designprocesser.....	13
2.2.3 Computational tankegang	14
2.2.4 Teknologisk handleevne.....	16
2.3 Supplerende stof.....	17
2.4 Omfang	17
3 Tilrettelæggelse	19
3.1 Didaktiske principper.....	19
3.2 Arbejdsformer	21
3.3 It.....	21
3.4 Samspil med andre fag.....	22
4 Evaluerings.....	23
4.1 Løbende evaluering	23
4.2 Prøveform	23
4.3 Bedømmelseskriterier.....	24
4.3.1 Oversigt over karakterskalaen	24
4.4 Vejledende karakterbeskrivelse.....	25

Indledning

Vejledningen præciserer, kommenterer, uddyber og giver anbefalinger vedrørende udvalgte dele af læreplanens tekst, men indfører ikke nye bindende krav.

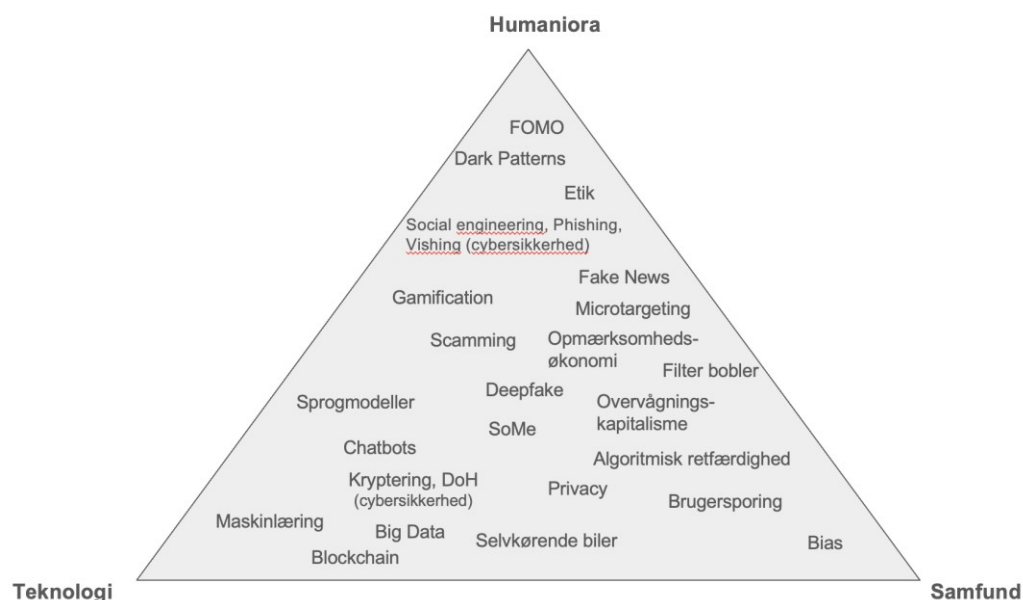
Citater fra læreplanen er anført i citationstegn.

Nærværende vejledning er gældende for informatik C forsøgslæreplanen (2025), som er generisk, fælles for hhx-, htx-, stx-, og hf-uddannelserne. Paradigmatiske eksempler og andre eksempler på uddannelsesstøttede undervisningsforløb, grundstof mm kan ses på fagets hjemmeside <https://informations-teknologi.wikidot.com> samt på <https://graspit.dk/category/informatik/>.

1 Identitet og formål

1.1 Identitet

Informatik består af emner inden for de teknisk-naturvidenskabelige, samfundsvidenskabelige og humanistiske områder, herunder udvikling og implementering af it-systemer. Informatik er tværvenskabelig og har ikke hovedvægt inden for et bestemt videnskabeligt hovedområde. Dette er illustreret i figur 1 herunder.



Figur 1: Informatik som et tværvenskabeligt fag.

1.2 Formål

I informatik skal eleverne opnå forståelse af informatikkens potentiale og rolle som katalysator for forandringer i samfundet med henblik på at styrke deres forudsætninger for at forstå og agere meningsfuldt i et demokratisk og digitalt samfund, herunder konstruktivt og kritisk at kunne medvirke til at forme også den digitaliserede virkelighed.

I informatik prioriteres det at fremme elevernes evne til at designe og realisere it-systemer, herunder teste og videreudvikle disse, og undervisningen har et særligt fokus på eleverne som kreative producenter.

Det er en væsentlig pointe, at undervisningen gennem valg af temaer/genstandsfelter knytter sammenhænge mellem faget og natur-, kultur-, praktiske- og/eller musiske fag og derigennem afspejler informatikkens rige relationer til og implikationer for alle andre fagfelter.

Blandt de faglige mål indgår, at eleverne skal kunne:

- behandle problemstillinger i samspil med andre fag
- demonstrere viden om fagets identitet og metoder
- løse et mindre problem ved at beskrive problemet, samt designe, realisere og afprøve et it-system gennem brugerorienterede teknikker.

Dette betyder, at der ikke alene skal undervises i informatik, men også om informatik. Når faget indgår i et samarbejde med andre fag, uanset om det er NV, SO, SOP, SRP eller, det er et fagligt samarbejde i

en studieretning, så skal eleverne kunne inddrage og anvende fagets relevante metoder, redegøre for disse i et sprog, som man også uden for faget kan forstå, samt forholde sig til fagets muligheder og begrænsninger i arbejdet med den konkrete problemstilling.

2 Faglige mål og fagligt indhold

2.1 Faglige mål

De faglige mål beskriver det, eleven skal kunne, og det er de faglige mål, eleven vurderes på ved løbende vurdering og eksamen. Kernestof og supplerende stof er det stof, vi gennemgår, for at eleverne når de faglige mål.

Generelt er både skaberkompetencer og fagstof til stede i alle de faglige mål. Desuden er der en tæt kobling mellem fagets identitet og formål på den ene side og de faglige mål på den anden side.

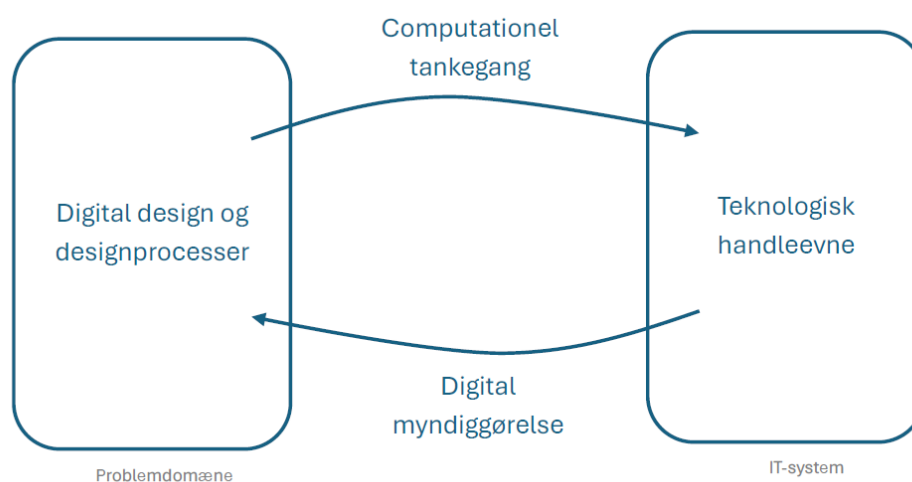
Når der i læreplanen omtales et it-system, menes et it-system i bredeste forstand. Her kan fx være tale om en app til en pizza-butik, en webshop, en førerløs bil (fx en robot, der kan køre og holde afstand) eller en helse-app. Filer eller dokumenter uden selvstændig, automatisk funktionalitet betragtes ikke som IT-systemer. Dette omfatter blandt andet billedfiler, rene HTML-filer og filer, der beskriver tredimensionelle figurer, uanset om disse bearbejdes eller manipuleres af eleven.

”løse et mindre problem ved at beskrive problemet, samt designe, realisere og afprøve et it-system gennem brugerorienterede teknikker”

Dette mål er overordnet for de andre faglige mål.

At skabe it-systemer er kernen i informatik-faget, og det er derfor naturligt, at undervisningen drejer sig om design af et it-system til løsning af et problem. Den proces eleven skal igennem for at designe it-systemet medtænker de andre faglige mål i informatik. De problemstillinger, som eleverne skal arbejde med, skal være relevante og kan beskrives af underviseren, eleverne selv eller en ekstern samarbejdspartner. Det kunne være webshops, apps til forskellige formål, robotter, reklamespil eller lignende. Processen består af forundersøgelse (beskrivelse af problemet), produktion (design og realisering) og afprøvning af en løsning. I den første fase findes der mange hensigtsmæssige metoder og redskaber, såsom problemformulering, problemtræ, observationer, interviews, personas og scenarier. Overvejelser om disse aspekter er nødvendige, før eleverne kaster sig ud i produktionsfasen. I forhold til afprøvning af løsningen refereres både til kvalitative og kvantitative tests, herunder fokusgrupper, interviews, spørgeskemaer, videoobservation, tænkehøjtmeter mv.

De faglige mål består overordnet af fire kompetenceområder: 1) Digital myndiggørelse, 2) Digital design og designprocesser, 3) Computational tankegang og 4) Teknologisk handleevne. Sammenhængen mellem disse fire kompetenceområder er illustreret i figur 2.



Figur 2:

Sammenhængen mellem de fire kompetenceområder, som tilsammen udgør de faglige mål.

I forbindelse med *konstruktion* (af dele) af et it-system viser kassen til venstre et problemdomæne for hvilket, der skal udvikles et it-system. Problemområdet analyseres gennem iterative designprocesser, brugerinddragelse og principper for interaktionsdesign (Digital design og designprocesser). Herefter undersøges det, hvordan det kommende it-system kan digitaliseres vha. datamodeller og algoritmer (Computational tankegang). Kassen til højre illustrerer implementering af it-systemet (Teknologisk handleevne). Dette indebærer den teknologiske implementering herunder programmering og valg af systemarkitektur. Pilen mod venstre viser, at det skal overvejes, hvilken indflydelse it-systemet har på det oprindelige problemdomæne (Digital myndiggørelse). På den måde kan de fire kompetenceområder ses som en naturlig del af en cyklus ifm. specificering og implementering af (dele af) et it-system.

I forbindelse med *afkodning* af (dele af) et eksisterende it-system, illustrerer kassen til venstre undersøgelse af et eksisterende design (Digital design og designprocesser), mens pilen til højre illustrerer en dekomposition af det betragtede it-systems data, datamodeller og algoritmer (Computational tankegang). I kassen til højre arbejdes der med it-systemets systemarkitektur og it-sikkerhed (Teknologisk handleevne). Endelig illustrerer pilen mod venstre, hvordan man med udgangspunkt i et eksisterende it-system kan beskrive, analysere og vurdere it-systemets oprindelige intention på den ene side og dets faktiske brug på den anden side, samt hvordan it-systemet har betydning for mennesker, natur eller samfund (Digital myndiggørelse).

Der er således en central dualitet indbygget i alle fire kompetenceområder: *Konstruktion* (eller udvikelse/modificering) (af dele) af et it-system på den ene side samt *afkodning* (af dele) af et eksisterende it-system på den anden.

2.1.1 Digital myndiggørelse

"Beskrive, analysere og vurdere it-systemers intentionalitet og anvendelse, samt deres etiske og sociale konsekvenser"

Eleverne skal kunne skelne mellem it-systemets tilsigtede formål og dets faktiske brug på baggrund af viden om kernestoffet og derigennem vurdere, hvordan forskelle mellem intention og anvendelse kan skabe nye betydninger, muligheder eller dilemmaer. Med intentionalitet forstås de holdninger, værdier og formål, som designere har indlejret i it-systemet. Alle it-systemer er skabt med den hensigt at påvirke brugere, handlinger eller oplevelser på bestemte måder. Gennem valg og fravalg i designprocessen har designere formet it-systemets egenskaber, og dermed også den måde, det inviterer brugeren til at handle på.

I undervisningen kan der arbejdes med, at eleven både skal kunne analysere et it-systems intentionalitet og dets anvendelse. Det vil sige afdække de hensigter og værdier, der ligger bag dets design og samtidig analysere, hvordan it-systemet faktisk anvendes.

"Give eksempler på hvordan eksisterende it-systemer og deres indhold har betydning for mennesker, natur eller samfund."

Eleverne skal kunne identificere udvalgte it-systemers muligheder og begrænsninger og give eksempler på hvilke tilsigtede og/eller utilsigtede konsekvenser et it-system kan have for mennesker, natur eller samfund.

I undervisningen arbejdes der med, at eleverne skal kunne undersøge konkrete it-systemer og identificere it-systemets betydning ud fra flere perspektiver fx etisk, socialt, teknisk, økonomisk, kulturelt eller miljømæssigt.

2.1.2 Digital design og designprocesser

"Undersøge et eksisterende design ud fra simple brugercentrerede analyser "

Eleverne skal kunne undersøge et eksisterende design og have blik for, hvad der kendetegner godt og dårligt design. Det indebærer, at eleverne skal kunne redegøre for, om udvalgte interaktionsdesignprincipperne er overholdt, og hvilken betydning det har for brugeroplevelsen, hvis det ikke er tilfældet.

”Konstruere et konkret it-system gennem iterative processer og brugerinddragelse samt redegøre for principper bag interaktionsdesign”

Eleverne skal selv kunne konstruere et it-system (eller dele heraf) i en iterativ arbejdsproces, hvor udarbejdede skitser og prototyper løbende evalueres med (eventuelt fiktive) brugere. Eleverne skal kunne redegøre for de designvalg, som er truffet i tilblivelsen af skitser og prototyper ved at anvende principper for interaktionsdesign.

2.1.3 Computational tankegang

Eleverne skal kende til datamodellering og til algoritmebegrebet samt kunne bruge disse til på den ene side afkodning og på den anden side specificering og implementering af it-systemer eller dele heraf.

Miniordbog

Computational tankegang er en samlet betegnelse for de processer, som indgår i at modellere et problem, så det kan behandles effektivt af en computer. Dette omfatter analyse, modellering og strukturering af data, datarepræsentationer og dataprocesser. Centrale begreber i den forbindelse er data, algoritmer, strukturering og modellering.

”Redegøre for et eksisterende it-systems data, datamodeller og algoritmer”

Eleverne skal kunne afkode hvilke typer af data, datamodeller og algoritmer, der anvendes i (dele af) et eksisterende it-system. Det vil sige systemets begrebsmæssige informationsgrundlag, herunder hvilke begreber systemet bygger på, og hvordan virkeligheden er forenklet og modelleret i systemet.

Eleverne skal kunne beskrive data i en datamodel, samt hvordan algoritmer opfører sig (fx beskrevet ved et rutediagram).

”Identificere potentialet til digitalisering af en problemstilling vha. datamodeller og algoritmer samt konstruere dele af et it-system, der anvender disse”

Eleverne skal – på basis af en overordnet systembeskrivelse – kunne specificere dele af et it-system i form af datamodeller for systemets informationsgrundlag og algoritmer for systemets opførsel i udvalgte brugssituationer.

Eleverne skal – på basis af en eksisterende specifikation (fx et rutediagram eller pseudokode) – kunne implementere dele af et it-system vha. konkrete digitale teknologier (programmeringssprog, evt. softwarebiblioteker m.m.).

2.1.4 Teknologisk handleevne

”Anvende simple strukturer i programmeringssprog og anvende programmering og fejlfinding til udvikling af simple it-systemer”

Eleverne skal kunne anvende simple strukturer i programmeringssprog som variabler, sekvenser, forgreninger, løkker og funktioner. På C-niveau er det nok, at eleverne kan anvende blokbaserede programmeringsværktøjer, og det er således ikke et krav, at eleverne lærer tekstbaseret programmering. Eleverne skal beherske det eller de valgte programmeringssprog på et sådant niveau, at de bliver i

stand til at udvikle simple it-systemer eller dele heraf. Desuden skal eleverne præsenteres for forskellige metoder til fejlfinding i egen og/eller andres kode.

”Redegøre for generelle principper bag it-systemers systemarkitektur”

Eleverne skal kunne redegøre for de overordnede principper for internettet herunder client-server arkitektur. Desuden skal eleverne kunne redegøre for, hvordan kommunikation på internettet kan beskyttes vha. kryptering.

”Redegøre for beskyttelse af egen digital identitet og egne data på internettet samt give eksempler på tekniske og menneskelige aspekter af it-sikkerhed.”

Eleverne skal kunne redegøre for, hvordan de beskytter deres egen digitale identitet og egne data på internettet ved blandt andet valg af gode passwords og 2-faktor godkendelse. Desuden skal eleverne kunne give eksempler på, hvilken data de selv eller andre deler på internettet, og hvordan disse data eventuelt kan misbruges af aktører med ondsindede hensigter. Eleverne skal også have forståelse for at it-sikkerhed både handler om teknologi og om de mennesker, som bruger teknologien.

2.2 Kernestof

Gennem arbejdet med kernestoffet udvikler eleverne forståelse for, at et it-systems betydning ikke alene ligger i dets tekniske funktioner, men også i de værdier, det repræsenterer, og de måder, det tages i brug på.

Faget skal tones i forhold til uddannelsen, og hvis faget indgår i en studieretning, tones der til denne. Når man vælger cases, vil det være oplagt at vælge emner, der skeler til den studieretning klassen har. Toningen kan være i en merkantil retning, af teknologisk, designorienteret, samfundsvidenskabelig eller naturvidenskabelig art.

2.2.1 Digital myndiggørelse

Kernestoffet omfatter analyse og vurdering af, hvordan it-systemer både er udtryk for bestemte intentioner og anvendes på måder, der kan afvige fra disse intentioner. Der kan arbejdes med begreber som formålsanalyse, brugsstudier, konsekvensvurdering og eventuelt redesign:

Miniordbog

Formålsanalyse: Analyse af et it-systems tiltænkte brug og indbyggede værdier, fx en MobilePay app, der har mulighed for betaling (som intentioneret brug) direkte mellem brugere (som indbygget værdi).

Brugsstudier: Undersøgelse af, hvordan et it-system anvendes og opleves af brugere i konkrete situationer, fx gennem observationer, interviews eller tænke-højt-tests.

Konsekvensvurdering: Vurdering af konsekvenser for mennesker, natur eller samfund på baggrund af teknologianalyse, formålsanalyse og brugsstudier herunder særligt forskelle mellem it-systemets intentioner og faktisk brug.

Redesign: Udvikling af alternativer til eksisterende it-systemer med afsæt i analyser og vurderinger, fx gennem skitser, mock-ups eller simple prototyper.

“it-systemers og menneskers gensidige påvirkning, herunder særligt etiske og sociale forhold ved brug af it-systemer og data”

I undervisningen kan der arbejdes med en lang række eksempler på, hvordan it-systemer på den ene side påvirker de mennesker, som bruger it-systemerne, og på den anden side hvordan mennesker påvirker selv samme it-systemer på forskellig vis.

Der kan fx arbejdes med, hvordan brugen af kunstige intelligens i forskellige sammenhænge påvirker mennesker. Omvendt, vil de menneskevalgte træningsdata, som den kunstige intelligens er trænet på i sidste ende påvirke den endelige model, som kommer til at udgøre den kunstige intelligens (fx dating-apps, hvor brugernes ændrede adfærd påvirker det datagrundlag, som match-algoritmen anvender). Der kan også arbejdes med placering af ansvar, hvis en chatbot giver et forkert svar, eller hvis en førerløs bil forulykker. Et konkret eksempel er trolley-dilemmaet, hvor en trolley kører, men hvor man har mulighed for at lave et sporskifte. Skiftes der ikke spor, kører trolleyen videre, hvilket resulterer i, at fem personer blive kørt ned. Træffes valget om sporskifte vil én person blive ofret til fordel for, at fem andre overlever.

For at kvalificere samtalen med eleverne om disse dilemmaer, kan det være hensigtsmæssigt at inddrage teorier som nytte- og pligtetik.

Privatliv eller privacy er et andet vigtigt begreb i forbindelse med it-systemers og menneskers gensidige påvirkning, specielt med henblik på sociale medier. Eleverne skal indse de utilsigtede konsekvenser ved, at de deler store mængder af data, herunder fotografier og holdninger til bestemte emner. Udover, at de kan komme til at fortryde det senere, skal de forstå, at deres data utilsigtet kan bruges til at påvirke dem til altid at træffe de samme typer valg i fremtiden.

De skal ydermere forstå at de annoncebaserede forretningsmodeller, der anvendes på sociale medier, har som mål at øge brugernes engagementstid på platformene så meget som muligt. Den øgede eksponering for reklamer og opslag fra andre brugere, kan forstærke synspunkter og skabe ekkokamre, hvor mangfoldigheden af holdninger, som brugerne udsættes for, er meget begrænset, hvilket resulterer i stærkt polariserede fællesskaber og en nedsat evne til kritisk tænkning. Dette er således eksempler på, hvordan it-systemer påvirker mennesker. Men påvirkningen kan også pege den anden vej, når brugere fx ikke længere vil bruge platforme, som udelukkende er annoncebaserede. Det kan påvirke it-systemerne på den måde, at nogle platforme tilbyder betalingsversioner, som er frie for reklamer, mens også helt nye firmaer vinder frem med det erklærede mål at være et socialt medie, som ikke har til hensigt at fastholde deres brugere i et endeløst scroll. Eller når apps som BeReal forsøger at lave et opgør med perfektionskulturen på fx Instagram ved at opfordre sine brugere til at dele autentiske hverdagsbilleder.

I diskussionen omkring it-systemers og menneskers gensidige påvirkning, kan der tages udgangspunkt i et væld af forskellige typer af it-systemer: Fx sociale medier, sundhedsapps, chatbots, webshops, digitale læringsplatforme, systemer i offentlig forvaltning, førerløse biler eller lignende for derigennem at analysere, hvordan sådanne it-systemer kan have etiske og sociale konsekvenser.

“it-systemers anvendelighed, intentionalitet og konsekvenser for mennesker, natur eller samfund.”

Der kan arbejdes med, hvordan designvalg indlejrer bestemte værdier og formål i systemer, og hvordan disse kommer til udtryk eller forandres i den faktiske brug.

I undervisningen kan der arbejdes med forskellen mellem et it-systems intentionalitet, og hvordan it-systemet reelt anvendes. Her kan der inddrages utilsigtede effekter, etiske og juridiske dilemmaer eller nye sociale og miljømæssige konsekvenser. Der kan arbejdes med dark patterns og med de utilsigtede konsekvenser af ellers velmenende designvalg, fx når anbefalingsalgoritmer, der skal gøre indhold relevant for brugeren, samtidig bidrager til dannelsen af filterbobler.

Der kan også konkret tages udgangspunkt i sociale medier, som ofte har til formål, at maksimere den tid brugeren er på mediet (intentionalitet). Her kan B. F. Skinners teori om belønningsmetoder (Skinner-boks), og B. J. Foggs teori om behavior design danne udgangspunkt for et teoretisk begrebsapparat, som kan bruges til at analysere, hvordan sociale medier næsten til perfektion kan fastholde brugerens opmærksomhed - herunder hvilke konsekvenser det får for det enkelte menneske, når sociale medier kan ende med at opleves som en decideret afhængighed. Herefter kan teorien om behavior design anvendes konstruktivt til en samtale om, hvad man kan skrue på, hvis man som bruger gerne vil minimere sin tid på et givent medie.

Der kan også arbejdes med intentionaliteten og konsekvensen af de "AI-bot venner", som flere sociale medier stiller til rådighed for deres brugere.

2.2.2 Digital design og designprocesser

"Principper for design af brugergrænseflader og tilhørende interaktion"

I undervisningen kan der fx arbejdes med Don Normans designprincipper, Gestaltlovene, Jakob Nielsens 10 heuristikker og/eller andre relevante designprincipper som grundlag for design. De forskellige principper for design kan både bruges til at forklare, hvorfor et eksisterende design enten er godt eller dårligt, men også til at konstruere elevernes egne brugergrænseflader i de it-systemer, som de udarbejder.

"Brugerundersøgelser til afdækning af brugeres krav til et it-system, gennem en iterativ udviklingsproces (f.eks. ved hjælp af mock-ups og prototyper) sammen med brugere"

Dette afsnit lægger op til tæt samarbejde med brugere. Dette er dog ikke altid praktisk muligt. Derfor kan der gøres brug af fiktive brugere.

Eleverne kan arbejde med udformning af personaer, brugsmønstre (use cases) og/eller user stories, som en metode til afdækning af krav til et it-system.

Herefter kan der arbejdes med forskellige typer af prototyper fra low fidelity til high fidelity. Det kan også være hensigtsmæssigt, at eleverne kender forskel på en horisontal og en vertikal prototype, hvor en horisontal prototype omfatter en stor del af hele brugergrænsefladen, men uden særlig meget underliggende kode og data. En sådan horisontal prototype kan både være low og high fidelity og kan tidligt i processen bruges i en evaluering med brugere. En vertikal prototype vil omvendt kun dække en begrænset del af brugergrænsefladen, til gengæld vil den tilhørende bagvedliggende kode og data stort set være fuldt implementeret. En sådan vertikal prototype kan udarbejdes på baggrund af et enkelt eller nogle få brugsmønstre og/eller user stories og kan på alle tidspunkter i udviklingsarbejdet danne udgangspunkt for en evaluering med brugere. Prototyperne kan suppleres med fremtidsscenarioer for at forstå brugen af dem.

Derudover kan der arbejdes med planlægning og gennemførelse af brugervenlighedstest eventuelt med efterfølgende kategorisering af de fundne brugervenlighedsproblemer (kosmetiske, alvorlige og kritiske problemer). Der kan også udarbejdes redesign forslag på baggrund af en gennemført brugervenlighedstest.

"Eksempel på model for designprocesser"

Double Diamond-modellen kan anvendes som udviklingsmodel i designprocesser. I Double Diamond modellen arbejdes der i 4 faser: Discover: Indsamling af indsigt om problemstillingen, fx gennem deskresearch, fieldresearch, problemtræ, brugerundersøgelser eller samtaler med domæneeksperter. Define: Afgrænsning og præcisering af problemstillingen, fx gennem udarbejdelse af personaer eller nutidsscenarioer, der tydeliggør brugernes behov og udfordringer. Develop: Idégenerering samt design, udarbejdelse og iterativ afprøvning af prototyper med brugere og med fokus på principper for interaktionsdesign. Deliver: Udvikling og præsentation af det færdige it-system eller løsningsforslag.

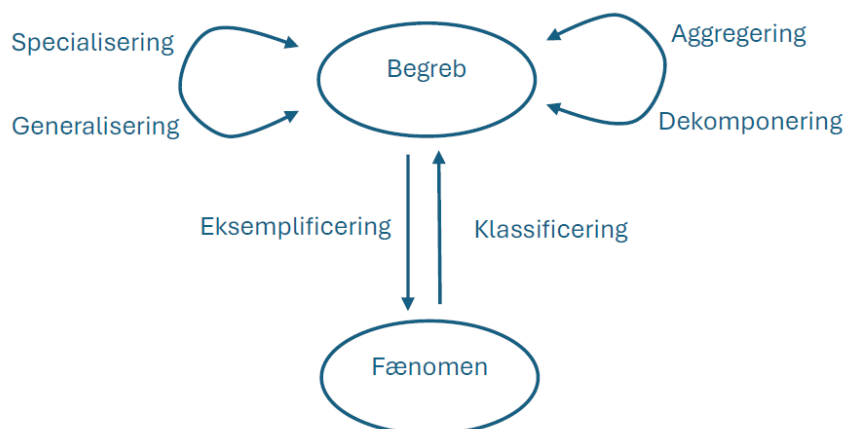
Et andet eksempel er stepwise improvement: I en iterativ udviklingsproces, hvor der arbejdes med at udvide en vertikal prototype ét brugsmønster (eller én user story) ad gangen, vil prototypen gradvist nærme sig det færdige produkt. Når eleverne på den måde tager udgangspunkt i et enkelt brugsmønster (eller user story) og arbejder med implementering af det, svarer det i stepwise improvement til en konkretisering (refinement). Implementering af det næste brugsmønster (eller user story) vil være en ny udvidelse (extension). På et tidspunkt, når prototype har fået en vis størrelse, kan der opstå behov for omstrukturering (restructure). Når eleverne arbejder på denne måde, vil de stort set hele tiden have funktionsdygtige prototyper (der kan evalueres med brugere), og som i sidste ende vil udvikle sig til det færdige it-system.

2.2.3 Computational tankegang

”Begrebs- og datamodeller, som eksempler på abstraktion og strukturering i forhold til fænomener i verden”

For at kunne lave en (computer)model af et domæne må man begrebsmodellere. En begrebsmodel er en overordnet beskrivelse af et domænes begreber og deres relation.

Æggestokmodellen (se figur 3) viser, at begreber generelt dannes på tre måder: klassificering/eksemplificering (begreb-fænomen), generalisering/specialisering (begreb-begreb) og aggregering/dekomponering (begreb-begreb). Sammenhænge mellem begreber kan desuden beskrives med såkaldte associationer.



Figur 3: Æggestokmodellen til beskrivelse af hvordan begreber dannes ved brug af abstraktion (klassificering, generalisering, aggregering) og konkretisering (eksemplificering, specialisering og dekomponering).

Eksempel

Hvis domænet er musik og musikafspilning kan man identificere begreber som sang/musikstykke, album, kunstner, komponist og playliste. Her vil eksempler på fænomener være "Let It Be", "Abbey Road", "Beatles", "John Lennon & Paul McCartney" og "The Beatles - Greatest Hits". Med hensyn til relationer mellem nogle af disse begreber er der eksempler på både aggregering/dekomponering (et album består af sange) og associationer (en komponist komponerer sange). Et eksempel på en generalisering af en sang/musikstykke kan være kunstværk, som yderligere vil kunne specialiseres i malerkunst og teaterstykke.

Begrebsmodeller er relevante mange steder i informatikken, både ifm. analyse- og designaktiviteter, og finder anvendelse inden for alle fire kompetenceområder af informatik. Som konkret eksempel kan nævnes design af brugergrænseflader, som jo skal afspejle begreber og fænomener, der er relevante for brugerne.

Når et domænes vigtigste begreber er forstået, vil der ofte være brug for at lave en egentlig datamodel, som beskriver, hvordan data mere konkret skal implementeres. Hvis der arbejdes med E/R-diagrammer, vil det være relevant, at eleverne kender til begreberne entitet, attribut, relation, kardinaliteter (1-1, 1-M og M-M), samt hvordan man kommer fra et E/R-diagram til tabeller i en database.

Der kan også arbejdes med digital teknologianalyse.

Miniordbog

Digital teknologianalyse: Analyse af et it-systems konstruktion, herunder input/output, sensorer, og it-systemets bagvedliggende data- og procesmodeller.

“Data og datatypers repræsentation og manipulation”

De fleste it-systemer er ikke meget værd uden data. Eleverne skal kende forskellige typer af data. Det kan være simple datatyper som tal, tekst og boolske værdier eller mere komplicerede datatyper som lister og dictionaries (objekter i JavaScript). Der skal arbejdes med manipulation af data, fx forskellen på at lægge to tal og to tekststreng sammen.

“Algoritmebegrebet og dets rolle i forbindelse med modellering af processer ”

Algoritmebegrebet har flere betydninger, men i denne kontekst forstås det som en utvetydig beskrivelse af, hvordan et problem kan løses. Det betyder, at algoritmen er formuleret i et sprog, der entydigt kan forstås af den eller det, som skal udføre den. En algoritme fungerer således som en opskrift, der angiver, hvordan man gennem et antal trin kan nå frem til en løsning.

Algoritmebegrebet kan perspektiveres ved hjælp af hverdagseksempler på algoritmer og ikke-algoritmer, og det anbefales at lade eleverne se flere eksempler på algoritmer med henblik på modellering af processer og derefter lade dem modificere disse algoritmer eller udarbejde lignende eller helt nye algoritmer.

Eleverne skal se eksempler på beskrivelse af algoritmer, fx grafiske fremstillinger som rutediagrammer eller tekstuelle beskrivelser som pseudokode, hvor de algoritmiske strukturer er tydelige, herunder sekvens, selektion og iteration, og samt eventuelt brug af under- eller delalgoritmer. Eleverne kan for eksempel få udleveret en algoritme beskrevet ved et rutediagram, som efterfølgende skal oversættes til pseudokode/kode eller omvendt.

Det er væsentligt at illustrere brugen af algoritmer både ifm. afkodning af (dele af) et eksisterende it-system og ifm. specificering og implementering af dele af et it-system.

Eleverne kan i den forbindelse arbejde videre på eksemplariske eksempler. Det kan fx gøres ved, at eleverne får en algoritme (evt. beskrevet vha. et rutediagram), som kan beregne et BMI og efterfølgende klassificere om det beregnede BMI er i kategorien “normalvægtig”. Herefter kan eleverne udvide algoritmen, så den kan bruges til at afgøre om BMI falder inden for andre kategorier (fx “undervægtig”, “overvægtig” og “svær overvægtig”). Slutteligt kan eleverne på baggrund af det udarbejdede rutediagram selv implementere et it-system, som kan bruges til beregning og klassificering af BMI.

Hvis det af den ene eller den anden grund er uhensigtsmæssigt at arbejde med BMI, kan et tilsvarende eksempel omhandle en webshop, hvor der er gratis fragt, hvis kunden eksempelvis køber for mere end 500 kr. eller, hvor der fx gives 10% rabat, hvis kunden har tilmeldt sig et nyhedsbrev (eller en gradueret rabat, som gives alt efter hvor mange produkter af den samme type, som kunden har købt).

Eleverne kan også præsenteres for en hjemmeside til udregning af told, moms og importgebyr ved handel med lande uden for EU. På baggrund heraf kan eleverne afkode den bagvedliggende algoritme og fx opstille et tilhørende rutediagram.

”Simple metoder til maskinlæring”

Eleverne skal kende de forskellige trin, der indgår i en maskinlæringsproces. Det er her oplagt at behandle maskinlæringsmodeller til klassifikation. Fx k-nærmeste naboer, klassifikationstræer eller kunstige neuroner. Det er ikke meningen, at eleverne skal kunne redegøre for modellerne, men de skal kende hovedprincipperne i deres tilblivelse, herunder forstå, hvad der menes med trænings- og test-data. Desuden skal eleverne kunne træne modeller på udvalgte datasæt og foretage en overordnet vurdering af, hvor god modellen er til at klassificere. Dette kan fx ske ved anvendelse af værktøjer som Orange, maskinlæringspakker til Python, Google Teachable Machine eller Micro:bit Create AI.

”Databasers anvendelse og simple databaseforespørgsler”

Eleverne skal have kendskab til databaser samt simple forespørgsler som tabelsopslag, indsætte, ændre, slette poster og oprettelse af tabeller (fx i SQL: SELECT.. FROM.. WHERE, INSERT INTO, UPDATE, DELETE samt CREATE TABLE). Hvis man benytter App Inventor eller Python er der mulighed for at forbinde til en MySQL database ved at opsætte en server med værktøjet XAMPP. Hvis man bruger App Lab, er dette ikke en mulighed. Her kan man i stedet bruge et værktøj som DB Browser for SQLite til at lave forespørgsler. Det er også en mulighed at arbejde med oprettelse af tabeller i SQLite, som efterfølgende importeres til datalaget i App Lab.

For at der i denne kontekst er tale om en del af et it-system, skal der som minimum være en datamodel (fx et E/R-diagram) og en teknisk forespørgsel, der følger datamodellen (fx en SQL-forespørgsel).

2.2.4 Teknologisk handleevene

”Basale strukturer i programmeringssprog, herunder variabler, sekvens, forgrening, løkker og funktioner”

At lære at programmere fra bunden i et sprog som fx Python kan være en tidskrævende affære. Derfor kan det give god mening at anvende visuelle programmeringsværktøjer (fx Scratch, App Lab eller App Inventor), da eleverne herved fint kan få en forståelse for de centrale programmeringsbegreber såsom variable, sekvens, forgrening, løkker og funktioner.

Hvis man har robotter (fx Spike Prime) eller mikrocontrollere (fx Micro:bit) til rådighed, er det også oplagt at lade eleverne arbejde med programmering af disse.

Der skal vælges et programmeringssprog, der indeholder variable, sekvenser, løkker, forgreninger og funktioner. HTML og CSS indeholder ikke disse strukturer. Til eksamen bedømmes, i hvilken grad eleven opfylder de faglige mål, og her indgår kodning med HTML og CSS ikke. Gode valg af programmeringssprog er Scratch, hvis eleverne fx skal programmere læringsspil eller Spike Prime-robotter. Skal eleverne arbejde med app-udvikling er App Inventor eller App Lab gode bud på programmeringsværktøjer.

Afhængig af elevernes forudsætninger (hvis eleverne allerede kan programmere eller går på en studieretning med faget programmering), kan de prøve kræfter med et tekstbaseret sprog som fx Python eller p5.js.

“Programmering og fejlfinding”

Programmering er meget mere end at kunne kode i et programmeringssprog. For at kunne programmere skal man til en vis grad mestre det programmeringssprog, som man skal udtrykke sig i, men endnu vigtigere er det at kunne “problemløse”, når koden ikke virker efter hensigten.

Det er vigtigt, at eleverne får en oplevelse af, at fejl er en naturlig og nødvendig del af det at programmere, og at det i sidste ende leder til bedre kode.

“Sikre kommunikationsformer, overordnede principper for internettet og client-server arkitektur”

Grundlæggende principper for internettet behandles, så eleverne forstår de vigtigste aspekter i kommunikationen mellem klient og server herunder betydningen af IP-adresser. Eleverne skal kende til nødvendigheden af kryptering for at sikre kommunikation på nettet - herunder forskellen på symmetrisk og asymmetrisk kryptering.

“Beskyttelse af egen digital identitet og data”

Eleverne skal lære, at beskyttelsen af egen data både i en personlig og professionel sammenhæng er tæt forbundet med sikringen af individers og organisationers sikkerhed. Det er af enorm betydning i det moderne samfund, hvor de fleste data nu findes i digital form, og hvor stadig flere interaktioner foregår gennem digitale enheder.

I forbindelse med IT-sikkerhed anvendes begrebet security, som handler om beskyttelse mod bevidste, ondsindede handlinger – altså trusler, der skyldes mennesker eller organisationer, der forsøger at forårsage skade. Eleverne skal lære at formålet med security er at forhindre angreb, uautoriseret adgang, datatyveri, sabotage osv. Det kan gøres gennem beskyttelse mod hacking, phishing og malware. Derudover er adgangskontrol fx login, to-faktor-godkendelse og kryptering af data vigtige redskaber. Eleverne skal kende deres egen rolle i forbindelse med beskyttelse af egen digitale identitet og data (awareness). Som eksempel kan der konkret arbejdes med en simpel app, som kan bruges til at afgøre om et indtastet password er godt (fx om det har en passende længde og evt. også om det indeholder store bogstaver og/eller specialtegn).

2.3 Supplerende stof

Det supplerende stof er i høj grad med til at tænde faget i forhold til uddannelsen, og hvis faget indgår i en studieretning, tones der til denne. Her vil det også være oplagt at vælge emner, der skeler til den studieretning klassen har. Toningen kan være i en merkantil retning, af teknologisk, designorienteret, samfundsvidenskabelig eller naturvidenskabelig art.

2.4 Omfang

“Det forventede omfang af fagligt stof er normalt svarende til 90-150 sider.”

Fagligt stof i faget omfatter fx netbaserede tutorials (herunder video-tutorials), netbaserede udviklingsværktøjer, -biblioteksmoduler, -dokumentationer og vejledninger, podcasts, dokumentarer, i- og e-bøger og traditionelle undervisningsmaterialer i form af bøger, udleveret tekstmateriale m.m.

Omfanget af fagligt stof anføres i beskrivelsen af den gennemførte undervisning (undervisningsbeskrivelsen), der færdigredigeres ved afslutningen af undervisningen i det enkelte fag. Omfanget angives normalt med en sådan detaljeringsgrad, så det af undervisningsbeskrivelsen fremgår, hvorledes det faglige stof har været vægtet i undervisningsforløbet. Dette gøres ved at angive et skønsmæssigt sidetal. Opgivelsen af omfang har til formål at sikre den faglige kvalitet, så eleverne når et passende fagligt niveau uden at blive fagligt overbelastet. Der kan være stor forskel på sværhedsgraden af materialerne. Derfor er der tale om en kvalificeret vurdering på baggrund af omfang og sværhedsgrad, når sidetal optælles. Er der store niveauforskelle i klassen, er det muligt at give ekstra materialer til de elever, der udviser særlig talent eller overskud.

Det kan være en god øvelse at overveje forberedelsestiden til et 2 siders dokument med tekst, koder, modeller osv. sammenholdt med en forberedelse af eksempelvis en video-tutorial over samme tema og faglige indhold.

3 Tilrettelæggelse

3.1 Didaktiske principper

“Undervisningen består af et antal aktuelle virkelighedsnære cases, som er motiverende for eleverne, og som tager afsæt i alle fire kompetenceområder – design- og designprocesser, computationel tænkning, digital myndiggørelse, og teknologisk handleevne, så fagets centrale principper, tænkemåder, udtryksformer, arbejdsformer og implikationer såvel konstruktivt som analytisk fremstår klart.”

For at undervisningen skal være vedkommende, kan man således gøre brug af det didaktiske princip “Be application-oriented”, som sigter på, at de it-systemer, som eleverne kommer til at arbejde med i undervisningen, skal være velkendte og meningsfyldte for eleverne. Det kan eksempelvis give mening og være motiverende at arbejde med cases som light udgaver af skoleadministrationsplatforme, sociale medier og/eller musiktjenester, som eleverne allerede kender og anvender på daglig basis. Andre eksempler på virkelighedsnære cases, som eleverne kan arbejde med, er førerløse biler, lærings- eller reklamespil og sundhedapps.

Som et konkret eksempel på en virkelighedsnær case, kan der arbejdes med en light udgave af Instagram. Her kan eleverne anvende principper for design til dels at analysere Instagrams brugergrænseflade og dels til selv at designe deres egen light udgave af Instagram. Det vil i den forbindelse også give mening, at arbejde med personaer, brugsmønstre eller user stories og senere brugervenligheds-evaluering (Digital design og designprocesser). Eleverne kan dernæst arbejde med Instagrams bagvedliggende datamodel og selv lave en simpel udgave af en Instagram database (Computationel tankegang). Herefter kan eleverne implementere en Instagram app i fx App Lab evt. med en simpel loginside (Teknologisk handleevne). Endelig kan man i undervisningen arbejde med, hvordan og hvorfor platforme som Instagram forsøger at fastholde sine brugere på mediet så længe som muligt. Ligesom der fx kan arbejdes med intentionen bag like-knappen og hvilke konsekvenser, det har fået for brugerne (Digital myndiggørelse).

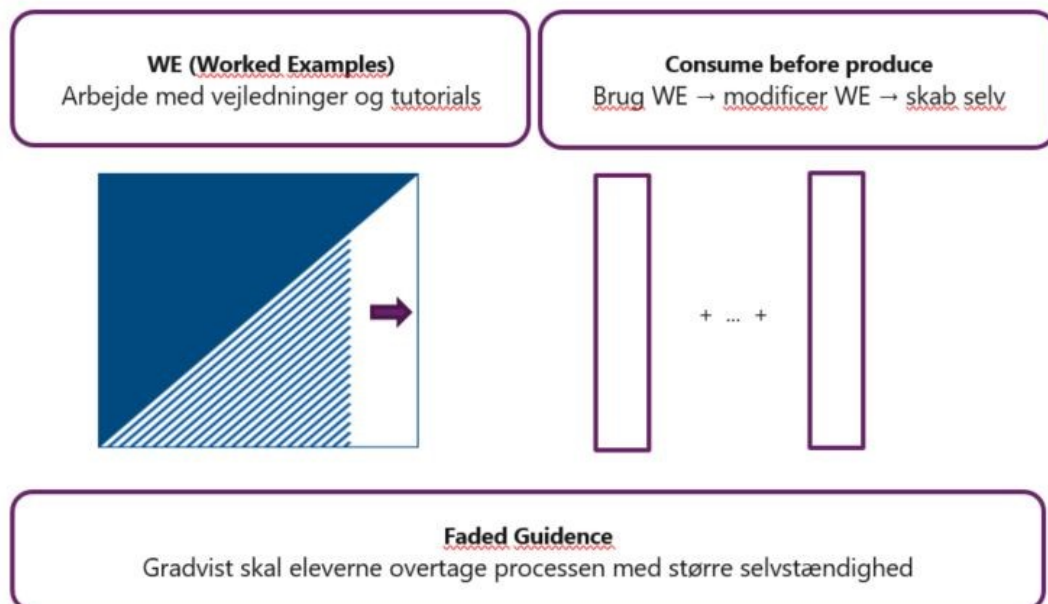
Det er ikke et krav, at alle fire kompetenceområder dækkes i hvert enkelt forløb, men i stedet at eleverne præsenteres for virkelighedsnære cases - evt. delt ud på flere forskellige forløb - hvor alle fire kompetenceområder behandles for hver case, som det er eksemplificeret ovenfor med Instagram. Det anbefales at de enkelte forløb dækker flere faglige mål.

“ Undervisningen tilrettelægges ved brug af anerkendte didaktiske principper, herunder ‘use-modify-create’-progression fra at anvende udleverede it-systemer til at modificere disse, for til sidst selvstændigt at skabe (nye dele af) it-systemer. ‘Stepwise Improvement’ anvendes generelt som metode til trinvis, iterativ og systematisk udvikling af it-systemer herunder brugerinddragende designprocesser som led i udviklingen af meningsfulde og brugbare løsninger.

‘Worked Examples’ kombineret med ‘Faded Guidance’ bruges til illustration af eksemplariske analyse- og løsningsprocesser.”

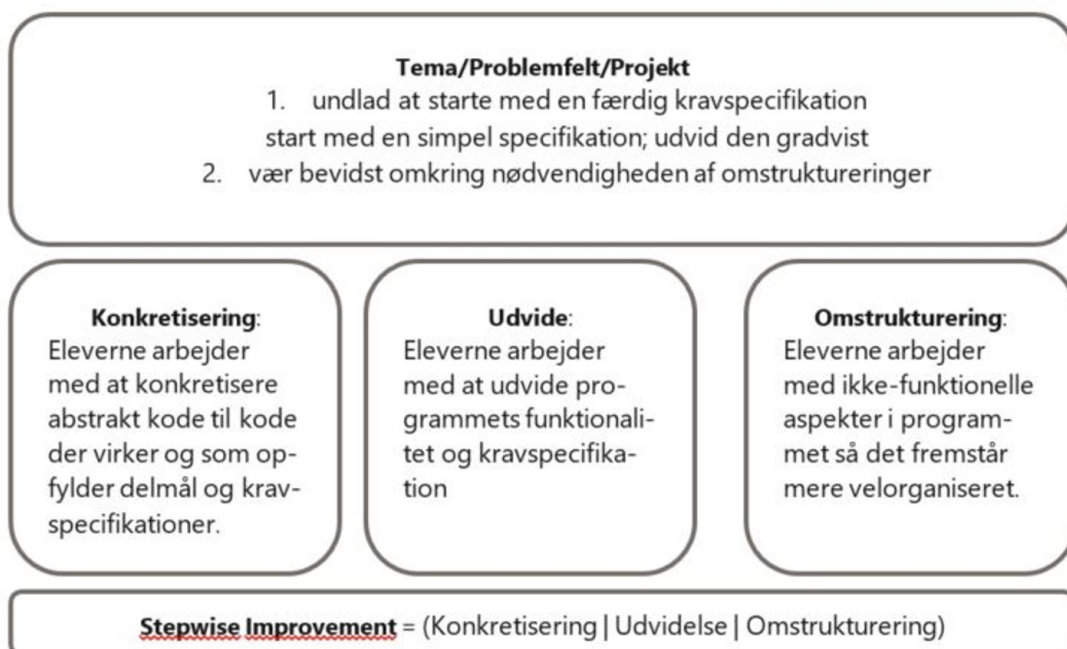
Den bedste stilladsering opnås ved, at eleverne bruger de eksemplariske ‘worked examples’, arbejder videre på dem og i sidste ende selv bliver i stand til selv at skabe nyt (se figur 4). På denne måde kan det også sikres, at eleverne får et overblik over og en delvis forståelse af hele strukturen af systemet.

For alle projektforsløb gælder, at selve processen med fordel kan brydes ned i flere enkeltelementer, i starten med en høj grad af lærerstyrede elevarbejder med gennemprøvede eksempler (vejledninger, tutorials m.m.), og worked examples (WE), og senere skal eleverne gradvist overtage processen med større grad af selvstændighed (faded guidance).



Figur 4: Worked Examples, Use-Modify-Create og Faded Guidance.

Som tidligere nævnt er stepwise improvement (se figur 5) et eksempel på både en didaktisk- og metodisk tilgang til arbejdet med programmering og udvikling af it-systemer generelt. Modellen kan bruges som et planlægningsværktøj til, hvordan man kommer fra A til B til C, og som så-
dan er modellen ret lavpraktisk. I stedet for at gå mere eller mindre tilfældigt frem mod et færdigt pro-
gram, kan eleverne bevæge sig mere systematisk i de 3 dimensioner: Konkretisering, udvidelse og om-
strukturering, som det er beskrevet i afsnit 2.2.2.



Figur 5: Stepwise improvement.

Disse didaktiske principper er også velegnede for elever, der ikke har programmeringsmæssige færdigheder med sig fra grundskolen, og understøtter dermed overgangen fra grundskole til gymnasium.

”Undervisningsformen differentieres således, at alle elever udvikler sig i undervisningsforløbet.”

Erfaringsmæssigt har eleverne vidt forskellige it - og især programmeringsmæssige forudsætninger ved starten af forløbet, og undervisningsdifferentiering er et vigtigt redskab til at fastholde en tilstrækkelig individuel progression.

Differentieringen kan eksempelvis ske gennem udstrakt inddragelse af eleverne i undervisningen gennem valg af problemstillinger, opgaver, eksempler, elevoplæg mv.

3.2 Arbejdsformer

Den enkelte elev dokumenterer løbende sin faglige udvikling i en logbog. Det betyder, at eleven skal have et sted at gemme forskellige former for dokumentation fx it-systemer, noter, synopser, journaler, programbeskrivelser og rapporter. Det kan fx gøres i Teams, OneNote, eller via skolens egne it-systemer.

Skriftlighed og mundtlighed i faget

Der kan arbejdes med skriftlighed i faget i mange dele under projektarbejdsformen. Det kan være når eleverne udarbejder brugsmønstre eller user stories, rutediagrammer, begrebsmodeller, E/R-diagrammer, designer brugergrænseflade eller tilrettelægger brugervenlighedstests. Man kan også træne eleverne i at kommentere deres programmer og navngive variable og metoder med passende navne. Der er krav om, at eleverne løbende dokumenterer deres faglige udvikling i logbogen, og her bliver den skriftlige del af faget også tilgodeset. Den skriftlige del af faget bliver også medtænkt, hvis eleverne laver rapporter i forbindelse med projekterne. For at sikre progression, kan det være en god idé at lade eleverne læse og kommentere de andre gruppers arbejder.

Det mundtlige aspekt er sidestillet med det skriftlige, dels fordi den mundtlige præstation indgår som en del af bedømmelsesgrundlaget til eksamen, og dels fordi italesættelsen af fagets mange elementer er vigtig. Det mundtlige aspekt kan trænes ved at lade eleverne fortælle hinanden om deres it-system, om udvalgte kodedele, brugergrænseflade og/eller datalag og i det hele taget om alle dele af projektudviklingen.

Karrierekompetencer

Elevernes karrierekompetencer, og udvikling af digital myndighed kan styrkes ved at lade eleverne overveje, hvordan de enkelte forløb og projekter kan bruges i et fremtidigt job og i deres personlige liv. Såfremt virksomhedsbesøg eller eksterne oplæg på skolen er en mulighed, kan oplægsholderne opfordres til at redegøre for deres karrierevej og de erfaringer, der har bragt dem til deres nuværende job.

Globale kompetencer

Man kan arbejde med elevernes globale kompetencer i informatikundervisningen på mange forskellige måder. Man kan fx lade eleverne reflektere over it-systemers betydning for det globale samfund. Ligeledes kan man lade eleverne arbejde med hvordan it-systemer, kan løse problemstillinger relevante for de globale udfordringer, vi står over for. Det er her oplagt at inddrage aktuelle begivenheder, hvor det giver mening.

3.3 It

Det er vigtigt at have en åben dialog med eleverne om evt. brug af generativ kunstig intelligens og understøtte dem til hensigtsmæssig brug heraf, så der forekommer læring. Dette kan fx være brug af generativ kunstig intelligens som redskab til at forstå kode, at debugge kode, at sparre med om emner for at øge forståelsen eller at få hjælp til at løse en opgave, uden at svaret bliver genereret. Ved anvendelse af værktøjer, som ikke understøtter drag-and-drop-baseret udvikling af grafiske brugergrænseflader, kan eleverne udarbejde en skitse af den ønskede grafiske brugergrænseflade og anvende en chatbot til generering af den kode, der hører til den grafiske brugergrænseflade. Prompten skal her tydeligt afgrænse opgaven til udelukkende at omfatte den grafiske brugergrænseflade og ikke programlogik. Ge-

nerativ kunstig intelligens kan også bruges til at opnå domænekendskab ved forundersøgelser til databasemodellering eller udvikling af interaktionsdesign. I interaktionsdesign kan man lade eleverne udvikle en persona og derefter foretage et interview med personaen via en chatbot, som simulerer personaens perspektiv. Det anbefales desuden, at eleverne reflekterer over, hvad man opnår ved brug af disse værktøjer, samt at eleverne er klar over, hvad der ikke er hensigtsmæssig brug af disse værktøjer. Det er fx ikke hensigtsmæssig brug at generere hele eller dele af opgavebesvarelser eller at programmere med autokorrektur med indbygget generativ kunstig intelligens som fx Github Copilot. Det anbefales derfor at deaktivere disse værktøjer, hvis de er indbygget i udviklingsmiljøet.

Der gælder ved brug af disse værktøjer (akkurat som når eleverne finder kodestumper på nettet), at eleverne skal sætte sig grundigt ind i den kode, de gør brug af, ligesom de skal kunne kildehenvise.

3.4 Samspil med andre fag

”Dele af kernestof og supplerende stof skal vælges og behandles, så det kan bidrage til styrkelse af det faglige samspil mellem fagene og i studieretningen. I tilrettelæggelsen af undervisningen inddrages elevernes viden og kompetencer fra andre fag, som eleverne hver især har, så de bidrager til perspektivering af temaer og belysning af fagets almindelige sider. Når faget indgår i flerfaglige forløb, lægges der vægt på, at eleverne får mulighed for løbende at reflektere over, hvordan deres valg og behandling af viden og metoder fra de indgående fag påvirker kvaliteten af den flerfaglige problemløsning.”

Det er et krav, at dele af kernestoffet og det supplerende stof skal vælges og behandles, så det kan bidrage til at styrke det faglige samspil mellem fagene og i studieretningen. Alt efter hvilken uddannelse og studieretning faget indgår i, kan det stof, der behandles, udvælges så det tones til uddannelsen og understøtter studieretningen.

Samspilsdimensionen er meget vigtig, bl.a. fordi det indgår i fagets faglige mål og fordrer en kvalificering af elevernes innovationskompetence. Det er vigtigt at pointere forskellen mellem C-niveau og B-niveau. På C-niveau er de omtalte problemstillinger af ”enkel” karakter. På B-niveau kan der være tale om mere komplicerede problemstillinger, der ligger til grund for samspillet. Løsningen af en problemstilling kan gøres på mange forskellige måder, og her er listet nogle forskellige tilgange til samspilsdimensionen, uanset hvilken uddannelse faget måtte indgå i.

Analyserende: I faget er der mange forskellige værktøjer til at analysere en given problemstilling. Eksempelvis kan et problemdomæne være givet i et andet fag. Her kan user stories eller brugsmønstre danne udgangspunkt for en udviklingsproces, hvor man efterfølgende forsøger at beskrive, hvorledes problemet kan løses på design-, data- og/eller software niveau (eksempelvis beskrevet diagrammatiske).

Designende: I faget er det muligt at tale om, hvordan brugerens oplevelse af softwaren bedst inddrages, når systemet skal konstrueres. Brugerinteraktion og interaktionsdesign kan på denne måde fungere som samspilselementer, hvor informatik enten bidrager til andre fags indhold eller drager nytte af de metoder, eleverne anvender i disse.

Konstruerende: Faget giver nogle tekniske færdigheder, der gør eleven i stand til at producere forskellige produkter. Faget kan i en samspilssituation ses som værende produktudviklende og på den måde kan det indgå i en situation, hvor et andet fag analyserer produktet og evt. forsøger at markedsføre det. Et andet fag kan også bidrage med genstandsfeltet.

De analyserende og designende tilgange kan være en fordel, da de åbner mulighed for analyse og vurdering.

I det faglige samspil mellem fagene kan begrebsmodeller bringes i spil, da de er relevante inden for alle andre fagområder, så de med fordel kan bruges erkendelsesmæssigt til at få styr på et fagområdes begreber og fænomener (se figur 3).

4 Evaluering

4.1 Løbende evaluering

Som beskrevet i afsnittet om arbejdsformer skal eleven dokumentere sin faglige udvikling i sin logbog. Denne logbog anvendes løbende i forbindelse med elevens selvevaluering og evalueringssamtaler med læreren. Ved afslutning af et undervisningsforløb kan elevernes præstationer evalueres, blandt andet via logbogen, og det er en god ide at evalueringen giver en individuel vurdering af niveau og udvikling i det faglige standpunkt i forhold til den forventede udvikling og de faglige mål.

4.2 Prøveform

”Der afholdes en mundtlig prøve på grundlag af en opgave, hvor der skal udvikles et it-system eller en del af et sådant. Opgaverne fordeles ved lodtrækning. Tidskrævende dele af opgaven løses kun i skitseform.”

”Eksaminationstiden er ca. 24 minutter. Der gives ca. 24 timers forberedelsestid, dog ikke mindre end 24 timer, hvor eksaminanderne arbejder i grupper på to til tre.

Hvor dette ikke er muligt eller ønskeligt, kan man lade eksaminanderne arbejde individuelt.”

Hver gruppe udgør således en trækningssenhed ved lodtrækningen af opgaverne. Antallet af trækningssmuligheder skal således tage hensyn til antal grupper, at den enkelte opgave højst må anvendes to gange på samme hold samt kravet om at antallet af trækningssmuligheder skal overstige antallet af eksaminander/grupper med mindst 3. Eftersom eleverne i en gruppe eksamineres i den samme eksamensopgave, bør skolen sikre, at elever, der er blevet eksamineret, og elever, der venter på at blive eksamineret, ikke kan kommunikere sammen. Det kan fx ske ved, at eksaminander, der endnu ikke er blevet eksamineret, venter i et rum uden kommunikationsudstyr.

”Opgaverne, der indgår som grundlag for prøven, skal tilsammen dække de faglige mål. Den enkelte opgave må højst anvendes to gange på samme hold”

Den enkelte opgave skal være så bredt formuleret, at den giver eksaminanden mulighed for at inddrage flere faglige mål i løsningen. Det er ikke et krav at eksaminanden udarbejder et komplet implementeret it-system, men præsentationen af opgaveløsningen skal demonstrere eksaminandens overblik over hele løsningen med inddragelse af relevante metoder og fagtermer. Det er en god ide at en mindre del af it-systemet udarbejdes i detaljer – såvel i den skriftlige dokumentation som i det udarbejdede it-system. Hvis opgaven løses i grupper, kan opgaveløsningen involvere en større problemstilling, være mere detaljeret og/eller indeholde flere detaljløsninger. Udtryk som ”du kan eventuelt komme ind på...” eller ”hvis der er tid, kan du...” skal undgås, fordi de skaber usikkerhed om, hvornår opgaven er fuldt besvaret.

Det skal fremgå af eksamensopgaverne, hvilke faglige mål de retter sig imod. Det kan eksempelvis ske ved at inddrage ord og formuleringer fra de faglige mål eller ved at notere de relevante faglige mål nedst på opgaven.

Eksamensopgaverne skal afspejle den tematiske undervisningsform som eleverne kender fra undervisningen, derfor er det en god ide at eksamensopgaverne opbygges tematisk og baseres på flere faglige mål.

”Eksaminationen er individuel. Eksaminationen indledes med eksaminandens præsentation af opgaveløsningen og former sig derefter som en samtale mellem eksaminand og eksaminator med udgangspunkt i opgaven.”

Eksaminationen består af to dele:

En præsentation af opgaveløsningen. Eksaminanden præsenterer det udviklede it-system/opgaveløsningen. Det er en stor fordel, hvis eksaminanden forbereder præsentationen, herunder dokumentation m.m. løbende under forberedelsen. Det er en god ide at eksaminanden selv har initiativet ved eksaminationens start, så eksaminanden får plads til selvstændigt at strukturere sin præsentation af opgaveløsningen. Præsentationen har udgangspunkt i det, under forberedelsestiden, fremstillede it-system, og i mindre grad i teoretiske præsentationer (fx PowerPoint).

En samtale, hvor også faglige elementer fra undervisningen, ud over hvad der indgår i opgaveløsningen, inddrages. I samtalen kan opgaveløsningen sættes i relation til eksaminandens øvrige it-produkter. Den efterfølgende samtale vil typisk handle om at gå i dybden med et eller flere faglige elementer, der er berørt i præsentationen af opgaveløsningen, for at afdække i hvilket omfang eksaminanden kan gøre rede for disse. Evt. kan der stilles supplerende spørgsmål, hvis der er behov for at afklare i hvilket omfang eleven har nået de faglige mål, som eksamensopgaven omfatter. Der kan både være behov for at stille generelle faglige spørgsmål ved meget produkt nære præsentationer eller spørgsmål til elevens egne produktioner ved mere teoretiske præsentationer.

Som tommelfingerregel anbefales det, at eksaminationstiden fordeles ligeligt mellem elevens præsentation og samtalen.

”Opgaverne og bilag sendes til censor forud for prøvens afholdelse”

Opgaver og eventuelle bilag sendes til censor mindst 5 hverdage før prøvens afholdelse, medmindre særlige forhold er til hinder herfor. Det kan betyde, at udsendelsen må foretages, før eksamensplanen er offentliggjort. Udsendelsen af opgaver og materialer må da kun ske i et omfang, der ikke medfører, at andre dele af eksamensplanen kan udledes.

Det fremgår af hjælpemiddeloversigten, som findes på [denne side](#), hvilke regler der gælder for adgang til Internettet og værktøjer med generativ kunstig intelligens.

4.3 Bedømmelseskriterier

- ”Bedømmelsen er en vurdering af, i hvilken grad eksaminandens præstation opfylder de faglige mål, som de er angivet i pkt. 2.1.
- Ved prøve, hvor faget indgår i samspil med andre fag, lægges der vægt på at eksaminanden
 - o kan demonstrere viden om fagets identitet og metoder
 - o behandle problemstillinger i samspil med andre fag”

Når faget indgår i samspil med efterfølgende prøve i fx nv-forløb (stx og htx), i SO og SOP (hhx), i SRP (stx) og SOP (hf); lægges der vægt på, at eksaminanden kan reflektere over, hvordan valg og behandling af viden og metoder fra de indgående fag påvirker kvaliteten af den flerfaglige problemløsning.

4.3.1 Oversigt over karakterskalaen

Karakter	Betegnelse	Beskrivelse
12	Fremragende	Karakteren 12 gives for den fremragende præstation, der demonstrerer udtømmende opfyldelse af fagets mål, med ingen eller få uvæsentlige mangler.
7	God	Karakteren 7 gives for den gode præstation, der demonstrerer opfyldelse af fagets mål, med en del mangler.

Karakter	Betegnelse	Beskrivelse
02	Tilstrækkelig	Karakteren 02 gives for den tilstrækkelige præstation, der demonstrerer den minimalt acceptable grad af opfyldelse af fagets mål.

4.4 Vejledende karakterbeskrivelse

Nedenstående er vist en vejledende karakterbeskrivelse for informatik C valgfag for karaktererne 12, 7 og 02.

Beskrivelsen er udarbejdet med udgangspunkt i læreplanens faglige mål og bedømmelseskriterier.

Karakter	Beskrivelse	Informatik C
12	Fremragende	It-løsningen præsenteres glimrende og fagligt sikkert mht. planlægning, gennemførsel og evaluering. It-løsningen lever op til de stillede krav i opgaven med kun få uvæsentlige mangler. Der argumenteres fagligt velbegrundet for valg af faglige teorier og metoder. Eksaminanden perspektiverer fagligt kvalificeret sin viden til såvel egne it-produkter som til opgavens teoretiske indhold. Eksaminanden besvarer glimrende og fagligt sikkert uddybende og supplerende spørgsmål under samtalen.
7	God	It-løsningen præsenteres mht. planlægning, gennemførsel og evaluering. It-løsningen lever trods en del mangler op til de stillede krav. Der argumenteres for valg af faglige teorier og metoder. Eksaminanden perspektiverer sin viden til såvel egne it-produkter som til opgavens teoretiske indhold. Eksaminanden besvarer uddybende og supplerende spørgsmål under samtalen.

Karakter	Beskrivelse	Informatik C
02	Tilstrækkelig	It-løsningen præsenteres sparsomt og knapt mht. planlægning, gennemførelse og evaluering. It-løsningen lever minimalt acceptabelt op til de stillede krav. Der argumenteres minimalt acceptabelt for valg af faglige teorier og metoder. Eksaminanden perspektiverer tilstrækkeligt sin viden til såvel egne it-produkter som til opgavens teoretiske indhold. Eksaminanden besvarer sparsomt og knapt uddybende på supplerende spørgsmål under samtalen.