



Informatik C, hhx, htx, stx, hf

Vejledning

Ministeriet for børn og undervisning

Styrelsen for Undervisning og Kvalitet

Gymnasie- og Tilsynskontoret, august 2017

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

Citater fra læreplanen er anført i kursiv.

Indholdsfortegnelse

1. Identitet og formål	3
1.1. Identitet	3
1.2. Formål	3
2. Faglige mål og fagligt indhold	4
2.1. Faglige mål	4
2.1.1. hhx (uddannelsesspecifikt afsnit)	4
2.1.2. htx (uddannelsesspecifikt afsnit)	5
2.1.3. stx/hf (uddannelsesspecifikt afsnit)	7
2.2. Kernestof	9
2.2.1. hhx (uddannelsesspecifikt afsnit)	9
2.2.2. htx (uddannelsesspecifikt afsnit)	11
2.2.3. stx/hf (uddannelsesspecifikt afsnit)	12
2.3. Supplerende stof	14
2.3.1. hhx (uddannelsesspecifikt afsnit)	14
2.3.2. htx (uddannelsesspecifikt afsnit)	14
2.3.3. stx (uddannelsesspecifikt afsnit)	15
2.3.4. hf (uddannelsesspecifikt afsnit)	15
2.4. Omfang	15
3. Tilrettelæggelse	16
3.1. Didaktiske principper	16
3.2. Arbejdsformer	18
3.2.1. hhx (uddannelsesspecifikt afsnit)	18
3.2.2. htx (uddannelsesspecifikt afsnit)	18

3.2.3. stx/hf (uddannelsesspecifikt afsnit)	19
3.3. It	19
3.4. Samspil med andre fag.....	19
3.4.1. hhx (uddannelsesspecifikt afsnit).....	19
3.4.2. htx (uddannelsesspecifikt afsnit).....	20
3.4.3. stx (uddannelsesspecifikt afsnit)	21
3.4.4. hf (uddannelsesspecifikt afsnit)	21
4. Evaluering.....	21
4.1. Løbende evaluering.....	21
4.1.1. hhx (uddannelsesspecifikt afsnit).....	21
4.1.2. htx (uddannelsesspecifikt afsnit).....	22
4.1.3. stx/hf (uddannelsesspecifikt afsnit).....	22
4.2. Prøveform	22
4.3. Bedømmelseskriterier	23
4.4 Vejledende karakterbeskrivelse	24

Nærværende vejledning er gældende for informatik C læreplanen (2017), som er generisk, fælles for hhx-, htx-, stx-, og hf uddannelserne.

Bemærk dog, at der i vejledningen er oplyst uddannelsesspecifikke/uddannelsesstonede afsnit.

Paradigmatiske eksempler, eksempler på uddannelsesstonede undervisningsforløb, grundstof mm kan ses på fagets hjemmeside www.iftek.dk.

1. Identitet og formål

1.1. Identitet

Informatik er sit eget fagfelt, og faget har stigende betydning for alle andre fagfelter. Fagfeltet informatik er baseret på at verden (såvel den reelle som den imaginære) er 'computable', og man er optaget af at konstruere og forstå (prototyper for) computationelle strukturer, processer, artefakter og systemer. Informatik har rige relationer til og implikationer for alle andre fagfelter.

Der arbejdes i faget med centrale elementer af informatikkens-

- grundlæggende principper (f.eks. digitalisering, design og evaluering)
- tænkemåder (f.eks. strukturanalyse, abstraktion, dekomponering)
- udtryksformer (f.eks. systembeskrivelse, data- og procesbeskrivelser, design, programmering) og
- arbejdsformer (f.eks. problemanalyse, begrebsmodellering, iterative metoder, brugerinddragelse, trinvis forbedring, test og fejl retning).

1.2. Formål

I informatik skal eleverne opnå forståelse af informatikkens muligheder 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 digitale systemer, herunder teste og raffinere disse, og undervisningen har et særligt fokus på eleverne som kreative og innovative producenter.

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

Blandt de faglige mål indgår, at eleverne skal kunne:

- demonstrere viden om fagets identitet og metoder
- behandle problemstillinger i samspil med andre fag
- 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 relevante informatik-faglige 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.

I læreplanens afsnit 1.1 og 1.2 er der givet en kompakt beskrivelse af fagets identitet og af det overordnede formål med undervisningen.

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 faget informatiks 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 system i bredeste forstand. Her kan både være tale om en hjemmeside til en pizza-butik, en webshop, men også et computerspil med et problemorienteret fokus eller en helse-app.

2.1.1. hhx (uddannelsesspecifikt afsnit)

– *Konstruktion af it-system som løsning til en problemstilling*

Dette faglige mål er samlende for de øvrige 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, reklamespil eller lignende.

– *It-systemers og menneskelig aktivitets gensidige påvirkning*

Eleverne præsenteres for en række it-systemer og it-innovationer, der har betydning for mennesker og virksomheder. Det kan være it-systemer, der er tæt på eleverne i dagligdagen (som sociale medier, udvalgte apps og Nemt Id) og systemer, der har stor betydning for virksomheder (som webshops, CRM- og ERP-systemer). Det er målet, at eleverne indser, hvor stor betydning it-systemer har for vores velfærd, for virksomheders konkurrenceevne og for muligheden for at imødegå de udfordringer globaliseringen bringer.

Når man udvikler og designer et IT-system, designer man ikke blot systemet, men de arbejdsgange, brugsmønstre og processer anvendelsen af systemet medfører. Elever bør derfor være i stand til at komme med eksempler på, hvordan it-systemer har haft betydning for menneskelig aktivitet indenfor eksempelvis handel, økonomi, drift af virksomhed og kommunikation. Eleverne bør kunne udarbejde en model over en problemstilling, som kan strukturere og simplificere emnet ud fra de krav, der er til systemet, det kunne være gennem rige billeder, use case og aktivitetsdiagrammer.

– *It-sikkerhed, netværk og arkitektur*

For at kunne beskytte deres digitale identitet og data kan eleverne arbejde med såvel fysisk -, logisk -, kommunikations- og datasikkerhed. Ligesom de bør introduceres til forskellige kategorier af trusler og trusselsaktører (computervira, hackere osv.) samt for forskellige modmidler, der formindsker eller fjerner truslerne f.eks. (brugerhåndtering og passwords, firewall, digital signatur, kryptering, åbne og lukkede netværk, VPN, SSL, HTTPS, backup-

procedurer). Forholdet mellem datas tilgængelighed, integritet og fortrolighed, kan for eksempel belyses gennem CIA-modellen.

Mange af de it-systemer, som eleverne anvender til dagligt, fungerer over internettet eller andre netværk, ligesom der i virksomhedssammenhæng sjældent anvendes enkeltstående systemer. Derfor bør eleverne kunne redegøre for, hvordan arkitekturen af et it-system har betydning for systemets funktionalitet og design, samt hvordan flere forskellige it-systemer kan levere data til hinanden.

– *Repræsentation og manipulation af data*

Data er grundlæggende for it-systemer, og derfor bør eleverne få en forståelse for hvilke typer af data it-systemer anvender, og hvordan data repræsenteres og er modelleret. Eleverne introduceres for forskellige typer af data (tal, tekst, billeder, lyd m.m.) og bør kunne redegøre for principperne bag de udvalgte typer af data, og hvordan data kan manipuleres, herunder konsekvenserne af manipulation, f.eks. tab af information. Eleverne skal også kunne strukturere data i f.eks. E/R-modeller eller datastruktur-modeller, så systemet kan leve op til de krav brugerne af systemet har. På baggrund af datamodeller skal eleverne kunne forklare, hvordan data kan organiseres i databaser, og hvordan data i databaser kan oprettes, ændres og vises i it-systemer.

– *Programmering*

Eleverne bør være i stand til at identificere de mest basale strukturer og begreber inden for programmering som sekvenser, variable, forgreninger, løkker og funktioner. Derudover skal de også kunne modellere et simpelt it-system, samt udvikle og tilpasse kode til it-systemet, uden det dog er et krav at de skal kunne konstruere it-systemet fra grunden. Programmering kan ske med tekstbaserede programmeringssprog og igennem værktøjer som f.eks. Scratch, App lab, App Inventor eller lignende. I udviklingen af it-systemer bør eleven få forståelse for, hvordan man opdeler en udviklingsopgave i delproblemer og modellerer flowet i systemet (se også afsnit 3.1).

– *Interaktionsdesign*

Interaktionsdesignet har stor betydning for oplevelsen og brugen af et it-system og derfor bør eleverne kunne redegøre for interaktionsdesigns ud fra eksempelvis gestaltlove, farvelære, brugervenlighedstest og målgrupper. Derudover skal de kunne udarbejde egne interaktionsdesign via f.eks. roughs, wireframes eller mockups og implementere eller tilpasse it-systemer ud fra disse.

– *Innovation*

Eleverne bør introduceres for innovative it-systemer, der har betydet noget for den måde, hvorpå vi lever, kommunikerer og laver forretning. Eleverne bør kunne redegøre for hvorvidt innovationen ligger i interaktionsdesign, funktionalitet eller dataanvendelse, og om det er en inkrementel eller radikal innovation. De bør også kunne sammenholde kendte innovative it-systemer med egne it-systemer og vurdere egne løsningers grad af innovation, og hvordan anvendelsen af it-systemet kan gøre en forskel.

2.1.2. htx (uddannelsesspecifikt afsnit)

– *Konstruktion af it-system som løsning til en problemstilling*

Dette overordnede faglige mål understreger, at informatik er et fag med særligt fokus på, at eleverne skaber forskellige it-systemer med henblik på at afhjælpe et givent problem. Det kunne eksempelvis være en hjemmeside, der sælger produkter, en app, der hjælper med at finde vej, et læringsorienteret computerspil m.m. Problemet kan enten være formuleret af læreren, eller eleverne kan selv finde og undersøge et problem. Processen består af forundersøgelse, produktion og afprøvning af en løsning. I den første fase findes der mange hensigtsmæssige metoder og redskaber, som man også kender fra teknologifaget og kommunikation/IT, såsom problemformulering, problemanalyse, problemtræ, målgruppeanalyse mv. Overvejelser om disse aspekter er nødvendigt før eleverne kaster sig ud i produktionsfasen. I forhold til afprøvning af løsningen refereres både til kvalitative og kvantitative tests, herunder fokusgrupper, interviews, spørgeskemaer, spiltests, videoobservation, tænkehøjtmetoder mv.

– *It-systemers og menneskelig aktivitets gensidige påvirkning*

Dette faglige mål handler om den betydning it-systemerne har for menneskers handlinger og tankemønstre. På C-niveau er det tilstrækkeligt at eleverne blot kan redegøre og give eksempler herpå. Ved at inddrage innovative it-systemer i denne sammenhæng berøres det faglige mål vedr. innovation også. Hvilke nye arbejdsgange og brugsmønstre dukker op? Det kan også handle om, hvordan it-systemer ændrer virksomheders kommunikationsprocesser og markedsposition. Viden om it-systemers menneskelige påvirkning kan både erhverves gennem allerede etableret teori og statistik, eller ved at eleverne selv udarbejder kvalitative og/eller kvantitative undersøgelser heraf.

– *It-sikkerhed, netværk og arkitektur*

Emnet om it-sikkerhed, og hvordan man opnår beskyttelse af digital identitet, kan både behandles fra et brugerperspektiv og fra et udviklerperspektiv. Som bruger er det vigtigt at kende til love om ophavsret, etik og sikkerhedsindstillinger såsom passwords og firewalls. Som udvikler er det endvidere vigtigt at kende til krypterings- og sikkerhedsmekanismer såsom SSL, HTTPS, VPN, åbne og lukkede netværk m.m. Hacking- og vira-beskyttelse er også oplagte temaer, og som cases kan nævnes Edward Snowdens afsløringer eller aflytninger af profilerede politikere eller terrorister.

Eftersom de fleste it-systemer er forbundet til andre systemer i et netværk, er undervisning i, hvordan data overføres på netværk oplagt. Eleverne skal kunne redegøre for hvordan arkitekturen af et it-system har betydning for systemets funktionalitet og design, samt hvordan flere forskellige it-systemer kan levere data til hinanden.

– *Repræsentation og manipulation af data*

Eleverne bør få en forståelse for, hvilke typer af data, der forefindes i et givent system, og hvad grundlæggende begreber som input og output betyder. Af forskellige datatyper kan nævnes tal, tekst, billeder, lyd, videoer m.m. Gennem elevernes arbejde med it-systemer bør de også få kendskab til, hvordan disse forskellige datatyper kan manipuleres, og hvilke konsekvenser denne manipulation kan have. Eleverne bør også lære hvordan data i et it-system kan trækkes ud og vises til brugeren, der møder systemet.

Eleverne bør også få en forståelse for hvordan data kan organiseres i databaser - herunder hvordan data kan oprettes, ændres, flyttes og slettes heri - dog kun på et redegørende niveau.

– *Programmering*

Det at lære at programmere fra bunden i eksempelvis C++ eller C# er for de fleste meget tidskrævende og kan være frustrerende specielt, hvis eleverne ikke har stiftet bekendtskab med det før. Der er mange ting at huske på, og det stiller store krav til logisk tænkning og struktur. Med de få timer, der ligger i et C-niveau-fag, kan det være svært at nå at komme dertil, hvor eleverne kan lave et brugbart it-system, hvis alting skal kodes fra bunden. Heldigvis findes der flere programmer, hvor eleverne på en mere simpel, visuel og letfordøjelig måde kan lære de mest basale principper for og begreber bag programmering, f.eks. spil- og app-udviklingsprogrammerne Scratch, GameMaker og App Inventor. Alternativt kan man – jf. afsnit 3.1 - som underviser give eleverne et allerede etableret program og så bede dem om at modificere enkelte programmeringslinjer for derefter at observere effekten heraf.

– *Interaktionsdesign*

Eleverne bør få kendskab til forskellige principper for interaktionsdesign - det kunne eksempelvis være Jakob Niensens 10 bud eller brugen af gestaltlove, kontraster, farveteori og typografi. Med denne viden kan eleverne få til opgave at analysere udvalgte it-systemers interaktionsdesign med henblik på, hvordan interaktionsdesignprincipperne er overholdt, og hvordan brugeroplevelsen virker. I sådanne øvelser er det naturligvis vigtigt at have øje for it-systemets målgruppe. Endelig kan eleverne udarbejde deres egne brugertest, enten på allerede eksisterende it-systemer eller - måske mere interessant - på deres egenproducerede. Her kan undervisning om kvalitative og kvantitative undersøgelser, fokusgrupper, prototyping, mock-ups m.m. være oplagte at inddrage. Disse brugertest kan enten foretages ved afslutningen af et projekt eller - mere formativt - i midten af et projekt, hvorved eleverne får mulighed for at tilpasse deres it-system til brugertestens resultater.

– *Innovation*

Eleverne bør kunne redegøre for, hvad der gør et it-system innovativt. Her kan man inddrage eksisterende innovative it-systemer og redegøre for, hvordan de virker. Begreber som brugerinddragelse, disruption og Blue Ocean-strategi kan være relevante i denne sammenhæng. Eleverne kan også blive bedt om at vurdere deres eget it-systems innovativitet ved at sammenligne med lignende it-systemer. I denne forbindelse kan man også undervise i, hvilken påvirkning it-systemet har for menneskelige aktivitet jf. dette andet faglige mål.

Eleverne skal kunne redegøre for graden (inkrementel/radikal) og typen (paradigme/position/proces/produkt) af innovation.

2.1.3. stx/hf (uddannelsesspecifikt afsnit)

– *Konstruktion af it-system som løsning til en problemstilling*

Dette faglige mål er samlende for de øvrige faglige mål. Da informatik i udpræget grad er et skaberfag, drejer undervisningen sig hovedsageligt om at konstruere en løsning til en it-problemstilling. I denne proces medtænkes de øvrige faglige mål.

Processen består af faserne problembeskrivelse og design, realisering og afprøvning af en løsning. Denne iterative metode er universel og unik for faget.

De it-problemstillinger, eleverne designer løsninger til, bør være relevante og kan være beskrevet af underviser eller eleverne selv. Eksempler er web-butikker, apps til helseformål eller spil til undervisning i grammatiske regler.

– *It-systemers og menneskelig aktivitets gensidige påvirkning*

I undervisningen arbejdes der med it-systemer og brugeres gensidige påvirkning. På den ene side kan der arbejdes med eksempelvis sociale medier og deres påvirkning at brugernes sociale liv, og på den anden side kan der arbejdes med brugernes påvirkning af de selvsamme sociale medier.

Eleverne skal opnå viden om, hvordan deres egne løsninger til it-problemer påvirker de involverede brugere, og denne viden skal bruges aktivt i konstruktionsprocessen. Desuden bør brugernes for forståelse for løsningstypen til en given it-problemstilling aktivt implementeres i løsningerne. Her er et eksempel en web-butik, hvor placeringen af indkøbskurven er den samme, som den placering brugeren kender i forvejen. Hermed opnås, at kunden nemt kan finde den, og det er i denne kontekst vigtigt.

– *It-sikkerhed, netværk og arkitektur*

Eleverne skal kunne identificere it-systemer, hvor sikkerheden er afgørende. Et særligt eksempel er elevernes egen it-sikkerhed. De skal kunne identificere trusselaktører (tekniske og menneskelige) og forholde sig til, hvordan de kan beskytte sig vha. modmidler. Et andet eksempel er it-systemer, der bruges over netværk, og hvilke sikkerhedsproblemer der dukker op i denne kontekst.

Eleverne bør se eksempler på simple arkitekturer såsom klient-server modellen og kunne placere egne og andres it-systemer i disse.

– *Repræsentation og manipulation af data*

Da ethvert it-system arbejder med data som input og output, skal eleverne have et indgående kendskab til begrebet. De bør se eksempler på repræsentation af data i form af datatyper såsom tal, tekst, lyd, billeder etc., og de bør se på relevante manipulationer.

Desuden bør de se på organisering af data i databaser, herunder oprettelse, ændring, flytning og indsættelse/sletning. Her kan det f.eks. dreje sig om et elevsystem, der anvender elev- og fagdatabaser eller en web-butik, der anvender kunde- og varedatabaser.

– *Programmering*

På c-niveau er det nok at eleverne stifter kendskab med blok-programmering såsom Scratch, Lego-programmering eller lignende. Fordelen er at syntaksfejl undgås.

Selvom man vælger blokprogrammering, skal eleverne have overblik over og kunne genkende grundstrukturernes sekvens, forgrening og løkke. Desuden skal de have kendskab til de vigtigste datastrukturer og deres egenskaber, der er i det valgte programmeringssprog.

Eleverne skal kunne konstruere simple it-systemer og tilpasse givne it-systemer.

– *Interaktionsdesign*

Groft sagt er et it-system ikke meget værd, hvis det ikke løser brugernes behov og er tilgængeligt på en måde, så brugerne kan anvende det. Eleverne skal have kendskab til elementer af interaktionsdesign såsom Jakob Nielsens 10 bud eller brugen af gestaltlove og farver, og de skal både kunne forholde sig til interaktionsdesigns af andres it-systemer, og de skal kunne implementere valgte interaktionsdesigns i egne it-systemer.

– *Innovation*

Eleverne bør kunne redegøre for, hvad der gør et it-system innovativt. De bør således på basis af en klar definition kunne afgøre om eget eller andres it-system er innovativt. Eleverne bør kunne redegøre for graden (inkrementel/radikal) og typen (paradigme/position/proces/produkt) af innovation.

2.2. Kernestof

Gennem kernestoffet skal eleverne opnå faglig fordybelse, viden og kundskaber. Kernestoffet er:

2.2.1. hhx (uddannelsesspecifikt afsnit)

It-systemers og menneskelig aktivitets gensidige påvirkning

– *it-systemer og brugeres gensidige påvirkning i forhold til etik og adfærd*

Introduktion til forskellige typer af systemer, der har stor betydning i privatlivet (f.eks. sociale medier, apps, NemID), for virksomheder (f.eks. ERP, CRM, BI, e-business hjemmesider) og for samfundet (f.eks. CPR-register, elektroniske patientjournaler, tastselv løsninger på f.eks. skat.dk).

– *modellering som middel til at forstå et problemområde*

Modellering af et problemområde kan gøres med rige billeder med fokus på, hvilke flow af data der er, mens arbejdsgange kan modelleres med aktivitets- og/eller flow/rutediagrammer

– *brugsmønstre til afdækning af brugertypers krav til et it-system*

Eleverne fastlægger brugstilfælde for it-systemer i form af use cases og use case diagrammer med udgangspunkt i de krav, brugerne stiller til systemet.

– *brugertest til kvalitetssikring af et it-system i forhold til brugertypers krav*

Planlægning, udførelse og efterbehandling af flere forskellige former for brugertest. Både kvalitative (fokusgruppeinterview, observationer m.m.) og kvantitative (spørgeskemaer m.m.) bør inddrages.

It-sikkerhed, netværk og arkitektur

– *Internettets teknologi og sikre kommunikationsformer*

Under Internettets teknologi er det oplagt at arbejde med kryptering, webservere, SSL, overvågning, IP-protokoller og hacking. Ligesom valg af password, digital signatur, compu-

tervira og modmidler, firewall og phishing alle er begreber, der er centrale for beskyttelse af digital identitet.

Det er oplagt at arbejde med virksomheders sårbarhed overfor digitale trusler, som f.eks. hacking og hvordan virksomheder kan imødekomme problemerne. En model til beskrivelse af dette kunne være CIA-modellen.

- *client-server arkitektur*

Introduktion til client-server arkitektur og forskellige eksempler på anvendelse. F.eks. de fordele virksomheder kan opnå med centrale data, der kan tilgås fra flere steder og på samme tid.

Repræsentation og manipulation af data

- *Abstraktion og strukturering, begrebs- og datamodeller*

Data kan f.eks. struktureres og modelleres i E/R- eller datastruktur modeller.

- *data og datatypers repræsentation og manipulation*

Forskellige datatyper som tekst, tal, dato, billeder, lyd og video.

Filformater og komprimering.

- *databasers anvendelse og simple databaseforespørgsler*

Elever skal kunne redegøre for databasers opbygning og hvordan de kan anvendes i forretningsøjemed, samt redegøre for simple databaseforespørgsler, f.eks. i SQL – SELECT, UPDATE, INSERT INTO.

Programmering

- *Funktioner*

Eleven introduceres til funktioner og de fordele, man opnår ved anvendelsen af disse.

- *variable, sekvenser, løkker og forgreninger*

Introduktion og træning i forståelse og anvendelse af variabler, sekvenser, forgreninger og løkker i et programmeringssprog eller værktøj.

Interaktionsdesign

- *design af en brugergrænseflade og den tilhørende interaktion*

Interaktionsdesign er en kreativ og konstruktiv proces, der skal sikre at interaktionen mellem brugeren og it-systemet fungerer efter hensigten. Eleverne kan arbejde med kommunikationsforhold og modeller, målgrupper og målgruppeanalyser. Eleverne bør også lære at styre deres udviklingsprocesser gennem agile og strukturerede udviklingsprocesser

- *prototyper til i samarbejde med brugerne at udvikle it-systemets interaktionsdesign*

Eleverne kan arbejde med mockups, roughs, wireframes og lignende metoder til at lave prototyper.

- *principper for interaktionsdesign*

Eleverne kan introduceres til grundlæggende brugervenligheds principper, typografiske valg, gestalt lovene og brug af multimedier i it-systemer

Innovation

– *eksempler på og kategorisering af innovative it-systemer*

Eleverne kan arbejde med forskellige typer af og definitioner på innovation, samt forskellen på innovation og invention.

Eleverne præsenteres for innovative it-løsninger som f.eks. Skype, Facebook, smartphones osv. og introduceres til metoder til kategorisering af innovative it-løsning som 4-P modellen, inkrementel/radikal innovation og analyse af innovationshøjden.

2.2.2. htx (uddannelsesspecifikt afsnit)

Grundlæggende behandles kernestoffet i faget ved at eleverne konstruerer et it-system som løsning på en problemstilling. Af eksempler på it-systemer, der kan løse et mindre problem, kan nævnes hjemmesider, computerspil og apps. Som forarbejde til elevernes konstruktion af et it-system kan de være hensigtsmæssigt at introducere dem for grundlæggende kommunikationsteori om afsender, budskab og målgruppe. Udvalgte målgruppeanalyseværktøjer såsom Minerva-modellen, Gallups-kompasset, Bartles spillertyper, Conzoom m.m. er oplagte at inddrage.

I forhold til hjemmesider kan man vælge at træne eleverne i basal HTML og CSS. Det at lære et nyt sprog at kende kan dog være ret tidskrævende, hvorfor man kan benytte sig af et CMS-system.

I forhold til computerspil findes der forskellige værktøjer, såkaldte game engines. De mest kendte er Scratch, GameMaker og Unity. Scratch er det mest simple, hvor man relativt hurtigt kan udvikle et ganske basalt 2D-spil, da programmeringen foregår via et grafisk interface. Dette gælder også Game Maker, men her er der flere muligheder for at ændre i den bagvedliggende kode. Unity muliggør også at lave 3D-spil, men har et noget kompliceret interface, som kan være svært at nå at lære på den begrænsede tid der ligger i Informatik C. Unity benytter programmeringssproget C#, hvilket er et tekstbaseret programmeringssprog. Eftersom spillene ville skulle løse en problemstilling, vil det være naturligt at inddrage teori om begrebet "serious games" - computerspil, der netop har et andet formål end bare underholdning. Det kunne være spil, der fungerer i en reklamekampagne, en politisk kampagne eller i en læringssammenhæng.

I forhold til udvikling af apps kan man anvende AppInventor, Appy Pie, og App Institute, der også muliggør skabelse af apps uden brug af kode.

It-systemers og menneskelig aktivitets gensidige påvirkning

Dette kernestofområde kan behandles i et forløb med sociale medier, som Facebook og Twitter, hvor man undersøger mediernes påvirkning på og resultat af vores adfærd, selvforståelse og sociale udfoldelse. Også emner som big data, robotteknologi, kunstig intelligens og virtual reality kunne være relevante i denne sammenhæng. Man kan også undersøge begreber som nudging og affordances i forhold til, hvordan IT-systemer ansporer dets brugere til givne handlinger. Megen teori inden for den anvendte psykologi kunne være relevante at inddrage i denne sammenhæng.

I forhold til brugertests findes der både kvalitative og kvantitative metoder, herunder fokusgrupper, interviews, spørgeskemaer, spilstests, videoobservation, tænkehøjtmetoder mv.

It-sikkerhed, netværk og arkitektur

Klient-server arkitektur og tre-lags-arkitektur kunne være et tema i et spilforløb - eksempelvis i forhold til integration af et highscore-system. Netværk er et oplagt tema i forhold til et forløb om hjemmesider men også i apps og spiludvikling - særligt i forhold til online multiplayer-spil. At implementere netværk i en app eller et computerspil er dog en vanskelig og tidskrævende affære, så i stedet kan eleverne nøjes med at analysere, hvordan et eksisterende spil benytter et netværk til at sende og holde styr på informationer.

Repræsentation og manipulation af data

Repræsentation og manipulation af data, det være sig tal, billeder, video, lyd, tekst e.lign., handler om at identificere, hvordan data i det hele taget vises og ændres i et it-system - eksempelvis i form af en highscore i et computerspil eller et varelager i en webshop. Eleverne kan som forarbejde hertil arbejde med E/R diagrammer og relationelle databaser. Emnet kan også behandles i forhold til, hvordan big data opsamles, bearbejdes og præsenteres.

Programmering

At lære at programmere fra bunden i et sprog som f.eks. C# eller C++ kan være en tidskrævende affære. Derfor kan det give god mening med it-systemer, der anvender visuelle programmeringsværktøjer (såsom Scratch), da eleverne herved fint kan få en forståelse for centrale programmeringsbegreber såsom variable, sekvenser, løkker og forgreninger. Med denne tilgang slipper eleverne for at arbejde med syntaks, hvilket kan være et nødvendigt kompromis på c-niveau.

Interaktionsdesign

Interaktionsdesign handler bl.a. om brugervenlighed. Her kan Jakob Nielsens 10 bud være relevante - særligt i forbindelse med webdesign - fordi de er meget konkrete for eleverne at arbejde med selvstændigt. Brugen af gestaltlove, typografi, kontraster, farver og andre visuelle virkemidler, der guider brugeren, er også relevante. Teknologier som eye tracking og heatmaps er også relevante i forhold til at evaluere et interaktionsdesign. I forhold til spiludvikling handler interaktionsdesign også om at skabe et fængende og motiverende gameplay, hvorfor principper for godt gameplay design er et relevant emne. Med emner som Virtual Reality og Augmented Reality kan man også behandle interaktionsdesign i et bredere perspektiv. Makey Makey, HTC Vive, Oculus Rift, Samsung Gear VR er alle oplagte teknologier i denne sammenhæng.

Innovation

Eleverne kan arbejde med forskellige definitioner på og typer af innovation, og de kan præsenteres for innovative it-systemer som f.eks. Airbnb eller GoMore og analysere, hvad der gør dem innovative og i hvilken udstrækning. Begreber som disruption, 4-P modellen, inkrementel/radikal innovation og analyse af innovationshøjden kan der arbejdes med her.

2.2.3. stx/hf (uddannelsesspecifikt afsnit)

It-systemers og menneskelig aktivitets gensidige påvirkning

Eleverne bør præsenteres for en række it-systemer og på baggrund af disse diskutere, hvordan de påvirker brugerne og vice versa. Diskussionerne kan dreje sig om hvordan it-systemerne påvirker brugernes holdninger og arbejdsmetoder. Der kan tages udgangspunkt i sociale medier, computerspil, patientjournalssystemer, health tracker apps o.lign. Helt konkret kan man diskutere om skydespil gør spillerne mere voldelige.

Desuden kan diskussionerne omhandle hvordan brugere præger it-systemer - også utilsigtet. Her kan sociale medier igen anvendes som eksempel. Det er brugerne, der udfylder indholdet.

Eleverne bør i egne it-produktioner medtænke hensigtsmæssige brugsmønstre og brugertests.

It-sikkerhed, netværk og arkitektur

Eleverne kan arbejde med it-sikkerhed og emner som fortrolighed, integritet og tilgængelighed (CIA-modellen). Desuden kan de arbejde med, hvordan man sikrer delaspekter vha. for eksempel kryptering.

Eleverne bør kende principperne for gode valg af passwords og sikker adfærd i forhold til færdsel på internettet.

Eleverne kan arbejde med client/server arkitektur, for eksempel i forbindelse med et computerspil, hvor highscorelisten ligger på en server.

Repræsentation og manipulation af data

Eleverne skal have kendskab til data i form af tal, tekst, billeder osv. De skal have kendskab til de basale datatyper og databasebegrebet. De bør se data organiseret i databaser, for eksempel E/R-diagrammer, og se disses anvendelse i et it-system. De bør have kendskab til simple SQL-forespørgsler. Big data kan også behandles.

Programmering

Da det er tidskrævende at lære tekstbaseret programmering, er det nok på c-niveau at lære blokprogrammering. Eleverne skal alligevel kende variable, sekvenser, løkker og gentagelser. Derfor er det ikke tilstrækkeligt kun at arbejde med HTML. Gode valg af programmeringssprog er Scratch, hvis eleverne vil programmere spil, Lego Mindstorm EV3, hvis man vil programmere Lego-robotter eller App Inventor, hvis eleverne skal lave en app.

Interaktionsdesign

Eleverne skal arbejde med design af brugergrænseflader, der opfylder krav om brugervenlighed. Her kan man for eksempel anvende Jakob Nielsens 10 bud. Gestaltlovene kan også anvendes i vurdering og design. Når eleverne designer brugergrænseflader, kan de forholde sig til brugergrupper, og de kan arbejde med prototyper, der testes.

Innovation

Eleverne kan sammenligne egne it-systemer med andre innovative it-systemer. De kan arbejde med it-systemer som Skype, smartphones, Airbnb etc. og analysere, hvorfor de er innovative. Eleverne skal yderligere kunne kategorisere it-systemerne i forhold til 4-P modellen og inkrementel/radikal innovation.

Eleverne bør præsenteres for et eller flere it-systemer, som indeholder forskellige grader/typer af innovation.

2.3. Supplerende stof

Eleverne vil ikke kunne opfylde de faglige mål alene ved hjælp af kernestoffet.

I forhold til de faglige samspil med de øvrige fag i uddannelsen vælges der supplerende stof med henblik på at bibringe faglig fordybelse og styrke toningen af kernestoffet. Dele af det supplerende stof vælges i samarbejde med eleverne, når det er muligt

2.3.1. hhx (uddannelsesspecifikt afsnit)

Det supplerende stof er med til at tone faget i en mere merkantil retning og derfor er det oplagt at tage udgangspunkt anvendelsen af it i virksomheder. Der er mange muligheder for udvælgelse af supplerende stof både ved at gå dybere ned i kernestoffet eller supplere med helt nyt stof. Eleverne kan arbejde mere indgående med databaser og deres anvendelse i virksomheder som grundlag for CRM-, ERP- eller Business Intelligence systemer og dermed også lave mere komplekse databaseforespørgsler. Eleverne kan arbejde med Big data og de fordele, virksomheder kan opnå ved anvendelsen af data som beslutningsgrundlag. Eleverne kan også arbejde med interaktionsdesignets betydning for salg på e-handelssider eller algoritmer til optimering af reklameindsats og som værktøj til at opnå mersalg. Man kan også vælge at behandle globalisering som et supplerende tema, hvor eleverne opnår forståelse for hvordan it er med til at muliggøre globalisering og virksomhederne mulighed for at agere nye markeder.

Som supplerende stof kan man ligeledes arbejde med flere af de it-begreber der spiller en stadig større rolle for mennesker og virksomheden, som web 2.0, Internet of Things, kunstig intelligens og cloud computing.

Det er oplagt at behandle dele af det supplerende stof som tværfaglige forløb.

2.3.2. htx (uddannelsesspecifikt afsnit)

Det supplerende stof er i høj grad med til at tone faget i forhold til den pågældende studieretning. Derfor vil det være oplagt at vælge emner, der skeler til den studieretningstuning klassen har. Toningen vil ofte være af teknologisk, designorienteret, samfundsvidenskabelig eller naturvidenskabelig art.

Som supplerende stof kan man tone faget i en mere teknisk/teknologisk retning ved eksempelvis at arbejde med alternative controllere til it-systemer herunder virtual/augmented reality, hvilket er oplagt sammen med teknologifaget eller teknikfag. Her kan man have fokus på rent faktisk at udforme controllere, med hvilke man kan styre it-systemer på alternative måder, hvilket vil styrke det faglige mål ang. interaktionsdesign. Internet of Things er et oplagt tema i denne sammenhæng. Sammen med samfundsfag kan man arbejde med kunstig intelligens, it-historie, big data, web 2.0, it-sikkerhed, it og globalisering, it og sundhed, deleøkonomi mv. Sammen med matematik kunne emner som kunstig intelligens og algoritmik være oplagte emner. Alle disse er aktuelle emner - særligt i nutidens mediebillende. Sammen med naturvidenskab kunne man arbejde med naturvidenskabelig formidling herunder læringsspil, der underviser elever inden for et naturvidenskabeligt område. Man kunne også lave fysiksimulationer i eksempelvis Unity for at forklare/formidle, hvordan fysiske love manifesterer sig, eller lave helse-apps i samarbejde med biologi.

2.3.3. stx (uddannelsesspecifikt afsnit)

Det supplerende stof er i høj grad med til at tone faget i forhold til den gymnasiale retning. Derfor vil det på STX være oplagt i valgene af emner at skele til studieretningen i stamklasser og det almene aspekt på valghold. På valghold kan der ligeledes perspektiveres til det humanistiske, det samfundsfaglige og det naturvidenskabelige område.

Specielt på det naturvidenskabelige område har faget en forpligtelse til at perspektivere til matematik. Specielt på det naturvidenskabelige område har faget en forpligtelse til at perspektivere til matematik. Her er en mulighed at tage udgangspunkt i et it-system, der anvendes i matematik, og undersøge det. Et eksempel kunne være en trekantsberegner

Når faget indgår i naturvidenskabeligt grundforløb, er en mulighed at it-faget udvikler et it-system, som er relevant for det emne, man behandler i NV. Det kan for eksempel være et it-system, der beregner energi ud fra madprodukter, hvis NV-emnet er kroppen.

Den faglige fordybelse er central, og fagets metode skal stå tydeligt i undervisningen. Både it-fagets faglighed, men også it-fagets anvendelse skal være central.

Emner der kan være relevante at behandle under supplerende stof: kunstig intelligens, webteknologier, netværk, it-historie, algoritmik, informationsarkitektur, Model View Control-arkitektur og virtual reality. Det er også relevant at behandle aktuelle it-emner, der dukker op i pressen.

2.3.4. hf (uddannelsesspecifikt afsnit)

Det supplerende stof er i høj grad med til at tone faget i forhold til den gymnasiale retning. Derfor vil det på HF være oplagt i valgene af emner at skele til de andre fag på studiet og de professionsrettede perspektiver. Eleverne kunne arbejde med it-systemer til sundhedssektoren, til politiet og lignende.

Et konkret eksempel på et it-system, der vil være relevant hos sundhedsvæsenet er et system, der på baggrund af patientens stamdata og madindtag kan afgøre, om patienten får næring nok. Både it-fagets faglighed, men også it-fagets anvendelse skal være central.

Emner der kan være relevante at behandle under supplerende stof: kunstig intelligens, webteknologier, netværk, it-historie, algoritmik, informationsarkitektur, Model View Control-arkitektur og virtual reality. Det er også relevant at behandle aktuelle it-emner, der dukker op i pressen.

2.4. Omfang

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

Fagligt stof i faget omfatter f.eks. netbaserede tutorials (herunder video-tutorials), netbaserede udviklingsværktøjer, -biblioteksmoduler, -dokumentationer og vejledninger, i- og e-bøger og traditionelle undervisningsmaterialer i form af bøger, udleveret tekst materiale m.m.

Det er ikke meningen at disse eksempler på netbaserede materialer skal konverteres til konkrete sidetal, men der bør være en sammenhæng mellem studieplanens angivelse af fagligt bearbejdet stof og det forventede omfang på 90 - 150 sider.

Opgivelsen af omfang har til formål at sikre den faglige kvalitet, så eleverne hverken under- eller overbelastes fagligt. 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 niveauforskellige 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 elevtiden til forberedelse af et 2 sideres dokument med tekst, koder, modeller osv. sammenholdt med en forberedelse af eksempelvis en videotutorial over samme tema og faglige indhold.

3. Tilrettelæggelse

3.1. Didaktiske principper

- *Undervisningen tilrettelægges ved brug af anerkendte didaktiske principper, herunder ‘use-modify-create’-progression fra at anvende udleverede programmer til at modificere disse for til sidst selvstændigt at skabe (nye dele af) it-systemer; ‘Stepwise Improvement’, som teknik til trinvis, iterativ og systematisk udvikling af programmer og ‘Worked Examples’ (kombineret med ‘faded guidance’), til illustration af eksemplariske løsningsprocesser.*

Læreplanen udelukker ikke anvendelse af andre didaktiske tilgange til informatikundervisningen.

Stepwise improvement (fig. 1) er et eksempel på en didaktisk- og metodisk tilgang til arbejdet med programmering. For alle projektførøb 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.

Tema/Problemfelt/Projekt

1. undlad at starte med en færdig kravspecifikation
1. start med en simpel specifikation; udvid den gradvist
2. vær bevidst omkring nødvendigheden af omstruktureringer

Konkretisering:

Eleverne arbejder med at konkretisere abstrakt kode til kode der virker, og som opfylder delmål og kravspecifikationer.

Udvide:

Eleverne arbejder med at udvide programmets funktionalitet og kravspecifikation

Omstrukturering:

Eleverne arbejder med ikke-funktionelle aspekter i programmet, så det fremstår mere velorganiseret.

Stepwise Improvement = (Konkretisering | Udvidelse | Omstrukturering)

Fig 1

Eleverne skal gradvist kunne overtage processen med egen produktion af it-systemer (fig. 2), dels gennem forbedring og løsning af konkrete delopgaver i deres projekt med basis i de gennemprøvede eksempler (WE), dels gennem arbejdet med at udvide kravspecifikationerne til produktet (udvide) og til den færdige produktion (omstrukturere).

Modellen kan bruges som et planlægningsværktøj til hvordan man kommer fra A til B til C, og som sådan er modellen ret lavpraktisk. I stedet for at gå mere eller mindre tilfældigt frem mod et færdigt program, kan eleverne bevæge sig mere systematisk i 3 dimensioner ved dels at forbedre deres eksisterende programmer (f.eks. rette fejl), udvide dem (tilføje mere funktionalitet) eller omstrukturere (dvs. ændre på strukturen i deres programmer).

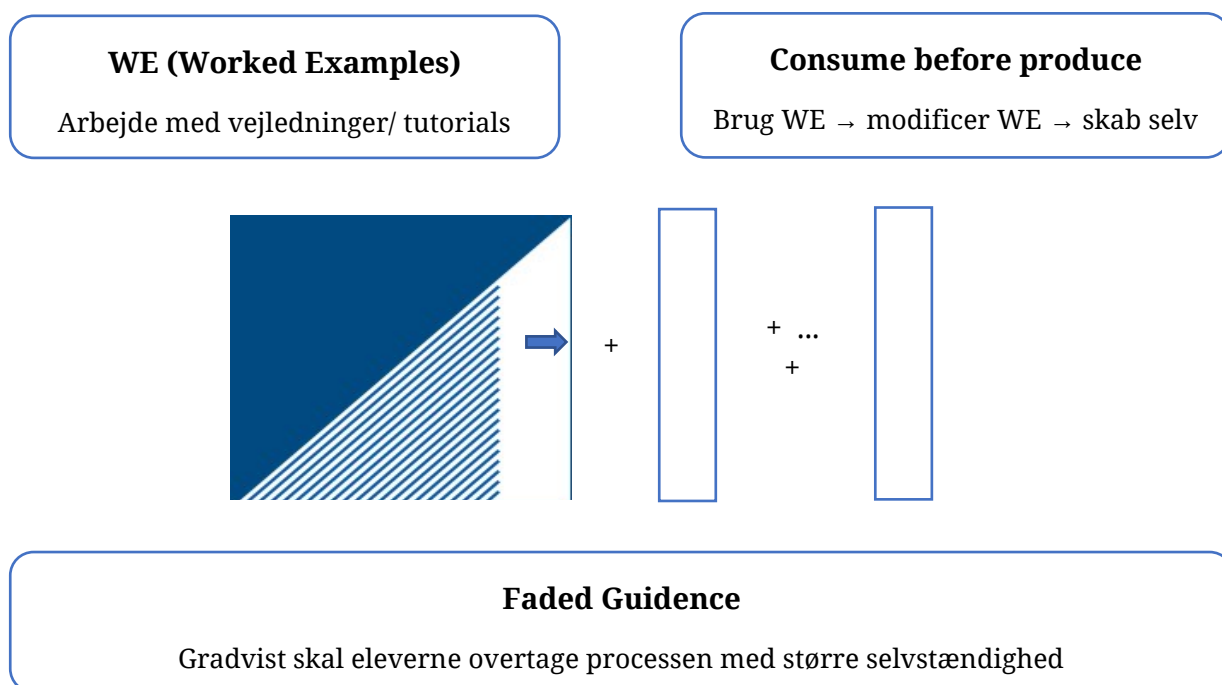


Fig 2

– *Undervisningsformen differentieres således, at alle elever udvikler sig i undervisningsforløbet. Der veksles mellem overbliksskabende forløb, eksperimenter, øvelser og projekter.*

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 f.eks. it-systemer, noter, synopser, journaler, programbeskrivelser og rapporter. Det kan f.eks. gøres i Google drev, en ftp-mappe eller via skolens egne it-systemer.

3.2.1. hhx (uddannelsesspecifikt afsnit)

Projektarbejdsformen bør være fremtrædende i undervisningen, og flere af undervisningsforløbene bør planlægges, så eleverne løser en problemstilling i form af et mindre projekt, der enten kan løses i grupper eller individuelt. Det er vigtigt, at der er fokus på at eleverne arbejder selvstændigt med at opnå projektets mål. Projekterne vil oftest dække mere end et enkelt fagligt mål, da eleverne i processen i at skabe it-systemer naturligt kommer til at beskæftige sig med flere af målene, f.eks. modellering, databehandling, interaktionsdesign, programmering og test. Dette ses som en fordel, da det giver en sammenhæng mellem emnerne i faget.

Specielt i starten af fagets undervisning kan eleverne godt behandle opgaver, der kun træner en bestemt færdighed, hvilket ikke kan betegnes som et projekt. Det kan være nødvendigt, da der ikke kan stilles så høje krav til selvstændighed hos eleverne på dette tidspunkt, ligesom det kan være nødvendigt med øget indholdsmæssig styring i starten jævnfør de didaktiske principper for faget omkring Faded Guidance og Stepwise Improvement. Man kan så senere anvende de færdigheder, som eleverne har opnået, i projekter der dækker flere faglige mål for at opnå sammenhængen mellem målene.

For at give eleverne indsigt i, hvordan faget anvendes i virksomheder og hvilke karrieremuligheder faget giver, bør et eller flere af de projekter eleverne arbejder med, arrangeres som en udadrettet aktivitet i samarbejde med en ekstern partner. Dette går godt i spænd med kravet om, at mindst et af projekterne i undervisningen skal tilrettelægges, så elevernes innovative kompetencer udvikles. Det betyder, at eleverne skal give forslag til løsning af en mindre konkret og virkelighedsnær problemstilling. Løsningsforslaget skal være et værdiskabende it-system, der præsenteres og evalueres.

3.2.2. htx (uddannelsesspecifikt afsnit)

Ved udarbejdelse af et it-system er projektarbejdsformen oplagt. Som regel har eleverne gavn af at kunne gå sammen i grupper, da det kan være svært at overskue alle aspekter af it-udvikling alene. Ofte vil der både være grafiske elementer, lyd, programmering, interaktionsdesign, projektstyring, dokumentationskrav m.m., der vil skulle tages højde for. Ofte vil eleverne have gavn af at kunne sparre med hinanden om disse aspekter. Og de vil som oftest også have forskellige kompetencer, som de kan supplere hinanden med. Det bør sikres, at alle elever arbejder med centrale faglige mål, eksempelvis programmering. Her er det hensigtsmæssigt med undervisning, som sikrer, at alle eleverne bliver trænet heri, før de slippes løs i projektarbejdet. Eleverne kan arbejde med tutorials - enten i form af video eller tekst - eller læreren kan gennemgå procedurer i plenum, for derefter at lade eleverne arbejde med dem selvstændigt for at øge deres kompetencer.

Alternativt kan mindre projekter - eksempelvis involverende simple og mere overskuelige programmer som Scratch o.lign. - tilrettelægges som individuelt arbejde. Herved sikres, at eleverne kommer igennem alle dele af udviklingen hver især.

I forhold til brugertests af elevernes it-systemer kan matrixgrupper være en mulighed. Herved kan eleverne teste hinandens systemer og give hinanden feedback.

3.2.3. stx/hf (uddannelsesspecifikt afsnit)

Projektarbejdsformen bør være den altoverskyggende arbejdsform i faget. Derved opnår man autencitet i forhold til professionsfaget it's arbejdsform. I denne forbindelse er gruppearbejde en god ide, da det at diskutere design og løsningsforslag dels gør det endelige it-system kvalitativt bedre og dels træner mundtlighed, som kommer i spil til eksamen. For at undgå at eleverne, under udvikling af it-systemerne, fordeler delopgaver såsom programmering og rapportskrivning imellem sig, kan man rammesætte undervisningen, så alle elever i en gruppe skal lave alle dele i udviklingen.

Der kan tages udgangspunkt i cases, hvor alle dele af udvikling af et it-system kommer i spil. Disse cases kan eventuelt defineres i samarbejde med eksterne partnere og dermed ses fagets samfundsmæssige og karrieremæssige relevans.

I starten af kursets afvikling er det en god ide at tilrettelægge undervisningen forholdsvis lærerstyret, for derefter løbende at slippe kontrollen og lade eleverne selv styre forløbet. Det er også i starten en god ide at give eleverne delvist udviklede it-systemer og så lade dem forbedre og ændre disse. Dermed kommer eleverne hurtigere i gang med at se dele af it-systemer, der virker.

3.3. It

Eleverne skal undervises, så de kan begå sig digitalt. Ser vi på eleven som forbruger af teknologier og kilder, skal de forholde sig kildekritiske, som de forholder sig kildekritiske i andre fag. Når eleverne designer it-systemer, skal de lave fornuftig dokumentation, og de skal kunne udtrykke sig via digitale medier såsom videopræsentationer og hjemmesider. Eleverne skal reflektere over digitale mediers generelle påvirkning af brugerne, og eleverne skal forholde sig til, hvordan it-værktøjer kan anvendes til at understøtte udviklingsprocessen i et it-projektforløb. Endelig skal eleverne via logbogsarbejde og gruppearbejde anvende it til at dokumentere et forløb, hvor et it-system udvikles, og de skal reflekteret deltage i den fælles læring.

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 almendannende 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.

3.4.1. hhx (uddannelsesspecifikt afsnit)

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 studieretning faget indgår i, kan det stof, der behandles, udvælges så det understøtter studieretningen. For eksempel kan faget i økonomiske studieretninger bidrage med viden om virksomheders anvendelse af data til styring af virksomheden og måske skitsere en it-løsning til et virksomhedsøkonomisk problem. På samme måde kan faget bidrage til en

reflekteret tilgang til anvendelsen af it til markedsføring og salg på marketingsorienterede studieretninger, mens den rolle it spiller i samfundet og internationalt mht. til samarbejde og handel, kan behandles på samfunds- og internationalt orienterede studieretninger.

Desuden kan samspillet med andre fag sikres ved at lade eleverne udvikle it-systemer med udgangspunkt i et andet fags problemfelt.

I SO-forløbet om digitalisering indgår informatik i følgende samspil:

- Et samarbejde mellem fagene matematik, samfundsfag og informatik, hvor der arbejdes med indsamling, behandling og analyse af data og med brug af it og matematik omkring en samfundsfaglig problemstilling.
- Et samarbejde mellem fagene samfundsfag, dansk og informatik, hvor der arbejdes med analyse af debatten og adfærden på de sociale medier.

3.4.2. htx (uddannelsesspecifikt afsnit)

I designorienterede studieretninger kan faget bidrage til, hvordan den kreative proces udspiller sig inden for udviklingen af it. Derudover kan faget bidrage med diskussioner og perspektiver på funktionalitet/æstetik-paradigmet. Hvordan bevares en æstetisk form i et system som også skal være funktionelt, brugervenligt og problemløsende? Hvilke udfordringer støder man på som udvikler af et system, der både skal levere en oplevelse til brugeren og samtidig hjælpe vedkommende til at løse et problem?

I teknologiske studieretninger deler fagene teknologi og informatik et problemløsende fokus, et fokus på at eleverne skal "have fingrene" i et system, et innovationsfokus samt et fokus på, hvordan teknologiske systemer påvirker menneskers sociale liv og samfundet i det hele taget. Informatik kan endvidere supplere teknologifaget gennem fagets fokus på programmering og databehandling til udarbejdelse af teknologiske produkter. Eksempelvis kan et forløb vedr. udviklingen af et "serious game" være oplagt som et tværfagligt teknologi/informatik forløb. Et sådant forløb vil i øvrigt også kunne blive til gennem samspil med samfundsfag, idet eleverne kunne sættes til at lave et computerspil med en politisk agenda.

I forhold til studieområdet er der mange studiekompetencer, der bliver trænet gennem udviklingen af it-systemer, herunder projektarbejdsformen, tidsplanlægning, projektstyring, rollefordeling, innovative processer, idegenerering, datadeling, dokumentation, problemløsningsprocesser, informationssøgning, kvalitative/kvantitative undersøgelser mm.

I forhold naturvidenskabelige fag kan eleverne også arbejde med eksempelvis at simulere et fysik-forsøg gennem et it-system eller udarbejde it-systemer, der formidler naturvidenskabelig læring. I forhold til det faglige mål, der omhandler, hvordan data kan organiseres i databaser, kan der arbejdes med empiriske data fra naturvidenskabelige og tekniske fag. Endelig kan informatik-fagets arbejdsmetode - herunder den iterative proces - sættes over for den naturvidenskabelige arbejdsmetode for at anskueliggøre deres forskelle og særlige kendetegn.

Informatik C i htx grundforløbet.

Informatik C kan indgå i det obligatoriske samspil Naturvidenskabeligt grundforløb (NV).

Flere af de nedennævnte temaer kan tillige adresseres af Kommunikation it/C, hvis skolen udbyder begge C-fag i grundforløbet:

- Digital billedrepræsentation og billedmanipulation.
Kommunikation og it C: digital billedbehandling, billedmanipulation og billedkommunikation.
Informatik: udvikling af små it-systemer der kan foretage billedmanipulationer, fx simple filtre der korrigerer farver, digital repræsentation af billeddata.
- Webjournalistisk og simple hjemmesider; CMS baseret.
Kommunikation og it C: interaktionsdesign, HTML/CSS.
Informatik: simple programmer til styring af interaktivitet/dynamisk indhold – JavaScript.
- Helt basale spil (Scratch) til visualisering af problemstillinger i andre fagfelter.
Kommunikation og it C: Små læringsspil eller simuleringer.
Informatik: biologi (økologiske/fysiologiske modeller); kemi (simple reaktioner); naturvidenskabeligt grundforløb (modeller); fysik (spilmuligheder der viser noget om fysikkens love / linearitet m.m.)

Som det ses af forslagene, vil eleverne arbejde forskelligt med emner og kernestof i Kommunikation og it C og Informatik C ud fra fagenes forskellige faglige mål. Trods forskelle i elevernes kompetencer efter grundforløbet, vil de have opnået de grundlæggende faglige forudsætninger for at fortsætte på studieretninger med enten Kommunikation og it A eller Informatik B.

3.4.3. stx (uddannelsesspecifikt afsnit)

Overordnet set skal faget bidrage til det faglige samspil. Er faget et valghold, kan eleverne være repræsentanter for deres studieretning og derigennem knytte sammenhænge mellem informatik og andre fagfelter.

I naturvidenskabelige studieretninger kan undervisningen f.eks. tage udgangspunkt i empiriske data eller eleverne kan designe it-systemer der simulerer f.eks. et fysikforsøg. I samfundsvidenskabelige studieretninger kan eleverne betone emner som it-sikkerhed og virksomhedsøkonomi. I humanistiske studieretninger kan eleverne særligt arbejde med interaktionsdesign og kommunikationsformer på nettet.

Generelt kan eleverne udvikle it-systemer, der tager udgangspunkt i et andet fags genstandsfelt.

3.4.4. hf (uddannelsesspecifikt afsnit)

Overordnet set skal faget bidrage til det faglige samspil. Eleverne kan være repræsentanter for et enkelt fag og dermed udveksle fags særkende i forhold til it-faget. Et par eksempler: Med udgangspunkt i fysik, kan undervisningen tage udgangspunkt i empiriske data og eleverne kan designe et it-system, der simulerer et fysikforsøg. Med udgangspunkt i dansk kan undervisningen arbejde med kommunikationsformer på nettet og sociale medier. I de tilfælde, hvor faget indgår i en fagpakke, bør faget arbejde med problemstillinger, der tager udgangspunkt i det andet fag i fagpakkens genstandsfelt.

4. Evaluering

4.1. Løbende evaluering

4.1.1. hhx (uddannelsesspecifikt afsnit)

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 evalueres elevernes præstationer, blandt andet via logbogen, og evalueringen bør give en individuel vurdering af niveau og udvikling i det faglige standpunkt i forhold til den forventede udvikling og de faglige mål.

4.1.2. htx (uddannelsesspecifikt afsnit)

Udover at elevernes dokumentationer kan udgøre en form for løbende evaluering, kan der også planlægges tests af elevernes it-systemer undervejs i deres proces. Dette vil styrke den iterative arbejdsproces, som er central for faget og i it-virksomheders arbejde generelt. Man kan som lærer fastsætte deadlines, for hvornår forskellige udgaver af et givent system skal være færdige - altså med andre ord arbejde med versioneringer - hvorved eleverne løbende får mulighed for at stoppe op, få feedback, evaluere deres arbejde og så fortsætte arbejdet. Det er oplagt at lave it-systemer til andre virksomheder, som med fordel kan inddrages i evalueringsprocessen enten løbende eller til sidst i forløbet.

Efter afslutningen af et spilforløb kan man arbejde med såkaldte postmortems - en særlig evalueringsform brugt i spilbranchen med henblik på at identificere positive og negative aspekter ved henholdsvis arbejdsprocessen og det endelige produkt.

4.1.3. stx/hf (uddannelsesspecifikt afsnit)

Den løbende evaluering er både en evaluering af delaspekter af de it-systemer eleverne bruger en stor del af undervisningen på at udvikle og en evaluering af elevernes aktuelle faglige standpunkt.

Den løbende evaluering af elevernes udvikling af it-systemer, kan organiseres som del-deadlines, hvor eleverne skal præsentere delmålene. Her kan gives feedback, enten af underviseren eller af de andre elever. Delmålene kan også beskrives i logbogen. Laves der it-systemer med virksomheder er det oplagt at lade disse deltage i evalueringen.

Den løbende evaluering af den enkelte elevs faglige standpunkt kan tage udgangspunkt i logbogsmaterialet. Her vurderes elevens udvikling af fagligt 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 trækningsenhed ved lodtrækningen af opgaverne. Antallet af eksamensopgaver/trækningsmuligheder 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 trækningsmuligheder skal overstige antallet af eksaminander/grupper med mindst 3.

- *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. En mindre del af it-systemet bør 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 detailløsninger.

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 nederst på opgaven.

Eksamensopgaverne skal afspejle den tematiske undervisningsform som eleverne kender fra undervisningen, dvs. eksamensopgaverne bør også 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. Eksaminanden bør selv have initiativet ved eksaminationens start, så eksaminanden får plads til selvstændigt at strukturere sin præsentation af opgaveløsningen. Præsentationen bør have udgangspunkt i det, under forberedelsestiden, fremstillede it-system, og i mindre grad i teoretiske præsentationer (f.eks. 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 produktære præsentationer eller spørgsmål til elevens egne produktioner ved mere teoretiske præsentationer.

- *Opgaverne og bilag sendes til censor forud for prøvens afholdelse*

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*
 - *kan demonstrere viden om fagets identitet og metoder*
 - *behandle problemstillinger i samspil med andre fag*

Når faget indgår i samspil med efterfølgende prøve i f.eks. 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.

Oversigt over karakterskalaen

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.
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	<p>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.</p> <p>Der argumenteres fagligt velbegrunderet for valg af faglige teorier og metoder.</p> <p>Eksaminanden perspektiverer fagligt kvalificeret sin viden til såvel egne it-produkter som til opgavens teoretiske indhold.</p> <p>Eksaminanden besvarer glimrende og fagligt sikkert uddybende og supplerende spørgsmål under samtalen.</p>

7	God	<p>It-løsningen præsenteres mht. planlægning, gennemførelse og evaluering.</p> <p>It løsningen lever trods en del mangler op til de stillede krav.</p> <p>Der argumenteres for valg af faglige teorier og metoder.</p> <p>Eksaminanden perspektiverer sin viden til såvel egne it-produkter som til opgavens teoretiske indhold.</p> <p>Eksaminanden besvarer uddybende og supplerende spørgsmål under samtalen.</p>
02	Tilstrækkelig	<p>It-løsningen præsenteres sparsomt og knapt mht. planlægning, gennemførelse og evaluering.</p> <p>It løsningen lever minimalt acceptabelt op til de stillede krav.</p> <p>Der argumenteres minimalt acceptabelt for valg af faglige teorier og metoder.</p> <p>Eksaminanden perspektiverer tilstrækkeligt sin viden til såvel egne it-produkter som til opgavens teoretiske indhold.</p> <p>Eksaminanden besvarer sparsomt og knapt uddybende og supplerende spørgsmål under samtalen.</p>