

# Eksamensprojektet i programmering

Søren Præstegaard  
Odense Tekniske Gymnasium

## Resumé

*Her beskrives et forløb i den afsluttende undervisningstid i programmering. Forløbet passer lige godt til C og B-niveau, og inkluderer en række forskellige aktiviteter, med det formål at gøre elevernes projekter relevante for den eksamen der venter.*

Eksamensprojekterne i programmering har en central rolle i eksamensformen - både på C- og B-niveau - og de skal inkludere så mange forskellige aktiviteter, at de danner grundlag for en dækkende og fair vurdering af eleven til eksamen. Uden vejledning fra læreren ender mange elever uden at have arbejdet væsentligt med alle dele af fagets faglige mål. Ved at inkludere sociale, kreative og strukturelle aktiviteter i projektperioden understreges at programmering er mere end blot kodning.

tidligt igang. Planlægningen omfatter beslutninger som:

- Vil man afvikle projektet individuelt eller i mindre grupper? Ved individuelle projekter får læreren travlt med at nå rundt til alle elever, og svage elever kan have svært ved på egen hånd at opnå et resultat med et omfang, der giver mulighed for en god snak til eksamen. Hvis man vælger gruppeprojekter er der omvendt en risiko for at svage elever ikke kommer til at deltage aktivt i arbejdet, og kan have svært ved at forsvare deres deltagelse til eksamen. De fleste elever klarer sig bedre til eksamen med et simpelt program, som de selv har lavet, end de gør med et komplekst program, som de ikke har fuldt overblik over.
- Vil man give eleverne frie hænder til at vælge indholdet i deres projekt, eller vil man diktere et eller flere projekter, som eleverne kan vælge mellem? Ofte er der meget forskellige interesser blandt eleverne på et hold, og det kan være svært at motivere alle elever ligeligt med et begrænset antal projekter. Omvendt er det en stor opgave for læreren at vejlede alle elever til et projekt, der både har passende fagligt indhold og omfang.
- Vil man give eleverne frie hænder til at vælge programmeringssprog og teknologi? Enkelte elever kan have erfaring med sprog der er mere eller mindre ukendte for læreren, og man skal kende eleverne

## INDHOLD

<b>1 Forberedelse af forløbet</b>	<b>1</b>
<b>2 Afvikling af projektet</b>	<b>2</b>
2.1 Opstartsfasen . . . . .	2
2.2 Arbejdsfasen . . . . .	2
2.3 Afslutningsfasen . . . . .	3
<b>3 Evaluering af forløbet</b>	<b>3</b>

### 1. FORBEREDELSE AF FORLØBET

På begge niveauer skal eksamensprojektet have et omfang svarende til 20 undervisningstimer, og derfor skal overvejelserne om eksamensprojektet starte tidligt på skoleåret. Særligt på C-niveau, hvor der ikke er så mange undervisningstimer, fylder forløbet det meste af foråret, og hvis eleverne skal have mulighed for at forberede projektet ordentligt skal man

---

godt, for at vide om de kan håndtere et eventuelt skifte op til eksamen.

## 2. AFVIKLING AF PROJEKTET

Denne beskrivelse deler projektforsløbet op i tre faser, *opstart*, *arbejde* og *afslutning*, men en konkret plan kan have endnu finere granularitet. Det er vigtigt under hele forløbet at hjælpe eleverne med at holde overblikket over tidsplanen, da mange elever oplever at tiden til projektet forsvinder meget hurtigt.

### 2.1. Opstartsfasen

Fra starten vil de fleste elever have en idé om hvordan de kommer igang, men de vil have en tendens til at fokusere på at skrive kode. Behøvet for at arbejde konkret på at producere det færdige produkt er naturligt nok, og for mange elever vil det være nødvendigt for at konkretisere billedet af deres endelige program.

Problemet med at komme for langt med kodningen på dette tidspunkt er, at eleverne ikke bliver bevidst om den proces de er i. De får ikke gjort sig overordnede tanker om projektet, og hvis de senere vender tilbage til at producere abstrakt dokumentation bliver det ofte søgt og konstrueret, da det ikke har indgået som en naturlig del af produktets udformning. For at undgå dette, og for at give eleverne en reel chance for at kunne tale om andet end koden til eksamen, kan man i opstartsfasen overveje at sætte struktur på en eller flere af nedenstående aktiviteter:

- Korte præsentationer af projektidéen for deres kammerater. Dette tvinger eleverne til at formulere konkrete mål for deres projekt, og kammeraternes spørgsmål og input kan hjælpe med at forme idéen.
- Fælles idégenereringsteknikker. Teknikker til at generere en lang række idéer, og efterfølgende teknikker til udvælgelse af de bedste idéer til deres produkt, kan være en god måde at sikre sig at eleverne får foretaget en relevant analyse af deres krav til programmet. Efterfølgende vil de,

på baggrund af denne analyse, kunne redegøre for de valg de har foretaget.

- Delaflevering af for eksempel kravspecifikation, brugerhistorier, skitser eller lignende. Man kan hjælpe eleverne med at holde progressionen i projektet, ved at lade dele projektet op med nogle specifikke delmål undervejs, men man kan også argumentere for, at eleverne i eksamensprojektet selv skal kunne styre deres projekt. Man løber en risiko for at kravspecifikationen bliver afleveret og indgår i elevens synopsis, men at den ikke reelt har været anvendt til noget konstruktivt i projektet.

### 2.2. Arbejdsfasen

Når eleverne slippes fri til at udforme den væsentligste del af deres program, forfalder de nemt til en ustruktureret form, hvor deres overblik over processen sjældent rækker længere end den konkrete ting, de netop nu er igang med. Til gengæld vil de ofte kunne komme langt med deres program, drevet af motivationen for at klare sig godt til eksamen og arbejdet med et projekt de finder interessant. Derfor er det ofte lærerens vigtigste opgave at gøre eleverne bevidst om de aspekter af de faglige mål, der ikke lige handler om koden.

I vejledningen af eleverne kan man hjælpe med at huske den iterative udviklingsmodel. I mange projekter kan den være en naturlig del af udviklingen, f.eks. hvis programmets krav er formuleret som brugerhistorier der kan implementeres én efter én. I projekter med mere tekniske krav kan man vejlede eleverne til at bruge *stepwise improvement*<sup>1</sup> og løbende dokumentere programmets udvikling.

Til eksamen skal eleverne helst demonstrere at de kan bruge de programmeringsteknikker man har arbejdet med i undervisningen. Det vil typisk være de mest gængse kontrolstrukturer og datatyper, måske design- eller algoritmemønstre, og mange elever vil have langt nemmere ved at demonstrere dette, hvis de kan tale om hvordan de konkret har anvendt

---

<sup>1</sup>Se afsnit 3.1 i læreplanerne eller artiklen om fagets metoder: <http://iftek.dk/fagets-metoder>

---

disse ting i deres projekt. Når eleverne løbende udvider deres program, kan man vejlede dem i retning af de udvidelser, der giver det mest varierede indhold i programmet.

sition for deres præsentation af projektet. Eleverne kan så mødes i mindre grupper og hjælpe hinanden med at få opbygget en hensigtsmæssig præsentation.

### 2.3. Afslutningsfasen

Mod slutningen af projektperioden vil mange elever begynde at udforme deres synopsis, og afhængig af hvor meget man har trænet det skriftlige arbejde på forhånd kan de få brug for vejledning til at opbygge den. En skabelon kan være en måde at skabe en kontrakt mellem elev og lærer, men ofte er de enkelte projekter så forskellige, at en skabelon ikke kan passe til alle projekter på holdet. Så opnår man blot at eleverne inkluderer afsnit, som ikke har nogen relevans for deres projekt, i deres synopsis.

I stedet bør man tale med eleverne om hvordan deres synopsis skal bruges til at forberede censor og eksaminator til eksamen. Eleverne sender et signal om hvad de har lagt vægt på når de udvælger indhold til deres synopsis, og det er med til at forme den samtale man har til eksamen. En fyldestgørende synopsis indeholder både overordnede beskrivelser af programmets opbygning og funktion, samt relevante forklaringer om de programmeringsteknikker, der var nødvendige for at opnå det resultat der er nået. Eleven kan f.eks. demonstrere at de behersker nogle abstrakte dokumentationsformer, og dermed behøver man ikke bruge så meget tid på dette til selve eksaminationen.

## 3. EVALUERING AF FORLØBET

Ved afslutningen af projektet ved eleverne ikke om de skal til eksamen i programmering, og der kan nå at gå mange ting gennem deres hovede inden de ser deres program igen. En grundig refleksion over hvad eleven har gjort og lært af projektet, kan hjælpe med at få tingene til at bundfælde sig.

Hvis man har gemt et enkelt modul af undervisningstiden til at tale om eksamensformen efter aflevering af projektet, kan man bede eleverne forberede sig ved at lave en dispo-