



Softwareudvikling

**Retningslinier for planlægning,
gennemførelse og dokumentation**

Forord

Denne lille vejledning er udarbejdet som et hjælpemiddel ved udvikling af software og kan med fordel anvendes såvel af virksomheder, der allerede arbejder med softwareudvikling, som af de, der skal i gang med det.

Der er grund til at understrege, at det for mindre projekter kan virke overvældende at skulle følge de foreslåede retningslinier. Det anbefales imidlertid, at man ikke går på akkord, da brugen af reglerne i høj grad kan hjælpe til at sikre også et mindre projekts succes.

Teknologistyrelsen takker

Kvalitetschef Jørgen N. Møller, Terma Elektronik A/S
Civilingeniør Poul Grav Petersen, Elektronikcentralen og
konsulent Ole Skyum, Jydsk Teknologisk Institut
for faglig bistand ved vejledningens udarbejdelse.

Mogens Kring

Udvikling af software

Udvikling og vedligeholdelse af software er kompliceret og omkostningskrævende. Derfor er det nødvendigt at planlægge og gennemføre arbejdet efter retningslinjer, som i praksis har vist sig at føre til bedst mulige resultat.

Disse regler har flere formål:

- Totalt set at billiggøre udvikling og vedligeholdelse af software.
- Gennem dokumentation at vise over for omgivelserne, at der rent faktisk eksisterer et software-produkt.
- Desuden vil det opfylde Teknologistyrelsens krav til softwareudvikling, som kan opnå støtte.

Et sådant sæt retningslinjer beskrives i det følgende.

Teknologistyrelsen vil i sin fremtidige medfinansiering af softwareudvikling forudsætte, at disse som et minimum følges. Andre indarbejdede metoder, som på tilsvarende måde kan sikre et positivt resultat, vil også kunne accepteres.

Ved større projekter (5–10 mandår) og i store udviklingsorganisationer har det vist sig nødvendigt at supplere med yderligere dokumentation og en mere formel indarbejdet kvalitetssikring.

I forbindelse med forskningsopgaver kan det være nødvendigt med en anden form for udviklingsmetodik, end den her beskrevne.

Planlægning og organisering

Før et udviklingsarbejde sættes i gang er det nødvendigt at udarbejde en plan for, hvordan arbejdet skal gribes an. Denne plan kan deles op i underplaner, f.eks.:

- A. En programudviklingsplan, der tids- og arbejdsmæssigt opdeler projektet i en række faser.
- B. En konfigurationsstyringsplan, der redegør for, hvordan man holder styr på resultaterne af udviklingsarbejdet.

C. En kvalitetssikringsplan.

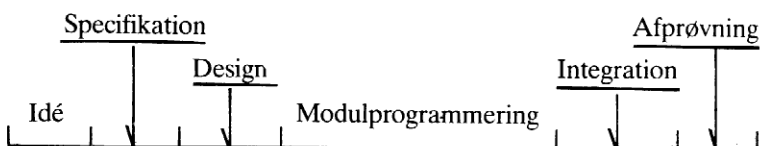
Hvis ikke de her nævnte planer overholdes, kan et softwareprodukts pålidelighed og kvalitet ikke sikres. Men følges de, er der øget sandsynlighed for, at kravene til projektets økonomi, tidsforløb og tekniske indhold opfyldes.

A. Programudviklingsplan

Den tidsmæssige og arbejdsmæssige plan vil normalt kunne opdeles i seks faser, der hver omfatter en logisk defineret og afgrænset del af det samlede udviklingsarbejde.

For at kunne opnå en heldig gennemførelse af projektet, er det en forudsætning, at hver fase gennemføres i henhold til det mål, der er sat for den. Desuden skal de opnåede resultater dokumenteres, så de kan danne grundlag for en vurdering af, om projektet skal standses i denne fase, eller om det kan gå videre til næste fase. Denne vurdering skal naturligvis være positiv, hvis næste fase skal igangsættes.

De seks faser er vist på følgende tidsakse, og hver fase vil blive nærmere beskrevet.



1. Idéfasen

Denne fase beskriver produktideen. En projektbeskrivelse skal redegøre for det tekniske indhold, de markedsmæssige aspekter, den forventede rentabilitet, projektets budget og tidsplan samt de krav til personale, organisation og anlæg, der er nødvendige for en succesfuld gennemførelse af projektet.

Formålet med denne beskrivelse er at fremskaffe et materiale, der kan bruges som:

- Beslutningsgrundlag for bestyrelsen/direktionen om projektets igangsættelse
- Grundlag for ansøgning om medfinansiering f.eks. fra Teknologistyrelsens erhvervsfremmeordninger.

2. Specifikationsfasen

Kravene til det samlede produkt skal opstilles i denne fase. Eventuelle kundeønsker skal med på dette tidspunkt. Desuden vedtages der standards m.v.

Specifikationsfasen afsluttes med en »Produkt- og Kravspecifikation« og en redegørelse for, hvordan det til sin tid bliver verificeret, d.v.s. kontrolleret og dokumenteret, at de stillede krav er opfyldt (Acceptance Test Plan).

Det gælder som en hovedregel ved udvikling af software, at der aldrig må stilles et krav uden samtidig angivelse af, hvordan det skal verificeres, at resultatet også overholder de stillede krav.

3. Designfase

»Produkt- og Kravspecifikationen« er udgangspunkt for, hvordan software-produktet tænkes designet. I designfasen skal der tages stilling til:

- hvilke fremgangsmåder, der skal benyttes ved programmeringen
- hvilke moduler programmet skal opdeles i (modulspecifikation)
- hvordan der kommunikeres internt mellem modulerne
- hvordan der kommunikeres med omverdenen (interfacespecifikation)

Det er en god regel ved opdeling i moduler at nedbryde til så små enheder, at de kan forstås, håndteres og løses.

Endelig skal der i denne fase tages stilling til, hvordan integrations-testen gennemføres – d.v.s. hvordan det samlede software-produkt skal afprøves, efter at modulerne er bragt til at arbejde sammen.

I designfasen skal der udarbejdes følgende dokumentation:

- En designbeskrivelse med
 - Modulspecifikation
 - Interfacespecifikation
 - Integrationstestbeskrivelse

Dokumentationen skal bruges ved den vurdering, som skal finde sted inden der fortsættes med programmeringen af de enkelte moduler. Netop denne vurdering (critical review) er af stor vigtighed for en heldigt gennemført softwareudvikling.

Vurderingen skal omfatte en stillingtagen til tid- og pengeforbrug, om målet for designfasen er nået, og om der er mulighed for at gennemføre projektet som planlagt.

Hvis der opstår tvivl om mulighederne, er det vigtigt at arbejde videre med problemerne og få dem løst inden der går videre til næste fase.

4. Modulprogrammering

I denne fase gennemføres tre hovedaktiviteter:

1. Moduldesign
2. Modulkodning
3. Modulafprøvning

Denne rækkefølge skal holdes for hvert enkelt modul. Men der er ikke noget i vejen for at lave samtlige moduldesign først, lave alle tre aktiviteter for et modul ad gangen eller en mellemting – blot rækkefølgen holdes.

Moduldesignet skal dokumenteres i en modulbeskrivelse. Samtidig skal der tages stilling til, hvordan modulafprøvningen skal ske, og dette må beskrives i en modulafprøvningsplan. Moduldesignet gennemføres i overensstemmelse med de standarder for strukturering af programmer og for anvendelse af designteknikker, der er fastlagt for projektet.

Modulkodningen skal ske på basis af modulbeskrivelsen i det valgte programmeringssprog. Projektets standarder for kodning og kommentering følges.

Selve modulafprøvningen foretages i overensstemmelse med den plan, der er udarbejdet. De fejl, der findes, skal rettes uden at overskride de regler, der gælder for moduldesign og modulkodning.

5. Integrationsfasen

De færdige, afprøvede og godkendte modulprogrammer skal i denne fase sammensættes til det komplette produktprogram. Derefter sker der en afprøvning for at vise, at programmet opfylder de specifikationer, der er angivet i designfasen.

6. Afprøvning

Den endelige afprøvning af det færdige softwareprodukt foregår nu i henhold til »Acceptance Test Plan«, og afprøvningen dokumenteres i en slutrapport.

B. Konfigurationsstyringsplanen skal beskrive hvordan:

- dokumenter og programdele identificeres
- man styrer og kontrollerer gennemførelsen af ændringer
- man undervejs i projektforløbet kan gøre status over projektets delelementer.

C. Kvalitetssikringsplanen skal gøre rede for følgende:

- hvilken dokumentation, der skal udarbejdes, og hvorledes tilstrækkeligheden af den sikres gennem reviews eller godkendelser
- hvilke standarder, der skal følges for dokumentation og programmering, og hvorledes overholdelsen af dem sikres
- hvilke reviews, der skal afholdes, med hvilke deltagere og hvordan
- hvilke metoder, teknikker og værktøjer, der anvendes for at understøtte kvalitetssikringen
- hvorledes organisationsstrukturen ser ud, hvem der er projektleder, og hvilket ansvar og beføjelser vedkommende har og hvilke andre, der er ansvarlige for, at de enkelte faser bliver gennemført efter de stillede krav

Dokumentation

Dokumentation er det gennemgående nøgleord i de foranstående 3 planer. Det skyldes, at uden dokumentation eksisterer software-produktet groft sagt ikke.

I hele projektforløbet skal udarbejdes en fyldestgørende dokumentation f.eks. i form af beskrivelser, specifikationer, vejledninger, afprøvningsforskrifter, rutediagrammer, rapporter – i det hele taget alle de tanker, som har ligget til grund for de valg, man har truffet. Dokumentation er ikke noget der skal laves, når udviklingsarbejdet er færdigt.

Dokumentation er den væsentligste del af selve udviklingsarbejdet på samme måde som det ved mekanisk konstruktion er nødvendigt at lave samlingstegninger og detailtegninger, inden selve produktet kan fremstilles. Formålet med dokumentationen er at:

- danne grundlag for beslutninger
- medvirke til at problemer blive gennemanalyseret
- muliggøre fejlfinding
- muliggøre vedligeholdelse
- muliggøre opfølgning
- sikre at firmaet bevarer know-how
- muliggøre dynamisk ressource fordeling

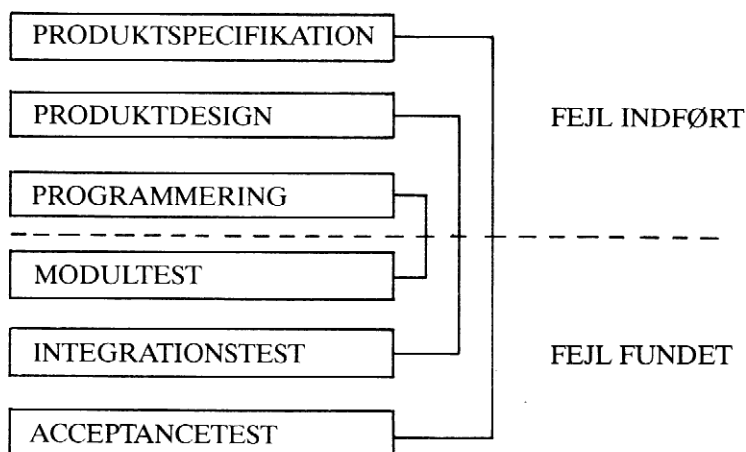
Endelig skal der ske reviews, d.v.s møder hvor de personer, der er med i udviklingen af software foretager en teknisk,- tidsmæssig- og økonomisk opfølgning.

Omkostningsmæssige fordele

Erfaringsmæssigt gør en gennemført kvalitetssikring ikke udviklingsarbejdet dyrere. Tværtimod er det blevet anslået, at omkostningerne til udvikling og vedligeholdelse af software – set over et produkts totale livscyklus – kan blive op til 10–30 procent lavere med en konsekvent kvalitetsstyring.

Det skyldes, at det er muligt at finde fejl på et tidligt tidspunkt i udviklingsforløbet, hvor det endnu ikke er blevet for kostbart at rette dem.

Jo senere en fejl opdages, jo dyrere bliver den at rette. Hvis der slipper fejl igennem i de første faser i et projekt, viser figuren, hvornår disse fejl normalt først opdages.



En tidligt begået fejl, som først opdages sent, er dyr at rette, da den normalt vil kræve rettelser i alle mellemliggende faser. Det betyder endvidere stor risiko for, at nye fejl opstår som følge af rettelser.

Fasernes økonomiske belastning

De enkelte faser i projektet belaster den samlede økonomi forskelligt. Erfaringsmæssigt tales der om følgende belastninger:

- | | |
|---------------------------|--------|
| ● Specifikationer | 5–15% |
| ● Designfase | 15–25% |
| ● Modulprogrammeringsfase | 40–60% |
| ● Integrationsfase | 10–20% |
| ● Afprøvningsfase | 5–10% |

Som det fremgår, er der vide rammer for de enkelte faser. Hovedtenden-
den er, at med voksende projektstørrelse vil tidsforbruget til
programmeringen falde, mens det vil stige til integration og test.
Desuden spiller det ind, at jo mere erfaring en virksomhed har med
lignende projekter, jo mindre tid går der med design og jo mere til
programmering.

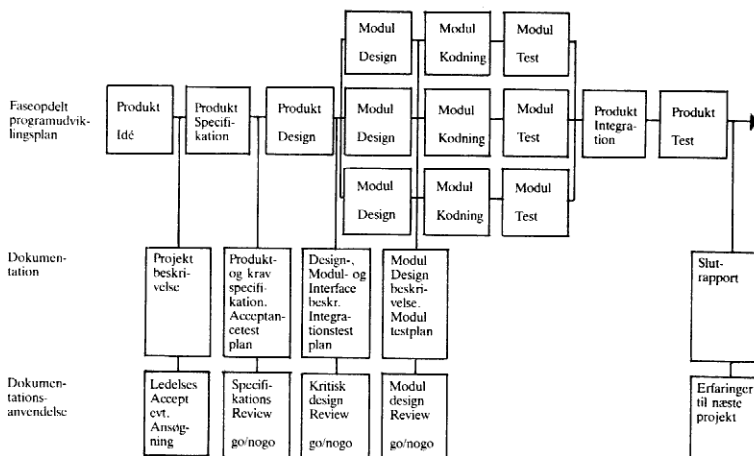
Litteraturliste

- ANSI/IEEE std 730-1981, IEEE Standard for Software Quality Assurance Plans.
- E. H. Bersoff, »Elements of Software Configuration Management«, IEEE Transactions on software Engineering, Vol. 10, No. 1, pp. 79–87, 1984.
- B. W. Boehm, »Software Engineering Economics«, Prentice-Hall, Inc., 1981.
- B. W. Boehm, »Software Engineering Economics«, IEEE Transactions on Software Engineering, Vol. 10, No. 1, pp. 4–21, 1984.
- F. P. Brooks, Jr., »The Mythical Man-Month, Essays on Software Engineering«, Addison-Wesley Publ. Co., 1975.
- W. Bryan and S. Siegel, »Making Software Visible, Operational and Maintainable in a Small Project Environment«, IEEE Transactions on software Engineering, Vol. 10, No. 1, pp. 59–67, 1984.
- R. Dunn og R. Ullman, »Kvalitetssikring af Edb-programmel«. Jydsk Teknologisk Instituts Forlag, 1983.
- S. L. Hansen, P. G. Petersen og B. Runge, »Metoder til organisation af programmeludviklingsprojekter«, SMT-02, Elektronikcentralen, 1981.
- IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans.
- IEEE Std 829-1983, IEEE Standard for Software Test Documentation.
- IEEE Std 830-1984, IEEE Guide to Software Requirements Specifications.

- S. Klim, »Systematic software testing for microcomputer systems«, ECR-155, Elektronikcentralen, 1984.
- P. G. Petersen, »Kvalitetssikring af software«, ECR-112, Elektronikcentralen.
- H. Scherwin, »Softwaredokumentation«, ECR-141, Elektronikcentralen, 1983.
- R. C. Tausworthe, »The Work Breakdown Structure in Software Projects Management«, The Journal of Systems and Software, Vol. 1, No. 3, pp. 181–186, 1980.
- Kvalitetsstyring af Softwareudvikling, Dansk Forening for Kvalitetsstyring, 1984.

Modelskitse over forløbet

Det samlede forløb i det totale softwareprojekt er vist i følgende figur. Den viser samtidig, hvilken dokumentation der skal udarbejdes samt hovedanvendelsen af den.



Støtte til udvikling af software

I forbindelse med Folketingets vedtagelse af Det Teknologiske Udviklingsprogram er der nu forøget mulighed for at opnå medfinansiering af software udvikling. Dette program skal bl.a. støtte:

1. Fremme af informationsteknologi.
2. Udvikling og indførelse af produktionssystemer baseret på informationsteknologi.
3. Udvikling og indførelse af informationsteknologi i produkter og processer.

De her beskrevne regler/retningslinjer for udvikling af software vil Teknologistyrelsen normalt forlange fulgt for på den måde at efterkomme programmets overordnede mål om størst mulig samfundsmæssig nytte af de bevilgede midler.

Adresse:

Telefon:

Teknologistyrelsen
Tagensvej 135
DK-2200 København N

01 - 85 10 66