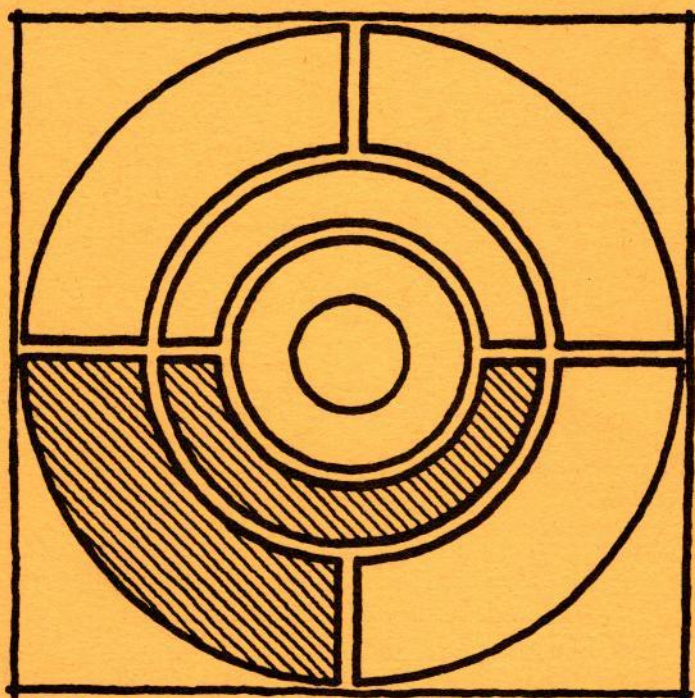
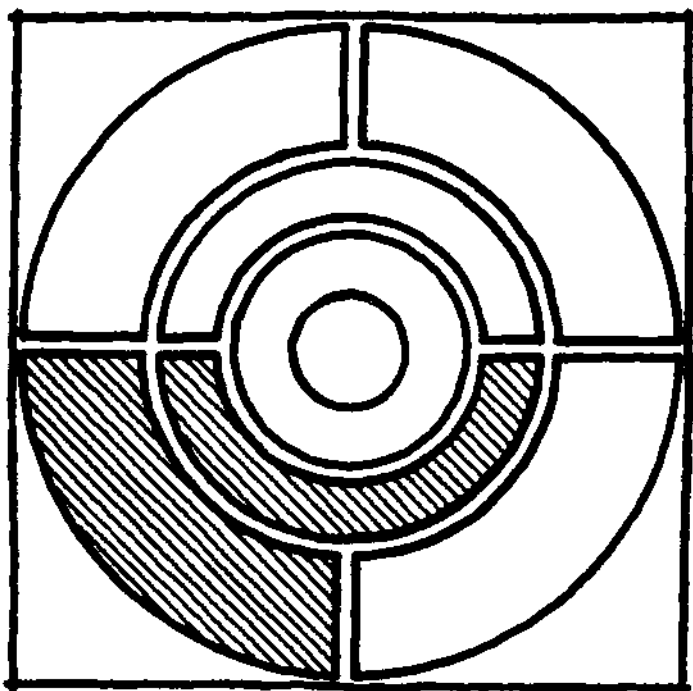




Metodesektionen



BRUG AF PROTOTYPER



BRUG AF PROTOTYPER

Brug af Prototyper

© I.S. Datacentralen af 1959

Første udgave sep.1987

udgiver og vedligeholder:

Metodesektionen i Teknikstaben

fordelingsnøgle: **MHPT**

INDHOLDSFORTEGNELSE

| | |
|--------------------------------------|----|
| Indledning | 4 |
| Hvad, Hvorfor og Hvordan? | |
| Hvad er prototyping? | 6 |
| Hvorfor prototyping? | 6 |
| Hvordan prototyping? | 7 |
| Fremgangsmåde ved prototyping? | 13 |
| Problemer og Faldgruber | 14 |
| Råd og vink om fremgangsmåder | |
| Planlægning | 18 |
| Projektets forudsætninger | 18 |
| Mulighed for genforhandling | 19 |
| Kvaliteten af beskrivelser | 20 |
| Brugerdeltagelse | 20 |

| | |
|-----------------------------------------------|-----------|
| Eablering af projekt | 21 |
| Ambitionsniveau ud fra mål | 21 |
| Prototypen i forhold til et produktionssystem | 22 |
| Design af funktioner | 22 |
| Design af datamodel | 23 |
| Flere fremgangsmåder | 23 |
| Sammenhænge og grænseflader | 23 |
| Vurdering af prototyperne | 24 |
| Standarder for system og dokumentation | 24 |
| Udviklingsmiljø og produktionsmiljø | 25 |
| | |
| Referencer | 27 |

BRUG AF PROTOTYPER
I
DC-PROJEKTER

INDLEDNING

- * Brug af prototyper erstatter ikke grundig analyse og design, men bidrager til øget viden om krav, ønsker og konsekvenser.*
- * Brug af prototyper understøtter brugerdeltagelse i systemudvikling og forbedrer kvaliteten af edb-systemet.*
- * Brug af prototyper er en læreproces, der kræver omhyggelig styring og muligheder for revision af ambitioner og tidsplaner.*

Dette er vigtige egenskaber ved brug af prototyper, som man skal have for øje, når man overvejer at bruge prototyper i et projekt.

Formål

Dette skrift er en vejledning i brug af prototyper i DC-projekter. Formålet med vejledningen er at give støtte til beslutningen om at bruge prototyper i et projekt, og til hvordan prototyper skal anvendes. Vejledningen giver råd og vink om gennemførelse af et projekt med brug af prototyper. Vejledningen er tænkt brugt ved diskussioner mellem projektdeltagere på seminarer, kurser og møder.

I det følgende bruges det kortere engelske ord 'prototyping' i stedet for vendingen 'brug af prototyper'.

Målgruppe

Den primære målgruppe for vejledningen er chefkonsulenter, projektledere og tilbudsgivere, som er ansvarlige for den overordnede udformning af systemudviklingsprojekter. Derudover er målgruppen alle systemudviklere på DC samt andre deltagere i DC-projekter.

Indhold

Vejledningen omfatter først en præcisering af, *hvad* vi forstår ved prototyping, *hvorfor* det er relevant at anvende prototyping, og *hvordan* det kan gøres. Der beskrives hvilke fordele, der er ved prototyping, og hvordan de forskellige fremgangsmåder ved prototyping kan variere med hensyn til hensigt, grundlag, udformning, afprøvning og udviklingsforløb. Der skitseres modeller for, hvordan prototyping kan indgå i et systemudviklingsforløb sammen med andre metoder, og hvordan fremgangsmåden ved prototyping kan forløbe.

De næste afsnit gennemgår problemer og faldgruber, som man kan komme ud for ved prototyping, og giver gode råd og vink om fremgangsmåder. De vigtigste blandt disse er måske følgende:

- planlæg og dokumenter fremgangsmåden,
- skriv kontrakt med muligheder for genforhandling,
- vær omhyggelig ved etablering af brugerdeltagelse,
- gør opmærksom på mangler i prototyperne,
- opstil kriterier og procedurer for vurdering,
- lav prototyper ud fra en normaliseret datamodel.

Men der er mange flere.

Baggrund

Vejledningen bygger på erfaringer i DC-projekter, som er beskrevet i referater fra interview med projektdeltagere. Desuden bygger den på noget af den efterhånden omfattende litteratur om prototyping. På side 27 findes referencer til baggrundsmaterialet.

Udgiver

Vejledningen er udarbejdet og udgivet af Metodesektionen i Teknikstaben. Kommentarer til vejledningen skal gives til Metodesektionen, som er ansvarlig for vedligeholdelsen.

HVAD, HVORFOR OG HVORDAN ?

Hvad er prototyping ?

Hvad

Prototyping er en fremgangsmåde, der består i at udvikle tidlige prøveversioner - prototyper - af produktet og derudfra vurdere og justere de forskellige forslag til design. I forbindelse med udvikling af edb-systemer benævnes fremgangsmåden også eksperimentel systemudvikling, systemudvikling med prototyper eller interaktiv systemudvikling. Prototyperne består ofte af kørende edb-systemer, men de kan også bestå af f.eks. papirattrapper af skærbilleder.

Hvorfor prototyping ?

Hvorfor

Det er svært at danne sig et billede af det færdige edb-system ud fra beskrivelser på papir, især beskrivelser i naturligt sprog. Det er derfor meget begrænset, hvad der bliver givet af kritik af designforslag, der præsenteres på papir. Mange krav og ønsker til systemet kommer ofte først frem, når det færdige system bliver præsenteret, og så er det dyrt og svært, ja måske tidsmæssigt umuligt at tilgodese kravene og ønskerne. Prototyping er et forsøg på at gøre noget ved de vanskeligheder.

Prototyping kan forbedre brugerdeltagelsen i systemudviklingen og give brugerne bedre mulighed for at forstå, vurdere og komme med forslag til det kommende edb-system.

Fordele

Eksempler på fordele, der er oplevet i DC-projekter:

- Prototyping forøger brugernes tilfredshed med det endelige system, idet der skabes et tillidsforhold mellem brugere og systemudviklere, og det endelige system lever i højere grad op til brugernes forventninger.

- Prototyping forbedrer produktiviteten i systemudviklingen, idet der anvendes værktøjer, der hurtigere giver synlige resultater, således at systemudviklernes overblik og motivation øges. Samtidig giver prototyping en stor ansvarsfølelse hos deltagerne over for udformningen af det nye edb-system.
- Prototyping processen giver systemudviklerne en viden om det arbejdsområde, hvor edb skal anvendes. Og brugerne får en viden om muligheder og begrænsninger ved anvendelse af edb. Denne viden er af uvurderlig betydning for gennemførelse af hele systemudviklingsprocessen.
- Prototyping understøtter brugerdeltagelse ved både overordnet og detaljeret design af edb-systemet.
- Prototyping af skærbilleder giver et godt grundlag for design af databasen, idet sammenhængen mellem dataene bliver kendt.
- Prototyping giver et mere gennemtænkt design. Der bruges også mere tid ved design af database og funktioner end traditionelt, men det spares rigeligt ind senere ved den endelige kodning.
- Prototyperne er gode til demonstration af det nye system over for en bred kreds af brugere ved informationsmøder og brugeruddannelse.

Hvordan prototyping ?

Hvordan

Prototyping kan praktiseres på mange forskellige måder. De forskellige fremgangsmåder ved prototyping kan variere med hensyn til hensigt, grundlag, udformning, afprøvning og udviklingsforløb.

Hvad er hensigten med prototyperne ?

Hensigt

Er hensigten at udvikle visioner om det nye edb-system og forbedre beskrivelsen af krav og ønsker til det nye system, eller er hensigten trinvis at udvikle det endelige system ?

I litteraturen, Cerveny(86), beskrives flere niveauer af prototyping:

1. 'Input/Output Design'. Rapporter og/eller skærmbilleder demonstreres nogenlunde som de vil komme til at fremtræde i det endelige system, men uden et kørende system bagved, der kan gemme, hente og behandle data. Nyttige værktøjer er rapportgeneratorer, skærmbilledgeneratorer, tekstbehandlingssystemer og simuleringssystemer.
2. 'Heuristic Design'. Brugergrænseflader kan demonstreres som under 'Input/Output Design', men derudover kan de bagvedliggende edb-funktioner udføres i begrænset omfang, herunder databasehåndtering. Nødvendige værktøjer er en skærmbilledgenerator, et databasehåndteringssystem og en rapportgenerator, der hænger sammen (= et 4.generationstværværktøj).
3. 'Adaptive Design'. Prototyperne ændres gradvist til at blive det endelige system. Brugere og systemudviklere eksperimenterer med prototyper, indtil der er udviklet et system, der kan bruges i drift. Prototypeværktøjerne skal være så effektive, at der kan laves et driftsklart system.

Hvad er grundlaget for prototyperne ?

Grundlag

Udvikles prototyperne ud fra en meget løs skitse af en ide til et system, eller udvikles prototyperne ud fra en omfattende beskrivelse af det funktionelle design af et system, f.eks. en datamodel og funktionsbeskrivelser ?

Hvad omfatter prototyperne ?

Udformning

Omfatter prototyperne en fuldstændig realisering af udvalgte dele af det funktionelle design (vertikale prototyper), eller kan prototyperne demonstrere alle sider ved det funktionelle design, men uden at alle detaljer er realiseret (horisontale prototyper) ?

Forskellige aspekter af et nyt system kan afklares ved brug af prototyper: brugergrænseflader, systemets funktioner og tekniske krav (f.eks. behandlingstid og lagerkrav). I litteraturen, bl.a. Cerveny(86), diskuteres hvilke typer af systemer, der er bedst egnet til prototyping. Der skelnes mellem tre typer af systemer: Transaktionsbehandling, Rapportudskrivning og Beslutningsstøtte (Sagsbehandling). Et systemudviklingsprojekt vil ofte omfatte udvikling af alle tre typer af systemer, idet der vil være delsystemer af hver af de tre typer. Til de tre typer af systemer anbefales følgende niveauer af prototyping:

Transaktionsbehandling: 'Input/Output Design'(niveau 1)

Rapportudskrivning: 'Heuristic Design'(niveau 2)

Beslutningsstøtte: 'Adaptive Design'(niveau 3)

Eksempler på prototyper:

- Attrapper (mock-ups) af skærbilleder.
- En serie af skærbilleder i en dialogstruktur uden data.
- En skærmdialog, hvor data kan gemmes og vises.
- Et brugerudviklet system på en PC.
- Et standardsystem.
- Et driftsklart system med lille kapacitet og begrænsede sikkerhedsfaciliteter.

Hvordan vurderes og justeres prototyperne ?

Afprøvning

Vurderes prototyperne af systemudviklerne, af udvalgte brugere eller af samtlige brugere ?

Afprøves prototyperne på DC, i simulerede brugsomgivelser eller i en virkelig brugssituation ?

Tilrettelægges vurderingen som en uformel afprøvning, som en systematisk undersøgelse ud fra på forhånd fastlagte spørgsmål, eller som noget imellem disse to ekstremer ?

Justeres prototyperne under afprøvningen eller sker justeringen mellem to afprøvninger ?

Hvordan indgår prototyping i systemudviklingsforløbet ?

Udviklingsforløb

Indgår prototyping som et supplement i et traditionelt systemudviklingsforløb med analyse-, design- og programdokumentation, eller er prototyperne i sig selv dokumentationen ?

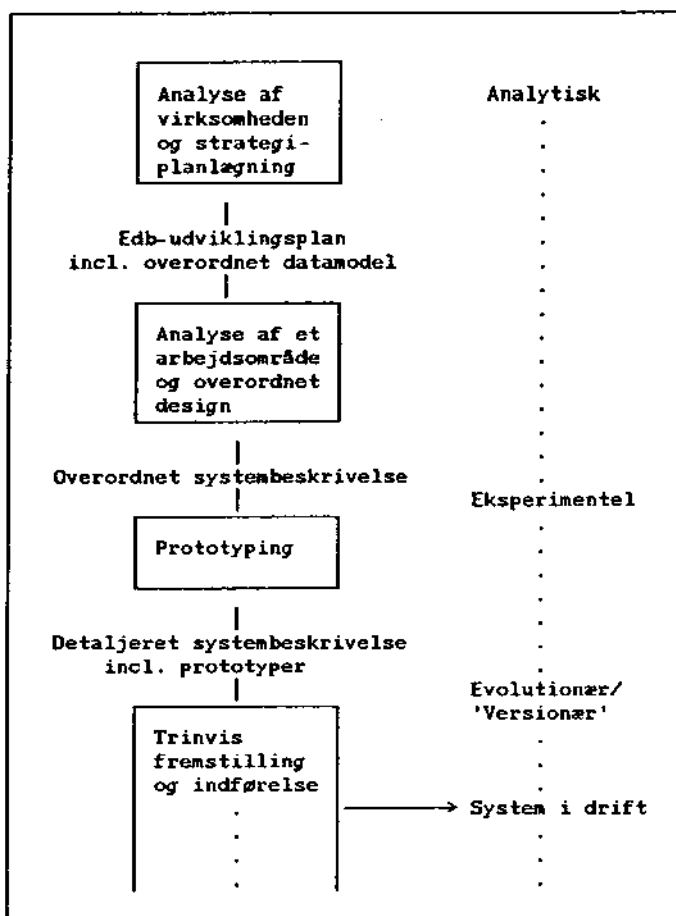
Bliver prototyperne videreudviklet til at blive det endelige system, eller bliver prototyperne kasseret og prototyping efterfulgt af en nyudvikling af det endelige system, f.eks. i et andet programmeringssprog ?

Hvordan kommer man fra prototyperne til det endelige system med optimering, dokumentation, test m.m.?

Eksempler på udviklingsforløb:

Prototyping kan anvendes i en kombineret analytisk, eksperimentel og evolutionær fremgangsmåde som illustreret i figur 1. En analytisk fremgangsmåde anvendes i starten ved analyse af virksomheden totalt og af enkelte arbejdsområder. Der laves edb-udviklingsplaner og overordnet design af virksomhedens data samt design af edb-

systemer til brug i de enkelte arbejdsområder. Ud fra det overordnede design anvendes en eksperimentel fremgangsmåde med prototyping ved det detaljerede design af edb-systemerne. Derefter udvikles og indføres de driftsklare systemer i en evolutionær fremgangsmåde, hvor systemerne indføres trinvis, og hvor vedligeholdelse



Figur 1.

Brug af prototyping i et systemudviklingsforløb med en kombination af en analytisk, eksperimentel og evolutionær fremgangsmåde.

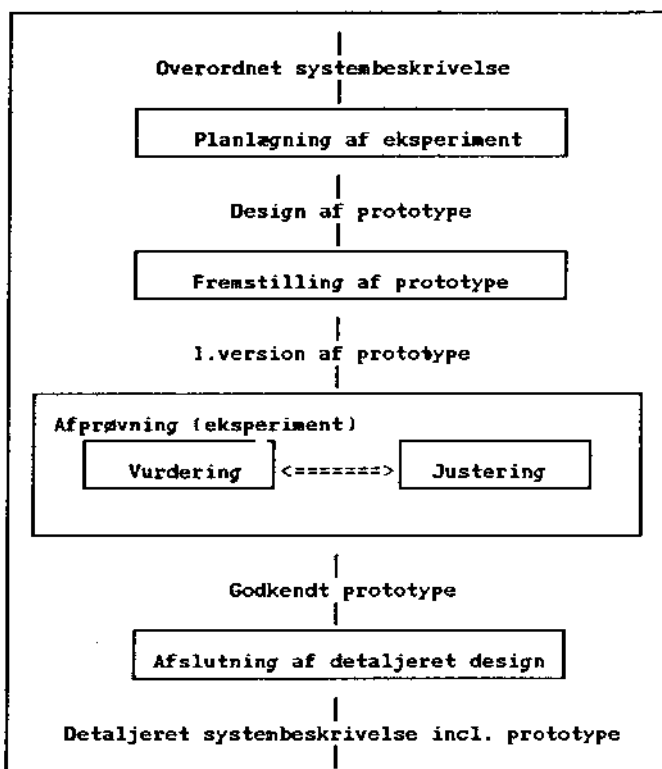
flyder sammen med udvikling. I DC-projekter vil det dog altid være nødvendigt at lave en klar adskillelse mellem udvikling og vedligeholdelse, idet DC altid vil have et kundetilhørsforhold til brugerorganisationen. En rendyrket evolutionær fremgangsmåde kræver et meget tæt samarbejde mellem systemudviklere og brugere, helst med personsammenfald således at enkelte brugere i perioder er systemudviklere. Men den trinvise igangsætning kan med fordel anvendes i DC-projekter. Denne fremgangsmåde kan lidt populært betegnes 'versionær'.

Et projekt, der er baseret på et tilbud i en udbudsforretning, kan tage udgangspunkt i en overordnet systembeskrivelse fra udbudsmaterialet, og projektet kan med fordel starte med en eksperimentel fremgangsmåde med prototyping. Dette stiller dog krav til kvaliteten af udbudsmaterialet og til udformningen af kontrakten jvf. beskrivelsen af problemer med prototyping.

Prototyping kan også anvendes sideløbende med en analytisk fremgangsmåde ved udarbejdelsen af det overordnede design.

Fremgangsmåde ved prototyping.

Prototyping omfatter et eller flere eksperimenter med prototyper. Ved planlægningen af eksperimentet skal formålet fastlægges. Kriterier og procedurer for vurdering af prototypen skal beskrives, og de funktioner, der skal indgå i prototypen, skal udvælges. Derudfra fremstilles 1. version af prototypen, og selve eksperimentet, der består af en iterativ udførelse af vurdering og justering, kan gennemføres. Resultatet af eksperimentet indgår i det videre arbejde med det detaljerede design.



Figur 2.
Fremgangsmåde ved et eksperiment med en prototype.

PROBLEMER OG FALDGRUBER.

Problemer og faldgruber

Hvis man skal opnå fordelene ved prototyping, er det vigtigt at være opmærksom på mulige faldgruber for gennem planlægning at forsøge at undgå dem. Viden om faldgruber er også nyttig for at kunne håndtere de problematiske situationer, der opstår.

Ud fra erfaringer i DC-projekter kan der peges på følgende problemer og faldgruber. Afsnittet med råd og vink indeholder en nærmere beskrivelse af de råd til fremgangsmåden, som er angivet i forbindelse med de enkelte problemer og faldgruber.

Styring

Prototyping er svær at styre. Der skal gives tilstrækkelig tid til at udvikle, vurdere og justere prototyperne, men der skal også afsættes tid til at gøre et produktionssystem færdigt. Hvornår skal den iterative proces stoppes? Skal alle nye krav og ønsker tilgodeses? Er det realistisk at tilgodesse alle krav og ønsker i produktionssystemet? Hvordan kan graden af fremdrift i projektet vurderes og dokumenteres? En prototyping proces kan virke kaotisk og uden fremdrift for udenforstående. Mange beslutninger bliver ændret undervejs, når ny viden og nye ideer kommer på banen. Råd:

- Planlæg og dokumenter fremgangsmåden.
- Opstil kriterier og procedurer for vurdering.
- Vurder sammenhænge og grænseflader tidligt.

Blindhed

Prototyper kan skabe blindhed. En tidlig prototype kan føre til, at både brugernes og systemudviklernes opmærksomhed koncentrerer sig om prototypen. Andre helt

anderledes muligheder overses. Det er farligt at forsøge sig med en prototype, før man kender brugerorganisationen og dens behov. Råd:

- Anvend prototyping sammen med andre fremgangsmåder.

Fokus på værktøj

Prototyping kan sætte fokus på værktøjer til udvikling af prototyper i stedet for på design af edb-systemer, f.eks. når værktøjerne er nye eller mangelfulde. Der bliver så lagt mindre vægt på at planlægge processen og effektivt inddrage brugere i vurdering af prototyperne. Derved kommer udbyttet af prototyping ikke til at svare til tids- og ressourceindsatsen. Råd:

- Planlæg og dokumenter fremgangsmåden.
- Vurder projektets forudsætninger.

Forventninger

Prototyper kan skabe forventninger hos brugerne om, at udviklingstiden kan blive kortere end det er realistisk. Disse forventninger eller økonomiske og tidsmæssige betingelser kan føre til, at en prototype sættes i drift uden at være tilstrækkelig robust med fejlhåndtering, back up faciliteter og andre sikkerhedsfaciliteter.

Prototyping kan også skabe forventninger om, at brugerne kan få rettet edb-systemet ligeså hurtigt og smertefrit efter igangsætning af produktionssystemet som under prototyping processen. Det uformelle samarbejde mellem brugere og systemudviklere omkring rettelser i prototyper er vanskeligt at bibeholde ved vedligeholdelse af produktionssystemet, idet det er vanskeligere at vurdere konsekvenserne af rettelser i produktionssystemet. Råd:

- Gør opmærksom på mangler og begrænsninger i prototypen i forhold til et produktionssystem.

Dokumentation Prototyping kan gennemføres i stedet for andre analyse- og designaktiviteter. Det fører til en mangelfuld dokumentation og til systemer, der er umulige at teste. Omfattende brug af prototyping kan føre til, at overblikket over datamæssige og systemmæssige sammenhænge forsvinder eller ikke bliver etableret. Råd:

- Anvend prototyping sammen med andre fremgangsmåder.
- Lav prototyper ud fra en normaliseret datamodel.
- Beskriv sammenhænge mellem funktioner.
- Fastlæg og beskriv standarder for udformning af system og dokumentation.

Ressourceforbrug

Det er vanskeligt at vurdere tids- og ressourceforbrug ved prototyping. Det er specielt typisk, at prototyping tager mere kalendertid end forventet. Der er en konflikt mellem brug af prototyping og kontrakter med fast pris og tid, idet resultat og omfang af prototyping processen er vanskeligt at forudsige. Det er forøvrigt sværere at bruge prototyping i projekter, hvor en tredje part, f.eks. et konsulentfirma, har interesse i at fastholde aftalerne i kontrakten. Brugere får selv stor forståelse for udbyttet af prototyping og vanskelighederne ved styring af prototyping, mens konsulentfirmaet i mindre grad får denne forståelse. Råd:

- Skriv kontrakt med muligheder for genforhandlinger.
- Vær på vagt over for beskrivelser kvaliteten.

Ambitionsniveau Ved brug af prototyping bliver systemudviklerne så motiverede, at det er svært at styre ambitionerne med hensyn til at tilgodese alle brugerønsker til funktioner og udviklingshastighed. Systemudviklerne bliver næsten for

pligtopfyldende og påtager sig for meget overarbejde.
Råd:

- Styr ambitionsniveauet ud fra mål med prototyping.

Bruger- deltagelse

Prototyping forudsætter en aktiv og repræsentativ brugerdeltagelse. Udvælgelsen af brugerdeltagere er derfor vigtig. Når DC ikke kender kundens organisation, er det svært at sikre en god brugerdeltagelse. En beskrivelse af brugerprofiler er ikke sikkerhed nok. Det er en fordel at kunne pege på bestemte personer som mulige deltagere.
Råd:

- Vær omhyggelig ved etablering af brugerdeltagelse.

Kvalifikationer

Prototyping stiller nye krav til den enkelte systemudvikler, fordi kontakten til kunden ikke er begrænset til få personer, men alle i projektgruppen har kontakt til kunden. Alle skal have interesse i kundekontakt og den nødvendige viden samt de nødvendige kvalifikationer af bl.a. psykologisk og politisk art. Det er derfor vanskeligt at inddrage nye medarbejdere i en prototyping proces.
Råd:

- Vær omhyggelig ved udvælgelse af systemudviklere.

Udviklings- værktøj

Det er en fordel at kende begrænsningerne i det udviklingsværktøj, der skal anvendes ved fremstilling af det endelige system, ellers kan der opstå problemer med at leve op til alle faciliteter i designet i prototyperne. Ved at kende begrænsningerne er det muligt at styre uden om et u hensigtsmæssigt design. Når prototyperne videreudvikles i et 4.generations værktøj kan det betyde lange svartider og lange køretider. En kodning af det endelige system i f.eks. Cobol kunne forbedre dette, men som regel ikke så meget, at det modsvarer indsatsen. Råd:

- Vær opmærksom på muligheder og begrænsninger i udviklingsmiljøet og produktionsmiljøet.

RÅD OG VINK OM FREMGANGSMÅDER.

Planlægning.

Planlæg og dokumenter fremgangsmåden

For at få en effektiv udnyttelse af prototyping er der brug for en veltilrettelagt og dokumenteret fremgangsmåde, hvor følgende spørgsmål er afklaret:

- Hvad ønsker vi at opnå med prototyping ?
- Hvad skal være grundlaget for prototyping ?
- Hvilke værktøjer skal vi bruge, og hvordan ?
- Hvordan skal prototyperne vurderes og justeres ?
- Hvilke kriterier skal være bestemmende for at afslutte prototyping ?
- Hvordan skal prototyperne dokumenteres ?
- Hvordan kommer vi fra prototyperne til det endelige system ?

Ved planlægningen kan spørgsmålene i afsnit 2.3 om 'Hvordan prototyping ?' bruges som en checkliste ved afklaring af ovennævnte spørgsmål.

Projektets forudsætninger.

Vurder projektets forudsætninger

Følgende forudsætninger er nødvendige for at få udbytte af prototyping:

- Brugerorganisationen har accepteret fremgangsmåden og forståelse for kravene til brugerdeltagelse.

- Systemudviklerne er interesseret i at forny arbejdsformerne i forhold til traditionel systemudvikling med fastlagte procedurer for fremgangsmåden.
- Systemudviklingsmiljøet giver mulighed for hurtigt at udvikle og ændre prototyper.
- De tidsmæssige og økonomiske rammer giver mulighed for at få etableret de nødvendige forudsætninger for eksperimenter med prototyper, bl.a. en høj grad af brugerdeltagelse, kendskab til muligheder og begrænsninger i tilgængelige prototypeværktøjer, viden om edb hos brugerdeltagerne og viden om anvendelsesområdet hos systemudviklerne.
- Der foreligger et overordnet design af systemet mindst i form af en simpel datamodel og en liste af ønskede funktioner.

Det følgende indeholder råd og vink om etablering af nogle af disse forudsætninger.

Mulighed for genforhandlinger.

Skriv kontrakt med muligheder for genforhandlinger

Tilbud om brug af prototyping og en høj grad af brugerdeltagelse i systemudviklingen bør modsvares af muligheder for genforhandling af priser og afleveringstider. Ved afgivelse af tilbud kan der f.eks. gives forskellige tilbud afhængig af, om der skal anvendes prototyping eller ej. En mere traditionel systemudvikling uden brug af prototyping kan lettere styres med hensyn til tid og økonomi, idet der kan gennemføres en tidlig fastfrysning af design af edb-systemet, og brugerdeltagelsen i systemudviklingen kan begrænses.

Kvaliteten af beskrivelser.

*Vær på vagt
over for beskri-
velsens kvalitet*

Flot indbundne og omfangsrige beskrivelser i et udbuds- materiale er ingen sikkerhed for, at beskrivelserne er kendte og bredt accepteret i brugerorganisationen. Brugere har brug for at få kød og blod på beskrivelserne for at kunne vurdere konsekvenserne af det nye edb-system. Samtidig er det begrænset, hvilken viden om brugernes ønsker og arbejdsområde, som kan formidles i skriftlige beskrivelser. Planlæg derfor altid med tid til at etablere en fælles forståelse af systembeskrivelserne hos brugere og systemudviklere. Det kan evt. være et af formålene med prototyping.

Brugerdeltagelse.

*Vær omhyggelig
ved etablering af
brugerdeltagelse*

Normalt er det en fordel med en fast brugergruppe, der deltager i afprøvningen af prototyperne i hele projektforløbet. Det er vigtigt, at så mange forskellige brugerkategorier som muligt deltager i brugergruppen. Alle arbejdsområder, der bliver berørt af edb-systemets indførelse, skal være repræsenteret både på udførende og ledende niveau. Teknologiaftalers krav til brugerindsfyldelse skal tilgodeses. Desuden er det en god ide at få både kritiske og opinionsdannende brugere med, både blandt ledere og primærbrugere. Det er afgørende, at brugergruppen kommer til at bestå af aktive og interesserede brugere, og at deltagerne får afsat tid til deltagelse i brugergruppens arbejde. Fra starten skal der laves aftaler om forventninger til brugergruppens arbejde, herunder tidsforbrug og beslutningskompetence. Det er en fordel med stor beslutningskompetence i brugergruppen.

Etablering af projekt.

Vær omhyggelig ved udvælgelse af systemudviklere

Systemudviklerne skal være med fra projektstart og være indstillet på et tæt samarbejde med brugere uden præcise specifikationer af det nye system. Systemudviklere skal kunne diskutere og gennemtænke brugernes ideer og ønsker til edb-systemet. Det er en erfaring, at det er svært for brugere at forestille sig mulighederne for brug af edb-teknologien i deres arbejde. Systemudviklerne skal være indstillet på at være udfarende med hensyn til at komme med forslag til udformning af edb-systemerne ud fra egne erfaringer.

Det er en god ide at gennemføre en grundig start af projektet, hvor brugere og systemudviklere får en fælles forståelse af betingelserne for projektet og i fællesskab planlægger arbejdsformen i projektet. Som inspiration kan projektet få personer fra andre projekter, der har anvendt prototyping, til at fortælle om erfaringer og give gode ideer. Der kan f.eks. gennemføres et seminar eller kursus om prototyping som led i projektstarten.

Ambitionsniveau ud fra mål.

Styr ambitionsniveauet ud fra mål med prototyping

For at kunne begrænse omfanget af prototyping processen er det nødvendigt at have præciseret forventningerne til udbyttet af processen. Forventningerne skal beskrives ved nogle kriterier og procedurer for, hvornår prototyping processen skal afsluttes. Det er vigtigt at være bevidst over for, at ambitionsniveauet for prototypens omfang ikke øges gradvist under processen.

Prototypen i forhold til et produktionssystem.

Gør opmærksom på mangler og begrænsninger i prototypen i forhold til et produktionssystem

Eksempler på mangler og begrænsninger i en prototype:

- Der opsamles ikke data om edb-kørslernes forløb.
- Der er ingen konverteringsprocedurer.
- Der er ingen brugerdokumentation.
- Der er ingen back up og genstarts procedurer.
- Der er ingen fuldt udbygget database og database-kommunikation.
- Der er ingen procedurer ved fejl i maskinel eller programmel.
- Der er ingen testprocedurer.
- Al datakommunikation er ikke etableret og testet.
- Der er ingen planer for brugeruddannelse.
- Der anvendes ikke fortrykte blanketter.

Design af funktioner.

Beskriv sammenhænge mellem funktioner

Prototyper er gode til at afklare krav og ønsker til detaljeret design af enkeltfunktioner, men det er nødvendigt at supplere prototyperne med beskrivelser af sammenhænge mellem funktioner. Beskrivelsesværktøjerne fra Struktureret Analyse, Struktureret Design og Struktureret Programmering kan anvendes her, bl.a. datastrømsdiagrammer, struktureret dansk, dataordbogsnotation, moduldiagrammer og JSP-diagrammer.

Systemudviklingsværktøjet CASE 2000 kan anvendes for at lette vedligeholdelsen af beskrivelserne og for at få støtte til kontrol af sammenhænge i beskrivelserne.

Design af datamodel.

Lav prototyper ud fra en normaliseret data-model

Et centralt element i de systemer, der med fordel kan udvikles ved brug af prototyper, er databasen. Det er vigtigt at gøre meget ud af at få en logisk og overskuelig struktur på de data, der skal gemmes. Det er vigtigt, at databasen ikke indeholder samme data flere steder. Prototyper bør derfor bygge på normaliserede datamodeller.

Det er vigtigere, at strukturen i databasen bliver fastlagt tidligt i processen, end at funktionerne bliver fastlagt, idet det er lettere at ændre og tilføje databehandlingsfunktioner end at ændre datastrukturen. Det er dog nødvendigt at have et vist kendskab til ønskede funktioner i systemet for at kunne fastlægge datastrukturen. Det er nødvendigt med et samspil mellem funktionsanalyse og dataanalyse i hele projektføreløbet.

Flere fremgangsmåder.

Anvend prototyping sammen med andre fremgangsmåder

Prototyping skal opfattes som et supplement til fremgangsmåder ved funktionsanalyse og dataanalyse, jvf. rådene om at beskrive sammenhænge mellem funktioner og at lave prototyper ud fra en normaliseret datamodel.

Prototyping skal bruges til at støtte brugernes evne til at få ideer til brug af edb og vurdere konsekvenser af edb-anvendelse. Lav derfor flere prototyper ud fra forskellige ideer til anvendelse af edb.

Sammenhænge og grænseflader.

Vurder sammenhænge og grænseflader tidligt

Det er ressourcekrævende at ændre i næsten færdigprogrammerede delsystemer. Det er derfor vigtigt at sikre en fælles opfattelse af datasammenhænge mellem delsystemer tidligt i projektføreløbet. Udviklingen af prototyper bør planlægges, således at grænseflader mellem delsystemer bliver afklaret og fastlagt tidligt i forløbet. Data-kommunikation mellem delsystemer bør indgå i de første prototyper.

Vurdering af prototyperne.

Opstil kriterier og procedurer for vurdering

Som et led i styringen af prototyping processen bør der opstilles kriterier og procedurer for vurdering af prototyperne. Kriterierne bør være konkrete og afgrænset til bestemte dele af systemet og bestemte aspekter af systemet. DC's Kvalitetshåndbog indeholder beskrivelser af forskellige egenskaber ved edb-systemer og af, hvordan disse egenskaber kan vurderes. DC's vejledning for teknologivurdering indeholder en checkliste med spørgsmål, der bør indgå i en vurdering af konsekvenserne af at anvende et bestemt edb-system.

I nogle situationer kan det være nødvendigt at begrænse prototyping processen i kalendertid eller til gennemførelse af bestemte vurderingsprocedurer, f.eks. gennemførelse af en 'tænke højt test' af nogle skærmdialoger med deltagelse af et bestemt antal forsøgspersoner.

Standarder for system og dokumentation.

Fastlæg og beskriv standarder for udformning af system og dokumentation

Standarder for udformning af system og dokumentation bør ideelt set fastlægges tidligt i projektførløbet. Ved udformning af brugergrænseflader kan det være nødvendigt i starten af prototyping processen at udforme systemet og standarderne parallelt. Det kan anbefales at starte med at arbejde med en mindre, repræsentativ del af systemet for at få nogle erfaringer med standarder og fremgangsmåder før, der parallelt arbejdes med hele systemet. På skift kan forskellige undergrupper i projektgruppen arbejde som udforskningsgruppe før igangsætning af en aktivitet bredt i projektgruppen. Specielle DC-standarder er beskrevet i DC-systemdirektiver. Det er vigtigt at skelne mellem standarder for udviklingsdokumentation og vedligeholdelsesdokumentation. Standarder bør omfatte realistiske minimumsstandarder og ikke kun standarder for den ideelle dokumentation.

Udviklingsmiljø og produktionsmiljø.

Vær opmærksom på muligheder og begrænsninger i udviklingsmiljøet og produktionsmiljøet

Ikke overraskende er det en erfaring, at det er en stor fordel, at systemudviklerne kender både udviklingsmiljøet og produktionsmiljøet. Men da der sker store ændringer i disse miljøer hele tiden, er det normalt, at en del af de tekniske miljøer er nye for systemudviklerne. Det er vigtigt at være opmærksom på dette og få afsat tid til og få gennemført den nødvendige uddannelse af både brugere og projektgruppe.

REFERENCER.

- DC-projekter* Erfaringer fra prototyping i nedennævnte DC-projekter er beskrevet i notater fra interview af DC-deltagere. Notaterne kan fås ved henvendelse til Metodesektionen i Teknikstaben.
- * **BT-projektet.**
Udvikling af et reservationssystem til Bornholmstrafikken ved brug af værktøjet Powerhouse. (Center 6)
 - * **TAPAS-projektet.**
Udvikling af et patientadministrationssystem til Tandlægehøjskolen i København ved brug af værktøjet ADABAS/Natural. (Center 4)
 - * **ELEVREG-projektet.**
Udvikling af et studieadministrationssystem til Handelsskolerne i Danmark ved brug af værktøjet ADABAS/Natural. (Center 4)
 - * **PAI-projektet.**
Udvikling af et standardssystem til personaleadministration ved brug af værktøjet Focus. (Center 3)
- DC-vejledninger* Nedennævnte vejledninger supplerer denne vejledning omkring kvalitetsvurdering og konsekvensvurdering af prototyper. Vejledningerne kan fås ved henvendelse til Metodesektionen i Teknikstaben.
- * **'Kvalitetshåndbogen'**, der indeholder beskrivelser af egenskaber ved edb-systemer og af, hvordan disse egenskaber kan vurderes.
 - * **'Principper for teknologivurdering'**, der indeholder en checkliste med spørgsmål, der bør indgå i en vurdering af konsekvenserne af anvendelse af et bestemt edb-system.

Litteratur

Ud af den omfattende litteratur om prototyping kan følgende anbefales som inspiration til systematisk anvendelse af prototyping. Litteraturen kan lånes på DC-biblioteket.

- * N.E.Andersen m.fl.: '**Professionel Systemudvikling - Erfaringer, muligheder og handling**', der bl.a. indeholder ideer til fremgangsmåder ved ledelse af systemudviklingsprojekter, som kan bruges ved styring af en prototyping proces.
Teknisk Forlag, 1986.
- * Bernhard H. Boar: '**Application Prototyping: A Requirements Definition Strategy for the 80s**', som i detaljer beskriver, hvordan prototyping kan anvendes til afklaring af brugernes behov for funktioner og data samt krav til udformning af brugergrænsefladen.
Wiley Interscience, 1984.
- * R.P.Cervany et al: '**The Application of Prototyping to Systems Development: A Rationale and Model**', s.52-62 i *Journal of Management Information Systems/Fall 1986, Vol III, no.2*. I artiklen præsenteres en model for, hvor og hvordan prototyping kan indgå i et systemudviklingsforløb. Modellen omfatter en beskrivelse af sammenhænge mellem typer af systemer, typer af prototyper, prototypeværktøjer, prototypeudformning og status for prototypen (hvordan prototypen indgår i udviklingsforløbet).
- * Søren Christensen m.fl.: '**Arbejdsformer under anvendelse af 4.generationsværktøjer**', der beskriver og vurderer erfaringer fra anvendelse af 4.generationsværktøjer i 9 systemudviklingsprojekter i 5 forskellige udviklingsorganisationer. Konklusionen er, at 'hvis systemudvikling baseret på 4.generationsværktøjer skal resultere i bedre edb-baserede systemer og en nedsættelse af den samlede udviklingstid, så skal indførelse af værktøjerne i udviklingsorganisationerne indgå som et led i en bevidst forandring af arbejdsformerne i systemudvikling.' Rapporten peger især på bevidst brug af prototyper i systemudviklingen.
Datalogisk Afdeling, Aarhus Universitet, DAIMI IR-69, Maj 1987.

- * Christiane Floyd: '**A Systematic Look at Prototyping**', s. 1-18 i R.Budde et al (ed): '**Approaches to Prototyping**', som indeholder en række arbejdsdokumenter fra en konference om prototyping afholdt i Namur i oktober 1983.
Springer Verlag, 1984.
- * Klaus Viby Mogensen: '**Afprøvning af systemudvikling med prototyper**', der bl.a. indeholder en kommenteret liste over litteratur om prototyping.
Datalogisk Institut, Københavns Universitet, Rapport nr. 85/10.
- * Rolf Molich (red): '**Brugervenlige Edb-systemer**', der bl.a. indeholder ideer til fremgangsmåder ved afprøvning af brugergrænseflader.
Teknisk Forlag, 1986.
- * Xephon Consultancy Report: '**Prototyping for IBM Users**', der indeholder en række indlæg om erfaringer med prototyping, specielt ved brug af udviklingsværktøjerne ADS/Online, FOCUS, MANTIS, RAMIS II og GENER/OL. Indlæggene er skrevet af forskellige konsulenter.
Xephon Technology Transfer Ltd, 1985.

