

Århus Købmandsskole  
Biblioteker  
Sønderhøj 30 · 8260 Viby J  
Tlf. 8936 3364  
E-mail: bib-sh@aabc.dk

Kim Thorshøj

# Eksperimenter i systemudvikling

Roskilde Universitetscenter

Erhvervsakademi Aarhus  
Biblioteket  
Sønderhøj 30 · 8260 Viby J  
Tlf. 7228 6040  
Eaa-bibliotek@eaaa.dk

LAYOUT: SØREN MAARBJERG. TRYK: ROSKILDE UNIVERSITETSCENTERS TRYKKERI, 1989.

---

# INDHOLD

---

## **Indledning**

- 3    Mirakelkur eller hårdt arbejde?    OM FORVENTNINGER OG PROBLEMER

## **Præsentation**

- 5    Kogebog eller idébank?    OM HENSIGTEN MED DENNE BOG

## **Eksperimenter - hvorfor**

- 7    Samarbejde og læreproces    OM SYN PÅ SYSTEMUDVIKLING

## **Eksperimenter - hvilke**

- 10    Udforske, afprøve, udvikle    OM FORMÅL MED EKSPERIMENTER  
12    Demonstration eller drift    OM AFPRØVNING AF PROTOTYPER  
14    Smid væk eller genbrug    OM ANVENDELSE AF PROTOTYPEN  
15    Modeller    OM PROJEKTFORLØB MED EKSPERIMENTER

## **Eksperimenter - hvordan**

- 18    Tilrettelæggelse    OM PUNKTER FOR FORBEREDELSE  
19    Anvendelse    OM EKSPERIMENTER ELLER EJ  
21    Ledelse    OM PLANLÆGNING OG ORGANISERING  
23    Forudsætninger    OM KRAV TIL DELTAGERNE  
25    Målsætning    OM PRÆCISERING AF FORMÅLET  
27    Konstruktion    OM PROTOTYPERNS INDHOLD OG OMFANG  
29    Afprøvning    OM STYRING OG UDFOLDELSE  
31    Dokumentation    OM OPSAMLING AF NY VIDEN  
32    Vurdering    OM NIVEAUER AF PLANLÆGNING

## **Afslutning**

- 35    Alternativ eller supplement?  
      OM TRADITIONEL OG EKSPERIMENTEL SYSTEMUDVIKLING  
37    Referencer  
38    Litteratur

---

# INDLEDNING

Århus Købmandsskole  
Biblioteket  
Sønderhøj 30 - 8260 Viby J  
Tlf. 8936 3364  
E-mail: bib-sh@aabc.dk

---

## Mirakelkur eller hårdt arbejde?

### OM FORVENTNINGER OG PROBLEMER

"A prototypical success story", "Shortcut to systems design", "Prototyping: Orchestrating for success", "Breaking the systems development bottleneck".

Overskrifter som disse har gennem den sidste halve snes år optrådt i tidsskrifter om systemudvikling. Eksperimentel systemudvikling (i engelsksproget litteratur kaldet 'prototyping') har vundet udbredelse i takt med fremkomsten af stadig bedre prototypeudviklingsværktøjer. Eksperimentel systemudvikling har fundet anvendelse som supplement eller alternativ til traditionel systemudvikling.

Forventningerne til eksperimentel systemudvikling har f.eks. været at opnå bedre kravspecifikation, øget produktivitet i programmeringsarbejdet, større brugertilfredshed og -engagement, og bedre kommunikation og samarbejde mellem brugere og udviklere. Men anvendelse af nye værktøjer til udvikling af prototyper har ikke altid indfriet forventningerne og har i stedet givet nye problemer. Nogle gennemgående problemer har været:

**BLINDHED** - Udvikling af en prototype uden forudgående analyse kan føre til, at man 'stirrer sig blind' på en uhensigtsmæssig løsning.

**DÅRLIGT DESIGN** - Eksperimenter med prototyper har tendens til at fokusere på detaljer på bekostning af det overordnede design.

**MANGLENDE STYRING** - Eksperimentel systemudvikling er svær at styre, fordi kravene til edb-systemet udvikles undervejs.

**DÅRLIGE SYSTEMER** - Prototyper bliver ofte taget i anvendelse, selvom omprogrammering af prototypen er planlagt.

**MANGLENDE FORSTÅELSE** - Afprøvning af prototyper giver ikke altid det ønskede resultat, fordi brugere og udviklere ikke forstår formålet med at udvikle og afprøve prototyper eller ikke kender deres egen rolle.

Bøger og artikler om eksperimentel systemudvikling rummer derfor i stigende grad forslag til, hvad man kan gøre ved disse problemer. Forslagene kan sammenfattes i følgende overordnede principper for anvendelse af eksperimenter i systemudvikling:

- Et eksperiment er en gensidig læreproces, resultatet er ny viden
- Eksperimenter skal tilrettelægges og afvikles systematisk
- Deltagerne skal forstå og acceptere anvendelse af eksperimenter
- Eksperimenter kræver regelmæssig projektvurdering og -planlægning
- Eksperimenter overflødiggør ikke traditionelle fremgangsmåder
- Formålet med eksperimentet skal afgrænses og præciseres
- Prototypen skal konstrueres ud fra eksperimentets formål
- Afprøvning af prototypen skal forholde sig til anvendelsessituationen
- Et eksperiment skal vurderes i forhold til alle niveauer af planlægning

Denne bog handler om disse principper. Hvad er baggrunden for principperne, og hvordan kan de føres ud i livet?

---

# PRÆSENTATION

---

## Kogebog eller idébank?

### OM HENSIGTEN MED DENNE BOG

Denne bog er et forsøg på at samle og videregive nogle af de erfaringer og forslag, som findes i nyere litteratur om eksperimentel systemudvikling. Bogen henvender sig til professionelle systemudviklere, som har læst eller hørt om eksperimentel systemudvikling, og som tænker: "Ideen er jo god nok, men hvad med problemerne?". Men også systemudviklere med praktisk erfaring med eksperimenter kan måske falde over nye ideer eller indfaldsvinkler.

Min baggrund for at skrive denne bog er førstegangserfaringer med eksperimentel systemudvikling i forbindelse med et projekt på datalogiuddannelsen ved Roskilde Universitetscenter. Projektet gik ud på at afprøve en metode for eksperimentel systemudvikling i praksis. Erfaringerne fra dette projekt førte til nogle overvejelser over, hvad man som systemudvikler har brug for af baggrundsviden og retningslinjer, når man kaster sig ud i eksperimentel systemudvikling. Mit næste projekt blev derfor at skrive denne bog, på baggrund af et studie af teori, metoder, undersøgelser og erfaringer.

Bogen handler ikke om at anvende værktøjer til udvikling af prototyper. Den handler om at anvende eksperimenter som en aktivitet i systemudvikling. Derfor anvender jeg betegnelsen 'eksperimenter i systemudvikling' i stedet for 'prototypeudvikling'.

Bogen er ikke en metode for eksperimentel systemudvikling. Den er ikke en 'kogebog' med præcise anvisninger på, hvordan et systemudviklingsprojekt skal forløbe under anvendelse af eksperimenter. Bogen er i stedet tænkt som en 'idebank', der rummer retningslinjer for anvendelse af eksperimenter i systemudvikling. Disse retningslinjer skal så vurderes i forhold til et konkret projekt.

Bogen er i tre dele:

**HVORFOR** - Første del præsenterer ideerne bag eksperimenter i systemudvikling. Her fremhæves, at systemudvikling også er en læreproces, og at læreprocesser understøttes af eksperimenter.

**HVILKE** - Anden del viser omfanget af anvendelsesmuligheder. Her præsenteres forskellige former for eksperimenter og forskellige måder, hvorpå eksperimenter og prototyper kan indgå i et systemudviklingsforløb.

**HVORDAN** - Tredje del går i detaljer med det enkelte eksperiment. Her diskuteres - på baggrund af eksempler - hvordan et eksperiment kan tilrettelægges og afvikles, så udbyttet øges og problemerne mindskes.

Bogen afsluttes med litteraturhenvisninger fra de enkelte afsnit. Forslag til uddybende læsning er kommenteret i litteraturlisten sidst i bogen.

Tak til alle, der har bidraget med kritik og forslag.

*Kim Thorshøj, Roskilde Universitetscenter, september 1989*

---

# HVORFOR

---

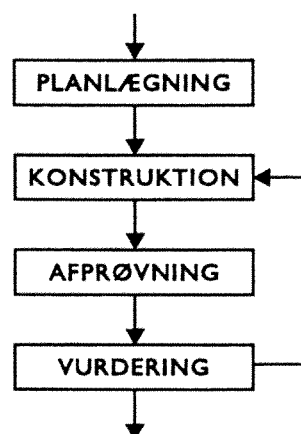
## Samarbejde og læreproces

### OM SYN PÅ SYSTEMUDVIKLING

Hvorfor laver man eksperimenter i systemudvikling? Hvad er ideen i et eksperiment? Eksperimenter i systemudvikling handler om at gøre en ufærdig model af et edb-system - en prototype - til genstand for afprøvning og vurdering. Forløbet i et eksperiment kan illustreres af fire trin:

- Eksperimentet planlægges, en grov kravspecifikation opstilles.
- Prototypen konstrueres ved hjælp af diverse udviklingsværktøjer.
- Prototypen afprøves af udvalgte brugere og udviklere.
- Kritik og forslag vurderes.

Udfra vurderingen afgøres, om eksperimentet har opfyldt sit formål og skal afsluttes, eller hvordan det skal fortsættes.



### Eksperimenter er fremgangsmåder i systemudvikling

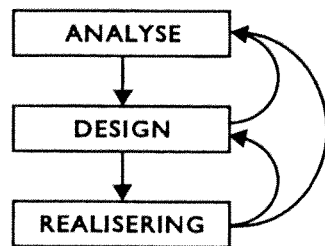
Baggrunden for anvendelse af eksperimenter i systemudvikling er problemerne med på den ene side at afklare brugernes behov for edb, på den anden side at beskrive et edb-system overfor brugerne. Brugere har svært ved at forestille sig muligheder og begrænsninger i edb. De har også svært ved at forholde sig til formaliserede beskrivelser af et edb-system. Brugerens forestillinger om anvendelse af edb udspringer af erfaringer. Eksperimenter med prototyper kan give brugerne konkrete oplevelser med et edb-system og dermed bedre grundlag for at komme med kritik og forslag. Samtidig kan brugerne opleve, at der sker noget, at deres kritik og forslag

får hurtig indflydelse på udformning af edb-systemet, og at de deltager direkte i design af deres eget edb-system. Det øger brugernes accept af såvel systemudviklingsprocessen som af det færdige system.

Eksperimenter kan således ses som fremgangsmåder, der gør kravspecifikationen bedre og brugerne mere tilfredse. Men eksperimenter kan også sættes i forhold til en mere teoretisk opfattelse af, hvad systemudvikling er, og hvordan en systemudviklingsproces forløber.

### Eksperimenter er iteration

Systemudvikling kan opdeles i tre hovedaktiviteter: analyse, design og realisering, som vist i figuren. Traditionel systemudvikling anbringer disse tre aktiviteter i rækkefølge. Eksperimentel systemudvikling bryder i princippet med denne rækkefølge og bygger i stedet på gentagelser (iterationer) over de tre aktiviteter.



I praksis forekommer iterationer også i traditionel systemudvikling, men ofte ufrivilligt, i forbindelse med afsluttende test eller indkøring af systemer, og iterationer ses som udtryk for fejl i forløbet.

I eksperimentel systemudvikling planlægges der bevidst med iterationer, de foregår hurtigere, og de lægges tidligere i forløbet.

### Eksperimenter er læreprocesser

Systemudvikling kan ses som en produktionsproces, hvor et mere eller mindre komplekst produkt udvikles ud fra en specifikation. Problemet er kompleksiteten, og løsningen er formelle beskrivelser af produktet. Fokus er på produktet og dets tekniske sundhed. Systemudvikling kan også ses som en læreproces, hvor en specifikation udvikles ud fra mere eller mindre usikre forestillinger om anvendelse af edb. Problemet er usikkerheden, og løsningen er praktiske eksperimenter. Fokus er på processen og på samarbejdet mellem brugere og udviklere.

Disse to syn på systemudvikling udelukker ikke hinanden, de supplerer hinanden.

Eksperimentel systemudvikling bygger på den grundlæggende antagelse, at systemudvikling også er en læreproces, og at læreprocesser forløber i vekselvirkning mellem refleksion og handling, dvs. mellem analyse, design og realisering.

### Eksperimenter er samarbejde

Drivkraften i denne læreproces er samarbejdet, den gensidige udveksling mellem systemudviklere og systembrugere. Udviklingen er en balance mellem accept og provokation. Udviklerne tager udgangspunkt i brugernes forestillinger om anvendelse af edb, men provokerer brugerne ved at



komme med en prototype, ved at komme med alternative forslag og ideer og ved at sætte spørgsmålstejn ved brugernes ønsker. Brugerne tvinges til at se deres egen arbejdssituation i nyt lys. Brugerne tager udgangspunkt i udviklernes opfattelse af arbejdet, i form af en prototype, men provokerer udviklerne ved at kritisere prototypen og ved at komme med nye forslag og ønsker. Udviklerne tvinges til at revidere deres opfattelse af brugernes arbejdssituation og til at overveje nye muligheder for edb-anvendelse.

### Opsummering

Man laver eksperimenter i systemudvikling, fordi det styrker samarbejdet med brugerne. Man laver eksperimenter, fordi systemudvikling også er en læreproces, som forudsætter iteration mellem analyse, design og realisering. Eksperimenter i systemudvikling har to sider. Den ene side handler om at fremstille et produkt, en prototype, som er en mere eller mindre udbygget model af et fremtidigt edb-system. Den anden side handler om at tilrettelægge en proces, der kan øge deltagernes viden om, hvordan et edb-system kan indgå i en fremtidig arbejdssituation.

Denne bog handler mest om processiden af eksperimenter i systemudvikling. Anden del - Hvilke - gennemgår forskellige former for eksperimenter og ser på, hvordan de påvirker samarbejdet og læreprocessen. Tredje del - Hvordan - diskuterer tilrettelæggelse af et eksperiment og fokuserer på, hvordan problemer kan mindskes ved at styrke samarbejdet og læreprocessen.

---

# HVILKE

---

## Udforske, afprøve, udvikle

### OM FORMÅL MED EKSPERIMENTER

Eksperimenter kan være vidt forskellige i formål og omfang. Man kan dele eksperimenter op efter tre overordnede formål - udforskende, afprøvende og udviklende eksperimenter. Disse tre formål udelukker ikke hinanden, et eksperiment kan omfatte forskellige grader af alle tre formål.

#### Udforskende eksperimenter

Udforskende eksperimenter ('exploratory prototyping') retter sig mod at generere ideer og muligheder for edb-anvendelse. Gennem sammenligning og diskussion af forskellige ideer præciseres brugernes behov for edb. Et udforskende eksperiment kan f.eks. gå ud på at:

- præsentere brugere for et katalog af prototyper fra tidligere projekter. Brugere kan så udvælge grænseflade, skærmlayout, funktioner m.m. til nærmere afprøvning.
- demonstrere eller afprøve relevant standardprogrammel eller andre eksisterende edb-systemer, f.eks. ved virksomhedsbesøg.
- forsyne en prototype med flere løsninger på udvalgte designspørgsmål. Konsekvenser af forskellige løsninger kan så direkte sammenlignes under afprøvning af prototypen.
- lave flere prototyper, ud fra forskellige ideer til, hvordan edb kan indgå i brugernes arbejde.

Udforskende eksperimenter kan nedsætte risikoen for blindhed, hvis væsentligt forskellige løsninger bliver præsenteret. De kræver ikke nødvendigvis udvikling af en prototype. Afprøvning af eksisterende programmel kan være en hurtig og billig form for idegenerering.

## Afprøvende eksperimenter

Afprøvende eksperimenter ('experimental prototyping') retter sig mod design af udvalgte dele af et edb-system. På baggrund af en bestemt ide til edb-anvendelse konstrueres en prototype af edb-systemet eller en del af det. Ved iteration mellem prototypeafprøvning og videreudvikling vurderes og præciseres ideen.

Områder for afprøvende eksperimenter kan f.eks. være:

- afgrænsning af system
- detaljering af funktionalitet
- den enkelte brugergrænseflade
- sammenhæng mellem skærbilleder
- databasedesign
- fejlbehandling, sikkerhed
- effektivitet, svartider
- brugeres arbejdsituation
- brugeres accept af prototypen
- indvirkning på brugerorganisation

Afprøvende eksperimenter giver mulighed for at gå i detaljer med et designforslag. Et problem er, at et sådant designforslag ofte præsenteres tidligt i et projektforslag, uden forudgående analyse af anvendelsesområdet, og uden udforskning af alternative forslag. Det øger risikoen for, at man lægger sig fast på et uhensigtsmæssigt design eller fortaber sig i detaljer på bekostning af overblikket.

## Udviklende eksperimenter

Udviklende eksperimenter ('evolutionary prototyping') retter sig mod trinvis realisering og ibrugtagning af et edb-system. Formålet kan være at tilpasse systemudviklingen til et omskifteligt anvendelsesmiljø eller at opdele et stort udviklingsprojekt i flere mindre projekter.

Tilpasning til et omskifteligt brugermiljø kan foregå ved at tilrettelægge forløbet som en - principielt uendelig - videreudvikling af edb-systemet, i form af gentagen revurdering, redesign og rekonstruktion. Denne form for iteration, ofte kaldet versionsudvikling, afviger ikke væsentligt fra en traditionel vedligeholdelsessituation, men kan ses som udtryk for en mere dynamisk opfattelse af systemudvikling, hvor ændringer ikke er nødvendigheden vedligeholdelse, men planlagt videreudvikling.

Opdeling af et stort edb-projekt kan foregå på to leder. F.eks. kan første version udgøre kernen af det projekterede system, som derefter tages i anvendelse, mens mere perifere dele designes og implementeres. Eller

systemet kan opdeles i en række sideordnede delsystemer, som udvikles og ibrugtages hver for sig. Typisk udvikles en fuldt funktionel prototype for den mest kritiske, usikre eller typiske del af edb-systemet, et såkaldt pilotsystem.

Trinvis udvikling er eksperimenter i det omfang, erfaringer fra afprøvning og anvendelse af versioner eller pilotsystemer indgår i analyse og design af senere versioner eller delsystemer. Udviklende eksperimenter reducerer tiden fra projektstart til der begynder at komme resultater, hvilket specielt i store projekter kan øge motivationen. Udviklende eksperimenter er blevet mere aktuelle med fremkomsten af prototypeværktøjer, men er ikke nødvendigvis knyttet til anvendelse af sådanne værktøjer.

---

## Demonstration eller drift

### OM AFPRØVNING AF PROTOTYPER

Kernen i et eksperiment er samarbejdet mellem udviklere og brugere omkring afprøvning af en prototype. Afprøvningen og samarbejdet kan organiseres på forskellige måder.

#### Demonstration

Udviklerne demonstrerer prototypen for brugerne. Afprøvnings starter typisk med en demonstration. Her gennemgås f.eks. prototypens opbygning, funktioner, begrænsninger samt ændringer i forhold til tidligere prototyper. Ved mere 'primitive' prototyper – f.eks. i udforskende eksperimenter – kan hele afprøvningen foregå som en blanding af demonstration og diskussion.

Demonstration giver brugerne mulighed for at blive lidt trygge ved situationen, før de selv skal have fingrene i maskinen, og giver udviklerne mulighed for at forberede brugerne på indholdet af en senere afprøvning. Demonstration uden efterfølgende afprøvning giver ikke brugerne direkte oplevelser og giver derfor mindre viden om brugernes opfattelse af prototypen.

#### Fælles afprøvning

Brugerne afprøver og kommenterer prototypen, f.eks. ved at 'tænke højt'. Udviklerne sørger for, at alle sider af prototypen bliver afprøvet, og noterer brugernes kommentarer. Udviklerne kan holde sig i baggrunden som observatører eller deltage mere aktivt ved at stille spørgsmål og diskutere prototypen med brugerne.

Fælles afprøvning gør det muligt at styre eksperimentet stramt efter et tilrettelagt forløb. En mindre tilrettelagt form kan styrke diskussionen mellem brugere og udviklere. Fælles afprøvning er af økonomiske grunde typisk kortvarig og afslører derfor ikke fejl og mangler, som knytter sig til langvarig og mere erfaren brug af prototypen.

### Fælles konstruktion

Udviklerne opbygger prototypen sammen med brugerne. I det omfang, konstruktion eller tilretning af prototypen kan udføres 'på stedet', kan brugere deltage. Konstruktion, afprøvning og vurdering af prototypen samles så i én aktivitet.

Fælles konstruktion giver mulighed for hurtig iteration og meget tæt samarbejde, typisk mellem én udvikler og én bruger. Fælles konstruktion kan medføre, at brugeren fokuserer på udviklingsværktøjets muligheder og begrænsninger frem for på reelle behov for anvendelse af edb.

### Brugerafprøvning

Prototypen installeres i brugermiljøet. Brugere afprøver selv prototypen og melder tilbage til udviklerne. Brugerafprøvning kan anvendes som supplement til demonstration eller fælles afprøvning. Som støtte for brugerafprøvning kan prototypen forsynes med hjælpefacilitet og med 'notesblok' eller 'dagbog' til brugernes kommentarer.

Brugerafprøvning giver brugere mulighed for at afprøve prototypen i vante omgivelser, i deres eget tempo, uden at være overvåget eller styret. Det giver brugere større frihed til at eksperimentere og 'komme ud i krogene' og derved opdage muligheder og problemer, som en mere styret afprøvning ikke vil opfatte. Brugerafprøvning giver ringe udbytte, hvis ikke brugere har fået undervisning i prototypen og kender målene med afprøvningen.

### Driftsafprøvning

Brugere anvender prototypen i det daglige arbejde over en periode. Anvendelse af en version eller et pilotsystem, under trinvis udvikling, er eksempel på driftsafprøvning. Også afprøvende eksperimenter kan afvikles i en driftslignende situation.

Driftsafprøvning gør det muligt at afprøve prototypens indvirkning på arbejdssituationen og på brugerorganisationen. Det kan også give indtryk af, hvordan en øvet bruger arbejder med systemet, hvor andre afprøvningsformer mere svarer til undervisning af en uøvet bruger. Driftsafprøvning er dyr, da det stiller store krav til prototypens robusthed og funktionalitet indenfor det afgrænsede arbejdsområde samt sikkerhed for, at brugere ikke risikerer at miste vigtige data.

## Paralleldrif

En variant af driftsafrøvning er paralleldrif, hvor prototypen anvendes sideløbende med det eksisterende (manuelle eller edb-baserede) system.

Paralleldrif koster brugerne ekstra arbejde, men mindsker risikoen ved eksperimentet, idet brugerne hele tiden har det gamle system som 'sikkerhedsnet'. Samtidig giver det et godt sammenligningsgrundlag for prototypen.

---

# Smid væk eller genbrug

## OM ANVENDELSE AF PROTOTYPEN

En prototype, udviklet i et prototypeværktøj, kan anvendes på forskellige måder, efter eksperimentet er afsluttet. Man skelner specielt mellem to former for anvendelse, som kan kaldes omprogrammering og færdigudvikling.

### Omprogrammering

Ved omprogrammering ('throw-away/requirements prototyping') anvendes prototypen til at kvalificere og præcisere en kravspecifikation, hvorefter prototypen 'kasseres'.

Omprogrammering styrker de udforskende og afprøvende sider af et eksperiment. Da prototypen ikke skal indgå i det færdige system, er de tekniske krav mindre, og man kan koncentrere sig om at afprøve flere muligheder og nå flere iterationer. Til gengæld bliver konstruktion af prototypen på kort sigt en ekstra omkostning, da det færdige system skal konstrueres på traditionel vis.

### Færdigudvikling

Ved færdigudvikling ('keep-it/implementation prototyping') fortsættes udvikling af prototypen, til den opfylder alle krav til edb-systemet, eller kan indgå som en del af dette.

Færdigudvikling reducerer programmeringsarbejdet og øger dermed produktiviteten i systemudvikling. Et problem er, at færdigudvikling ofte anvendes ukritisk og ustruktureret, uden tilstrækkelig 'produktmodning' af prototypen. Når man nu har lavet en prototype, og tiden er knap, er det fristende for projektledelsen at give efter for brugerpres og tage prototypen i anvendelse. Det giver dårligere systemer. Dårlige erfaringer med ukritisk færdigudvikling er en af hovedårsagerne til kritik af eksperimentel systemudvikling.

Muligheden for færdigudvikling afhænger af prototypeværktøjet. Hvor effektivt og fleksibelt er det? Kan det integreres i det eksisterende edb-

miljø? Er der mulighed for optimering gennem omprogrammering af centrale funktioner til et lavere sprog?

### Andre anvendelser

Hvad enten en prototype omprogrammeres eller færdigudvikles, kan den have andre anvendelsesmuligheder under eller efter udviklingen af det endelige edb-system:

- Undervisning af kommende brugere
- Demonstration for nye, potentielle brugere eller kunder
- Demonstration for deltagere i andre udviklingsprojekter, i udforskende eksperiment
- Afprøvning af forslag til vedligeholdelse eller udvidelse af det færdige system

## Modeller

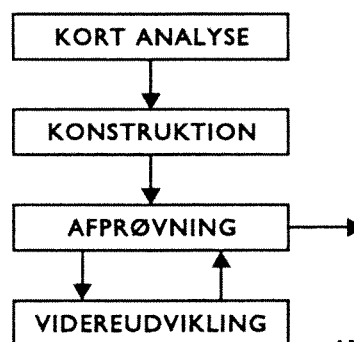
### OM PROJEKTFORLØB MED EKSPERIMENTER

Hovedaktiviteterne i et systemudviklingsprojekt er analyse, design og realisering, i nævnte rækkefølge. Disse tre aktiviteter udgør hovedbevægelsesretningen i traditionelle systemudviklingsmetoder. Selv om eksperimenter i systemudvikling handler om iteration mellem analyse, design og realisering, kan eksperimenter indgå i et traditionelt systemudviklingsforløb. Et eksperiment kan udgøre et afgrænset forløb indenfor en af hovedaktiviteterne.

Alle tre hovedaktiviteter kan understøttes af eksperimenter. I grove træk kan de tre hovedformål for eksperimenter - udforskende, afprøvende og udviklende - knyttes til hver sin hovedaktivitet og dermed placeres tidsmæssigt i et forløb: udforskende eksperimenter under analyse, afprøvende eksperimenter under design og udviklende eksperimenter under realisering. I praksis kan de tre hovedformål kombineres i et og samme eksperiment og optræde i vilkårlig rækkefølge. F.eks. kan man under et overvejende afprøvende eksperiment få behov for at udforske alternative løsninger på et konkret problem.

### En forenklet model

Figuren viser en forenklet model for et forløb, bestående af fire trin. Efter en kort undersøgelse af de grundlæggende krav konstrueres prototypen, hvorpå der itereres mellem afprøvning og videreud-

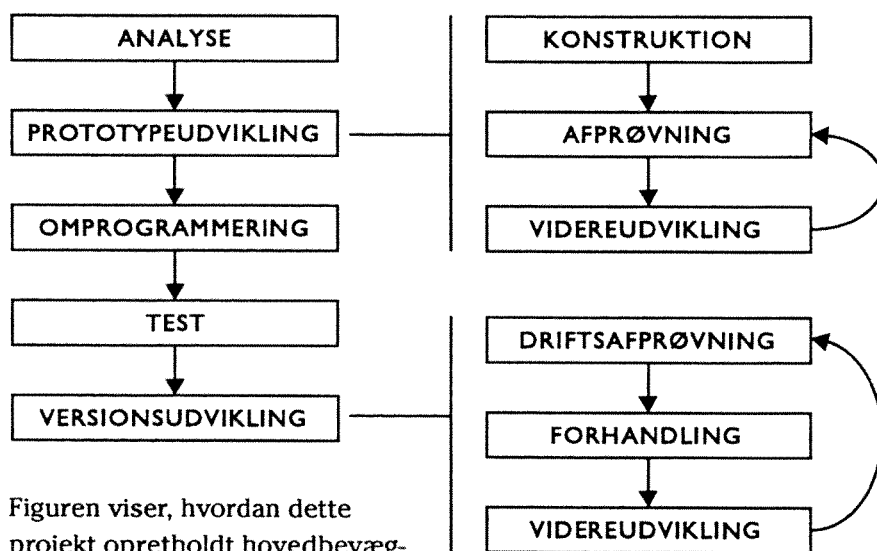


vikling, til prototypen vurderes færdig.

En sådan model kan illustrere forløbet af et enkelt, afgrænset eksperiment, nærmere betegnet et afprøvende eksperiment, hvor der ikke udforskes forskellige grundideer, og hvor prototypen ikke udvikles til det færdige system. Som model for et projektforbøb kan den kun illustrere små, trivielle projekter. Det er ikke muligt at opstille en generel model for projektforbøb med eksperimenter. Det er muligt at opstille en model for et konkret projekt, som illustrerer den overordnede udviklingsstrategi.

### Eksempler på projektforbøb

- I et større projekt blev prototypeudvikling rettet mod en kravspecifikation kombineret med versionsudvikling. Projektet omfattede såvel nye former for edb-anvendelse som nye programmeringsmetoder. Parallelt med analysen af anvendelsesområdet blev der derfor eksperimenteret med værktøjer til prototypeudviklingen, og under denne blev der eksperimenteret med værktøjer til det færdige system. Under omprogrammeringen foretoges enkelte afprøvende eksperimenter med brugergrænsefladen på grund af begrænsninger i det endelige maskinel.



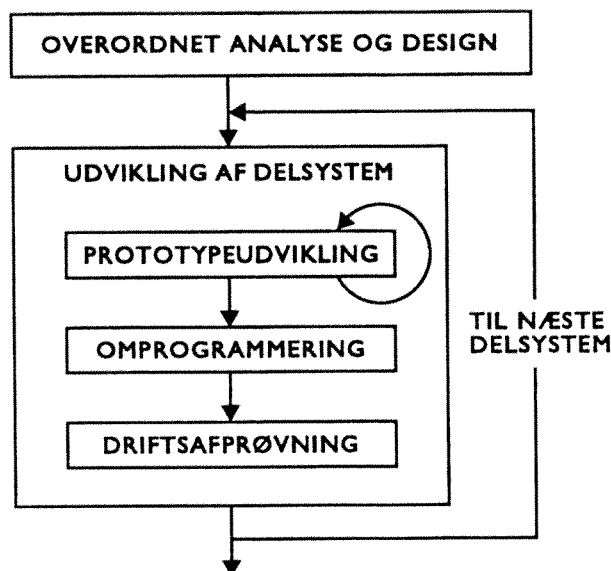
Figuren viser, hvordan dette projekt opretholdt hovedbevægelsesretningen i en traditionel projektforbøb, trods udstrakt anvendelse af små og store eksperimenter.

- Et stort metalindustriforetagende besluttede at udvikle et højt specialiseret CAD-system. Under den indledende analyse opstod det problem, at brugerne fandt det svært at afse tid til at diskutere detaljer i et edb-system, som - grundet dets størrelse - først ville blive en realitet efter flere år. Ledelsen viste også modvilje mod at afsætte større ressourcer i så lang tid uden at se resultater undervejs.



Det blev derfor besluttet at følge en trinvis udviklingsmodel, baseret på opdeling i delsystemer. CAD-systemet blev delt op i en række sideordnede delsystemer. Hvert delsystem skulle understøtte en bestemt gruppe af brugere. En prototype af et af delsystemerne blev udviklet og afprøvet i tæt samarbejde med brugere og derpå omprogrammeret og taget i brug. Undervejs opsamledes erfaringer med design, udviklingsværktøjer og arbejdsformer, samt erfaringer fra brug af delsystemet, til brug for udvikling af næste delsystem.

Dette udviklingsprojekt er et eksempel på iterationer på flere niveauer, små iterationer inden i store, som vist i figuren. Den trinvise udviklingsstrategi gav større engagement fra brugerne og mulighed for at afprøve generelle designideer undervejs, også i realistiske anvendelsessituationer.



- Udvikling af et toldsystem til en statslig institution foregik ved, at udviklere og slutbrugere blev isoleret i et kontor med terminaler. Først blev krav til systemet diskuteret, og skærbilleder blev opbygget og afprøvet. Mens brugerne indtastede stamdata, fortsatte udviklerne med at programmere funktionerne. Så snart en funktion var programmeret, blev den afprøvet. Tvivlsspørgsmål blev diskuteret og afgjort på stedet. Efter en uges tæt samarbejde var systemet færdigt.

Eksemplet viser den anden ende af spektret for eksperimenter i systemudvikling. Iterationerne er så hurtige, som de kan blive. Det er umuligt at opstille en model for et sådant forløb. Et eksperiment af denne form kan også indgå i større forløb, hvor prototypen omprogrammeres, eller hvor eksperimentet kun omfatter en del af et større system.

## Opsummering

Eksperimenter i systemudvikling er et bredt område. Eksperimenter kan have vidt forskellige formål. Prototyper kan være alt fra designskitser eller standardprogrammel til funktionsdygtige delsystemer. Eksperimenter kan indgå i forskellige stadier af et projektførløb.

Tredje del af bogen - Hvordan - vil gå mere i detaljer med, hvordan det enkelte eksperiment tilrettelægges og afvikles.

---

# HVORDAN

---

## Tilrettelæggelse

### OM PUNKTER FOR FORBEREDELSE

- Eksperimenter skal tilrettelægges og afvikles systematisk

Anvendelse af prototypeværktøjer udgør ikke i sig selv et eksperiment. Først når en prototype gøres til genstand for systematisk afprøvning og vurdering, er der tale om et eksperiment.

Hvad vil det sige at tilrettelægge og afvikle et eksperiment systematisk? Er det ikke en selvmodsigelse? Nedenfor er opstillet en række spørgsmål, som kan inddrages ved planlægning af et eksperiment. Rækken af spørgsmål er hverken obligatorisk eller udtømmende. Omfang og prioritering af planlægningen skal stå i et rimeligt forhold til omfanget af eksperimentet. Samtidig skal planlægningen ikke være så stram, at der ikke bliver plads til det egentlige: at eksperimentere.

Hvert af de følgende afsnit behandler et af nedenstående spørgsmål. Behandlingen tager udgangspunkt i et overordnet princip. Ud fra eksempler diskuteres problemer og muligheder. Til sidst i hvert afsnit opsummeres retningslinjer for handling. Heller ikke disse retningslinjer er obligatoriske eller udtømmende.

- ◆ Hvorfor skal der laves et eksperiment?
- ◆ Hvordan ledes eksperimentet?
- ◆ Hvem skal deltage, og hvad kræves af deltagerne?
- ◆ Hvad er formålet med eksperimentet?
- ◆ Hvordan skal prototypen konstrueres?
- ◆ Hvordan skal afprøvningen foregå?
- ◆ Hvordan dokumenteres eksperimentet?
- ◆ Hvordan skal resultatet vurderes?

## Anvendelse

### OM EKSPERIMENTER ELLER EJ

- Eksperimenter overflødiggor ikke traditionelle fremgangsmåder

Anvendelse af eksperimenter er ikke noget mål i sig selv. Det skal overvejes ud fra forholdene i det aktuelle projekt. Kan man sige noget generelt om, hvornår man skal anvende eksperimenter? Hvilke typer af edb-systemer egner sig for eksperimenter med prototyper? Her følger nogle eksempler på valgsituationer.

- En bank skulle have udviklet et system til budgetplanlægning for private kunder. Ideen med systemet var, at en budgetplan skulle kunne udvikles på stedet, sammen med kunden. Selv om anvendelsessituationen var anderledes end for bankens øvrige edb-systemer, foregik analyse og design udelukkende ved hjælp af formelle, strukturerede teknikker. Budgetsystemet blev konstrueret på normeret tid, men ibrugtagningen trak ud på grund af problemer med tilpasning til bankens praksis, hvilket krævede større ændringer i systemet.

I denne situation kunne udviklerne have valgt også at eksperimentere med den nye form for edb-anvendelse, f.eks. med et design til en større skærm, så kunden kan følge med og deltage direkte i tilpasning af budgetplanen.

- Under udvikling af kravspecifikation til et lager- og værkstedsstyringssystem blev udviklerne frustreret over værkstedspersonalets mangel på interesse for processen. De besluttede derfor at udvikle en prototype over værkstedsstyringen, hvilket tog lang tid på grund af dårlige værktøjer. Under afprøvning af prototypen viste det sig, at den manglende interesse skyldtes interne konflikter mellem lager og værksteder. Specifikationen havde taget udgangspunkt i lagerets krav, og lageret ville anvende edb-systemet til at kontrollere og styre værkstedernes arbejde. Det blev derfor nødvendigt at begynde delvis forfra med kravspecifikationen.

I denne situation førte et eksperiment ganske vist til en afklaring af modstridende krav til edb-systemet. Men det blev dyrt, selv om det var blevet endnu dyrere, hvis modstriden først var opdaget ved afprøvning af det færdige system. Hvis udviklerne havde analyseret årsagerne til den manglende interesse nøjere, kunne to personår måske være sparet.

- Forundersøgelse af en telefonomstilling viste tre mulige områder for edb-anvendelse: vedligeholdelse af telefonlister, opslag under betjening af opkald, og diverse udtræk af listerne til administrativt brug. Vedligeholdelse af lister på edb havde været prøvet nogle år tidligere. De administrative funktioner berørte kun én medarbejder nogle timer om

ugen og var lige til at definere. Opslag berørte alle ansattes vigtigste arbejdsfunktion. Udviklerne var usikre på, om de ansatte ville acceptere en så intens anvendelse af edb, og valgte derfor dette område ud til et afprøvende eksperiment. Eksperimentet medførte, at opslagsfunktionen blev godkendt som krav til edb-systemet.

I denne situation var der ikke tid til at lave en prototype over hele systemet. Afgrænsningen til en enkelt funktion blev foretaget ud fra to kriterier: usikkerhed og funktionens vigtighed.

Eksemplerne illustrerer, at eksperimenter ikke udelukker, men supplerer andre fremgangsmåder. Det er vigtigt at vurdere hvilken fremgangsmåde, der egner sig til den aktuelle situation. Traditionel analyse kan danne grundlag for udvælgelse og planlægning af eksperimenter.

Eksperimenter egner sig for situationer, der er præget af usikkerhed. Det er vigtigere at eksperimentere med det, man ikke kender, end med det, man kender. Men eksperimenter løser ikke usikkerhed, der skyldes intern uenighed i brugerorganisationen. Eksperimenter kan højst bidrage til at bringe uenighed for dagens lys.

### **Ingen klare retningslinjer**

Det er svært at opstille klare regler for, hvilke typer og størrelser af edb-systemer, der egner sig for eksperimenter med prototyper. Det afhænger af synet på eksperimentel systemudvikling og af prototypeværktøjernes udviklingsstade. Der er dog nogle hovedtendenser i litteraturen.

Udforskende og afprøvende eksperimenter anbefales til interaktive transaktions- og rapporteringssystemer, udviklende eksperimenter til beslutningsstøttesystemer.

Eksperimenter, hvor prototypen omprogrammeres, anbefales til store projekter, hvor designændringer sent i forløbet vil være meget bekostelige. Høj grad af usikkerhed, lang forventet levetid, et stort antal brugere eller høje kvalitetskrav til systemet taler for sådanne eksperimenter. Som forudsætning nævnes, at eksperimenter skal kunne gennemføres for mindre end 25% af det samlede budget.

Eksperimenter, hvor prototypen færdigudvikles, anbefales til mindre projekter, hvis hurtig udvikling er vigtigere end krav til teknisk kvalitet og effektivitet, eller hvis gevinsten ved omprogrammering ikke står mål med omkostningerne. Edb-systemer, der kræver tunge beregninger, håndtering af store datamængder eller avancerede, grafiske brugergrænseflader, egner sig - endnu - ikke for færdigudvikling i et prototypeværktøj.

Principielt kan alle former for edb-anvendelse gøres til genstand for eksperimenter, mens omvendt de færreste edb-systemer kan udvikles alene ved hjælp af eksperimenter med prototyper. Valg af fremgangsmåde hænger ikke altid sammen med valg af værktøj. Selv om et prototypeværk-

tøj vurderes at være egnet til anvendelse i det endelige edb-system, er det ikke nødvendigvis muligt at eksperimentere sig frem til et godt design.

- ◆ Anvend eksperimenter, når:
  - graden af usikkerhed er høj
  - nye former for edb-anvendelse skal udvikles
  - nyt maskinel/programmel skal anvendes
  - centrale arbejdsfunktioner skal understøttes
- ◆ Suppler eksperimenter med andre teknikker

## Ledelse

### OM PLANLÆGNING OG ORGANISERING

- Eksperimenter kræver regelmæssig projektvurdering og -planlægning.

Hvordan adskiller projektledelse under anvendelse af eksperimenter sig fra projektledelse i almindelighed? Erfaringer viser, at eksperimenter i systemudvikling giver specielle problemer med projektledelsen. Det er svært at vurdere fremdriften i iterative projektforløb. Det er svært at styre og stoppe udvikling af prototyper. Det er svært at vurdere tidsforbruget, når kravene til det endelige system ændres undervejs. Det er ikke muligt at planlægge hele projektforløbet fra starten, planlægningen må justeres undervejs.

Der er derfor behov for at tilpasse ledelsesaktiviteterne til de nye arbejdsformer. Her skal kort omtales nogle forslag til ledelsesaktiviteter og -strategier, der kan imødegå nogle af de specielle problemer i eksperimentel systemudvikling.

### Planlægning

Eksperimenter bryder mere eller mindre hovedrækkefølgen i traditionel systemudvikling, fra analyse over design til realisering, og dermed sammenhængen mellem tid og indhold (analyse herfra og dertil, derefter design indtil ...). Planlægning med referencelinjer ('baselines') er en teknik til projektplanlægning, som netop adskiller den indholdsmæssige fra den tidsmæssige planlægning. Det gør det nemmere at vurdere fremdriften i projekter, hvor hovedrækkefølgen ikke overholdes.

En referencelinje er en beskrivelse af en projektilstand, en 'milepæl'. En referencelinje beskriver bl.a. en række mellemprodukter samt kriterier og fremgangsmåde for disses vurdering. Et mellemprodukt kan f.eks. være en kravspecifikation eller en prototype. Ved planlagte kontrolpunkter vurderes, om en referencelinje er nået.

En referencelinje kunne bl.a. indeholde planlægningen af et eksperiment, med formål, afgrænsning af prototypen, fremgangsmåde ved afprøvninger og kriterier for vurdering. Et andet mellemprodukt i samme referencelinje kunne så være en præcisering af næste referencelinje, på baggrund af vurdering af eksperimentet. Denne næste referencelinje kunne f.eks. indeholde planlægning af et nyt eksperiment med samme prototype, og så videre.

### Vurdering af usikkerhed

Hvornår skal man starte et eksperiment, og hvornår skal man stoppe det igen? Regelmæssig vurdering af usikkerhedsfaktorer ('risk analysis') kan danne grundlag for styring af projektforsøb med såvel traditionel som eksperimentel systemudvikling. En såkaldt 'spiralmodel' for systemudvikling opstiller et cyklisk projektforsøb. Hver cyklus indledes med at opstille mål og vurdere usikkerhed for hvert mål. For hver usikkerhedsfaktor vælges passende fremgangsmåde, f.eks. et eksperiment, til at mindske usikkerheden. På baggrund af usikkerhedsfaktorer og fremgangsmåder besluttet videre udviklingsstrategi (cyklisk, lineær, eller en blanding) og omfang af andre aktiviteter (planlægning, specifikation, test). Hovedideen i modellen er, at karakteren og omfanget af projektkิจกรรมterne ikke er ensartet og fastlagt, men regelmæssigt tilpasses en vurdering af usikkerhedsfaktorer.

### Organisering

Projekstyringen kan styrkes ved at planlægge med forholdsvis korte, intensive forsøb. Det kan være bedre at planlægge tre korterevarende eksperimenter end ét længerevarende. Ved at afgrænse omfanget af de enkelte eksperimenter kan man mindske risikoen for, at et eksperiment fortaber sig i detaljer eller lægger sig fast på et dårligt design.

Eksperimenter fungerer bedst i små projektkupper. Små kupper giver mulighed for tættere samspil mellem analyse, design og realisering og mellem ledelses- og udførelsesaktiviteter. Store projektkupper med udtalt arbejdsdeling vil kræve meget kommunikation. Der er derfor tendens til at organisere eksperimenter i forholdsvis små projektkupper med et par udviklere, evt. flere i programmeringstunge perioder.

### Genforhandling

Hvordan skriver man kontrakt om levering af et edb-system, hvis der skal laves eksperimenter med prototyper? Hvordan lægger man rammer, der passer til en læreproces? En mulighed er at skrive separat kontrakt for en indledende prototypefase ('front-end contract'), så vurdering af prototypen kan danne grundlag for forhandling om den endelige kontrakt. Selv om der foreligger en overordnet kravspecifikation fra brugerorganisa-

tionens side ved projektstart, er det ikke sikkert, den holder til nærmere afprøvning i eksperimenter. Det anbefales derfor at indføre mulighed for genforhandling i projektkontrakter ('dynamic contracting'). Genforhandling kan indgå som planlagt trin i et projekt, baseret på versionsudvikling.

- ◆ Lav projektplanlægning ved hjælp af referencelinjer
- ◆ Lav regelmæssig vurdering af usikkerhedsfaktorer
- ◆ Planlæg med korte, intensive forløb i små projektgrupper
- ◆ Indfør mulighed for genforhandling i kontrakt

## Forudsætninger

### OM KRAV TIL DELTAGERNE

- Deltagerne skal forstå og acceptere anvendelse af eksperimenter  
Hvem skal deltage i eksperimentet, og hvad kræver det af deltagerne?
- Et reservationssystem blev udviklet gennem eksperimenter med prototyper. Terminaler blev installeret i brugermiljøet, så brugerne løbende kunne afprøve prototyperne ind i mellem det daglige arbejde. Ledelsen i brugerorganisationen afsatte ikke tid til oplæring af brugerne eller til, at brugerne kunne afprøve prototyperne. Resultatet blev derefter. Det blev overladt til hver enkelt bruger at stjæle sig tid til at 'lege' med prototyperne, som visse kolleger kaldte det. Det var derfor kun ganske få brugere, overvejende mellemledere, der fik afprøvet prototyperne. Slutbrugere blev først inddraget under den afsluttende test, hvor de kom med flere rimelige kritikpunkter, som burde være fremkommet under prototypeafprøvningen.

Eksemplet viser nødvendigheden af, at udviklerne blander sig i, hvilke brugere der deltager, og sikrer en vis forståelse hos brugerne og deres ledelse for, hvad det kræver at indgå i en eksperimentel systemudviklingsproces.

### Udvælgelse af brugerdeltagere

Eftersom et af formålene med eksperimenter er at få større indsigt i anvendelsesområdet, er det vigtigt at inddrage brugere på alle niveauer, og specielt slutbrugere. Det er slutbrugere, som ved hvordan arbejdet hidtil har foregået, og det er slutbrugere, der skal arbejde med det endelige system. Ledere og andre berørte brugere på alle niveauer skal også inddrages på et tidspunkt. De vil have forskellige syn på prototypen, hvilket også kan medvirke til at bringe eventuelle uenigheder for dagen. Hvilke brugere, der skal deltage, afhænger af formålet med det enkelte

eksperiment og kan derfor variere gennem et projektforsøg.

Det væsentligste krav til brugerdeltagerne er faglig kompetence indenfor det arbejdsområde, der omfattes af eksperimentet. Beslutningskompetence er en forudsætning ved eksperimenter, hvor der skal vælges mellem alternative løsningsforslag, eller hvor prototypen skal godkendes. Motivation er en forudsætning, men kritisk holdning til edb er bedre end udelte begejstring, hvis deltagerne skal repræsentere andre brugere. Kendskab til edb kan være en fordel, f.eks. under udforskende eksperimenter, men kan også være uønsket, f.eks. under afprøvning af brugergrænseflader.

## Nye krav

Eksperimenter stiller nye krav til projektdeltagerne. Samarbejdet mellem brugere og udviklere bliver tættere, og programmeringsarbejdet ændrer karakter. Udviklingsforløbet bliver mere dynamisk og mindre forudsigeligt, kort sagt mere kaotisk. Det kan give forskellige problemer for projektdeltagerne.

Brugerne kan blive overrasket over, hvor tidskrævende det kan være at deltage i eksperimenter, specielt hvis de har erfaring med traditionel systemudvikling. Specielt ledelsen forventer umiddelbart, at indførelse af edb giver lettelse i det daglige arbejde, ikke yderligere belastninger. Det kan være en uønsket rolle for brugerne at skulle komme med forslag til edb-anvendelse. Brugere kan have svært ved at acceptere, at en prototype har fejl og mangler. Efter endt eksperimenteren kan det være svært for brugere at forstå, at der kan være langt fra en prototype til et færdigt system.

Brugerne skal derfor forberedes på deltagelse. F.eks. kan brugerdeltagere fra andre projekter fortælle om erfaringer med eksperimenter. Brugerne skal først og fremmest afsætte tid til at deltage. Det kan være nødvendigt med vikardækning.

Udviklerne kan også have svært ved at acceptere det rolleskift, eksperimenter medfører. Samarbejde med brugere stiller større krav til arbejdsanalytiske og sociale evner. Nye programmeringsværktøjer og -metoder skal indlæres. Manglende kendskab til værktøjet kan distrahere og stjæle tid fra selve eksperimentet. Udviklere kan have svært ved at acceptere kritik af 'deres' prototype fra brugere. Udviklere kan nære modvilje mod at konstruere 'mangelfulde' prototyper og mod at 'kassere' prototypen.

Udviklere skal derfor også forberede sig på deltagelse. Lav f.eks. en studiekreds og et praktisk forsøg, et eksperiment med eksperimenter. Sørg for uddannelse i prototypeværktøjet, og evt. teknisk assistance ved programmering. Udviklere kan også placere sig i brugerorganisationen i en periode. Det styrker kommunikationen med brugerne og giver bedre mulighed for at vurdere brugernes forslag og selv komme med forslag.



Ledelsen i såvel bruger- som i udviklerorganisationen kan være utryk ved tendenser til decentralisering og uddelegering af ansvar, som ligger i eksperimenter i systemudvikling. Eksperimenter udføres bedst i forholdsvis små og selvstændige projektgrupper. Centralistiske ledelsesformer virker hæmmende på eksperimenter.

Arbejd for følgende forudsætninger for eksperimentet:

- ◆ Brugerne skal:
    - være motiverede
    - afsætte tilstrækkelig tid til deltagelse
    - have faglig og beslutningsmæssig kompetance
    - være repræsentativt udvalgt
    - orienteres om muligheder og begrænsninger i eksperimenter
    - kende deres rolle og ansvar i eksperimentet
  - ◆ Udviklerne skal:
    - være indstillet på at samarbejde med brugerne
    - have kendskab til de valgte udviklingsværktøjer
    - sætte sig ind i arbejdsområdet, før en prototype udvikles
  - ◆ Ledelsen skal:
    - være åben overfor eksperimenter og uddelegering af ansvar
- 

## Målsætning

### OM PRÆCISERING AF FORMÅLET

- Formålet med eksperimentet skal afgrænses og præciseres
 

Hvordan planlægger man et eksperiment, så indsatsen står i rimeligt forhold til det forventede udbytte?
- Udvikling af en prototype på et beslutningsstøttesystem tog længere tid end beregnet. Brugernes afprøvning af prototypen blev hele tiden afbrudt af fejl. Brugerne var vant til at anvende færdige edb-systemer og var ikke blevet forberedt på fejlsituationer. For nogle brugere gik der ligefrem sport i at fremprovokere fejl. Udviklerne brugte meget tid på at undersøge fejlene og gøre prototypen mere robust, selvom den var planlagt til at skulle omprogrammeres.
 

Eksemplet viser en situation, hvor formålet med eksperimentet - og med eksperimenter i det hele taget - har været uklart. Udviklerne havde ikke

præciseret prototypens begrænsninger overfor brugerne og satte ikke grænser for videreudvikling af prototypen.

### Præcisering af formålet

Brugere har en vis evne til at finde fejl i edb-systemer. Denne evne udnyttes f.eks. i ekstern test. Den udnyttes også i eksperimenter med prototyper, men her kan den give problemer. En prototype er bevidst mangelfuld. Den konstrueres til at udfylde en bestemt, afgrænset del af en tænkt edb-anvendelse. Under afprøvning af en prototype vil udviklere mere eller mindre bevidst holde sig indenfor disse begrænsninger, mens brugere netop vil træde udenfor. Brugere har ikke samme forudsætninger som udviklere for at skelne mellem, hvad der er fejl og mangler indenfor den aktuelle prototype, og hvad der skyldes bevidste begrænsninger.

Det er nødvendigt at afgrænse og præcisere formålet med eksperimentet overfor deltagerne. Hvilke usikkerheder skal afklares? Hvilke delsystemer og funktioner skal udforskes, afprøves og/eller udvikles? Hvilke konsekvenser af edb-systemet skal undersøges? Ellers kan der spildes tid på fejl, som ikke er fejl, men udtryk for prototypens udviklingsgrad. Der kan spildes tid på at udbygge prototypen med detaljer, som ikke er relevante for afprøvningen. Det kan være svært at vurdere, hvornår eksperimentet er færdigt. Jo større udviklingsprojekt, eksperimentet indgår i, jo vigtigere er det at afgrænse og præcisere formålet.

### Anvendelse af prototypen

Formålet med et eksperiment hænger sammen med den forventede anvendelse af prototypen. Anvendelse af prototypen bør derfor planlægges sammen med formålet. Skal prototypen omprogrammeres eller færdigudvikles? Skal den anvendes i andre eksperimenter? Skal den anvendes til undervisning eller demonstration?

Beslutningen om anvendelse kan evt. udskydes til den afsluttende vurdering af eksperimentet. Hvorvidt prototypen skal omprogrammeres, kan f.eks. gøres betinget af udfaldet af afprøvende eksperimenter rettet mod effektivitet og svartider.

Planlægning af anvendelse skal ikke hindre, at beslutningen kan ændres efter vurdering af eksperimentet. Det skal blot sikre, at en senere kursændring følges op af en revurdering. Før en prototype tages i anvendelse som færdigt system, kræver det grundig vurdering af, hvad der skal til for at gøre prototypen 'produktmoden'.

- ◆ Præciser formålet med eksperimentet overfor deltagerne
- ◆ Lad formålet danne udgangspunkt for konstruktion, afprøvning og vurdering af prototypen
- ◆ Planlæg anvendelsen af prototypen sammen med formålet

# Konstruktion

## OM PROTOTYPSENS INDHOLD OG OMFANG

- Prototypen skal konstrueres ud fra eksperimentets formål  
Hvor omfattende og detaljeret skal prototypen være?
- Til undersøgelse af muligheden for administrativ edb-anvendelse afsatte en mindre virksomhed to måneder til udvikling af en prototype. Udvikleren valgte at 'kopiere' det eksisterende, manuelle system og at konstruere en prototype med alle centrale funktioner og en fuldt færdig brugergrænseflade. Der nåedes kun to iterationer indenfor den fastsatte tid. Ved de to afprøvninger var brugerne umiddelbart meget positive, men det viste sig at være uhensigtsmæssigt at kopiere det eksisterende system. Der blev ikke tid til at udvikle og afprøve andre muligheder.

Eksemplet viser en situation, hvor ambitionsniveauet for konstruktion af prototypen ikke svarede til eksperimentets overvejende udforskende formål. Her kunne det have været bedre at nå flere iterationer eller præsentere flere forslag i en mindre færdig form.

En prototype er en ufærdig model af et edb-system. Den er et oplæg til diskussion og vurdering. Det er afprøvningen, og ikke prototypen selv, der er målet med eksperimentet. Det er de nye og ukendte sider af edb-systemet, der er mest interessante. En tilsyneladende komplet prototype, uden alternative løsningsforslag, lægger ikke op til at involvere brugere i diskussion af muligheder for anvendelse af edb. En sådan prototype egner sig bedre til eksperimenter, hvor formålet er langtidsafprøvning eller endelig godkendelse.

## Bredde og dybde

En prototypes udviklingsgrad er et udtryk for, i hvor høj grad prototypen afspejler det tænkte edb-system. Udviklingsgraden har to dimensioner: bredden og dybden. Disse dimensioner går igen i betegnelserne 'vandrette/lodrette' prototyper. Bredden er udtryk for, hvor stor en del af det endelige edb-system, prototypen omfatter. Dybden er udtryk for, i hvor høj grad prototypen kan simulere eller udføre funktioner på data.

Prototypens udviklingsgrad er afgørende for udbyttet af eksperimentet. Og omvendt: det erklærede formål med et eksperiment stiller krav til prototypens udviklingsgrad. Mere realistiske eksperimenter kræver større udviklingsgrad. Men større prototyper tager længere tid at udvikle, hvilket betyder større udgifter eller færre iterationer. Det centrale i eksperimenter i systemudvikling er iteration og samarbejde. Jo hurtigere prototypen konstrueres, jo flere iterationer kan der nås, og jo nemmere er det at opretholde brugernes interesse og aktive deltagelse i samarbejdet.

Valget af udviklingsgrad er altså et kompromis. Valget afhænger af det overordnede formål med eksperimentet. Ved udforskende eksperimenter lægges vægt på hurtig præsentation af flere alternative løsningsforslag. Ved afprøvende eksperimenter lægges vægt på, at udviklingsgraden inden for det afgrænsede område er tilstrækkelig høj til at give udbytte af eksperimentet. Udviklingen er typisk i dybden, med mindre formålet er at afprøve sammenhænge mellem delsystemer, f.eks. i brugergrænsefladen. Ved udviklende eksperimenter lægges vægt på høj udviklingsgrad på begge leder, i dybden i den enkelte version eller det enkelte delsystem, og i bredden i løbet af den trinvis udvikling.

Vægtningen mellem bredde og dybde er en afvejning mellem overblik og realisme. Arbejder man kun i bredden, risikerer man, at afprøvningen af prototypen bliver for kedelig for brugerne. Arbejder man kun i dybden, risikerer man at fortabe sig i detaljer. Generelt er realismen vigtigst for udbyttet af et eksperiment. Overblikket kan opnås ved andre teknikker end eksperimenter med prototyper.

### Omfang og udseende

- Under forundersøgelse af et enkeltbrugersystem til løbende registrering af igangværende opgavers tidsforbrug blev brugeren præsenteret for skitser af skærbilleder. Skitserne var lavet på baggrund af et par interviews og gennemgang af eksisterende, papirbaseret registrering. I et af skærbillederne kritiserede brugeren den prioritering, der lå i de forskellige felters placering og størrelse. Den afspejlede de forskellige datas formelle vigtighed, men ikke deres reelle betydning i det daglige arbejde. Kritikken førte til afklaring af, hvordan edb-systemet skulle indgå i arbejdet.

Eksemplet viser, hvordan designskitser - på papir eller på skærm - kan danne grundlag for hurtige og billige eksperimenter, hvis brugerne er i stand til at forholde skitserne til en anvendelsessituation. Prototypen skal afpasses efter brugerne. I hvor høj grad er de i stand til at se bort fra mangler og begrænsninger? I hvor høj grad vil fejlsituationer gøre dem utrygge eller negative overfor afprøvningen?

Eksemplet illustrerer også, at det kan være en god ide at inddrage det grafiske design af skærbilleder i tidlige eksperimenter. Grafisk design er ikke noget, man bare føjer til bagefter. Det visuelle udtryk påvirker opfattelsen af indholdet og kan give ideer til design af såvel database som brugergrænseflade og funktionalitet. Hvis prototypen skal omprogrammeres, skal der også tages højde for muligheder og begrænsninger i det endelige maskinel og programmel. Det er specielt farligt at forsyne prototypen med faciliteter, som er umulige eller urimeligt dyre at realisere i det endelige programmeringsmiljø.

- ◆ Styr ambitionsniveauet ind efter formålet med eksperimentet
  - ◆ Afpas prototypens udviklingsgrad efter brugernes vilje og evner
  - ◆ Konstruer og ret prototypen hurtigt
  - ◆ Prioriter alternative forslag ved udforskende eksperimenter
  - ◆ Prioriter funktionalitet ved afprøvende eksperimenter
  - ◆ Prioriter realisme højere end overblik
  - ◆ Inddrag det grafiske design fra starten
  - ◆ Tag højde for det endelige maskinel og programmel
- 

## Afprøvning

### OM STYRING OG UDFOLDELSE

- Afprøvning af prototypen skal forholde sig til anvendelsessituationen

Når en prototype er udviklet, skal den demonstreres, afprøves, diskuteres, der skal afvikles et eksperiment med prototypen. Hvordan skal afprøvningen foregå?
- I to udviklingsprojekter blev udarbejdet 'horisontale' prototyper i form af en række skærbilleder og deres indbyrdes sammenhæng. Prototyperne var tænkt som alternativ til formelle beskrivelser, som oplæg til godkendelse og eventuelt justering af design. Udviklerne vurderede, at det ville blive for tidskrævende at udbygge prototypen med simuleret funktionalitet, dvs. funktioner på simulerede data.

I det ene projekt blev prototypen installeret i brugermiljøet. Efter en fælles demonstration blev brugerne opfordret til at afprøve systemet og fremkomme med forslag på senere projektmøder. Responsen på disse møder var langt mindre, end udviklerne havde forventet. Dette blev udlagt som almindelig tilfredshed med designet. Det viste sig senere, at brugerne havde været usikre overfor, hvordan de skulle afprøve prototypen.

I det andet projekt blev prototypen ikke installeret i brugerorganisationen. I stedet fik hver af brugerne en demonstration på tomandshånd. Under demonstrationen beskrev udvikleren, hvordan funktionaliteten var tænkt, og hvordan arbejdsgangen ville blive under anvendelse af edb-systemet, og diskuterede dette med brugeren. Denne fremgangsmåde gav god respons, bl.a. flere konstruktive designforslag.

I begge projekter blev edb-systemernes funktionalitet først afprøvet af brugere under den afsluttende test. Her fremkom nye og væsentlige kritikpunkter, hvoraf nogle blev ret tidskrævende at imødekomme.

## Udbytte af afprøvning

Eksemplet illustrerer flere faktorer, som påvirker udbyttet af et eksperiment.

Afprøvningen af en prototype skal tilrettelægges. Det er sjældent gjort med at sende en prototype med posten og så gå og vente på, at brugerne skal ringe. Hvad enten udviklerne skal deltage eller ej, skal der laves retningslinjer for, hvordan afprøvningen skal forløbe. Hvad skal der fokuseres på, hvad skal der ikke fokuseres på? Her er det igen en fordel at have præciseret formålet med eksperimentet.

En prototype er ikke bare et visuelt medie for en kravspecifikation, som fremlægges til godkendelse. En prototype er et oplæg til diskussion. Det handler om at styre og stimulere diskussionen, ikke om at undgå den. Udviklerne kan stimulere samarbejdet ved selv at stille forslag og ved at sætte spørgsmålstegn ved brugernes forslag. Brugeres viden om muligheder i edb kan være ret begrænset, og deres ønsker afspejler ikke nødvendigvis reelle behov.

Afprøvningen skal forholde sig til anvendelsessituationen. Det er den, og ikke prototypen i sig selv, der er interessant for brugerne. Selv ved demonstration af skærbilleder er det muligt at forholde sig til, hvordan systemet skal betjenes i dagligt arbejde.

## Afprøvningsform

Valg af afprøvningsform afhænger af formålet med eksperimentet og af prototypens udviklingsgrad. Valget er en balancegang mellem faktorer som styring, frihed og tryghed.

Styring er nødvendig, i form af instrukser for forløbet af afprøvningen. Løse instrukser øger risikoen for, at eksperimentet ikke opfylder sit formål. Stramme instrukser kan gå ud over kreativiteten i samarbejdet. Rammerne påvirker også brugernes tryghed i afprøvningssituationen. Stramme instrukser, som en bruger udfører under overvågning, giver nemt det indtryk, at det er brugeren og ikke prototypen, der er til afprøvning. Afprøvning i brugermiljøet uden overvågning giver brugerne friere rammer for udfoldelse, men kan også øge utrygheden, hvis ikke brugerne er fortrolige med prototypen og med formålet med afprøvningen. Tryghed afhænger også af udviklernes indstilling til samarbejdet med brugerne. F.eks. kræver afprøvning af en brugergrænseflade, at udviklerne kan holde kæft og lade brugeren afsløre grænsefladens utilstrækkelighed uden indblanding.

Generelt egner brugerafprøvning og driftsafprøvning sig bedst til afprøvende og udviklende eksperimenter senere i forløbet, hvor prototypen har nået en vis udviklingsgrad og robusthed. Under udforskende og

afprøvende eksperimenter, hvor det drejer sig om at generere og vælge mellem løsningsmuligheder, er styring og tæt samarbejde vigtigst.

- ◆ Vælg afprøvningsform efter behov for samarbejde, styring, frihed og tryghed
- ◆ Lav retningslinjer for forløbet af afprøvningen
- ◆ Præciser prototypens mangler og begrænsninger
- ◆ Sæt spørgsmålstejn ved brugernes forslag, stil selv forslag
- ◆ Lad være med at forsvare prototypen

## Dokumentation

### OM OPSAMLING AF NY VIDEN

- Et eksperiment er en gensidig læreproces, resultatet er ny viden
- Gode prototypeudviklingsværktøjer indeholder faciliteter til dokumentation af prototypen. Men resultatet af et eksperiment er ikke bare en prototype.
- Et programmefirma gik i samarbejde med en virksomhed om at afprøve en ide til edb-anvendelse. Der blev først udviklet en omfattende prototype. Erfaringerne med denne var så positive, at der blev skrevet kontrakt om udvikling af et færdigt system. Forinden blev prototypen 'oversat' til en formel kravspecifikation, der kunne danne grundlag for en kontrakt.

Under afprøvning af det endelige system viste det sig, at det på en række punkter afveg fra brugernes opfattelse af prototypen. En del af den fælles forståelse af prototypen var gået tabt i 'oversættelsen'. Deltagerne var ikke gået kritisk ind i formuleringerne i kravspecifikationen, de havde jo godkendt prototypen. Samtidig var størstedelen af udviklerne blevet skiftet ud i overgangen fra prototypeudvikling til omprogrammering. Resultatet var, at udviklerne forholdt sig til kravspecifikationen, mens brugerne forholdt sig til prototypen.

Eksemplet illustrerer to forhold. For det første løser eksperimenter med en prototype ikke nødvendigvis alle kommunikationsproblemer omkring kravspecifikation. Der kan stadig være brug for at supplere prototyper med en formel kravspecifikation. For det andet kan en prototype alene ikke dokumentere et eksperimenterende forløb. Prototypen er blot det synlige slutresultat af dette forløb. Under eksperimentet konfronteres brugerne med en prototype og med udviklernes nærgående spørgsmål og tvinges derved til at reflektere over deres egen og hele organisationens

arbejdsgang. Denne refleksion kan give brugere og udviklere ny indsigt i brugerorganisationen og den fremtidige edb-anvendelse. Denne indsigt udmønter sig i nye krav og ønsker til prototypen og danner grundlag for designbeslutninger. Bag prototypen ligger således en opfattelse af en fremtidig anvendelsessituation og dermed en række begrundelser for designvalg. Hvis denne opfattelse kun eksisterer i hovederne på deltagerne, kan der opstå problemer i forbindelse med ombemanning af projektet.

Dokumentation af læreprocessen kan være dagbøger eller beslutningsreferater fra afprøvningen, hvor også begrundelser for kritik, ønsker m.m. noteres. Hvis flere udviklere deltager i afprøvningen, kan den ene referere. Hvis kun én udvikler deltager, kan referatskrivning være for tungt. En anden mulighed er at opsamle stikord undervejs og slutte af med at opsummere beslutninger og kritikpunkter. Opsummeringen kan så refereres, evt. optages på bånd. Afprøvningen kan også optages på video, hvis udviklerne ikke er med til afprøvningen eller af andre grunde vil kunne gentage forløbet.

- ◆ Dokumentér ikke kun prototypen, også afprøvningen
  - ◆ Opsaml ikke kun krav og beslutninger, også begrundelser
- 

## Vurdering

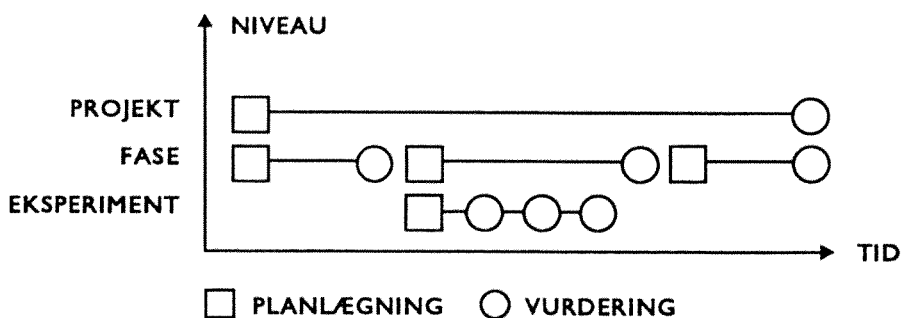
### OM NIVEAUER AF PLANLÆGNING

- Et eksperiment skal vurderes i forhold til alle niveauer af planlægning  
Under afprøvning af en prototype fremkommer en række ønsker og forslag til ændringer og udvidelser af prototypen. Såvel bruger som udvikler kan blive ved med at få nye ideer til udvidelser. Hvilke forslag skal med i prototypen? Hvad stiller man op med resten af forslagene?
- Under udvikling af prototypen på et stort, administrativt edb-system opstod ofte usikkerhed om, hvordan man skulle prioritere alle de små og store krav til forbedring af prototypen. Der var ingen retningslinjer for, hvorvidt et krav skulle med i prototypen, skulle indføres i kravspecifikationen, eller helt afvises. Det endte som regel med, at de fleste krav førte til ændring eller udvidelse af prototypen, som brugerne ønskede det. Dette var medvirkende til, at prototypeudviklingen kom til at koste fire gange så meget persontid som forventet.  
Eksemplet viser først og fremmest nødvendigheden af en stram styring af prototypeudvikling. Men det viser også nødvendigheden af retningslinjer for vurdering af resultatet af en prototypeafprøvning. En forudsætning for



vurdering er, at eksperimentet har et afgrænset formål. Hvilke delsystemer og funktioner skal udforskes, afprøves og/eller udvikles? Hvilke konsekvenser af edb-systemet skal undersøges? Hvordan skal prototypen kunne anvendes efter eksperimentet?

Ud fra formålet kan opstilles kriterier for, hvornår eksperimentet anses for gennemført. Men vurderingen begrænser sig ikke til det aktuelle eksperiment. Eksperimentet må også vurderes i forhold til rammerne for den igangværende fase og for hele projektet.



Figuren viser en projektsituation på tidspunktet for vurdering i et igangværende eksperiment.

Formålet med det igangværende eksperiment er en del af planlægningen. Sammen med formålet er opstillet kriterier for vurdering af, om formålet er opfyldt. Det er ét niveau for vurdering af resultatet af en afprøvning. Er de nye krav til prototypen nødvendige for at opfylde formålet med det igangværende eksperiment? Skal formålet udvides, eller skal der laves et andet eksperiment med et andet formål?

Eksperimentet indgår som en aktivitet i en projektfase. Planlægningen af fasen beskriver en projekttilstand, som skal nås, for at fasen er slut. Planlægningen rummer kriterier for vurdering af, om tilstanden er nået. Det er et andet niveau for vurdering. Er de nye krav til prototypen relevante for den igangværende fase, eller skal de udskydes til en senere fase?

Projektfasen er en del af det samlede projektforsløb. Mål og rammer for projektet er fastsat i projektgrundlaget eller kontrakten. Kravene til det færdige system samles i en kravspecifikation, evt. suppleret af prototypen. Det er et tredje niveau for vurdering. Er de nye krav til prototypen indenfor projektets rammer, eller kræver de genforhandling af kontrakten? Skal de udskydes til en eventuel senere version? Skal de indføres i kravspecifikationen uden at komme med i prototypen?

Ovenstående model af niveauer for vurdering beskriver en idealsituation, som ikke er dækkende for praktiske systemudviklingsprojekter. Vurdering af et eksperiment er ikke bare en rationel proces. Det er også en

politisk og økonomisk forhandlingsproces, som påvirkes af konflikter i såvel bruger- som udviklerorganisationen. Formålet med modellen er at understrege nødvendigheden af at afklare niveauerne af planlægning og vurdering i det konkrete projekt. Hvilke niveauer er der, hvad er kriterierne for vurdering, og hvem træffer afgørelserne på hvert niveau?

- ◆ Fastlæg kriterier for vurdering af eksperimentet
- ◆ Afklar niveauer for vurdering af eksperimentet
- ◆ Suppler prototypen med en kravspecifikation

---

# AFSLUTNING

---

## Alternativ eller supplement?

### OM TRADITIONEL OG EKSPERIMENTEL SYSTEMUDVIKLING

Der er ikke noget nyt i at eksperimentere i systemudvikling. Behovet for at udskille eksperimentel systemudvikling som en særlig tilgang til systemudvikling er udsprunget dels af problemer i praktisk systemudvikling, dels af muligheder i nye værktøjer.

Der har fra starten været to forskellige syn på eksperimentel systemudvikling: som supplement eller alternativ til traditionel systemudvikling. De to forskellige syn ser ud til at nærme sig hinanden. Fra den ene side åbnes for at lade en optimeret prototype indgå i det færdige system. Fra den anden side lægges vægt på at supplere en eksperimentel udviklingsmodel med formelle fremgangsmåder til analyse, design og dokumentation.

Der er en modsætning mellem eksperimentel og traditionel systemudvikling. De tre overordnede formål med eksperimenter – udforskende, afprøvende, udviklende – repræsenterer stigende grader af brud med traditionel systemudviklings anvendelse af formelle fremgangsmåder og adskillelse af analyse, design og realisering. Der er en modsætning mellem det at eksperimentere og det at planlægge og styre. Eksperimenter er i sig selv uforudsigelige og fungerer dårligt under bureaukrati. Det handler om at administrere disse modsætninger og finde en balance fra skridt til skridt. Det handler ikke om at vælge side og hoppe videre på ét ben.

Eksperimentel systemudvikling er ikke en ny standardmetode. Eksperimentel systemudvikling er en samling af fremgangsmåder, strategier og arbejdsformer. Det handler om at vælge ud og prioritere. I denne bog stilles derfor flere spørgsmål, end der gives svar. Handlingsanvisninger er forslag til overvejelse, ikke direktiver.

Morale: Man lærer ikke at svømme alene ved at få et korkbælte om livet og instruktion på bassinkanten. Man lærer det heller ikke ved bare at springe ud på det dybe med det samme. Muligheden er der, men resultatet er sjældent kønt eller effektivt, og risikoen er for stor. Udvikling af eksperimenter i systemudvikling handler ikke bare om at anskaffe nye værktøjer, følge nye kurser eller læse nye bøger. Det handler først og fremmest om at prøve eksperimenter i praksis, at eksperimentere med systemudvikling. God fornøjelse!

---

# REFERENCER

---

## Indledning

Erfaringer med eksperimentel systemudvikling er f.eks. refereret i Budde m.fl., Andersen & Andersen og Christensen m.fl.

## Hvorfor

Opdelingen af systemudvikling i analyse, design og realisering er taget fra Andersen m.fl. Fremstillingen af de to syn på systemudvikling trækker på Mathiassen & Stage. Balancen mellem accept og provokation er inspireret af Ehn. Opfattelsen af systemudvikling som læreproces for brugere og udviklere beskrives nærmere i flere artikler, f.eks. Floyd og Floyd m.fl. Jørgensen beskriver læreprocessen i systemudvikling ved hjælp af eksempler på menneskelig indlæring i almindelighed. Kauber argumenterer for iteration og eksperimenter som en generel strategi for beslutning og udvikling i organisationer.

## Hvilke

Opdelingen af eksperimenter efter tre overordnede formål er taget fra Floyd. Opremsningen af områder for afprøvning er baseret på Mogensen. Eksperimenter med standardprogrammel behandles af Janson.

Den forenklede model af et projektforsøg er baseret på Jenkins, som giver eksempler på, hvordan den kan indbygges i en traditionel projektmodel. En mere nuanceret (og indviklet) model for projektforsøg, hvor en prototype anvendes til afklaring af kravspecifikation, findes i Boar. En model for versionsudvikling findes i Floyd m.fl..

## Hvordan

### Tilrettelæggelse

Opstillingen af spørgsmål til forberedelse af et eksperiment er overvejende baseret på Christensen m.fl., som desuden opstiller underspørgsmål til hvert spørgsmål.

### Beslutning

Kriterier for typer af edb-systemer er fra Cervený m.fl. og Boar. Lantz kombinerer eksperimentelle og formelle fremgangsmåder i en metode for versionsudvikling. Boehm opstiller en projektmodel, hvor vurdering af usikkerhedsfaktorer bestemmer valget af eksperimentelle og formelle teknikker.

### Ledelse

Projektplanlægning med referencelinjer er beskrevet af Andersen m.fl. og præsenteres kort af Christensen m.fl. og Floyd m.fl. Spiralmodellen er fra Boehm. Projektledelse under prototypeudvikling behandles også af Boar og Lantz.

### Deltagerne

Christensen m.fl. og Molich har overvejelser over betingelser for samarbejde med brugere.

### Konstruktion

Afvejningen af prototypens udviklingsgrad i forhold til antal iterationer og til brugernes forudsætninger er inspireret af Mogensen.

### Afprøvning

Afvejningen mellem faktorerne styring, frihed og tryghed er baseret på Reisin & Wegge. Tilrettelæggelse og afvikling af eksperimenter med brugergrænseflader er detaljeret beskrevet i Molich.

### Dokumentation

Molich behandler dokumentation af eksperimenter med brugergrænseflader.

---

# LITTERATUR

---

Listen omfatter såvel referencer som forslag til videre læsning. Bøger og rapporter, som har været centrale i arbejdet med bogen, og som er almindeligt tilgængelige, er kort kommenteret.

## Andersen m.fl.

Niels Erik Andersen, Finn Kensing, Monika Lassen, Jette Lundin, Lars Mathiasen, Andreas Munk-Madsen, Pål Sørgaard:

### **Professionel Systemudvikling - erfaringer, muligheder og handling**

Teknisk Forlag, København 1986

Bogen er et resultat af MARS-projektet ("Metodiske ARbejdsformer i Systemudvikling"). Projektet undersøgte praktisk systemudvikling i store danske edb-virksomheder og -afdelinger, og eksperimenterede med indførelse af nye arbejdsformer. I bogen opstilles en teori for systemudvikling i form af en model over systemudviklingens hovedelementer og en række teser. Teseerne danner udgangspunkt for behandling af aktiviteter i praktisk systemudvikling. Der afsluttes med strategier for forandring af arbejdsformer.

Eksperimentel systemudvikling omtales ganske kort, men mange af bogens teser og handlingsanvisninger understøtter anvendelse af eksperimenter. Det gælder f.eks. projektplanlægning med referencelinjer, som bygger på begrebet 'baselines' fra engelsksproget litteratur om styring af systemudviklingsprojekter.

## Andersen & Andersen

Poul Elmo Andersen, Leif Andersen:

### **Erfaringer med eksperimentel systemudvikling**

Afsluttende opgave fra datanom-uddannelsen, nr. 84/7

Rapporten beskriver erfaringerne med anvendelse af eksperimentel systemudvikling i fem danske virksomheder. Erfaringer er indsamlet ved interview af udviklere og brugere. Alle de undersøgte projekter har anvendt et 4. generationsværktøj til at udvikle en prototype, der er blevet taget i brug som det endelige

system. Rapporten rummer således ingen erfaringer med eksperimenter rettet mod udvikling af en kravspecifikation, men ser alligevel sådanne eksperimenter som et overgangsfænomen, til bedre værktøjer og metoder til styring af færdigudvikling er udviklet. I afsluttende overvejelser over en sådan metode lægges vægt på en præcis målsætning med kriterier for vurdering, og på en stramtidsplan, baseret på opgaver af et par ugers varighed.

## Boar

Bernard Boar:

### **Application Prototyping - a requirements definition strategy for the 80s**

John Wiley & Sons, New York 1983

I denne bog opstilles en model for prototypeudvikling, kaldet "The Prototype Life Cycle". Modellen anbringes i definitionsfasen i en projektmodel, dvs. rettet mod udvikling af krav til edb-systemet. Modellen omfatter udvikling af en formel kravspecifikation som supplement til prototypen. Modellen retter sig mod udvikling af store transaktionssystemer, men der foreslås modifikationer ved udvikling af andre størrelser og typer af edb-systemer. Der åbnes også op for eksperimenter tidligere i forløbet, under forundersøgelse, og præsenteres en revideret model for færdigudvikling af prototypen.

Bogen rummer afsnit om opbygning af et "prototype-center", hvor maskinel, programmel og bemanding bidrager til at understøtte projekter, der skal anvende prototypeudvikling. Et andet afsnit giver forslag til, hvordan prototypeudvikling introduceres i en organisation eller et projekt. Til sidst argumenteres mod de mest almindelige indvendinger mod udvikling af prototyper.

Bernard Boar:

### **Application Prototyping: A Life Cycle Perspective**

Journal of Systems Management, February 1986

I denne artikel præsenteres kort ovennævnte "Prototype Life Cycle" i let revideret form.

**Boehm**

Barry W. Boehm:

**A Spiral Model of Software Development and Enhancement**

IEEE Computer, May 1988

**Budde m.fl.**

Reinhard Budde, Karin Kuhlenkamp, Lars Mathiassen, Heinz Züllighoven (red.):

**Approaches to Prototyping**

Springer Verlag, Berlin-Heidelberg, 1984

Bogen er en samling af oplæg til en konference om prototyping i Namur, oktober 1984. Oplæggene er grupperet i fire temaer:

"Prototyping: Concepts, Tools, Experiences"

"Prototyping and Database Design"

"Systems Development and Prototyping"

"Very High Level Languages for Prototyping"

**Cerveny m.fl.**

Robert P. Cerveny, Edward J. Garrity, G. Lawrence Sanders:

**The Application of Prototyping to Systems Development: A Rationale and Model**

Journal of Management Information Systems/FALL 1986, Vol. III, No. 2

**Christensen m.fl.**

Søren Christensen, Kaj Grønbæk, Tove Rolskov:

**Arbejdsformer under anvendelse af 4. generationsværktøjer**

Specialrapport, DAIMIIR-69, Århus 1987

Rapporten bygger på interview med systemudviklere fra ni systemudviklingsprojekter, fordelt på fem danske udviklingsorganisationer. Alle projekter har anvendt 4. generationsværktøjer. Resultaterne fra undersøgelsen præsenteres fordelt på fire hovedspørgsmål: Hvorledes indfris de store forventninger til 4. generationsværktøjernes betydning for udførelsesaktiviteterne?

Hvordan foregår samspillet mellem udførelsesaktiviteterne, når der anvendes 4. generationsværktøjer?

Hvordan anvendes 4. generationsværktøjerne i samarbejdet med brugere?

Hvordan påvirker anvendelsen af 4. gene-

rationsværktøjer og eksperimentelle teknikker behovene for projektledelse?

Hovedkonklusionen er, at ibrugtagning af nye udviklingsværktøjer ikke i sig selv løser nogle problemer. Hvis forventningerne til værktøjerne skal indfris, kræver det omlægning af arbejdsformerne. Undervejs præsenteres forslag til nye arbejdsformer.

**Ehn**

Pelle Ehn:

**Work-Oriented Design of Computer Artifacts**

Arbetslivscentrum, Stockholm 1988

**Floyd**

Christiane Floyd:

**A Systematic Look at Prototyping**

i Budde m.fl.

Denne artikel står som indledning i Budde m.fl. og er et forsøg på - ud fra oplæggene til konferencen - at opstille en fælles begrebsramme for diskussionen. Artiklen giver derfor et overblik over forskellige tilgange til prototyping. Floyd ser ikke prototyping som en metode for systemudvikling, men som en teknik, der skal kombineres med andre. Floyd ser prototyping som primært rettet mod udvikling af "interaktive applikationssystemer til støtte for menneskeligt arbejde". Floyd opstiller tre forskellige tilgange til prototyping, som også anvendes i denne bog: "exploratory/experimental/evolutionary prototyping".

**Floyd m.fl.**

Christiane Floyd, Fanny-Michaela Reisin, Gerhard Schmidt:

**STEPS to Software Development with Users**

Proceedings of ESEC 1989, Warwick, England

**Janson**

Marius A. Janson:

**Applying a Pilot System and Prototyping Approach to Systems Development and Implementation**

Information & Management 10 (1986)

**Jenkins**

A. Milton Jenkins:

**Prototyping: A Methodology for the Design and Development of Application Systems**

Discussion paper #227, Indiana University, 1983

**Jørgensen**

Anker Helms Jørgensen:  
**On the Psychology of Prototyping**  
 i Budde m.fl.

**Kauber**

Peter G. Kauber:  
**Prototyping: Not a Method but a Philosophy**  
 Journal of Systems Management, September 1985

**Lantz**

Kenneth E. Lantz:  
**The Prototyping Methodology**  
 Prentice-Hall, New Jersey 1986  
 Lantz præsenterer bogen som et alternativ til traditionelle systemudviklingsmetoder. Han opstiller en metode, baseret på versionsudvikling, hvor "prototypen bliver det endelige system". Lantz skelner sådanne prototyper fra "mock-ups", beregnet til udforskende eksperimenter. Metoden er bygget op om en model med syv trin, hvor der itereres mellem de tre centrale: "Define/Build/Exercise Prototype". Lantz er meget detaljeret i sine anvisninger i de forskellige trin, men giver ingen retningslinjer for, hvordan metoden tilpasses det konkrete projekt, eller hvordan der kan afviges fra det foreskrevne forløb.

**Mathiassen & Stage**

Lars Mathiassen, Jan Stage:  
**The Principle of Limited Reduction**  
 Indlæg til 12. IRIS-konference, Skagen 1989

**Mogensen**

Klavs Viby Mogensen:  
**Afprøvning af systemudvikling med prototyper**  
 Specialerapport, DIKU rapport nr. 85/10, København 1985  
 Rapporten er baseret på studie af litteratur, suppleret med to måneders

praktisk udvikling af prototyper for en mindre virksomhed. Mogensen gennemgår baggrunden for at anvende prototyper i systemudvikling og diskuterer begreber og opsamler erfaringer med prototypeteknikken, herunder dens begrænsninger. Derefter gennemgås det praktiske udviklingsprojekt og erfaringerne herfra. I rapportens konklusion gør Mogensen status over hidtidige erfaringer med prototypeudvikling og peger på muligheder for fremtidig anvendelse og forbedring af teknikken.

**Molich**

Peter Beyer, Peter Carstensen, Anker Helms Jørgensen, Rolf Molich (red.), Finn Hindkjær Pedersen:

**Brugervenlige edb-systemer**

Teknisk Forlag, København 1987

I bogen opstilles fem gyldne regler for brugervenligt design:

Kend brugerne

Inddrag brugerne

Koordiner hele brugergrænsefladen

Mål brugervenligheden løbende

Design iterativt

Bogen handler om design af brugergrænseflader. De meget konkrete og detaljerede anvisninger på afprøvning af brugergrænseflader rummer gode pointer om tilrettelæggelse og afvikling af eksperimenter i al almindelighed.

**Reisin & Wegge**

Fanny-Michaela Reisin, Daniela Wegge:  
**On Experimental Prototyping in User-Oriented System Development**  
 Indlæg til 12. IRIS-konference, Skagen 1989