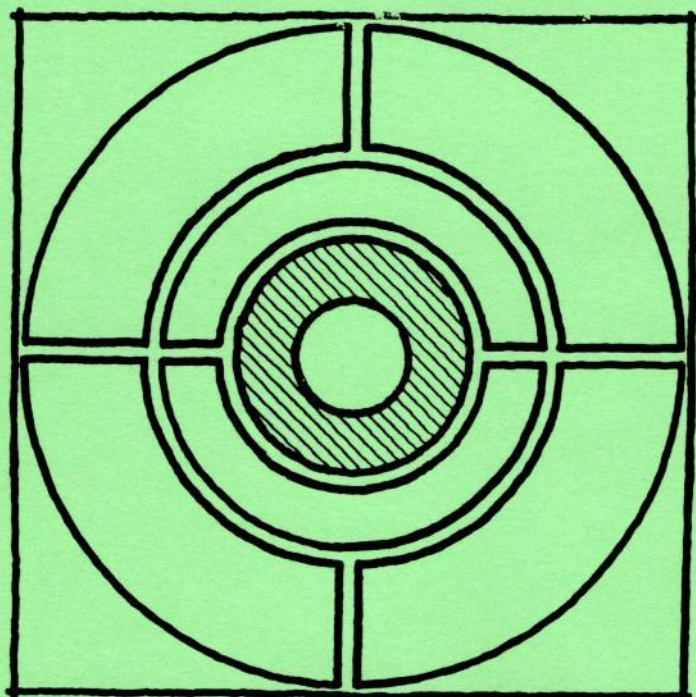




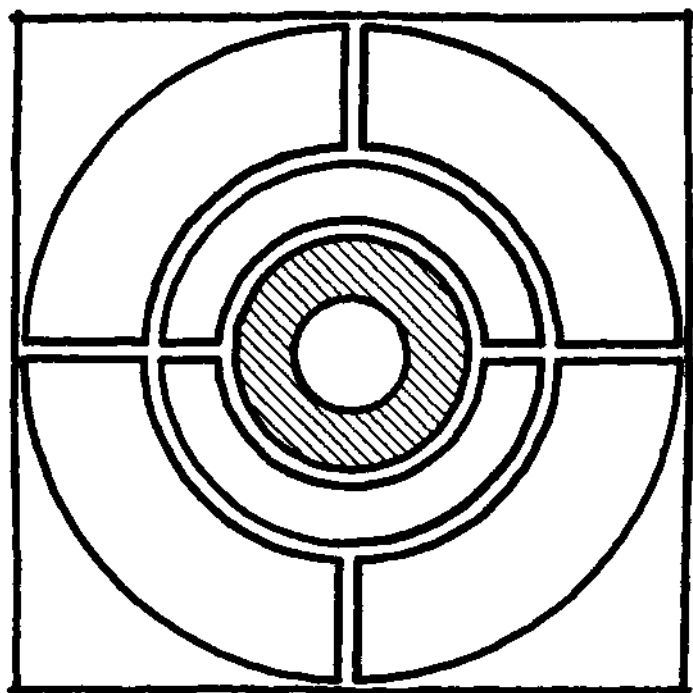
Metodesektionen



PLANLÆGNINGS VEJLEDNING



Metodesektionen



PLANLÆGNINGS VEJLEDNING

Planlægningsvejledning

© I/S Datacentralen af 1959

Første udgave dec. 1987

Anden udgave feb. 1988

udgiver og vedligeholder:

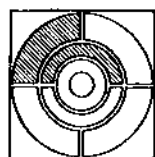
Metodesektionen i Teknikstaben

fordelingsnøgle: **MHPLV**

INDHOLDSFORTEGNELSE

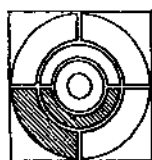
Indledning

Formål	4
Afgrænsning og referencer til anden dokumentation	5
Planlægning af systemudviklingsprojekter på Datacentralen	6



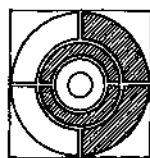
Styringsmetoder

Systemudviklingsforløb	8
Aktivitetsplanlægning	14
Tidsstyring	16
Ressourcestyring	19
Økonomistyring	21
Kvalitetsstyring	23
Projektorganisation	27



Udførelsesmetoder

Teknologivurdering	29
Soft Systems Analysis *	
Information Resource Management *	
Prototyping	31
Struktureret Analyse	34
Dataanalyse	36
Dialogudformning	39
Struktureret design	41
Jackson Struktureret Programmering	44
Databasedesign	47
Test	50



Dokumentation

Styringsdokumentation	54
Systemdokumentation	55

Værktøjer

Projektstartseminar *

Case2000 *

SuperProject + *

Sårbarhedsanalyse SBA *

* Under udarbejdelse.

INDLEDNING

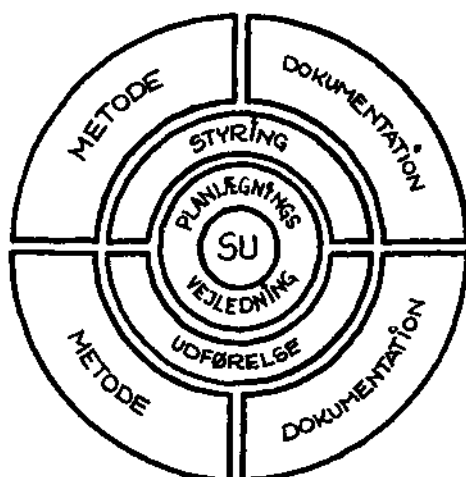
Formål

Denne vejledning er udarbejdet for at støtte planlægning af systemudviklingsprojekter.

Vejledningen henvender sig primært til projektledere, men også andre deltagere i projektorganisationer, såsom styregruppemedlemmer og projektdeltagere, skal have kendskab til indholdet.

Indholdet i planlægningsvejledningen er en gennemgang af Datacentralens Systemudviklingskoncept med påpejning af hvilke overvejelser, der bør foretages inden anvendelse af de enkelte dele i konceptet.

Indholdet er inddelt i afsnit efter Systemudviklingskonceptet



og således grupperet i

- Styringsmetoder
- Styringsdokumentation
- Udførelsesmetoder
- Udførelsesdokumentation (Systemdokumentation).

Hertil kommer et afsnit om værktøjer, der kan anvendes til understøttelse af de metoder m.v. projektet vil anvende, med beskrivelse af, hvilke fordele og ulemper, der er ved de enkelte værktøjer.

Afgrænsning og referencer til anden dokumentation

Denne planlægningsvejledning indeholder en oversigt over, hvad man kan gøre for at få et projekt til at lykkes. Planlægningsvejledningen indeholder således ikke beskrivelser af hvordan man gør.

Afsnittet om styringsmetoder indeholder en kort gennemgang af hvad styring omfatter. En mere detaljeret beskrivelse findes i Projektstyringshåndbogen.

Afsnittet om udførelsesmetoder indeholder korte vurderinger af udviklingsmetoder. Beskrivelse af metoderne kan findes via henvisninger til brochurer, artikler, bøger m.v. der er samlet og løbende bliver ajourført i brochuren "Systemudviklingskoncept" (fordelingsnøgle i DC's fordelingsssystem: MHSUK).

Planlægning af systemudviklingsprojekter på Data-centralen

DC er en stor virksomhed med mange typer af projekter, og derfor kan og skal der ikke gives faste retningslinier for det enkelte projekt. Derimod kan der fra DC's mangfoldige erfaringer udledes nogle generelle principper for gennemførelse af systemudviklingsprojekter.

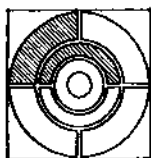
Disse principper er udmøntet i Systemudviklingskonceptet. Dette koncept giver en oversigt over, hvad systemudvikling kan bestå i omkring styring og udførelse af et projekt, samt hvilke metoder og dokumentations elementer, der erfaringsmæssigt er hensigtsmæssige at anvende.

Det enkelte projekt må fastlægge sin konkrete udformning gennem valg ud fra disse generelle muligheder, på baggrund af projektets rammer.

Fremgangsmåden for dette kan være følgende:

- gennemfør interessentanalyse og situationsanalyse
- planlæg projektets hovedforløb
- gennemfør aktivitetsnedbrydning
- klarlæg projektets rammer i form af tidsplan, ressourceplan, økonomiplan og kvalitetsplan
- fastlæg projektorganisation
- undersøg mulige metoder og dokumentation

- vælg metoder og dokumentation, og beskriv valgene samt overvejelserne bag
- overvej ideel anvendelse af valgte metoder og dokumentation
- bestem og beskriv konkret anvendelse af metoder og dokumentation i den aktuelle projektramme
- få ledelsens og kundens og evt. projektdeltageres accept på projektets udformning, valgte metoder og dokumentation.



STYRINGSMETODER

Dette afsnit beskriver anvendelsesmuligheder for metoder til projektstyring. Metoderne er nævnt i den rækkefølge, projektet forventes at anvende dem.

Systemudviklingsforløb

At planlægge et systemudviklingsforløb er at planlægge rækkefølgen af systemudviklingsaktiviteterne. Med systemudviklingsaktiviteter menes såvel det arbejde, der skal udføres som resultatet heraf i form af ydelser og produkter.

Et projekts systemudviklingsforløb skal være præget af bevidste overvejelser og valg og ikke af fornemmelser, vaner og tilfældige omstændigheder.

En gennemtænkt planlægning af systemudviklingsforløbet er en forudsætning for at projektet kan gennemføres med succes.

Planlægning af systemudviklingsforløb afhænger af opgaven, kundeforhold, ressourcer og af mange andre forhold.

Herefter følger tre skitser af forskellige eksempler på systemudviklingsforløb, som kan anvendes som inspiration. Eksemplerne er skitserede som netværksdiagrammer, hvor kasserne angiver aktiviteter og pilene angiver afhængighed mellem aktiviteter.

Diagrammerne er opdelt i to dele:

i venstre side vises aktiviteter, der udføres af DC og i højre side vises aktiviteter, der udføres af kunden.

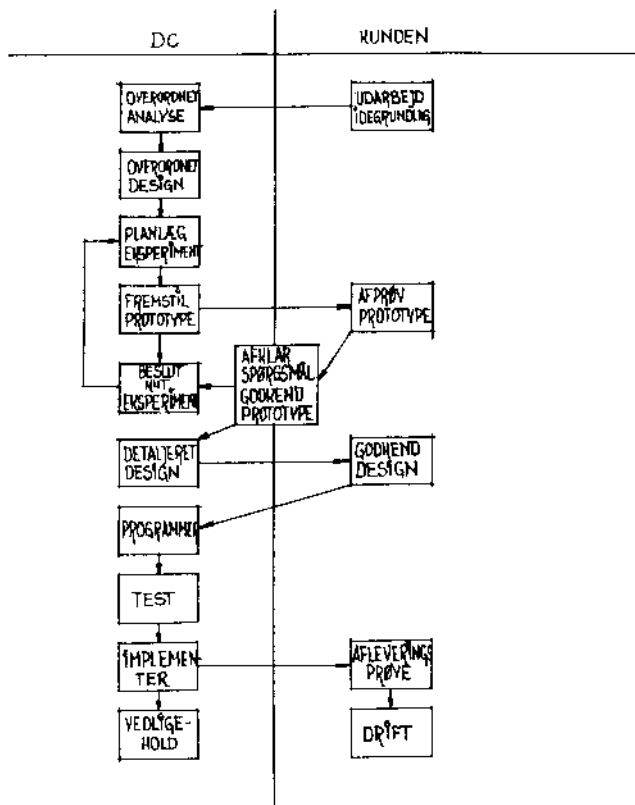
Prototyping

Første eksempel er et projekt, hvor kunden beder om anvendelse af prototyping til afklaring af brugergrænsefladen. Prototypen skal være en skærmdialog med data, der kan gemmes.

Forløbet starter med en overordnet indledende analyse efterfulgt af en skitse eller overordnet design af det nye system.

Dernæst følger en iterativ brug af prototyping, der har karakter af at være eksperimenterende.

Når prototypen og dermed grundlaget for systemet er på plads, fortsættes som ved anden systemudvikling.



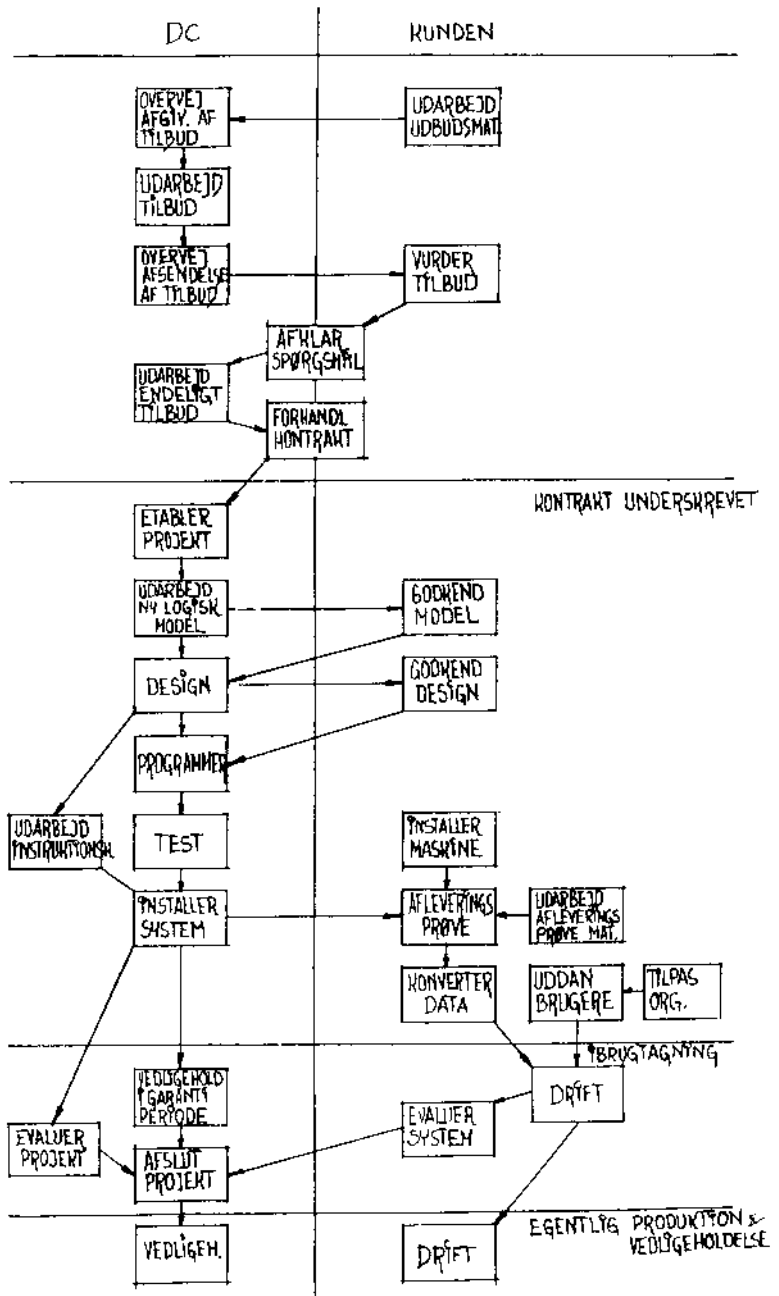
Fast tid/fast pris

I andet eksempel er forløbet initieret af et udbud vedr. et system incl. hardware til installering lokalt hos kunden.

Forløbet falder i to dele, hvor første del omfatter tilbudsfasen. I denne fase skal projektet rådføre sig med Råd og Tilsynsfunktionen i Teknikstaben som anvist i vejledning for tilbudsgivning.

Anden fase af forløbet foregår som anden systemudvikling, men med det forhold, at pris og leveringstidspunkt samt systemegenskaber m.v. er beskrevet i kontrakten og er svære at ændre både for brugere og for DC.

Fordelen ved dette forhold er at man har præcise mål at styre efter. Til gengæld kan målet blive at opfylde kontrakten i stedet for at opfylde kundens behov.



Versionsudvikling

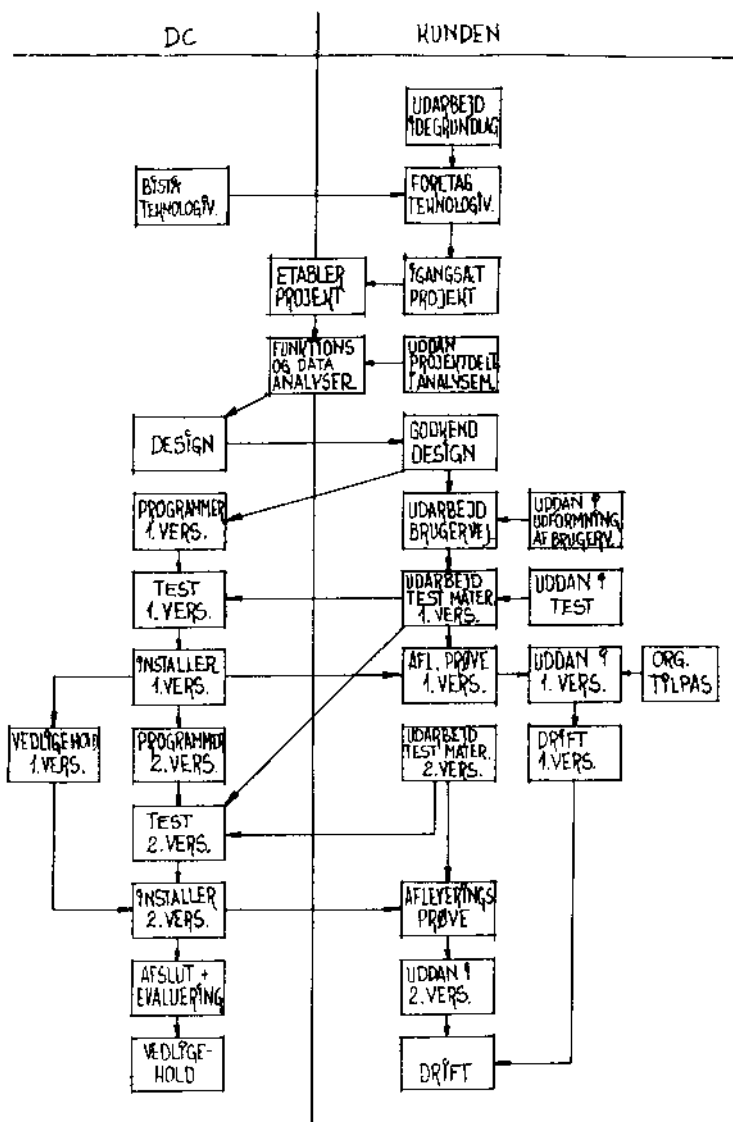
Tredie eksempel er et projekt, hvor kunden og DC samarbejder om at udvikle et system, der skal køre på DC's centrale anlæg.

Systemet skal leveres i to omgange, således at en del af funktionaliteten først leveres i 2. version. Grunden til dette er, at kunderne har presserende behov for funktionaliteten i 1. version, og ønsker gradvis indføring af teknologi i organisationen.

I projektet deltager brugerne aktivt i analyse og test, og det giver stor sandsynlighed for at systemet kommer til at svare til brugernes forventninger. Til gengæld er projektet afhængig af brugere med egnede profiler, og at de reelt har tid til at deltage.

IRM

Projekter, der beskæftiger sig med Information Resource Management kan henvende sig til Metodesektionen for vejledning i udførelse af systemudviklingsforløb.



Planlægning af et projekts udviklingsforløb består som sagt ikke i at vælge et af ovennævnte forløb, men i at planlægge det specifikke forløb, der er relevant for netop dette projekt.

Aktivitetsplanlægning

Aktivitetsplanlægning består i en top-down nedbrydning af projektopgaven karakteriseret ved et skitseret systemudviklingsforløb og idegrundlaget for projektet.

Projektopgaven skal nedbrydes til aktiviteter, dvs målelige dele af den samlede opgave. For hver aktivitet beskrives anvendelse af metode/værktøj, startgrundlag, resultat (afslutningskriterier), afhængighed af andre aktiviteter, omfang (mandtid), ressourcekategori m.v.

Beskrivelsen af omfang volder problemer, her skal anvendes estimering. Estimeringsteknikken vælges afhængig af tidspunktet og det tilgængelige materiale. I starten af et projekt bruges generelle erfaringer, ved senere detailplanlægning og planrevision bruges projektets hidtidige forløb.

Estimeringsteknikker er baseret på systemelementer og projektelementer, som kvantificeres ud fra erfaring. Det kan være egne erfaringer eller andres fra lignende opgaver.

Opsamling af en erfaringsbase for grupper med fælles præg i projekter er mulig og overkommelig, hvis der er enighed om standard for system- og projektelementer.

Ved angivelse af omfang er det praktisk at anvende den enhed, man senere vil bruge ved opfølgning (f.eks. timer).

Aktiviteters omfang på overordnet niveau beregnes som summen af de underliggende aktiviteter omfang. Erfaringsmæssigt bør der tages højde for uforudsete aktiviteter ved tillæg af ekstra enheder.

For at opnå effektiv styring, skal aktiviteter nedbrydes således, at en aktivitet maksimalt omfatter 3-6 mand-uger.

Detaljeret aktivitetsnedbrydning kræver grundigt kendskab til opgaven og lægger op til stærk tids- og ressourcestyring.

Summarisk aktivitetsnedbrydning og -beskrivelse med mange usikkerheder lægger op til dynamisk styring med løbende planlægning.

Ofte vil detaljeringsgraden være forskellig fra aktivitet til aktivitet, idet de nærmest forestående aktiviteter er mest detaljeret beskrevet.

Man må acceptere, at nogle aktiviteter vil være summarisk beskrevet og estimeret ved gæt, fordi man endnu ikke ved nok om opgaven. Til gengæld må man sørge for at ajourføre aktivitetsplanen løbende i projektet, så den viden, man har, altid er afspejlet i planen.

I aktivitetsplanen indgår både styringsaktiviteter og udførselsaktiviteter.

Henvisning

Eksempler på aktiviteter er beskrevet i Projektstyringshåndbogen.

SuperProject + kan anvendes til beskrivelse af aktiviteter og deres afhængigheder.

Tidsstyring

Tidsstyring er planlægning og opfølgning på kalender-tid.

Ved tidsplanlægning forstås placering af aktiviteter i en tidsplan, hvis udstrækning er afhængig af opgavens omfang og af projektkrav så som ressourcer, økonomi, deadlines m.v.

I tidsplanen skal tages højde for afhængigheder mellem aktiviteter.

Hvordan findes projektets udstrækning, hvis der ikke er ydre tidskrav?

For enhver opgave er der en grænse for, hvor hurtigt opgaven kan løses. Uanset hvor mange ressourcer man tilfører projektet, kan denne grænse ikke overskrides.

Undersøgelser i USA (Barry Boehm) af projekter inden for forsvaret med rigelige ressourcer har vist, at normal tidsudstrækning for et projekt med krav om hurtig færdiggørelse kan beregnes efter følgende formel:

$$\text{normal kalendertid} = 2.5 \sqrt[3]{\text{samlet mandtid}}$$

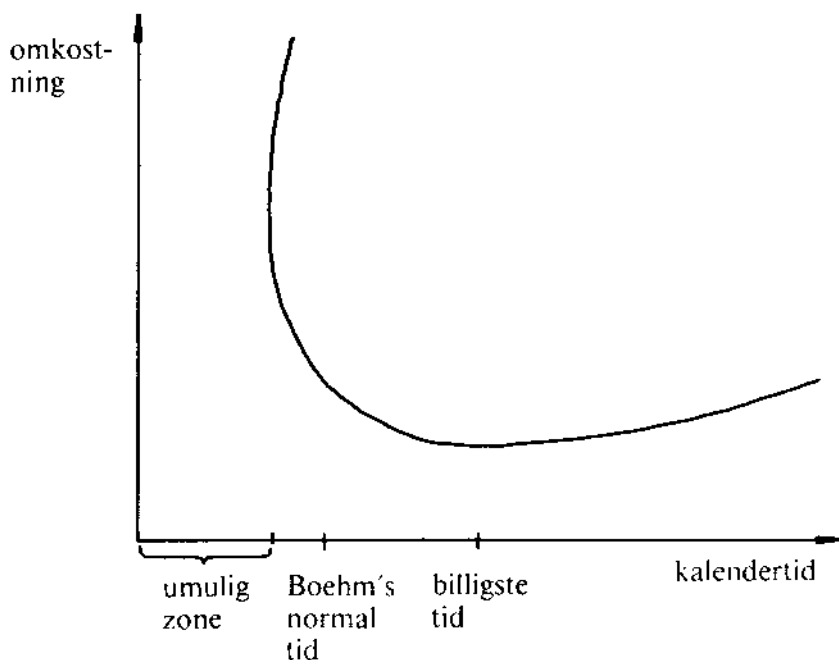
dvs. hvis et projekts omfang er 27 mandmåneder, er normal gennemførelsestid

$$2.5 \sqrt[3]{27} = 2.5 * 3 = 7.5 \text{ kalendermåneder}$$

dvs. gennemsnitlig bemanning 3-4 personer.

Barry Boehm har desuden erfaret, at det er umuligt at gennemføre et projekt på under 3/4 af normal kalender-tid, i eksemplet under 3/4 af 7.5 = 4.7 kalendermåneder.

Hvis et projekt skal gennemføres forceret, stiger omkostningerne. Sammenhængen mellem et projekts samlede omkostninger og gennemførelsestid kan ses af følgende kurve:



En almindelig reaktion på problemet med at overholde tidsplanen, er at sætte flere personer på projektet. Dette har i praksis ofte vist sig at give modsat resultat, - jævnfør kurven, der svinger tilbage ved høje omkostninger.

Overarbejde virker på samme måde, dvs. forringer projektets økonomi, i hvert fald i det omfang overarbejdet ikke viderefaktureres.

En anden fejlagtig antagelse er at stramme tidsplaner bevirker at projektet går fremad med maximal hastighed. Men et projekt under pres laver flere fejl og vil gå langsommere frem med større omkostninger end et projekt med realistiske tidsplaner.

Når man har fastlagt projektets udstrækning, skal aktiviteterne lægges i en tidsplan. Her skal tages højde for tid til kunde- og ledelsesbeslutninger, som ofte tager længere tid end man umiddelbart skulle tro. Desuden skal man huske ferieperioder, luft til sygdom, uddannelse, aktiviteter udenfor projektet m.v.

I tidsplanen afsættes statuspunkter (milepæle, reference-linier o.lign.), hvor aktiviteterne forventes at have nået et veldefineret resultat. På disse tidspunkter kan man så følge op på, om planerne overholdes, yderligere detaljere og evt. korrigere fremtidige planer.

Afstanden mellem statuspunkter afhænger desuden af interessenternes behov for information og rapportering samt vigtigheden af at overholde tidsplanen.

Tidsplaner kan opdeles i et antal niveauer og kan maksimalt have samme detaljeringsgrad som aktivitetsplanen.

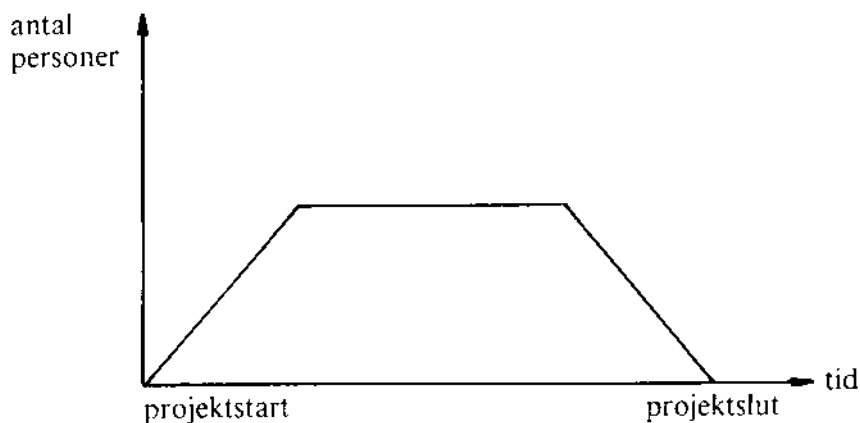
SuperProject + kan gengive tidsplaner i form af Gantt-diagrammer og netværksdiagrammer.

Ressourcestyring

Ressourcestyring er planlægning af hvilke ressourcer, der skal bruges i projektet og opfølgning på hvor mange ressourcer, der er anvendt. Ved edb-projekter drejer det sig hovedsagelig om personer og mandtimer.

Med udgangspunkt i aktivitetsplanens beskrivelse af ressourcekategori (personkvalifikationer) og tidsplanens beskrivelse af aktiviteterne tidsmæssige placering bemannes projektet.

En tommelfingerregel siger, at projektets bemanning over tid skal være trapetzformet:



Det tager lang tid for en person at komme ind i et projekt, og af den grund er det en fordel at beholde samme personer i projektet så længe som muligt. For hver ny person skal afsættes tid til indslusning.

Til gengæld kan det også være en fordel at have faglige specialister i projektet i kortere tid for at tilføre projektet erfaringer eller udføre specielle aktiviteter.

Hvis det ikke er muligt at skaffe personer med de nødvendige kvalifikationer til projektet, må tidsplanen ændres ved at indføje tid til uddannelse, eksperimenter og erfaringsudveksling.

I det hele taget er tidsplaner og ressourceplaner uløseligt knyttet sammen, og under planlægning kan man tage udgangspunkt i enten den ene eller den anden, og så justere efterhånden som planlægningen skrider frem.

Ressourceplaner kan beskrives i niveauer som tidsplaner, men af det mest detaljerede niveau skal det fremgå, hvilken aktivitet, den enkelte person skal beskæftige sig med, hvornår og hvormeget.

I ressourceopfølgning er det ikke kun interessant at vide hvor stort ressourceforbruget har været, men også hvor meget der mangler i at aktiviteten er færdiggjort. Det er dog vanskeligt at vurdere varigheden af resten af aktiviteten.

Resultater fra opfølgning på ressourceforbrug er værdifulde oplysninger i den løbende planlægning og til estimering af fremtidige projekter.

SuperProject+ kan vise belastning både for den enkelte ressource og for projektet som helhed.

Økonomistyring.

Økonomistyringen af et projekt er en integreret del af projektstyringen.

Formålet med økonomistyringen er at gennemføre projektet med færrest mulige omkostninger samt at sikre, at projektets budget overholdes.

Det forudsættes, at der inden projektets start er udarbejdet et omkostningsbudget på aktivitetsniveau for hele projektet.

Der er ikke nødvendigvis sammenhæng mellem det enkelte projekts salgspris og omkostningsbudgettet, idet der kan være tale om en større eller mindre fortjeneste (= projektets salgspris - omkostninger). Fortjenesten kan endda være negativ.

Men det er klart, at der for DC's projekter som helhed skal være en balance mellem indtægter og registrerede omkostninger.

Projektets samlede omkostninger omfatter interne og eksterne omkostninger.

Interne omkostninger består af:

- Forbrug af mandtimer omregnet til kroner efter aftalt sats.
- Anvendelse af maskinressourcer, TSO-ressourcer eller andel i anskaffelse og drift af decentralt udstyr.
- Øvrige interne omkostninger, der kan direkte relateres til projektet.

Eksterne omkostninger består af:

- Køb af software/hardware
- Ekstern konsulentbistand
- Øvrige eksterne omkostninger, der kan direkte relateres til projektet.

De faktisk registrerede omkostninger sammenholdes løbende med projektets budgetterede omkostninger for at konstatere budgetoverskridelser eller budgetforbedringer.

Projektlederen skal endvidere til enhver tid i hele projektføreløbet kende projektets forventede restbudget.

Herved sikres en løbende opfølgning og rapportering af projektets totaløkonomi.

Det bemærkes, at projektets indtægter (salgspris, a'conto fakturering, indtægter fra salg af software, hardware m.v.) ikke er omfattet af økonomistyringen, selvom det ofte vil være projektlederens ansvar at sikre, at den aftalte fakturering sker.

Kvalitetsstyring

Kvalitet opnås, når en leveret ydelse opfylder kundens forventninger.

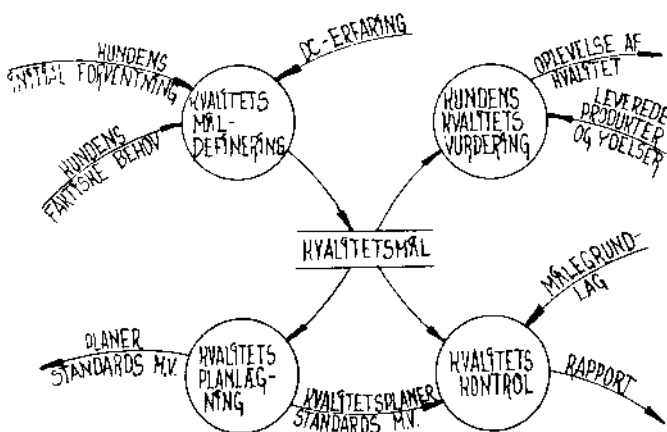
Styring af kvalitet er en forudsætning for at opnå kvalitet, og derfor er kvalitetsstyring den fjerde parameter i projektstyring.

Hvor de tre andre, tidsstyring, ressourcestyring og økonomistyring er almindeligt anvendte, er kvalitetsstyring først blevet udbredt i de senere år.

Udgangspunktet for kvalitetsstyring er, at man har defineret kvalitetsmål. Kvalitetsstyring består herefter i kvalitetsplanlægning og kvalitetskontrol.

Ved kvalitetsmåledefinerings afstemmes kundens initialforventninger med kundens faktiske behov og DC-erfaringer, og der opstilles kvalitetsmål, som kunden skal godkende. Dette skal foregå på salgstidspunktet.

De opstillede kvalitetsmål skal være et udtryk for kundens faktiske forventninger og det grundlag, som kunden vil vurdere de leverede ydelser og produkter på.



Ud fra kvalitetsmålene foretages kvalitetsstyringen. Først kvalitetsplanlægning, dvs. planlægning af, hvorledes den ønskede kvalitet skal indbygges i produkter og opnås i projektet.

På baggrund af de udarbejdede planer, standards m.v. samt kvalitetsmålene udvikles systemet, gives ydelser m.v.

Under systemudviklingsforløbet fremkommer nogle forskellige resultater f.eks. i form af modeller og udgaver af systemet, som bruges som målegrundlag for kvalitetskontrol, hvor målegrundlag sammenholdes med kvalitetsmål. Resultatet dokumenteres i kvalitetsrapporter, rewieurapporter eller statusrapporter.

Alle projekter skal foretage kvalitetsstyring, dvs. planlægge og kontrollere kvalitet, på lige fod med tidsstyring, ressourcestyring og økonomistyring.

Resultatet af kvalitetsstyring er kontrol over system- og projektkvalitet.

Fordele

Kvalitetsstyring giver mulighed for tættere samarbejde mellem kunde og leverandør, mellem projekter og internt i projekter. Dette betyder, at man kan få glæde af andres viden og erfaring, og dermed højere kvalitet i arbejdet.

Kvalitetsstyring flytter kvalitetsmålinger så langt frem i projektføreløbet som muligt, således at afleveringsfasen ikke præges af mange fejlrettelser og deraf voldsomt tidspres.

Fejl og mangler opdages i den fase, de er opstået i, hvilket reducerer omkostningen ved fejlrettelse betydeligt.

Antallet af fejlrettelser efter aflevering reduceres og forventningerne til systemet er afstemt mellem leverandør og kunde.

Kvalitetsstyring medvirker til overholdelse af aftalte tidsterminer ved at fremtvinge klarere statusrapportering og projektstyring.

Forudsætninger

Det er vigtigt, at kvalitetsmål defineres af eller i samarbejde med kunden.

Ledelsesopbakning og et samarbejdspræget projektmiljø er en forudsætning for kvalitet.

Der skal være afsat tid og ressourcer i projektplanen til kvalitetsstyringsaktiviteter.

Det kan være en god ide at anvende personer udenfor projektet som deltagere i reviews.

Den person, der står for kvalitetsstyringen på projektet, bør have grundig kendskab til kvalitetsstyring.

Faldgruber

Måldefinering er en vanskelig sag, som man kan være tilbøjelig til at gå let hen over, men

"You can't control, what you can't measure"
(Tom DeMarco)

Fundamentet for kvalitetsstyring er prioriterede, entydige og målbare egenskaber.

En traditionel fejl er at slække på kvalitetsstyringsaktiviteter p.g.a. tidspres. Det bevirker, at kvaliteten ikke er under kontrol, hvilket ofte giver store omkostninger på længere sigt i form af større vedligeholdelsesudgifter, dårligt renome o.s.v.

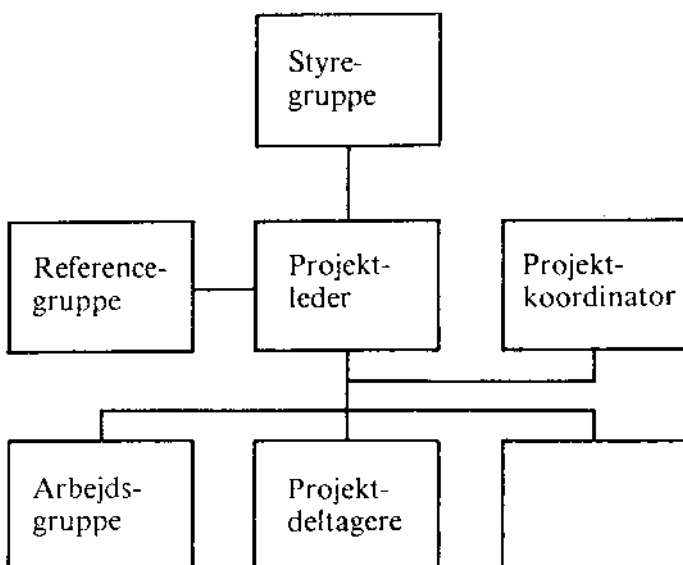
Kvalitetskontrol kan opleves som en belastning i dårligt fungerende grupper. Tvangsindføring af kvalitetskontrol i et anspændt miljø er ingen god ide. Det er vigtigt, at projektdeltagerne tager aktivt del i udformningen af kvalitetsstyringen.

Henvisning Emnet kvalitet behandles i Kvalitetshåndbogen (fordelingsnøgle: MIKVA).

Projektorganisation

Umiddelbart virker det ret enkelt at udarbejde projektorganisation.

Den velkendte model



anvendes som udgangspunkt, tilpasses og funktionerne bemandes.

Men i virkeligheden er projektorganisering et ømtåleligt emne, der berører personlig prestige, samarbejdsrelationer, kompetence m.v.

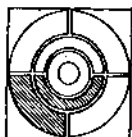
Det er vigtigt, at finde frem til en projektorganisation, der kan fungere i praksis, og som interessenterne kan gå ind for.

Forudsætninger herfor er:

- At der tages stilling til, hvorledes kunden og kundens organisation indgår i projektet
- At basisorganisations og projektorganisations indbyrdes relationer afklares
- At styregruppen sammensættes hensigtsmæssigt, dvs.
 - * at den ikke er for stor
 - * at den overvejende består af mellemledere eller ledere på højere niveau
 - * at formanden er projektejer, dvs. den chef, der har størst interesse i at projektet lykkes
- At der til organisationsdiagrammet knyttes beskrivelser af arbejdsopgaver, kompetance og ansvar
- At evt. referencegruppes rolle afklares
- At koordineringsfunktioner fastlægges og placeres.

Ud over den formelt fastlagte organisation kan der være uformelle informationsstrømme, som man også må være opmærksom på at anvende.

Projektorganisationen er ikke nødvendigvis den samme i hele projektets levetid.



UDFØRELSESMETODER

Ved aktivitetsplanlægningen blev nævnt, at der til hver aktivitet kan knyttes en metode.

I dette afsnit beskrives anvendelsesmuligheder for metoder, der knytter sig til udførelse af systemudvikling.

Teknologivurdering

Teknologivurdering er en samlet vurdering af konsekvenserne af at indføre ny teknik i en organisation. Det er en vurdering af konsekvenserne på alle områder, dvs.: for brugergrænsefladen, for økonomien, for arbejdsfunktionerne, for jobindholdet, for de organisatoriske relationer (internt og eksternt), for arbejdsmiljøet (fysisk, socialt og psykisk) samt for servicen.

En sådan vurdering bør foretages flere gange i løbet af systemudviklingen. Vurderingen skal primært laves af de kommende brugere, samt af de der iøvrigt vil blive berørt af systemet. De forskellige brugergrupper har forskelligt syn på det nye system og er opmærksomme på forskellige aspekter og konsekvenser. Derfor bør alle grupper inddrages i udviklingen. For at kunne gennemføre teknologivurderingen må brugerne støttes af en konsulent med erfaring i, hvilke konsekvenser edb-systemer har haft andre og eventuelt lignende steder.

For brugerne er formålet med teknologivurderingen:

- at få et bedre beslutningsgrundlag
- at kravene til systemet præciseres og diskuteres
- at ønskede konsekvenser fremmes og opnås
- at uønskede konsekvenser undgås

For systemudviklerne er formålet med teknologivurderingen:

- en grundig analyse af brugerorganisationen
- at kravene til systemet præciseres og diskuteres

De forskellige brugergruppers interesser betyder, at der ofte opstilles modstridende krav og ønsker om konsekvenser. Dette betyder, at konsulentens rolle er vanskelig og at konsulenten nemt kan blive upopulær hos mindst en af brugergrupperne.

En forudsætning for at en teknologivurdering kan gennemføres, er at brugeren har eller tager sig tid til at udføre vurderingen. I teknologivurderingen er det en god ide at inddrage dem der er mest skeptiske over for edb.

I SD415 "Principper for teknologivurdering" er der opremset mange mulige konsekvenser af indførelsen af et nyt edb-system. Disse konsekvenser er opremset som spørgsmål, således at de umiddelbart er brugbare til at igangsætte en teknologivurdering med brugerne.

Prototyping.

Prototyping er en fremgangsmåde, der består i at udvikle tidlige prøveversioner - prototyper - af produktet og derudfra vurdere og justere de forskellige forslag til udformning af systemet. I forbindelse med udviklingen af edb-systemer benævnes fremgangsmåden også eksperimentel systemudvikling, systemudvikling med prototyper eller interaktiv systemudvikling.

Prototyping erstatter ikke grundig analyse og design, men bidrager til øget viden om krav, ønsker og konsekvenser.

Prototyping er velegnet til ustrukturerede opgaver, dynamiske miljøer, eksperimentelle situationer, udformning af skærmdialoger og til modning af institutioner for edb.

Resultat

Forskellige aspekter af et nyt system kan afklares ved brug af prototyper: brugergrænseflader, systemets funktioner og tekniske krav. Prototyping kan praktiseres på mange forskellige måder. Forskellene kan karakteriseres ved hensigten med, grundlaget for, udformningen og afprøvningen af prototyperne samt måden prototyperne indgår i systemudviklingsforløbet på.

Af forskellige eksempler på prototyper kan nævnes:

- Attrapper af skærmbilleder.
- En serie af skærmbilleder i en dialogstruktur uden data.
- En skærmdialog, hvor data kan gemmes og vises.
- Et brugerudviklet system på en PC.
- Et standardsystem.

Fordele

Det er svært at danne sig et billede af det færdige edb-system ud fra beskrivelser på papir. Brugerens mulighed for at forstå og tage stilling øges, hvis der fremstilles en automatiseret model af systemet i form af en prototype.

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

Prototyping forbedrer produktiviteten i systemudviklingen, idet der anvendes værktøjer, der hurtigere giver synlige resultater, således at systemudviklernes overblik og motivation øges.

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.

Forudsætninger

Prototyper er gode til at afklare krav og ønsker til detaljeret design af enkeltfunktioner, men det er nødvendigt at supplere prototyperne med overordnet analyse og design af funktioner og data.

Prototyping forudsætter en repræsentativ brugerdeltagelse, hvor deltagerne har tid til at være med.

Prototyping stiller nye krav til den enkelte systemudvikler. Alle får kundekontakt og skal derfor have den nødvendige viden samt de nødvendige kvalifikationer også af psykologisk og politisk art.

For at sikre effektiv udnyttelse af prototyping er der brug for en veltilrettelagt og dokumenteret fremgangs-

måde, hvor følgende spørgsmål er afklaret:

- Hvad ønsker vi at opnå med prototyping ?
- Hvad skal grundlaget være 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 ?

Faldgruber

Prototyping er svært at styre, og har tendens til at løbe løbsk. Opstil derfor kriterier og procedurer for vurdering.

Det er vanskeligt på forhånd at vurdere projektets samlede tids- og ressourceforbrug ved prototyping. Det skal der tages hensyn til i kontrakten.

Prototyping kan sætte fokus på værktøjer til udvikling af prototyper i stedet for på design af edb-systemer.

Prototyper kan skabe forventninger hos brugerne om, at udviklingstiden bliver 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 f.eks. med hensyn til fejlhåndtering, back up 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. Gør derfor brugeren opmærksom på de begrænsninger og mangler der er i prototypen i forhold til produktionssystemet.

Henvisning

Der henvises til skrift 'Brug Af Prototyper' (fordelingsnøgle: MHPT).

Struktureret Analyse

Struktureret Analyse (SA) er en metode beregnet til at udvikle en analysemodel af et system. Metoden resulterer i en funktionsnedbrydning, der udmøntes i en grafisk model af systemet.

Alle systemudviklingsprojekter skal anvende Struktureret Analyse, med mindre der er vægtige grunde for at vælge alternative analysemetoder.

Metoden giver en samlet beskrivelse af det nye systems maskinelle og manuelle funktioner, og dermed et beslutningsgrundlag for brugeren. Beskrivelsesværktøjerne er datastrømsdiagram, dataordbog samt procesbeskrivelser.

Fordele

Metodens værktøjer til beskrivelse af systemet sikrer fuldstændighed.

Metoden er velkendt, og ved anvendelse af Struktureret Analyse opnår projektet, at brugeres og nye systemmedarbejderes indsigt i systemet lettes.

Struktureret Analyse er velegnet både ved udvikling af nye systemer og ved ændring af eksisterende.

Metoden støtter at systemets kompleksitet nedbrydes, således at fejl og mangler lettere afsløres og analysearbejdets kvalitet forbedres.

Forudsætninger

Brug af SA forudsætter et vidtgående samarbejde med bruger i form af interviews og diskussioner omkring brugers krav og mål, samt feedback på modeloplægget.

Struktureret Analyse skal suppleres med dataanalyse.

Det må anbefales, at man, for projektarbejde med SA påbegyndes, deltager i kursus om metoden.

Værktøjer

Case2000 integrerer tekst og grafik. Derudover er der diverse analyse-, design-, kontrol-, og valideringsfunktioner, der understøtter SA.

Ved analysearbejde kan man bruge forskellige teknikker, f.eks. interview-teknik, brugergruppemøder evt. med væggraffer, spørgeskemaundersøgelser m.f.

Faldgruber

Man skal ikke fokusere så meget på metoden, at man hver gang beskriver nuværende fysisk system og nuværende logisk system. Det er ikke alle opgaver der kræver det.

Man skal sørge for altid at have de essentielle krav for øje.

Supplerende metoder

En metode, der er udviklet på baggrund af Struktureret Analyse, er 'Essential Systems Analysis', som ved hjælp af en blitzingteknik skaber et hurtigt overblik over systemet opbygning.

Dataanalyse skal anvendes til at fastlægge systemets data og deres sammenhænge.

Dataanalyse

Dataanalyse er en metode til at analysere informationsgrundlaget for et virksomhedsområde. Resultatet er en logisk model af disse data, som bl.a. anvendes ved databasedesign.

Det må anbefales altid at lave en datamodel. Minimumskravet er at datastrukturerne normaliseres.

Dataanalyse gennemført i fuldt omfang giver:

- en datamodel for virksomhedsområdet,
- normaliserede datastrukturer,
- oversigt over livsforløb for data og
- accessbeskrivelser i tilknytning til datas repræsentation for bruger.

Fordele

Datamodellen giver en dybere forståelse af virksomhedsområdet, og den kan afdække uoverensstemmelser og uklarheder i politikker og krav.

Sammen med oversigten over livsforløb for data giver datamodellen mulighed for kontrol af Struktureret Analyse: er alle funktioner til vedligeholdelse, oprettelse og nedlæggelse af forekomster i databasen definerede?

Ved at bygge datamodellen over et virksomhedsområde i stedet for at begrænse sig til et systems dataelementer får man et bedre grundlag for opbygning af fælles databaser. Dette kan være et led i en overordnet styring af informationsressourcerne, hvor datamodellen går ud fra og integreres i virksomhedens informationsmodel.

Datamodellen kan dog også afgrænses til datagrundlaget for det aktuelle system.

Ved at lave accessbeskrivelser sikres de nødvendige oplysninger til konstruktion og optimering af databasen.

Forudsætninger

Omfang og placering af dataanalysen afhænger af kundens indstilling. Jo mere omfattende og generelt virksomhedsområdet er, jo større indsats kræves af kunden og brugerne.

Dataanalysen skal altid suppleres med en analyse af de krævede funktioner i et system. Dataanalysen kan udføres før, sideløbende med eller efter funktionsanalysen.

Sideløbende analyse af funktioner og data er det optimale; men det kræver erfaring og sikkerhed i alle analysemetoderne. At koncentrere sig om een analyse ad gangen og så afstemme resultaterne vil være nemmere.

Dataanalysen kan have forskellige udgangspunkter, der også kan kombineres. Det kan være

- virksomhedens informationsmodel,
- en funktionsmodels beskrivelse af data-elementer, datastrømme osv.,
- helt fra bunden med interviews, studier af eksisterende systemer, blanketter, skrifter om virksomhedsområdet m.m. med omfattende brugermedvirken.

Accessbeskrivelsen har til formål at give de nødvendige oplysninger for optimering af datas tilgængelighed ved databasedesign. De basale mængdeoplysninger for dataforekomster, og hvor tit data skal accesses, vil altid være nødvendige. Oversigter over access kan koncentreres om kritiske svartider og mængder.

Da accessbeskrivelsen går på tilgang til data, kan det være hensigtsmæssigt at tage afklaring af sikkerhedskravene op i tilknytning hertil.

Værktøjer Case2000 kan benyttes til dokumentation af både grafiske og tekstmæssige dele.

Datadictionary-systemer kan benyttes til dokumentation, definition af komponenter i datamodel og datastrukturer.

Faldgruber En uklar afgrænsning af, hvad analysen omfatter, kan resultere i, at der arbejdes på kundens informationsmodel, og ikke på en datamodel for et virksomhedsområde.

Sammenhængen med funktionsanalyse kan indebære, at de samme ting laves to gange. Men det kan også føre til, at noget ikke bliver gjort, fordi alle regner med, at det beskrives i den anden model. Projektet må lave sin egen standard for, 'hvem' der gør hvad, og hvordan der dokumenteres til fælles brug.

Datamodellen skal give et logisk billede af virksomhedsområdet data, så man skal passe på ikke at blande argumenter om implementation ind i udarbejdelsen.

Supplerende metoder Struktureret analyse til analyse af funktionerne er velegnet til koordinering med dataanalyse.

Dialogudformning

Dette afsnit handler alene om brugervenlighed og brugerdeltagelse som forudsætning for god dialogudformning.

Vi definerer et brugervenligt system som et system der er:

- let at lære
- let at bruge
- let at huske
- robust mod fejl
- behageligt at bruge

Der findes ingen skudsikre forskrifter for, hvordan et system skal se ud for at opfylde disse krav. Der er dog udviklet en hel del retningslinier for, hvordan systemer skal være udformet for at opfylde disse krav, fx retningslinier for menuudformning, bogstavsstørrelser, skærmbilledlayout osv.

Generelle retningslinier vil dog aldrig kunne beskrive, hvordan det specifikke system skal se ud for at passe til og være brugervenligt overfor den bestemte brugergruppe, det er lavet til. Det er derfor nødvendigt at bruge følgende 3 principper/metoder i systemudviklingen:

- tidligt fokus på brugerne inden systemdesignet foretages. Man må i direkte kontakt med brugerene for at undersøge og forstå deres særlige måde at tænke og arbejde på

- brugerne skal deltage i systemudviklingen. Lige fra begyndelsen bør brugerne være en del af udviklingsgruppen.
- gentagne afprøvninger af brugergrænsefladen og korrektion som følge heraf undervejs i systemudviklingen og gerne så tidligt som muligt. Brugergrænseflader kan afprøves ved papirmodel eller med prototyper. Afprøvningsmetoden kan være tænk-højt test.

Kun ved at bruge disse udviklingsprincipper har man mulighed for at sikre at systemet bliver brugbart for brugerne.

Struktureret Design

Struktureret design (SDe) er en metode til udarbejdelse af en hierarkisk modulopdeling af et system udfra en logisk beskrivelse af systemet.

Alle systemudviklingsprojekter bør anvende SDe i en eller anden veldefineret grad, med mindre der er dokumenteret vægtige grunde til at lade være.

Resultatet af metoden er en komplet beskrivelse af det nye fysiske system opdelt i moduler med klart definerede grænseflader. Der kan herefter kodes efter beskrivelsen.

Fordele

Metoden indeholder regler, som sigter mod en modulopdeling, der er vedligeholdelsesvenlig, testvenlig, udbygningsvenlig m.v.

Metoden indeholder standarder for beskrivelse, der sikrer, at systemdokumentation udformes ensartet og entydigt.

Metoden er velkendt hos mange kunder og kræves stadig hyppigere anvendt i udbud. Anvendelse af metoden vil ofte kunne medvirke til at give projektet et positivt image.

Forudsætninger

Struktureret design beskæftiger sig med et systems funktioner, og skal suppleres med metoder til dataanalyse og databasedesign.

Udgangspunktet for SDe er en beskrivelse af systemet i form af datastrømsdiagrammer med dataordbog og procesbeskrivelser. Det må anbefales, at hovedstrukturen i form af modulhierarki og data er fastlagt, inden versionsudvikling (hvor en del af systemet konstrueres, af testes og installeres ad gangen) påbegyndes.

Det er nødvendigt, at projektet opstiller egne standarder for design, dokumentation, anvendelse af metoden m.v., der er afpasset efter kvalitetsmålene for den specifikke opgave, som supplement til de regler metoden anbefaler.

Struktureret design er en metode for cdb-folk.

Et godt design er ikke noget man kan ryste ud af ærmet eller læse sig til. Et projekt bør mindst råde over en person, der tidligere har anvendt metoden i praksis.

Hvis projektdeltagerne er usikre på metodens anvendelse i det konkrete projekt, bør eksperter i form af erfarne kolleger, metodeafdelingen eller eksterne konsulenter bistå projektet ved seminarer, reviews o.lign.

Værktøjer Case2000 kan anvendes til dokumentation af designet.

Faldgruber Ved anvendelse af metoder er der generelt det problem, at man kan komme til at fokusere på at anvende metoden i stedet for at løse opgaven. Sagt med andre ord: Ingen metode kan udforme et design. Man skal tænke selv.

Metoden støtter ikke opbygning og anvendelse af databaser. Man skal derfor huske at foretage dataanalyse og databasedesign sideløbende.

Metoden omfatter ikke alle designaktiviteter.

Supplerende metoder

Der kan være overlap til Jackson Struktureret Programmering, men JSP kan også supplere SDe ved at strukturere komplekse moduler.

Dialogudformning er en selvstændig disciplin, som har indflydelse på modulstrukturen.

Som nævnt skal dataanalyse og databasedesign anvendes til at fastlægge databasen.

Jackson Struktureret Programmering

Jackson Struktureret Programmering (JSP) er en metode, hvormed man designer programmer, som bliver modeller af de datastrømme der skal behandles.

Datastrømmene tegnes med strukturdiagrammer, der bagefter kombineres til en (program)struktur. Den påføres de operationer (beregninger, flytning af data osv.), som programmet skal udføre. Herfra kan programkode kodes i 3. eller 4. generationssprog.

JSP har vist sig at være en stabil metode, den er stort set forblevet uændret siden '75, selvom edb-branchen er gået fra batchkørsler med hulkort til systemer med on-line-dialoger.

Fordele

Datastrømmene tegnes som strukturer, der angiver den tidsmæssige rækkefølge, hvori data produceres eller anvendes. Strukturdiagrammer kan bruges til at beskrive mange ting, for eksempel serielle filer, input fra en skærm, en serie af databasekald og "transaktioner" i en tele-applikation.

JSP er en sammenhængende metode, der er bredt anvendelig og ikke skal ikke suppleres - eller tilpasses i hvert enkelt tilfælde.

Det har vist sig, at JSP giver vedligeholdelsesvenlige programmer, blandt andet fordi programkoden er velstruktureret forståelig.

JSP er fleksibel, fordi der er meget små krav til detaljeringniveau. Man tegner ikke diagrammerne for deres egen skyld.

Diagrammerne skal tegnes efter præcise regler, og det letter kommunikationen med andre, at der ikke er fortolkningsmuligheder.

JSP-løsninger kan kodes i et hvilket som helst af de sprog, der anvendes på DC.

På DC findes flere erfaringer med JSP, end med nogen anden metode. Mest omfattende er formentlig CPR-systemet, hvor 600 programmer er designet med JSP. Der findes flere forskellige rapporter fra projekter, der har anvendt JSP.

Forudsætninger

JSP er beregnet på administrative systemer, og baserer sig på tidsmæssige/hierarkiske strukturer. Hvis de ikke findes, er JSP næppe til megen hjælp.

Værktøjer

På IBM-mainframes findes et sæt MetaCOBOL-makroer, der helt dækker de grundstrukturer og specialteknikker, der anvendes i JSP.

Faldgruber

Man starter med at beskrive datastrømmenes struktur, det er den mest kreative fase, men også den mest krævende. Man kan løbe sur i det, hvis man ikke kan få hjælp, når der er brug for det. Det er vigtigt, at projektet på forhånd gør sig klart hvordan uerfarne i JSP kan blive hjulpet.

Overladt til sig selv risikerer man at bruge diagrammerne som flowcharts, hvad man ikke altid slipper godt fra. Man kan ikke altid "bare gøre et eller andet" i JSP.

Pæne tegninger er naturligvis en dyd - i dokumentationen. Mens det er en fristelse at koncentrere sig om at tegne pænt, hvor man skulle have brugt energien på at finde det rigtige design.

Medtager man bevidstløst alle datastrømme - eller hvis man går i unødvendige detaljer, kan man ende med endnu mindre overblik, end man startede med.

JSP er en god hjælp, men en dårlig krykke.

Databasedesign

Databasedesign er en aktivitet, der resulterer i en definition af datastrukturer tilpasset et udvalgt databasesystem. Dens udgangspunkt er en logisk beskrivelse af informations- indholdet i et system samt mængde-, frekvens- og accessoplysninger. Databasesystemet kan være et færdigt databasestyresystem eller filer under et operativsystem.

Databasedesign skal altid foretages i en eller anden form ved nyudvikling.

Fremgangsmåden afhænger fuldstændig af det valgte databasesystem. Leverandører af databasesystemer angiver til tider metoder for deres system.

Resultatet af databasedesignet er en definition af databasen i de termer, databasesystemet betjener sig af. Termerne kan f.eks. være recordtyper, tabeller, pointere, accessmetoder.

Databasedesign skal sikre optimal tilgængelighed af data og minimal redundans. Dette kan indebære modstridende interesser, som skal afvejes mod hinanden.

Forudsætninger

Det kræver medarbejdere med nøje kendskab til det valgte databasesystems konventioner, stærke og svage sider, for at lave et godt design med afvejning af forskellige hensyn.

Baggrundsmaterialet i form af datamodel og accessbeskrivelser skal give et korrekt billede af sammenhænge i data, være normaliseret og vise de kritiske accesser.

Databasedesign skal ligge tidligt blandt designaktiviteter, da det påvirker udformningen af funktioner, der skaffer og kombinerer data.

I tilknytning til databasedesign er det hensigtsmæssigt at behandle følgende emner:

Konvertering.

Der skal laves en plan for, hvorledes der til start kommer data i systemet. Specielle programmer til udtræk fra det eksisterende system og indlægning i det nye (evt. med indtastning) kan være nødvendige.

Back-up og recovery.

Hvordan, hvor meget og hvor tit skal der tages back-up? Hvordan undgås tab af transaktioner ved recovery? Benyttes et databasestyringsystem, vil dette oftest indeholde faciliteter, der understøtter disse opgaver.

Sikkerhed.

Hvem har adgang til hvad? Hvilken kontrol skal udføres? Også her vil databasestyringsystemet indeholde faciliteter.

Værktøj

Datadictionary-systemer kan benyttes til dokumentation. Nogle kan danne definitioner til brug i programmer.

Databasestyringsystemer kan indeholde definitionsværktøj. Det kan være integreret med datadictionary.

Faldgruber

Hvis der udelukkende fokuseres på optimering af access i systemet, bliver nye eller ændrede funktioner vanskelige at honorere; herunder muligheder for ad-hoc forespørgsler. Selv små forskydninger i datagrundlag og sammenhænge mellem data kræver omfattende omstrukturering.

Accesstider er ofte kritiske ved svartidskrav til online-systemer.

Test

Test er en aktivitet, der udføres for at finde fejl og mangler ved et produkt.

En test forbedrer ikke et produkts kvalitet; en efterfølgende fejlrettelse er en forudsætning for dette.

Fordele

Ved test konstateres, om et produkt opfylder sine kvalitetsmål, og hvis ikke hvordan man kan forbedre kvaliteten, eller hvordan man må ændre målene.

Forudsætninger

Man kan ikke teste til alle fejl er fundet, eller til alt er afestet. Det er vigtigt, at tage beslutninger om, hvor man vil lægge sin testindsats, og i denne forbindelse fastlægge:

- testorganisation
- økonomi
- tid
- ressourcer
- testkvalitetsmål
- testform og -område
- testmetode
- standard for testdokumentation

Afhængigt af hvilke typer af fejl, man leder efter, og tidspunkt i systemudviklingsforløbet kan der anvendes forskellige former for test.

Forskellige testformer er:

- gennemgang af dokumentation (f.eks. systembeskrivelse, program)
- modultest
- integrationstest (flere moduler)
- funktionstest (funktionalitet, dialog)
- systemtest (performance, sikkerhed, kapacitet)
- afleveringsprøve (foretages af kunden)
- installationstest

Værktøjer

Der findes flere forskellige metoder til test, heraf kan nævnes

- Structured Walk Through (SWT)
velegnet til gennemgang af dokumenter
- Tænke højt test
velegnet til test af dialog og funktionalitet
- Maskinel test
velegnet til modultest, integrationstest, systemtest m.fl.

Faldgruber

Testmaterialet må ikke være mere omfattende, end man har tid til at gennemgå resultatet. Det forekommer ofte, at testmaterialet dokumenterer fejl, som alligevel bliver overset, fordi testresultatet ikke gennemgås omhyggeligt nok.

Man må ikke teste egne produkter. Ekspertter mener endda, at man bør adskille testere og systemudviklere organisatorisk, og at man kun bør beskæftige sig med en af delene.

Testmaterialet skal indeholde forventet resultat til sammenholdelse med det faktiske testresultat.

Det er aldrig tilstrækkeligt at anvende produktionsmateriale til test.

Det er vigtigt at registrere alle fejl. Dels for at sikre at de bliver rettet, og dels for at kunne overvåge systemets og systemudviklingens kvalitet.

Det er vigtigt at man selv retter sine fejl, ellers lærer man ikke af dem.

Mange tror, at fejl er naturlige, men man skal bestræbe sig på at lave fejlfrie produkter. Ved test findes erfaringsmæssigt max 38% af fejlene, ifølge Tom DeMarco.



DOKUMENTATION

Projektets dokumentation kan deles i to dele: styringsdokumentation og systemdokumentation.

Før disse to dele beskrives særskilt, kan der gives nogle generelle retningslinier for dokumentation.

Det er vigtigt allerede fra starten at fastlægge principper for hvilken dokumentation, der som minimum bør udformes og vedligeholdes i forbindelse med systemarbejdet. Man kan skelne mellem dokumentation, der er nødvendig under udviklingen og vedligeholdelsesdokumentation.

Der skal fastlægges et ambitionsniveau, der kan overholdes inden for de afsatte projektrammer. I hele projektføreløbet skal man være opmærksom på at ambitionsniveauet holdes.

Hvis ambitionsniveauet for dokumentationen sættes for højt i starten, kan det medføre, at der ikke bliver lavet det minimum af dokumentation, som systemet kræver i vedligeholdelsessituationen. Fastlæg derfor minimumsstandarder.

Der skal være afsat tid og ressourcer i projektplanen til dokumentationsarbejdet.

Det er vigtigt altid at få en eller anden form for feedback på sin dokumentation.

At dokumentere er ikke nødvendigvis at sætte ord på papir, dokumentation kan også være en grafisk afbild-

ning, som mange gange vil være lettere at opfatte og diskutere end flere sider prosa om samme emne.

Uensartet og unøjagtig dokumentation er ikke til nogen hjælp.



Styringsdokumentation

Formålet med styringsdokumentation er at dokumentere arbejdsprocessen i projektet, bl.a. de aftaler, der er indgået mellem projektet og projektets interessenter og i projektet internt.

Styringsdokumentationen omfatter desuden rapporter, referater m.v., der udarbejdes i forbindelse med projektstyringen.

Dokumentationen samles i en projekthåndbog med følgende hovedafsnit:

- indledning
- idegrundlag
- mål
- opgaver/aktiviteter
- tids-, ressource-, økonomi- og kvalitetsplaner
- projektorganisation
- rapportering og information til omgivelserne
- metoder og værktøjer
- eksternt arkiv
- internt arkiv

Der skal i starten og løbende i et projektføreløb tages stilling til, hvilket konkret indhold de enkelte afsnit skal have.

Der bør være regler for udformning af mødereferater, rapporter, notater, forskellige typer af breve for at sikre mod misforståelser og fejl.

Der skal tages stilling til modtagerkredsen for de enkelte dele af styringsdokumentationen.

Under rapportering og information er det en god ide med en projektlogbog, hvori vigtige hændelser i projektet noteres. Projektlogbogen kan bl.a. være et godt grundlag for udarbejdelse af projektevalueringsrapporten.



Systemdokumentation

Formålet med systemdokumentation er at dokumentere produktet, dvs. systemet som resultat af forskellige faser af systemudviklingen.

Systemudvikling kan opfattes som en gradvis konkretisering af indledende forestillinger om et system til et kørende edb-system, hvor der under processen udarbejdes forskellige modeller eller beskrivelser af systemet.

På DC findes standarder for indhold af disse beskrivelser opdelt i følgende dokumentationsenheder:

- opgaveformulering
- systembeskrivelse
- teknikspeficikation
- systemspeficikation
- testspeficikation
- instruktionshåndbog.

Standarden indeholder ikke faste krav for udformning og omfang af systemdokumentation. På det enkelte projekt må man udforme egne standarder for dokumentation, men det må anbefales at tage udgangspunkt i DC's standard.

Større projekter kan med fordel lade en lille gruppe arbejde forud på et delområde og lave forslag til standarder, før alle skal i gang.

Ved at udforme egne retningslinier for omfang af systemdokumentation kan man undgå unødigt papirarbejde.

Ved udformning af projektets dokumentationsstandarder skal man tage stilling til:

- Hvilke dokumenter skal betragtes som arbejdspapirer, som kun har midlertidig anvendelse, og hvilke dokumenter skal udgøre den blivende dokumentation, som løbende vedligeholdes?
- Hvilke dele af et dokument kan genbruges i andre dokumenter? (f.eks. dataordbog)
- Hvilken redundans er nødvendig?
- Hvilke dokumenter udgør et kontraktgrundlag, og hvilke krav stilles af den grund til disse?
- Hvilke dokumenter skal godkendes af kunden?
- Hvem er modtagere af dokumentet, og hvilke krav stiller de pågældende til, f.eks.
 - detaljering
 - ensartethed

- læsbarhed
- aktualitet
- distribution

afhængig af, hvad dokumentet skal bruges til.

- Hvordan skal konfigurationsstyring af beskrivelser og programmer foretages?
(Ved konfigurationsstyring forstås styring af den uendelige række af ændringer, der overgår systemet og modeller af systemet i udviklingsprocessen og i vedligeholdelsesfasen)
- Hvem er ansvarlige for de enkelte dokumenters korrekthed?
- Hvilke edb-baserede værktøjer kan medvirke til at effektivisere papirarbejdet, herunder vedligeholdelse?
- Hvilke standarder for brug af edb-værktøjer skal man have?
- Hvordan undgås afskrivning, fra et dokument til et andet?
Afskrivning medfører fejl og tager tid!
- Hvordan undgås redundans, dvs. samme information beskrevet flere steder?
- Hvordan anvendes review til at sikre dokumentationskvaliteten?

Faldgruber

Konfigurationsstyring er en ofte undervurderet aktivitet, som bør opprioriteres og gerne understøttes af edb-værktøjer for at undgå misforståelser og fejlliverancer.

DataCentralen