

Over succesvol begroten van software projecten

Drs. P. Rowold

Iedereen kent wel een verhaal over een automatiseringsproject dat te laat werd opgeleverd, met kostenoverschrijdingen die tot 300% opliepen en dat niet die prestaties levert die er in het begin aan werden toegedicht. Is het inherent aan automatiseringsprojecten of horen we alleen maar de negatieve verhalen? Na publikaties over het Walradar project, de wet op de studiefinanciering, en de Gemeentelijke Basis Administraties zou de gedachte kunnen ontstaan dat het vrijwel onmogelijk is om automatiseringsprojecten op tijd en binnen het gestelde budget op te leveren. Waar zijn de succesverhalen?

Het begroten van automatiseringsprojecten levert, zoals ook uit een recente enquête¹ blijkt, veel problemen op. Daarbij blijkt dat slechts een klein aantal (22%) van de schriftelijk geënquêteerden het de moeite waard vindt om op de enquête te reageren. Is de belangstelling voor het begroten van automatiseringsprojecten dan zo gering of is het probleem zo moeilijk dat reageren op de enquête onmogelijk was?

Het ontwikkelen van informatiesystemen is (te) lang een moeilijk grijpbare problematiek geweest. Maar de jaren dat automatiseren achter gesloten deuren plaatsvond en omringd was door een waas van geheimzinnigheid lijken voorbij.

Bij gesprekken over het schatten van de omvang van automatiseringsprojecten blijkt dat begrippen als begroten, plannen en budgetteren nog al eens door elkaar worden gehaald en verkeerd worden gebruikt. Het is daarom goed om even stil te staan bij deze begrippen.

Begroten of schatten is een middel ter voorberei-

ding van een zo doelmatig mogelijke uitvoering van de betrokken (project)werkzaamheden.

Budgetteren en *plannen* hebben veel meer een taakstellend en/of autoriserend karakter. Budgetteren gebruiken we dan meestal voor een taakstelling in geld, terwijl plannen een taakstelling in de tijd kan inhouden.

Beide begrippen hebben betrekking op automatiseringsprojecten, die een heel scala kunnen bestrijken van de aanschaf van één personal computer tot het ontwikkelen en installeren van een groot informatiesysteem met honderden aansluitingen.

Voor zover hier sprake is van projecten bedoelen men die groep automatiseringsprojecten waarbij de inzet van personeel, dat software (programma's) ontwikkelt, meer dan substantieel is. Niet dat alle andere automatiseringsprojecten niet begroot of gebudgetteerd moeten worden, maar het lijkt alsof projecten waarbij (veel) mensen betrokken zijn, de tendens hebben forse overschrijdingen op de leveren. Een project kan gedefinieerd worden als: een verzameling werkzaamheden die dient tot de realisatie van een vooraf gespecificeerd doel; namelijk, het op tijd en tegen de geraamde kosten opleveren van een, door de opdrachtgever gewenst, informatiesysteem.

Het zichtbare gedeelte van een automatiseringsproject bestaat meestal uit een grote hoeveelheid papier. Daarnaast zijn er veel ongrijpbare zaken zoals programma's, gegevensbestanden, apparatuur, enzovoort. Dit maakt een automatiseringsproject voor velen weinig inzichtelijk.

Drs. P. Rowold studeerde economie, afstudeerrichting bestuurlijke informatiekunde aan de Universiteit van Amsterdam. Thans werkzaam als organisatie-adviseur bij Deloitte, Van Dien, Van der Torn + Buningh.

Al heel lang hebben mensen, vooral in Amerika, getracht om inzicht te krijgen in de vraag wat nu de omvang van een automatiseringsproject bepaalt.

Globaal kunnen er twee stromingen voor de bepaling van de omvang worden onderkend. De ene stroming gaat uit van het feit dat de opgeleverde programmaregels het enige meetbare eindprodukt van een automatiseringsproject zijn. In deze categorie treft men namen aan van Boehm, Putnam, en Jensen met respectievelijk de COCOMO-, SLIM- en JS/2-methode. De tot deze categorie behorende methoden worden omschreven als op programmaregels gebaseerde methode. De andere stroming gaat uit van de (gebruikers-)functies die het opgeleverde informatiesysteem zal vervullen en schat de omvang aan de hand van het aantal functies en de zwaarte van de functie. De methoden van Albrecht met FPA en van DeMarco met 'design weights' kunnen tot een op functies gebaseerde methode worden gerekend.

De afgelopen jaren is een convergentie opgetreden van beide stromingen en een aantal auteurs of ondernemingen heeft methoden ontwikkeld die met de term 'hybride' kunnen worden aangeduid. Deze auteurs maken gebruik van de uitgangspunten van beide stromingen en brengen hun methode meestal, ondersteund door programmatuur, op de markt. De namen die hier bij horen zijn: Rubin met het produkt ESTIMACS en Gordon Group met het produkt 'Before you Leap'. Met de komst van hybride methoden en de bijgeleverde computerprogramma's treedt er een volgend groot probleem op: er kan nu wel een schatting gemaakt worden, maar hoe goed is die schatting?

Wat is een goede schatting?

Conte definieert de criteria van een goede schatting als volgt:

- 1 75% van de voorspellingen vertonen een afwijking van minder dan 25% van de werkelijke uitkomst;
- 2 de gemiddelde relatieve fout van de voorspellingen is kleiner dan 25%.

In zijn beschouwing van een aantal schattingsmethoden concludeert Conte dat geen van de door hem onderzochte schattingsmethoden aan deze criteria voldoet. Uit Conte's conclusie mag niet worden afgeleid dat goed schatten per definitie onmogelijk is. Uit een door de auteur uitgevoerd (beperkt) onderzoek² blijkt dat er bedrijven zijn die wel goed schatten. Afwijkingen van minder dan 10% op totale projecten en minder dan 2% op delen van projecten blijken tot de mogelijkheden te behoren.

Het is interessant om te kijken waarom deze bedrijven beter schatten dan andere bedrijven. Er is een aantal redenen aan te geven dat maakt dat bij deze bedrijven schatting en werkelijkheid niet ver uit elkaar liggen.

De vier belangrijkste redenen zijn:

- 1 Alle bedrijven hebben over een lange periode gegevens verzameld over hun projectverloop.
- 2 De geschatte onderdelen zijn in omvang beperkt, dat wil zeggen hele grote projecten worden in onderdelen geschat en uitgevoerd.
- 3 Er wordt tijdens het projectverloop een aantal malen geschat.
- 4 Schatten wordt niet alleen door de automatiseerders, maar ook door de bedrijfsleiding gezien als een bedrijfsbelang.

De eerste reden geeft al aan waarom schattingen vaak afwijken. Bij veel bedrijven is een automatiseringsproject geen alledaagse bezigheid. Dit maakt het verzamelen en vergelijken van de gegevens over het projectverloop moeilijk voor kleine bedrijven met een beperkte automatiseringsstaf. Wanneer we de andere drie redenen zien als voorwaarden voor een nauwkeurige schatting, kunnen alle bedrijven aan deze voorwaarden voldoen.

Goede en slechte schattingen

Hoe kan men een schatting beoordelen zonder de werkelijkheid te kennen? Is het mogelijk om aan de hand van criteria een schatting te beoordelen. Op grond van ervaringen van bedrijven, waarvan is gebleken dat ze goede schattingen maken, kunnen criteria worden ontwikkeld die een onderscheid tussen goede en slechte schattingen aan-

geven. Een aantal van deze criteria wordt hier toegelicht.

Fasering

Omdat het nauwkeurig schatten van een (nog) onbekend probleem onmogelijk is, moet tijdens het verloop van een project een aantal malen geschat worden. In het projectverloop moet hiervoor een aantal meetpunten worden vastgesteld. Het is daarom noodzakelijk dat een project in fasen wordt uitgevoerd. Ontwikkelingsmethoden zoals SDM, PROMPT, PARAET enzovoort kennen alle een fasering van de systeemontwikkeling. Deze faseringen gaan uit van groepen activiteiten die één of meer documenten of produkten opleveren.

Het gebruik van een fasering heeft 3 neveneffecten op de schattingen:

- 1 Naarmate meer fasen zijn afgerond, wordt het probleem steeds duidelijker en zal de schatting steeds nauwkeuriger worden.
- 2 Als aan het begin van elke fase opnieuw een schatting wordt gemaakt, zal de absolute nauwkeurigheid toenemen, dat wil zeggen de afwijkingen in tijd en geld verminderen omdat een steeds kleiner deel van het nog af te lopen traject wordt geschat.
- 3 Aan het einde van iedere fase wordt de achterstand op de geschatte voortgang vastgesteld. Tijdens de uitvoering van het verdere project kunnen dan al maatregelen worden genomen.

ad 1 De onnauwkeurigheid van een schatting aan het begin van een project wordt in de literatuur gesteld op $\pm 40\%$. Ook de praktijk bevestigt dit. Naarmate het ontwikkelings-traject verder wordt uitgevoerd, daalt de onnauwkeurigheid. Bij de oplevering van het project zal de onnauwkeurigheid immers nul zijn. Londeix hanteert de volgende onnauwkeurigheid bij de schatting aan het begin van iedere fase:

– subsystem definition	$\pm 40\%$
– module design	$\pm 40\%$
– code	$\pm 20\%$
– test	$\pm 10\%$
– na test	$\pm 5\%$

Deze Amerikaanse aanduidingen van fasen komen niet overeen met de in Nederland gangbare aanduidingen. Bij één van de bedrijven uit het vermelde onderzoek² dat SDM hanteert, komen de volgende cijfers naar voren:

Onnauwkeurigheid aan het begin van de fase,

– Definitiestudie	$\pm 20\%$
– Functioneel ontwerp	$\pm 10\%$
– Technisch ontwerp en programmeren en testen	$\pm 5\%$

Een aantal bedrijven geeft aan dat vóór de start van een project een onnauwkeurigheid van 50% niet ongewoon is.

ad 2 De verschillende fasen verdelen het totale ontwikkeltraject. Daar de inhoud van een fase per methode verschilt, is er ook een verschillende verdeling van de totale inspanning over die fasen. Voor twee (Nederlandse) methoden volgen hier de uit de praktijk verkregen gemiddelden:

SDM ³	
Definitiestudie	12%
Basisontwerp	8%
Detailontwerp	40%
Realisatie	20%
Invoering	30%
PROMPT ⁴	
Opstart	2%
Specificatie	15%
Ontwerp	10%
Ontwikkeling	43%
Invoering	27%

ad 3 Aan het einde van iedere fase kan evaluatie van de voortgang in de tijd en van de uitputting van het financiële budget plaatsvinden. Als voorbeeld: aan het einde van de definitiestudie zijn de grenzen van het probleem en het project bepaald. Dan kan blijken dat de omvang van het project groter is dan verwacht. Er kunnen direct maatregelen worden genomen om: of de omvang te verkleinen, of de schatting te verhogen. Dit werkt

helaas ook tegengesteld: als om andere redenen dan een te laag geschatte omvang de definitiestudie uitloopt, zal men meestal de volgende fasen (te) ruim inschatten. De produktiviteit binnen een project wordt dan ongunstig beïnvloed.

Als de eindprodukten van iedere fase gedefinieerd zijn, wordt het moeilijk om een eventueel opgelopen achterstand steeds naar een volgende fase te verschuiven. Zijn de fasen niet duidelijk gedefinieerd, dan cumuleert de achterstand tot aan het einde van een project en blijkt pas dan hoe groot de achterstand werkelijk is. Dit effect, dat soms het 'Walrus effect' genoemd wordt naar de uit de hand lopende bouw van onderzeeboten, wordt met een stringente afloopcontrole per fase vermeden.

Een limiet aan de doorlooptijd

Voorspellen is nu eenmaal moeilijk, aangezien het over de toekomst gaat. Naarmate die toekomst verder weg is, zijn de uitspraken daarover meer onzeker. Het is duidelijk dat de onnauwkeurigheid in de schatting van de doorlooptijd bij een grote projectomvang in absolute zin groter is dan bij een project van kleine omvang. Het is dan ook zinvol om de doorlooptijd van een project te beperken door opsplitsing van het project in deelprojecten.

Enkele auteurs (Gilb en Bemelmans 1987) bepleiten een evolutionaire strategie, waarbij de specificaties stukje voor stukje worden geformuleerd en het systeem steeds wordt uitgebreid. Voordelen zijn ondermeer het ontbreken van een bevriezende effect op de organisatie en de mogelijkheid om mee te evolveren met de veranderende informatiebehoefte. De evolutionaire strategie, ook wel de 'kleine stap'-strategie genoemd, beperkt tevens de doorlooptijd van de deelprojecten. Metzger geeft als voordelen voor deze beperking: grote projecten zijn moeilijk beheersbaar door de grote hoeveelheid communicatie en door fragmentatie van de werkzaamheden. Maar de belangrijkste reden is dat een afwijking van bijvoorbeeld 20% op de doorlooptijd bij projecten

met een korte doorlooptijd minder pijnlijk is dan bij projecten met een veel langere doorlooptijd. Binnen de onderzochte bedrijven ligt de maximale doorlooptijd tussen één en anderhalf jaar. Ook auteurs van schattingsmethoden beperken de doorlooptijd van een te schatten project. Maar de bovengrens ligt op ongeveer 5 jaar, en veel hoger dan bij de onderzochte² bedrijven. Dit is te verklaren uit de omgeving waarin de schattingsmethoden zijn ontwikkeld. Het betrof meestal hele grote automatiseringsprojecten in vliegtuigen geleide wapenindustrie.

Riesewijk en Warmendam signaleren dat de doorlooptijd van een project een succesfactor kan zijn. Bij projecten korter dan 1 jaar kan 72,2% van de projecten als een succes worden gekenmerkt. Bij projecten met een doorlooptijd tussen de 1-3 jaar is dat nog maar 40,0%.

De doorlooptijd is een factor die het slagen of falen van projecten kan bepalen. Bedraagt de schatting van de doorlooptijd van een automatiseringsproject die meer dan een jaar, dan moet het project zeer sceptisch worden bekeken.

Een specificatie van de kostensoorten

De totale kosten voor een project worden vaak uitgedrukt in mensmaanden. Deze mensmaanden kunnen volgens sommige auteurs en schatters eenvoudig naar geld worden omgerekend. Grote onduidelijkheid blijft bestaan welke kosten nu in de schatting zijn opgenomen. Zijn dat nu de ontwikkelingskosten, de projectkosten of de totale kosten?

Londeix onderscheidt twee verschillende kostencurven die hij respectievelijk aanduidt met de ontwikkelcurve en de projectcurve. Het oppervlak onder de ontwikkelcurve geeft de kosten van de inzet van analisten, ontwerpers, programmeurs en hun directe management weer. De ontwikkelaars werken binnen een bepaalde omgeving en worden ondersteund door mensen die indirect bij het ontwikkelproces betrokken zijn zoals: secretariaatsmedewerkers, programma- en systeembeheerders, specialisten ten aanzien van methoden en technieken enzovoort. Als deze kosten bij de ontwikkelcurve worden opgeteld, ontstaat de

projectcurve. Londeix signaleert dat bij kleine projecten het verschil tussen projectcurve en ontwikkelcurve klein is. Naarmate het project groter wordt, zal volgens Londeix de ontwikkelcurve een ander verloop vertonen en zal de projectcurve gaan afwijken van de ontwikkelcurve.

Naast ontwikkelings- en projectkosten worden er ook kosten gemaakt na de oplevering van het informatiesysteem. Putnam, maar ook Boehm en Myer geven aan dat het beslag gedurende de ontwikkelingstijd ongeveer 40% is van de totale inspanning over de gehele levensduur van een informatiesysteem. Globaal gesteld betekent dit dat 60% van de totale inspanning geleverd wordt *na* oplevering van een systeem.

Bij elke schatting moet worden aangegeven welke kosten er wel en niet in de schatting zijn opgenomen. Kosten van het gebruik van het systeem na de oplevering worden meestal niet in een schatting opgenomen, maar kunnen bij de evaluatie van twee alternatieven van doorslaggevende betekenis zijn.

Voor- en nacalculatie

Bedrijven, waarvan blijkt dat ze goed schatten, hebben vaak over een lange periode gegevens verzameld. Schattingsmethoden gaan er vanuit dat de uitgangspunten van een methode getoetst worden aan de werkelijkheid en eventueel worden bijgesteld. Dit bijstellen, vaak calibreren genoemd, kan alleen maar plaatsvinden als de begroting ook wordt vergeleken met de werkelijkheid. Het klakkeloos overnemen van een schattingsmethode is geen waarborg voor een goede schatting. Het werkelijke projectverloop moet worden vastgelegd in een werk- of urenregistratie. Ontbreekt deze registratie dan is geen ervaring van voorgaande projecten opgebouwd en is het gebruik van een schattingsmethode weinig zinvol.

Nacalculatie is noodzakelijk om de gebruikte schattingsmethode te calibreren. Uit onderzoek¹ blijkt dat van de bedrijven die aan het begin van een project een begroting opstellen slechts 43% nacalculeren. De bedrijven die in staat zijn om nauwkeurig te begroten doen dat wel. Nacalcula-

tie is daarom een noodzakelijke voorwaarde om tot goede begrotingen te komen, want 'de grootste fout die we kunnen maken, is niet te leren van de fouten die we maken'.

Tijdens het gehouden onderzoek² gaven de onderzochte bedrijven nog een aantal andere oorzaken aan waarom projecten uitlopen. De belangrijkste oorzaken zijn:

- wijzigingen tijdens het ontwikkelproces;
- de organisatorische omgeving.

Wijzigingen tijdens het ontwikkelproces

'Wijzigingen tijdens het ontwikkelproces' wordt genoemd als één van de grootste veroorzakers van verschillen tussen schatting en werkelijkheid. Een tweetal redenen voor deze wijzigingen kan worden onderkend:

- een (te) omvangrijk project,
- onduidelijke specificaties van het te ontwikkelen systeem.

Al eerder is gepleit voor een 'kleine stap'-strategie, die de omvang van een project beperkt waardoor de nauwkeurigheid van de schatting kan worden verhoogd. Als gevolg van een lange doorlooptijd zullen de uitgangspunten waarmee het project is gestart, bij de oplevering al weer achterhaald zijn. Omdat men geen project op wil leveren dat niet voldoet aan de (inmiddels veranderde) wensen van de gebruiker, worden tijdens het ontwikkelproces wijzigingen aangebracht. De invloed van deze wijzigingen op de oorspronkelijke schatting wordt vaak niet geschat of begroot, zodat de afwijking tussen de schatting en de realisatie steeds groter wordt.

De specificaties van het te ontwikkelen systeem worden in de eerste fasen van het project vanuit een globaal concept of probleemstelling steeds gedetailleerder beschreven. Deze beschrijvingen, gemaakt door en meestal voor automatiseerders zijn vaak in een vorm die voor de opdrachtgever en/of de gebruiker te technisch en te abstract is. Dit kan tot gevolg hebben dat later bij de gedetailleerde beschrijvingen of bij oplevering blijkt dat er allerlei bijstellingen moeten plaatsvinden. In de oorspronkelijke schatting waren deze bijstellin-

gen niet voorzien. Voor de voortgang en de realisatie van het project is dit desastreus.

De organisatorische omgeving

Onder de organisatorische omgeving verstaat men zowel de projectorganisatie waarbinnen het automatiseringsproject wordt uitgevoerd als de gebruikersorganisatie waarbinnen het uiteindelijke informatiesysteem moet functioneren.

De projectorganisatie kan een bepalende factor zijn voor het welslagen van het project. Hoewel hierover veel literatuur is geschreven, worden de problemen die ontstaan bij projectmatig werken vaak genegeerd, waardoor bijvoorbeeld de besluitvorming veel trager verloopt dan werd verwacht, of eenmaal genomen besluiten worden in een later stadium teruggedraaid. Beide hebben gevolgen voor de voortgang van het project en daarmee voor het realiseren van de begrote inspanning.⁵

Naast de projectorganisatie worden vaak de aan een project toegewezen middelen als belangrijke factoren voor een succesvol realiseren van een automatiseringsproject bestempeld. Aan toverwoorden als 'Computer Assisted Software Engineering' (CASE) of 'Software Engineering Workbenches' wordt een (te) grote produktiviteitsverbetering toegedicht. Geen enkel onderzoek bevestigt dat deze hulpmiddelen de produktiviteit van de systeemontwikkeling daadwerkelijk verbeteren.⁶ Het succes van een geslaagd project is slechts gedeeltelijk afhankelijk van deze hulpmiddelen.

Uit onderzoek is wèl gebleken dat de bedrijven die vrij rigide ontwikkelingsmethoden en -technieken gebruiken minder met overschrijdingen te maken hebben dan bedrijven die informatiesystemen 'uit de losse pols' ontwikkelen.

De problemen die kunnen ontstaan bij de invoering van systemen worden vaak niet onderkend. In hoeverre de gebruikersorganisatie het systeem kan en ook wil invoeren, wordt met name in de Amerikaanse literatuur onderbelicht. Enid Mumford is één van de weinigen die in haar systeemontwikkelingsconcept ETHICS zowel technische als organisatorische aspecten opneemt. Toch blijkt dat de organisatorische omgeving van groot

belang is. Riesewijk en Warmendam hebben gesignaleerd dat bij 68% van de projecten de organisatorische inpassing een probleem is. Acceptatie van het systeem door het personeel levert in 58% van de projecten problemen op.

Eén van de conclusies in een Nederlands onderzoek naar de sociaal-organisatorische implicaties van automatisering luidt:

'Een geringe aandacht voor sociaal-organisatorisch aspecten in de planning- en startfase van projecten blijkt in sommige gevallen een aanzienlijke probleemfactor te vormen (...). Ruime aandacht voor dergelijke aspecten kan zich daarentegen als een succesfactor openbaren.'⁷

Projectrisico's

In zijn eerste publikatie over de schattingsmethode 'Function Point Analysis' zegt Albrecht dat voorafgaand aan het starten van het project eerst gekeken moet worden naar het risico van het project:

'(...) we estimate the project. Then we go through the first of the feedback loops, where we assess the project risk, with a 28-question structured questionnaire'.⁸ Bij de verschillende implementaties van de methode FPA blijkt het schatten van projectrisico's verdwenen te zijn, terwijl alle andere uitgangspunten van deze methode wel worden overgenomen.

Risico-analyse

Risico-analyse is een geformaliseerde methode om risico's ten aanzien van overschrijding van het budget of ten aanzien van afbreuk bij automatiseringsprojecten te bepalen. De meeste methoden bestaan uit een gestructureerde vragenlijst aan de hand waarvan voor een aantal mogelijke probleemgebieden het geschatte risico wordt bepaald.

De methoden voor risico-analyse vallen uiteen in een analyse vooraf en een evaluatie achteraf. Bij de evaluatie kunnen eventuele probleemgebieden, voor de volgende projecten als aandachtsgebieden fungeren. Beoordelingscriteria zoals die van Baily en Pearson, gebruikt en toegevoegd door zowel Bemelmans (1983) als Dikken,

worden gebruikt voor de evaluatie van automatiseringsprojecten.

Beoordeling vooraf vraagt om een ietwat andere benadering. Binnen een aantal softwarehuizen⁹ is een risico-analyse in de vorm van een enquête in gebruik. Deze analyse, afkomstig uit Zweden, is ontwikkeld voor de Zweedse overheid en wordt ook gebruikt bij grote bedrijven.

De Security by Analysis (SBA) richt zich ondermeer op vijf aspecten van een automatiseringsproject:

– De omvang.

Zowel de menskracht, de doorlooptijd als de organisatorische reikwijdte worden in risicofactoren omgezet.

– De gebruikersorganisatie.

De acceptatiebereidheid, de veranderingsbereidheid en de ervaring van de gebruiker zijn hier de risicobepalende factoren.

– De gebruikte automatiseringstechnologie.

De mate waarin de gebruikte automatiseringstechnologie nieuw is voor de gebruiker, de leverancier of de projectmedewerkers bepalen hier het risico.

– De projectorganisatie.

Kennis van het werken in projecten en ervaring met het werken in projecten van zowel de projectmedewerkers als van de in de projectgroep opgenomen gebruikers bepalen de mate waarin deze risico's gelopen worden.

– Randvoorwaarden.

Aan het project kunnen impliciet of expliciet randvoorwaarden worden gesteld. Deze randvoorwaarden bepalen in hoge mate het succes van een project. Aanwezigheid van veel randvoorwaarden verhoogt het risico.

De enquête bestaat uit 77 vragen verdeeld over deze vijf aspecten. Uiteindelijk wordt er per aspect en voor alle aspecten een risicofactor bepaald. Is de totale risicofactor hoog, dan is het slagen van het project zeer onzeker en kan men een keuze maken tussen het nemen van risicobeperkende maatregelen of het geheel stoppen van het project.

Als het totale risico gemiddeld is, kan toch in één van de groepen een verhoogd risico optreden. Risicobeperkende maatregelen voor dat aspect zijn dan aan te bevelen.

Conclusies

Een goede begroting van een project is gebaseerd op ervaringscijfers van het projectverloop van vorige projecten of (met enig scepsis) van soortgelijke projecten van andere organisaties. De totale begroting is opgebouwd uit deelbegrotingen per fase, waarbij de uitgangspunten van de begroting(en) zijn aangegeven. Per project ligt de totale doorlooptijd op maximaal één jaar of is het project opgesplitst in deelprojecten met een doorlooptijd van maximaal één jaar. De begroting geeft aan welke kostensoorten wel én welke niet zijn opgenomen. Er zijn maatregelen genomen om wijzigingen tijdens de uitvoering te vermijden of de impact van deze wijzigingen in de begroting op te nemen. Naast een schatting van de kosten is ook een schatting gemaakt van de projectrisico's en is aangegeven welke maatregelen er zijn genomen om (te grote) projectrisico's te vermijden. Met deze maatregelen en in een omgeving waarin automatisering geen gave maar een doodgewoon vak is, kan het mogelijk zijn om een informatiesysteem te bouwen dat op tijd en binnen het budget wordt opgeleverd en datgene doet wat ervan verwacht wordt.

Literatuur

- Albrecht, A. J.; Measuring application development and productivity of computersystems, *Proceedings of the Guide conference on application and productivity Symposium*, pp. 83-92, New York 1979.
- Bailey, J. E., en Pearson, S. W.; *Development of a tool for measurement and analyzing computer user satisfaction*, *Management Science*, vol. 29 nr. 5, mei 1983.
- Bemelmans, (1983) T. M. A. en Rijn, Th. M. J. van; Een methode voor evaluatie van automatiseringstoepassingen, *Informatie*, jaargang 25 nr. 12, pp. 22-29, 1983.
- Bemelmans, (1987), T. M. A.; *Bestuurlijke informatiesystemen en automatisering*, Stenfert Kroeze, 1987, pp. 141-142.
- Boehm, B. W.; *Software engineering economics*, Prentice Hill, New Jersey, 1981.
- Conte, S. D. Dunsmore, H. E. en Shen V. Y.; *Software engineering, metrics and models*, Benjamin/Cummings Publication Comp., 1987, pp. 173-176.
- DeMarco, T.; *Controlling software projects*, Yourdon Press, New York 1982.
- Dikken, J. A. H.; Gebruikerswaardering van automatiseringstoepassingen, *Informatie*, jaargang 29 nr. 2, pp. 101-196, 1987.

- Gilb, T.; *Software Metrics*, Wintrop Publishers, Cambridge 1977, pp. 214-218.
- Gordon Group; *Before you leap*, User's guide rel. 2.1 A(1), San José, 1987.
- Jensen, R. W.; *An improved macrolevel development estimation model*, Hughes Aircraft Compagny, verm. 1986.
- Londeix, B.; *Cost estimation for software development*, Addison Wesley Publishing Comp., 1987.
- Metzger, P. W.; *Managing a programming project*, Prentice-Hall, New Jersey, 1981.
- Mumford, E. en Henshall, D.; *A participative approach to computer design*, London, 1979.
- Myers, G. J.; *Software reliability*, Wiley & Sons, New York, 1976.
- Pauw, J. W.; Interfasetijd bij systeemontwikkeling, *Informatie* jaargang 31 nr. 4, pp. 241 t/m 320 1989.
- Putnam, L. H.; The real economics of software development, in: *Economics of information processing*, vol. 2, Operations, programming and software models, John Wiley & Sons, pp. 167-176, 1982.
- Riesewijk, B. en Warmerdam, J.; Het slagen en falen van automatiseringsprojecten, Instituut voor toegepaste sociale wetenschappen, vakgroep bestuur en beleid, KU Nijmegen, 1988, pp. 89.
- Rijsenbrij, D. B. B. en Bauer A.H.; Projectdiagnose, een goed begin is het halve werk, *Informatie* jaargang 31 nr. 3, pp. 153 t/m 240, maart 1989.
- Rowold, P.; *Schatten en begroten van automatiseringsprojecten*, Tutein Nolthenius, verschijnt oktober 1989.
- Rubin, H. A.; In: *Estimacs, product concept and facilities manual*, Computer Associates, 1987.
- Siskens, W. J. A., Heemstra F. J. en van der Stelt H.; Kostenbeheersing bij automatiseringsprojecten: een empirisch onderzoek, *Informatie*, jaargang 31 nr. 1, pp 1 t/m 72, 1989.
- Weghe, R. M. van de en Kraak R.; *Kwaliteits- en voortgangsbewaking bij softwareprojecten*, BSO 1987.

Noten:

- 1 Zie Siskens pag. 35.
- 2 Een door de auteur gehouden onderzoek, nog te publiceren zie Rowold.
- 3 Cursus SDM, Pandata. 1988. De vermelde verhouding maakt geen deel uit van de officiële cursusdocumentatie.
- 4 Zie Weghe, p. 9. De percentages sommen niet tot 100% omdat het midden van een bandbreedte is aangegeven.
- 5 Zie Pauw, een vertragsfactor (interfase tijd) van 0,8 is een gemiddelde.
- 6 Zie Siskens, p. 41.
- 7 Zie Riesewijk, pp. 232-233 voor twaalf geconstateerde sociaal-organisatorische aspecten.
- 8 Zie Albrecht, p. 83. Op het NGI congres 'FPA in beweging' in nov. 1988 geeft Albrecht aan dat FPA zonder de risicofactoren niet geschikt is om mee te schatten.
- 9 Zie Rijsenbrij.

Vestigingsplaats- beoordeling en de optimale winkelformule

Een praktische illustratie in de
kappersbranche

Drs. P. A. M. Versteijne

Inleiding

Twee problemen waarmee praktisch alle startende ondernemers in de detailhandel geconfronteerd worden zijn de bepaling van de meest geschikte vestigingsplaats voor het nieuwe bedrijf en de bepaling van de beste winkelformule. De doelstelling van dit artikel is duidelijk te maken hoe een ondernemer *met weinig middelen* een *concrete* oplossing kan krijgen voor de gestelde *samenhangende* problemen.

De hierna te presenteren oplossing is gebaseerd op een onderzoek uitgevoerd voor een startende ondernemer in de kappersbranche.

Er is daarbij gebruik gemaakt van secundaire bronnen, persoonlijke observatie en een passantenquête. Het onderliggende onderzoeksmodel is geconstrueerd door bruikbare elementen uit de marketingliteratuur te combineren.

Literatuuronderzoek

De concrete vraag van de startende ondernemer was: 'Is het winkelcentrum dat ik op het oog heb

Drs. P. A. M. Versteijne studeerde bedrijfseconomie aan de Rijksuniversiteit Groningen. Thans als Universitair Docent verbonden aan de Katholieke Universiteit Brabant, Sectie Marketing en Marktonderzoek.