

HET ONTWIKKELEN VAN NIEUWE APPLICATIES IS ÉÉN VAN DE HOOFDACTIVITEITEN BINNEN DE ICT AFDELING VAN HET BANK- EN VERZEKERINGS-BEDRIJF KBC. TERECHT VRAAGT DIT BEDRIJF ZICH AF IN HOEVERRE HET ONTWIKKELEN VAN NIEUWE APPLICATIES VOLDOENDE WAARDE OPLEVERT, GEGEVEN DE GEÏNVESTEERDE TIJD EN MIDDELEN. OM EEN ANTWOORD TE GEVEN OP DE VRAAG 'HOE PRODUCTIEF IS ICT-ONTWIKKELING BINNEN ONS BEDRIJF?' WERD VOOR APPLICATIEVE ONTWIKKELINGEN OP HET BELANGRIJKSTE MAINFRAME-PLATFORM EEN MEETOMGEVING OPGEZET GEBRUIK MAKEND VAN HET COCOMO II-MODEL. MET DEZE MEETOMGEVING WIL KBC HAAR PRODUCTIVITEIT IN KAART BRENGEN EN ZICHELZELF POSITIONEREN TEN OPZICHTE VAN ANDERE BEDRIJVEN. MAAR OOK, EN MISSCHIEN WEL BELANGRIJKER, WIL MEN DETECTEREN OP WELKE PLAATSEN IN HET ONTWIKKELINGSPROCES PRODUCTIVITEITSVERBETERING MOGELIJK IS.

### COCOMO II-MODEL ALS PRODUCTIVITEITSMETING

COCOMO II, het Constructive Cost Model [1], is een inschattingmodel ontwikkeld door Barry Boehm. De effort die nodig is om een stuk software te ontwikkelen, wordt ingeschat op basis van de grootte van het opgeleverde product, namelijk het aantal lijnen code (kSLOC<sup>1</sup>). Daarnaast worden ook een aantal andere projectkenmerken in rekening gebracht die de projectinspanning positief of negatief

beïnvloeden. Deze karakteristieken of cost drivers zijn enerzijds schaalfactoren (SF) die schaalvoordelen kunnen opleveren en anderzijds effort multipliers (EM) die een lineair effect hebben op de effort. We schatten het aantal persoonsmaanden (PM) met volgende COCOMO II-formule:

$$PM = A \times \text{Size}^E \times \prod_{i=1}^n EM_i$$

$$\text{met } E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

Iedere SF en EM wordt ingeschat op een schaal van *very low* tot *extra high* met een bijhorend gewicht. Voor een EM duidt dit gewicht de hoeveelheid extra effort aan die nodig is in vergelijking met de *nominale* schaal (gewicht 1). Bijgevolg zal een inschaling van het effect van een EM op 1,10 erop wijzen dat dit project 10% extra effort nodig heeft vergeleken met projecten waarvoor het effect van deze EM wordt ingeschaald op *nominaal*. Terwijl een gewicht van 0,91 erop wijst dat het project 9% minder effort nodig heeft dan in het *nominale* geval. De schaalfactoren bepalen de exponent. Afhankelijk van de gewichten zal deze groter of kleiner zijn dan 1 en krijgen we een diseconomy of economy of scale. De gewichten en de constante parameters A en B zijn bepaald in het COCOMO II-model door lineaire regressie op de 161 projecten in de COCOMO II-databank.

Aangezien COCOMO II een inschatting geeft van de benodigde workload voor een project, kunnen we die inschatting gebruiken om een uitspraak te doen over het al dan niet productief zijn van het project. De werkelijke effort vergelijken

met deze inschatting geeft een indicatie of het project beter of slechter scoorde dan de voorspelling volgens COCOMO II. Met andere woorden, de COCOMO II-inschatting wordt als norm gebruikt voor onze productiviteitsmeting. Eens er voldoende eigen projectdata aanwezig is, kunnen we de verschillende parameters in de formule kalibreren naar de huidige situatie binnen een bedrijf. Bij de volgende metingen zullen de projecten dan niet langer gebenchmarkt worden ten opzichte van de COCOMO II-norm, maar ten opzichte van een norm eigen aan het bedrijf.

### COST DRIVERS: EEN BRON VAN INFORMATIE

De cost drivers (EM en SF) uit het COCOMO II-model bieden een uitgebreide lijst van factoren die een positieve of negatieve invloed hebben op de projectinspanning binnen softwareontwikkeling (zie tabel 1). De gewichten toegekend aan de verschillende classificaties van een cost driver bepalen de grootte van die invloed. De 161 projecten in de COCOMO II-databank werden gebruikt om deze initiële gewichten toe te kennen. Er is echter weinig informatie bekend over deze 161 projecten en het is aannemelijk dat deze zullen afwijken van de eigen projecten. Bijgevolg is het aan te raden om het model te kalibreren en na te gaan of deze waarden ook binnen de eigen omgeving correct zijn. Om dit uit te zoeken hebben we een rapport ontwikkeld waaruit we het werkelijke effect (zoals waargenomen in de projectdata) van een effort multiplier afleiden.

Met dit rapport wensen we na te gaan dat de EM geïdentificeerd in COCOMO II inderdaad factoren zijn die invloed

# Productiviteit meten en verbeteren

Lotte De Rore, Monique Snoeck, Geert Poels en Guido Dedene

TABEL 1

Schaalfactoren en effort multipliers in COCOMO II

Schaalfactoren	VL	L	N	H	VH	EH
Precedentedness	6.20	4.96	3.72	2.48	1.24	0.00
Development flexibility	5.07	4.05	3.04	2.03	1.01	0.00
Architecture/Risk resolution	7.07	5.65	4.24	2.83	1.41	0.00
Team cohesion	5.48	4.38	3.29	2.19	1.10	0.00
Process maturity	7.80	6.24	4.68	3.12	1.56	0.00
Effort multipliers	VL	L	N	H	VH	EH
Product factoren						
Required software reliability	0.82	0.92	1.00	1.10	1.26	
Database size		0.90	1.00	1.14	1.28	
Product complexity	0.73	0.87	1.00	1.17	1.34	1.74
Developed for reusability		0.95	1.00	1.07	1.15	1.24
Documentation match to life-cycle needs	0.81	0.91	1.00	1.11	1.23	
Platform factoren						
Execution time constraint			1.00	1.11	1.29	1.63
Main storage constraint			1.00	1.05	1.17	1.46
Platform volatility		0.87	1.00	1.15	1.30	
Personeelsfactoren						
Analyst capability	1.42	1.19	1.00	0.85	0.71	
Programmer capability	1.34	1.15	1.00	0.88	0.76	
Personnel continuity	1.29	1.12	1.00	0.90	0.81	
Applications experience	1.22	1.10	1.00	0.88	0.81	
Platform experience	1.19	1.09	1.00	0.91	0.85	
Language and tool experience	1.20	1.09	1.00	0.91	0.84	
Project factoren						
Use of software tools	1.17	1.09	1.00	0.90	0.78	
Multisite development	1.22	1.09	1.00	0.93	0.86	0.80
Required development schedule	1.43	1.14	1.00	1.00	1.00	

1 kSLOC = kilo Source Line of Code

hebben op de projecteffort binnen dit bedrijf. En indien dit zo is, willen we ook nagaan of de invloed volgens COCOMO II vergelijkbaar is met het effect dat we meten in de projectdata.

### RAPPORT: INVLOED VAN EFFORT MULTIPLIER $EM_i$

Met ons rapport wensen we na te gaan hoe perfect de schatter is voor de invloed van een effort multiplier  $EM_i$  binnen dit bedrijf. Daarvoor isoleren we eerst het effect door de invloed van alle  $EM_j$  die we buiten beschouwing laten te elimineren. Zo bekomen we de *genormaliseerde effort ten opzichte van  $EM_i$* . Neem bijvoorbeeld twee effort multipliers  $EM_1$  en  $EM_2$ , met een respectievelijke waarde van 1,10 en 1,20. De totale effort wordt berekend als  $A \times \text{Size}^E \times 1,10 \times 1,20 = A \times \text{Size}^E \times 1,32$ . Indien we enkel het effect van  $EM_1$  wensen te onderzoeken, moeten we de berekende effort delen door  $EM_2$  om zijn effect te elimineren: *Genormaliseerde Effort* =  $(A \times \text{Size}^E \times 1,10 \times 1,20) / 1,20$ . Merk op dat we hier dus impliciet veronderstellen dat de invloed van de andere EM correct geschat wordt door het COCOMO II-model. Algemeen kunnen we dus stellen dat:

$$(1) \quad \text{GenormaliseerdeEffort}_i = \frac{\text{Effort}}{\prod_{j \neq i} EM_j} = A \times EM_i \times \text{Size}^E$$

We verkrijgen een lineair verband tussen een effort meting (de genormaliseerde effort) en een maat van de grootte van het project ( $\text{size}^E$ ) en dit verband is recht evenredig met de waarde van  $EM_i$ .

Wanneer we in een rapport de genormaliseerde effort uitzetten tegenover de  $\text{size}^E$  en de projecten groeperen volgens hun

classificatie voor  $EM_i$ , zal de regressielijn (met intercept gelijk aan nul) doorheen elke groep met eenzelfde classificatie er als volgt uitzien:

$$(2) \quad GenormaliseerdeEffort_i = R \times Size^E$$

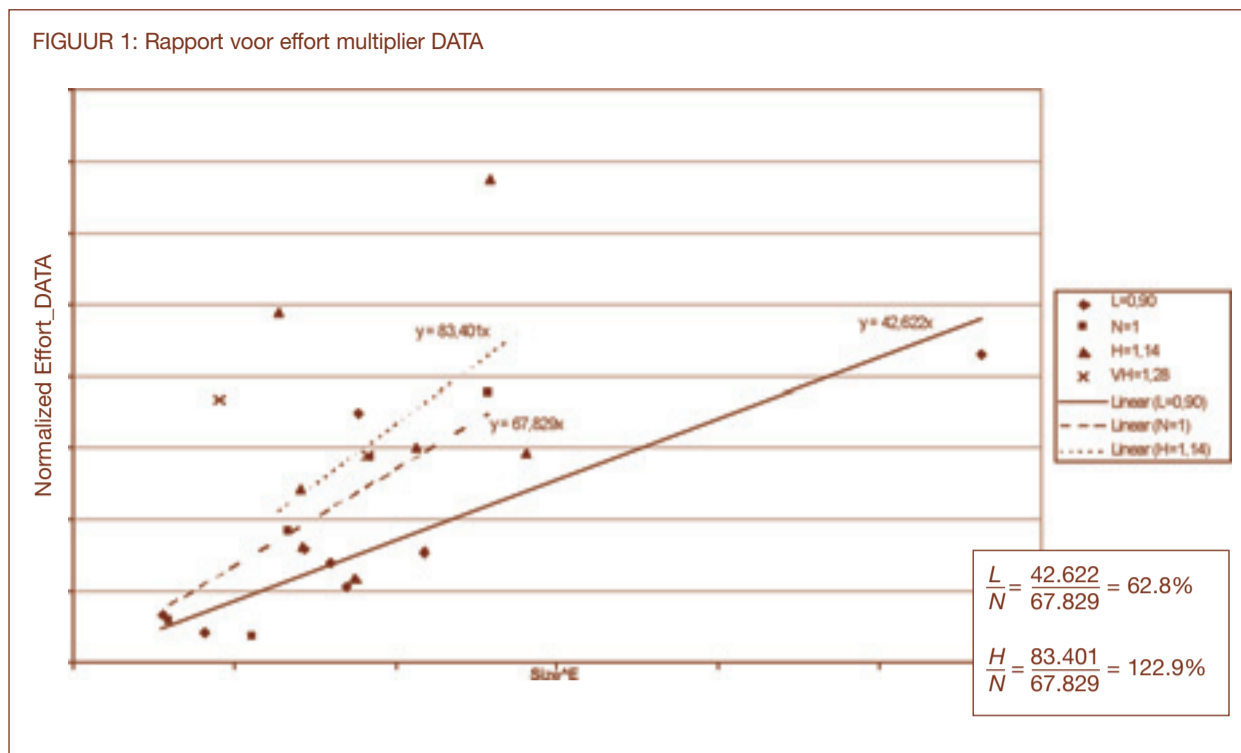
Vergelijken we (1) en (2), dan zien we dat de richtingscoëfficiënt van de regressielijn, genoteerd met  $R$ , proportioneel in verhouding staat met de waarde van  $EM_i$ . Op deze manier kunnen we de invloed van de effort multiplier op de geanalyseerde projecten kwantificeren.

## INTERPRETATIE VAN HET RAPPORT

Het gewicht voor een classificatie van een EM duidt aan hoeveel extra of minder effort een project nodig heeft in vergelijking met een *nominale* classificatie. Bijgevolg, wanneer we de richtingscoëfficiënt van iedere classificatie vergelijken met die van de *nominale*, verkrijgen we informatie over het effect van de verschillende categorieën van de effort multiplier.

Veronderstel bijvoorbeeld dat de COCOMO II-waarde voor een *high* classificatie van  $EM_i$  gelijk is aan 1,10, wat overeenstemt met 10% meer effort dan een *nominaal* project. De richtingscoëfficiënt van de *high* en *nominale* regressielijn zullen aantonen of het effect van  $EM_i$  voorspeld door COCOMO II ook zichtbaar is in de geanalyseerde projectdata.

Bekomen we eenzelfde ratio (in het voorbeeld 1,10), dan kunnen we besluiten dat  $EM_i$  inderdaad een effect heeft op de effort. Meer nog, de grootte van het effect gemeten in de



reductie van 37% in de effort, terwijl een *high* classificatie (voor het project wordt een aanzienlijke hoeveelheid eigen testdata gecreëerd) zal leiden tot een toename van 23% in de effort vergeleken met een *nominale* classificatie (voor het project wordt een minimum aan eigen testdata gecreëerd). Deze af- en toename zijn groter dan voorspeld door COCOMO II (-10% voor *low* en +14% voor *high*).

Gezien de kleine hoeveelheid aan projecten, kunnen we de richtingscoëfficiënten niet zien als absolute waarden.

bouwen en naar het management toe te visualiseren dat het inderdaad belangrijk is om te investeren in software process improvement om zo het (negatieve) effect van deze effort multiplier te verminderen.

Dit onderzoek kadert binnen de KBC-leerstoel over 'Managing efficiency aspects of software factory systems' gesponsord door KBC Bank en Verzekeringen.

# en met behulp van COCOMO II

geanalyseerde projecten is vergelijkbaar met het ingeschatte effect door COCOMO II. Krijgen we echter een significant verschillende ratio, kunnen we besluiten dat er wel invloed is maar dat het effect van  $EM_i$  voor dit bedrijf verschillend is. Stel bijvoorbeeld dat we een ratio van 1,20 verkrijgen: dan heeft een *high* project niet 10% maar 20% meer effort nodig dan een *nominaal* project. Verder onderzoek naar de reden van afwijking kan opportuniteiten voor productiviteitsverbetering opleveren. Anderzijds, wanneer de ratio ongeveer gelijk is aan 1, kunnen we besluiten dat  $EM_i$  geen invloed heeft. Bijgevolg mogen we *high* projecten geen extra effort toewijzen omdat ze anders productiever lijken.

Samengevat, het vergelijken van elke regressielijn met de nominale regressielijn levert ons twee soorten informatie. Ten eerste wijst een verschil erop dat  $EM_i$  inderdaad een effect heeft op de effort. Ten tweede, toont de ratio ook aan of dit effect vergelijkbaar is met wat COCOMO II voorschrijft.

## TOEPASSING BIJ KBC

KBC zette een meetomgeving op met behulp van het COCOMO II-model om hun productiviteit in kaart te brengen alsook de mogelijke plaatsen in het ontwikkelproces te identificeren waar verbetering mogelijk is [2]. Na één jaar meten werd een eerste analyse uitgevoerd op de data van 22 projecten [3]. Figuur 1 illustreert het rapport van DATA (database size), de effort multiplier voor de invloed van testdata-behoefte op de projecteffort.

Bij vergelijking van de richtingscoëfficiënten van de drie regressielijnen (*low*, *nominaal* en *high*) zien we een grotere invloed van DATA dan voorspeld door COCOMO II. Een *low* classificatie (het project gebruikt een kopie van de productiedata in de acceptomgeving als testdata) zal leiden tot een

Toch mogen we uit dit rapport afleiden dat DATA een groot effect heeft binnen KBC en dat het effect van deze factor zeker verder onderzocht moet worden. Na bevraging bij het kernteam "testen" (vertegenwoordigers van alle testers) werden de huidige processen voor het vinden en beheren van geschikte testdata als een mogelijke verklaring voor de (te) grote extra inspanningen naar voren geschoven. Een voorbeeld van een element dat meespeelt is dat alle projecten werken met dezelfde testgegevens. Dit betekent dat wanneer testdata gewijzigd worden door een project, de originele data verloren gaan en mogelijks niet langer bruikbaar zullen zijn voor een ander project. Hoewel men zich bewust was van het niet optimaal zijn van deze werkmethode, had men geen idee van de impact van de werkwijze rond testen op de projectinspanning. Het rapport bood voldoende materiaal om de directie van ICT te overtuigen het ophalen en beheren van fysieke testdata in aanmerking te nemen voor verder onderzoek en verbetering.

## BESLUIT

Eén van de grote troeven van het COCOMO II-model is de uitgebreide lijst van projectkarakteristieken. Niet alleen duiden die karakteristieken factoren aan die een invloed hebben op de inspanning nodig om een softwareproduct te ontwikkelen, maar COCOMO II kwantificeert ook hoe groot deze invloed is. Het hier voorgesteld rapport biedt de kans om de werkelijke invloed binnen de eigen projecten te visualiseren. Het is een instrument om probleemgebieden, en bijgevolg ook domeinen waar verbetering mogelijk is, te identificeren. Wanneer de resultaten van dit rapport overeenstemmen met het gevoel en de ervaring van de experts is dit een duidelijke indicatie dat er een opportuniteit voor productiviteitsverbetering bestaat. Het rapport kan dan ook gebruikt worden om een business case op te

**Lotte De Rore** is als wetenschappelijk medewerker verbonden aan de onderzoeksgroep Beleidsinformatica (LIRIS)  
E-mail: lotte.derore@econ.kuleuven.be



**Monique Snoeck** is als hoogleraar verbonden aan de onderzoeksgroep Beleidsinformatica (LIRIS)  
E-mail: monique.snoeck@econ.kuleuven.be



**Geert Poels** is als docent verbonden aan de Faculteit Economie en Bedrijfskunde van de Universiteit van Gent  
E-mail: geert.poels@ugent.be



**Guido Dedene** is als gewoon hoogleraar verbonden aan de onderzoeksgroep Beleidsinformatica (LIRIS) en als hoogleraar Informatie- & Communicatie Systemen aan de Business School van de Universiteit van Amsterdam  
E-mail: guido.dedene@econ.kuleuven.be



## REFERENTIES

- [1] Boehm, B.W. "Software cost estimation with COCOMO II", Prentice Hall, New Jersey, 2000.
- [2] De Rore, L., Snoeck, M. and Dedene, G., "Cocomo II applied in a bank and insurance company: Experience report", Proceedings of the 3rd Software Measurement European Forum, May 10-12 2006, Rome, pp. 247-257.
- [3] De Rore, L., Snoeck, M., Poels, G. and Dedene, G., "Deducing software process improvement areas from a Cocomo II-based productivity measurement", Proceedings of the 5th Software Measurement European Forum, May 2008, Milan (to appear).