Beter meten met Cffp

Omvangbepaling voor eigentijdse ontwikkelmethoden

Functiepuntanalyse is de meest gebruikte methode voor omvangbepaling van softwareontwikkelprojecten. De telrichtlijnen hiervan zijn echter lastig toepasbaar bij nieuwe methoden zoals objectgeoriënteerd of componentgebaseerd ontwikkelen. Voor deze methoden zijn *Cosmic full function points* ontwikkeld.

Ton Dekkers

Bij veel succesvolle projecten is de projectomvang de basis voor de budgetbeheersing (tijd en kosten). Het uitgangspunt hierbij is een correlatie tussen de omvang van het project en de benodigde inspanning. Wanneer de omvang en de aan het project bestede uren bekend zijn, is de productiviteit te berekenen. Omdat er verschillen zijn tussen projecten, is het uiteraard belangrijk de omstandigheden van de projecten te kennen.
De ontwikkelomgeving is sterk bepalend voor de productiviteit. Wanneer de productiviteit tussen projecten, branches of leveranciers wordt vergeleken, dan gebeurt dat normaliter per ontwikkelomgeving. Lastiger zijn de projectspecifieke invloeden op het budget. Dat zijn de maatregelen om risicovolle omstandigheden te beheersen als tijdsdruk, ervaringsniveau, besluitvorming en de aard van de software.

Analyse van gerealiseerde projecten met het model in figuur 1 levert diverse prestatiecijfers, zoals de productiviteit. Bij budgetteren wordt dit model gebruikt, bij voorkeur met prestatiecijfers op basis van eigen ervaringen. Bij onvoldoende of geen ervaring worden cijfers van derden gebruikt. Een belangrijke onafhankelijke aanbieder van ervaringscijfers is de International Software Benchmarking Standards Group (ISBSG). De basis voor het budgetteren blijft de omvang.

Omvangbepaling

Functiepuntanalyse is wereldwijd de meest gebruikte methode voor omvangbepaling. Een methode in opkomst is *Cosmic full function points* (Cffp). Beide methoden zijn in december 2002 Iso-gecertificeerd, waarmee is aangetoond dat de methoden voldoen aan de definities van Iso/IEC-standaard 14143. Hierin staat beschreven waar een meetmethode voor functionele omvang aan moet voldoen. Andere in Nederland gangbare methoden zijn 'functiepuntanalyse in onderhoud' en testpuntanalyse. Functiepuntanalyse (fpa) wordt vooral gebruikt bij nieuwbouwprojecten in de zakelijke software. 'Functiepuntanalyse in onderhoud' is een aanvulling op fpa, zodat ook onderhoudsprojecten methodisch kunnen worden begroot. Cosmic full function points is toepasbaar bij zowel nieuwbouw als onderhoud. Testpuntanalyse is een

Samenvatting

Elke *entry*, *exit*, *read* en *write* heeft binnen *Cosmic full function points* (Cffp) de omvang van 1 *Cosmic functional size unit*. De omvang van een functioneel proces is gelijk aan het aantal entry's, exits, reads en writes van het proces. Cffp biedt verschillende voordelen boven functiepuntanalyse zoals de eenvoud van meten en een betere omgang met complexiteit van software.

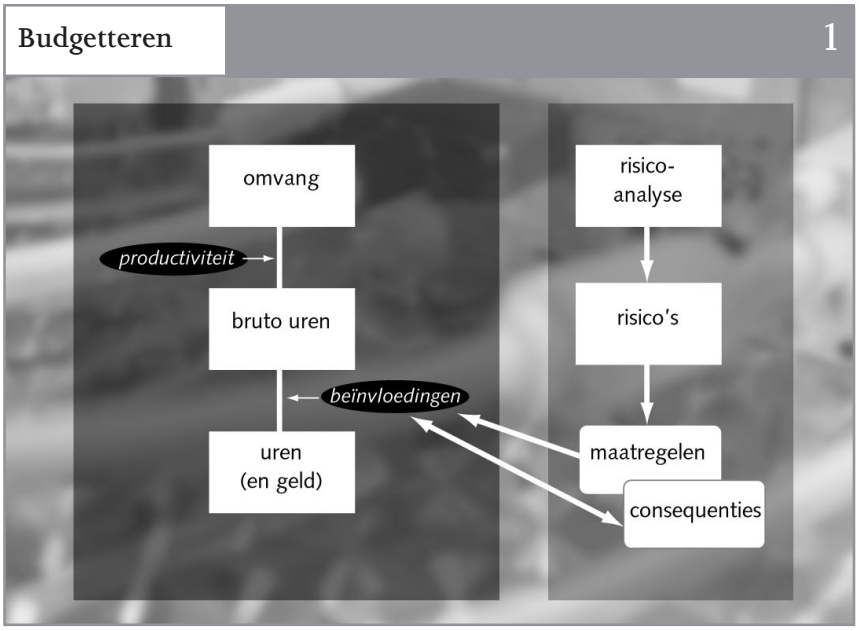
methode om de testinspanning te begroten. Vooralsnog is deze methode gebaseerd op fpa, een versie op basis van Cffp is in ontwikkeling.

Cosmic full function points

Fpa is ontstaan in een periode van het mainframe en ontwikkeling aan de hand van specificaties, gebaseerd op functionele decompositie. In deze omgeving bestaat een sterke correlatie tussen omvang en inspanning. De nieuwe ontwikkelmethoden (objectgeoriënteerd, componentgebaseerd, rad) en het toepassen van architectuur (naast de klassieke monolithische architectuur, multi-tier of web) hebben ook gevolgen voor de specificaties. De telrichtlijnen van fpa zijn daardoor lastiger toepasbaar.
De toename van de complexiteit van software door vergaande integratie van systemen, de plaats van software in andere domeinen en de aard van software (real time oplossingen en ingebedde software) zetten de veronderstelde correlatie tussen de functionele omvang in functiepunten en inspanning onder druk. Wat echter blijft is dat ook ontwikkeling met een nieuwe aanpak moet worden begroot en bewaakt. Het Common Software Measurement International Consortium (Cosmic) ontwikkelde Cffp om aan de nieuwe vraag te kunnen voldoen.

Uitgangspunten

Op het hoogste niveau gaat Cffp uit van het softwarecontextmodel. Op dit niveau wordt aangegeven



waarom en vanuit welk gezichtspunt (viewpoint) de meting wordt uitgevoerd. Hoewel de te meten componenten per gezichtspunt kunnen verschillen, zijn de meetprincipes zijn gelijk. Voor de eenvoud worden in dit artikel een generiek softwaremodel als voorbeeld gebruikt.
De functionele omvang is zoals bij alle Iso-14143-gecertificeerde methoden afgeleid van de *functional user requirements* (functionele gebruikersspecificaties – *fur's*). Voor het meten worden *base functional components* (functionele basiscomponenten – *bfc's*) gebruikt. Om de bfc's te kunnen identificeren, hanteert Cffp twee principes. De software wordt geactiveerd door invoer en levert uitvoer of resultaten bruikbaar voor de gebruiker. En de software verwerkt, bewerkt en presenteert brokken informatie in de vorm van een gegevensgroep die bestaat uit één of meerdere attributen.

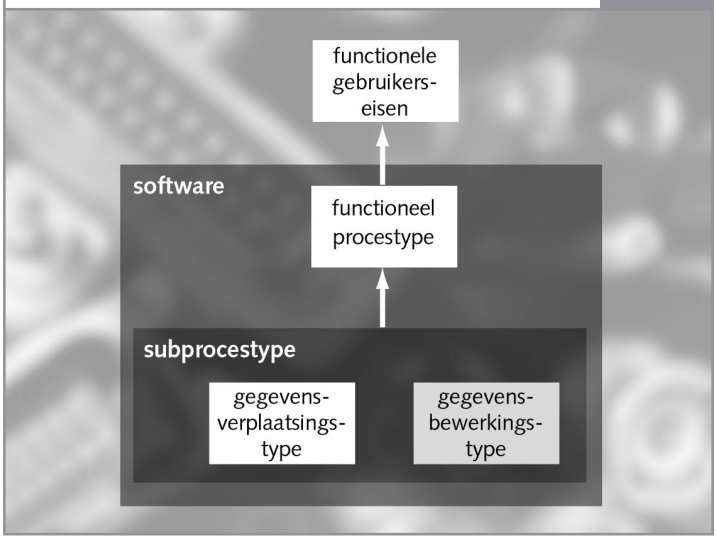
Cosmic
De ervaren beperkingen van de toepasbaarheid van functiepuntanalyse zijn in 1998 aanleiding geweest voor de oprichting van Cosmic (COmmon Software Measurement International Consortium). Cosmic is een initiatief van experts op het gebied van softwaremetrieken uit alle werelddelen, zowel uit de academische wereld als uit de praktijk.
Cosmic heeft zich tot doel gesteld een nieuwe meetmethode te definiëren die geschikt is om de prestatiecijfers (productiviteit, levertijd en kwaliteit) te bepalen. De meetmethode moet toepasbaar zijn in zo veel mogelijke software'domeinen' en besturingssystemen. Denk hierbij aan zakelijke software, real time software en ingebedde software. Cosmic coördineert het valideren, beschikbaar stellen en de acceptatie van de methode.

Volgens het model in figuur 2 worden de fur's onderverdeeld in een aantal functionele processen. Elk van die functionele processen bestaat uit een unieke verzameling van bfc's van het type gegevensverplaatsing of datamanipulatie.
Cffp onderkent vier soorten subprocessen die gegevens verplaatsen: *entry*, *exit*, *read* en *write* (zie figuur 3). Entry's leveren gegevens (van de gebruiker) aan het functionele proces. Exits bieden gegevens aan aan de gebruiker. Reads en writes verplaatsen gegevens van en naar opslag. Het bewerken van gegevens wordt gezien als onderdeel van de gegevensverplaatsingen. Bij zakelijke en real time software blijkt deze aanname op te gaan. Voor wetenschappelijke software, met veel algoritmen, is de methode niet geschikt.

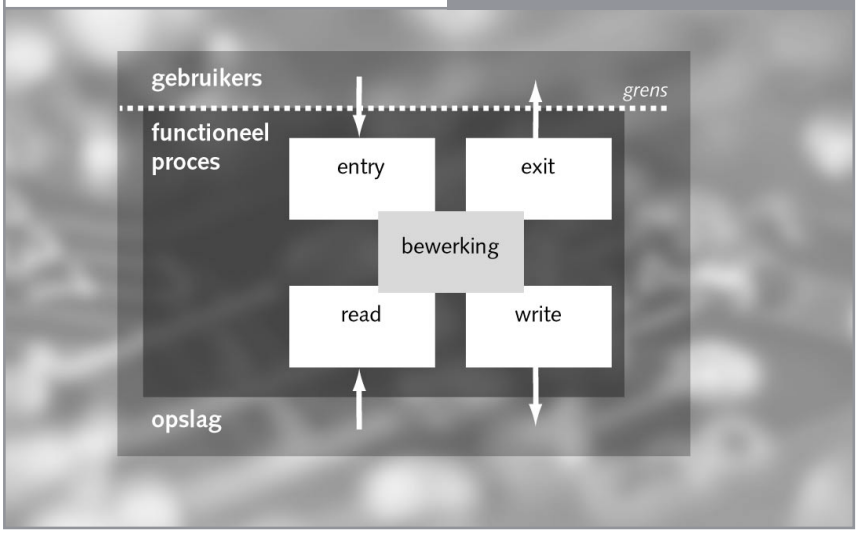
Definities

In het meethandboek worden de relevante begrippen uitgebreid beschreven. De belangrijkste zijn de grens en de gebruiker. De (systeem)grens is de grens tussen de te meten software en zijn omgeving. De omgeving wordt bepaald door het perspectief van de gebruiker. De gebruiker is iedere persoon of ieder object dat communiceert of samenwerkt met de software. Met deze definities zijn multi-tier, componenten-, real time software en interacties met apparatuur binnen de meetmethode te plaatsen.
Een *functioneel proces* is een elementair onderdeel van de fur's die bestaat uit een unieke samenhangende verzameling gegevensverplaatsingen en die zelfstandig uitvoerbaar is. Een proces wordt geactiveerd door een gebruiker en is beëindigd als is opgeleverd wat wordt verwacht.
Een *gegevensgroep* is een onderscheidende verzameling attributen waarvan elk attribuut een aanvullend aspect beschrijft van hetzelfde item. De aard van een gegevensgroep wordt bepaald door de rol.
Een *object of interest* is gedefinieerd binnen de eisen en wensen en kan zijn een fysiek iets of een begrip of deel van een begrip in de wereld van de gebruiker dat van belang is voor het proces en/of voor opslag.
De *gegevensgroeppersistentie* wordt bepaald door de rol van de gegevens in het functioneel proces. Cffp kent permanente en vluchtige gegevens. Permanent zijn gegevens die na afloop van het proces blijven bestaan, gegevens die worden opgeslagen. Vluchtige gegevens zijn na het proces niet meer beschikbaar. De gegevens worden doorgegeven door de gebruiker aan functionele processen, door processen aan andere processen en door processen aan gebruikers.
Een *entry* is een gegevensverplaatsing die een gegevensgroep van de

Gegevensmodel functionele gebruikerseisen 2



Vier typen gegevensverplaatsing 3



gebruiker over de (systeem)grens brengt naar het functionele proces dat deze gegevens nodig heeft.
Een *exit* is een gegevensverplaatsing die een gegevensgroep van het functionele proces over de (systeem)grens brengt naar de gebruiker die deze gegevens nodig heeft.
Een *read* is een gegevensverplaatsing die een gegevensgroep van de opgeslagen permanente gegevens brengt naar het functionele proces dat deze gegevens nodig heeft.
Een *write* is een gegevensverplaatsing die een gegevensgroep brengt van het functionele proces naar de opgeslagen permanente gegevens.
In figuur 4 is de meetmethode schematisch weergegeven.

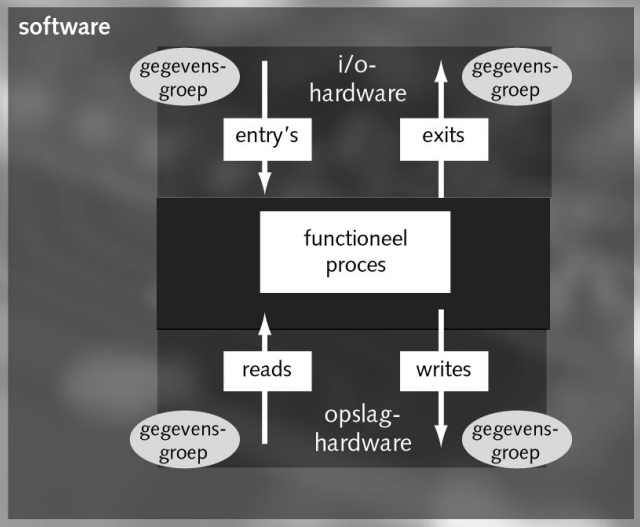
De Cffp-meeteenheid is *Cosmic functional size unit* (Cfsu). Elke gegevensverplaatsing heeft standaard de omvang van 1 Cfsu. De omvang van een functioneel proces in Cfsu's is daardoor gelijk aan het aantal entry's, exits, reads en writes waaruit het functioneel proces bestaat. De omvang van de software is gelijk aan de som van de omvang van de functionele processen die samen de software bepalen.

Meten

Twee voorbeelden maken duidelijk hoe de regels in de praktijk worden toegepast. Als eerste een voorbeeld uit de wereld van de zakelijke software. Het functionele proces bestaat hier uit het aanmaken van facturen van de transacties van week 40. Hiervoor zijn acht gegevensverplaatsingen nodig (1 entry, 2 reads, 1 write en 4 exits; zie figuur 5). De omvang van dit functionele proces is 8 Cfsu's.

Het andere voorbeeld is een real time toepassing. Dit functionele proces bestaat uit het op vaste tijdsintervallen controleren van de temperatuur om een ruimte op de gewenste temperatuur te houden. Hierbij zijn vier gegevensverplaatsingen nodig (2 entry's, 1 read en 1 exit; zie figuur 6). De omvang van dit functioneel proces is 4 Cfsu's.

Meetmethode 4



Gegevensverplaatsingen voor transacties week 40 5

soort verplaatsing	omschrijving actie
entry	geef weeknummer in (vluchtig); activerende procesgegevens, van gebruiker naar functioneel proces
read	lees leveringen (permanent); benodigde procesgegevens, van opslag naar functioneel proces
read	lees klantgegevens (permanent)
write	sla factuurgegevens op (permanent); resultaat proces, van functioneel proces naar opslag
exit	print klantgegevens op factuur (vluchtig); resultaat proces, van functioneel proces naar gebruiker
exit	print leveringen op factuur (vluchtig)
exit	print totalen op factuur (vluchtig)
exit	verantwoord proces (vluchtig)

Gegevensverplaatsingen temperatuurcontrole 6

soort verplaatsing	omschrijving actie
entry	geef kloksignaal in (vluchtig)
entry	geef huidige temperatuur in (vluchtig)
read	lees temperatuurinstelling (permanent)
exit	zend aan/uit-signaal naar verwarming (vluchtig)

Voordelen

Het gebruik van Cffp biedt verschillende voordelen boven dat van fpa. Een belangrijk voordeel van Cffp is de eenvoud van meten. Bij Cffp worden de functionele processen gewaardeerd door de aantallen gegevensverplaatsingen (entry, exit, read en write), die zijn gewaardeerd als 1 Cfsu. Bij fpa moeten de functionele processen worden benoemd als invoer-, uitvoer- of opvragingsfunctie. Het type functie en de aantallen gerefereerde gegevensverzamelingen en attributen zijn bij fpa bepalend voor het aantal functiepunten.

Datamodel

Een ander voordeel van Cffp is dat permanente gegevens niet worden gewaardeerd. Bij fpa is dit wel het geval. Om fpa te kunnen toepassen moet er een datamodel aanwezig zijn. Om fpa te kunnen toepassen en de complexiteit van de functies goed te kunnen bepalen bij een ontwikkelmethode waarbij datamodellering geen specifieke activiteit is, moet dit worden opgesteld. Dit is bijvoorbeeld het geval bij een objectgeoriënteerde ontwikkelmethode. Bij Cffp speelt een datamodel geen bepalende rol. Gegevens worden niet als zelfstandig item gewaardeerd.

Complexiteit

De ervaring leert dat de complexiteit van de functionele processen bij nieuwe software toeneemt. Deze groei in complexiteit heeft binnen fpa geen evenredige groei van functiepunten tot gevolg. Een belangrijke factor binnen Cffp is de nuancering in de omvang van een functie. Binnen fpa is de omvang minimaal 3 functiepunten en maximaal 7 functiepunten. Bij Cffp is een functioneel proces is minimaal 2 Cfsu. Er is in principe geen maximum.

Architectuur

Met het Cffp-softwarecontextmodel is het mogelijk de omvang van architectuurafhankelijke specificaties apart te bepalen, zoals bij een meerlagenarchitectuur. Ook van real time toepassingen in zowel zakelijke software als in ingebedde software (consumentenelektronica, mobiele telefoons) waar de definities van fpa tekortschieten is met Cffp de omvang goed te bepalen. Overigens is Cffp goed toepasbaar waar fpa dit is.

Nadelen

Natuurlijk er ook nadelen. Door de jarenlange toepassing van fpa zijn er veel meer prestatiecijfers bekend op basis van functiepunten. ISBSG is inmiddels begonnen projecten in haar database op te nemen waarvan de omvang is bepaald met Cffp. Met fpa kan op basis van een datamodel in een vroeg stadium van het project een indicatie worden gegeven van de omvang van een project. Cffp kent ook een indicatieve methode, *early ffp*, maar deze moet zich nog bewijzen in de praktijk.

Conclusie

Omdat de mogelijkheden van Cffp groter zijn dan die van fpa, zal Cffp fpa op termijn gaan vervangen. Naarmate meer ervaring wordt opgedaan met de methode, duidelijke richtlijnen ontstaan voor de praktische toepassing van Cffp en ook representatieve prestatiecijfers beschikbaar komen, zal de overgang van fpa naar Cffp versnellen. De *Cosmic/ISBSG-benchmarking study* 2003 is een eerste verzameling cijfers.

Reviewer Hans van der Mey

Literatuur

Cosmic (2003). *Measurement Manual Cosmic Full Function Points 2.2.*

Cosmic/ISBSG (2003). *Cosmic Benchmarking Invitation.*

ISBSG (2002). *Software Metrics Compendium.*

Iso (1998). *Iso/IEC 14143 Software Engineering – Software Measurement – Functional Size Measurement – Part 1: Definitions of concepts.*

Iso (2003). *Iso/IEC 19761 Software Engineering – Software Measurement – Cosmic Full Function Points.*

Nesma (april 1996). *Definities en telrichtlijnen voor de toepassing van functiepuntanalyse 2.0.*

Pol, M., R. Teunissen & E. van Veenendaal (1999). *Testen volgens Tmap®.* 2de druk (hoofdstuk 12). Uitgeverij Tutein Nolthesius.

Toivonen, H. (2002). Defining Measures for Memory Efficiency of the Software in Mobile Terminals. *Proceedings 12th International Workshop Software Measurement.* Nokia.

Links

www.Cffp.nl
www.Cosmicon.com
www.isbsg.org
www.lrgl.uqam.ca/Cosmic-ffp
www.nesma.org

Ton Dekkers
is senior projectconsultant bij de divisie Engineering & Projecten van Sogeti Nederland B.V. en coördinator van het Expertise Centrum Metrieken. E-mail: ton.dekkers@sogeti.nl