



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden_nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.



De atomaire produktkwaliteit van een informatiesysteem

J. Acohen

Moret Ernst & Young management consultants, Utrecht

Inleiding

De laatste tien jaren is de interesse van opleidingsinstituten en ondernemingen verschoven van de programmeerkunst naar relationele databases, case tools en dergelijke. Toch wordt er in ongeveer tien procent van de gestarte nieuwbouwprojecten geen informatiesysteem opgeleverd en de kosten van het testen en onderhouden van 'geslaagde' systemen neemt grote vormen aan. De kunst van het specificeren/programmeren dient weer in ere te worden hersteld.

Stapt u in een vliegtuig dat geheel door computers wordt bestuurd?

Bij het beantwoorden van bovenstaande vraag, die ook wat minder dramatisch kan worden gesteld zoals bijvoorbeeld 'besteedt u uw geld uit aan een bank die de administratie geautomatiseerd voert?', vraagt u zich niet af of er een 4GL is gebruikt. U denkt ook niet aan relationele databases, case tools, administratieve organisatie, projectbeheer, informatieplanning of Information Engineering. Toch zijn dit de begrippen die de laatste tien jaar in automatiserings(software)wereld de boventoon voeren. Hiermee wil ik niet zeggen dat dit ten onrechte het geval is, want al deze elementen spelen wel degelijk een rol in de kwaliteit van informatiesystemen, maar ik wil hier een lans breken voor de stelling dat de kunst van het specificeren/programmeren meer aandacht verdient dan die het momenteel krijgt.

In de beginjaren tachtig werd voorspeld dat Cobol-programmeurs overbodig zouden worden, er was vooral behoefte aan systeemanalisten en systeemontwerpers. De opleidingen op HBO-niveau gingen zich meer met informatie-analyse en SDM bezig houden dan met Pascal of programmeermethoden. Er werd in die tijd zelfs over Cobol-krassers gesproken. Programmeren werd een VWO/MBO-activiteit. Schoolverlaters van het HEAO wilden direct als systeemontwerpers aan de slag.

Het wordt echter tijd weer eens met beide benen op de grond te gaan staan en toe te geven dat specificeren/programmeren zeer moeilijk is. Zo moeilijk zelfs dat bij grote systemen evenveel tijd wordt besteed aan het testen als aan het ontwikkelen zelf. Tijdens het testen komt men tot de conclusie dat er steeds weer nieuwe fouten in de programma's zitten en dat de functionaliteit toch ook weer enigszins anders is dan men vooraf veronderstelde. Alle systeemontwikkelmethoden ten spijt bestaan systemen nog steeds uit programma's en deze programma's zullen goed moeten zijn. Ook automatisch gegenereerde programma's zijn niet beter dan de specificaties die erin worden gestopt.

Waar ligt dan het werkelijke probleem? Het werkelijke probleem blijft het vertalen (simuleren) van de werkelijkheid in (formele) specificaties. Deze formele specificaties waren vroeger vooral de statements in de programma's en zijn nu gedetailleerde functionele ontwerpen. Die programma's en ontwerpen waren en zijn echter een aaneenschakeling van voorbeelden die een gebruiker, ontwerper of programmeur zich reëel kan voorstellen. Het niet-bedachte voorbeeld blijkt echter ook voor te kunnen komen, zodat de atomaire kwaliteit (de kwaliteit van elk afzonderlijk programma) niet echt verbetert. Ook ontwikkelingen om via nieuwe formele specificatietalen het probleem op te lossen, kunnen mank lopen op het (gebrek aan) abstraheringsvermogen van de ontwikkelaar/ontwerper. Na minstens tien jaar ontwikkelingen van 4GL tot CASE-tools moeten we onder ogen zien dat het formele vertaalproces van de werkelijkheid in een geautomatiseerd systeem in wezen deze atomaire kwaliteit van het informatiesysteem bepaalt. En deze kwaliteit wordt nauwelijks verbeterd door betere hulpmiddelen.

Slot

Hoe kunnen we nu deze atomaire produktkwaliteit van informatiesystemen verbeteren?

Er is mijns inziens een aantal aspecten waarmee opleidingsinstituten en bedrijven rekening moeten houden:

- voor de vertaling van de werkelijkheid in eenduidige formele specificaties is kennis van de materie en van formele specificatiemethoden nodig
- om te kunnen abstraheren van enkele voorbeelden naar een formeel model is een formele en analytische denkwijze vereist
- opleidingsinstituten (WO, HBO) zullen in studierichtingen die studenten opleiden voor ontwerper moeten selecteren op de vereiste analytische denkwijze en moeten trainen in het opstellen van formele specificaties
- systeemontwikkelmethoden, zoals SDM, die slechts aangeven welke activiteiten in het ontwikkeltraject uitgevoerd moeten worden, zullen zoals Information Engineering, aangevuld moeten worden met de wijze waarop deze activiteiten moeten worden uitgevoerd. De vrijblijvende invulling, geroemd als flexibiliteit, leidt vaak tot onjuiste (of onvolledige) specificaties
- de methodische begeleider, dit is de vaktechnische evenknie van de projectbeheerder, moet toezien op het juiste gebruik van onder andere de formele specificatiemethode
- onderzoek naar en ontwikkeling van formele specificatiemethoden en vertalers moet worden gestimuleerd en de resultaten moeten in het bedrijfsleven worden geïntroduceerd
- de kloof tussen de werkelijkheid en (het opstellen van) de formele specificaties kan alleen worden overbrugd door de goede mensen met de goede hulpmiddelen (specificatietalen) te laten werken.