

Dertiende en laatste toespraak tot mijn studenten, voorjaar 1976.

Omdat ik in het vijfde jaar geen colleges geef, wordt dit de laatste keer dat ik U toespreek, en ik wil daarom proberen vanuit het perspectief van Uw latere beroepsuitoefening een terugblik op ons onderwijs te laten vallen.

In de opleiding tot wiskundig ingenieur bevindt de informatica zich in een bijzonder gunstige positie: geen van de andere specialisaties is immers zo duidelijk enerzijds een wiskundige aangelegenheid en anderzijds een ingenieursaangelegenheid. Tot de synthese van deze twee componenten hebben de informatici relatief de meeste mogelijkheden, en ik onderschrijf van ganser harte C.A.R.Hoare's uitspraak, dat in geen ingenieursdiscipline het succesvol nastreven van academische idealen zo rijke vruchten afwerpt als in "software engineering". Hier is inderdaad een terrein, dat zo inspirerend is, omdat de overdekking van wat theoretisch gezond en praktisch zinvol is, zich, mits men zich daar met zorg op toelegt, zo schitterend laat uitbreiden.

Toch treft ons het verwijt, dat wij onze studenten niet adequaat op de praktijk voorbereiden, een verwijt, waarvan het de moeite loont om te analyseren wat het voorstelt. Wij ontwikkelen en demonstreren formele methoden, en niet alleen dat, wij propageren zelfs, dat ze in de praktijk gebruikt worden, een praktijk die door de doorsnee programmeur intuïtief benaderd wordt.

Nu geeft de overgang van ambacht tot wetenschappelijke discipline altijd moeilijkheden: de oude gildebroeders voelen zich erdoor bedreigd en neigen ertoe zich tegen de ontwikkeling te verzetten. Enig anti-intellectualistisch tegengas moeten we dus als normaal accepteren.

In de informatica ligt het door externe omstandigheden, die niets met de wetenschap te maken hebben, echter aanzienlijk gecompliceerder. Als programmeurs weten wij uit ervaring en kunnen wij zelfs op grond van simpele analogieën overtuigend aantonen, dat een noodzakelijke voorwaarde voor goede en efficiënte programma's gevormd wordt door de beschikbaarheid van een eenvoudige programmeertaal waarvan syntax en semantiek rigoureuze en non-operationeel gedefinieerd zijn zonder enige verwijzing naar computer of compiler. Zonder zulk gereedschap blijft het modderen.

De complicerende bijkomstigheid is evenwel dat de International Business Machines, de computerfabrikant die een zo sterk overheersende positie op de markt inneemt, niet het minste commercieel belang heeft bij de verbreiding van het gebruik van goedgedefinieerde programmeertalen. In tegendeel zelfs! Bij een goedgedefinieerde programmeertaal zou IBM ten eerste de plicht hebben om hem correct te implementeren, en ten tweede de kans lopen, dat een concurrent het beter zou doen. Bij ontstentenis van scherpe definitie is het begrip "correctheid van een implementatie" echter leeg, en IBM kan hiermee uit de voeten, omdat --wederom wegens de ontstentenis van een scherpe definitie en de overheersende positie van zijn producten-- de IBM-implementatie door het klootjesvolk als de zg. "de facto standaard" wordt geaccepteerd. En vanaf dat moment is per definitie wat IBM doet wel gedaan en mag de concurrentie amechtig achterna hollen, per definitie twee jaar achter. U begrijpt, dat er IBM alles aan gelegen is deze ondoorzichtige toestand te handhaven. U begrijpt, waarom IBM de verbreiding van een zo welgedefinieerde programmeertaal als ALGOL 60 te vuur en te zwaard bestreden heeft, en U mag een keer raden naar de koker waaruit de laster komt, waaraan Niklaus Wirth bloot staat sinds zijn geesteskind PASCAL zich toch verbreidt.

Na U aldus op de praktijk te hebben voorbereid geef ik nog eenmaal er uiting aan, dat ik overigens van oordeel ben, dat er op Hilversum 4 minder moet worden gezeverd.