

## Architectuur is systeem

Roel Wieringa  
Faculteit Informatica  
Universiteit Twente

De vraag of architectuur hetzelfde is als systeemontwerp is voor buitenstaanders wat moeilijk te volgen vanwege de vreemde betekenis die wij software engineers aan het begrip "ontwerp" toekennen. In de software engineering is een systeemontwerp de decompositie van de software in elementen die gezamenlijk de software-functies realiseren. In alle andere ingenieurswetenschappen is ontwerpen het nemen van beslissingen over de gewenste eigenschappen van een product. De gewenste eigenschappen kunnen eigenschappen zijn van de uiterlijke vorm of de innerlijke structuur van een systeem of zelfs over de materialen waarvan het gemaakt is. Het resultaat van de ontwerpactiviteit is een document die die gewenste eigenschappen vastlegt. Dat kan een blauwdruk zijn, of een bouw instructie, een specificatie van onderdelen, etc.

Dit bredere begrip van ontwerp omvat wat wij in de software engineering "requirements" noemen zowel als wat wij tegenwoordig vaak "architectuur" noemen. Kern van het bredere begrip dat ik hier voorstel, is dat ontwerpen probleemoplossen is. Elke ontwerpbeslissing is een beslissing over de eigenschappen die een product moet hebben om een probleem in de uiteindelijke gebruiksomgeving op te lossen. In nog algemenere termen houdt ontwerpen in dat we de slogan "bezint eer ge begint" serieus nemen. Een ontwerper bedenkt wat er gebouwd moet worden; een bouwer bouwt wat er ontworpen is.

De discussies in de software engineering over de "beste" definitie van het begrip "architectuur" geven soms blijk van een gebrek aan datgene wat juist essentieel is voor een goede architectuur: contextgevoeligheid en historisch besef. Een simpele woordenboekdefinitie van het begrip "architectuur" is dat het een constructie is die er uit ziet alsof het geproduceerd is door een bewuste handeling en niet bij toeval ontstaan is (*Webster's Ninth Collegiate Dictionary*). De architect is een ontwerper: hij of zij denkt na over de gewenste producteigenschappen en beslist wat deze eigenschappen zullen zijn, voordat het product gebouwd wordt. Het gebrek aan contextgevoeligheid in de architectuurdiscussie in de software engineering spreekt onder andere uit het feit dat in de software engineering een beperkt ontwerpbegrip gebruikt wordt, wat het zicht ontnemt op het feit dat producteigenschappen gerelateerd moeten worden aan het probleem waarvoor het product een oplossing moet bieden: aan de context van het product dus. Een ontwerper moet beginnen met de context van het ontwerpprobleem te analyseren. Een tweede uitingsvorm van de geringe contextgevoeligheid van de architectuurdiscussie is dat binnen de software engineering een terminologie gebruikt wordt die niet aansluit bij wat er in

aangrenzende ontwerpwetenschappen zoals bijvoorbeeld de werktuigbouwkunde of industrieel productontwerp gebruikelijk is.

Een tweede interessante definitie van "architectuur" die door *Webster's* gegeven wordt, is dat het een unificerende of samenhangende vorm of structuur is. Dit is de essentie van het klassieke systeembegrip: een samenhangend stel van elementen. Door de samenhang is het geheel meer dan de som der delen. Een architect is een ontwerper die de producteigenschappen zodanig kiest dat ze systematisch samenhangen, waardoor er systeemgedrag ontstaat dat het oorspronkelijke ontwerpprobleem oplost. Een architect is dus een systeemontwerper: een ontwerper van samenhang. Het spreekt van gebrek aan historisch besef dat de tientallen jaren oude begrippen uit de systeemleer niet in de architectuurdiscussie gebruikt worden. Zou dat wel gebeuren, dan zouden we wat kunnen leren van de aspecten van systeem (dus de aspecten van architectuur) die reeds 40 jaar geleden relevant geacht werden voor systeemontwerp (dus voor architectuurdefinitie): zoals bijvoorbeeld de sociale, fysische, financiële, juridische, ergonomische, economische, en vele andere aspecten die toen reeds relevant geacht werden, zowel afzonderlijk maar vooral ook in hun samenhang. (Zie bv. A. D. Hall, *A Methodology for Systems Engineering*, Van Nostrand, 1962.) Het vereist geen genie om vast te stellen dat ook het informatie-aspect bij software erg belangrijk is. Maar zien we softwareontwerp als systeemontwerp, dan komen als vanzelf vele andere relevante aspecten van softwarearchitectuur boven water.