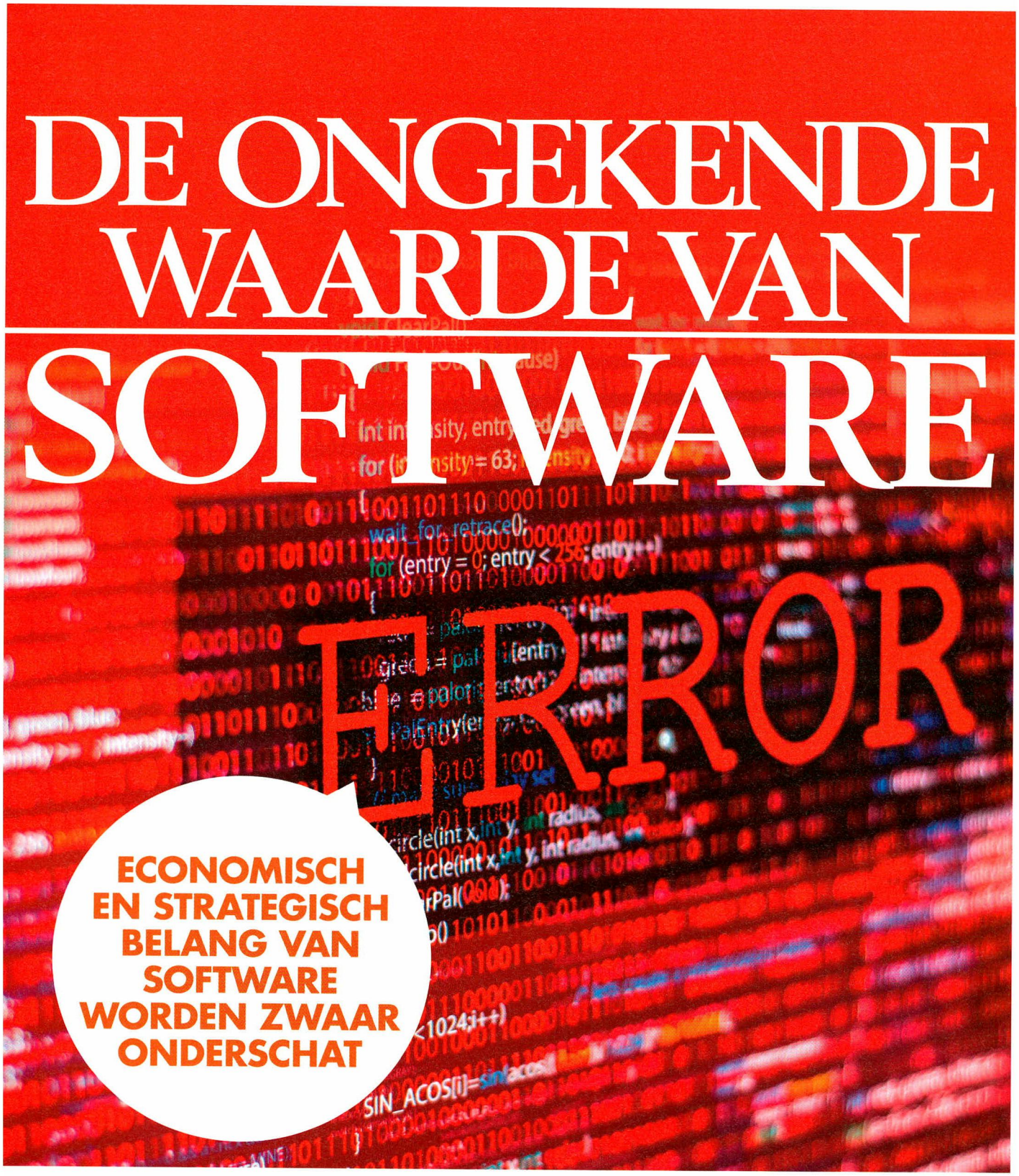
Publicatie : AG Connect
Datum : 31 dec 2017
cm2 : 1.175
Pagina : 62
Advertentiewaarde : € 26.250,00
Regio : Nederland
Frequentie : 10x per jaar
Oplage : 10.000

DE ONGEKENDE WAARDE VAN SOFTWARE



Publicatie : AG Connect
Datum : 31 dec 2017
cm2 : 1.175
Pagina : 62
Advertentiewaarde : € 26.250,00
Regio : Nederland
Frequentie : 10x per jaar
Oplage : 10.000

We zijn ons allemaal bewust van de razendsnelle digitalisering van de samenleving. Maar we zijn verbazingwekkend laconiek over de kwaliteit van de software die schuilgaat achter die digitalisering. Software moeten we op waarde leren schatten, zegt Jos Baeten: als volwaardig onderzoeksgebied en als belangrijke, economische drijfveer.

door Jos Baeten beeld Shutterstock

WE ZIJN STEEDS MEER AFHANKELIJK VAN SOFTWARE. Neem bijvoorbeeld onze infrastructuur, onze financiële transacties en communicatiekanalen – voor al deze diensten is software onontbeerlijk. We vertrouwen dagelijks op navigatieapps die ons de weg wijzen, op de veiligheid van internetbankieren en op een constante, toegankelijke informatiestroom op het web. Maar is dat vertrouwen wel gerechtvaardigd? Een hoge kwaliteit van software achter al deze systemen is van ongekend belang, maar die kwaliteit krijgt vaak niet de aandacht die het verdient.

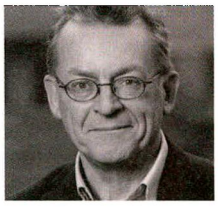
ALGOL 68

Bijna vijftig jaar geleden, in 1968, maakte software een belangrijke stap naar volwassenheid. Aad van Wijngaarden, een van mijn voorgangers als CWI-directeur, stond toen aan het roer van het ontwerp van de programmeertaal Algol 68. Algol 68 was de eerste programmeertaal met een formele syntax en semantiek. Dat wil zeggen: de regels die bepalen hoe programmeurs zinnen in deze taal formuleren, en de interpretatie van de gemaakte zinnen lagen voor het eerst op een wiskundige manier vast. Daardoor was de taal eenduidig. Er was geen ruimte meer voor verschillende interpretaties. Het formaliseren van Algol 68 had grote gevolgen: het opende de deuren om wiskunde en logica te gebruiken om over programma's in deze taal te redeneren. Sterker nog, met Algol 68 konden wiskundigen voor het eerst de juistheid van programma's bewijzen. Dat was een grote mijlpaal. Het maakte software tot een volwaardig, volwassen onderzoeksonderwerp. Het nieuwe vakgebied van formele methoden was geboren.

We kunnen ons geen fouten in de code permitteren

Dat onderzoeksgebied, waarin wetenschappers dus wiskunde en logica gebruiken om de werking van software te specificeren en de juistheid van programma's te verifiëren, is ook nu nog ongekend belangrijk. Neem bijvoorbeeld een recente doorbraak. In 2015 repareerden CWI-onderzoekers een bug in TimSort, een standaardsorteeralgoritme waarvan onder andere Java, Python en Android gebruikmaken. Er zijn miljarden computers, cloudservices en telefoons die TimSort gebruiken. De bug, die al een lange tijd bekend was, veroorzaakte dat programma's af en toe crashten. Met behulp van formele methoden wisten de onderzoekers de juistheid van een nieuwe, bugvrije versie van TimSort te bewijzen. Inmiddels hebben Python en Android deze verbeterde versie doorgevoerd.

AUTEUR



JOS BAETEN (jos.baeten@cwi.nl) is algemeen directeur van het Centrum Wiskunde & Informatica (CWI) in Amsterdam. Daarnaast is hij parttimehoogleraar Theory of Computing aan de Universiteit van Amsterdam (UvA).

Publicatie : AG Connect
Datum : 31 dec 2017
cm2 : 1.175
Pagina : 62
Advertentiewaarde : € 26.250,00
Regio : Nederland
Frequentie : 10x per jaar
Oplage : 10.000

> "Nederland is altijd sterk geweest in het maken van innovatieve software."

Het is natuurlijk slechts een voorbeeld, maar dit werk onderstreept het belang van formele methoden: het is de meest grondige, fundamentele manier om de juistheid van software te garanderen. Daarbij is het vakgebied een kracht van de Nederlandse wetenschap. Vijftig jaar geleden waren Aad van Wijngaarden en Edsger Dijkstra voorlopers, maar het is nog steeds een gebied waarop we wereldwijd op topniveau meespelen.

HOOGMOED

Er zijn dus manieren om softwarekwaliteit te garanderen, of althans om dat te proberen. Toch gaat het in de praktijk vaak anders. Er is veel software in omloop die vol zit met bugs. Sterker nog, het is eerder regel dan uitzondering dat een bedrijf software uitbrengt waarvan bekend is dat er nog verschillende iteraties nodig zijn voordat het bugvrij is. Denk bijvoorbeeld aan de regelmatige software-updates van smartphones. Gaandeweg passen bedrijven hun software aan om zo veelvoorkomende crashes op te lappen. Dat is logisch vanuit businessperspectief, want zo kan het product vroeg de markt op. Maar ik ben van mening dat we hier niet te lichtzinnig mee om moeten gaan. Want wanneer burgers echt afhankelijk zijn van software, kunnen we ons geen fouten in de code permitteren.
Een bekend historisch voorbeeld dat illustreert hoe het verkeerd kan gaan, is het bestralingsapparaat Therac-25, dat Canadese artsen gebruikten voor de behandeling van kankerpatiënten. Programmeerfouten in de Therac-25 leidde ertoe dat tussen 1985 en 1987 verschillende patiënten een veel te hoge stralingsdosis kregen, in een aantal gevallen zelfs met de dood tot gevolg. Een ander bekend voorbeeld is de raket Ariane 5, die tijdens haar proefvlucht in 1996 explodeerde. Het ongeluk bleek te wijten aan een softwarefout: doordat programmeurs een verkeerd gegevenstype gebruikten, ging de conversie van een getal mis – een getal dat de horizontale snelheid van de raket bepaalde. De hoogmoed van programmeurs kwam, in dit geval letterlijk, voor de val.
En ook vandaag de dag gaat het nog regelmatig mis. Recentelijk, in november 2017, bleek bijvoorbeeld dat er 280 miljoen dollar aan digitale valuta was bevroren. Een ontwikkelaar bij het bedrijf Parity had per ongeluk een library gewist, waardoor de digitale portemonnees van gebruikers niet meer toegankelijk waren. Dit was niet de eerste keer dat een fout in de code digitale valuta bevroor of zelfs deed verdampen. Dus ook de financiële sector is gevoelig voor softwarefouten.
Het moge duidelijk zijn dat zorgvuldigheid met software geboden is. Nu meer dan ooit. Inmiddels is software namelijk zo doorgedrongen in onze samenleving, dat kleine missers in een code soms letterlijk levensgevaarlijk zijn. Denk bijvoorbeeld aan onze treinen, die vertrouwen op digitale verkeersregelaars. Of aan alle medische apparatuur in ziekenhuizen. Of, in de nabije toekomst, aan zelfrijdende auto's. Wanneer je je leven toevertrouwt aan zulke systemen, moet je er absoluut op kunnen rekenen dat zij functioneren.

ECONOMISCH BELANG

Niet alleen het belang van softwarekwaliteit is onderbelicht. Ik ben van mening dat ook het economisch en strategisch belang van software zwaar onderschat wordt. Historisch gezien is Nederland een land dat altijd al sterk is geweest in het

OVERHEID EN ICT

Om grote missers te voorkomen, ondergaan grote ICT-projecten van de rijksoverheid sinds 2015 een toetsing door een onafhankelijk bureau, BIT (Bureau ICT-Toetsing). Onlangs is een project dat al jarenlang liep – het invoeren van een nieuw bevolkingsadministratiesysteem – naar aanleiding van BIT-advies beëindigd. BIT concludeerde dat het simpelweg niet haalbaar was een systeem te bouwen met de benodigde functionaliteiten voor zoveel gebruikers.
De kosten zouden de pan uit rijzen. Het BIT identificeerde daarnaast een significant risico: dat het project opnieuw vertraagd zou raken of zelfs zou vastlopen. De conclusie: de voordelen van een gecentraliseerd systeem, ter vervanging van de huidige gemeentelijke bevolkingsregisters, wogen niet op tegen de kosten en de risico's. Dergelijke kritische softwareadviezen – weliswaar niet van onafhankelijke partijen – kunnen het risico op onnodige kosten flink verkleinen.

Publicatie : AG Connect
Datum : 31 dec 2017
cm2 : 1.175
Pagina : 62
Advertentiewaarde : € 26.250,00
Regio : Nederland
Frequentie : 10x per jaar
Oplage : 10.000

ITERATIEF ONTWERPEN

Het bouwen van software gaat regelmatig mis omdat er alleen in het beginstadium een inventarisatie plaatsvindt onder gebruikers, terwijl eisen en omstandigheden tijdens het bouwen van de software veranderen. Een softwareontwerp zou dynamischer moeten zijn, als een iteratief proces, zodat ontwerpers er zeker van kunnen zijn dat hun eindresultaat past bij de gebruikers.

maken van innovatieve software. En Nederlandse bedrijven wisten hun softwarediensten wereldwijd te verkopen. Maar tegenwoordig maken we in Nederland regelmatig softwareproducten die we vervolgens al snel overdragen aan de Verenigde Staten of aan landen in Azië. Denk bijvoorbeeld aan het van origine Nederlandse booking.com. In 2015 kocht The PriceLine Group deze grote boekingssite. In hetzelfde jaar kocht het Amerikaanse Tripadvisor de Nederlandse restaurantsite Iens. Nederland maakt dus nog steeds sterke, waardevolle producten, maar we geven die vervolgens uit handen. Dat is een aderlating, gezien het strategisch en economisch gewicht van zulke producten. We maken ons op deze manier steeds meer afhankelijk van slechts enkele digitale grootmachten. Dat kan en moet anders. Zo stimuleert bijvoorbeeld China voornamelijk Chinese producten en diensten, zoals het e-commercebedrijf Alibaba en het technologiebedrijf Tencent.

Ik ben van mening dat ook Nederland, en in bredere zin Europa, zou moeten investeren in software. We moeten sterke opensourceproducten maken en de softwarediensten wereldwijd verkopen. Op die manier kunnen we waardevolle innovaties behouden, persoonlijke data zelf blijven beheren en de controle op onze IT-infrastructuur in eigen handen houden. Om dat te realiseren, moeten er meer middelen komen om een softwareprototype door te ontwikkelen tot een volwaardig, bruikbaar product. Dit is een vorm van wetenschapsvalorisatie, met potentieel veel impact, waar momenteel weinig aandacht voor is. Er is in de wetenschap vrijwel geen financiering beschikbaar voor softwareonderzoekers om de 'last mile' tot een product af te kunnen leggen. De afgelopen jaren heeft het CWI tweemaal zo'n doorontwikkeling kunnen steunen, maar dat vergt flinke investeringen. Daar is niet altijd ruimte voor. Er zou daarom een breder gedragen beleid voor moeten komen, waarin de Nederlandse overheid een duidelijke rol speelt. Alleen op die manier kunnen we garanderen dat Nederland zijn positie als software-exporterend land niet verliest.

FUNDAMENT

Terugblikkend kunnen we concluderen dat software in vijftig jaar tijd veel verschillende rollen heeft gekend: van jong, onontgonnen, vakgebied tot een bepalende factor in de wereld. Ook de komende vijftig jaar zal dat zeker zo blijven. De wereld is digitaal en wordt alleen maar digitaler. Dat voorzien we allemaal. Software is daarbij een onmisbare bouwsteen. En als we willen blijven bouwen, moeten we ons sterk maken voor een fundament van de hoogste kwaliteit.

REACTIES EN BIJDRAGEN

Voor reacties en nieuwe bijdragen van IT-experts:
Henk Ester
020-2356415
h.ester@agconnect.nl