

Clive Thompson

DE CODERS

**< Een kijkje in het hoofd van
programmeurs – de machtigste
beroepsgroep ter wereld >**

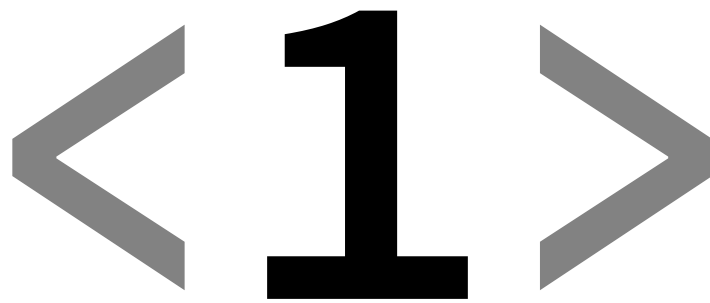
Vertaald uit het Amerikaans door Piet Dal,
Nannie de Nijs Bik-Plasman en Marianne Palm

MAVEN
PUBLISHING

Inhoud

<1>	De software-update die de werkelijkheid heeft veranderd	7
<2>	De vier golven coders	35
<3>	Voortdurende frustratie en uitbarstingen van vreugde	69
<4>	Te midden van INTJ's	99
<5>	De efficiencycultus	129
<6>	10X'ers, supersterren en de mythe van meritocratie	161
<7>	De ENIAC-meisjes verdwijnen	199
<8>	Hackers, crackers en vrijheidsstrijders	243
<9>	Komkommers, Skynet en de opkomst van AI	281
<10>	Schaalgrootte, trollen en Big Tech	321
<11>	De softwarearbeider	361
	Woord van dank	397
	Noten	400

01001000 01101111 01101111
01100110 01100100 01110011
01110100 01110101 01101011



De software-
update die de
werkelijkheid
heeft veranderd

8 In de kleine uurtjes van 5 september 2006 herschreef Ruchi Sanghvi de wereld met één enkele software-update.

Sanghvi, een openhartige programmeur met een rond gezicht, was 23 jaar toen ze bij Facebook ging werken. Ze was opgegroeid in India en had er lang van gedroomd om te gaan werken in haar vaders bedrijf, dat zware machinerieën verhuurde voor de bouw van havens, olieraffinaderijen en windmolens. Tijdens haar studie aan Carnegie Mellon University raakte ze echter geboeid door computertechniek en werd er verliefd op. Het was alsof je voortdurend puzzels zat op te lossen: zoeken naar een manier om een algoritme sneller te laten functioneren, om een stom stukje code dat niet goed werkte te debuggen. Dit mentale schaakspel nestelde zich in haar brein en ze merkte dat ze de hele dag liep na te denken over programmeerproblemen. ‘Je bent er uren mee bezig, je eet niet, je slaapt niet, je kunt het niet loslaten,’ vertelde ze me.

Sanghvi was, naar programmeermaatstaven, een laatbloeier. Haar studiegenoten waren kinderen, bijna allemaal jongens, die al sinds ze negen waren programmeerden en videogames speelden, en ze leken het zonder enige moeite te snappen. Maar ze hield vol, haalde goede cijfers, studeerde af en kreeg haar eerste baan in

Manhattan, waar ze zich bezighield met het schrijven van wiskundige modellen voor een bureau dat in derivaten handelde.

Toen ze in New York aankwam, schrok ze ontzettend van al die grijze kantoorhokjes op haar werkplek. Hier zou ze niet echt haar stempel op de wereld kunnen drukken. Ze wilde geen radertje in het grote geheel zijn dat code schreef om financiële activiteiten te ondersteunen, maar wilde wel dolgraag bij een bedrijf werken waar de technologie zelf het kernproduct was, waar computerwetenschappers de hoofdrol speelden. Ze wilde eigenlijk iets maken wat zou worden gebruikt, iets tastbaars, iets nuttigs. Ze wilde iets Facebookachtigs doen, een site waar ze zich het laatste jaar van haar studie bij had aangemeld. Dát was pas een verslavend stukje software. Ze logde de hele tijd in om in contact te blijven met haar pas afgestudeerde studievrienden en keek telkens op hun pagina of er updates waren.

Dus Sanghvi zei haar baan op nog voor haar eerste werkdag om was en vertrok uit Manhattan. Ze vluchtte naar San Francisco, waar ze een baan kreeg bij Oracle, een bedrijf dat databasesyste-
men ontwierp. En toen werd ze op een dag door een studievriend uitgenodigd om eens bij Facebook zelf langs te komen.

9

Het was een piepklein bedrijf dat alleen voor studenten was bedoeld. Gewone mensen mochten Facebook niet gebruiken. In het kantoor op de eerste verdieping, boven een Chinees restaurant, trof ze een stel voornamelijk jonge blanke mannen, van wie er een paar onlangs waren gestopt met hun studie aan Harvard: de 21-jarige Mark Zuckerberg liep er rond op bijna versleten sandalen, Adam D'Angelo (de gozer die ooit een jongere Mark Zuckerberg een beetje had leren programmeren) en Dustin Moskovitz, Zuckerbergs huisgenoot op Harvard. Ze werkten in een waas van diepe concentratie, met opengeklapte laptops op bureaus vol rommel, ook als ze videogames speelden in hun vlakbij gelegen studentikoze noodslaapplaatsen of lagen te zonnen op het dak van het kantoor van Facebook. De graffitikunstenaar David Choe werd in die tijd aangenomen om de muren te versieren met schilderijen, waaronder 'een reusachtige weelderige vrouw met enorme borsten die een pak in Mad Max-stijl droeg en een bulldog

bereed' (zoals Ezra Callahan, een medewerker uit de begintijd, het omschreef).¹

Ze waren constant bezig Facebook te optimaliseren en te veranderen, en stuurden geregeld nieuwe code naar gebruikers om features als Facebooks beroemde 'porren' te installeren of een 'notities'-app waarmee langere berichten konden worden geschreven. Ze waren waaghalzen, soms was een nieuwe feature zo enthousiast en haastig geschreven dat hij onverwachte neveneffecten had, waar ze pas achter kwamen als, oeps, de code al op de site in werking was. Dan rolden ze de code om middernacht uit en wachtten gespannen af of Facebook zou crashen of niet. Als alles werkte, gingen ze weg. Als het een ramp veroorzaakte gingen ze als gekken aan de slag om het te herstellen en zwoegden ze vaak tot vroeg in de morgen door. Soms zetten ze alles weer terug naar de oude situatie als ze de nieuwe feature gewoonweg niet aan de gang kregen.² Zuckerbergs veelgeciteerde motto was dan ook: 'Move fast and break things.'³

10 Sanghvi vond het geweldig. 'Het was anders, het was opwindend, het lééfde,' zei ze. 'Iedereen daar leek op volle toeren te draaien, iedereen had het echt druk, iedereen ging helemaal op in waar hij mee bezig was... de energie was zo tastbaar.' En Facebook bleek wanhopig op zoek naar meer *coders*. Je kunt het je nu bijna niet voorstellen, nu het bedrijf zo'n wereldwijd opererende kolos is, maar in 2005 had het moeite om werknemers aan te trekken. De meeste ervaren software-engineers in Silicon Valley meenden dat Facebook een hype was, zo'n webverschijnsel dat kortstondig uit-zinnig populair is voordat het vreselijk passé wordt. Ze waren niet geïnteresseerd in een baan daar. Sanghvi kwam er dus op een heel gunstig moment: jong genoeg om Facebook te hebben gebruikt en te weten hoe verslavend het was, maar oud genoeg om te zijn afgestudeerd en op zoek te zijn naar een baan als programmeur. Een week na haar bezoek, werd ze aangenomen als de eerste vrouwelijke software-engineer van het bedrijf.

Ze kreeg al snel een belangrijke taak. Zuckerberg en de andere oprichters vonden dat Facebook te langzaam en te gebruiksonvriendelijk was. In die begindagen kon je alleen weten waar je vrienden mee bezig waren door op hun Facebookpagina te kijken.

Je moest er dus eerst actief aan denken. Als iemand een sappig stukje informatie postte – iets over een net beëindigde relatie, een roddeltje, een pittige profielfoto – zou je dit niet zien als je vergat om die dag zijn pagina te bekijken. Facebook was eigenlijk zoiets als wonen in een flatgebouw waar je je neus telkens bij iemand binnen moest steken als je wilde weten wat er aan de hand was.

Zuckerberg wilde de boel stroomlijnen. Hij liep rond met een notitieblok waarin hij (in zijn kleine, precieze handschrift) een idee had geschetst voor een ‘nieuwsoverzicht’. Zodra je inlogde, kreeg je een overzichtspagina met een lijst van dingen die je vrienden hadden gepost sinds de laatste keer dat je had ingelogd. Het was een soort ESP voor je sociale leven. Zodra iemand een update postte – ping! – verscheen die aan de rand van je blikveld. Het nieuwsoverzicht was niet slechts een kleine cosmetische aanpassing van Facebook, zoals een mooi nieuw lettertype of een nieuwe kleur. Het zou de manier waarop we aandacht voor elkaar hadden reorganiseren.

En Sanghvi moest dit nieuwsoverzicht ontwerpen. Ze ging aan het werk met een kleine groep medewerkers, onder wie Chris Cox, Matt Cahill, Kang-Xing Jin (die KX werd genoemd) en Andrew ‘Boz’ Bosworth, Zuckerbergs vroegere student-assistent op Harvard. Ze werkten er negen maanden heel hard aan. Eerst bespraken ze uitgebreid ideeën en daarna sloegen ze aan het programmeren, en al die tijd knalde James Brown of Johnny Cash uit de laptop van Cox. Net als de andere coders ging Sanghvi bijna de klok rond programmeren, bleef ze tot het ochtendgloren bij Facebook en ging dan slaapdronken naar huis in San Francisco. Nadat ze bijna haar auto in de prak had gereden door slaapegebrek, verhuisde ze naar een huis vlak bij het kantoor van Facebook, van waaruit ze soms in haar pyjama naar haar werk liep. Dat kon niemand iets schelen. Bij alle programmeurs liepen sociaal leven en werk dooreen, en ze speelden poker of videogames op het werk. In een video-interview in 2005 zat Zuckerberg te praten met een biertje in een rode plastic beker in zijn hand, en maakte een werknemer een handstand op een biervat en dronk er tegelijkertijd uit.⁴

Het was een jongensclubje, maar voor Sanghvi was dat niets nieuws. Zo was het altijd geweest in de wereld van de computer-

wetenschappen die zij kende. Er zaten maar een paar vrouwen in haar 150 studenten tellende klas. Ze had geleerd om terug te schreeuwen als anderen begonnen te schreeuwen, wat in een vertrek vol zelfingenomen jonge mannen vaak het geval was. Als je als vrouw luidkeels van je liet horen, lokte dat wel reacties uit. ‘Iedereen noemde me superagressief,’ zei ze, ‘en dat deed pijn. Ik vind mezelf niet agressief.’

Maar ze liet zich er niet door afleiden, zwoegde op de code, omdat vooral die haar interesseerde, en het was ontzettend leuk, bizar en moeilijk. Om het nieuwsoverzicht te kunnen maken, moesten zij en haar collega’s zich ook verdiepen in pittige filosofische vragen over vriendschap, zoals ‘Wat voor soort nieuws willen vrienden van elkaar weten?’ Het overzicht kon niet de hele dag door alles laten zien wat elke afzonderlijke vriend van je deed. Als je tweehonderd vrienden had die ieder tien dingen postten, waren dat tweeduizend items. Niemand had tijd om die allemaal te bekijken. Sanghvi en de programmeurs moesten dus een stel regels maken die ieders overzicht zouden filteren, er een bepaald ‘gewicht’ aan zouden geven, een getal dat iets als meer of minder interessant zou rangschikken. ‘Welk gewicht zou jij toekennen aan de relatie tussen twee mensen,’ vroegen ze elkaar als ze ’s avonds laat in het kantoor van Facebook zaten. ‘Welk gewicht zou jij toekennen aan de relatie tussen een mens en een foto?’

12

Halverwege 2006 hadden ze een prototype aan het werk. Op een avond zat Chris Cox thuis te kijken naar het allereerste bericht van het nieuwsoverzicht dat knipperend tot leven kwam: **MARK HEEFT EEN FOTO TOEGEVOEGD**. (‘Het was het Frankenstein-moment waarop de vinger bewoog,’ grapte hij later.) Tegen het einde van de zomer werkte het nieuwsoverzicht soepeltjes genoeg om op het publiek te worden losgelaten. Sanghvi schreef een openbaar bericht met de titel ‘Facebook Gets a Facelift’ om het product wereldkundig te maken. ‘Het geeft je de hele dag door een gepersonaliseerd overzicht van nieuwtjes, zodat je weet wanneer Mark Britney Spears bij zijn favorieten heeft geplaatst of wanneer degene op wie je smoorverliefd bent weer single is. Dus telkens wanneer je inlogt, krijg je de laatste hoofdpunten van de activiteiten van je vrienden en sociale groepen te zien,’ legde Sanghvi uit.

De veranderingen, zo schreef ze, zouden ‘heel anders zijn dan alles wat je op het web kon vinden’.⁵

Kort na middernacht stuurden Sanghvi en haar collega’s de update de wereld in. Het nieuwsoverzicht was er, de teamleden trokken wat flessen champagne open en omhelsden elkaar. Juist vanwege zulke momenten was ze in de computerwereld gestapt, om code te schrijven die ons dagelijks leven verandert.

Maar er was één probleem: de gebruikers hadden er een hekel aan.

Toen Sanghvi en het team in die kleine uurtjes de update hadden uitgerold, hadden ze zich rond een laptop op haar bureau verzameld in afwachting van het commentaar van de gebruikers. Ze hurkte op de grond terwijl Zuckerberg, in een rood CBGB-T-shirt, naar het scherm zat te staren. Haar collega KX stond achter Zuckerberg en torende boven hem uit. Iedereen stond te stuiteren van opwinding. ‘Ze dachten dat ze goed nieuws te horen zouden krijgen,’ zei Zuckerberg later.

Dat was niet zo. ‘Dit is drie keer niks’ was zo’n typisch commentaar dat over het scherm rolde. De gebruikers kwamen massaal in opstand, vele dreigden van Facebook te gaan of het te boycotten. Er hadden zich groepen gevormd met een naam als ‘Ruchi is de Duivel’. Eén student, Ben Parr, had een Facebookgroep gemaakt die ‘Studenten tegen Facebooks nieuwsoverzicht’ heette en in amper een dag meer dan 250.000 leden kreeg.

Waar hadden ze nu precies zo’n hekel aan? ‘Er zijn maar heel weinig mensen die willen dat iedereen automatisch een update krijgt van wat we doen,’ legde Parr uit. ‘Het nieuwsoverzicht is te griezelig, te stalkerachtig.’⁶ Natuurlijk, Facebook was traag en inefficiënt geweest, zoals Zuckerberg had opgemerkt, maar de gebruikers van Facebook waren blijkbaar op die inefficiëntie gaan vertrouwen. Daardoor hadden ze een klein, aangenaam stukje voor zichzelf gehad. Ze konden een nieuwe profielfoto plaatsen, besluiten dat die niet mooi was en hem een paar minuten later snel weer omruilen voor de oude foto in de wetenschap dat waarschijnlijk niet veel vrienden de verandering hadden gezien. Maar nu was daar het nieuwsoverzicht dat als een brutale, bemoeizieke robot elke post van je van de daken schreeuwde. ‘Hé, Rita

heeft het uitgemaakt met Jeff! Ze is weer single! Kijk zelf maar!’

De programmeurs hadden gelijk. Hun uitvinding had inderdaad de manier waarop iemand iets hoorde uit zijn sociale kring veranderd. De gebruikers waren er echter nog niet over uit of zij wel zaten te wachten op zulke snelle en ingrijpende upgrades voor hun aandachtssysteem.

Het tumult nam in de loop van de dag nog toe en de dag erna bivakkeerden demonstrerende studenten voor het kantoor van Facebook, waardoor Sanghvi en haar collega’s via de achterdeur naar binnen of naar buiten moesten glippen. Online was het nog erger. Minstens een miljoen Facebookgebruikers – 10 procent van het totale aantal – was lid geworden van Facebookgroepen die eisten dat het nieuwsoverzicht zou worden uitgezet.

14 De staf discussieerde over wat ze moesten doen. Er ontstonden twee facties, de ene was ervoor om het nieuwsoverzicht te sluiten, de andere groep betoogde dat het een kwestie van wennen was. Zuckerberg zat in het tweede kamp. Hij meende dat de gebruikers het leuk zouden vinden als het eerste schokeffect zou zijn weggeëbd. Sanghvi was het hier helemaal mee eens, al moest ze toegeven dat ze deels ook vasthield aan het nieuwsoverzicht vanuit haar gevoel van trots dat ze een van de makers was. ‘Ik ben hier negen maanden van mijn leven mee bezig geweest, en ik ga het verdomme niet weggooien,’ zei ze.

Zuckerbergs visie won die dag, maar hij gaf wel toe dat ze te haastig waren geweest en dat ze hun woedende gebruikers halverwege moesten tegemoetkomen. Daarom broedden de ICT’ers van Facebook een plan uit om wat extra privacy-instellingen te maken waarmee Facebookgebruikers konden voorkomen dat gevoelige updates in het nieuwsoverzicht zouden verschijnen. Na een haastklus van 48 uur ging die privacycode online. Zuckerberg plaatste een openbaar excuus op Facebook. ‘We hebben hier echt een bende van gemaakt,’ erkende hij. ‘[We] hebben niet goed uitgelegd wat de nieuwe features inhielden en wat nog erger was: we hebben jullie er niet zelf de controle over gegeven.’⁷ Toch was hij er nog steeds van overtuigd dat het nieuwsoverzicht op de lange termijn een succes zou worden.

Hij had gelijk. Het overzicht was verwarrend en schokkend,

maar het was ook fascinerend. Er school, zo bleek, een enorme waarde in dat dagelijkse krantje met de activiteiten van je vrienden. Als je het overzicht bekeek en de statusupdates zag langskomen, dan kreeg je een genuanceerd beeld van wat er in het leven van je vrienden speelde. Sterker nog, de dag nadat het nieuws-overzicht was toegevoegd, merkten Sanghvi en de rest van het team dat de gebruikers twee keer zo lang op Facebook zaten als daarvoor. Ook werden er veel sneller groepen gevormd. Dat was logisch. Als je kon zien dat je vriend lid werd van een groep met een bepaald politiek doel of van een fangroep van een band, ging je er zelf misschien ook over nadenken of je dat ook moest doen. Dat de 'Ik haat het nieuws-overzicht'-groepen zo hard hadden kunnen groeien, was ironisch genoeg helemaal te danken aan het feit dat ze putten uit de kracht van het overzicht. (En er ontstonden niet alleen flauwe groepen. Heel kort nadat het nieuws-overzicht was gelanceerd, was de op een na grootste groep gewijd aan het onder de aandacht brengen van de genocide in Darfur, en pleitte de op drie na grootste groep voor meer borstkankeronderzoek.)

15

Je zou zelfs kunnen zeggen dat het nieuws-overzicht uiteindelijk het stuk computercode was met de meest verstrekkende gevolgen van de afgelopen twintig jaar. De gevolgen ervan zijn overal te zien, dwars door de patronen van ons leven heen. Facebookgebruikers zien dat hun vrienden een kind hebben gekregen, ze zien snapshots van hun kantoorhokjes en vakanties, ze lezen zo af en toe een grap en klikken op links naar kattenmemes. Door de enorme, gedeelde aandacht voor hetzelfde onderwerp is het nieuws-overzicht een van de betrouwbaarste kompassen om een stuk cultuur viral te laten gaan, van een sentimenteel verhaal over een goede daad tot een fragment van Beyoncé, van het hoopvolle, pro-democratische begin van de Arabische Lente tot kwaadaardige wervende video's van IS. Het nieuws-overzicht verbond mensen en zette aan tot een heleboel tot acroniem gereduceerde kwalen uit de populaire psychologie, van TMI tot FOMO.

Door het overzicht gingen gebruikers ook veel op Facebook kijken, gemiddeld 35 minuten per dag per Amerikaan.⁸ Hoe dat komt, is niet zo moeilijk te begrijpen. Het algoritme dat het overzicht rangschikt is zo ontworpen dat het je meer geeft van wat je

leuk vindt. Het houdt heel goed in de gaten wat je allemaal doet op Facebook – je likes, je reposts, je commentaren – om je daarvoor beter nieuwe items te kunnen laten zien die, zo hopen de programmeurs, keurig aansluiten bij jouw voorkeuren.⁹ Iemand voor het grootste deel voorschotelen wat hij wil zien, betekent uiteraard een geweldig verdienmodel, en daarmee verdiende Facebook in 2017 ongeveer 40 miljard dollar aan advertenties.¹⁰ Dat iemands aandacht in de eigen bubbel werd geconcentreerd door het overzicht van Facebook, leidde echter ook tot verontrustende neveneffecten. Hierdoor ontstond een belangrijke zwakke plek in de maatschappelijke discussie. Het nieuwsoverzicht bleek een bijzonder doeltreffend middel om verkeerde informatie of geruchten te verspreiden, of om op te roepen tot haat. Journalisten ontdekten dat er bij de Amerikaanse presidentsverkiezingen eind 2016 en in Trumps eerste jaar als president allerlei vileine krachten – van blanke racisten tot clickbait-handelaren in politieke desinformatie – vrolijk het overzicht misbruikten door het te voeden met verhalen die waren geschreven om politieke hysterie op te wekken. Erger nog, zeer waarschijnlijk vergrootte het overzicht de kloof tussen de verschillende groeperingen in Amerika, omdat het zo was ontworpen dat nieuws dat niet aansloot bij wat je al had ‘geliket’ er voor het grootste deel werd uitgefilterd.¹¹

In februari 2017 leek zelfs Zuckerberg zich af te vragen wat voor creatuur hij in het leven had geroepen. Hij schreef een notitie van 5700 woorden, die aandeed als een indirecte en defensieve verontschuldiging voor de rol van Facebook in de huidige politieke schisma's. 'Het is de taak van Facebook om mensen te helpen de grootst mogelijke positieve invloed uit te oefenen door daar waar technologie en sociale media verdeeldheid en isolement kunnen vergroten de risico's daarop te verkleinen,' schreef hij.¹²

Dat is een merkwaardig voorzichtige doelstelling: door daar waar technologie en sociale media verdeeldheid en isolement kunnen vergroten de risico's daarop te verkleinen. Een heel wat meer weloverwogen uitspraak dan 'Move fast and break things'. Je zou het misschien kunnen lezen als een stille erkenning van het feit dat sommige dingen beter niet kunnen worden gebroken.

‘Software eet de wereld op,’ zoals de durfkapitalist Marc Andreessen verkondigde.¹³

Dat klopt. Je gebruikt software op vrijwel elk moment dat je wakker bent, bij voor de hand liggende zaken als je telefoon, je laptop, e-mail, sociale netwerken, videogames en Netflix, de manier waarop je een taxi en eten bestelt. Maar overal om je heen is ook allerlei minder evidente software onzichtbaar aanwezig. Bij de totstandkoming van bijna elk papieren boek of pamflet dat je aanraakt, is software gebruikt. Het remsysteem van je auto wordt ondersteund door software, ‘zelflerende’ algoritmes van je bank houden je aankopen nauwlettend in de gaten zodat eerder kan worden ontdekt of een crimineel je bankpas misbruikt en spullen aanschaft met jouw geld.

Het is misschien zo zonneklaar dat het belachelijk klinkt, maar al die stukjes software zijn door een programmeur geschreven, door iemand als Ruchi Sanghvi of Mark Zuckerberg. De kans is groot dat degene die oorspronkelijk het product bedacht een coder was. Programmeurs proberen dag in dag uit computers nieuwe dingen te laten doen, en begrijpen dus vaak heel goed de ‘wat als’-vragen die door computers worden mogelijk gemaakt. (Wat als je een computer zou hebben die elk woord dat je typt op de achtergrond stil en voortdurend en automatisch zou vergelijken met een lijst met veelvoorkomende Nederlandse woorden? Hallo, spellingscontrole!) Soms is het net of de software die we gebruiken er zomaar vanzelf is, zoals het gras dat in je tuin groeit. Maar dat is niet zo. Hij is gemaakt door iemand die – in code – een lange, zorgvuldig samengestelde reeks instructies heeft uitgeschreven die de computer precies vertellen wat hij moet doen, stap voor stap, om een bepaalde taak uit te voeren. Rond het woord ‘algoritme’ wordt een haast religieus mysterie gecultiveerd, maar een algoritme bestaat alleen uit instructies: doe dit, dan dit, daarna dit. Het nieuwsoverzicht is nu een buitengewoon ingewikkeld algoritme met onder meer wat getraind zelflerend vermogen, maar ook dit is als puntje bij paaltje komt nog steeds niet meer dan een lijst met regels.¹⁴ Die regelmakers zijn dus machtig. Sterker nog, tegenwoordig zijn de oprichters van hightechbedrijven – degenen die bepalen welke

producten worden gemaakt, welke problemen worden opgelost en wat überhaupt een probleem is – steeds vaker techneuten, mensen die ervaring hebben opgedaan door eindeloos codes te schrijven en het prototype van hun nieuwe bedrijf zelf in elkaar hebben geflanst.

Programmeurs behoren dus tot de meest onopvallende invloedrijke personen ter wereld. Wij wonen in een wereld die is gemaakt van software en zij zijn de architecten ervan. De beslissingen die zij nemen sturen ons gedrag. Wanneer zij iets makkelijker maken om te doen, doen we het veel vaker. Als zij iets moeilijker of onmogelijk maken om te doen, doen we het minder vaak. Toen coders eind jaren negentig en begin deze eeuw de eerste tools creëerden om te bloggen, volgde er een explosie van zelfexpressie. Toen het opeens makkelijk werd om iets te publiceren, gingen miljoenen mensen dit doen. En toen programmeurs in ongeveer dezelfde periode tools bedachten om bestanden te delen, ging er een rilling door de entertainmentindustrie, die haar greep op de distributie opeens zag verdwijnen. Ze ging in de tegenaanval en nam zelf programmeurs in dienst die software moesten schrijven voor het 'beheren van digitale rechten'. Die werd opgenomen in uit te brengen muziek en films, waardoor gewone mensen die lastiger konden kopiëren en aan hun vrienden geven. Ze probeerden een kunstmatige schaarste te scheppen. Als rijke belanghebbenden niet blij zijn met wat een code doet, dan betalen ze voor het maken van software die dit tegengaat. De code geeft en de code neemt.

18

Er zijn in het verleden altijd momenten geweest waarop een bepaald beroep plotseling heel belangrijk werd en de beoefenaars ervan heel machtig. De wereld had opeens behoefte aan hun specifieke vaardigheden en beloonde ze.

Tijdens de Amerikaanse Onafhankelijkheidsoorlog eind achttiende eeuw vormden juristen de belangrijkste beroepsgroep. De Amerikaanse regeringsstijl is volledig opgebouwd uit wetten, dus advocaten en schrijvers van juridische teksten – iedereen die in zijn hoofd een rechtsstelsel kon samenstellen, die overtuigend en hartstochtelijk kon pleiten voor een bepaalde structuur – waren machtig. Zij hadden invloed. Als je naar de carrière van de grondleggers van de Verenigde Staten kijkt, zie je dat de meesten een

juridische opleiding hadden (John Adams, Alexander Hamilton, John Jay, Thomas Jefferson), en dat degenen voor wie dit niet opging (James Madison) niettemin meesterlijke legalisten waren. Deze mannen schreven de regels die van Amerika Amerika maakten. Zij schreven het besturingssysteem van zijn democratie.¹⁵ Hun kleinste beslissingen over het ontwerp hebben enorme, langdurige invloed gehad op de ontwikkeling van de republiek. Door het creëren van het kiescollege hebben de grondleggers bijvoorbeeld onbedoeld een systeem geschapen waarin, tweehonderd jaar later, presidentskandidaten zouden ontdekken dat ze eigenlijk alleen maar aandacht hoefden te besteden aan een handjevol *swing states*. Zodra onomstotelijk vaststaat dat een staat Democratisch of Republikeins zal stemmen, verdwijnt hij van de politieke kaart. Geen enkele presidentskandidaat brengt daar een bezoek of probeert de kiezers daar over te halen. Een andere keus in het ontwerp – laten we zeggen: de president wordt gekozen omdat hij de meeste stemmen heeft gekregen in het hele land – zou hebben geleid tot een wezenlijk andere stijl bij de hedendaagse verkiezingen. De grondleggers hebben echter een systeem ontworpen waaraan de Amerikanen vastzitten, tenzij de grondwet zou worden veranderd.

19

Honderd jaar na de Amerikaanse Onafhankelijkheidsoorlog werd een nieuwe beroepsgroep belangrijk. Nu de industriële revolutie op volle toeren draaide, begon het land te verstedelijken en verrezen er wolkenkrabbers in New York, Boston en Chicago. Ineens was het van cruciaal belang om uit te zoeken hoe je miljoenen mensen dicht op elkaar kon laten wonen zonder dat ze omkwamen in hun eigen uitwerpselen en hoe je ervoor kon zorgen dat ze redelijk schoon water en schone lucht hadden en een manier om zich te verplaatsen. Hiervoor waren hopen ingenieuze mechanismen nodig, dus plotseling hadden civiel ingenieurs, architecten en stadsplanologen het voor het zeggen. Iedereen die op een van deze terreinen kon werken – metrobouwers, bruggenbouwers, landschapsarchitecten – had een buitengewoon grote rol bij het bepalen van de leefstijl van de stadbewoners in de VS. En ook nu had één enkele keus in het ontwerp vele jaren later nog een enorme invloed op het leven van de bevolking. Robert Moses was een

beroemde stadsplanoloog in het New York van rond 1950, die in een geweldige vloed van activiteit tientallen snelwegen en parken liet aanleggen die bepalen hoe het hedendaagse New York City werkt. Door enkele van zijn besluiten werden levens verwoest. In 1948 begon hij met de aanleg van de Cross Bronx Expressway, die volgens hem nodig was om de verkeersdruk van Long Island naar New Jersey te verlichten. Dat deed hij ook, maar tegen een heel hoge prijs voor de overwegend zwarte buurten waar de snelweg doorheen sneed. Al snel trok de snelweg zoveel lawaaiig, smerige lucht uitstotend vrachtverkeer aan dat de waarde van de nabijgelegen huizen kelderde, waardoor honderden meestal Afro-Amerikaanse gezinnen plotseling een stuk armer werden en in een veel minder aangename buurt woonden. (Moses hield zo te zien ook van ‘Move fast and break things’.)¹⁶

20 Om de huidige wereld te begrijpen, moeten we iets van coders snappen. Wat zijn dit voor mensen die de wereld van nu bouwen? Wat drijft hen? Wat voor persoonlijkheidstype heeft iemand die zich aangetrokken voelt tot het schrijven van software? Welke invloed heeft hun werk op ons? En de misschien wel interessantste vraag: wat doet het met hen?

Bijna iedere programmeur heeft een vergelijkbaar verhaal over het moment waarop hij in de ban raakte van programmeren.

Vaak gebeurt dit wanneer ze hun eerste, piepkleine stukje code schrijven dat de computer ‘Hallo, wereld!’ laat zeggen.

```
print ("Hallo, wereld!")
```

Druk op enter om die code uit te voeren en de computer print wat je verwacht:

Hallo, wereld!

Niet zo heel erg ingewikkeld, hè? Maar het heeft een opwindend en subliem effect op de beginnende programmeur. ‘Het is dat gevoel dat jij het voor het zeggen hebt,’ vertelde een coder in Noisebridge, de beroemde hackerspace van San Francisco, me. ‘Ik was

dertien en die machine kwam tot leven en deed wat ik wilde. Voor een kind is dat een te gek gevoel. Alsof je de baas bent over een klein universum, dat je zelf hebt gecreëerd.’

Het eerste gedocumenteerde gebruik van ‘Hallo, wereld!’ was in 1972, toen de jonge computerwetenschapper Brian Kernighan de handleiding schreef voor het programmeren in de programmeertaal B. Hij wilde het eenvoudigste tonen wat je B kon laten uitvoeren, en dat was een bericht printen. Ooit had hij een tekenfilm gezien waarin een kuiken uit een ei kroop en ‘Hallo, wereld’ zei, en hij had die grappige, eigenwijze ondertoon leuk gevonden. Daarom schreef hij een eenvoudig fragmentje in B-code dat die kleine boodschap liet zien. Al snel werd Kernighans geestige idee door programmeurs gejat, en sindsdien begint praktisch elke handleiding voor een programmeertaal – en dat zijn er meer dan 250 – met die ene formule.¹⁷ ‘Hallo, wereld!’ haalde er keurig de existentiële schok van programmeren uit: het scheppen van een levensvorm die opeens tot leven komt.

Rond programmeren heeft altijd een vleugje mysterieuze wonderdoenerij gehangen. Het is een soort constructie, absoluut, maar anders dan bij elk ander type constructie – mechanisch, industrieel, civiel – worden de machines die we met software maken, geweven van woorden. Code is spraak, spraak die een mens uit in silicium, die de machine tot leven wekt en laat doen wat wij willen. Hierdoor is code eigenaardig schrijftalig. Sterker nog, deze aard van code zien we zelfs terug in de wet. Tastbare machines of hulpmiddelen, zoals automotoren of blikopeners, vallen onder patentwetten, op software rust een copyright, waardoor het een eigenaardig zusje van het gedicht of de roman wordt. Toch verschilt software ook duidelijk van een gedicht of roman, omdat hij zulke directe fysieke effecten heeft op hoe we leven. (Deels hierom vinden sommige programmeurs het rampzalig dat code wordt gereguleerd door het copyright.) Code bevindt zich in twee werelden: half metaal en half idee.

‘Een programmeur werkt net als een dichter op maar heel korte afstand van het pure gedachtespul,’ schreef de software-engineer Fred Brooks in 1975. ‘Hij bouwt zijn kastelen in de lucht, van lucht, hij creëert ze door zijn verbeeldingskracht aan het werk te zetten

[...] Maar de programmaconstructie is, anders dan de woorden van de dichter, echt in de zin dat ze beweegt en werkt, zichtbare output produceert los van de constructie zelf. Ze print resultaten, tekent afbeeldingen, produceert geluiden, beweegt armen. De magie van mythen en legenden is werkelijkheid geworden in onze tijd. Je tikt de juiste toverspreuk in op je toetsenbord en er komt een scherm tot leven dat je dingen laat zien die er nooit waren, noch konden zijn.’¹⁸

Daarom heeft dat zinnetje ‘Hallo, wereld!’ zo’n enorme metaforische lading. Het roept alle religieuze tradities op waarin een god de schepping tot leven laat komen: ‘In den beginne was het woord.’ (Vooraf christelijke programmeurs zijn fan van dit verband. Robyn Miller, een aanhanger van de evangelische leer die meewerkte aan het programmeren van de gamewereld *Myst*, pauzeerde af en toe tijdens het programmeren nadat hij iets had gemaakt dat heel cool was, en dacht dan: het is goed.)¹⁹ Maar aan ‘Hallo, wereld!’ zit ook een sinistere kant. Het doet je denken aan de onverwachte neveneffecten die kunnen optreden als je iets tot leven brengt wat je niet meer in de hand hebt, zoals de verwaarloosde en wraakzuchtige stakker van dr. Frankenstein die uiteindelijk zijn geliefde naasten zou doden, of de onbedwingbare, zich steeds vermeerderende bezems uit *The Sorcerer’s Apprentice*. Dat geldt ook voor code. Dankzij jouw nieuwsoverzicht kunnen vrienden zorg organiseren voor een vriend die aan kanker lijdt, maar het geeft mensen die misselijke memes plaatsen ook de kans om volslagen geschifte samenzweringstheorieën te verspreiden. Deze harmonieuze weerklank tussen magie en code is de reden dat begin jaren tachtig programmerende nerdy tieners zich zo makkelijk konden inleven in *Dungeons & Dragons* (een spel dat fantasie vermengde met kansberekening op basis van dobbelstenen) of de bezweringen roepende tovenaars uit Tolkiens boeken. In de jaren zestig, toen programmeurs een soort code bedachten die voortdurend op de achtergrond draaide, noemden ze die een ‘daemon’. Larry Wall, de computerwetenschapper die de computertaal Perl schiep, nam er een functie in op die hij ‘bless’ noemde. De coder Danny Hillis merkte ooit op: ‘Een paar honderd jaar geleden zou een correcte omschrijving van mijn vak me in mijn geboortestreek

New England op de brandstapel hebben gebracht.’²⁰

Dit magische gevoel van de macht in handen hebben kan bedwelmend en leuk zijn. Het leent zich vaak voor een soort wereldvreemd idealisme, maar het kan ook leiden tot grote overmoed, vooral bij jonge coders die nog niet tot nederigheid neigen door levenservaring en eigen blunders. Heel goede programmeurs, zo merkte de coder en techcultuurrecensent Maciej Cegłowski op, ‘raken er vaak van overtuigd dat ze dankzij hun superieure analytische capaciteiten het unieke vermogen hebben om elk willekeurig systeem vanaf de eerste beginselen te begrijpen zonder enige voorafgaande opleiding. Succes in de kunstmatig gebouwde wereld van het softwareontwerp bevordert een gevaarlijk zelfvertrouwen.’²¹ Of zoals de computerwetenschapper Joseph Weizenbaum in 1976 zei: ‘Het zou heel vreemd zijn als Lord Actons opmerking dat macht corrumpeert niet van toepassing zou zijn in een omgeving waarin je zo makkelijk almachtig kunt worden.’²²

Programmeren is niet makkelijk. Je zit urenlang in je eentje te proberen je de kronkelige nuances van een stukje software eigen te maken: hoe wordt deze loop hier getriggerd door die input van de gebruiker, behalve als deze andere subroutine dan al draait, want in dat geval zou deze functie moeten opstarten. Het werd me op verschillende manieren beschreven als ‘proberen om de structuur van elke straat in Londen te bouwen en te onthouden’ en ‘in je geest een enorm kaartenhuis optillen en op zijn plaats zetten’. Daarom trekt het vaak introverte mensen en logische denkers aan die graag puzzels oplossen. Van die gasten die het heerlijk vinden om op zaterdagavond om elf uur stuurprogramma’s te maken voor niet meer verkrijgbare webcams uit 1997 alleen omdat ze in een la een oude webcam hebben gevonden en die aan de praat willen krijgen, en het een verdomd interessant probleem is en beslist voorstelbaarder dan omgaan met mensen. (Maar het is geen volledig introvert zijn. In feite is het bouwen van software tegenwoordig een steeds socialer gebeuren; het vereist teamwork en, eerlijk gezegd, gaat er vaak meer tijd zitten in bijeenkomsten met collega’s om te praten over wat er gedaan moet worden dan in het eigenlijke werk.)

Maar met introvert zijn of logisch denken ben je er nog niet.