

DE 'E' VAN SOFTWARE ENGINEER

Koop een willekeurig computertijdschrift en je leest de prachtigste verhalen over computers. Maar ga je naar een verjaardag, dan hoor je hele andere verhalen anders. Als het problemen zijn, komt het door de computer. Als hij het al doet, dan is het weer te ingewikkeld. Bovendien: als je computers wel leuk vindt, dan ben je een 'nerd'. Het lijkt erop dat wij, software engineers, iets fout doen. Maar wat?

En hoe kunnen we dat verbeteren?

Verwarring

Er is nogal wat verwarring over computers en over alles en iedereen die werkt met computers. Vooral over de makers van de computer, of eigenlijk, van de software. Dat begint al met de naamgeving. Waren computers vroeger het domein van ingenieurs en programmeurs, nu zijn computers alom aanwezig en spreken we van 'ICT'. Maar om werk te zoeken kijken we onder het kopje 'automatiseerder'; niet bij 'techniek'. Toch is ons vak erg technisch.

In sommige kringen is 'automatisering' nog steeds een populair beroepsveld: de 'ICT' verdient immers goed (of was alleen dát vroeger beter?). Een strak pak en goede babbel lijken soms voldoende. Maar ben je ook automatiseerder als je vooral computers en software verkoopt? Of als je computers open en dicht schroeft? En heb je dan verstand van computers? Natuurlijk, deze mensen doen goed werk. Maar het zijn geen technici. Ze hebben geen opleiding of ervaring met de technische details van een computer. Van mij mag iedereen in de ICT werken, maar laten we niet doen alsof je dan automatisch verstand hebt van computers. Of van het programmeren daarvan.

Kennelijk wordt de term 'automatiseerder' (of ICT'er) gebruikt voor iedereen die iets met of in de ICT doet. Daarmee zijn ook wij automatiseerders. Maar wat maken we dan automatisch? We maken programma's, daarmee zijn we programmeurs! Maar iedereen programmeert tegenwoordig: minimaal de video- of dvd-recorder. Ik vind daarom ook de term programmeur niet correct voor het benoemen van ons beroep. Het schrijven van embedded software, drivers en zelfs een 'gewoon' C programma is heel wat complexer dan de toch al zo lastige video, toch?

Vaak wordt de term 'Software Engineer' gebruikt. Soms terecht en vaak ook niet. Maar als we letterlijk kijken naar die term, dan is het helemaal niet zo'n slechte. De naam bevat zowel software, wat handig is. Maar ook 'engineer'. Het is alleen niet altijd duidelijk wat dat betekent.

Software ingenieur!

Het Engelse woord 'engineer' werd als eerste gebruikt in het leger; het betrof een elite-onderdeel om bruggen en andere complexe

Ik leg de lat graag hoog. Het moet altijd beter; dat moeten we althans proberen. Waarom? Omdat ons mooie vak van 'Software Engineering' anders ten onder zal gaan! Het is een jong vak maar ook een erg onvolwassen vak! Natuurlijk zijn er uitzonderingen, maar het imago van de meeste software is gewoon slecht. Er zijn weinig gebruikers van software (buiten ons eigen wereldje) die enthousiast zijn over 'computers' of over 'software'. Dat kan zo niet doorgaan. Dat de technische opleidingen leegstromen is geen goed voorteken. En als we software niet beter maken, zullen er ooit gewoon geen kopers meer voor zijn.

Vergelijk het met andere producten zoals de auto. De verplichting, begin vorige eeuw, om iemand met een vlag (en 's avonds met een rode lamp) vooruit te laten lopen beperkte het praktische nut van de eerste generaties; toch waren de bezitters trots en waren auto's gewild. Al hadden ze zeker hun 'bugs'. Tegenwoordig is dat heel anders; elke auto heeft dezelfde interface: 3 pedalen, een stuur, etc. Het is allemaal hetzelfde. Er is veel concurrentie en auto's worden vooral verkocht op - technisch onbelangrijke - details als vorm, kleur of 'stoelzachtheid'. Zelfs een klein, technisch onvolkomenheidje is al 'killing'; denk maar aan de kantel-benz.

Als dat ook zou gelden voor software, dan blijft er - vrees ik - weinig over. Daarom moet het beter! Het kan beter. Niet dat alle software evengoed moet zijn als een Rolls-Rolls. Er is ook ruimte voor een Golfje! Desnoods voor een Lelijk eendje of een Mini. Maar de tijden van de Trabant liggen achter ons!

Wat is mijn missie: software beter maken. Software, waar mensen trots op zijn. Software die werkt, zoals je mag verwachten. Wellicht ben ik een moderne 'Don Quichote', misschien ook niet. Maar proberen zal ik het! En ach, als het niet lukt, word ik wellicht net zo bekend als 'Don'.

Waarom wil Albert het beter?

dingen te maken. Dat geldt ook voor het Nederlandse woord 'ingenieur', dat via 'civiel ingenieur' ook algemeen gebruikelijk werd. We zouden dus best kunnen stellen dat wij 'software ingenieurs' zijn: we behoren tot de elite die echt verstand heeft van computers. We zorgen ervoor dat het leger van ICT'ers zijn werk kan doen.

Maar zijn we dat ook? Die elite? Er valt van alles vóór te zeggen maar we kunnen onszelf niet benoemen tot elite. Dat zijn we pas als anderen ons zo zien. En helaas hebben we een ander imago.

Toch hebben we heel wat gedaan; zonder ons - zonder software ingenieurs - zouden TV's veel eenvoudiger zijn, was de telefoon nog analoog en immobiel en was er zeker geen internet. Maar dat zijn de positieve uitzonderingen. Er gaan ook een hoop zaken fout.

Zo vinden alleen wij het normaal om al trots te zijn als een product (slechts) voor 99% werkt en zien alleen wij elke verandering als een verbetering. We zijn erg onkritisch op onszelf of op ons vak. Natuurlijk, de hele computerindustrie is jong. Programmeren is zelfs nog jonger; zo bestaat de 'oude' C-taal nog geen 40 jaar! Realiseer je je wat dat betekent? O.a. dat er nog nooit een in 'C' geschoolde Software Ingenieur met pensioen is gegaan!

Bij een jong vak horen snelle ontwikkelingen, evenals 'onvolwassen' gedrag. Dat zie je elders ook. De automobiel bestaat iets meer 100 jaar en ook daar kunnen we slechts glimlachen als we naar de auto's van een halve eeuw oud terugkijken. Auto's van nu zijn veel, veel beter.

Ook software wordt beter. Toch zijn er verschillen. Zo lijkt de wereld tegenwoordig sneller te draaien: we zullen dus sneller volwassen moeten worden. En ik vraag me af of de autobouwers van zo'n 60 jaar terug 'nerds' waren. Ik zie ze meer als voorbeelden, als helden. Iedereen wilde wel een auto ontwerpen.

Auto's versus software

Er zijn vele varianten van hetzelfde grapje. Maar alle bevatten ze dezelfde kern van waarheid: auto's en software lijken deels op elkaar en deels moeten wij software echt beter maken.

Bill Gates vergeleek de computer-industrie met de auto-industrie en hij beweerde: "Als zij, net als wij, waren bijgebleven met de ontwikkelingen in de technologie, dan zouden we auto's rijden van minder dan 300 euro met een snelheid van 1000 kilometer per uur".

De reactie was een persbericht dat dat bevestigde en aanvulde met o.a.:

- Auto's zouden, zonder enige reden, twee keer per dag crashen.
- Elke keer als de witte strepen op het wegdek werden overgeschilderd, zou iedereen een nieuwe auto aan moeten schaffen.
- Iedereen zou accepteren dat een auto regelmatig midden op de snelweg stopt, de ramen open en dicht doet, opnieuw start en dan gewoon verder rijdt.
- De meeste auto's kunnen maar 1 persoon vervoeren; er bestaan wel versies met meerdere stoelen, maar per keer dat die gebruikt worden, moet je bijbetalen.
- De olie, watertemperatuur en de waarschuwinglichten zouden allemaal door één en hetzelfde lampje worden aangegeven.
- Voordat de airbag uit zou klappen, zou er eerst een venstertje verschijnen met: Weet u het zeker? Bij de rem is dat een optie.
- Er zouden gratis Linux-auto's zijn, gemaakt door vakmensen, maar daarvan zijn zoveel varianten dat niemand weet wat de beste is. Bovendien moet je het onderhoud zelf doen.
- Elke keer dat een nieuw model uitkomt, moet je opnieuw rijles nemen, want de bediening is compleet 'verbeterd'.
- Je moet minimaal twee auto's kopen, omdat de essentiële gebruiksaanwijzing niet in één auto past.

Software beter maken

Er zijn meerdere manieren om software beter te maken. We zouden kunnen afspreken dat we geen 'bugs' meer maken, maar dat is niet realistisch en waarschijnlijk ook onvoldoende.

Ik kies daarom liever voor een meervoudige aanpak. Zoals in de ToolView serie uit de afgelopen PTSers: betere tools en ze beter gebruiken. Maar ook betere softwaretechnieken en -methodieken. Het softwareproces zal aan bod komen; dat kan beter en kan helpen om de software zelf te verbeteren. Natuurlijk is testen belangrijk, evenals diverse vormen van documenteren. En beter communiceren! Daarom een oproep voor feedback: heb je een idee laat het me weten!

Beter imago

Als we willen dat ons vak een beter imago krijgt, zullen we daar anders mee om moeten gaan. Want bij 'SE' lijkt het vooral om de software te gaan. Het stukje 'engineering' krijgt wat weinig aandacht. Kijken we bijvoorbeeld naar PTS, dan zien we dat (bijna) iedereen de titel ingenieur mag dragen. Maar dragen we die ook uit?

Het woord ingenieur is afgeleid van het Latijnse 'ingenium' - letterlijk: 'wat is ingeboren' - en betekent vindingrijkheid en verstand(ig). Een ingenieur is iemand die zijn wetenschappelijke kennis gebruikt om praktische problemen op te lossen. Een software ingenieur zou dus een verstandig persoon moeten zijn die vindingrijk is en op systematische, kundige en het liefst op wetenschappelijke manier problemen met behulp van software oplost. Het gaat dus niet om 'code kloppen'. Ook testen, ontwerpen en vele andere aspecten vallen daaronder.

Software-ingenieurs zijn dol op recursie, daarom: om het probleem van het imago van de software ingenieur op te lossen, moet het imago verbeteren naar probleemoplossend. Want een belangrijk aspect van 'ingenieur zijn' is het oplossen van problemen. Bij elke stukje software dat we maken moeten we de vraag 'Welk probleem lossen we voor wie op?' in ons achterhoofd houden. Want als we vooral software schrijven om de software, dan zijn we aan het hobbyen. Dat is prima overigens: veel goede software is als hobby ontstaan. Maar als we gezien willen worden als professioneel software ingenieur moeten we dat niet doen. De meeste auto's worden toch ook niet gemaakt door een 'auto-nerd'? Het woord 'auto-nerd' bestaat overigens niet. Zo iemand heet 'auto-liefhebber'; dat is hetzelfde maar dan met een beter imago. Dat is wat wij fout doen; wij zijn nerds, zij liefhebbers!

