



Ron Tolido

Steve Phone Home

Software zonder fouten is niet van deze aarde. Tenminste, als ik een recent memo van Steve Ballmer aan zijn klanten goed heb begrepen. In een zeldzame bui van openhartigheid laat hij los dat zelfs de software van Microsoft zo nu en dan wel eens een bug bevat. Een verpletterende mededeling die mij persoonlijk minutenlang in diep, somber gepeins heeft doen verzakken. Diep in je hart wist je het allang! Maar als de waarheid zo expliciet wordt, komt hij toch als een mokerslag aan.

Volgens Ballmer bevat zelfs een klein softwareproduct al miljoenen regels code, 'en daarin verbergen zich vele plekken voor bugs om zich te verbergen'.

Hij zal het wel weten.

Microsoft heeft eigenhandig Visual Basic grootgemaakt; een programmeertaal die ordinair gillend uitnodigt tot een ondoorzichtige brij van code en variabelen. Daarnaast zijn veel programma's van Microsoft geschreven in C++, een van de meest misdadige uitvindingen van de IT-wereld in tientallen jaren. Met C++ kun je zelf geheugen alloceren en vrijgeven, een trefzekere manier om in de loop van een sessie steeds meer geheugenlekken op te lopen. Ook kun je met deze programmeertaal snel en efficiënt delen van de *stack* overschrijven of anderszins de grenzen van buffers verkennen (een wetenschap die overigens ook het hackersgilde elke ochtend overal ter wereld vrolijk fluitend doet opstaan).

Bugs zijn een voldongen feit. *Software Will Crash*. Zelfs met rigoureuze module-, acceptatie- en stresstests is een perfecte werking blijkbaar niet te garanderen. Al helemaal niet in een hijgerig tijdperk waarin snel scoren en bij elkaar geraapte oplossingen de agenda bepalen. In zo'n context hoorde ik een gerespecteerde vakgenoot onlangs uitleggen dat een nieuwe, iteratieve teststrategie warempel had geleid tot *minder foutvolle software*. Tel uit je winst.

Het siert Ballmer dan wel dat hij een in wezen genante situatie naar een positieve sfeer weet om te buigen. In zijn memo vertelt hoe de in zijn nieuwe producten ingebouwde 'Phone Home' software helpt om meer greep te krijgen op fouten. Als een programma crasht biedt het de gebruiker de optie om een compact statusrapport online te versturen naar

Microsoft. Deze *minidump* bevat gegevens over onder andere modulen, variabelen en de stack. Waardevolle informatie die werkelijk helpt om de oorzaak van fouten op te sporen en snel *patches* te maken. Het inzicht in de aard van de bugs neemt daardoor toe. Statische analyse door Microsoft wijst uit dat 20% van de programmeerfouten leidt tot 80% van de crashes. Nog opmerkelijker is dat slechts 1% van de fouten garant blijkt te staan voor de helft van de uitglijders. Als je maatregelen weet te nemen om uitgerekend die categorie van fouten te vermijden, zet je als softwareproducent echt stappen voorwaarts.

Als we eenmaal het rouwproces door zijn dat software doorgaans niet in één klap perfect werkt, biedt Phone Home software bruikbare perspectieven om de schade te beperken. Het online versturen van een discrete minidump werkt alvast een stuk beter dan het tonen van een *Blue Screen* met ridicule, hexadecimale data. Als de gedupeerde gebruiker ook nog eens via het Internet kan volgen hoe het zit met afhandeling van zijn fout, neemt de tevredenheid toe. Een *Bug* wordt daarmee een *Opportunity* en de falende programmeur kan trots vaststellen dat hij in ieder geval contact heeft gemaakt met de klant.

In de ultieme variant haalt Phone Home software ter plekke de patch op vanaf de thuisbasis en draait de applicatie onopgemerkt door. Een applicatie wordt daarmee zelfhelend. Dat kenmerkt succesvolle biologisch organismen en zou ook de norm moeten zijn voor complexe IT-systemen, waarin haperende onderdelen nu eenmaal onvermijdelijk zijn. Dat Microsoft in de uitwerking nog een steekje laten vallen, moet op de koop toe worden genomen: zelf heb ik crashes meegemaakt bij het versturen van de minidump, waarna er weer een nieuwe minidump werd verstuurd. Maar dat is overmacht.