

Een juiste diagnose

*diagnostiek als integraal onderdeel
van de architectuur*

Erik Philippus
Improvement
erik@agile-architecting.com

Onlangs heeft Microsoft erkend dat de User Account Control (UAC) een van de meest controversiële onderdelen is van Windows Vista. Deze waakhond vraagt om toestemming als er iets ingrijpends staat te gebeuren, bijvoorbeeld als een applicatie of devicedriver wordt geïnstalleerd of verwijderd of als een systeeminstelling wordt veranderd. Vanwege het grote aantal zinloze meldingen ervaren gebruikers de functie als opdringerig en irritant. Doordat ze blind klikken op de Allow-knop, kan de UAC zelfs leiden tot grotere onveiligheid. Veel Vista-gebruikers schakelen de functie uit bij het installeren van software, of stellen haar zelfs permanent buiten gebruik, om de waterval aan beveiligingsmeldingen te stoppen.

Deze perikelen geven mij een fors déjà vu. Zo'n vijftientig jaar geleden werkte ik als junior software-engineer bij de firma Foxboro aan de userinterface van grote industriële controlesystemen. Om gevoel te krijgen voor hoe operators met een dergelijk systeem werken, ging ik een dagje meekijken bij de inbedrijfstelling van een naftakraker ergens in de buurt van Pernis. Met grote regelmaat kwamen er foutmeldingen op het scherm. Klaarblijkelijk was dat normaal bij het opstarten van een installatie, aangezien de operators de meldingen routineus negeerden. Ook miste ik het gebruikelijke geluidssignaal, zelfs bij een red alert. Toen ik onder het dashboard keek, zag ik dat de twee draadjes naar de speaker waren doorgeknipt. De operator was blijkbaar zo horendol geworden van de stroom aan schelle alarmsignalen dat hij de boel onklaar had gemaakt.

Anno 2009 is het nog steeds een forse uitdaging om foutboodschappen op een effectieve manier onder de aandacht te brengen van gebruikers. Laatst voegde een vastgelopen applicatie mij op strenge toon toe: 'Pure virtual function call'. Zonder C++-kennis zou je zomaar kunnen denken dat je zelf iets fout hebt gedaan. Dergelijke meldingen reizen via diverse softwarelagen naar boven om uiteindelijk bij de eindgebruiker op het bordje te komen, zonder dat de omschrijving van de fout onderweg is aangepast aan de context van de lezer. Aandacht voor de 'taalkundige ergonomie' van software-intensieve systemen is geen overbodige luxe.

Belangrijk is ook om te waken voor excessieve foutafhandeling. Die kan de omvang en complexiteit van systemen flink opdrijven, en vaak ook de performance negatief beïnvloeden. Maar wat nog erger is: de eindgebruiker zal 'doof' worden voor elke foutboodschap, waardoor werkelijk relevante informatie verloren gaat. De architecten bij Microsoft lijken lering te hebben getrokken uit de ervaringen met Vista: in Windows 7 beperken ze onnodige of dubbele beveiligingsmeldingen tot een minimum, zodat echt belangrijke boodschappen makkelijker zijn te herkennen.

Deze aspecten van foutafhandeling hebben een directe relatie met het vermogen van systemen om bruikbare diagnostische informatie te verschaffen. Afwijkend gedrag van een geïnstalleerd systeem kan ingrijpende gevolgen hebben, en het kunnen stellen van een snelle en juiste diagnose is vaak van groot economisch belang. Ook diagnosticeren (en upgraden) op afstand wint terrein. Zeker voor multidisciplinaire systemen is diagnosticeerbaarheid een steeds belangrijkere eis. Daarbij is vaak een sleutelrol weggelegd voor het ontwerp van de embedded software.

Probleem is dat diagnostiek, net als foutafhandeling, nog vaak wordt gezien als *add-on* in plaats van als integraal onderdeel van de systeemarchitectuur. Daardoor kent de diagnosefunctionaliteit die we bijvoorbeeld in een moderne auto vinden nog diverse tekortkomingen. De boodschap moge duidelijk zijn: doelgerichte foutafhandeling en diagnostiek worden pas echt succesvol als de (systeem)architectuur ze expliciet ondersteunt. Hier ligt een mooie en belangrijke taak voor de architect.

*Erik Philippus
maart 2009*

Verschenen in de architectencolumn in Bits & Chips, 11^e jaargang nr 5/6