Ron Tolido

Een veilig gevoel, dat we vanaf nu in ons vakgebied een Top 25 van Meest Gevaarlijke Programmeerfouten hebben (even "Top 25 errors" *googelen* voor wie dat nog niet wist). Als we de lijst van fouten doornemen komen we geheide klassiekers tegen als 'de input onjuist valideren' en 'niet binnen de grenzen van een geheugenbuffer blijven' (met speciale dank aan de op dit gebied buitengewoon getalenteerde talen C en C++). Maar ook moderner spul als 'onveilige scripts van andere websites gebruiken' en 'cruciale informatie in foutmeldingen opnemen'.

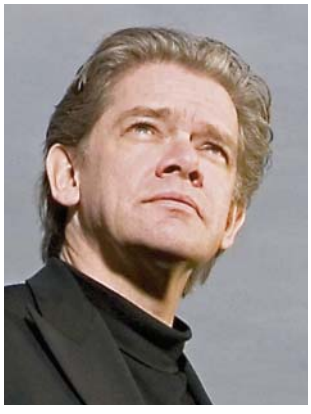
Dweilen

Heel leerzaam materiaal – ook voor hackers trouwens – en als ontwikkelaar zou je deze Top 25 eigenlijk uit het hoofd moeten leren (of gewoon *taggen*, onthouden is misschien teveel gevraagd).

Toch ontbreekt de allergrootste oorzaak van kritieke fouten in software: er wordt *teveel software* geschreven. Dit is een vakgebied waarin we hardnekkig zelf problemen blijven veroorzaken die we daarna met veel aplomb – desnoods begeleid door lijstjes – oplossen. We zetten eerst alle kranen in huis wagenwijd open en komen dan trots aanhuppelen met de allernieuwste vochtabsorberende dweil. Het lijkt me daarom een goed idee om de categorie 'Onnodig Coderen' aan de lijst toe te voegen. Ik heb alvast zeven suggesties voor de typerende misdragingen die je in deze categorie tegenkomt:

1. **Hergebruik van code vermijden**. Veel basisfunctionaliteit wordt gedekt in *frameworks* of andere – doorgaans grondig geteste – herbruikbare componenten. Toch zijn er veel ontwikkelaars die denken dat zelfs de meest elementaire functies toch maar beter door henzelf kunnen worden ontwikkeld. Schadelijke zelfoverschatting.
2. **Generen van code vermijden**. Laten we er eens vanuit gaan dat er ontwerpmodellen zijn of specificaties in een al dan niet natuurlijke taal. Daaruit kan veel foutloze code worden gegenereerd; een te mooie kans om te missen. Grootste valkuil: zelf de generator willen ontwikkelen. Zie ook 4.
3. **Standaardpakketten niet overwegen**. Steeds meer software die we vroeger zelf moesten bouwen, komt nu uit de laboratoria van de ERP-leveranciers. Laat ze zich in Waldorf maar druk maken over het vermijden van bugs. Standaard bedrijfsfuncties moeten door standaard software worden ondersteund.
4. **Zelf middleware ontwikkelen**. In een schimmig verleden heb ik ooit ook een windowing systeem en een transactiemonitor geschreven. Vroeg of laat word je altijd achterhaald door veel betere producten van gespecialiseerde leveranciers. Tijd voor alarmbellen bij de neiging om eigen *ESB's*, codegeneratoren, portals en *object management systemen* te bouwen.
5. **Proceslogica en bedrijfsregels hard coderen**. Met de komst van een nieuwe generatie *Business Process Management* (BPM) en *Business Rule Engine* (BRE) systemen moet elke ontwikkelaar zich serieus afvragen waarom er nog allerlei proceslogica en bedrijfsregels *hardcoded* worden ingebakken in de software. Ze samen met de gebruikers in een apart systeem vastleggen - in een taal die door de bedrijfsvoering wordt begrepen – scheelt niet alleen een hoop coderen maar slaat ook nog eens een brug tussen technologie en bedrijfsvoering: het gebied waar van oudsher de meeste fouten ontstaan.
6. **Complexe constructies opzoeken**. Zinloze complexiteit in de IT: vroeg of laat gaan we er de eerste Stille Tocht omheen meemaken. Dynamisch polymorfisme, multiple inheritance, xml-to -object-to-relational mapping (en terug): we zwermen als vliegen rond complexe constructies. Niet omdat we ze nodig hebben, maar omdat we worden aangetrokken door lekker foute, analytische uitdagingen.
7. **Verkeerde programmeertalen toepassen**. Als we de eerstgenoemde zes verschijnselen serieus hebben genomen, zouden we wel eens kunnen merken dat er helemaal niet meer zoveel valt te programmeren. Misschien willen we vooral kunnen orchestreren en lijmen. Een mooi moment om eens na te gaan of we werkelijk moeilijk, foutgevoelige programmeertalen als Java of C# nodig hebben.

Als je geen code schrijft, kun je kortom ook geen fouten produceren. Draai de kraan maar dicht. En vergeet die dweil. «



Ron Tolido
tolido.blogspot.com