

## **GEEN FUNCTIONEEL ONTWERP AUB!**

**Door Gerwin Berentschot**

**Voordat je iets nieuws koopt wil je het eerst kunnen zien, voelen, ruiken, proeven, etc. Van een TV wil je eerst de beeldkwaliteit zien en die nieuwe schoenen wil je toch eerst wel even passen. Met software is het niet anders. Klanten willen graag vooraf weten welke software ze aanschaffen en wat dat gaat kosten. Dat levert problemen op bij maatwerksoftware, simpelweg omdat de software er nog niet is op het moment van aanschaf. Wat nu?**

### **Een functioneel ontwerp is niet functioneel**

In een poging de klant vooraf toch een beeld te geven van de software die hij kan verwachten, stellen softwarebouwers vaak een document op waarin de software wordt beschreven. Dit noemen we dan een functioneel ontwerp. Het probleem is alleen dat er niets functioneels aan is. Wie kan zich nou op basis van een tekstdocument een goede voorstelling maken van de software en hoe deze zal gaan werken? Twee personen die los van elkaar een roman lezen vormen zich waarschijnlijk allebei een ander beeld van de hoofdpersonen en locaties zoals die in het boek worden beschreven. Zo werkt het ook met een functioneel ontwerp.

Als softwarebouwer moet je de werkelijkheid met je klant afstemmen en niet één of andere abstractie daarvan. Je vertelt iemand toch ook niet hoe een lied klinkt? Het lied neuriën is zinniger. Hoe verder je van het echte product afstaat, hoe groter het gevaar dat je denkt dat je elkaar begrijpt, terwijl dat in werkelijkheid niet zo blijkt te zijn. En dat leidt vrijwel altijd tot teleurstellingen.

## **Laat gewoon schermen zien!**

Klanten zijn praktisch ingesteld. Dat geldt helemaal voor eindgebruikers. Zij moeten immers met de software werken en er resultaten mee boeken. Als je eindgebruikers vraagt welke software zij voor ogen hebben, dan kom je er al snel achter dat ze denken in schermen en niet in tabelstructuren, coderegels, queries en andere zaken onder de motorkap. Daarin zijn ze vaak totaal niet geïnteresseerd. Klanten hebben verwachtingen aan het voor hen zichtbare deel van de software. Daarop baseren zij of een product voor hen de moeite waard is of niet.

Als klanten software voor zich zien in schermen, laat ze die dan in hemelsnaam gewoon zien! In eerste instantie door een uurtje met de klant om tafel te gaan zitten en schetsen te maken van enkele belangrijke schermen. Gewoon *old school*, dus met een stift op papier of een whiteboard. Je stemt in dat ene uurtje al meer verwachtingen af dan je met een duimendik en tijdrovend functioneel ontwerp ooit zult doen. De schermschetsen geven een goed beeld van wat er moet worden gebouwd. Dit voorkomt misverstanden. Ook kun je als softwarebouwer al gelijk een concrete prijs en doorlooptijd noemen.

## **Te paard?**

Oké, we kunnen aan de slag. Toch? We hebben overeenstemming over wat er moet worden gebouwd en over doorlooptijd en prijs. Te paard! Aanzetten die computers en bouwen! Maar wacht even. Je hebt samen met de klant schermschetsen gemaakt, maar het kan best zo zijn dat bepaalde dingen in het echt minder goed uitpakken. Dat moet dan worden aangepast. Als je dat pas in het kant-en-klare product constateert, dus in een compleet afgeronde versie van de software inclusief tabellen en code, dan kan het zijn dat hele dagen werk in de prullenbak belanden. En dat is erg kostbaar. We moeten er dus voor zorgen dat we het ontwerp nog

met een gerust hart kunnen aanpassen zonder al teveel werk te vernietigen.

### **Bouw eerst alleen de interface**

De oplossing is simpel: bouw eerst alleen de interface (de schermen) van de software en laat de klant daarmee spelen. Tijdrovende dingen zoals het aanleggen van tabellen of het schrijven van code zijn uit den boze! De klant kan naar harte lust typen en klikken in de interface en bepalen of alles werkt zoals het moet. Constateert de klant een tekortkoming, dan is het een kwestie van wat schermen aanpassen. Er hoeft geen onderliggende logica te worden vernietigd. Het finetunen van de software blijft hierdoor 'licht' en goedkoop.

Pas als het prototype van de interface helemaal naar tevredenheid is ga je de voor de klant onzichtbare structuren zoals tabellen en code in het prototype bouwen. Je weet nu zeker dat de arbeidsintensieve onderdelen van de software voldoen aan wat de klant verwacht. En daar is geen functioneel ontwerp aan te pas gekomen. Lekker praktisch en ook nog eens sneller, beter en goedkoper!

Gerwin Berentschot

Gerwin Berentschot is oprichter/eigenaar van Mr. Access. Het team van Mr. Access richt zich op het ontwikkelen van maatwerkdatabases in Microsoft Access.

Meer informatie is te vinden op [www.mraccess.nl](http://www.mraccess.nl).