

User Experience

Vaak sta je er niet bij stil waarom een product op een bepaalde manier ontworpen is. Neem bijvoorbeeld het toetsenbord. Fabrikanten komen met de geweldigste designs voor toetsenborden. Ergonomische vormgeving, aanpasbare sneltoetsen, backlights, je kunt het zo gek niet bedenken of het wordt op een toetsenbord geplaatst. Gaaf natuurlijk. Maar wat is de belangrijkste bestaansreden voor een toetsenbord? Het invoeren van tekst. Waarom staan dan de letters zo inefficiënt op het toetsenbord. De reden hiervoor komt uit 1874 toen de QWERTY-indeling bedacht werd door Sholes & Glidden typewriters om sneller te kunnen tikken met een typemachine. Dit had een puur technische reden, omdat de typemachine dan minder snel vastliep. Stel dat we de indeling van een toetsenbord opnieuw zouden kunnen bedenken zonder rekening te houden met het verleden, maar wel duidelijk te kijken naar hoe de gebruiker het toetsenbord het beste en snelste kan gebruiken. Waarschijnlijk komt er dan iets uit dat helemaal niet meer lijkt op een toetsenbord zoals we dat vandaag de dag kennen. Veel efficiënter en geen last van RSI? Net als bij het voorbeeld van het toetsenbord, wordt software vaak geoptimaliseerd voor de verwerking op een computer. Vroeger was dit een prima aanpak, maar tegenwoordig wordt computerhardware steeds goedkoper, terwijl menselijke arbeid steeds duurder wordt. Wordt het niet eens tijd om software te optimaliseren voor de eindgebruiker?

Als je de gebruiker centraal wilt stellen in een applicatie moet je kijken naar de complete gebruikerservaring, ook wel user experience of UX genaamd. Dit gaat veel verder dan alleen een mooie user-interface. Aspecten als hoe de applicatie wordt gebruikt, waar deze applicatie wordt gebruikt en door wie, hebben grote invloed op de optimale user experience. Waarom is user experience dan zo belangrijk? Simpel, het levert geld en tijd op. De gebruiker kan efficiënter zijn werk doen en heeft minder training nodig. Zelfs het terugbrengen van een paar clicks voor een veelvoorkomende handeling kan bij een flink aantal gebruikers uiteindelijk enorme besparingen opleveren. Denk ook eens aan het terugbrengen van het aantal dure helpdesk-calls. Door een applicatie begrijpbaarder te maken kan hier ook flink op bespaard worden.

Mensen hebben vaak van nature een weerzin tegen veranderingen in hun werksituatie. Dit treedt ook vaak op als een nieuwe applicatie wordt geïntroduceerd. Soms kan dit zo heftig zijn dat vele softwareontwikkelprojecten falen doordat ze niet door de eindgebruikers worden geaccepteerd. Door tijdens het softwareontwikkelp proces de gebruiker centraal te stellen en te focussen op een betere user experience kan deze weerstand uiteindelijk aanzienlijk worden verkleind. Uiteraard wordt een mooie user-interface ook steeds belangrijker. Als je kijkt naar commerciële software voor bijvoorbeeld de consumptie, dan zal deze sneller voor iets kiezen dat er mooier en beter uitziet dan voor een product dat wellicht veel beter is, maar er totaal niet uitziet.

Bij softwareontwikkeling speelt user experience dus een erg belangrijke rol. Helaas zie je in de praktijk dat hieraan weinig tot geen aandacht wordt besteed. Veelal zie je de programmeur de schermen maken en bedenken of wordt een 'gebruiker' bijbetrokken die voor iedereen bepaalt waar welk knopje komt. Heeft de programmeur verstand van hoe de eindgebruiker het beste met de software kan werken? Grote kans van niet. Hij denkt voornamelijk vanuit de techniek. Het is dus belangrijk om gebruikers bij het softwareontwikkelp proces te betrekken, maar het blijft lastig om de juiste gebruikers te selecteren. Ze moeten natuurlijk wel representatief zijn. Bij Microsoft maken we hiervoor veel gebruikt van personas. Dit zijn fictieve personen die gebruikmaken van onze software. Door het opstellen van personas verlies je niet het contact met je eindgebruikers. Hoe vaak praat een programmeur daadwerkelijk met de eindgebruiker? Met personas krijgt hij hierbij ook een duidelijk beeld. Wireframes en storyboards zijn ook middelen om de user experience op voorhand te bepalen. Met behulp van prototypen kun je dit zelf met de verschillende gebruikersgroepen testen.

Als je een huis laat bouwen, wil je ook vooraf zien hoe het er uit gaat zien en of het huis wel is wat je er van verwacht. Met software wil je dit ook, dus maak je een prototype. Je kunt dit op twee manieren doen, namelijk op papier met schetsen (low fidelity prototyping). Dit heeft als voordeel dat het laagdrempelig is en je kunt de gebruikers erbij betrekken. Of je kunt een prototype maken dat net echt lijkt, wat als voordeel heeft dat iedereen ziet wat het echt gaat worden. Alleen bestaat het gevaar dat als je het bijna echt lijkende prototype aan de directieleden laat zien, ze zullen zeggen dat het al af is? Waar teken ik! Of waarom heb je nog een jaar nodig? Let dus op aan wie je welk prototype laat zien en wat hun verwachting hierbij zal zijn.

Om een goede user experience te kunnen bieden, zijn er extra rollen nodig in het softwareontwikkelp proces, zoals een interaction designer, een UI-designer of usability expert. Deze rollen vereisen echter heel andere vaardigheden dan die van een programmeur of architect, en kunnen daardoor het beste worden ingevuld door echte UX-specialisten.

Wist je dat Microsoft ook een masterclass heeft over user experience?

www.microsoft.com/netherlands/msdn/events/masterclasses.aspx
Meer over user experience is te vinden op MSDN:
msdn2.microsoft.com/en-us/architecture/aa699447.aspx

'Uit de praktijk' is een column van Clemens Schotte en Erwin van der Valk. Beiden zijn werkzaam als Development Consultant bij Microsoft Services.