

## SPECIALISEREN: USABILITY TESTER

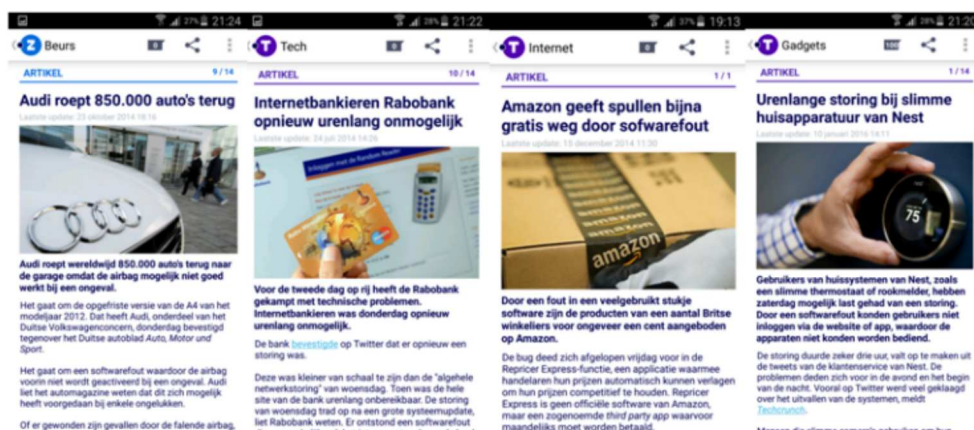
Door Erik van Veenendaal • [www.erikvanveenendaal.nl](http://www.erikvanveenendaal.nl) @ErikVeenendaal



*De veranderingen in het IT-landschap hebben ook gevolgen op de rol van tester. De tester heeft feitelijk twee opties: verbreding van kennis en kunde of specialiseren. In dit artikel worden kort beide opties besproken alvorens nader wordt ingegaan op de specialisatie Usability Tester. Het belang van usability zou inmiddels duidelijk moeten zijn in een wereld van mobile apps en websites waarbij veelal de stelling: 'the competition is only click away' van toepassing is. Een aantal technieken wordt aangereikt voor het testen van usability die goed passen in de huidige Agile wereld. Heuristic evaluation, cognitieve walkthrough en vragenlijsten worden behandeld onder de noemer 'Discount Usability Testen'.*

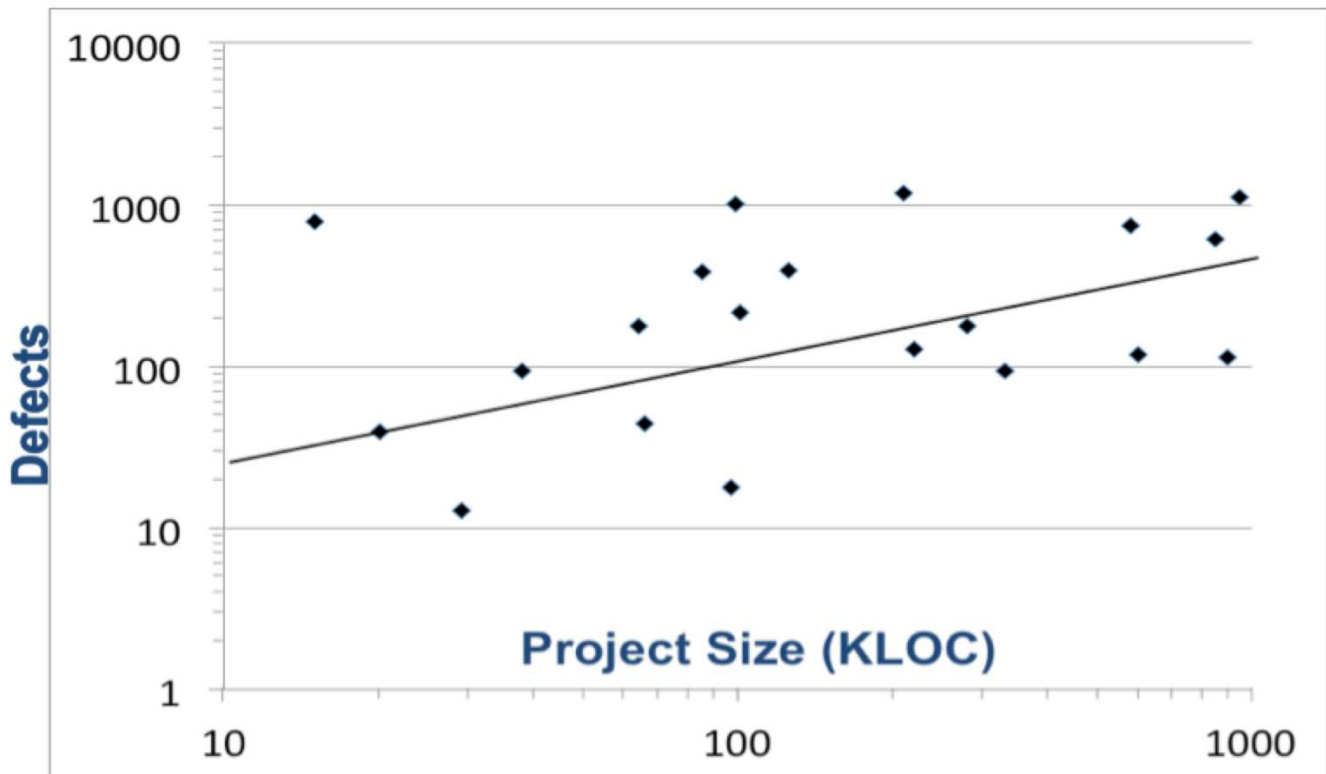
**No more Testers!**

Met enige regelmaat hoor of lees ik zogenaamde experts verklaren dat binnen een aantal jaren er geen testers meer nodig zullen zijn. Zonder hierover meteen een lange discussie te willen starten, is dit wat mij betreft praktijk versus theorie. In de dagelijkse praktijk zie ik de kwaliteits- en complexiteits-uitdagingen vooralsnog alleen maar groeien. Het aantal fouten is zeker niet aan het afnemen en je hoeft maar een krant open te staan, een nieuws website te bezoeken om weer een verhaal te lezen over de gevolgen van zoveelste software fout (zie figuur 1).



Figuur 1 : Softwarefouten [www.nu.nl](http://www.nu.nl)

Agile heeft allerlei positieve effecten, maar het is een misverstand dat daarmee ineens het kwaliteitsprobleem is opgelost. Een recent Amerikaans onderzoek laat zien dat het aantal fouten per KLOC in een Agile project niet vanzelfsprekend minder zijn, dan bij een traditionele project (zie figuur 2). Ongestructureerd ontwikkelen en testen en dan de transitie maken naar Agile ontwikkelen lost uiteraard niets op in ten aanzien van het kwaliteitsprobleem. Alleen indien tevens stappen worden gemaakt in de volwassenheid van het ontwikkelen en testen, dan pas kunnen verbeteringen worden bereikt. Wat mij betreft nog meer dan voldoende uitdagingen voorlopig voor de tester. Een vergelijkbare conclusie staat ook in het World Quality Report van 2014: 'despite continuous growth in adoption of Agile, many organizations are still facing the the same challenges as in previous years'.



Figuur 2 : Aantal fouten per KLOC Sequentieel vs. Agile  
(lijn = sequentiele projecten, punten = Agile projecten)

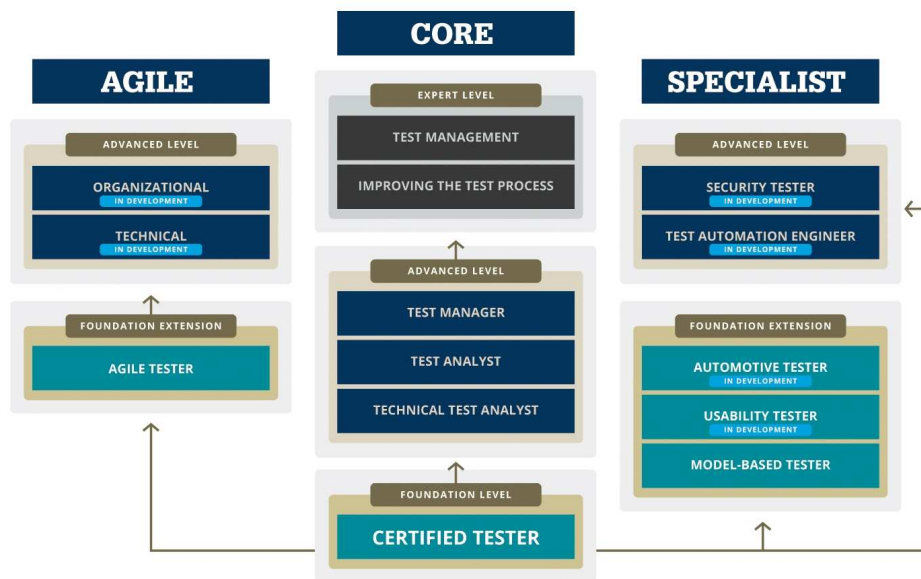
### Verbreiding van kennis en kunde

Wel is er een echter nadrukkelijk sprake van een verandering van de rol van de tester. De IT-wereld is de laatste jaren sterk veranderd en veel bedrijven zijn, op zijn minst bij een deel van de projecten, overgestapt naar Agile softwareontwikkeling. Een voorbeeld hiervan is dat de tester bij Agile nog meer betrokken bij de requirements dan ooit tevoren. Hij levert een bijdrage aan het specificeren van requirements en de bijbehorende acceptatiecriteria. Zo zijn er diverse trends in het vakgebied testen waarvan de (traditionele) tester op de hoogte dient te zijn en in staat moet zijn hierop te reageren. Kennis en kunde wordt een uitdaging voor veel testers in de nabije toekomst. Het is niet meer voldoende om alleen testkennis en kunde te hebben en alleen te beschikken over een ISTQB of TMap certificaat. Testers zijn veel minder vaak werkzaam in de veilige omgeving van het onafhankelijk testteam. We zullen nauw moeten samenwerken met vertegenwoordigers van de business en ontwikkelaars, elkaar moeten helpen daar waar nodig en als team trachten een product van hoogwaardige kwaliteit op te leveren. Van testers wordt verwacht dat zij onder andere over kennis beschikken betreffende de volgende zaken: domeinkennis, requirements engineering kennis & kunde, kunde ten aanzien van ontwikkelscripts, en sterke sociale vaardigheden, zoals communicatie en onderhandelen. In feite zal de tester zich sterk moeten gaan verbreden ten aanzien van kennis en kunde om ook in de toekomst een goede rol te kunnen vervullen in de veelal Agile projecten.

### Specialiseren

Er is echter nog een andere mogelijkheid voor de tester. Aangezien software producten steeds complexer worden en veelal moeten worden geïntegreerd in een open-omgeving, wordt het zogeheten niet-functioneel testen steeds belangrijker maar ook een steeds grotere uitdaging. Tegelijkertijd verwachten de business, de gebruikers en

klanten een hoog kwaliteitsniveau. Hiermee bedoelen ze veelal meer dan alleen de juiste en werkende functionaliteit. Ze willen een systeem dat betrouwbaar is, een goede performance heeft, gebruikt kan worden in diverse omgevingen, uitermate gebruikersvriendelijk is, et cetera. Om in deze uitdagende omgevingen niet-functionele aspecten zoals beveiliging, performance, betrouwbaarheid en usability goed te kunnen testen, zijn specialistische testers nodig. Deze full-time testspecialisten weten 'alles' van het testen van één bepaalde kwaliteitskarakteristiek. Zoals gezegd voorzie ik geen revolutie waarbij de testers zullen verdwijnen. Echter, er zijn trends en we zullen ons (geleidelijk) moeten aanpassen. Als tester heb je de keuze: het verbreden van je kennis en kunde of specialiseren in een bepaald (niet-functioneel) gebied. Interessant in dit opzicht is ook de ontwikkeling binnen het testcertificatieschema van ISTQB. In het zeer onlangs gepubliceerde nieuwe productportfolio (zie figuur 3) zie je een duidelijke stroming gericht op specialisatie.



Figuur 3 : ISTQB Product Portfolio

Eén van de specialistische modules die nu binnen ISTQB volop aandacht krijgt is Usability Tester. Tijdens de general assemblee te Singapore is de bèta versie van de gelijknamige syllabus gereleased. In het vervolg van dit artikel zal ik inzoomen op de usability specialisatie en daarbij een aantal testtechnieken aanreiken die ik zelf als uitermate bruikbaar heb ervaren.

### Usability

Het blijft vreemd dat niet minder dan zo'n 40% van de code die wereldwijd wordt ontwikkeld, direct of indirect te relateren is aan de user-interface, dat inmiddels in diverse onderzoeken 'voldoende aandacht voor usability' als een kritieke succesfactor voor IT-projecten is aangeduid, maar we nog steeds slechts een minimaal percentage van de testtijd besteden aan usability testen. Toch zijn er voldoende 'best practices'. In een tijd waarin alles Agile moet worden uitgevoerd met een sterke focus op resultaatgerichtheid; zo min mogelijk documentatie en in een korte tijdsbestek (sprint), denk ik in eerst instantie aan een aantal relatief eenvoudige usability technieken. Dit zijn technieken die snel zijn te leren en vaak niet veel meer kosten dan een één of twee dagen en derhalve een uitstekende kosten-batenverhouding hebben. In dit artikel zal ik nader ingaan op de Heuristic Evaluation, de Cognitieve Walkthrough en de SUMI vragenlijst methode. Deze drie technieken voldoen aan de hierboven gestelde

eisen en passen als zodanige prima in een Agile ontwikkelproject. Een mooie samenvattende term voor deze drie technieken is wat mij betreft 'Discount Usability Testen'. Natuurlijk kan ook gekozen worden voor de traditionele gebruikerstest, maar ook die moet dan anders worden benaderd en meer lean worden uitgevoerd. Vaak blijkt echter een gebruikerstest toch lastiger uit te voeren dan gedacht en ontbreekt de nodig kennis, kunde en ervaring.

### Heuristic Evaluation

Heuristic evaluation, ontwikkeld door Jacob Nielsen, is wellicht de meeste bekende en toegepaste techniek om usability te testen. Tijdens een heuristic evaluation wordt de usability van de user-interface systematisch beoordeeld door een aantal usability experts of testers ten opzichte van een aantal usability principes, de zogeheten 'heuristics' (zie tabel 1). Dit kan gebeuren op basis van het user-interface-ontwerp, een prototype of een gerealiseerd systeem. Uiteraard geldt ook hier, hoe eerder hoe beter. De herstelkosten van een fout zullen sterk toenemen indien het usability probleem pas in een laat stadium wordt ontdekt. Het uiteindelijke doel is potentiële usability problemen te vinden (en op te lossen). De heuristics die worden gebruikt, zijn opgesteld op basis van uitgebreid onderzoek naar problemen in de praktijk met usability. De praktijkproblemen zijn in dit onderzoek geanalyseerd en geclusterd. Per cluster is vervolgens een heuristic opgesteld waarmee de problemen (deels) hadden kunnen worden voorkomen. Veruit de meeste problemen in de praktijk blijken terug te herleiden zijn tot het niet voldoen aan deze elementaire usability principes.

1	Zichtbaarheid van de systeemstatus
2	Overeenkomst tussen systeem en de echte wereld
3	Controle en vrijheid voor de gebruiker
4	Consistentie en voldoen standaards
5	Foutpreventie
6	Herkenning in plaats van herinneren
7	Flexibiliteit en efficiëntie van gebruik
8	Esthetisch en minimalistisch ontwerp
9	Hulp aan gebruiker om fouten te herkennen, oorzaak te achterhalen en te herstellen
10	Help en documentatie

*Tabel 1 : Tien heuristics van Jacob Nielsen*

De heuristic evaluation is gebaseerd op software-inspectietechniek. Nielsen heeft de inspectietechniek aangepast en geschikt gemaakt om te kunnen gebruiken bij usabilitytesten. In het algemeen kun je stellen dat een heuristic evaluation lastig effectief is uit te voeren door slechts één persoon, aangezien die persoon nooit alle usabilityproblemen van de user-interface zal vinden. Praktijkervaringen in een groot aantal projecten hebben aangetoond dat verschillende testers veelal verschillende usabilityproblemen vinden. Het is daarom mogelijk de effectiviteit van de techniek aanmerkelijk te verbeteren door er meerdere testers (en wellicht ook ontwikkelaars, usability-experts en gebruikers) bij te betrekken. De effectiviteit neemt nog verder toe indien aan de verschillende testers verschillende rollen (op basis van de heuristics) worden toegewezen. Ervaren teams zijn zelfs in staat zo'n 100 usability problemen per uur te vinden.

Tijdens de heuristic evaluation inspecteert iedere tester de interface zelfstandig. Slechts nadat alle individuele evaluaties zijn afgerond is het toegestaan aan de testers om onderling te communiceren en de bevindingen samen

te voegen of te clusteren. Dit aspect is belangrijk en dient ervoor te zorgen dat de evaluaties onafhankelijk zijn en er geen onderlinge beïnvloeding plaatsvindt. Een heuristic evaluation sessie duurt normaal 1 à 2 uur. Langere sessies zijn soms nodig indien de user-interface erg complex of omvangrijk is met een groot aantal schermen en opties. Het is echter aan te raden om dit soort sessies dan op te splitsen in meerdere kortere sessies en eventueel te verdelen over meerdere personen. Het blijkt namelijk dat na zo'n twee uur de concentratie afneemt en er minder problemen worden gevonden. Tijdens een sessie zal de tester normaal gesproken meerdere keren door de interface heen gaan en het beoordelen ten opzichte van de heuristics. Hierbij wordt soms gebruikgemaakt van een checklist die veel voorkomende fouten aangeeft per heuristic. De heuristics worden gebruikt om fouten te vinden, maar veelal ook om de gevonden problemen te categoriseren.

In principe bepaalt de tester zelf hoe hij de evaluatie uitvoert. Een algemene richtlijn is wel dat de tester tenminste tweemaal door de user-interface heen gaat. De eerste keer is het doel inzicht te krijgen van de procesgang binnen de interface en kennis te maken met het systeem. Tijdens de tweede doorgang kan de tester dan de verschillende interface elementen beoordelen daarbij wetende hoe zij passen in het groter geheel. Zoals gezegd kan de heuristic evaluation uitstekend worden toegepast in een vroegtijdig stadium van ontwikkeling. De praktijk heeft aangetoond dat deze techniek, die vrij eenvoudig te leren is, een uitermate efficiënte usability testtechniek is.

### **Cognitieve Walkthrough**

De cognitive walkthrough is een usability reviewtechniek die begint met het opstellen van een aantal gebruikers-scenario's op basis van de requirements (user-stories) en/of een prototype. In tegenstelling tot de heuristic evaluation staan bij de cognitieve walkthrough de taken van de gebruiker centraal en wordt vanuit de taken getest. Je zult dus ook veelal andere fouten vinden dan bij een heuristic evaluation. Met de taakgerichte aanpak wordt getracht zoveel mogelijk fouten te vinden die ook in de praktijk kunnen optreden. De cognitieve walkthrough is derhalve een taakgerichte testtechniek die wordt uitgevoerd door een aantal usability experts / testers om op die manier fouten in het ontwerp van de user-interface te ontdekken.

De usability tester speelt feitelijk de rol van de gebruiker waarbij hij door de user-interface heen gaat en probeert de gebruikerstaak uit te voeren. Elke stap binnen de taak die de gebruiker zou moeten nemen wordt uitgebreid geëvalueerd. Als de usability tester vastloopt in de user-interface of de taak niet (volledig) kan uitvoeren duidt dit op een probleem. Wellicht dient de interface eenduidiger en eenvoudiger worden gemaakt. De analysefase bestaat uit het beoordelen van elke stap binnen de taak en telkens trachten te bedenken waarom de gebruiker de juiste actie zal kiezen (of juist niet). Deze analyse zijn mede gebaseerd op kennis en aannames over de achtergrond en doelstellingen van de gebruiker, maar ook het begrijpen wat het mentale proces is wat de gebruiker doormaakt bij het kiezen van de juiste actie.

Tijdens de cognitieve walkthrough stelt de tester zich telkens de volgende vier vragen:

1. Is het voor de gebruikers duidelijk welke actie hij moet uitvoeren?  
Bijvoorbeeld is de taak het printen van een document, waarbij eerst een printer moet worden geselecteerd. Weet de gebruiker dat hij dat eerst een printer moet selecteren?
2. Wordt de beschikbaarheid van betreffende actie door de gebruiker opgemerkt?  
Dit heeft te maken met de zichtbaarheid en begrijpbaarheid van de mogelijkheden in de user-interface.
3. Zal de gebruiker de juiste actie associëren met het resultaat dat hij wil bereiken?

4. Als de juiste actie is uitgevoerd, is het voor de gebruiker duidelijk dat er voortgang is in het afronden van de actie?

Dit heeft te maken met het beoordelen van de feedback van het systeem nadat de gebruiker de actie heeft uitgevoerd.

In de praktijk blijkt dat de walkthrough techniek ietwat lastiger te leren is dan bijvoorbeeld de heuristic evaluation en men ook meer ervaring moet hebben om deze goed te kunnen uitvoeren. Kennis van zowel de gebruikers van het systeem, maar ook algemene kennis en ervaring ten aanzien van het gedrag van gebruikers is noodzakelijk. Ook een cognitive walkthrough kan in elk stadium van de ontwikkeling van het ontwerp worden uitgevoerd, waarbij er gebruik kan worden gemaakt van een prototype of het reeds gerealiseerde systeem.

### **SUMI vragenlijst**

Een hele andere wijze van beoordelen is gebruikmakend van vragenlijsten. Met behulp van vragenlijsten wordt de mening van gebruikers gevraagd over de usability van het systeem. Hoewel in principe ook toepasbaar op prototypes of zelfs een user-interface-ontwerp document, worden vragenlijsten met name toegepast wanneer het systeem in (acceptatie)test is of als het in productie is. Dit is meteen een groot verschil (en nadeel) met de twee voorgaande testtechnieken. Daar waar de heuristic evaluation en cognitieve walkthrough al in een vroegtijdig stadium van de ontwikkeling kunnen worden toegepast, zal een vragenlijst in het algemeen pas in een relatief late fase van de ontwikkeling kunnen worden uitgevoerd. Soms wordt een vragenlijst gebruikt om de mening van acceptatietesters en gebruikers te krijgen over usability, tijdens of na afloop van een testfase.

Zodra er voldoende gebruikers de vragenlijst hebben ingevuld om er statistisch verantwoorde uitspraken mee te kunnen doen, volgt de verwerking en vervolgens de analyse van de resultaten. Hoewel de vragenlijst techniek een relatief goedkope manier is om usability te testen dan wel inzicht te krijgen in de usability van een user-interface, is een belangrijk ander nadeel dat de vragenlijsttechniek normaal gesproken geen gedetailleerde bevindingen oplevert. Als bijvoorbeeld de usability als onvoldoende wordt beoordeeld, zullen aanvullende testen moeten worden uitgevoerd om de concrete problemen op te sporen.

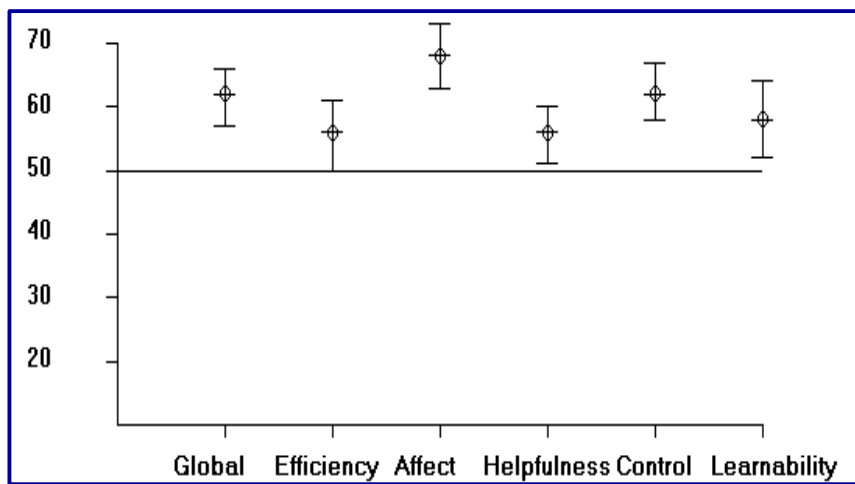
Natuurlijk kan men zelf een vragenlijst ontwikkelen, maar het is aan te raden gebruik te maken van een standaard vragenlijst. Het ontwikkelen van een onderbouwde en uitgebalanceerde vragenlijst met goede vragen is niet zo makkelijk als het wellicht lijkt. Bekende voorbeelden van veelgebruikte standaard vragenlijsten zijn onder andere SUMI (Software Usability Measurement Inventory) en WAMMI (Website Analysis and Measurement Inventory). In het vervolg van het artikel zal ik met name ingaan op de SUMI vragenlijst, WAMMI volgt feitelijk dezelfde structuur en opzet als SUMI, maar is specifiek ontwikkeld voor websites.

In principe is SUMI bruikbaar voor elk software product dat een user-interface heeft. Bij de SUMI vragenlijst wordt usability gemeten ten opzichte van een referentie database. Deze vertegenwoordigt als-het-ware de usability zoals gebruikers deze ervaren in hun dagelijkse praktijk. SUMI bestaat uit vijftig vragen waarbij elke stelling moet worden beantwoord met 'eens', 'geen oordeel' of 'oneens'. Hieronder een aantal voorbeelden van de stellingen waarop de gebruiker moet reageren:

- Dit product reageert te traag op de invoer van de gebruiker;
- Ik zou mijn collega's aanraden dit product te gebruiken;
- Het product is er op een bepaald tijdstip onverwachts mee gestopt;
- Het leren omgaan met het product levert in het begin veel problemen op;

- Met dit product weet ik vaak helemaal niet wat het volgende is dat ik moet doen;
- Ik vind het werken met dit product leuk;
- De berichten van het product zijn niet erg behulpzaam.

Zowel de SUMI als WAMMi vragenlijst zijn overigens beschikbaar in een groot aantal talen waaronder Nederlands. De verwerking van de gegevens levert bij SUMI een algemene score op ten aanzien van usability, maar ook een vijftal subscores ten aanzien van een aantal usability aspecten (zie figuur 4). Dit is uiteraard erg behulpzaam bij het analyseren van het resultaat. De scores worden bij SUMI berekend door een vergelijking te maken met de reeds eerdergenoemde referentiedatabase van eerder uitgevoerde SUMI-tests op allerlei soorten software producten.



*Figuur 4 : Voorbeeld van de SUMI score grafiek*

Figuur 4 geeft een voorbeeld van de output die wordt verkregen na een SUMI-test (waarbij gebruik is gemaakt van het ondersteunde tool). De scores van de test en bandbreedte van de scores (bepaald op basis van de standaard deviatie) worden afgezet tegen de gemiddelde score in de referentie database (weergegeven door de waarde 50). De usability scores zoals deze worden weergegeven in dit voorbeeld zijn derhalve positief; dit wil zeggen hoger dan de gemiddelde usability score en met een relatief kleine bandbreedte (weergegeven door de verticale lijn met een boven- en ondergrens).

Een productevaluatie met de SUMI vragenlijst geeft een duidelijke en objectieve meting van de mening van de gebruiker over de usability van het product voor zijn of haar taken. Het kwantitatieve karakter van SUMI is uiteraard een erg sterke eigenschap; usability wordt hiermee meetbaar gemaakt. Indien de score niet conform verwachting is, zullen echter aanvullende tests moeten worden gedaan om de details van de problemen te achterhalen. Soms wordt, om dit te ondervangen, een SUMI-test gecombineerd met een heuristisch evaluatie of cognitieve walkthrough waardoor men zowel een kwantitatieve usability meting heeft alsmede een gedetailleerde lijst met usability bevindingen.

### Ten slotte

Reeds een groot aantal jaren verzorg ik cursussen en spreek ik op (test)conferenties over het onderwerp Usability. Tijdens EuroSTAR heb ik al eens een 'award-winning' tutorial verzorgd met als onderwerp 'Discount Usability Testing'. Toch heb ik niet het idee dat het baanbrekend werk al heel veel heeft opgeleverd. Misschien waren we er

nog niet klaar voor, of was gewoon de behoefte er nog niet (in voldoende mate). Met de beweging naar meer specialisaties binnen het vakgebied testen, komen er wellicht usabilitytesters die dit verder gaan uitdragen en oppakken. In deze tijd van mobile, internet, et cetera zou je toch haast denken dat het niet anders kan. Tijdens een recente conferentie werd ik een aantal keren aangesproken door enthousiaste ex-cursisten die vertelden dat ze de bovengenoemde technieken met uitstekende resultaten nu echt gebruiken in hun projecten. Gaat het dan toch nog gebeuren?! ←