

**WHITE PAPER**

**Usability Testen volgens TMap NEXT**

**AUTEUR**

**IDEA: usability testen**



White Paper  
Usability Testen volgens TMap NEXT®

IDEA: Usability Testen



december 2009

Copyright Sogeti Nederland B.V. © te Vianen

Niets uit deze uitgave mag verveelvoudigd en/of openbaar worden gemaakt (voor willekeurig welke doeleinden) door middel van druk, fotokopie, microfilm, geluidsband, elektronisch of op welke andere wijze dan ook zonder voorafgaande schriftelijke toestemming van Sogeti Nederland B.V..

## INHOUDSOPGAVE

### Inhoudsopgave

<b>Voorwoord</b>	<b>6</b>
<b>1 Inleiding</b>	<b>7</b>
<b>1.1 Probleem</b>	<b>7</b>
<b>1.2 Oplossing</b>	<b>7</b>
<b>1.3 Doel van het whitepaper</b>	<b>7</b>
<b>1.4 Leeswijzer</b>	<b>7</b>
<b>2 Wat is usability?</b>	<b>8</b>
<b>2.1 Wat speelt een rol bij usability?</b>	<b>8</b>
<b>2.2 Hoe meet men usability?</b>	<b>9</b>
<b>2.3 Aspecten van usability</b>	<b>10</b>
<b>2.4 Conclusie</b>	<b>11</b>
<b>3 Usability testen volgens TMAP NEXT®</b>	<b>12</b>
<b>3.1 Usability en business doelstellingen</b>	<b>12</b>
<b>3.2 Usability en het Mastertestplan</b>	<b>14</b>
<b>3.3 Een aanpak voor usability testen</b>	<b>16</b>
3.3.1 Usability testplan	17
3.3.2 Iteraties	18
<b>4 Testtechnieken</b>	<b>20</b>
<b>4.1 Heuristic Evaluation</b>	<b>20</b>
4.1.1 Stappenplan	20
4.1.2 Heuristieken	21
4.1.3 Varianten	22
4.1.4 Situatie	22
4.1.5 Resultaat	22
<b>4.2 Perspective-Based Inspection</b>	<b>23</b>
4.2.1 Stappenplan	23
4.2.2 Perspectieven	23
4.2.3 Varianten	24
4.2.4 Situatie	24
4.2.5 Resultaat	24
<b>4.3 Interviews</b>	<b>25</b>
4.3.1 Stappenplan	25
4.3.2 Soorten interviews	25
4.3.3 Openheid vragen	26
4.3.4 Situatie	26
4.3.5 Resultaat	26
<b>4.4 Thinking aloud</b>	<b>28</b>
4.4.1 Stappenplan	28
4.4.2 Situatie	29
4.4.3 Resultaat	29
<b>4.5 Logging Actual Use</b>	<b>30</b>
4.5.1 Stappenplan	30
4.5.2 Situatie	31
4.5.3 Resultaat	31
<b>4.6 Observatie</b>	<b>32</b>
4.6.1 Soorten observaties	32
4.6.2 Stappenplan	32
4.6.3 Situatie	33
4.6.4 Resultaat	34
<b>4.7 A/B Testing</b>	<b>35</b>

4.7.1	KPI's	35
4.7.2	Stappenplan	35
4.7.3	Multivariate testing	36
4.7.4	Situatie	36
4.7.5	Resultaat	36
<b>5</b>	<b>Toetstechnieken</b>	<b>37</b>
<b>5.1</b>	<b>Consistency Inspection</b>	<b>37</b>
5.1.1	Verschillende methoden	37
5.1.2	Stappenplan	37
5.1.3	Situatie	38
5.1.4	Resultaat	38
<b>5.2</b>	<b>Cognitive Walkthrough</b>	<b>39</b>
5.2.1	Stappenplan	39
5.2.2	Varianten	40
5.2.3	Situatie	40
5.2.4	Resultaat	40
<b>5.3</b>	<b>Model-Based Inspection</b>	<b>42</b>
5.3.1	Stappenplan	42
5.3.2	CPM-GOMS	42
5.3.3	Situatie	42
5.3.4	Resultaat	43
<b>5.4</b>	<b>Pluralistic Walkthrough</b>	<b>44</b>
5.4.1	Stappenplan	44
5.4.2	Situatie	45
5.4.3	Resultaat	45
<b>5.5</b>	<b>Focus Groups</b>	<b>46</b>
5.5.1	Stappenplan	46
5.5.2	Varianten	47
5.5.3	Situatie	47
5.5.4	Resultaat	47
<b>5.6</b>	<b>Reverse Card Sorting</b>	<b>48</b>
5.6.1	Stappenplan	48
5.6.2	Situatie	48
5.6.3	Resultaat	49
<b>6</b>	<b>Preventieve maatregelen</b>	<b>50</b>
<b>6.1</b>	<b>Card Sorting</b>	<b>50</b>
6.1.1	Stappenplan	50
6.1.2	Situatie	51
6.1.3	Resultaat	51
<b>6.2</b>	<b>Prototype testing</b>	<b>52</b>
6.2.1	Soorten Prototypes	52
6.2.2	Stappenplan	52
6.2.3	Varianten	53
<b>6.3</b>	<b>Persona's</b>	<b>54</b>
6.3.1	Verschillende soorten persona's	54
6.3.2	Toepassing	55
6.3.3	Persona's opstellen	55
<b>7</b>	<b>Hulpmiddelen</b>	<b>57</b>
<b>7.1</b>	<b>Checklists</b>	<b>57</b>
<b>7.2</b>	<b>Infrastructuur</b>	<b>57</b>
7.2.1	Prototypen	57
7.2.2	Usability Laboratorium	58
7.2.3	Eyetracking	59
7.2.4	Mobile Device Camera	59
7.2.5	Spectacles Camera	60
7.2.6	Domecamera	60
<b>7.3</b>	<b>Tools</b>	<b>61</b>
7.3.1	Log data analyse tools	61
7.3.2	Automated inspection tools	61

7.3.3	Development tools	62
7.3.4	Usability laboratorium tools	62
7.3.5	Reverse card sorting tools	63
<b>8</b>	<b>Usability maturity model</b>	<b>64</b>
<b>8.1</b>	<b>Het Usability Maturity Model</b>	<b>64</b>
8.1.1	Stand van zaken	64
8.1.2	Verbetering	64
8.1.3	Verbetermodel	64
8.1.4	Usability Maturity Model	65
8.1.5	Onderverdeling	65
8.1.6	Aandachtsgebieden	66
8.1.7	Controlepunten	66
8.1.8	Usability Maturity Matrix	68
8.1.9	Gebruik van de Matrix	68
<b>8.2</b>	<b>Aandachtsgebieden en niveaus</b>	<b>69</b>
8.2.1	Toepassing fasering	69
8.2.2	Moment van betrokkenheid	70
8.2.3	Begroting en Planning	72
8.2.4	Metrieken	73
8.2.5	Gebruiker Onderzoek	75
8.2.6	Usability Tools	76
8.2.7	Usability werkplek	77
8.2.8	Stakeholder commitment	78
8.2.9	Usability awareness	79
8.2.10	Usability eisen	81
<b>Bijlage A</b>		<b>83</b>
<b>Bijlage B checklists</b>		<b>84</b>
<b>Nuttige links</b>		<b>84</b>
Navigatie:		84
Zoekmachines:		84
Design patterns		84
<b>Algemeen toepasbare checklist</b>		<b>84</b>

## VOORWOORD

Het document dat voor u ligt is een verzameling van usability testkennis die gedurende 5 jaar verzameld is door Sogeti. Het is een levendig document geworden dat een overzicht biedt over verschillende technieken en methoden. Dit document is dan ook vooral bedoeld als naslagwerk en hulpmiddel. Het is niet bedoeld om van voor tot achter te lezen.

In het begin van deze whitepaper wordt uiteengezet wat usability precies is (Hoofdstuk 2) en hoe men usability kan testen volgens de TMAP NEXT aanpak (hoofdstuk 3).

De volgende hoofdstukken zijn beschrijvingen van verschillende technieken, waar op een praktische manier wordt aangegeven in welke situaties men de technieken kan gebruiken en hoe men deze precies moet toepassen. De technieken zijn gegroepeerd in testtechnieken (hoofdstuk 4), Toetstechnieken (hoofdstuk 5) en preventieve maatregelen (hoofdstuk 6). De gereedschapskist voor usability testen wordt beschreven in het hoofdstuk over hulpmiddelen (hoofdstuk 7).

Hoofdstuk 8 gaat als laatste hoofdstuk in op de usability volwassenheid van een organisatie. In dit hoofdstuk wordt de TPI methode gebruikt om iets te zeggen over usability volwassenheid.

Bij het samenstellen van dit document zijn veel verschillende mensen betrokken geweest. In de eerste plaats is dit document in grote mate schatplichtig aan de schrijvers van het document "Usability testen: kenmerken van methoden" dat verscheen in 2006.

Daarnaast zijn er reviewers betrokken geweest van buiten de IDEA groep. Dit zijn Hugo Wijntjes van AfAS Software, Wouter Crama van Cap Gemini en Johan Vink, Andréas Prins en Leo van der Aalst van Sogeti.

Verder zijn in het verleden een hoop mensen betrokken geweest bij de gedachtevorming die heeft geleid tot dit document. Dit zijn de leden van de expertisegroep vanaf 2006 tot nu. Met name Mervyn Audhoe, Filip Joele, Robin Klein, Joost Janssen, René Bultema, Ricky Alves, Ivo ten Kate, Randy Semeleer, Thomas Veltman, Richard A. de Vries, Niek Fraanje, Bas Hoogendijk, Vincent Wijnen, Richard Enthoven, Martien Adema, Mark Schut en Cor van der Berg.

Niek Fraanje verdient daarbij extra aandacht omdat hij het leeuwendeel van het schrijfwerk en het beheersbaar houden van het reviewproces op zich heeft genomen. Bas Hoogendijk heeft in de laatste fase van het document een grote bijdrage geleverd door de eindredactie te voeren en te zorgen dat het geheel goed in lijn bleef.

Thomas Veltman

Trekker IDEA Software Control, usability testen.

# 1 INLEIDING

Usability of gebruiksvriendelijkheid gaat over het gemak waarmee software bediend wordt door eindgebruikers. Dit paper biedt een complete methode om vanaf het begin van het ontwikkelen van software te sturen op de gewenste mate van usability van het eindproduct.

## 1.1 Probleem

De concurrentiestrijd die op dit moment binnen veel bedrijfstakken op het web wordt uitgevochten is enorm. Denk hierbij aan branches zoals banken, verzekeringen, reisorganisaties, telecom, detailhandel en marketing. Voor klanten is het essentieel dat een aanbieder een aantrekkelijke, bruikbare en efficiënte website heeft. Als zij te lang moeten wachten, niet vinden waar ze naar zoeken of een site niet mooi vinden, zoeken ze direct een concurrent op. Een bedrijf dat rendement op het web wil behalen, kan daarom niet om usability heen.

Daarnaast staan sinds de crisis ook de marges steeds meer onder druk. Dit noopt bedrijven om efficiënter met hun IT om te gaan. Tegelijkertijd zien gebruikers dikwijls dat zij niet optimaal productief kunnen zijn, omdat de applicaties die zij gebruiken niet optimaal aansluiten op hun werkprocessen. Het verbeteren van de usability van applicaties, en het borgen van de usability van nieuwe applicaties, kan bedrijven helpen hun werknemers effectiever in te zetten.

## 1.2 Oplossing

Om deze redenen is het voor bedrijven van belang om de usability van sites en applicaties te borgen. De kennis en ervaring die de afgelopen decennia is opgedaan met het testen en toetsen van usability, heeft geleid tot een scala aan technieken, die in diverse stadia van het ontwikkelproces toegepast kunnen worden. Daarnaast is er voor het gestructureerd opzetten en uitvoeren van een testtraject de methode TMAP NEXT beschikbaar in de markt.

## 1.3 Doel van het whitepaper

Dit whitepaper heeft tot doel om de lezer een praktische handleiding te geven om usability aan de hand van TMAP NEXT® een duidelijke plek te geven in het testtraject. Dit kan zowel door testmanagers gebruikt worden om usability een duidelijke positie te geven in de teststrategie als door testers als praktische handleiding bij het uitvoeren van usability testen en toetsen. Daarnaast geeft het organisaties een methode om structureel bewust om te gaan met usability en de borging van haar producten.

## 1.4 Leeswijzer

In hoofdstuk 2 wordt het begrip usability uitgelegd en onderverdeeld in aspecten. In hoofdstuk 3 wordt beschreven hoe usability een plek gegeven kan worden in een teststrategie die volgens TMAP NEXT opgesteld wordt. Hoofdstuk 4, 5 en 6 geven een overzicht van respectievelijk test-, toets-, en realisatietechnieken, waarbij voor elke techniek uiteengezet wordt hoe dit in de praktijk kan worden toegepast. Hoofdstuk 7 staat in het teken van de infrastructuur waarmee deze technieken ondersteund kunnen worden. Tenslotte beschrijft hoofdstuk 8 het usability maturity model, waarmee een organisatie kan nagaan hoe volwassen zij zijn in het omgaan met usability, en kunnen bepalen welke acties zij uit moeten zetten om het gewenste niveau van volwassenheid te bereiken.

## 2 WAT IS USABILITY?

De meeste mensen binnen de ICT weten dat usability 'iets' te maken heeft met gebruiksvriendelijkheid of gebruiksgemak. Zonder gedetailleerdere kennis en begrip van usability, is het niet haalbaar om de kwaliteit van een product op dit gebied te borgen. Deze paragraaf heeft als doel dit inzicht te geven.

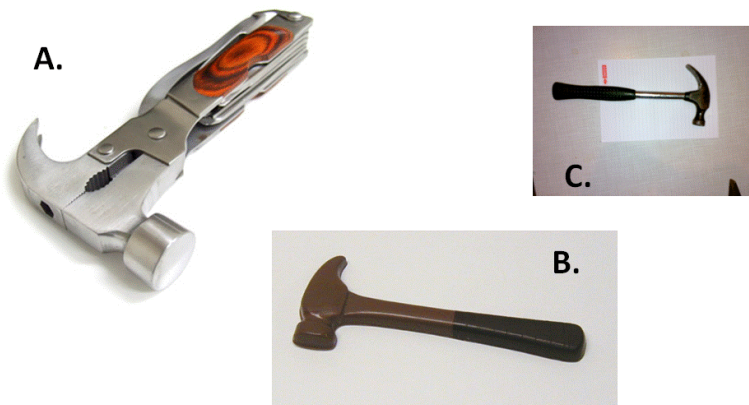
ISO hanteert de volgende definitie van usability:

*Usability is de mate waarin een product door bepaalde gebruikers in een bepaalde gebruikersomgeving kan worden gebruikt om bepaalde doelen effectief, efficiënt en naar tevredenheid te bereiken.*

Aan de hand van deze definitie zal een beeld gegeven worden van wat usability is.

### 2.1 Wat speelt een rol bij usability?

Usability is een kwaliteitsattribuut aan de hand waarvan bepaalde kenmerken van een product, het *testobject*, gekwalificeerd en beoordeeld kunnen worden. Dit wordt geïllustreerd met het volgende voorbeeld. Bekijk de volgende drie hamers:

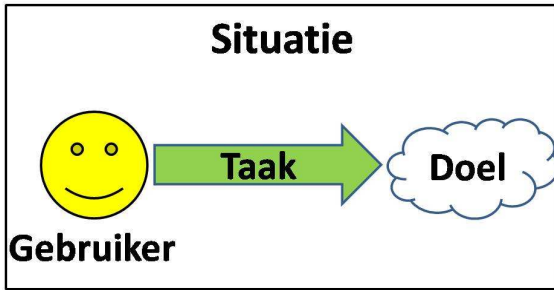


Welke van deze drie hamers heeft de beste usability?

Op het eerste gezicht is een mening snel gevormd. Veel mensen kiezen voor hamer C omdat dit de meest gangbare hamer is om spijkers ergens in te slaan. Maar in die uitspraak zit een onbewuste aanname: dat er een gebruiker als taak heeft om met de hamer een spijker in te slaan. Hamer A heeft veel andere functies en is daarom niet meteen het handigste hulpmiddel om ergens een spijker in te slaan. Maar als je een fles moet ontkurken is deze hamer verreweg de beste. Dus voor het bepalen van de usability is het doel van belang. Wat wil men met het hulpmiddel bereiken?

Verskillende gebruikers hebben verschillende eisen aan hulpmiddelen. De een hamert beter met een zware hamer, de ander met een lichte. De een vindt een hamer met veel functies zoals A beter, een andere ziet meer in hamer die eenvoudig is zoals C.





Voor het doen van een uitspraak over de usability van een testobject is dus meer van belang dan alleen de eigenschappen van het testobject en het doel waarvoor het is ontworpen. Vandaar dat in de definitie van usability: *Usability is de mate waarin een product door bepaalde gebruikers in een bepaalde gebruikersomgeving kan worden gebruikt...De gebruiker is degene die het doel bepaalt en tevens degene met het testobject de taak heeft om dit doel te gebruiken. Daarnaast is de gebruikersomgeving of situatie hiervoor van belang.*

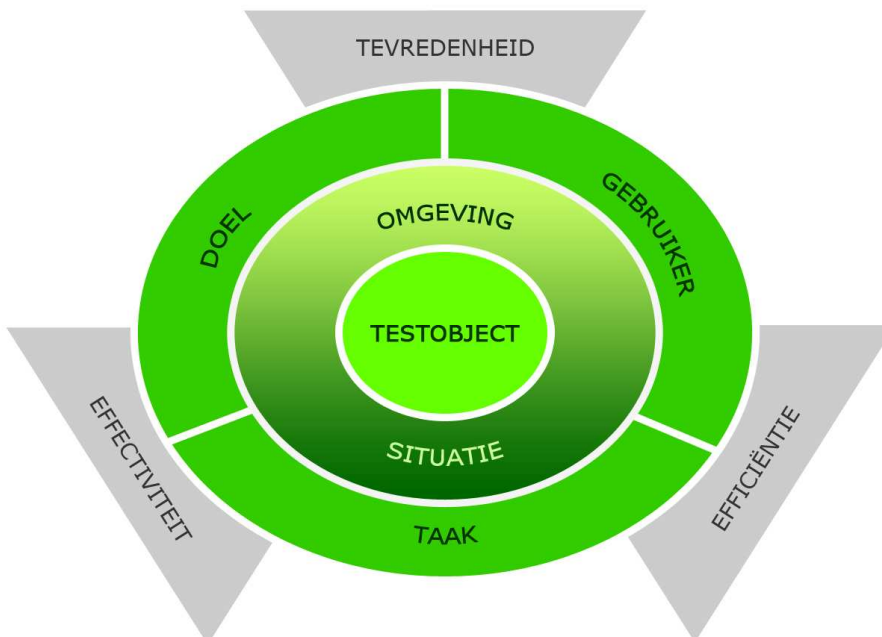
Deze onderlinge afhankelijkheid van situatie, taak, testobject, doel en gebruiker maakt het soms lastig om algemene uitspraken te doen over usability. Smaken verschillen en wat voor de ene gebruiker werkt, werkt niet voor de andere. Dit kan worden verholpen door naar gebruikerssegmenten te kijken: heeft een object een bepaalde usability voor gebruikers met bepaalde overeenkomstige kenmerken?

Gelukkig is het ook vaak zo dat wél een algemene uitspraak mogelijk is over usability. In ons voorbeeld is hamer B van chocola; om spijkers in te slaan heeft niemand daar wat aan.

## 2.2 Hoe meet men usability?

Nu is vastgesteld dat usability een uitspraak doet over hoe de applicatie gebruikt wordt door een gebruiker om een bepaald doel te bereiken, kan de usability van een object bepaald worden en kan hierover worden gecommuniceerd.

Er is alleen nog wel terminologie nodig om aan te geven of de usability van een product goed of slecht, hoog of laag is. Hiervoor is een meer objectieve, specifieke en meetbare manier nodig om usability te beschrijven. Deze manier van uitdrukken moet ook faciliteren bij het uitvoeren van tests en toetsen. Het tweede deel van de definitie van usability hierboven geeft de hiervoor benodigde terminologie: *...om bepaalde doelen effectief, efficiënt en naar tevredenheid te bereiken.*



Efficiëntie geeft een bepaalde mate van productiviteit aan. "Gebruiker X slaat met hamer Y 15 spijkers per minuut in". Effectiviteit geeft een bepaalde mate van resultaat aan "Gebruiker X slaat met hamer Y 7 van de 8 spijkers er goed in". Tevredenheid kan men op een andere manier specifiek en meetbaar maken "Gebruiker X geeft aan dat hij tevreden is over het gebruik van hamer Y. Hij geeft het gebruik een 7 op een schaal van 10".'

Deze termen stellen ons in staat exacter te formuleren, hoe de usability van een product beoordeeld is. De termen stellen ons tevens in staat om op hoog niveau over usability te communiceren en er eisen aan te stellen. Het maakt het ook meetbaar. Ze stellen ons in staat om afspraken te maken over usability met de opdrachtgever.

## 2.3 Aspecten van usability

Met deze definitie van usability kan tegen requirements vanuit de opdrachtgever getest worden. Echter, wanneer aan een requirement niet voldaan wordt ("de gebruiker werkt niet efficiënt, hij doet te lang over een transactie") levert dit weinig informatie op voor het oplossen van dit probleem.

Daarnaast geven deze kwaliteitsaspecten geen ruimte om rekening te houden met verschillende typen gebruikers. De voorkennis van de gebruiker kan bijvoorbeeld invloed hebben op de usability. Een applicatie die ondoorzichtig is voor beginnende gebruikers, kan bijzonder efficiënt zijn voor ervaren gebruikers.

Voor het uitvoeren van usability testen is usability onderverdeeld in een verzameling aspecten: de usability aspecten van Jakob Nielsen. Deze zijn hieronder weergegeven en kort toegelicht aan de hand van het voorbeeld van de hamer (paragraaf 2.1):

- *Leerbaarheid*: Hoe snel leert een onervaren gebruiker het gebruik van de hamer? Hoe lang duurt het voordat hij evenveel spijkers erin slaat als een ervaren gebruiker?
- *Efficiëntie (van de ervaren gebruiker)*: Als de gebruiker ervaren is, hoeveel moeite kost het dan om spijkers erin te slaan?
- *Memorabiliteit*: Als gebruikers de hamer een tijd niet hebben gebruikt, in hoeverre kunnen ze hem dan na een tijd weer als ervaren gebruiker gebruiken?
- *Fouten*: Wat kan er allemaal misgaan bij het gebruik van de hamer? Hoeveel van deze fouten maken de gebruikers bij het timmeren? Hoe ernstig zijn deze fouten en hoe gemakkelijk kunnen gebruikers deze fouten herstellen?
- *Tevredenheid*: Hoe plezierig is het om het product te gebruiken?

In een aantal van deze aspecten (leerbaarheid, memorabiliteit, efficiëntie) is de voorkennis van de gebruiker al meegenomen; dit maakt het gemakkelijker om algemene uitspraken te doen. "Onervaren gebruikers hebben moeite om de applicatie te leren gebruiken". Het benoemen van het soort gebruiker maakt het gemakkelijker om de impact van problemen of het belang van requirements te bepalen. Wordt een testobject meer gebruikt door ervaren gebruikers of door meer onervaren gebruikers? Wordt de applicatie af en toe gebruikt of regelmatig? Het werken met deze aspecten van usability stelt de opdrachtgever in staat om te kiezen op welke aspecten het wordt geëvalueerd.

Het 'Fouten' aspect focust op gebeurtenissen die het de gebruiker bemoeilijken of onmogelijk maken zijn doel te bereiken. Soms is een fout alleen een misstap zonder gevolgen en kan de gebruiker weer verder met slechts wat vertraging. Dit zijn minimale fouten. Het kan ook dat de gebruiker per ongeluk (onherstelbare) schade aanricht. Dit zijn uiteraard kwalijkere fouten. Denk bijvoorbeeld aan een simpele tekstverwerker die je in staat stelt slechts drie keer 'ongedaan maken' te gebruiken. Een gebruiker kan hierdoor per ongeluk stukken tekst weggooiden zonder dat deze fout hersteld kan worden.

## 2.4 Conclusie

We hebben dus twee verschillende verzamelingen kwaliteitsaspecten waarin usability kan worden onderverdeeld, die beide hun voordelen hebben voor verschillende situaties. Ze geven beide een bepaalde invalshoek op de usability van een applicatie. In principe kan men deze begrippen door elkaar gebruiken, men moet dan echter wel in de gaten houden welke begrippen welke precieze betekenis hebben en wat de onderlinge relatie is. Bijvoorbeeld: de leerbaarheid van een applicatie zegt wat over de algemene efficiëntie zoals ISO dit aangeeft.

De relaties tussen de ISO normen en de kwaliteitsaspecten van Nielsen staan hieronder aangegeven. Een kruis in een cel betekent, dat uitspraken over deze kwaliteitsattributen op elkaar van toepassing kunnen zijn.

	ISO	Efficiëntie	Effectiviteit	Tevredenheid
Nielsen				
Leerbaarheid	X		X	
Efficiëntie	X			
Memorabiliteit	X		X	
Fouten			X	
Tevredenheid				X

Over hoe deze aspecten worden gebruikt in een usability test kan men meer lezen in hoofdstuk 3.

### **3 USABILITY TESTEN VOLGENS TMAP NEXT®**

Barry Boehm (1985) toonde aan dat de herstellkosten van een fout in de software exponentieel stijgen naarmate hij later in het ontwikkeltraject wordt gevonden. Dit geldt voor usability fouten nog eens extra: de grootste usability problemen komen meestal voort uit ontwerpfouten. Het is dus zaak om zo vroeg mogelijk in het traject eventuele usabilityproblemen te ontdekken.

Voor usability is het dus van extra groot belang om het testproces gestructureerd uit te voeren. Omdat een gestructureerd testproces een van de essenties is in TMap NEXT®, is dit hét hulpmiddel bij uitstek om een usability test (en toets) traject op te zetten. In dit hoofdstuk wordt uitgelegd hoe een dergelijk traject kan worden aangepakt.

#### **3.1 Usability en business doelstellingen**

De usability van een applicatie kan grote invloed hebben op het behalen van de business doelstellingen die de opdrachtgever probeert te bereiken. Usability kan bijvoorbeeld van invloed zijn op sales, marketing en op interne bedrijfsprocessen.

Wanneer een applicatie voor e-business ontwikkeld wordt, is het van belang dat gebruikers de producten die zij willen aanschaffen kunnen vinden. Bijna net zo belangrijk is het dat gebruikers makkelijk toegang hebben tot andere producten waar zij mogelijk ook in geïnteresseerd zijn. Daarnaast is het ook vitaal dat zij snel en gemakkelijk deze producten af kunnen rekenen. Uit studies blijkt dat klanten bij problemen met websites zeer snel afhaken en een vergelijkbare (concurrerende) website opzoeken. Tenslotte moeten zij kunnen vertrouwen op het bedrijf achter de website. Hoewel het aantal internettransacties elk jaar toeneemt, is het kopen van producten op internet voor veel gebruikers nog steeds een drempel. Een professionele indruk, duidelijke productbeschrijvingen met illustraties en een uitgebreide beschrijving van de geschiedenis van het bedrijf achter de website kunnen onervaren gebruikers over die drempel heen helpen.

Veel van de technieken en tools die gebruikt worden om usability te testen of toetsen, worden ook gebruikt in de wereld van marketing. Sinds de opkomst van e-business zijn marketing en usability steeds meer naar elkaar toe gegroeid. De techniek A/B testing (zie paragraaf 4.8) is zelfs oorspronkelijk ontwikkeld als marketingtest. Het analyseren van klikgedrag van gebruikers, kan gebruikt worden om usability problemen op te sporen, maar ook om de effectiviteit van advertenties te meten. Met behulp van usability testen kan ook bepaald worden of gebruikers makkelijk dié acties kunnen of gaan doen die de opdrachtgever wil dat zij gaan doen.

### Marketing en usability

Hieronder is een online advertentie van The Economist weergegeven. In deze advertentie krijgt de bezoeker van de site 2 opties. The Economist had als doel om zoveel mogelijk "print" subscriptions te verkopen. Uit onderzoek bleek 68% van de bezoekers de eerste optie te kiezen, en 32% de tweede optie.

The image shows two side-by-side screenshots of a subscription page for The Economist. Both screenshots have the heading "SUBSCRIPTIONS" and "Welcome to The Economist Subscription Centre". They both ask the user to "Pick the type of subscription you want to buy or renew." The left screenshot shows two options: "Economist.com subscription - US \$59.00" with a 68% selection rate, and "Print & web subscription - US \$125.00" with a 32% selection rate. The right screenshot shows three options: "Economist.com subscription - US \$59.00" with a 16% selection rate, "Print subscription - US \$125.00" with a 0% selection rate, and "Print & web subscription - US \$125.00" with an 84% selection rate. The descriptions for each option are identical to those in the left screenshot.

Op zich lijkt hier onmogelijk een usability probleem in te zitten. De pagina is duidelijk en je zou hier niet op vast kunnen lopen. Het lijkt er dus op dat het maximale resultaat voor deze pagina bereikt is.

Door met de opmaak van de pagina te experimenteren en te usability testen, is het toch mogelijk een pagina te creëren die een beter resultaat geeft. Door het toevoegen van een extra optie voor dezelfde prijs als de "print en web subscription" lijkt het alsof men een web abonnement van 59 euro gratis krijgt. Nu kiest 84% van de bezoekers voor een "print en web subscription". Dit beïnvloed mensen om de "print en web subscription" te kiezen.

Het ontwikkelen van dit scherm lijkt in eerste instantie geen problemen te bevatten. Maar op het moment dat men vanuit marketing optiek denkt: er is nog  $100 - 32 = 68\%$  te winnen, dan kan dit een grote verbetering van het resultaat opleveren.

(Voorbeeld: Copyright Dan Ariely 2009)

Daarnaast kan een bepaalde doelgroep onderzocht worden. Wanneer het bijvoorbeeld de bedoeling is om jongeren te bereiken, is het cruciaal om te onderzoeken hoe zij tegen de website aankijken, en hoe zij zich gedragen op de site. Slaat de website aan, of laten zij bepaalde delen van de site geheel links liggen? Hoe tevreden zijn zij als zij de site gebruiken? Kan de website zo aangepast worden, dat een bredere doelgroep zich ook tot de site aangetrokken voelt?

Wanneer een nieuwe applicatie ontwikkeld wordt, is het vergemakkelijken van een bestaand proces een vaak genoemde doelstelling. Het is in dit geval cruciaal om te onderzoeken of deze vergemakkelijking ook ervaren wordt door de mensen die de applicatie daadwerkelijk moeten gebruiken. Als zij niet met de applicatie uit de voeten kunnen, is het niet ondenkbaar dat de verwachte opbrengsten niet gehaald zullen worden. In extreme gevallen kan het zelfs zo zijn dat gebruikers een applicatie weigeren te gebruiken uit ontevredenheid. Leerbaarheid kan ook een factor zijn: wanneer een afdeling veel verloop kent, is het belangrijk dat de bediening van de applicaties die zij gebruiken snel en gemakkelijk aan te leren is. Daardoor kunnen de opleidingskosten

beperkt worden, en zullen nieuwe medewerkers snel productief zijn en een tevreden gevoel krijgen over hun nieuwe werk.

### 3.2 Usability en het Mastertestplan

Zoals eerder genoemd is usability een kwaliteitsattribuut dat, om het beste resultaat te bereiken, aandacht verdient in het volledige ontwikkelproces. De opdrachtgever en de testmanager zullen daarom ook voor usability moeten bepalen wat de testdoelen zijn. Hoe belangrijker de rol van usability is bij het behalen van de business doelstellingen, des te meer aandacht verdient het formuleren van de testdoelen voor usability. Als algemene stelregel geldt dat hoe groter en diverser de gebruikersgroep is en hoe groter de afstand is tot die groep (zijn het werknemers, klanten of potentiële klanten?) hoe groter het belang wordt van usability.

Hieronder staan een aantal voorbeelden van testdoelen die de klant kan hebben die een usabilitytest in meer of mindere mate rechtvaardigen.

- De applicatie is een website
  - Er is productverkoop afhankelijk van de website en dit moet goed blijven lopen en/of
  - de applicatie wordt veel gebruikt door klanten die geen hinder mogen ondervinden en/of
  - De applicatie is voor veel gebruikers de eerste kennismaking met de organisatie
  - er zijn eisen vanuit de overheid m.b.t. usability en toegankelijkheid
- De applicatie is een verkoopbaar software product
  - De gebruikersgroep is geneigd onderling te communiceren over de bruikbaarheid van de applicatie en/of
  - De kopers zijn ook de gebruikers en/of
  - De gebruikersgroep heeft weinig ervaring met software en/of
  - Het wordt op afwijkende middelen gebruikt (bv. mobiel)
- De applicatie wordt intern gebruikt bij een organisatie.
  - Efficiëntie is van belang
  - De gebruikersgroep is groot en moeilijk te bereiken met cursussen
  - Medewerkertevredenheid is van belang
  - De opleidingskosten zijn hoog

Hierbij moet worden opgemerkt dat bij een website en een verkoopbaar software product aandacht voor usability van een *nóg* groter belang is dan voor een interne applicatie. Bij de ontwikkeling van dit soort software wordt gewoonlijk rekening gehouden met usability en usability testen. De extra punten achter de bullets zijn dan triggers om het gewicht van de usability test te bepalen.

Hierna worden in de productrisicoanalyse de risico's van het niet halen van deze testdoelen ook meegenomen. Bij webapplicaties kan bijvoorbeeld gedacht worden aan klanten die voortijdig afhaken, vermindering van het professionele imago, het niet bereiken van de doelgroep of zelfs sancties vanuit de overheid. Bij software producten kunnen de verkopen lager uitvallen, kan de service afdeling onnodig veel vragen krijgen (met alle bijbehorende kosten) of kan een bedrijf zelfs vaste klanten verliezen. Bij interne applicaties spelen vaak hele andere risico's; denk bijvoorbeeld aan een verminderde efficiëntie in een bedrijfsproces, aan ontevredenheid onder werknemers, aan hoge opleidingskosten voor nieuwe gebruikers, of aan het geheel niet gebruiken van een applicatie. Aan de hand van de toegekende risicoklassen kan gestructureerd vastgesteld worden hoeveel aandacht usability verdient ten opzichte van andere kenmerken en testdoelen.

Bij usability testen gaat het echter niet alleen om risico's, maar ook om *kansen*. Een kans is een omgekeerd risico. Als een bepaald onderdeel van een applicatie veel efficiënter kan, dan kan dat een risico zijn, namelijk dat er geld wordt verloren door werknemers die minder productief zijn. Aan de andere kant kan je het ook zien als een kans om de productiviteit van je medewerkers te verhogen door een efficiëntere applicatie te ontwikkelen. Usability testen kan, zeker in een vroeg stadium, worden gebruikt om verschillende oplossingen uit te proberen

en te experimenteren en zo tot de meest efficiënte oplossing te komen. Het voorbeeld van The economist uit paragraaf 3.1 illustreert ook dit proces.

Vervolgens moet bij het uitwerken van de teststrategie al nagedacht wordt over de vraag hoeveel aandacht usability testen in elke testsoort krijgt. Vanaf het toetsen van ontwerpdocumentatie tot de laatste testen op een volledig ontwikkeld systeem kunnen usability testen plaatsvinden. Beperk daarom de usability inspanningen niet tot de gebruikersacceptatiefase, maar geef usability door het hele testproces aandacht.

Het is zaak om zo snel mogelijk een usabilityspecialist te betrekken die kan gaan bepalen op welke manier de usability van het testobject het beste getest kan worden. De usabilityspecialist is verantwoordelijk voor het opstellen van de usabilitytestaanpak voor het hele ontwikkeltraject.

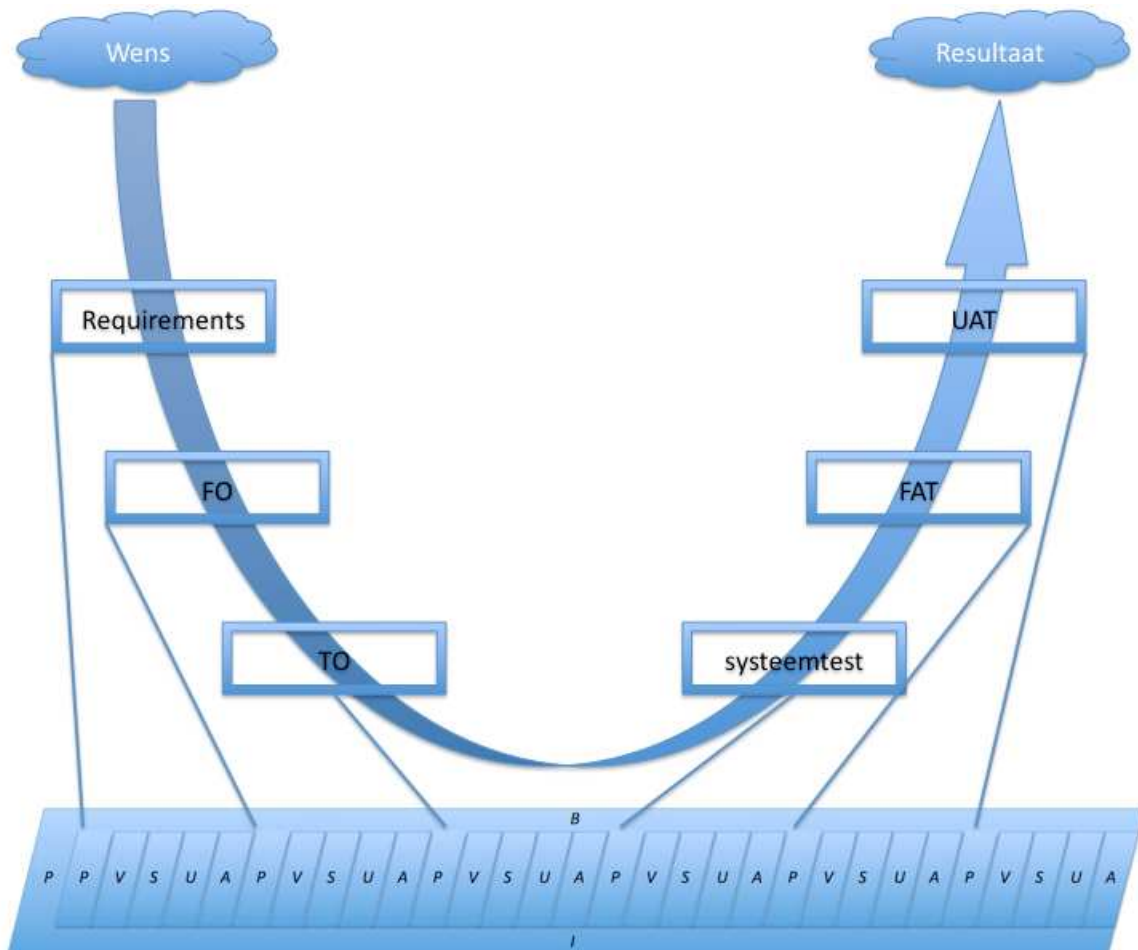
### 3.3 Een aanpak voor usability testen

Het testen en toetsen van de usability van een applicatie is een sterk iteratief proces. Elke actie is erop gericht te komen tot verbeteringen die de applicatie meer usable maken. De beste resultaten worden verkregen, als hier vroeg in een ontwikkelproces mee begonnen wordt. Wanneer de applicatie al ontwikkeld is, is veel minder winst te behalen, en worden verbeteringen kostbaar.

In een iteratief proces kan ervoor gekozen worden elke iteratie met gebruikers te testen, of zelfs ingrijpendere methoden als user-centered design toe te passen. Door in nauwe samenwerking met ontwerpers en ontwikkelaars te experimenteren en verschillende oplossingen te testen kan tot de beste oplossingen gekomen worden.

Ook in een traditioneel waterval traject kan dit iteratieve proces uitgevoerd worden. Het is dan belangrijk zoveel mogelijk usability testen uit te voeren in elke fase van ontwikkeling. Op deze manier kan het borgen van de usability toch een iteratief karakter krijgen, ook al is het proces eigenlijk lineair.

Een gestructureerde usability testaanpak bestaat uit dezelfde fasen waaruit het testproces volgens TMap NEXT® bestaat. Het traject is vergelijkbaar met een iteratief testtraject: na een overkoepelende planningsfase, waarin de usability specialist het usabilitytestplan opstelt, volgen meerdere trajecten waarin de fasen voorbereiding, specificatie, uitvoering en afronding zich herhalen voor elke fase van het project waarin usability testen of toetsen uitgevoerd worden. Door het hele traject loopt een fase beheer waarin met name bevindingenbeheer en rapportage uitgevoerd worden, en een fase infrastructuur waarin de benodigde omgevingen en tooling gerealiseerd en beheerd worden.





### 3.3.1 Usability testplan

De onderdelen van een usability testplan zijn in principe hetzelfde als de onderdelen van een normaal testplan, zoals beschreven in TMap NEXT®. Een aantal onderdelen verdienen speciale aandacht bij een usabilitytestplan. Deze staan hieronder toegelicht.

#### Vaststellen testbasis

Zie ook pagina 171 van TMap NEXT®. Testbasis is in het geval van usability een geval apart. In het begin van het project zijn het de *bewuste* en *onbewuste* eisen die de gebruiker stelt aan een applicatie. Het probleem zit hem natuurlijk in de woorden *onbewuste eisen*. Als de gebruiker zelf niet weet dat hij een bepaalde eis heeft, dan wordt het ook moeilijk om deze op een of andere manier vast te leggen.

Het is dus zaak dat vroeg wordt gekozen hoe de testbasis vastgelegd gaat worden.

Hierin kunnen een aantal keuzes gemaakt worden.

- Deze eisen vastleggen door een persona (zie ook paragraaf 7.3) te definiëren.
- door focus groups (zie ook paragraaf 6.5) met gebruikers in contact komen
- individuele sessies doen met gebruikers.
- Deze eisen niet vastleggen. Dan komen deze pas op het moment van testen boven water.

#### Bepalen planning

Zie ook pagina 183 van TMap NEXT®. Vanwege het sterk iteratieve karakter van usability testen vormt de planning vaak een uitdaging: want hoe goed is goed genoeg?

Een manier om hiermee om te gaan is timeboxen. Geef een bepaalde test een van te voren afgebakende hoeveelheid tijd om te besteden. Als er geen tijd meer is voor testen, wordt het testobject opgeleverd. Dit maakt het proces beheersbaar, zodat voorkomen wordt dat onevenredig veel tijd besteed wordt aan details. Er zit echter geen harde controle in of het testobject van voldoende kwaliteit is.

Wanneer kwaliteit een grote rol speelt, kan daarom ook gewerkt worden met acceptatiecriteria. Het testobject wordt opgeleverd wanneer aan alle criteria voldaan wordt, ongeacht hoe lang dat duurt. Het is helaas lastig om goede criteria te bedenken. Ze moeten voldoende SMART zijn, anders kan niet gecontroleerd worden of de test geslaagd is. Voorbeelden van SMART acceptatiecriteria zijn:

- Het gemiddelde cijfer dat gebruikers geven aan een product is 7,5
- 85 % van de gebruikers zegt tevreden te zijn over dit tussenproduct.
- 80 % van de geteste gebruikers kan de taak afronden binnen 1 minuut.

In een traditioneel waterval traject is het plannen van een iteratief usability testtraject een uitdaging. In elke fase van het ontwikkeltraject kan ruimte ingepland worden voor usability testen of toetsen. Dit kan echter op verzet stuiten bij het projectmanagement. Het beeld bestaat dat in een dergelijk lineair traject, usability testen pas aan het einde uitgevoerd worden. Vaak is er angst om gebruikers te confronteren met tussenproducten die mogelijk onvolkomenheden bevatten, of functionaliteiten die uiteindelijk geschrapt moeten worden.

Maak daarom duidelijk, dat de tijdens het gehele traject uitgevoerde activiteiten de planning van de acceptatietesten aanzienlijk inkort. Gebruikers hebben immers al tijdens het gehele traject meegekeken, of hun belangen zijn behartigd door de usability specialist. De gedane bevindingen kunnen vroeg in het traject nog snel en goedkoop verholpen worden. Wanneer het testobject al af is, zijn verbeteringen vaak veel moeilijker door te voeren en veel duurder.

#### Toewijzen testeenheden en testtechnieken

In een traditioneel waterval traject is het cruciaal om exact te plannen welke techniek in welke fase uitgevoerd wordt, om de technieken zo effectief mogelijk in te zetten. Dit moet dan ook terugkomen in het usability testplan. Hoe dit precies uit te voeren is afhankelijk van diverse factoren: denk aan deadlines, externe leveranciers, beperkte betrokkenheid van gebruikers, late betrokkenheid van testers etc.

De onderstaande tabel geeft voor elke fase in het V-model weer, welke technieken het meest geschikt zijn om in die fase te hanteren. Deze technieken worden in hoofdstukken 4, 5 en 6 uitgebreid besproken.

<b>Fase</b>	<b>Technieken</b>						
<b>Requirements</b>	Personas	Focus groups	Interviews				
<b>Functioneel ontwerp</b>	Card Sorting	Cognitive walkthrough	Consistency inspection	Model-based inspection	Pluralistic walkthrough	Focus groups	Reverse card sorting
<b>Technisch ontwerp</b>	Card Sorting	Cognitive walkthrough	Consistency inspection	Model-based inspection	Reverse card sorting		
<b>Ontwikkeling</b>	Prototype testing	Consistency inspection	Focus groups				
<b>Ontwikkeltest</b>	Heuristic evaluation						
<b>Systeemtests</b>	Heuristic evaluation	Perspective Based inspection	Interviews	Thinking aloud	observatie	Focus groups	
<b>Acceptatietests</b>	Heuristic evaluation	Perspective Based inspection	Interviews	Thinking aloud	Logging actual use	Focus groups	Reverse card sorting
<b>Gebruik en beheer</b>	A/B Testing	Card Sorting	Logging actual use	observatie	Interviews	Focus groups	Reverse card sorting

Zie ook Bijlage A voor een gedetailleerder overzicht van deze technieken

### Definiëren infrastructuur

Zie ook TMap NEXT® pag 211. Binnen een usability testtraject is speciale aandacht voor infrastructuur gewenst. Wanneer usability veel budget heeft, kunnen uitgebreide omgevingen en tools ingezet worden om het effect van de testen te verhogen. Maar infrastructuur wordt pas echt belangrijk wanneer het budget beperkt is. Dan zal naar slimme oplossingen gezocht moeten worden om de testen te ondersteunen.

Voor een prototype test kan bijvoorbeeld ook gebruik gemaakt worden van een paper prototype, in plaats van een werkend prototype. Bij de keuze voor tools kan gekozen worden voor freeware of donationware oplossingen. In plaats van een uitgebreid laboratorium, kan ervoor gekozen worden een kantoor in te richten als laboratorium, met webcams en microfoons, en eventueel remote desktop om met de gebruiker mee te kunnen kijken.

Zie voor een uitgebreidere verhandeling van alle infrastructurele mogelijkheden hoofdstuk 7.

### 3.3.2 Iteraties

Elk usability traject bestaat uit meerdere iteraties. Elke usability test of toets is een subproces op zich. Zoals eerder gezegd bestaat elke iteratie uit de fasen planning, voorbereiding, specificatie, uitvoering en afronding.

In de fase planning wordt voor de huidige iteratie bepaald welke activiteiten in welke volgorde uitgevoerd zullen worden. Belangrijk is om daarbij aan te sluiten op de actuele planning van de iteratie of fase waar het project zich in bevindt.

In de fase voorbereiding worden alle stappen uitgevoerd die randvoorwaardelijk zijn voor de test. Denk hierbij aan het reviewen danwel uitwerken van de testbasis, het specificeren en regelen van infrastructurele zaken en eventuele organisatorische werkzaamheden, zoals het uitnodigen van deelnemers.

In de fase specificatie vinden de activiteiten plaats die leiden tot de documentatie aan de hand waarvan de test of toets uitgevoerd kan worden. Denk hierbij aan het uitwerken van scenario's, beschrijvingen, instructies e.d.

In de fase uitvoering vindt de daadwerkelijke test of toets plaats. De activiteiten kunnen gedaan worden door usability experts, of de echte eindgebruikers, al naar gelang de techniek. Belangrijk is dat in alle gevallen de bevindingen centraal gelogd worden door de usability specialist.

In de fase afronding worden alle activiteiten gedaan, die moeten leiden tot een succesvol vervolg. Denk hierbij aan het debriefen van de deelnemers, maar ook aan het analyseren van de resultaten en het rapporteren van de bevindingen.

Een uitgebreide beschrijving van de exacte stappen die voor elke techniek in elke fase doorlopen moeten worden is te vinden in hoofdstuk 4, 5 en 6.

## 4 TESTTECHNIEKEN

Testtechnieken zijn technieken die kunnen worden toegepast bij het evalueren van eindproducten van de software. Deze technieken worden in onderstaande paragrafen volgens een vast stramien besproken.

### 4.1 Heuristic Evaluation

Heuristic evaluation is een informele individuele techniek waarbij een aantal evaluatoren, vaak usability experts, een interface toetsen aan vooraf vastgestelde usability principes. Deze principes worden heuristics genoemd.

#### 4.1.1 Stappenplan

Dit stappenplan kan een houvast bieden aan een testcoördinator die een heuristic evaluation moet organiseren. Alleen bij de uitvoeringsfase wordt het oogpunt van de evaluator gekozen.

##### *Vorbereiding*

1. Bepaal het aantal evaluatoren. Standaard wordt een aantal van 3 tot 5 evaluatoren aangeraden, maar dit hangt sterk af van de grootte en de complexiteit van het testobject.
2. Bepaal of gebruik gemaakt wordt van een observator. Voordeel hiervan is dat de evaluatoren ontlast worden, en dat niet achteraf nog eens alle bevindingen samengevoegd hoeven te worden.
3. Bepaal het testobject. In een iteratief ontwikkelproces zou dit al een paper prototype van de interface kunnen zijn.

##### *Specificatie*

4. Kies een basistechniek. Meestal wordt error guessing gebruikt, zodat evaluatoren zelf kunnen bepalen hoe zij door de interface navigeren. Wanneer een systeem erg domeinspecifiek is en de evaluatoren onvoldoende domeinkennis hebben, is het aan te raden scenario's te specificeren. De aan te raden technieken om dergelijke scenario's te ontwikkelen zijn de use-case techniek en de procescyclus test, omdat deze technieken in de regel representatieve scenario's opleveren. Als testbasis kunnen use-cases, procesbeschrijvingen of functioneel ontwerpen dienen.
5. Specificeer indien nodig de testscripts of scenario's.

##### *Uitvoering*

Alle evaluatoren voeren de uitvoering onafhankelijk van elkaar uit. Zij mogen geen ervaringen uitwisselen voordat de test afgelopen is.

6. Verken eerst de interface op een hoog niveau, zodat een duidelijk beeld ontstaat van de interactie en de scope van het systeem.
7. Ga nog een keer door de interface, maar nu met de focus op specifieke onderdelen van het systeem. Denk aan bepaalde functies, processen of deelsystemen.
8. Wanneer een bevinding optreedt, wordt deze direct genoteerd, aangevuld met een toelichting en een ernstcategorie. De observator kan indien nodig hulp bieden om verder te komen, maar niet voordat het probleem is benoemd door de evaluator. Alle bevindingen worden apart genoteerd, ook al treden ze op binnen één dialoog element.

##### *Afronding*

9. Verzamel de bevindingen van alle evaluatoren en distribueer deze naar de ontwerpers van de interface volgens de geldende bevindingenprocedure.

10. Optioneel: organiseer een debriefing met alle evaluatoren, de observator en de ontwerpers van de interface, waarin de deelnemers kunnen brainstormen over mogelijke herontwerpen. Ook kan stil gestaan worden bij de sterke punten van de interface.

#### **4.1.2 Heuristieken**

Usability experts hebben doorgaans een eigen specifieke verzameling van heuristieken. Een veel gebruikte verzameling is de onderstaande, ontwikkeld door Jakob Nielsen, de bedenker van deze techniek.

##### ***Helderheid over de status van het systeem***

Het systeem moet de gebruikers altijd informeren over de status van het systeem.

##### ***Laat het systeem overeenkomen met de echte wereld***

Het systeem moet dezelfde taal spreken als de gebruikers, ook het taalgebruik moet conformeren met die van de gebruikers. Het systeem moet taal gebruiken die past bij de materie waar het systeem voor bedoeld is. De informatie moet natuurlijk in een logische volgorde op het scherm worden getoond.

##### ***User control en vrijheid***

Gebruikers gebruiken vaak per ongeluk bepaalde systeemfuncties en hebben daarom een duidelijke "nooduitgang" nodig om de ongewenste status waarin het systeem zit, ongedaan te maken zonder uitgebreide dialogen. Ongedaan maken (undo) en opnieuw doen (redo) zijn functies die ondersteund moeten worden.

##### ***Consistentie en standaarden***

Gebruikers zouden zich niet moeten afvragen of verschillende woorden, situaties of acties dezelfde betekenis hebben. Gebruik standaarden.

##### ***Fouten voorkomen***

Een juiste foutmelding is goed, maar een goed ontwerp dat voorkomt dat de fouten gemaakt worden is beter. Probeer fouten die vaak worden gemaakt te voorkomen of onderken ze en laat de gebruikers hun keuze op dit punt in het systeem bevestigen.

##### ***Herkennen in plaats van herinneren***

De gebruiker moet geen informatie hoeven te onthouden die eerder is getoond. Instructies voor het gebruik van het systeem moeten zichtbaar zijn wanneer van toepassing.

##### ***Flexibiliteit en efficiency van gebruik***

Sneltoetsen – niet zichtbaar voor de beginnende gebruiker – kunnen de interactie tussen het systeem en de ervaren gebruiker versnellen zodat het systeem toepasbaar is voor zowel ervaren als onervaren gebruikers. Sta toe dat de gebruiker snel de frequentere acties kan uitvoeren.

##### ***Esthetisch en duidelijk ontwerp***

Dialogen moeten geen informatie bevatten die irrelevant of zelden nodig is. Extra informatie zorgt ervoor dat de relevante informatie minder benadrukt wordt.

##### ***Help de gebruikers met het herkennen, vaststellen, en herstellen van fouten***

Foutmeldingen moeten getoond worden in simpele taal (niet in code), waarin duidelijk wordt aangegeven wat het probleem is en waarin een constructieve oplossing wordt voorgesteld.

##### ***Hulp en documentatie***

Het is beter dat het systeem gebruikt kan worden zonder te moeten zoeken in documentatie desondanks kan het noodzakelijk zijn om hulp en documentatie beschikbaar te stellen. Deze informatie moet makkelijk te vinden zijn en gericht zijn op de taak van de gebruiker. Een lijst met gebruikersinstructies voor een bepaalde taak mag niet teveel stappen bevatten.

### **4.1.3 Varianten**

Een bekende variant op deze techniek is de heuristic estimation. Hierbij proberen de evaluatoren op basis van richtlijnen de usability van twee of meer ontwerpen kwantitatief te schatten. Hiervoor kunnen eenheden gebruikt worden zoals doorlooptijd, performance, aantal fouten etc. Op deze manier kan een onderbouwde, afgewogen keuze gemaakt worden welk ontwerp gebruiksvriendelijker zou moeten zijn.

### **4.1.4 Situatie**

De heuristic evaluation is ontwikkeld voor gebruik tijdens een iteratief ontwikkelproces. Omdat deze techniek ook gebruikt kan worden met dialogontwerpen of paper prototypes kunnen de evaluaties al starten in de ontwerpfase van een project. De techniek is relatief goedkoop en daarom ook goed in te zetten bij kleinere projecten of in kleinere organisaties. De techniek kent eigenlijk maar één harde randvoorwaarde: de evaluatoren moeten voldoende begrip hebben van usability principes.

### **4.1.5 Resultaat**

Het succes van de heuristic evaluation ligt in de kennis en betrokkenheid van de usability experts. Indien deze voldoende is, kan deze techniek in een vroeg stadium van het V-model tegen lage kosten aanzienlijke verbeteringen in de usability van een interface teweeg brengen.

Een risico van het gebruik van deze techniek kan optreden omdat usability experts vaak buiten de gebruikersafdeling of zelfs buiten de organisatie gevonden moeten worden. Voor hen kan het lastig zijn om rekening te houden met specifieke eigenschappen van het project of de organisatie. Hierdoor kunnen usability problemen gemist of te zwaar aangezet worden. Het is dan ook aan te raden naast deze techniek zeker ook usability testen met eindgebruikers te blijven doen.

## 4.2 Perspective-Based Inspection

Perspective-Based inspection is een informele individuele techniek, waar één of meer evaluatoren een ontwerp beoordelen door vanuit verschillende invalshoeken verschillende scenario's uit te voeren.

### 4.2.1 Stappenplan

Dit stappenplan kan een houvast bieden aan een testcoördinator die een perspective-based inspection moet uitvoeren.

#### *Vorbereitung*

1. Bepaal de perspectieven die gebruikt kunnen worden,
2. Bepaal per perspectief de inspectiedoelen waarop het testobject geëvalueerd moet worden.
3. Bepaal per perspectief de procedure die uitgevoerd moet worden, zodat de evaluator een houvast heeft aan een nieuw perspectief.
4. Nodig indien van toepassing extra evaluatoren uit.

#### *Specificatie*

5. Kies een basistechniek om de testgevallen uit te werken. Aan te raden technieken zijn de use-case techniek en de procescyclus test, omdat deze technieken in de regel representatieve scenario's opleveren. Als testbasis kunnen use-cases, procesbeschrijvingen of functioneel ontwerpen dienen.
6. Specificeer de testscripts.
7. Bepaal voor elk perspectief welke scenario's uitgevoerd moeten worden.

#### *Uitvoering*

8. Voer voor het eerste perspectief en het eerste scenario de procedure uit. Let hierbij op de inspectiedoelen.
9. Stel bij elke stap in de procedure vragen en schrijf deze op. Gaandeweg zullen deze vragen leiden tot een vaste verzameling vragen waar bij elke stap een antwoord op gegeven moeten worden.
10. Maak een bevinding aan wanneer de vragen niet naar tevredenheid beantwoord kunnen worden.
11. herhaal stap 8 t/m 10 voor elk scenario en elk perspectief.

#### *Afronding*

12. Verzamel de resultaten en analyseer deze.
13. Maak bevindingen aan volgens de geldende procedure.
14. Rapporteer de resultaten aan de opdrachtgever.

### 4.2.2 Perspectieven

Zhang et al. Beschreven een concrete aanpak voor perspective-based inspection. Zij gebruiken modellen uit de mens-computer interactie om te komen tot de volgende perspectieven, inspectiedoelen en procedures:

#### *Beginner*

De gebruiker kan zijn kennis en ervaring niet aanwenden om zijn doel te bereiken.

Inspectiedoelen:

1. fundamentele taken kunnen uitgevoerd worden met minimale kennis van het systeem.

Procedure:

1. bepaal het effect dat in de interface bereikt moet worden om het doel te halen.
2. identificeer de acties die nodig zijn om het effect te bereiken.
3. voer de acties uit.
4. observeer de feedback die de interface geeft.

5. probeer de gemaakte voortgang te begrijpen.

### **Expert**

De gebruiker weet hoe het systeem te gebruiken, en geeft de voorkeur aan efficiënt werken en het bereiken van complexere doelen.

Inspectiedoelen:

1. Gebruikers kunnen elke taak efficiënt en eenvoudig afronden.
2. Gebruikers kunnen het systeem customizen aan hun behoeften.
3. Gebruikers kunnen productiever worden door complexere features te gebruiken.

Procedure:

1. Scan de instructies en de interface.
2. Voer de acties zo efficiënt mogelijk uit.
3. Wacht indien nodig op respons.

### **Foutafhandeling**

De gebruiker heeft een probleem met het resultaat van een handeling en probeert dit op te lossen.

Inspectiedoelen:

1. De kans dat een gebruiker een fout maakt is minimaal.
2. De gebruikersinterface helpt de gebruiker een fout te begrijpen.
3. De gebruikersinterface helpt de gebruiker een fout te herstellen.
4. Systeemfouten worden adequaat afgehandeld.

Procedure:

1. Overweeg alle mogelijke fouten die de gebruiker kan maken. Denk hierbij aan omissies, tikfouten, perceptiefouten, fouten door uitproberen en fouten door een verkeerde systeemmodus.
2. Probeer de fouten te voorkomen.
3. Probeer te realiseren dat er een fout gemaakt is.
4. probeer het effect van de fout te minimaliseren.
5. Probeer de fout met zo min mogelijk impact te herstellen.

## **4.2.3 Varianten**

Bij een persona-based inspection neemt elke evaluator het perspectief aan van een persona. Het gekozen persona bepaalt de inspectiedoelen van de evaluator, de procedure die doorgelopen moet worden en de scenario's die uitgevoerd kunnen worden. Voor het ontwikkelen van persona's zie hoofdstuk 6.

## **4.2.4 Situatie**

Deze techniek kan al vroeg in het V-model ingezet worden, maar is het meest effectief wanneer de interface vrij definitief is omdat de techniek vrij gedetailleerd is. Het geniet de voorkeur om voorafgaand de belangrijkste issues op te sporen met een eenvoudiger techniek. Een ervaren usability expert kan deze techniek vrij snel tegen lage kosten uitvoeren. Wanneer geen expert voorhanden is, is deze techniek waarschijnlijk te complex en te tijdrovend om uit te voeren. De techniek kan door één evaluator uitgevoerd worden, maar ook een opzet met meerdere evaluatoren die ieder een eigen perspectief kiezen, of meerdere evaluatoren die hetzelfde perspectief kiezen, is mogelijk. Deze hoeven niet allemaal usability experts te zijn, zolang er maar één is om de procedures uit te werken.

## **4.2.5 Resultaat**

Deze techniek kan al vroeg in de ontwikkeling op detailniveau usability bevindingen aan het licht brengen. De sterke kant van deze techniek is dat met één evaluator vaak veel meer gedetailleerde bevindingen gedaan kunnen worden dan bij andere individuele technieken. Het is een ideale techniek, wanneer gebruikers niet beschikbaar zijn voor testen. Nadeel is dat deze inspecties in de praktijk erg veel tijd kosten, en dat deze niet zonder een usability expert gedaan kunnen worden.



## 4.3 Interviews

Het interview is niet zozeer een testtechniek, maar een aanvulling op een usability test waarmee extra testresultaten verkregen kunnen worden. Deze techniek is zeer belangrijk voor usability testen om twee redenen: het is een zeer goedkope techniek, en het is tot nu toe de enige techniek waarmee bruikbare gegevens over pleasurability verkregen kunnen worden.

### 4.3.1 Stappenplan

Dit stappenplan biedt een houvast aan een testcoördinator die een interview moet voorbereiden en uitvoeren.

#### *Vorbereitung*

1. Bepaal het doel van het interview. Neem hierin op welke gegevens verkregen moeten worden, en of het om kwalitatieve of kwantitatieve gegevens gaat.
2. Bepaal aan de hand van het doel wat voor type interview gehanteerd gaat worden. De verschillende typen zijn beschreven in de volgende paragraaf.
3. Bepaal het moment waarop het interview afgenomen wordt. Dit is meestal na afloop van een usability test. Ook kan vooraf een interview afgenomen worden, waarbij de gebruiker gevraagd wordt naar zijn verwachtingen.

#### *Specificatie*

4. Bepaal de structuur van het interview. Zorg ervoor dat deze voor de gebruiker logisch in elkaar zit.
5. Stel de vragen op. Let hierbij op de openheid van de vragen. Zorg ervoor dat de vragen begrijpelijk zijn voor de doelgroep, ondubbelzinnig zijn en enkelvoudig opgesteld zijn. Vermijd suggestieve vragen!

#### *Uitvoering*

6. Laat de gebruiker kennis maken met het systeem, bijvoorbeeld door een van de testtechnieken met hem uit te voeren.
7. Neem het interview af. Wanneer het geen mondeling interview betreft, leg de vragen dan tijdig, liefst zo snel mogelijk, voor aan de tester.
8. Begin bij een mondeling interview altijd met een uitleg van het doel van het interview, de verwachte duur en een kort overzicht van de structuur.
9. Stel de vragen een voor een, en pas hierbij LSD toe: Luisteren, Samenvatten en Doorvragen.
10. Bedank na afloop de tester.

#### *Afronding*

11. Analyseer de gegevens. Verzamel bij open vragen eventuele genoemde bevindingen, en illustratieve citaten. Verzamel bij gesloten vragen de antwoorden over alle interviews, en probeer hieruit waar mogelijk conclusies te trekken.
12. Stel een rapport op van de resultaten.

### 4.3.2 Soorten interviews

Interviews kunnen op zeer uiteenlopende wijzen uitgevoerd worden. In de regel verschillen interviews van elkaar op 3 dimensies: medium, structuur en openheid van de vragen.

#### *Medium*

Interviews kunnen op verschillende mediums uitgevoerd worden. Wanneer over een interview gesproken wordt, wordt doorgaans een mondeling interview bedoeld. Ook telefonisch of via een chatprogramma is hierbij mogelijk.

Andere media vereisen vaak een zeer gestructureerde vorm, zoals schriftelijk, email, internet en voice-response systemen.

### *Structuur*

Interviews kunnen nogal verschillen in de gestructureerdheid. Volledig ongestructureerde interviews zijn mogelijk, maar vaak niet goed bruikbaar voor gebruik bij usability testen. Bovendien vergt dat in de uitvoering veel ervaring van de interviewer. Tenslotte is de analyse van dergelijke interviews nogal problematisch.

Volledig gestructureerde interviews worden over het algemeen enquêtes genoemd. Hierbij zijn alle vragen voorbereid, en is nagenoeg geen ruimte voor improvisatie. Deze vorm kan noodzakelijk zijn voor het gekozen medium, of wanneer van tevoren exact bekend is welke gegevens verzameld moeten worden. Ook is het hiermee makkelijker kwantitatieve gegevens te verkrijgen, bijvoorbeeld cijfermatige beoordelingen van het testobject. De analyse van de resultaten is vaak relatief eenvoudig.

In de praktijk wordt vaak een tussenvorm gebruikt. Een aantal vragen met ieder een aantal subvragen zijn voorbereid, waarbij ruimte is voor improvisatie, mocht een antwoord verduidelijking behoeven, of interessant genoeg zijn om op door te vragen. Dit geeft de interviewer de flexibiliteit om voor elke tester te bepalen welke onderwerpen de meeste aandacht krijgen.

### *4.3.3 Openheid vragen*

Als laatste kunnen interviews sterk verschillen in de openheid van de vragen. Open vragen bevatten een vragend voornaamwoord (wie, wat waar, wanneer, hoe, ...), waarbij een relatief uitgebreid antwoord wordt verwacht. Dergelijke vragen zijn goed om kwalitatieve gegevens te verkrijgen, zoals verklaring van gedragingen van gebruikers, of subjectieve oordelen over het testobject. Het is alleen lastig analyses uit te voeren over meerdere resultaten.

Gesloten vragen kennen slechts een bepaald aantal korte antwoorden, denk aan de ja/nee vraag, de meerkeuze vraag of de schaalvraag. Deze vragen worden vaak gebruikt bij enquêtes. Hiermee kunnen kwantitatieve gegevens verzameld worden, bijvoorbeeld een subjectief cijfermatig oordeel over het testobject. Het cijfermatige karakter maakt het makkelijker analyses uit te voeren over meerdere resultaten. Nadeel bij deze vorm is dat de resultaten niet altijd juist worden geïnterpreteerd, omdat de antwoorden beïnvloed kunnen zijn. Een bekend voorbeeld hiervan is dat mensen vaak extreme waarden vermijden bij schaalvragen.

### *4.3.4 Situatie*

Op elk moment in een project, kunnen gebruikers geïnterviewd worden. Het moment hangt vaak af van het doel van het interview. Een veelvoorkomende toepassing is na afloop van een usability test. Een combinatie van een mondeling interview met een schriftelijke enquête wordt vaak gehanteerd. Op die manier kunnen objectieve resultaten aangevuld worden met subjectieve meningen over het testobject. Interviews kunnen in elke organisatie uitgevoerd worden, de kosten zijn laag en de benodigdheden vallen binnen het gangbare kantoorassortiment.

Het is echter wel belangrijk dat de interviewer weet wat hij doet. Kleine fouten in de vraagstelling kunnen de verkregen resultaten enorm beïnvloeden. Wanneer gevraagd wordt naar kwantitatieve gegevens kan dit een verkeerd beeld geven. Bij onervarenheid met statistiek en onderzoeken, is het raadzaam een interview alleen te gebruiken voor het verzamelen van kwalitatieve gegevens.

### *4.3.5 Resultaat*

Interviews zijn doorgaans een zeer nuttige aanvulling op usability testen tegen zeer lage kosten. Dit is tot nu toe de enige techniek waarmee testers kunnen laten weten welke emoties het testobject bij hun heeft opgeroepen. Zo kan een tester tijdens een interview een indruk geven van het gebruiksplezier dat hij heeft ervaren. Alternatieven om pleasurable te bepalen, zoals observatie, gezichtsherkenning en fysiologie, leveren resultaten

die (nog) niet bruikbaar genoeg zijn. Ook kan de tester uitleg geven over de motivatie achter zijn handelingen. Dit kan nuttige inzichten opleveren voor de analyse van andere testresultaten.

Op zichzelf is de techniek wel te gebruiken, alleen zijn de resultaten vaak onbetrouwbaar, wanneer deze niet vergeleken kunnen worden met de daadwerkelijke handelingen van de gebruiker. Wat de gebruiker zegt, is immers niet hetzelfde als wat de gebruiker doet. Vul deze techniek dan ook altijd aan met objectieve testresultaten.

## 4.4 Thinking aloud

Thinking aloud is een informele individuele techniek, waar een tester hardop nadenkt terwijl hij het testobject gebruikt. Dat klinkt erg simpel, maar het is voor een tester niet vanzelfsprekend dat hij continu hardop vertelt welke afwegingen hij maakt en hoe hij zijn ervaringen beoordeelt.

### 4.4.1 Stappenplan

Dit stappenplan biedt houvast aan een testcoördinator die een thinking aloud test moet organiseren.

#### *Vorbereiding*

1. Bepaal het aantal testers. Volgens experts wordt met 3 testers gemiddeld de beste kosten/baten ratio bereikt. Voor een systeem met een brede doelgroep zouden als het budget het toelaat beter minimaal 5 testers gebruikt kunnen worden.
2. Bepaal de methode om de ervaringen van de testers vast te leggen en zorg indien nodig voor apparatuur. In de praktijk blijkt de methode waarbij gebruik wordt gemaakt van een headset microfoon het meest effectief. Een goedkoper alternatief is om een observator mee te laten schrijven.
3. Rekruteer de testers uit de gebruikersgroep. Deze techniek leent zich in principe alleen voor testen met eindgebruikers.

#### *Specificatie*

4. Kies een basistechniek voor de testgevallen. Aan te raden technieken om dergelijke scenario's te ontwikkelen zijn de use-case techniek en de procescyclus test, omdat deze technieken in de regel representatieve scenario's opleveren. Als testbasis kunnen use-cases, procesbeschrijvingen of functioneel ontwerpen dienen. Het is ook mogelijk geen testgevallen te specificeren en de gebruiker de vrije hand te geven.
5. Specificeer indien nodig de testscripts.

#### *Uitvoering*

6. Begin altijd met een briefing aan de tester. Leg uit wat het doel van de test is, en vertel hem hoe de test zal verlopen. Vertel hem welke taken hij moet uitvoeren, wanneer hij klaar is en hoe de afronding plaats zal vinden. Vertel hem ook dat hij hardop moet uitspreken wat hij denkt tijdens het uitvoeren van de test, en dat hij opgenomen wordt.
7. Laat de tester een oefening doen met thinking aloud. Een veelgebruikte oefening is om de tester een potlood en een stuk papier te geven. De tester wordt gevraagd wat hij vindt van het potlood, wat hij er mee wil gaan doen, hoe hij dit wil gaan doen, of hij geslepen moet worden of niet, vervolgens kan hij beginnen met schrijven en zeggen wat hem opvalt.
8. Wanneer de tester thinking aloud onder de knie heeft, kan hij beginnen met het een voor een uitvoeren van de testgevallen.
9. Neem de uitspraken van de tester op, of maak aantekeningen. Grijp in wanneer de gebruiker stilvalt, vraag hem wat hij denkt en waarom hij zwijgt. Wees terughoudend met aanwijzingen.

#### *Afronding*

10. Doe altijd een debriefing met de gebruiker. Dit is te combineren met een interview. Laat de gebruiker in ieder geval weten hoe hij het gedaan heeft en wat er met de resultaten gaat gebeuren.
11. Werk de opnames, of de aantekeningen uit. Identificeer problemen en log deze als bevinding.
12. Stel een rapport samen met de belangrijkste bevindingen en conclusies en stuur dit naar de opdrachtgever.

#### **4.4.2 *Situatie***

Thinking aloud kan op zeer kleine projecten ingezet worden als zelfstandige techniek, maar ook op grotere projecten als onderdeel van een gecontroleerde observatie. Deze techniek is alleen geschikt om kwalitatieve gegevens mee te verkrijgen. Vaak wordt een thinking aloud test na de FAT uitgevoerd, zodat de gebruiker geen technische of functionele fouten te zien krijgt. Het is echter ook mogelijk thinking aloud uit te voeren op (paper) prototypes.

#### **4.4.3 *Resultaat***

Thinking aloud is bij uitstek een techniek om tegen lage kosten de belangrijkste usability bevindingen te vinden. Ook is het een makkelijke manier om gebruikers in een vroeg stadium te betrekken bij een iteratief ontwikkelproces. Een nadeel is dat veel afhangt van de capaciteit van testers om hun gedachten te verwoorden. Gebruik deze techniek daarom bij voorkeur in combinatie met een of meer andere usability testtechnieken.

## 4.5 Logging Actual Use

Logging actual use is een individuele informele techniek, waarmee automatisch verzamelde gegevens over het gebruik van het systeem geanalyseerd worden. Achteraf kan met deze gegevens het gedrag van één of meerdere gebruikers in kaart gebracht worden. Voor het loggen kan extra functionaliteit ingebouwd worden, standaard functionaliteit gebruikt worden of een speciaal pakket ingezet worden. Voor de beschrijving van pakketten verwijzen wij naar paragraaf 7.3 “Tools”.

### 4.5.1 Stappenplan

Dit stappenplan kan een houvast bieden aan een testcoördinator die een logging actual use test moet organiseren.

#### *Vorbereiding*

De gehele voorbereiding is optioneel. In de meeste gevallen kunnen achteraf logs uitgelezen en bekeken worden. Het verdient echter wel de aanbeveling vooraf na te denken over te loggen gegevens en de noodzaak om hiervoor code te bouwen danwel een pakket te implementeren.

1. Bepaal het aantal testers. Met 3 testers wordt gemiddeld de beste kosten/baten ratio bereikt. Voor een systeem met een brede doelgroep zouden als het budget het toelaat beter minimaal 5 testers gebruikt kunnen worden.
2. Bepaal de te loggen gegevens. Denk hierbij aan toetsaanslagen, muisklikken, klikratio, foutmeldingen, waarschuwingen, performance, wachttijd, functiegebruik, helpgebruik,
3. Bepaal de noodzaak voor extra programmatuur. Veel systemen kennen van zichzelf al logging van belangrijke events zoals fouten en waarschuwingen, maar voor bijvoorbeeld toetsaanslagen, muisklikken en performance zijn vaak tools of programmatuur nodig.
4. Rekruteer de testers uit de gebruikersgroep. Deze techniek leent zich in principe alleen voor testen met eindgebruikers.

#### *Specificatie*

5. Kies een basistechniek voor de testgevallen. Aan te raden technieken om dergelijke scenario's te ontwikkelen zijn de use-case techniek en de procescyclus test, omdat deze technieken in de regel representatieve scenario's opleveren. Als testbasis kunnen use-cases, procesbeschrijvingen of functioneel ontwerpen dienen. Het is ook mogelijk geen testgevallen te specificeren en de gebruiker de vrije hand te geven.
6. Specificeer indien nodig de testscripts.

#### *Uitvoering*

7. Begin altijd met een briefing aan de tester. Leg uit wat het doel van de test is, en vertel hem hoe de test zal verlopen. Vertel hem welke taken hij moet uitvoeren, wanneer hij klaar is en hoe de afronding plaats zal vinden.
8. Zet indien nodig de logging aan en start de test.

## *Afronding*

9. Doe altijd een debriefing met de gebruiker. Dit is te combineren met een interview. Laat de gebruiker in ieder geval weten hoe hij het gedaan heeft en wat er met de resultaten gaat gebeuren.
10. Verzamel de loggegevens en analyseer deze. Wees terughoudend met conclusies: beoordeel zelf, eventueel in samenspraak met de tester, of opvallende gegevens daadwerkelijk een bevinding betekenen. Bijvoorbeeld:
  - een lange wachttijd kan betekenen dat de gebruiker niet weet wat hij moet doen. Maar de gebruiker kan ook geïnteresseerd een stuk tekst aan het lezen zijn, of even koffie aan het halen zijn.
  - Het niet gebruiken van een functie kan betekenen dat de gebruiker deze functie niet nodig heeft. Het kan ook betekenen dat hij niet weet wat de functie doet, de functie niet kan vinden of de functie niet herkent als zijnde een interactieobject.
  - Intensief gebruik van een helpfunctie kan betekenen dat de gebruiker niet zo ervaren is. Het kan ook betekenen dat de interface te complex is, dat het lang duurt voor de gebruiker de applicatie onder de knie heeft, of dat de helpfunctie niet duidelijk genoeg is.
11. Stel een rapport samen met de belangrijkste bevindingen en conclusies, en stuur dit naar de opdrachtgever.

### *4.5.2 Situatie*

Logging actual use is een van de weinige usability testtechnieken waarbij de gebruikers in principe vanaf hun werkplek de testen uit kunnen voeren. Dit schept mogelijkheden voor testen over afstand, zeker bij een beperkt budget. Daarnaast is de techniek goed te combineren met observatie en A/B testing. De techniek levert in principe alleen kwantitatieve gegevens op. Met deze gegevens kunnen in beperkte opzet alleen zeer voorzichtig conclusies getrokken worden. Vaak zullen de kwantitatieve gegevens een indicatie geven van plekken waar zich problemen voordoen. Deze techniek wordt dan ook vaak gebruikt om op het spoor van mogelijke usability problemen te komen. Wanneer grootschalige onderzoeken gedaan worden, bijvoorbeeld op drukbezochte websites, kan logging actual use zeer bruikbare gegevens opleveren. Deze techniek kan in elke fase, waarin een werkend testobject opgeleverd is uitgevoerd worden, en is dan ook bruikbaar in iteratieve ontwikkeltrajecten. Ook in de beheerfase wordt deze techniek toegepast om verbeterpunten te kunnen ontdekken.

### *4.5.3 Resultaat*

Zoals gezegd levert logging actual use alleen kwantitatieve gebruiksgegevens op. Dit heeft als voordeel dat de gegevens objectief zijn, daadwerkelijke over het gebruik gaan en verzameld kunnen worden zonder dat de gebruiker het merkt. Het nadeel is dat deze informatie niet weergeeft waarom gebruikers bepaalde handelingen doen. Daarvoor kan deze techniek aangevuld worden met interviews.

Deze techniek kan een goede indicatie geven van probleemgebieden binnen een applicatie. Alleen in groots opgezette testen kunnen harde conclusies getrokken worden op basis van de verzamelde gegevens. Wees ook dan terughoudend, en betrek bij voorkeur een professionele onderzoeker bij de analyse. De meeste resultaten worden gehaald bij grootschalig onderzoek naar websites: op dit gebied bevinden zich dan ook de meeste commerciële logging pakketten.

## 4.6 Observatie

Observatie is een informele individuele techniek, waar een tester geobserveerd wordt terwijl hij het testobject gebruikt. Deze techniek kan op hele verschillende manieren uitgevoerd worden, van eenvoudige observatie van eindgebruikers tot grootschalige gebruikersonderzoeken in professionele laboratoria.

### 4.6.1 Soorten observaties

Observaties kennen 2 dimensies waarop ze van elkaar verschillen:

#### *Naturalistisch vs. gecontroleerd*

Naturalistische observaties, ook wel veld studies, vinden plaats in een natuurlijke omgeving, vaak een kantoor of een huiskamer. Testers kunnen ook zelf observeren, door een log of dagboek bij te houden. Resultaten zijn kwalitatief.

Gecontroleerde observaties vinden plaats in een gecontroleerde ruimte, ook wel laboratorium, waarbij de tester vaak gestuurd wordt in zijn acties. Een dergelijke opstelling biedt veel mogelijkheden tot het verzamelen van kwantitatieve gegevens. Overigens kan een simpele kantooruimte ook vrij eenvoudig als lab ingericht worden.

#### *Passief vs. actief*

Bij passieve observatie concentreert men zich op wat de gebruiker doet. De observator bemoeit zich niet met de gebruiker. Uitleg van de taak of het beantwoorden van mogelijke vragen wordt zoveel mogelijk voor aanvang van de test gedaan zodat de observator zich gedurende de test afzijdig houdt.

Bij actieve observatie is de observator meer betrokken bij de gebruiker tijdens de test. Uitleg van de applicatie, en het stellen en beantwoorden van vragen zijn toegestaan. Deze methode is alleen geschikt om de mening van de gebruiker vast te stellen, conclusies over het gedrag zijn moeilijker te doen door de invloed van de observator.

### 4.6.2 Stappenplan

Dit stappenplan kan een houvast bieden aan een testcoördinator die een observatie moet organiseren en uitvoeren.

#### *Planning*

1. Als eerst moet de opzet van de observatie bepaald worden:
  - a. Wat willen we observeren: welk product, of welke deelproducten?
  - b. Wat is het doel van de observatie? Is deze alleen bedoeld om verbeterpunten te identificeren, of moet ook een oordeel gevormd worden over de gebruiksvriendelijkheid van het testobject. Het doel staat aan de basis van alle keuzen, die voor de uitvoering van de observatie gemaakt moeten worden.
  - c. Wat voor soort observatie wordt gedaan? Naturalistisch of gecontroleerd, passief of actief?
  - d. Waar moet op gelet worden? Correcte uitvoering, foutafhandeling, snelheid, gemoedstoestand, ... maak eventueel vooraf sets van mogelijke gebeurtenissen, zodat deze snel geregistreerd kunnen worden tijdens de test.
2. Tijdens de planning moeten ook een aantal beslissingen genomen worden ten aanzien van de inrichting van de observatie. Zorg hierbij altijd dat deze aansluiten op de keuzes die bij stap 1 zijn gemaakt.
  - a. Welk materiaal gaat gebruikt worden om waarnemingen vast te leggen? Denk aan pen en papier, audio recorders, camera's, ...
  - b. Waar gaat de test uitgevoerd worden? een professioneel laboratorium of een ingericht kantoor? Op een werkplek of bij de gebruiker thuis?



- c. Wie gaat de testen uitvoeren? Welke categorieën gebruikers kunnen onderscheiden worden? Gebruikers kunnen verschillende kenmerken hebben, zoals ervaring met IT, domeinkennis, handicaps, leeftijd, ethniciteit, ...
- d. Welke tooling wordt gebruikt ter ondersteuning van de test. Zie het hoofdstuk "Tools".

### *Vorbereiding/Infrastructuur*

3. Nadat in stap 1 en 2 alle belangrijke keuzes gemaakt zijn, moeten deze ook uitgevoerd gaan worden:
  - a. Het materiaal moet verzameld, en eventueel aangeschaft en uitgeprobeerd worden
  - b. De lokatie moet ingericht worden. Een laboratorium moet geïnstalleerd worden, of afgehuurd, een natuurlijke plek moet eventueel ook uitgezocht en ingericht worden. Het testobject moet bereikbaar zijn vanaf de lokatie.
  - c. Testers moeten gerekruteerd en uitgenodigd worden.
  - d. Tools moeten aangeschaft en geïnstalleerd worden. De observator zal eventueel ook opgeleid moeten worden in het gebruik van de tools.

### *Specificatie*

4. Kies een basistechniek voor de testgevallen en schrijf deze. Aan te raden technieken om dergelijke scenario's te ontwikkelen zijn de use-case techniek en de procescyclus test, omdat deze technieken in de regel representatieve scenario's opleveren. Als testbasis kunnen use-cases, procesbeschrijvingen of functioneel ontwerpen dienen. Bij naturalistische observatie verdient het de voorkeur geen testgevallen te specificeren, zodat het geobserveerde gedrag zo natuurlijk mogelijk is.

### *Uitvoering*

5. Begin altijd met een briefing aan de tester. Leg uit wat het doel van de test is, en vertel hem hoe de test zal verlopen. Vertel hem welke taken hij moet uitvoeren, wanneer hij klaar is en hoe de afronding plaats zal vinden. Laat hem ook weten op welke manieren hij geobserveerd wordt en op welke wijze de observator tewerk zal gaan. Geef daarbij ook aan in hoeverre de observator actief is: geeft de observator uitleg, is hij beschikbaar voor vragen of zal hij in het geheel niet ingrijpen.
6. Observeer de gebruiker bij het uitvoeren van zijn taken.
7. Wanneer iets bijzonders gebeurt, noteer dit direct. Gebruik waar mogelijk vooraf bepaalde annotaties. Maak hierbij, vooral wanneer opnames gemaakt worden, gebruik van een tijdslijn. Houdt wel altijd ruimte voor vrije opmerkingen.
8. Doe altijd een debriefing met de gebruiker. Dit is te combineren met een interview. Laat de gebruiker in ieder geval weten hoe hij het gedaan heeft en wat er met de resultaten gaat gebeuren.

### *Afronding*

9. Analyseer de data. Afhankelijk van de opzet en het doel van de test, kan dit relatief weinig tijd in beslag nemen, of meer dan de uitvoering van de test. Kwalitatieve bevindingen kunnen geprioriteerd en gelogd worden, kwantitatieve gegevens moeten zorgvuldig geanalyseerd worden om conclusies te kunnen trekken.
10. Stel een rapport samen met de belangrijkste bevindingen en conclusies, en stuur dit naar de opdrachtgever.

#### **4.6.3 Situatie**

Omdat er zoveel verschillende soorten observaties zijn, is deze techniek erg breed inzetbaar. De techniek is eigenlijk alleen niet optimaal in te zetten, wanneer de gebruikersgroep zeer homogeen is, en kwantitatieve gegevens gewenst zijn. In dat geval zou namelijk een erg groot aantal testers nodig zijn voor een representatief resultaat.

Observatie met eindgebruikers als testers is niet verstandig voordat een correct functionerend product is opgeleverd. Meestal wordt deze techniek daarom pas na de FAT ingezet. De techniek kent zeer goedkope varianten, waardoor budget zelden een issue hoeft te zijn. De goedkopere varianten kunnen ook gebruikt worden voor prototype testing.

#### **4.6.4 Resultaat**

Observatie is een techniek waarmee snel en goedkoop kwalitatieve resultaten geboekt kunnen worden. Zo kunnen snel algemene indrukken en concrete bevindingen gevonden worden. Daarnaast kan in een grotere opzet veel kwantitatieve informatie vergaard worden met een grote gebruikersgroep. In een dergelijke opzet is het mogelijk harde gegevens te vinden waarmee bijvoorbeeld vergelijkingen tussen verschillende ontwerpen of trendanalyses gemaakt kunnen worden.

Nadeel is dat deze techniek vaak pas tijdens de GAT uitgevoerd kan worden. Dit verhoogt de herstelkosten van gevonden bevindingen aanzienlijk. Observatie kan ook een goede manier zijn, om usability binnen een organisatie te verkopen. Met deze techniek kunnen bijvoorbeeld ontwerpers en ontwikkelaars, maar ook managers met eigen ogen zien wat het belang kan zijn van gebruiksvriendelijke producten.

Grootschalige observaties met kwantitatieve resultaten kunnen veel nuttige concrete resultaten opleveren. Het verdient echter de aanbeveling om in deze gevallen ervaren onderzoekers in te schakelen om de test op te zetten en de resultaten te analyseren. Kwantitatief onderzoek kent namelijk veel valkuilen. De testers kunnen niet representatief zijn voor de doelgroep, omgevingsfactoren kunnen een invloed gespeeld hebben (denk bijvoorbeeld aan storende geluiden), en correlaties kunnen niet significant blijken, of geen causaal verband hebben vanwege de invloed van een derde variabele.

## 4.7 A/B Testing

A/B testing is een usability testtechniek die uitsluitend gebruikt kan worden voor websites. Bij A/B testing worden twee varianten van één website online geplaatst. Variant A en B verschillen op maximaal één onderdeel van elkaar. Een deel van de bezoekers wordt naar variant A gestuurd, het andere deel naar variant B. Om dit te ondersteunen zijn diverse pakketten beschikbaar, zie hiervoor paragraaf 7.3 “Tools”.

### 4.7.1 KPI's

A/B testing levert altijd statistische gegevens over het gebruik van verschillende versies van websites op. Daarom is het van belang om van tevoren te bedenken hoe het succes van een website gemeten kan worden. Hiervoor worden vaak KPI's gebruikt die ook bij marketing gebruikt worden. Een bij A/B testing veelgebruikte KPI is conversieratio: de verhouding tussen het aantal bezoekers van de website, die een bepaald doel bereiken, en het totaal aantal bezoekers van de website. Ook kan gebruik gemaakt worden van de gemiddelde transactiespijs, de duur van het bezoek of loyaliteit.

### 4.7.2 Stappenplan

#### *Vorbereiding*

1. Bepaal het doel van de website
2. Breng middels een andere usability testtechniek boven water wat de belangrijkste usability problemen van de website zijn.
3. Kies een te wijzigen sectie. Geef voorkeur aan secties waar relatief veel usability problemen in zitten, en die zo veel mogelijk bijdragen aan de doelstelling van de website.
4. Bepaal de doelstelling van de test. Let hierbij op het doel van de site en de usability problemen die spelen. Maak de doelstelling SMART, zodat harde conclusies getrokken kunnen worden uit de test.

#### *Specificatie*

5. Kies de te wijzigen eigenschappen. Geef voorkeur aan eigenschappen die de grootste usability problemen bevatten. Waak er bij bestaande websites voor niet te veel eigenschappen te kiezen, als de site afhankelijk is van trouwe bezoekers. Deze kunnen negatief reageren als de hun vertrouwde website onverwachts op veel punten is veranderd.
6. Maak varianten voor de gekozen eigenschappen. Denk aan variërende locaties voor buttons of menu's, verschillende kleuren of alternatieve navigatiestructuren.
7. Zorg voor goede tooling om de test te ondersteunen en de resultaten te loggen. Hierbij moet de tool zodanig ingesteld worden dat de doelstelling gemeten kan worden.

#### *Uitvoering*

8. Zet de gekozen varianten online, en start de tooling. Bij A/B testing staan altijd twee varianten online: er kan ervoor gekozen worden twee nieuwe varianten met elkaar te vergelijken, of om een nieuwe variant tegen de bestaande website af te zetten.
9. Laat de tool de resultaten van de test meten totdat een betrouwbaar resultaat bereikt is. Hierbij wordt minimaal een periode van een week aangehouden, zodat zoveel mogelijk soorten bezoekers deelnemen aan de test. De periode hangt ook af van het aantal bezoekers van de site. Let er ook op dat de verhouding bezoekers tussen de varianten gelijk blijft. Houdt ook rekening met eventuele seizoenseffecten; met name bij e-business sites kunnen verschillende maanden hele verschillende resultaten opleveren. Voor webshops is eind december bijvoorbeeld altijd een relatief drukke periode.

### **Afronding**

10. Controleer of de doelstelling, die in de voorbereidingsfase is bepaald, met één van de varianten gehaald is.
11. Als een van de varianten de doelstelling haalt, adviseer dan deze variant te implementeren
12. Als beide varianten de doelstelling halen, en ongeveer gelijk presteren, kan ervoor gekozen worden langer door te testen. Ook kan simpelweg een van beide varianten geïmplementeerd worden.
13. Als geen van de varianten de doelstelling halen, controleer dan of de doelstelling niet te hoog lag. Als een variant duidelijk betere resultaten oplevert, adviseer dan deze toch te implementeren. Zoniet, evalueer dan grondig het testproces om uit te zoeken waarom de verwachte verbeteringen niet gehaald zijn. Begin dan opnieuw met de specificatie, of zelfs met de voorbereiding.
14. Sla alle verzamelde gegevens zorgvuldig op, zodat deze achteraf nog verder geanalyseerd kunnen worden, mocht daar aanleiding toe zijn.

### **4.7.3 Multivariate testing**

Een usability testtechniek die sterk op A/B testing lijkt is multivariate testing. Bij multivariate testing worden meer dan twee varianten per keer worden getest. Ook worden bij multivariate testing meerdere secties per testtraject gewijzigd.

Zo kan in één testtraject bijvoorbeeld:

- de positie van de zoekfunctie gewijzigd worden (rechtsboven, linksboven)
- de achtergrondkleur van het menu gevarieerd worden (rood, blauw, geel)
- de lettergrootte van tekstkoppen aangepast worden (14, 18)

Op deze manier ontstaan er  $2$  (positie zoekfunctie)  $\times$   $3$  (kleur menu)  $\times$   $2$  (lettergrootte) =  $12$  varianten.

### **4.7.4 Situatie**

A/B testing is een techniek die alleen in vrij specifieke situaties toegepast kan worden. Er moet een bestaande website zijn, die dagelijks minimaal 50 bezoekers trekt, om zinvolle conclusies te kunnen trekken uit de resultaten. Daarnaast moet ook gekeken worden naar de verwachte baten; als die er niet zijn, heeft het weinig zin een A/B test uit te voeren.

In dergelijke situaties is A/B testing een uitermate geschikte methode om harde statistieken te verzamelen over het gedrag van grote groepen gebruikers. Het is een goede methode om in een iteratief verbetertraject wijzigingen uit te proberen, voor ze definitief door te voeren. Vaak wordt de testtijd benut om een volgende A/B test met nieuwe varianten voor te bereiden.

### **4.7.5 Resultaat**

A/B testing levert harde statistieken over het gedrag van de gebruikers, en de effectiviteit van verschillende varianten van één website. Na een A/B test kan een gewogen beslissing gemaakt worden over wijzigingen aan de website. Ook kan een A/B test aanleiding zijn, om wijzigingen die bijzonder succesvol blijken ook door te voeren op andere secties van een site, of zelfs op andere websites.

A/B testing is ook een manier om de klant beter te leren kennen. Uit de resultaten kan opgemaakt worden wat hem wel bevalt en wat niet.

## 5 TOETSTECHNIEKEN

Toetstechnieken zijn technieken die kunnen worden toegepast bij het evalueren van tussenproducten, zoals ontwerpen of mockups van de software. Deze technieken worden in onderstaande paragrafen volgens een vast stramien besproken.

### 5.1 Consistency Inspection

Consistency inspection is een informele groep techniek. Ontwerpers beoordelen het ontwerp van een gebruikersinterface van een collega op interne consistentie en op consistentie met hun eigen ontwerpen.

#### 5.1.1 Verschillende methoden

Er zijn twee manieren om een consistency inspection uit te voeren. Deze manieren hebben gemeen dat vooraf de inconsistenties in en tussen verschillende systemen gezocht worden, waarna met alle lead ontwerpers besproken wordt op welke wijze de applicaties consistent gemaakt zullen gaan worden.

Het verschil zit hem in het voorwerk. Sommigen kiezen ervoor de ontwerpers zelf naar de ontwerpen van collega's te laten kijken om inconsistenties te zoeken met hun eigen ontwerpen. Anderen laten een usability expert naar de verschillende ontwerpen kijken en een rapport opstellen over de interne en de onderlinge inconsistenties. Beide methoden hebben hun voor- en nadelen, zoals beschreven in onderstaande tabel. In elk project zal gekeken moeten worden welke variant mogelijk is, en hoe de voor- en nadelen tegen elkaar opwegen.

Uitvoerder	Voordelen	Nadelen
Ontwerpers	Ontwerpers zien met eigen ogen de inconsistenties, en ervaren zo waarom eventuele wijzigingen nodig zijn. Zo raken zij gemotiveerd om mee te helpen aan het oplossen van deze inconsistenties.	Ontwerpers moeten vrijgemaakt worden van hun eigen werk.
		Ontwerpers kijken niet altijd vanuit een kritisch oogpunt, zoals een tester dat doet.
		Ontwerpers zijn niet altijd gemotiveerd voor het uitvoeren van testwerk.
Usability expert	De expert heeft doorgaans een beter oog voor usability problemen en inconsistenties	Het inhuren van een usability expert kost geld.
	De usability expert is een neutrale figuur van buitenaf. Zijn oordeel zal minder beladen zijn dat dat van een collega ontwerper.	

#### 5.1.2 Stappenplan

Dit stappenplan kan houvast bieden aan een testcoördinator die een consistency inspection wil organiseren en begeleiden.

##### Vorbereiding

1. Bepaal welke systemen geanalyseerd zullen worden. Kies voor goed vergelijkbare systemen, bijvoorbeeld systemen die op hetzelfde platform ontwikkeld gaan worden.
2. Nodig de evaluatoren uit. Neem minimaal een usability expert en de lead designer van elk systeem. Zorg ervoor dat de designer mandaat heeft om wijzigingen in het ontwerp - en eventueel het product - aan te brengen. Bij voldoende kennis kan de rol van usability expert ingevuld worden door de testcoördinator zelf.
3. Zorg voor een vergaderzaal waar de interfaces besproken kunnen worden, en indien nodig voor handouts of een PC met beamer en scherm.

## *Specificatie*

4. Laat de usability expert de interfaces van de systemen analyseren. Hij noteert opvallende verschillen die een interface vertoont ten opzichte van de interfaces van de andere systemen. De volgende typen inconsistenties kunnen onderscheiden worden:
  - a. Visuele inconsistenties, denk aan layout, kleur, design etc.
  - b. Interactie inconsistenties: verschillende manieren om dezelfde actie uit te voeren.
  - c. Navigatie inconsistenties: bijvoorbeeld scrollen versus bladeren door pagina's.
  - d. Foutafhandeling inconsistenties: bijvoorbeeld het afdwingen van een datumformaat, versus instructies of een foutmelding bij een verkeerd formaat.
  - e. Terminologie inconsistenties: bijvoorbeeld "username", "accountname" en "loginname".
5. verzamel alle bevindingen in een document. Dit document zal als uitgangspunt gebruikt worden voor de inspectie.

## *Uitvoering*

6. Brief de ontwerpers over het doel en het verloop van de sessie.
7. Bediscussieer de eerste inconsistentie uit het document
8. Wanneer een beslissing genomen wordt, noteer deze. Het kan hierbij mogelijk zijn dat om een bepaalde reden besloten wordt de inconsistentie niet aan te pakken. Inconsistente terminologie op portals is bijvoorbeeld verklaarbaar, wanneer de portals een andere doelgroep hebben, zoals particulier en zakelijk.
9. Parkeer het element wanneer de discussie niet snel tot een oplossing leidt.
10. Herhaal stap 7 t/m 9 voor alle inconsistenties uit het document.

## *Afronding*

11. Debrief de deelnemers en bedank ze voor hun aanwezigheid.
12. Plan vervolgspraken in voor de geparkeerde inconsistenties.
13. maak bevindingen en eventueel wijzigingsverzoeken aan voor alle genomen besluiten.
14. Rapporteer de resultaten aan de opdrachtgever.

### **5.1.3 Situatie**

Deze techniek kan het best zo vroeg mogelijk in de ontwikkeling van de onderzochte systemen ingezet worden, omdat bevindingen mogelijk tot ingrijpende ontwerpwijzigingen kunnen leiden. De techniek kan tegen zeer lage kosten ingezet worden. Het is wel raadzaam dat degene die de verschillende systemen analyseert, voldoende kennis heeft van usability. Omdat alleen ontwerpers nodig zijn, kan deze techniek ook in kleine en relatief onvolwassen organisaties uitgevoerd worden. Voorwaarde is wel dat de deelnemers allen mandaat hebben om op basis van de resultaten wijzigingen aan te brengen in hun ontwerp.

### **5.1.4 Resultaat**

Consistency inspection kan in een vroeg stadium van het ontwikkelproces inconsistenties tussen ontwerpen aantonen zonder gebruikers in te schakelen. Dergelijke inconsistenties hebben vaak een negatieve impact op de gebruiksvriendelijkheid van meerdere systemen. Deze inconsistenties wegnemen heeft dan ook een positief effect op de efficiëntie en de effectiviteit waarmee gebruikers hun taken uitvoeren. Een nadeel van de techniek is dat deze techniek zich exclusief richt op het vinden van inconsistenties en geen uitspraken oplevert over de andere usabilityaspecten. Completeer deze techniek dan ook altijd met andere technieken, vooral met usability testtechnieken.

## 5.2 Cognitive Walkthrough

Cognitive walkthrough is een formele groeptechniek, waar met ontwerpers en ontwikkelaars beoordeeld wordt hoe eenvoudig het zou zijn voor een gebruiker om taken uit te voeren met het product.

### 5.2.1 Stappenplan

Dit stappenplan biedt houvast aan een testcoördinator die een cognitive walkthrough moet organiseren en begeleiden.

#### Vorbereiding

1. Bepaal het doel van de walkthrough. Selecteer indien nodig welke (deel) producten getoetst worden.
2. Nodig deelnemers uit. Nodig minimaal één ontwerper en één ontwikkelaar uit, deze groep kan uitgebreid worden met meer ontwerpers en ontwikkelaars, een (extra) usability expert of marketing medewerkers, afhankelijk van de aard en grootte van het project.
3. Zorg voor een vergaderzaal waar de interface gepresenteerd kan worden. Zorg bij een paper prototype voor schermafdrukken, en bij een werkend prototype voor een PC, beamer en scherm.

#### Specificatie

4. Identificeer de doelgroep van het product. Persona's kunnen hierbij als input dienen. Belangrijk is hoeveel ervaring ze hebben met IT in het algemeen, en met vergelijkbare producten in het bijzonder. Voor webapplicaties is ook internet ervaring belangrijk.
5. Bepaal welke taken geanalyseerd moeten worden. Zorg ervoor dat de verzameling taken niet te groot is, maar nog wel representatief. Gebruik als input bijvoorbeeld de eisen, een feasibility study, een quick scan of een marketing analyse
6. Beschrijf de interface: in welke staat is de interface voor elke taak, en hoe reageert de interface op acties van de gebruiker. Gebruik hiervoor een interface ontwerp of een (paper) prototype.
7. Beschrijf voor elke taak, welke acties de gebruiker uit moet voeren met de huidige interface om de taak te voltooien. Ga ervan uit dat de gebruiker nog niet bekend is met de interface.

#### Uitvoering

8. Brief de deelnemers. Laat ze weten hoe de sessie gaat verlopen en wat er van hun verwacht wordt.
9. Begin de sessie. Stel de deelnemers bij de eerste actie van de eerste taak de volgende vier vragen:
  - a. Zal de gebruiker proberen het juiste resultaat te bereiken?
  - b. Zal de gebruiker zien dat de juiste actie beschikbaar is?
  - c. Zal de gebruiker de juiste actie correleren aan het te bereiken resultaat?
  - d. Zal de gebruiker, wanneer hij de juiste actie uitvoert, zien dat hij vooruitgang maakt bij het uitvoeren van zijn taak?
10. Probeer aan de hand van de antwoorden van de deelnemers een onderbouwd "succesverhaal" te construeren. Lukt dat niet, probeer een onderbouwd "doemscenario" te construeren.
11. Noteer de bevindingen en ga verder met de volgende actie.
12. Herhaal stap 9 t/m 11 voor alle acties van alle taken.

#### Afronding

13. Debrief de deelnemers kort en bedank hun voor hun inzet.
14. Stel een rapport samen met de belangrijkste bevindingen en conclusies, en stuur dit naar de opdrachtgever.

### **5.2.2 Varianten**

Rick Spencer ervaarde vier problemen bij het uitvoeren van deze techniek: tijdsdruk, een overdaad aan discussie, veel documentatie en defensieve reacties. Om deze problemen te verhelpen introduceerde hij de streamlined cognitive walkthrough techniek. Hij bedacht 4 grondregels voor de discussie en reduceerde het aantal tijdens de uitvoering te stellen vragen tot 2.

Grondregels:

1. Niet ontwerpen
2. Geen ontwerpen verdedigen
3. Geen discussie over cognitieve theorieën
4. De usability expert leidt de discussie

Vragen:

1. Zal de gebruiker weten wat hij hier moet doen?
2. Als de gebruiker de juiste actie uitvoert, weet hij dan dat zijn actie de juiste was, en dat hij naar zijn doel toewerkt?

### **5.2.3 Situatie**

Deze techniek is het meest geschikt om toe te passen op (paper) prototypes, maar kan ook voor volledige systemen gebruikt worden. De techniek hoeft niet duur te zijn, maar kan erg tijdrovend worden bij grote en complexe systemen. Belangrijk is wel dat degene die de sessie voorbereid enige kennis van mens-computer interactie heeft. Omdat alleen ontwerpers en ontwikkelaars nodig zijn, kan deze techniek ook in kleine en relatief onvolwassen organisaties uitgevoerd worden, zolang de deelnemers maar open staan voor opbouwende kritiek op hun werk.

### **5.2.4 Resultaat**

Cognitive walkthrough kan in een vroeg stadium usability problemen aantonen zonder gebruikers in te schakelen. Dit heeft als voordeel dat ontwerpers en ontwikkelaars een gevoel krijgen bij usability, en dat geen gebruikers gerekruteerd hoeven te worden. Nadelen hieraan zijn dat de deelnemers zich mogelijk anders gedragen dan echte gebruikers, en dat de techniek niets aantoont over de tevredenheid van de eindgebruiker. Completeer deze techniek daarom altijd met andere technieken, bij voorkeur met usability testtechnieken.





### 5.3 Model-Based Inspection

Model-based inspection is een formele individuele toetstechniek. Aan de hand van een ontwerp wordt een model gemaakt van de gebruikershandelingen, die nodig zijn om taken uit te voeren. Hiermee wordt geëvalueerd of het testobject geschikt is om de taken uit te voeren waarvoor het systeem bedoeld is, en zo ja, hoeveel tijd dit zal kosten.

#### 5.3.1 Stappenplan

Dit stappenplan biedt houvast aan een reviewer om een model-based inspection uit te voeren

##### *Vorbereiding*

1. Kies een techniek aan de hand waarvan het model ontwikkeld gaat worden. In de volgende subparagraaf wordt als voorbeeld de CPM-GOMS techniek beschreven.
2. Bepaal welk ontwerp getoetst zal worden volgens deze methode. Vaak zal dit een functioneel ontwerp, of een interface ontwerp zijn. Zorg ervoor dat het ontwerp voldoende detailniveau bevat om een model te ontwikkelen.
3. Indien van toepassing voor het model: interview gebruikers om een idee te krijgen van hun manier van werken. Ook een persona, zie paragraaf 6.3, kan indien beschikbaar nuttige input leveren voor het model.

##### *Specificatie*

4. Maak volgens de gekozen techniek het model van de gebruikershandelingen per taak.
5. bepaal per taak hoe lang deze in de praktijk zal duren.

##### *Uitvoering*

6. Evalueer het model. Zoek naar acties die onevenredig veel stappen bevatten en tijd kosten, deze geven vaak een indicatie van een probleem in het ontwerp. Probeer ook acties te identificeren die volgens het model bijzonder efficiënt verlopen. Deze kunnen als input dienen voor wijzigingen aan het ontwerp.

##### *Afronding*

7. Verzamel de bevindingen en rapporteer deze aan de opdrachtgever.

#### 5.3.2 CPM-GOMS

In de GOMS technieken (Goals, Operators, Methods and Selection rules) worden de doelen (Goals) van een gebruiker beschreven, waarbij de acties (Operators) samen de methodes (Methods) vormen waar de gebruiker uit kan kiezen (Selection rules) om de doelen te bereiken.

De oorspronkelijke vorm van deze techniek heet CMN-GOMS.

#### 5.3.3 Situatie

Model-based testing wordt in de regel alleen ingezet voor het toetsen van functionele of interface ontwerpen, met voldoende detailleringniveau. Voor deze techniek is het gewenst dat de reviewer enig verstand heeft van mens-computer interactie. In dat geval is het een techniek die breed ingezet kan worden, zowel op kleine als op grotere projecten. Een mogelijke toepassing van deze techniek is om een ontworpen systeem te vergelijken met een huidig systeem. Zo kan in een vroeg stadium inzicht gegeven worden in de verbetering, of het gebrek daaraan, van de efficiëntie van de taken die met het systeem uitgevoerd gaan worden.

### **5.3.4 Resultaat**

Model-based testing kan in een vroeg stadium belangrijke knelpunten in een gebruikersinterface vinden. Voordeel is dat de techniek sterk gericht is op taken, en daardoor concrete problemen identificeert. De techniek heeft echter een aantal nadelen. De techniek richt zich uitsluiten op het uitvoeren van taken, en houdt daarbij geen rekening met (gebrek aan) functionaliteit, het maken van fouten en hoe de gebruiker de interface ervaart.

## 5.4 Pluralistic Walkthrough

Pluralistic walkthrough is een formele groep toetstechniek, waarbij een groep van gebruikers, ontwikkelaars en usability experts beschrijft hoe zij scenario's en taken uit zouden voeren met het testobject en hierover in discussie gaat.

### 5.4.1 Stappenplan

Dit stappenplan kan een houvast bieden aan een reviewer om een pluralistic walkthrough te organiseren

#### *Vorbereiding*

1. Bepaal de deelnemers van de sessie, en nodig deze uit. Neem in ieder geval (een representant van) de eindgebruikers, ontwerpers, ontwikkelaars en usability experts. Denk goed na over de aantallen deelnemers, te veel kan de sessie moeilijk controleerbaar maken. Neem minimaal twee gebruikers, twee ontwikkelaars en een usability expert. Verdubbel deze aantallen wanneer het budget het toelaat.
2. Bereid de deelnemers voor op de sessie. Ontwikkelaars zouden veel kritiek kunnen krijgen. Vertel ze dat ze alle opmerkingen positief moeten benaderen voor een maximaal effect. Laat de deelnemers ook van tevoren weten dat opmerkingen niet per definitie tot aanpassingen hoeven te leiden.

#### *Specificatie*

3. Kies een basistechniek voor de testgevallen. Aan te raden technieken om dergelijke scenario's te ontwikkelen zijn de use-case techniek en de procescyclus test, omdat deze technieken in de regel representatieve scenario's opleveren. Als testbasis kunnen use-cases, procesbeschrijvingen of functioneel ontwerpen dienen.
4. Specificeer de scenario's globaal.
5. Bereid instructies voor en zorg voor schermprints per scenario waar de deelnemers tijdens de sessie opmerkingen bij kunnen schrijven.
6. Bereid per scenario een korte enquête voor over de gebruiksvriendelijkheid van het testobject.

#### *Uitvoering*

7. Geef de deelnemers altijd vooraf instructies over het verloop van de sessie en de regels die tijdens de sessie gelden. Zie hiervoor de onderstaande stappen. Geef ze ook de schermprints.
8. Laat een ontwikkelaar een globaal overzicht geven van het eindproduct, zodat de gebruikers de scenario's in een context kunnen plaatsen.
9. Vraag de gebruikers voor het eerste globale scenario op te schrijven, welke acties zij zouden uitvoeren om het scenario uit te voeren. Overleg is niet toegestaan.
10. Vraag de gebruikers elkaar te vertellen wat zij opgeschreven hebben en te discussiëren over elkaars antwoorden en potentiële usability problemen. De ontwikkelaars moeten stil blijven.
11. De usability experts identificeren de usability problemen die naar hun idee uit de discussie naar voren zijn gekomen aan de hele groep.
12. De ontwerpers en ontwikkelaars reageren op de bevindingen en leggen uit waarom zij bepaalde beslissingen hebben gemaakt. Zij moeten hierbij open staan voor opbouwende kritiek.
13. Laat de ontwikkelaars indien nodig vertellen hoe het scenario uitgevoerd had moeten worden.
14. Laat de gebruikers de enquête invullen.
15. Herhaal stap 8 t/m 14 voor elk scenario.

#### *Afronding*

16. Debrief de deelnemers. Laat ze weten wat er met de resultaten gaat gebeuren en bedank ze voor hun deelname.
17. Verzamel de resultaten, bevindingen en verbeterpunten en rapporteer deze aan de opdrachtgever.

### **5.4.2 Situatie**

Pluralistic walkthroughs zijn vooral effectief voor het toetsen van tussenproducten. Het toetsen van ontwerpen is mogelijk, maar de techniek is effectiever wanneer het testobject een (paper) prototype is. Deze techniek is bij uitstek geschikt voor iteratieve ontwikkelmethoden in het algemeen, en in het bijzonder user-centered design methodes, omdat hierbij ontwikkelaars al veel samenwerken met eindgebruikers. Voor kleinere projecten is deze techniek vaak te arbeidsintensief, maar voor grotere projecten kunnen in een vroeg stadium resultaten bereikt worden. Deze techniek is heel geschikt om het belang van usability binnen de ontwikkelorganisatie aan te tonen, maar ook om de acceptatie van een nieuw systeem door gebruikers te verhogen.

### **5.4.3 Resultaat**

Pluralistic walkthrough kan belangrijke design issues aantonen voordat ook maar begonnen is met de bouw van het systeem. Het gebruik van een multidisciplinaire groep kan de creativiteit van de deelnemers aanwakkeren en wederzijds begrip kweken. De techniek is echter niet eenvoudig in te zetten. In een druk project is het vaak lastig tien mensen langere tijd bij elkaar te krijgen. Daarnaast kunnen ontwikkelaars moeite hebben met de kritiek die zij krijgen. Tenslotte is de techniek beperkt in zijn scope: slechts een beperkt aantal scenario's kan behandeld worden, en met zoeken- en verkengedrag kan geen rekening gehouden worden

## 5.5 Focus Groups

Focus groups is een informele toetstechniek, waarbij gebruikers in een groep discussiëren over een product. Deze techniek wordt veel toegepast in de marketingwereld, en is “overgewaaid” naar systeemontwikkeling doordat marketing medewerkers invloed kregen op de ontwikkeling van websites.

### 5.5.1 Stappenplan

Dit stappenplan biedt houvast aan een organisator van een focus Group sessie.

#### *Vorbereiding*

1. Bepaal het onderwerp van de sessie. Dit kan zijn een compleet product of een prototype, maar ook een specifieke design beslissing.
2. Omdat focus group sessies vrij chaotisch kunnen verlopen, wordt veel gevraagd van de kwaliteiten van de moderator van de sessie. Overweeg een professionele moderator in te huren. De organisator kan dan ook de rol van observator invullen.
3. Regel faciliteiten voor de sessie. Een vergaderzaal met tafels en stoelen is logisch, maar vergeet niet indien nodig opname apparatuur, dranken, lunch, een monitor of een beamer met scherm, eventueel teleconferencing apparatuur, en denk na of nog andere zaken nodig zijn.
4. Rekruteer deelnemers uit de gebruikersorganisatie. Denk goed na over de aantallen deelnemers, te veel kan de sessie moeilijk controleerbaar maken. Gebruikelijk is tussen de zes en negen deelnemers.

#### *Specificatie*

5. Werk de onderwerpen uit. Denk hierbij aan een demo van een (tussen)product, een aantal designopties, een stelling, een beschrijving van features waarover gediscussieerd kan worden, etc. bepaal ook hoe lang de discussie per onderwerp mag duren.

#### *Uitvoering*

6. Geef de deelnemers altijd vooraf instructies over het verloop van de sessie, de agenda en waar hun uitspraken voor gebruikt gaan worden.
7. Introduceer het eerste onderwerp.
8. Zet eventueel de opnameapparatuur aan en start de discussie. De moderator moet er hierbij voor zorgen dat de discussie vloeiend verloopt, maar er wel voor zorgen dat de deelnemers bij het onderwerp blijven, en dat alle deelnemers gehoord kunnen doen.
9. Wanneer geen opnameapparatuur gebruikt wordt moet de observator aantekeningen maken over de onderwerpen en de doelen die vooraf bedacht zijn.
10. Wanneer geen nieuwe inzichten meer op tafel gelegd worden, of de tijd op is, wordt de discussie stop gezet. Herhaal eventueel de belangrijkste punten.
11. Herhaal stap 7 t/m 10 voor elk onderwerp

#### *Afronding*

12. Bespreek na met de deelnemers. Laat ze weten wat er met de resultaten gaat gebeuren en bedank ze voor hun deelname.
13. Analyseer de resultaten. Werk aantekeningen of opnames uit tot argumenten, meningen, verbeterpunten en conclusies, afhankelijk van de onderwerpen, doelen en resultaten. Ook kan algemeen commentaar toegevoegd worden over de stemming van de groep. Diepgaande analyses op focus group sessies zijn vaak lastig uit te voeren en leveren zelden betrouwbare resultaten op.
14. Rapporteer de resultaten aan de opdrachtgever.

### **5.5.2 Varianten**

Een variant op deze techniek is om de discussie op een intra- of internet forum te houden. Dit elimineert de kosten voor ruimtes, opnameapparatuur en een moderator. Het is verstandig de discussie te modereren op het moment dat de groep van het onderwerp af dwaalt. Een nadeel is dat de (vaste) deelnemers aan een forum vaak niet representatief zijn voor de hele gebruikersgroep. Daarom kan gekozen worden voor een gesloten discussie op een afgeschermd discussieforum.

### **5.5.3 Situatie**

Focus groups kunnen in elke fase van systeemontwikkeling ingezet worden, tot ver na in gebruik name. De beste resultaten zijn echter te behalen tijdens de eisendefinitie en de ontwerpfase. Zeker bij een grote gebruikersgroep verdient het de voorkeur meerdere focus group sessies uit te voeren om betrouwbare conclusies te kunnen trekken. Hoewel de onderwerpen sterk uiteen kunnen lopen, worden in de praktijk de bruikbaarste resultaten bereikt wanneer gebruikers gevraagd wordt, wat zij willen van een product. Voor een eenvoudig onderwerp kan een sessie in twee uur uitgevoerd worden, bij omvangrijkere discussies kan dit oplopen tot twee dagen. Denk hieraan bij de voorbereiding. Deze techniek kan tegen lage kosten toegepast worden in een kleinere organisatie, het vergt echter wel een open en volwassen cultuur van de gebruikersorganisatie. Ook is een capabele moderator een noodzaak, om te voorkomen dat de discussie afdwaalt, of gedomineerd wordt door één of enkele deelnemers.

### **5.5.4 Resultaat**

Focus groups is een techniek waarmee in een vroeg stadium al belangrijke inzichten verkregen kunnen worden in de meningen van de eindgebruikers van een te ontwikkelen product. Het kan ook ingezet worden om ontwerpvragestukken voor te leggen. Omdat deze techniek de gebruikers de gelegenheid geeft uitgebreid te discussiëren over hun werkzaamheden, kunnen resultaten in de context van het dagelijkse werk geplaatst worden. Deze techniek kent echter wel een aantal beperkingen. Belangrijk is om te onthouden dat wat gebruikers zeggen te doen, niet altijd is wat gebruikers in de praktijk doen. Dat gebruikers aangeven iets te willen, betekent nog niet dat dat de beste manier is waarop hun werk ondersteund kan worden. Voer daarom altijd naast deze techniek ook usability testen uit. Daarnaast is één focus group in een grote organisatie vaak niet representatief voor de hele gebruikersgroep. In dat geval kunnen meerdere focus group sessies uitgevoerd worden. Uiteraard worden de kosten van de techniek dan hoger.

## 5.6 Reverse Card Sorting

Reverse card sorting is een toetstechniek om de structuur van een ontworpen of al gebouwde applicatie te evalueren. Een gebruiker krijgt de taak om bepaalde informatie op te zoeken in een hiërarchische structuur. De naam is afgeleid van de techniek Card Sorting, waarmee gebruikers een hiërarchische informatiestructuur creëren. Die techniek wordt in het volgende hoofdstuk behandeld.

Reverse card sorting wordt vaak Tree testing genoemd. Omdat deze techniek juist zijn meerwaarde bewijst als toetstechniek, wordt die naam in dit whitepaper niet gebruikt.

### 5.6.1 Stappenplan

Dit stappenplan biedt houvast aan de testcoördinator of de beheerder die een reverse card Sorting sessie wil organiseren.

#### *Vorbereiding*

1. Rekruteer gebruikers uit de (toekomstige) gebruikersorganisatie. Uit onderzoek is gebleken dat minimaal 5 gebruikers nodig zijn om een representatief resultaat te bereiken. Meer dan 30 gebruikers heeft over het algemeen geen toegevoegde waarde.
2. Werk de structuur van de te testen website uit in een simpele text versie, of op papier, waarbij elk menu een eigen pagina heeft.

#### *Specificatie*

3. Ontwikkel diverse taken voor de gebruiker, waarbij typische doelen bereikt moeten worden. Houd hierbij rekening met het doel van de website. Vergeet ook niet de gebruiker te laten zoeken naar standaardinformatie, zoals contactgegevens of de sitemap.

#### *Uitvoering*

4. Brief de gebruiker. Leg hem in ieder geval uit wat er van hem verwacht wordt.
5. Geef de gebruiker de eerste taak.
6. Toon de gebruiker het hoofdmenu.
7. Laat de gebruiker een menu element kiezen, en toon hem het submenu.
8. Laat de gebruiker op deze wijze door de menustructuur gaan. Hij mag ook teruggaan naar bovenliggende menu's. Ga verder met de volgende taak, wanneer hij een onderwerp heeft gevonden waarvan hij denkt dat hij de gevraagde informatie daar aan zal treffen, of wanneer hij het opgeeft.
9. Herhaal stap 4 t/m 8 voor alle taken en alle gebruikers.

#### *Afronding*

10. Analyseer de resultaten. Kijk voor elke taak of gebruikers deze konden uitvoeren, of ze veel tijd nodig hadden en of ze vaak terug moesten gaan. Bepaal zo of specifieke delen van de structuur wel of niet goed werken en waar verbeteringen mogelijk zijn.
11. Rapporteer de resultaten aan de opdrachtgever.

### 5.6.2 Situatie

Deze methode wordt in de praktijk gebruikt om menustructuren en informatie architecturen te testen. Het is een snelle en goedkope manier om een ontwerp te testen, of om issues in een bestaande applicatie te onderzoeken. Omdat de techniek met pen en papier tegen lage kosten uit te voeren is, is deze techniek ook bij een krap budget in te zetten. Deze techniek kan ook ondersteund worden middels tooling, zie hiervoor paragraaf 7.3.



Aanleiding voor het uitvoeren van reverse card sorting op een bestaande applicatie kan zijn dat gebruikers structureel problemen ondervinden met het zoeken naar informatie of functionaliteiten in een applicatie, of dat (delen van) een webapplicatie tegenvallende bezoekersaantallen hebben. Het nadeel hiervan is dat een redesign in dit stadium veel tijd en moeite zal kosten. Dit gebruik wordt vooral aangeraden wanneer de problemen kritisch zijn, of een nieuwe versie van de applicatie al ingepland is.

### **5.6.3 Resultaat**

Reverse Card sorting kan een ontwerper van een menustructuur of een informatie architectuur snel feedback geven over zijn ontwerp. Bij voorkeur is het ontwerp nog niet volledig uitgewerkt, zodat wijzigingen snel te maken zijn. Gunstig hierbij is dat de techniek ook inzicht kan geven in waar de fout zich precies bevindt.

Ook kan reverse card sorting problemen in de structuur van een bestaande applicatie aan het licht brengen en duidelijk maken waar die problemen zich bevinden. Op deze manier kan het een goed middel zijn om vooronderzoek te verrichten voor een nieuwe versie van de applicatie.

## 6 PREVENTIEVE MAATREGELEN

De technieken die in dit hoofdstuk worden genoemd zijn niet onder te brengen onder de noemer “test” – of “toetstechnieken”. Ze kunnen namelijk al worden ingezet voordat er een stuk software of een ontwerp is gemaakt. Ze zijn al in eerdere fasen bruikbaar, bijvoorbeeld om tot een usable ontwerp te komen of om het hele project of beheerteam een bewustzijn te geven over het soort gebruiker dat de applicatie zal hebben.

Deze technieken zijn dus niet uitsluitend het domein van de tester. Ze zijn voor alle partijen van belang. Ze horen met nadruk óók bij de toolkit van de tester. De technieken zijn namelijk op verschillende manieren en in verschillende fasen inzetbaar. Een techniek als persona’s is bijvoorbeeld dusdanig adaptief dat als deze niet in een vroege fase is toegepast nog wel in een latere projectfase een toegevoegde waarde kunnen hebben.

De term “preventieve maatregel” komt voort uit het feit dat de technieken door vroeg aandacht te geven aan usability kunnen voorkomen dat er niet voldoende aandacht wordt gegeven aan usability tijdens het hele ontwikkeltraject.

### 6.1 Card Sorting

Card sorting is een techniek die ingezet kan worden om input te genereren voor een gebruiksvriendelijk ontwerp van een applicatie. Het resultaat is een mogelijke structuur voor deze applicatie, gebaseerd op de verwachtingen van een groep gebruikers. Card sorting kan ook gebruikt worden om de structuur van een al gebouwde applicatie te evalueren. In dat geval kan het resultaat input bieden voor een wijzigingsverzoek, of een nieuwe versie.

Bij een Cardsort wordt proefpersonen gevraagd de onderwerpen die op een website of in een programma te vinden zijn, te groeperen en deze groepen te benoemen. Door dit met een aantal proefpersonen te doen en hier het gemiddelde van te nemen komt men tot een voor gebruikers begrijpelijke indeling.

#### 6.1.1 Stappenplan

Het onderstaande stappenplan biedt houvast aan de testcoördinator of de beheerder die een Card sorting sessie organiseert.

##### *Vorbereiding*

1. Rekruteer gebruikers uit de (toekomstige) gebruikersorganisatie. Uit onderzoek is gebleken dat met 15 gebruikers doorgaans een representatief resultaat bereikt kan worden. Nodig indien mogelijk voor extra zekerheid 20 tot 30 gebruikers uit. De testers worden doorgaans uitgenodigd voor individuele sessies, maar het is ook mogelijk sessies uit te voeren met groepen van 3 t/m 5 gebruikers.

##### *Specificatie*

2. Verzamel alle in de testbasis, of in de bestaande applicatie gebruikte begrippen en zet elk begrip op een kaartje. De begrippen kunnen bijvoorbeeld zijn taken die een gebruiker moet vervullen, verschillende informatieonderdelen die geraadpleegd moeten kunnen worden, functies die uitgevoerd moeten kunnen worden, etc.

##### *Uitvoering*

3. Brief de gebruiker, leg hem in ieder geval uit wat er van hem verwacht wordt.
4. Schud de kaarten met de begrippen van de applicatie en geef deze aan de gebruiker.
5. Vraag de gebruiker de kaarten te verdelen in stapels, zodat kaarten die bij elkaar horen in dezelfde stapel liggen. Stel geen eisen aan het aantal stapels.
6. Optioneel: bij grote aantallen begrippen kan gevraagd worden om ook de stapels te groeperen.

7. Optioneel: vraag de gebruikers de stapels en groepen stapels van een naam te voorzien. Bij een zogeheten open card sort moeten de gebruikers zelf namen verzinnen, bij een closed card sort krijgen zij de keuze uit een verzameling namen.
8. Herhaal stap 3 t/m 7 voor alle (groepen) gebruikers.

### *Afronding*

9. Analyseer de resultaten. Voor elke kaart kan gekeken worden hoeveel gebruikers deze in dezelfde stapel hebben geplaatst. Dit levert scores op waarmee de, in de ogen van deze groep gebruikers, meest logische structuur vastgesteld kan worden. Ook de namen van de stapels en groepen kunnen op dezelfde wijze geanalyseerd worden.
10. Optioneel: vergelijk de verkregen structuur met de structuur van de bestaande applicatie, en identificeer de grootste verschillen.
11. Rapporteer de resultaten aan de opdrachtgever.

### **6.1.2 Situatie**

Deze methode wordt in de praktijk gebruikt om menustructuren, workflows en navigatiepaden voor gestructureerde applicaties te ontwikkelen. Vanwege de lage kosten is deze techniek ook bij een klein budget in te zetten. Wanneer een applicatie ontwikkeld wordt voor een domein, waarvoor een gestandaardiseerde taxonomie bestaat, geniet het vaak de voorkeur deze over te nemen.

Als testtechniek is Card sorting relatief beperkt inzetbaar, omdat de techniek niet verder kijkt dan de structuur van een applicatie. Het is aan te raden deze techniek alleen in te zetten, wanneer vermoed wordt dat de applicatiestructuur problemen oplevert. Verschillen tussen de uitkomst en de gerealiseerde structuur, hoeven namelijk niet automatisch te betekenen dat gebruikers met de gerealiseerde structuur problemen ondervinden. Aanleiding voor het uitvoeren van Card sorting op een bestaande applicatie kan zijn dat gebruikers structureel problemen ondervinden met het zoeken naar informatie of functionaliteiten in een applicatie, of dat (delen van) een webapplicatie tegenvallende bezoekersaantallen hebben.

### **6.1.3 Resultaat**

Card sorting is een techniek waarmee de gebruikers al voor het eerste ontwerp input kunnen geven. Het voordeel is dat een applicatie gestructureerd wordt volgens het mentale model van de gebruiker, en niet volgens die van de ontwerper. Dit kan in latere fasen ingrijpende ontwerpbevindingen voorkomen.

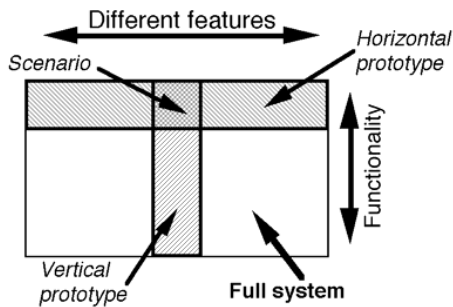
Daarnaast kan na de ontwikkeling Card sorting input geven voor een redesign, wanneer de structuur van de applicatie problemen lijkt te veroorzaken. Nadeel van dit gebruik is dat de herstellkosten vaak relatief erg hoog zijn. Deze toepassing biedt het meeste voordeel, wanneer toch al een nieuwe versie van een systeem ingepland is.

## 6.2 Prototype testing

Prototype testing is een techniek waarmee usability getest kan worden op tussenproducten tijdens de ontwikkeling. Deze techniek gaat uit van een iteratief ontwikkelproces, waarbij usability testing een onderdeel vormt van elke iteratie.

### 6.2.1 Soorten Prototypes

Er zijn grofweg 3 typen prototypes te onderscheiden, zoals te zien in onderstaande figuur:



- Verticale prototypes: bevatten alle functionaliteiten van een subset van een deel van het systeem.
- Horizontale prototypes: bevat het hele systeem, maar met gereduceerde functionaliteit.
- Scenario's: bevatten gereduceerde functionaliteiten van een deel van het systeem.

De verschillende typen prototypes zijn voor verschillende doelen in te zetten. Waar scenario's vaak geschikt zijn om interactievormen uit te testen, zijn horizontale prototypes vooral van belang bij applicaties waar navigatie en zoekgedrag een grote rol spelen. Vertical prototypes worden veel ingezet op projecten waar deelproducten achter elkaar afgerond en gefaseerd geïmplementeerd worden.

Paper prototypes zijn een specifieke toepassing van scenario's en worden doorgaans nog tijdens de ontwerpfasen ingezet. Deze prototypes hebben als voordeel dat usability testing uitgevoerd kan worden nog voordat er één regel code is geprogrammeerd. Hierdoor kunnen in een vroeg stadium ontwerpbevindingen gevonden en opgelost worden zonder ontwikkelingspanning, waar dat later in het ontwikkelproces veel moeite had gekost.

Meer over prototypes is te vinden in het hoofdstuk infrastructuur.

### 6.2.2 Stappenplan

Dit stappenplan biedt houvast aan een testcoördinator die verantwoordelijk is voor prototype testing. De stappen herhalen zich voor elke iteratie.

#### Planning

1. Bepaal welke technieken uitgevoerd gaan worden voor de iteratie. Houd hierbij rekening met het tussenproduct dat de iteratie oplevert: een paper prototype vraagt om een andere aanpak dan een werkend product.
2. Optioneel: Schrijf een detail testplan, waarin o.a. de technieken beschreven worden, en de planning en begroting van de test uiteengezet wordt. Bij korte iteraties kan deze stap overgeslagen worden.

#### Vorbereiding

3. Zorg ervoor dat alle benodigde faciliteiten en tools beschikbaar zijn.
4. Zorg ervoor dat alle benodigde testers beschikbaar zijn.

5. Vergaaf feedback op bevindingen gedaan in eerdere iteraties.
6. Voer een intake uit op de testbasis. Let hierbij specifiek op de wijzigingen t.o.v. eerdere versies, waaronder bevindingen die opgelost zijn.

### *Specificatie*

7. Kies de te gebruiken basistechnieken.
8. Verzamel alle herbruikbare testscripts en actualiseer deze indien nodig.
9. Specificeer indien nodig extra testscripts.
10. Specificeer hertesten naar aanleiding van bevindingen uit eerder iteraties.

### *Uitvoering*

11. Voer de geplande testen uit en registreer de resultaten.

### *Afronding*

12. Log bevindingen volgens de bevindingenprocedure.
13. Stel een rapport van de test samen en stuur dit naar de opdrachtgever.

## **6.2.3 Varianten**

Een door usability expert Bruce Tognazzini geïntroduceerde variant is closed-coupled testing. Deze techniek is vooral bruikbaar in de eerste iteraties van de ontwikkeling. De techniek kent 3 stappen:

1. test de ontwikkelde interface
2. analyseer de bevindingen
3. breng wijzigingen aan.

Deze stappen worden herhaald totdat geen bevindingen meer gevonden worden. Deze techniek is uitermate geschikt om in een vroeg stadium de belangrijkste usability bevindingen uit een interface te krijgen. Het testobject kan variëren van paper prototypes tot de eerste werkende iteraties. Met name bij kleinere applicaties is dit een goedkope en snelle methode om bruikbare interfaces te ontwikkelen. Het vereist echter wel commitment van het management, om eindgebruikers op korte termijn hiervoor beschikbaar te stellen. Deze techniek werkt het best wanneer het ontwikkelteam klein is.

## 6.3 Persona's

Persona's zijn archetypische gebruikers van een applicatie. Een persona geeft de doelen en taken van een bepaalde groep gebruikers weer, maar ook hun persoonlijke karakteristieken. Hoewel de personas fictieve gebruikers zijn, zijn hun doen en laten gebaseerd op kennis die men bezit over echte gebruikers. Deze kennis kan op verschillende manieren worden verzameld, zoals interviews, de analyse van marktonderzoek en gesprekken met mensen binnen de organisatie met veel klantcontact. Welke onderzoeksmethode er wordt gebruikt hangt af van de situatie waar men zich in bevindt en over wat voor gegevens men beschikt.



Bob is 52 years old and works as a mechanic with an organisation offering road service to customers when their car breaks down. He has worked in the job for the past 12 years and knows it well. Many of the younger mechanics ask Bob for advice when they meet up in the depot as he always knows the answer to tricky mechanical problems. Bob likes sharing his knowledge with the younger guys, as it makes him feel a valued part of the team.

Bob works rolling day and night shifts and spends his shifts attending breakdowns and lockouts (when customers lock their keys in the car). About 20% of the jobs he attends are complex and he occasionally needs to refer to his standard issue manuals. Bob tries to avoid using the manuals in front of customers as he thinks it gives the impression he doesn't know what he's doing.

Bob has seen many changes over the years with the company and has tried his best to move with the times. However he found it a bit daunting when a new computer was installed in his van several years ago, and now he has heard rumours that the computer is going to be upgraded to one with a bigger screen that's meant to be faster and better.

Bob's been told that he will be able to access the intranet on the new computer. He has heard about the intranet and saw once in an early version on his manager's computer. He wonders if he will be able to find out what's going on in the company more easily, especially as customers' seem to know more about the latest company news than he does when he turns up at a job. This can be embarrassing and has been a source of frustration for Bob throughout his time with the company.

Bob wonders if he will be able to cope with the new computer system. He doesn't mind asking his grandchildren for help when he wants to send an email to his brother overseas, but asking the guys at work for help is another story.

### 6.3.1 Verschillende soorten persona's

Er zijn verschillende soorten personas. Soms worden ze heel ver uitgewerkt, met veel persoonlijke details, andere keren worden ze minder ver uitgewerkt en bestaan ze uit een lijst met punten die voor de gebruiker van belang zijn. Hierboven een voorbeeld van een persona. In dit geval gaat het om een persona die is ontwikkeld voor het ontwerp van een intranet site voor een wegenwachtorganisatie in Amerika.

De volgende lijst geeft concrete toepassingen van persona's:

- het identificeren van de eigenschappen, functionaliteiten en inhoud die moeten worden ontwikkeld voor een intranet of website release, waarbij de waarde voor de gebruiker is verzekerd
- het bepalen of één gebruikersinterface voldoende is om aan de eisen van alle gebruikers te voldoen, of dat het nodig is om twee of meer interfaces te ontwikkelen voor verschillende gebruikersgroepen
- het communiceren naar leidinggevenden of opdrachtgevers over de visie op een nieuwe website of intranet en hoe deze aan de wensen en eisen voldoen
- het maken van een ontwerpbeslissing over hoe een stukje functionaliteit precies zal werken over het creatieve ontwerp ervan

- het samenstellen van het inhoudelijke ontwerp van een applicatie, bijvoorbeeld welke informatie bevat het wel en niet
- het begeleiden van een expert review
- het ontwikkelen van scenario's voor een usability test met proefpersonen of een cognitive walkthrough
- Het voorbereiden van focus group sessies

### 6.3.2 Toepassing

Persona's zijn bijvoorbeeld een prima manier om de val te ontlopen dat er ontwikkeld wordt waar een gebruiker om vraagt in plaats van wat hij in de praktijk zal gebruiken. Ook zijn ze vrij snel te ontwikkelen en daarom snel toepasbaar in iteraties. Persona's kunnen ook een goede manier zijn om performance en defect requirements te kunnen koppelen aan design requirements.

In algemene zin kunnen persona's op veel momenten in het usabilitytesttraject worden ingezet om het perspectief van de gebruiker te beleven. Ze vormen een hulpmiddel om argumentatie als het ware namens de gebruiker te kunnen vormen. Concreet kunnen persona's worden gebruikt bij een persona based inspection, een variant op de perspective based inspection (zie paragraaf 5.2 en 5.2.3). Ook kunnen ze als hulpmiddel worden ingezet bij een focus group sessie (zie paragraaf 6.5) of bij het opstellen van testscenario's voor proefpersonen (zie paragraaf 5.7).

### 6.3.3 Persona's opstellen

Door verschillende informatiebronnen te gebruiken wordt een beeld gevormd van een archetypische gebruiker. Hierdoor wordt de focus van het systeemontwerp gelegd bij het ondersteunen van gebruikersdoelen in plaats van de ideeën van het ontwerpteam of de wensen van de manager. Uiteindelijk wordt de informatie gebundeld en worden er een of meerdere persona's ontwikkeld.

#### *Stap 1: Kies een onderzoeksmethode voor het inventariseren van eigenschappen van de persona's*

Het doel van het onderzoek is om de trends en patronen van het gedrag, verwachtingen en motivaties van gebruikers in kaart te brengen. Dit vormt de basis van de persona's. De beste manier om dit te doen is door gebruikers te interviewen. Dit kan eenvoudig bij een intranet, maar wordt lastiger als het een website betreft waarvan de doelgroep minder toegankelijk is.

Als de gebruikersgroep bereikbaar is, kies dan voor interviews. Maak een selectie van gebruikers voor de interviews. Denk bij het vernieuwen van en bestaande applicatie niet alleen aan huidige gebruikers, maar ook aan potentieel nieuwe gebruikers. Trends worden meestal zichtbaar bij het interviewen van ongeveer 10 gebruikers. Ook hier geldt dat als er meerdere doelgroepen zijn, er ook meer gebruikers geïnterviewd moeten worden omdat elke groep zijn eigen trends kan hebben.

Als de gebruikersgroep om een of andere reden niet bereikbaar is, dan kan een combinatie van de onderstaande technieken een uitkomst bieden.

- Interview stakeholders die vaak contact hebben met gebruikers.
- Review marktonderzoek en interview de marketing experts van de organisatie.
- Maak een enquête en analyseer de kwantitatieve onderzoeksdata (vooral handig om demografische kenmerken en ervaringsniveaus in kaart te brengen).
- Interview mensen die representatief zijn voor de gebruikers en die wel toegankelijk zijn. Denk hierbij aan mensen uit de kennissenkring die de applicatie zouden kunnen gebruiken.

### Stap 2: voer het onderzoek uit

Nu een keuze gemaakt is voor het type onderzoek kan het opgezet en uitgevoerd worden. Zorg voor een goede voorbereiding en structuur van de interviews zodat de gegevens van de verschillende interviews vergelijkbaar is. Neem ongeveer een uur tot anderhalf uur om een persoon te interviewen. Wees waakzaam voor meningen van de geïnterviewden. De gebruikers zijn geen experts en de waarde van hun mening is daarom niet vast te stellen. Het gaat om hun gedrag. De volgende tabel geeft aandachtspunten voor de interviews:

Informatie om te verzamelen bij interviews t.b.v. persona's	
Intranet	website
Demografische gegevens als leeftijd, werk, lengte in tijd van de baan en tijd bij de organisatie.	Demografische gegevens als leeftijd, werk, familie, hobby's en interesses.
Verantwoordelijkheden in het werk en hoe een typische dag verloopt.	Hoe verloopt een typische dag.
Taken die heel lang duren om ze uit te voeren, kritiek zijn of heel vaak worden uitgevoerd.	Gangbare vragen of taken met betrekking tot het domein van de website.
Grote frustraties.	Grote frustraties bij het behalen van doelen met betrekking tot de website of het domein van de website.
Wat vinden gebruikers leuk aan hun werk.	Grote punten van tevredenheid m.b.t. de website of het domein.
Met welke afdelingen of individuen heeft de persoon contact binnen de organisatie.	Met wie heeft de persoon het meeste contact bij het vervolbrengen van zijn taak (bijv. reisagent, partner, huisgenoten enz...)
Wat is het ervaringsniveau van de geïnterviewde?	Vaardigheidsniveaus m.b.t. taken en technologie.
Doelen, houdingen en overtuigingen	Doelen, houdingen en overtuigingen.

### Stap 3: Analyseer de onderzoeksdata en identificeer een set persona's

Review de onderzoeksdata en zoek naar patronen in houdingen en gedrag. Als het bijvoorbeeld een website voor een reisorganisatie betreft kan er een groep mensen zijn die duurdere reizen zoeken of langere afstanden willen reizen. Cluster de patronen zo veel mogelijk. Als de clusters geïdentificeerd zijn, geef ze dan een naam en een korte beschrijving. Dit vormt de basis voor de verschillende persona's. Beperk het aantal tot maximaal vijf. Minder mag ook. Een groter aantal persona's is moeilijk werkbaar.

### Stap 4: Schrijf de persona's

Vul de korte beschrijvingen aan met concrete voorbeelden uit de onderzoeksdata. Schrijf de persona's alsof je een persoonsbeschrijving maakt. Laat de persoon 'leven'. Geef elke persona bijvoorbeeld een naam en een foto.



## 7 HULPMIDDELEN

Dit hoofdstuk beschrijft drie vormen van hulpmiddelen die kunnen worden ingezet in het usabilitytesttraject. Als eerste wordt aandacht besteed aan checklists voor het snel scannen van testobjecten. Vervolgens gaat paragraaf 8.2 in op de infrastructuur die nodig is om bepaalde tests uit te voeren. Tot slot komt in paragraaf 8.3 een overzicht van softwarematige tools die kunnen worden ingezet bij het testen.

### 7.1 Checklists

Checklists kunnen helpen bij het snel scannen van testobjecten op gebruikersvriendelijkheid. Een checklist bevat een lijst met zeer concrete punten waaraan bijvoorbeeld een website zou moeten voldoen in termen van usability.

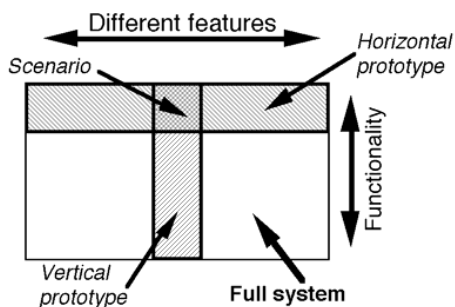
Op het internet is een ruime verzameling checklists in omloop. Ze zijn altijd specifiek voor een bepaald doel ontwikkeld. Zo zijn er checklists voor een website, een menustructuur, navigatieonderdelen enzovoorts. Het is ook goed mogelijk om voor een specifieke test zelf een checklist samen te stellen uit andere beschikbare checklists. In bijlage B staat een breed toepasbare checklist en een lijst met links naar nuttige websites met verschillende soorten checklists.

### 7.2 Infrastructuur

Deze paragraaf beschrijft de mogelijkheden die omgevingen bieden om usability testen zoveel mogelijk te faciliteren. In de eerste paragraaf beschrijven wij hoe prototype omgevingen ingezet kunnen worden om usability al in vroege stadia van het ontwikkelproces te testen. In de tweede paragraaf beschrijven wij de mogelijkheden die een usability laboratorium kunnen bieden.

#### 7.2.1 Prototypen

In de praktijk vindt het testen van usability vaak plaats op de acceptatie omgeving als onderdeel van de GAT. Hoewel de acceptatie omgeving daar zeer geschikt voor is, kan ook in vroegere stadia van het ontwikkelproces al veel werk verricht worden op prototypen. Dit wordt optimaal ondersteund in een iteratief ontwikkelproces, maar kan ook in traditionele watervalprocessen ingezet worden. Er zijn grofweg 3 typen prototypen te onderscheiden, zoals te zien in onderstaande figuur:



- Horizontale prototypen: bevat het hele systeem, maar met gereduceerde functionaliteit.
- Verticale prototypen: bevatten alle functionaliteit van een subset van een deel van het systeem.
- Scenario's: bevatten gereduceerde functionaliteit van een deel van het systeem.

Horizontale prototypes bieden de mogelijkheid om de interface van een applicatie als geheel te bekijken, voordat de functionaliteit tot in detail is doorontwikkeld. Daarom worden deze vooral ingezet bij het testen van applicaties waar navigatie en zoekgedrag een grote rol spelen. Deze vorm is bijzonder geschikt voor het uitvoeren van usability tests in een vroegtijdig stadium.

Vertical prototypes bevatten alle functionaliteit van een deel van het systeem. Deze worden veel ingezet op projecten waar deelproducten achter elkaar afgerond en gefaseerd geïmplementeerd worden. Een vertical prototype biedt niet de mogelijkheid om usability te testen voordat alle functionaliteit is ontwikkeld voor dat deel. Toch kan usability testen op een vertical prototype voordeel bieden: het kan inzichten bieden die toegepast kunnen worden bij latere deelproducten om de usability van het hele systeem te verbeteren.

Scenario's bevatten beperkte functionaliteit van een deel van het systeem. Deze zijn uitermate geschikt om in een vroeg stadium interactievormen uit te testen. Op die manier kan ruim voor de ontwikkeling van het hele systeem is afgerond al getest worden op usability, zodat ontwerpfouten in de interface hersteld kunnen worden tegen relatief lage kosten.

Paper prototypes zijn een specifieke toepassing van scenario's, en worden doorgaans al tijdens de ontwerpfase ingezet. Deze prototypes hebben als voordeel dat usability testing uitgevoerd kan worden nog voordat er één regel code is geprogrammeerd. Hierdoor kunnen in een vroeg stadium ontwerpbevindingen gevonden en opgelost worden zonder ontwikkelingspanning, waar dat later in het ontwikkelproces veel moeite had gekost.

## 7.2.2 Usability Laboratorium

Om usability testen gecontroleerd uit te kunnen voeren, worden vaak usability laboratoria gebruikt. Een laboratorium kan heel eenvoudig van opzet zijn, bijvoorbeeld een kantoor met een werkplek en een extra stoel voor de observator, of heel uitgebreid, met een spiegelwand, camera's, microfoons, etc.

Deze paragraaf geeft een overzicht van alle hardware tools waarmee een laboratorium voor usability testen of onderzoek ingericht kan worden. In de volgende paragraaf volgen nog enkele software tools voor het verzamelen en analyseren van gegevens in een laboratorium.

### *Observatiekamer*

Een usability lab is meestal uitgerust met een observatie kamer. Het is een 'geheime' kamer naast de kamer waar de proefpersoon zit. De twee kamers zijn gescheiden door een spiegelwand: de observeerder kan de proefpersoon zien, andersom niet. De observatiekamer is voorzien van allerlei apparatuur.

Zo kan de observeerder de controle van de test pc overnemen en heeft hij controle over alle camera's die in de test ruimte hangen. Ook kijkt hij mee op het scherm van de proefpersoon, zodat exact het gedrag van de proefpersoon op de testapplicatie gevolgd kan worden. Verder is er een microfoon aanwezig zodat er met de proefpersoon gepraat kan worden. Als de observeerder ziet dat het mis gaat, kan hij dus de proefpersoon corrigeren. Meestal is er ook een groot TV scherm in de observatiekamer aanwezig zodat er meer mensen mee kunnen kijken wat de proefpersoon aan het doen is. Bovenstaande foto is genomen in het usability lab van Noldus.

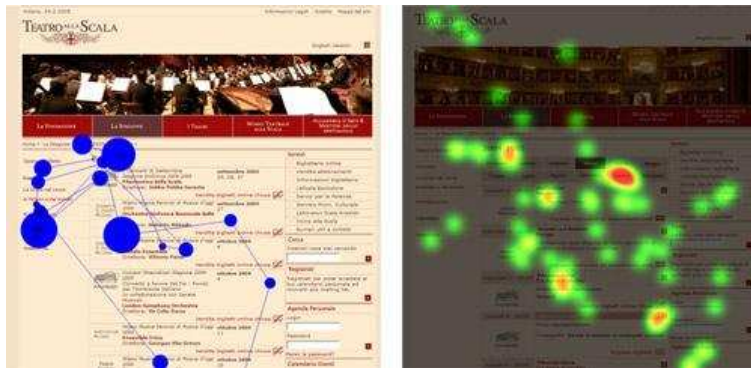


### 7.2.3 Eyetracking

Eyetracking tools zijn tools die de ogen van een proefpersoon in detail kunnen volgen. Op bovenstaande foto zit een proefpersoon achter een beeldscherm, dat uitgerust is met eyetracking. In de monitor zitten verschillende sensoren die de ogen in de gaten houden. Op deze manier worden alle oogbewegingen geregistreerd en wanneer de ogen op een bepaald punt gefocused zijn, wordt dat dus ook geregistreerd.



Als deze gegevens kunnen later door de software geanalyseerd worden. Figuur 6 laat zien hoe de verschillende analyse figuren eruit kunnen zien. De lijnen zijn de wegen die de ogen hebben bewandeld en de dikke stippen zijn de punten waarop gefocused is (op de linkerfoto). Vervolgens wordt er door sommige eyetracking tools een soort samenvatting gemaakt van de 'hotspots': de punten waar de meeste proefpersonen op gefocused waren. Deze hotspots zijn de gekleurde stippen op het scherm: groen (niet zo intensief) -> rood (intensief). Op deze manier kan er bekeken worden welke elementen op bijvoorbeeld een website veel aandacht krijgen en welke niet.



### 7.2.4 Mobile Device Camera

De laatste tien jaar heeft eigenlijk iedereen een mobiele telefoon. Ook worden de functionaliteiten die op de mobiele telefoon zitten, steeds uitgebreider. Zo kon je op de eerste mobiele telefoon alleen telefoneren, tegenwoordig kun je internetten, foto's maken etc. Omdat het display op de mobiele telefoon erg beperkt is, is het natuurlijk een zeer interessant testobject voor usability. Ook omdat de mobiele telefoon veel verschillende gebruikers heeft, van kinderen tot ouderen en van digibeten tot IT'ers. Noldus heeft een mobile device camera ontwikkeld die op een mobiele telefoon bevestigd kan worden en precies gevolgd kan worden wat de gebruiker doet. De camera kan gericht worden op het toetsenbord van de telefoon of op het beeldscherm van de telefoon. Ook is het mogelijk om vanaf maximaal 25 meter de videobeelden te volgen. Nadeel is wel dat het een behoorlijk groot apparaat is wat op de telefoon is bevestigd en de vraag is of de proefpersoon neutraal zijn test kan uitvoeren zonder last te hebben van de device camera.



### 7.2.5 Spectacles Camera

De spectacles camera is ook een dergelijk product van Noldus om de echte gebruikerservaring op te nemen op video. Zoals op bovenstaande figuur te zien is, heeft de proefpersoon een redelijk normale bril op. De bril is voorzien van een camera en op deze manier kan er precies opgenomen worden waar een proefpersoon naar kijkt. Dit kan gebruikt worden als men wil weten waar een proefpersoon allemaal naar kijkt tijdens een autorit of waar hij naar kijkt, wanneer hij aan het winkelen is door een winkelcentrum. Voordeel is dat de proefpersoon niet heel veel last heeft van de bril, omdat deze redelijk licht is. Nadeel is dat de camera in de bril zit, en dus de bewegingen van het hoofd volgt en niet de exacte bewegingen van de ogen. Op deze manier is het niet altijd even betrouwbaar. Duidelijke instructies kunnen er echter voor zorgen dat de resultaten wel betrouwbaar worden.



### 7.2.6 Domecamera

Tenslotte staan we even stil bij de domecamera. Deze camera is niet weg te denken uit een gemiddeld usability lab. Het voordeel van de domecamera is dat hij op afstand te besturen is. Op deze manier kan hij 360 graden ronddraaien en goed in –en uitzoomen. Hierdoor is het mogelijk om de camera te richten op een bepaald object dat belangrijk is voor de test. Is bijvoorbeeld het gezicht van de proefpersoon belangrijk, kan de camera precies gericht worden op het gezicht.



## 7.3 Tools

Voor het uitvoeren van de diverse test- en toetstechnieken die in de voorgaande hoofdstukken zijn genoemd zijn diverse al dan niet geautomatiseerde tools beschikbaar. Zo zijn er tools waarmee routinematige checks uitgevoerd kunnen worden, tools die het vastleggen van data tijdens observatietests kunnen vergemakkelijken, en tools die statistische gegevens over een usability test kunnen vergaren en analyseren. Voor de methode om tools in te voeren verwijzen wij naar TMap NEXT® paragraaf 8.5.

### 7.3.1 Log data analyse tools

Deze tools zijn specifiek gebouwd om de technieken “AB testing” en “logging actual use” te ondersteunen. Deze programma’s draaien op de achtergrond, terwijl een gebruiker testhandelingen uitvoert met het testobject. De tool registreert gegevens over het gedrag van de gebruiker, en kan na afloop van de test ondersteuning bieden bij het analyseren van deze resultaten.

#### *Google Website Optimizer*

Google Website Optimizer is één van de meeste gebruikte website statistieken pakketten op internet. Het is een gratis online multivariate testing tool, gebaseerd op de Google Analytics software. Met deze tool kunnen eenvoudig verschillende A/B testen worden opgezet.

Elke test begint bij een test site, en heeft als succesvol eindpunt de conversiesite. Op de testsite worden variaties aangebracht, die voor een bepaald deel van de bezoekers te zien zullen zijn. Website optimizer houdt dan tijdens de test bij wat de conversieratio’s zijn van de verschillende varianten, zodat de gebruiker na de test kan zien welke variant het meest succesvol is geweest.

#### *Lyris Hq Web Analytics*

Deze tool, voorheen Clicktracks geheten maar tegenwoordig onderdeel van marketingplatform Lyris HQ, is toegespitst op de analyse van (klik) gedrag van gebruikers van e-business toepassingen. Het wordt vooral gebruikt bij de analyse van effectiviteit van websites en advertentiecampagnes om omzet te genereren. Deze tool biedt o.a. de mogelijkheid om bezoekers te groeperen, ROI te bepalen van advertentiecampagnes en zoekmachine gebruik te onderzoeken. Zo kunnen veel diepgaandere uitspraken gedaan worden over het gedrag van gebruikers dan met traditionele datalogging tools.

#### *Nist Web Metrics Flud*

Het Framework for Logging Usability Data (FLUD) is een onderdeel van NIST Web Metrics, een verzameling tools voor het geautomatiseerd testen van websites. Deze tool logt acties en events die plaatsvinden terwijl een gebruiker testhandelingen uitvoert. Met behulp van visualisatie tools, die ook in NIST Web Metrics zitten, kan ook het pad dat de gebruiker door een website aflegt grafisch weergegeven worden. NIST Web Metrics is gratis down te loaden, maar is niet de makkelijkste in gebruik. Bovendien lijkt NIST al enige tijd geen nieuwe versies meer te hebben uitgegeven.

### 7.3.2 Automated inspection tools

Deze tools controleren geheel automatisch een webpagina op potentiële usability en/of accessibility problemen. Hoewel deze tools volautomatisch uitgevoerd kunnen worden, moet uiteraard wel zorgvuldig omgegaan worden met de parametrisering vooraf, en de analyse van de resultaten achteraf. Houd er rekening mee dat dergelijke tools niet automatisch kunnen bepalen of een website usable dan wel accessible is, ze sporen slechts (mogelijke) bevindingen op!

#### *Rational Policy Tester*

Policy Tester is een onderdeel van IBM’s Rational productgroep. Deze bevat o.a. een Quality Edition, voorheen WebXM geheten, welke usability issues opspoor, en een Accessibility edition, voorheen Bobby. De Quality Edition doet in feite een review van webpagina’s: het zoekt naar o.a. spelfouten, broken links, navigatieproblemen,

interactieproblemen, performance, zoekmachine ineffectiviteit, en andere worst practices. De accessibility edition kan valideren of een website voldoet aan een aantal accessibility standaarden, waaronder de W3C Web content accessibility guidelines en de Amerikaanse overheidsstandaard Section 508.

### **WAVE**

WAVE is een gratis te gebruiken online accessibility tool, die de HTML code van een webpagina controleert op vooraf gedefinieerde accessibility richtlijnen. Wanneer op de website een URL ingetikt wordt, verschijnt na enige tijd een rapport waarmee met iconen fouten en waarschuwingen aangegeven worden. WAVE heeft een eigen set richtlijnen, maar heeft daarbij wel richtlijnen overgenomen van Section 508 en W3C. WAVE pretendeert echter niet te kunnen controleren of aan deze richtlijnen wordt voldaan.

### **Nist Web Metrics Websat**

De Web Static Analyzer Tool (WebSAT) is net als FLUD een onderdeel van NIST Web Metrics. Deze tool controleert de HTML code van een webpagina, of een hele site, aan een set usability richtlijnen. Hiervoor zijn door NIST een eigen set, inclusief accessibility richtlijnen, gemaakt, maar ook een richtlijn van het IEEE is ingebouwd. Deze gratis tool is helaas al enige tijd niet bijgewerkt: met name de performance gerelateerde richtlijnen kunnen achterhaald zijn.

## **7.3.3 Development tools**

Deze tools kunnen gebruikt worden tijdens ontwerp en ontwikkeling van user interfaces. Ze kunnen vaak gebruikt worden in combinatie met de x technieken uit hoofdstuk 6. In sommige gevallen zijn ze ook inzetbaar voor usability testen.

### **Hiser Element Toolkit**

Deze verzameling tools is gebaseerd op de gelijknamige user-centered design methode. Het bevat tools voor de ondersteuning van het ontwerpproces, card sorting en paper prototypes. Daarnaast worden diverse testtechnieken ondersteund, zoals heuristic evaluation en scenario based testing. De toolkit is zo opgesteld dat ook ontwerpers zonder usability ermee uit de voeten moeten kunnen. Veel nadruk wordt gelegd op de ondersteuning, met checklists, templates, tool advisering, instructies en tips.

### **Optimal Workshop**

Deze verzameling tools specifiek gericht op het verbeteren van de usability van websites. Optimal Workshop bestaat uit een tool voor Card Sorting, een tool voor Reversed Card sorting en een tool voor prototype testing. Met deze tools kunnen deze technieken online uitgevoerd worden. De tools zijn individueel en samen te bestellen en gratis uit te proberen.

### **Nist Web Metrics Webcat**

Web Category Analysis Tool (WebCAT), wederom onderdeel van de NIST Web Metrics, biedt een geautomatiseerde variant van de Card Sorting techniek. Een ontwerper maakt eerste een experiment, met namen, een categorisatie methode, instructies e.d. en zet deze online als JAVA applet. Gebruikers kunnen dan met een "drag and drop" interface een structuur maken. Resultaten kunnen op elk moment afzonderlijk of geclusterd weergegeven worden.

## **7.3.4 Usability laboratorium tools**

Deze software tools zijn speciaal ontwikkeld voor gebruik in een laboratorium. Vaak is het hier nodig om grote hoeveelheden data afkomstig van verschillende bronnen te verzamelen en te analyseren. Deze tools zijn erop gericht dat proces te faciliteren.

### **The Observer**

Dit pakket van Noldus, uit Wageningen, is oorspronkelijk opgezet om diergedragsonderzoek te ondersteunen. Vandaar dat de scope van het pakket breder is dan alleen usability. Het pakket kan data verzamelen van een grote hoeveelheid bronnen, zoals video, audio, eye-tracking, fysiologie etc. Deze data kan met het pakket

gecodeerd, gevisualiseerd en geanalyseerd worden. Noldus gebruikt het pakket als integratiecomponent voor al hun laboratoria. Ze verkopen het pakket ook los, maar bieden ook aan om complete laboratoria te installeren, of om een van hun eigen laboratoria af te huren.

### *Morae*

Dit pakket van Techsmith, oorspronkelijk begonnen met screen capture tools, is de grote concurrent van Noldus op het gebied van usability labtools. Zij richten zich specifiek op usability testen, en ondersteunen expliciet prototype testing en focus groups. Dit pakket verzamelt ook data van verschillende bronnen, en stelt de gebruiker in staat deze te coderen, analyseren, annoteren en uit te zetten in grafieken. Morae lijkt de nadruk gelegd te hebben op het analyseren van data, waar The Observer meer gericht lijkt op de vele mogelijkheden om data te verzamelen.

### *Datalogger*

Dit is een gratis te downloaden Excel-gebaseerde tool om usability testen mee voor te bereiden en gegevens mee te verzamelen en te analyseren. De tool biedt ruimte om direct opmerkingen te plaatsen, en om uitkomsten van interviews aan gegevens van een tester toe te voegen en na afloop te analyseren. Ook kunnen automatisch grafieken gegenereerd worden. Het grote verschil in functionaliteit met de professionele labtools, is dat alle input direct van de onderzoeker moet komen; gegevens worden niet automatisch verzameld.

## **7.3.5 Reverse card sorting tools**

Specifiek voor de toetstechniek reverse card sorting zijn een aantal tools ontwikkeld. De tool Treejack is opgenomen in de eerder genoemde Optimal Workshop. Het idee achter deze tools is vergelijkbaar en wordt hieronder uitgelegd aan de hand van C-Inspector.

### *C-Inspector*

Met deze service kan een reverse card sorting online uitgevoerd worden. Eerst moet de informatie architectuur geupload worden, en moeten de taken en antwoorden gedefinieerd worden. Daarna kunnen alle deelnemers een link toegestuurd krijgen om de taken uit te voeren. Alle acties worden bijgehouden, zodat na de test een grondige analyse uitgevoerd kan worden. C-Inspector richt zich vooral op intranets en informatieve websites en is 30 dagen gratis te proberen in een beperkte versie.

## 8 USABILITY MATURITY MODEL

Dit hoofdstuk richt zich op het volwassen worden van usability binnen een organisatie. Hiermee wordt bedoeld de mate waarin een organisatie op een weloverwogen manier omgaat met de borging van het kwaliteitsattribuut usability. Om het belang hiervan te illustreren is een beeld geschetst van hoe in de praktijk veelal omgegaan wordt met usability, en waarom verbetering gewenst is.

### 8.1 Het Usability Maturity Model

#### 8.1.1 *Stand van zaken*

De tijd dat usability over het algemeen een non-issue was, ligt inmiddels achter ons. Met name de opkomst van e-commerce heeft ertoe geleid dat en algemeen belang van usability vaak bekend is. Een concrete invulling geven aan usability is helaas vaak nog een stap te ver. Dit kan zich bijvoorbeeld als volgt uiten:

- Er is één algemeen requirement dat een product usable moet zijn, zonder dat bekend is hoe dat bepaald gaat worden. Hierdoor wordt het requirement ontoetsbaar.
- Gebruikers worden pas betrokken wanneer een product opgeleverd wordt, terwijl gebruikers al vanaf de ontwerpfase en in de ontwikkelfase belangrijke input kunnen leveren.
- Als al gestructureerde usability testen gedaan worden, worden de resultaten vaak niet hergebruikt tijdens andere projecten.
- Usability wordt pas getoetst tijdens de Gebruikers Acceptatie Test. Omdat het ontwerp meestal ten grondslag ligt aan usability problemen, vallen de kosten relatief hoger uit voor het oplossen van de gevonden problemen.
- Wanneer een usability probleem gevonden wordt, wordt niet berekend wat het rendement van de investering (ROI) van het oplossen van het probleem is. Hierdoor kunnen problemen als “cosmetisch” beoordeeld worden, terwijl ze na in gebruik name dagelijks negatieve gevolgen kunnen hebben voor de productiviteit en de omzet.

#### 8.1.2 *Verbetering*

De huidige situatie, waarin usability niet gestructureerd geborgd wordt, lijkt met name veroorzaakt te worden door een gebrek aan kennis over usability. Tegelijkertijd zorgt deze ongestructureerde aanpak ervoor dat er geen kennis en ervaring opgedaan wordt tijdens projecten. Om deze spiraal te verbreken, is een model nodig waarmee een gestructureerde aanpak gekozen kan worden om verbetering te brengen.

Procesverbetering bestaat over het algemeen uit een vast aantal stappen:

- Bepalen doel en beschouwinggebied
- Vaststellen huidige situatie
- Vaststellen gewenste situatie
- Implementeren veranderingen
- Beoordelen veranderingen

Een gestructureerd en praktisch model voor verbetering van de usability volwassenheid moet dus al deze stappen kunnen faciliteren.

#### 8.1.3 *Verbetermodel*

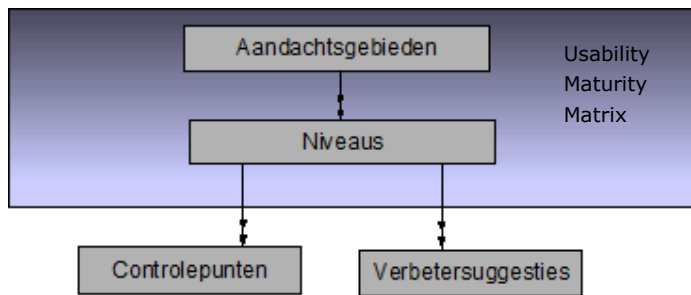
Usability-expert Jakob Nielsen beschreef 8 niveaus, waarin een organisatie zich kan bevinden op het gebied van usability volwassenheid. Deze beschrijving is zeer nuttig bij het bepalen van het doel, en kan een leidraad zijn bij het vaststellen van de huidige en de gewenste situatie. Het ontbreekt helaas aan de nodige handvatten om de huidige situatie objectief vast te stellen en veranderingen te implementeren en beoordelen. Om deze reden zijn er elementen toegevoegd die ook in het TPI® NEXT model voor testprocesverbetering beschreven worden:



aandachtsgebieden, controlepunten en verbeteracties. Tenslotte zijn ook valkuilen bij het implementeren van veranderingen beschreven. Op deze manier is het Usability Maturity Model (UMM) tot stand gekomen. In de rest van dit hoofdstuk is het UMM uiteen gezet, zonder al te diep in te gaan op de algemene details van het uitvoeren van een veranderproces. Deze details worden uitstekend beschreven in het boek TPI® NEXT, Business Driven Test Process Improvement.

### 8.1.4 Usability Maturity Model

Het UMM omvat 3 niveaus: bewust, systematisch en optimaliserend. Deze niveaus worden gedefinieerd door 10 aandachtsgebieden. Op welk niveau een aandachtsgebied zich bevindt, kan beoordeeld worden door de controlepunten voor het niveau, en voor alle onderliggende niveaus, te toetsen. Wanneer een bepaald niveau als doel gesteld is, geven de verbeteracties aanwijzingen om dit niveau te halen. De valkuilen geven een overzicht weer van veel voorkomende fouten waar men in de praktijk mee te maken kan krijgen in het verbeterproces.



**Usability Maturity Model**

De verschillende aandachtsgebieden zijn afhankelijk van elkaar. Wanneer het niveau van een ander aandachtsgebied achterloopt, kan het lastig of zelfs onmogelijk worden om een niveau te halen. Deze afhankelijkheden zijn voor een deel verwerkt in de verschillende niveaus waarin een organisatie zich kan bevinden. Dit wordt geïllustreerd in de Usability Maturity Matrix. Aangeraden wordt om de organisatie met één niveau tegelijk te verbeteren, zodat verbeteracties niet gehinderd kunnen worden door achterblijvende aandachtsgebieden. Daarnaast zal de organisatie beter omgaan met de nodige veranderingen, wanneer deze stap voor stap geïmplementeerd worden.

### 8.1.5 Onderverdeling

De drie niveaus van usability volwassenheid kennen ieder een geheel eigen invalshoek:

#### **Bewust**

Dit niveau is primair gericht op de bewustwording van de organisatie met het belang van de gebruiker. Binnen een organisatie worden concrete activiteiten uitgevoerd om usability bij de uitvoering van een project te waarborgen en te testen. Het doel is dat usability een vaste plek krijgt binnen de organisatie.

#### **Systematisch**

Dit niveau richt op een systematische invulling van het usability proces. Op organisatiebreed niveau loopt een proces waarmee de usability van producten gewaarborgd wordt, bijvoorbeeld door designstandaards te introduceren, testresultaten te centraliseren en door samen met gebruikers iteratief te ontwikkelen.

#### **Optimaliserend**

Dit niveau is erop gericht om het usability proces binnen de gehele organisatie continu te optimaliseren. Usability wordt niet alleen een kwantitatief gemeten key performance indicator (KPI) waarop alle projecten worden beoordeeld, maar ook een doel van de organisatie als geheel. De gebruikerservaring staat centraal binnen de organisatie en al haar interactievormen.

### 8.1.6 Aandachtsgebieden

Het UMM wijst 10 aandachtsgebieden aan als pijlers onder een gestructureerd usability proces. Deze aandachtsgebieden bestrijken het gehele usability proces en zijn afgeleid van de beschrijving van Nielsen, de aandachtsgebieden van TPI, en vanuit de best practices.

Aandachtsgebied	Omschrijving
Toepassing testfasering	De fasering voor usability tests verschilt niet van de algemene TMap fasering. Voor een effectief usability proces is het namelijk van belang niet alleen de hoofdfasen toe te passen, maar ook een gedegen voorbereiding en met name een afronding waarbij resultaten opgeslagen worden zodat deze herbruikt kunnen worden bij andere projecten. Door het late moment van betrokkenheid waar veel usability testers mee te maken krijgen blijken juist deze fasen vaak niet uitvoerbaar.
Moment van betrokkenheid	Net als bij andere testvormen levert het vinden van usability bevindingen in een vroeg stadium van het ontwikkelproces veel winst op. Juist in usability tests worden veel bevindingen op het ontwerp gedaan. Toch worden eindgebruikers vaak pas ingeschakeld wanneer er een werkend product is. Effectiever is het om usability al tijdens het ontwerp te behandelen, of om de usability van een ontwerp te borgen middels prototyping
Begroting en planning	Aandacht voor usability kan pas echt structureel plaatsvinden als usability activiteiten begroot en gepland worden. Aangegeven moet worden wanneer welke activiteiten uitgevoerd moeten worden door welk aantal mensen.
Metrieken	Omdat in een gestructureerd usability proces met name gekeken wordt naar gedragingen van eindgebruikers, is het van belang metrieken over de resultaten van usability testen te vergaren en op te slaan. Wanneer deze informatie over verschillende producten of projecten vergaard wordt, kan deze informatie hergebruikt worden voor nieuwe projecten, bijvoorbeeld door richtlijnen of standaarden te ontwikkelen.
Usability onderzoek	Gerelateerd aan usability testen is usability onderzoek, van het vergaren van kwalitatieve gegevens over een product (meningen, ervaringen, anekdotes) tot onderzoek over verschillende producten en projecten heen, tot kwantitatief (statistisch) onderzoek.
Usability tools	Tools zijn op zichzelf geen voorwaarde voor volwassenheid. Het getuigt echter wel van volwassenheid om tools waar nodig toe te kunnen passen.
Usability werkplek	Ook voor usability testers en consultants zijn geschikte werkplekken van groot belang. Goed bruikbare werkplekken en nabijheid van collega's kunnen direct invloed hebben op de motivatie en prestaties. Daarnaast kunnen bepaalde voorzieningen, zoals een laboratorium, meer mogelijkheden scheppen voor usability tests en onderzoek.
Stakeholder commitment	Commitment van de stakeholders is cruciaal voor het inrichten van een efficiënt en gestructureerd usability proces. Zonder commitment van lijn- of projectmanagement blijven usability initiatieven steken op de lagere schalen, en is groei naar een efficiënt en gestructureerd usability proces onmogelijk.
Usability awareness	Waar commitment zich met name richt op management, is usability awareness een factor die organisatiebreed invloed heeft. Als ontwerpers en ontwikkelaars geen inzicht hebben in het belang van de eindgebruiker, dan is usability testen een druppel op een gloeiende plaat. In een efficiënt proces moet het besef groeien voor de invloed die usability heeft op de ROI van een product, project of zelfs op de bedrijfsstrategie.
Usability eisen	In een gestructureerd usability proces worden specifieke eisen gesteld aan de usability. Daardoor kan een helder en onderbouwd oordeel gevormd worden van het niveau van de usability van een product, en kan bepaald worden hoe dit niveau te verhogen. Daarnaast kunnen deze eisen als input dienen voor ontwikkelaars en ontwerpers, om zo usability bevindingen te voorkomen.

### 8.1.7 Controlepunten

Elk aandachtsgebied heeft 2 of 3 niveaus. Elk niveau is onderverdeeld in 1 tot 4 controlepunten. Om een niveau te verkrijgen, is het noodzakelijk dat niet alleen aan de controlepunten voor dat niveau voldaan wordt, maar ook aan de controlepunten voor alle onderliggende niveaus. Deze controlepunten kunnen betrekking hebben op een project, op de mogelijkheden binnen de organisatie, of op de manier waarop binnen de organisatie gewerkt wordt.

Wanneer bijvoorbeeld het aandachtsgebied usability werkplek op optimaliserend staat, betekent dit dat indien nodig een project de beschikking kan krijgen over een laboratorium. Dat hoeft niet te betekenen dat voor elk project een laboratorium gebruikt moet worden, of dat de organisatie zelf een laboratorium ingericht moet hebben. Laboratoria kunnen immers ook gehuurd worden.

<b>Aandachtsgebied</b>	<b>Bewust</b>	<b>Systematisch</b>	<b>Optimaliserend</b>
Toepassing testfasering	Hoofdfasen	Volledig	
Moment van betrokkenheid	Acceptatietesten	Input bij ontwerp	Initiatie project
Begroting en planning		Onderbouwd	Statistisch onderbouwd
Metrieken	Per project	Archief	Over organisatie
Usability onderzoek		Kwalitatief	Kwantitatief
Usability tools	Losse tools	Ingebed in methode	Herbruikbaar
Usability werkplek	Werkplek	Afdeling	Laboratorium
Stakeholder commitment	Budget en tijd	In projectorganisatie	In organisatie
Usability awareness	Algemeen nut	ROI	Strategie
Usability eisen		Richtlijnen	Subset van eisen

### 8.1.8 Usability Maturity Matrix

Het naar een bepaald niveau tillen van één of meer aandachtsgebieden is nooit een doel op zich. Alle verbeteractiviteiten moeten in relatie staan tot een visie op de gehele organisatie. Binnen het model zijn namelijk afhankelijkheden aanwezig tussen verschillende aandachtsgebieden. Daarnaast hebben bepaalde aandachtsgebieden bij een laag niveau minder of geen prioriteit, omdat andere aandachtsgebieden op dat moment veel effectiever zijn in het volwassen maken van het usability proces. Deze afhankelijkheden en prioriteiten zijn samengevat in de Usability Maturity Matrix. In de matrix zijn verticaal de aandachtsgebieden benoemd en horizontaal de niveaus van testvolwassenheid. Binnen de niveaus zijn de controlepunten (0 tot en met 4) weergegeven.

Aandachtsgebied	Bewust	Systematisch	Optimaliserend
Toepassing fasering			
Moment van betrokkenheid			
Begroting en planning			
Metrieken			
Gebruiker onderzoek			
Usability tools			
Usability werkplek			
Stakeholder commitment			
Usability awareness			
Usability eisen			

### 8.1.9 Gebruik van de Matrix

De Usability Maturity Matrix wordt ingevuld na de inventarisatie van het usability proces. Zo kan een goede afweging gemaakt worden voor de verbetervoorstellen. Hierbij is het de bedoeling om van links naar rechts te werken, dus de aandachtsgebieden met een laag niveau als eerste te verbeteren. Goed gemotiveerde afwijkingen zijn wel mogelijk.

Kijk niet alleen naar de matrix, maar neem ook de eigenschappen van de organisatie zelf in acht. Niet elke organisatie hoeft als doel na te streven om op het niveau optimaliserend terecht te komen. Bedenk goed welke doelen voor de organisatie relevant zijn, alvorens hiervoor een plan op te stellen.

In het onderstaande voorbeeld gebruikt de organisatie losse tools (bewust) en hebben usability medewerkers een werkplek (bewust), er is er nog geen stakeholder commitment (startniveau).

Aandachtsgebied	Bewust	Systematisch	Optimaliserend
...			
Usability Tools			
Usability werkplek			
Stakeholder commitment			

Op basis van deze matrix kan over verbeteringen worden gediscussieerd. In dit voorbeeld wordt ervoor gekozen zoveel commitment te krijgen dat usability budget en tijd, en een fulltime medewerker krijgt (2/3 bewust) en een vaste ruimte voor usability testen in te richten (1/2 systematisch). Omdat deze laatste stap gedaan wordt, terwijl de organisatie zich op dat moment nog niet op niveau bewust bevindt, is dat alleen aan te raden als deze beslissing weloverwogen genomen is. Aan het inbedden van tools in de methode wordt (voorlopig) nog niet gewerkt. Uit de matrix blijkt ook dat dit nog geen prioriteit heeft. De gewenste situatie staat weergegeven in de volgende matrix.

Aandachtsgebied	Bewust	Systematisch	Optimaliserend
...			
Usability Tools			
Usability werkplek			
Stakeholder commitment			

## 8.2 Aandachtsgebieden en niveaus

Nu de theorie achter het model is beschreven, volgt in deze paragraaf de praktische uitwerking van het model. Per aandachtsgebied worden de niveau's beschreven, met de controlepunten waarmee bepaald kan worden of de organisatie zich op dat niveau bevindt, en de verbetersuggesties als handvat bij het bereiken van het niveau. Het model probeert echter nadrukkelijk niet voor te schrijven op welke wijze de doelen behaald worden, dat zal per organisatie bekeken moeten worden.

### 8.2.1 Toepassing fasering

Net als bij alle andere testvormen, verdient het ook bij usability testen sterk de aanbeveling een fasering van de activiteiten toe te passen, zodat de benodigde effort zo min mogelijk op het kritieke pad ligt. Een inhoudelijke beschrijving van de fasen is te vinden in het boek TMap NEXT®

#### *Bewust: hoofdfases*

Op dit niveau worden tenminste de fasen Planning, beheer, specificatie en uitvoering onderkend. Het proces begint met het opstellen van een testplan, waarna testgevallen gespecificeerd worden aan de hand van de testbasis. Daarna worden deze testgevallen uitgevoerd.

#### Controlepunten:

- De acceptatietesten zijn onderverdeeld in de fasen specificatie en uitvoering. Deze worden achtereenvolgens uitgevoerd, eventueel per deelsysteem. Enige overlap tussen deze fasen is gewenst voor een efficiënt proces.
- De fase planning wordt onderkend. Deze fase markeert het begin van het testproces. In deze fase wordt het testplan geschreven.
- De fase beheer wordt onderkend. Deze fase vindt plaats tijdens het hele testproces.

#### Verbetersuggesties:

- Breng onder de aandacht dat de fasering ervoor zorgt dat het testen zo min mogelijk tijd op het kritieke pad doorbrengt. Opdrachtgevers en projectleiders zouden hier gevoelig voor moeten zijn.

#### Valkuilen:

- Deze verandering blijkt zeer lastig door te voeren middels een bottom-up benadering wanneer testers niet full-time beschikbaar zijn.

#### *Systematisch: Volledige fasering*

Op dit niveau worden naast de hoofdfasen ook de fasen voorbereiding, afronding en infrastructuur uitgevoerd. Voorafgaand aan de specificatie vindt een intake plaats op de testware, aan de hand van checklists. Na afloop van de uitvoering wordt de testware geüpdate en geconserveerd voor gebruik, en wordt het testproces geëvalueerd. Tijdens het proces wordt ook de benodigde infrastructuur beheerd.

#### Controlepunten:

- De fase afronding wordt onderkend. Deze fase volgt op de fase uitvoering.
- De fase voorbereiding wordt onderkend. Deze fase gaat vooraf aan de fase specificatie.
- De fase infrastructuur wordt onderkend. Deze fase vindt plaats tijdens het testproces.

#### Verbetersuggesties:

- Ontwikkel checklists en stimuleer het gebruik hiervan bij het beoordelen van de testbasis.

- Zorg ervoor dat een usability medewerker officieel deel uitmaakt van het algehele reviewproces binnen het project.
- Maak resultaten van de intake zichtbaar, zodat duidelijk wordt dat deze activiteit bijdraagt aan het verkrijgen van een gemeenschappelijk begrip van het eindproduct.
- Werk na het testen de testscripts bij
- Zorg ervoor dat de beheerorganisatie de testscripts kan herbruiken, bijvoorbeeld voor latere releases.
- Bewaar rapportages op een centrale plek.
- Stel een testinfrastructuurcoördinator aan, die de verantwoordelijkheid heeft de oplevering van de infrastructuur te monitoren, en om de infrastructuur te beheren.

#### Valkuilen:

- Hoewel het uitvoeren van intakes zonder checklists nuttige resultaten kan bieden, betekent het vaak wel dat tijdens de specificatie toch nog onduidelijkheden naar boven komen.
- Wanneer checklists gebruikt worden zonder te kijken naar de context van een project, kunnen onterechte bevindingen gedaan worden. Wanneer dit vaak gebeurt, kan dit de acceptatie van usability medewerkers als reviewpartij schaden.
- Testware bewaren is niet voldoende. De testware moet ook in de toekomst gebruikt (kunnen) worden door beheer. Houdt hierover contact met de beheerorganisatie.
- Wanneer geen testinfrastructuurcoördinator aangewezen wordt, ontstaat vaak onduidelijkheid over verantwoordelijkheden voor omgevingen en andere infrastructuurelementen.

### 8.2.2 *Moment van betrokkenheid*

Omdat usability nauw verbonden is met (interface) ontwerp, speelt dit aandachtspunt een cruciale rol in ons model. Immers, bevindingen op het ontwerp zijn, wanneer gevonden na de ontwikkeling, extra kostbaar om te herstellen. Veel voorkomend is de situatie waarin usability testen pas aan het einde van de acceptatietesten nog even uitgevoerd worden. Een gevolg hiervan is dat vaak alleen de meest eenvoudig op te lossen bevindingen aangepakt kunnen worden binnen de planning en het budget. Om de usability op een systematische manier te kunnen borgen, is het nodig om in de ontwerpfase al te beginnen met usability, bij voorkeur middels prototyping.

#### ***Bewust: Acceptatietesten***

Op dit niveau worden tijdens de acceptatietesten gebruikers ingeschakeld om een oordeel te geven over de gebruiksvriendelijkheid van het product. Dit kan nuttige inzichten geven, zowel in fundamentele fouten als in quick wins. Veel gebruikte technieken hierbij zijn observatie en interviews.

#### Controlepunten

- Voor de acceptatietesten wordt een usability test uitgevoerd met eindgebruikers.
- De bevindingen die uit deze test voorkomen worden volgens de gebruikelijke bevindingenprocedure behandeld.

#### Verbetersuggesties:

- Wanneer het aandachtsgebied usability awareness op niveau A zit, zou het niet heel lastig moeten zijn om dit niveau te halen. Belangrijk is om te benadrukken dat usability testbaar is, en testen vrij makkelijk uitgevoerd kunnen worden. Vaak wordt bij usability gedacht aan laboratoria met camera's voor eye-tracking, terwijl veel goedkopere alternatieven mogelijk zijn.
- Zorg ervoor dat resultaten niet in de spreekwoordelijke la verdwijnen, maar zoveel mogelijk opgepakt worden.
- Start een pilotproject. Belangrijk is dat aanwijsbaar resultaat getoond kan worden, bijvoorbeeld dat gebruikers tevredener zijn over het eindproduct, of sneller handelingen uit kunnen voeren na het oplossen van bevindingen.

#### Valkuilen

- Op dit niveau wordt ook vaak al een heuristic evaluation uitgevoerd, meestal door een expert. Zijn oordeel kan echter afwijken van dat van de gebruiker. Gebruik deze techniek niet op zichzelf, behalve wanneer de doelgroep zeer breed is. (denk aan algemene webapplicaties)
- Kies een pilot project zorgvuldig. Zorg ervoor dat het om een project gaat, waar usability van belang is, en het usability testen dus resultaat kan hebben.
- Pas bij het presenteren van resultaten op voor al te veel statistiek, dat zal een opdrachtgever niet altijd veel zeggen. Daarnaast zijn statistieken op dit volwassenheidsniveau vaak onbetrouwbaar.

#### *Systematisch: input bij ontwerp*

Op dit niveau is usability niet meer iets dat na ontwikkeling getest wordt, maar een factor waar tijdens de ontwerpfase al rekening mee gehouden wordt. Veelal komt deze input voort uit de kennis die is opgedaan tijdens de diverse usability testen. Het daadwerkelijke testen met eindgebruikers vindt in eerste instantie nog wel vaak plaats tijdens de acceptatietesten, maar zal steeds vaker uitgevoerd worden tijdens het ontwikkelen van prototypes, als onderdeel van het ontwerpproces. Hiervoor kunnen een aantal technieken gebruikt worden, zoals card sorting, heuristic evaluation en walkthroughs.

#### Controlepunten

- Usability medewerkers worden betrokken in de ontwerpfase.
- Ervaringen uit eerdere usability tests worden stelselmatig gebruikt als input bij het ontwerpproces.
- Usability testen met eindgebruikers worden uitgevoerd op prototypes van het eindproduct.
- De resultaten van deze testen worden gebruikt om nieuwe prototypes te ontwikkelen.

#### Verbetersuggesties

- Stel resultaten van usability tests beschikbaar aan ontwerpers.
- Maak ontwerpers bewust van het belang van een usable ontwerp, zodat ze bij hun werkzaamheden rekening gaan houden met eindgebruikers.
- Stel ontwerprichtlijnen op (of voor), op basis van eerdere testresultaten, zodat ontwerpers houvast hebben bij het ontwikkelen van usable ontwerps.
- Maak inzichtelijk dat prototyping ervoor zorgt dat usability bevindingen op het eindproduct voorkomen worden, wat veel herstelkosten bespaart.
- Zorg ervoor dat gebruikers weten dat hun input gebruikt wordt.
- Een veelgehoord argument om geen usability tests met eindgebruikers te doen, is dat gebruikers het niet op prijs zullen stellen om deelproducten te zien. Maak duidelijk en inzichtelijk dat gebruikers het op prijs zullen stellen intensief betrokken te worden bij de ontwikkeling van een applicatie, zeker wanneer zij deze regelmatig zullen gaan gebruiken.

#### Valkuilen

- Op dit niveau zullen usability testers zich moeten gaan bemoeien met het werk van de ontwerpers. Wanneer daar niet zorgvuldig mee omgegaan wordt, kan dat verkeerd overkomen bij de ontwerpers. Waak ervoor dat ontwerpers input willen ontvangen, voordat het gegeven wordt.
- Stel niet te snel richtlijnen op. Zorg ervoor dat deze gebaseerd zijn op een representatieve hoeveelheid testen.
- Prototype testing vergt nogal een omslag in de manier van werken van de ontwerpers. Het is lastig uit te voeren zolang het ontwerpproces niet iteratief is. Wel is het een goed idee om voor een iteratief ontwerpproces te pleiten, maar
- Zet de testen per iteratie niet al te groots op. Regelmatig testen met weinig gebruikers levert sneller meer resultaten dan weinig groots opgezette testen.
- Wanneer eindgebruikers vaker meewerken aan testen, laat ze dan weten wat er gebeurt met hun input, en geef hun echt het gevoel dat ze betrokken zijn bij het ontwerpproces.

### ***Optimaliserend: Initiatie project***

Op het niveau optimaliserend zal in elk project usability een belangrijke rol spelen bij de initiatie van een project. Vaak zal het niet gaan om de betrokkenheid van de eindgebruikers zelf, alswel het gebruik van de gegevens die verzameld zijn tijdens het uitvoeren van usability testen in het verleden.

#### Controlepunten

- Usability gegevens worden gebruikt als input voor de projectdefinitie en de requirements fase.
- Voor elk project worden concrete meetbare doelen gesteld op het gebied van usability.

#### Verbetersuggesties

- Om het niveau optimaliserend te bereiken moeten zowel het hele projectteam als alle betrokken managers zich ervan bewust worden dat usability een onderdeel is van hun werk, en dat onderzoek het hele project helpt om betere resultaten te bereiken.
- Wat bij het vorige niveau van toepassing is voor ontwikkelaars, is op dit niveau van toepassing voor projectmanagers. Belangrijk is dat zij open staan voor de input die vanuit de kant van usability onderzoek gegeven kan worden.

#### Valkuilen

- Probeer dit niveau niet te bereiken voordat herhaalde, onderbouwde gegevens verkregen zijn uit testen, toetsen en onderzoek. Wanneer de als input gebruikte gegevens onjuist of verkeerd geïnterpreteerd blijken te zijn, kan dit catastrofale gevolgen hebben voor het project.

## **8.2.3 Begroting en Planning**

De begroting en de planning geven aan wanneer welke activiteiten door wie uitgevoerd moeten worden en hoeveel capaciteit daarvoor gereserveerd moet worden. Dit aandachtspunt verschilt in principe weinig van het gelijknamige aandachtspunt uit TPI, echter in de praktijk blijkt de onderbouwing vaak extra lastig voor usability testen.

### ***Systematisch: Onderbouwde begroting en planning***

Om een gestructureerd usability testproces te verkrijgen, is het noodzakelijk dat de gemaakte plannings en begrotingen goed onderbouwd zijn, bijvoorbeeld aan de hand van ervaringen uit het verleden, begrotingstechnieken en de overkoepelende teststrategie.

#### Controlepunten

- De begroting en planning kunnen worden onderbouwd.
- Er vindt bewaking plaats van de begroting en de planning, met zo nodig bijsturing.

#### Verbetersuggesties

- Hanteer meerdere begrotingstechnieken om de begroting te onderbouwen en te valideren.
- Werk standaardprocedures uit voor het opstellen van begrotingen en plannings.
- Houd in de planning rekening met hertesten.
- Tref voorbereidingen voor eventuele problemen. Denk bijvoorbeeld aan afspraken maken over overwerk, of het prioriteren van de werkzaamheden.

#### Valkuilen

- Gebruik gegevens uit vorige projecten alleen bij vergelijkbare projecten.
- Wanneer 1 begrotingstechniek gebruikt wordt, is het risico op afwijkingen groter. Meerdere technieken leveren betrouwbaardere begrotingen op.



### ***Optimaliserend: Statistisch onderbouwde begroting en planning***

Het opstellen van een begroting wordt verder geoptimaliseerd door statistische gegevens te gebruiken over de doorlooptijden en urenbesteding van eerdere usability testprocessen.

#### Controlepunten

- Statistische gegevens over het proces worden gebruikt bij het onderbouwen van de begroting en planning.
- Tijdens het proces verzamelde statistische gegevens worden gebruikt voor sturing op planning en budget.

#### Verbetersuggesties

- Gebruik metrieken ter ondersteuning van het opstellen van en het sturen op planning en budget.
- Ook al brengen statistisch verkregen gegevens normaliter meer zekerheid, blijf meerdere begrotingstechnieken gebruiken voor betrouwbaardere resultaten.

#### Valkuilen

- Gebruik ook statistische gegevens alleen bij vergelijkbare projecten.

## **8.2.4 Metrieken**

Metrieken spelen een belangrijke rol bij usability, wanneer het proces volwassener wordt. Vergeleken bij TPI kan eerder gestart worden met verbeteringen op dit aandachtspunt. Dat heeft alles te maken met het feit dat inzichtelijk gemaakt moet worden welke concrete resultaten usability testen biedt, voordat genoeg commitment verkregen kan worden om de hogere schalen te bereiken.

### ***Bewust: Statistieken per project***

Per project worden statistieken bijgehouden over het product en het proces, zodat inzichtelijk gemaakt kan worden welke waarde usability heeft voor het project, en voor de organisatie.

#### Controlepunten

- Er worden inputstatistieken bijgehouden:
  - Gebruikte resources: uren
  - Uitgevoerde activiteiten: uren en doorlooptijd
  - Omvang en complexiteit van de te testen interface
- Er worden outputstatistieken bijgehouden:
  - Testproducten: specificaties en testgevallen, logverslagen
  - Testvoortgang: uitgevoerde tests, status
  - Aantal bevindingen: per deelsysteem, naar oorzaak, prioriteit, status, ...
- Statistieken over het product, bijvoorbeeld:
  - Doorlooptijden voor handelingen
  - Aantallen door de gebruikers gemaakte fouten
  - Perceptie van kwaliteit bij de gebruiker
- De statistieken worden gebruikt voor rapportage.

#### Verbetersuggesties

- Begin klein, en breid daarna uit.
- Begin zo vroeg mogelijk met het verzamelen van gegevens, zodat later altijd referentiemateriaal beschikbaar is.
- Maak de metrieken onderdeel van de rapportage templates.

#### Valkuilen

- Te veel metrieken tegelijk invoeren. Metrieken bijhouden is een proces waar alle betrokkenen langzaam aan zullen moeten wennen.
- Metrieken gebruiken om testers, bouwers of ontwerpers af te rekenen op hun functioneren.

### ***Systematisch: Statistieken in archief***

Op dit niveau worden de usability gegevens over de diverse uitgevoerde projecten verzameld in een archief, zodat best en worst practises opgesteld kunnen worden, trends inzichtelijk worden en uiteindelijk organisatiespecifieke ontwerp richtlijnen opgesteld kunnen worden. Wanneer gegevens over meerdere systemen zijn verzameld, worden deze gebruikt om te bepalen welke systemen scoren op gebied van usability en welke niet. Ook wordt per gebruikersgroep geanalyseerd of zij specifieke usability behoeften hebben. Controlepunten

- Als onderdeel van de afronding van de usability test worden de verzamelde gegevens gebundeld en gearchiveerd.
- Van alle systemen in de organisatie zijn metrieken verzameld.
- Deze metrieken worden gebruikt om analyse te doen naar de (relatieve) usability van specifieke systemen, en om analyse te doen naar behoeften van groepen eindgebruikers.
- De analyses worden gebruikt voor verbetering van het ontwerp- en het usability testproces.

#### Verbetersuggesties

- Zorg voor een goed geordend archief, waarin het makkelijk is specifieke resultaten terug te vinden.
- Maak van het archiveren van resultaten een vast onderdeel van de afronding van de usability testen.
- Neem de tijd om resultaten van verschillende projecten te vergelijken.
- Stel de resultaten breed beschikbaar. Deze kunnen nuttig zijn voor bijvoorbeeld projectmanagement, ontwerpers, informatie analisten, testmanagement, opdrachtgevers, etc.
- Maak het gebruik van de verzamelde gegevens een vaste actie in de voorbereiding op de testen.
- Faciliteer de ontwikkeling van een ontwerpstandaard voor gebruiksvriendelijke interfaces.
- Richt het metrieken proces zo in, dat de resultaten gebruikt kunnen worden door de organisatie als KPI.

#### Valkuilen

- De belangrijkste valkuil voor dit niveau is het archief niet beschikbaar stellen aan andere groepen. De belangrijkste winsten die met een archief gehaald kunnen worden liggen namelijk buiten het usability testdomein. Met name ontwerp kan heel veel voordeel halen uit gegroepeerde resultaten van usability tests.
- Er worden zware analyses gedaan wanneer er nog niet genoeg beschikbaar zijn. Dit levert onbetrouwbare resultaten op, waardoor verbeteracties mogelijk niet het gewenste effect hebben.

### ***Niveau D Statistieken over organisatie***

Op dit niveau is de gehele organisatie onderwerp van de te verzamelen usability metrieken. Ook buiten het domein van de IT worden producten beoordeeld op usability, en ook hiervan worden gegevens bijgehouden en geanalyseerd, zodat alle processen van de organisatie gericht zijn op de gebruiker.

#### Controlepunten

- Statistieken over usability worden niet langer alleen voor IT bijgehouden, maar voor alle producten die de organisatie oplevert. Het gaat hierbij niet alleen om de primaire producten van de organisatie, maar ook zaken als vergaderzalen, ontvangstruimtes, kantoren, kantines, ...
- Deze statistieken worden door de gehele organisatie gebruikt om producten te ontwikkelen en te beoordelen.

#### Verbetersuggesties

- Laat de usability afdeling niet (langer) een onderdeel zijn van de IT afdeling, maar geef deze een eigen plaats in de organisatie. Op die manier kan de afdeling zijn scope uitbreiden tot alle producten van de organisatie
- In dit stadium komen usability en ergonomie erg dicht bij elkaar te liggen. Houd daar rekening mee bij de verzameling van gegevens, en combineer deze waar mogelijk.

#### Valkuilen

- Dit niveau kan alleen bereikt worden wanneer de hele (productie) organisatie meewerkt en verandert. Te vroeg dit niveau proberen te bereiken kan tot veel weerstand leiden.

### 8.2.5 Gebruiker Onderzoek

Gerelateerd aan usability testen is gebruiker onderzoek, onderzoek doen naar de eindgebruikers van het product, voordat de ontwerpfase start.

#### *Systematisch: Kwalitatief*

Onderzoek naar gebruikers beperkt zich tot kwalitatief onderzoek.

Controlepunten

- Bij belangrijke projecten wordt voor de ontwerpfase onderzoek gedaan naar de eindgebruikers.
- Dit onderzoek richt zich op het gedrag van de gebruikers
- Dit onderzoek gebeurt op een gestructureerde wijze

Verbetersuggesties

- Laat gebruiker onderzoek opnemen in de projectplanning, voor de start van de ontwerpfase
- Begin met kwalitatief onderzoek van beperkte omvang. Hiermee zijn namelijk snel resultaten te boeken.
- Borg de resultaten voor toekomstig gebruik.
- Gebruik waar mogelijk ook opgeslagen testresultaten als input of referentiemateriaal.

Valkuilen

- Laat dit onderzoek alleen uitvoeren door medewerkers die weten hoe een onderzoek opgezet en uitgevoerd moet worden. Als de eerste resultaten tegenvallen, zal het commitment voor gebruiker onderzoek snel verdwijnen.
- Start niet met kwantitatief onderzoek. Omdat een kwantitatief onderzoek harde cijfers oplevert, is het van belang dat de resultaten op een betrouwbare manier vergaard zijn, en grondig geanalyseerd worden. Deze processen zijn zeer gevoelig voor de kleinste afwijkingen.
- Usability komt in een vroeg stadium van het project op het kritieke pad terecht. Zorg ervoor dat onderzoekers en eindgebruikers tijdig beschikbaar zijn.
- Gebruiker onderzoek is geen doel op zich. Zorg er van tevoren voor dat ontwerpers open staan voor resultaten uit gebruiker onderzoek.
- Kijk uit met het onderzoeken van de mening van de eindgebruiker. Wat de gebruiker zegt, is vaak niet hetzelfde als wat hij doet.

#### *Optimaliserend: Kwantitatief*

Op dit niveau wordt naast kwalitatief onderzoek ook kwantitatief onderzoek uitgevoerd. Kwantitatief onderzoek biedt mogelijkheden om statistische analyses uit te voeren en zo bijvoorbeeld tendensen te ontdekken.

Kwantitatief onderzoek is echter kostbaar, en gevoelig voor fouten.

Controlepunten

- Bij (belangrijke) projecten wordt kwantitatief onderzoek gedaan naar de gebruikers.
- Dit onderzoek wordt uitgevoerd door professionele onderzoekers.
- De uitkomsten van het onderzoek worden gebruikt voor statistische analyses, en dienen als input voor het ontwerpproces.

Verbetersuggesties

- Voer kwantitatief onderzoek uit met groepen gebruikers.
- Zorg voor geschikte faciliteiten. Kwantitatief onderzoek is gevoelig voor invloeden van buitenaf, deze moeten daarom zoveel mogelijk beperkt worden.
- Neem een professionele onderzoeker aan, of leidt deze op.
- Maak gebruik van tools voor statistische analyse.

#### Valkuilen

- Kwantitatief onderzoek uitvoeren zonder de juiste mensen en faciliteiten te gebruiken. Een verkeerd ontwerp, een invloed van buitenaf of een verkeerde interpretatie kunnen volledig onjuiste resultaten opleveren.
- Kijk uit met resultaten die (nog) niet gereproduceerd zijn. Hoe goed een onderzoek en daaropvolgende analyse ook zijn, statistisch gezien toont 1 op de 40 onderzoeken een correlatie aan, die in werkelijkheid niet bestaat.

### 8.2.6 Usability Tools

Net als bij testen in het algemeen, spelen tools ook bij usability testen een belangrijke rol. Zie hiervoor ook het hoofdstuk Tools in dit white paper.

#### ***Bewust: Gebruik van op zichzelf staande tools***

Op dit niveau worden wel tools ingezet, maar nog niet systematisch. Diverse losse tools kunnen gebruikt worden om bijvoorbeeld gegevens te vergaren en te analyseren, of om de testuitvoering te ondersteunen

#### Controlepunten

- Er wordt gebruik gemaakt van (geautomatiseerde) hulpmiddelen die bepaalde activiteiten in de fase planning en uitvoering ondersteunen.
- De gebruikte tools bieden aantoonbare voordelen in het usability testproces.

#### Verbetersuggesties

- Onderzoek welke tools er op usability gebied zijn. (zie hiervoor ook hoofdstuk Tools)
- Overweeg welke activiteiten geautomatiseerd kunnen worden, of welke baat hebben bij ondersteuning van een tool.
- Toon aan, bijvoorbeeld via metriecken, dat tools voordelen bieden.

#### Valkuilen

- Tools blijven hulpmiddelen. Gebruik alleen tools wanneer deze voordeel kunnen bieden bij het uitvoeren van bepaalde handelingen.
- Bij usability tools wordt vaak gedacht aan dure pakketten waarmee laboratoria uitgerust zijn. Er zijn echter ook kleinere softwarepakketten, of zelfs freeware beschikbaar. Ook met spreadsheet applicaties kunnen hulpmiddelen gemaakt worden.

#### ***Systematisch: Tools ingebed in methode***

Op dit niveau maken tools deel uit van het usability testproces vanaf de fase planning tot de fase afronding. Bij de beslissing om een tool wel of niet te implementeren en te gebruiken worden altijd de voordelen en de nadelen afgewogen.

#### Controlepunten

- Er is een overwogen beslissing genomen welke delen van de testuitvoering wel of niet te automatiseren.
- Indien de besluitvorming rond automatisering van testuitvoering positief is uitgevallen, is er inmiddels ook sprake van het gebruik van tools voor testuitvoering.
- De introductie van nieuwe usability testtools wordt voorafgegaan door een inventarisatie van technische aspecten en mogelijke randvoorwaarden die aan het testproces gesteld worden.

#### Verbeteracties

- Maak het selecteren van tooling een vast onderdeel van de fase planning, en van het template MTP.
- Verkrijg voor aanschaf inzicht in de verwachte kosten/baten verhouding van gewenste tools.

#### Valkuilen

- De behoefte aan tooling zal groter worden wanneer usability testen meer, vaker en groter opgezet zullen worden. Het is valkuil om de volgende redenen:
  - Tools die eerder te duur bleken niet meer te overwegen.
  - Aan te sturen op uitbreiding van de tooling, zonder dat het usability testproces daar klaar voor is.
- De beheerorganisatie zou op dit niveau op zijn minst af moeten weten van de gebruikte tools, aangezien deze een vast onderdeel worden van een proces.

#### *Optimaliserend: herbruikbare tools*

Op dit niveau worden tools zo geselecteerd en ingezet, dat ze herbruikbaar zijn, bijvoorbeeld voor regressietesten of nieuwe releases. Usability tools maken hierdoor deel uit van het beheerproces.

#### Controlepunten

- De gebruikte testtools zijn overwegend herbruikbaar voor een volgend testproces. Hiertoe is het beheer van de testtools geregeld.
- Het gebruik en beheer van de testtools past binnen de (gewenste) werkwijze van het gehele testproces.

#### Verbeteracties

- Regel het beheer van testtools in. Stem dit af met de beheerorganisatie.
- Houd de kosten/baten verhouding van de gebruikte tools bij, en stuur hierop.

#### Valkuilen

- De beheerorganisatie niet op de hoogte brengen van de aanwezigheid van de gebruikte tools.
- Geen afspraken maken over het beheer van de tools in het kader van het algehele beheerproces.

### **8.2.7 Usability werkplek**

#### *Bewust: Werkplek*

Op dit niveau zijn werkplekken beschikbaar voor die mensen die zich bezighouden met usability testen op een project.

#### Controlepunten

- De voor testen benodigde kantoorinfrastructuur is tijdig geregeld.
- Zaken rondom kantoorinrichting hebben minimale impact op het verloop van het usability testproces.

#### Verbetersuggesties

- Maak inzichtelijk dat het niet hebben van een (geschikte) werkplek extra kosten met zich meebrengt, of de effectiviteit verlaagt.

#### Valkuilen

- Het is onverstandig om usability werkplekken fysiek ver van de rest van de projectorganisatie én van de gebruikersorganisatie te plaatsen. Er kan veel gericht getest worden wanneer de testers een goed beeld hebben van de eindgebruikers, hun werkzaamheden en hun processen.

#### *Systematisch: Permanente usability afdeling*

Op dit niveau is er binnen de organisatie een vaste plek waar usability medewerkers bij elkaar zitten.

#### Controlepunten

- Een ruimte, of een deel van een ruimte, is permanent gereserveerd voor meerdere usability medewerkers en hun spullen.
- In deze ruimte zijn geschikte werkplekken aanwezig, en is voldoende kantoorinrichting beschikbaar.

#### Verbetersuggesties

- Ruimte beschikbaar stellen brengt vaak hoge kosten met zich mee. Maak inzichtelijk welke voordelen het biedt om de usability medewerkers bij elkaar te laten zitten.
- Op projectbasis kan het voordelen bieden om medewerkers (full-time of parttime) bij het projectteam te plaatsen.

#### Valkuilen

- Als de afdeling fysiek moeilijk bereikbaar is, schiet het oprichten ervan zijn doel voorbij. Het blijft belangrijk om veel te communiceren met projectleden en eindgebruikers, de ruimte moet daar wel geschikt voor zijn.

#### *Optimaliserend: usability lab beschikbaar*

Op dit niveau is een ruimte beschikbaar die zo ingericht is dat deze geschikt is voor het uitvoeren van usability onderzoek.

#### Controlepunten

- De usability groep heeft altijd de beschikking over een ruimte
  - Waar een standaard werkplek staat
  - Voldoende mogelijkheden zijn tot observatie van gebruikers
  - Geen hinderlijke geluiden van buitenaf door kunnen dringen

#### Verbetersuggesties

- Begin met een laboratorium van beperkte omvang, en toon met concrete resultaten aan wat het de organisatie oplevert.
- Bij wijze van pilot kan voor een eerste onderzoek ook gebruik gemaakt worden van een extern laboratorium.
- Een extern laboratorium kan ook op regelmatige basis gebruikt worden voor onderzoek, wanneer de kosten van het inrichten te hoog gevonden worden.

#### Valkuilen

- Bij een laboratorium wordt vaak gedacht aan een ruimte vol camera's en microfoons, en een spiegelwand. Een laboratorium kan echter ook kleiner van opzet zijn. Een standaard kamer met werkplek en voldoende ruimte voor observatie kan al dienst doen als laboratorium.
- De kosten en de baten van een intern en een extern laboratorium niet tegen elkaar uitzetten.

### **8.2.8 Stakeholder commitment**

#### *Bewust: Toewijzing van budget en tijd*

Op dit niveau krijgt usability testen op projectbasis tijd en budget toegewezen.

#### Controlepunten

- Usability testen krijgt een hoeveelheid budget en tijd toegekend. Hierop wordt ook gestuurd door het (test)management.
- Minimaal één werknemer is fulltime bezig met usability.
- Er is een goede verhouding tussen usability en andere disciplines in het project en de organisatie.

#### Verbetersuggesties

- Signaleer usability problemen in bestaande systemen, en geef aan hoe op korte termijn verbeteringen tot verbeteringen kunnen leiden.
- Toon aan dat vroege signalering (of zelfs preventie) van usability issues tot kostenbesparing in het project, en hogere kwaliteit van het eindproduct kan leiden.

#### Valkuilen

- Wanneer een organisatie in een lage schaal zit, zal het vaak niet meevallen om een nieuwe fulltime usability medewerker ingezet te krijgen. Probeer op dit niveau fulltime inzet van al aanwezige medewerkers op usability te bewerkstelligen.

### ***Systematisch: Testen geïntegreerd in projectorganisatie***

Op dit niveau heeft usability een vaste plek in de organisatie, en geldt als een aparte discipline binnen de projectorganisatie.

#### Controlepunten

- Alle betrokkenen vinden dat usability een duidelijk waarneembare positieve invloed heeft op de kwaliteit van een product.
- Het (test)management stuurt de usability afdeling aan op tijd, geld en kwaliteit.
- Bij onvoldoende usability van producten kunnen bouwers en ontwerpers hierop aangesproken worden.
- De adviezen vanuit usability worden besproken in het projectoverleg.

#### Verbetersuggesties

- Toon aan dat de aandacht die tot nu toe op projectbasis is gegeven aan usability, geleid heeft tot kostenbesparingen in het project en hogere kwaliteit van eindproducten.
- Bepleit met gehaalde resultaten een structurele rol voor usability binnen projecten.

#### Valkuilen

- Vanaf schaal 4 zullen de activiteiten van de usability medewerkers niet langer beperkt blijven tot zuiver testwerk. Waak ervoor dat usability een eigen begroting krijgt

### ***Optimaliserend: Usability geaccepteerd in organisatie***

Op dit niveau is binnen de hele organisatie het belang van usability bekend. Uiteindelijk zal usability deel uit gaan maken van de strategie van de organisatie

#### Controlepunten

- Usability wordt bij elk project als kwaliteitsindicator gebruikt.
- In elke fase van een project gelden gebruikersgegevens als belangrijke input.
- Usability bepaalt de richting en de prioriteiten van de organisatie

#### Verbetersuggesties

- Op het niveau optimaliserend moet het accent voor het verkrijgen van stakeholder commitment verschuiven naar commitment van het hogere management.
- Signaleer ook buiten het IT domein usability problemen in producten van de organisatie, en geef aan hoe verbeteringen kunnen leiden tot winst voor de organisatie.

#### Valkuilen

- Waak ervoor dat gebruikersgegevens betrekking hebben op wat de gebruiker doet, en niet wat hij zegt.
- Een marketingonderzoek is geen substituut voor het verzamelen van harde gebruikersgegevens.

## **8.2.9 Usability awareness**

Een groot deel van de verbetersuggesties die genoemd worden in dit hoofdstuk, vallen of staan bij voldoende awareness van usability in de uitvoeringsorganisatie.

### ***Bewust: Algemeen nut bekend***

Op dit niveau wordt onderkend dat een gebruiker serieuze problemen kan ondervinden wanneer een applicatie niet gebruiksvriendelijk is en dat ontwerpers en ontwikkelaars niet geschikt zijn als beoordelaars van gebruiksvriendelijkheid.

#### Controlepunten

- Ontwerpers en ontwikkelaars houden, in welke vorm dan ook, rekening met gebruiksvriendelijkheid bij de uitvoering van hun werkzaamheden.
- Ontwerpers en ontwikkelaars staan open voor usability bevindingen, en schatten deze over het algemeen belangrijker in dan “cosmetisch”.
- Ontwikkelaars en ontwerpers zien de noodzaak om gebruikers mee te laten doen in tenminste de testfase van een project.

#### Verbetersuggesties

- Maak bij het loggen van usability bevindingen ook inzichtelijk wat de gevolgen kunnen zijn. Denk aan gebruikers die langer doen over hun taken, klanten die afhaken of fouten die gemaakt kunnen worden.
- Laat ontwerpers en ontwikkelaars, indien mogelijk, in een vroeg stadium weten dat gekeken zal worden naar gebruiksvriendelijkheid. Neem dit ook op in het MTP.
- Nodig ontwerpers en ontwikkelaars uit bij tests met gebruikers zodat zij met eigen ogen kunnen zien dat usability issues een hoge impact kunnen hebben en dat gebruikers waardevolle input kunnen leveren.

#### Valkuilen

- Probeer usability alleen te “verkopen” wanneer zich concrete verbeterpunten manifesteren. Wanneer een organisatie nog niet aan dit niveau voldoet, zou dat niet lang moeten duren.
- Waak ervoor de de boodschap, dat gebruikersinput nodig is om gebruiksvriendelijke producten te creëren, overkomt als een diskwalificatie van ontwerpers en ontwikkelaars. Een manier om dat te doen is de nadruk te leggen op het feit dat zij eigenlijk veel te veel kennis hebben, om een goed beeld te hebben bij het perspectief van de eindgebruiker.

#### ***Systematisch: Usability ROI (Return On Investment)***

Op dit niveau is binnen de (project)organisatie bekend dat voor usability (issues) een ROI berekend kan worden.

#### Controlepunten

- Wanneer een bevinding of een wijzigingsvoorstel gedaan wordt op het gebied van usability, wordt de prioriteit bepaald aan de hand van een ROI berekening.
- Binnen de organisatie is bekend dat een goede usability ook voor interne applicaties van waarde kan zijn.

#### Verbetersuggesties

- Laat door de usability testen de ROI van de verbeteringen op het gebied van usability zien. Probeer ook bij het registreren van een bevinding de prioriteitsstelling zo te onderbouwen.

#### Valkuilen

- Richt de voordelen voor de organisatie niet al te veel op alleen maar besparingen. Dan kan het beeld ontstaan dat usability gebruikt kan worden om met minder medewerkers toe te kunnen. Geef ook aan dat usability verbeteringen de productiviteit van medewerkers kunnen verhogen, waardoor een organisatie kan groeien, of zijn service kan verbeteren.

#### ***Optimaliserend: Usability input bij besluitvorming***

Usability data bepaalt in sterke mate welke projecten uitgevoerd gaan worden en welke producten dit oplevert.

#### Controlepunten

- Gegevens over usability worden op management niveau gebruikt om te bepalen welke projecten uitgevoerd gaan worden.
- Gegevens over usability worden op projectniveau gebruikt om te bepalen welke producten opgeleverd gaan worden.



#### Verbetersuggesties

- Zorg ervoor dat op management niveau bekend wordt wat usability betekent voor een project en voor de organisatie. Nodig managers uit om mee te kijken met usability tests en onderzoek.
- Doe zelf suggesties voor projecten, op basis van testresultaten en gebruikersonderzoek. Laat zien hoe usability gegevens gebruikt kunnen worden op strategisch niveau.

#### Valkuilen

- Probeer dit niveau alleen te bereiken wanneer er hele concrete resultaten geboekt zijn.
- Niet elke organisatie zal dit niveau moeten of willen bereiken. Bedenk van vooraf of de organisatie deze kant wel uit moet.
- Voor het gebruiken van klantgegevens in het beslisproces wordt al langere tijd gepleit. In de praktijk worden klantgegevens echter zelden gebruikt. Waak ervoor dat op dit niveau opinie peilingen niet gebruikt worden als klant, of usability gegevens.

### **8.2.10 Usability eisen**

Dit aandachtsgebied beschrijft in hoeverre usability meegenomen wordt bij het opstellen van de eisen voor een project.

#### ***Systematisch: Richtlijnen worden gebruikt bij ontwikkeling***

Op dit niveau zijn aan de hand van ervaringen uit usability tests richtlijnen opgesteld voor de ontwikkeling van een gebruiksvriendelijk product. Dit zijn geen harde eisen, maar zaken waar bij het opstellen van eisen, het ontwerpen en het ontwikkelen van een product rekening mee gehouden moet worden. Uiteindelijk zullen de richtlijnen de vorm gaan krijgen van standaards.

#### Controlepunten:

- Er is een officieel document met richtlijnen voor ontwerp en ontwikkeling van gebruiksvriendelijke user interfaces.
- Dit document is gebaseerd op ervaringen uit usability tests.
- Dit document wordt minimaal gebruikt bij de toetsing van user interface ontwerpen.
- De richtlijnen zijn vaste standaarden voor user interfaces geworden. Deze standaarden zijn gebaseerd op usability tests en onderzoek, en hebben als doel gebruiksvriendelijke interfaces op te leveren.

#### Verbetersuggesties

- Stel een document met richtlijnen op in samenspraak, of zelfs samenwerking, met ontwerpers.
- Geef aan hoe richtlijnen tot stand zijn gekomen zodat indien nodig achteraf bepaalde beslissingen onderbouwd heroverwogen of juist bekrachtigd kunnen worden.
- Gebruik dit document ook bij het uitvoeren van toetsen en testen als leidraad.
- Zorg ervoor dat ontwerpers weten van het bestaan van deze richtlijnen en stimuleer het gebruik hiervan.
- Vaak zullen al ontwerpstandaarden in de organisatie aanwezig zijn. Probeer hierbij aan te sluiten, maar maak indien nodig duidelijk, met behulp van testresultaten, dat deze standaarden heroverwogen moeten worden, of dat zelfs nieuwe standaarden ontwikkeld moeten worden.
- Zorg ervoor dat deze standaarden gebruikt worden tijdens de ontwerpfase, en toets hier ook op.

#### Valkuilen

- Richtlijnen niet gebruiken bij toetsen en testen. Het is belangrijk dat getoetst blijft worden of de richtlijnen gehandhaafd worden.
- Het document niet onderbouwen. Deze richtlijnen kunnen kritische vragen oproepen van ontwerpers. Hier geldt namelijk hetzelfde als bij "Moment van betrokkenheid" niveau B: usability testers gaan zich begeven op het terrein van de ontwerpers.
- Voor dit niveau tellen alleen standaarden die gestoeld zijn op tijdens usability testen en gebruikersonderzoek behaalde resultaten.

- Alle aanwezige standaarden zijn waarschijnlijk ontwikkeld door een ontwerper. Kritiek hierop kan averechts werken. Maak daarom altijd duidelijk dat verbeter suggesties gestoeld zijn op objectief verkregen resultaten.

***Optimaliserend: Usability eisen subset van de requirements***

Op dit niveau worden voor alle projecten als vast onderdeel van de requirements definitie afzonderlijke usability eisen bepaald.

**Controlepunten**

- Tijdens de requirements definitie worden als subset van het totale eisenpakket usability eisen opgesteld.
- Deze eisen worden gemanaged in een iteratief proces kunnen daarom naar aanleiding van testresultaten bijgesteld worden.

**Verbetersuggesties**

- Voor dit niveau kunnen ook al usability gerelateerde eisen gedefinieerd worden. Deze zijn in de praktijk vaak vrij algemeen, omdat kennis ontbreekt over hoe usability SMART gemaakt kan worden. Zorg ervoor dat er organisatiebreed een goed beeld is hoe usability eisen SMART gedefinieerd kunnen worden. Pas dan kunnen deze eisen in het project echt getest en getoetst worden.

**Valkuilen**

- In de praktijk komt het voor dat algemene usability eisen specifiek gemaakt worden door het “hoe” te beschrijven. Denk bijvoorbeeld aan een eis “de interface moet uitgaan van het WYSIWYG (What you see is what you get) principe”. Echter, deze eis op zichzelf zegt nog niets over de usability van een user interface. Zorg ervoor dat usability eisen alleen het “wat” beschrijven, bijvoorbeeld “een nieuwe gebruiker moet taak x binnen y minuten uit kunnen voeren”.

## BIJLAGE A

Hieronder wordt voor elke fase in het ontwikkelproces weergegeven uit welke technieken gekozen kan worden, wanneer in deze fase een usability test plaatsvindt.

<b>Testtechnieken</b>	<b>Korte omschrijving</b>	<b>Fase in V-model</b>
Heuristic evaluation	Evaluatie op basis van vuistregels.	Hele traject
Perspective-based inspection	Inspectie op basis van rollen.	Hele traject
Interviews	Vraaggesprek met personen uit de doelgroep.	Hele traject
Thinking aloud	Een proefpersoon denkt hardop.	Hele traject
Logging actual use	Automatisch verzamelde gegevens over het gebruik van het systeem worden geanalyseerd.	Rechter kant V-model
Observatie	Observatie van gebruikers in actie.	Hele traject
<b>Toetstechnieken</b>		
Cognitive walkthrough	Gebruikers worden door het systeem of het ontwerp geleid volgens voorgeschreven paden.	Hele traject
Consistentie inspection	Inspectie van consistentie in gebruikte termen, vormen en interacties.	Hele traject
Model-based Inspection	Een mockup op basis van een ontwerp of requirements wordt als testobject gebruikt.	Linker kant V-model
Pluralistic walkthrough	Cognitive walkthrough met een groep mensen tegelijk.	Hele traject
Focus groups	Interview met een groep mensen uit de doelgroep.	Hele traject
Reverse card sorting	Gebruikers krijgen zoektaak taken binnen een hiërarchische structuur.	Hele traject
<b>Realisatietechnieken</b>		
Personas	Gebruikersprofielen worden opgesteld en gebruikt tijdens het ontwerp.	Linker kant V-model
Card Sorting	Hulpmiddel om een logische hiërarchische indeling van informatie te krijgen.	Linker kant V-model
Prototype testing	Het testen van een prototype van de software.	Linker kant V-model

## BIJLAGE B CHECKLISTS

### Nuttige links

URL's waar informatie is te vinden voor verschillende onderdelen.

#### *Navigatie:*

<http://www.frankwatching.com/archive/2009/05/27/checklist-10-tips-voor-een-effectieve-navigatie/>

#### *Zoekmachines:*

[http://louisrosenfeld.com/home/bloug\\_archive/000290.html](http://louisrosenfeld.com/home/bloug_archive/000290.html)

<http://www.adaptivepath.com/ideas/essays/archives/000341.php>

#### *Design patterns*

Design patterns voor verschillende onderdelen, ook als inspiratie voor checklist en suggestie om dingen beter te maken:

<http://www.welie.com/patterns/>

### Algemeen toepasbare checklist

De volgende checklist is ontwikkeld door gerenomeerd usability specialist van de Nielsen Norman Group, Bruce Tognazzini en komt van <http://www.asktog.com/basics/firstPrinciples.html>.

<b>Anticipation</b>	Applications should attempt to anticipate the user's wants and needs. Do not expect users to search for or gather information or evoke necessary tools. Bring to the user all the information and tools needed for each step of the process.
<b>Autonomy</b>	<p>The computer, the interface, and the task environment all "belong" to the user, but user-autonomy doesn't mean we abandon rules.</p> <p>Give users some breathing room. Users learn quickly and gain a fast sense of mastery when they are placed "in charge." Paradoxically, however, people do not feel free in the absence of all boundaries (Yallum, 1980). A little child will cry equally when held too tight or left to wander in a large and empty warehouse. Adults, too, feel most comfortable in an environment that is neither confining nor infinite, an environment explorable, but not hazardous.</p> <p>Use status mechanisms to keep users aware and informed.</p> <p>No autonomy can exist in the absence of control, and control cannot be exerted in the absence of sufficient information. Status mechanisms are vital to supplying the information necessary for workers to respond appropriately to changing conditions. As a simple example, workers, failing status information, will tend to maintain heightened pressure on themselves during slow periods, until the moment the work actually runs out. This will stress and fatigue them unnecessarily, so that when the next rush occurs, they may be lacking the physical and mental reserves to handle it.</p> <p>Keep status information up to date and within easy view</p>

Users should not have to seek out status information. Rather, they should be able to glance at their work environment and be able to gather at least a first approximation of state and workload. Status information can be quite subtle: the inbox icon could be switched to show an empty, somewhat full, or stuffed state. This, however, should not be overdone. The Macintosh, for years, showed an icon of a trashcan of imminent danger of explosion if a single document was placed therein. Users quickly formed the habit of emptying the trashcan as soon as the first document hit. This not only turned a single-step operation into a two-step operation (drag to the trash, then empty the trash), it negated the entire power of the trashcan, namely, undo.

As another positive example, a search field icon can change color and appearance to indicate that the search is in progress or has been completed with too many matches, too few matches, or just enough. (Like any element of the interface, just color is not enough; 10% of males show some indication of color blindness. Even a higher percentage may have temporary alterations in perception of blue under varying conditions.)

#### **Color Blindness**

Any time you use color to convey information in the interface, you should also use clear, secondary cues to convey the information to those who won't be experiencing any color coding today.

Most people have color displays nowadays, but they are not universal. In addition, approximately 10% of human males, along with a rare sprinkling of females, have some form of color blindness.

The cones in the eye are the source of color vision. We have cones separately sensitive to red, green, and blue. If the red ones are not functioning that is called protanopia. If the green are not functioning, that is called deuteranopia. Absence of blue, extremely rare, is called tritanopia.

Protanopia and deuteranopia are the most popular forms of color blindness, collectively called red/green blindness. (There are, in fact, significant differences in their effects, but those differences have no real effect on interaction design.) While tritanopia is far more rare, it nonetheless rules out dependence on yellow-blue differentiation without secondary cues.

Secondary cues can consist of anything from the subtlety of gray scale differentiation to having a different graphic or different text label associated with each color presented.

#### **Consistency**

The following principles, taken together, offer the interaction designer tremendous latitude in the evolution of a product without seriously disrupting those areas of consistency most important to the user.

Levels of consistency: The importance of maintaining strict consistency varies. The following list is ordered from those interface elements demanding the most faithful consistency effort to those demanding the least. Paradoxically, many people assume that the order of items one through five should be exactly the reverse, leading to applications that look alike, but act completely different in unpredictable ways:

1. Interpretation of user behavior, e. g., shortcut keys maintain their meanings.
2. Invisible structures.
3. Small visible structures.
4. The overall "look" of a single application or service--splash screens, design elements.
5. A suite of products.
6. In-house consistency.
7. Platform-consistency.

"Invisible structures" refers to such invisible objects as Microsoft Word's clever little right border that has all kinds of magical properties, if you ever discover it is there. It may or may not appear in your version of Word. And if it doesn't, you'll never know for sure that it isn't really there, on account of it's invisible. Which is exactly what is wrong with invisible objects and why consistency is so important. Other objects are, strictly speaking, visible, but do not appear to be controls, so users, left to their own devices, might never discover their manipulability. The secret, if you absolutely insist on one, should be crisp and clean, for example, "you can click and drag the edges of current Macintosh windows to size them," not, "You can click and drag various things sometimes, but not other things other times."

"Small visible structures" refers to icons, size boxes, scroll arrows, etc. The appearance of such objects needs to be strictly controlled if people are not to spend half their time trying to figure out how to scroll or how to print. Location is only just slightly less important than appearance. Where it makes sense to standardize location, do so.

Inconsistency: It is just important to be visually inconsistent when things must act differently as it is to be visually consistent when things act the same.

Avoid uniformity. Make objects consistent with their behavior. Make objects that act differently look different.

The most important consistency is consistency with user expectations.

The only way to ascertain user expectations is to do user testing. No amount of study and debate will substitute.

#### **Defaults**

Defaults should be easy to "blow away:" Fields containing defaults should come up selected, so users can replace the default contents with new material quickly and easily.

Defaults should be "intelligent" and responsive.

Do not use the word "default" in an application or service. Replace with "Standard," "Use Customary Settings," "Restore Initial Settings," or some other more specific terms describing what will actually happen.

#### **Efficiency of**

Look at the user's productivity, not the computer's.

## the User

People cost a lot more money than machines, and while it might appear that increasing machine productivity must result in increasing human productivity, the opposite is often true. In judging the efficiency of a system, look beyond just the efficiency of the machine.

For example, which of the following takes less time? Heating water in a microwave for one minute and ten seconds or heating it for one minute and eleven seconds?

From the standpoint of the microwave, one minute and ten seconds is the obviously correct answer. From the standpoint of the user of the microwave, one minute and eleven seconds is faster. Why? Because in the first case, the user must press the one key twice, then visually locate the zero key, move the finger into place over it, and press it once. In the second case, the user just presses the same key—the one key—three times. It typically takes more than one second to acquire the zero key. Hence, the water is heated faster when it is "cooked" longer.

Other factors beyond speed make the 111 solution more efficient. Seeking out a different key not only takes time, it requires a fairly high level of cognitive processing. While the processing is underway, the main task the user was involved with—cooking their meal—must be set aside. The longer it is set aside, the longer it will take to reacquire it.

Additionally, the user who adopts the expedient of using repeating digits for microwave cooking faces fewer decisions. They soon abandon figuring out, for example, whether bacon should be cooked for two minutes and ten seconds or two minutes and twenty-three seconds. They do a fast estimate and, given the variability of water content and bacon thickness, end up with as likely a successful result with a lot less dickering up front, again increasing human efficiency.

Keep the user occupied.

Since, typically, the highest expense in a business is labor cost. Any time the user must wait for the system to respond before they can proceed, money is being lost.

To maximize the efficiency of a business or other organization you must maximize everyone's efficiency, not just the efficiency of a single group.

Large organizations tend to be compartmentalized, with each group looking out for its own interests, sometimes to the detriment of the organization as a whole. Information resource departments often fall into the trap of creating or adopting systems that result in increased efficiency and lowered costs for the information resources department, but only at the cost of lowered productivity for the company as a whole.

For example, one large California corporation used floppy disks as the medium for collecting benefit enrollment information. At the beginning of open enrollment, each employee would receive a disk with the enrollment applications on which he or she would insert into their computer and run. After asking for the employee's name, address, phone number, department name, etc., the employee would be permitted to step through all the various benefits, ultimately returning the disk which now contained all their answers and decisions. The IR department then sucked the data off each disk and entered it into their system, all automatically. The IR department saved a great deal of money over the old system, where they had to key in the employee's decisions from a paper form.

What was the problem? Instead of the IR department bearing the burden of keying in the employees' decisions, each and every employee now bore the burden of typing in his or her name, address, phone number, department name, etc. The system was just as inefficient as before, but now the cost was borne by all departments, rather than having it concentrated in the IR department's budget.

The great efficiency breakthroughs in software are to be found in the fundamental architecture of the system, not in the surface design of the interface.

This simple truth is why it is so important for everyone involved in a software project to appreciate the importance of making user productivity goal one and to understand the vital difference between building an efficient system and empowering an efficient user. This truth is also key to the need for close and constant cooperation, communication, and conspiracy between engineers and human interface designers if this goal is to be achieved.

Write help messages tightly and make them responsive to the problem: good writing pays off big in comprehension and efficiency.

Menu and button labels should have the key word(s) first.

Example from a fictitious word processor:

Wrong:

Insert page break  
Add Footnote  
Update Table of Contents

Right:

Insert:

Page break  
Footnote  
Table of contents



Here, the first example, with its leading words, is actually more informative and more accurate: one does not "insert" a footnote if it is to be placed after all the other footnotes. And one does not insert a table of contents if there is already a table of contents there. Instead, one updates it. Still, the second example will prove much more efficient in time-trials. Why? Because the extra information the first example offers does not outweigh the advantage of being able to scan only the first word in each menu item to find the specific menu item you are after.

### **Explorable Interfaces**

Give users well-marked roads and landmarks, then let them shift into four-wheel drive.

Mimic the safety, smoothness, and consistency of the natural landscape. Don't trap users into a single path through a service, but do offer them a line of least resistance. This lets the new user and the user who just wants to get the job done in the quickest way possible and "no-brainer" way through, while still enabling those who want to explore and play what-if a means to wander farther afield.

Sometimes, however, you have to provide deep ruts.

The closer you get to the naive end of the experience curve, the more you have to rein in your users. A single-use application for accomplishing an unknown task requires a far more directive interface than a habitual-use interface for experts.

Offer users stable perceptual cues for a sense of "home."

Stable visual elements not only enable people to navigate fast, they act as dependable landmarks, giving people a sense of "home."

Make Actions reversible

People explore in ways beyond navigation. Sometimes they want to find out what would happen if they carried out some potentially dangerous action. Sometimes they don't want to find out, but they do anyway by accident.

By making actions reversible, users can both explore and can "get sloppy" with their work.

Always allow "Undo."

The unavoidable result of not supporting undo is that you must then support a bunch of dialogs that say the equivalent of, "Are you really, really sure?" Needless to say, this slows people down.

In the absence of such dialogs, people slow down even further. A study a few years back showed that people in a hazardous environment make no more mistakes than people in a supportive and more visually obvious environment, but they worked a lot slower and a lot more carefully to avoid making errors.

Always allow a way out.

Users should never feel trapped. They should have a clear path out.

However, make it easier to stay in.

Early software tended to make it difficult to leave. With the advent of the web, we've seen the advent of software that makes it difficult to stay. Web browsers still festoon their windows with objects and options that have nothing to do with our applications and services running within. Our task can become akin to designing a word process which, oh, by the way, will be using Photoshop's menu bar. Having 49 options on the screen that lead directly to destruction of the user's work, along with one or two that just might help is not an explorable interface, it is the interface from hell. If you are working with complex transactions using a standard web browser, turn off the menu bar and all of the other irrelevant options, then supply our own landmarks and options.

#### **Fitts' Law**

The time to acquire a target is a function of the distance to and size of the target.

While at first glance, this law might seem patently obvious, it is one of the most ignored principles in design. Fitts' law (properly, but rarely, spelled "Fitts's Law") dictates the Macintosh pull-down menu acquisition should be approximately five times faster than Windows menu acquisition, and this is proven out.

Fitts' law dictates that the windows task bar will constantly and unnecessarily get in people's way, and this is proven out. Fitts' law indicates that the most quickly accessed targets on any computer display are the four corners of the screen, because of their pinning action, and yet, for years, they seemed to be avoided at all costs by designers.

Use large objects for important functions (Big buttons are faster).

Use the pinning actions of the sides, bottom, top, and corners of your display: A single-row toolbar with tool icons that "bleed" into the edges of the display will be many times faster than a double row of icons with a carefully-applied one-pixel non-clickable edge between the tools and the side of the display.

[For more on Fitts' Law, see my A Quiz Designed to Give You Fitts](#)

#### **Human Interface Objects**

Human-interface objects are not necessarily the same as objects found in object-oriented systems. Our objects include folders, documents, and the trashcan. They appear within the user's environment and may or may not map directly to an object-oriented object. In fact, many early gui's were built entirely in non-object-oriented environments.

Human-interface objects can be seen, heard, touched, or otherwise perceived.

Human interface objects that can be seen are quite familiar in graphic user interfaces. Objects that play to another sense such as hearing or touch are less familiar. Good work has been done in developing auditory icons (Gaver).

Human-interface objects have a standard way of interacting.

Human-interface objects have standard resulting behaviors.

Human-interface objects should be understandable, self-consistent, and stable.

**Latency Reduction**

Wherever possible, use multi-threading to push latency into the background.

Latency can often be hidden from users through multi-tasking techniques, letting them continue with their work while transmission and computation take place in the background.

Reduce the user's experience of latency.

Acknowledge all button clicks by visual or aural feedback within 50 milliseconds.

Display an hourglass for any action that will take from 1/2 to 2 seconds.

Animate the hourglass so they know the system hasn't died.

Display a message indicating the potential length of the wait for any action that will take longer than 2 seconds.

Communicate the actual length through an animated progress indicator.

Offer engaging text messages to users informed and entertained while they are waiting for long processes, such as server saves, to be completed.

Make the client system beep and give a large visual indication upon return from lengthy (>10 seconds) processes, so that users know when to return to using the system.

Trap multiple clicks of the same button or object. Because the Internet is slow, people tend to press the same button repeatedly, causing things to be even slower.

Make it faster

Eliminate any element of the application that is not helping. Be ruthless.

**Learnability**

Ideally, products would have no learning curve: users would walk up to them for the very first time and achieve instant mastery. In practice, all applications and services, no matter how simple, will display a learning curve.

Limit the Trade-Offs.

Usability and learnability are not mutually exclusive. First, decide which is the most important; then attack both with vigor. Ease of learning automatically coming at the expense of ease of use is a myth.

<b>Methaphors</b>	<p>Choose metaphors well, metaphors that will enable users to instantly grasp the finest details of the conceptual model.</p> <p>Good metaphors are stories, creating visible pictures in the mind.</p> <p>Bring metaphors alive by appealing to people’s perceptions–sight, sound, touch, and kinesthesia–as well as triggering their memories.</p> <p>Metaphors usually evoke the familiar, but often add a new twist. For example, Windows 95 has an object called a briefcase. Like a real-world briefcase, its purpose is to help make electronic documents more portable. It does so, however, not by acting as a transport mechanism, but as a synchronizer: Documents in the desktop briefcase and the briefcase held on portable media are updated automatically when the portable media is inserted in the machine.</p>
<b>Protect Users' Work</b>	<p>Ensure that users never lose their work as a result of error on their part, the vagaries of Internet transmission, or any other reason other than the completely unavoidable, such as sudden loss of power to the client computer.</p> <p>(Even here, it has become completely inexcusable that today's computers and operating systems do not support and encourage continuous-save. That, coupled with a small amount of power-protected memory could eliminate the embarrassment of \$5000 machines offering the reliability of 10-cent toys.)</p>
<b>Readability</b>	<p>Text that must be read should have high contrast. Favor black text on white or pale yellow backgrounds. Avoid gray backgrounds.</p> <p>Use font sizes that are large enough to be readable on standard monitors. Favor particularly large characters for the actual data you intend to display, as opposed to labels and instructions. For example, the label, "Last Name," can afford to be somewhat small. Habitual users will learn that that two-word gray blob says "Last Name." Even new users, based on the context of the form on which it appears, will have a pretty good guess that it says "Last Name." The actual last name entered/displayed, however, must be clearly readable. This becomes even more important for numbers. Human languages are highly redundant, enabling people to "heal" garbled messages. Numbers, however, unless they follow a very strict protocol, have no redundancy, so people need the ability to examine and comprehend every single character.</p> <p>Pay particular attention to the needs of older people. Presbyopia, the condition of hardened, less flexible lenses, coupled with reduced light transmission into the eye, affects most people over age 45. Do not trust your young eyes to make size and contrast decisions.</p>
<b>Track State</b>	<p>Because many of our browser-based products exist in a stateless environment, we have the responsibility to track state as needed.</p> <p>We may need to know:</p>

Whether this is the first time the user has been in the system  
Where the user is  
Where the user is going  
Where the user has been during this session  
Where the user was when they left off in the last session

and myriad other details.

In addition to simply knowing where they've been, we can also make good use of what they've done.

State information should be held in a cookie on the client machine during a session with a transaction service, then stored on the server when they log off.

Users should be able to log off at work, go home, and take up exactly where they left off.

A private service for doctors, Physicians On Line, does an excellent job with this. Doctors can be 95% of the way through a complex transaction, log off, log in again six weeks later from another part of the world, and the service will ask them if they want to be taken right back to where they were.

Avoid invisible navigation.

**Visible  
Navigation**

Most users cannot and will not build elaborate mental maps and will become lost or tired if expected to do so.

The World Wide Web, for all its pretty screens and fancy buttons, is, in effect, an invisible navigation space. True, you can always see the specific page you are on, but you cannot see anything of the vast space between pages. Once users reach our applications, we must take care to reduce navigation to a minimum and make that navigation that is left clear and natural. Present the illusion that users are always in the same place, with the work brought to them. This not only eliminates the need for maps and other navigational aids, it offers users a greater sense of mastery and autonomy.

As with the inherent statelessness of the web (see Track State, above), our job is not to accept blindly what the architects have given us, but to add the layers of capability and protection that users want and need. That the web's navigation is inherently invisible is a challenge, not an inevitability.