

Bijlage 7 - 10 UX principes van Jacob Nielsen

Voor PEP 3.0 is een goede UX/UI essentieel. PEP hanteert daarom als basis de 10 UX principes van Nielsen, zoals hieronder verder toegelicht. Om een goed beeld te krijgen van de toepassing van deze principes in het product dat Leverancier aanbiedt als oplossing voor PEP 3.0 en hierop een beoordeling te kunnen doen, dient Leverancier de gevraagde voorbeelden te leveren middels een schermopname ("screen recording"). Indien van een principe geen voorbeeld kan worden aangeleverd dient toegelicht te worden waarom dit het geval is.

1	Laat zien wat er gebeurt. Zichtbaarheid van de systeem status. Een systeem moet gebruikers op de hoogte houden van wat er aan het gebeuren is. <i>B.v. laat je bezoeker altijd weten waarom hij bijvoorbeeld moet wachten en hoe lang het gaat duren (b.v. het tonen van wachtanimatie tijdens het laden van inhoud). Denk ook aan het aantal stappen dat de bezoeker nog moet doorlopen. Laat bij het proces een stappenplan zien, hoe ver de gebruiker is en hoeveel stappen hij nog moet doen.</i>	Laat zien hoe de gebruiker een account aanmaakt.
2	Overeenkomst met de reële wereld Spreek dezelfde taal en beeldtaal als je bezoeker. Gebruik termen die hij ook gebruikt en afbeeldingen en icoontjes waaraan men is gewend. Dit is belangrijk voor de beleving van de bezoeker. <i>B.v. het 'bureaublad' als digitale werkomgeving.</i>	Laat zien hoe dit principe is toegepast in de applicatie die jullie al hebben ontwikkeld.
3	Gebruikerscontrole en vrijheid. Zorg dat je bezoeker beslissingen ongedaan kan maken. De website moet de bezoeker altijd een gemakkelijke uitweg bieden. Dit geeft meer vertrouwen en meer zekerheid. <i>B.v. een undo en redo functie.</i>	Geef een voorbeeld van een functionaliteit in de applicatie om handelingen of beslissingen ongedaan te maken.
4	Consistentie en standaardisering Gebruik elementen en acties waar je bezoeker aan gewend is. Dat kunnen bepaalde woorden zijn, maar ook waar je bepaalde functies en lay-out van een pagina plaatst. Bepaalde onderdelen worden altijd verwacht door bezoekers. <i>B.v. een logo, een zoekfunctie, taalgebruik.</i>	Laat de locatie van de zoekfunctie zien binnen minimaal 3 verschillende onderdelen van de applicatie.

5	Preventie van fouten Wanneer een bezoeker iets ziet wat niet klopt, bestaat de kans dat de pagina wordt weggeklikt. Fouten zijn belangrijk om te voorkomen. Het aangeven van een foutmelding is altijd snel gedaan, maar als de bezoeker niet weet wat hij fout heeft gedaan zal hij de ervaring als vervelend ervaren en je pagina verlaten. Je probeert de gebruiker te behoeden van eventuele fouten en te helpen met het aangeven. Dit geeft een gevoel van zekerheid. <i>B.v. als je een wachtwoord wilt veranderen, dan moet deze tweemaal ingevuld worden om typefouten te voorkomen. Je wachtwoord is dan zeker goed ingevuld. Of als de combinatie van 2 specifieke acties een error geeft, zorg dan dat deze acties niet gecombineerd kunnen worden.</i>	Laat zien welke oplossing er wordt geboden voor de gebruiker ter voorkoming van een 'dubbele registratie' (nieuwe gebruiker maakt account aan met gegevens van een account dat al in het systeem voorkomt).
6	Bereikbaarheid van informatie Zorg dat alle functionaliteiten en instructies binnen bereik zijn op de plaatsen waar je ze nodig hebt zodat de gebruiker tijdens een proces niets moet onthouden of onnodig moet terugkeren. Zorg dat je website de dingen onthoudt die belangrijk zijn, of vermeld de informatie gewoon nog een keer waar nodig is. <i>B.v. zorg dat de zoekvraag getoond wordt bij het zoekresultaat.</i>	Laat zien hoe de applicatie gebruikers (organisaties) faciliteert bij het plaatsen van een vacature.
7	Flexibiliteit en efficiëntie Nieuwe bezoekers lezen eerder de uitleg dan een ervaren bezoeker. Je website moet afgestemd zijn op je bezoeker. Zorg dat je systeem bruikbaar is voor ervaren én onervaren gebruikers. Laat gebruikers toe om acties die veel uitgevoerd worden, snel te bereiken. <i>B.v. shortcodes zorgen ervoor dat ervaren gebruikers sneller kunnen werken zonder dat onervaren gebruikers hier hinder van ondervinden.</i>	Laat zien dat de applicatie onderscheid maakt tussen nieuwe bezoekers en ervaren bezoekers.
8	Minimalistisch ontwerp Elk woord of elke functie die niet essentieel is, staat in concurrentie met essentiële informatie. Zorg ervoor dat je bezoeker zijn taak op je website kan uitvoeren. Door eenvoud weet de gebruiker hoe hij te	Laat zien hoe een dashboard van een gebruiker eruit ziet.

	werk moet gaan. Dit geldt zowel voor functionaliteiten als teksten als de vormgeving. <i>B.v. een welkom-boodschap is niet essentieel, en staat in concurrentie met het feit dat je een gebruiker snel de weg wil wijzen naar de functie waarvoor hij/zij de interface wil gebruiken.</i>	
9	Gebruikers helpen met fouten Maak van foutmeldingen netjes een melding. Vertel wat er aan de hand is. Gebruik tekst in plaats van codes bij foutmeldingen. Wijs op het specifieke probleem en stel een oplossing voor. <i>B.v. zorg ervoor dat een gebruiker weet of zijn/haar paswoord OF de gebruikersnaam fout is. Als het paswoord fout is, geef dan meteen de optie om een nieuw aan te maken.</i>	Geef een voorbeeld van een foutmelding als de combinatie van e-mailadres en wachtwoord niet klopt bij inloggen.
10	Helpfunctie en documentatie Een systeem zou bruikbaar moeten zijn zonder documentatie en zonder hulp. Bied dus alleen extra hulp als het nodig is. Het is eigenlijk een extra service. Wel geeft het voorzien van helpfuncties vertrouwen in het systeem. Zorg ervoor dat gebruikers gemakkelijk de weg vinden in en naar de documentatie. <i>B.v. contextgevoelige help geeft gebruikers op de plaats waar ze zich bevinden relevante tips zonder dat er gezocht moet worden in een handleiding.</i>	Geef een voorbeeld van een toepassing van een contextgevoelige helpfunctie binnen de applicatie.