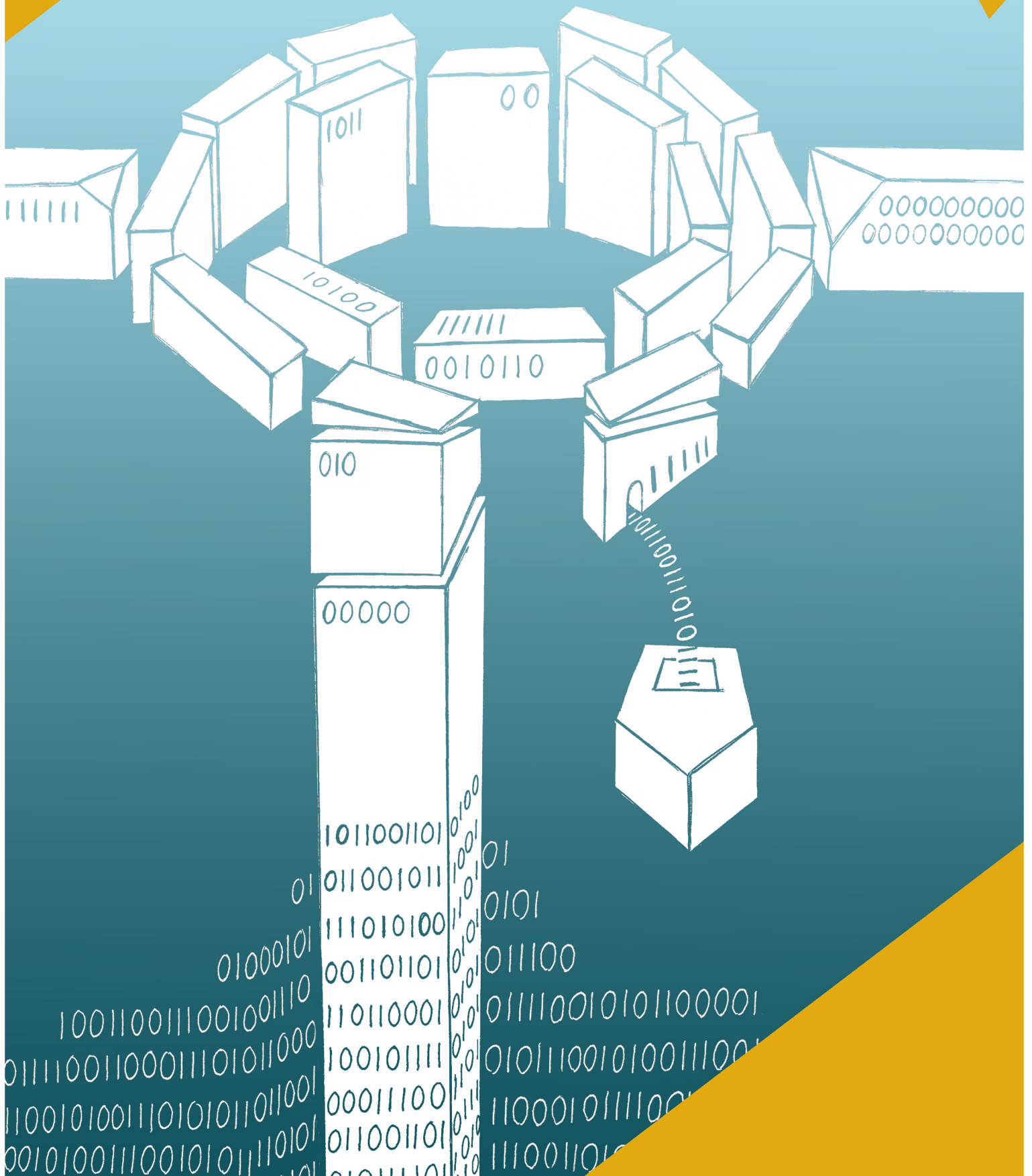


Digitale Architectuur

Bas Elbers



Radboud Universiteit



BACHELORSCHRIJF

De menselijke maat in een zich ontwikkellende techniek

Auteur:
Bas ELBERS

Begeleider:
prof. dr. ir. Th.P. (Theo) van
der WEIDE

21 januari 2015

Inhoudsopgave

1	Inleiding	4
2	De organisatie en de mens daarbinnen	5
2.1	Het belang van de organisatie binnen de IT	5
2.2	Verdeling van taken binnen een organisatie	6
2.3	Middelen om de rol van de werknemer te vervullen	9
2.4	Context waarbinnen een werknemer zijn of haar rol vervult	11
2.5	De gevolgen van evolutie binnen een organisatie	12
3	Koppeling mens met techniek	15
3.1	Menselijke eigenschappen	15
3.1.1	Waarom wordt er naar de eigenschappen van een mens gekeken? .	15
3.1.2	Cognitieve eigenschappen	16
3.2	Interactie mens en computer	17
3.2.1	Werkomgeving versus buitenwereld	17
3.2.2	Opbrengst en bedieningsgemak van het systeem	18
3.2.3	De verschillende typen interactie	19
3.2.4	Reductie in complexiteit tussen systeem en gebruiker	20
3.2.5	Bestaat de perfecte interface?	23
3.3	Principes	24
3.3.1	Aanpassing aan de gebruiker	24
3.3.2	Focus van een systeem	25
3.3.3	Waarden van de mens	27
3.3.4	Specifieke eisen door gebruik van principes	28
4	Conclusie	29
5	Bibliografie	31

1 Inleiding

Als Informatiekundige is het de bedoeling dat je een expert wordt op het gebied van systemen. Kennis over systemen is daarbij cruciaal. Je moet systemen analyseren om de overkoepelende architectuur bloot te leggen. Je leert over logica, je leert over intelligente systemen, maar je leert bovendien over mensen. Het doel van IT is immers om de mens en de omgeving van dienst te zijn. De manier waarop de IT de mens en omgeving van dienst is verandert en wordt daarnaast steeds ingrijpender.

Bij het gebruik van systemen is er een groep die over het algemeen het belangrijkste is. Dat is namelijk de groep gebruikers. Doordat gebruikers behoeften hebben die nog niet of niet genoeg zijn beantwoord ontstaan nieuwe of aangepaste systemen. De mening van de auteur is het volgende: ondanks dat de gebruiker centraal staat in het gebruik van systemen wordt er te weinig gebruik gemaakt van zijn of haar expertise. Daarnaast kan er veel meer worden gedaan om de gebruiker te ondersteunen bij het gebruik van systemen.

Het onderwerp van deze scriptie is dan ook de menselijke maat in een zich ontwikkelende techniek. Daarbij wordt onderzocht hoe het mogelijk is om ervoor te zorgen dat de gebruiker zich niet moet aanpassen aan het systeem, maar dat het systeem zich aanpast aan de gebruiker. Gebruikers klagen dat het systeem niet goed aansluit en ontwikkelaars klagen dat de gebruiker niet begrijpt hoe er met het systeem moet worden omgegaan. Om te onderzoeken hoe een systeem zich aan de gebruiker kan aanpassen moet eerst worden onderzocht wie de gebruiker is en wat de context is waarbinnen een gebruiker met het systeem interacteert. Alleen als je de gebruiker begrijpt kan je kijken hoe het systeem zich kan aanpassen. Manieren waarop een systeem zich aan de gebruiker kan aanpassen komen in deze scriptie dan ook aan bod. Daarnaast wordt besproken tot op welke hoogte de mens kan worden ondersteund.

We zullen zien dat organisaties een belangrijke rol spelen met betrekking tot systemen. De reden hiervoor wordt in hoofdstuk 2 eerst besproken. Daarbij komen de taken van de werknemers en de ondersteuning aan bod. Verder wordt de invloed van de persoonlijke eigenschappen van werknemers binnen de IT beschreven. In hoofdstuk 2 wordt ook ingegaan op de evolutie van een organisatie. In hoofdstuk 3 wordt het begrip van gebruiker als werknemer losgelaten. In plaats daarvan wordt de gebruiker meer als mens benaderd waarbij de techniek kan helpen. De menselijke eigenschappen komen eerst aan bod, waarnaar er wordt ingegaan op de interactie tussen mens en systeem. Hoofdstuk 3 wordt afgesloten met een aantal principes waar een ontwikkelaar rekening mee kan houden bij het maken van een systeem. Tot slot volgt de conclusie in hoofdstuk 4.

2 De organisatie en de mens daarbinnen

2.1 Het belang van de organisatie binnen de IT

Software wordt voor diverse doelen gebruikt. Een van de meest toonaangevende voorbeelden waarbinnen software wordt gebruikt is een organisatie. Het begrip organisatie wordt eerst uitgelegd: wat is een organisatie precies en waarom wordt hier aandacht aan besteedt? De precieze definitie van een organisatie is door verschillende interpretaties en invalshoeken niet eenduidig samen te vatten. Het belangrijkste aan een organisatie is¹ het feit dat het een verband is van mensen die kennis en vaardigheden bundelen om een doel te bereiken of om in een bepaalde behoefte te voorzien. Om dit te realiseren gebruikt men onder andere systemen.

Organisaties zijn er in verschillende soorten en maten, met elk een ander doel voor ogen. Toch zijn ze in twee soorten op te delen. De ene organisatie is gericht op de vervaardiging van fysieke producten en de ander op het leveren van diensten. De onderliggende informatieverwerking is van belang voor deze scriptie. De informatie die verwerkt wordt bevat bijvoorbeeld de status van de koppeling tussen de interne en externe wereld.

Het verzamelen en verwerken van informatie resulteert in kennis. Deze kennis is nodig als een organisatie wil overleven en de slag met de concurrentie niet wil verliezen. *'Companies who are successful are those who keep up with the change by creating new knowledge and disseminate it throughout the organization at greater speed than competitors and include that in new product developments.'* (de Klerk, 2012)[4] De kennis die wordt vergaard kan gaan over de concurrentiepositie, de toekomst of de soort klant. Het verkrijgen en behouden van kennis is nodig voor de continuïteit en het optimaal functioneren van de organisatie.

Wat een organisatie interessant maakt is het feit dat de systemen vaak meer zijn dan losstaande applicaties. De systemen zijn in meer of mindere mate met elkaar verbonden en moeten onderling communiceren. Het afstemmen van systemen kan problemen opleveren, zeker naar mate de organisatie evolueert en er steeds meer systemen bijkomen of veranderen. *'...architecture is a system's organizational structure [10, 11]. Some organizations allow that structure to evolve unintentionally or through neglect; others focus on designing or deriving it by following a planned process. However, allowing systems to evolve haphazardly often results in failure.'* (McBride, 2007)[10]

¹Belangrijk voor de scope van deze scriptie, niet voor de 'officiële' definitie.

Bij het lezen van het bovenstaande citaat is de gedachte die gelijk naar boven schiet die van een bestemmingsplan voor een gemeente of stad. Die veranderen of groeien ook ‘van-zelf’. De transformatie moet echter in goede banen worden geleid, goede infrastructuur is vereist. Deze (fysieke) infrastructuur is vergelijkbaar met de digitale infrastructuur zoals die van een organisatie. Aanpassingen of vernieuwingen moeten doorgerekend worden op basis van de huidige, goed vastgelegde, situatie. De verandering moet in lijn zijn met een beoogd doel die hopelijk beschreven is in de Mission/Vision statement. Conflicten ontstaan bij een ontbrekende of onvolledige vastlegging van de huidige situatie, een onvolledig plan of door onvoorziene interconnectiviteit.

2.2 Verdeling van taken binnen een organisatie

‘Humans use technologies not for the sake of those technologies, but to support tasks that are relevant or meaningful to their jobs or personal needs. In addition, people carry out tasks in settings or contexts that impose constraints on doing and completing the tasks. Four contexts are identified: group, organizational, social, and global.’ (Zhang en anderen, 2007)[20]

Zoals gelezen in sectie 2.1 is het de taak van de werknemers om informatie te verwerken (in kennis) en worden systemen gebruikt ter ondersteuning hiervan. Voordat verder gekeken kan worden op welke manier het systeem de werknemer ondersteunt in zijn taken moet eerst gekeken worden wat de taken precies zijn. Bij de verschillende taken horen namelijk verschillende soorten ondersteuning. Een manager wil bijvoorbeeld weten hoe effectief en efficiënt iemand met een systeem werkt, terwijl een werknemer meer geïnteresseerd is in hoe goed het systeem aansluit bij zijn wensen en behoeften. Deze tweedeling in belangen bij het gebruik van systemen is typerend. Vandaar dat dit principe een naam heeft gekregen: *prestatiedichotomie*².

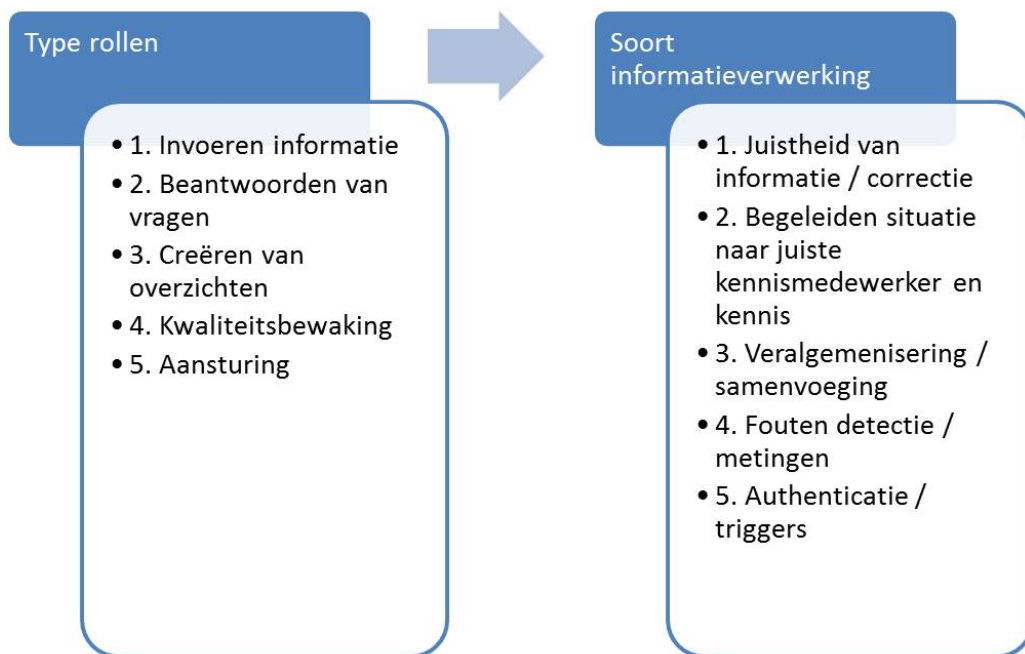
Om ervoor te zorgen dat de systemen geaccepteerd en gebruikt worden moeten deze goed aansluiten bij de individuele taken van de werknemer. Iets wat niet altijd wordt begrepen. *‘Organizations that spend millions of dollars on information technologies (IT) are primarily concerned with how their investments will influence organizational and individual performance (Torkzadeh et al. 1999). However, the expected productivity gains and organizational benefits delivered by IT cannot be realized unless IT is actually accepted and used (Hackbarth et al. 2003).’ (Zhang en anderen, 2007)[20]* In deze scriptie gaat het met name om het benadrukken van de menselijke kant bij het werken met systemen, ofwel de prestatie van het systeem ten opzichte van de gebruiker (later ook welzijn/bedieningsgemak genoemd).

Een tweede principe ontstaat doordat organisaties een beperkte visie kunnen hebben. Vanwege sub-optimalisatie is de organisatie niet bezig met wat belangrijk is voor het geheel. Die sub-optimalisatie ontstaat met name doordat een te grote focus wordt gelegd

²Prestatie van de gebruiker ten opzichte van het systeem en het systeem ten opzichte van de gebruiker.

op de prestatie van een afdeling. Dit is bijvoorbeeld het geval wanneer verdeling van hulpmiddelen³ plaatsvindt. Terwijl een juiste en rechtmatige verdeling de organisatie meer profijt biedt, wil elke afdeling voor zijn eigen prestatie zoveel mogelijk ontvangen. Iemand moet die verdeling maken, bij het maken van een systeem is dat de architect. Later in deze sectie zal blijken dat de architect in een lastige positie kan worden gedrukt wanneer alleen hij/zij degene is die het grotere plaatje ziet.

Zoals aangegeven in sectie 2.1 richten we ons in deze scriptie op taken die te maken hebben met informatieverwerking. In figuur 2.1 staat een overzicht van de type rollen die in die taken naar voren (kunnen) komen. Daarbij staat een eerste indruk van de consequenties voor de informatieverwerking bij het type rol. Elk type informatieverwerking zorgt voor verschillende principes, per rol zijn andere principes belangrijk. Deze principes kunnen met elkaar botsen. Het is aan de ontwikkelaar om de principes te identificeren en een goede afstemming voor de taken en rollen te vinden.



Figuur 2.1: Eén op één relatie tussen de type rol en de soort informatieverwerking.

Het is dus belangrijk om de verschillende taken van de werknemers te weten. Bij het ontbreken van ondersteuning voor deze taken kunnen de gevolgen vrij ernstig zijn. Nu komt een aantal mogelijke gevolgen aan bod.

Het opleggen van een systeem van bovenaf, dat is wat je vaak hoort als er een nieuw systeem wordt geïntroduceerd. “Wij moeten...”. RU-medewerkers die met BlackBoard

³Een hulpmiddel kan in deze context de functionaliteiten van een systeem zijn.

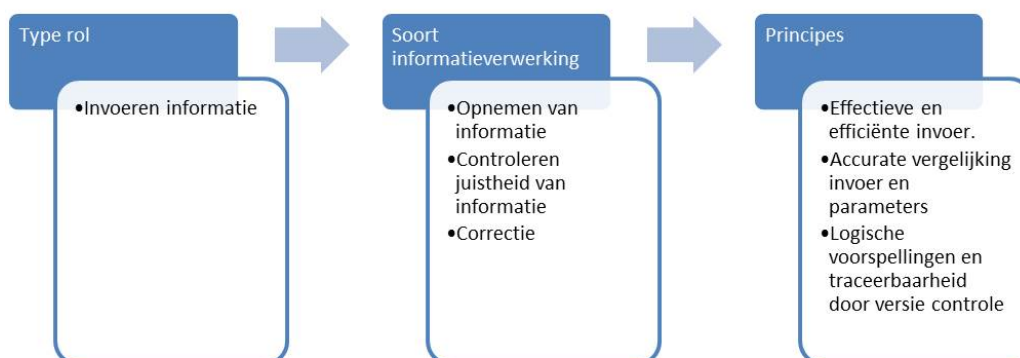
moeten werken of artsen met het EPD⁴. Elke medewerker heeft eigen ideeën bij het nieuwe systeem. Als het systeem niet de functionaliteiten heeft die je als werknemer voor ogen had kan dit gevolgen hebben. De wijze waarop het systeem tot stand komt en hoe erover wordt gecommuniceerd beïnvloedt de gevolgen. Elke ontwikkelaar moet concessies doen. Belangrijk is dat de mensen die met het systeem werken de concessies begrijpen en accepteren. Voor begrip en acceptatie is namelijk betrokkenheid nodig.

Als gebruikers ideeën hebben, maar deze niet mogen uiten, niet serieus genomen worden of de ideeën worden niet netjes afgehandeld, dan is er bij voorbaat al een negatieve kijk op het nieuwe systeem. Het is voor een werknemer ergerlijk als er wel een ideeënbus is, maar op een verzoek nooit wordt gereageerd. Als een voorstel technisch niet haalbaar is dan hoeft dit niet erg te zijn. Door een persoonlijke reactie ontstaat er begrip over de afwijzing.

Een voorbeeld van een gebrek aan begrip zijn de artsen die met het EPD moesten gaan werken. Zij kregen nooit een antwoord op hun verzoeken. Het gevolg daarvan was dat artsen elkaar hetzelfde verzoek lieten doen waardoor het belangrijker moest lijken. De architect die de verzoeken moest afhandelen had door het grote aantal verzoeken niet genoeg tijd om erop te reageren en nam ze bovendien niet meer serieus. Zoals eerder is uitgelegd hadden de werknemers hier niet door dat ze de functionaliteiten onder de afdelingen moeten verdelen. De architect moet selectief zijn zodat alle cruciale elementen voor de verschillende afdelingen in elk geval in het systeem komen. Doordat de terugkoppeling van de architect naar de artsen ontbrak was er geen plaats voor begrip en acceptatie en ontstond er onvrede bij de artsen.

BlackBoard kan ook als voorbeeld worden gebruik. Het wordt door docenten het liefst vermeden omdat het invoeren van gegevens onnodig arbeidsintensief is. Door ontoereikende ondersteuning ontstaat een wildgroei van alternatieve systemen die hetzelfde doen, maar dan sneller en gebruiksvriendelijker. De oplossing van de leidinggevenden is om de alternatieve systemen te verbieden i.p.v. de essentie hiervan op te nemen en het huidige systeem aan te passen. Dat is in de opleverfase namelijk te duur. Door het ontbreken van zeggenschap van gebruikers in het nieuwe systeem sluit het niet aan bij de verschillende taken van de werknemers. In figuur 2.2 volgt een verder uitgewerkt diagram met de uitvergroete principes van de docent die resultaten van de studenten wil invoeren in BlackBoard. Dit is alvast een voorproefje voor wat in het volgende onderdeel naar voren komt, namelijk wat er voor de werknemer nodig is om zijn of haar rol goed te kunnen vervullen.

⁴Elektronisch patiëntendossier



Figuur 2.2: Voorbeeld van enkele principes voor een docent die gegevens invoert over de resultaten van een student.

2.3 Middelen om de rol van de werknemer te vervullen

In de vorige sectie is gezien dat er bij elk (type) gebruiker en systeem principes kunnen worden aangeduid. Het vertalen van deze principes naar concrete systeemeigenschappen is een taak van de ontwikkelaar. Doordat er een enorm aantal principes zijn waar potentieel rekening mee gehouden moet worden zal eerst gekeken worden naar de situaties die een werknemer tegen kan komen. In de sectie die hierop volgt zal er worden gekeken hoe de context van de gebruiker verschilt in het gebruik. Daarnaast komt aan bod hoe het systeem en de hulpmiddelen kunnen helpen door zich aan te passen aan de mens.

Een werknemer die een bepaalde rol in een organisatie speelt moet met verschillende scenario's om kunnen gaan. De mogelijke scenario's kunnen worden verdeeld in drie categorieën. De eerste en voornaamste categorie is die van de standaard procedure (ook wel happy flow genoemd). Als we het voorbeeld nemen van een navigatie systeem dan is de standaard procedure om met de auto vanaf de huidige locatie naar locatie B te verplaatsen. De tweede categorie is een non-standaard categorie, in dit geval de alternatieve paden. Een alternatief pad kan zijn dat je niet met de auto van locatie A naar B wilt, maar met de fiets. In de meeste gevallen bestaat er nog een derde categorie, de uitzonderingen. Dit is ook een non-standaard categorie. Een uitzondering bij het gebruik van een navigatie systeem kan zijn dat je onmogelijk van A naar B kan komen.

Waarom wordt nu onderscheid in scenario's gemaakt? Het is belangrijk dat het onderscheid wordt gemaakt omdat de representatie in het systeem overeen moet komen met de werkelijkheid. Problemen vallen veelal in de non-standaard categorie. Dit laatste is vrij evident. Situaties die niet tot nauwelijks voorkomen zijn moeilijker vast te leggen

en worden vaak als minder belangrijk aangeduid. Vandaar dat het in de informatiewereld ook gebruikelijk is om verschillende test fases te gebruiken bij het identificeren van non-standaard gevallen. Wat de gebruiker bij non-standaard gevallen nodig heeft, is een systeem dat deze gevallen herkent, ondersteuning biedt en ook een mogelijkheid tot educatie geeft.

Hoe er met uitzonderlijke gevallen wordt omgegaan ligt voor een groot deel aan de context. Bij het besturen van een vliegtuig wordt veel aandacht besteedt aan het omgaan met uitzonderingen. De gevolgen van deze uitzonderingen kunnen desastreus zijn. Bekende uitzonderingen kunnen door de werknemer worden geleerd, mogelijkheid tot (zelf)educatie is daarbij het voornaamste. Het systeem kan helpen met een help functie of een eigen probleem analyse. Wanneer uitzonderingen echter niet bekend zijn bij het systeem wordt het gebruik hiervan problematisch. Dan ligt het meest voor de hand om bij een volgende update van het systeem het probleem en de mogelijke oplossingen hierin op te nemen. Het bieden van een mogelijkheid tot het geven van feedback en foutrapportages is van groot belang. De uitzondering wordt anders nooit opgemerkt en erkend.

'The scope of exceptions covers use problems generated by: 1) erroneous or incomplete information input, 2) requests to deviate from standard procedures, or 3) situations that the system was never designed to handle (Strong 199 5). Such exceptions are normally caused by three conditions: 1) operating errors (user error), 2) design errors (design flaw or a missed requirement), and 3) uncontrolled change inherent in dynamic organizations.' (McGann, Lyytinen, 2005)[11]

Dat een systeem overeen moet komen met de werkelijkheid kan het beste worden uitgelegd met een voorbeeld. Als gekeken wordt naar de beveiliging van een systeem en wat daarbij nodig is, dan zie je dat een werknemer meer nodig heeft dan alleen een waterdicht systeem. De manier waarop een werknemer met het systeem omgaat heeft ook gevolgen voor de veiligheid. Het systeem moet op een realistisch manier aansluiten op de behoeftigheden voor een veilige situatie, daarbij wel rekening houdende met de gebruikers en de omgeving. *'Stanton and colleagues defined behavioral information security as the human actions that influence the availability, confidentiality, and integrity of information systems (Stanton et al. 2006). They noted that despite the multibillion dollars spent on information security by commercial, nonprofit, and governmental organizations around the world, the success of security appears to depend upon the behavior of the individuals involved (Stanton et al. 2006). Appropriate and constructive behavior by end users, system administrators, and others can enhance the effectiveness of information security while inappropriate and destructive behaviors can inhibit its effectiveness.'* (Zhang en anderen, 2007)[20]

Een typerend voorbeeld van de invloed van gedrag op de beveiliging is het plakken van Post-its op het beeldscherm met de gebruikersnaam en wachtwoord van het systeem. Dit kwam opvallend genoeg voor bij een bezoek aan het Radboud ziekenhuis. Wie dat ziet denkt gelijk aan hoe onprofessioneel het daar aan toe gaat, maar aan wie ligt dit nu werkelijk? Veel vaker moet je bij wachtwoorden zowel kleine letters, hoofdletters, een

cijfer en een speciaal teken gebruiken dat elke 3 maanden vervangen moet worden door een nieuw, ongebruikt, wachtwoord.

Als je bedenkt hoeveel wachtwoorden een gemiddeld persoon moet onthouden is het ineens niet meer zo gek dat deze af en toe worden opgeschreven. Hetgeen wat je aan de gebruiker vraagt moet realistisch zijn en passen bij de hoeveelheid veiligheid die nodig is. Je moet dus niet, zoals de eerste versies van Windows Vista, elke keer aan een gebruiker vragen of het wel veilig is om de Windows verkenner te openen. Dat gaat zijn doel voorbij.

2.4 Context waarbinnen een werknemer zijn of haar rol vervult

Ieder mens is anders. Wat kan een systeem doen om te helpen? Om een systeem zich aan te laten passen aan de gebruiker moet er verder worden gekeken dan de type rol en taak van de werknemer. Elke werknemer bevindt zich namelijk in een bepaalde context. Verschillende eigenschappen hebben invloed op de werknemer. Hieronder verstaan we o.a. achtergrond, leeftijd en opleiding. Het is van belang om in de soort gebruiker te verdiepen en een onderscheid te maken tussen de werknemers.

Elke gebruiker leeft bijvoorbeeld binnen een eigen cultuur. Onder die cultuur verstaan we het nationaal cultuur zoals Westers of Aziatisch, maar ook de diverse subculturen binnen een gemeenschap of bedrijf. De gebruiker wordt beïnvloed door de normen en waarden die heersen binnen de verschillende culturen. Als je alle factoren die de gebruiker beïnvloedt verzamelt en benoemt krijg je uiteindelijk de context waarbinnen de gebruiker het systeem gebruikt. Vanwege deze context kan het systeem binnen de ene organisatie perfect werken, maar ergens anders falen. Het is vooral belangrijk om grote verschillen tussen contexten te benoemen en op te vangen, hoewel die informatie diep verborgen kan zijn.

Domeinkennis van beide (sub)culturen is nodig wanneer sprake is van een transformatie van een systeem van de ene plek naar een ander. Multinationals werken met systemen die overal ter wereld gebruikt moeten worden. Het enkel inschakelen van een vertaler is bij een migratie dan niet genoeg. Dit is vergelijkbaar met het vertalen van spreekwoorden. Je kan spreekwoorden niet letterlijk vertalen. Je hebt de algemene kennis van beide talen nodig om iets soortgelijks te produceren. Zo wordt ‘een ongeluk komt zelden alleen’ vertaald naar ‘it never rains, it pours’. Dit kun je niet weten als je de context niet kent, hetzelfde geldt voor systemen.

Het verschil in context komt terug in het onderzoek naar waarden tussen Amerikanen en niet-Amerikanen (Tully, 1989).[\[16\]](#)

1) *‘The most important value for Americans was having challenging tasks to perform, but for non-Americans it was having an opportunity to make real contributions to the success of their company or organization.’*

Wat heeft dit nu te maken met het maken van een systeem? Het hoeft geen enorm verschil te betekenen, maar dit verschil in waarden kan terugkomen in de representatie van een opdracht. Bij Amerikanen laat het systeem wellicht zien wat de moeilijkheidsgraad van een opdracht is, terwijl het bij niet-Amerikanen laat het zien hoeveel waarde het bedrijf aan de opdracht hecht. Dit kleine verschil kan bijdragen aan de voldoening van een werknemer.

10) *‘Non-Americans felt that decisions should be made by the one who carries the responsibility while Americans felt that the decisions should be made by the person or persons with the most knowledge about the problem.’*

Als beslissingen gemaakt worden met behulp van een systeem kan een verschil in de omgang met beslissingen een impact hebben op de rechten in een systeem. Meestal zijn verantwoordelijkheden vastgelegd in een bepaalde hiërarchie. Wanneer beslissingen gemaakt worden door degene die de verantwoordelijkheid draagt, kan deze hiërarchie in het systeem vastgelegd worden. Wanneer beslissingen gebaseerd zijn op kennis kan in theorie elke werknemer de beslissing nemen. Een dynamisch systeem waarbij werknemers tijdelijk ergens bevoegd voor worden sluit dan beter aan. Een klein verschil in cultuur heeft dan grote gevolgen op het systeem.

Wat we uit deze voorbeelden kunnen leren is dat er bij de analyse van de context van de gebruiker(s) meer ontwerpprincipes ontstaan waar rekening gehouden mee moet worden. Deze kunnen dan weer gecombineerd worden met de principes die voortkwamen uit de type rol die de gebruiker speelt en de soort informatieverwerking daarbij nodig. Het resultaat is een gebruikersmodel waarbij rekening wordt gehouden met zowel de rollen en taken van de gebruikers als de context waarbinnen die rollen en taken worden gespeeld. Het enige wat daarbij nog ontbreekt zijn de algemene (cognitieve) eigenschappen van een mens bij het omgaan met een systeem. Dit wordt in hoofdstuk 3 besproken.

‘Systems often meet their technical ‘requirements’ but are considered to be a ‘failure’ because they do not deliver the expected support for the real work in the organisation. The source of the problem is that techno-centric approaches to systems design do not properly consider the complex relationships between the organisation, the people enacting business processes and the system that supports these processes (Norman, 1993; Goguen, 1999).’(Gordon Baxter, Ian Sommerville, 2011)[1]

2.5 De gevolgen van evolutie binnen een organisatie

In sectie 2.1 werd uitgelegd dat organisaties moeten oppassen bij de evolutie van een systeem. Systemen evolueren doordat deze met updates worden geoptimaliseerd, maar ook omdat de organisatie zelf verandert. Die verandering kan betekenen dat er wordt uitgebreid, ingekrompen, of dat er wordt verschoven⁵. Verandering gebeurt zowel bewust

⁵De verandering in een organisatie kan bijvoorbeeld betrekking hebben op een product/dienst, maar ook op een hele markt.

als onbewust, door noodzaak of mogelijkheden, zowel intern als extern, geleidelijk of abrupt en positief of negatief⁶. Als organisatie probeer je veranderingen enigszins te controleren, maar externe factoren vermoeilijken het om een accurate verwachting van de toekomst te hebben.

Zoals beschreven in sectie 2.1 is een van de grootste waarden van een (informatie verwerkend) bedrijf in veel gevallen de kennis. Kennis over het bedrijf, de klanten, de competitie, noem het maar op. Kennis over het systeem en hoe het werkt hoort hier ook bij. Iets wat bij reorganisaties nogal eens over het hoofd wordt gezien. Zeker specifieke, op maat gemaakte, systemen kunnen veel kennis vereisen om mee te werken. Kennis die vaak maar voor een klein deel is gedocumenteerd. De ongedocumenteerde kennis kan in meer of mindere mate worden overgedragen, maar dat levert vaak problemen op. Niet alleen kost overdragen veel geld en tijd, er is ook kans dat werknemers de boel saboteren. *'Due to the high level of turnover in the labour market, organisations sometimes have to face the fact that senior managers will refuse to transfer knowledge to newcomers.'* (Urbancová, Linhartová, 2012)[17] De reden voor het weigeren van overdracht van kennis ligt aan het simpele feit dat mensen niet graag worden vervangen. Waarom moet je anders al je kennis aan iemand overdragen?

'In today's organisations, it is not unusual to find that only a small percentage of the knowledge which enables an organisation to function is documented. The rest, whether it be information or process, resides in the heads of employees, contractors or consultants... In today's working environment of high staff turnover, the capture of this knowledge becomes increasingly important. Too many organisations have seen their corporate memory disappearing as people walk out of the door.' (Vitlin, 2001)[18]

De manier waarop een systeem evolueert draagt veel bij aan de acceptatie van het systeem. Het is belangrijk om te weten dat er over het algemeen bij gebruikers terughoudendheid is ten opzichte van veranderingen. De eerste plek waaraan je verandering merkt is vaak het systeem dat wordt gebruikt. Bij verandering zijn er verschillende redenen waarom we bang of onwillend zijn. Een aantal wordt hier besproken.

Veranderingen betekenen onzekerheid. Onzeker of het systeem wel werkt, of banen blijven behouden en onzeker of het nieuwe systeem wel net zo goed aansluit als het oude, vertrouwde, systeem. Daarnaast zijn mensen nog steeds gewend aan bepaalde gewoontes. Deze gewoontes kunnen verstoord worden als de manier van werken verandert. Daar willen we over het algemeen niks van weten. Het huidige systeem werkt toch prima? Werknemers hebben over het algemeen geen visie voor de toekomst nodig als het gaat om het uitvoeren van hun taken. Ze zien het nut niet in van het nieuwe systeem.⁷

Je kunt mensen niet veranderen, je kunt de negatieve elementen voor een groot deel wel opvangen. Het doel van het nieuwe of gewijzigde systeem is om de taken van de werknemers te ondersteunen, met een visie op de langere termijn. Het huidige systeem

⁶Waarom een organisatie evolueert is voor deze scriptie niet van belang. Het gaat erom dat zowel de structuur als de systemen van een organisatie evolueren.

⁷<http://www.peterstark.com/why-employees-resist-change/>

heeft goede en slechte eigenschappen bij de ondersteuning van die taken. Het in kaart brengen hiervan levert inzichten op voor de ontwikkeling van het nieuwe of aangepaste systeem.

Bij het in kaart brengen van het huidige systeem wordt nogal eens vergeten dat de werknemers de experts zijn. Zij werken er immers iedere dag mee. Als ontwerper of ontwikkelaar kun je de negativiteit tegenover verandering doorbreken door de werknemer de kans te bieden om onderdeel te zijn van het ontwikkelingsproces. Er zijn verschillende manieren waarop je de gebruiker onderdeel kan laten zijn van het proces. Het gaat voor deze scriptie te ver om deze te benoemen, dit onderzoek gaat hier verder op in: (Harris, Weistroffer, 2008).[\[8\]](#)

3 Koppeling mens met techniek

3.1 Menselijke eigenschappen

3.1.1 Waarom wordt er naar de eigenschappen van een mens gekeken?

Limieten zorgen voor optimalisatie met een grens in efficiëntie. Roeien met de spaanders die je hebt. In de computerwereld is er in een vrij korte tijd een verandering ontstaan in wat de limiet is. In het begin werd er alles aan gedaan om de computer zo efficiënt mogelijk te laten lopen. Daar werd puur de focus op berekeningen gelegd. De snelheid en opslag van de computer had namelijk een limiet. Het gebruik was daarmee ondergeschikt aan de functionaliteit.

De techniek is zodanig verbeterd dat de snelheid niet meer de bottleneck is. Ja, er zijn nog altijd processen, zoals een brute-force met veel data. De snelheid kan dan niet hoog genoeg zijn. Voor de 'normale gebruiker'¹ is snelheid niet meer het probleem. De beperkende factor is daarmee veranderd van de computer naar de interactie tussen mens en computer².

Dat de beperkende factor nu de interactie is tussen mens en computer heeft een reden. Een mens kan maar een aantal processen of instanties tegelijkertijd bijhouden. Het aantal processen dat een alledaags systeem kan draaien is vele maler groter dan het aantal processen een gebruiker mentaal kan verwerken. Dit verschil wordt steeds groter omdat de mens relatief gezien niet sneller of beter wordt in het verwerken van informatie. Door middel van educatie en ontwikkeling kan het zo zijn dat we als mens steeds meer informatie kunnen verwerken, maar de techniek groeit veel sneller dan dat wij als mens ooit kunnen ontwikkelen.

Doordat de mens in de mens-computer interactie de zwakste schakel is, is het belangrijk om de gebruiker zo goed mogelijk te helpen in het verwerken van informatie. Om deze ondersteuning zo effectief en efficiënt mogelijk te laten verlopen moet de mens zelf goed worden begrepen. Kennis over bijvoorbeeld gedrag, gedachtes, limieten, mogelijkheden, diversiteit en meer geven de ontwikkelaar de kans om rekening te houden met bepaalde karakteristieke eigenschappen van een gebruiker. Het is voor deze scriptie ondoenlijk om al deze eigenschappen na te gaan, maar het is belangrijk om te beseffen dat de eigenschappen van mensen invloed kunnen hebben op het gebruik van een systeem.

¹Een normale gebruiker is bijvoorbeeld iemand die een PC of smartphone gebruikt voor alledaagse applicaties.

²<http://www.ent.mrt.ac.lk/~ekulasek/en577/hciihome.html>

3.1.2 Cognitieve eigenschappen

De begrippen EPIC³ en GOMS⁴-model komen veel naar voren bij het zoeken naar cognitieve eigenschappen die belangrijk zijn bij de interactie tussen mens en computer. Deze modellen van de mens als systeem staan echter ver weg van bruikbare design principes waar een ontwikkelaar rekening mee moet houden.

Het oorspronkelijke idee was om basale cognitieve eigenschappen om te zetten in concrete design principes. Er is echter teveel domeinkennis nodig om zowel de cognitieve eigenschappen te begrijpen als de transitie van die eigenschappen in design principes. Vandaar dat de focus meer ligt op de design principes. Deze zijn makkelijker te begrijpen en bevatten concrete aandachtspunten voor de ontwikkelaar.

Nielsen en Molich (1990)[13] en Nielsen (1994)[12] kwamen met tien design principes waar een user interface op gecontroleerd kan worden. Aan de hand van deze tien principes kan een ontwikkelaar zien waar de sterktepunten en zwaktepunten van de user interface liggen. Deze principes komen echter uit het categoriseren van algemene problemen die gepaard gaan met het ontwikkelen van user interfaces. De achterliggende oorzaak kan te maken hebben met cognitieve eigenschappen, maar de principes hoeven niet direct hieraan gerelateerd aan te zijn.

In 1996 kwam Jill Gerhardt-Powals[7] met een set van heuristieken die gebaseerd was op de cognitieve theorie. Deze principes richten zich vooral op de reductie van de cognitieve last bij het uitvoeren van een taak. Op basis van cognitieve theorie staan deze principes het dichtst bij de bruikbare eigenschappen die een ontwikkelaar kan gebruiken voor het ontwikkelen van user interfaces. In tabel 3.1 staan de 10 eigenschappen om een idee te krijgen waar op gelet moet worden.

Automate unwanted workload
Reduce uncertainty
Fuse data
Present new information with meaningful aids to interpretation
Use names that are conceptually related to function
Group data in consistently meaningful ways to decrease search time
Limit data-driven tasks
Include in the displays only that information needed by the user at a given time
Provide multiple coding of data when appropriate

Tabel 3.1: De 10 heuristische eigenschappen van Gerhardt-Powals

De primaire taak bij interactie tussen mens en computer is het verwerken van informatie. De manier van representatie van de informatie zorgt voor een bepaalde cognitieve last. Het gaat te ver om dit exact uit te leggen, maar het heeft te maken met je visuele

³Executive-Process/Interactive Control

⁴Goals, Operators, Methods en Selection rules

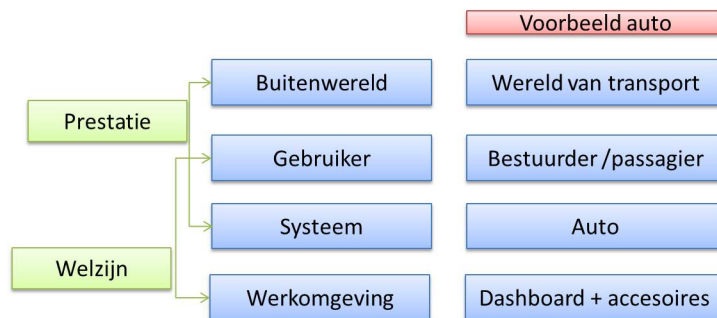
geheugen, je (korte termijn) werkgeheugen en de geheugen op lange termijn. Voor een uitgebreide uitleg over de werking hiervan wordt graag verwezen naar het artikel van Hollender en anderen(2010)[9].

3.2 Interactie mens en computer

3.2.1 Werkomgeving versus buitenwereld

We gaan nu verder in op de relatie tussen mens en systeem en de omgeving die daarbij hoort. De omgeving van een systeem kan worden opgedeeld in twee soorten, de werkomgeving en de buitenwereld. Dit onderscheid wordt gemaakt om een beter begrip te kweken in waar de aandacht (onterecht) op wordt gericht. Ondanks dat er kan worden beargumenteerd dat de werkomgeving een deelverzameling is van de buitenwereld is er vaak een verschil in zichtbaarheid en tastbaarheid tussen de twee.

De werkomgeving is de omgeving die direct betrokken is bij de gebruiker en de interactie. Bij een auto bijvoorbeeld is de werkomgeving het dashboard met de accessoires, dus alles wat met directe interactie te maken heeft. De buitenwereld is in dit geval zowel de rest van de auto als de wereld van transport (inclusief de wegen, andere weggebruikers etc.). Figuur 3.1 geeft hiervan een overzicht.



Figuur 3.1: Model van een systeem in de omgeving.

Wat kunnen we hier nu mee? Geconcludeerd wordt dat de werkomgeving zodanig moet worden afgesteld dat de gebruiker zo effectief en efficiënt mogelijk de beoogde taken kan uitvoeren. In een auto is de hoofdtaak het veilig (snel, comfortabel etc.) voortbewegen van A naar B. Naar mate de principes verder worden uitgediept ontstaan verdere specifieke eisen. De afstemming van de gebruiker op de werkomgeving zorgt voor het welzijn van de gebruiker. De afstemming van het systeem op de buitenwereld zorgt voor de prestatie.

Het probleem bij software is vaak dat er wel wordt gekeken naar de prestatie (opbrengst),

maar niet naar het welzijn (bedieningsgemak)⁵. Dit is eerder besproken in sectie 2.2, we noemden dit de *prestatiedichotomie*. Daarom is het belangrijk dat de gebruiker invloed heeft op het design van het systeem. Niet alleen de managers en ‘experts’ op dit gebied. De gebruiker is namelijk een expert op het gebied van welzijn, daarom geven J.Dul en anderen (2012) ook aan dat: *‘another potential value of HFE⁶ is that it ensures that system actors can influence system design.’*^[5]

3.2.2 Opbrengst en bedieningsgemak van het systeem

In sectie 3.2.1 is onderscheid gemaakt tussen de buitenwereld en werkomgeving. Nu kan er worden uitgelegd waarom er een verschil in tastbaarheid is tussen de twee. De reden waarom het een wel tastbaar is en de ander niet, is vanwege de mogelijkheid om het te meten. Wanneer je vergelijkt hoe een systeem zich gedraagt in de buitenwereld, oftewel hoe een auto zich gedraagt op de weg, kan van alles worden gemeten. Je kan namelijk meten hoe snel een auto rijdt, hoe snel een auto remt en hoeveel kilometer een auto kan rijden met 1 liter brandstof.

Aan de andere kant wil je meten hoe de gebruiker zich gedraagt in de werkomgeving, oftewel hoe de bestuurder achter het stuur zit. Het is dan veel moeilijker te meten hoe goed het display weergeeft wat de bestuurder nodig heeft. Ook hoe comfortabel de bestuurder zit en of deze niet onnodig wordt afgeleid van de weg is moeilijk te meten.

Het gedrag van de gebruiker in de werkomgeving is uiteindelijk meetbaar, maar het is niet eenduidig samen te vatten in een simpele relatie tussen A en B. Hier komt ook een andere discipline om de hoek kijken, namelijk (Human-Computer Interaction). Het gaat meer richting psychologie, hoe het systeem een invloed heeft op de mentale processen⁷. *‘Human-computer interaction (HCI) is a multidisciplinary field in which psychology and other social sciences unite with computer science and related technical fields with the goal of making computing systems that are both useful and usable.’* (M.Olsen, S.Olsen, 2003)^[14]

Het is belangrijk om de invloed van andere velden te erkennen, als klassieke Informaticus of Informatiekundige worden deze niet of nauwelijks geleerd. Erkennen dat je zelf niet alles weet of kunt bevatten is de eerste stap. Communiceren met lotgenoten, het inschakelen van domeinexperts en samen stilstaan bij mogelijke aannames zijn nodig in het proces om tot een beter geheel te komen.

Kader 3.2: Een kleine uitweiding over een expert zonder oogkleppen.

⁵Je kunt ook beargumenteren dat de afstemming van de gebruiker op de werkomgeving de prestatie van de interactie is.

⁶Human Factors Ergonomics

⁷De invloed van psychologie bij de meetbaarheid van de gebruiker in de werkomgeving is ook een van de redenen waarom we in sectie 3.1.2 naar de cognitieve eigenschappen hebben gekeken.

3.2.3 De verschillende typen interactie

Zichtbaar versus onzichtbaar

Wat in de bediening van het systeem belangrijk is, zijn de zichtbare en onzichtbare elementen voor de gebruiker. In het voorbeeld van de auto lijkt dit vrij evident, maar daar zijn ook een aantal subtiele kenmerken waar rekening mee moet worden gehouden. In zowel de werkomgeving als de buitenwereld zijn er elementen zichtbaar en onzichtbaar. Zo kan de bestuurder binnen de werkomgeving zien hoe hard hij rijdt en of de handrem erop zit. In de buitenwereld kan hij zien of een verkeerslicht op rood staat. Wat hij binnen de werkomgeving niet kan zien is bijvoorbeeld hoe de elektronica alles aanstuurt. In de buitenwereld is de trigger van een verkeerslicht onzichtbaar voor de gebruiker. Alle technische details hiervan zijn verborgen voor de gebruiker. Die hoeft hij namelijk niet te zien.

Als iets voor de gebruiker verborgen is betekent dit echter niet dat het ook ontoegankelijk is. Een monteur moet namelijk bij de elektronica kunnen komen om aanpassingen te verrichten. Hier moet bij het design allemaal rekening mee worden gehouden. Waarom is dit nu belangrijk? Bij software zijn er elementen die op het ene moment wel belangrijk zijn en op het andere moment niet. Daar wordt geschakeld tussen zichtbaar en onzichtbaar.

Een voorbeeld van een verschil in zichtbaarheid zijn de bekende ‘geavanceerde’ functies in programma’s. Deze functies zijn te ingewikkeld voor de ‘gewone’ gebruiker en worden alleen van toepassing naar mate meer (of andere) functionaliteit wordt gevraagd. Dit wordt gedaan om de complexiteit voor de gebruiker te verminderen. Het is daarbij belangrijk dat de gebruiker niet te snel wordt overvraagd. Laten zien wat de gebruiker ook echt wil zien is een complex proces, zeker naar mate de type gebruiker varieert. Dit wordt vaak opgelost door middel van een dynamische omgeving. Eentje die aanpasbaar is aan de gebruiker.

Bij gebruik van een dynamische omgeving kan de gebruiker zelf beslissen wat voor hem belangrijk is en wat niet. De standaard, simpel en geavanceerde standen doen aannames over wat iemand moeilijk vindt of nodig heeft. Dit is iets wat niet altijd te voorspellen valt. Wanneer je één enkele instelling nodig hebt dat toevallig geavanceerd is wordt je gelijk overspoeld met allerlei andere onzinnige instellingen. Voor elk systeem is het weer een uitdaging om hier een balans in te vinden. Het is daarbij noodzakelijk te weten wie je gebruiker is.

Bewust versus onbewust

Wie aan interactie met systemen denkt komt vrij snel uit bij de PC, de personal computer. *‘However, in most of the cases, computers are only parts serving to the functioning larger technical systems, so our interaction with them is not as explicit as when a personal computer is in use.’ (J.Cañas, B.Velichkovsky, M.Velichkovsky, 2011)[3]* Interactie

met computers gebeurt zowel bewust als onbewust. Doordat het lastig kan zijn om het onderscheid tussen bewuste en onbewuste interactie direct te herkennen staan in figuur 3.3 een aantal voorbeelden genoemd.

Bewust	Onbewust
• PC • Smartphone • Navigatie	• Camera gezichtsherkenning • Verkeerssensor wegdek • ABS

Figuur 3.3: Focus van interactie.

Het onderscheid in bewuste en onbewuste interactie wordt gemaakt omdat er in deze scriptie alleen wordt gekeken naar bewuste interactie. Doordat computers in allerlei hulpmiddelen verwerkt worden, gebeurt interactie vaak onbewust en onzichtbaar achter de schermen. Daarbij moet alleen het noodzakelijke voor de gebruiker zichtbaar zijn. *'The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.'* (Mark Weiser, 1991)[19] Als het om rekening houden met de gebruiker gaat zijn onzichtbare computers echter niet het probleem. Het zit hem namelijk in het directe contact met de gebruiker. De voornaamste bewuste interactie tussen mens en computer gebeurt tijdens het gebruik van user interfaces. *'Today's multimedia machines makes the computer screen into a demanding focus of attention rather than allowing it to fade in the background.'* (Mark Weiser, 1991)[19]

3.2.4 Reductie in complexiteit tussen systeem en gebruiker

In sectie 2.2 hebben we gezien wat de verschillende werkzaamheden zijn van een werknemer in een organisatie. In de vorige sectie werd gekeken welke type interactie daarbij komt kijken. Nu is het tijd om wat verder in te gaan op welke manier die interactie tot stand komt bij het gebruik van interfaces.

J.Dul geeft aan dat: *'HFE can also contribute to the design of natural user interfaces in human-computer interactions.'* (2012)[5] Het subtiel hierin is het woordje 'natuurlijk'. De mens is gehecht aan zijn gewoontes. Structuur en consistentie zijn gekoppeld aan ons levenspatroon. Dat vind je terug in de user interfaces. Structuren kunnen in de loop van tijd echter veranderen. Dit wil niet altijd zeggen dat de rest ook mee verandert.

Een goed voorbeeld hiervan zijn symbolen. Deze kunnen blijven bestaan ondanks de verandering in structuur.

Zo zijn er nog veel systemen die het opslaan van iets weergeven met behulp van een diskette. Voor de jongere generatie heeft dit tekenje echter totaal geen betekenis anders dan dat ze hebben *geleerd* dat ze daardoor iets kunnen opslaan. Jongeren zullen alleen het symbool aan die actie linken als er constant gebruik van wordt gemaakt. De verbinding met de oorspronkelijke symboliek is verdwenen. Het is van een concreet symbool in een abstract symbool veranderd.

Symbolen zijn het liefst algemeen bekend, dan hoeft namelijk niemand ze te leren. Bij algemene symbolen bestaat echter het gevaar dat ze hun betekenis kunnen verliezen of dat de intuïtie verloren gaat. Doordat je het moet leren ontstaat een cognitieve last wat de interactie lastiger maakt.

De vraag is of iets beters te bedenken is voor de floppy als symbool voor opslaan. Doordat deze vraag niet alleen in de digitale wereld geldt, volgt hier een uitspraak die het gebruik van symbolen goed weergeeft: *‘All that is really required of a good symbol is that it is unique, simple to reproduce, and memorable’*⁸ Een floppy disk bevat al deze eisen. Of we het begrip ‘opslaan’ niet langzaam aan het verliezen zijn (en of we dat wel kunnen accepteren) wordt in het kader nog nader bekeken.

*Uitweiding van een forum over de keerzijde van automatisch opslaan.*⁹

”For a somewhat related anecdote, we built an app recently that did away with the save button, because as google does, we were using AJAX to save in the background. Good God you wouldn’t believe how many complaints we got from the client because they were confused about how to ”save.” So we just added a save button that fires a toast notification that the file was saved.

It’s ridiculous, but that’s the way the corporate world works.

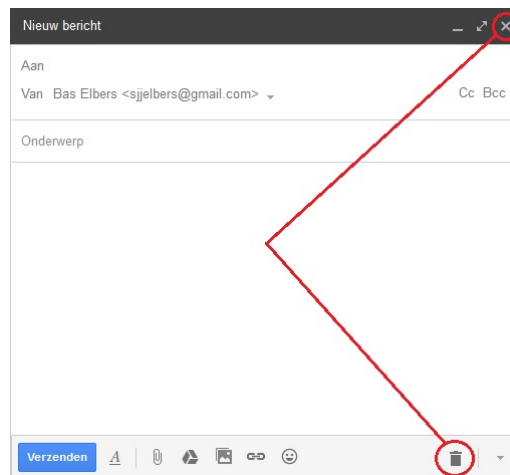
Well, it wasn’t the icon they were looking for. It was missing a save button completely. Even though we had a toast notification showing the save in the background that saved as they clicked out of the input field. The complaints we got were that they weren’t sure it was saving and they were scared they would lose their information when they left the page. It appeared to be the physical act of clicking the ”save button” that gave them the confidence that it was saved.

Keep in mind that this app wasn’t built for teens and people familiar with the newest tech gadgets. It wasn’t a commercial app. This app was built for enterprise, so it was a lot of lower level data entry people. Most of them probably didn’t have smartphones or if they did probably got one more recently.

⁸<http://connortomas.com/2013/04/in-defence-of-the-floppy-disk-save-symbol/>

It just shows how much your audience really matters when designing user interfaces. Trying to make sexy applications for enterprise is not an easy thing to do! I learned my lesson.”

Commentaar: *Wat hier goed uit gedestilleerd kan worden is het onwennige nieuwe verschijnsel van automatisch opslaan. De oude structuur, handmatig opslaan, is zodanig ingeburgerd dat de gebruiker overtuigd moet worden dat iets automatisch wordt opgeslagen. Nu kan wel gelijk worden gezegd dat bijvoorbeeld Google ook alles er aan doet om duidelijk te maken dat iets automatisch wordt opgeslagen. Zo krijg je bij het typen van een mail in Gmail onderaan regelmatig te zien dat er tussentijds wordt opgeslagen. Daarnaast staat bij het afsluiten van het bericht 'opslaan en sluiten' zodat je niet bang bent iets te verliezen. Helaas voor Google betekent dit wel dat het aantal nutteloze concept emails vanzelf groeit. Dit komt omdat je geneigd bent om iets af te sluiten met het kruisje als je het niet wilt hebben. Het knopje 'concept verwijderen' is duidelijk zichtbaar, maar niet de eerste keuze (zie figuur 3.4).*



Figuur 3.4: Mogelijkheden voor het sluiten van een concept in Gmail.

De maker van het bericht op het forum is vrij overtuigd van het feit dat het eigen ontwerp duidelijk moet zijn. Het is dan ook niet vreemd dat de maker de gebruikers in een hokje stopt. Aannames over je gebruiker zijn soms nodig om het geheel minder complex te maken. Aannemen dat een administratief medewerker geen smartphone heeft en dus het begrip automatisch opslaan niet kent is te voorbarig. Daarnaast is het zo dat iemand die automatisch opslaan niet kent, bij de juiste notificatie, overtuigd moet kunnen worden dat het werk is opgeslagen. Je publiek kennen is belangrijk. De vraag blijft echter wel wat een 'sexy' applicatie precies is en wat dit voor een onderneming nu zo moeilijk maakt. Het Googlen van 'sexy applications' blijkt echter geen goed idee.

Kader 3.5: Een uitweiding van een forum bericht over automatisch opslaan

3.2.5 Bestaat de perfecte interface?

Wat is een goede user interface? Begrippen als easy-to-use en intuïtieve user interface zijn daarbij niet duidelijk genoeg. Raskin (1994) geeft aan dat intuïtief betekent bekend^[15]. Het moet op iets lijken wat je al kent. Dat geeft vrij weinig ruimte voor iets creatiefs en nieuws (Adream Blair-Early, Mike Zender, 2008)^[2]. Terwijl in debat kan worden gegaan of dat erg is maakt dat het ontwerpen van nieuwe vormen van interactie (zoals spraak en touch) in elk geval wel moeilijker.

Niet alleen moet het gebruik met een user interface natuurlijk over komen, het moet ook nog eens mooi eruit zien. *‘A specific aspect of affect is the pleasantness or unpleasantness that may be generated by visual attractiveness, or aesthetics, as Tractinsky puts it (Tractinsky 2006). Tractinsky made a strong argument that aesthetics has become a major differentiating factor between IT products that many products now provide the same functionality and meet the same needs; this has happened because aesthetics satisfies basic human needs and because human needs are increasingly supplied by IT (Tractinsky 2006). Perceived aesthetics (Tractinsky et al. 2000), perceived visual attractiveness (van der Heijden 2003) and first impression (Everard et al. 2006; Schenkman et al. 2000) have all been found to influence people’s judgment of IT, as they regard what is beautiful is usable (Tractinsky et al. 2000). As Norman stated, beautiful things work better (Norman 2004).’ (Zhang en anderen, 2007)*^[20]

Schoonheid in systemen is een relatief nieuw begrip. Voorheen was het vooral functionaliteit dat voorop stond. Zoals in sectie 3.1.1 beschreven komt dit mede vanwege het feit dat snelheid en opslag toen nog een bottleneck was. Tegenwoordig zijn systemen snel en soepel en kan er veel meer mee worden gedaan.

Een goede user interface combineert natuurlijkheid met schoonheid. Beide begrippen zijn subjectief, dus moeilijk te definiëren. Er zijn wel principes waar een interface aan moet voldoen, maar dat maakt het niet concreet. Daarnaast verandert de manier van interactie met systemen ook. Hier moet je denken aan touch screens of interactie met behulp van spraakherkenning.

‘However, one can at present witness a paradigm shift in the interaction that requires a change in how to understand and to evaluate usability. The change is motivated by the design of new applications and services under the influence of increasingly fast convergence of nano-, bio- and information technologies with cognitive science (see, e.g., NBIC-Report, 2006). This new development has several features that force cognitive ergonomists to re-think the concept of interaction. Those features are, first of all, that applications are designed to be used in intelligent and mobile environments... ubiquitous computing... supposed to be adaptable to the current intention, state of knowledge and emotion of the user.’ (J.Cañas, B.Velichkovsky, M.Velichkovsky, 2011)^[3]

⁹<http://www.sitepoint.com/challenge-redesigning-save-icon/>

De tijd moet nog komen dat de user interface zich aanpast aan de identiteit en de emoties van een gebruiker. Computers worden wel steeds beter in het lezen van emoties, bijvoorbeeld door het lezen van emoties in gezichtsuitdrukkingen. Alleen, wat moet een user interface anders doen als iemand kwaad of vrolijk is? Een veelbelovend gebied waar momenteel wordt geëxperimenteerd met emotie is in de auto-industrie ¹⁰. Daar wordt onder andere gekeken naar mogelijkheden om een bestuurder te kalmeren als de auto via een sensor constateert dat de bestuurder boos is. Hoe ver een systeem of interface zich kan aanpassen en tot waar dit toevoeging biedt aan de functionaliteit is momenteel nog onduidelijk. Meer onderzoek is hiervoor nodig.

3.3 Principles

3.3.1 Aanpassing aan de gebruiker

‘Most information systems are developed in response to a need that is determined to exist. That need could originate from the organizational level, as in an enterprise-wide system, or at the individual level, for making decisions more accurately, strategically, or quickly. In either case, individuals will use the system, facing its screens and needing to understand and respond to its prompts. Developing systems from the perspective of users is therefore a logical, yet sometimes neglected, strategy.’ (Galletta, Zhang, 2006)[6]

Dit is dus het precieze verschil tussen een gebruiker die zich aanpast aan het systeem en een systeem dat zich aanpast aan de gebruiker. Wat er nodig is voor dit systeem moet zo basaal mogelijk worden opgeschreven. Dat betekent zonder aannames over hoe iets wordt gedaan. Het gaat er eerst om wat in het systeem moet komen (functionele en non-functionele requirements) en wat de restricties daarbij zijn (business rules). Daarna begin je pas te oriënteren hoe iets mogelijk kan worden gedaan. Vervolgens kies je de beste optie uit. Het is daarbij belangrijk het perspectief van de gebruiker in acht te nemen.

Een voorbeeld van een perspectief vanuit de gebruiker is het project van een boekenwinkel die 2e hands boeken wilde verkopen. De boekenwinkel doneerde standaard 10 procent aan het goede doel. Het enige probleem was wie het goede doel moest bepalen, de koper of verkoper. Zonder na te gaan of een dynamische structuur kon worden gehandhaafd (dit zou te ingewikkeld zijn voor de gebruiker) maakte het management uiteindelijk de keuze.

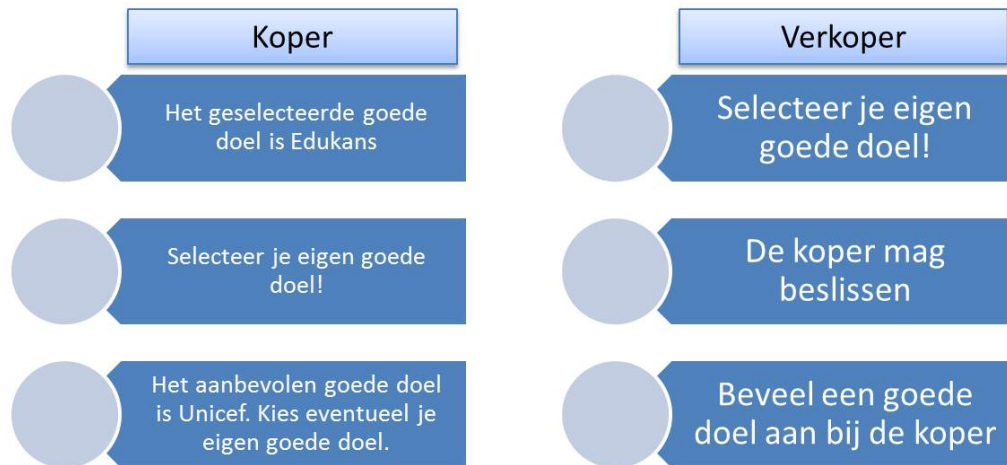
Een conflict tussen koper en verkoper komt naar voren wanneer wordt vastgesteld dat:

- 1) Koper wil bepalen welk goed doel hij steunt.
- 2) Verkoper wil bepalen welk goed doel hij steunt.

In plaats van er een te kiezen was het beter geweest om zich te verplaatsen in de gebruiker en te constateren dat beide opties mogelijk zijn met een eenvoudige weergave. De

¹⁰<http://www.sciencedaily.com/releases/2014/03/140313173001.htm>

verkoper kan bepalen om het goede doel echt vast te zetten, eentje aan te bevelen of aan de koper over te laten. De keuze van de koper hangt af van de verkoper. Een voorbeeld van een representatie staat in figuur 3.6.



Figuur 3.6: De 'complexe' representatie van de keuzes voor de verkoper en de gevolgen voor de koper.

Zonder het onnodig complex te maken kan zowel de wens van de koper als verkoper worden nageleefd. In dit specifieke geval werd door het management van tevoren bepaald dat de koper het goede doel moest vaststellen. Halverwege het project werd dit plotseling gewijzigd naar de verkoper, een indicatie dat een conflict was ontstaan tussen de belangen. Dit hoeft echter niet, zolang de gebruiker maar wel centraal staat.

3.3.2 Focus van een systeem

'Olson and Olson note that in a user-centered design strategy, a design begins with detailed considerations of users' tasks and capabilities: Who are the potential users? How varied are they? What is their current work like? Which aspects of their work are difficult? What are their needs? There are three key aspects to this design strategy: involving users, iterative design, and the role of theory about users (Olson et al. 1991).' (Zhang en anderen, 2007)[20]

Bij het analyseren van je gebruikers is het belangrijk om de taken, en mogelijkheden om die taken uit te voeren, in kaart te brengen. Het subtiele van het in kaart brengen van taken zijn de alternatieven en uitzonderingen die een taak kunnen hebben. In sectie 2.3 is hier over gesproken. Daar werd uitgelegd dat een werknemer met verschillende scenario's om moet kunnen gaan. Vaak is er een bepaalde standaard weg die wordt afgelegd, die zich soms vertakt in andere wegen. De frequentie van de takken zorgt voor de focus van een systeem. Normaliter is het zo dat de belangrijkste takken op de voorgrond treden

en dat voor de minder belangrijke elementen relatief meer moeite moet worden gedaan. Dit gaat echter nog weleens mis.

Bijvoorbeeld bij een taxibedrijf waar een nieuw systeem is geïntroduceerd. Elke rit wordt digitaal afgehandeld, maar er zijn vraagtekens bij te stellen. Niet alleen is het systeem ongelofelijk traag (terwijl ze dit rijdend bedienen), maar het systeem vraagt na elke rit of de rit wellicht geprint moet worden. In de drie maanden dat de auteur met de taxi chauffeurs meereed, printte geen een een taxirit uit. Ondanks dat vraagt het systeem toch elke rit apart of er geprint moet worden. Deze optie blijkt niet te kunnen worden uitgezet. Het proces van A tot Z is dan niet goed in kaart gebracht en irriteert daarmee de chauffeurs. Hoe meer handelingen er zijn, des te meer de bestuurder van de weg is afgeleid. Een design principe moet de vereiste interactie tot een absoluut minimum houden.

Uitweiding over het systeem van de taxi:

Het ergste van het verhaal over het systeem van de taxi is dat bij navraag bij een van de chauffeurs werd aangegeven dat het vorige systeem veel minder aandacht nodig had. In het geval van de actie om iets uit te printen werd dit als een van de opties aangegeven die tijdens de gehele rit kon worden aangevinkt. Op die manier kon op elk moment dat de reiziger aangaf graag een bon te willen het worden ingevoerd in het systeem. Sommige reizigers geven aan het begin van de rit al aan graag een bon te willen. Bij het huidige systeem moet de chauffeur onthouden dit op het eind te selecteren. De natuurlijke reactie van de chauffeur is om 'die irritante pop-up' weg te drukken, zonder dat daarbij de mogelijkheid bestaat om die actie te herstellen.

Het interessante is hetgeen wat in sectie 2.5 is besproken. Het is namelijk de vraag of de ontwikkelaars de goede en slechte eigenschappen van het vorige systeem in kaart hebben gebracht. Aan de specifieke chauffeur is in elk geval geen enkele mogelijkheid geboden om deel te nemen aan het ontwerp van het nieuwe systeem. De manager gaf aan erg blij te zijn met het nieuwe systeem. De chauffeurs willen graag weer terug naar het oude systeem.

Kader 3.7: Vergelijking tussen het oude en nieuwe systeem van de taxi.

Bij het in kaart brengen van de verschillende takken die worden afgelegd blijft het belangrijk om de context erbij te betrekken. Alleen op de frequentie van paden afgaan is niet genoeg, de factor van de zwaarte moet erbij worden betrokken. Een voorbeeld hiervan zijn vangnetten. Een noodstop moet goed toegankelijk zijn, ook al wordt deze misschien nooit gebruikt. Hetzelfde geldt wanneer een additionele vraag wordt gesteld als 'Weet u het zeker?'. Hoewel 99 van de 100 keer iemand het zeker weet, als de gevolgen groot zijn is het opvangen van die een keer dat iemand twijfelt wel belangrijk. In het geval van de taxi is de vraag om iets uit te printen niet te rechtvaardigen. Dit moet een aparte optie zijn.

Het systeem mag niet het vermogen van een gebruiker om er mee om te gaan onderschatten. Hier moet rekening mee worden gehouden bij het creëren van het systeem. Hierbij moet wel worden genoteerd dat er mogelijk een wisselwerking kan optreden tussen gebruikers of taken. De gebruiker van nu kan wellicht zonder hulp met het systeem om gaan, maar dit moet ook zo zijn als er nieuwe gebruikers komen. Om het geheel in goede banen te leiden moet je ook naar de toekomst kijken en voorspellen wat er nodig is. Als de taken veranderen moet ook naar de toekomst worden gekeken. Dit is niet altijd van tevoren te voorspellen, maar moet zeker onderzocht worden.

3.3.3 Waarden van de mens

‘According to Friedman and colleagues, an important and longstanding interest in designing information and computational systems should be to support enduring human values (Friedman et al. 2006). Value-sensitive design is a theoretically grounded approach to the design of IT that accounts for human values in a principled and comprehensive manner throughout the design process.’ (Zhang en anderen, 2007)[20]

Wanneer je de gebruiker begrijpt kan je bij het maken van een systeem rekening houden met de mens. Vooral privacy is een aspect die voor veel mensen belangrijk is, evenals het wekken van vertrouwen. Dat laatste kan al snel mis gaan, bijvoorbeeld bij het registreren bij bepaalde websites (zie kader).

Een uitweiding over de veiligheid van je wachtwoord bij sommige websites:

Een aantal websites zijn geneigd om bij registratie het ingevoerde wachtwoord via de mail op te sturen. Als ze dit combineren met het enkelvoudig invoeren van het e-mailadres daalt het vertrouwen in de website al direct naar een minimum. Er zijn altijd mensen die het e-mailadres per ongeluk verkeerd invoeren. De kans dat je wachtwoord (die natuurlijk bij meerdere websites wordt gebruikt) dan naar een onbekend e-mailadres wordt gemaild is vrij groot.

Daarnaast wordt het nog erger als je bedenkt dat het wachtwoord dan ook te achterhalen is. Er zijn twee mogelijke gevallen, de een is dat de website je wachtwoord als gewone tekst opslaat en de andere is dat er een omkeerbare encryptie (symmetrisch) mogelijk is. Beiden zijn kwalijk. Er is zelfs een website opgericht om de ‘plain text offenders’ aan het licht te stellen.¹¹ De correcte manier is om het wachtwoord te hashen (met salt¹²) en bij het invoeren van het wachtwoord de salt toe te voegen en deze weer te hashen (aan de server kant) en te vergelijken. Als een database dan in het geding komt is de data onbruikbaar.¹³

Kader 3.8: De keerzijde van het verkrijgen van je wachtwoord door het registreren bij bepaalde websites.

Voor het maken van een systeem moet al vroegtijdig de belangrijke waarden worden geïdentificeerd. Het is onmogelijk om achteraf nog iets toe te voegen dat het systeem

bijvoorbeeld eenvoudig maakt. In sommige gevallen is het zeer belangrijk om bepaalde waarden in het systeem terug te laten komen. In sectie 3.3.1 hadden we het over een online boekenwinkel. Hetgeen wat deze boekenwinkel zo uniek maakt is het feit dat van elk aankoopbedrag 10 procent naar het goede doel gaat. Deze vrijgevigheid moet op de website terugkomen en is het hart van de organisatie. Evenals de analyse van de gebruikers kunnen de doelen van een organisatie bijdragen bij het identificeren van de waarden.

3.3.4 Specifieke eisen door gebruik van principes

‘User-centred design aims at describing the human being who interacts with the system from the viewpoint of cognitive science. Then, based on those characteristics cognitive ergonomists provided engineers with a set of principles to be considered in the design.’ (J.Cañas, B.Velichkovsky, M.Velichkovsky, 2011)[3]

Het is belangrijk om veel te weten over de gebruikers van het systeem. Hoe meer informatie je kan vertalen in concrete principes hoe beter het uiteindelijke systeem aansluit bij de gebruiker. Dit is nodig voor de acceptatie van het systeem. Gebruikers moeten het gaan zien als een verlengstuk en niet als het noodzakelijk kwaad. De vraag is alleen hoe die principes eruit zien. Zijn dit vage begrippen of worden deze heel specifiek opgesteld?

Zoals in sectie 3.2.1 beschreven kan het welzijn van de gebruiker en de prestatie van het systeem negatief worden beïnvloed als de omgeving niet aansluit bij de gebruiker. Wanneer het systeem buiten de limieten van de mens treedt, of te weinig rekening houdt met wat de gebruiker wil bereiken, is er kans dat zowel de prestatie als het welzijn vermindert. *‘Reduced performance and well-being can occur when there is a lack of fit between the environment and human capabilities and aspirations.’ (J.Dul en anderen, 2012)[5]* Door onder andere eisen te stellen aan de werkomgeving kunnen vage principes als ‘eenvoudig’ of ‘zorgvuldig’ concreet worden gemaakt. Die concrete eisen kun je dan weer testen voor de verificatie van een systeem.

4 Conclusie

Zoals aangegeven in hoofdstuk 2 proberen organisaties de competitie voor te blijven door het verzamelen en verwerken van informatie in kennis. Het optimaal gebruik maken van systemen helpt de medewerkers om de taken en rollen te vervullen die bij het informatie verwerken horen. Omdat de systemen van organisaties vaak met elkaar verbonden zijn, is dit onderwerp apart besproken. Het afstemmen van systemen kan problemen opleveren, zeker naar mate er steeds meer systemen bij komen of veranderen. Aanpassingen of vernieuwingen moeten worden doorgerekend op de huidige, goed vastgelegde, situatie. Terwijl de toekomst onzeker blijft moet toch rekening worden gehouden met een mogelijke wisselwerking tussen werknemers en hun taken¹. Om het systeem te verbeteren, de negativiteit ten opzichte van veranderingen te doorbreken, begrip te kweken en om sub-optimalisatie te voorkomen moeten werknemers bij aanpassingen of vernieuwingen worden betrokken. Elke ontwikkelaar moet concessies doen. Het is voor de acceptatie van het systeem belangrijk dat die concessies worden begrepen en geaccepteerd.

Bij de verscheidene taken en rollen van werknemers horen verschillende soorten ondersteuning. Een belangrijk contrast bij deze ondersteuning is het verschil in wensen tussen een manager en kenniswerker. Een manager wil weten hoe effectief en efficiënt iemand met een systeem werkt, terwijl een werknemer zich meer interesseert hoe goed het systeem aansluit bij de wensen en behoeften. Dit verschil wordt de *prestatiedichotomie* genoemd.² De afstemming van de gebruiker op de werkomgeving zorgt voor het welzijn van de gebruiker. De afstemming van het systeem op de buitenwereld zorgt voor de prestatie. Het probleem is alleen dat er een te grote focus ligt op de prestatie van het systeem en er te weinig oog is voor het welzijn van de werknemer.

We hebben in meerdere secties gezien dat het welzijn van de werknemer op verschillende manieren kan worden verhoogd. Bij het creëren van van het systeem moet niet alleen de werknemer betrokken worden. Ook moet er rekening worden gehouden met de karakteristieken van de desbetreffende werknemer. Door de gebruiker te analyseren op het gebied van persoonlijke eigenschappen³, cognitieve eigenschappen, belangrijke waarden en de type rol (met de bijbehorende informatieverwerking) krijg je een gebruikersmodel waar een ontwerper rekening mee kan houden. Het voornaamste van een systeem is dat de gebruiker om kan gaan met de verschillende scenario's. In het bijzonder met de non-standaard categorie die de meeste problemen kan opleveren. Wat de gebruiker bij

¹Zie sectie 3.3.2

²Zie sectie 2.2

³Zie sectie 2.4 over de context van een werknemer

non-standaard gevallen nodig heeft is een systeem dat deze gevallen herkent, ondersteuning biedt en daarbij een mogelijkheid tot educatie geeft.

De mate waarin een gebruiker wordt ondersteund is moeilijk meetbaar. De prestatie van de gebruiker ten opzichte van het systeem is echter wel makkelijk te meten. Om toch iets te kunnen zeggen over de mate waarin een gebruiker wordt ondersteund is de hulp nodig van andere deeldisciplines, waaronder psychologie. We hebben in sectie 3.1.2 beargumenteerd dat de 10 heuristieken van Gerhardt-Powals en 10 principes van Nielsen helpen bij het identificeren van problemen bij de interactie met user interfaces. Bij het gebruik van user interfaces is het belangrijk dat alleen zichtbaar is wat op dat moment nodig is. Om te kunnen schakelen tussen zichtbaar en onzichtbaar moet een dynamische omgeving aanpasbaar zijn aan de gebruiker. Om de juiste balans te vinden kan het gebruikersprofiel daarbij helpen.

De conclusie van sectie 3.2.5 en de zoektocht naar een goede interface is dat deze natuurlijkheid met schoonheid combineert. Beide begrippen zijn subjectief, dus moeilijk te definiëren. Verder maakt een user interface gebruik van de structuur en consistentie die wij als mensen kennen. Het komt echter voor dat die structuur verandert, maar het gebruik wel blijft bestaan. Dat hoeft niet erg te zijn. De cognitieve last wordt dan wel verhoogd omdat de gebruiker de structuur moet leren. Er is verder onderzoek nodig in hoeverre een systeem zich aan de gebruiker kan aanpassen. Samenvattend kunnen we concluderen dat er bestaande middelen zijn om de gebruiker te ondersteunen in het gebruik van systemen en dat de ontwikkelende techniek de grenzen opzoekt in de missie om de gebruiker niet te laten aanpassen aan het systeem, maar het systeem te laten aanpassen aan de gebruiker.

5 Bibliografie

- [1] Gordon Baxter and Ian Sommerville. Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, 23(1):4–17, 2011.
- [2] A Blair-Early and M Zender. User interface design principles for interaction design. *Design Issues*, 24(3):85–107, July 2008.
- [3] José J. Cañas, Boris B. Velichkovsky, and Boris M. Velichkovsky. *Human Factors and Ergonomics*, pages 316–337. Wiley-Blackwell, 2011.
- [4] Johan De Klerk. The knowledge economy: Importance of information resources for improved organization performance. *Available at SSRN 2015333*, 2012.
- [5] Jan Dul, Ralf Bruder, Peter Buckle, Pascale Carayon, Pierre Falzon, William Marras, John Wilson, and Bas van der Doelen. A strategy for human factors/ergonomics: developing the discipline and profession. *Ergonomics*, 55(4):377–395, January 2012.
- [6] D.F. Galletta and P. Zhang. *Human-computer Interaction and Management Information Systems: Applications*. Advances in management information systems. M.E. Sharpe, 2006.
- [7] Jill Gerhardt-Powals. Cognitive engineering principles for enhancing human-computer performance. *International Journal of Human-Computer Interaction*, 8(2):189–211, 1996.
- [8] Mark A Harris and H Roland Weistroffer. Does user participation lead to system success? *SAIS 2008 Proceedings*, Paper 4, 2008.
- [9] Nina Hollender, Cristian Hofmann, Michael Deneke, and Bernhard Schmitz. Integrating cognitive load theory and concepts of human-computer interaction. *Computers in Human Behavior*, 26(6):1278 – 1288, 2010. Online Interactivity: Role of Technology in Behavior Change.
- [10] Matthew R McBride. The software architect. *Communications of the ACM*, 50(5):75–81, 2007.
- [11] Sean T McGann and Kalle Lyytinen. How information systems evolve by and for use. *Sprouts: Working Papers on Information Systems*, 5(15), 2005.

-
- [12] Jakob Nielsen. Usability inspection methods. chapter Heuristic Evaluation, pages 25–62. John Wiley & Sons, Inc., New York, NY, USA, 1994.
 - [13] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 249–256, New York, NY, USA, 1990. ACM.
 - [14] GM Olson and JS Olson. Human-computer interaction: Psychological aspects of the human use of computing. *ANNUAL REVIEW OF PSYCHOLOGY*, 54:491–516, 2003.
 - [15] Jef Raskin. Viewpoint: Intuitive equals familiar. *Commun. ACM*, 37(9):17–18, September 1994.
 - [16] Paul Tully et al. Cross-cultural issues affecting information technology use in logistics. *Proceedings Cultural Attitudes Towards Communication and Technology*, 1998.
 - [17] Hana Urbancová and Lucie Linhartová. Ensuring knowledge continuity and elimination of employee turnover in organisations in the czech republic. *Ekonomická revue*, 15(2):95–106, 2007.
 - [18] Geoff Vitlin. Knowledge management and information management: What's the difference? *Doll Martin Associates*, 2001.
 - [19] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.
 - [20] Zhang, P. D. Galletta, N. Li, H. Sun, and forthcoming. *HUMAN-COMPUTER INTERACTION*. University Press, Beijing, China., 2007.