RWTH Aachen University
Media Computing Group
Prof. Dr. Jan Borchers

Mensch-Maschine-Interaktion (Human-Computer Interaction)
SS 2007

Socio-organizational issues and stakeholder requirements

Timo Henrich

Thorsten Lennartz

Tutor: Daniel Spelmezan

Inhaltsverzeichnis

1 Einführung

Neue Software-Technologien werden von Menschen zur Lösung von Problemen oder zur Vereinfachung der Arbeit genutzt und sollten daher auch entsprechend entwickelt werden. Es ist fatal, eine neue Technologie nur unter dem Gesichtspunkt ihrer bloßen Funktionalität zu betrachten und alle anderen Aspekte außer Acht zu lassen: Ob eine Technologie erfolgreich wird, hängt nicht zuletzt davon ab, ob sich diese in bestehende Strukturen integrieren kann (socio-organizational issues) und/oder von den jeweils Betroffenen (stakeholder). Der Begriff SStakeholderümfasst hierbei alle Personen, die von dem Einsatz einer neuen Technologien in irgendeiner Form betroffen sind, also nicht nur die unmittelbar beteiligten Benutzer. Im Folgenden werden nun zunächst einige Aspekte vorgestellt, die in diesem Zusammenhang beachtet werden müssen. Danach wird kurz diskutiert, ob und wie der Erfolg neuer Technologien gemessen werden kann. Zudem wird der Begriff des Stakeholders genauer gefasst. Abschließend werden verschiedene Verfahren erörtert, die beim Entwurf neuer Software-Systeme zum Einsatz kommen können.

2 Konflikte beim Einsatz von neuer Technologien

Der Einsatz einer neuer Technologie (z.B. Software-Systeme) in einem Unternehmen kann mehr bewirken als die bloße Lösung einer vorliegenden Problematik. Je nach Konzeption bzw. erlaubter Funktionalität können Nebeneffekte auftreten, die zu Problemen mit der ursprünglichen Unternehmenstruktur führen. Hierbei kann es passieren, dass auch Personen von diesen Effekten betroffen werden, die eigentlich nicht von der Software betroffen sein bzw. mit dieser arbeiten sollten. Im folgenden Abschnitt werden einige Aspekte hierzu vorgestellt.

2.1 Probleme bei CSCW-Systemen

In einem Unternehmen existieren für gewöhnlich mehrere Abteilungen mit unterschiedlichen Aufgabenfeldern. Bei CSCW-Systemen (*computer-supported cooperative work*) handelt sich um Software, die helfen soll die Zusammenarbeit zwischen diesen zu verstärken und damit Arbeitsabläufe effizienter zu gestalten. Durch ein unzureichendes Konzept kann es durch den Einsatz solcher Software allerdings zu Problemen kommen, die den Nutzen des ganzen Systems in Frage stellen. So könnten durch das System z.B. Benutzer plötzlich Zugriff auf nicht für sie bestimmte Informationen erhalten, oder aber unkontrolliert Einfluss in Arbeitsabläufe nehmen, die ursprünglich nur von einer bestimmten Abteilung ausgeführt werden sollten. Bei der Entwicklung solcher Software sollten daher folgende Aspekte berücksichtigt werden: Wer wird das System später benutzen? Wer wird durch die Einführung des Systems wie betroffen werden? Ergeben sich überschneidende Tätigkeiten/Arbeitsabläufe, die durch das System in einen Konflikt zueinander geführt werden könnten? Erlaubt das System einzelnen Benutzern mehr/weniger Zugriff auf Informationen als diese ursprünglich haben sollten? In wie weit solche Effekte gewünscht sind, musss letztlich mit der Unternehmensführung abgestimmt werden; wichtig ist jedoch, dass sie soweit wie möglich im Voraus erkannt werden.

2.2 Einfluss auf die Unternehmenshierarchie

Jedes Unternehmen verfügt über eine mehr oder weniger ausgeprägte Hierarchie. Diese und die sozialen Kontakte der einzelnen Mitarbeiter haben einen großen Einfluss auf die Arbeitsabläufe in einem Unternehmen. Der einzelne Mitarbeiter einer Abteilung wird in der Regel nur mit den direkt unter bzw. übergeordneten Hierarchiestufen zusammenarbeiten. Die einzelnen Vorgesetzten entscheiden, welche Informationen in welche Richtung weitergeleitet werden. Zudem findet oft kein richtiger Austausch zwischen den normalen Angestellten der einzelnen Abteilungen statt, was ggf. durch die räumliche Aufteilung der einzelnen Arbeitsplätze unterstrichen wird. Ermöglicht nun ein neues Software-System die Kommunikation der Mitarbeiter aller Abteilungen miteinander und insbesondere der unterschiedlichen Hierarchiestufen untereinander, so hat dies natürlich gerade bei Unternehmen mit einer ausgeprägten Struktur massive Auswirkungen. So ist es nun möglich, dass ein Angestellter unter Auslassung seines Vorgesetzten mit einer höheren Stufe der Unternehmenshierachie kommuniziert und diesen damit eines Teiles seiner Autorität "beraubt". Der Einsatz von E-Mails in einem Unternehmen ist das Beispiel für das beschriebene Problem. Heutzutage geht der Trend ohnehin zu flacheren Kommandostrukturen, und der beschriebene Effekt ist oftmals sogar erwünscht. Sollte das nicht der Fall sein, so können natürlich technische Maßnahmen zur Einschränkung/Überwachung der Kommunikation vorgenommen werden, jedoch führt dies ggf. zu einer verringerten Akzeptanz des Systems durch seine Benutzer (vgl. Abs. 3).

2.3 Benachteiligung durch Heimarbeit

Durch den Einsatz neuer Kommunikationsmittel und dem bereits angesprochenen CSCW-System ist es oftmals unerheblich, von wo aus die Mitarbeiter eines Unternehmens ihre Arbeit erledigen. Ob diese Art zu arbeiten praktikabel ist, hängt natürlich von der zu erledigenden Arbeit ab: besteht die Arbeit darin, im Büro präsent zu sein und dort z.B. spontan anfallende Aufgaben zu erledigen oder etwa eingehende Aufträge von Kunden anzunehmen, spricht dies natürlich gegen eine Arbeit von Zuhause aus. Besteht die Arbeit hingegen darin, bestimmte Aufgaben/Projekte, ggf. bis zu einem bestimmten Zeitpunkt zu erledigen, so kann es dem Mitarbeiter freigestellt werden, von wo und wann er zu arbeiten hat - er muss lediglich Resultate bis zu einem gewissen Zeitpunkt vorweisen können (z.B. eine Team-Besprechung). Die Vorteile dieser Art der Beschäftigung liegen klar auf der Hand: das Unternehmen spart Arbeitsräume und sonstige Ressourcen, die sie solchen Mitarbeitern sonst zur Verfügung stellt. Ausserdem kann man annehmen, dass die jeweiligen Mitarbeiter mehr motiviert sind, als solche die an feste Büro-Zeiten gebunden sind, da sie die Arbeit besser ihrem Leben anpassen können. Der größte Nachteil zeigt sich nicht unmittelbar: obwohl Heimarbeiter ihren vom Büro aus arbeitenden Kollegen technisch durchaus gleichgestellt werden und auch qualitativ gleichwertig arbeiten können, müssen sie damit rechnen, bei personalpolitischen Entscheidungen wie Beförderung oder Entlassung jeweils schlechter behandelt zu werden, als ihre präsenten Kollegen. Das muss keine objektive Ursache haben und ist in der menschlichen Natur begründet. Man kann versuchen diese, durch mangelnde Präsenz hervorgerufene Benachteiligung mit Hilfe fortschrittlicher Videokonferenzlösungen zu verkleinern; vollkommen gelöst werden kann dieses

Problem damit nicht.

2.4 Workflows und BPR

Die Arbeitsabläufe (engl. Workflow) in einem Unternehmen setzen sich normalerweise aus mehreren einzelnen Tätigkeiten zusammen. So könnte vereinfacht gesprochen der Ablauf Ännahme einer Bestellungäus den 3 Tätigkeiten - Telefonat mit Kunden - Anfertigen einer schriftlichen Bestellung für diesen - Weiterleitung der Bestellung an die Warenausgabe bestehen. Diese Abläufe ähneln sich häufig und meist geht es im Kern darum, dass an einer Stelle Informationen erfasst werden und dann von diversen anderen Stellen weiterverarbeitet werden. Eine Gattung von Software über die versucht wird solche Abläufe in elektronischer automatisierter Weise darzustellen sind die sogenannten Workflow-Engines. Solche Software stellt ein Interface zur Verfügung, über das dann die entsprechenden Prozesse der "realen Welt"modelliert werden können und anschließend eine entsprechende Software-Lösung generiert wird. Problematisch ist, dass sich bei diesem Vorgehen nicht immer alle real existieren Arbeitsabläufe 1:1 durch die Workflow-Engine abbilden lassen. Ob und wie stark dieses Problem auftritt hängt natürlich auch von der verwendeten Software-Lösung ab. Oftmals kann der problematische Ablauf in einer allgemeineren Form formuliert werden, oder er wird von der elektronischen Verarbeitung ausgeklammert und in seiner bisherigen Art weitergeführt. Als alternatives progressiveres Vorgehen bietet sich das *business process re-engineering*(BPR). Hierbei werden im Vorfeld die bestehenden Arbeitsabläufe in einem Unternehmen erfasst. Anschließend werden diese mit Hinblick auf eine automatisierte Verarbeitung analysiert und gegebenenfalls angepasst. Man kann also sagen, das Unternehmen passt sich somit der Software an.

3 Faktoren für die Akzeptanz

Der Erfolg einer Software bzw. eines Systems hängt nicht zuletzt davon ab, ob diese von den betroffenen Menschen gerne benutzt wird - also von den entsprechenden Benutzern akzeptiert wird. Im Folgenden werden einige Faktoren vorgestellt, die über die Akzeptanz und damit auch letztlich über den Erfolg eines Systemes, ausschlaggebend sein können.

3.1 Profiteure eines Systems

Die Arbeit vieler Systeme besteht darin, von seinen Benutzern eingegebene Daten zu speichern bzw. zu verarbeiten und auf den dadurch gewonnenen Daten gewisse Informationen bzw. Funktionen bereitzustellen. Gerade bei solchen Systemen tritt der Effekt auf, dass manche Benutzer mehr für das System arbeiten, also mehr Daten in dieses bringen und andere eher von den Vorzügen des Systems profitieren. Diese Situation ist natürlich unbefriedigend für jene Benutzer, die mehr Arbeit in das System stecken als sie von diesem profitieren und kann dazu führen, dass das System von diesen nicht mehr genutzt wird. Das wiederum würde die gesamte Nutzbarkeit des Systems in Frage stellen, da die Datenbasis immer geringer wird. Bei Anwendungen in einem Unternehmen tritt dieser Fall am

ehesten zwischen unterschiedlichen Hierarchie-Stufen auf, wobei obere Stufen eher profitieren, die unteren Stufen für das System arbeiten. In einem solchen Fall könnte natürlich die Verwendung des Systems, z.B. durch eine Dienstanweisung weiter vorgeschrieben werden und somit das gesamte System am Leben erhalten werden, allerdings wird eine solche Maßnahme nicht unbedingt dem Arbeitsklima der Betroffenen zuträglich sein. Besser ist es, ein gewisses Maß an Symmetrie zwischen geleisteter Arbeit und erhaltenem Nutzen herzustellen. Dies lässt sich oft nicht nur mit der eigentlichen Grundfunktionalität leisten, allerdings besteht manchmal Potential für diverse Zusatzfunktionen durch die die Balance wieder verbessert werden kann.

3.2 Free-Rider Problematik

Als "Free-Rider"bezeichnet man Benutzer eines System, die nichts zu diesem beitragen und dieses lediglich nutzen. Die im Folgenden beschriebene Problematik ähnelt jener aus 3.1., nur dass in den hier beschriebenen Problemen kein Unterschied in der Position der einzelnen Benutzer besteht. Auch in diesem Fall kann ein Übergewicht an bloßen Nutznießern das System zum Mißerfolg führen. Früher oder später werden die Benutzer, die sich hauptsächlich an einem System beteiligen ihre Arbeit einstellen und damit auch Free-Ridern den Grund zu dessen Verwendung entziehen. Es ist generell darauf zu achten, dass ein System jedem Benutzern unter dem Strich mehr bietet, als er an Arbeit dafür aufwenden muss. Bei einzelnen Operationen muss diese Forderung allerdings nicht immer erfüllt werden.

3.3 Kritische Benutzeranzahl

Manche Systeme müssen erst eine gewisse Anzahl von Benutzern haben, bevor sie dem einzelnen einen Vorteil bringen können (Analog zur "Free-Rider Problematikällerdings liegt dort der Fokus auf aktiven Benutzern). Als Beispiel hierfür kann z.B. die Verbreitung von Kommunikationsmitteln wie Telefon oder E-Mail dienen. Diese Systeme bringen überhaupt erst einen Nutzen, wenn eine hinreichend große Anzahl von Personen über Zugang zu einem dieser Mittel verfügt. Was genau die hinreichende Anzahl von Personen ist, hängt von der Situation und Zielsetzung der Lösung ab. Soll z.B. nur die Kommunikation innerhalb einer bestimmten Arbeitsgruppe verbessert werden, so genügt es wenn lediglich die Beteiligten Zugang z.B. zu einem Telefon haben.

4 Beurteilung des Erfolges

Angenommen, ein System erfüllt einerseits die an es gestellten Spezifikation um ein Problem zu lösen, andererseits wird es von seinen Benutzern als nützlich angenommen (vgl. Abs. 3). Wie kann nun der Erfolg bzw. der Vorteil beurteilt werden, den z.B. ein Unternehmen durch den Einsatz dieses Systems erhält? Dies kann ggf. gewünscht sein, um die getätigte Investition in die Entwicklung der Lösung in Relation mit seinen Kosten zu setzen. In vielen Fällen wird es schwer, wenn nicht unmöglich sein, eine messbare Größe für den durch den Einsatz des System verschafften Vorteil zu finden. In kaum einem Fall

wird eine Software direkt zur Generierung von Umsatz beitragen, vielmehr sind indirekte Auswirkungen in Bezug auf Arbeitsabläufe der beteiligten Mitarbeiter zu erwarten. Als beste Kenngröße bietet sich hierbei natürlich die Zeit an: Wenn durch die Einführung des Systems die benötigte Arbeitszeit für bestimmte Aufgaben gesunken ist, so lässt sich dies durchaus in einem geldwerten Vorteil ausdrücken. Anders sieht es allerdings bei "weichen"Vorteilen wie etwa einer gestiegenen Arbeitsmoral oder einer besseren Kommunikation zwischen den beteiligten Benutzern aus. Diese lassen sich nicht gut messen (ggf. durch eine Befragung der Benutzer o.ä.) und vor allem nicht in einem finanziellen Vorteil beziffern.

5 Benutzer und Betroffene

Wer oder was sind eigentlich Stakeholder? Hierunter versteht man die Personen, die vom Einsatz der Software betroffen sind. Direkt oder indirekt. Als Beispiel kann man einen Verkaufshandel heranziehen. Wer benutzt dieses System und wer profitiert davon und wer könnte dadurch eventuell auch negativ betroffen sein? Als direkte Benutzer müssen alle Kassierer, Informationen, Lager etc. mit diesem System etwas anfangen können und darüber vernetzt sein. Liegt z.B. ein Warenfehler an der Information vor, so muss direkt eine Meldung ins Lager geschickt werden. Dadurch kann natürlich auch der Kunde profitieren, falls alles reibungslos funktioniert. Der Kunde, welcher eigentlich mit dem System direkt nichts zu tun hat, bekommt natürlich dennoch die Funktionalität des Systems an der Kasse zu spüren und ist somit in dieser Situation ein Stakeholder. Also sind Stakeholder (Benutzer) alle Personen vom Management, übers Lager, bis hin zum Endverbraucher.

Man unterscheidet grundsätzlich 4 Arten von Stakeholdern:

- *Primäre Stakeholder*: Sind alle die dieses System nutzen - die Endverbraucher
- *Sekundäre Stakeholder*: Sind Benutzer, welche nicht direkt etwas mit dem System zu tun haben, aber Informationen durch dieses erhalten - z.B. ein Kunde, der einen Kassenzettel erhält
- *Tertiäre Stakeholder*: Benutzer, welche nicht in die ersten beiden Klassen fallen, aber dennoch durch das System beeinflusst werden - wie z.B. der Berater, welcher es in Auftrag gegeben hat, aber selbst nichts mit diesem Unternehmen zu tun hat
- *Unterstützende Stakeholder*: alle diejenigen, die in das System involviert sind, aber nicht direkt mit ihm arbeiten - z.B. Designer, Entwickler

An dieser Aufteilung orientieren sich auch die Ziele der Entwickler. Das Hauptziel ist natürlich möglichst alle durch Software Betroffenen zufrieden zu stellen. Natürlich ist das nicht immer möglich und man hält sich daher an eine gewisse Priorisierung: Die Befriedigung der primären Stakeholder hat Vorrang vor der der Sekundären usw. Doch auch hierbei gilt es vorsichtig zu sein. Nicht immer ist der sekundäre Stakeholder der größte Betroffene eines Systems. Beispiel Krankenhaus: Den größten Nutzen einer Beatmungsmaschine hat nicht der Arzt, der diese bedient, sondern der Patient, dessen Leben an der

Maschine hängt. Somit also der sekundäre Benutzer. Die Reihenfolge ist also vom Projekt abhängig und muss jedes mal neu überdacht werden.

6 Erfassung der Anforderungen

6.1 Sozio-technische Systeme

Schnell wurde klar, dass sich die zunehmende Technisierung auch auf das soziale Leben der Menschen auswirkt. Deshalb wurden Modelle entwickelt, um diese beiden Welten zu vereinen und verknüpfen, sodass die Mensch-Maschinen Interaktion eine Art chemisches Gleichgewicht eingeht, indem beide funktionieren.

Beispiel: Die Argumented Reality in der Produktion: Benutzer wurden untersucht, um eine optimale Kabelbaumkonfektionierung zu entwickeln(MCK1, S.55-64)

Bei Sozio-technischen Systemen geht es darum zuerst das Problem zu erkennen, danach erst einmal Benutzer zusammenzuführen und zu beobachten - am Arbeitsplatz und in Arbeitsgruppen -, um seine Annahmen zu bestätigen und dann darüber zu spekulieren wie sich das System auf das Unternehmen auswirken könnte und wie es von Außen beeinflusst werden kann.

Um zu verstehen, was ein Unternehmen für Anforderungen an das System hat und wie weit die Technologien führen sollen, ist es wichtig durch Interviews, Beobachtungen, Analysen das Unternehmen zu verstehen und kennen zu lernen. Da wäre zum einen die *CUSTOM Methode*, welche besonders für kleine Unternehmen geeignet ist. Sie beschäftigt sich in erster Linie mit den Belangen aller Benutzer und nicht nur der primären oder sekundären Benutzer. Die CUSTOM Methode orientiert sich an sechs Punkten:

1. Den organisatorischen Inhalt beschreiben - d.h. die Ziele der Designer und die Struktur des Unternehmens
2. Benutzer personalisieren - Alle Daten der Benutzer erfassen (Name, Daten, Position etc.) und in die vier Stakeholder Klassen einteilen.
3. Arbeitsgruppen erfassen - Wer arbeitet mit wem in welcher Gruppe und an welchem Thema
4. "task-object pairs"herausarbeiten - Aufgaben, welche erledigt werden müssen, um die Ziele zu erreichen
5. Anforderungen an Benutzer erfassen - z.B. herausfinden, ob eine Fortbildung nötig ist
6. Bedürfnisse der Benutzer erfassen

Die Punkte 2-4 beschreiben die Situation des Unternehmens vor und nach der Einführung des neuen Systems. Benutzer werden befragt und daraus die ersten Schlüsse gezogen.

Um nun sicher zu stellen, dass das System gut läuft und sich gut in die Firmenstruktur integriert, werden die Punkte 5 und 6 durchgeführt.

Eine Andere Methode ist die *Open System Task Analyse* - kurz OSTA - welche sich genauso wie CUSTOM mit der sozialen und technologischen Sicht auseinandersetzt. Anders ist allerdings, dass sich OSTA auf die folgenden sieben Aufgaben bezieht, welche - im Gegensatz zu den bisher behandelten Methoden - nicht direkt auf den Benutzer eingehen:

1. Die erste Aufgabe besteht darin das System mit den Wünschen der Benutzer abzugleichen
2. Die Aufgaben sind erkannt und müssen mit dem Design vergleichen werden, da diese eventuell unterschiedlich sind
3. Die Unternehmensumgebung ist analysiert, in welche das System integriert werden soll
4. Die Veränderungen des Systems sind erkannt
5. Die Arbeitsbeziehungen sind analysiert, wie z.B. die Existenz von Arbeitsgruppen oder Beziehungen zu anderen Unternehmen
6. Die Rohfassung des Systems ist fertig entworfen und muss mit anderen Systemen abgeglichen werden und kompatibel sein
7. Der Feinschliff wird vollzogen - Das System wird optimiert und an die soziale Umgebung und die vorhandenen technischen Systeme angeglichen

Als Beispiel für sozio-technische Systeme könnte man das, sich in der Entwicklung befindliche, System STWT (*Sociotechnical Walkthrough*) erwähnen, das 2002 vorgestellt wurde und auf die sozio-technischen Methoden zurückgreift(MCK, S.323-332).

6.2 Soft System Methodik

Im Gegensatz zu den eben behandelten soziologischen Methoden, die versuchen Mensch und Maschine gleichermaßen zu beachten, beschäftigen wir uns nun mit einer Methode, welche beide "Weltenßusammenfasst und bei der das geplante System im Vordergrund steht. Hierbei ist die Mensch - Maschinen Interaktion als Teil des Systems zu sehen und nicht das System als Teil der Interaktion. Die so genannte *Soft System Methodik* - kurz SSM - wurde von Checkland begründet und hilft dem Designer sich in das Projekt gut einzufügen. Checkland unterscheidet in seiner Methode die Ebenen der realen Welt die Ebenen der technischen Welt. Insgesamt handelt es sich um folgende sieben Schritte:

1. Das Problem muss erkannt und die Analyse begonnen werden
2. Genaue Erfassung des Problems: erstellen eines "Rich Picture"
3. Grundkonzept des Systems erstellen

4. Vorläufiges System erstellen
5. Ebene 4 mit Ebene 2 abgleichen
6. Nötige Änderungen herauskristallisieren
7. Feststellen, was für diese Änderungen benötigt wird

Die Punkte 1 und 2 und 5 bis 7 befinden sich, laut Checklands Modell, auf der Ebene der realen Welt und die Punkte 3 und 4 in der technischen Welt. Näher eingehen sollte man aber auf die Erstellung des "Rich Picture". Es handelt sich hierum um eine Zeichnung, welche die Positionen, Belange und Aufgaben der Benutzer beinhaltet. Um diese Informationen zu erhalten sind alle Befragungstechniken, welche auch schon bei vorherigen Modellen erwähnt wurden, zu benutzen, um an eine reichhaltige und möglichst dichte Fülle an Informationen zu kommen. Je mehr Informationen, desto genauer wird die Zeichnung und desto besser sind die Designer im Bilde. Es gibt keine feste Richtlinie für die Erstellung der Zeichnung, aber es hat sich als nützlich erwiesen Sprechblasen für die Belange der Benutzer, sich kreuzende Schwerter für Konflikte im System und ein Auge für externe Beeinflussungen zu verwenden. Das "Rich Picture"hat sich als sehr informativ erwiesen und bietet einen recht kompletten Überblick über die Unternehmensstruktur. Zudem können an dieser Zeichnung Designer und Benutzer gleichermaßen arbeiten, da sie für alle Beteiligten - selbst ohne Vorkenntnisse - sehr verständlich ist und sogar von Kindern problemlos verstanden werden kann. Auch für die Grundkonzepterstellung gibt es eine nützliche Hilfe in diesem Modell; das Initialwort **CATWOE**:

- **C** lients - Benutzer, welche in das System integriert sind
- **A** ctors - Nutzer, welche das System nutzen, aber nicht integriert sind
- **T** ransformations - Änderungen, welche durch das System hervorgerufen wurden
- **W** eltanschauung - Wie das System wahrgenommen wird
- **O** wner - Besitzer, welche verantwortlich sind und Änderungen vornehmen können
- **E** nvironment - Umgebung und Beeinflussung des Systems

Alles in allem ist die SSM sehr gut geeignet um die Benutzer mit in die Entwicklung des Systems einzubeziehen, da es sehr einfach zu verstehen ist. Es gibt einen Leitfaden vor, aber ist in sich individuell verwendbar und vereinfacht es dem Designer ungemein die Wünsche und Bedürfnisse des Unternehmens und der Benutzer zu verstehen und in die Tat umzusetzen. Als Beispiel kann hier das COMPILE System der Uni Münster erwähnt werden. Das System beschäftigt sich mit der Interaktion verschiedener Institutionen und Angehöriger einer Uni und wird auch in mehrere Benutzergruppen, mit verschiedenen Rechten, unterteilt(MCK, S.55-64).

6.3 Participatory Design

Unter *Participatory Design* versteht man so viel wie mitwirkendes Design. In diesem Konzept wird der Benutzer direkt in den Designprozess involviert und ist ein Teil des Designteams. Diese Methode ist darauf begründet, dass die Benutzer selbst am besten wissen was Sie von dem neuen System erwarten und sind zudem auch noch die besten Testpersonen. Das Participatory Design basiert im Grundlegenden auf drei Faktoren: Man versucht die Arbeitsumgebung in den Designprozess mit einwirken zu lassen, man bezieht den Benutzer in die Entwicklung ein. Die Auswertungen und neuen Elemente ergänzen sich ständig. Da bei diesem Konzept eine andere Herangehensweise herrscht, benutzt man auch andere Kommunikationsmittel, als in den vorangehenden Methoden. Zum einen wäre dort das *Brainstorming*, welches alle Entwickler (Designer und Benutzer) in einem Gedankenpool zusammenwirken lässt, in dem jeder seine Ideen in die Menge wirft und diese von den anderen weiterentwickelt oder verworfen werden. Eine weitere Art der Ideensammlung wurde schon in Kapitel 6 (im HCI Buch) erwähnt, nämlich das *Storybording*, eine Art Tagesplaner, indem Tagesaktivitäten festgehalten werden. Nützlich sind auch *Workshops*, in denen man die Entwickler auf einen gemeinsamen Wissensstand bringt, um so ein Arbeiten auf ähnlichem Niveau zu garantieren. So verstehen die Nutzer die Gedankengänge der Designer besser und umgekehrt. Eine gemeinsame Basis kann das Arbeiten an einem Projekt ungemein erleichtern. Zuletzt wären da noch die *pencil and paper exercises* zu erwähnen, welche oft die Lücken zwischen dem aktuellen Stand des Systems und der Wünsche der Benutzer aufweisen und dies in einer kostengünstigen Art und Weise. Hauptsächlich findet man diese Arbeitsform in Skandinavien, wo es aber auch noch nicht sehr gut getestet wurde. Eine spezielle Methode dieses Konzepts wurde durch die britische Wissenschaftlerin Enid Mumford entwickelt, die ETHICS (***E**ffective **T**echnical and **H**uman **I**mplementation of **C**omputer-based **S**ystems*) - Methode. Diese verbindet die Aspekte der sozial-technischen Methoden und des Paticipatory Design, d.h. die Benutzer sind in den Designprozess involviert und werden nicht wie bei den sozial-technischen Methoden außen vor gelassen und nur als Informationsquellen betrachtet. Aber dennoch unterteilen sie die Entwickler in drei Gruppen: In *Berater* - nur zu Interviews herangezogen -, *Bevollmächtigte* - werden in Entscheidungen mit einbezogen - und *Gleichgestellte*, welche dann schließlich ein Teil des Entwicklerteams sind. In den gebildeten Arbeitsgruppen bespricht man dann schließlich wie das System aufgebaut wird, welches seine Hauptaufgaben sind und wie es in das Unternehmen aktiv integriert werden soll. Im Allgemeinen versucht man die Jobzufriedenheit zu steigern, indem die Benutzer selbst kritische Entscheidungen treffen sollen, um den Entwicklern mögliche Schwachstellen aufzuzeigen. Insgesamt kann diese Methode sehr zeitaufwändig und dadurch auch kostspielig werden, aber als Entschädigung bekommt man ein besser funktionierendes System, welches die Arbeitsmoral steigert.

6.4 Ethnografische Methoden

Bisher wurde sich bei allen Methoden hauptsächlich mit dem Beobachten der Stakeholder beschäftigt. Dies ist auch in der ethnografischen Sicht so, doch aus einer anderen Perspektive. Die *Ethnografie* beschäftigt sich viel mehr mit dem gesamten unbeeinflussten Umfeld des Stakeholders und hält sich bei der Beobachtung viel mehr im Hintergrund. Hierfür wird

zwar mehr Zeit beansprucht, aber man erhält einen Überblick des sozialen und kulturellen Umfeldes des Beobachteten. Speziell ist zudem noch, dass nicht darüber spekuliert wird, was ein Benutzer in einer bestimmten Situation machen würde, sondern nur das beachtet wird, was auch wirklich aufgezeichnet wurde. Es entsteht ein relativ unvollständiges, aber dennoch informatives Bild der Arbeitssituation.

In engem Zusammenhang mit der ethnologischen Perspektive steht die *kontextabhängige Befragung*. Sie betrachtet den Benutzer im Sinnzusammenhang mit seiner Arbeitsgewohnheit/tätigkeit. Das Ziel der kontextabhängigen Befragung ist ein neues System zu entwerfen, welches in einer reinen ethnografischen Befragung zu keinem Ende führen würde. Bei diesem Modell wird der Forscher [1] von dem Benutzer in seine Arbeit und seine Arbeitsumgebung eingeführt. Die Kommunikation zwischen den beiden Parteien findet direkt am Arbeitsplatz statt, sodass sie sich direkt auf die Arbeit bezieht und oft auch praktisch erklärt werden kann - anhand von spezifischen Beispielen vor Ort, welche die direkte Arbeit betreffen. So wird der Forscher ein aktiver Teil der Arbeit und kann sich somit ein besseres Bild von dem zu entwerfenden System machen. Das Interview am Arbeitsplatz dauert meist nicht länger als 2-3 Stunden und wird sehr intensiv geführt. Dazu muss der Forscher natürlich vorher ein grobes Bild von dem neuen System vor Augen haben, um gezielter Fragen stellen zu können. Während des Interviews zeichnet der Forscher alle Tätigkeiten des Benutzers auf, um ein möglichst komplettes Bild über seine Arbeitsumgebung und -einstellung zu bekommen. Dabei wird nicht nur auf die berufliche Tätigkeit an sich geachtet, sondern zusätzlich auch auf seine Kommunikation mit anderen Mitarbeitern. Der Forscher bedient sich dabei folgender Modelle:

- Das *Sequenz Modell* arbeitet die Schritte raus, welche nötig sind um eine spezielle Aufgabe zu erfüllen.
- Das *physische Modell* beschäftigt sich mit der Arbeitsumgebung und der Arbeitseinstellung des Benutzers.
- Das *"flow"* (Ablauf) *Modell* zeigt die Kommunikation des Benutzers am und außerhalb des Arbeitsplatzes.
- Das *kulturelle Modell* spiegelt die Arbeitskultur und -verfahren wieder und in wie weit sie die Umgebung beeinflussen.
- Das *Artefakt Modell* beschreibt die Struktur und den Nutzen bestimmter Objekte im Arbeitsalltag.

Schließlich, wenn der Forscher das Interview abgeschlossen hat, werden die Ergebnisse der einzelnen Interviews zusammengetragen und den Entwicklern vorgelegt. Diese erstellen dann Zusammenhangsdiagramme und kristallisieren die Hauptthemen der Interviews heraus und stellen daraus ihre Vorstellungen für das System zusammen.

[1] Führt nur die Interwievs durch und ist in die Entwicklung sonst nicht involviert

Literatur

[MCK] Mensch & Computer Konferenz 2002 *Vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten*. B.G. Teubner,Stuttgart,2002

[MCK1] Mensch & Computer Konferenz 2005 *Kunst und Wissenschaft . Grenzüberschreitungen der interaktiven ART.*. Oldenbourg Verlag,München,2005

[HCI] A. Dix, J. Finlay, G. D. Abowd, R. Beale *Human Computer Interaction*. Prentice Hall, 3. 2003