

VISUALISIERUNG VON NUTZERTESTDATEN

Robert Wlček

Robert Wlček
Visualisierung von Nutzertestdaten
Bachelorarbeit, 03. November 2014

Betreuer: Prof. Dr.-Ing. Bernhard Preim
Otto-von-Guericke-Universität
Fakultät für Informatik
Universitätsplatz 2
Magdeburg 39106
Deutschland

Betreuerin: Dipl.-Ing. Nadine Kempe
UCDplus GmbH
Hegelstraße 23
39104 Magdeburg
Deutschland

Eidesstattliche Erklärung

Hiermit versichere ich, Robert WIček, an Eides statt, dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

.....
Ort, Datum

.....
Unterschrift

Inhalt

I Einleitung 1

- I.1. Motivation 1
- I.2. Ziele der Arbeit 2
- I.3. Gliederung der Arbeit 2

II Grundlagen 3

- II.1. Nutzerzentriertes Design 3
- II.2. Qualitative Sozialforschung 5
- II.3. Informationsvisualisierung 7
 - II. 3. 1. Abgrenzung der Begriffe 7
 - II. 3. 2. Visualisierungspipeline 9
 - II. 3. 3. Gütekriterien 9

III Konzeption 11

- III.1. Domänencharakterisierung 11
 - III. 1. 1. Domäne 11
 - III. 1. 2. Zielgruppen und Probleme 13
 - Testleiter 13
 - Usability-Experten 13
 - Projektleiter 14
 - Auftraggeber/Produktmanager 14
- III.2. Datencharakterisierung 15
 - III. 2. 1. Aufgabenzeit 15
 - III. 2. 2. Zufriedenheit 16
 - III. 2. 3. Qualitative Daten 18
 - Codes und Marker 19
- III.3. Diskussion bisheriger Ansätze 20
 - III. 3. 1. Aus den Domänen 20
 - Usability 20
 - Qualitative Sozialforschung 20
 - III. 3. 2. Aus der Informationsvisualisierung 21
 - Parallelkoordinaten 21
 - Scatterplots 21
 - Tabellen 22
 - Permutationsmatrizen 24
 - Heatmaps 25
- III.4. Funktion und Form 25
 - III. 4. 1. Heatmap 25
 - Differenzraster 28
 - III. 4. 2. Farben 30
 - Quantisierte Farben 30

	Farbskalen	30
III. 4. 3.	Boxdiagramm	32
III. 4. 4.	Interaktion	33
III.5.	Workflow	35
III. 5. 1.	Testdesign	35
III. 5. 2.	Testdurchführung	36
III. 5. 3.	Testauswertung	37

IV Anekdotisches Feedback 38

V Evaluierung 41

V.1.	Problembeschreibung	41
V.2.	Abstraktion des Problems	42
V.3.	Kodierungs- und Interaktionsdesign	42
V.4.	Algorithmen- und Systemdesign	43
V.5.	Usability-Testing des komplexen Systems	43

VI Ausblick und Fazit 45

VII Literaturverzeichnis 47

I EINLEITUNG

Man müsse schon bald Schriftsteller sein, um sich ein rein analoges Leben überhaupt noch vorstellen zu können, scherzt Oliver Reichstein und prophezeit das Ende der analogen Welt in seinem Agentur-Blog auf ia.net. Es ist kein Geheimnis mehr, dass Computer fast alle Lebensbereiche der Menschen in den Industrienationen durchdringen. Ihrer Anwesenheit wird längst nicht mehr mit Missmut und Skepsis entgegnet und viele Menschen begrüßen die tiefe Symbiose von Lifestyle und Technologie. Wir haben eine gewisse Vorstellung davon, welche Aufgaben wir im Leben erfüllen möchten, in beliebiger Granularität, vom großen Lebensziel bis zum Schreiben des Einkaufszettel. Die Menschen sind bereit, für ihre ganz persönliche digitale Revolution zu bezahlen und es besteht reges wirtschaftliches Interesse daran, ihnen die Chance dafür zu geben. In Spike Jonzes Film „Her“ manifestiert sich die Utopie eines Computerdesigns, das sich so gut bedienen lässt, dass der Protagonist es nicht mehr von einem Menschen unterscheiden möchte.

I.1. Motivation

Der Mensch rückt zunehmend in den Fokus der Entwicklung neuer Anwendungssoftware und Elektronik. Den Mensch als Quelle für Designentscheidungen heranzuziehen, ist zu einem florierenden Gewerbe geworden. Software soll direkt an den Anwendenden getestet werden. Nutzerzentriertes Design erhält einen immer höheren Stellenwert bei der Gestaltung neuer Software und der Evaluierung der alten. Die Nutzer sind die Kunden und sie wissen am besten, ob die Software funktioniert. Ein zentrales Werkzeug ist der Nutzertest, bei dem die Kundschaft direkt mit dem Produkt interagiert und dessen Ergebnisse wertvolles Feedback bedeutet. Die auf dem Weg entstehenden Artefakte sind Protokolle, Tabellen, Umfrageergebnisse, Balkendiagramme, Zahlen, Bildschirmaufzeichnungen, hundertseitige Berichte. In dieser Arbeit soll ein ganzheitliches Framework zur Aufbereitung der gesammelten Nutzertestdaten vorgestellt werden.

I.2. Ziele der Arbeit

Ziel dieser Arbeit soll es sein, ein Framework zur Visualisierung von Nutzertestdaten bereitzustellen, das folgende Kriterien erfüllt:

- » Die Visualisierung ist ein hilfreiches Werkzeug bei der Auswertung der Daten für die Zielgruppen (s. Kapitel II. 1. 2.), gemessen an den Gütekriterien einer Visualisierungssoftware (s. Kapitel II. 3. 3.). Anekdotisches Feedback, gesammelt von Vertretern der Zielgruppe, soll diesen Punkt erfüllen.
- » Das präsentierte Framework lässt sich in einen bestehenden nutzerzentrierten Workflow ohne erhöhten Ressourcenaufwand implementieren. Das Analysieren des bestehenden Workflows und ein Vorschlag zur Erweiterung soll diesen Punkt erfüllen.

I.3. Gliederung der Arbeit

Zu Beginn gibt diese Arbeit Einblicke in die relevanten Forschungsbereiche der Informationsvisualisierung, qualitativen Sozialforschung und nutzerzentrierten Designs. Letzteres fasst die Einsatzdomäne des vorgestellten Frameworks zusammen, die in ihrer konkreten Ausprägung als Nutzertest in der Domänencharakterisierung betrachtet wird. Danach werden die Daten, die bei einem Nutzertest anfallen, genau charakterisiert und bewertet. Im Anschluss werden bisherige Ansätze zur Visualisierung gleicher oder ähnlicher Daten diskutiert. Aus diesen Ansätzen wird dann im Kapitel II. 4. Funktion und Form die Visualisierung und ihre Interaktionskomponente hergeleitet und ihre Konzeption theoretisch begründet. Ein beispielhafter Workflow soll dann verdeutlichen, wie das vorgestellte System real eingesetzt werden kann. Es folgt die Auswertung einer anekdotischen Feedbackrunde mit UX-Experten und Vorschlägen zu einer reproduzierbaren, validen Evaluierung der Visualisierung und des vorgeschlagenen Workflows. Ein Ausblick soll Ansatzpunkte zur Weiterentwicklung und Evaluierung bereitstellen.

II GRUNDLAGEN

Die folgenden Abschnitte sollen Aufschluss über die in dieser Arbeit verwendeten Begrifflichkeiten und Konzepte geben. Design wird seit jeher als eine intuitive, durch Übung, Erfahrung und Stil komplettierte Tätigkeit betrachtet, die erst seit wenigen Jahren verstärkt in akademische Betrachtungen berücksichtigt wurde. Im Zuge dessen wuchs das Bedürfnis nach Definitionen, die die relevanten Autoren zwar bieten, die sich aber oft ob ihrer Unschärfe und Verallgemeinerung überschneiden. Mit weitergedachten Schlussfolgerungen und Zusammenführungen soll hier kohärenter Rahmen für die später vorgestellten Konzepte geschaffen werden.

II.1. Nutzerzentriertes Design

Nutzerzentriertes Design, englisch *user-centered design* (UCD), ist eine Designphilosophie des User Experience Designs: Wie nimmt der Nutzer die Anwendung wahr? Dabei steht der Anwender, in dieser Arbeit synonym für die Nutzer, in allen Prozessschritten eines Produkts im Fokus von Forschung und Entwicklung. Das soll zur Folge haben, dass ein Design so genau auf die Anwender zugeschnitten wird, dass es "unsichtbar" wird. Diese Designschule steht im Kontrast zu anderen wie dem Ad-hoc-Design, Aufgabenzentriertem Design oder Expertendesign, denen alle unterschiedliche Analyse- und Bewertungsmethoden zu Grunde liegen (SPOOL, 2009). Tabelle 1 zeigt eine Übersicht.

Nutzerzentriertes Design zielt auf eine Maximierung der Usability für die angestrebte Zielgruppe ab. Die ISO-Norm EN ISO 9241 (ISO, 1995) definiert Usability, oder Gebrauchstauglichkeit, als das Ausmaß, mit dem ein Produkt dem Anwender eine Aufgabe

- » *effektiv*: der Nutzer kann die Aufgabe erfolgreich durchführen,
- » *effizient*: der Nutzer kann sie ohne großen Aufwand beenden, und
- » *zufriedenstellend*: die Bearbeitung der Aufgabe hinterlässt einen positiven

	Ad-hoc-Design	Eigendesign	Expertendesign	Aufgaben-zentriertes Design	Nutzerzentrier-tes Design
Quelle	Günstigste Implementierung, „Weg des geringsten Widerstands“	Eigene, intrinsische Bedürfnisse	Erfahrung, Best Practices	Anfallende Aufgaben für den Nutzer	Nutzer
Werkzeuge	Keine	Technische Anforderungen, Kollegen fragen	heuristische Evaluation, Cognitive Walkthrough	Use Cases, Flowcharts	Personas, Feld-studien, Usability-Tests, iteratives Prototyping
Beispiel	Wissenschaftlicher Prototyp	Interne Bug-tracking-Software	Portfo-lio-Webseite	Recher-chesoft-ware für Bibliotheken	OP-Planungs-software

Tabelle 1: *Designphilosophien beim Gestalten eines Softwareproduktes. Die genannten Beispiele bezeichnen Arten, in denen die jeweilige Philosophie typisch ist.*

emotionalen Eindruck beim Nutzer bewältigen lässt.

Im Softwaredesign beziehen sich diese Attribute auf die Benutzungsschnittstelle und wie sie die intendierte Aufgabe der Software unterstützt. Nach GOULD & LEWIS ist dieses Ziel zu erreichen durch

- » das Verständnis, wer die Nutzer sind und welche Aufgaben, Ziele und Wünsche sie haben,
- » enge Zusammenarbeit mit potenziellen späteren Nutzern in der Planung,
- » aktives Testen der Prototypen durch eben jene Nutzer und
- » dem Beheben gefundener Fehler und erneutem Testen, was einer iterativen Entwicklung entspricht (vgl. GOULD & LEWIS, 1985).

Die Anwender, oder ihre repräsentativen Vertreter, stehen demnach den Entwicklern und Designern direkt in zwei Funktionen zur Seite: zum Beobachten und Lernen (Anwenderforschung), und zur Evaluierung. Letzteres findet ihre Umsetzung in Nutzertests unterschiedlicher Ausprägung. Formative Tests geben Antworten auf Designentscheidun-

gen während der Entwicklung, summative Tests dienen der Validierung reifer Produkte oder der Evaluierung eines bestehenden Systems. Remote-Tests werden via VoIP und Screencasting durchgeführt, Labortests finden unter kontrollierten Bedingungen in speziell eingerichteten Räumlichkeiten statt. Ein Usability-Test soll den Prototypen auf seine Gebrauchstauglichkeit hin untersuchen ("Wird das Problem gut gelöst?"), ein Utility-Test, ob das zu Grunde liegende Software-System überhaupt die Ziele, Wünsche und Fähigkeiten der Zielgruppe widerspiegelt ("Gibt es dieses Problem?"). Das in dieser Arbeit vorgestellte Framework ist gegenüber den Differenzierungen nicht voreingenommen und kann in jedem dieser Fälle zur Anwendung kommen.

Der genau Ablauf eines Nutzertests ist maßgebend für die Form und Funktion der Visualisierung und ist Gegenstand der Domänencharakterisierung in Kapitel II. 1. .

II.2. Qualitative Sozialforschung

Qualitative Sozialforschung untersucht soziale Phänomene mittels nicht-standardisierter, empirischer Datenerhebung und deren Analyse. Nicht-standardisiert bedeutet in diesem Kontext, dass dabei nicht primär auf Fragebögen, Demographien oder anderen Statistiken zurückgegriffen wird. Die Natur dieses Ansatzes resultierte in einem Methodenstreit, infolgedessen der qualitativen Sozialforschung ein gewisser Grad an Unwissenschaftlichkeit vorgeworfen wird. Nichtsdestotrotz gibt sie fundierte Antworten auf viele Fragen der Human- und Geisteswissenschaften (MILES et al., 2013, S. 11).

Ein Forschungsgenre der qualitativen Sozialforschung ist die Grounded Theory, ein Methodenframework zur Theorienbildung auf Basis empirischer Datenerhebung. Der Begriff Grounded Theory bezeichnet dabei nicht die Methodik an sich, sondern ihr eigenes Ergebnis: eine in empirischen Daten begründete Theorie. Sich aus der Analyse der Daten formende Theorien für soziale Phänomene haben dabei einen rückwirkenden Einfluss auf die Analyse, sodass sich die Theorien und die Analyse wechselseitig bedingen (STRÜBING, 2009, S. 13 f.).

Eine der Methoden der Grounded Theory ist das Erfassen sogenannter Codes aus einem Datensatz. Ein Code weist einer kleinen Teilmenge des gesamten Datensatzes symbolisch ein zusammenfassendes, hervorstechendes und/oder in seiner Bedeutung essen-

tielles Attribut zu, um eine Kategorien übergreifende Theorie zu erschließen (MILES, 2013, S. 70 ff.). In weiterführenden Untersuchungen helfen die Code-Muster bei der kognitiven Verarbeitung des Gesamtbildes.

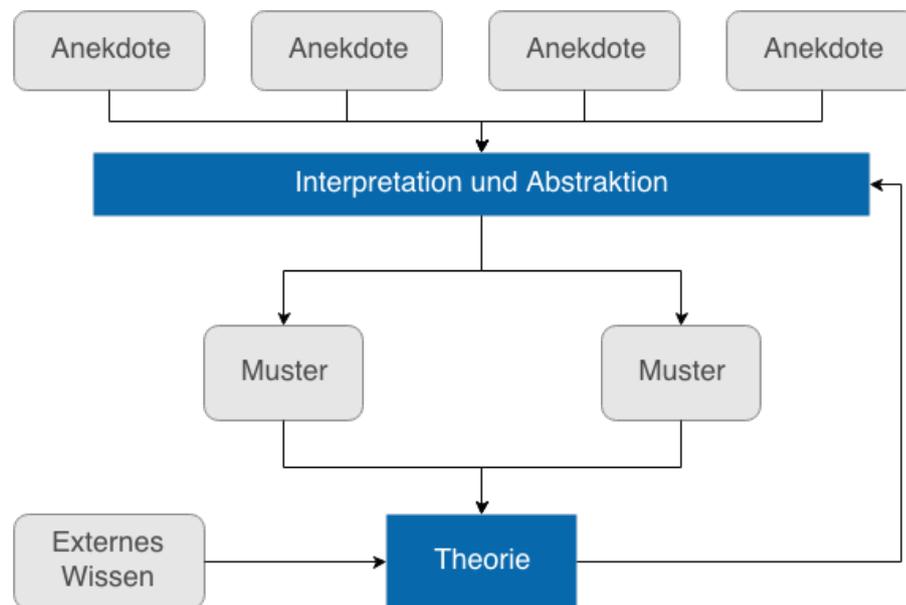


Abb. 1: Schematischer Ablauf beim Coding.

MILES führt weiter aus, dass Coding nicht nur ein tieferes Verständnis der Zusammenhänge zwischen den Daten vorbereite, sondern selbst Analyse sei. Es biete die Möglichkeit zur Reflexion und damit einhergehend eine Interpretation der möglichen Bedeutung der Daten. Eine schematischer Ablauf findet sich in Abb. 1.

Ein Beispiel soll den Vorgang verdeutlichen: Eine Forschungsteam versucht, eine Ursache für die fehlgeschlagene Implementierung einer neuen Unterrichtsmethode in einem Gymnasium zu finden. Dazu werden qualitative Interviews mit dem Lehrkörper durchgeführt. Eine Physiklehrerin erzählt:

„Letztes Schuljahr standen wir alle sehr unter Druck. Wir sollten ‚mehr mit weniger‘ erreichen, aber eigentlich erreicht man mit weniger immer nur noch weniger. Ich finanziere einen Teil des Unterrichts aus eigener Tasche mit, aber so läuft das hier heutzutage.“

Der Analyst notiert folgende Codes an diese Stelle im Transkript:

- » *conflict*: wenige Ressourcen
- » *emo(tion)*: Frustration
- » *tactic*: Widerstand
- » *tactic*: Aufopferung
- » *att(itude)*: Status Quo akzeptieren

Die Forscher können in den Transkripten der anderen Interviews und in anderen Dokumenten nach ähnlichen Codes suchen, um ein Muster zu finden. Anhand der Muster lassen sich Theorien bilden. Lag das Scheitern an der Resignation der Lehrer gegenüber dem gesunkenem Budget? Die formulierten Theorien können neues Licht auf die gesammelten Daten werfen.

Die Anwenderforschung, und im speziellen Nutzertests, um die es in dieser Arbeit geht, haben scheinbar einige Punkte mit einer Forschungsstudie gemein. DUMAS & REDISH (1993, S. 36) verweisen aber besonders auf die unterschiedlichen Ziele beider Genres. Die Ergebnisse der qualitativen Sozialforschung müssen eine aufgestellte Theorie mit anekdotischen Fakten untermauern, der Nutzertest muss die gefundenen Fehler und Probleme nicht mit Fakten stützen. Die Probleme sind in einem realistischen Umfeld aufgetreten und können später bei realen Nutzern wieder einen negativen Einfluss auf die Usability haben. Es geht darum, die Fehler zu sehen, nicht sie zu beweisen. Das Coding bietet eine Methode zur systematischen Erschließung von Mustern und Theorien, das sich auch beim Auswerten der Daten aus Nutzertests als hilfreich erweisen kann.

II.3. Informationsvisualisierung

II. 3. 1. Abgrenzung der Begriffe

Visualisierung ist der Prozess, eine Gegebenheit auf eine graphische Repräsentation abzubilden; Informationsvisualisierung ist die Kommunikation von Information zwischen einem Medium und einem menschlichen Betrachter (WARD et al., 2010, S. 1) Graphische Repräsentation ist das Abbilden von Objekten, Beziehungen und Strukturen auf die Se-

miologie und das Vokabular der Graphik, und Graphik ist das visuelle Mittel zum Lösen logischer Aufgaben (BERTIN, 1981, S. 16) Die inhaltliche Grundlage der Informationsvisualisierung sind Daten, die in einem Kontext Informationen bilden und deren Interpretation formt Wissen, ein Erkenntnisgewinn – das ist das Ziel, dass alle Visualisierungen gemein haben. Ohne das Ziel, Wissen zu schaffen, steht der Betrachter vor einer allenfalls ansprechenden Kunstgrafik oder bedient ein Spielzeug.

Graphischen Repräsentationen können visuelle Variablen zugeordnet werden. Jede dieser Variablen ist dazu im Stande, Daten zu kodieren. In Abb. 2 sind die visuellen Variablen nach BERTIN dargestellt. Die Zuordnung von Datum und / oder Information zu graphischen Repräsentationen heißt visuelle Kodierung.

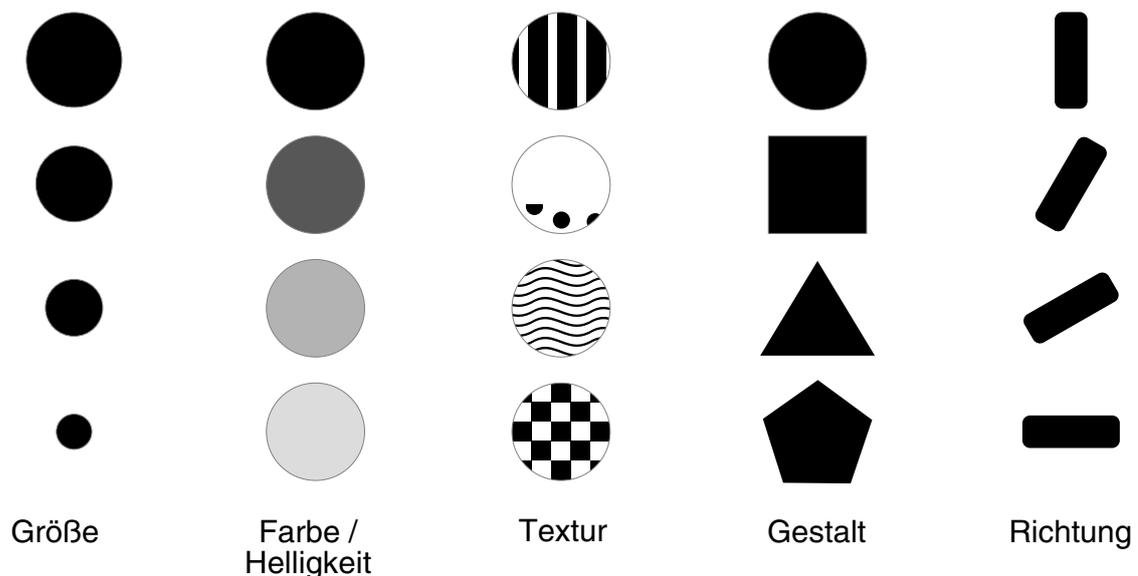


Abb. 2: Die visuellen Variablen in der Semiologie der Graphik (BERTIN, 1983)

ENGELHARDT (2002, S.4) legt dar, dass eine graphische Repräsentation ein Ausdruck, eine semantische Einheit, einer visuellen Sprache ist. Die Syntax dieser visuellen Sprache lautet:

- » Graphischer Raum, jener Raum, der die beinhaltenden graphischen Objekte oder graphischen Beziehungen enthält. Auch scheinbar leerer Raum kann Information tragen. Der Raum wird bedeutungsvoll, wenn die Positionen seiner

Objekte Bedeutung für die Interpretation haben.

- » eine Menge graphischer Objekte innerhalb des graphischen Raums und
- » eine Menge graphischer Beziehungen von Objekt zu Raum oder Objekt zu Objekt.

Das Anwenden des Graphikdesigns auf die Ausgestaltung des Raums, der Objekte und ihrer Beziehungen heißt Informationsdesign. Dazu werden unstrukturierte und komplexe Daten in wert- und bedeutungsvolle Information übersetzt (BAER, 2008, S. 12).

II. 3. 2. Visualisierungspipeline

Die Visualisierungspipeline beschreibt den Informationsfluss und die Zwischenprodukte von den Rohdaten zur Visualisierung und schlussendlich zum Erkenntnisgewinn beim Anwender. Abb. 3 zeigt die Pipeline für interaktive Informationsvisualisierung.

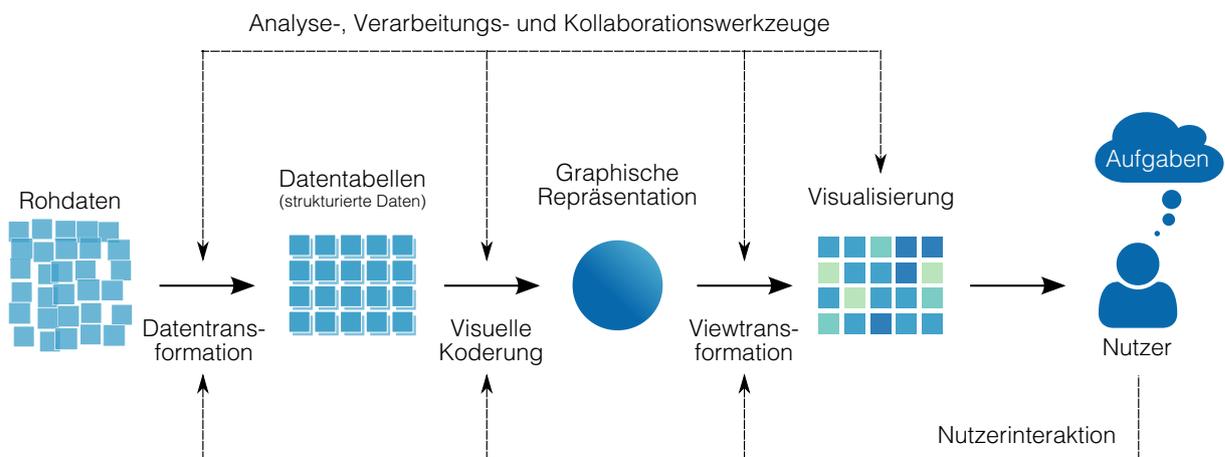


Abb. 3: Die Visualisierungspipeline (vgl. WARD et al., 2010, S. 129).

II. 3. 3. Gütekriterien

Die Gütekriterien einer Visualisierung zeigt unverkennbar Parallelen zur ISO-Norm für Gebrauchstauglichkeit, oder Usability, vgl. Kapitel II. 1. Die Visualisierung selbst ist ein Produkt, was der Anwender benutzt und unterliegt damit den gleichen Anforderungen.

- » *Expressivität*: kommuniziert die Visualisierung alle Information, die den erhofften Erkenntnisgewinn bringen, und nicht zu viel? (vgl. WARD, 2010, S. 131)
- » *Effizienz*: kommuniziert die Visualisierung die Informationen schnell, können sie vom Visualisierungsnutzer leicht aufgenommen werden? WARD (2010, S. 132) spricht von "Effektivität", dieser Begriff beinhaltet aber keine Betrachtung des Aufwandes, sondern nur der Qualität. Effizienz stellt das Maß für die Wirtschaftlichkeit bereit.
- » *User Experience*: ist es angenehm, dem Visualisierungsziel nachzugehen? Das findet bei WARD keine Erwähnung mehr, ergibt sich aber aus der Tatsache, dass die Visualisierung ein interaktives System ist und damit Regeln und Prinzipien der Mensch-Maschine-Interaktion, wie eingangs erwähnt, untersteht.

III KONZEPTION

III.1. Domänencharakterisierung

Die Domäne ist das Fachgebiet, aus der die Daten gewonnen werden und in denen die Menschen arbeiten, die später im Idealfall aus der Visualisierung Erkenntnisse gewinnen werden. Die Domäne für die in dieser Arbeit vorgestellten Visualisierung ist die Analyse und Entwicklung von Software mit Methoden des nutzerzentrierten Designs, speziell des Nutzertests.

Die Prozesse zum Gewinn der Daten und zur eigentlichen Visualisierung sind allerdings so gewählt und formuliert, dass diese Visualisierung auch für qualitative, empirische Daten aus anderen Domänen verwendet werden kann.

III. 1. 1. Domäne

Die gesammelten Daten, die den Input für die Visualisierungsanwendung darstellen, entstammen einem Nutzertest. DUMAS (1999, S.22) fasst die Eigenschaften eines Nutzertests folgendermaßen zusammen:

- » Primäres Ziel ist die Verbesserung der Usability, oder Gebrauchstauglichkeit, eines Systems.
- » Die Probanden repräsentieren reale Nutzer.
- » Die Probanden vollführen reale Aufgaben.
- » Der Testleiter beobachtet und zeichnet auf, was die Probanden tun und sagen.
- » Die Daten werden analysiert, reale Probleme diagnostiziert und Änderungen werden vorgeschlagen, um die Probleme zu lösen.

Die Ausgangslage des Testverfahrens ist abhängig von dessen Ziel. An diesem wird die Formulierung der Aufgaben orientiert und die Probanden ausgewählt. Diese Schritte werden ausführlich in der Literatur (RICHTER, 2007; NIELSON 1994) behandelt. Die Frage nach gut gewählten Aufgaben und Probanden hat einen starken Einfluss auf die Qualität der Testergebnisse, kann aber nicht durch das vorgestellte System beantwortet werden.

Die Daten werden folgendermaßen akquiriert: Jeder Proband führt der Reihe nach die Aufgaben am zum testenden System durch. Dabei wird der Bildschirm samt aller Ein- und Ausgaben aufgezeichnet. Zusätzlich werden auch alle Aussagen und Fragen des Testleiters und des Probanden aufgenommen. Eine Aufgabe wird beendet, sobald sie erfüllt wurde, der Proband aufgibt oder der Testleiter abbricht. Während des Tests führen der Testleiter oder Assistenten Protokoll. Nach der Aufgabe kann dann ein Fragebogen (Questionnaire) ausgefüllt werden, der die Zufriedenheit des Probanden ermitteln soll, nachdem sie das System für die Aufgabe benutzt hat. Je nach Intention kann auch nur ein Fragebogen am Ende des Tests pro Proband durchgeführt werden. Vorgehensweise und ihr Einfluss auf die Visualisierung sind im Kapitel II. 2. 2. Zufriedenheit erläutert.

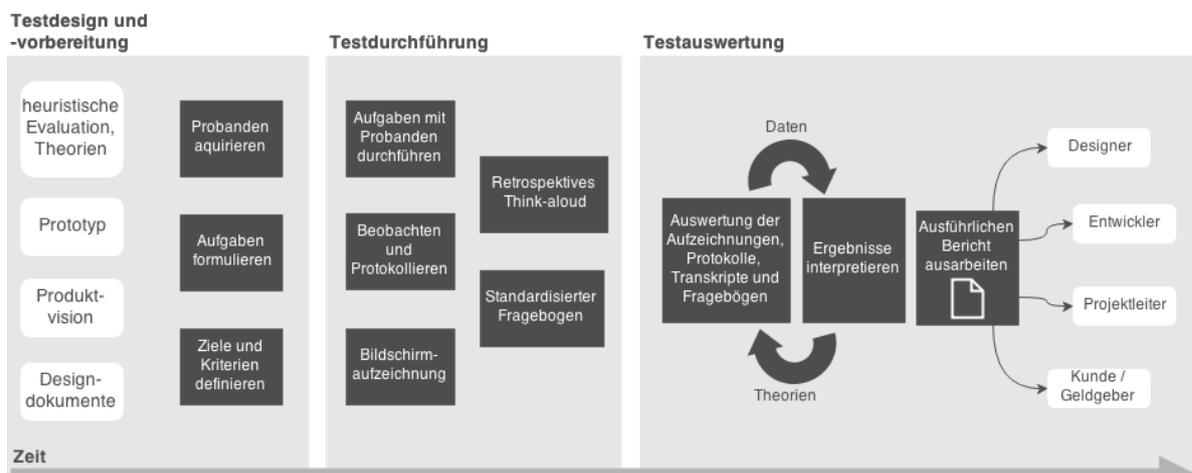


Abb. 4: Der Workflow für einen Nutzertest. Weiß hinterlegt sind die Artefakte, die als Eingangsparameter dienen oder Empfänger der Ergebnisse sind.

Im Idealfall absolvieren alle Probanden alle Aufgaben, um den Ressourcen entsprechend möglichst viele Daten zu sammeln. Manchmal soll nicht nur eine Version einer Softwareoberfläche getestet werden. Dies kann entweder bei einer Neuentwicklung, in der sich zwei Prototypen gegenüber stehen, der Fall sein, oder wenn sich eine bereits veröffentlichte Software an einem Prototypen messen lassen muss, da beispielsweise ein Update der Benutzungsoberfläche oder Bedienphilosophie bevorsteht. Ob die gleichen Probanden beide Versionen testen, muss individuell vom Designer des Tests entschieden werden und macht in manchen Fällen mehr Sinn als in anderen: wie gut kennen Benutzer das alte System und wie gut können sie mit dem neuen arbeiten? Beide Szenarien liefern geeignete Daten für die Visualisierung. Ein Blick auf die Zielgruppen soll helfen, Visualisierungsziele festzulegen.

III. 1. 2. Zielgruppen und Probleme

Die Ergebnisse aus einem Nutzertest sind für viele Menschen von Interesse, die an der (Neu-)Entwicklung eines Produkts beteiligt sind. Repräsentative Fragen sollen zu jeder Zielgruppe verdeutlichen, welche Probleme sie durch eine Analyse versucht zu lösen.

TESTLEITER

Testleiter führen die Nutzertests mit den Probanden durch. Sie müssen nicht zwingend Experten aus dem Fachgebiet des User-centered Designs sein. Testleiter, die sich nur auf die Durchführung der Tests beschränken, werden nicht mit der Visualisierungskomponente des System in Kontakt kommen.

USABILITY-EXPERTEN

Sie werten die Daten aus, die während eines Nutzertests gesammelt werden. Dabei können sie Fehler im Design aufdecken und Änderungen entwickeln, die diese Fehler vermeiden sollen. In manchen Fällen sind die Usability-Experten auch die zuständigen Interface oder Interaction Designer.

- » Welche Probleme gab es wo und bei wem?
- » Welche Aufgabe verursachte die meisten Probleme welcher Art?
- » Welcher Proband oder Persona hatte die meisten Probleme und warum?
- » Mit welcher Aufgabe waren die meisten Probanden sehr unzufrieden und warum?
- » Gab es Aufgaben, die je nach Proband sehr unterschiedliche Aufgabenzeiten hatte, und wenn ja, warum?
- » etc.

PROJEKTLEITER

Ausgehend von den Ergebnissen des Nutzertests können die Projektleiter entscheiden, welche Teile der Entwicklung priorisiert werden müssen, um mit den verfügbaren Ressourcen die besten Resultate zu erzielen.

- » In welchen Aspekt der UI müssen zusätzliche Ressourcen investiert und durch welche Aufgabe wird er getestet?
- » Entspricht die Performance des Prototypen dem Projektplan?

AUFTRAGGEBER / PRODUKTMANAGER

Im Agenturumfeld beauftragt ein Auftraggeber eine Agentur mit der Umgestaltung einer Softwareoberfläche, ein Produktmanager übernimmt diese Aufgabe meist für Firmen-interne Designteams. Die Testergebnisse sind hier in einer mit aussagekräftigen Ereignissen und Statistiken fundierte Zusammenfassung mit aussagekräftigen Trends, die den Stand des Projekts verdeutlichen, von Interesse.

- » Wie gut schneidet der Prototyp ab?
- » Woran kann ich die Performance überhaupt messen?
- » Kann ich die Entscheidungen der Designer nachvollziehen?

Eine taugliche, also effiziente, effektive und zufriedenstellende Visualisierung ist sicher für alle Zielgruppe interessant und ein Fortschritt beim Erfassen und der Analyse der Daten, jedoch sollen Usability-Experten die Kernzielgruppe dieser Arbeit darstellen. Sie sind in umfassender Weise mit den Daten und dem Test beschäftigt und kommunizieren die Ergebnisse letztendlich auch an alle anderen Beteiligten. Die vorgeschlagene Visualisierung bietet aber auch Antworten auf die anderen Fragen.

III.2. Datencharakterisierung

Die Datencharakterisierung soll Aufschluss über die Daten geben, die als Eingangsparameter der Visualisierung dienen.

III. 2. 1. Aufgabenzeit

Die Aufgabenzeit ist eine der am häufigsten gemessenen Werte in Usability-Tests (SAURO, 2012, S. 10). Es ist zu unterscheiden zwischen der Zeit bis zur Vollendung einer gegebenen Aufgabe durch einen Probanden und der Zeit bis zum Abbruch einer Aufgabe. Damit einher geht die Erfassung der Beendigungsrate: Welche Aufgaben wurden wie oft erfolgreich beendet?

Aufgabenzeiten einer Aufgabe sind untereinander vergleichbar. Es handelt sich um quantitative Daten, die genau genommen durch reelle Zahlen dargestellt werden, aber häufig im Zuge von Vereinfachungen auf Ganzzahlen abgebildet werden. So wird die Zeit bis zur Vollendung einer Aufgabe nicht bis auf einen beliebig kleinen Teil einer Sekunde bestimmt, da ein Vergleich in dieser Größenordnung wenig Aussagekraft besitzt.

STATISTIK

Die Verteilung der Aufgabenzeit ist schief (*skewed*): Einige Probanden können wesentlich länger für eine Aufgabe brauchen, als die restlichen Probanden aus der Stichprobe, da sie z.B. auf einen schwerwiegenden Fehler gestoßen sind. Damit diese Extrema statistische Aussagen nicht zu sehr verzerren, diskutiert SAURO (2012, S. 30 f.) alternative

Berechnungen für den Durchschnitt, der zur Aufstellung von Konfidenzintervallen benötigt wird.

Das arithmetische Mittel ist ungeeignet, da Extrema das Ergebnis zu stark beeinflussen. Der Median nutzt nicht die komplette Information der ohnehin schon relativ kleinen Stichprobenzahl und neigt dazu, mit jeder zusätzlichen Stichprobe stark zu springen. Zusätzlich ist es wahrscheinlich, dass der Stichprobenmedian den Bevölkerungsmedian durchgehend überschätzt. Der Median empfiehlt sich als Mittelwert für Stichprobenzahlen über 25, wobei dann eine Konfidenzintervallberechnung für Perzentile, im Falle des Medians 0,5, zum Einsatz kommen muss.

SAURO empfiehlt daher das geometrische Mittel zur Berechnung der Konfidenzintervalle oder die vorangehende Log-Transformation aller Stichprobenwerte.

III. 2. 2. Zufriedenheit

Mit Hilfe der Charakteristik Zufriedenheit soll die quantifizierte, subjektive Einstellung eines Probanden gegenüber dem Testsystem oder einer im Zusammenhang stehenden Untermenge der in diesem System möglichen Aktionen (Aufgaben) bestimmt werden (NIELSEN, 1994, S. 33). Die Zufriedenheit ist verwandt mit dem Konzept der Einstellung, dem Grad des negativen oder positiven Affekts gegenüber einem psychologischen Objekt (EDWARDS, 1983, S. 2). Daher lassen sich ähnliche Verfahren zur Ermittlung der Einstellung aus der Psychologie auch im Nutzerzentrierten Design anwenden. Die ermittelte Einstellung des Probanden ist zwar subjektiv, im Mittel über alle Probanden ist sie aber signifikant, d.h. mit einer anderen Stichprobenmenge mit ähnlichen Eigenschaften reproduzierbar (NIELSEN, 1994, S. 34).

Eine Anforderung an die Art der Erhebung für die Visualisierung ist lediglich, dass das Ergebnis ein skalarer Wert ist, der jedem Probanden zugeordnet werden kann. Ein für Nutzertests übliches Verfahren ist ein Fragebogen auf Basis einer Likert-Skala, der aus einer Zahl sogenannter Items besteht, die mit ordinalskalierten Antworten eingestuft werden. Ein typisches Item einer Likert-Skala ist eine positiv oder negativ formulierte Aussage und dazu Antworten von „trifft zu“ bis „trifft nicht zu“. Das arithmetische Mittel oder der Median über alle auf Zahlen abgebildeten Antworten ist dann der gesuchte skalare Wert.

Auch der Einsatz semantischer Differentiale ist möglich. Semantische Differentiale versuchen durch die Balance zwischen zwei gegensätzlichen Konzepten indirekt die Einstellung eines Probanden gegenüber eines psychologischen Objekts zu ermitteln (OSGOOD, 1957, S. 76 ff.). Auch hier kann mit Hilfe arithmetischer Operationen ein einfacher skalarer Wert ermittelt werden.

Der Bezug der Items lässt sich wie folgt kategorisieren:

- » Die Items können sich allgemein auf das System beziehen, was zu einer Gleichschaltung aller Zufriedenheitswerte pro Proband führt. (z. B. "Ich fühlte mich sehr sicher bei der Bedienung des Systems."). Ein Beispiel dafür ist SUS (BROOKE, 1996).
- » Zu jeder Aufgabe gibt es ein gleich bleibendes Set Items. So wird für jeden Probanden pro Aufgabe und pro getesteten Prototypen ein eigener Zufriedenheitswert erhoben. (z. B. "Das System hat mich beim Erfüllen dieser Aufgabe unterstützt.")
- » Zu jeder Aufgabe gibt es ein Set maßgeschneiderter Items. Einerseits ist es in diesem Fall schwierig, eine Aufgaben übergreifende Vergleichbarkeit zu gewährleisten, andererseits werden so Aufgaben-spezifische Aspekte mit in Betracht gezogen. ("Das Dropdown-Menü war mir eine Hilfe bei der Auswahl des Bundeslandes.")

Welche Items konkret genutzt werden, ist Gegenstand ausführlicher Forschung und Literatur (EDWARDS, 1983, S. 13). Bezug und die Fragen selbst obliegen dem Testdesigner und sind auch von Faktoren abhängig wie dem Ziel des Tests.

Es sind im User-centered Design auch psychophysiologische Messmethoden für die Reaktionen der Probanden bekannt, z. B. Blutdruckmessung, Hirnstrommessung, Pupillenweitung, Handfeuchte etc. Diese Art der Datenerhebung ist wesentlich komplexer und einschüchternder für die Probanden und kommt nur für einige ganz spezielle Fälle in Frage, wie beispielsweise Flugsimulatoren für die Pilotenausbildung. Da die Daten auch hier quantifizierbar sind, bildet der nicht-alltägliche Einsatz allerdings keine Hürde für den Einsatz in der in dieser Arbeit vorgestellten Visualisierung.

III. 2. 3. Qualitative Daten

Abseits der quantitativen Daten Aufgabenzeit und Zufriedenheit, letzteres nach der Evaluierung durch einen Fragebogen, werden noch zahlreiche qualitative Daten in Form von Beobachtungen gesammelt. MILES (2013, S. 11) argumentiert, dass ein großer Vorteil der Fokus auf natürlich vorkommende Begebenheiten in einem natürlichen Setting sei. Dies gilt nicht nur in der qualitativen Sozialforschung.

Welche Daten sollen gesammelt werden? Qualitative Daten sind sämtliche Beobachtungen, die vom Nutzer ausgehen. Einschließlich, aber nicht erschöpfend, beinhaltet das:

- » nicht zielführende Interaktion im User Interface, z.B. falscher Button,
- » Verzögerungen, z.B. zu langes Suchen nach einem Element,
- » Verwirrung: Eine Anweisung auf dem Bildschirm wurde falsch interpretiert, oder nicht verstanden,
- » Gedächtnis: Der Proband hat eine Information vergessen und muss noch einmal zurücknavigieren,
- » der Proband hat einen Fehler gemacht und kann ihn nicht rückgängig machen,
- » Ablenkung: Der Proband ist von einem Element abgelenkt,
- » Proband stößt auf eine physische Barriere, z. B. bei Sehschwäche,
- » Reaktionen des Probanden: Irritation, Seufzer durch Genervtsein, Lachen
- » Antworten auf vom Testleiter gestellte, offene Fragen,
- » laut denken und verbales Protokoll.

Im angemessenen Fall kann sich der Testdesigner auch dazu entscheiden, die Probanden grundsätzlich zum lauten Mitdenken anzuregen und ein verbales Protokoll zu führen oder führen zu lassen. Dies kann Einblicke in das Kurzzeitgedächtnis, das mentale Modell und andere kognitive Prozesse der Probanden geben (DUMAS & REDISH, 1999, S.279). In diesem Fall ist davon auszugehen, dass die Aufgabenzeiten nicht mehr repräsentativ für die tatsächliche Performance ist, da sich durch das laute Aussprechen der Gedanken die Interaktion verzögert.

Eine Alternative stellt das retrospektive laute Denken dar, bei dem dem Probanden am Ende der Aufgabenbearbeitung die Aufnahme gezeigt wird und sie dann in Retrospektive kommentiert. Probanden neigen dazu, dann ihre Vorgehensweise hintergründiger zu beschreiben und Vorschläge zur Verbesserung zu geben (DUMAS & REDISH, 1999, S. 279 f.). Für den Test stellen diese Kommentare wertvolle qualitative Daten dar, die ebenfalls für die Visualisierung herangezogen werden können.

CODES UND MARKER

Marker übertragen das Konzept der Coding (vgl. Kapitel II. 2.) auf Nutzertests. Die Codes werden in Markern erfasst. Dabei steht ein Code, oder der ihn repräsentierende Marker, nicht unbedingt für einen konkreten Fehler, sondern vorerst nur für eine Beobachtung. Dies kann ein Fehler sein, oder auch nur eine Teilinformation, die dann nach dem Sammeln weiterer Marker des gleichen oder eines ähnlichen Codes, zur Identifizierung eines Fehlers führen.

Ein Marker enthält folgende Informationen:

- » den Code, den er repräsentiert,
- » die Aufgabe und der Proband, dem diese Codeinstanz eindeutig zugeordnet werden kann,
- » eine Zeitmarke, wann der Code innerhalb eines Durchlaufs aufgetreten ist und nach Bedarf
- » eine Unterkategorisierung, um Codeinstanzen innerhalb des Code-eigenen Kontexts priorisieren zu können bzw. ordnen zu können, z. B. nach Schweregraden (vgl. NIELSEN, 1995).

Pro Proband, pro Aufgabe und pro Prototyp kann eindeutig ein Durchlauf zugeordnet werden. Dieser Durchlauf enthält eineindeutig 0~n Marker, wobei die Codes, die von den Markern repräsentiert werden, über das gesamte Projekt (oder nur je Prototyp, Aufgabe, Persona oder Proband) in Relation stehen können. Wie in Kapitel II. 2. 3. erwähnt, können auch Daten aus einem retrospektivem Think-aloud verwertet werden. Der Testleiter setzt dabei an die betreffende Stelle einen Marker mit dem entsprechenden Code, wenn er

noch einmal mit dem Probanden den Aufgabenverlauf durchgeht.

III.3. Diskussion bisheriger Ansätze

III. 3. 1. Aus den Domänen

USABILITY

Üblich sind Tabellen mit allen Aufgaben und Probanden gefüllt mit quantitativen und qualitativen Daten in Form schriftlicher Anmerkungen. Spezielle Tabellenformen unterscheiden sich je nach Art der Aufgaben, was notiert und worauf geachtet wurde. Dies ist ein den Daten entsprechendes natürliches Format, in dem je Aufgabe und Proband Beobachtungen notiert werden. Die Fachliteratur empfiehlt keine einheitliche Vorgehensweise.

Die US-amerikanische TechSmith® Corporation hat mit Morae® eine Software im Portfolio, die es unter anderem ihren Nutzern erlaubt, in Bildschirmaufzeichnungen Marker für bestimmte Ereignisse zu setzen (TECHSMITH CORPORATION, 2014). Das Konkurrenzprodukt Silverback von Clearleft Ltd. (CLEARLEFT LTD., 2014) unterstützt ebenfalls sogenannte *session highlights*. Im Kapitel II. 2. 3. wird dieser Ansatz theoretisch fundiert und für den Einsatz mit der entwickelten Visualisierung systematisiert. In Morae® kommen Balkendiagramme zur Visualisierung einfacher quantitativer Beziehungen zum Einsatz.

QUALITATIVE SOZIALFORSCHUNG

Aufgrund wesentlich komplexerer Szenarien haben sich hier in der Fachliteratur einige systematische Herangehensweisen etabliert, wovon die meisten auf visuelle Kodierung verzichten und Information über ihre räumliche Struktur vermitteln. Die Visualisierungen basieren hauptsächlich auf Matrizen und Netzwerkdarstellungen, wie der *partially ordered meta matrix* (MILES, 2013, S. 137). Bei dieser Darstellung werden alle Quellen – Interviewpartner, Dokumente, etc – zu allen relevanten Forschungsaspekten in eine Beziehung gesetzt, die durch einen Kommentar oder die Nennung eines passenden

Codes gekennzeichnet ist. Die Daten sind hier noch in verbaler, verschriftlichter Form hinterlegt; als Zitate, Codes, Anmerkungen, Umschreibungen. Die unterschiedlichen Darstellungen dienen unterschiedlichen Arbeitszielen, wie z.B. Erforschen, Beschreiben, Ordnen, Erklären und Voraussagen.

Das dort vorherrschende Informationsdesign basiert auf verschiedenen Gestaltgesetzen wie das der Geschlossenheit, der Kontinuität und der Nähe, um die textliche Information zu kommunizieren. Labels werden in Rastern und Netzwerken angeordnet und Beziehungen zwischen ihnen durch Linien oder zugehörigen Tabellenzellen hergestellt. BERTIN (1981, S.3) argumentiert, dass sich ohne graphische Repräsentationen die Zusammenhänge nur im Gedächtnis ergeben können und dieses klaren Einschränkungen unterliegt. Solche sprachliche Konstrukte mit graphischen Mitteln zu abstrahieren und zu symbolisieren, erweist sich als eine schwierige Aufgabe.

III. 3. 2. Aus der Informationsvisualisierung

PARALLELKOORDINATEN

Parallelkoordinaten sind ein beliebtes Mittel zur Visualisierung multivariater Daten. Sie scheinen jedoch ob ihrer Tabellen-fremden Form unintuitiv für die Zielgruppe. Die Effizienz und Effektivität ihres Einsatzes, vor allem in Kombination mit weiteren in diesem Kapitel angesprochenen Konzepten, bleibt zu klären. Abb. 5 zeigt eine interaktive Implementierung.

SCATTERPLOTS

Bezeichnen die Reihen von Tabellen bestimmte Charakteristika für die Objekte im Tabellenkopf, so kann man zwei dieser Charakteristika je auf der Ordinate und der Abszisse eines Koordinatensystem abtragen und die Punkte repräsentieren dann die Objekte. Das ist die allgemeine Transformation einer Tabelle in einen Scatterplot: Die Bedeutung von graphischem Raum, graphischem Objekt und graphischer Beziehung wird rotiert.

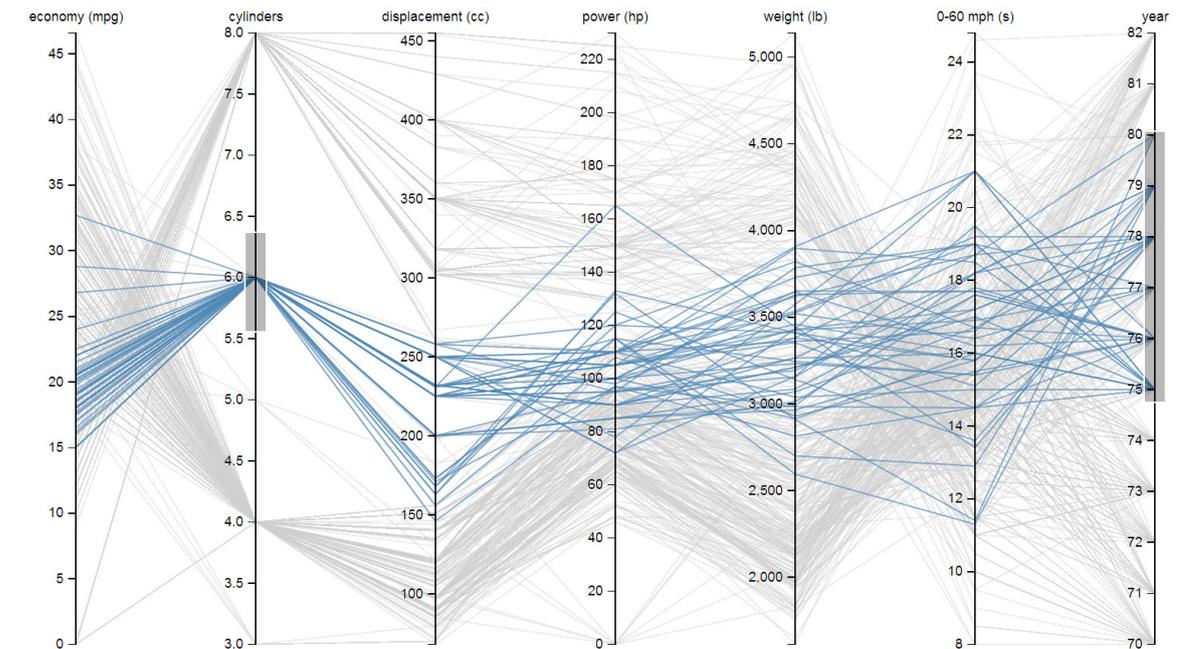


Abb. 5: Parallelkoordinaten, implementiert in D3.js von MIKE BOSTOCK (2014). Eine Kurve repräsentiert einen Datensatz. Auf der zweiten und siebten Achse ist ein sog. Brush aktiv, der einen Teil der Daten filtern, um sie gezielter explorieren zu können.

HARTIGAN (1975) erweitert das Konzept des Scatterplots um die Anordnung miteinander in Beziehung stehender Plots in einer Matrix für multivariate Daten, sogenannte Scatterplotmatrizen (SPLOMs). SPLOM-Implementierungen mit einer Interaktionskomponente gab und gibt es viele, wie beispielsweise die auf modernen und offenen Webtechnologien aufbauende Arbeit von BOSTOCK (2012). SPLOMs eignen sich gut, um Korrelationen unter allen Variablen auszumachen, verlieren aber ab einer bestimmten Datensatzbreite an Übersicht. Zudem scheinen jene Korrelationen zwischen Variablen wie Zufriedenheit und der Markerquantität für das Betätigen eines UI-Elements bestenfalls zweifelhaft.

TABELLEN

Seit über einhundert Jahren werden in Tabellen Einträge farblich unterlegt, um ihnen präattentiv eine Beziehung zuzuordnen zu können, so auch im *Atlas statistique de la population de Paris / Statistikatlas der Pariser Bevölkerung 1873* von TOUSSAINT LOUA (1873). Heutige Ausprägungen dieser Technik sind vom Inhalt abhängige Zellenfarben in Tabellenkalkulationsprogrammen.

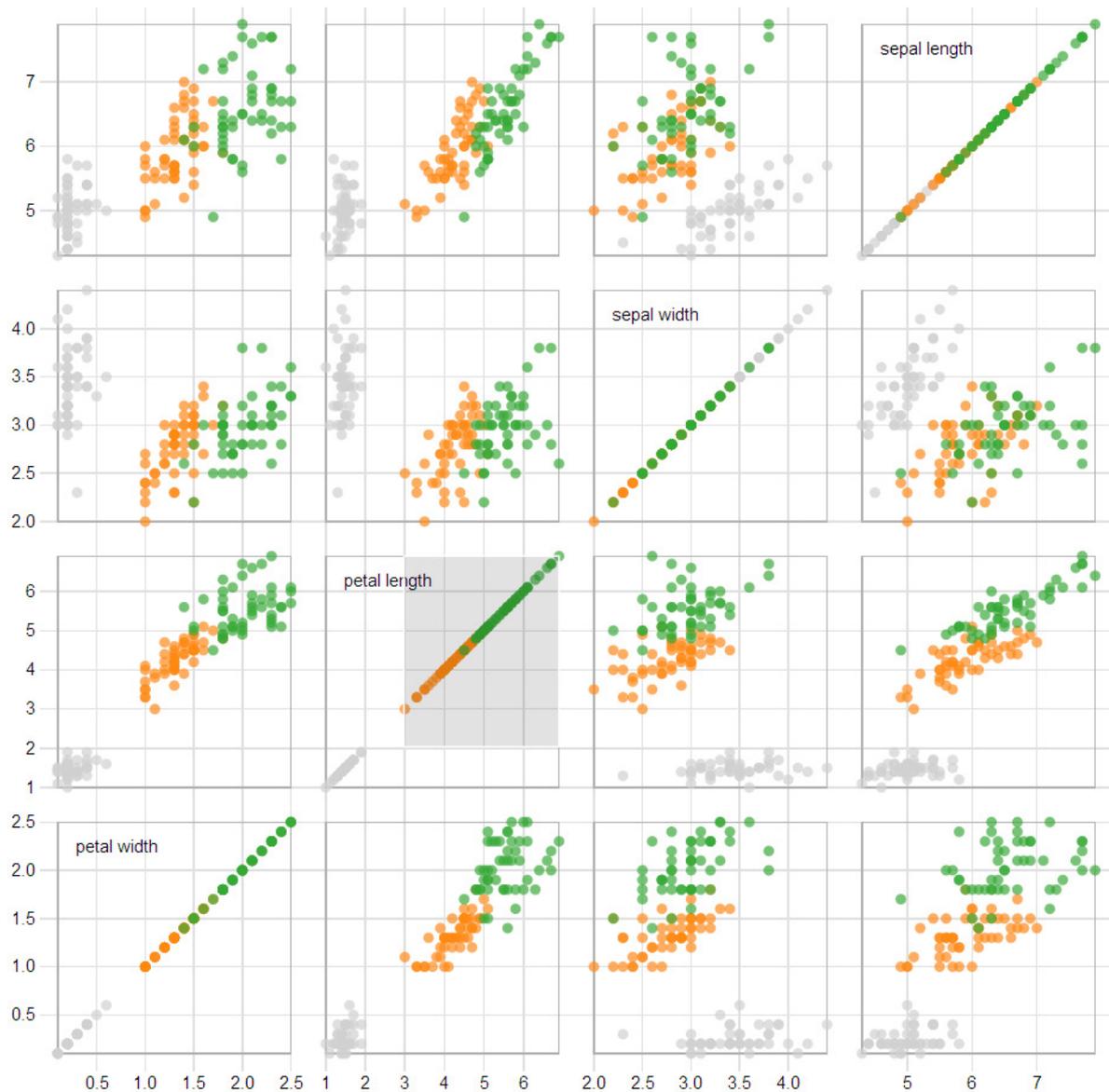


Abb. 6: Interaktive Implementierung einer Scatterplotmatrix mit vier Charakteristiken von MIKE BOSTOCK (2014) nach HARTIGAN (1975). Im zweiten Scatterplot und unten und links sind ein Teil der Daten zum Brushing ausgewählt. Die korrespondierenden Objekte in den anderen Plots werden der Auswahl entsprechend rekodiert.

H	I	J	K	L	M
August	September	Oktober	November	Dezember	
234	167	100	23	123	18-30
134	234	123	43	14	31-64
456	312	213	122	24	65+
678	456	325	232	75	Touristen
34	12	65	34	26	Lokale

Abb. 7: Ausschnitt aus einem Tabellenkalkulationsprogramm (LibreOffice Calc). Relevante Zellen werden farbig hinterlegt, um sie präattentiv identifizieren zu können.

PERMUTATIONSMATRIZEN

BERTIN veröffentlichte 1981 sein Werk „Graphics and Graphic Information Processing“ (BERTIN, 1981), in dem er sich mit der visuellen Analyse quantitativer Daten befasste. Er beschreibt die Datentabelle als fundamentales Format (BERTIN, 1981, S.26). Die mögliche Konstruktion einer einzigen Tabelle aus einem Datensatz zeugt von der Homogenität eines Problems, den Konstruktionsprozess nennt er Matrixanalyse eines Problems (ebd., S. 3).

Scatterplotmatrizen mit $n(n-1)/2$ bivariaten Scatterplots sind laut BERTINS Synoptik der graphischen Konstruktion (ebd., S.27 ff.) Tabellen mit n Reihen, eine Reihe je Charakteristik und je Spalte ein Objekt, welches von je einem Punkte im Scatterplot repräsentiert wird. Ein Scatterplot alleine kann demnach nur eine Tabelle mit zwei Charakteristika visualisieren; oder drei Charakteristika, wenn den Punkten noch eine zusätzliche visuelle Variable wie Größe oder Farbe zugeordnet wird. Für Tabellen mit mehr Reihen müssen mehrere Diagramme die Beziehungen der Charakteristika repräsentieren (SPLOMs), wobei allerdings der globale Zusammenhang der Daten keine visuelle Repräsentation mehr findet und Interpretationen scheinen dann, wie unter Scatterplot im vorherigen Abschnitt erwähnt, anekdotisch:

“Vor allem aber ist ein Problem mit n Charakteristika nicht die Summe von n Problemen mit zwei Charakteristika.“ (BERTIN, 1981, S. 26)

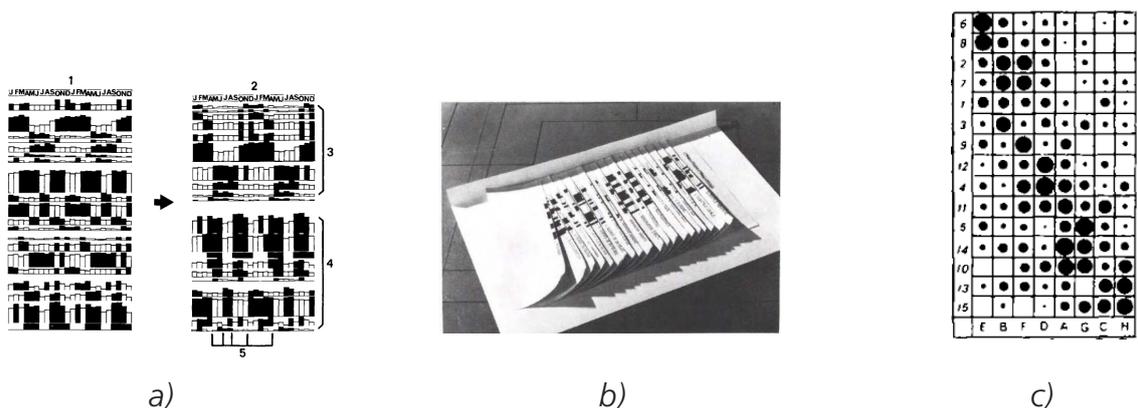


Abb. 8: Graphische Konstruktion aus BERTIN (1981): a) das Ergebnis der Matrixanalyse eines Problems mit sortierbaren (permutierbaren) Reihen, b) manuelle Permutationen, c) sortierbare Spalten und Reihen.

Die Beziehung zwischen Objekten und Charakteristika wird dann mit einer graphischen Repräsentation kognitiv leichter vergleichbar gemacht. In BERTINS Beispiel werden aus Zahlen dann Balken (Spalten sind untereinander nicht vergleichbar) bzw. unterschiedlich große Kreisscheiben (Reihen und Spalten sind untereinander vergleichbar) pro Tabellenzelle; schwarz gefüllt, sobald der Wert den Durchschnitt überschreitet. Sind die Reihen und/oder Spalten sortierbar, also untereinander vergleichbar, können Zusammenhänge mit Hilfe von Matrixpermutationen erschlossen werden. Reihen und Spalten werden neu-sortiert, um Cluster aufzuzeigen.

HEATMAPS

Diese Permutation waren nach WILKINSON (2008) erste Schritte in die Richtung, solche Matrizen als Werkzeug zur Strukturentdeckung in Datenbeständen (Clustering) zu nutzen. Dabei wird eine Distanzfunktion definiert, durch die die (mitunter abstrakte) Distanz zwischen Daten berechnet werden kann. „Nahe“ Daten gehören einem Cluster an. Als Heatmaps finden sie weite Verbreitung in Disziplinen wie der Statistik oder Bioinformatik; meist in Kombination mit einem Dendrogramm, welches das hierarchische Clustering nachvollziehbar macht (WILKINSON, 2008).

Die Heatmap soll auch das zentrale graphische Artefakt des in dieser Arbeit vorgestellten Visualisierungsprototypen sein. Sie soll dem Nutzer schnell Schwerpunkte und Beziehungen durch graphische Repräsentation im Datenbestand vermitteln und durch die Kombination von Variablen Zusammenhänge erkennen lassen.

III.4. Funktion und Form

III. 4. 1. Heatmap

Erster Schritt zum Entwurf und zur Ausarbeitung einer Visualisierung war die Domänen- und Problemcharakterisierung. Das entspricht der in der Fachliteratur vorgeschlagenen Vorgehensweise (MUNZNER et al, 2010) und zugleich BERTINS Matrixanalyse und Kons-

truktion der Datentabelle. Führt man diese Konstruktion vollständig aus, ist der nächste Schritt laut MUNZNER, *operation and datatype konstruktion*, ebenfalls abgeschlossen. Datentyp ist dann die multidimensionale Tabelle, ein übliches Ausgangsformat für interaktive Visualisierungen (SHNEIDERMAN, 1996, S. 4). Da alle Daten quantifiziert vorliegen, kann die Matrix folgende Form annehmen: Probanden auf der Abszisse, Aufgaben auf der Ordinate abtragen, die Beziehung zwischen beiden Komponenten als Tabellenzellen. Die konstruierte Heatmap ist in Abb. 9 abgebildet.

Das entspricht einer intuitiven Darstellung aus dem UCD und Übersichtsmatrizen aus der qualitativen Sozialforschung. Ausgehend von der Matrix kann Bertins Ansatz für die Matrixanalyse weiter verfolgt werden. Zunächst müssen die Eigenschaften der Spalten und Reihen bestimmt werden. BERTIN unterscheidet zwischen sortierbar (ordinal), geordnet (nominal) und topographisch (intrinsisch räumliche Koordinaten). Die Beziehungen, die den Probanden zugeordnet sind, sind sortierbar: es können Cluster erkannt werden.

Die Aufgabenzeiten der Aufgaben sind nicht direkt anhand ihres absoluten Wertes sortierbar: Die Aufgabenzeit einer Aufgabe ist kein Vergleichswert für die Zeit einer anderen. Die Abstraktion in eine visuelle Variable ermöglicht aber eine Normalisierung der Werte. So kann für jede Aufgabe eine Reihe kritischer Werte definiert werden, wonach die Zuordnung zu den, für alle Aufgaben gleichen, Farbfeldern erfolgt.

Markerquantitäten hingegen können als Aufgaben-übergreifend sortierbar betrachtet werden. Ein Code bezieht sich nicht nur auf eine Aufgabe, sondern auf das gesamte Interface, mit der die Aufgabe bearbeitet worden ist. Somit bilden die Codes eine sortierbare Quantität, oder Beziehung, für das zu testende System — unabhängig von der bearbeiteten Aufgabe.

Da nun sowohl in x-Richtung, als auch in y-Richtung sortierbare Objekte vorliegen, können in den Schnittpunkten die jeweiligen Charakteristiken einer graphischen Repräsentation zugeordnet werden. BERTIN (1981) nutzt in seiner Veröffentlichung dafür durchgängig die visuelle Variable Größe in Form kreisrunder Flächen unterschiedlichen Flächeninhalts.

OGRADY & OGRADY (2008, S.98) beschreiben das Raster als ein wesentliches ästheti-

ches Designprinzip. Im Prototypen wird die Wirkung des Rasters mit Hilfe durchgängig gleichmäßiger Kachelzwischenräume verstärkt. Es entsteht eine natürliche Führung des Blicks von den Reihen- und Spaltenköpfen zu der entsprechenden Kachel, oder bezogen auf die Aufgabe ausgedrückt: von der Aufgabe und dem Probanden zum entsprechenden Testdurchlauf. Größe fällt dabei als visuelle Variable heraus, da hierbei das Raster aufgebrochen werden würde — auch wenn man wie Bertin die unterschiedlich großen Kreise in ein Tabellenraster setzt, da durch das Gesezt der Ähnlichkeit eher die Kreise zueinander in eine räumliche Beziehung gesetzt werden, nicht die Linien der Tabelle.

Wählt man an dieser Stelle die visuelle Variable Farbe, ist das Ergebnis eine Heatmap. Der graphische Raum, der von einer Kachel eingenommen wird, repräsentiert einen Durchlauf, also die Bearbeitung einer Aufgabe durch einen Probanden. Die Farbe der Kachel repräsentiert den relativen Wert der Charakteristik, die momentan ausgewählt wurde. Bei einem Wechsel der Charakteristik ändern sich die Farbwerte aller Kacheln entsprechend der zu Grunde liegenden Daten der Charakteristik. Selektiert der User einen der Codes als zu visualisierende Charakteristik, werden die Kacheln aller Durchläufe, bei denen der Code auftrat, kontrastreich zu den anderen Kacheln eingefärbt. Wird ein weiterer Code ausgewählt, werden nur noch jene Kacheln eingefärbt, in deren Durchlauf beide Codes auftraten. Es verhält sich analog mit weiteren Codes. Die Hervorhebung einer Kombination aus bestimmten Codes erlaubt es dem Nutzer, Zusammenhänge zwischen Durchläufen, Aufgaben, Probanden oder Personas zu erkennen. Abb. 12 illustriert diese Funktion.

Die Variablen der Aufgaben und Probanden pro Durchlauf bilden eine flache Hierarchie der Daten: Die Werte, die eine Variable annehmen kann, sind stets ganzheitlich in Form des farbkodierten Kachelrasters im Blick. Das kommt der *Uncertainty Reduction Theory* zu Gute, wonach menschliche Betrachter bei Ungewissheit aktiv nach Informationen sucht (OGRAY & OGRADY, 2008, S. 88). Gewissheit wird durch Ähnlichkeit verstärkt und durch Unterschiede verringert. Die Farben der Kacheln sind zwar unterschiedlich, aber nicht ihre räumliche Struktur im Raster.

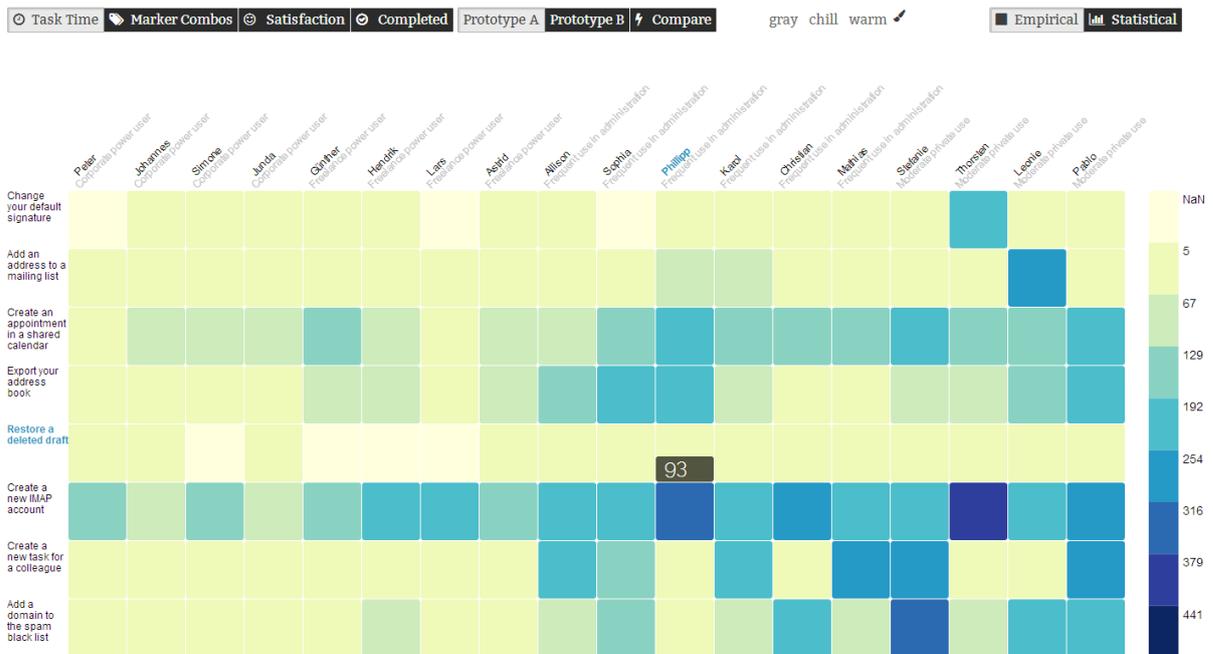


Abb. 9: Die Heatmap. Über den Kacheln sind die Probanden mit ihren Personas abgetragen, auf der linken Seite stehen die Aufgaben für den Test. Farbwerte kodieren Zahlenwerte der Aufgabenzeit, wie oben in den Buttonbars ausgewählt. Auf der rechten Seite ist die Legende, um den Farben Wertebereiche zuordnen zu können. Eine Kachel in der Mitte wird von der Maus überfahren und der konkrete Wert für den Probanden in der jeweiligen Aufgabe wird angezeigt.

DIFFERENZRASTER

Im Differenzraster liegt pro Kachel die Differenz zweier Werte einer Charakteristik vor. Diese beiden Werte stammen von zwei Prototypen, die direkt miteinander verglichen werden sollen. Die zwei stark unterschiedlichen Farbtöne der divergierenden Farbskala (s. Kapitel II. 4. 2.) repräsentieren jeweils einen der Prototypen. Eine Kachel bildet dann die Differenz der ausgewählten Charakteristik zwischen zwei Prototyp für den jeweiligen Probanden und jeweiligen Aufgabe ab. Es kann eindeutig anhand der Farbtöne bestimmt werden, welcher Prototyp bei welchem Probanden/Persona und bei welcher Aufgabe besser in der gewählten Charakteristik abgeschnitten hat.



Abb. 10: Im Differenzraster werden zwei Prototypen miteinander verglichen. In den Durchläufen der magentafarbenden Kacheln schnitt Prototyp A schlechter ab als Prototyp B, und in bei den grünen schnitt A besser ab als B.



Abb. 11: Das Differenzraster für die Charakteristik „Aufgabe erfolgreich“. Prototyp A erzielte offensichtlich wesentlich mehr Erfolge.

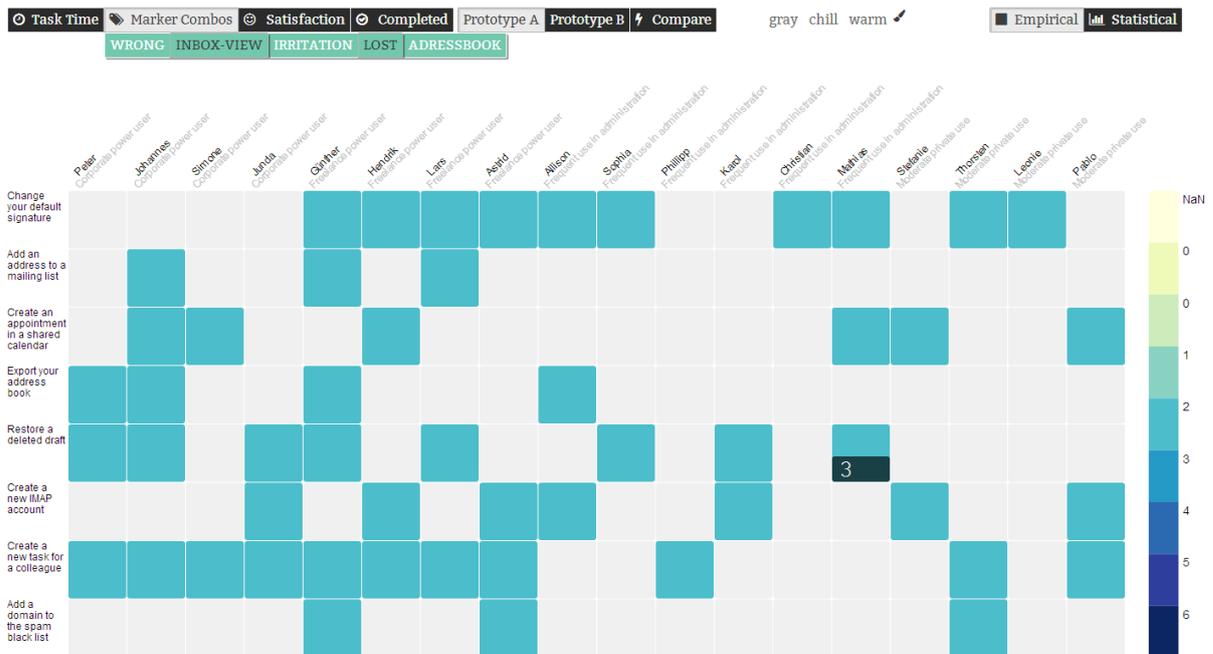


Abb. 12: Markerkombinationen. Die Durchläufe der farblich markierten Kacheln beinhalten den Code `INBOX-VIEW` und `LOST`. Warum verirren sich so viele Probanden mit Büroroutine so oft im Postfach bei Aufgabe 7?

III. 4. 2. Farben

QUANTISIERTE FARBEN

Die Farbe einer Kachel als graphische Repräsentation soll dem Betrachter primär dabei helfen, Trends in den visualisierten Ergebnissen zu finden und auf dem Weg dahin wesentliche Unterschiede in den Quantitäten zu erkennen. Quantisierte Farbfelder ermöglichen es, nah beieinander liegende Quantitäten durch einen gleichen Farbwert präattentiv zu gruppieren, so wie es auch bei Chloroplethenkarten der Fall ist. Die gleichen Farben werden auch bei den Markerquantitäten verwendet, wobei hier bis zu einer bestimmten Markeranzahl eindeutig ein Farbwert für eine Zahl steht.

FARBSKALEN

Sequentielle Farbskalen implizieren Ordnung zwischen den von Farbfeldern repräsentierten Klassen und sind damit für die Kodierung kardinaler Daten geeignet (Brewer et al.,

2014, S. 3). Um einen möglichst hohen Kontrast innerhalb der Skale zu erreichen, wird nicht nur die Helligkeit und Sättigung des jeweiligen Farbtons variiert, sondern auch der Farbton selbst. Der Prototyp bietet die Möglichkeit, die Farbskala auf Graustufen umzustellen, um auch bei Farbfehlsichtigkeiten eine Unterscheidbarkeit zwischen den Kacheln zu gewährleisten. Datenpunkte aus einem Differenzdatensatz werden mit einer quantisierten, divergierenden Farbskala kodiert, die um einen Messpunkt herum, in diesem Fall die Differenz 0, in stark unterschiedliche Farbtöne divergiert. Stark unterschiedlich heißt hier, dass die Farbtöne in entgegengesetzten Hälften des Farbkreises liegen und implizieren somit nicht nur eine Ordnung, sondern auch eine Qualität: Einer der Datenpunkte, aus dem der Differenzdatenpunkt gebildet wurde, war größer oder kleiner als der andere. Die Skala kodiert also Betrag (empfundene Helligkeit, Stärke) und Vorzeichen (Farbton) des Differenzdatensatzes. Abb. 12 soll die Unterschiede verdeutlichen.

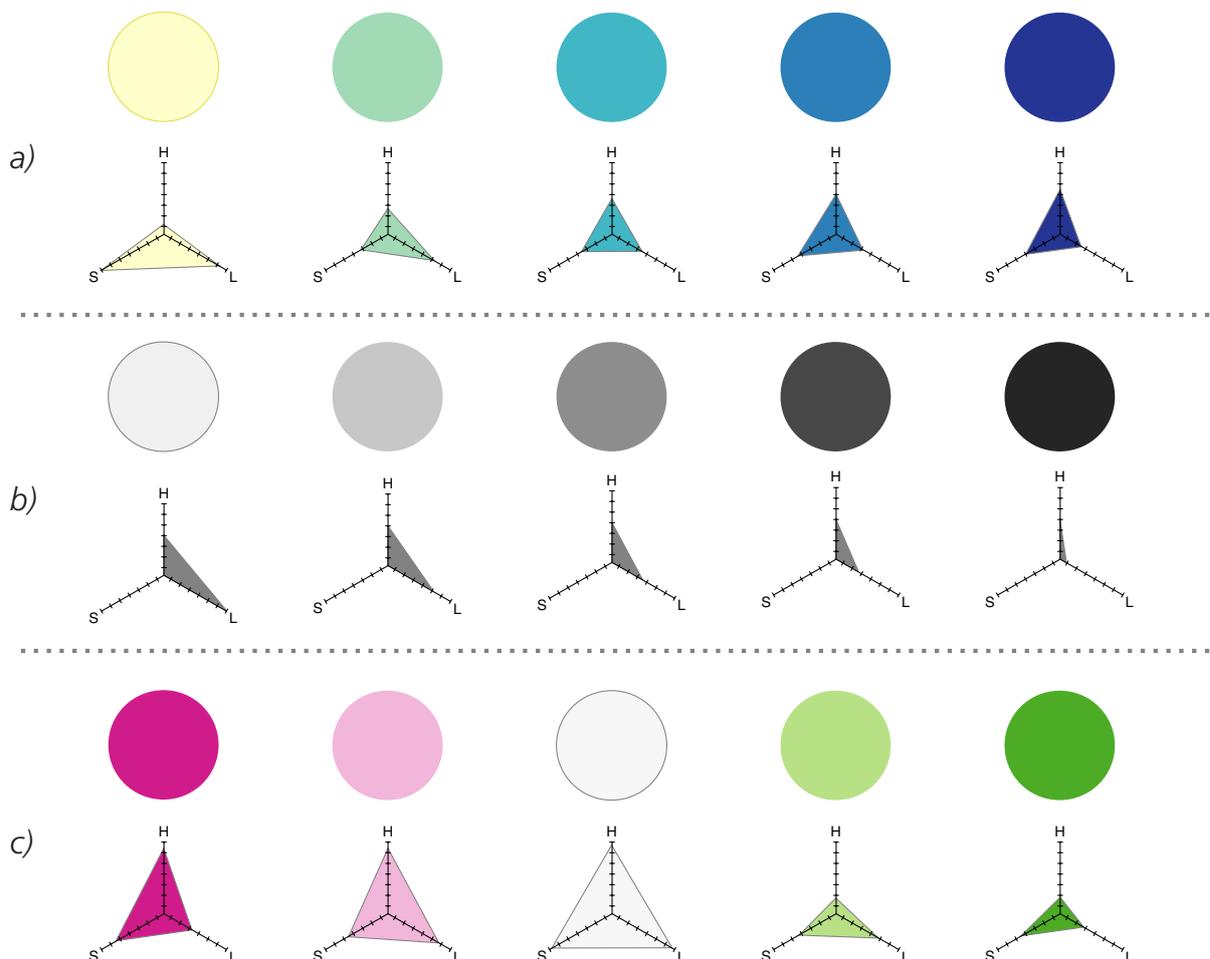


Abb. 13: Die verwendeten Farben mit ihren Farbtonen (H), Sättigungen (S) und Helligkeiten (L) auf Koordinatensystemen abgetragen. a) Sequentielle Skale: starke Unterschiede in der Helligkeit, schwache, aber bemerkbare im Farbton. b) Graustufen: lediglich Veränderungen in der Helligkeit. c) Divergierende Skale: Starke Farbtonsprünge

Die Farben könnten algorithmisch generiert werden, zum Beispiel unter Nutzung des HSL-Farbraums. Der H-Komponente, Hue, könnte eine Funktion zugeordnet werden, die einen Farbton nach Vorzeichen wählt. Den S- und L-Komponenten, Saturation und Lightness, könnten dann mit einer stetigen oder diskreten Funktion innerhalb der maximalen Amplitude des Betrags auf diesen abgebildet werden. Der Prototyp benutzt allerdings von BREWER (2014) entworfene Skalen (sequentiell in grün/blau, Graustufen, orange/rot und divergierend) mit je neun Farbfeldern.

III. 4. 3. Boxdiagramm

Im Boxdiagramm wird pro Aufgabe eine statistische Aufbereitung der Daten visualisiert. Im Gegensatz zu einem Balkendiagramm kodiert nicht nur die Länge des Balken, sondern auch sein Ursprungspunkt Information. Auf der Ordinate bleiben die Aufgaben erhalten und auf der Abszisse werden Prozente abgetragen. Die Breite der Balken kodiert die Breite des Konfidenzintervalls und sein Mittelpunkt markiert das Mittel aus der Stichprobe. Das Mittel selbst hat im Konfidenzintervall keine statistische Aussage, da Konfidenzintervalle eine Methode zur Berechnung des Bevölkerungsmittelpunktes sind, die nur mit einer gewissen Wahrscheinlichkeit, Signifikanz genannt, ein mögliches Ergebnis liefert, welches in der Regel nicht überprüft werden kann. Das Mittel wird daher in der Visualisierung nicht hervorgehoben.

Diese Ansicht stellt keine absoluten Werte bereit; es sind statistische Werte, die sich über alle Probanden auf das gesamte System beziehen. Der Nutzer kann abhängig von der Stichprobengröße folgende Aussage treffen:

„Mit einer Konfidenz von 95 % liegt der wahre Durchschnitt des Ereignisses zwischen $p-d$ und $p+d$ %. Oder anders ausgedrückt: Ein Bevölkerungsdurchschnitt zwischen $p-d$ und $p+d$ % ist plausibel, außerhalb dieser Grenzen ist er es nicht.“,

wobei p das arithmetische Mittel aus dem Test und d der Fehler aus der Berechnung des Konfidenzintervalls ist. Das Ereignis ist die jeweils untersuchte Charakteristik, so auch der Marker zum erfolgreichen Abschluss einer Aufgabe.



Abb. 14: Konfidenzintervalle für die Charakteristik „Zufriedenheit“. Aufgabe 6 schneidet auch unter Beachtung der oberen und unteren Grenzen, die die Konfidenzintervalle ausdrücken, gegenüber aller anderen Aufgaben schlecht ab. Es ist plausibel, dass die durchschnittliche Zufriedenheit nach einem Fragebogen bei 45 bis 55 % der realen Nutzer bei 1,6 bis 2,2 Punkten liegt.

III. 4. 4. Interaktion

WARD & YANG (2004) und YI et al. (2007) beschreiben in ihren Arbeiten Interaktionsoperanden aus Interaktionsräumen, auf die entsprechende Interaktionsoperator angewandt werden. Für den Prototypen sind diese Konzepte folgendermaßen definiert:

Bildschirmraum. Der Rendering-Bereich des Browsers, beschrieben durch HTML- und SVG-Markup und verknüpften CSS-Stilen. Das Kachelraster passt sich automatisch der Bildschirmauflösung an; Navigieren sollte sich also im Bestfall erübrigen. Die Auflösung dieses Raums, die Pixeldichte, ist wesentlich höher als die Dichte der noch unterscheidbaren graphischen Objekte und ihrer graphischen Beziehungen.

- » Kodieren. Der Nutzer kann nach Bedarf die Zoom-Funktion des Browser benutzen, um die gerenderten Elemente verlustfrei zu skalieren. Im eigentlichen Prototypen findet keine Transformation der Bildschirmpixel statt.

Datenstrukturraum. Die Daten sind strukturiert in multivariaten Datensätzen, einer pro

Durchlauf, und sind als Operanden in diesem Raum der primäre Aspekt in der Visualisierung. Zusammen mit der relativ zum Bildschirmraum niedrigen Operandendichte (siehe vorheriger Punkt) ist die flache Hierarchie der Datensätze der Grund für dieses Fokus. Das folgt nicht intrinsisch aus den Daten, sondern aus dem Design der Visualisierung.

- » *Erkunden.* Das Auswählen einer anderen Charakteristik aus dem Menü reko-
diert die Farben aller Kacheln, eine Zuordnung zu der entsprechenden Aufga-
be und Probanden bleibt durch das starre Raster gewährleistet.
- » *Filtern/Verbinden.* Im Code-Menü können Filter für Codes zugeschalten wer-
den und Kacheln, die der ausgewählten Filterkombination genügen, werden
farblich gekennzeichnet. Die Farbe der Kachel kodiert die Summe der ausge-
wählten Codes.
- » *Abstrahieren/Rekonfigurieren.* Die Boxdiagrammansicht abstrahiert die Wer-
te aller Probanden pro Aufgabe in Form einer statistischen Auswertung mit
Konfidenzintervallen.

Werteraum. Dieser Raum umfasst die Wertebereiche der einzelnen Variablen, oder Cha-
rakteristika je Proband und Aufgabe. Der Prototyp verfügt über keine Operatoren in
diesem Raum. Möglich wäre jedoch:

- » *Filtern.* Kacheln, die in einem spezifiziertem Wertebereich liegen, werden
hervorgehoben.

Attributraum. Hier werden die Variablen der graphischen Repräsentation betrachtet, wie
Größe, Farbe und Anordnung der Kacheln. Einziger relevanter Interaktionsoperator
ist das Kodieren — die Wahl der graphischen Repräsentation —, da die Attribute nur
über diese Funktion in Beziehung zu den Daten stehen.

- » *Kodieren.* Die Farbkodierung kann zwischen verschiedenen Farbpaletten, da-
runter auch Graustufen, gewechselt werden.

Visualisierungsraum. Dieser beinhaltet graphische Objekte aus der Datenstruktur und aus
Metadaten, wie zum Beispiel die Namen der Probanden und Aufgaben, die Abszisse-
nachse in des Boxdiagramms oder die Kacheln als Quadrate selbst.

- » *Auswählen/Verbindung.* Das Überfahren mit dem Cursor der Kacheln führt

Hervorhebung der dazugehörigen Aufgabe und des dazugehörigen Probanden. Das ermöglicht eine schnelle Einordnung.

Ward listet noch den Objektraum, in dem Daten auf transformierbare Objekte im 3D-Raum abgebildet wird, der allerdings für diesen Prototypen keine Rolle spielt.

III.5. Workflow

Neben der Güte der Visualisierung ist auch die Möglichkeit zur Integrierbarkeit in bestehende Arbeitsabläufe des UCD ein Ziel dieser Arbeit. Übliche Vorgehensweisen bei einem Nutzertest wurden in den Vorbetrachtungen in Kapitel II. 1 Grundlagen erläutert.

Zum Zeitpunkt der eigentlichen Datenvisualisierung müssen die Marker zur Verfügung stehen, ansonsten gibt die Heatmap und das statistische Boxdiagramm nur Aufschluss über Aufgabenzeit, Zufriedenheit und den Erfolg oder Misserfolg einer Aufgabe, jeweils pro Prototyp. Wie in den Vorbetrachtungen erwähnt, gibt es bereits Softwarepakete, die das Markieren besonderer Vorkommnisse ermöglichen. Diese Methode ist demnach einem gewissen Teil der Zielgruppe nicht fremd. Die Frage nach einer Evaluierung der Akzeptanz unter diesen Nutzern wird im Kapitel V Evaluierung gestellt.

Die folgenden Erläuterungen geben Aufschluss darüber, wie die Arbeit mit einem fertigen System vonstattengehen kann. Abb. 15 zeigt eine Übersicht des angepassten Workflows. Einige der Features sind nicht im Prototypen enthalten, siehe dazu das Kapitel VI Ausblick.

III. 5. 1. Testdesign

Während der Vorbereitung des Nutzertests sollten – abhängig von den Testparametern – Ziele, Aufgaben, Probanden und dem Design des Prototypen, Codes vorgeschlagen werden, die beim Test eine Rolle spielen könnten. Das umfasst nicht erschöpfend:

- » UI-Elemente, denen besonderer Gestaltungsaufwand zukam,

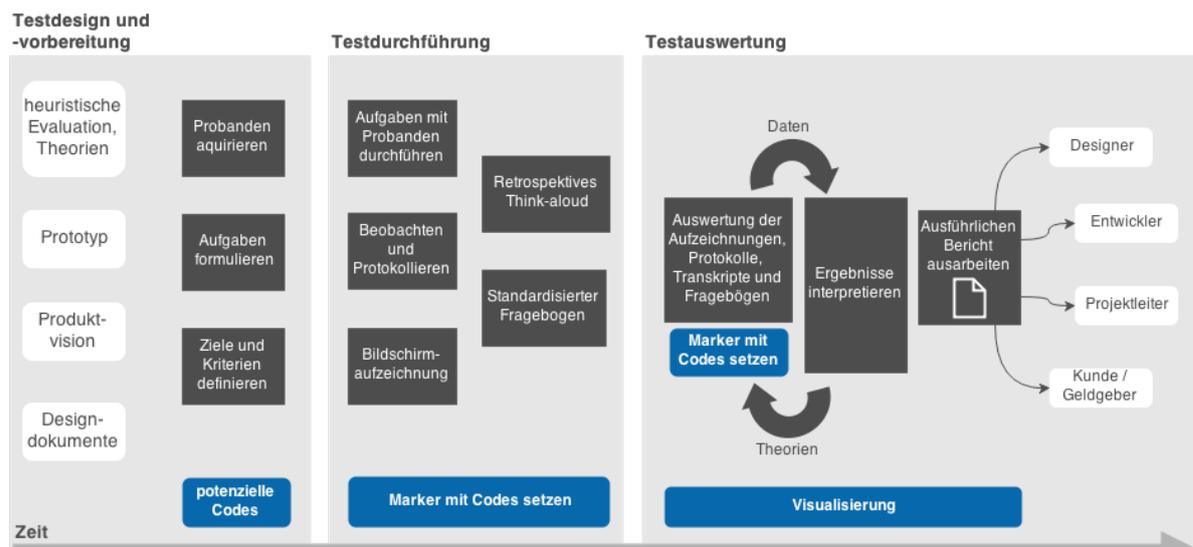


Abb. 15: Der angepasste Workflow für Nutzertests. Blau hinterlegt sind die neuen Schritte.

- » Erwartete Reaktionen der Probanden,
- » Fehler, wie falsch geklickte Buttons und langwierige Suchen und
- » Theorie-relevante Codes, deren Auftreten eine bestimmte Theorie bestätigen kann.

Bis auf den letzten Punkt stellen die Codes Aspekte dar, die ohnehin vor einem Test in Betracht gezogen werden.

III. 5. 2. Testdurchführung

Jede Beobachtung, die die Interpretation der Testergebnisse beeinflussen kann, kann durch einen Marker mit Code und Zeitpunkt repräsentiert. Die Codes bestehen aus einem Teil vorgefertigter Codes aus den Testvorbereitungen und einem Teil, der während und des Tests ad hoc festgelegt werden. Dank der Client-Server-Architektur kann dies in Echtzeit und von mehreren Punkten aus verfolgt werden, zum Beispiel vom Protokollanten im Beobachtungsraum an seinem Laptop und von der Moderatorin auf ihrem Tabletcomputer.

Sieht der Test ein retrospektives Think-aloud vor, kann währenddessen ebenfalls wertvolle qualitative Daten durch die Kommentare der Probanden in Form von Codes gesammelt werden. Am Ende dieses Schrittes oder zu Beginn des nächsten sollten die Angaben zur Aufgabenzeit, die Punktzahl aus dem Fragebogen und ob die Aufgabe erfolgreich abgeschlossen wurde in das System übernommen werden.

III. 5. 3. Testauswertung

An dieser Stelle stellt das System bereits eine Visualisierung bereit. Während der Analyse können noch weitere Codes hinzugefügt werden, wenn sich der Fokus der Analyse durch neue Erkenntnisse verschoben hat, neue Muster ergeben oder weil Details übersehen worden sind.

Die Visualisierung ist ein Werkzeug zur Erschließung und Interpretation der Daten; sie und die erfassten Codes und deren Marker als Form einer qualitativen Datenerfassung treten jedoch nicht ersatzbietend an die Stelle herkömmlicher Analysemethoden. Eine Evaluierung muss klären, inwiefern eine Arbeitserleichterung geboten werden kann.

IV ANEKDOTISCHES FEEDBACK

Bei einer Feedbackrunde wurde der Prototyp drei UI- und User-Experience-Designern, die mit dem Prozess des Nutzertests vertraut sind, vorgestellt und erklärt. Es wurden danach folgende Fragen gestellt und beantwortet:

Wäre das Setzen der Marker eine Behinderung im Test-Workflow?

Morae®-User (s. Kapitel II. 3. 1.) sind das Setzen von Markern während der Test-Sitzung gewohnt. Oft ist zusätzlich zum Moderator noch Protokollant vor Ort, der dann Marker setzen könnten. Spezielle Eingabegeräte könnten den Aufwand weitestgehend auf ein Minimum reduzieren, es lege letztendlich an der Gebrauchstauglichkeit der Marker-Software. Neben Tabletcomputern wurden auch spezifisch eingefärbte und beschriftete Tastaturen vorgeschlagen. Es bestünde die Möglichkeit, sofort nach dem Test die Visualisierung zur Hand zu haben oder auch ohne Aufnahmemöglichkeit rekonstruierbare Ergebnisse zu produzieren.

Könnten Testergebnisse besser mit dem Projektkunden oder Projektleiter kommuniziert werden?

Die Gesamtübersicht über Aufgabenzeit oder erfolgreich abgeschlossene Aufgaben eigne sich besonders zur Einleitung des ausführlichen Testreports und gibt einen besseren Überblick als eine Tabelle. Der Kunde könne sich in einer Ergebnispräsentation selbst interaktiv, geleitet durch die Moderatorin, die Ergebnisse erforschen. Auch eine Kommentarfunktion pro Kachel, deren Inhalt angemessen in den graphischen Raum eingebettet wird, kann hilfreich sein, um Informationen zwischen Teammitgliedern auszutauschen. Die vergleichende Ansicht mit der divergierenden Farbskala verdeutliche besonders effizient die Unterschiede in der Zufriedenheit zwischen zwei Prototypen. Für eine ausführliche Analyse des semantischen Differentials, wenn eingesetzt, sei es aber kein Ersatz, da nicht

die einzelnen Items miteinander verglichen werden.

Lassen sich quantitative Werte wie Aufgabenzeit und Zufriedenheit angemessen innerhalb einer Aufgabe einschätzen?

Die farblich kodierten Kacheln helfen dabei, problematische Durchläufe schnell auffindig zu machen. Die Effizienz könne verbessert werden, indem für jede Aufgabe die Farbskala angepasst werden könnte. Für jede Aufgabe definiert der Testdesigner eine minimale Aufgabenzeit und eine kritische maximale Zeit, nach der eine Aufgabe „zu lange“ dauert. Innerhalb dieser Spanne werden dann die Werte den Farbfeldern zugewiesen.

Ist die visuelle, statistische Auswertung nützlich?

Sie gebe ein zusammenfassendes und vergleichendes Bild für die einzelnen Aufgaben. Die Effizienz könne verbessert werden, indem auch hier die Balken entsprechend ihrer Position mit der jeweiligen Farbe kodiert werden. Bestimmte Durchläufe mit auffallenden oder unvorhergesehenen Ergebnissen sollte man aus der statistischen Auswertung entfernen können und sie später individuell genau untersuchen. Eine Vergleichsansicht wie das Differenzraster für die Heatmap wäre hier wünschenswert, um die Statistiken der beiden Prototypen miteinander vergleichen zu können.

Kann es bei der Interpretation helfen, Code-Kombination einzusehen?

Es könne hilfreich sein, die Vielzahl an Markern in ihrem Auftreten strukturiert zu sehen.

Welche Aspekte wären bei einer Evaluierung mit validen und reproduzierbaren Ergebnissen zu beachten?

Wichtig wäre ein Vergleich zu bestehenden, üblichen Visualisierungen wie dem klassischen Balkendiagramm für quantitative Daten wie der Aufgabenzeit und den damit einhergehenden Workflows. Die Probanden der Evaluierung sollten sich über alle Möglichkeiten des Visualisierungssystems im Klaren sein und im Bestfall im Vorfeld die Möglichkeit haben, das System selbst auszuprobieren. Besonders interessant sei hier eine Evaluierung mit Nutzern der Morae®-Software, da diese bereits die Arbeit mit Markern innerhalb einer Software gewohnt sind.

Ein Designer betonte das angenehme Umschalten zwischen Farbskalen und Charakteristika wegen des weichen Gradienten beim Farbwechsel. Unabhängig von den Fragen gab es viele Anregungen zur Usability des Prototypen, einschließlich:

- » Gruppierungen: Header von Body trennen, Überschriften für Buttongroups, größere Abstände zwischen Buttongroups und
- » Kohärenz: Dunkle Farbskalenwerte implizieren negatives Ergebnis, helle ein positives Ergebnis.

V EVALUIERUNG

Eine valide, reproduzierbare Evaluierung des Prototypen ist nicht Teil dieser Arbeit, dennoch soll an dieser Stelle ein Überblick über mögliche Ansatzpunkte für eine Evaluierung geboten werden. MUNZNER (2010) beschreibt für jeden Schritt zum Visualisierungsdesign eine mögliche Downstreamvalidierung, die voraussetzt, dass alle nachfolgenden Schritte bereits umgesetzt worden sind. Das heißt, dass ein benutzbarer Prototyp zur Verfügung stehen muss. Dieser wiederum wird zusammen mit dieser Arbeit vorgelegt.

Eine zukünftige Evaluierung muss sich an drei Gütekriterien messen:

- » *Objektivität*: Die Ergebnisse sind unabhängig von den Rahmenbedingungen, insbesondere den Testleitern, entstanden.
- » *Reproduzierbarkeit*: Die Ergebnisse sind wiederholbar in ähnlichen Testdesigns und somit statistisch signifikanter für die Gesamtbevölkerung.
- » *Validität*: Die Testmethodik misst tatsächlich die Güte der Visualisierung und den Mehrwert, den sie der Zielgruppe bringt. Die folgenden Punkte konzentrieren sich besonders auf dieses Kriterium.

V.1. Problembeschreibung

Unmittelbar können die Ergebnisse aus der Domänencharakterisierung überprüft werden, in dem die Zielgruppe befragt und bei der Arbeit beobachtet wird, um die Charakterisierung zu überprüfen. Die Frage lautet an dieser Stelle, ob es das beschriebene Problem überhaupt gibt. Die in dieser Arbeit verwendete Charakterisierung entstand zum Teil aus der Zusammenarbeit mit der Zielgruppe und wurde als valide angenommen.

Eine Downstreamvalidierung wäre die Beobachtung der Akzeptanzrate oder die Einschätzung eines marktwirtschaftlichen Interesses für den Visualisierungsprototypen. Da dieser mit einer Anpassung des Nutzertest-Workflows einhergeht, müsste dazu zunächst

die gesamte Anwendung, nicht nur die Visualisierungskomponente, zur Testreife gebracht werden.

V.2. Abstraktion des Problems

Anekdotischen Feedbacks durch Vertreter der Zielgruppe kann die korrekte Abstraktion des Problems, das Abbilden des Problems auf Strukturen und Prozesse der Datenvisualisierung und Informatik, im Downstream validieren. Dies wird in Kapitel IV ausgeführt.

Eine nachhaltige und effektivere Methode wäre hier eine gründliche Feldstudie, der die Zielgruppe bei der Nutzung der Anwendung und des angepassten Testprozesses über einen längeren Zeitraum hinweg untersucht. Die Ergebnisse wären qualitativer Natur, aber weniger anekdotisch, da aus einem größeren Pool praxisnaher Beobachtungen geschöpft wird.

Wie in der Feedbackdiskussion angeführt, ist besonders ein Vergleich mit klassischen Best-Practice-Methoden zur Analyse interessant; sowie Probanden mit Erfahrung im Umgang mit TechSmith® Morae® zur Evaluierung heranzuziehen, um den Prozess des Markierens mit einer etablierten Methode aus der Industrie zu vergleichen.

V.3. Kodierungs- und Interaktionsdesign

Dieser Schritt kann über die Argumentation mit etablierten Prinzipien aus Kommunikationsdesign und Datenvisualisierung validiert werden.

MUNZNER empfiehlt für diesen Schritt einen kontrollierten, summativen Nutzertest unter Laborbedingungen, der dann selbstverständlich mit einer bereits validierten Methodik ausgewertet wird. Dieser Test in einem kontrollierten Umfeld soll Fehler in feinkörnigen Aspekten des Visualisierungsdesigns offenbaren. Ein vergleichender Test mit klassischen Analysemethoden und der Morae®-Software ist ebenfalls erwägenswert.

Informelle Usability-Tests sind ungeeignet, um die Überlegenheit dieser Visualisierung gegenüber einer anderen Form der Analyse zu beweisen. Solche formativen Tests helfen allerdings beim Bau eines besseren Systems, was wiederum die Validierungen der anderen Schichten effektiver gestaltet. Validierungsergebnisse würden dann nicht aufgrund mangelhafter Bedienung getrübt werden. Siehe dazu Kapitel V. 5. für eine mögliche Herangehensweise.

V.4. Algorithmen- und Systemdesign

Eine Anforderung an den Prototypen war seine Lauffähigkeit auf einer breiten Auswahl stationärer und mobiler Endgeräte. Während seiner Entwicklung wurde die Server- und Clientkomponente stets logisch voneinander durch eine virtuelle Maschine getrennt und ausschließlich vom W3C vorgeschlagene oder standardisierte Technologien verwendet, vor allem HTML, CSS, SVG und AJAX.

Für eine Evaluierung sollten Klassen von Zielplattformen (Kombination aus Layout Engine, Betriebssystem und Hardware) abgegrenzt werden, die mindestens das gesamte Featureset unterstützen. Erforderlich ist weiterhin die Definition maximaler Systemreaktionszeiten und minimaler Bildrate beim Rendering, um die Leistung auf der jeweiligen Plattform zu beurteilen. Eine Zusammenarbeit mit der Zielgruppe ist erstrebenswert, um mögliche Plattformen, Szenarien und Konfigurationen abzustecken.

V.5. Usability-Testing des komplexen Systems

REDISH (2007) formuliert einen Leitfaden für das Usability-Testing komplexer System, die wie die Systeme, die es betrachtet, wesentlich umfangreicher und gründlicher durchgeführt werden müssen. Der vorgestellte Prototyp sei als "komplex" anzunehmen, da er einige der aufgezählten Eigenschaften komplexer Systeme inne hat (REDISH, 2007, S. 103):

- » Es stehen möglicherweise mehr Informationen zur Analyse bereit, als bei initialer Sichtung verarbeitet werden können.
- » Datenanalyse und rekursive Entscheidungsfindung stehen zueinander in

Wechselwirkung.

- » Informationen sind unvollständig. Der Datensatz repräsentiert im Bestfall den Testablauf dank der Videoaufzeichnung und der Datenaufbereitung und -visualisierung, gibt aber naturgemäß keinen vollständigen Einblick in die Köpfe der Probanden.
- » Die Nutzer sind Experten in ihrer Domäne.
- » Analysten und Entscheidungsträger sind unterschiedliche Leute, zum Beispiel Usability-Experte, Interface Designer, Projektleiter und Auftraggeber.

Die formative Usability-Evaluierung stellt daher eine besondere Herausforderung dar. Nicht nur Spezialisten aus der Domäne, die eigentliche Zielgruppe, sondern auch Experten aus anderen Disziplinen sollten hinzugezogen werden (REDISH, 2007, S. 105). Analysten aus der qualitativen Sozialforschung oder Informationsdesigner und -architekten können einen entscheidenden Beitrag leisten. Starke Aufmerksamkeit muss auch der Suche nach geeigneten Probanden aus der Zielgruppe, oder einem repräsentativen Ersatz, und geeigneten Szenarios zukommen, um falsche Positivs oder falsche Negativs zu vermeiden. Das gilt sicher für Usability-Testing im Allgemeinen, jedoch stehen hier aufgrund der spezialisierten Domäne wesentlich weniger Leute und Raum für Tests bereit.

Testdesigner müssen sich auch über die schwierige Formulierung von Zielen und Aufgaben im Klaren sein. Ein Test mag mit der Aufgabe „Benutzen mindestens 50% der Besucher des Webshops die Merkliste?“ beginnen, aber der Proband formuliert vielleicht die Aufgabe um, da er bereits einen schlecht zu erkennenden Button oder ein wirres Merklisten-Layout verdächtigt. In diesem Fall untersucht er die anderen Durchläufe ebenfalls auf Probleme im Zusammenhang mit diesen Codes. Nicht nur die Aufgabe, auch das Ziel hat sich geändert. Eine Evaluierung muss diese Fälle abfangen und reproduzierbare Ergebnisse liefern.

Die genannten Punkte gelten für Usability-Tests im speziellen, aber auch für die anderen Evaluierungen der Schichten nach MUNZNER (2010), wie der Feldstudie zur Validierung der korrekten Problemabstraktion, welche nebenbei auch von REDDISH als Exempel genannt wird.

VI AUSBLICK UND FAZIT

Die Güte der in dieser Arbeit vorgestellten Visualisierung wurde lediglich anhand einer anekdotischen Feedbackrunde evaluiert. Es mangelt noch an einer Evaluierung mit reproduzierbaren Ergebnissen. Mögliche Vorgehensweisen und Aspekte dazu wurden in Kapitel V erläutert. Im Rahmen des Feedbacks wurden auch einige Ansätze zur Verbesserung und Erweiterung des Interaction Designs aufgezeigt, die iterativ in Abstimmung mit der Zielgruppe implementiert werden können. Als Ergänzung sei hier noch auf RESTfulness (URLs kodieren den Systemzustand) hingewiesen, die zwar im System beim Umgang mit den Ressourcen (Datenmodelle für Probanden, Aufgaben, Durchläufe, Projekte etc.), aber nicht bei der Visualisierungsanwendung in den URLs Anwendung findet. Der aktuelle Zustand der Visualisierung, wie beispielsweise die gewählte Codekombination, wird nicht durch die URL repräsentiert, wodurch der Zustand nicht durch die URL allein zwischen zwei Systeminstanzen übertragen werden kann. Es sollte auch möglich sein, markierte Codes zusammenzufassen, wenn sie das gleiche Muster ausdrücken. Dies kann unter anderem zu Stande kommen, wenn mehrere Analysten das Material sichten oder wenn sich einige Codes während der Analyse als ähnlich herausstellen. Nicht zuletzt sei darauf hingewiesen, dass das System zur Erfassung der Daten sowohl auf Server- als auch auf Client-Seite noch ausbaufähig ist und bisher nur ein grundlegendes Featureset bietet.

Der Prototyp aus dieser Arbeit verfügt über keine Möglichkeit, durch Permutationen nach Bertin zu clustern. Tabellenreihen und -spalten bleiben starr an ihrer Position und werden nur durch die Reihenfolge der Aufgaben und Probanden in der Datenbank bestimmt. Clustern als Strukturentdeckung findet wie im vorangegangenen Kapitel beschrieben durch Codekombinationen statt. Ein konsequenter Ansatz zur Erweiterung des Prototypen ist das Clustern via Permutationen. Reihen und Spalten können nach einer Charakteristik geordnet werden, um Probanden und Aufgaben mit ähnlichen Ergebnissen zu finden. Die Sortierung, oder Permutation, könnte auch durch Betrachtung mehrerer Charakteristika mit Hilfe einer Datenbank-ähnlichen Abfrage erfolgen. Eine Abfrage könnte dann beispielsweise folgendermaßen aussehen: „Finde alle Durchläufe mit Markern für die Codes ‚IRRITIERT‘ und ‚WARENKORB‘ und sortiere nach der Aufgabenzeit.“

Sie sollte über Interaktion mit der GUI erfolgen und nicht verbalisiert werden müssen, um kreatives Erkunden zu unterstützen.

Ein weiterer Ansatz ist die Konstruktion der Heatmap selbst. In dieser Arbeit wurden kategorische Daten mittels farblich kodierten Kacheln über diverse Charakteristika zueinander in Beziehung gesetzt. Scatterplotmatrizen (SPLOMs) wurden anfangs ausgeschlossen, da sie ein Problem nicht ganzheitlich betrachten, unintuitiv für die Zielgruppe sind und zu Unübersichtlichkeit neigen. Das waren Annahmen, wie sie häufig im Zuge von Designstudien gemacht werden, um ein Konzept zu differenzieren. SPLOMs sind daher nicht grundsätzlich ungeeignet und ihre Tauglichkeit gilt es zu untersuchen. Vor allem Entwicklungen aus jüngerer Zeit wie die Arbeit von ALBUQUERQUE et al. (2009) sind hier relevant, in der die Scatterplots selbst innerhalb der SPLOM mittels eines im Vorfeld definierten Qualitätsmaßes geordnet werden. Zur Visualisierung wird hierbei der graphische Raum des einzelnen Scatterplots durch eine Farbkodierung des Qualitätsmaßes ersetzt. Die primär zu klärende Frage ist die Definition des Qualitätsmaßes, das die Beziehung der Scatterplots untereinander herstellt.

In dieser Arbeit wurde ein Framework zur Analyse und Visualisierung von Nutzertestdaten für nutzerzentrierte Designprozesse vorgestellt und anhand einer Beispielimplementierung in einen bestehenden Workflow eingebettet. Der Visualisierung wurde anhand anekdotischen Feedbacks Potential für einen produktiven Einsatz nachgewiesen. Die Implementierung der im Ausblick aufgezählten Änderungen und eine anschließenden Evaluierung soll die Frage klären, ob dieses Potential auszuschöpfen ist.

VII LITERATURVERZEICHNIS

- ALBUQUERQUE, G.; EISEMANN, M.; LEHMANN, D. J.; THEISEL, H. & MAGNOR, M. 2009. *Quality-Based Visualization Matrices Proceedings of Vision, Modeling, and Visualization*.
- BAER, K. & VACARRA, J. 2010. *Information Design Workbook: Graphic Approaches, Solutions, and Inspiration + 30 Case Studies*. Rockport Publishers.
- BERTIN, JACQUES 1983. *Semiology of Graphics*. University of Wisconsin Press.
- BERTIN, J.; BERG, W.J. & SCOTT, P. 1981. *Graphics and Graphic Information Processing*. De Gruyter.
- BOSTOCK, M. 2014: *D3.js Examples*. d3js.org (02. November 2014)
- BROOKE, JOHN 1996. *SUS: A quick and dirty usability scale*.
- CLEARLEFT LTD. 2014. *Clearleft Silverback*. <http://silverbackapp.com/> (02. November 2014)
- DUMAS, JOSEPH F. & REDISH, JANICE C. 1993. *A Practical Guide to Usability Testing*. Greenwood Publishing Group Inc. Westport, CT, USA
- EDWARDS, A.L. 1983. *Techniques of Attitude Scale Construction*. Irvington Publishers
- VON ENGELHARDT, J. 2002, *The Language of Graphics: A Framework for the Analysis of Syntax and Meaning in Maps, Charts and Diagrams Institute for Language, Logic and Computation*. Universiteit van Amsterdam ILLC dissertation series
- FRIENDLY, MICHAEL 2009. *The history of the cluster heat map*. The American Statistician
- GOULD, JOHN D. & LEWIS, CLAYTON 1985. *Designing for Usability: Key Principles and What Designers Think*. ACM, 28, 3, March 300–311, New York, USA
- HARROWER, MARK & BREWER, CYNTHIA A. 2003. *ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps Cartographic Journal*. The Maney Publishing, June, 27–37
- HARTIGAN, J.A. 1975. *Printer graphics for clustering Journal of Statistical Computation and Simulation*. Journal of Statistical Computation and Simulation, Volume 4, Issue 3, 187 ff.
- EN ISO 9241 1995. *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten*. Geneva, Switzerland
- MILES, M.B.; HUBERMAN, A.M. & SALDAÑA, J. 2013. *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications
- MUNZNER, TAMARA 2009. *A Nested Model for Visualization Design and Validation*. IEEE Transactions on Visualization and Computer Graphics.
- NIELSEN, JAKOB 1995. *Usability Inspection Methods*. Conference Companion on Human Factors in Computing Systems CHI, ACM, New York

- NIELSEN, JAKOB 1994. *Usability Engineering*. Elsevier.
- O'GRADY, J.V. & O'GRADY 2008. *The Information Design Handbook*. RotoVision
- OSGOOD, C.E.; SUCI, G.J. & TANNENBAUM, P.H. 1957. *The Measurement of Meaning*. University of Illinois Press
- REDISH, JANICE (GINNY) 2007. *Expanding Usability Testing to Evaluate Complex Systems*. JUS Journal of Usability Studies 2 3 102–111
- RICHTER, M. & FLÜCKIGER, M. 2007. *Usability Engineering kompakt: benutzbare Software gezielt entwickeln*. Elsevier GmbH
- SAURO, J. & LEWIS, J.R. 2012 *Quantifying the User Experience: Practical Statistics for User Research*. Elsevier/Morgan Kaufmann
- SHNEIDERMAN, BEN 1996. *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. IEEE Computer Society, Proceedings of the 1996 IEEE Symposium on Visual Languages
- STRÜBING, J. 2008. *Grounded Theory*. VS Verlag für Sozialwissenschaften GmbH
- TECHSMITH CORPORATION 2014. TechSmith® Morae® <http://www.techsmith.com/> (02. November 2014)
- WARD, M.O.; GRINSTEIN, G. & KEIM, D. 2010. *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters, Ltd.
- WARD, MATTHEW O. & JING YANG 2004. *Interaction Spaces in Data and Information Visualization*. Eurographics Association
- YI, JI SOO; KANG, YOUN AH; STASKO, JOHN & JACKO, JULIE 2007. *Toward a Deeper Understanding of the Role of Interaction in Information Visualization*. IEEE Transactions on Visualization and Computer Graphics