



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Anwendungen 2

Torsten Rauschan

Usability Metrics

Torsten Rauschan

Usability Metrics

Ausarbeitung eingereicht im Rahmen des Moduls Anwendungen 2
im Studiengang Informatik (Master)
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuer : Prof. Dr. Olaf Zukunft

1. Prüfer: Prof. Dr. rer. nat. Kai von Luck
2. Prüfer: Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 31. August 2010

Inhaltsverzeichnis

1 Motivation	4
2 Verwandte Forschungsergebnisse	5
2.1 QUIM	5
2.1.1 Nutzungsqualität (Quality in use)	5
2.1.2 Faktoren (Factors)	6
2.1.3 Kriterien (Criteria)	7
2.1.4 Metriken (Metrics)	7
2.1.5 Daten (Data)	8
2.1.6 Zusammenhang der Ebenen	9
2.2 SUM	9
2.2.1 Quantitatives Modell	10
2.2.2 Hauptkomponentenanalyse	10
2.2.3 Standardisierung	11
2.2.4 Erstellung der SUM	12
2.3 UTUM	12
2.3.1 Testprozess	12
2.3.2 Metriken	13
2.4 Abgrenzung zur eigenen Arbeit	14
3 Schluss	16
3.1 Zusammenfassung	16
3.2 Ausblick	16
Literaturverzeichnis	18

1 Motivation

Im Usability Labor der Hochschule für angewandte Wissenschaften Hamburg (HAW Hamburg) wurden Usability Metriken bisher vernachlässigt. Es gibt aktuell nur einen ersten Ansatz, der auf die Erfassung von Maus- und Tastaturdaten beschränkt ist. Daher wurde im Modul Projekt 1 damit begonnen, diese Software komplett zu überarbeiten.

Der erste Ansatz besteht darin, bestimmte Thesen mit Zahlen zu unterstützen. Später wird es möglich sein, mit Hilfe von Metriken komplette empirische Studien durchzuführen. Darüber hinaus sollen Meta-Analysen über verschiedene Projekte durchführbar sein.

In dieser Ausarbeitung wurden vergleichbare Arbeiten gesucht und ausgewertet. Aus diesen wurden die in Kapitel 2 beschriebenen Arbeiten ausgewählt. Diese drei haben sich ebenfalls grundsätzlich mit Usability Metriken beschäftigt und diese auch detailliert beschrieben. Daher haben diese drei Arbeiten den größten Einfluss auf das parallel durchgeführte Projekt gehabt und sind daher von größerer Bedeutung.

2 Verwandte Forschungsergebnisse

In diesem Kapitel werden Projekte vorgestellt, die sich ebenfalls mit dem Thema Usability Metrics befassen. Dabei handelt es sich um eine Auswahl von drei Paper, die jeweils unterschiedliche Ansätze zur Erfassung und Interpretation von Metriken darstellen.

Im Folgenden wird auf die grundsätzlichen Eigenschaften der Modelle und die beschriebenen Metriken eingegangen.

2.1 QUIM

Das Paper „*QUIM: A Framework for Quantifying Usability Metrics in Software Quality Models*“ von [Seffah u. a. \(2001\)](#) beschreibt das Quality in Use Integrated Measurement-Framework. Dabei handelt es sich um ein hierarchisches Modell, welches in vier Level unterteilt ist.

- Faktoren
- Kriterien
- Metriken
- Daten

Abbildung 2.1 zeigt die Struktur des Frameworks detailliert. Im Folgenden werden die dargestellten Ebenen beschrieben:

2.1.1 Nutzungsqualität (Quality in use)

Die Quality in use beschreibt die Qualität der Software aus Sicht des Benutzers. Dabei wird nur die Benutzung der Software betrachtet, sowie deren Performance und ihr Nutzen. Der innere Aufbau spielt aus Sicht des Benutzers keine Rolle.

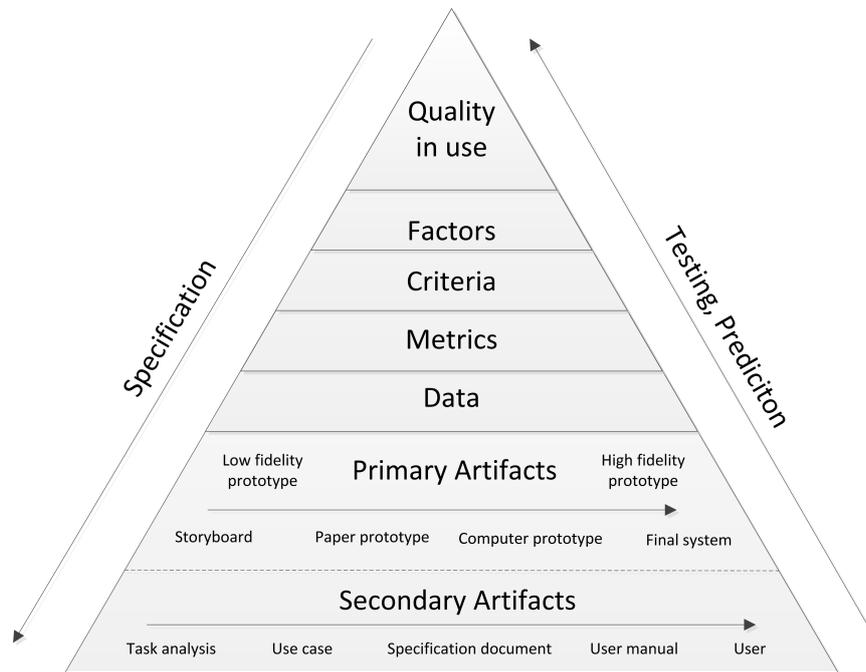


Abbildung 2.1: QUIM: Struktur und Verwendung (Seffah u. a., 2001)

2.1.2 Faktoren (Factors)

Die Faktoren bestimmen die Qualität der Software. Folgende Faktoren sind in QUIM beschrieben:

- **Effektivität:** Beschreibt den Grad der Fehlerfreiheit und Vollständigkeit, mit der der Benutzer eine bestimmte Aufgabe in einem bestimmten Kontext löst.
- **Effizienz:** Die Anzahl der verwendeten Ressourcen, in Relation zur Fehlerfreiheit und Vollständigkeit, mit denen der Nutzer ein angestrebtes Ziel mit dem System erreicht.
- **Nutzerzufriedenheit:** Wird durch die Abwesenheit von Problemen und eine positive Erfahrung im Umgang mit dem System geprägt.
- **Produktivität**
- **Sicherheit**
- **Internationalität:** Beschreibt den Grad, in wie weit das System für den internationalen Markt nutzbar ist. Dafür werden regionale, sprachliche und kulturelle Variationen betrachtet.

- **Erreichbarkeit:** Beschreibt die Breite der unterschiedlichen Personengruppen, die komfortabel mit dem System arbeiten können. Dazu zählen Nutzergruppen, die Assistenztechnologien (z.B. Bildschirmlupe, Sprachsteuerung) nutzen müssen.

2.1.3 Kriterien (Criteria)

Kriterien sind in QUIM, wie auch in einigen anderen Modellen, Subfaktoren. Das spezielle an den QUIM-Faktoren ist, dass sie wiederum durch mehrere Metriken gemessen werden. Im Folgenden werden einige beispielhafte Kriterien beschrieben:

- **Attraktivität:** Stellt einen Indikator dar, in wieweit der User das System während der Benutzung als hilfreich empfunden hat.
- **Konsistenz:** Attribute, die Einfluss auf die Konsistenz des User Interfaces haben.
- **Minimaler Aufwand:** Der minimale Aufwand, den der Nutzer betreiben muss, um eine spezielle Aufgabe zu lösen.
- **Befehle merken:** Die minimale Anzahl an Befehlen, die der Nutzer behalten muss, um eine spezielle Aufgabe zu lösen.
- **Vollständigkeit:** Das Ausmaß, indem der Nutzer seine Aufgabe mit Hilfe des Systems komplett lösen kann.

2.1.4 Metriken (Metrics)

Der IEEE-Standard definiert den Begriff Metrik wie folgt: Metriken können als Funktionen verstanden werden, die als Eingabe die Daten der Software besitzen und als Ausgabe wird ein numerischer Wert geliefert, der als Attribut interpretiert werden kann. Dieser beeinflusst die Qualität der Software.

Im Kontext von QUIM wird die Ausgabe der Funktion als ein numerischer Wert definiert, der den Status eines speziellen Kriteriums summiert. QUIM definiert ca. 100 Metriken, von denen einige Funktionen sind und andere einfache, zählbare Werte darstellen.

Als Beispiel werden im Folgenden zwei Metriken vorgestellt, die allgemein gehalten und so für die meisten Softwaresysteme anwendbar sind.

- **Aufgaben Übereinstimmung (Task Concordance):** Diese Metrik zeigt an, wie die Häufigkeit der Aufgaben zur Schwierigkeit der Aufgabe passt. Im Allgemeinen wird es bei einem guten System dazu kommen, dass häufig gestellte Aufgaben mit der Zeit besser gelöst werden. Um diese Metrik zu berechnen ist es zuerst nötig alle Aufgaben,

absteigend nach der erwarteten Häufigkeit sortiert, aufzulisten. Die Metrik wird nach folgender Formel berechnet:

$$A\ddot{U} = \frac{100 * D}{\frac{N*(N-1)}{2}} \quad (2.1)$$

Dabei stellt N die Anzahl der betrachteten Aufgaben dar. Mit D ist der „Discordance Score“ gemeint. Dieser wird gebildet, indem man die Aufgaben, deren Schwierigkeiten nicht in erwarteter Reihenfolge (NER) auftreten, von denen die in erwarteter Reihenfolge auftreten (ER), abzieht.

$$D = ER - NER \quad (2.2)$$

- **Visuelle Kohärenz (Visual Coherence):** Die visuelle Kohärenz zeigt an, wie gut ein User Interface zusammengehörige Komponenten auch zusammen anzeigt bzw. unabhängige Komponenten trennt. Sie wird nach folgender Formel berechnet:

$$VK = \frac{100 * G}{\frac{N*(N-1)}{2}} \quad (2.3)$$

Wobei G die Anzahl zusammengehöriger Komponenten und N die Gesamtanzahl an Komponenten in einer Gruppierung angibt.

2.1.5 Daten (Data)

Die Daten bilden die Grundlage der Metriken. Diese können qualitativ oder quantitativ sein. Darüber hinaus gibt es zwei unterschiedliche Methoden, um aus den Daten eine Metrik zu generieren.

- **Zählbar:** Einige Daten sind zählbar. In diesem Fall ist es möglich diese Daten als Metrik zu betrachten und direkt als Kriterium zu nutzen. Ein zählbares Datum ist zum Beispiel die Aufgabenzeit. Zusätzlich fallen Daten aus Fragebögen in diese Kategorie. Diese dienen als Eingabe für statistische Metrikfunktionen.
- **Berechenbar:** Andere Daten wiederum erfordern einen beliebig komplexen Rechenschritt, um aussagekräftig zu werden. Ein Beispiel dafür, ist die Aufgabenvollständigkeit in Prozent. Aus diesem Datum kann die Aufgaben-Effektivitäts-Metrik (AE) errechnet werden.

$$AE = \frac{\text{Quantität} * \text{Qualität}}{100} \quad (2.4)$$

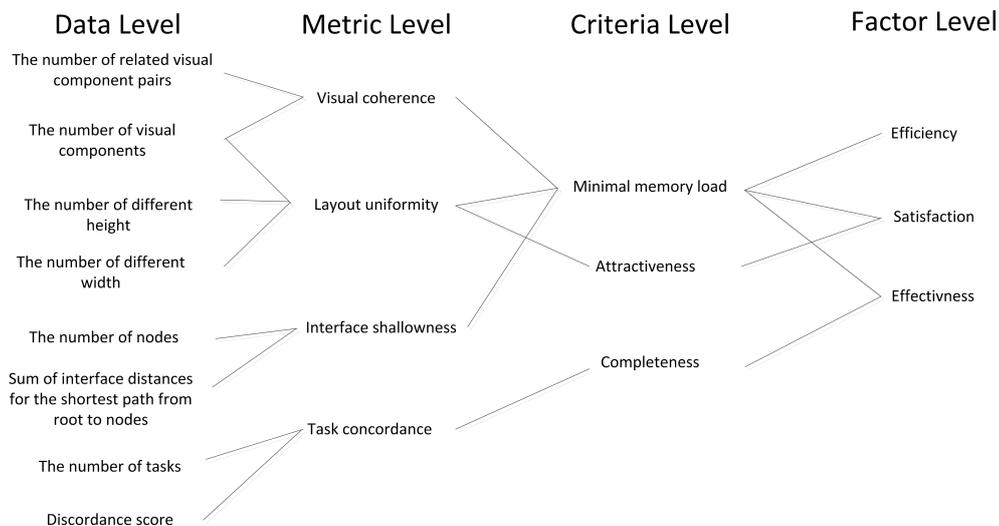


Abbildung 2.2: QUIM: Beispiel für den Zusammenhang der Komponenten (Seffah u. a., 2001)

Die Daten werden aus unterschiedlichsten Quellen gewonnen. Diese sind zum Beispiel die Nutzer selber, Usability Experten, Fragebögen, das System, die Dokumentation des Systems oder das Modell des Systems.

2.1.6 Zusammenhang der Ebenen

Die Struktur von QUIM kann nicht als Baum dargestellt werden. Eine Metrik kann auf mehr als ein Kriterium einwirken und damit mit mehreren Kriterien in Verbindung stehen. Dies gilt für alle Level. Grafik 2.2 zeigt ein Beispiel, wie die Level zusammenhängen können.

„Number of components“ zum Beispiel ist eine Eingabe für „Visual Coherence“ und „Layout Uniformity“. Diese Metriken messen das Kriterium „Minimal memory load“, welches wiederum die „Efficiency“, die „Satisfaction“ und die „Effectivness“ beeinflusst.

2.2 SUM

Das Paper „*A Method to Standardize Usability Metrics Into a Single Score*“ von Sauro und Kindlund (2005) beschreibt eine Methode, um mit Hilfe von Usability Metriken einen standardisierten Wert zu erstellen, der die Usability eines Systems anzeigt. Dafür wird die ISO-Norm in eine einzige, standardisierte und summierte Usability Metrik (SUM) vereinfacht. Daher ist diese Analyse als empirische Evaluation zu sehen und eignet sich für Regressionsanalysen, Hypothesen Tests und allgemein zur Unterstützung in Usability Reports.

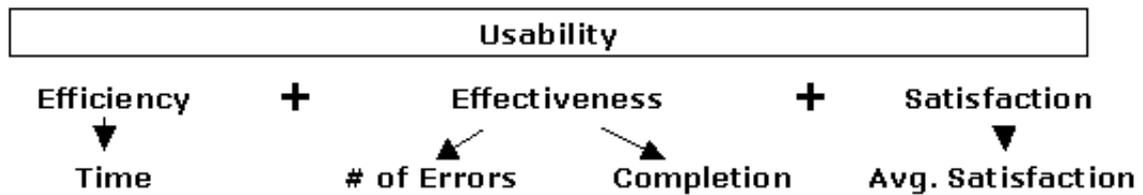


Abbildung 2.3: SUM: Quantitatives Modell von Usability (Sauro und Kindlund, 2005)

Ziel dieser Methode ist es, Usability Metriken einfacher darzustellen. Viele Metriken werden schnell unübersichtlich und sind schwer zu verstehen für Entscheider. Zusätzlich muss jede Metrik einzeln gemessen werden und einzeln in Analysen oder Berichte eingebracht werden.

2.2.1 Quantitatives Modell

Die Abbildung 2.3 zeigt das Quantitative Modell von Usability, welches SUM zugrunde liegt. Dabei handelt es sich um drei Dimensionen (Effizienz, Effektivität und Zufriedenheit), die aus der ISO Norm bekannt sind. Diese werden durch vier Metriken beeinflusst. Die Effizienz wird durch die Aufgabenzeit definiert, die Effektivität durch die Fehleranzahl und die Aufgabenvollständigkeit und die Zufriedenheit wird durch die durchschnittliche Zufriedenheit des Nutzers erfasst. Diese wird mit Hilfe von standardisierten Fragebögen erfasst und nach der jeweiligen Aufgabe oder nach dem kompletten Test ausgefüllt.

2.2.2 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse ist eine statistische Technik zum Vereinfachen von großen Datenmengen. Dabei werden viele statistische Variablen, durch wenige, aussagekräftige Linearkombinationen genähert. Diese Linearkombinationen werden dann als Hauptkomponenten bezeichnet. Das Ziel der Analyse ist es, die Dimension der Originaldaten zu reduzieren. Als Ergebnis erhält man vier Hauptkomponenten (Zeit, Fehlerrate, Nutzerzufriedenheit und Vollständigkeit). Aus diesen wird dann ein Endwert, der Single Score, ermittelt.

Daraus ergibt sich ein verbessertes Modell für Usability, da die Fehlerrate der Einzelmessungen minimiert wird. Zusätzlich werden redundante Daten entfernt, da zum Beispiel Fehler die Fehlerrate und die Aufgabenzeit beeinflussen können. Darüber hinaus ist es möglich zu erkennen, wenn eine der vier Variablen das Modell mehr beeinflusst, als andere.

2.2.3 Standardisierung

Damit die Datenmengen komprimiert werden können, müssen sie standardisiert werden. Die Standardisierung bewirkt, dass die Daten das selbe arithmetische Mittel und die gleiche Standardabweichung bekommen und so zusammengefasst werden können. Allgemein werden diskrete bzw. kontinuierliche Daten wie folgt standardisiert:

- **Für diskrete Daten (Zeit und Nutzerzufriedenheit):**

$$D = \frac{\text{Defekte}}{\text{Mögliche Defekte}} \quad (2.5)$$

- **Für kontinuierliche Daten (Fehlerrate und Vollständigkeit):**

$$K = \frac{\text{Spezifiziertes Maximum} - \text{Arithmetisches Mittel}}{\text{Standardabweichung}} \quad (2.6)$$

Daraus folgend gelten für die konkreten Metriken die unten aufgeführten Formeln.

- **Vollständigkeit (Task Completion):**

$$V = \frac{\text{Anteil fehlerhafter Aufgaben}}{\text{Begonnene Aufgaben}} \quad (2.7)$$

- **Fehlerrate (Number of errors):** Anders als bei den Fehlerhaften Aufgaben kann der Nutzer in diesem Fall mehr als einen Fehler pro Aufgabe machen. Auf Grund dessen müssen die Möglichkeiten für Aufgabenfehler im Voraus definiert werden. Dies muss nicht in jedem Fall möglich sein und ist daher ein möglicher Schwachpunkt. Komplexe Aufgaben sind hier natürlich anfälliger, als einfache Aufgaben.

$$F = \frac{\text{Anzahl Fehler}}{\text{Anzahl mögliche Fehler}} \quad (2.8)$$

- **Nutzerzufriedenheit (Avg. Satisfaction):** Zur Erfassung der Nutzerzufriedenheit gibt es nach jeder Aufgabe einen Fragebogen. Zur Bewertung dieser Fragen haben sich zwei Skalen etabliert. Die 1-5 Skala (5: sehr einfach bis 1: sehr schwer) und die 1-7 Skala (7: sehr einfach bis 1: sehr schwer). Wobei ein System mit „Guter Usability“ einen Wert von 4 (1-5) bzw. 5,6 (1-7) aufweist. Für die 1-5 Skala sieht die Berechnungsformel daher wie folgt aus:

$$N = \frac{4 - \text{Durchschn. Nutzerzufriedenheit}}{\text{Standardabweichung}} \quad (2.9)$$

Für die 1-7 Skala würde sich der Wert des spezifizierten Maximum (hier 4) auf 5,6 ändern.

- **Aufgabenzeit (Time):** Bei der Aufgabenzeit ist die Herausforderung das Finden der idealen Aufgabenzeit. Die Frage ist: Wann braucht eine Testperson zu lange? Wenn man diese allerdings definiert hat, kann die Aufgabenzeit wie folgt errechnet werden:

$$A = \frac{\text{Ideale Aufgabenzeit} - \text{Benötigte Aufgabenzeit}}{\text{Standardabweichung}} \quad (2.10)$$

2.2.4 Erstellung der SUM

Der letzte Schritt besteht darin, die eigentliche Kennzahl zu erstellen. Dafür wird aus den vier standardisierten Variablen für jede Aufgabe der Durchschnitt gebildet. Als Ergebnis erhält man eine Kennzahl, die aussagt wie gut benutzbar das getestete System ist. Das kann natürlich nicht eine qualitative Usability Studie ersetzen, allerdings erhält man (ähnlich einem IQ-Test) Informationen über ein hochkompliziertes System in Form von einer Zahl und es ist möglich daraufhin Entscheidungen zu treffen. Als Beispiel ist die Qualitätsentwicklung im Software-Entwicklungsprozess zu nennen. Dafür wird vor und nach einer Änderung gemessen. Aufgrund der Änderung der Zahl ist es möglich zu erkennen, ob die Änderung das System zum Positiven entwickelt hat.

2.3 UTUM

Das UIQ Technology Usability Metrics (UTUM) Testpackage wird in [Winter u. a. \(2007\)](#) vorgestellt und ist ein Paket von Usability Metriken für mobile Geräte. Es basiert auf empirischen Analysen, welche Metriken für die Nutzerzufriedenheit, die Effizienz und die Effektivität beinhalten. Diese werden durch Auffälligkeiten des Testleiters ergänzt.

2.3.1 Testprozess

Zu Beginn des Testvorgangs erhalten die Testpersonen einen Fragebogen zu ihrem Nutzungsverhalten von mobilen Geräten und zu ihren persönlichen Daten. Darauf folgt eine Einarbeitungsphase in die mobilen Geräte. Nach Abschluss dieser Einarbeitungsphase füllt die Testperson die „Hardware Evaluation“ aus. Dies ist wiederum ein Fragebogen, bei dem es allerdings um das Look and Feel des speziellen Gerätes geht.

Danach folgt die eigentliche Testphase. Die Testperson beginnt, die im Vorfeld erarbeiteten Use Cases, mit dem mobilen Gerät auszuführen. Während dessen notiert sich der Testleiter Auffälligkeiten und stoppt die Zeit für jede Aufgabe. Im Anschluss an die einzelnen Use Cases, füllt die Testperson einen „Task Effectiveness“ Fragebogen aus. Dieser enthält Fragen zur Meinung des Nutzers, wie gut das mobile Gerät die Ausführung Use Cases unterstützt. Hat die Testperson alle Use Cases ausgeführt, bekommt sie den SUS Fragebogen. Dieser ergibt eine subjektive Bewertung der Usability des Systems. Dieser Fragebogen ist eine „quick and dirty“ Lösung, die auf der ISO 9241-11 basiert und als Ergebnis eine Kennzahl liefert, welche die Usability des kompletten Systems anzeigt.

2.3.2 Metriken

Im Folgenden werden die Metriken beschrieben, die im UTUM Testpackage betrachtet werden.

Performance-Effizienz Metrik

Die Performanz-Effizienz Metrik wird aus dem Durchschnitt der relativen und speziellen Effizienz gebildet. Als Datengrundlage für die Relative Effizienz dient allein die Zeit. Für die spezielle Effizienz werden zusätzlich die Auffälligkeiten des Testleiters betrachtet. Diese Metrik gibt an, ob das getestete mobile Gerät effizient für die gestellte Aufgabe ist.

Nutzerzufriedenheit Metrik

Als Datengrundlage für die Nutzerzufriedenheit wird der SUS Fragebogen benutzt. Damit wird darauf eingegangen, ob das mobile Gerät leicht zu benutzen ist.

Aufgaben-Effektivität Metrik

Diese Metrik evaluiert die Aufgabeneffektivität. Dabei wird gezeigt, in wie fern es möglich ist, mit dem mobilen Gerät, die gestellte Aufgabe zu lösen. Also wie gut der User beim lösen seines Use Cases unterstützt wird.

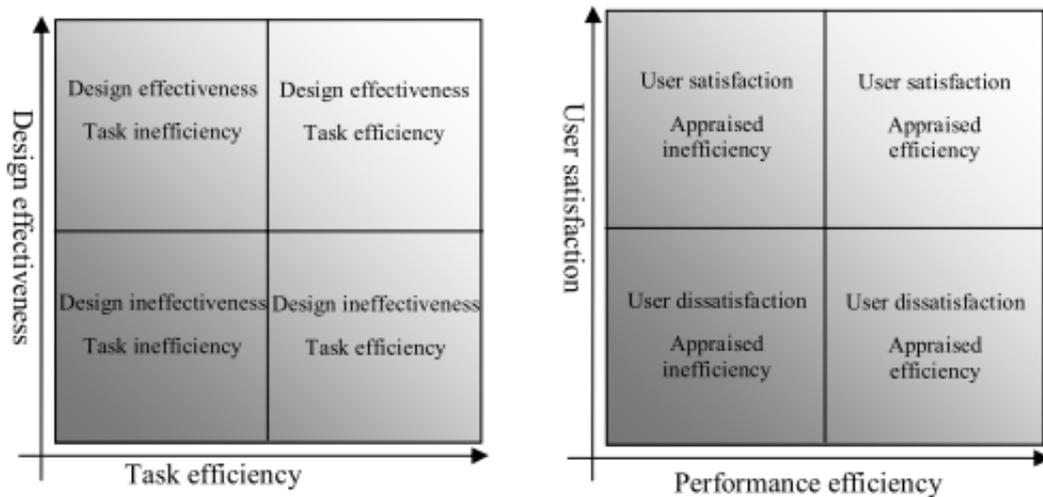


Abbildung 2.4: UTUM: Effektivität-Effizienz Metrik (l) und Zufriedenheit-Effizienz Metrik (r) (Winter u. a., 2007)

Total Usability Metric

Die „Total Usability Metric“ stellt eine Kombination aus Nutzerzufriedenheit, Performance-Effizienz und Aufgaben-Effektivität dar. Dies ist möglich, da die drei Metriken den selben Datenbereich haben, allerdings ist der Unterschied in der Natur der Daten zu groß. Dies bewirkt daher einen großen Datenverlust und ist daher nicht praktikabel.

Um diesen Verlust zu verringern werden jeweils nur zwei Metriken verglichen. Die Abbildung 2.4 zeigt die Effektivität-Effizienz- und die Zufriedenheit-Effizienz Metrik.

2.4 Abgrenzung zur eigenen Arbeit

Im Gegensatz zu den vorgestellten Arbeiten sollen hier die Metriken nicht das einzige Mittel zur Bewertung der Usability eines Systems sein. Sie dienen der Unterstützung von Thesen. Das heißt, dass es im End keine Kennzahl geben wird, die die Usability des Gesamtsystems darstellt. Dies wird nur für Teilsysteme oder einzelne Use Cases durchgeführt. Darüber hinaus wird es aber möglich sein, empirische Studien umzusetzen.

Die Erfassung der Rohdaten und die Auswertung dieser, wird so weit es möglich ist, vom System automatisiert werden. Dabei werden aber die verwendeten Metriken generalisierbar bleiben. Dieser Punkt wurde in keinem der vorgestellten Frameworks beschrieben.

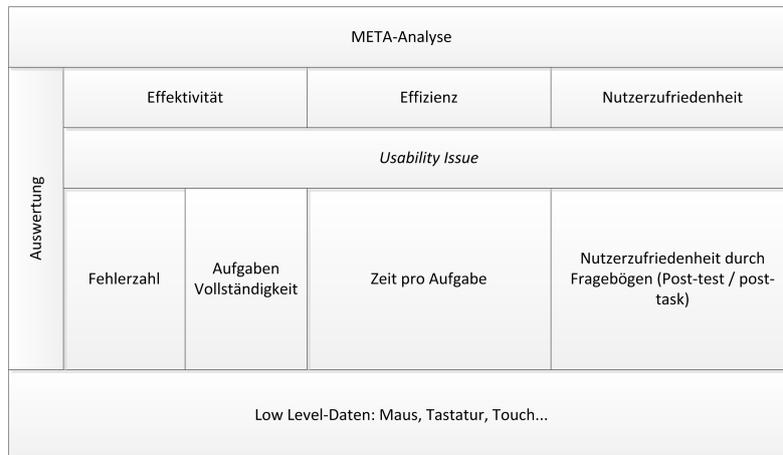


Abbildung 2.5: Usability Metrics Modell

Das Modell der Arbeit ist grundlegend ähnlich wie das quantitative Modell von SUM (siehe Abbildung 2.3). Dies wurde erweitert und den eigenen Ansprüchen angepasst (siehe Abbildung 2.5). Als Grundlage werden die Low-Level Daten erfasst. Dies können zum Beispiel Maus-, Tastatur- oder Touchdaten sein. Zu diesen Daten kommen die drei Metriken Fehlerzahl, Aufgabenzeit und Nutzerzufriedenheit. Im Modell werden diese dann durch die Usability Issues, im Grunde die Kompetenz von Usability Experten, ergänzt und bilden so die drei Dimensionen der Usability: Effizienz, Effektivität und Nutzerzufriedenheit. Danach ist es möglich Meta-Analysen durchzuführen.

3 Schluss

Im letzten Kapitel wird diese Ausarbeitung noch einmal zusammengefasst und ein Ausblick des weiteren Vorgehen gegeben.

3.1 Zusammenfassung

Zu dem Thema Usability Metrics wurden drei Metriken-Frameworks vorgestellt. Diese stellen jeweils einen empirischen Ansatz für Usability dar.

Mit QUIM wurde ein hierarchisches Modell, welches ca. 100 Metriken enthält, die teils berechnet und teils nur summiert werden, vorgestellt. Diese Metriken bestimmen Kriterien und Faktoren, welche wiederum die Nutzungsqualität beeinflussen. Diese nimmt der Benutzer des Systems wahr.

SUM hingegen ist ein Ansatz der auf Statistik basiert. In seinem quantitativen Modell wird gezeigt, dass sich Usability aus den drei Dimensionen Effektivität, Effizienz und Nutzerzufriedenheit zusammensetzt. Diese werden aus den vier Metriken Aufgabenzeit, Fehlerzahl, Aufgabenvollständigkeit und die durchschnittliche Nutzerzufriedenheit gebildet. Aus diesen Metriken wird eine Kennzahl berechnet, die die Usability des Gesamtsystems angibt.

Das letzte Framework, das UTUM Testpackage, ist speziell auf die Usability von mobilen Geräten ausgerichtet. Es beschreibt eine empirische Analyse auf der Basis von Metriken für die Nutzerzufriedenheit, Effizienz und Effektivität. Dies wird ergänzt um Auffälligkeiten des Testleiters. Das Vorgehen wird durch einige genormte Fragebögen unterstützt.

3.2 Ausblick

Wie bereits in Kapitel 1 beschrieben wird im Modul Projekt 1 ein System zur Erfassung der Rohdaten für Metriken entwickelt. Im zweiten Projektmodul wird voraussichtlich mit der Entwicklung einer Auswertungsumgebung für diese Rohdaten fortgefahren.

Darüber hinaus ist es möglich in unterschiedliche Richtungen weiterzuentwickeln. Es wurde in den Projekten eine Grundlage für komplette empirische Studien gelegt. Hier muss weiterentwickelt und getestet werden.

Des Weiteren ist es möglich den Erfassungsbereich zu erweitern und damit die Verwendung der Rohdaten von Eye-Trackern, Body-Monitoring-Systemen, Elektroenzephalogrammen oder ähnlichem implementiert werden. Dies kann dann ebenfalls im Auswertungsbereich getan werden.

Erweiterungsmöglichkeiten bestehen ebenso in den Bereichen Touchbedienung oder mobile Geräte.

Literaturverzeichnis

- [Feldman u. a. 2009] FELDMAN, Liam ; MUELLER, Carl J. ; TAMIR, Dan ; KOMOGORTSEV, Oleg V.: Usability testing with total-effort metrics. In: *ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. Washington, DC, USA : IEEE Computer Society, 2009, S. 426–429. – Abruf: 2010-04-12. – ISBN 978-1-4244-4842-5
- [Hornbæk und Law 2007] HORNBÆK, Kasper ; LAW, Effie Lai-Chong: Meta-analysis of correlations among usability measures. In: *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2007, S. 617–626. – Abruf: 2010-05-26. – ISBN 978-1-59593-593-9
- [McCabe 1976] MCCABE, Thomas J.: A complexity measure. In: *ICSE '76: Proceedings of the 2nd international conference on Software engineering*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1976, S. 407
- [Nielsen 2001] NIELSEN, Jakob: *Usability Metrics*. 2001. – URL <http://www.useit.com/alertbox/20010121.html>. – Abruf: 2010-04-12
- [Sauro und Kindlund 2005] SAURO, Jeff ; KINDLUND, Erika: A method to standardize usability metrics into a single score. In: *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2005, S. 401–409. – Abruf: 2010-05-26. – ISBN 1-58113-998-5
- [Sauro und Lewis 2009] SAURO, Jeff ; LEWIS, James R.: Correlations among prototypical usability metrics: evidence for the construct of usability. In: *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*. New York, NY, USA : ACM, 2009, S. 1609–1618. – Abruf: 2010-04-12. – ISBN 978-1-60558-246-7
- [Seffah u. a. 2001] SEFFAH, A. ; KECECI, N. ; DONYAE, M.: QUIM: A Framework for Quantifying Usability Metrics in Software Quality Models. In: *APAQS '01: Proceedings of the Second Asia-Pacific Conference on Quality Software*. Washington, DC, USA : IEEE Computer Society, 2001, S. 311. – Abruf: 2010-04-12. – ISBN 0-7695-1287-9
- [Winter u. a. 2007] WINTER, Jeff ; RONKKO, Kari ; AHLBERG, Marten ; HINELY, Mark ; HELLMAN, Mats: Developing Quality through Measuring Usability—The UTUM Test Package. In: *WoSQ '07: Proceedings of the 5th International Workshop on Software Quality*.

Washington, DC, USA : IEEE Computer Society, 2007, S. 2. – Abruf: 2010-04-12. – ISBN 0-7695-2959-3

[Winter u. a. 2009] WINTER, Jeff ; RONKKO, Kari ; HELLMAN, Mats: Reporting usability metrics experiences. In: *CHASE '09: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. Washington, DC, USA : IEEE Computer Society, 2009, S. 108–115. – Abruf: 2010-04-12. – ISBN 978-1-4244-3712-2

[Zeman u. a. 2009] ZEMAN, Jaroslav ; TANUSKA, Pavol ; KEBISEK, Michal: The Utilization of Metrics Usability to Evaluate the Software Quality. In: *ICCTD '09: Proceedings of the 2009 International Conference on Computer Technology and Development*. Washington, DC, USA : IEEE Computer Society, 2009, S. 243–246. – Abruf: 2010-04-12. – ISBN 978-0-7695-3892-1