



universität
wien

MASTERARBEIT

Titel der Masterarbeit

„Softwarelokalisierung von mobilen Apps“

Verfasser

David Rathbauer, BA

angestrebter akademischer Grad

Master of Arts (MA)

Wien, im August 2014

Studienkennzahl lt. Studienblatt:

A 060 342 351

Studienrichtung lt. Studienblatt:

Übersetzen Englisch Spanisch

Betreut von:

Univ-Prof. Mag. Dr. Gerhard Budin

Inhaltsverzeichnis

1	Einleitung.....	4
2	Der Computer	6
3	Lokalisierung.....	9
3.1	Dimensionen.....	9
3.1.1	GILT.....	9
3.1.2	Prozesse und Beteiligte	11
3.1.3	Produktkomponenten.....	12
3.1.4	Abstufungen der Adaptierung	13
3.2	Internationalisierung.....	18
3.2.1.	Herausbildung von Internationalisierung	19
3.2.2.	Internationalisierung von Software	20
3.2.3.	Sprachliche Vorbereitung auf Lokalisierung.....	21
3.2.4.	Probleme und Lösungsansätze	23
4	Die wirtschaftliche Dimension	25
4.1	Entwicklung der Lokalisierungsindustrie	25
4.2	Warum lokalisieren?	31
4.3	Kostenmanagement	34
4.3.1	Kostenaufstellung.....	35
4.3.2	Übersetzungskosten.....	36
4.3.3	Effizienzsteigerung.....	38
4.4	Alternative Ansätze	40
4.4.1	Freie Software	40
4.4.2	Crowdsourcing	41
5	Das Lokalisierungsprojekt.....	44
5.1	Projektmanagement.....	44
5.1.1	Kompetenzen und Aufgaben	44
5.1.2	Zusammensetzung der Projektbeteiligten.....	48
5.1.3	Projektablauf	50

5.1.4	Lokalisierungskit	51
5.2	Terminologie, Terminologiemanagement	52
5.2.1	Probleme, Ursachen und Folgen	52
5.2.2	Return on Investment und Terminologie.....	55
5.2.3	Terminologiemanagement	58
5.3	Übersetzung.....	64
5.3.1	Software.....	65
5.3.2	Hilfedatei	70
5.3.3	Dokumentation	71
5.3.4	Stil und Locales	72
5.3.5	Videospiele und Übersetzung oder <i>All your base are belong to us</i>	73
5.4	Qualitätsmanagement	75
5.4.1	Softwarequalität.....	76
5.4.2	Sprache	78
6	Technologie	89
6.1	Translation Memory-Tools	90
6.2	Terminologie-Tools.....	92
6.3	Lokalisierungstools	93
6.4	Sonstige Tools und Features	94
6.5	Maschinelle Übersetzung	95
6.5.1	Regelbasierte und statistische maschinelle Übersetzungssysteme	96
6.5.2	Einsatz von maschineller Übersetzung in der Praxis	97
6.6	Entwicklung	98
6.7	Standards	100
7	Mobile Apps	102
7.1	Smartphone und Tablet	102
7.1.1	Entstehung.....	102
7.1.2	iOS und Android	103
7.2	Mobile Apps.....	105
7.2.1	Grundlegendes.....	105

7.2.2	Wirtschaftliche Relevanz.....	108
7.2.3	Vertriebsmodell	109
7.2.4	App-Programmierung und Internationalisierung	110
7.3	Lokalisierung von Apps	112
7.3.1	Warum Apps lokalisieren?	112
7.3.2	Lokalisierung für Übersetzer/innen.....	113
7.3.3	App-Lokalisierung für Entwickler/innen	114
8	Fazit	116
9	Quellen.....	118
9.1	Bibliografie.....	118
9.2	Onlinequellen	125
	Kurzfassung.....	133
	Abstract	134
	Lebenslauf	135

1 Einleitung

Die Geschwindigkeit des technischen Fortschritts in unserer Zeit ist wahrhaft erstaunlich. Ein gutes Beispiel dafür ist das Phänomen App. Innerhalb weniger Jahre ist aus dieser Neuheit etwas Allgegenwärtiges geworden, das jeder Mensch kennt und meint, es zu brauchen. In der Tat haben Smartphones mit ihren kleinen Anwendungen den Alltag von vielen verändert; Unterhaltung, Dienstleistungen und vor allem Kommunikationsmöglichkeiten werden in einem Gerät vereint, das in die Hosentasche passt.

Bei der Verbreitung des Smartphones spielt Lokalisierung eine bedeutende Rolle. Ohne Softwarelokalisierung wäre es nicht möglich, die große Anzahl an Menschen zu erreichen, die Smartphones nutzen und Apps herunterladen. Deshalb wird hier die Frage gestellt: Wie können mobile Apps lokalisiert werden?

Um dieser Frage beantworten zu können, wird sie sozusagen bei der Wurzel gepackt. Es wird zuerst versucht, ein ganzheitliches Bild von Lokalisierung im Hier und Jetzt zu bilden. Welche Konzepte liegen diesem Phänomen zugrunde? Welche Aspekte haben sich herausgebildet und gelten jetzt als relevant? Einfacher ausgedrückt: Wie funktioniert Softwarelokalisierung im Jahr 2014?

Diese Ausführungen bilden schließlich die Grundlage für die ursprüngliche Frage. Es wird herausgearbeitet, inwiefern sich mobile Apps von PC-Software unterscheidet und welche Auswirkungen dies auf ihre Lokalisierung hat. Für die Informationsbeschaffung wurde das Internet als Informationsquelle herangezogen. Dieser Ansatz lässt sich dadurch begründen, dass über dieses Medium alles, was Apps betrifft, abgewickelt wird: Tools für die Entwicklung von Apps werden im Internet zur Verfügung gestellt, Informationen über die Entwicklung bereitgestellt und auch der Vertrieb erfolgt ausschließlich online. Es handelt sich also um ein Phänomen, das mit dem World Wide Web verbunden ist.

Der erste Teil dieser Arbeit beschreibt Softwarelokalisierung aus verschiedenen Perspektiven und versucht, den heutigen Stand wiederzugeben. Dazu ist zu sagen, dass es sich um ein Feld handelt, das sich sehr schnell entwickelt. Vieles von dem, was hier beschrieben wird, wird innerhalb kurzer Zeit nicht mehr aktuell sein. Dies macht Softwarelokalisierung jedoch zu einem spannenden und dynamischen Bereich und es bleibt abzuwarten, was die Zukunft noch bringt.

Der Ausgangspunkt ist die Entstehung und Entwicklung des Computers. Ohne dieses Medium gäbe es keine Lokalisierung. Es wird geschildert, wie sich diese eigentlich junge Technologie in den vergangenen 60 bis 70 Jahren zu dem entwickelt hat, was sie heute ist und wie aus riesigen Rechenmaschinen hosentaschengroße, multifunktionale Geräte geworden sind.

Das dritte Kapitel widmet sich Softwarelokalisierung als Phänomen. Dieses kann auf verschiedenen Ebenen betrachtet werden: Von der Einbindung der Industrie im Wirtschaftsgefüge über Lokalisierung auf Unternehmensebene bis hin zur Tätigkeit des/der Einzelnen. Es wird schließlich besprochen, was Lokalisierung für ein Produkt bedeutet und wie tiefgehend die Anpassungen an verschiedene Märkte und Kulturen sein können. Internationalisierung geht Hand in Hand mit Lokalisierung und wird deshalb hier auch ausführlich besprochen.

Kapitel 4 beschreibt die wirtschaftliche Seite. Den Ausgangspunkt bildet die Entstehung und Entwicklung der Lokalisierungsindustrie. Im Weiteren wird auf die Überlegungen der Entscheidungsträger/innen eingegangen: Warum sollte man Produkte lokalisieren und welche Investitionen sind damit verbunden. Schließlich werden auch Ansätze abseits der Mainstream-Softwareindustrie besprochen, um aufzuzeigen, dass es auch bei Softwareentwicklung verschiedene ideologische Hintergründe gibt.

Das folgende Kapitel stürzt sich tiefer in die Materie. Im Fokus liegen hier die wichtigsten Aspekte des Lokalisierungsprojekts: Projektmanagement, Terminologie, Übersetzung und Qualitätsmanagement. Projektmanagement bildet den Rahmen eines erfolgreichen Lokalisierungsprojekts. Da viele verschiedene Beteiligte ihre Arbeitsergebnisse beitragen, kann hier nichts dem Zufall überlassen werden. Es wird besprochen, welche Herausforderungen sich im Zuge eines Lokalisierungsprojekts ergeben. Terminologiemanagement ist, wie in anderen Bereichen der Sprachindustrie, ein elementarer Bestandteil auch bei Softwarelokalisierung. Hier wird auf Terminologiarbeit und die Implementierung von Terminologiemanagement innerhalb eines Unternehmens eingegangen. Übersetzung wird manchmal als Kern von Softwarelokalisierung bezeichnet. Dieses Teilkapitel widmet sich den spezifischen Herausforderungen, denen Übersetzer/innen im Rahmen eines Projekts begegnen. Qualitätsmanagement ist der letzte Aspekt, der einzeln behandelt wird. Es wird besprochen, was Qualität ist, wie sie überprüft werden kann und was das für die Praxis bedeutet.

Was wäre Softwarelokalisierung ohne Programme und Tools? Kapitel 6 widmet sich den technischen Hilfsmitteln, die Übersetzer/innen und andere Beteiligte einsetzen. Es werden die verschiedenen Programme vorgestellt und schließlich die Entwicklung in diesem Bereich skizziert.

Das Kapitel Apps kehrt zur Ausgangsfrage zurück. Den Ausgangspunkt bildet hier das Medium, auf dem Apps eingesetzt werden, das Smartphone. Weiters werden die Merkmale von mobilen Apps selbst analysiert, welche Besonderheiten es bei der Programmierung gibt und wie sie internationalisiert werden. Der letzte Punkt ist schließlich Softwarelokalisierung von Apps, der sich mit der ursprünglichen Frage beschäftigt. Es wird aufgezeigt, wie Apps lokalisiert werden können und, davon ausgehend, welche Unterschiede es zur Lokalisierung von PC-Software gibt.

2 Der Computer

Das zentrale Medium der Softwarelokalisierung ist der Computer in allen seinen Formen und Ausführungen. Neben den verschiedenen modernen Transportmitteln ist es jene Erfindung, die unser tägliches Leben, zumindest in den sogenannten Industrieländern, am meisten verändert hat. Die Omnipräsenz und Vielfalt der Geräte fallen den meisten Menschen wahrscheinlich gar nicht mehr auf. Vom Arbeitsplatz hat es der Computer als PC in die Privathaushalte geschafft und als Handheld hinaus auf die Straße und praktisch überallhin. Genauso wenig wie die Pionier/innen des Autos, eine andere große Erfindung die unsere Lebensweise revolutioniert hat, ahnen konnten, was sich im 21. Jahrhundert auf den Straßen fortbewegt, hat der Weg des Computers mit dem Smartphone begonnen. Im Folgenden soll ein kurzer Überblick über die Entstehung des Computers, so wie wir ihn kennen, gegeben werden, da damit natürlich auch die Entstehung und Entwicklung des Konzeptes Lokalisierung eng zusammenhängen.

Am Anfang stand das Ziel, Funktionen auszurechnen. Charles Babbage, Erfinder und Mathematiker aus England, entwarf in den 1830ern die „Analytische Maschine“, ein mechanischer Vorläufer des Computers, der mit einer Art Lochkarten funktionieren sollte. Dieses Gerät wurde erst Anfang des 20. Jahrhunderts durch den Sohn des Erfinders gebaut (vgl. Wurster 2002:8-15).

Der Deutsche Konrad Zuse, ein Bauingenieur aus Berlin, hatte ebenfalls das Ziel, mühsame, komplexe Rechengänge durch eine Maschine vornehmen zu lassen. Er stellte das erste Gerät, das die Bezeichnung Computer verdient, im Jahr 1941 vor. Seine programmierbare Maschine „Z 3“, deren Kern aus Telefonrelais besteht, wurde somit drei Jahre vor dem „Harvard Mark I“, der ersten Rechenmaschine eines amerikanischen Forschungsteams, fertiggestellt. Diese war 17 Meter lang und fünf Tonnen schwer; Ausmaße, die heute absurd erscheinen, in den ersten Jahrzehnten der Computertechnik jedoch völlig normal waren. In Folge investierte vor allem die US-amerikanische Regierung in Forschungsprogramme zur Weiterentwicklung der Großrechner, die zu Beginn vor allem für militärische Zwecke eingesetzt wurden (vgl. Wurster 2002: 8-25).

Die erste Norm wurde 1963 eingeführt. ASCII (American Standard Code for Information Interchange) sollte Kommunikation zwischen verschiedenen Computersystemen, deren es bereits mehrere gab, ermöglichen. In den Sechzigern und Siebzigern entwickelte sich der Rechner zu einem Gerät, das in den verschiedensten Gebieten eingesetzt wurde; so in der Raumfahrt, auf Flughäfen, in Atomkraftwerken, im Bankwesen und im Rechnungswesen (vgl. Wurster 2002: 48-55). Als sehr teure Anschaffung unterscheidet sich der Einsatz des Computers natürlich grundlegend vom Einsatzgebiet der heute weit verbreiteten PCs. Für dieselben Zwecke wie damals werden noch immer große Systeme, sogenannte Supercomputer und Großrechner, eingesetzt (vgl. Byte Notes 2014).

Für Unternehmen und Banken kam auch in den sechziger Jahren der Minicomputer auf, der allerdings nur im Vergleich mit dem Großrechner „mini“ war. Trotzdem waren diese Geräte für Unternehmen und Organisationen leistbar. Interagiert wurde noch mit Lochkarten über eine Bedienkonsole, wobei es für einen Rechner meistens mehrere Konsolen gab, weshalb sich sogar mehrere Organisationen einen Computer teilen konnten (vgl. Wurster 2002: 106-113).

In den 1970ern begann die Entwicklung in Richtung des Computers, den wir heute kennen, mit der Erfindung des Mikroprozessors durch die Firma Intel. Dadurch war es möglich, kleine Systeme zu konstruieren. Paul Allen und Bill Gates erfanden die bereits vorhandene Programmiersprache BASIC quasi neu. Es gab nun die Möglichkeit für Privatpersonen, Computer zu kaufen und damit zu experimentieren sowie viele neue Einsatzgebiete, wie zum Beispiel Spielhallengeräte (vgl. Wurster 2002: 132-137). Alle Geräte, die heute privat und in den meisten Büros benutzt werden, fallen im Prinzip in diese Kategorie der Mikrocomputer (vgl. Byte Notes 2014).

Ende der siebziger Jahre begann das Zeitalter des Personal Computers mit dem „Apple II“ von Steve Jobs und Steven Wozniak. Die Benutzerschnittstelle wurde mit Tastatur und Bildschirm revolutioniert und die ersten Schreib- und Tabellenkalkulationsprogramme entstanden und somit wurde die Softwareentwicklung zu einem eigenen, immer bedeutenderen Fach. Die Achtziger brachten Betriebssysteme wie MS-DOS von Microsoft und die Ausbreitung des PCs in der Gesellschaft (vgl. Wurster 2002: 140-146). Die Weiterentwicklung ging nun immer schneller vonstatten, mit grundlegenden Dingen wie dem Desktop, Grafikprogrammen und der Maus, die zwar schon früher erfunden wurden, aber erst in den achtziger Jahren wirklich integriert wurden. Durch die grafische Benutzeroberfläche, die in ihrer Struktur tatsächlich einen Schreibtisch nachahmt, wurde es viel einfacher, einen Computer als Laie zu benutzen. Der Aufschwung führte zu einem Konkurrenzkampf zwischen Apple und Microsoft und spätestens ab Beginn der Neunziger zur vollständigen Integration des Computers in den Arbeitsplatz und schließlich in den Privathaushalt (vgl. Wurster 2002: 224-235).

In den sechziger Jahren begann die vom US-amerikanischen Militär gegründete „Advanced Research Project Agency“ mit der Konzeption und Entwicklung eines Netzwerkes, dem ARPAnet. Dieses wurde 1972 erstmals vorgestellt, stieß aber nicht auf großes Interesse. Ein Jahr später wurde das Ethernet, das lokale Netzwerk erfunden. Tim Berners-Lee sorgte schließlich 1989 für die Verknüpfung aller lokalen Netzwerke untereinander; es entstand das World Wide Web oder Internet. Er erfand die Seitenbeschreibungssprache HTML, auf der Grundlage des Hypertext-Konzeptes. Dieses steht im Gegensatz zum sequenziellen Schreiben, nämlich der linearen Produktion und Rezeption von Texten. Ted Nelson stellte fest, dass Gedanken nicht sequenziell aneinandergereiht sind, sondern vielfach miteinander verknüpft sind. Mit dieser Grundlage erdachte er Xanadu, ein Netzwerk, das im Prinzip mit dem heutigen Internet realisiert worden ist (vgl. Wurster 2002: 256-257).

Das Internet selbst wurde erst durch einige Standards möglich gemacht. Der Adressstandard „Universal Ressource Locator“ (URL), die Seitenbeschreibungssprache HTML und das Übertragungsprotokoll „HyperText Transfer Protocol“ (HTTP) machen die Kommunikation zwischen den Computersystemen möglich und bilden so die technische Grundlage des Internets. Den Durchbruch schafft das Internet 1995 durch die Erfindung des grafischen Webbrowsers und später durch die Programmiersprache Java (vgl. Wurster 2002: 268-269). Mittlerweile sind große Teile der Welt miteinander vernetzt. Zwischen 2000 und 2012 wurde ein Anstieg von knapp 361 Millionen auf über 2,4 Milliarden User verzeichnet (vgl. Internet World Stats 2014). Der Zugang zum Internet ist natürlich nicht gleichmäßig verteilt. Die Downloadgeschwindigkeit hängt vom Ausbau sowie von der Technik des Netzes ab. Die höchsten Geschwindigkeiten erreichen Südkorea und Japan, während es in einigen afrikanischen Ländern fast noch gar kein Internet gibt (vgl. Netindex 2014). Das Internet ist, zumindest in den Industrienationen, allgegenwärtig und in den Alltag soweit eingebunden, dass es an eine Katastrophe grenzt, wenn kein schneller Internetanschluss verfügbar ist. Wie alle weltweiten Netzwerke ist es natürlich nicht vor Problemen gefeit, das heißt, dass Leitungen, vor allem jene, die durch Konfliktzonen gehen, beschädigt werden können. Dies kann zu Ausfällen in großen Gebieten führen, wobei Entwicklungsländer auch hier eher betroffen sind als Industrienationen (vgl. Cowie 2014).

Ein Blick in die Wohnzimmer Europas zeigt, dass der Stand-PC kein Standard mehr ist. Laptops sind leistungsfähig geworden und so gibt es mittlerweile tragbare Geräte in Hülle und Fülle: Vom teuren Apple-Gerät über günstige Notebooks bis zum Netbook ist für alle Zwecke und Geldbeutel etwas dabei. Kleinere tragbare Geräte laufen dem Laptop beim privaten Gebrauch den Rang ab (vgl. CTV News 2013). Apple hat zwar weder das Smartphone noch das Tablet erfunden, mit dem iPhone und dem iPad den Markt aber stark verändert. Das Mobiltelefon ist zum Smartphone geworden und bietet somit fast überall Zugang zum Internet. In manchen Ländern, vor allem Entwicklungsländern sind kleine tragbare Geräte sogar insgesamt stärker verbreitet (vgl. van Genabith 2009: 7). Durch die Veränderung im Design und der Handhabung hat sich auch die Benutzeroberfläche geändert. Es ist ein Unterschied, ob ein kleiner Bildschirm per Berührung gesteuert wird oder ein größerer mit Maus und Tastatur. Dies wirkt sich auf die Programme und deren Aufbau aus, somit auch auf den Bereich der Lokalisierung. Auf die Entwicklung des Smartphones, die Plattformen und Anwendungen wird im letzten Kapitel genauer eingegangen.

Insgesamt hat sich seit den ersten Einfällen, Maschinen rechnen zu lassen, sehr viel verändert. Vom Rechner von riesigen Ausmaßen ist im Alltag nicht mehr viel übriggeblieben. Verdeutlicht wird die Geschwindigkeit der Entwicklung durch das Mooresche Gesetz, wonach sich „die Zahl der Transistoren von integrierten Schaltungen jährlich verdoppelt“. (IT-Wissen: Mooresches Gesetz 2014) Auch wenn man mittlerweile von 18 Monaten ausgeht, bedeutet dies einen enormen Anstieg.

3 Lokalisierung

3.1 Dimensionen

Von außen betrachtet ist es kein schwieriges Unterfangen, das Konzept Lokalisierung zu beschreiben. Die allgemein anerkannte Definition der mittlerweile aufgelassenen „Localisation Industry Standards Association“ (LISA) lautet wie folgt: „Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold.“ (zit. n. Esselink 2000: 3¹) Ähnlich lautet auch Esselinks Definition: „In a nutshell, localization revolves around combining language and technology to produce a product that can cross cultural and language barriers. No more, no less.“ (Esselink 2003: 4) Bereits in diesen Definitionen sind jedoch verschiedene Dimensionen enthalten: Einerseits geht es um Kultur, andererseits um Sprache, auch Technologie als Schlagwort wird genannt. Beim Versuch, Lokalisierung als Ganzes zu definieren, werden oft, auch von anderen Autor/innen, die Grenzen zwischen verschiedenen Dimensionen oder Blickwinkeln nicht klar gezogen. Lokalisierung kann nach verschiedenen Gesichtspunkten betrachtet werden. Auf der höchsten Ebene steht Lokalisierung als Ganzes im globalen Gefüge. Darunter steht der Lokalisierungsprozess mit seinen verschiedenen Aufgabenbereichen. Als nächstes kommt das lokalisierte Produkt mit seinen Komponenten und schließlich die einzelnen Anpassungen, die an diesen vorgenommen werden.

3.1.1 GILT

Um dem Begriff Lokalisierung einen Kontext zu geben, wird oftmals das Gefüge „GILT“ herangezogen. Dies ist ein Akronym der Begriffe „Globalisation“, „Internationalisation“, „Localisation“ und „Translation“ (vgl. Dunne 2006^a: 4). Globalisierung bezieht sich hier auf das Agieren auf internationalen Märkten, es handelt sich also um ein Phänomen auf Unternehmensebene. Dies zeigt sich auch bei der LISA-Definition (vgl. Esselink 2000: 4), nach der es bei Globalisierung um die unternehmerischen Angelegenheiten geht, die beim globalen Vertrieb eines Produktes zum Tragen kommen. Esselink (2000: 4) erwähnt auch, dass Globalisierung mehrere Begriffe umfasst, nämlich die herkömmliche Interpretation der Integration der Weltwirtschaft, die hier verwendete Auffassung des globalen Agierens und außerdem auch Lokalisierung von Websites beschreiben kann. Hall (2002: 8) beschreibt Globalisierung, oder G11N, als Summe von Internationalisierung, oder I18N, und Lokalisierung, L10N, dies

¹ Da die Website der LISA zu diesem Zeitpunkt nicht mehr online ist, wird die Definition aus dem angegebenen Werk zitiert.

allerdings im Hinblick auf ein globalisiertes Produkt. Im Allgemeinen ist unter seiner Ausführung wohl auch die Gesamtheit aller Entscheidungen und Prozesse zu verstehen, die zu Internationalisierung und Lokalisierung führen, sowie diese organisieren und koordinieren. Der Weg des gesamten Prozesses sollte idealerweise mit der Entscheidung auf dieser Ebene beginnen (vgl. Dunne 2006^a: 6).

Als zweiter Schritt, zumindest konzeptionell, wenn auch nicht unbedingt zeitlich, ist Internationalisierung zu verstehen. Grundsätzlich bedeutet dies, ein (Software-)Produkt auf Lokalisierung vorzubereiten. Dabei wird „alles Lokale“ entfernt beziehungsweise so programmiert und gespeichert, dass es leicht ersetzt und verändert werden kann. So werden, zum Beispiel, alle Ausgabertexte in eigenen Sprachdateien gespeichert oder die Verwendung von unterschiedlichen Zeichensätzen durch Unterstützung von Unicode sichergestellt (vgl. Dunne 2006^a: 4). Internationalisierung ist zudem ein Prozess, der in die gesamte Entwicklung des Produktes eingebunden werden sollte. Nachträgliche Internationalisierung ist teuer und ist nicht immer sinnvoll, da es das Ziel dieses Konzeptes ist, das Produkt von vornherein leichter veränderbar zu machen. Internationalisierung wird von Softwareentwickler/innen vorgenommen (vgl. Hall 2002: 8). Auf Internationalisierung wird in Kapitel 3.2 näher eingegangen.

An nächster Stelle steht Lokalisierung selbst. Die Bezeichnung stammt vom Wort *Locale*, das im Englischen ursprünglich eine kleine Region bezeichnet, bezogen auf die Lokalisierungstätigkeit für eine Kombination aus Sprache und Region beziehungsweise Land verwendet wird (vgl. Esselink 2000: 1). In diese Ebene passen die zu Beginn des Kapitels aufgeführten Definitionen, die Lokalisierung als Adaptierung eines Produkts für ein bestimmtes Locale beschreiben.

Ein Bestandteil der Lokalisierung ist der letzte Punkt, Übersetzung, der oft als Kern des gesamten Prozesses angesehen wird, wobei. Wie dies nun konkret aussieht, wird in den nächsten Unterkapiteln erläutert.

Wird diese Reihenfolge, von oben nach unten, eingehalten, sollte der gesamte Prozess effizient ablaufen. Dunne (2006^a: 5-7) stellt jedoch fest, dass die Reihenfolge der Entstehung der GILT-Bestandteile genau umgekehrt verlaufen ist. Am Beginn steht Übersetzung. Damit begann auch die Geschichte der Lokalisierung; es wurden die Ausgabertexte der Benutzeroberfläche übersetzt (und die Eingabe verschiedener Sprachen ermöglicht), um zusätzliche Märkte zu erreichen. Schnell wurde ersichtlich, dass nur die Übersetzung dieser Texte nicht reichte, um der Nachfrage gerecht zu werden, so wurde das Konzept der Lokalisierung geboren. Dass es eine mühsame Arbeit war, Lokalisierung am fertigen Produkt zu vollziehen, wurde auch bald festgestellt. Als Antwort darauf entstand Internationalisierung. Da diese Prozesse gesteuert werden müssen, muss sich auch die Unternehmensleitung darauf einstellen. Lokalisierung und Internationalisierung wird organisiert, für die tatsächliche Erschließung neuer Märkte sind aber durchdachte Konzepte und Strategien nötig, dies bedeutet Globalisierung. Im Rahmen von vielen Unternehmen, die diesen Schritt wagen, gibt es weltweite

Trends zur Verknüpfung; die Welt wächst sozusagen weiter zusammen, was durch den herkömmlichen Begriff der Globalisierung ausgedrückt wird. Das Problem bei dieser Reihenfolge ist, dass die globale Sichtweise fehlt, das heißt, die Entscheidungsträger/innen haben keinen Überblick. Jeder Schritt wird als Antwort auf ein vorhandenes Problem beziehungsweise eine Herausforderung getan. Wie dieser Schritt getan wird, welche Ausmaße notwendig sind und welche Auswirkungen auftreten, wird in der Entscheidungsfindung nicht berücksichtigt, da all diese Aspekte nicht bekannt sind. Oft wird ad hoc gehandelt, also ein Produkt lokalisiert, weil sich gerade die Gelegenheit bietet. Wegen der fehlenden Internationalisierung kann bei einer erneuten Lokalisierung keine Kosten eingespart werden. Probleme ergeben sich außerdem bei der Entwicklung von Patches und Updates, da diese eigens für jede Sprachversion entwickelt werden müssen et cetera. Behält man von vornherein den Überblick und organisiert man von oben nach unten, kann man all diese Problemen vorbeugen und effizient lokalisieren.

Hierzu sollte erwähnt werden, dass Dunnes Ausführungen im Jahr 2006 herausgegeben wurden. Lokalisierungsexpert/innen arbeiten seit geraumer Zeit insbesondere auch an der Bewusstseinsbildung und an verschiedenen technischen Lösungen, um diesem Problem zu begegnen.

3.1.2 Prozesse und Beteiligte

„In sum, localization is not so much about specific tasks as much as it is about the processes by which products are adapted.“ (Dunne 2006^a: 2-3)

Kapitel 5 beschäftigt sich genauer mit dem Lokalisierungsprojekt, im Folgenden soll über die Projektteilnehmer/innen und Prozesse ein Überblick im Rahmen des Ansatzes der Lokalisierung als Prozess gegeben werden. Im Gegensatz zur herkömmlichen Übersetzung wird bei Lokalisierung kein Text übersetzt, sondern ein komplexes Softwareprodukt beziehungsweise eine Website adaptiert. Aufgrund der Beschaffenheit des Ausgangsmaterials und der Komplexität des Projektes ist die Zusammenarbeit und Expertise von Expert/innen verschiedener Fächer vonnöten. Esselink (2000: 13-23) gibt eine umfangreiche Aufstellung der beteiligten Personen und der Prozesse im Projekt. Im typischen outgesourceten Setting gibt es von Seiten des Softwareunternehmens zumindest eine/n Sachbearbeiter/in und/oder Lokalisierungsmanager/in, der/die für die Koordination und die Vorbereitung des Projekts zuständig ist. Außerdem gibt es meistens eine Qualitätssicherungsabteilung im Unternehmen selbst, die für die letztendliche Abnahme der lokalisierten Produktbestandteile zuständig ist. Qualitätsüberprüfungen werden zudem vom Personal in den Zweigstellen durchgeführt.

Für Organisation, Koordinierung und Projektmanagement werden von Seiten des Lokalisierungsunternehmens Account-Manager/innen und Projektmanager/innen eingesetzt. Ihnen fällt eine bedeutende Rolle vor und zu Beginn des Projekts zu, als auch die laufende

Überprüfung des Projektfortschritts sowie Lieferung des lokalisierten Materials und der Projektabschluss und Projektanalyse.

Übersetzer/innen sind für die Übersetzungen der Produktkomponenten zuständig und meistens auch für die Erarbeitung und Pflege der Terminologiedatenbanken. Auf die Komponenten wird im nächsten Unterkapitel eingegangen.

Für die Qualitätssicherung während des Projekts werden Korrektor/innen beschäftigt, die die Übersetzungen auf ihre sprachliche Qualität kontrollieren.

Softwaretechniker/innen kümmern sich um alle technischen Aspekte. Sie sind vor dem Übersetzen für die Vorbereitung des Materials zuständig, danach für dessen Einbindung und für das Testen der lokalisierten Software. Gegebenenfalls werden auch Expert/innen für die Bedienung von CAT-Tools eingesetzt.

Da zum Lieferumfang eines Softwareprodukts auch Broschüren, Handbücher und andere Texte gehören, ist auch der Einsatz von DTP-Spezialist/innen notwendig. Diese bearbeiten die übersetzten Texte für gedrucktes Material als auch virtuell mitgeliefertes Material.

Es ist ersichtlich, dass Lokalisierung viel mehr einschließt als Übersetzung. Dies soll nicht bedeuten, dass Management für die traditionelle Übersetzungsarbeit nicht von Bedeutung wäre, oder dass übersetzte Texte nicht vorbereitet oder weiterverarbeitet werden. Die Vielzahl an Aufgaben, Prozessen und Beteiligten machen das Behalten des Überblicks beim Lokalisierungsprojekt jedoch besonders notwendig.

Den Mangel an diesem Überblick beanstandet Dunne (2006^a: 1-3). Er schreibt, dass die Beteiligten oft nicht das Ganze sehen. Je nach Aufgabenbereich werden Teile hervorgehoben, so zum Beispiel kehren Übersetzer/innen den Übersetzungsaspekt hervor, Geschäftsleute beschäftigen sich mit der finanziellen Seite, dem gewünschten zusätzlichen Ertrag, während für Projektmanager/innen die Vereinbarung der Interessen der Beteiligten von Relevanz ist. Es ist, mangels Überblick, für die verschiedenen Bereiche oft schwierig, die Herausforderungen zu bewältigen. Softwareentwickler/innen fehlt demnach oft der Sinn für sprachliche und kulturelle Aspekte, da diese in ihrer Ausbildung keine Rolle spielen. Auf der anderen Seite stehen Übersetzer/innen, denen oftmals das fachliche Wissen im Computerbereich fehlt. Dies betont noch einmal, dass die Dimension von Lokalisierung als Projekt, das verschiedene Prozesse vereint, von grundlegender Wichtigkeit ist.

3.1.3 Produktkomponenten

Laut der weiter oben genannten Definitionen geht es bei Lokalisierung um die Anpassung eines Produktes an ein anderes Locale. Woraus besteht nun ein solches Produkt?

Grundsätzlich wird davon ausgegangen, dass es sich um Softwareprodukte oder um Websites handelt (vgl. Esselink 2000: 1). 2003 schreibt Esselink (6), dass die Grenze zwischen den beiden mehr und mehr verschwimmt, da Websites nicht nur aus Fließtext bestehen,

sondern über viele verschiedene Anwendungen und Funktionen verfügen. Softwareprodukte selbst bilden auch keine homogene Gruppe. Dies wird beim bloßen Gedanke an die Bandbreite der veröffentlichten Programme ersichtlich. Von sehr kostenintensiver, maßgeschneiderter Unternehmenssoftware bis zu frei verfügbaren Open Source-Programmen, von Betriebssystemen bis zu kleinen Spielen, es gibt im Prinzip alles. Von Websites soll in dieser Arbeit abgesehen werden, da eine genauere Beschäftigung damit den Rahmen sprengen würde. Bei Softwarelokalisierung geht man grundsätzlich von drei relevanten Komponenten aus: der Software selbst, der Hilfedatei und der Dokumentation (vgl. Esselink: 2000: 10-13).

Die grafische Benutzeroberfläche oder GUI (graphical user interface) einer Software beinhaltet übersetzbare Texte in drei Formen: Dialogfelder, Menüs und Zeichenketten oder Strings. Diese sind in Ressourcendateien oder direkt in Binärdateien gespeichert. Je nachdem können diese Dateien in einem Editorprogramm, einem Translation Memory-System oder einem Lokalisierungstool geöffnet und bearbeitet werden (vgl. Esselink 2000: 57-59). Nach der Übersetzung selbst muss die grafische Benutzeroberfläche nachbearbeitet werden, sodass die Texte korrekt dargestellt werden (vgl. Esselink 2000: 97).

Die Hilfedatei enthält (praktische) Informationen über die Software. Im Gegensatz zu einem Handbuch kann sie direkt aus einem Programm selbst geöffnet werden und leitet Nutzer/innen zum relevanten Thema. Außerdem ist sie als Hypertext aufgebaut, enthält also viele Verknüpfungen zwischen Themen und Schlagwörtern (vgl. Esselink 2000: 165-166).

Dokumentation fasst verschiedene Texte zusammen, zum Beispiel Installationsanleitungen, Kurzanleitungen, Handbücher und sonstige Texte im Lieferumfang (vgl. Esselink 2000: 289-290). Nicht immer wird die Hilfedatei separat erwähnt (vgl. Müller 2005: 17). Dokumentation und Hilfedateien sollten nach den Bestandteilen der grafischen Benutzeroberfläche übersetzt werden, da sie vielfach auf diese verweisen.

In Kapitel 5 wird näher auf die Produktkomponenten und deren Dateiformate sowie auf die Bearbeitung eingegangen. Die angeführten Aufstellungen beziehen sich auf Software für Desktopcomputer. In Kapitel 7 werden die Komponenten bei Apps sowie die Unterschiede zu „herkömmlicher“ Software erläutert.

3.1.4 Abstufungen der Adaptierung

Im vorigen Kapitel wurde über die Übersetzung der Produktkomponenten gesprochen, was natürlich nicht falsch ist; die einzelnen Dialogfelder, Menüs, Strings, Fließtexte werden natürlich übersetzt. Es sollte an dieser Stelle noch erwähnt werden, dass Übersetzen hier nicht als Transkodierung verstanden wird, sondern im Sinne der Skopostheorie (vgl. Reiß & Vermeer 1984). Dass es niemals nur eine Übersetzungsmöglichkeit gibt und dass Lokalisierung über die Übersetzung der sprachlichen Komponenten hinausgeht, wird in Dunnes Definition widergespiegelt:

„Localization involves: (a) translation of textual content into the language and textual conventions of the target locale; and (b) adaptation of non-textual content (from colors, icons and bitmaps, to packaging, form factors, etc.) as well as input, output and delivery mechanisms to take into account the cultural, technical and regulatory requirements of that locale.” (Dunne 2006^a: 2-3)

He et al (2002: 90-91) sprechen auch von verschiedenen Handlungen im Lokalisierungsprozess. So gehören dazu: Anpassungen, gegebenenfalls Hinzufügen oder Löschen von Features für eine bestimmte Zielkultur beziehungsweise Zielmarkt, Übersetzung der Benutzeroberfläche, Anpassung der Größe der Dialogfelder, Abänderung anderer Bestandteile, wie zum Beispiel Musik oder Bilder, Anpassung der Daten-, Zahlen- und Währungsformate, sowie nachfolgende Tests der lokalisierten Version. Was hier allerdings nicht beachtet wird, sind die unterschiedlichen Ebenen der an der Software vorgenommenen Arbeitsschritte. Trotz der Tatsache, dass sich die Übersetzung der Texte am Zweck, Kontext und den Konventionen der Zielkultur orientieren muss, ist eine bloße Übersetzung der Ausgabetexte noch nicht als Lokalisierung zu verstehen. Auf derselben Ebene wäre die Anpassung der Größe der Dialogfelder zu platzieren. Die Anpassung der verschiedenen Formate ist hingegen ein weiterer Schritt in Richtung Lokalisierung, während durch die Abänderung anderer Bestandteile und das Hinzufügen beziehungsweise Löschen von Features wirklich eine an die Zielkultur angepasste, lokalisierte Version der Software erstellt wird.

Was kulturelle Anpassung nun bedeutet, ist viel schwieriger zu beschreiben und zu argumentieren als, zum Beispiel, die korrekte Darstellung der Benutzeroberfläche. Die erste Kategorie von Elementen, die anzupassen sind, umfasst verschiedenste Formatkonventionen. Diese Anpassungen sind eher praktischer Natur, sie machen Software im Zielland besser nutzbar, bieten aber prinzipiell kein kulturelles Gefahrenpotenzial. Fissgus & Seewald-Heeg (2005: 222) führen einige Formate als Beispiele an, einschließlich des Datums, der Datums-kurzform, der Uhrzeit, des Kalenders, der Zahlenschreibweise, der Währung, der Adressangaben und der Maße. So wird das Datum in den USA in der Kurzform mit m/d/yy angegeben, während die in Deutschland übliche Form dd.mm.yy ist.

Weiter in Richtung kulturelle Adaptation bewegt sich Schäler (2008: 41-42). Er stuft diese weiter ab in oberflächliche und tiefgreifende Anpassung des Produkts, die sich wiederum durch alle Ebenen, von der äußerlichen Verpackung bis zum einzelnen String, ziehen. Unter den oberflächlichen, oder leicht erkennbaren Merkmalen, befindet sich die Farbe. So haben Farben in unterschiedlichen Kulturkreisen verschiedene Bedeutungen. Rot kann, zum Beispiel in westlichen Kulturen, eine Signalfarbe sein, in manchen asiatischen Ländern, wie in China, ist es die Farbe des Glücks. Grün wird auf der einen Seite mit Umwelt assoziiert, auf der anderen Seite, in arabischen Ländern, ist es eine heilige Farbe. Auch so grundlegende Gefühle wie Trauer werden durch verschiedene Farben, einmal Weiß, einmal Schwarz, wiedergegeben.

Bildmaterial ist oft sensibel. So ist, zum Beispiel, die Darstellung von Frauen mit Vorsicht zu handhaben; die falsche Kleidung oder Frauen als Führungspersonen darzustellen kann in einigen Ländern zu Missmut führen oder sogar verboten sein (vgl. Edwards 2009b: 29).

Gesten und Handzeichen bieten scheinbar die Möglichkeit, Botschaften auf begrenztem Platz unterzubringen. Doch können harmlose Gesten der einen Kultur in einem anderen Umfeld sogar Beleidigungen bedeuten, so wie zum Beispiel das US-amerikanische „Daumen hoch“ und „okay“, das aus einem Kreis besteht, der von Zeigefinger und Daumen gebildet wird.

Die Bedeutung von Symbolen erwächst auch aus dem Umfeld. Der US-amerikanische Briefkasten, bei dem ein aufgestelltes Fähnchen anzeigt, dass Post gekommen ist, wurde bereits von Microsoft entfernt, da User/innen in anderen Kulturkreisen den Zusammenhang mit einer neu erhaltenen E-Mail nicht erkannten. Dies bedeutet nicht, dass keine Symbole verwendet werden sollten, vor allem jene, die sich auf Systemkomponenten wie Drucker oder Monitor beziehen, sind allgemein gut verständlich. Vom Gebrauch von Fahnen und Flaggen zur Kennzeichnung der Sprache warnt Edwards (2006a: 25-27). Die saudi-arabische Fahne enthält beispielsweise das Glaubensbekenntnis des Islam und ist so denkbar ungünstig als Symbol für die arabische Sprache, da eine Produktverpackung mit der Fahne als Aufdruck im Prinzip nicht weggeworfen werden dürfte. Die taiwanesishe Fahne als Symbol für das Locale Taiwan zu verwenden scheint auf den ersten Blick in Ordnung zu sein, wird das Produkt jedoch auch in China vertrieben kann das zu Unstimmigkeiten führen, da die Volksrepublik China die Republik China nicht anerkennt. Ein weiteres Beispiel aus dem Bereich Websites ist die Verwendung eines Halbmonds, der dem islamischen Hilal gleicht, für astrologische Inhalte. Astrologie ist im Islam verboten, der Gebrauch des Symbols kann in diesem Fall unter Umständen sogar als Affront aufgefasst werden (vgl. Edwards 2007: 33).

Geräusche und Tonmaterial werden auch unterschiedlich assoziiert. Dazu führt Schaller (2008: 42) das Beispiel des Gongs als Begleitung einer Fehlermeldung an. Dies würde in westlichen Kulturkreisen als normal akzeptiert, jedoch wäre es in Japan einem/r Büroangestellten peinlich, vor den Kolleg/innen offensichtlich einen „Fehler“ gemacht zu haben. Aus dem religiösen Bereich führt Edwards (2006b: 30) das Spiel „Kakuto Chojin“ an, bei dem als Hintergrundmusik ein gesungener Koran-Vers verwendet wurde. Zuerst wurde das Prügelspiel im arabischen Raum verboten, dann sogar weltweit zurückgezogen.

Vielleicht weniger relevant für Bürosoftware, jedoch umso mehr bei Spielen oder Web-Inhalten ist Geschichte. Ein Beispiel ist das Strategiespiel „Age of Empires“ mit historischem Inhalt. Das südkoreanische Informationsministerium beanstandete die Darstellung einer japanischen Invasion im Reich der Joseon-Dynastie, die im Spiel, den historischen Quellen folgend, als erfolgreich dargestellt wurde. Laut der koreanischen Geschichtsauffassung war die Verteidigung jedoch nicht vollkommen zerschlagen worden, wie es im Spiel vorkam. Auch die griechische Regierung war nicht zufrieden mit der Darstellung von Ale-

xander dem Großen. Laut griechischer Deutung liegt seine Herkunft in Griechenland und nicht in Mazedonien (vgl. Edwards 2008: 27).

Der letzte Punkt, den Schäler (2008: 42) anführt, ist die Verwendung von Produktnamen und Akronymen, so zum Beispiel der Abkürzungen von internationalen Organisationen, die sich wegen der Grammatik der unterschiedlichen Sprachen verändern. Das im Englischen und im Deutschen verwendete Akronym der „North Atlantic Treaty Organization“, NATO, lautet im Französischen und Spanischen OTAN.

Diese kulturellen Anpassungen liegen, wie oben genannt, an der Oberfläche. Schwieriger wird es, wenn ein Produkt tiefergehend kulturell angepasst werden soll, das heißt die Struktur der Benutzerschnittstelle oder einer Website verändert wird. Der Grund für Anpassungen, die über die oben genannten hinausgehen, liegt laut Fort & Kotzé (2005: 716-717) darin, dass Menschen mit Computern, genauso wie mit anderen Menschen, interagieren. Da in unterschiedlichen Kulturen Kommunikation und Interaktion unterschiedliche Formen annehmen, ist auch die Herangehensweise an die Mensch-Computer-Interaktion nicht universal. Folglich wird der Umgang mit dem Computer vereinfacht, wenn dessen Bedienung die Verhaltens- und Kommunikationsmuster der jeweiligen Kultur widerspiegelt. Die Benutzeroberfläche für Angehörige einer Kultur intuitiver zu machen ist insofern eine Herausforderung, als dass Kultur an sich schwer messbar ist. Schäler (2008: 42-44), Marcus (2005: 43-65) und Ford & Kotzé (2005: 716-723) beziehen sich deshalb auf Geert Hofstede's Kulturmodell: Bei Hofstede wird Kultur als „programming of the mind“ bezeichnet (Geert Hofstede Centre 2014). Dabei fungieren fünf Dimensionen, um Kulturen zu unterscheiden: Machtdistanz (Power Distance Index PDI), Individualismus und Kollektivismus (Individualism IDV), Maskulinität versus Femininität (Masculinity MAS), Ungewissheitsvermeidung (Uncertainty Avoidance Index UAI) und lang- oder kurzfristige Ausrichtung (Long-Term Orientation LTO). Bei jeder Dimension gibt es zwei Extreme und jede Kultur, die analysiert wird, wird auf der dazwischen liegenden Linie platziert.

Die Machtdistanz gibt an, wie egalitär Menschen in einer Gesellschaft sind; wie weit sie Machtgefälle erwarten und akzeptieren. In Gesellschaften mit hoher Machtdistanz fällt es den schwächeren Mitgliedern schwerer, sich gegen Personen in Autoritätspositionen zu behaupten.

Die Dimension Individualismus und Kollektivismus bezieht sich auf die Rolle des Individuums und der Gruppe. In individualistischeren Gesellschaften steht jede Person für sich, hat dafür aber auch mehr Freiheit und Selbstbestimmungsrecht, während Menschen in kollektivistischen Gruppen stark in diese eingebunden sind und darin Schutz finden, dafür aber auf persönliche Freiheiten verzichten müssen.

Bei Maskulinität versus Femininität geht es um die Anwendung von traditionellen Gender-Rollen auf die Gesamtgesellschaft. Werden Konkurrenzverhalten, Erfolg und Macht hoch angesehen, ist die Gesellschaft eher maskulin, während eine feminine Gesellschaft eher

auf ein gutes Zusammenleben ausgerichtet ist. Außerdem gibt es in einer feminineren Gesellschaft mehr Gleichberechtigung zwischen den Geschlechtern.

Ungewissheitsvermeidung bezieht sich auf die Art und Weise, wie Menschen mit Unsicherheit und Risiko umgehen. Für Menschen und Gruppen, die in dieser Kategorie hoch eingestuft sind, ist es wichtig, unsichere Situationen zu vermeiden und sich Struktur zu schaffen. Dies führt zu genaueren und zahlreicheren Regeln und Gesetzen.

Die zeitliche Ausrichtung beschreibt, wie sehr Menschen an Vergangenheit, Gegenwart und Zukunft denken. In Kulturen, die langfristig ausgerichtet sind, spielt Tradition eine große Rolle und fixe Machtverhältnisse sowie ein hoher Status von „ehrllicher Arbeit“ sind die Norm. In kurzfristig orientierten Gesellschaften sind auch Individualismus und Egalität ausgeprägt. Außerdem werden Veränderungen leichter zugelassen.

Marcus (2005: 49-50) wendet diese Kriterien auf die Benutzeroberfläche an, um nicht nur die äußerlich sichtbaren Merkmale (Bilder, Symbole et cetera) besser an die Zielkultur anzupassen, sondern den gesamten Aufbau und das Konzept intuitiver zu machen. Er meint, dass es auf lange Sicht erstrebenswert wäre, Kataloge für verschiedene Kulturen zu erstellen, samt einer Inkludierung in ein Tool, das einem bei der Entwicklung Vorschläge für eine an die Kultur angepasste Benutzeroberfläche liefert. Hohe Machtdistanz bei der Benutzeroberfläche bedeutet hierbei, dass der Zugang zu Informationen strukturiert ist, dass das Hauptaugenmerk auf Führungspersonen liegt, Sicherheitsmaßnahmen angewendet werden und auch sichtbar sind. Zudem werden Autoritäten betont. Das Gegenteil gilt bei Benutzeroberflächen für Gesellschaften mit niedriger Machtdistanz. Bei Benutzeroberflächen, die eine hohe Unsicherheitsvermeidung aufweisen, sollte es, zum Beispiel, nur die notwendigsten Menüoptionen geben, die Hilfe leicht zu finden sein und die Navigationsstruktur möglichst zielführend sein, um Probleme zu vermeiden. Im Gegenteil dazu würden Benutzeroberflächen, die eine geringe Unsicherheitsvermeidung repräsentieren, die Erforschung und Eigeninitiative der Nutzer/innen fördern, also viele Menüoptionen enthalten. Maskulin orientierte Benutzeroberflächen sollten nur eine begrenzte Anzahl von Möglichkeiten enthalten, die dafür aber schnell zum Ziel führen. Die Navigationsstruktur sollte dem/der Nutzer/in die Kontrolle in die Hand geben. Der Inhalt sollte herausfordernd sein und gleichzeitig Geschlechts- und Altersgruppen explizit trennen. Grafiken und Animationen sollten praktisch und funktionell sein. Dagegen sollten feminine Benutzeroberflächen ästhetisch ansprechend sein, um die Aufmerksamkeit des/der Nutzer/in zu erlangen. Die Inhalte sollten Geschlechterrollen nicht hervorkehren und insgesamt sollten diese Benutzeroberflächen Kooperation und den Austausch von Ideen und Hilfe fördern. Individualistisch orientierte Benutzeroberflächen sollten durch materialistisches Bildmaterial die Verbindung der Konsumgesellschaft und Erfolg herstellen. Die Erregung der Aufmerksamkeit erfolgt über die Konzepte Jugend, Handlungen und Einzelpersonen. Bei den Inhalten sollte es um persönliche Ziele gehen, um neue Produkte und Konzepte. Außerdem sollten Kontroverse und persönliche Meinungen gefördert werden. Die Nutzer/innen sollten nicht dazu aufgefordert werden, persönliche Informationen weiterzugeben.

Bei kollektivistischen Benutzeroberflächen sollten soziopolitische Pläne mit Erfolg verbunden werden und erfahrene Charaktere zur Erregung der Aufmerksamkeit verwendet werden. Bei den Inhalten sollten Gruppenerfolge, Geschichte und Tradition hervorgehoben werden und die Bildung von persönlichen Meinungen nicht gefördert werden. Benutzeroberflächen, die zeitlich nicht langfristig ausgerichtet sind, sollten so strukturiert sein, dass Aufgaben schnell ausgeführt werden können. Regeln sollten angewendet werden, um den Wahrheitsgehalt von Informationen zu bestätigen. Bei langfristig orientierten Benutzeroberflächen können Navigation und Inhalte komplexer sein, da die Nutzer/innen bereit sind, mehr Zeit in deren Entdeckung zu investieren. Die Inhalte sollten praktischen Nutzen haben und der Wahrheitsgehalt kann durch persönliche Aussagen bestätigt werden (vgl. Marcus 2005: 50-54). Ford & Kotzé (2005: 724) kritisieren diese kategorische Einteilung als zu starr und zu sehr von Stereotypen geleitet. Sie kommen außerdem zum Schluss, dass eine Repräsentation von hoher Machtdistanz, Maskulinität, hoher Unsicherheitsvermeidung und Kollektivismus in der Benutzeroberfläche allgemein zu besseren Ergebnissen führen, auch wenn diese Merkmale in der Zielkultur nicht ausgeprägt sind.

Ungeachtet der Kritik ist die tiefergehende Anpassung der Benutzeroberfläche an die Kultur nach den Dimensionen von Hofstede ein interessanter Ansatz, der jedoch auch einen relativ starken Eingriff darstellt. Es ist wohl auch nicht bei allen Anwendungen und Websites sinnvoll, so weit zu gehen. Außerdem stehen bei den Entscheidungsträger/innen die wirtschaftlichen Überlegungen nach wie vor im Mittelpunkt, das heißt, es wird nur so wenig wie möglich, mit dem geringstmöglichen (finanziellen) Aufwand verändert. Die oben beschriebenen Überlegungen zeigen jedoch, dass Lokalisierung, zumindest theoretisch, viel mehr einschließt als bloße Übersetzung. Es gibt also Ansätze, die Lokalisierung auf eine ganz andere Ebene heben, von der Transkodierung der sichtbaren Texte, die für das Verständnis unbedingt notwendig sind, hin zu einer Anpassung nicht nur der sichtbaren Teile einer Anwendung, sondern sogar der kompletten Schnittstelle zwischen Mensch und Maschine.

3.2 Internationalisierung

„Internationalization is the process of generalizing a product so that it can handle multiple language and cultural conventions without the need for re-design. Internationalization takes place at the level of program design and document development.” (LISA zit. n. Esselink 2000: 25)

Internationalisierung wurde bereits im vorigen Kapitel im Zusammenhang mit Globalisierung und Lokalisierung erwähnt. Wie im Zitat ersichtlich, handelt es sich primär um die technische Erleichterung von Lokalisierung, vorrangig mit dem Ziel, kosten- und ressourceneffizienter zu arbeiten. Dabei erstreckt sich Internationalisierung auch auf die sprachliche Komponente, um die Texte, die schließlich übersetzt und bearbeitet werden, auch besser übersetzbar zu machen. Im folgenden Kapitel sollen die Entwicklung von Internationalisierung, die techni-

schen Anpassungen, die Möglichkeiten der sprachlichen Optimierung sowie verschiedene Probleme und Lösungen behandelt werden.

3.2.1. Herausbildung von Internationalisierung

Schon seit den neunziger Jahren versuchen Lokalisierungsexpert/innen, die Industrie auf diese Verknüpfung, vor allem auf die Interdependenz Internationalisierung – Lokalisierung, aufmerksam zu machen. Mittlerweile arbeiten die meisten, zumindest die großen, Softwarefirmen nach diesem Prinzip. Zu Beginn war davon noch keine Rede, im Folgenden wird die Entwicklung des Lokalisierungs- und Internationalisierungsprozesses dargestellt. Die Bezeichnungen des „Typs“ der Lokalisierung hängen zusammen mit der Programmphase, für die die jeweils veränderte Datei gebraucht wird.

Ganz zu Beginn wurde nur die ausführbare Datei verändert. Ein/e Programmierer/in öffnete die EXE-Datei und suchte nach zu lokalisierenden Elementen, um dann eine lokalisierte ausführbare Datei zu erstellen. He et al (2002: 91) bezeichnen dies als „Run-time Localisation“, also Laufzeit-Lokalisierung.

„Compile-time Localisation“ bezeichnet die nachfolgende Stufe, bei der der/die Entwickler/in Zugriff zum Quellcode hat. Ausgehend von diesem wird eine lokalisierte Version des Programms erstellt. Der Quellcode wird während der Übersetzungszeit vom Compiler in Maschinencode übersetzt. Die ersten beiden Typen erfordern relativ viel Arbeitsaufwand, da im Prinzip das Programm umgeschrieben werden muss. Es kann auch keine Zeit eingespart werden, wenn man mehrere lokalisierte Versionen erstellen möchte, das heißt, zu diesem Zeitpunkt wurde noch nicht internationalisiert.

Der dritte Ansatz kombiniert bereits Lokalisierung und Internationalisierung. Bei „Compile-time Internationalisation and Compile-time Localisation“ werden alle lokalen Elemente aller Locales in den Quellcode geschrieben und dann für die jeweiligen lokalisierten Versionen verlinkt und kompiliert.

„Compile-time Internationalisation and Link-time Localisation“: Dies geht noch einen Schritt weiter in Richtung Internationalisierung. Es werden alle zu lokalisierenden Daten und Codes aus dem Programm extrahiert, so dass nur diese Module verändert werden müssen. Bis jetzt muss die Hauptaufgabe noch immer von Softwareexpert/innen durchgeführt werden.

Beim Ansatz „Compile-time Internationalisation and Run-time Localisation“ werden wiederum alle zu lokalisierenden Daten extrahiert, jedoch komplett aus dem Programm isoliert, sodass zur Wiedereinfügung keine Programmierarbeit mehr nötig ist. Die lokalisierten Versionen existieren als Sprachdateien, die alle mit der internationalisierten ausführbaren Datei funktionieren.

Der finale Ansatz bei He et al (2002: 91-92) ist jener der „Design-time Internationalisation“. Dies bedeutet, dass Internationalisierung und Lokalisierung nicht an einem bereits bestehenden Programm vorgenommen werden, sondern bereits während der Entwicklung des Programmes selbst. Auf diese Weise wird die größtmögliche Effizienz erreicht, da die zu lokalisierende Anwendung nicht nachträglich verändert werden muss. Diese Reihenfolge spiegelt das Konzept GILT, beschrieben in Kapitel 3.1.1. wider.

3.2.2. Internationalisierung von Software

In diesem Kapitel soll näher auf die einzelnen technischen Aspekte eingegangen werden, die bei der Internationalisierung einer Anwendung beachtet werden müssen. Die wichtigsten Aspekte sind Zeichenkodierung, Speicherung der übersetzungsrelevanten Dateien, Design der grafischen Benutzeroberfläche und regionale Formatierungen (vgl. Esselink 2000: 31).

So wie jede andere visuelle Ausgabe eines Computers sind auch Zeichen, also Buchstaben, Zahlen et cetera, kodiert. Zu Beginn wurden Zeichen nach dem ASCII-Standard kodiert, der über 8 Bit, oder 256 Positionen verfügt, wobei den 26 Buchstaben des englischen Alphabets (Groß- und Kleinbuchstaben), den Ziffern 0-9 und mehreren Satzzeichen und Symbolen die Werte von 32 bis 126 zugewiesen wurden. 0-31, sowie 127 waren für Steuerzeichen reserviert und 128-255 waren nicht belegt. Dies waren relativ wenige Plätze für Sonderzeichen anderer Sprachen, ganz zu schweigen von anderen Schriftsystemen. Daher wurden die verfügbaren Werte von verschiedenen Anwendungen unterschiedlich interpretiert, was zu Problemen bei der Kompatibilität führte. Die Lösung dafür war der Standard Unicode, der über mehr als 95.000 Positionen verfügt. Es wurden nicht nur Zeichensätze von verschiedensten Schriftsystemen darin aufgenommen, sondern auch verschiedenste Kriterien integriert, die für die korrekte Darstellung der unterschiedlichen Schriften notwendig sind. Die Unterstützung von Unicode ist also eine der grundlegendsten Maßnahmen für internationalisierte Software, da dadurch die Übersetzung der Ausgabetexte in alle Sprachen als auch die Eingabe der relevanten Zeichen ermöglicht wird (vgl. Gillam 2005: 8-9).

Nicht zu vernachlässigen sind die Art und Weise der Speicherung sowie die Dateistruktur der Anwendung. Wie im vorigen Unterkapitel erwähnt, ist es üblich, den Code und die Daten einer Anwendung zu trennen. Die Ressourcendatei enthält somit alle übersetzungsrelevanten Elemente, zum Beispiel im Format DLL. Die Vorteile sind folgende: Für jede lokalisierte Version einer Anwendung muss nur die entsprechende Ressourcendatei verändert werden, es ist außerdem der Programmcode vor unabsichtlichen Veränderungen geschützt und die Wahrscheinlichkeit, dass übersetzungsrelevante Ausgabetexte übersehen werden, ist gering. Zudem ist es einfacher, die relevante Ressourcendatei zu finden, wenn eine eindeutige Speicherstruktur verwendet wird und nicht alle Dateien in einem Ordner enthalten sind (vgl. Esselink 2000: 32).

Nach der Übersetzung von Ausgabetexten ist es gut möglich, dass diese länger werden, abhängig von der Zielsprache und dem Schriftsystem. Daher sollte in Feldern und Menüs 30 % mehr Platz gelassen werden beziehungsweise die Größe der jeweiligen Felder nicht hartkodiert werden. Außerdem sollten Grafiken, die Text enthalten, einfach zu bearbeiten sein (vgl. Esselink 2000: 27; 33-34).

Wie bereits in Kapitel 3.1.4. erwähnt, gibt es in verschiedenen Ländern unterschiedliche Formate, wie zum Beispiel Maße, Adresse, Datum und Zeit. Diese sollten von der Anwendung unterstützt werden und am besten kodiert und an das Locale geknüpft werden, so dass die jeweilige Anzeige automatisch verändert wird, wenn das Locale des Betriebssystems erkannt wird (vgl. Esselink 2000: 33-35).

3.2.3. Sprachliche Vorbereitung auf Lokalisierung

Der Großteil der Software wird noch immer auf Englisch entwickelt. Da Sprache immer die Kultur widerspiegelt, in der sie verwendet wird, muss für Softwareentwickler/innen und technische Redakteur/innen klar sein, dass sie bei der Verfassung ihrer Texte das globale Zielpublikum der lokalisierten Versionen im Kopf behalten müssen. So wie bei Symbolen, die nicht universal verständlich sind, kann es auch bei Beispielen, Referenzen und anderen Assoziationen in ihren Texten zu Unklarheiten kommen, wenn sie zu kulturspezifisch sind. Für die Texte der grafischen Benutzeroberfläche und auch für die Texte in der Hilfedatei sowie der Dokumentation gibt es mehrere Dinge, die beim Verfassen zu beachten sind. Einerseits sind es Formatierung und Typografie, die eine Rolle spielen, andererseits Terminologie und Sprache.

Geht man davon aus, dass Übersetzer/innen Translation Memory-Systeme zur Bearbeitung der Texte verwenden, sollten verschiedene Aspekte beachtet werden. Zuerst, vor allem relevant für die Dokumentation, sollte auf das Dateiformat geachtet werden. Wenn PDF-Dokumente verwendet werden, muss die Datei zuerst in ein anderes Format konvertiert werden beziehungsweise der Text herauskopiert. Dabei schleichen sich üblicherweise Formatierungsfehler ein, die bearbeitet werden müssen, vor allem wenn ein Translation Memory-System verwendet wird. Daher ist es ratsam, auf PDF-Dateien zu verzichten.

Bei der Formatierung sollte auch die weitere Verarbeitung der Texte nicht vergessen werden. Verschiedene nicht korrekte Steuerzeichen führen dazu, dass der Text beim Importieren in ein Translation Memory-System nicht optimal segmentiert beziehungsweise erkannt wird. Wird ein Absatzzeichen innerhalb eines Satzes benutzt, weil es sich zum Beispiel um den Text eines Dialogfeldes handelt, der nicht in einer Zeile Platz findet, so erkennt das Translation Memory-System diesen Satz als zwei Segmente, da diese normalerweise durch Absätze, Punkte, Ausrufezeichen und Fragezeichen unterteilt werden. Dadurch wird wiederum ein Match oder Fuzzy Match nicht erkannt und die falsch abgetrennten Segmente müs-

sen zusammengefügt werden. Dasselbe, nur umgekehrt, ist der Fall, wenn einfache Zeilenumbrüche statt dem Absatzzeichen verwendet werden. Dadurch wird das Ende eines Segments nicht erkannt, was wiederum zu falscher Segmentierung führt.

Statt der Tabulatortaste mehrmals die Leertaste zu benutzen führt zu Verwirrung seitens der Übersetzer/innen. Für diese ist unklar, ob die Leerzeichen eine spezifische Funktion haben und deshalb belassen werden müssen, oder nicht.

Die richtigen Formate sind auch hier ein Thema. Wird eine Zahl oder ein Datum im richtigen Format eingegeben, so erkennt das Translation Memory-System diese als solche und überprüft im übersetzten Segment, ob das richtige Format in der Zielsprache verwendet wurde. Ist die Eingabe nicht korrekt, fällt auch die Überprüfung weg (vgl. Keller 2011: 43-44).

Terminologie ist grundsätzlich ein wichtiges Thema bei Übersetzung und Lokalisierung und wird in Kapitel 5.2 behandelt. Hier soll kurz auf jene Aspekte eingegangen werden, die bei der Ausgangstextproduktion von Bedeutung sind.

Lombard (2006: 156-158) beanstandet, dass Terminologearbeit üblicherweise erst mit dem Lokalisierungsprojekt beginnt. Vor der Übersetzung der Texte werden, normalerweise vom Übersetzungsteam, Glossare beziehungsweise Terminologiedatenbanken angelegt. Dies wird ungleich schwieriger, wenn nicht schon vorher Begriffe und Benennungen bewusst gefunden und verwaltet werden und während des Entwicklungsprozesses konsequent verwendet werden. Wird dies nicht gemacht, so ist es wahrscheinlich, dass für dieselben Befehle und Meldungen unterschiedliche Formulierungen gebraucht werden. Dies wird wiederum verstärkt, da normalerweise eine Vielzahl an Programmierer/innen an den verschiedenen Teilen eines Programms arbeitet. Es ist für die Übersetzer/innen und Terminologieverantwortlichen im Nachhinein nicht klar, ob zwei unterschiedliche Wortlaute, die eigentlich dasselbe bedeuten, gleich übersetzt werden sollen oder nicht. Dies führt zu wiederholten Rückfragen, längerer Arbeitszeit und somit zu höheren Kosten. Auch Esselink (2000: 28) rät dazu, vor beziehungsweise während der Softwareentwicklung Glossare zu erstellen und die aufgenommenen Begriffe und Benennungen kohärent zu verwenden. Außerdem weist er darauf hin, dass dies auch für die Hilfedatei und Dokumentation gilt, da diese sich ja auf die Anwendung beziehen.

Bei der Erstellung der Texte ist es außerdem wichtig, dass die Benennungen richtig geschrieben und gegebenenfalls richtig abgekürzt werden. Dies ist bei der Verwendung von Translation Memory-Systemen in Hinblick auf die integrierten Terminologie-Tools besonders zu beachten. Wird eine Abkürzung verwendet, die nicht in der Datenbank aufscheint, wie zum Beispiel „apx.“ statt „approx.“ für „approximately“, so wird dies erstens nicht erkannt und der Terminus im Terminologie-Tool nicht angezeigt, und zweitens am Punkt ein Segmentende erkannt und der Text wiederum falsch segmentiert (vgl. Keller 2011: 44-45).

Wie bereits angedeutet, sind auch Formulierungen und der Inhalt der Texte ein Aspekt, der Beachtung verdient. Bei der Übersetzung erschwert ohnehin der Umstand, dass man die Texte meistens nicht im Kontext sieht, die Arbeit. Deshalb ist es umso wichtiger, dass die Botschaften klar formuliert und nicht zweideutig sind. Dies gilt vor allem in Bezug auf Wortketten. Im Englischen ist es außerdem nicht üblich, hier Passiv zu verwenden, sondern kurze, einfach formulierte Sätze. Von Referenzen auf kulturspezifische Elemente, zum Beispiel politischer Natur, Sportereignisse, Humor oder Klischees, ist abzusehen. Es sollten auch keine Tiere oder Religionen erwähnt werden. Schwierig ist es auch bei Fixpunkten, die in einem Kulturkreis vollkommen klar sind, wie zum Beispiel Feiertage, Klima oder Jahreszeiten, da diese weltweit unterschiedlich sind. Um bei der Übersetzung Probleme zu vermeiden, sollte auf Akronyme und Abkürzungen eher verzichtet werden.

Um eventuellen Schwierigkeiten vorzubeugen, gibt es verschiedene Ansätze für kontrollierte Sprache, wie zum Beispiel „simplified English“. Diese enthalten gewisse Wörter, die verwendet beziehungsweise nicht verwendet werden dürfen, und geben auch Grammatikregeln vor, zum Beispiel den Satzbau betreffend. Dadurch sollen die verfassten Texte einfach zu verstehen und einfach zu übersetzen sein (vgl. Esselink 2000: 29-30).

3.2.4. Probleme und Lösungsansätze

So einfach dies alles klingt, die tatsächliche Abstimmung von Internationalisierung und Lokalisierung ist in der Praxis trotzdem ein komplexes Unterfangen. Ein Problem, das den reibungslosen Ablauf gefährdet ist die fehlende Zusammenarbeit und Abstimmung von Internationalisierungsteam und Lokalisierungsteam. Dies ist auch kein abwegiges Szenario, da in der momentanen Praxis das Outsourcing-Modell überwiegt und die verschiedenen Beteiligten oft geografisch voneinander getrennt arbeiten.

Sprachdienstleister werden oft erst beauftragt, wenn die Internationalisierung eines Produktes abgeschlossen ist. Das Problem dabei ist, dass die Internationalisierung mangels erfahrener Expert/innen so vorgenommen wurde, dass erst recht gewisse Hürden auftauchen. Henderson (2010: 4-5) spricht von Projekten, bei denen Dateiformate für die Speicherung des übersetzungsrelevanten Texts verwendet werden, die von den üblicherweise eingesetzten CAT-Tools nicht importiert werden können. Durch die zusätzliche Bearbeitung und Konvertierung entsteht ein Mehraufwand, der sich in den Kosten niederschlägt. Bei einem anderen Beispiel wurden die englischen Strings in einer mehrsprachigen Ressourcendatei jeweils unter die übersetzten Strings hinzugefügt. Dadurch musste der neue Text erst mühsam herausgesucht, übersetzt und schließlich wieder in eine Ressourcendatei zusammengefügt werden.

Durch die Einsetzung von unterschiedlichen Tools können Komplikationen entstehen, wenn die Codes für Sprachen beziehungsweise Locales nicht einheitlich verwendet werden. Henderson (2010: 5) nennt als Beispiel den Code „gr“, der bei einem Unternehmen Deutsch,

bei einem anderen Unternehmen Griechisch bedeutete, was sogar zur Übersetzung in die falsche Sprache führte.

Durch Zusammenarbeit des Internationalisierungs- und des Lokalisierungsteams von Beginn an können diese Missverständnisse vermieden werden. Dies ist jedoch nur dann eine realistische Option, wenn der Kunde regelmäßig Aufträge mit hohem Volumen vergeben. Es ist hilfreich, wenn alle Internationalisierungs- und Lokalisierungsmaßnahmen zentral geregelt werden, zum Beispiel mit einem „Translation Management System“. Dadurch kann sichergestellt werden, dass die Dateiformate, Terminologie et cetera durchgängig von allen Beteiligten ohne Probleme verwendet werden können (vgl. Henderson 2010: 4).

Die oben genannten Probleme tauchen auf, wenn, trotz der Einhaltung der „richtigen“ Reihenfolge, die Abstimmung nicht erfolgt. Das folgende Beispiel zeigt ein Problem, das aus der Umkehrung der Reihenfolge entsteht. Dies ist, wie in Kapitel 3.1.1. erläutert, vor allem durch die natürliche Entwicklung von Lokalisierung und Internationalisierung bedingt.

Bei nicht internationalisierten Anwendungen kann es sich trotzdem auszahlen, diese nachträglich zu internationalisieren. Peng et al (2009: 1-5) beschäftigen sich mit der Aufgabe, eine Lösung zu finden, die die ASCII-Kodierung mit Unicode ersetzt. Durch einen Algorithmus wird die Kodierung erkannt und im Quellcode mit der entsprechenden Unicode-Kodierung ersetzt. Obwohl nicht alle Zeichen erkannt werden, ist diese Methode doch erfolgreich, da der Arbeitsaufwand, die Kodierung manuell zu ändern, bei einem Quellcode von mehreren Tausend Zeilen ins Unermessliche steigen würde.

4 Die wirtschaftliche Dimension

4.1 Entwicklung der Lokalisierungsindustrie

Lokalisierung an sich und somit die Lokalisierungsindustrie entstanden in den 1980er Jahren. Zuvor war die Anpassung der Benutzeroberfläche an andere Sprachen und Kulturen kein Thema, da die Bedienung der Großrechner ohnehin eine Expertentätigkeit war. Auch mit dem allmählichen Einzug der Minicomputer in die Arbeitswelt in den Siebzigern war deren Nutzung auf den Arbeitsplatz beschränkt und vor allem die Benutzeroberfläche alles andere als selbsterklärend. Erst in den Achtzigerjahren, mit der Einführung und Verbreitung des Desktop-Computers, stieg die Anzahl der Nutzer/innen, die keine Fachleute auf dem Gebiet waren (vgl. Esselink 2003: 4). Einige Softwarehersteller kamen zu dem Schluss, dass es einfacher werden würde, Software und Computer in anderen Ländern zu verkaufen, wenn die Texte der Benutzeroberfläche übersetzt würden. Zu Beginn unterschied sich jedoch das Verständnis von Lokalisierung selbst deutlich von dem Umfang, den der Begriff heutzutage mit sich trägt. Das Ziel der Anpassung von Anwendungen an andere Märkte war, und ist nach wie vor, deren Erschließung und somit höhere Erträge. Von kulturellen Unterschieden war damals noch keine Rede, es ging um die reine Übersetzung der für die Nutzer/innen sichtbaren Texte der damals verfügbaren Programme. Grundlegende Anpassungen an das Schreibsystem et cetera mussten natürlich auch vorgenommen werden, so wie zum Beispiel die Integration von Sonderzeichen für Eingabe und Ausgabe. Bei den Anwendungen handelte es sich meistens um Bürosoftware, also Texterstellungsprogramme, Tabellenkalkulationsprogramme et cetera. Die Übersetzungen wurden von den unternehmenseigenen Übersetzungsabteilungen, Freelancer/innen oder dem Personal der Zweigstellen angefertigt. Diese Übersetzungen wurden außerdem am fertigen Produkt vorgenommen, weswegen die Texte hartkodiert waren und die Arbeit deshalb kompliziert und mühsam war. Auch fehlten Standards, zum Beispiel für die verwendeten Zeichensätze, was zu Kompatibilitätsproblemen führte. Ohne Internationalisierung gab es auch die Möglichkeit nicht, Arbeitsschritte einzusparen, wenn mehrere lokalisierte Versionen hergestellt werden mussten. Da das gesamte Feld neu war, gab es noch keine Best-Practice-Methoden, geschweige denn eingehende Forschung (vgl. Yeo 1996: 348; Esselink 2003: 4; Schäler 2008: 39).

Mitte der Achtzigerjahre wurden die ersten Sprachdienstleistungsunternehmen gegründet, die mehrere Sprachen anboten (Multi Language Vendors MLV). Trotzdem wurde die meiste Arbeit von den eigenen Unternehmensabteilungen oder kleinen Einzelbetrieben übernommen. Dabei stand auch noch die Übersetzung im Vordergrund, das spiegelt sich auch in der Abrechnung wider, die noch auf der Wortanzahl basierte (vgl. Esselink 2000: 5; Cyr & Lew: 354).

Das Outsourcing-Modell wurde Anfang der Neunziger immer beliebter, da die Menge der Arbeit je nach Projektphase zu- oder abnimmt. Die MLVs boten auch immer mehr Dienste an und wurden sozusagen zu Allroundern, die die Lokalisierung der Produkte von Anfang bis Ende übernahmen. Mit diesem Übergang von Übersetzung hin zu Lokalisierung begannen auch andere Bereiche, sich damit zu beschäftigen. So wurde, zum Beispiel, die „Localisation Industry Standards Organisation“ LISA im Jahr 1990 gegründet (vgl. Esselink 2000: 6). Ab der Mitte der Neunzigerjahre begann eine Konsolidierungsperiode, unter anderem angetrieben durch die Verbreitung des Internets, relativ günstige Hard- und Software und das Voranschreiten des Outsourcing-Modells. Es entstanden einige globale Akteure, die viele kleine Unternehmen aufkauften und sich auch untereinander zusammenschlossen. Einige von diesen waren Berlitz, Lionbridge, Lernout & Hauspie und ALPNET (vgl. Esselink 2000: 6-7; Cyr & Lew 2003: 354).

Interessanterweise wurde Irland zum Zentrum der noch relativ jungen Lokalisierungsindustrie. Einerseits hatten viele US-amerikanische Softwareunternehmen dort Zweigstellen, andererseits waren die Wirtschaftsbedingungen günstig (vgl. Esselink 2003: 4-5). Die irische Wirtschaftsförderungsagentur versuchte, Anreize für Investitionen aus dem Ausland zu schaffen. Dabei setzte sie, nach fehlgegangenen Versuchen, sich als Produktionsland zu etablieren, auf Forschung und Entwicklung, vor allem im IT-Bereich. Für Unternehmen, die sich in Irland ansiedelten, gab es finanzielle Vergünstigungen wie Zuschüsse, eine Körperschaftssteuerquote von 10 % und Umsatzsteuerbefreiung. Die Lohnkosten und Lohnnebenkosten waren außerdem niedriger als in der restlichen Europäischen Union und den USA. Zudem gab es in Irland zu dem Zeitpunkt viele junge, gut ausgebildete Arbeitskräfte, die Hälfte der Bevölkerung war unter 25 Jahre alt. Diese Faktoren, zusätzlich zur Landessprache Englisch, machte Irland zur idealen Brücke über den Atlantik (vgl. Esselink 2000: 7; Esselink 2003: 4-5).

Ab den späten Neunzigerjahren entwickelten sich auch die großen MLVs weiter in Richtung globale Lösungen. Im Zusammenhang mit der fortschreitenden Entwicklung des Internets und verschiedener Anwendungen wurde die komplette Vernetzung aller Beteiligten und Arbeitsschritte angestrebt. Der Arbeitsbereich ging immer mehr über Softwareanwendungen hinaus und schloss immer mehr mit ein, wie zum Beispiel Webinhalte (vgl. Cyr & Lew 2003: 354).

Zum Zeitpunkt der frühen 2000er wurde mit einem jährlichen Wachstum der Lokalisierungsindustrie um 30 % gerechnet. Lokalisierung war als Wirtschaftsfaktor nicht mehr wegzudenken, so stammten über 60 % der Erträge von Microsoft aus dem internationalen Verkauf. Durch die immer weiter laufende Verbreitung von IT-Geräten sah die Lage trotz der Auswirkungen der geplatzten Dotcom-Blase relativ gut aus. So stieg der Bedarf nach Lokalisierung stetig an, vergleichsweise waren 1998 noch über 61 % der Internetnutzer/innen US-amerikaner/innen, für 2002 wurde deren Anteil nur mehr auf 37 % geschätzt. Viel mehr als ein Problem angesehen wurde der Mangel an guten Fachleuten (vgl. Cyr & Lew 2003: 337-338; 341-342; Schäler 2003: 5). DePalma (2006: 17-18) stellt auch bis 2005 einen insgesamt

wachsenden Markt fest. Trotz geringerer Einbrüche in den USA als Folge einer schwierigen Wirtschaftslage gab es in vielen Sektoren gesteigerte oder zumindest gleich große Investitionen wie zuvor, was zeigt, dass die Vorteile von lokalisierten Produkten auf Fremdmärkten im Bewusstsein angekommen waren.

Outsourcing ist nach wie vor das bevorzugte Modell. DePalma spricht von 87 % im Jahr 2004. Irland als Zentrum der Lokalisierungsindustrie rückte jedoch etwas aus dem Fokus, da viele Hersteller sich für Länder mit niedrigeren Lohn- und Lohnnebenkosten entschieden, so wie zum Beispiel Indien, die tschechische Republik und Argentinien (vgl. DePalma 2006: 23).

Die ersten Zielmärkte für lokalisierte Produkte waren westeuropäische Länder und Japan. Die größten Märkte waren Frankreich, Deutschland und Japan, gefolgt von Brasilien, Italien, Spanien, Schweden, Norwegen und den Niederlanden (vgl. Esselink 2000: 8). In diesen Ländern gab es genug Kaufkraft um einen profitablen Absatzmarkt für elektronische Geräte und Software darzustellen. Außerdem wird in diesen Ländern, abgesehen von Japan, die lateinische Schrift verwendet, was Lokalisierung einfacher macht.

Schäler beanstandet im Jahr 2003 (5-7), dass die Lokalisierungsindustrie zu einseitig sei und führt folgende Punkte an: Da der Großteil der digitalen Produkte in den USA entsteht, laut Schäler sogar über 90 %, gibt es einen stetigen Fluss nur in eine Richtung, was zu wenig Diversität führt. Die vermeintliche Annahme, dass Lokalisierung zu einer stärkeren kulturellen Vernetzung und Verbreitung von Kulturgut führt, entspricht somit nicht der Realität. Dies liegt auch gar nicht im Blickfeld der Softwarehersteller, da es ihnen nur um die zusätzlichen Erträge durch die Erschließung neuer Märkte geht. Dabei geht es logischerweise nicht darum, möglichst viele Menschen zu erreichen, sondern nur jene, die sich den technischen Fortschritt auch leisten können. Auch das Internet, das eigentlich dazu führen könnte, Menschen besser mit Informationen zu versorgen, kritisiert Schäler. 80 % der 2001 verfügbaren Websites waren englischsprachig, die verbleibenden in anderen Sprachen der Nordhalbkugel; also Japanisch, Deutsch, Französisch, Spanisch oder Portugiesisch. Der Anteil der Menschheit, der mit diesen Sprachen erreicht werden kann, ist im Gesamten betrachtet nicht sehr hoch. Es wäre auch nicht überaus sinnvoll, Inhalte in „südliche“ Sprachen zu übersetzen, da der Zugang zum Internet damals noch eher ein Luxus als die Normalität war. Durch die Unerreichbarkeit des technischen Fortschritts für viele wird die Schere zwischen Arm und Reich noch vergrößert, was auch als „Digital Divide“ bezeichnet wird. Schlussendlich bemängelt Schäler auch die Verbreitung einer einheitlichen Konsumgesellschaft, die von den USA ausgeht. Produkte und Inhalte seien oberflächlich und gleichmachend und steuern somit gegen Diversität. Die Möglichkeiten, die Lokalisierung bietet, sollten eigentlich dafür genutzt werden, besseren Zugang zu Information und dem technischen Fortschritt für alle zu schaffen und die Verbreitung von vielen verschiedenen Kulturgütern zu fördern, um mehr Diversität zu erreichen.

Dies ist eine relativ negative Interpretation von Lokalisierung und vor allem der Lokalisierungsindustrie. Die Tatsachen, die genannt werden, spiegeln jedoch die Ausführungen

anderer Quellen wieder, wie zum Beispiel die Orientierung nach Westeuropa und Japan und die Auffassung von Lokalisierung als einzig wirtschaftliche Maßnahme. Schäler stellt mit seinen Anforderungen Lokalisierung auf eine höhere Ebene, also wie man dieses Konzept nutzen könnte, um die Welt zu verbessern.

Seit der Jahrtausendwende sind mittlerweile über zehn Jahre vergangen, was auch an der Lokalisierungsindustrie nicht spurlos vorübergegangen ist. Van Genabith geht etwas später, im Jahr 2009 (4-7), auf Veränderungen und neue Herausforderungen für die Lokalisierungsindustrie ein, auf die „nächste Generation“ der Lokalisierung. Seine Definition der Lokalisierung umschließt bereits viel mehr als die Software und Übersetzung und zeigt, wie sich nach eigentlich wenigen Jahren das Verständnis geändert hat:

„Localisation is the process of adapting digital content to culture, locale and linguistic environment. Localisation brings products and services to markets that are otherwise inaccessible. Because of this, localisation is a core multiplier and value-adding component of the global software, services, manufacturing and content distribution industry.” (van Genabith 2009: 4)

Dies ist ein recht ganzheitliches Verständnis von der Aufgabe der Lokalisierungsindustrie. Es zeigt auch, dass die technologische Landschaft sich verändert hat, anstatt von Softwareprodukten geht es nun allgemein um Inhalte. Esselink lässt dies bereits 2003 (6) anklingen, als er schreibt, dass die Grenzen zwischen den Bereichen Software- und Weblokalisierung verschwimmen (siehe Kapitel 3.1.3.). Auch van Genabith beschäftigt sich mit dem Digital Divide, sieht diesen aber als Herausforderung. Die ständig wachsende Menge an Inhalten und die große Anzahl der Sprachen übersteigen die Kapazitäten der Lokalisierungs- und Übersetzungsindustrie. Dies bedeutet nicht nur, dass der Unterschied zwischen Arm und Reich größer wird, gemessen am Zugang zu Information, sondern auch, dass viele Geschäftsmöglichkeiten nicht genutzt werden. Es gibt andererseits aber auch immer mehr Lösungen, mithilfe des technischen Fortschrittes diese Unterschiede zu überwinden. Kleine elektronische Geräte wie Smartphones erreichen in Schwellenländern eine viel größere Verbreitung als Desktop-Computer. Van Genabith plädiert auch für eine Weiterentwicklung und Lokalisierung von Sprachtechnologien, wie zum Beispiel Sprachsteuerung, um andere Barrieren zu überwinden. Im Bereich Locale geht er auch weiter. Vor allem bezogen auf Websites und dergleichen sollte das „ultimative“ Locale die Person selbst sein, das heißt die Informationen und die Präsentation sollte auf den/die einzelne/n User/in zugeschnitten sein (vgl. van Genabith 2009: 4-7). Auch Abel (2010: 39-42) spricht sich für die Personalisierung von Inhalten, so wie Marketingmaterialien und Soziale Netzwerke, aus. Die Antworten auf diese Herausforderungen sieht van Genabith vor allem im technischen Fortschritt, wie die vermehrte Einbindung der weiterentwickelten maschinellen Übersetzung, und alternative Methoden wie Crowdsourcing (vgl. van Genabith 2009: 5-10). Auf Crowdsourcing wird in Kapitel 4.4. näher eingegangen.

In der Tat hat die Lokalisierungsindustrie den „Rest der Welt“ entdeckt. Ob dies aus hehren Motiven geschieht, ist zu bezweifeln. 2012 zeichnet Sargent (49-50) ein recht positives Bild für Übersetzungs- und Lokalisierungsexpert/innen mit der Aussicht auf viele Auf-

träge. Das Internet spielt dabei eine große Rolle. Für Konsument/innen von Informationen über das Internet ändert sich die Sichtweise. Durch die Verwendung des Internets kommt man in Kontakt mit vielen Gütern und Dienstleistungen. Als Nutzer/innen von sozialen Netzwerken verstärkt sich die Vernetzung mit anderen Konsument/innen und man ist vielen Produkten und Marken viel stärker ausgesetzt. Dies bedeutet eine generelle Einbindung des Internets in das Konsumentenleben. Durch diese Verbreitung des Internetnetzes ist die Kaufkraft, die man online erreicht, zwischen 2009 und 2012 von 36,5 Billionen USD auf 44,6 Billionen USD gestiegen. Die verschiedenen Websites und Internetdienste erreichen Menschen optimal, wenn sie lokalisiert werden. Die Sprachenaufteilung der Internetnutzer/innen ist auch im Verändern begriffen, interessant ist hier jedoch die unterschiedliche Verteilung der Kaufkraft. Mit zwölf Sprachen erreicht man 80 % der Internetnutzer/innen. Neben den üblichen Sprachen finden sich hier auch Chinesisch, Indonesisch, Koreanisch, Russisch und Arabisch. Um jedoch 90 % des wirtschaftlichen Potenzials zu erreichen, sind dreizehn Sprachen notwendig. Aus dieser Liste fällt Indonesisch hinaus, dafür kommen Niederländisch und Schwedisch dazu. Der Unterschied ist teilweise enorm: mit Chinesisch können 22,7 % der Internetuser/innen erreicht werden, jedoch nur 6,1 % des wirtschaftlichen Potenzials, während der Prozentsatz der englischsprachigen User/innen annähernd gleich ist (21,6 %), diese jedoch über 36,6 % der online erreichbaren Kaufkraft verfügen.

Abgesehen vom Internetzugang ändert sich einiges bei den Zielmärkten für lokalisierte Produkte. Sargent (2012: 49) spricht von einer Krise der westlichen Mittelschicht, während die Mittelschicht in Afrika und Indien stark wächst und die betroffenen afrikanischen Länder sowie Indien als Märkte interessant werden. Auch Carter (2012: 25-29) erläutert die steigende Relevanz von Schwellenländern. Je besser die Wirtschaft und Kaufkraft eines Landes, desto mehr Potenzial gibt es auch für Sprach- und Lokalisierungsdienstleister. Brasilien, Russland, Indien und China stellen eine Gruppe von aufstrebenden Märkten da, die 2012 nach wie vor relevant sind, jedoch nicht mehr so stark wachsen wie zuvor. Dabei ist Indien, zum Beispiel, im Gegensatz zu Russland oder Brasilien eine besondere Herausforderung wegen der Vielfalt der Sprachen und Kulturen. Dies trifft auch auf andere, vor allem afrikanische, Länder zu, in denen viele Kulturen zusammenleben. Trotz der steigenden Anzahl an Lokalisierungsaufträgen in den klassischen Zielländern lässt sich doch die stetige Suche nach neuen Märkten beobachten. Muddyman (2012: 30-33) spricht hier die CIVETS-Länder an (Kolumbien, Indonesien, Vietnam, Ägypten, die Türkei und Südafrika). Diese ist wieder eine im Aufstreben begriffene Ländergruppe und auch hier gibt es spezielle Herausforderungen, die bei der Lokalisierungsarbeit nicht außer Acht gelassen werden dürfen. Die Religion ist hier, zumindest in Indonesien, Ägypten und der Türkei, ein nicht unwichtiger kultureller Faktor. In Südafrika leben wiederum, wie oben bereits erwähnt, viele verschiedene Kulturen in einem Land zusammen. Dies wirkt sich auch auf die Sprache aus; in Südafrika gibt es elf Amtssprachen und zahlreiche andere Sprachen, die gesprochen werden. Bahasa Indonesia oder Indonesisch weist hingegen ein recht komplexes Vokabular aufgrund von vielen Ein-

flüssen auf. Diese und viele andere Faktoren unterscheiden die genannten Länder von den klassischen Zielmärkten für lokalisierte Produkte.

Zwei Jahre später schreibt Bracken (2014: 18) über die weitere Entwicklung in Richtung Schwellenmärkte und die Herausforderungen und Probleme, die bei all diesen Ländern mehr oder weniger ähnlich sind. Er führt an, dass sich der Lokalisierungsprozess selbst seit Längerem nicht mehr grundlegend verändert hat. Im Prinzip ginge es nur mehr darum, die Sprache auszutauschen, da man glaube, die Grundstruktur auf alle Projekte anwenden könne.

Die momentan wichtigsten Gebiete, die es zu erschließen gilt, sind Osteuropa und Asien. Hierbei gibt es allerdings Einiges zu beachten, da man nicht mit denselben Bedingungen rechnen kann, die für Industrieländer gelten und somit auch die Prozesse überarbeiten muss. So werden viele der asiatischen Schriftsysteme von CAT-Tools nicht oder nicht optimal unterstützt. Dadurch kommt es zu Problemen bei der Segmentierung, mit dem Platz, Schriftgrößen und mit sonstigen Inkompatibilitäten. Bei einigen Schriften gibt es auch noch keine beziehungsweise nur mangelhafte Unicode-Unterstützung. Gemäß dem Outsourcing-Modell sind die Übersetzer/innen meistens Freelancer/innen aus den Zielländern. Hierbei gibt es wiederum das Problem, dass viele keine CAT-Tools benutzen, einfach deshalb, weil diese durch die Wechselkurse zwischen den Währungen zu teuer sind. Je kleiner die Sprachgruppe ist, desto weniger ausgebildete Übersetzer/innen gibt es auch. Außerdem variiert die Qualität dieser Ausbildungen oft drastisch. Dazu kommt, dass sich oft die Erarbeitung von Terminologie und Standards, zwei der wichtigsten Aspekte bei Lokalisierung, noch im Anfangsstadium befindet. Durch Schrift- und Grammatiksysteme, die sich sehr stark von den indogermanischen Systemen unterscheiden, können Messgrößen, so wie die noch akzeptierte Fehleranzahl pro tausend Wörter, nicht angewendet werden. Trotz der weltweiten Expansion des Internets gibt es noch gewaltige Unterschiede bezüglich Downloadgeschwindigkeit und Kosten. Dadurch wird die Vernetzung mit den vor Ort arbeitenden Freelancer/innen erschwert. In China, zum Beispiel, verlangsamt auch die internationale Firewall die Verbindungsgeschwindigkeit mit ausländischen Servern. Wie oben bereits angemerkt, spielt die Kultur eine große Rolle. Der kulturelle Unterschied zwischen den USA und westeuropäischen Ländern ist nicht so groß wie der Unterschied zu vielen Entwicklungsländern. Dies erfordert also mehr Aufklärungsarbeit und eine tiefere Auseinandersetzung mit ebendiesen Unterschieden. Es braucht auch andere Ansätze als bei der Arbeit mit Freelancer/innen in Ländern, wo die Lokalisierungsindustrie sich bereits etabliert hat und die Expert/innen auf mehrjährige Erfahrung zurückblicken können (vgl. Bracken 2014: 18-20). Genau dieses Problem, in Bezug auf die Geschäftsstrategie, spricht auch DePalma bereits 2006 (23) an. Für die Erschließung neuer Märkte wird weniger Marktforschung betrieben und das Geschäftsszenario, das in den USA funktioniert hat, eins zu eins übernommen.

Auf der Anbieterseite sind die USA und Europa nach wie vor stark. Irland hat mittlerweile an Bedeutung als ehemaliges Zentrum der Lokalisierungsindustrie verloren. Neue Gebiete sind der Norden Englands sowie London, München und Berlin in Deutschland und

Barcelona in Spanien. Aufschwung erfahren auch Polen, Rumänien, die baltischen Staaten und die ehemaligen Länder der Sowjetunion, begünstigt durch die niedrigeren Lebenshaltungskosten (vgl. Spacinsky 2014: 17).

In den wenigen Jahrzehnten, die Lokalisierung auf dem Buckel hat, hat sich also einiges getan. Einerseits veränderte sich das Verständnis von Lokalisierung, von der Übersetzung der Benutzeroberfläche hin zu einem gesamtheitlichen Konzept, das alle digitalen Inhalte betrifft, die, meistens noch immer aus dem Englischen, für alle möglichen Sprachen und Kulturen adaptiert werden. Diese Expansion geht natürlich einher mit der Entwicklung des Computers und des Internets. Vom Bürogerät ins Wohnzimmer und schließlich auf die Straße, vor allem im mobilen Bereich, der den Hauptgegenstand dieser Arbeit darstellt, gibt es rasante Fortschritte und viele neue Möglichkeiten. Andererseits hat sich der Fokus der Industrie stark ausgebreitet: von wenigen kaufkraftstarken Schlüsselmärkten hin zu allen Ländern, in denen der technische Fortschritt mit PCs, Smartphones und Tablets Einzug gefunden hat. Das Ziel von Schäler und van Genabith, den Digital Divide zu verringern, liegt nach wie vor nicht im Fokus der Industrie. Doch bedingt durch die Ausbreitung und Diversifizierung der Technologien und auch der Lokalisierungsbestrebungen gibt es auch für nicht priorisierte Zielmärkte immer mehr Möglichkeiten, an der globalen Kommunikationsgesellschaft teilzunehmen.

4.2 Warum lokalisieren?

Wie aus den Ausführungen des vorigen Kapitels ersichtlich, entschließen sich viele Unternehmen, ihre Produkte zu lokalisieren. Immer mehr werden auch neue Märkte angestrebt, es wird an Lösungen gearbeitet, die Software für alle möglichen Schriftsysteme kompatibel zu machen und auch tiefer unter die Oberfläche zu dringen und Anwendungen und Inhalte kulturell besser anzupassen. Dies geschieht nicht aus Jux und Tollerei, sondern um Profit zu erzielen. Im Endeffekt wird der Erfolg eines lokalisierten Produktes an der Rendite oder „Return on Investment“ gemessen, das heißt, am Verhältnis zwischen finanziellen Mehraufwand und Mehrertrag.

Rein finanziell gesehen lohnt sich die Investition im Allgemeinen. Bezogen auf US-amerikanische Hersteller beschreibt DePalma (2006: 15-16), dass die Erträge aus ausländischen Märkten 45 % der Gesamterträge ausmachen, bei 40 % der Assets. Dies bedeutet, dass überproportional viel Ertrag aus ausländischen Märkten erwirtschaftet wird. Insgesamt entstehen bei den 500 größten Unternehmen 20 bis 50 % der Geschäftstätigkeit im Ausland, was einen großen Anteil ausmacht. Da ausländische Märkte einen großen Faktor ausmachen, sollte auf diesen nicht leichtfertig gehandelt werden, sondern die Konkurrenzfähigkeit nicht unterschätzt werden, was ein guter Grund ist, die Lokalisierung der Produkte vor der Veröffentlichung und während des gesamten Produktzyklus sorgfältig zu planen und durchzuführen.

Die tatsächlichen Kosten betragen 2,5 % und weniger der Erträge aus den ausländischen Märkten. Dies bedeutet also, dass man für wenig Einsatz viel herausbekommt.

Die Berechnung der Lokalisierungskosten ist jedoch meistens nicht einfach. Der einzige Posten, der in der Buchhaltung direkt aufscheint, sind die Ausgaben für Lokalisierungsdienstleister. Viele Kosten sind auf vielen verschiedenen Ebenen verteilt und so nicht einfach auszumachen. Was Lokalisierung insgesamt kostet ist nicht bekannt, was dazu führt, dass die Rendite nicht berechnet werden kann (vgl. DePalma 2006: 19-21). Da nicht nur die Lokalisierung des originalen Produkts Geld kostet, sondern auch die Lokalisierung von Updates, des Supports, von Patches et cetera, sollten die Kosten in mehreren Modellen unterschiedlich berechnet werden. Die Kostenbasis ist das Produkt bei der Markteinführung. Darin inkludiert sind die Übersetzungskosten für die Software selbst, Hilfedatei und Dokumentation, sowie Produktinformation. Andere Kostenfaktoren sind, zum Beispiel, Marketingkosten, also die Übersetzung und Anpassung von Marketingmaterial, Websites et cetera. Die Ergebnisse dieser Berechnungen stellen die Kosten zu einem gewissen Zeitpunkt dar. Das zweite Modell baut auf der Kostenbasis auf und schließt die Kosten während des Produktzyklus ein, also die Übersetzung und Anpassung von Updates und Marketingmaterial. Schließlich sollte auch noch die Eigenkapitalrentabilität berechnet werden, das heißt, die Erträge in Relation zum eingesetzten Eigenkapital gesetzt werden. Dies stellt jedoch bereits eine höhere Ebene der Berechnung dar und erfordert unter Umständen mehr Informationen und Analysen als zur Verfügung stehen (vgl. DePalma 2006: 27-29).

„Return on Investment“ ist außerdem kein Konzept, dass sich rein auf die finanzielle Seite beschränkt. Bei DePalma lautet die Definition wie folgt:

„ROI is a multi-variable, time-variant calculation that will be made by many participants in a value chain. A comprehensive view of ROI should capture the broader range of economic benefits that companies expect from their international investments – including increases in sales and market share, goodwill from better localized branding, lower support costs due to in-language information, and increased customer service and loyalty.” (DePalma 2006: 15)

Dies bedeutet, dass nicht nur die einmaligen Erträge eine Rolle spielen, sondern die Ziele differenziert betrachtet werden. Auf lange Sicht sind, neben den Erträgen durch die tatsächlichen Verkäufe, folgende Ziele von Belang: die Vergrößerung des Marktanteils, die Beherrschung eines Segments und die bereits vorhandenen Marktanteile zu verteidigen, um nur einige zu nennen. Folgende Faktoren lassen sich nicht direkt in klingende Münze umrechnen, stellen jedoch auf lange Sicht auch wichtige Werte dar: stärkere Kundenbindung, die Loyalität der Kund/innen zur Marke zu stärken, die Support-Kosten zu verringern durch lokalisiertes Hilfe- und Dokumentationsmaterial, Ausfälle und Schadensfälle durch besseres, lokalisiertes Service und effektivere Kommunikation zu verringern, geringere administrative Kosten et cetera. Außerdem ist es immer von Vorteil, die Marke zu stärken, zum Beispiel durch die Sichtbarkeit und den Bekanntheitsgrad des Unternehmens und seiner Produkte, marktübergreifendes Markenmanagement und Anpassung der Marke an lokale Gegebenheiten so-

wie eine stärkere Bindung der Marke an die lokale Kultur. Diese letztgenannten Ziele lassen sich nicht durch Kapitalflüsse, sondern durch Umfragen, Fokusgruppen und andere Marktforschungsinstrumente überprüfen (vgl. DePalma 2006: 27).

Es wird trotz allem öfter die Frage gestellt, ob die Investition überhaupt notwendig ist, da ein Großteil der Konsument/innen von technischen Geräten und Software ohnehin zumindest ein bisschen Englisch spricht. Diese Sichtweise ist wohl auch verknüpft mit der traditionellen Ausrichtung auf westeuropäische Märkte, deren Konsument/innen generell der US-amerikanischen Konsumkultur und deren Produkte ausgesetzt sind. Es gibt viele Gründe, die dagegen sprechen, obwohl dieser Aspekt auch differenziert betrachtet werden sollte.

In diesem Zusammenhang ist eine Studie aus dem Jahr 2005 interessant (vgl. Husbands 2005: 3-7). Im Zuge dieser Studie wurden Konsument/innen aus mehreren Ländern bezüglich ihrer Akzeptanz von nichtlokalisierter Produktdokumentation und Marketingkommunikation befragt. 81 % der Befragten gaben an, dass sie erwarteten, dass Unternehmen, die Produkte und Dienstleistung in ihrem Land verkaufen wollen, in ihrer Muttersprache mit ihnen kommunizieren. 73 % der Befragten würden eher ein Produkt kaufen, das über lokalisierte Produktinformation verfügt, als ein gleichwertiges nicht-lokalisiertes Produkt zu wählen. Über 70 % würden kein Produkt kaufen, dessen Verpackung sie nicht verstehen würden. 61 % würden kein Produkt kaufen, dessen Produktinformation fehlerhafte Übersetzungen enthält und 71 % würden eher ein Produkt derselben Marke kaufen, wenn der Support in ihrer Muttersprache verfügbar sei. Dabei gibt es Sprachgruppen, die Kommunikation in der eigenen Sprache eher erwarten oder für wichtiger halten. Hierbei war es für Konsument/innen aus den portugiesischen, spanischen und deutschen Sprachräumen am wichtigsten, dass die Produktinformation und Kommunikation in ihrer Sprache verfügbar seien, am wenigsten wichtig war es den Befragten aus den skandinavischen Ländern, dem Baltikum und den Niederlanden. Außerdem war den Befragten aus dem asiatisch-pazifischen Raum wichtig, dass die Übersetzungen keine Fehler, sowie keine unangebrachten Symbole, Bilder und Sprache enthalten. Wurde diese Umfrage auch nicht spezifisch zu lokalisierter Software durchgeführt, zeigt sie doch, dass die Sprache und kulturelle Adaption zumindest beim Produktumfang eine große Rolle spielt.

Ob nun Lokalisierung des Produktes samt dem Produktumfang unbedingt notwendig ist, kommt auf verschiedene Gegebenheiten an, zum Beispiel das Bildungsniveau des Ziellandes, die rechtlichen Vorschriften und die Gesamtbetriebskosten. Bei DePalma wird die Grenze zwischen freiwilliger und notwendiger Lokalisierung, nach dem „Common Sense Advisory“, „Tipping Point“ genannt. Dieser variiert je nach Land, Produkt, Konsument/in und Vertriebsweg (vgl. DePalma 2006: 32). Reflektiert wird dies auch in der oben genannten Studie, wonach es auf die Art des Produktes ankommt, ob die Information in der Muttersprache der Konsument/innen verfügbar sein muss. Vor allem bei Bank- und Finanzdienstleistungen ist dies sehr wichtig, gefolgt von pharmazeutischen und kosmetischen Produkten und Verbraucherelektronik (vgl. Husbands 2005: 6).

DePalma (2006: 32-33) weist darauf hin, dass sich der „Tipping Point“ sehr schnell verschieben kann. Er unterscheidet auch zwischen verschiedenen Gruppen, für die Lokalisierung mehr oder weniger wichtig ist. Fachleute, vor allem mehrsprachige, im technischen und IT-Bereich, sowie Führungskräfte und Manager/innen haben oftmals kein Problem, mit englischsprachigen Produkten zu arbeiten, während für Endverbraucher/innen und Angestellte ohne Englischkenntnisse lokalisierte Produkte wichtig sind. Auch Scattergood (2003: 14), der sich mit Lokalisierung von Dokumentation beschäftigt, unterscheidet hier zwischen EDV-Expert/innen, fortgeschrittenen Nutzer/innen und User/innen ohne professionelle Unterstützung. Für die erste Gruppe ist die Lokalisierung der Dokumentation nicht notwendig, da diese sich ohnehin mit den Anwendungen auskennen (müssen) und denen demnach die verwendete Sprache geläufig ist. Bei der zweiten Gruppe sollten gegebenenfalls Teile lokalisiert werden und die Dokumentation, die für die dritte Gruppe relevant ist, also einfache Anleitungen für die Verwendung der Anwendung, ist ganz zu lokalisieren.

Im Endeffekt sollte man auch die Gegenfrage stellen: Was sind die Folgen der Entscheidung, ein Produkt nicht zu lokalisieren und trotzdem international zu vertreiben? Für Endverbraucher/innen ist es einfach nicht praktikabel, sich beim Gebrauch eines Produktes auf eine Fremdsprache konzentrieren zu müssen. Selbst, wenn die Konsument/innen Englisch sprechen, gibt es eine Reihe von Kostenfaktoren, die schließlich bei diesen hängenbleiben. Ein Beispiel sind die Kosten des Fachpersonals, das Installation und Wartung übernimmt. Diese sind in vielen Ländern höher, wenn das zu installierende Produkt nicht in der Landessprache verfügbar ist. Allgemein wird dieser Mehraufwand, sei er finanzieller oder auch praktischer Natur, dem/der Verbraucher/in zugeschoben, was insgesamt dazu führt, dass diese/r sich eher für ein lokalisiertes Produkt entscheiden wird (vgl. DePalma 2006: 33-34).

Die Frage, ob Lokalisierung eine berechtigte Investition darstellt, lässt sich bei internationaler Vermarktung eines Produktes also durchaus mit Ja beantworten. Bei der Entscheidung und Planung sollte jedoch eine umfassende Analyse der finanziellen und marktrelevanten Faktoren durchgeführt werden, um harte Fakten bei der Hand zu haben. Es sollte außerdem nicht außer Acht gelassen werden, dass der Erfolg sich nicht nur durch verkaufte Stückzahlen messen lässt, sondern andere Faktoren, wie Markterschließung und Kundenbindung, auf lange Frist gesehen von großer Bedeutung sind.

4.3 Kostenmanagement

Trotz der allgemein guten Rentabilität ist es klar, dass es sich kein Unternehmen leisten kann, Geld zum Fenster hinauszuwerfen. Es ist immer ein Ziel, kosteneffizienter zu arbeiten. In diesem Kapitel wird zuerst die übliche Kostenzusammensetzung innerhalb eines Lokalisierungsprojekts erläutert, gefolgt von den Kosten für Übersetzungen und schließlich wird auf

Wege eingegangen, wie möglichst effizient gearbeitet werden kann und unnötige Kosten vermieden werden.

4.3.1 Kostenaufstellung

Welche Arbeitsergebnisse werden geliefert und welche Arbeitsschritte sind dafür nötig? Wie bereits erwähnt, sind die Hauptteile bei Lokalisierung die Software selbst, die Hilfedatei und die Dokumentation. Zusätzlich zur Übersetzung und Anpassung diverser Merkmale müssen diese normalerweise in die ursprüngliche Form zurückkompiliert beziehungsweise – formatiert und getestet werden. Es werden die kompilierten beziehungsweise formatierten Dateien sowie die Ressourcendateien geliefert. Ein wichtiger Punkt, der nicht fehlen darf, sind Translation Memorys des Projekts sowie alle Glossare (vgl. Esselink 2000: 435-436).

Um die fertigen Dateien zu erstellen, sind die im Folgenden beschriebenen Arbeitsschritte nötig. Der Preis wird üblicherweise entweder nach Stunden berechnet oder nach der Anzahl der Einheiten, etwa Wörter, Einträge oder Seiten. Der erste Punkt ist Recherche, um die Terminologie zu erarbeiten. Diese Tätigkeit wird pro Stunde oder pro Eintrag abgerechnet. Der zweite Punkt ist die Übersetzung. Hierbei wird unterschieden zwischen Software, deren Übersetzung üblicherweise 20 bis 30 % mehr kostet, und der Hilfedatei sowie der Dokumentation. Ist eine Übersetzung wegen des Dateiformates (z.B. XML oder HTML) recht aufwendig, so wird der Preis auch höher. Die Kosten für die Übersetzung werden nach Wortanzahl berechnet. Zusätzlich gibt es oft noch Beispieldokumente, deren Übersetzung entweder nach Wortanzahl oder nach Stunden berechnet wird (vgl. Esselink 2000: 434). Je nach Angebot inkludiert dies nur die Übersetzung oder auch bereits Bearbeitung und Korrektur (vgl. Sweezey & Visuri 2013: 51).

Terminologierecherche und Übersetzung sind Aufgaben für Übersetzer/innen und Sprachexpert/innen, während die Softwarebearbeitung von Softwaretechniker/innen vorgenommen wird. Dazu zählen die Anpassung der Dialogfelder, die Überprüfung der geänderten Tastenkombinationen und das Kompilieren der Software. Diese Arbeit wird normalerweise nach Stunden abgerechnet, es kann aber auch vorkommen, dass der Preis nach der Anzahl der angepassten Dialogfelder berechnet wird.

Neben der sprachlichen Überprüfung der Übersetzungen muss auch die Funktion der lokalisierten Software getestet werden. Die Abrechnung erfolgt nach Arbeitsstunden, wobei die Abrechnung der Überprüfung der Hilfedatei auch nach der Anzahl der Seiten berechnet werden kann.

Um die Dokumentation und Begleitmaterial fertigzustellen, ist oft die Arbeit von Grafikdesigner/innen notwendig, um die übersetzten Texte in das Layout einzufügen. Hier wird nach Seiten oder Stunden abgerechnet. Auch Bildmaterial der Software, der Hilfe und der

Dokumentation muss gegebenenfalls verändert werden, so zum Beispiel Bildbeschreibungstexte. Diese Arbeiten werden nach Bildern oder nach Stunden abgerechnet.

Wenn Material gedruckt beigelegt wird, muss dieses vorbereitet und gegebenenfalls zuvor konvertiert werden. Bei der Abrechnung wird nach Stunden oder Seiten vorgegangen.

Schließlich bleiben noch der Posten der Projektvorbereitung, der üblicherweise nach Stunden abgerechnet wird, und der Posten Projektmanagement. Für das Projektmanagement wird üblicherweise ein Prozentsatz von 10 bis 15 % auf die Kosten aufgeschlagen. Manchmal werden für Kommunikation noch 2 bis 5 % addiert. Es kann sein, dass die Kosten für das Projektmanagement in die Kosten pro Einheit für die anderen Posten miteingerechnet werden (vgl. Esselink 2000: 434-435).

4.3.2 Übersetzungskosten

Aus den Ausführungen der vorigen Kapitel geht hervor, dass Übersetzung nur ein Baustein im Lokalisierungsunterfangen ist, was auch in der Aufstellung der Arbeitsschritte widergespiegelt wird. Wie viel macht nun die Übersetzung aus? Als Richtwert gibt Scattergood (2003: 13) an, dass 40 % der Kosten für externe Dienste ausgegeben werden, während 60 % der Lokalisierungskosten im Unternehmen selbst auftreten. Von den 40 %, die ausgegeben werden, werden 70 % für die Übersetzung ausgegeben, während 30 % für Management und Softwaretechnik anfallen. Dies bedeutet, dass von den gesamten Lokalisierungskosten nur 28 % für die tatsächliche Übersetzung der Texte ausgegeben werden.

Es wurde bereits erwähnt, wie der Preis für die Übersetzung der Texte berechnet wird. Hier ist noch hinzuzufügen, dass der Preis für die Übersetzungen im Bereich Lokalisierung üblicherweise nach der Anzahl der Wörter berechnet wird, auch wenn in vielen Ländern Übersetzungen anders, zum Beispiel nach Normzeilen, berechnet werden (vgl. Esselink 2000: 435). Wie viel ein Wort nun kostet, unterscheidet sich sehr stark, abhängig von der Sprache, der Region, der Anzahl an verfügbaren Übersetzer/innen, ob Freelancer/innen oder Lokalisierungsdienstleister beauftragt werden et cetera. Wird, wie es die Regel ist, ein Translation Memory-System verwendet, so kosten die Wörter und Segmente, für die es bereits Matches oder Fuzzy Matches gibt, weniger. Im Beispiel von Sweezey & Visuri (2013: 52) wird für 100 %-Matches gar nichts verrechnet und für Matches zwischen 95 und 99 % weniger als ein Viertel als für Matches unter 75 % beziehungsweise für neue Wörter.

Kann nun bei der Übersetzung gespart werden? Kostenreduktionen können durch die Einsetzung von CAT-Tools und Content Management Systemen erreicht werden, die Expert/innen warnen jedoch davor, die Übersetzungskosten um jeden Preis zu drücken und das billigste verfügbare Angebot anzunehmen (vgl. DiFranco 2006: 47; Sweezey & Visuri 2013: 51). Auf der einen Seite scheint es klar, dass Lokalisierungs- und Sprachexpert/innen für sich

argumentieren, doch dies tun sie mit guten Argumenten, die nicht einfach so wegzuwischen sind.

Zum einen ist die Marge für Übersetzungen im Bereich Lokalisierung ohnehin niedrig; sie liegt bei 5 bis 10 %. Dies bedeutet, dass das Drücken des Preises notwendigerweise zu Einbußen bei der Qualität führt, da Übersetzer/innen auf Qualitätskontrollen verzichten müssen, wenn sie auf einen angemessenen Stundensatz kommen wollen (vgl. Scattergood 2003: 13).

Makosiej (2013: 22-23) argumentiert für die Auftragsvergabe an einen Lokalisierungsdienstleister anstatt der direkten Vergabe an Freelancer/innen. Softwareunternehmen tendieren dazu, die Übersetzungsarbeit direkt an Freelancer/innen outzusourcen, da dies auf den ersten Blick günstiger ist. Dies ist laut van der Meer (2003: 9-11) auch verständlich, angesichts der Ebenen, die die Übersetzer/innen vom ursprünglichen Auftraggeber trennen. Die Freelancer/innen bekommen ihre Aufträge von Agenturen, die ihre Aufträge wiederum von Sprachdienstleistern bekommen. Im Extremfall verkaufen diese die Leistungen weiter an Sprachdienstleister, die mehrere Sprachen anbieten. Dies bedeutet im Endeffekt, dass die ursprünglichen Auftraggeber dreimal so viel bezahlen, wie die Übersetzer/innen einnehmen. Die zusätzlichen Kosten entstehen durch verschiedene Managementaufgaben und Qualitätssicherung. Genau hier liegt laut Makosiej (2013: 22-23) der Mehrwert. Arbeiten Freelancer/innen direkt für die Endkunden, so fallen wichtige Kontrollmechanismen weg, die den Output auf verschieden Weisen kontrollieren. Zum einen müssen freiberufliche Übersetzer/innen flexibel sein. Um die Kunden nicht zu verlieren müssen sie die Aufträge dann bearbeiten, wenn sie sie bekommen. Wenn ein/e Freelancer/in mehrere Aufträge innerhalb einer kurzen Zeitspanne bekommt, bleibt keine Zeit für Qualitätskontrolle, da die Deadline für den nächsten Auftrag heranrückt. Wird die Auftragsvergabe durch einen Dienstleister gesteuert, so kann eher auf ein regelmäßiges Arbeitspensum geachtet werden. Zum anderen sorgen Sprachdienstleister für Qualitätssicherung und Qualitätskontrolle, welche außerdem von anderen Übersetzer/innen übernommen werden. Dies ist ein wichtiger Punkt, da die Kontrolle der eigenen Arbeit meistens nicht zu gleich guten Ergebnissen führt. Nicht vergessen werden sollte außerdem die Organisations- und Koordinationsarbeit, die geleistet wird. Vergibt man Aufträge an einen Sprachdienstleister, bekommt man ein Gesamtpaket geliefert. Dieses ist zwar mit höheren Kosten verbunden, allerdings auch mit mehr Leistungen. Es ist klar, dass Makosiej als Geschäftsführer eines Sprachdienstleistungsunternehmens für diese Dienste spricht, seine Argumentation ist jedoch in sich schlüssig. Sweezey & Visuri (2013: 52-53) sprechen außerdem von den Vorteilen von guter Zusammenarbeit mit Sprachdienstleistungsunternehmen. Herrscht eine gute Basis, so bekommen Kunden Beratung und Wissen quasi umsonst, da es für die Dienstleister von Vorteil ist, die Kunden langfristig zu binden. Auch Harcz (2011: 25), Geschäftsführer einer Übersetzungsagentur, führt an, dass es bei der Preisberechnung nicht nur auf die harten Fakten ankommt, sondern auch auf Faktoren wie gute Zusammenarbeit und verlässliche Begleichung der Rechnungen.

4.3.3 Effizienzsteigerung

Wie nun die Wahl ausfällt, Freelancer/innen oder Sprachdienstleister, es gibt viele andere Wege, die Effizienz zu steigern und die Investition möglichst wirkungsvoll zu nutzen. Im Prinzip zielen alle Verbesserungen in den Bereichen Internationalisierung, Technologie, Projektmanagement und Qualitätssicherung darauf ab, Endprodukte von höherer Qualität mit weniger Zeitaufwand herzustellen. Diese Aspekte werden alle in den entsprechenden Kapiteln erläutert. Deshalb sollen hier lediglich die Fäden zusammengeknüpft werden, um vorab einen Überblick zu schaffen.

DiFranco (2006: 47-48) argumentiert, dass die Kostenkontrolle grundsätzlich nicht bei den einzelnen Ausgaben für die benötigten Dienstleistungen vorgenommen werden sollte, also am Ende des Prozesses, sondern auf höherer Ebene beginnen muss, das heißt, die Planung beginnt gleich nach der Entscheidung, zu lokalisieren. Sie spricht dabei von drei Gruppen: Tools, Projektmanagement und Globalisierung.

Die Automatisierung verschiedener Prozesse durch Softwareanwendungen ist ein wichtiger Faktor in der Lokalisierungsindustrie. Die Auswahl und die richtige Verwendung von CAT-Tools ist ein Schritt, auf den u.a. DiFranco (2006: 49-54), Scattergood (2003: 13-15) und Lin & Gurevich (2010: 24-27) eingehen. Auf die Funktionsweise von CAT-Tools wird in Kapitel 6 genauer eingegangen, hier werden Kriterien besprochen, die bereits bei der Planung eines Lokalisierungsprojektes beachtet werden sollten, um die gewählten Tools möglichst wirksam einsetzen zu können und Problemen vorzubeugen.

Der erste Faktor ist die Einsetzbarkeit der Tools. Es sollte sichergestellt werden, dass das verwendete Programm für alle Projektgrößen einsetzbar ist, sowie alle gewünschten Sprachen unterstützt. Der zweite Faktor ist die Integration in den Workflow, bezüglich der Angaben, die gespeichert werden und die Verwendbarkeit durch verschiedene Beteiligte (vgl. DiFranco 2006: 50). Auch Scattergood (2003: 14) geht auf die Wichtigkeit der Wiederverwendung von Translation Memorys ein. Durch Kontextinformationen werden die gespeicherten Segmentpaare besser wiederverwertbar, was zu einer Effizienzsteigerung und besserer Kohärenz bei der Übersetzung führt. Aus dem Lokalisierungsprozess ist dies heute nicht mehr wegzudenken.

Die Wiederverwertbarkeit von Translation Memory-Datenbanken wird beeinträchtigt, wenn die Tools nicht kompatibel sind. Deshalb sollten diejenigen Tools verwendet werden, die der Kunde auch nutzt (vgl. DiFranco 2006: 50). Auf der anderen Seite kann Kompatibilität durch die Nutzung von offenen Standards, wie zum Beispiel XLIFF, erreicht werden, da diese von allen Systemen importiert werden können (vgl. Scattergood 2003: 15). Zur Entwicklung und Weiterentwicklung der Standards und deren Bedeutung, siehe Kapitel 6.

Da die Anforderungen an CAT-Tools sich je nach Kunde und Projekt unterscheiden, sollten diese personalisierbar sein. Dadurch kann auch auf kundeneigene CAT-Tools verzichtet werden, die nicht unbedingt mit anderen Tools und Formaten kompatibel sind.

Praktische Überlegungen, wie die Verfügbarkeit von technischem Support und der Leichtigkeit, den Umgang mit dem Tool zu erlernen, sollten auch in die Entscheidung einfließen. Ist eine Anwendung sehr komplex, dauert die Einschulung von neuen Projektbeteiligten unter Umständen zu lang um effektiv zu sein.

Schließlich spielen noch technische Faktoren eine Rolle, wie die Unterstützung von Schriftsystemen, die Möglichkeit der Versionskontrolle und die Einbindung eines Terminologie-Tools (vgl. DiFranco 2006: 50-51).

Projektmanagement umfasst alle koordinierenden und organisatorischen Tätigkeiten vor, während und nach dem Projekt. Wegen der Vielzahl an Arbeitsschritten und Beteiligten ist diese Arbeit eine der grundlegendsten im ganzen Prozess, da sie das gesamte Projekt zusammenhält. Es müssen alle Beteiligten koordiniert werden, Arbeitsschritte geplant und überprüft und vor allem die Vernetzung der Expert/innen und die Bereitstellung des Materials zur richtigen Zeit stellen eine Herausforderung dar. DiFranco (2006: 54-59) stellt eine Liste mit den wichtigsten Arbeitsschritten auf. Vor dem Projekt werden die Ressourcen analysiert und ein Zeitplan für alle Beteiligten erarbeitet. Während des Projektes wird die Einhaltung des Zeitplans ständig überprüft, die eingereichten Materialien werden für die nächsten Arbeitsschritte weitergegeben, die Kommunikation zwischen den Beteiligten und deren Unterstützung muss koordiniert werden et cetera. Nach Abschluss des Projekts bleibt noch die Analyse desselben sowie die Überprüfung und Archivierung von Ressourcen, die für weitere Projekte nützlich sind, zum Beispiel Translation Memorys, Terminologiedatenbanken et cetera.

Lin & Gurevich (2010: 24-27) stellen die Optimierung des Lokalisierungsprozesses an einer Fallstudie dar. Anstatt bei einem nicht optimal verlaufenden Projekt einfach das Budget zu erhöhen, schildern sie, wie die Prozesse analysiert wurden und anschließend nicht optimale Praktiken verändert wurden. So wurde, zum Beispiel, die Personalstruktur des Testteams verändert, abgestuft nach Qualifikation. Für diejenigen Arbeitsschritte, die keine lange Einarbeitungszeit verlangen, wurden Techniker/innen aus anderen Abteilungen geholt, anstatt mehr Expert/innen einzustellen oder die Arbeit outzusourcen. Dies hätte einen bedeutenden finanziellen Mehraufwand dargestellt, der in diesem Falle jedoch gar nicht nötig war. Außerdem wurde die Struktur des Projektes verändert, das heißt, die Reihenfolge der Teilschritten optimiert, sodass voneinander unabhängige Schritte gleichzeitig verlaufen konnten und somit die benötigte Zeit verringert wurde.

Die Optimierung der verschiedenen Arbeitsschritte, die Kommunikation und Vernetzung der Expert/innen sowie deren Einsatz sind Aspekte, die meistens eher zu einer besseren Ressourcennutzung und Erhöhung der Qualität führen, als die Kosten der einzelnen Arbeitsschritte zu drücken. In Kapitel 5 wird noch näher auf Projektmanagement und Projektablauf eingegangen.

Ein grundlegender Aspekt, dem Kapitel 3.2. gewidmet ist, ist Internationalisierung. DiFranco(2006: 59-63) spricht von „Globalisierung von Software“, beschreibt jedoch jenes Konzept, das in dieser Arbeit als Internationalisierung bezeichnet wird. Auf Internationalisierung soll hier nicht weiter eingegangen werden, sondern lediglich deren Bedeutung für die Effizienzsteigerung betont werden.

4.4 Alternative Ansätze

Es gibt neben kommerzieller Software und Lokalisierung auch viele andere Projekte, deren Ziel nicht Profitschaffung ist. Seit dem Verkauf der ersten Kleincomputer haben Enthusiast/innen begonnen, Software selbst zu schreiben, sich auszutauschen, andere Projekte zu verbessern und die Programme auch gratis weiterzugeben. Für die Lokalisierung dieser Open-Source-Programme werden für gewöhnlich keine Lokalisierungsdienstleister beauftragt, sondern die Usercommunitys übersetzen die Bestandteile der Benutzeroberfläche, um die Software für User/innen in ihren eigenen Ländern besser nutzbar zu machen. Im folgenden Kapitel wird zuerst die Entstehung der freien Software besprochen, danach das Phänomen Crowdsourcing näher beleuchtet.

4.4.1 Freie Software

In den Sechzigerjahren war es unter Computerexpert/innen üblich, Software zu teilen und auch selbst weiterzuentwickeln. Es gab für jedes System eigene Software und Programmiersprachen und deshalb war der Austausch und somit die Möglichkeit, mehr aus den Geräten herauszuholen, für alle gewinnbringend. Erst in den Siebzigerjahren begann AT&T mit Unix, proprietäre Software zu entwickeln, die Rechte zu sichern und die Programmierer/innen an sich zu binden und Geheimhaltungsvereinbarungen zu unterwerfen. Dadurch konnte man Programmlizenzen verkaufen, wodurch aber die Codes nicht mehr weitergegeben werden konnten und die Zusammenarbeit der Community unterbunden wurde. Das Modell der kostenpflichtigen, geschützten Software setzte sich mit dem Aufschwung des Desktop-PCs durch und ist auch jetzt das gängigste, das heißt, Programme werden gekauft und benutzt (vgl. Mestre 2009).

Abseits dieser Software-Industrie gibt es aber auch andere Strömungen. Richard Stallman ist einer der Begründer/innen der freien Software-Bewegung. Anstatt einen Posten als Programmierer anzunehmen, gründete er 1983 das Projekt GNU (GNU is not Unix) und 1985 die „Free Software Foundation“. Deren Ziel ist es, die Verbreitung und Weiterentwicklung von Software zu fördern, ohne kommerzielle Interessen in den Vordergrund zu rücken. Die Grundlage ist die Philosophie, dass Software „frei“ sein sollte, dies bedeutet, dass nicht

ein Unternehmen ein Programm von den angestellten Programmierer/innen entwickeln lässt und dann Lizenzen verkauft, sondern dass alle User/innen die Freiheit haben, Software zu benutzen, zu kopieren, zu verbreiten, zu untersuchen, zu verändern und somit zu verbessern. Dies steht im Gegensatz zu proprietärer Software, deren Quellcodes nicht weitergegeben werden und somit nicht verändert werden können. Bei freier Software wird auch der Verkauf nicht ausgeschlossen, obwohl viele Programme gratis erhältlich sind. Die Freiheiten der User/innen werden durch „Copyleft“-Lizenzen geschützt, deren bekannteste die „GNU General Public License“ ist. Eines der bekanntesten Produkte ist das Betriebssystem GNU/Linux, das weltweit von vielen User/innen und vor allem Organisationen verwendet wird (vgl. Free Software Foundation 2014).

Einen ähnlichen Ansatz vertritt die „Open Source Initiative“, die 1998 gegründet wurde. Die beiden Strömungen trennen sich nicht durch eine klare Linie, vielmehr gibt es viele Überschneidungen und im Prinzip sind es dieselben Ziele, die verfolgt werden: Software zu verbreiten und für Veränderungen und Verbesserungen durch die Community zu öffnen. Der grundlegende Unterschied ist die Philosophie hinter den Projekten. Beim der freien Software-Bewegung stehen die Freiheiten im Vordergrund, die Verbreitung und Offenlegung von Software ist ein ethisches und politisches Unterfangen, bei Open Source geht es vielmehr um praktische Gründe: Um das Potenzial der Community zu nutzen, muss der Quellcode einsehbar sein und auch die Veränderung rechtlich möglich sein (vgl. Open Source Initiative 2014).

Raymond (2002) vergleicht die Gegensätzlichkeit der Software-Industrie und Open Source-Software mit einer Kathedrale beziehungsweise einem Basar. Auf der einen Seite arbeiten wenige Expert/innen isoliert an einem Programm, im Rahmen eines gut geplanten Projektes. Auf der anderen Seite tragen viele Einzelpersonen ihren Teil bei und durch regen Austausch und Zusammenarbeit, sowie durch viele verschiedene Ansätze entsteht das Programm.

4.4.2 Crowdsourcing

Hierbei handelt es sich um ein Phänomen, das erstmals von Howe (2006) benannt wurde. Analog zu Outsourcing, der Weitervergabe von Teilaufgaben an unternehmensexterne Anbieter, prägt er den Begriff Crowdsourcing, der Vergabe von Teilaufgaben an viele Einzelpersonen, die durch ein Netzwerk – das Internet – verbunden sind, für kein oder minimales Entgelt. Howe erwähnt auch die Open Source-Bewegung, die bewiesen hat, dass eine Vielzahl von passionierten Hobby-Programmierer/innen insgesamt genauso gute Anwendungen entwickeln kann wie ein exklusives Expertenteam eines hochrangigen Softwareunternehmens. Durch die Vernetzung durch das Internet kann die Gesamtheit der kleinen, oftmals kostenlosen, Beiträge für manche Zwecke sehr wertvoll sein.

Im Bereich Übersetzung und Lokalisierung gibt es auch großes Interesse am Crowdsourcing-Modell, einerseits für Projekte, um den Digital Divide zu verringern, andererseits auch, um Kosten zu sparen. Crowdsourcing ist jedoch kein Modell, das für alle Zwecke eingesetzt werden kann und dadurch herkömmliche Übersetzungsarbeit verdrängt. Kelly (2009: 62-63) stellt fest, dass es auf die Beschaffenheit des Projekts ankommt, ob Crowdsourcing sinnvoll ist oder nicht. So braucht es zuallererst eine Gruppe von Menschen, die bereit sind, daran teilzunehmen und die auch über die notwendigen Kenntnisse verfügt. Es muss auch die Infrastruktur geschaffen werden um die einzelnen Beiträge verwenden und kontrollieren zu können, wofür Investitionen nötig sind. Schlussendlich kommt es auch auf die Texte an, die übersetzt werden. Handelt es sich um Inhalte, die eine kurze Lebensdauer haben, kann es sein, dass der herkömmliche Übersetzungsprozess ganz einfach zu lange dauert und crowdgesourcete Übersetzung durch die Community effektiver ist, während es auf der anderen Seite Texte gibt, die genau überprüft und korrigiert werden müssen. Bei sensiblen Texten, die rechtlich bindend sind, oder Texten, die nicht an die Öffentlichkeit gelangen dürfen, macht es keinen Sinn, sie an die „Crowd“ weiterzugeben und darauf zu vertrauen, dass die Kontrollmechanismen funktionieren.

Grundsätzlich kann Crowdsourcing nur dann funktionieren, wenn die Technologie zur Verfügung gestellt wird und die Menschen motiviert sind, mitzumachen. Dies kann erreicht werden durch finanzielle Anreize, die meistens jedoch eher gering sind, oder durch die Freude an der Arbeit. Hierfür ist oft der „Community Spirit“ wichtig und sonstige Anreize, wie Punktesysteme, Auszeichnungen, oder aber das Recht, ein lokalisiertes Programm zu nutzen (vgl. Howe 2006; Exton et al 2009: 84; Kelly 2009: 63; Thicke 2013^b: 26-27).

Bei Exton et al (2009: 81-89; 2010: 46-62) geht es um die Möglichkeit, die Hürde zu überwinden, die Menschen in nicht kaufkräftigen Ländern vom Zugang zu Technologie trennt. Es steht hier Open Source beziehungsweise freie Software im Vordergrund. Die Kombination von freier Software und Crowdsourcing macht es möglich, lokalisierte Versionen von Software zu entwickeln, die es sonst für diese Locales nicht geben würde, da es sich finanziell nicht rentiert. Es wird eine Softwarearchitektur besprochen, die es freiwilligen Übersetzer/innen möglich macht, Strings et cetera im Programm selbst zu übersetzen. Diese Texte werden auf einen Server hochgeladen und werden so an andere User/innen weitergegeben. Die Qualitätskontrolle erfolgt über Rankingsysteme und die Möglichkeit, Alternativvorschläge zu geben.

Ähnliche Lösungen werden bereits verwendet. Bei Mozilla wird ein Open Source-Lokalisierungstool, Pontoon, eingesetzt. Dieses wird verwendet für die Lokalisierung der Firefox OS Web App, kann aber auch für die Lokalisierung von Websites eingesetzt werden. Es unterstützt Translation Memorys und maschinelle Übersetzung (vgl. Beatty & Małolepszy 2013: 28-32). Für diesen Bereich haben Wasala et al (2012: 15-18) auch ein Model entwickelt, wie über eine Web App Inhalte übersetzt werden können. Auch hier würden Translati-

on Memorys verwendet werden, um die Segmente zu speichern. Ein Problem ist jedoch, dass nur die Texte übersetzt werden können, Bilder, Video- und Audiomaterial aber nicht.

Es gibt mittlerweile vielerlei Implementierungen von Crowdsourcing, von Websites, wie Facebook (vgl. Thicke 2013^b: 24-27), Open Source-Programme wie OpenOffice (vgl. OpenOffice 2014) und auch kommerzielle Anbieter, die Crowdsourcing-Übersetzungen anbieten (z. B. Get Localization 2014). Insgesamt ist Crowdsourcing nicht die Alternative für herkömmliche Übersetzungs- und Lokalisierungsdienste, kann bei gewissen Projekten jedoch sehr gut funktionieren. Vor allem in Kombination mit freier Software ist Crowdsourcing eine gute Möglichkeit den Digital Divide zu verringern.

5 Das Lokalisierungsprojekt

In den vorigen Kapiteln wurde Lokalisierung als Gesamtkonzept besprochen und die finanzielle Seite betrachtet. Im folgenden Kapitel richtet sich der Blick näher auf die Arbeitsschritte und Bestandteile, die im Rahmen eines Lokalisierungsprojektes vorkommen. Natürlich unterscheiden sich in der Realität Projekte teils sehr stark voneinander, deshalb wird hier kein Anspruch auf Allgemeingültigkeit gestellt, jedoch sind es Themen, die in der Literatur oft unter die Lupe genommen werden und die deshalb Eingang in diese Arbeit finden. Der erste Teil beschäftigt sich mit den Besonderheiten des Projektmanagements im Zusammenhang mit Lokalisierung, es folgt eine Auseinandersetzung mit jenen Themen, die für Übersetzer/innen und andere Sprachexpert/innen von Relevanz sind: Terminologie, Übersetzung und Qualitätsmanagement. Die Aufgabengebiete von Softwarespezialist/innen werden der Vollständigkeit halber kurz angeschnitten, da der Hauptaugenmerk jedoch nicht auf ihnen liegt, wird nicht näher auf sie eingegangen.

5.1 Projektmanagement

Dies ist ein Thema, das in seiner Gesamtheit nahezu unermesslich groß ist, deshalb werden hier die Besonderheiten bei der Arbeit mit Lokalisierungsprojekten herausgearbeitet. Die zentralen Fragen beschäftigen sich den Kompetenzen und Aufgaben, die für Projektmanager/innen relevant sind, verbunden mit den Menschen, mit denen sie zusammenarbeiten, mit dem Ablauf eines Lokalisierungsprojektes, sowie mit dem Lokalisierungskit, das einen Fixpunkt in der Arbeit von Projektmanager/innen darstellen.

5.1.1 Kompetenzen und Aufgaben

Ein Lokalisierungsprojekt ist in der Regel mit einem großen Planungs- und Koordinierungsaufwand verbunden. Die zentrale Person ist der/die Projektmanager/in, bei dem/der alle Fäden zusammenlaufen. In kleinen Lokalisierungsunternehmen kann es vorkommen, dass ein/e Leiter/in des Übersetzungsteams die Projektleitung übernimmt, in größeren Unternehmen gibt es eigene Projektmanager/innen, teils wird diese Position auch an freiberufliche Projektmanager/innen vergeben (vgl. Esselink 2000: 427-429; Cerda 2010: 41).

Esselink (2000: 427-428), Cerda (2010: 40-42) und Spacinsky (2013: 28-29) beschäftigen sich mit den Kompetenzen und Fähigkeiten, die für diese Position wichtig sind. Die Grundvoraussetzung ist natürlich generelle Kenntnis (und Erfahrung) im Bereich Projektmanagement. Das bedeutet, Projektmanager/innen müssen Zeitpläne erstellen können, Ressourcen und Budget planen. Außerdem müssen sie mit Kunden und Mitarbeiter/innen sowie be-

auftragten Expert/innen kommunizieren und verhandeln und sich mit verschiedenen Arten von Software auskennen. Dazu zählt Workflowsoftware, Datenmanagementsoftware, aber auch Anwendungen, die üblicherweise im Büro eingesetzt werden. Im Lokalisierungsprojekt sollten Projektmanager/innen außerdem zumindest ein Grundverständnis der CAT-Tools haben, die von Übersetzer/innen und anderen Beteiligten verwendet werden, um sich über die benötigte Zeit der Arbeitsschritte im Klaren zu sein beziehungsweise um Unterstützung durch Expert/innen organisieren zu können. Außerdem sind Kenntnisse im Bereich Softwareentwicklung von Vorteil, sowie Wissen über Lokalisierung und allen damit verbundenen Aspekten.

Cerda (2010: 40-42) beschäftigt sich in diesem Zusammenhang mit den Unterschieden zwischen den Arbeitgebern und teilt in Projektmanager/innen im Kundenunternehmen, im Dienstleistungsunternehmen und Projektmanagement bei der Lokalisierung von Spielen. Bei Kunden sind Projektmanager/innen zuständig für alle Prozesse bevor das Projekt an den Lokalisierungsdienstleister weitergegeben wird, so zum Beispiel für Internationalisierung während des Entwicklungsprozesses, als auch für die Aufgaben nach der Lieferung durch den Dienstleister. Projektmanager/innen im Lokalisierungsunternehmen sind verantwortlich für eine Reihe von Aufgaben, die im Folgenden besprochen werden. Die Besonderheit des Projektmanagements im Rahmen der Lokalisierung von Spielen ist, dass eine künstlerische und multimediale Komponente hinzukommt; bei der Zusammenarbeit mit dem kreativen Personal ist besonderes Fingerspitzengefühl gefragt. Auf der praktischen Seite kommen auch die technischen und kulturspezifischen Besonderheiten dazu (Multimedialokalisierung, NTSC beziehungsweise PAL, Jugendschutzbestimmungen et cetera).

Sind bei allen Projekten die Verhandlungs- und Kommunikationsfähigkeit ein wichtiger Punkt, kommt bei Lokalisierungsprojekten noch hinzu, dass man mit Menschen aus verschiedenen Kulturkreisen arbeiten muss, so zum Beispiel mit beauftragten Übersetzer/innen in anderen Ländern. Deshalb sind Fremdsprachkenntnisse und nicht zuletzt Wissen über andere Kulturen von Vorteil. Die erste Herausforderung ist hier die interkulturelle Kommunikation selbst. Arbeitsweise, Ethik, die grundsätzliche kulturelle Sichtweise eines jeden Menschen machen es oft nicht einfach, die in Auftrag gegebenen Projektbestandteile auf die gewünschte Weise und zum gewünschten Zeitpunkt wiederzubekommen oder zumindest ein gutes Arbeitsklima herzustellen. Ist man mit der Umgangsweise der Kultur, in der man sich bewegt, nicht vertraut, kann das zu Unstimmigkeiten führen. Bell (2011: 49-50) beschreibt in einem Beispiel die Zusammenarbeit mit somalischen Übersetzer/innen. Hier wurde die Zusammenarbeit oft nach einem Auftrag abgebrochen oder nach der Bewerbung die benötigten Unterlagen nicht geschickt. Erst später erfuhr die Auftraggeberin, dass Geschäfte in der Kultur der Übersetzer/innen bereits nach mündlicher Vereinbarung gültig waren und es für sie somit nicht verständlich war, dass sie in den USA schriftliche Bewerbungen und Honorarnoten zu schicken hatten. Gelöst wurde dieses Problem schließlich durch eine Erklärung zu den Gegebenheiten, die mit jeder Anfrage mitgeschickt wurde, in der sich die Auftraggeberin für

die Unannehmlichkeiten entschuldigte und erklärte, dass es in den USA rechtlich notwendig war, diese Dinge in schriftlicher Form zu dokumentieren. Beispiele wie diese finden sich zahlreich in der Fachliteratur.

Die globale Verteilung der Beauftragten führt zu einer zweiten Herausforderung. Vor allem durch die Verbreitung des Internets, das grundsätzlich eine der stärksten Triebkräfte der internationalen Zusammenarbeit in diesem Bereich ist, befinden sich die meisten Projektbeteiligten nicht mehr in einem Gebäude, sondern arbeiten auf dem ganzen Globus verstreut. Deshalb spielen zusätzliche Faktoren eine große Rolle. Die Kommunikation über E-Mails und dergleichen sollte auch so gestaltet werden, dass die Nachrichten auch wirklich den/die Empfänger/in erreichen und nicht im Spam-Ordner landen. Es empfiehlt sich, ein System zur Kennzeichnung von Nachrichten zu verwenden, um sicherzustellen, dass dringende Botschaften auch so schnell wie möglich empfangen und bearbeitet werden. Wegen des Zeitunterschieds ist es schwieriger, direkte Kommunikation herzustellen, das heißt, Telefongespräche oder virtuelle Konferenzen müssen zu einem Zeitpunkt stattfinden, der für alle Teilnehmer/innen einhaltbar ist. Auch beim Zeitpunkt der Abgabe sind Zeitunterschiede zu beachten, sowie andere, profan wirkende, Unterschiede, wie zum Beispiel nationale Feiertage, die eingehalten werden müssen. In Kapitel 4 wurde beschrieben, dass sich Lokalisierungsvorhaben mittlerweile in der ganzen Welt stärker ausbreiten, das heißt, auch immer mehr Länder zum Ziel werden, wo der technische Standard nicht auf demselben Niveau liegt wie in den westlichen Ländern. Auch wenn Übersetzer/innen ihre Arbeit zeitgerecht abschließen, können sie diese nicht abliefern, wenn der Strom oder das Internet ausfallen. Dies sind Risikofaktoren, die eingeplant werden müssen (vgl. Astashkina 2005: 14-15; Jones 2007: 50-51).

Neben den praktischen Herausforderungen gibt es noch subtilere Aspekte, die zu beachten sind. So ist es viel schwieriger, das Team zu motivieren, wenn es keine Treffen gibt und ohnehin lokale Distanz herrscht. Es ist für die meisten Menschen nicht dasselbe, eine E-Mail zu bekommen oder mit dem/der Chef/in zu sprechen. Deshalb ist es wichtig, Möglichkeiten zu finden, einerseits die Fähigkeiten der Mitarbeiter/innen zu überprüfen und sicherzustellen, dass diese nach bestem Wissen und Gewissen arbeiten, und andererseits, sie so zu motivieren, dass sie das gerne tun (vgl. Astashkina 2005: 14-15).

Ein Projekt beginnt, bevor sich Softwareentwickler/innen, Übersetzer/innen und andere Beteiligte mit dem Ausgangsmaterial beschäftigen. Die Arbeit für Projektmanager/innen beginnt gegebenenfalls mit der Erstellung des Angebots, was meistens dann der Fall ist, wenn bereits eine Zusammenarbeit zwischen dem Softwarehersteller und dem Lokalisierungsunternehmen besteht. Wenn nicht, fällt dies üblicherweise in den Zuständigkeitsbereich des/der Vendor Manager/in.

Nach Erstellung und Annahme des Angebots erfolgt die eigentliche Planung des Projekts. Die Evaluierung des Projekts wird bereits für die Angebotserstellung durchgeführt, es sind jedoch normalerweise noch einige Dinge zu klären, bevor die Projektbeteiligten mit den ihnen zugewiesenen Arbeitsschritten beginnen. So sollte das Ausgangsmaterial auf Vollstän-

digkeit und Richtigkeit überprüft werden und eine detaillierte Aufgabenliste erstellt werden, die als Ausgangspunkt für den Projektzeitplan dient. Dazu ist es nötig, gewisse Fragen mit dem Auftraggeber abzuklären, zum Beispiel, wie und mit welcher Person kommuniziert werden soll, ob von Seiten des Kunden Korrekturen durchgeführt werden, wie oft Zwischenberichte erwartet werden, ob bereits Zeitpläne und Fristen vorliegen und ob es unterschiedliche Abgabefristen für verschiedene Sprachgruppen gibt. Oftmals wird Software in mehreren Schüben, oder „tiers“, ausgeliefert, zuerst in den Sprachen der größeren Märkte, wie Französisch, Italienisch, Deutsch und Spanisch (FIGS), danach kleinere Sprachen, zumeist die skandinavischen Sprachen und Niederländisch et cetera. Gibt es mehrere „tiers“, beginnt die Lokalisierung für den zweiten Schub womöglich erst nach der Veröffentlichung in den ersten vier Märkten, während die Lokalisierung für die restlichen Märkte kurz vor Abschluss dieser Projekte beginnt (vgl. Esselink 2000: 438-439; Westland 2002: 2).

Ein Schritt, der für einen problemlosen Projektverlauf notwendig ist, ist es, sicherzustellen, dass alle Beteiligten auf die jeweiligen Dateien zugreifen können. Dies schließt auch eine übersichtliche Ordnerstruktur ein. Außer den Projektdateien, von denen immer jeweils die aktuelle Version zur Verfügung stehen sollte, sollten auch Referenzmaterialien und proprietäre Tools inkludiert sein, sowie Templates, Stylesheets, Translation Memorys, Glossare und administrative Informationen, wie Zeitpläne, Kostenvoranschläge und Projektpläne. Die Benennung der Dateien sollte systematisch erfolgen, sodass nicht bereits bearbeitete Dateien und Ausgangsdateien verwechselt werden oder erstere sogar überschrieben. Wird bei einem Projekt in mehrere Sprachen beziehungsweise Locales lokalisiert, so sollten diese auch entsprechend gekennzeichnet werden. Wichtig ist auch, dass alle Beteiligten wissen, welche Arbeitsschritte sie zu erfüllen haben und wie sie dabei vorgehen sollen (vgl. Esselink 2000: 439-440).

Ein umfassender Zeit- und Ressourcenplan ist grundlegend für den Erfolg eines Projektes. Dafür müssen alle Aufgaben gesammelt werden, also welche Dateien wie bearbeitet werden sollen und was schließlich abzuliefern ist. Dazu gehören üblicherweise Aufgaben wie Terminologievorbereitung, Vorbereitung der zu übersetzenden Texte, Übersetzung aller Bestandteile, Zusammenfügung und Anpassung der lokalisierten Softwareversion, Tests, Erstellung der Zusatzmaterials und Qualitätssicherungsschritte. Im Endeffekt sollten alle Aufgaben inklusive Abgabedaten und Material verteilt sein. Während des Projekts sollte laufend kontrolliert werden, ob der Plan eingehalten wird und gegebenenfalls Probleme gelöst werden. Mit Meilensteinen wird festgestellt, ob das Projekt im Zeitplan ist, oder ob bereits Verzögerungen eingetreten sind. Ein besonderes Augenmerk bei der Kontrolle liegt auf der Abfolge der Aufgaben, es kann beispielsweise die lokalisierte Software nicht getestet werden, wenn die Wiedereinbindung der übersetzten Texte nicht erfolgt ist. Insgesamt sind also Projektmanager/innen bei der Durchführung des Projekts zuständig für die Überprüfung des Fortschritts, Zuteilung der Ressourcen, das Budget, Qualitätsmanagement und die Kommunikation (vgl. Esselink 2000: 438-461):

5.1.2 Zusammensetzung der Projektbeteiligten

In Kapitel 3.1.2 wurde bereits kurz auf die Personalzusammensetzung bei Lokalisierungsprojekten eingegangen. In diesem Kapitel wird dies aufgegriffen und auf die Kompetenzen eingegangen.

5.1.2.1 Sprachexpert/innen

In diese Kategorie fallen alle Beteiligten, die sich mit der sprachlichen Komponente auseinandersetzen, also Übersetzer/innen, Übersetzungskoordinator/innen, Korrektor/innen und Terminologiebeauftragte (vgl. Esselink 2000: 16). Vor allem Übersetzer/innen arbeiten meistens freiberuflich, können aber auch bei einem Lokalisierungsdienstleister oder einem Lokalisierungskunden angestellt sein. Die Vor- und Nachteile der unterschiedlichen Beschäftigungsmodelle wurden bereits in Kapitel 4.3.2 erläutert. Spacinsky (2013: 25) beschreibt Sprachexpert/innen als Kern der Sprachindustrie, die Sprachbegabung sowie ein Gefühl für Sprachnuancen und Kontext mitbringen sollen. Gute Kommunikationsfähigkeiten, Präzision und Detailgenauigkeit sind neben Sprachkenntnissen auf sehr hohem Niveau wichtige Kompetenzen. Außerdem ist Fachwissen in bestimmten Bereichen von großem Vorteil.

Durch den technischen Fortschritt hat sich die Arbeit für Übersetzer/innen verändert. CAT-Tools sind in den verschiedenen Bereichen immer grundlegender, durch das Internet ist der Arbeitsort immer unwichtiger geworden. Maschinelle Übersetzung sorgt für ein neues Einsatzgebiet für Übersetzer/innen: die Nachbearbeitung oder „Post-Editing“ von maschinell übersetzten Texten (vgl. Steiert et al. 2010: 46).

5.1.2.2 Expert/innen im technischen Bereich

Hier gibt es auf Kunden- sowie Dienstleisterseite viele Rollen. Esselink nennt Qualitätssicherungsexpert/innen, Lokalisierungsingenieur/innen, Tester/innen oder Expert/innen in der Qualitätskontrolle, sowie CAT-Tool-Expert/innen (vgl. Esselink 2000: 15-17). Spacinsky (2013: 25-27) fügt Internationalisierungsexpert/innen hinzu, sowie Solutions-Architekt/innen und technische Leiter/innen. Alle diese Berufe erfordern eine Fachausbildung und gute Kenntnisse der technologischen Materie.

Am vorderen Ende der Aufgabenkette stehen Internationalisierungsexpert/innen, die dafür sorgen, dass eine Anwendung so aufgebaut wird, dass sie später besser lokalisiert werden kann, dazu gehört, zum Beispiel, die Unterstützung verschiedener Formate und Zeichensätze.

Lokalisierungsingenieur/innen sind zuständig für die Vorbereitung der Dateien für Übersetzung und sonstige Anpassungen, außerdem für die Analyse der Ausgangsdateien für die Übersetzung. Nach Eingang der übersetzten Dateien sind sie für die Kompilierung der Zieldateien zuständig sowie für diverse Anpassungen, zum Beispiel der Größe der Dialogfelder.

Nach der Lokalisierung sind Softwaretester/innen und Qualitätskontroll-Expert/innen für die Überprüfung der lokalisierten Software verantwortlich. Sie testen die Software auf Funktion, während für die sprachliche Überprüfung Korrektor/innen oder andere Sprachexpert/innen zuständig sind. Qualitätssicherungs-Spezialist/innen überprüfen die Prozesse und Vorgänge während des gesamten Projektes.

Solutions-Architekt/innen beschäftigen sich mit der Entwicklung von Lokalisierungslösungen auf einer höheren Ebene, also mit der Art und Weise, sowie Struktur, wie die Lokalisierung aufgebaut werden kann. Technische Leiter/innen stehen an der Spitze des technischen Fachteams. Sie planen und koordinieren die Abläufe, Aufgabenzuweisung und Ausführung. Auch wenn die beiden letztgenannten Positionen nicht unbedingt direkt mit Codes und Programmierungsarbeit zu tun haben, sind profunde Kenntnisse der Materie erforderlich, die auch durch längere Erfahrung erworben beziehungsweise vertieft wurden (vgl. Spacinsky 2013: 25-27).

5.1.2.3 Management

In Softwareunternehmen gibt es Managementpositionen, die für Lokalisierung zuständig sind; Vendor Manager/innen und Lokalisierungsmanager/innen. Diese Positionen sind vor allem in kleineren Unternehmen oft in einer zusammengefasst. Zu den Aufgaben von Vendor Manager/innen zählen die Anfrage nach Angeboten und Kostenvoranschlägen, die Pflege der längerfristigen Zusammenarbeit mit Lokalisierungsdienstleistern und deren Auswahl, Verhandlung von Preis und Konditionen, Überwachung der Verträge, Qualitätsüberprüfung und Budgetverwaltung. Lokalisierungsmanager/innen sind für die laufenden Projekte zuständig, also für Kick-off-Meetings, die Definition der Prozesse und Kontaktpunkte, Weiterleitung der Projektmaterialien, generelle Überwachung des Projektverlaufs, -budgets und der -verträge, Koordination der Veröffentlichung, Assistenz des/der Vendor Manager/in und die Koordination des Projektteams auf Kundenseite.

Im Lokalisierungsunternehmen gibt es Account Manager/innen, die für den Kontakt und die Zusammenarbeit mit Kunden verantwortlich sind und Projektmanager/innen, die die Projekte selbst leiten, also für die Zeitpläne, Ressourcenzuteilung, Kundenkommunikation und Überwachung des Budgets und des Projektfortschritts zuständig sind (vgl. Esselink 2000: 16-18). Wie in anderen Unternehmen gibt es Geschäftsführer/innen, Manager/innen und Abteilungsleiter/innen, die das Unternehmen beziehungsweise die Abteilungen führen. Eigens

angeführt werden sollten noch Beschaffungsmanager/innen und Verkaufsleiter/innen, die für die Arbeit mit Kunden zuständig sind (vgl. Spacinsky 2013: 27-30).

5.1.3 Projektablauf

Wie bereits erwähnt, unterscheiden sich Lokalisierungsprojekte oft erheblich voneinander, Esselink (2000: 17-23; 428-461) beschreibt typische Phasen, die meistens vorkommen und eine Reihe von Aufgaben, die Projektmanager/innen zu bearbeiten haben. Lewis et al (2009: 30) beschreiben auch den Projektablauf, oder Workflow, schließen jedoch die Vor- und Nachbereitung sowie die organisatorische Arbeit nicht mit ein.

Ein Projekt beginnt im Prinzip bereits vor der Auftragsvergabe mit der Ausschreibung oder einer Angebotsanfrage. Hierbei wird ein Lokalisierungsdienstleister gesucht, der das beste Angebot liefern kann. In der Anfrage wird auch bereits Material mitgeliefert, sodass die Mitarbeiter/innen ein adäquates Angebot erstellen können.

Ist ein Projekt vergeben worden, findet üblicherweise ein Kick-off-Meeting statt, bei dem das Projekt besprochen wird, Fragen geklärt werden können und die Beteiligten die Möglichkeit haben, sich persönlich kennenzulernen.

Das Ausgangsmaterial muss vor der Erstellung von Zeit- und Ressourcenplan analysiert werden. Dazu werden die Dateienpakete, meistens in Form eines Lokalisierungskits, an das Lokalisierungsunternehmen geliefert. Inhalt und Umfang des Lokalisierungskits werden weiter unten näher beschrieben.

Nach der Analyse werden Zeit und Ressourcen geplant. Diese ist eine der wichtigsten Phasen, die grundlegend für das gesamte Projekt ist.

Der erste Arbeitsschritt, der sich mit dem Ausgangsmaterial beschäftigt, ist üblicherweise die Erarbeitung der Terminologie. Danach wird das Ausgangsmaterial für die Übersetzung vorbereitet und Projekt-Translation Memorys erstellt. In der Übersetzungsphase wird zuerst die Software übersetzt und danach die Hilfedatei sowie Dokumentation. Die Texte müssen alle überprüft werden, bevor die Zielformate erstellt werden und, im besten Fall, bevor die Translation Memorys upgedatet werden. Nach der Einbindung der Übersetzungen wird die Software auf Funktion und Sprache getestet und auch Hilfe sowie Dokumentation fertiggestellt. Im Falle von Multimediateilen, die ein wichtiger Bestandteil etwa von Spielen sind, kommen Schritte hinzu, wie zum Beispiel die Untertitelung oder Synchronisation von Videos und sonstigen gesprochenen Texten (vgl. Lutsuk 2014: 28-29).

Nach Abschluss und Lieferung der Projektbestandteile werden oft Nachbesprechungen und Analysen organisiert. Hierbei wird festgestellt, was gut funktioniert hat, was nicht und wo Verbesserungen möglich sind.

In Kapitel 4 wurde Crowdsourcing als Alternative diskutiert. Bei Projekten dieser Art sieht der Projektablauf im Allgemeinen anders aus. Es wird zuerst die Möglichkeit für die „Crowd“ geschaffen, Übersetzungs- und Terminologievorschläge einzubringen. Auch hier sind Qualitätssicherungsschritte von großer Bedeutung, so werden diese entweder über ein Ratingsystem von den User/innen bewertet oder von professionellen Übersetzer/innen oder Korrektor/innen überprüft (vgl. Lewis et al 2009: 31-32).

5.1.4 Lokalisierungskit

Das Lokalisierungskit ist ein Paket an Dateien und Informationen, das dem Lokalisierungsunternehmen vom Kunden übermittelt wird. Damit ist es für den/die Projektmanager/in einfacher, das Projekt zu planen und Rückfragen werden so auf ein Minimum reduziert.

Der erste Teil gibt die wichtigsten Informationen über das Projekt. So sollte das Produkt beschrieben werden, inklusive Namen, Zweck, Produkthomepage, Vorgängerversionen, bereits veröffentlichte Versionen und die Wortanzahl. Es sollten bereits lokalisierte Vorgängerversionen bekannt gegeben werden, einschließlich der Locales und der verantwortlichen Personen. Ein Überblick über das Projektteam, samt Namen, Position, Zuständigkeitsbereich, sowie ein Organigramm sollte mitgeliefert werden, ebenfalls ein Zeitplan mit Projektumfang und Meilensteinen.

Beim zweiten Teil geht es um die einzelnen Schritte der Lokalisierung selbst, also wer welche Dateien bearbeiten soll, in welchem Format diese abgegeben werden sollen et cetera. Dazu gehören Informationen über die Benennung der Dateien, die Übermittlung der Dateien und auch, welche Dateien in welche Sprachen übersetzt werden sollen. Insgesamt wird hier, mehr oder weniger detailliert, beschrieben, welche Aufgaben ausgeführt werden sollen.

Zu diesem Zweck ist es auch nötig, die Tools anzugeben, die für die Bearbeitung der Dateien zu benutzen sind. In einem Softwarepaket gibt es verschiedenste Formate für Textdateien, Systemdateien, Ressourcendateien et cetera. Hierbei sollte beachtet werden, dass die Version oftmals eine Rolle spielt, da nicht alle Tools und Formate abwärtskompatibel sind. Neben der Angabe der Tools selbst sollte auch angegeben werden, wer im Problemfall zu kontaktieren ist, Informationen und Hilfen zu den Tools und gegebenenfalls bekannte Probleme.

Weiters enthalten sind bereits bestehende Translation Memorys, Templates und Fonts für die Bildbearbeitung, Wortanzahl, Checklisten und Testskripts für alle beteiligten Personen. Referenzmaterial schließt Informationen über Vorgängerversionen mit ein, Glossare der Ausgangssprache, mehrsprachige Glossare, Originaldateien, Zusatzmaterial, Informationen über die Zieltextrezipient/innen, Style-Guides und Terminologieinformationen. Sonstige Informationen betreffend die Organisation sind Daten der Kontaktpersonen, Informationen über Kommunikationswege und Beschreibungen der Prozesse.

Im Endeffekt sind all dies Informationen, die der/die Projektmanager/in ohnehin braucht und erfragen müsste, wenn sie nicht von vornherein mitgeliefert würden. Werden diese detailliert im Rahmen eines Lokalisierungskits geliefert, stellt dieses eine gute Grundlage für die Planung des Projektes dar (vgl. Zerfaß 2005: 8-11).

5.2 Terminologie, Terminologiemanagement

Terminologearbeit ist grundsätzlich ein Schwerpunkt im Bereich Fachübersetzen und verwandten Gebieten, sowie im Bereich Forschung und Entwicklung. Da bei Lokalisierung diese Bereiche aufeinandertreffen, ist die Bedeutung von Terminologie in der Lokalisierungsindustrie nicht abzustreiten. Terminologearbeit beginnt nicht erst im Lokalisierungsprojekt, sondern sollte idealerweise bereits im Entwicklungsprozess des Ausgangsprodukts eingebunden werden. In der Realität gab es in der Softwareindustrie lange kein Bewusstsein darüber, weshalb Expert/innen seit Langem Überzeugungsarbeit leisten, Terminologearbeit in die Entwicklung zu integrieren. In diesem Kapitel wird die Bedeutung von Terminologearbeit erläutert, wie dies in der Praxis aussieht und welche Aufgaben Terminologieexpert/innen zufallen.

5.2.1 Probleme, Ursachen und Folgen

Bereits Esselink (2000: 397) schreibt, dass Terminologie ein oft unterschätzter Aspekt ist und verweist auf die Wichtigkeit von Terminologearbeit für Softwareentwicklung und Lokalisierung. In späteren Quellen wird meistens beschrieben, dass Terminologearbeit zwar im Bereich der Lokalisierung Einzug gefunden hat, jedoch Terminologieverantwortliche und Übersetzer/innen mit mangelhafter Terminologearbeit in den Ausgangstexten zu kämpfen haben. Lombard (2006: 156) beschreibt ein typisches Szenario wie folgt: Nach Abschluss des Softwareprojekts sammelt ein/e Dokumentationsspezialist/in Termini und Definitionen in einem ausgangssprachlichen Glossar. Dieses wird an den Lokalisierungsdienstleister weitergegeben, wo Übersetzer/innen oder gegebenenfalls ein/e Terminologieexpert/in ein zwei- oder mehrsprachiges Glossar erstellen. Diese können zwar Rückfragen stellen, haben jedoch nicht viel Zeit, da der Zeitplan üblicherweise relativ eng ist. Trotzdem ist die Terminologie im lokalisierten Produkt normalerweise kohärenter als im ausgangssprachlichen Produkt, da sich aktiv jemand damit befasst. Lombard mutmaßt, dass in Softwareunternehmen ganz einfach kein Bewusstsein darüber besteht, wie wichtig gemanagte Terminologie auch im Ausgangsprodukt ist. Der Programmcode sei heilig, doch Strings und andere Elemente der Benutzeroberfläche bestünden „nur aus Wörtern“. Auch Karsch (2011: 46, 48) schreibt, dass, zumindest in den USA, Terminologearbeit mit Lokalisierung und Übersetzung verbunden wird. Eine mögliche

Erklärung für das fehlende Bewusstsein ist, dass für die meisten Beteiligten und Kunden nicht ersichtlich ist, ob ein Fehler oder sonstiges Problem aus einem terminologischen Problem entstanden ist. Im Gegensatz zu Softwarefehlern sind diese oft nicht klar zu identifizieren und selbst wenn im Lokalisierungsprozess terminologische Bugs gefunden werden, werden diese der zielsprachlichen Terminologie zugeordnet. Um dem entgegenzuwirken und in den Unternehmen das Bewusstsein zu schaffen, dass Terminologiemanagement ein grundlegender Bestandteil der Softwareentwicklung ist, kann, wie bei Lokalisierung selbst (siehe Kapitel 4.2.), die Rendite, oder „Return on Investment“, herangezogen werden. Deren Berechnung gestaltet sich jedoch schwierig (vgl. Lombard 2006: 160-161; 163-164). Weiter unten in diesem Kapitel werden Ansätze zur Errechnung der Rendite diskutiert.

Eine weitere Problemquelle, neben dem fehlenden Bewusstsein, ist die Arbeitsweise in Softwareunternehmen. Viele Programmierer/innen arbeiten an Programmteilen, die letztlich zu einem komplexen Paket zusammengefasst werden. Diese Arbeit findet oft isoliert statt, das heißt, ohne zentral geregelte Terminologie ist es für die Entwickler/innen gar nicht möglich, dieselben Termini zu verwenden. Außerdem gab es durch die Vielzahl an neuen Begriffen viele neue Benennungen zu finden. Bei der Programmierung waren viele Entwickler/innen gerne kreativ oder bauten Referenzen ein, sodass diese insgesamt noch schwerer zu durchschauen waren (vgl. Lombard 2006: 162-163).

Mangelhafte Terminologie umschließt falsche Benennungen für Begriffe, Synonyme und Homonyme. Synonyme oder Quasisynonyme beziehungsweise Homonyme führen allgemein zur Verwirrung beim Übersetzen und beim Verwenden der Produkte, da für die Nutzer/innen nicht klar ist, ob zwei oder mehrere verwendete Benennungen nun wirklich für dasselbe stehen oder nicht. Der Grund dafür ist häufig die oben genannte isolierte Arbeit (vgl. Dunne 2007: 33). Auch bei der Wahl der Benennungen für neue Begriffe ist darauf zu achten, dass diese möglichst leicht verständlich sind. Werden Benennungen, wie es oft bei der Softwareentwicklung geschieht, aus der Gemeinsprache übernommen, so ist es umso schwieriger, die Begriffe zu verstehen, je abstrakter der Zusammenhang ist (vgl. Schmitz 2009). Insgesamt ist die kohärente Verwendung der geprägten Benennungen im gesamten Produkt, inklusive Hilfedatei, Dokumentation und Zusatzmaterial, der wohl wichtigste Punkt (vgl. Dunne 2007: 33; Schmitz 2009). Karsch (2011: 46-48) identifiziert mehrere Probleme, die teilweise in die oben genannten Kategorien fallen: falsche Verwendung von Benennungen, schlecht gewählte Benennungen, inkohärente Verwendung, inklusive verschiedener Schreibweisen und Abkürzungen, sowie fehlende Erklärungen von unbekannten Begriffen.

Was sind nun die Folgen mangelnder Terminologiarbeit? Bereits im ausgangssprachlichen Produkt führt inkohärente und fehlerhafte Terminologie zu einer Einschränkung der Benutzerfreundlichkeit. Mahajan & Shneiderman (1997) geben an, dass User/innen durch inkohärente Terminologie der Benutzeroberfläche bis zu 34 % mehr Zeit für die Ausführung einer Aufgabe benötigen, als sie bei einer Anwendung mit kohärenter Terminologie brauchen

würden. Auch Lombard (2006: 165) führt einige Beispiele an, wie vor allem nicht professionelle User/innen durch mangelhafte Terminologearbeit verwirrt werden.

Rechtfertigt bereits die erhöhte Benutzerfreundlichkeit im ausgangssprachlichen Produkt die zentral gesteuerte Terminologearbeit, so wird deren Bedeutung im Zusammenhang mit lokalisierter Software beziehungsweise Übersetzung noch um Einiges gesteigert. Zunächst stellt die Übersetzung der in der Software verwendeten Terminologie grundsätzlich eine Herausforderung dar vor allem durch grammatikalische und lexikalische Unterschiede. So ist es, zum Beispiel, bei der Übersetzung aus dem Englischen nicht immer klar, ob es sich bei einem Wort um ein Nomen oder ein Verb handelt, vor allem, wenn es sich um Strings aus einzelnen Wörtern ohne Kontext handelt. Außerdem werden, wie oben beschrieben, im Bereich der Softwareentwicklung neue Benennungen oft aus der Gemeinsprache gebildet, was zu Problemen führt, wenn kulturell stark verankerte Konzepte herangezogen werden. Eine wörtliche Übersetzung funktioniert in der Zielkultur oft nicht oder ist im Extremfall sogar anstößig (vgl. Dunne 2007: 33-34; Schmitz 2009). Wird Terminologearbeit, in vielen Fällen aus Zeitmangel, nicht ordentlich durchgeführt, so ist es für Übersetzer/innen schwierig, eine adäquate Übersetzung anzufertigen, das heißt, fehlerhafte Ausgangstexte führen zu fehlerhaften Zieltexten. Hierzu verweist Dunne (2007: 36) auf Wright (2001: 491-492), die aus der Annahme, dass Übersetzer/innen und technische Redakteur/innen die Gemeinsprache beherrschen, schließt, dass Terminologie jener Aspekt eines Textes ist, der am fehleranfälligsten ist.

Dunne (2007: 37) beschäftigt sich schließlich mit dem Risiko von ungemanagter Terminologie anhand der Auswirkungsanalyse. Zusammengefasst kann hier festgestellt werden, dass gemanagte Terminologie in der Ausgangssprache die Wahrscheinlichkeit senkt, dass fehlerhafte Terminologie Eingang in die Produktion findet. Durch Terminologiemanagement in allen Sprachen wird das Wissen über das Produkt weitergegeben, wodurch alle weiteren Prozesse viel weniger fehleranfällig sind. Dies bedeutet auch, dass Arbeitsschritte, die sonst doppelt ausgeführt werden, wie im oben beschriebenen Beispiel, nur einmal vorkommen und somit die Prozesse effizienter werden. Insgesamt sinkt dadurch das Fehlerpotenzial und somit die Wahrscheinlichkeit eines fehlerhaften Produkts, das den Ruf des Unternehmens und der Marke beschädigt beziehungsweise zu hohen Kosten führt, wenn ein Produkt nachträglich verbessert werden muss oder, im schlimmsten Fall, zurückgezogen wird.

Neben Fehlern, die während der Übersetzung entstehen, führen ebenjene terminologischen Ungenauigkeiten beziehungsweise Fehler in der Ausgangssprache zu erheblichen Zeit- und Kostensteigerungen. Dunne (2007: 34-35) beschreibt ein Beispiel mit einem fehlerhaften Terminus, der in einer Information auftaucht, die in zehn Arbeitsergebnissen verwendet wird, welche wiederum in fünf Produkten implementiert werden. Wenn diese Produkte in zehn Sprachen übersetzt werden, multipliziert sich das Auftreten dieses Fehlers und er ist schließlich an 500 Stellen vorhanden. Lombard (2006: 157-159) stellt an einem Beispiel anschaulich dar, wie nicht nur Fehler, sondern auch inkohärente Terminologie zu Kostensteigerungen führen. Er zieht hierzu Fehlermeldungen heran, von denen es in Softwareprodukten eine

Vielzahl gibt. Wenn nun drei Programmierer/innen, die isoliert arbeiten, keine standardisierte Terminologie konsultieren können, ist es möglich, dass sie für ein und denselben Fehler drei unterschiedliche Meldungen verfassen. Auch wenn das Konzept hinter diesen Meldungen dasselbe ist, ist dies durch die unterschiedlichen Formulierungen sowohl für Nutzer/innen als auch Übersetzer/innen nicht ersichtlich. Die Kostenersparnis durch den Gebrauch von CAT-Tools, so wie Translation Memory-Systemen, stellt im Bereich Lokalisierung einen erheblichen Faktor dar, da für bereits übersetzte Segmente weniger berechnet wird. Im Falle von drei unterschiedlichen Formulierungen ist es jedoch unwahrscheinlich, dass ein Translation Memory-System eine Übereinstimmung findet und deshalb wird jede Meldung einzeln übersetzt. Die Kostensteigerung allein durch Synonyme wird von Lombard berechnet. Dazu geht er in seinem Beispiel von Kosten von 0,25 \$ pro Wort aus. Bestehen die Meldungen aus zehn, elf und 25 Wörtern mit jeweils einem, zwei und drei Synonymen, so errechnen sich Kosten in der Höhe von 2,50 \$, 2,75 \$ und 6,25 \$ pro Übersetzung. In einem realistischen Szenario werden diese Einzelkosten mit 20 (für 20 Zielsprachen) und mit 1000 (für 1000 ähnliche Meldungen in einer Anwendung) multipliziert, wodurch man auf Kosten von 50.000 \$, 55.000 \$ und 125.000 \$ kommt. Wird jedoch die ausgangssprachliche Terminologie verwaltet, so reduziert sich die Wortanzahl auf neun, neun und 22 und die Anzahl der Synonyme auf null. Die Kosten pro Einzelübersetzung betreffen hiermit 2,25 \$, 2,25 \$ und 5,50 \$, was mit derselben Rechnung zu einem Endergebnis von 45.000 \$, 45.000 \$ und 110.000 \$ führt. Dadurch ergeben sich Einsparungen von 5.000 \$, 10.000 \$ und 15.000 \$. Im Endeffekt bedeutet dies, dass sich durch die Standardisierung der Terminologie nicht unerhebliche Summen einsparen lassen.

5.2.2 Return on Investment und Terminologie

Dies ist im Prinzip auch die Basis für die Berechnung des ROI. Die Rendite, die das Verhältnis zwischen Mehraufwand und Mehrertrag darstellt, wird öfter als Mittel vorgeschlagen, um für die unternehmensweite Einbindung von zentral gesteuertem Terminologiemanagement zu argumentieren. Wie bereits in Kapitel 4.2 erwähnt, sollte ROI ein umfassendes Konzept sein, bei dem nicht nur die klingende Münze eine Rolle spielt, jedoch ist dies für viele Entscheidungsträger/innen die einzige Möglichkeit, Vor- und Nachteile zu messen. Grundsätzlich ist bei Terminologiemanagement der Mehrertrag nicht im Bereich der Einkünfte zu suchen, sondern als Verringerung der Ausgaben zu verstehen. Das eingesetzte Kapital kann, im Fall eines Projektes, eine direkte Investition darstellen, im Geschäftsbetrieb aber auch die laufenden Kosten, die in die Einrichtung und Betreuung des Terminologiemanagements fließen beziehungsweise eine Kombination beider Faktoren (vgl. Childress 2007: 43; Wittner 2007: 52).

Um eine finanzielle Rechnung durchzuführen, müssen zuerst Zahlen identifiziert werden. Dies bedeutet, dass die Kosten des Terminologiemanagements aufgestellt werden müssen, als zweiten Schritt die Ausgaben, die durch nicht gemanagte Terminologie entstehen und schließlich werden beide Werte verglichen und die Einsparungen als Mehrertrag in Bezug zu den Mehrausgaben gesetzt.

Bei der Kostenaufstellung gibt es drei grundlegende Faktoren: Erstellung der Terminologie, Recherche und die benötigte Infrastruktur. In die erste Kategorie fällt die Erstellung der ausgangssprachlichen und zielsprachlichen Einträge. Die zweite umfasst die Evaluierung der Quellen, Diskussionen unter den beteiligten Expert/innen und die Überprüfung der Einträge. Die dritte Kategorie setzt sich aus der Hardware und Software zusammen, die für die Terminologearbeit benötigt wird, sowie aus der Entwicklung und dem Support, aus der Koordination der Standards und Prozesse, sowie der Schulung der Mitarbeiter/innen. Um die Arbeit, die von Mitarbeiter/innen ausgeführt wird, beziffern zu können, ist es am einfachsten, auf die investierte Arbeitszeit zurückzugreifen. In der Buchhaltung gibt es normalerweise bereits Werte zu den Gesamtkosten pro Stunde, sodass die Gesamtkosten von Terminologiemangement gut ausgerechnet werden können (vgl. Childress 2007: 43-44).

Um einen Vergleichswert zu erhalten, werden die Ausgaben errechnet, die bei nicht zentral gemanagter Terminologie entstehen. Hierzu gehört die Suche nach nicht standardisierten Termini in Produkten, inklusive Dokumentation, Marketingmaterialien und Websites in den verschiedenen Sprachen, die Arbeitszeit, die in die Bearbeitung von Anfragen vonseiten der Übersetzer/innen fließt, die Überarbeitung von maschinell übersetzten Dokumenten und die Kosten, die durch Verspätungen wegen terminologiebezogener Probleme auftreten. Außerdem müssen auch die Kosten herangezogen werden, die durch den vermehrt notwendigen Kundensupport entstehen: Durch inkohärente Terminologie sinkt die Nutzerfreundlichkeit der Produkte, was zu mehr Anfragen führt. Es sollten hier die Anfragen heraussortiert werden, die direkt mit terminologischen Problemen zu tun haben, um diese Zahl nicht zu verzerren. Die ersten Punkte werden wiederum durch die Kosten der Arbeitszeit ausgedrückt, während es für den letzten Punkt oft andere, standardisierte Ansätze gibt, beispielweise eine gewisse Summe pro bearbeiteter Anfrage (vgl. Childress 2007: 44). Auch Wittner (2007: 53) beschreibt in einer Fallstudie, dass die Kosten der Terminologearbeit nach Arbeitszeit errechnet wurden. So wurden die Mitarbeiter/innen angehalten, in einem gewissen Zeitraum jeden Tag aufzuschreiben, wie viel Zeit sie in Terminologearbeit investiert hatten. Dies erstreckte sich über alle Unternehmensbereiche. Obwohl am meisten Arbeitszeit bei der Nachbearbeitung und Korrektur der Übersetzung für terminologische Belange verwendet wurde (zirka ein Viertel), wurde in anderen Bereichen auch 1 bis 5 Prozent mit Terminologearbeit verbracht; der zweite Bereich in der Aufstellung war Übersetzung mit 5 %, gefolgt von der technischen Redaktion mit 3 %, der Schulungsabteilung und dem Informationsservice mit 2 % und der Entwicklungsabteilung mit über 1 %. Auch in den Rechts- und Finanzabteilungen wurde Zeit für Terminologearbeit aufgewendet, jedoch weniger als ein Prozent, und so-

mit wurden diese Daten für die weitere Rechnung ausgeklammert. Auch hier waren die relevanten Tätigkeiten die Suche nach den richtigen Termini, die Suche nach den richtigen zielsprachlichen Äquivalenten und die Abhandlung der terminologiebezogenen Kundenanfragen.

Childress (2007: 44) bringt hier ein Beispiel, das dem im Vorkapitel ähnelt, und rechnet aus, was die Standardisierung eines Terminus und dessen weitere Verwendung kostet. Wird ein Schlüsselterminus vor der Übersetzung standardisiert, können folgende Kosten addiert werden: Erstellung und Überprüfung des Eintrags, Suche und Korrektur von ungewollten Synonymen, Erstellung der zielsprachlichen Einträge und deren Übersetzung in den zielsprachlichen Dokumenten. Erfolgt die Standardisierung des Terminus erst nach der Übersetzung, müssen die Kosten mit der Anzahl der Zielsprachen multipliziert werden, da im Prinzip alle Arbeitsschritte wiederholt werden müssen. Errechnet man nun den ROI nach dem oben beschriebenen Prinzip, also betrachtet man die Einsparungen als Mehrertrag, so kommt man bei zehn Zielsprachen auf einen ROI von 900 %, das heißt, wenn die Standardisierung auf einen späteren Arbeitsschritt verschoben wird, ist dies neunmal teurer. Mit noch höheren Kosten kann gerechnet werden, wenn Terminologearbeit noch später gemacht wird, also zum Beispiel nach der Auslieferung der lokalisierten Produkte mittels Update.

Ähnlich wie bei einzelnen Termini schlägt sich zentral gemanagte Terminologie auch auf Unternehmensebene nieder. Durch die einmalige Investition in die Errichtung einer zentralen Terminologie-Infrastruktur werden die nachfolgenden Kosten pro Eintrag geringer. Wird Terminologie von jeder Unternehmensabteilung einzeln erstellt und gewartet, so steigen die Kosten linear: Für jede Abteilung entstehen dieselben Kosten, das heißt, es können keine Einsparungen durch Wiederverwendung erzielt werden, jedoch ist keine Erstinvestition nötig. Es ist jedoch anzunehmen, dass alle Termini öfter von verschiedenen Abteilungen verwendet werden, und somit amortisiert sich die erste Investition relativ schnell und man kann insgesamt mit Einsparungen rechnen (vgl. Childress 2007: 45-46). In Wittners Fallbeispiel (2007: 54) werden durch zentral gesteuertes Terminologiemanagement jährlich 170.000 \$ eingespart, was bei einer Profitmarge von 50 % zu 85.000 \$ pro Jahr führt. Bei der Einführung des zentralisierten Systems kommen einmalige Kosten in der Höhe von 63.500 \$ auf das Unternehmen zu, die für Hardware und Software, Lizenzen, Softwareanpassungen, Installationskosten und diverse Kommunikationswege ausgegeben werden. Diese werden über drei Jahre abgeschrieben. Die jährlichen Fixkosten betragen 10.000 \$, die in Wartung, Softwareupdates, Datenbankupdates, Workflow-Management und Weiteres fließen und jedes Jahr um 20 % sinken. Die Berechnung für die ersten drei Jahre nach Einführung sehen nach dem Schema „Ertrag dividiert durch Investition“ wie folgt aus: $(85.000 - 31.200) / 31.200 = 172 \%$. Nach diesem Zeitraum fallen die Einführungskosten weg, was zu einem noch viel höheren ROI führt: $(85.000 - 5000) / 5000 = 1.600 \%$. Diese Zahlen sind lediglich ein Beispiel, doch beweisen sie, dass die einmalige Investition in ein zentralisiertes Terminologiesystem auf jeden Fall ihr Geld wert ist.

5.2.3 Terminologiemanagement

Nach der Besprechung der Mängel und deren Folgen soll in diesem Kapitel Terminologiemanagement nach verschiedenen Gesichtspunkten behandelt werden. Zuerst wird besprochen, worauf bei der Terminologiarbeit selbst zu achten ist, daraufhin wird auf die Organisation im Unternehmen eingegangen und schließlich werden Methoden vorgestellt, wie ROI berechnet werden kann.

5.2.3.1 *Termini und Einträge*

Der Lokalisierungs- und allgemein der Computerbereich stellt in Hinsicht auf terminologiewissenschaftliche Grundlagen natürlich keine Ausnahme dar. Es gibt hier, wie in allen Arbeitsbereichen, auch gewisse Eigenheiten und Merkmale. Im Folgenden sollen die Grundlagen nach Wright & Budin (1997) im Zusammenhang mit Lokalisierung besprochen werden.

Am Beginn der Terminologiarbeit stehen die neuen Technologien, Prozesse und Produkte, die im Rahmen der Entwicklung des Computers entstanden sind (vgl. Schmitz 2009). Aus diesen Dingen werden durch Abstrahierung Begriffe gebildet, die durch Benennungen repräsentiert werden. Neben den Hardware-Begriffen, die für angreifbare Dinge stehen (zum Beispiel „Drucker“, „Maus“, „Bildschirm“, „Prozessor“), gibt es das nahezu unbegrenzte Feld der Software-Begriffe. Da es Anwendungen zu einer nahezu unermesslichen Anzahl von Themen gibt, ist es nicht möglich, diesen Bereich abzugrenzen. Da praktisch jeden Tag neue Programme veröffentlicht werden, erweitert sich Software-Terminologie stetig. Die Begriffe stammen zumeist aus der Benutzeroberfläche. Es handelt sich um computerspezifische Begriffe aus den Betriebssystemen und Anwendungen, aber auch um Begriffe aus anderen Fachsprachen, wie es, zum Beispiel, bei Buchhaltungssoftware der Fall ist. Diese Begriffe haben auf den ersten Blick mit Lokalisierung nichts zu tun, sind jedoch von Belang, wenn sie in einem Programm verwendet werden (vgl. Corbolante & Irmeler 2001: 517; 520).

Benennungen im Softwarebereich können, so wie in anderen Fachgebieten, aus einzelnen beziehungsweise mehreren Wörtern bestehen, aus Phrasen, Kollokationen und Abkürzungen (vgl. Schmitz 2009). Beispiele hierzu wären „Datei“, „Neues Fenster“, „Speichern und Senden“, „Format übertragen“ und „RAM“. Die Bildung der Benennungen erfolgt ebenso nach den bei Sager (2001: 25-41) aufgeführten Regeln. Schmitz (2009) merkt an, dass bei der Entwicklung eines Programms sehr auf die Architektur geachtet wird. Je intuitiver diese aufgebaut ist, desto bessere und schnellere Ergebnisse lassen sich beim Benutzen erzielen. Die Texte sind innerhalb der Architektur einer der wichtigsten Informationsübermittler und deshalb ist die richtige Wahl der Terminologie oft ausschlaggebend für die Benutzerfreundlichkeit der gesamten Anwendung. Je transparenter die Benennungen sind, desto einfacher sind auch neue Begriffe zu durchschauen. Werden neue Benennungen morphologisch moti-

viert gebildet, so ist es relativ einfach, ihre Bedeutung von den Teilen abzuleiten. Werden allgemeinsprachliche Wörter terminologisiert, so ist es stark vom jeweiligen Fall abhängig, ob sie ohne weiteres verstanden werden können. Es ist auch zu beachten, dass die Benennungen angemessen sind, das heißt, es sollten an die neuen Benennungen möglichst keine (negativen) Konnotationen geknüpft sein. Es wird das Beispiel der EDV-Benennungen „master/slave“ genannt, die aus dem Bereich Hydraulik stammen. Hierbei wurde die negative Konnotation der ursprünglichen Begriffe außer Acht gelassen. Im Englischen ist der Gebrauch anerkannt, da er auch im Zusammenhang mit Hydraulik verwendet wird, in anderen Sprachen jedoch wäre eine wörtliche Übersetzung nicht angebracht, da die Verbindung zu diesem Bereich fehlt.

Lassen sich die oben genannten Konzepte auf ausgangssprachliche Terminologearbeit anwenden, so konzentrieren sich Corbolante & Irmeler (2001: 516-520; 522-523) auf die speziellen Eigenheiten der, zumindest zweisprachigen, Terminologearbeit bei Lokalisierung. Ihre Ausführungen knüpfen an die allgemeinen Erläuterungen in Kapitel 3.1.4 an, die sich mit den kulturellen Adaptierungen von Software und deren Benutzeroberfläche befassen. Stil, Register und Tonfall der Texte der Benutzeroberfläche sind Aspekte, die im Rahmen von Lokalisierung angepasst werden müssen. Dies erstreckt sich im Prinzip bis auf die Begriffsebene, weshalb auch bei der Erarbeitung der Terminologie auf die Rezipient/innen geachtet werden muss. Das Locale ist hierbei nicht die unterste Ebene der Anpassung; wie tiefgreifend die Anpassung ist, hängt auch von der Subgruppe ab. Dies bedeutet, dass die Kompetenz der User/innen ausschlaggebend dafür ist, wie ein äquivalenter Terminus in der Zielsprache aussehen sollte. Handelt es sich bei der Zielgruppe zum Beispiel um EDV-Expert/innen, ist es oftmals besser, die ausgangssprachliche (englische) Benennung zu verwenden, wenn diese sich in diesem Umfeld durchgesetzt hat. Für manche neue Begriffe im EDV-Bereich gibt es Benennungen, die sich ohnehin international durchgesetzt haben und deshalb nicht übersetzt werden sollten. Bei der Entlehnung von ausgangssprachlichen Benennungen sind die üblichen Vorgehensweisen zu beachten; ob ein Wort beispielsweise mit der ausgangssprachlichen Rechtschreibung übernommen wird oder an die Grammatik angeglichen wird. Die Angleichung ist besonders dann zu beachten, wenn sich die Schriftsysteme unterscheiden. Es werden bei der Benennungsbildung auch kulturelle Referenzen, vor allem bei Symbolen, erwähnt. An früherer Stelle wurde bereits diskutiert, dass dies mittlerweile als nicht zielführende Praktik kritisiert wird und deshalb weitgehend von vornherein abgelehnt wird.

Nach diesen generellen Überlegungen setzen sich Corbolante & Irmeler (2001: 520-530) mit den konkreten Eigenschaften der Termini im Hinblick auf Übersetzung auseinander. Die erste Gruppe sind Benennungen, die aus der Gemeinsprache stammen. Ihre Übersetzung scheint auf den ersten Blick einfach zu sein, da es meistens Äquivalente in den Zielsprachen gibt. Es gilt hier jedoch, Vorsicht walten zu lassen, da es sich nicht immer um hundertprozentige Äquivalente handelt, sondern unterschiedliche Konnotationen auftreten können. Außerdem kann es sein, dass die ausgangssprachliche Benennungsbildung durch Metaphern in den

Zielsprachen nicht erfolgt ist und deshalb alle anderen Benennungen, die zu einem System gehören, nicht direkt übersetzt werden können. Hier wird das Beispiel der Navigationsstruktur angeführt, bei der im Englischen eine Baummetapher gewählt wurde („tree“, „branch“, „leaf node“ et cetera), im Deutschen „tree“ jedoch mit „Struktur“ übersetzt wurde und somit die einzelnen Bestandteile nicht mit den jeweiligen deutschen Wörtern übersetzt werden können.

Bei der Gruppe der Fachtermini aus anderen Gebieten, die im EDV-Bereich auftauchen, ist die Verwendung der zielsprachlichen Äquivalente weniger risikoreich, da diese ohnehin gut dokumentiert sein sollten. Um die Arbeitszeit zu verringern, ist es hilfreich, den Terminologieverantwortlichen die notwendigen Unterlagen zur Verfügung zu stellen (vgl. auch Esselink 2000: 402).

Im EDV-Bereich werden sehr oft Akronyme und Abkürzungen eingesetzt, vor allem für Hardware-Teile und bei Betriebssystemen. Akronyme zu übersetzen ist meistens nicht sinnvoll, da sich diese oft bereits international durchgesetzt haben, beispielsweise RAM, CPU, DVD, PDF. In manchen Fällen, wie bei den Dateikürzeln verschiedener Formate, sind die Akronyme auch technisch eingebunden und können somit gar nicht übersetzt werden. Wichtiger in diesem Zusammenhang ist die Art und Weise, wie diese Akronyme dem/der User/in erklärt werden. Abkürzungen sind leichter zu übersetzen, wenn auch die entstehenden zielsprachlichen Benennungen an Transparenz einbüßen können.

Die verschiedenen Traditionen der Benennungsbildung können auch zu Problemen führen. So ist es oft unmöglich, für ausgangssprachliche Neologismen ein zielsprachliches Äquivalent zu finden. Sind diese Komposita oder Kofferwörter, ist es meistens nicht möglich, einfach die Teile zu übernehmen. Hier kann auf frühere Entwicklungsschritte zurückgegriffen werden, für den Fall, dass ein Alternativvorschlag existiert, der leichter zu übersetzen ist.

Abgesehen von der Verständlichkeit und der Übermittlung von Wissen werden manche Begriffe auch gezielt eingesetzt, um im Marketing auf besondere Features eines Programms hinzuweisen. So wurde für einige Zeit auf viele Hardware-Komponenten, die „Plug&Play“-fähig waren, dies auf der Packung angepriesen, da es noch kein Merkmal war, über das alle Geräte verfügten. Handelt es sich also um marketingwirksame Termini, ist es wichtig, dass das zielsprachliche Äquivalent auch ansprechend ist (vgl. Corbolante & Irmeler 2001: 522-524).

Eine Herausforderung bezüglich der Äquivalenz ist das Auftauchen von Quasisynonymen in der ausgangssprachlichen Gemeinsprache, die für eine nuancierte Benennungs- und somit auch Begriffsbildung herangezogen werden können. Diese Möglichkeit existiert unter Umständen in den Zielsprachen nicht, weshalb innerhalb der Terminologie nicht zwischen mehreren Begriffen unterschieden werden kann. Dies kann gelöst werden, indem man dieselbe Benennung für mehrere Begriffe verwendet, Erklärungen oder Adjektive hinzufügt oder einen Neologismus bildet.

Wie bereits an anderer Stelle erwähnt, stellen Unterschiede zwischen den Sprachen eine der größten Herausforderungen dar. Dies ist in Bezug auf die Textlänge ersichtlich, sowie bei den grammatikalischen Gegebenheiten, wenn Strings verbunden werden (vgl. Corbolante & Irmeler 2001: 526-530). Darauf wird in Kapitel 5.3 näher eingegangen.

Die Datenkategorien der terminologischen Einträge sind von mehreren Faktoren abhängig, so wie der Organisation und dem Fachgebiet. Neben den üblichen Kategorien, wie Benennung, Definition und Quelle sind bei lokalisierungsbezogener Terminologiearbeit jene Kategorien besonders von Belang, die kontextbezogene Information liefern:

- Stringkategorie: Diese Kategorie enthält Informationen über die Funktion des Terminus in einer Softwareanwendung, dient also als kontextuelle Referenz. Mögliche Datenelemente sind hier „Menüname“, „Befehl“, „Kontrollkästchen“, aber auch „generelle Termini“, wobei jene eingeschlossen sind, die nicht in der Benutzeroberfläche vorkommen, jedoch in der Dokumentation.
- Wortklasse: Diese Kategorie ist keine, die speziell für Lokalisierung relevant ist, sollte jedoch hier der Vollständigkeit halber auch erwähnt werden. Hilfreich sein kann sie trotzdem vor allem bei der Übersetzung von Strings, die aus einem Wort bestehen, aus dem Englischen, da sich Wörter verschiedener Klassen oft formal nicht unterscheiden. Taucht ein Wort in der Terminologie nur als Verb auf, fällt bei einer Übersetzung ins Deutsche zumindest die Möglichkeit weg, dass es sich um ein Nomen handeln könnte.
- Umgebung oder Plattform: Die verschiedenen Hersteller haben, aus unterschiedlichen Gründen, oftmals nicht dieselbe Terminologie. Deshalb ist es notwendig, das Betriebssystem oder sonstige Plattformen (inklusive Version) einzutragen.
- Produktname: Ähnlich wie die Plattform und die Stringkategorie bietet diese Kategorie auch kontextuelle Information.
- Produktversion: Dies ist relevant, da Termini in neueren Versionen verändert werden können.
- Anmerkung: Diese Kategorie ist für zusätzliche Informationen bestimmt, die nicht in die bereits genannten Kategorien passen. So kann es vorteilhaft sein, bei Begriffspaaren, die Gegensätze ausdrücken (zum Beispiel „links/rechts“), den jeweils anderen Begriff anzugeben, vor allem, wenn ein Begriff mehrere gegensätzliche Begriffe hat, die unterschiedlich übersetzt werden müssen.
- Zeit und Datum
- Updates
- Zugriffsschlüssel

Diese Liste ist nicht vollständig, stellt jedoch eine Sammlung der wesentlichen Kategorien dar und kann je nach Anforderung erweitert werden (vgl. Corbolante & Irmeler 2001: 531-532). Grundsätzlich sollten die Kategorien mit den besten verfügbaren Elementen befüllt werden, auch wenn das bedeutet, dass einige leer bleiben. Ein nicht vollständiger Eintrag

kann trotzdem hilfreich sein, ob die enthaltene Information zuverlässig ist, lässt sich am Statusfeld und am Workflowschritt erkennen. Deshalb sollten diese auch immer eingetragen werden. Insgesamt ist hier noch hinzuzufügen, dass der Wert eines Eintrags auch abhängig von der Vollständigkeit beziehungsweise vom Umfang der gesamten Terminologiedatenbank ist, weil Begriffe nicht isoliert existieren, sondern innerhalb von Systemen. Deshalb ist es hilfreich, wenn das Begriffssystem durch Grafiken, Querverweise oder die Datenkategorien ersichtlich ist (vgl. Karsch 2000: 175-177).

Die Quellen für die Einträge im Rahmen eines Lokalisierungsprojektes sind verschiedene. Es sollte ein Projektglossar beziehungsweise eine Projektdatenbank erstellt werden, mit dem während des Projektes gearbeitet wird. Dies ist vor allem für die Kohärenz zwischen den verschiedenen übersetzten Produktbestandteilen notwendig; es kann in der Hilfedatei beispielsweise kein Terminus anders übersetzt werden als in der Benutzeroberfläche, da sie sich ja auf diese bezieht. Quellen, die für die einheitliche Verwendung in der Softwareumgebung sorgen, sind die Glossare der jeweiligen Betriebssysteme. Wurde bereits mit einem Kunden zusammengearbeitet, so ist es ratsam, auf dessen Glossare und Datenbanken zurückzugreifen, um auch hier die einheitliche Verwendung durchzusetzen. Wie bereits beschrieben, gibt es im Rahmen von Lokalisierungsprojekten meistens Termini, die aus anderen Bereichen stammen. Hierfür sollten Ressourcen verwendet werden, die sich in diesen Bereichen durchgesetzt haben (vgl. Esselink 2000: 401-402).

5.2.3.2 Organisation der Terminologiearbeit

Dieses Kapitel beschäftigt sich mit der Organisation von zentral gesteuertem Terminologiemanagement im Unternehmenskontext. Ein besonderer Fokus liegt dabei auf dessen Einführung und auf der Rolle von Terminologiesoftware.

Karsch (vgl. 2007: 177-179) merkt an, dass der kritische Punkt bei der Organisation von Terminologiemanagement nicht die Software oder die Workflows sind, sondern die Unternehmenskultur. Da Terminologie eine Form von explizitem Wissen ist, die das Produkt vom impliziten Wissen der Mitarbeiter/innen ist, ist es fundamental, dass die Unternehmenskultur das Teilen von Wissen fördert. Haben die Mitarbeiter/innen, im Gegenteil, den Eindruck, dass es von Vorteil ist, Wissen zu horten, wird es schwierig sein, eine Terminologiedatenbank zu füllen, also das Wissen für alle zugänglich zu machen. Dieses Verhalten wird beispielsweise durch strikte Hierarchien und wenig Kommunikation zwischen den verschiedenen Abteilungen gefördert. Wird die Einführung von Terminologiemanagement beschlossen, ist es also wichtig, dass das gesamte Unternehmen dahintersteht, da speziell die Erstellung und Weitergabe der Einträge die Mitarbeit aller erfordert. Der Anstoß kann dabei von einer Gruppe im Unternehmen kommen, wahrscheinlich einfacher ist es jedoch, wenn vor allem die Unternehmensführung an Bord ist (vgl. auch Lombard 2006: 168-169; Wittner 2007: 52-

53). Dies ist deshalb von erhöhter Bedeutung, da ja der Aufbau der Infrastruktur und der Prozesse eine Investition darstellt und diesbezüglich Entscheidungen seitens der Führungsetage zu treffen sind.

Im Prinzip geht es darum, ein zentrales Datenbanksystem zu haben, auf das alle zugreifen können, um Einträge zu erstellen und abzurufen. Ferner muss dieses System betreut werden, neben der technischen Seite (Updates, neue Softwareversionen et cetera) sollten Terminologieexpert/innen für den reibungslosen Ablauf sorgen. Das bedeutet, dass ein Eintrag, der neu erstellt wird, von einem/r oder mehreren Terminolog/innen überprüft wird und einen Prozess durchläuft, bis er freigegeben wird. Außerdem sollten auch existierende Einträge überprüft werden, wenn dies nötig ist. Dies wird beispielsweise durch Feedback von Nutzer/innen ersichtlich. Insgesamt sind also etablierte Workflows nötig (vgl. Karsch 2006: 179-187).

Neben den Terminologieexpert/innen stellt das verwendete Tool den zentralen Punkt dar. Es ist zwar möglich, mit Word- oder Excel-Listen zu arbeiten, diese bieten jedoch nicht genügend Funktionen, um effizient größere Datenbanken zu erstellen und zu managen. Erstens ist die Weitergabe mühselig, denn einfache Listen müssen per E-Mail oder auf andere Weise versendet werden, was bei großen Datenbanken oft nicht einfach ist, vor allem wenn Mitarbeiter/innen nur über schlechte Internetverbindungen verfügen. Der Zugriff kann hierbei auch nicht differenziert vergeben werden, da die einzige Möglichkeit ist, die gesamte Liste per Kennwort zu schützen. Außerdem stellt sich die Frage, wie oft die aktualisierte Version verschickt wird; wird innerhalb eines kurzen Zeitraums viel verändert, muss unter Umständen täglich eine neue Liste an alle Nutzer/innen gehen. Zweitens stellt sich die Frage nach der Strukturierung der Liste. Alphabetisch geordnete Glossare können einsprachig noch relativ gut verwaltet werden, doch bereits bei zweisprachigen Listen stellt sich die Frage, wie die Begriffe vernetzt werden. Wird mit mehreren Sprachen gearbeitet, ist eine sinnvolle Gegenüberstellung der Einträge nahezu unmöglich.

Deshalb ist es unumgänglich, Terminologiesoftware zu verwenden, will man effizientes Terminologiemanagement betreiben. Durch den begriffsorientierten Aufbau können Synonyme und Homonyme den richtigen Einträgen zugeordnet werden und auch die Querverweise eingetragen werden. Außerdem können Abkürzungen, Akronyme und unterschiedliche Schreibweisen in einen Eintrag inkludiert werden. Die begriffsorientierte Struktur macht es auch möglich, die jeweiligen äquivalenten Einträge in verschiedenen Sprachen zu verknüpfen, sodass nicht unabsichtlich ein Homonym gewählt wird, wie es bei einer Strukturierung nach Lemma vorkommen kann. Ein weiterer Vorteil ist die Kapazität der Einträge selbst. Nicht nur grammatikalische Angaben, Kontextinformationen und andere Datenkategorien erleichtern das Verstehen des Begriffs, auch bei der Definition kann auf verschiedene Möglichkeiten zurückgegriffen werden, wie zum Beispiel Abbildungen und Grafiken des Begriffssystems. Terminologie-Tools erlauben außerdem verschiedene Suchvarianten, das heißt, es kann von jeder Sprache ausgehend ein Begriff gesucht werden und auch unscharfe Suchen,

Platzhaltersuchen oder Suchen in mehreren Datenbanken führen zu Ergebnissen, während in einer Wordliste ein Wort so gesucht werden muss, wie es geschrieben ist.

Nicht zuletzt die Vernetzungsmöglichkeiten von serverbasierter Terminologiesoftware bieten einen erheblichen Vorteil. Im momentan vorherrschenden outgesourcten Setting gibt es eine Vielzahl an Personen an verschiedenen Orten, die Zugriff brauchen. Die Zugriffsrechte können genau geregelt werden, außerdem wird sichergestellt, dass alle, egal von wo sie zugreifen, immer alle aktuellen Einträge finden können. Durch offene, meistens XML-basierte Standards ist es möglich, bereits existierende Terminologiebestände zu einer Datenbank hinzuzufügen beziehungsweise benötigte Einträge zu exportieren. Die Integration der Terminologiesoftware in andere Workflow-Tools macht die Arbeit insgesamt effizienter (vgl. Fidura 2007: 39-41). Kapitel 6.2 beschäftigt sich näher mit Terminologie-Tools.

Massion (2007: 47-50) und Dunne (2007: 37-38) beschäftigen sich mit der Neueinführung von zentral gesteuertem Terminologiemanagement. Grundsätzlich gilt hier der Ansatz, langsam und überlegt vorzugehen. Es ist ratsam, klein anzufangen, trotz des Ziels, schließlich ein unternehmensweites System aufzubauen. So kann ein Pilotprojekt in einem bestimmten Bereich begonnen werden, zum Beispiel eine Datenbank mit Schlüsselterminologie der größten Kunden, um nicht zuletzt den Entscheidungsträger/innen die Effizienz vorzuführen. Es sollte grundsätzlich vor dem Beginn des Aufbaus das Ausmaß und die Ziele definiert werden und abgewogen werden, welche Prozesse und Tools notwendig sind, um diese zu erreichen. Beginnt man in kleinerem Rahmen, kann dieser immer erweitert werden, umgekehrt ist es nicht ratsam, Strukturen aufzubauen, die schließlich das Ziel übersteigen. Vor der Auswahl der Terminologiesoftware sollten die Prozesse und Workflows definiert werden, sodass diese von dem gewählten Programm optimal unterstützt werden, dazu gehört auch, festzulegen, wer welche Zugriffsrechte bekommt.

Abschließend kann festgestellt werden, dass Terminologearbeit innerhalb des Lokalisierungsgefüges ein wichtiger Bestandteil ist, dessen Vernachlässigung zu erhöhten Kosten führt und weitere negative Folgen haben kann. Dabei geht die Arbeit über ein Lokalisierungsprojekt hinaus, nur die optimale Vernetzung aller Beteiligten, von den Programmierer/innen bis zu den Übersetzer/innen, führt zu den besten Ergebnissen. Deshalb ist auch logisch, dass Terminologearbeit zentral gesteuert werden muss, um die negativen Folgen von isolierten Arbeitsschritten zu vermeiden.

5.3 Übersetzung

Wie man es betrachtet, im Gefüge GILT oder im Rahmen eines Lokalisierungsprojektes, ist Übersetzung nur ein Teil der Lokalisierung, wenn auch ein wichtiger. Seit den Anfängen in den Achtzigerjahren, als die Übersetzung der Benutzeroberfläche neben der Anpassung der

Eingabe das einzige war, was gemacht wurde, hat sich der Fokus stark verschoben und andere Bereiche werden stärker beobachtet und weiterentwickelt. Dies ist wahrscheinlich darauf zurückzuführen, dass es in diesen Bereichen mehr zu tun gab; auf der technischen Seite, zum Beispiel, die Weiterentwicklung der CAT-Tools, im Bereich Projektmanagement die Optimierung der Abläufe. Übersetzung stellt gewissermaßen einen Fixpunkt dar, bei dem eher die damit verbundenen Aspekte, wie Terminologie oder die Beschäftigungsverhältnisse der Übersetzer/innen, abgehandelt werden. Da diese auch in dieser Arbeit als eigene Kapitel auftreten, soll hier der Fokus auf der eigentlichen Übersetzungsarbeit liegen: was übersetzt wird, wie die Übersetzungsarbeit einzuteilen ist und worauf dabei zu achten ist.

5.3.1 Software

In Kapitel 3.1.3 wurde bereits genannt, welche die typischen Bestandteile einer Anwendung sind, die im Zuge eines Lokalisierungsprojektes übersetzt und bearbeitet werden. Es handelt sich dabei grundsätzlich um die Softwaredateien selbst, die Hilfedatei und die Dokumentation. Die Software sollte dabei grundsätzlich zuerst übersetzt werden, da Hilfe und Dokumentation sich darauf beziehen. Im Lokalisierungsprojekt kommt die Übersetzung idealerweise nach der Erstellung der Terminologiedatenbank und wird zumeist von freiberuflichen Übersetzer/innen angefertigt, die die benötigten Dateien als Translation-Kit bekommen.

Es gibt eine sehr große Bandbreite bei der Größe von Anwendungen für Endkund/innen; von einigen Megabytes bis hin zu vielen Gigabytes. Im Prinzip enthalten die verschiedenen Dateien Informationen darüber, wie Eingabe (durch Maus, Tastatur, Touchscreen, natürliche Sprache oder Bewegung) interpretiert werden soll und welche Ausgabe daraufhin erfolgt (über den Bildschirm, Lautsprecher oder, zum Beispiel bei Smartphones, Vibrationen). Bei der Entwicklung eines Programms werden Quellcode und Ressourcendateien erzeugt. Der Quellcode enthält das eigentliche Programm in einer Programmiersprache, die natürlich nicht übersetzt werden darf, die Ressourcendateien enthalten unterschiedlichste Informationen, auf die bei der Ausführung des Programms zurückgegriffen werden. Dazu gehören auch die für die Lokalisierung relevanten Bestandteile der grafischen Benutzeroberfläche. Um eine Anwendung ausführbar zu machen, müssen diese Dateien in die Maschinensprache kompiliert oder interpretiert werden, sodass die Hardware diese „versteht“. Diese Dateien bestehen dann im Prinzip nur mehr aus 0 und 1 und sind für Menschen nicht mehr verständlich. Bei Windows-Programmen sind die ausführbaren Dateien, die aus dem Programmcode kompiliert werden, meistens jene mit der Endung .exe, andere Binärdateien haben die Endung .dll (dynamische Programmbibliotheken) oder sind als proprietäre Formate gespeichert. Letztere enthalten zumeist Ressourcen (vgl. Esselink 2000: 58-59).

Zur Übersetzung der grafischen Benutzeroberfläche können die Binärdateien dekompiert werden und die Ressourcen, die normalerweise als Textdateien gespeichert werden,

geöffnet werden. Dies ist mit einem einfachen Texteditor oder einem Ressourceneditor möglich, mit einem Translation Memory-System zu arbeiten birgt jedoch gewisse Vorteile. Auch in Ressourcendateien steht vorrangig Code, in den die verschiedenen Ausgabetexte eingebettet sind. Prinzipiell unterscheidet man hier Dialogfelder, Menüs und Strings. Außerdem sind in Ressourcendateien Versionsinformationen und Kommentarzeilen enthalten.

Dialogfelder sind jene Fenster, die in einem Programm geöffnet werden können und die Optionen und Einstellungen enthalten. Ein Beispiel ist das Fenster „Absatz“, das im Office-Programm Word geöffnet werden kann und das Einstellungen zum Textformat innerhalb eines Absatzes enthält. Der Code gibt dem Programm die Informationen zum Dialogfenster, also wie es zu öffnen ist, wie groß es ist, der Font, der verwendet wird, die verschiedenen Auswahlmöglichkeiten et cetera. Der Text, der zu übersetzen ist, ist mit Tags gekennzeichnet. Meistens sind es Anführungszeichen, dies ist jedoch von der Programmiersprache abhängig.

Menüs sind jene Drop-Down-Listen, die beim Klick auf einen Button ausgeklappt werden. Ein Beispiel sind die klassischen Elemente der Windows-Menüleiste „Datei“, „Bearbeiten“, „Anzeige“. Ein Menü enthält wiederum Buttons, die eine Aktivität ausführen können oder ein Dialogfenster öffnen. In der Windows-Umgebung sind die letzteren mit drei Punkten gekennzeichnet.

Strings sind schließlich jene Elemente, die an verschiedenen Stellen im ausgeführten Programm angezeigt werden. Es handelt sich um Zeichenketten, also meistens Wörter oder Sätze, die Fehlermeldungen, Statusmeldungen, Fragen und Kurzinfos enthalten. Diese können fix sein oder variabel und werden an verschiedenen Stellen in Dialogfeldern, Menüs oder Buttons eingefügt. Diese Kategorie ist üblicherweise jene mit dem größten Umfang und auch die, die am schwierigsten zu übersetzen ist, da Strings meistens in Listen stehen und keine Kontextinformationen bieten.

In Windows-Ressourcendateien gibt es zumeist einen Versionsstempel, der über das Programm Auskunft gibt, also den Namen des Unternehmens und des Programms, Copyright-Informationen, die Versionsnummer und die Sprache beziehungsweise das Locale enthält. Übersetzt wird hier wenig, zumeist lediglich der Name des Programms (wenn sich dieser in der Zielsprache unterscheidet) und die Copyright-Information. Natürlich muss auch der Wert verändert werden, der die Sprachversion angibt.

Kommentarzeilen sind nicht zu übersetzen, jedoch für Übersetzer/innen deshalb relevant, weil sie Informationen für diese enthalten können. Diese Zeilen sind auch getagt und enthalten beispielsweise Informationen zum Kontext oder zur Maximallänge von Strings.

Wird für die Übersetzung eine Datei in einem Texteditor geöffnet, sind also nicht nur die zu übersetzenden Texte, sondern auch der Code zu sehen. Der Code darf auf keinen Fall verändert werden, weil dies zu Fehlern bei der Ausführung des wieder kompilierten Programms führt, zu sogenannten Bugs. Wie oben beschrieben, sind die relevanten Texte an den Anführungszeichen oder anderen Tags zu erkennen. Anführungszeichen sind deshalb bei

Windows-Programmen im Text selbst nicht zu verwenden, genauso wie andere Codeelemente, die eigene Funktionen haben, wie das „&“, das für Hot-Keys benutzt wird. Sind Datums-, Zeit- und Zahlen in Formaten angegeben, die in der Ausgangssprache üblich sind, sollten diese auch nicht verändert werden, da deren Anpassung bei der Ausgabe im Programm meistens über andere Mechanismen erfolgt. Grundsätzlich sollten diese Informationen ohnehin als Code oder in Form von Platzhaltern angegeben sein, sodass der Computer bei der Ausführung die richtigen Formate wählt.

Wegen der Gefahr, unabsichtlich einen Teil des Codes zu löschen, ist die Verwendung von Translation Memory-Tools von Vorteil, da mithilfe von Filtern alle nicht zu übersetzenden Teile geschützt werden können. Das Translation Memory-Programm erkennt die Tags und zeigt nur die relevanten Texte in den Segmenten an. Eine andere Möglichkeit ist die Öffnung der Binärdateien selbst mit einem Ressourceneditor oder einem Lokalisierungstool. Bei letzteren wird auch der Code geschützt und außerdem wird die Benutzeroberfläche direkt angezeigt. Dadurch ist erstens ein Kontext gegeben, das heißt, für die Übersetzer/innen ist ersichtlich, wie ein Element aussieht und zweitens kann die Größe von Dialogfeldern und Buttons angepasst werden, um genügend Platz für Übersetzungen zu haben, die länger sind (vgl. Esselink 2000: 60-65; 76).

Was ist nun bei der Übersetzung zu beachten? Neben dem vorsichtigen Umgang mit Codes nennt Esselink (2000: 66-73) sprachliche und stilistische Aspekte und einige andere Faktoren, die eine Rolle spielen.

Auf Terminologie wurde bereits im letzten Kapitel eingegangen, diese spielt nicht zuletzt aufgrund der Kürze der Meldungen eine große Rolle. Auch stilistisch sollten die zielsprachlichen Texte eine gewisse Kohärenz aufweisen. Würde man beispielsweise statt „Möchten Sie die an Dokument_1 vorgenommenen Änderungen speichern?“, „Willst du speichern, was du in Dokument_1 verändert hast?“ schreiben, würde dies die Nutzer/innen irritieren. Hierzu gibt es in jeder Sprache beziehungsweise Locale unterschiedliche Konventionen. So ist es im Englischen üblich, einen relativ lockeren Stil anzuwenden, wie im Beispiel: „Congratulations, you have successfully installed this application“, was in anderen Sprachen nicht erwartet werden würde. Auch innerhalb eines Programms sollte die Kohärenz der Übersetzungen gewahrt werden, also ein Dialogfeld, das über den Button „Speichern unter“ geöffnet wird, sollte keinen anderen Namen tragen. Grundsätzlich ist auf Klarheit der Botschaften zu achten, dies gilt nicht nur für die Formulierung, sondern auch für Grammatik und Rechtschreibung. Welche Folgen inkohärente Terminologie hat, wurde im vorigen Kapitel erläutert, im weiteren Sinne ist dies für die gesamte Sprachausgabe gültig.

Eine Herausforderung stellt die Größe der Felder dar, die Texte enthalten, vor allem in Kombination mit der üblichen Ausdehnung der zielsprachlichen Texte im Vergleich mit dem Ausgangstext. Diese können bis zu 30 % länger sein. Deshalb werden die Felder nach der Übersetzung üblicherweise angepasst oder dies durch den/die Übersetzer/in selbst vorgenommen, wenn diese/r ein Lokalisierungstool verwendet. Es kann jedoch sein, dass die Grö-

ße nicht verändert werden kann, was dazu führt, dass der/die Übersetzer/in Kreativität beim Formulieren an den Tag legen muss und/oder Abkürzungen verwenden. Hierbei ist wiederum auf die durchgängige Verwendung einer Abkürzung im gesamten Programm zu achten (vgl. Esselink 2000: 59-67). Ein besonderer Bereich ist hier die Lokalisierung von Gerätesoftware, wenn es Ausgabetexte auf kleinen Bildschirmen oder Anzeigen gibt. Hier kann die Größe natürlich nicht verändert werden und somit ist die Kreativität der Übersetzer/innen sowie eine gute Zusammenarbeit mit den Entwickler/innen gefragt (vgl. Massion 2011: 41).

Noch ein Beispiel für Codes, die im Text vorkommen, sind Platzhalter oder Variable. Diese werden, bei Windows-Programmen, mit dem Prozentzeichen und einem Buchstaben dargestellt, so steht, zum Beispiel, „%s“ für einen String, der während der Ausführung eingefügt wird. Die Variablen selbst dürfen natürlich nicht verändert werden, auch für die Formulierung selbst bedeutet dies eine Herausforderung. Die englische Sprache, die meistens die Ausgangssprache ist, ist für die Darstellung von Variablen vorteilhaft, da die Grammatik Verbindungen gut zulässt. Alle Objekte haben, zum Beispiel, dasselbe Geschlecht und die Pluralformen sind (fast immer) auch gleich. Deshalb ist es kein Problem, wenn die Mehrzahlform eines Hauptwortes angezeigt werden muss, da nur ein „s“ angehängt werden muss. Im Deutschen sind jedoch Geschlecht, Zahl und Fall morphologisch gekennzeichnet, was sich auch im Satz ausdrückt, und deshalb gilt es, unterschiedliche Lösungen zu finden. Bei „Copying %d file%s.“ oder „Copying %d folder%s.“ wird der Wert „%d“ durch eine Zahl ersetzt, je nachdem, wie viele Dateien oder Ordner kopiert werden. Sind es mehr als 1, wird das „s“ angehängt. Im Deutschen ist dies nicht möglich, da der Plural von Datei mit dem Suffix „en“ gebildet wird, der Plural von Ordner jedoch formgleich mit dem Singular ist.

Unter den Elementen, die nicht gelöscht beziehungsweise verändert werden dürfen, befinden sich noch Formatierungscodes, wie jene, die Umbrüche anzeigen oder Links zu anderen Dokumenten enthalten. Es darf, wie angegeben, auch nichts übersetzt werden, was sich außerhalb der Tags befindet.

Hot-Keys und Tastenkombinationen sind Elemente, bei denen Vorsicht geboten ist. Hot-Keys sind Tasten, die innerhalb eines Dialogfeldes oder Menüs unterstrichen sind und anstatt eines Mausklicks direkt einen Button aufrufen. In der Ressourcendatei sind sie durch das „&“ vor dem verwendeten Zeichen gekennzeichnet. Hierbei ist darauf zu achten, dass innerhalb eines Dialogfeldes beziehungsweise Menüs nicht zweimal dasselbe Zeichen zugewiesen wird. Da Hot-Keys unterstrichen sind, sollten keine Buchstaben verwendet werden, die Unterlänge haben, außerdem keine, die Sonderzeichen sind, da eine spezifische Tastatur zum Aufrufen notwendig wäre. Hot-Keys sind für Übersetzer/innen deshalb relevant, weil sich die Befehle in den Sprachen unterscheiden und somit der in der Ausgangssprache verwendete Buchstabe in der Zielsprache unter Umständen nicht im Wort enthalten ist, obwohl grundsätzlich auf Einheitlichkeit innerhalb eines Betriebssystems Wert gelegt wird. Außerdem sind manche Zeichen auf verschiedenen Tastaturen nicht einfach zu erreichen, so wie,

zum Beispiel, die Zahlen auf der französischen Tastatur. Für nicht-lateinische Schriften wird der Hot-Key meist extra in Klammer angegeben.

Tastenkombinationen sind, im Gegensatz zu Hot-Keys, für das gesamte Programm festgelegt. Übliche Tastenkombinationen sind „Strg + P“ für Drucken, „Strg + C“ für Kopieren und „Strg + V“ für Einfügen. Die Anweisungen zu den Hot-Keys sind grundsätzlich auch hier gültig, es ist anzumerken, dass die üblichen Tastenkombinationen in den meisten Programmen einer Plattform durchgängig verwendet werden.

Neben der Übersetzung von Strings, die ohne Kontext in einer Liste stehen, stellen verbundene Strings (concatenated strings) eine besondere Herausforderung dar. Hier werden zwei oder mehrere Strings zu komplexeren Meldungen verknüpft. Das Problem, das zuvor bei Variablen geschildert wurde, tritt hier noch stärker auf (vgl. Esselink 2000: 68-73). Als anschauliches Beispiel dient die automatische Generierung von Meldungen in einem Online-Spiel (vgl. Arthur et al 2010: 45). In dem Rollenspiel, das in der Fallstudie untersucht wird, werden viele Meldungen automatisch generiert, weil es einfacher und kostengünstiger ist, als alle möglichen Meldungen durch die Programmierer/innen verfassen zu lassen. Darunter fällt, zum Beispiel, die Nachricht, wenn der Spielcharakter einen Gegenstand gefunden hat: „You have found an ancient Kovalli copper bracelet“, wobei sich diese zusammensetzt aus den Strings „You have found“, „an“, „ancient“, „Kovalli“, „copper“ und „bracelet“. Der erste Teil ist das Grundelement dieser Nachricht, der Artikel ist abhängig vom nächsten Anfangsbuchstaben und der Rest, der Gegenstand, wird generiert. Es gibt beliebig verschiedene Möglichkeiten, die Grundstruktur ist jedoch „Alter“, „Material“, „Quelle“, „Name des Gegenstandes“. Im Englischen funktioniert dies ohne Probleme, da es egal ist, ob es sich um ein „bracelet“, „hat“, „crown“, „gauntlets“ oder sonstiges handelt. Die einzige Anpassung muss am Artikel vorgenommen werden; hier gibt es jedoch auch nur drei Möglichkeiten: „a“, „an“ oder gar nichts, im Falle einer Mehrzahlform. In anderen Sprachen, wie Deutsch oder Spanisch, werden Attribute jedoch dekliniert, weshalb die komplette Struktur der Generierung verändert werden muss. Für „ancient“ gibt es, je nach Geschlecht und Zahl des Gegenstandes, die Übersetzungsmöglichkeiten „alten“, „alte“ und „altes“ beziehungsweise „anciano“, „anciana“, „ancianos“ und „ancianas“. Im Falle des Spiels müssen die Strings also extra gekennzeichnet werden, um die richtige Variante zu erhalten, bei anderen Meldungen ist es meistens einfacher, alternative Lösungen zu finden.

Die angeführten Aspekte sind natürlich nicht alle Herausforderungen, die im Rahmen der Übersetzung der Elemente der Benutzeroberfläche einer Anwendung auftauchen können. Grundsätzlich ist es hilfreich, wenn nicht sogar notwendig, dass sich Übersetzer/innen der Beschaffenheit dieser Bestandteile und des damit verbundenen Codes bewusst sind. Auch Stolze (1999: 160) meint, dass Übersetzer/innen mit der Computerumgebung und der Struktur von Anwendungen vertraut sein müssen, um gute Ergebnisse zu erzielen. Je tiefgehender die Kenntnisse von Programmstrukturen und Codes sind, desto besser findet man sich zurecht und kann auch ohne direkte Kontextangaben mit den zu übersetzenden Texten umgehen. Da

es sich nicht um Fließtexte handelt, sondern gewissermaßen um eine „zerrissene“ Textform sind auch die Informationen aus dem Lokalisierungs- beziehungsweise Translation-Kit besonders wertvoll. Zugriff auf Terminologiedatenbanken und sonstige Referenzmaterialien sowie die Vernetzung mit den übrigen Projektbeteiligten stellen eigentlich eine Grundvoraussetzung dar, was jedoch nicht immer so einfach ist. Übersetzer/innen sind oftmals über mehrere Sprachdienstleister oder Agenturen beauftragt und arbeiten teilweise an nicht gut vernetzten Orten überall am Globus. Deshalb ist auch verständlich, dass sich viele Autor/innen und Forscher/innen eher mit diesen Aspekten befassen als mit der Übersetzung selbst.

5.3.2 Hilfedatei

Die Programmhilfe oder Hilfedatei ist, wie schon der Name ausdrückt, zur Unterstützung der User/innen da. Der Name Online-Hilfe, analog zu „online help“ wird im Deutschen nicht so gebraucht, da es mittlerweile verschiedene Wege gibt, diese Art des Supports zur Verfügung zu stellen. Das heißt, Online-Hilfe bedeutet wirklich ein Hilfesystem, das online abgerufen werden kann, im Gegensatz zur Offline-Hilfe, die ein „klassisches“ Dokument ist, das im Produktumfang inkludiert ist. Es gibt bei der Online-Lösung die Möglichkeit, die Hilfe innerhalb des Programms aufzurufen, wobei die Inhalte direkt heruntergeladen werden, oder es wird ein Link in einem Browserfenster geöffnet. Bei Office-2010-Produkten oder bei LibreOffice ist die aufgerufene Hilfe beispielsweise eine Mischung, die eigene Inhalte enthält, zusätzliche hingegen abrufen (vgl. Microsoft Office-Support 2014; LibreOffice Versionshinweise 2014). Die Hilfe kann teilweise auch kontext-sensitiv sein, das heißt, sie ist in jedem Dialogfeld aufrufbar beziehungsweise erscheinen Instruktionen und Erklärungen als Kurzinfo. Das Prinzip ist jedoch immer dasselbe, es handelt sich um Informationen, die in Themen gruppiert und als Hypertext organisiert sind. Um alle Möglichkeiten einzuschließen wird hier von der Hilfedatei gesprochen.

Dieser Produktbestandteil hat umfassende gedruckte Handbücher schon vor längerer Zeit abgelöst und stellt üblicherweise das größte Übersetzungsvolumen dar, da hier größere Mengen Fließtext enthalten sind. Die Vorteile gegenüber einem Handbuch liegen auf der Hand: Erstens ist es die Möglichkeit, die Hilfedatei direkt aus dem Programm aufzurufen, wenn die Informationen benötigt werden beziehungsweise Erklärungen als Kurzinfo zu erhalten. Der zweite Vorteil ist die Hypertext-Struktur, die es den User/innen erlaubt, von einem relevanten Thema mittels Link zu einem anderen zu wechseln, das danach benötigt wird. Es können zudem hilfreiche Grafiken und Pop-up-Fenster oder Multimedia-Dateien aufgerufen werden; alles Dinge, die in einem gedruckten Handbuch nicht möglich sind. Drittens ist die Aktualisierung der Hilfedatei einfach: Online-Hilfen können grundsätzlich immer upgedatet werden, Updates für Offline-Hilfen auch heruntergeladen. Viertens verfügen Hilfedateien über Suchfunktionen, die viel schneller zu Ergebnissen führen als das Herumblättern in ei-

nem Druckwerk und schließlich ersparen sich Softwareunternehmen den Kostenaufwand für eine Druckversion (vgl. Esselink 2000: 165-166).

Es gibt nun verschiedene Dateiformate, in denen Hilfedateien entwickelt und gespeichert sind. Seitens Microsoft gab es lange Zeit das Format WinHelp mit der Endung .hlp, das von HTMLHelp mit der Endung .chm abgelöst wurde. Seit Windows Vista gibt es wiederum eine neue Hilfeplattform, genannt „Assistance Platform Help“ oder AP Help. Die meisten Hilfedateien sind HTML-basiert, wie der Standard für alle Windows-Plattformen, Microsoft HTML Help 1.4, JavaHelp oder Apple Help (vgl. Windows Help 2014; Techscribe 2014; Mac Developer Library 2014). Ungeachtet des Formats sind die Materialien, die in der Hilfedatei angezeigt werden, auch in Ressourcendateien gespeichert. Dies sind der Text und Formatierungsinformationen, gegebenenfalls Bilddateien, zum Beispiel in den Formaten GIF oder JPG, Multimediadateien und Layout-Dateien, wie zum Beispiel CSS-Dateien. Auch hier muss darauf geachtet werden, nicht versehentlich Codes zu löschen. Falls sich Formatierungen und Layout ändern oder Bildmaterial angepasst werden muss, wird dies nachträglich gemacht.

Obwohl es sich meist um längere Texte handelt als bei der Übersetzung von Strings, werden in der Hilfedatei normalerweise nur jene Absätze angezeigt, die ein gewisses Thema beschreiben. Dies bedeutet, dass man nicht davon ausgehen kann, dass der/die User/in zuvor ein anderes Hilfethema gelesen hat. Stilistisch ist hier, wie bei der Software selbst, auch auf klare Formulierungen zu achten, ist der Zweck der Hilfedatei ja jener, den User/innen Softwarefeatures zu erklären und nicht, sie noch weiter zu verwirren. Wegen des Bezugs auf die Features ist die genaue Verwendung der Terminologie grundlegend. Auch innerhalb der Hilfedatei ist auf Kohärenz zu achten, da die Verlinkungen mithilfe von Schlüsselwörtern gemacht werden, die dann auch in einem Index aufgelistet sind. Indizes und Inhaltsverzeichnisse werden normalerweise nach der Übersetzung automatisch generiert, weshalb nachträgliche Veränderungen im Text nicht ohne weitere Überprüfung dieser Bestandteile vorgenommen werden sollten.

Übersetzung, Überprüfung und Freigabe der Software sollten im Prinzip vor Bearbeitung der Hilfedatei abgeschlossen sein. Erstens, weil die Terminologie der Software sich wegen der Verweise nicht mehr verändern sollte und zweitens, weil der/die Übersetzer/in dann außerdem die Software auf einem anderen Gerät ausführen kann und somit den Kontext vor sich hat. Die Reihenfolge einzuhalten ist jedoch aufgrund der zeitlichen Einteilung nicht immer möglich (vgl. Esselink 2000: 166-172).

5.3.3 Dokumentation

Die Dokumentation umfasst verschiedenste Dokumente, wie Installationsanleitungen, Einstiegshilfen und Handbücher, sowie sonstiges Material, zum Beispiel Schnellreferenzkarten,

Marketingmaterial, Datenträgeraufdrucke und die Verpackung. Die Texte für die Übersetzung sind in vielen verschiedenen Formaten gespeichert, so wie DTP-Formate, PDF-Dokumente et cetera. Die übersetzten Dokumente müssen normalerweise von DTP-Expert/innen nachbearbeitet werden, die das Layout überprüfen und gegebenenfalls anpassen. Dies ergibt sich vor allem dann, wenn die Endformate konvertiert werden müssen, um sie in einem Translation Memory-Programm öffnen zu können. Außerdem unterscheidet sich der benötigte Platz von Ausgangs- und Zieltext oft erheblich, vor allem wenn nicht-lateinische Schriften in der Zielsprache verwendet werden. Handelt es sich um eine Schrift, die von rechts nach links verläuft, muss das gesamte Layout an diese Richtung angepasst werden (vgl. Esselink 2000: 289-295). Auch die Dokumentation hängt sehr stark mit der Software zusammen, weswegen es hier wichtig ist, die Anwendung als Referenz zu haben.

Bernal Merino (2008: 33) nennt im Zusammenhang mit der Lokalisierung von Spielen die Produktwebsite. Diese taucht zwar sonst nicht im Zusammenhang mit Dokumentation auf, durch die Verbindung mit dem Produkt ist jedoch eine Nutzung der Ressourcen (Translation Memorys, Terminologiedatenbanken et cetera) für die Lokalisierung der Website sinnvoll.

5.3.4 Stil und Locales

Schon an der Menge der lokalisierten Software lässt sich ableiten, dass die Texte die auf den virtuellen Schreibtischen der Übersetzer/innen landen ein großes Spektrum abdecken. Bereits in Kapitel 5.2 wurde, in Bezug auf Terminologie, erwähnt, dass die Bandbreite an Texten nahezu unbegrenzt ist, da sich der Computer in alle Bereiche ausgebreitet hat und somit Software für alle erdenklichen Zwecke existiert. Dies hat auch bei der Übersetzung Auswirkungen auf den Stil und nicht zuletzt Recherche und Referenzmaterial. Es ist ein großer Unterschied, ob Anwendungen im medizinischen Bereich oder in der Buchhaltung eingesetzt werden. Hier gilt es, Referenzen aus der „echten“ Welt hinzuzuziehen, um qualitativ hochwertige Übersetzungen im Fachkontext zu erstellen.

An dieser Stelle soll noch einmal auf Locales verwiesen werden. Diese Kombination von Sprache und Land beziehungsweise Region hat für die lokalisierte Software nicht nur Veränderungen in Bezug auf Formate, Schriften, Vorschriften und kulturelle Inhalte zu Folge, sondern erfordert auch eine Auseinandersetzung mit sprachlichen Varietäten. Es kann also vorkommen, dass Texte einer US-amerikanischen Anwendung auch für Großbritannien bearbeitet werden (vgl. Safar 2014: 31-35). Dies ist zugegeben eher für Online-Inhalte oder Inhalte mit Marketing-Zwecken von Belang als für Strings, jedoch zählen zur Dokumentation eines Softwarepakets auch Verpackung und Datenträgeraufdrucke. Es ist hier noch anzumerken, dass die Beschäftigung mit verschiedenen Locales oft vernachlässigt wird. Im Falle von Spanisch gäbe es, aufgrund der großen Zahl an Sprecher/innen oft doch Bedarf an Anpassung

an das Locale. Oft gibt es, neben dem europäischen Spanisch, lediglich ein Locale für latein-amerikanisches Spanisch oder internationales Spanisch auszuwählen (vgl. Ryan et al 2009: 20; Jiménez-Crespo 2010: 13-14).

5.3.5 Videospiele und Übersetzung oder *All your base are belong to us*

Videospiele sind interaktive audio-visuelle Produkte, deren Wurzeln bis in die Fünfzigerjahre zurückreichen und die ab den Siebzigern weite Verbreitung fanden. Auch wenn sie oft kritisiert werden, lässt sich nicht leugnen, dass es sich um einen großen Industriezweig handelt. Die weltweiten Umsätze erreichten im Jahr 2012 eine Höhe von fast 79 Milliarden Dollar, bis 2015 wird mit einem Anstieg auf 111 Milliarden Dollar gerechnet (vgl. Gartner 2013). Der Titel „Grand Theft Auto 5“ erreichte in den ersten drei Tagen einen Rekord-Umsatz von einer Milliarde Dollar und mittlerweile wurden 33 Millionen Einheiten verkauft (vgl. WinFuture 2013; Forbes GTA 5 2014). Davon sind nicht wenige lokalisierte Ausgaben. Auch wenn man das Medienecho beobachtet, mit Vorschauen, Leaks und Spekulationen über weitere Spiele einer Reihe, lässt sich nicht bestreiten, dass Videospiele den Eingang in viele Kulturen gefunden haben und im Prinzip zu einer eigenen Kunstform geworden sind. Die Computer- und Videospielindustrie ist vor allem in den USA und Japan entstanden und auch hier hat sich Lokalisierung von einem rudimentären Übersetzungsansatz zu einem hochprofessionellen Bereich entwickelt. Das Zitat in der Kapitelüberschrift ist das Ergebnis der Übersetzung des japanischen Arcade-Spiels „Zero Wing“ aus dem Jahr 1989 (vgl. Arcade Museum 2014). Der Satz, der auch im Spiel keinen Sinn macht, ist mittlerweile legendär geworden und geistert gemeinsam mit anderen kuriosen Übersetzungsfehlern im Internet herum. Heutzutage wird viel mehr in Lokalisierung und Übersetzung investiert, wie der Aufgabenbereich von Ben Bateman (2012: 32-34) zeigt, der für eine finale Bearbeitung der Übersetzungen von Spielen aus dem Japanischen zuständig ist, um die Charaktere, die Spielwelt und die Geschichten so rund wie möglich zu gestalten.

Lässt sich bei vielen Softwareanwendungen eine klare Verbindung zu einem Fachgebiet ziehen, ist es bei der Lokalisierung von Spielen gar nicht unähnlich. Das Fachwissen und –vokabular stammt jedoch aus einem vollkommen anderen Bereich. Unter den verschiedenen Genres, von Ego-Shootern über Jump’n’Run bis zu Strategiespielen, gibt es eine unermesslich hohe Anzahl von Konzepten und Spielwelten. Manche Spielwelten sind der Realität nachempfunden, von ernst gemeinten bis zu satirischen Varianten, manche Welten sind komplette Fantasieprodukte, manche Spiele haben eine grundsätzlich andere Struktur, so wie Puzzle- und Ratespiele. Viele Spiele bilden die Realität ab, wie zum Beispiel Mannschaftssportspiele, bei denen die echten Mannschaften und Ligen nachgebildet sind. Oft gibt es auch Spiele, die bereits vorhandene, aus Filmen und Büchern bekannte Fantasiewelten nutzen, wie zum Beispiel jene aus „Star Wars“ oder „Harry Potter“. Es stecken in jedem Fall viele kultu-

relle Hintergründe in diesen Produkten. Deshalb ist das (sub-)kulturspezifische Wissen der Übersetzer/innen gefragt (vgl. Edwards 2014: 14-15). Neben den einzelnen Welten, auf die sich Spiele beziehen, müssen Übersetzer/innen auch über genrespezifisches Wissen verfügen. Es gibt vielerlei Begriffe, die sich im Spielbereich etabliert haben, wie „deathmatch“ oder „cooperative multiplayer“, bis hin zu den einfachen Befehlen in den Menüs und beispielsweise den Grafikeinstellungen. Dies bedeutet, dass die Geringschätzung, die manchmal beobachtet wird – es handle sich ja nur um Spiele – nicht gerechtfertigt ist (vgl. Dietz 2006: 126). Mit dem Inhalt von Spielen müssen sich Übersetzer/innen und andere Beteiligte genauso auseinandersetzen und die Terminologie managen, wie es bei Programmen zu anderen Fachgebieten der Fall ist (vgl. Lutsuk 2014: 28).

Eine Schwierigkeit, die bei der Übersetzung der Benutzeroberfläche eines Videospiels verstärkt auftritt, ist der fehlende Kontext. Mithilfe von Lokalisierungstools lässt sich die Oberfläche der meisten anderen Programmen darstellen, bei Spielen ist das bisher nicht möglich. Auch wenn es sehr kleine Spiele mit wenigen Ausgabetexten gibt, wie zum Beispiel Puzzle- und Ratespiele, sind manche mittlerweile so große, komplexe Umgebungen geworden, dass für die Lokalisierung eines Spieles Tausende von Strings übersetzt werden müssen. Groß angelegte Rollen- und Abenteuerspiele erzählen meistens eine nicht-lineare Geschichte, die erst durch den/die Spieler/in aufgebaut wird, weshalb es schwierig ist, den Überblick zu behalten. Im Idealfall haben die Übersetzer/innen eine (Beta-)Version des ausgangssprachlichen Spieles, um die Spielwelt zu erkunden, dies ist meistens jedoch nicht möglich (vgl. Bernal Merino 2008: 32-33; 35).

Das Problem der Variablen wurde bereits mit einem Beispiel aus einem Spiel erläutert. Hier soll noch einmal angemerkt werden, dass dies vor allem in diesem Bereich verstärkt vorkommt. Meistens kann der/die Spieler/in zumindest einen Namen, oft sogar noch mehr Merkmale auswählen, die später in verschiedene Textausgaben eingefügt werden. Viele Aktionen in einem Spiel verlaufen nach demselben Muster, sodass es logisch ist, für alle diese Aktionen nur einen Satz von Strings zu verwenden, der mithilfe der eingefüllten Wörter und Sätze für die jeweilige Situation verändert wird. Genauigkeit bei der Übersetzung ist hier besonders wichtig, einerseits, um die Platzhalter nicht versehentlich zu entfernen, andererseits um sicherzustellen, dass die schließlich angezeigten Meldungen grammatikalisch korrekt sind. Dazu sind oft Veränderungen der Sätze nötig, da die zielsprachliche Grammatik mehr Übereinstimmungen erfordert als die Ausgangssprache (vgl. Bernal Merino 2008: 36-37).

Bei vielen Spieltiteln gibt es nicht nur als Dialogfeld angezeigte Texte, sondern gesprochene. Es gibt verschiedene Ansätze, diese zu übersetzen, vorrangig mit Untertitel und Synchronisation. Dadurch ähnelt die Arbeit der Übersetzer/innen der Arbeit im Zusammenhang mit Filmen und anderen audio-visuellen Medien. Um die Charaktere authentisch klingen zu lassen ist nicht nur die Auswahl der richtigen Synchronsprecher/innen wichtig, sondern auch der Tonfall und die Formulierungen des Sprechtextes. Es geht bei den Zieltexten

nicht unbedingt um korrekte Grammatik, diese kann sogar „falsch“ sein, wenn eine Figur bewusst eine grammatikalisch fehlerhafte Sprechweise aufweist. Vielmehr ist Kreativität gefragt, um die Figur so wirken zu lassen, dass sie ins Konzept passt (vgl. Lutsuk 2014: 28-29).

Wie bereits angemerkt, stellen viele Spiele einen Bezug zur Realität her und/oder enthalten Material, das nicht in allen Kulturen geduldet wird oder sogar in manchen Ländern verboten ist. In Kapitel 3.1.4 wurden Vorfälle beschrieben, wo Titel zurückgezogen oder verändert werden mussten, weil sie als blasphemisch, nicht jugendgerecht oder geschichtsverfälschend eingestuft wurden. Diese Aspekte sollten zwar schon in der Planung identifiziert und verändert werden, jedoch kann gut möglich sein, wohl auch wegen der Unmenge an übersetzungsrelevantem Material, dass erst die Übersetzer/innen Steine des Anstoßes entdecken. In solchen Fällen ist es wichtig, sich auf die beratende Funktion als Kulturmittler/in zu besinnen und Rücksprache mit den Verantwortlichen zu halten (vgl. Steiert & Steiert 2014: 58).

Insgesamt handelt es sich bei Spielen um Werke, in denen die Kreativität vieler Beteiligten steckt. Die Sprache ist ein Teil des Gesamtgefüges, das eine Geschichte, oft eine Parallelwelt und ein gewisses Spielgefühl vermitteln möchte. Um den Spieler/innen an verschiedenen Orten der Welt das Spielvergnügen zu garantieren, ist ein Grundwissen über Spiele und deren Aufbau wichtig, genauso wie das Auseinandersetzen mit der Materie. Als letzte Anmerkung, die an verschiedenen Stellen genannt wird, sei hier noch hinzugefügt, dass bei der Zusammenarbeit mit den Spieleentwickler/innen Fingerspitzengefühl gefragt ist. Dies ist beim Umgang mit den kreativen Köpfen geboten, da diese anscheinend mehr als andere Entwickler/innen hinter ihren Produkten stehen und sich mehr Sorgen darüber machen, dass die Lokalisierung gut gemacht wird und auch das „gewisse Etwas“ in der Zielkultur nicht verloren geht (vgl. Cerda 2010: 41; Edwards 2014: 15).

5.4 Qualitätsmanagement

Dieses Kapitel befasst sich mit dem Qualitätsaspekt im Rahmen von Lokalisierung, vor allem damit, wie ein qualitativ hochwertiges Endprodukt geschaffen werden kann. Bei lokalisierten Produkten wird zumeist unterschieden nach der Qualität der Software und der Qualität der Sprache. Dies ist darauf zurückzuführen, dass sich die Qualität eines Softwareprodukts – die Funktion, die Architektur, die Erfüllung der Aufgaben – relativ gut messen lassen, während die Gesamtqualität von Sprache zum Teil subjektiv zu bewerten ist (vgl. Dunne 2006^b: 95-96).

5.4.1 Softwarequalität

Kommerziell vertriebene Softwareanwendungen sind in der Regel höchst komplexe Produkte. Selbst bei kleinen Hilfsprogrammen und einfach gestrickten Spielen ist das Zusammenspiel der Code-Teile und der Ressourcen empfindlich. Da ein Computer nicht selbst denkt, kann er fehlerhafte Codes und falsche Verweise auf Ressourcendateien nicht interpretieren, was bei Ausführen der Anwendungen zu Fehlermeldungen und Abstürzen führt. Deshalb ist das Testen von Software und ihrer Bestandteile im Rahmen der Qualitätssicherung ein wichtiger Teil der Entwicklung. Da eine Anwendung im Rahmen von Internationalisierung und Lokalisierung verändert wird, gibt es auch hier verschiedene Tests, um die Funktion dieser Versionen sicherzustellen. Manche dieser Tests werden vom Softwareunternehmen ausgeführt, da sie die Software eigentlich noch in der Entwicklungsphase befindet. Andere Tests werden vom Lokalisierungsunternehmen durchgeführt. Welche Tests vom Dienstleister übernommen werden ist unterschiedlich und hängt vor allem von den Ressourcen des beauftragenden Unternehmens ab.

Internationalisierungstests sind integraler Bestandteil der Softwareentwicklung, oder sollten es zumindest sein. Wie bereits mehrfach erwähnt, sollte die Internationalisierung einer Anwendung so früh wie möglich in die Entwicklung eingebaut werden, um die bestmöglichen Resultate zu erreichen. Daraus folgt, dass Internationalisierung auch so früh wie möglich getestet werden sollte, um spätere Probleme bei der tatsächlichen Lokalisierung zu verhindern. Internationalisierungstests setzen sich aus zwei Schritten zusammen: Zum einen ist es das Testen auf die Unterstützung von verschiedenen Sprachen, regionalen Einstellungen und Formaten, Kompatibilität mit lokalisierten Betriebssystemen und anderen international relevanten Belangen. Nicht nur sind diese Faktoren für später lokalisierte Anwendungen von Belang, sondern auch für ausgangssprachliche Versionen, die international vertrieben werden. In manchen Ländern und für manche Bereiche sind nicht alle Softwareanwendungen lokalisiert erhältlich, weswegen die Konsument/innen auf (meistens) englische Pakete zurückgreifen. Trotzdem ist es für die User/innen wichtig, dass regionale Formate oder nicht-lateinische Schriften unterstützt werden, um damit arbeiten zu können.

Die zweite Art von Internationalisierungstest ist das Testen auf Lokalisierbarkeit. Hiermit wird sichergestellt, dass eine Anwendung gut lokalisiert werden kann. Verschiedene potenzielle Probleme werden gesucht, zum Beispiel nicht externalisierte Ausgabetexte, Stringverkettungen, die problematisch werden können, Felder und Layouts, die nicht genügend Platz für längere übersetzte Texte bieten, Bilder, die Text enthalten, der in der Bilddatei verändert werden muss und Farben, Grafiken oder andere Komponenten, die in einer lokalisierten Version kulturell nicht akzeptabel sind und vielleicht durch andere ersetzt werden können, die überall problemlos rezipiert werden. Bereits in dieser Phase ist es ratsam, verschiedene Expert/innen der späteren zielsprachlichen Locales miteinzubeziehen. Eine gute Methode, um Lokalisierbarkeit zu überprüfen, ist Pseudo-Lokalisierung oder Pseudo-

Übersetzung. Mit speziellen Programmen oder Lokalisierungstools, die diese Funktion enthalten, werden die Strings der Sprachdateien mit längeren Strings und/oder Sonderzeichen ersetzt. Zum einen kann man so feststellen, ob noch hartkodierte Strings vorhanden sind und zum zweiten, ob die Zeichensätze auch unterstützt werden, wobei letztere Funktion eigentlich in den oben genannten Bereich fällt (vgl. Esselink 2000: 145-149; Sikes 2010: 33-35).

Die zweite Testgruppe wird nach der Lokalisierung mit allen lokalisierten Versionen einer Anwendung durchgeführt. Lokalisierungstests sind sprachliche Tests, kosmetische Tests und Funktionstests. Die Sprache wird von zielsprachlichen Expert/innen getestet. Da dieser Aspekt im nächsten Kapitel behandelt wird, soll an dieser Stelle nicht näher darauf eingegangen werden.

Kosmetische Tests werden zumeist von Softwareexpert/innen durchgeführt, die idealerweise mithilfe einer Anleitung, des Testskripts, alle Dialoge und Meldungen durchgehen, um festzustellen ob Probleme mit der Darstellung vorliegen. Sie überprüfen, ob alle Menüs, Optionen und Befehle auch in der lokalisierten Version vorhanden sind, ob alle Strings korrekt angezeigt werden und in ihre Felder passen, ob alle Zeichen angezeigt werden, ob alle Bestandteile der Benutzeroberfläche bei allen Bildschirmauflösungen auf dem Bildschirm Platz finden, ob die Ästhetik der Benutzeroberfläche nicht verändert wurde, ob alle Hot-Keys einzigartig sind und ob alle Formate den Regeln des Locales entsprechen. Kosmetische Tests werden zuerst in Ressourcen-Editoren oder Lokalisierungstools ausgeführt, die den Code auf manche dieser Aspekte automatisch überprüfen können. Als zweiter Schritt wird alles noch einmal in der ausgeführten Anwendung getestet (vgl. Esselink 2000: 151-152; Sikes 2010: 35).

Funktionstests am lokalisierten Produkt sind im Prinzip eine Wiederholung der Tests an der Ausgangssprachlichen Version. Sie sind nicht Standard, und wenn sie gemacht werden, geschieht dies oft nicht ausführlich. Wenn die anderen Tests genau durchgeführt werden, ist es unwahrscheinlich, an diesem Punkt Probleme zu finden. Dies wäre auch höchst unerfreulich, da die Behebung von Softwarefehlern, genannt Bugs, immer teurer wird, je später im Entwicklungsprozess sie geschieht (vgl. Westland 2002: 3). Neben der grundsätzlichen Funktion der Anwendung wird bei diesen Tests vor allem die Kompatibilität mit der lokalen Umgebung getestet, zum Beispiel, ob die Software auf dem lokalisierten Betriebssystem läuft, ob das Zusammenspiel mit anderen Programmen funktioniert und ob die übliche Hardware kompatibel ist (vgl. Esselink 2000: 152-154; Sikes 2010: 36).

Tests sind wichtige Bestandteile der Qualitätssicherung bei Softwareprodukten. Werden Fehler gefunden, müssen sie natürlich behoben werden. Dies geschieht im Rahmen von Workflows. Normalerweise wird ein Test-Team mit der Suche nach Bugs beauftragt und ein anderes Team mit deren Behebung. Die Tester/innen des ersten Teams folgen den Skripten, das heißt sie durchkämmen das Programm. Wenn das Programm abstürzt, eine Fehlermeldung angezeigt wird oder ein Schritt auf eine andere Art und Weise nicht durchgeführt werden kann, wird dies in der Regel noch einmal versucht, um sicherzugehen, dass ein Bug vor-

liegt. Der nächste Schritt ist das Erstellen eines Bug-Reports mit genauen Informationen über den Fehler, der an das zweite Team weitergegeben wird. Bug-Reports können auch von manchen Programmen automatisch erstellt werden. Die Bugs durchlaufen nun meistens mehrere Phasen der Problembeseitigung und nochmaligen Überprüfung, bis sie abgehakt werden oder als nicht relevant eingestuft werden. Das heißt, der Workflow bei Tests verläuft dynamisch und nicht linear (vgl. Esselink 2000: 160-163).

Ein Sonderfall ist hier wieder der Bereich Spiele. Ist es bei den meisten Programmen gut möglich, sich durch alle Funktionen durchzuklicken, müssen Spiele tatsächlich gespielt werden. Natürlich bedeutet dies nicht, dass der/die Tester/in sich nach Lust und Laune mit dem Spiel beschäftigen kann, auch hier gilt es, systematisch alle Funktionen zu überprüfen. Je offener das Spieluniversum und je weniger linear die Geschichte aufgebaut ist, desto komplexer wird auch hier das Testen. Es gibt bei Rollenspielen unzählige Möglichkeiten, das Spiel durchzuspielen, also müssen so viele Wege wie möglich ausgetestet werden, um Bugs zu identifizieren. Deshalb ist es auch für Spieltester/innen grundlegend, dass sie sich mit der Materie gut auskennen und je genauer die Vorgehensweise beschrieben wird, desto zielführender wird die Suche nach Problemen (vgl. Barceló 2011: 38-39).

5.4.2 Sprache

Die Frage, die sich zunächst stellt, ist, was sprachliche Qualität ist und wie sie kontrolliert werden kann. Wie Rueda (2010: 38) anmerkt, ist Sprache keine exakte Wissenschaft, die in einem Reagenzglas überprüft werden kann. Ein sprachliches Produkt wird von einer Vielzahl von Menschen rezipiert, die dieses jeweils anders interpretieren und auch unterschiedlich beurteilen. Auch Arnsperger (2013: 44) weist auf den subjektiven Charakter von Sprache hin.

5.4.2.1 DIN EN 15038:2006

In der Norm für Übersetzungs-Dienstleistungen, DIN EN 15038:2006-08, werden trotzdem verschiedene Aspekte beschrieben, auf die Übersetzer/innen während der Übersetzung achten müssen. Der erste Punkt hier ist Terminologie. Ähnlich wie bereits in Kapitel 5.2 erläutert, sollten folgende Vorgaben eingehalten werden: „Übereinstimmung mit der Kundenterminologie, den terminologischen Konventionen des Fachgebietes oder mit anderen zugrunde gelegten Terminologien; kohärente Terminologianwendung im gesamten Übersetzungstext.“ Weiters werden angeführt:

- „b) Grammatik: Satzbau, Schreibweisen, Interpunktion, Rechtschreibung, diakritische Zeichen.
- c) Lexik: Lexikalische Kohäsion, phraseologische Korrektheit.
- d) Stil: Einhaltung der eigenen oder kundenseitigen Stilrichtlinien; Nutzung des adäquaten Sprachregisters und Wahl der korrekten Sprachvariante.

- e) Locale: Lokale und regionale Konventionen und Normen.
 - f) Formatierung [...]
 - g) Zielgruppe und Zweck der Übersetzung.“
- (Herzog & Mühlbauer 2007: 21)

Um die Machbarkeit für den/die Übersetzer/in zu gewährleisten, müssen die einzelnen Punkte zuvor so gut wie möglich festgelegt werden. So heißt es:

„Der Übersetzungsdienstleister muss dafür Sorge tragen, dass Informationen über spezielle sprachliche Anforderungen an das Übersetzungsprojekt aufgezeichnet werden. Zu diesen Anforderungen können die Einhaltung von kundenseitige Stilrichtlinien, die Adaption der Übersetzung an eine vereinbarte Zielgruppe, die Ausrichtung auf einen bestimmten Übersetzungs- oder Verwendungszweck, die Verwendung geeigneter Terminologie oder die Aktualisierung von Glossaren gehören.“ (Herzog & Mühlbauer 2007: 20)

Nach der Übersetzungsarbeit muss der Zieltext vom/von der Übersetzer/in selbst geprüft werden und schließlich auch von einer andere Person korrekturgelesen werden. Bei diesen Schritten soll sichergestellt werden, dass die oben genannten Aspekte nicht übergangen wurden. Gegebenenfalls ist der Zieltext auch von einem/einer Fachprüfer/in geprüft werden. Diese/r ist ein/e einsprachige Expert/in im Fachgebiet des Textes (vgl. Herzog & Mühlbauer 2007: 21-22).

5.4.2.2 Beteiligte Personen

Die Bestimmungen in der Norm sind natürlich relativ offen und so liegt es an den Expert/innen selbst, im jeweiligen Projekt die Anforderungen zu identifizieren und zu dokumentieren. Es ist deshalb klar, dass die Kompetenzen der betroffenen Personen eine große Rolle spielt. Die Norm spezifiziert auch hier die Anforderungen für Übersetzer/innen und Korrektor/innen:

- „a) Übersetzerische Kompetenz: Die übersetzerische Kompetenz besteht in der Fähigkeit, Texte auf dem erforderlichen Niveau [...] zu übersetzen. Dazu gehören die Einschätzung der Problematik des Textverständnisses und der Texterstellung sowie die Fähigkeit, den Zieltext nach der Vereinbarung zwischen dem Kunden und dem Übersetzungsdienstleister [...] zu erstellen und die Ergebnisse rechtefertigen zu können.
- b) Sprachliche und textliche Kompetenz in der Ausgangs- und Zielsprache: Die sprachliche und textliche Kompetenz sind die Fähigkeit, die Ausgangssprache zu verstehen und die Zielsprache zu beherrschen. Textliche Kompetenz erfordert die Kenntnis von Textsortenkonventionen für eine möglichst große Bandbreite an allgemeinsprachlichen und Fachtexten sowie die Fähigkeit, dieses Wissen bei der Erstellung von Texten anzuwenden.
- c) Recherchierkompetenz, Informationsgewinnung und –verarbeitung: Recherchierkompetenz ist die Fähigkeit, effizient sprachliches und fachliches Zusatzwissen zu erwerben, das für das Verständnis des Ausgangstextes und das Erstellen des Zieltextes erforderlich ist. Recherchierkompetenz erfordert auch Erfahrung beim Einsatz von Recherche-Tools sowie die Fähigkeit, passende Strategien für den effizienten Einsatz verfügbarer Informationsquellen zu entwickeln.

d) Kulturelle Kompetenz: Kulturelle Kompetenz ist die Fähigkeit, Informationen über das Locale, Verhaltensmuster und Wertesysteme einzusetzen, die für die Ausgangs- und die Zielkultur charakteristisch sind.

e) Fachliche Kompetenz: Fachliche Kompetenz umfasst Fähigkeiten und Fertigkeiten, die für die professionelle Vorbereitung und Anfertigung von Übersetzungen erforderlich sind. Dazu gehört auch der Umgang mit technischen Ressourcen [...].

Mindestens eine der folgenden Voraussetzungen sollte erfüllt sein:

- eine formale höhere Übersetzungsausbildung (anerkannter Hochschul-Abschluss);
- eine vergleichbare Ausbildung in einem anderen Fachbereich mit mindestens zwei Jahren dokumentierter Übersetzungserfahrung;
- mindestens fünf Jahre dokumentierte professionelle Übersetzungserfahrung.“

(Herzog & Mühlbauer 2007: 17)

Es gibt an der Vollständigkeit dieser Liste wohl wenig auszusetzen. Auch die Voraussetzungen sind legitim; nach fünf Jahren Erfahrung beziehungsweise mit einem Abschluss in der Tasche sollten alle Übersetzer/innen über die notwendigen Kompetenzen verfügen. Die Schwierigkeit, die hier in der Praxis auftaucht, liegt jedoch im vorherrschenden Agentur-Freelancer-System. Tishin (2014: 35-38) veranschaulicht dies an einem Fallbeispiel aus dem Bereich Spielelokalisierung: Sein Übersetzungsunternehmen arbeitete mit Freelancer/innen und war auf wenige Sprachkombinationen spezialisiert. Mit der Änderung des Marktes und dem Aufschwung von mobilen Spielen mit geringem Umfang änderten sich auch die Anforderungen seitens der Kunden. Es wurden vermehrt Dienstleister gesucht, die diese Spiele in alle gewünschten Sprachen, bis zu 14, übersetzen konnten. Deshalb wurden im Unternehmen innerhalb kürzester Zeit eine große Anzahl von Freelancer/innen in online-Datenbanken gesucht und in den Übersetzerpool aufgenommen. Nach der ersten Welle von abgeschlossenen Projekten gab es sehr viele negative Rückmeldungen seitens der Konsument/innen, die an das Übersetzungsunternehmen weitergegeben wurden. Da es sich meistens um schlechte Übersetzungen handelte, wurde die Liste der Freelancer/innen überprüft und ein Evaluierungssystem eingesetzt. Das Fazit war, dass nur 27 % der eingetragenen tatsächlich gute Übersetzungen lieferten und andere teilweise sogar falsche Angaben gemacht hatten, um an Aufträge zu kommen. Die Folge war, dass ein umfassendes Einstiegssystem für neu eingetragene Freelancer/innen eingeführt wurde, um nicht geeignete Übersetzer/innen ablehnen zu können.

Der springende Punkt ist hier, dass es schwierig ist, übers Internet festzustellen, ob Übersetzer/innen die notwendigen Kompetenzen aufweisen. Für Personen, die ohne Qualifikationen Übersetzungen anbieten, ist es nicht schwierig, es einfach zu versuchen. Selbst wenn die Übersetzer/innen Abschlüsse und andere Qualifikationen vorweisen können, ist es für die Unternehmen nicht unbedingt klar, ob diese auch die notwendige Ausbildungsqualität garantieren können. Die in Kapitel 4 beschriebene Expansionen der Lokalisierungsindustrie über beinahe den gesamten Erdball führt dazu, dass auch in Sprachen übersetzt wird, für die es wenig anerkannte oder bekannte erwerbbare Qualifikationen gibt. So ist es für Sprach-

dienstleistungsunternehmen nahezu unmöglich, Ausbildungsstätten auf der anderen Seite der Erde zu bewerten.

Verschiedene Formen der Kompetenzüberprüfung sind auf jeden Fall eine gute Möglichkeit, die Eignung der Übersetzer/innen sicherzustellen. Auch bei Rueda (2010: 39) wird angesprochen, dass Einstiegstests und spätere Stichproben Rückschlüsse auf die Kompetenzen der Übersetzer/innen zulassen und somit eine gute Methode sind, ein gewisses Niveau zu garantieren.

Die oben zitierten Anforderungen gelten sowohl für Übersetzer/innen als auch für Korrektor/innen. Es ist nicht unüblich, dass ein/e Korrektor/in beziehungsweise ein/e Fachprüfer/in eingesetzt wird, der/die aus einer Zweigstelle des Kundenunternehmens stammt. Auf diese Weise werden einerseits bestehende Ressourcen eingesetzt, andererseits sind diese Personen mit den sprachlichen Präferenzen des Unternehmens vertraut. Wenn es sich um Fachprüfer/innen handelt, die lediglich wegen ihrer zielsprachlichen Kompetenz ausgewählt werden, kann es zu Problemen führen. Für die Kundenunternehmen ist es oft schwierig, die tatsächliche Sprach- und Fachkompetenz der Fachprüfer/innen einzuschätzen, mitunter wird darauf auch gar nicht geachtet. Unter Lai/innen fehlt meistens das Bewusstsein, dass professionelle Sprachkompetenz über das bloße Sprechen einer Sprache hinausgeht. Das bedeutet im Endeffekt, dass die fachliche Prüfung des Zieltextes zu mehr Unstimmigkeiten und Fehlern führt. Hinzu kommt auch hier der subjektive Charakter von Sprache. Ein/e Prüfer/in, dem/der die fachliche Kompetenz fehlt und der/die auch keine konkreten Vorgaben bekommt, verändert womöglich die richtige Terminologie oder korrekte Formulierungen, um ein Ergebnis zu erhalten, das ihm/ihr subjektiv besser gefällt. Erschwerend wirkt, dass unerfahrenen Prüfer/innen und Korrektor/innen oft den Eindruck haben, sie müssten etwas verändern, da ja sonst ihre eigene Kompetenz infrage gestellt würde. Wichtig ist deshalb, dass die Prüfer/innen sich nicht als Gegner/innen der Übersetzer/innen sehen und dass hier, genauso wie in den anderen Bereichen, die Ziele der Prüfung klar formuliert sind. Das heißt, das Ziel dieser Überprüfungen ist nicht, Texte umzuformulieren, sondern offensichtliche Fehler, wie Auslassungen, Fehlübersetzungen, falsche Rechtschreibung, Grammatik und Terminologie sowie Nichteinhaltung der Style-Guides zu finden und zu beheben (vgl. Bass 2006: 86-88).

5.4.2.3 Messen der Sprachqualität und Automatisierung

Die oben beschriebenen Ansätze helfen zwar, Qualität zu kontrollieren und stellen zumindest die Anforderungen relativ klar dar. Dennoch liegt die Produktion von qualitativ hochwertigen Ergebnissen und vor allem deren Bewertung in der Hand der beteiligten Personen. Die Kontrolle dieser Personen ist für manche nicht genug, deshalb wurden Methoden erstellt, die Subjektivität aus dem Kontrollprozess so gut wie möglich zu entfernen und objektive Maßstäbe zu schaffen. Dies ist für sehr technische Texte sinnvoll, also auch für viele, die im Rahmen

von Lokalisierungsprojekten entstehen. Hier liegt der Fokus eher auf Richtigkeit, Klarheit und Vollständigkeit und sehr wenig auf nur subjektiv bewertbare Aspekte, wie es zum Beispiel bei literarischen Texten der Fall ist.

In den vergangenen zehn bis 15 Jahren wurden einige Systeme zur Quantifizierung von Übersetzungsqualität erstellt. Der Gesamteindruck eines Textes oder der Stil kann schwerlich eingestuft werden, doch für andere Aspekte gibt es sehr wohl diese Möglichkeit. Dazu zählen offensichtliche Fehlübersetzungen, Auslassungen, Zusätze, Grammatik, Rechtschreibung, Punktierung und Terminologie. Die Messsysteme enthalten verschiedene Fehlerkategorien und jeder gefundene Fehler wird anhand seiner Schwere beurteilt. So kann im Endeffekt ein Punktwert errechnet werden, der dann auf einer Skala eingeordnet wird.

J2450 ist eine Methode der „Society of Automotive Engineers“. Diese wurde, wie der Name schon sagt, für die Autoindustrie entwickelt, mittlerweile aber auch für andere Bereiche weiterentwickelt. Dieser Standard hat sieben Fehlerkategorien, wie zum Beispiel Rechtschreibfehler und Auslassungen, und für die Schwere der Fehler zwei Unterkategorien. Jeder dieser Kategorien und Unterkategorien wird ein Wert zugeteilt. So kann schließlich ein Gesamtwert errechnet werden. Dieser Standard muss auf jedes Unternehmen zugeschnitten werden, das heißt, die Werte und die daraus folgende Bewertung werden an die Anforderungen des Unternehmens angepasst.

Bevor die „Localization Industry Standards Association“ eingestellt wurde, wurde das LISA QA-Modell entwickelt. Auch hier gibt es Unterkategorien für die Schwere der Fehler, die Kategorien dazu werden von den jeweiligen Unternehmen gemeinsam mit den Kunden erstellt. Dazu gibt es vorgefertigte Modelle für Software, Hilfedatei und Dokumentation.

Das Modell des „Translation Bureau“ der kanadischen Regierung heißt „Canadian Language Quality Measurement System“. Allgemeine Fehler und Fehler in der Übersetzung werden als schwer oder leicht eingestuft. Die Qualität wird an der Anzahl der Fehler in Passagen von 400 Wörtern gemessen. Für jede Qualitätsstufe ist eine gewisse Anzahl von schweren und/oder leichten Fehlern zugelassen, woraus sich die Einstufung ergibt.

Das Unternehmen „ForeignExchange Translations“ hat das System METRiQ für die Evaluierung von medizinischen Übersetzungen entwickelt. Auch hier werden Fehler anhand von Kategorien und Grad der Schwere gezählt und Zahlenwerte zugewiesen. Aus dem Schlusswert wird die Statistik für eine 3000-Wörter-Passage errechnet. Interessant ist hier die Abstufung der zugelassenen Fehlerhäufigkeit: Je nach Projektstatus und Projektart werden mehr oder weniger Fehler akzeptiert.

Wird ein Messsystem eingesetzt, muss es an das Unternehmen beziehungsweise den Kunden angepasst werden. Dies geschieht, indem ein Satz aus Fehlerkategorien, Schweregraden und zugewiesenen Werten erstellt wird. Weiters sollte noch abgeklärt werden, welche Ressourcen für die Evaluierungsverantwortlichen zur Verfügung gestellt werden, nach welchen Projektschritten Evaluierungen durchgeführt werden und welche Werte noch akzeptiert werden (vgl. Arnsparger 2013: 46-49).

Diese Methoden orientieren sich an leicht greifbaren, objektiven Fehlern. Diese können nicht nur für Messsysteme herangezogen werden, sondern auch von Software entdeckt werden. Die Rechtschreib- und Grammatiküberprüfung von Textprogrammen und Browsern sind auch solche Tools und mittlerweile weit verbreitet. Spezielle Programme können noch mehr und sind somit für die Automatisierung mancher Überprüfungsprozesse gut geeignet.

Die Überprüfung von Texten durch Menschen hat grundsätzlich einige Schwachstellen, wo automatische Lösungen einspringen können. Erstens müssen Korrektor/innen bezahlt werden und brauchen eine gewisse Zeit für ihre Arbeit. Dies ist nur logisch, handelt es sich um eine wertvolle Dienstleistung und manche Fehler und Unstimmigkeiten im Text werden von Programmen nicht erkannt. Nicht immer ist jedoch eine volle Überprüfung für alle Texte nötig, vor allem wenn es sich um kurzlebige handelt. Programme können außerdem innerhalb relativ kurzer Zeit große Dokumente durcharbeiten und sind bei manchen Fehlerarten sogar effektiver. Deshalb ist es sinnvoll, automatische Lösungen für zumindest einen Teil der Qualitätsüberprüfung einzusetzen. Folgende Aspekte können von diesen überprüft werden: die kohärente Verwendung der zuvor beschlossenen Terminologie, die Kohärenz im Text, also ob ein Satz (oder Segment) immer gleich übersetzt wurde, Zahlen und Zahlenformate, Vollständigkeit, Typografie und Formatierung und verschiedene technische Aspekte, zum Beispiel Einhaltung der maximalen Stringlänge oder die Gültigkeit von Tags. Es gibt natürlich andere Kategorien, für die nach wie vor menschliche Überprüfung notwendig ist, zum Beispiel die kontextabhängige Verwendung der richtigen Terminologie, falsche Übersetzungen und Stilfehler. Außerdem besteht immer die Gefahr, dass Wörter oder Sätze als Fehler erkannt werden, obwohl sie korrekt sind. Wird, zum Beispiel, das Format für Dezimalzahlen überprüft, erkennt das Programm die Verwendung des Punktes als Dezimaltrennzeichen im Deutschen als falsch, obwohl es sich bei einem Wert von „1.2“ auch um ein Aufzählungszeichen einer Kapitelüberschrift handeln kann. Nichtsdestotrotz gibt es auch in diesem Bereich Fortschritte und es werden stetig neue Algorithmen entwickelt, um diese Programme effizienter zu machen. Im technischen Bereich ist der Einsatz von automatisierten Prozessen auf jeden Fall nicht zu vernachlässigen (vgl. Massion 2010: 42).

5.4.2.4 Qualitätshindernisse

Es gibt einige Faktoren, die sprachlicher Qualität im Wege stehen. Zuerst sei hier das sogenannte magische Dreieck genannt, das aus Zeit, Kosten und Qualität besteht. Qualität kann somit beeinflusst werden, indem man die beiden anderen Faktoren verändert; ist höhere Qualität bei einem Projekt gewünscht, muss man also entweder mehr Zeit, mehr Geld oder eine Kombination aus beidem investieren. Oft steht aber nicht die Qualität im Vordergrund, sondern die Kosten, da in kommerziellen Unternehmen der Profit zählt. Im Wettbewerb konkur-

rieren viele Unternehmen auf Preisebene, das heißt, sie bieten ihre Leistungen günstiger als andere Firmen an, um einen Vorteil zu erhalten. Wenn jedoch nur auf diesen Aspekt geachtet wird, so ist die Konsequenz oft schlechtere Qualität im Endprodukt, da sehr billige Dienstleister es meistens nicht schaffen werden, gute Übersetzer/innen einzustellen oder zu beauftragen. Dass sich verschiedene Übersetzungsdienstleister preislich unterbieten, wird durch Globalisierung verstärkt, da der Markt der Anbieter global wird und so mehr Wettbewerb herrscht. Es muss für Unternehmen, die Übersetzungsdienstleistungen in Anspruch nehmen, klar sein, dass qualitativ hochwertige Übersetzungen nicht billig sind und dass die finanziellen Folgen der mangelnden Qualität im Endeffekt meistens höher sind, als der Unterschied zwischen den Angeboten. Ein Produkt mit offensichtlichen qualitativen Mängeln wird auf dem Zielmarkt nicht sehr erfolgreich sein; werden durch sprachliche Fehler rechtliche Regelungen verletzt, kommt es zu noch schlimmeren Konsequenzen. Der Aspekt Zeit spielt natürlich auch eine große Rolle, meistens ist es mangelhaftes Projektmanagement, wenn für die verschiedenen Schritte, die mit dem Übersetzungsprozess zu tun haben, nicht genügend Zeit eingeplant ist. Für Übersetzer/innen ist es nicht möglich, die Qualitätsstandards zu erfüllen, wenn nicht genügend Zeit verfügbar ist. Überprüfung und Korrektur sind wichtige Schritte, die als erste übersprungen werden wenn die Deadline näher rückt. Wie gesagt stammt Zeitmangel meistens aus schlechtem Projektmanagement oder unrealistischen Erwartungen. Es sind zwar ungefähre Größen für den täglichen Output von Übersetzer/innen bekannt, jedoch kann es aufgrund der Fachlichkeit oder der Qualität des Ausgangstextes große Abweichungen geben. Nicht zu vernachlässigen sind hier größere Unterschiede zwischen verschiedenen Sprachkombinationen. Für Übersetzungen zwischen Englisch und romanischen Sprachen kann man mit einem täglichen Volumen von 2500 Wörtern rechnen, bei Englisch und asiatischen Sprachen jedoch eher mit 1500 bis 2000 Wörtern. Einen Einfluss auf die für Übersetzungen benötigte Zeit hat auch die zu geringe Zeitzuteilung für Schritte, die vorher gemacht werden, wie die Erstellung der Terminologie. Wird hier gespart, ist dies überaus wertvolle Ressource für Übersetzer/innen mangelhaft, was dazu führt, dass diese selbst umso mehr Zeit in Recherche stecken müssen.

Bei Planung und Projektmanagement ist oft mangelnde Kommunikation ein Hindernis. Wenn der Dienstleister nicht richtig mit dem Kunden kommuniziert, wird es schwierig sein, die notwendigen Informationen zu bekommen und das Projekt dementsprechend zu planen. Als Folge wird dann meistens während des Projektes herausgefunden, dass Deadlines nicht eingehalten werden können oder manche Prozesse nicht funktionieren. Es liegt an den Dienstleistungsunternehmen, sicherzustellen, dass alle benötigten Informationen vorliegen und dass Kommunikationswege festgelegt werden, um alle auftauchenden Fragen stellen zu können.

Ein Aspekt, der besonders heikel ist, ist das Ausgangsmaterial. Lokalisierungsunternehmen können sich dieses nicht aussuchen, sondern müssen mit dem Material arbeiten, dass sie bekommen. Da Software in der Regel nicht von Sprachexpert/innen geschrieben wird und

auch die Erstellung der Dokumentation immer öfter aus Ressourcenmangel an Ingenieur/innen weitergegeben wird, ist nicht auszuschließen, dass die Ausgangstexte sprachlich fehlerhaft sind und Grammatik, Rechtschreibung und Punktierung Mängel aufweisen sowie die Terminologie nicht richtig verwendet wurde.

Schließlich steht auch die Technologie mitunter eine Hürde dar. Da Übersetzer/innen und sonstige Sprachexpert/innen im Lokalisierungsbereich sehr viel mit verschiedenen Programmen arbeiten müssen, entstehen oft Probleme mit der Meisterung der Programme selbst, mit Konvertierungen und Formaten sowie mit Kompatibilität. Dies birgt natürlich eine gewisse Ironie, da CAT-Tools und andere Programme ja eigentlich die Produktivität fördern sollten, jedoch auf diese Weise dieselbe behindern (vgl. Bass 2006: 70-78).

Insgesamt ergeben sich die meisten Probleme, wenn die Erwartungen und Anforderungen der Kunden nicht mit dem tatsächlich machbaren übereinstimmen, vor allem wenn die Lokalisierungsunternehmen unrealistische Vorstellungen noch verstärken, weil sie nicht professionell arbeiten und, zum Beispiel, Leistungen viel zu billig anbieten (vgl. Bass 2006: 84-86).

5.4.2.5 *Qualität im und als Prozess*

In diesem Kapitel wurde besprochen, welche Aspekte die sprachliche Qualität der Übersetzungen bestimmen. Haben die bisherigen Kapitel ihr Ziel nicht verfehlt, sollte klar sein, dass Übersetzung jedoch nur ein Teil im Gesamtgefüge des Lokalisierungsprojekts ist. Somit kann daraus geschlossen werden, dass auch im Zusammenhang mit Qualität Übersetzung nicht in der Luft hängt, sondern im Projekt eingebettet ist und es daran liegt, die Prozesse zu optimieren, um die gewünschte Qualität zu erreichen. Qualität und Qualitätsmanagement sind Schlagwörter, die nur allzu oft aufgegriffen und behandelt werden. Im Folgenden wird zuerst über die Entwicklung von Qualitätsmanagement gesprochen und danach überlegt, wie dieses in den Lokalisierungsprozess eingebaut werden kann.

In diesem Zusammenhang stellt Dunne (2006^b: 96-102) fest, dass traditionelle Arten, Qualität zu messen und zu managen, wie bei der Herstellung von Gütern, für die Sprachindustrie nicht zielführend sind. Qualitätsmanagement als solches entstand in den 1940er Jahren, als begonnen wurde, fertige Produkte Qualitätskontrollen zu unterziehen. Als erster Schritt ist dies zwar besser, als fehlerhafte Produkte zu liefern, doch werden Fehler im Herstellungsprozess dadurch nicht unbedingt ausgeräumt. Ab den Siebziger Jahren wurde schließlich der Fokus auf die Produktion und die verschiedenen Prozesse gerichtet, um von vornherein die Wahrscheinlichkeit zu senken, fehlerhafte Ergebnisse herzustellen. Dies nennt sich Qualitätssicherung. Es wird also nach jedem Schritt überprüft, ob dieser erfolgreich war. Im Prinzip ist also Qualitätssicherung die Kombination von verschiedenen Qualitätskontrollschritten. Um dies bewerkstelligen zu können, muss zuvor definiert werden, wie das Ergebnis aussehen

soll. Im industriellen Herstellungsprozess ist dies geradlinig: Ein Produkt oder Zwischenprodukt wird mit zuvor definierten Merkmalen verglichen, weiters werden Statistiken über Ausfallsquoten et cetera errechnet, um feststellen zu können, ob die Prozesse optimal verlaufen. Der prozessbasierte Ansatz ist bereits in der ISO 9001:2000 festgehalten, die Qualität definiert als Fähigkeit, Produkte herzustellen, die den Anforderungen von Kunden und Regulierungen genügen und die Abläufe zu wiederholen. Vorausgesetzt wird, dass ein Unternehmen von einem Kunden beauftragt wird, ein Produkt von Grund auf herzustellen und dass die Beschaffenheit des gewünschten Produkts genau spezifiziert wird. Das Unternehmen hat die Kontrolle über die Qualität, weil es die Baupläne bekommt oder vor Herstellung selbst konstruiert, die Ausgangsmaterialien beschafft und alle Bestandteile, Ausgangsmaterialien und Endprodukte überprüft. Diese Voraussetzungen treffen bei Sprachdienstleistungen und somit Lokalisierung nicht zu. Das Endprodukt wird nicht von Grund auf vom Lokalisierungsunternehmen hergestellt. Es kann auch bei der Beschaffung der Ausgangsmaterialien beim Lokalisierungsprozess nicht auf andere zurückgegriffen werden, da diese vom Kunden zur Verfügung gestellt werden. Zudem ist es schwierig zu spezifizieren, wie genau das Produkt des Lokalisierungsunternehmens, die an ein Locale angepasste Version einer Anwendung, aussehen soll, da der Kunde normalerweise in der Zielkultur keine Expertise aufweisen kann. Damit ist die Messung und Überprüfung der Qualität kein geradliniges Unterfangen. Ob die lokalisierte Version ausgeführt werden kann, kann natürlich überprüft werden, doch sind es ja eigentlich die sprachlichen und kulturellen Adaptionen, die vom Lokalisierungsunternehmen gemacht werden, also das Produkt, für das der Kunde bezahlt. Was nun grundsätzlich Qualität im Zusammenhang mit Übersetzung betrifft, wurde bereits besprochen. Wie kann nun auf Projektebene sichergestellt werden, dass diese Prozesse eingebaut werden und welche Faktoren sind dabei noch ausschlaggebend?

Grundsätzlich sollte die Arbeit innerhalb des Dienstleistungsunternehmens mit einem Qualitätsmanagementsystem kontrolliert werden. Laut der Norm DIN EN 15038:2006 müssen darin mindestens folgende Komponenten enthalten sein:

- „a) Die dokumentierte Zielsetzung des Qualitätsmanagementsystems.
- b) Ein Verfahren zur Überwachung der Qualität der Übersetzungsdienstleistungen und gegebenenfalls Korrekturmaßnahmen nach erfolgter Lieferung.
- c) Ein Verfahren zur Handhabung aller vom Kunden zur Verfügung gestellten Informationen und Unterlagen.“ (Herzog & Mühlbauer 2007: 18)

Dies bedeutet im Prinzip, dass die oben besprochenen Maßnahmen im Projektmanagement als Workflowschritte eingebaut werden. Es ist nun nicht nur wichtig, Qualität abzuliefern, sondern auch dem Kundenunternehmen deutlich zu machen, dass auf Qualität Wert gelegt wird. Dies geschieht mithilfe von transparenten Prozessen, die die Existenz dieses Systems und die dementsprechende Arbeitsweise aufzeigen.

Deshalb ist es vor Projektbeginn wichtig, die Anforderungen des Kunden genau zu verstehen und zu spezifizieren. Wie bei der Produktion eines Werkstücks, wird so sozusagen der Plan erstellt, mit dem das Produkt gemessen werden kann. Die Verantwortung für diesen Schritt obliegt dem Dienstleister, da dieser ja über die Expert/innen verfügt. Es kann vom Kunden nicht erwartet werden, dass er weiß, was für die bestmögliche Bearbeitung vonnöten ist, wie oben bereits erwähnt wurde. Mithilfe von genauen Listen sollten die Dienstleister abklären, wie die Anforderungen genau aussehen. Dazu gehören: Rezipient/innen, Zielmarkt, Medium, Zweck des Translats, Abgabefristen und Referenzmaterial, das den Übersetzer/innen zur Verfügung gestellt wird. Je besseres Referenzmaterial vorhanden ist, desto zielführender ist die Arbeit der Übersetzer/innen. Referenzmaterial umschließt: Paralleltexte und frühere Übersetzungen, Informationen über das Thema des Textes, Informationen über bevorzugte Terminologie beziehungsweise Glossare und/oder Terminologiedatenbanken. Da es in manchen Fällen vorkommen kann, dass Kunden Einwände bezüglich des Referenzmaterials haben und vielleicht vertrauliche Texte nicht herausgeben wollen, liegt es am Dienstleister, ein Vertrauensverhältnis herzustellen und klarzumachen, dass es für die Übersetzer/innen ungleich schwieriger ist, das gewünschte Ergebnis zu liefern wenn sie kein geeignetes Material haben.

Wenn die Kommunikation zwischen Kunden und Dienstleistern optimal verläuft, werden nicht nur die Anforderungen spezifiziert und unrealistische Erwartungen berichtigt, sondern als Resultat auch ersichtlich, welche Ressourcen das Dienstleistungsunternehmen benötigt. Kein Unternehmen sollte Aufträge annehmen, die es nicht bearbeiten und zu vollster Kundenzufriedenheit abschließen kann. Kein potenzieller Kunde wird die Professionalität des Dienstleisters anzweifeln, wenn dieser aus guten Gründen absagt.

Neben der Sicherstellung der Kompetenz der Übersetzer/innen ist Technologie ein Faktor, der Qualität merkbar beeinflusst. Nicht ohne Grund ist dieser Punkt bereits unter den Hindernissen aufgetaucht. Die technischen Ressourcen sind für jedes Unternehmen von Belang, im Lokalisierungs- oder Übersetzungsunternehmen bedeutet dies, die nötige Hardware und Software zur Verfügung zu haben, die für die Bearbeitung aller Aufträge nötig ist. Dazu gehören Computer, Server, CAT-Tools et cetera. Eine EDV-Abteilung sollte dafür sorgen, dass diese auf dem neuesten Stand sind und funktionieren, außerdem sollten Workflow-Schritte eingesetzt werden, um das Verlieren von Daten zu verhindern und Datenschutz und –sicherheit zu gewährleisten. Da für die meisten Projekte Dateien und Informationen über Netzwerke weitergegeben werden, ist ein funktionierendes Sicherheitssystem eine Grundvoraussetzung.

Wie aus der oben zitierten Norm hervorgeht, ist die Übersetzungsarbeit mit Workflows zu steuern. Dazu gehört zuerst die Kontrolle der zu bearbeitenden Dateien, also wie diese benannt werden, wo sie gespeichert werden und Versionskontrolle, sodass nicht unabsichtlich fertige Dateien mit früheren Versionen überschrieben werden. Es sollte klar vorgegeben werden, wie die Dateien an die Übersetzer/innen und Korrektor/innen weiterzugeben sind und wie und in welcher Form die bearbeiteten Dateien abgegeben werden. Im Prinzip

geht es darum, so genau wie möglich vorzuschreiben, welche Schritte aufeinanderfolgen und was wann zu tun ist. Es nutzt nichts, wenn die Arbeit der Übersetzer/innen makellos ist, wenn die Ergebnisse im Prozess verschwinden. Die Prozeduren müssen genau festgehalten werden, sodass nicht von einem/einer einzelnen Angestellten abhängt, ob die Abwicklung funktioniert oder nicht. So ist es auch einfacher, wenn neue Kräfte eingestellt werden, da dieses explizite Wissen weitergegeben werden kann.

Zu den Spezifikationen eines Projekts gehört natürlich auch, wie schließlich die Qualität des Ergebnisses sichergestellt werden kann. Es muss im Vorhinein abgeklärt werden, welche Maßnahmen ergriffen werden, ob Korrektor/innen vom Dienstleister beauftragt werden oder vom Kundenunternehmen, welche automatisierten Methoden angewandt werden und wie tiefgreifend die Überprüfung vorgenommen wird. Schlussendlich sollte noch geklärt werden, über welche Kanäle und auf welchen Medien die Lieferung der Ergebnisse erfolgt und wie der Erhalt bestätigt wird (vgl. Rueda 2010: 38-40).

Qualität ist von vielen Faktoren abhängig. Diese ziehen sich durch alle Ebenen, das heißt von den einzelnen Beteiligten über ihre Arbeitsergebnisse bis zu den Prozessen, die alle Schritte koordinieren ist Qualitätsmanagement ein umfassendes Konzept, das hohe Komplexität aufweist und sich im Laufe der Zeit auch immer wieder verändert. Dass es sich um einen wichtigen Aspekt handelt, ist wohl nicht abzustreiten, denn nur durch Qualität kann die Zufriedenheit der Endverbraucher/innen sichergestellt werden.

6 Technologie

Lokalisierung ist gewissermaßen die Verbindung von Sprache, Kultur und Technik. Deshalb ist es nicht verwunderlich, dass von Beginn an versucht wurde, den technischen Fortschritt im Sinne von Computersoftware für die Verbesserung der Arbeit an sich zu verwenden beziehungsweise Programme zu entwickeln, die Arbeitsschritte und Aufgaben automatisieren. Im Umfeld von Lokalisierung werden Anwendungen für vielerlei Aufgaben eingesetzt, natürlich, allen voran, bei der Softwareentwicklung, -internationalisierung und Vorbereitung auf Übersetzung und kulturelle Anpassung. Da diese Arbeit sich nicht mit Informatik beschäftigt, wird hier nicht näher auf Entwicklungs- und Programmierarbeit eingegangen. Die mittlerweile weit verbreiteten integrativen Lösungen für Projekt-, Workflow- und Qualitätsmanagement sowie Bürosoftware sind natürlich auch für Übersetzer/innen und andere Beteiligte im Lokalisierungsprozess relevant, würden jedoch in ihrer Vielfalt Stoff für eine Vielzahl von Abschlussarbeiten liefern. Von einer tieferen Auseinandersetzung damit wird in dieser Arbeit abgesehen. In diesem Kapitel liegt der Fokus auf Anwendungen, die im Arbeitsalltag der Übersetzer/innen in der Lokalisierungsbranche auftauchen: Translation Memory-Systeme, Terminologie-Tools, Lokalisierungstools und maschinelle Übersetzungssoftware. Ferner wird ein Überblick über die Entwicklung dieser Software und über neuere Ansätze gegeben. In diesem Zusammenhang sollte erwähnt werden, dass die Weiterentwicklung zu diesem Zeitpunkt nicht anhält und dass die Entwicklung neuer Tools auch in Zukunft spannende Lösungen mit sich bringen wird. Es werden schließlich auch die relevanten Standards und Formate besprochen, die die Zusammenarbeit und Vernetzung erleichtern beziehungsweise erst möglich machen.

Spätestens seit CAT-Tools und maschinelle Übersetzung weitere Verbreitung gefunden haben, gibt es selten Lösungen, die exklusiv nur ein Feature enthalten. Dies ist natürlich sinnvoll, da Translation Memory-Tools selten ohne Terminologie-Tool benutzt werden und es unpraktisch wäre, beide Programme nebeneinander laufen zu lassen und ständig umschalten zu müssen. Deshalb sollen nicht Softwarepakete, die auf dem Markt sind, besprochen werden, sondern die einzelnen Tools selbst beschrieben werden. Grundsätzlich wird in diesem Bereich zwischen Tools unterschieden, die Übersetzer/innen bei der Arbeit unterstützen, sogenannten CAT-Tools (computer-assisted translation oder computer-aided translation), und maschineller Übersetzung, die die Arbeit der Übersetzer/innen übernehmen.

6.1 Translation Memory-Tools

Translation Memorys sind, wie der Name schon sagt, Übersetzungsspeicher. Der Sinn und Zweck dieser Anwendungen ist die Speicherung von Übersetzungen in Datenbanken, sodass diese später selbst oder von anderen wiederverwendet werden können. Dazu wird der eingelesene Text in Segmente, die kleinsten übersetzbaren Einheiten, unterteilt. Diese bestehen zumeist aus Sätzen oder Wörtern, können aber auch andere sprachliche Einheiten sein, wie zum Beispiel Kapitelüberschriften oder Listenelemente. Die Unterteilung in Segmente erfolgt üblicherweise nach Punkten, Strichpunkten und Zeilenumbrüchen. Diese Segmentierungsregeln können spezifiziert und angepasst werden. So wird, zum Beispiel, verhindert, dass nach Punkten, die als Dezimaltrennzeichen verwendet werden, ein neues Segment beginnt oder man teilt einen Text auf ganze Absätze auf.

Bei der Übersetzung werden die einzelnen Segmente übersetzt und die Ausgangssprachlichen mit den Zielsprachlichen Segmenten verbunden. Wird ein bestehendes Translation Memory verwendet, vergleicht das Programm die Segmente des neuen Textes mit bereits bestehenden Segmenten aus der Datenbank und findet so „Matches“ oder „Fuzzy Matches“. Der Grad der Übereinstimmung wird in Prozent angegeben. 100 %-Matches können meistens übernommen werden, es kann jedoch sein, dass der Kontext ein gänzlich anderer ist und die vorgeschlagene Übersetzung des Segments in diesem Fall nicht korrekt wäre. Je kürzer ein Segment ist, desto mehr ist darauf zu achten. Fuzzy Matches, also Segmente, die teilweise übereinstimmen, sind nur ab einem bestimmten Prozentsatz wertvoll, da einige wenige Prozent rasch erreicht werden, zum Beispiel durch Artikel oder bestimmte Verbformen.

Ein wichtiger Bestandteil von Translation Memory-Tools sind Filter. Diese konvertieren importierte Dateien verschiedener Formate in ein Format, das vom Programm bearbeitet werden kann, wie zum Beispiel XLIFF oder LCX. Es gibt eine Reihe von Dateiformaten, für die standardmäßig in den meisten Translation Memory-Tools Filter enthalten sind, so wie XML/HTML/SGML, RC, RTF, DOC oder DOCX. Der Filter erkennt sozusagen das Format und gegebenenfalls Kodierung und Tags und schützt diese vor unabsichtlicher Veränderung. Filter können üblicherweise für nicht-Standard-Formate selbst erstellt werden. Nach dem Bearbeiten des Dokuments wird dieses wieder ins ursprüngliche Format exportiert.

Ein Feature, das alle Translation Memory-Tools enthalten, ist die Berechnung von Statistiken. Es werden dabei Wörter, Dateien, externe und interne Matches gezählt. Externe Matches sind Übereinstimmungen mit Segmenten aus bestehenden Translation Memorys, interne Matches sind Wiederholungen im Dokument. Dabei werden auch Fuzzy Matches unterschiedlichen Grades angegeben, normalerweise jene zwischen 50 und 99 %. Interessant ist auch die Angaben von Wörtern und Segmenten, für die keine Matches gefunden werden. Die Statistik ist vor allem für die Berechnung des Kostenvoranschlags wertvoll. Ein Problem hierbei ist, dass verschiedene Tools, manchmal sogar verschiedene Versionen eines Herstel-

lers, zu unterschiedlichen Wortanzahlen kommen. Deshalb sollten, wenn möglich, mehrere Tools für die Errechnung der Statistik verwendet werden.

Translation Memory-Tools sind nur für die Übersetzung von bestimmten Texten zu gebrauchen. Einsparungen können nur dann erzielt werden, wenn es sich um fachliche Texte mit vielen Wiederholungen handelt. Dies ist bei Lokalisierungsprojekten der Fall. Einerseits gibt es oft innerhalb der Dokumente Wiederholungen, andererseits wird Software oft upgedatet, das heißt es muss bei diesen Projekten nur neu übersetzt werden, was neu hinzukommt. Außerdem führt die Verwendung von Translation Memory-Tools dazu, dass die Sprache über das gesamte Produkt oder die Produktkette kohärent ist; dies gilt auch für die verschiedenen Bestandteile innerhalb eines Projekts. Durch die Weitergabe der Translation Memorys können mehrere Übersetzer/innen von bereits bestehenden Übersetzungen profitieren. Außerdem können mittels Translation Memory-Systemen viele Dokumente bearbeitet werden, für die sonst ein eigener Editor nötig wäre. Durch die Filter werden nur die übersetzungsrelevanten Texte innerhalb eines Dokuments angezeigt, wodurch erstens der/die Übersetzer/in nicht mit dem Code vertraut sein muss und diesen zweitens nicht unabsichtlich verändern kann. Durch die Wiederverwendung und diese anderen Vorteile werden Produktivitätssteigerungen um die 30 %, manchmal sogar bis zu 50 % beobachtet. Diese Steigerung hängt natürlich von vielen Faktoren ab, wie der Vertrautheit des/der Übersetzer/in mit dem Programm, der bestehenden Translation Memorys und der Beschaffenheit der Texte. So sind fachliche Texte gut zu bearbeiten, bei Marketingmaterial oder literarischen Texten ist die Verwendung nicht zu empfehlen, da erstens die Match-Quote nicht sehr hoch sein wird und zweitens der Überblick über den Gesamttext dafür nötig ist, der bei der segmentierten Ansicht im Translation Memory-Programm wegfällt.

Einige Faktoren erschweren die Arbeit mit Translation Memory-Programmen. Durch den Import ins Programm werden die Formatierungen und das Layout nicht angezeigt, weshalb eine nachträgliche Überprüfung der Übersetzungen im ursprünglichen Format notwendig ist. Effiziente Verwendung von Translation Memory-Systemen setzt auch gewisse Prozesse voraus. So sollte geregelt werden, wer wie Zugriff auf die Translation Memorys hat und auch nachträgliche Korrekturen am übersetzten Dokument im Translation Memory nachgetragen werden. Probleme können auch entstehen, wenn Filter für gewisse Formate nicht vorhanden sind, was vor allem dann geschehen kann, wenn sich neue Formate auf dem Markt schnell verbreiten und die Hersteller der TM-Systeme die Filter nicht rasch nachliefern (vgl. Esselink 2000: 362-367).

Grundsätzlich sind Translation Memory-Tools heutzutage aus der Übersetzungsarbeit in bestimmten Bereichen nicht mehr wegzudenken. Es sollte jedoch nicht angenommen werden, dass diese Programme immer und überall mit Zeit- und Kosteneinsparungen verbunden sind. Erstens kommt es, wie oben beschrieben, auf die Art der Texte an, ob der Einsatz von Translation Memory-Tools überhaupt sinnvoll ist und zweitens garantieren auch scheinbar gute Voraussetzungen nicht die erwarteten Ergebnisse. Orfall (2010: 30-33) schildert an einer

Fallstudie, wie der Einsatz von Translation Memory-Tools bei einem Projekt mit mehreren Sprachen zu sehr stark variierenden Kosten führte. Bei dem Projekt sollten Betriebsanleitungen aktualisiert werden, das heißt, nur Teile der Dokumente übersetzt werden. Die Unterschiede zwischen den Honoraren für die Übersetzungen in die verschiedenen Sprachen ergaben sich daraus, dass manche der Übersetzer/innen für 100 %-Matches auch Geld verlangten. Dies ist durchaus üblich, da ja Bearbeitungszeit für die Übersetzer/innen anfällt und außerdem ihre Investitionen gedeckt werden müssen, die später durch den Einsatz der CAT-Tools zu Produktivitätssteigerungen und niedrigeren Gesamtkosten für die Kunden führen. In besagtem Fall war jedoch in sehr großen Dokumenten nur sehr wenig zu verändern, weshalb fast der gesamte Text als 100 %-Matches bestätigt wurde. Es wäre also günstiger gewesen, die neuen Teile aus dem Text herauszusuchen und auf traditionelle Art und Weise übersetzen zu lassen. Dies ist natürlich nicht die Regel und, wie schon erwähnt wurde, führt der Einsatz von Translation Memory-Tools üblicherweise zu Produktivitätssteigerungen. Es zeigt jedoch, dass die Evaluierung eines Projekts unbestritten wichtig ist und auch der Einsatz von CAT-Tools genau bedacht werden sollte.

6.2 Terminologie-Tools

Esselink (2000: 379) fasst den Begriff Terminologie-Tool relativ weit und nennt auch Glossar-Tools für die Verwaltung von umfangreichen mehrsprachigen Glossaren und elektronische Wörterbücher. Diese werden natürlich nach wie vor verwendet. Wird der Begriff Terminologie-Tool oder Terminologiesoftware hier gebraucht, bezieht er sich jedoch ausschließlich auf Terminologiemanagementsoftware.

Kommerzielle Terminologieprogramme gibt es seit fast 30 Jahren (vgl. Zetzsche 2012: 50). Die Bedeutung von Terminologie-Tools wurde bereits in Kapitel 5.2 erläutert, deshalb soll hier nicht mehr auf die Vorteile eingegangen werden. Es ist jedoch unumstritten, dass Terminologiesoftware zu den Grundprogrammen gehört, die beim Fachübersetzen verwendet werden. Es finden natürlich auch ein- und mehrsprachige Glossare Verwendung, umfassende Terminologiearbeit ist jedoch ohne die passende Software schwer möglich, da der begriffsorientierte Aufbau und die Vielzahl an Vernetzungen sowie die Optimierung der Workflows bei Excel- oder Wordlisten nicht funktioniert. Terminologie-Tools sind meistens in andere CAT-Tools integriert oder im Umfang des Programmpakets enthalten. So können Einträge zumeist in Translation Memory-Programmen über eigene Fenster gesucht werden oder die Einträge werden automatisch vorgeschlagen.

Zusätzliche Funktionen, die neben der Eintragserstellung und -verwaltung in Terminologie-Tools enthalten sind, sind Terminologie-Extraktion und Terminologieüberprüfung. Erstere durchsucht automatisch Dokumente nach Wörtern, die für Einträge infrage kommen. Trotzdem die Algorithmen dieser Programme laufend verbessert werden, sollten jene Begriff-

fe, die tatsächlich in die Terminologiedatenbank aufgenommen werden, auf jeden Fall von Terminologieexpert/innen oder Übersetzer/innen ausgewählt werden. Ein Feature, das bereits im Zusammenhang mit Qualitätskontrolle genannt wurde, ist die Überprüfung der kohärenten Verwendung der Terminologie. Vor allem wenn die Terminologiedatenbanken auch verbotene Termini enthalten, kann so sichergestellt werden, dass nur die richtigen im Text vorkommen.

Da der Zweck von Terminologearbeit die kohärente Verwendung der Termini im gesamten Produkt beziehungsweise der ganzen Produktkette ist, sind der Austausch und die Vernetzung von Datenbanken ein bedeutender Faktor. Mithilfe von Standards kann garantiert werden, dass Übersetzer/innen unabhängig vom Tool, das sie verwenden, Datenbanken importieren können (vgl. Esselink 2000: 379-381, Fidura 2007: 41; Zerfaß 2008: 9).

6.3 Lokalisierungstools

Lokalisierungstools sind, wie der Name schon sagt, Programme, die eigens für die Lokalisierung von Software entwickelt werden. Damit können Ressourcendateien, aber auch binäre Dateiformate direkt geöffnet werden und so die Bestandteile der Benutzeroberfläche innerhalb einer WYSIWYG-Oberfläche (Echtbilddarstellung, Akronym von „What You See Is What You Get“) übersetzt werden. Dies hat den bedeutenden Vorteil, dass die Übersetzer/innen im Kontext arbeiten können. Außerdem kann in diesen Tools auch die Benutzeroberfläche selbst verändert werden, also zum Beispiel Dialogfelder und Menüs für übersetzte Texte, die länger als die Ausgangstexte sind, vergrößert werden.

Lokalisierungstools können Dateiformate öffnen, die sonst nur mit Ressourceneditoren bearbeitet werden können, zum Beispiel RC, EXE oder DLL. Da sie aber für Sprachdienstleistungen konzipiert werden, enthalten sie Features, die auf diese Arbeit zugeschnitten sind. So gibt es meistens Schnittstellen zu Translation Memory-Tools und Terminologiesoftware, Berechnung der Statistiken, Rechtschreibüberprüfung und andere automatische Überprüfungsfunktionen. Dazu gehört, zum Beispiel, Pseudo-Übersetzung, ein Feature, das im vorigen Kapitel bereits beschrieben wurde.

Der Hauptvorteil ist wohl, dass die ausführbaren Binärdateien nicht dekompiert werden müssen, um die ursprünglichen Ressourcendateien zu erhalten. So können Arbeitsschritte eingespart werden. Wie auch bei Translation Memory-Programmen kann der/die Übersetzer/in nur den Text bearbeiten während der Code geschützt bleibt, was Softwarebugs im lokalisierten Programm verhindert. Außerdem sind Ressourceneditoren üblicherweise nur als Teil einer Entwicklungsumgebung verfügbar, was bedeutet, dass relativ große Pakete installiert werden müssen, die Funktionen enthalten, die Übersetzer/innen nicht brauchen. Lokalisierungstools sind entweder einzeln erhältlich und somit kleiner im Umfang, oder als Teil von Programmpaketen, die für Übersetzer/innen und Lokalisierungsexpert/innen entwickelt wur-

den. Somit kann ein/e User/in bei der Arbeit die verschiedenen Funktionen der Programmteile nutzen und kombinieren (vgl. Esselink 2000: 383, Zerfaß 2008: 8).

6.4 Sonstige Tools und Features

Abgesehen von den oben beschriebenen Kernfunktionen gibt es noch einige Tools, die teilweise als alleinstehende Programme oder aber auch als Features innerhalb von Programmpaketen erhältlich sind. Der Zweck dieser Tools ist die Automatisierung von Arbeitsschritten, sodass sich Übersetzer/innen auf ihre Kernaufgabe konzentrieren können. Ebenso wie Translation Memory-Tools das übersetzerische Gedächtnis erweitern, können viele mühsame Schritte auch von diesen Tools übernommen werden.

Eine dieser Aufgaben ist „Alignment“, oder die Nebeneinanderstellung von Ausgangs- und Zieltexten. Wird ein Text nicht mithilfe eines TM-Programms übersetzt, sind die Segmentpaare natürlich nicht in einem Translation Memory gespeichert und können so lediglich als Referenz eingesetzt werden. Deshalb gibt es die Alignment-Funktion, die die Texte nebeneinanderstellt, segmentiert und dann versucht, die Segmentpaare zu bilden. Je mehr die Textstruktur des Zieltextes der des Ausgangstextes entspricht, desto besser kann das Programm die Segmente verknüpfen, da es ja nicht die Bedeutung der Sprache versteht, sondern lediglich die Punktierung und Formatierung interpretiert. Wurden ganze Absätze oder Sätze umgeformt, so werden keine guten Ergebnisse erzielt werden. Grundsätzlich sollten die aus der Alignment-Funktion entstandenen Translation Memorys von Übersetzer/innen oder anderen Expert/innen überprüft werden. Heute ist diese Funktion meistens als Features in Softwarepaketen enthalten, früher waren oft alleinstehende Tools erhältlich.

Konvertierungstools und Textextraktions-Tools zählen nicht unbedingt zu den CAT-Tools, sind jedoch Programme, die auch für Übersetzer/innen nützlich sind. Bei Aufträgen von verschiedenen Kunden erhält man oft die unterschiedlichsten Formate, die mit den vorhandenen Tools nicht zu bearbeiten sind, vor allem wenn es sich um proprietäre Dateiformate handelt. Hier ist es notwendig, diese Dokumente zuerst zu konvertieren oder den Text herauszuholen. Nach der Übersetzung müssen diese Dokumente wieder in die ursprünglichen Formate rückkonvertiert werden.

Das Zählen der Wörter im Text ist eine Funktion, die prinzipiell in der Statistikfunktion enthalten ist. Es gibt jedoch auch eigenständige Tools, falls kein TM-Programm oder ähnliches verwendet wird, oder aber, um deren Angaben überprüfen zu können. Wie oben erwähnt, liefern verschiedene Programme die unterschiedlichsten Resultate. Es gibt mittlerweile auch für den „Word Count“ einen Standard, der in Kapitel 6.4 näher besprochen werden.

Andere Features fallen in die Sparte Qualitätsüberprüfung und wurden bereits in Kapitel 5.4.2.3 erwähnt. Meistens auch schon in den Kerntools enthalten sind Rechtschreib- und Grammatiküberprüfung sowie Überprüfung auf die richtigen Zahlenformate und Punktierung.

Da Übersetzungen im Lokalisierungsbereich üblicherweise in größere Projekte eingebunden sind, kommen Übersetzer/innen auch mit Projektmanagementsoftware und Workflow-Tools in Berührung. Mit Projektmanagement-Tools können Arbeitspakete oder Translation-Kits erstellt werden, die alle nötigen Informationen und Referenzmaterial für die Übersetzer/innen enthalten. Workflow-Tools sind einerseits für die Automatisierung von Arbeitsschritten verantwortlich, aber auch für deren Verknüpfung. So können, zum Beispiel, fertiggestellte Übersetzungen hochgeladen werden und werden über das Tool automatisch an den/die Korrektor/in weitergeleitet, um den nahtlosen Übergang zwischen den Schritten zu gewährleisten. Heutzutage sind Übersetzungssoftware-Komplettlösungen oft mit Workflow- oder Projektmanagement-Tools verbunden oder enthalten diese (vgl. Zerfaß 2008: 8-9)

Es gibt natürlich noch eine Vielzahl an Features in den neueren Versionen der Softwarepakete. Als Beispiel seien die Konkordanzsuche, „PerfectMatch“, „AutoSuggest“, „Context Matching“ und „QuickPlace“ aus der SDL Trados Studio-Reihe genannt. Ersteres ist die Suche nach Segmenten im verwendeten Translation Memory falls kein Match vorgeschlagen wird. Das zweite Feature speichert angenommene Segmentpaare, sodass diese, wenn sie in einem Dokument noch einmal vorkommen, nicht zusätzlich überprüft werden müssen. „AutoSuggest“ schlägt Übersetzungen für Segmente vor und verbindet dabei zielsprachliche Segmente aus dem Translation Memory mit Einträgen aus der Terminologiedatenbank. „Context Matching“ ist für Matches, die im Prinzip über 100 % übereinstimmen, wenn die vorherigen und nachfolgenden Matches auch dieselben sind und „QuickPlace“ wird für die automatische Übertragung der Tags und Formatierungen aus dem Ausgangssprachlichen ins Zielsprachliche Segment verwendet (vgl. Waßmer 2011: 14-15). Welche Features nun zu den Kernfunktionen zählen und welche eigene Tools darstellen, ist immer von den Programmpaketen abhängig. Viele Features, die ursprünglich als eigene Tools angeboten wurden, sind mittlerweile bei den bekannten Komplettlösungen standardmäßig enthalten.

6.5 Maschinelle Übersetzung

Ab dem Zeitpunkt, an dem die Pionier/innen der Computerentwicklung ernstzunehmende Resultate hervorbrachten, wurde versucht, Systeme zu entwickeln, die menschliche Übersetzer/innen ersetzen. Da man damals Übersetzung als Transkodierung sah, schien es logisch, ein Programm mit dem „Code“, der Sprache, zu füttern und dann anstatt eines/einer Übersetzer/in den Code auszutauschen. In den 1950ern und 1960ern war man zuversichtlich und hoffte darauf, innerhalb von ein paar Jahren Rechnerleistungen hervorzubringen, die dies bewerkstelligen könnten. Da es beim Übersetzen nicht um das bloße Ersetzen von Wörtern und Sätzen geht, gelangte man nie an den Punkt, an dem wirklich brauchbare Ergebnisse erzielt wurden. Deshalb wurden schließlich die meisten Forschungsprogramme eingestellt. Die Forschung wandte sich anderen Möglichkeiten zu, den Computer für Übersetzer/innen nütz-

lich zu machen, was nach und nach, ab Mitte der 1980erjahre, die oben beschriebenen Tools hervorbrachte. In den 90erjahren ist maschinelle Übersetzung auf die Bildfläche zurückgekehrt, zuerst mit kommerziellen Programmen, später erfolgte die weitere Verbreitung im Internet. Mittlerweile wird maschinelle Übersetzung an vielen Orten eingesetzt, jedoch hat sich der Blickwinkel geändert: Anstatt den Computer die gesamte Arbeit machen zu lassen, wurde das Potenzial erkannt, das in den nicht perfekten Ergebnissen steckt. Im Folgenden werden die beiden Systeme besprochen, die sich durchgesetzt haben und der Einsatz von maschineller Übersetzung in der Praxis erläutert.

6.5.1 Regelbasierte und statistische maschinelle Übersetzungssysteme

Die verbreiteten Systeme werden grundsätzlich in drei Typen eingeteilt: regelbasierte maschinelle Übersetzung, statistische maschinelle Übersetzung und Hybridsysteme, die beide vereinen.

Regelbasierte Systeme funktionieren, wie der Name schon sagt, mit sprachlichen Regeln. Es wird ein kompliziertes Regelsystem erarbeitet, sodass das Programm die Grammatik und Rechtschreibung von einem Sprachpaar lernt. Der Wortschatz stammt aus (Fach-) Wörterbüchern. Diese Kombination hat den Vorteil, dass man durch die Wörterbücher und Terminologiedatenbanken die Kohärenz der Terminologie gut steuern kann. Rollenbasierte Systeme müssen immer für ein Sprachenpaar entwickelt werden. Diese Art von Übersetzungssystemen gibt es schon länger als statistische Systeme und ist momentan auf dem Markt stärker vertreten. Statistische Systeme arbeiten nicht mit vorgefertigten Regeln, sondern mit statistischen Modellen, deren Parameter aus der Analyse von Text stammt. Es sind große Textkorpora nötig, um das Programm zu trainieren. Mit beiden Ansätzen können gute Ergebnisse erzielt werden, abhängig von den Voraussetzungen. Thicke (2010: 42) beschreibt den Unterschied mit folgendem Beispiel: Die Phrase „the black cat“ soll auf Französisch übersetzt werden. Das regelbasierte System erkennt die Wortarten und die Konstellation und übersetzt auf „le chat noir“. Das Hauptwort und das Adjektiv tauschen Platz, weil das regelbasierte System die Wortstellung der französischen Grammatik kennt. Ein statistisches System übersetzt höchstwahrscheinlich auch korrekt, weil anzunehmen ist, dass diese bekannte Kombination im Textkorpus bereits aufgetaucht ist. Ersetzt man nun das Adjektiv mit „blue“, wird ein regelbasiertes System dies wieder korrekt übersetzen. Bei statistischen Systemen besteht jedoch die Möglichkeit, dass die Phrase im Trainingskorpus noch nicht vorgekommen ist und übersetzt möglicherweise zwar die Wörter korrekt, ändert die Satzstellung aber nicht, mit dem Ergebnis „le bleu chat“.

Das Beispiel zeigt, dass sich die beiden Ansätze grundlegend unterscheiden. Es gibt noch einige Vor- und Nachteile: Die Zeit, die gebraucht wird, um ein maschinelles Übersetzungsprogramm zu installieren und funktionsfähig zu machen, ist bei regelbasierten Systeme-

men deutlich kürzer, da es ja bereits „fertig“ ist. Bevor ein maschinelles Übersetzungsprogramm produktiv verwendet werden kann, muss es jeweils „trainiert“ werden, also an die Fachsprache et cetera angepasst. Bei regelbasierten Systemen muss also Terminologie hinzugefügt werden. Statistische Systeme brauchen relativ große Mengen an Texten, sodass man die Terminologieverwendung nicht gut steuern kann, außerdem ist es natürlich möglich, dass Korpora solcher Größe nicht vorhanden sind. Durch diese Analyse und Verarbeitung sind nicht nur Softwareexpert/innen nötig, sondern auch relativ hohe Rechnerleistungen. Dafür sind die Ergebnisse von statistischen Systemen meist flüssiger, die Satzstrukturen natürlicher. Ein weiterer großer Vorteil ist die Möglichkeit, Sprachen hinzufügen zu können und auch wie benötigt zu verknüpfen. Dies funktioniert bei regelbasierten Systemen nicht, da komplexe Regelsysteme beider Sprachen erstellt und verbunden werden müssen.

Hybridsysteme sind eine Kombination aus beiden, sodass die jeweiligen Vorteile genutzt werden können. Es wird, zum Beispiel, mit einem regelbasierten System eine Übersetzung angefertigt, die danach mit einem statistischen Übersetzungsprogramm automatisch nachbearbeitet wird. Das Ergebnis ist eine Senkung der benötigten Nachbearbeitung durch Personen und somit weniger Arbeitszeit. Die Effizienzsteigerung ist im Prinzip der Zweck des Einsatzes von maschineller Übersetzung (vgl. Tatsumi & Sun 2008: 22-33; Kuhn et al 2010: 43-46; Thicke 2010: 42).

6.5.2 Einsatz von maschineller Übersetzung in der Praxis

Anstatt von einer maschinellen Übersetzungs-Engine perfekte Ergebnisse zu erwarten, geht es heutzutage darum, die von Menschen erbrachte Arbeitszeit zu verkürzen. Es gibt natürlich Fälle, bei denen maschinelle Übersetzung eingesetzt wird, um die grundlegenden Informationen eines fremdsprachigen Textes zu verstehen, sogenanntes „gisting“. Material, das veröffentlicht werden soll, wird jedoch normalerweise nachbearbeitet.

„Post-editors“ suchen Fehler und unpassende Formulierungen im Output des Übersetzungssystems, stellen diese richtig und produzieren so einen funktionierenden Zieltext. Nachbearbeiter/innen sind meistens Übersetzer/innen (vgl. Guerberof Arenas 2010: 34-35). Es gibt jedoch Versuche mit Personen, die die Ausgangssprache nicht beherrschen. Dabei werden auch gute Resultate erzielt (vgl. Koehn 2010). Übersetzer/innen haben jedoch den Vorteil, dass sie über mögliche Fehlerquellen Bescheid wissen und bei gänzlich unverständlichen Formulierungen auf den Ausgangstext zurückgreifen können. Auch Übersetzer/innen sollten in die Arbeit mit maschinell übersetzten Texten eingeschult werden. Zusätzlich ist der Einsatz von Post-Editing-Richtlinien zu empfehlen. Diese Guide-Lines unterscheiden sich von Style-Guides und anderen anweisenden Dokumenten. Sie sollten eine Beschreibung des verwendeten Systems enthalten, Informationen über den Ausgangstext, Qualitätserwartungen des Kunden, Szenarien, die das Verwerfen eines Segments erfordern, typische Fehler des

Outputs der Sprachkombination, Angaben zum Stil und Angaben zur Terminologie (vgl. Guerberof Arenas 2010: 35-36).

Eine andere Art, maschinelle Übersetzung einzusetzen, ist die Einbindung in CAT-Tools. Es gibt mittlerweile viele Lösungen, die eine Schnittstelle zu maschineller Übersetzung enthalten. Das bedeutet, dass Segmente des Ausgangstextes, für die keine verwertbaren Matches gefunden wurden, von maschinellen Übersetzungssystemen vorübersetzt werden (vgl. Cattelan 2014: 40-41).

Die Brauchbarkeit der Resultate und die tatsächliche Zeitersparnis hängen von der idealen Kombination aus dem richtigen System, der Vorbereitung der Engine, der Beschaffenheit des Ausgangstexts und der Kompetenz und verfügbaren Zeit der Nachbearbeiter/innen ab. Manche Sprachkombinationen funktionieren mit regelbasierten Systemen gut, manche mit statistischen. Wenn der Ausgangstext eher umgangssprachlich geschrieben wurde, sind statistische Systeme besser. Dafür lassen sich gute Ergebnisse mit regelbasierten Systemen erzielen, wenn der Ausgangstext speziell für maschinelle Übersetzung optimiert wurde. Dies fällt in den Bereich kontrollierte Sprache: Manche Satzkonstruktionen sind nicht eindeutig zu verstehen und verlangen deshalb die Interpretation des Kontexts, etwas, was bislang nur Menschen können. Viele Sätze und Phrasen können aber umformuliert werden, sodass sie eindeutig sind und später besser maschinell übersetzt werden können. Dazu gehört, zum Beispiel, die Vermeidung von Passivkonstruktionen und Pronomen, deren Bezug nicht eindeutig ist. Natürlich ist dies stark von der Sprache abhängig.

Ein Kriterium ist auch die Qualität des maschinellen Translats. Davon hängt die Länge der Arbeitszeit ab, die Nachbearbeiter/innen benötigen. Wenn eine Engine nicht passend vorbereitet beziehungsweise trainiert wurde, oder von vornherein nicht die richtige Lösung ist, ist das Resultat meistens mehr als mangelhaft. Dies frustriert nicht nur die für die Nachbearbeitung zuständigen Personen, sondern verlangt auch einen erheblichen Mehraufwand an Zeitressourcen. Das bedeutet im Endeffekt, dass auch hier die Evaluierung des Projekts fundamental für den effizienten Ablauf ist und es von vielen Kriterien abhängt, welche automatischen Lösungen eingesetzt werden können (vgl. Thicke 2011: 37-40; 2013: 42-45)

6.6 Entwicklung

Die Geschichte von Software, die direkt mit Übersetzung zu tun hat, hängt mit der Computergeschichte an sich zusammen. Wie Wyld (2010: 44) schildert und auch in Kapitel 2 beschrieben wurde, gab es alle zehn Jahre einen Paradigmenwechsel: Um das Jahr 1945 begann die tiefere Forschung und Entwicklung im Computerbereich, was dazu führte, dass das englische Wort „computer“ nicht mehr professionelle Dienstleister/innen bezeichnete, die Rechenaufgaben durchführten, sondern Maschinen. In den Sechzigerjahren begann die Ära der Großrechner, gefolgt von den Minicomputern der Siebziger und dem PC in den Achtzigerjah-

ren. In den Neunzigern hob das Internet den Computer auf eine andere Ebene und im neuen Jahrtausend begann sich der Computer in Form von tragbaren Geräten noch weiter auszubreiten, verbunden mit einem steilen Anstieg der Verfügbarkeit und der Geschwindigkeit des Internets.

In den ersten Jahrzehnten wurde zunächst auf maschinelle Übersetzung gesetzt. Ab den 1980er Jahren lag der Fokus hauptsächlich auf computergestützter Übersetzung, als mit den Terminologie-Tools die ersten CAT-Tools entwickelt wurden und auf den Markt kamen. Dies wäre natürlich ohne die Verbreitung des PCs am Arbeitsplatz nicht möglich gewesen. Anfang der Neunziger begann der Vertrieb der ersten Translation Memory-Systeme. Diese Produkte erforderten große Investitionen und zielten grundsätzlich auf die Verwendung in großen Unternehmen ab. Einige wenige Ausnahmen sorgten aber auch für die Verbreitung von TM-Tools unter Freelancern und letztendlich wurde es zur Regel, CAT-Tools zu verwenden. Mit dem Internet und der Entwicklung von Standards wurde es immer leichter, das Potenzial dieser Software auszuschöpfen, indem man Datenbanken und Übersetzer/innen vernetzte. Gleichzeitig wurden auch die Softwarepakete immer umfassender. „Translation Management Systems“ und „Translation Environment Tools“ kombinieren alle möglichen Arten von CAT-Tools und maschineller Übersetzung. Dies hat den Vorteil, dass man erstens in einem Programmpaket alles enthalten hat und nichts zukaufen muss und, dass für Kompatibilität dieser Tools gesorgt ist (vgl. Wyld 2010: 44-45; Zydroń 2012: 20; Zetzsche 2012: 50-51).

Der nächste logische Schritt ist die „Cloud“. Dieser Begriff umschließt die Verschiebung von Anwendungen und Inhalten in das Internet. Dazu gehören verschiedene Dienste, wie „Software as a Service“ (SaaS), „Platform as a Service“ (PaaS) und „Infrastructure as a Service“ (IaaS), die verschiedene Grade von Dienstleistungen umfassen. Bei SaaS wird die gesamte Umgebung, inklusive Anwendungen, von einem Anbieter zur Verfügung gestellt, bei PaaS die Umgebung, jedoch nicht die Anwendungen und bei IaaS nur mehr die Infrastruktur. Im Endeffekt bezahlt man für die Dienste, die man in Anspruch nimmt und muss die Informationstechnologie nicht mehr selbst aufbauen und warten (lassen). Die einzige Voraussetzung ist ein Gerät, das ständig mit dem Internet verbunden ist. Diese Verbindung setzt natürlich die nötige Infrastruktur voraus: fällt der Server aus, steht auch die Arbeit (vgl. Wyld 2010: 44-45; Gambín 2014: 46-47).

Für Übersetzungs- und Lokalisierungsdienstleister bietet die Cloud und SaaS gewisse Vorteile. Der Erwerb der EDV-Infrastruktur bringt hohe Kosten mit sich. Neben der Hardware und der Bürosoftware erfordern CAT-Tools enorme Investitionen für Lizenzen. Diese müssen immer wieder erneuert werden, erstens wegen des Supports und zweitens, weil immer wieder neue Dateiformate auftauchen, die von älteren Versionen nicht mehr bearbeitet werden können. Es ist natürlich nicht möglich, diese Investitionen nicht zu tätigen, wenn die betreffende Anwendung über längere Zeit nicht verwendet wird. Nimmt man SaaS in Anspruch, muss man sich in der Regel nicht über mehr als drei Monate verpflichten und kann so

sicherstellen, dass man nicht für mehr bezahlt, als man tatsächlich verwendet. Dies ist vor allem für kleinere Unternehmen ein bedeutender Vorteil. Für große Unternehmen zahlt es sich meistens aus, in die nötige Software zu investieren, außerdem verfügen sie auch über die nötigen Mittel. Kleineren Unternehmen fehlt meistens dieser Spielraum und dadurch, dass sie nur für die tatsächlich in Anspruch genommenen Dienste bezahlen, können sie effizienter wirtschaften. Die Installation von Updates und die ständige Wartung werden auch vom Anbieter übernommen, was dadurch möglich ist, da alles zentral gesteuert wird. Für die Übersetzer/innen fällt die Installation weg und die Zusammenarbeit mit anderen Beteiligten ist viel effizienter, da die zu bearbeitenden Dokumente online gespeichert sind. So müssen diese nicht mühsam per E-Mail verschickt werden, außerdem können mehrere Beteiligte direkt auf die Dokumente zugreifen und deren Aufteilung und Zusammenfügung fallen als Arbeitsschritte auch weg. Es muss auch nicht mehr auf maximale Dateigrößen geachtet werden (vgl. Zydrón 2012: 20-21). Cloud Computing eröffnet auch für Crowdsourcing neue Möglichkeiten. Ein zentraler Faktor bei Crowdsourcing ist die Infrastruktur. SaaS bietet umfangreiche Komplettlösungen, die dafür effizient genutzt werden können (vgl. Roland 2014: 21-23).

Der Schritt in die Cloud birgt natürlich auch Risiken. Dadurch, dass alles „irgendwo im Netz“ gespeichert ist, ist dieses System anfälliger für Hacks. Andererseits könnte man einwenden, dass die Vernetzung über das Internet auch für Desktop-basierte Software und internationale Zusammenarbeit schon seit längerem Fakt ist und zentrale Datenbanken und die Kommunikation via E-Mail auch nicht vor Hack-Angriffen gefeit ist. Es steht außer Frage, dass Sicherheitsmechanismen eingesetzt werden, um diese Eventualitäten zu verhindern. Da sich auch (potenzielle) Kunden um die Sicherheit Sorgen machen, ist es auch hier wichtig, dies zu kommunizieren (vgl. Zimmerman 2014: 42-43).

6.7 Standards

Standards sind Regeln, Leitlinien oder Merkmale, die für Produkte und Prozesse erstellt werden. In der Sprachtechnologie sind Standards für die Kompatibilität der verschiedenen Anwendungen wichtig. Da die einzelnen Tools oft proprietäre Dateiformate haben, wäre es für mehrere Übersetzer/innen eines Teams schwer möglich, zusammenzuarbeiten und ihre Datenbanken auszutauschen, wenn sie verschiedene Programme verwenden. Obwohl es auf der Hand liegt, dass alle Beteiligten profitieren würden, wenn alle Programme mit denselben Formaten arbeiteten, werden die normierten Formate nicht überall eingesetzt und auch ihre Unterstützung funktioniert nicht immer reibungslos (vgl. DePalma 2012: 40).

Die LISA hat fünf Standards erarbeitet: „TermBase eXchange“ (TBX), „Translation Memory eXchange“ (TMX), „Segmentation Rules eXchange“ (SRX), „Global information management Metrics eXchange – Volume“ (GMX-V) und „XML Text Memory“ (xml:tm).

TBX, TMX, SRX und xml:tm sind XML-basierte Formate, die vorschreiben, wie die Information in Terminologiedatenbanken, Translation Memorys, Segmenten und im Text gespeichert wird. GMX-V legt Regeln fest, wie die Wortanzahl in Dokumenten ermittelt wird. Mit diesen Standards können Übersetzer/innen, Terminolog/innen und andere Beteiligte die Früchte ihrer Arbeit austauschen, da die üblichen CAT-Tools diese Formate exportieren und importieren können. Da die proprietären Formate, falls vorhanden, Informationen anders speichern, kann es aber zu Problemen beim Export beziehungsweise Import kommen. Da die LISA seit 2011 nicht mehr existiert, wurden die fünf Formate zu offenen Standards gemacht und vom Europäischen Institut für Telekommunikationsnormen übernommen. Dieses sorgt für die Weiterentwicklung der Standards, sodass diese den Anforderungen in ihrem sich ständig wandelnden Umfeld genügen (vgl. Guillemin & Trillaud 2012: 38-41).

Ein weiterer bedeutender Standard ist „XML Localisation Interchange File Format“ (XLIFF), der von der „Organisation for the Advancement of Structured Information Standards“ (OASIS) entwickelt wurde. XLIFF ist ein Format, das Ausgangs-, Zieltext und weitere Informationen enthält, um so ganze Projekte weitergeben zu können. XLIFF ist, so wie die oben genannten Standards, Teil der Open-Standards-Initiative OAXAL, zu der auch Unicode gehört (vgl. Savourel 2014: 42-43; Guillemin & Trillaud 2012: 38-41).

7 Mobile Apps

Apps sind ein relativ neues Phänomen, das jedoch innerhalb weniger Jahre Eingang in die Gesellschaft gefunden hat. Hätte von zehn Jahren niemand gewusst, worum es sich handelt, ist der Begriff mittlerweile sogar im Duden zu finden (vgl. Duden: App 2014). Die Idee, diese Arbeit dem Thema Apps zu widmen, entstand aus der Allgegenwärtigkeit dieser Programme gepaart mit dem Gedanken: Wenn mein Smartphone meine Sprache spricht muss irgendwo jemand sitzen, der übersetzt. Da sich mobile Apps und ihr Medium, das Smartphone, von Desktopsoftware und dem PC unterscheiden, stellen sich folgende Fragen: Wie können Apps lokalisiert werden? Lassen sich die besprochenen Konzepte der Softwarelokalisierung auch auf diesen Bereich anwenden?

Um diesen Fragen auf den Grund zu gehen, werden zunächst die Medien, Smartphones und Tablet-PCs, vorgestellt, danach auf die Software eingegangen und schließlich die Lokalisierung von mobilen Apps besprochen.

7.1 Smartphone und Tablet

7.1.1 Entstehung

Das „kluge Telefon“ ist aus dem heutigen Alltag nicht mehr wegzudenken. Smartphones haben sich in der westlichen Kultur etabliert. Assoziiert man vielleicht vorerst Teenager damit, die selten den Blick von den kleinen Bildschirmen abwenden, sind sie in Wirklichkeit nicht nur in den privaten, sondern auch den Business-Alltag eingedrungen. Grundsätzlich sind Smartphones wohl eher als Computer denn als Telefone einzuordnen, denn die Rechnerleistungen haben Höhen erreicht, die vor nicht allzu langer Zeit noch bei Desktop-PCs Standard waren. Telefonieren ist zu nur einer von vielen Funktionen dieser Geräte geworden. Definieren lässt sich das Smartphone als:

„Mobiltelefon mit erweitertem Funktionsumfang. Dazu zählen neben der Telefonie und Short Message Service (SMS) üblicherweise Zusatzdienste wie Electronic Mail (E-Mail), World Wide Web (WWW), Terminkalender, Navigation sowie Aufnahme und Wiedergabe audiovisueller Inhalte. Auf Smartphones laufen gegenüber herkömmlichen Mobiltelefonen komplexere Betriebssysteme wie etwa Symbian OS, Blackberry OS oder das iPhone OS. Die hierdurch geschaffene Möglichkeit zur Installation weiterer Applikationen durch den Endnutzer verleiht Smartphones einen erweiterbaren und individualisierbaren Funktionsumfang.“ (Springer Gabler 2014)

Ein weiteres Kriterium ist die Bedienung über Touchscreen (vgl. Duden: Smartphone 2014). Dadurch ergibt sich eine gänzlich andere Bedienung als bei der Arbeit mit einem PC.

Die Idee des multifunktionalen Gadgets ist jedoch keineswegs neu. Die Anfänge des tragbaren Taschencomputers reichen bis in die 80erjahre zurück, diese PDAs fanden ab Anfang der Neunziger Verbreitung. Das erste Gerät, das diese Funktionen mit dem Telefonieren kombinierte, war der „IBM Simon Personal Communicator“, der 1994 auf den Markt kam, jedoch insgesamt nur geringe Erfolge verbuchen konnte und ab 1995 nicht mehr verkauft wurde. Die Hauptgründe für den Misserfolg waren die mangelnde Infrastruktur und die schlechte Akkuleistung. Sonst verfügte der Simon über ähnliche Features wie moderne Smartphones: er wurde per Touchscreen bedient und hatte mehrere Funktionen, die man mit den heutigen Apps vergleichen kann. Die folgenden Geräte, die eher in der Geschäftswelt Verbreitung fanden, waren der „Nokia Communicator“ und die „BlackBerry“-Geräte (vgl. Sager 2012).

Es gab noch viele weitere Handheld-Geräte, die entwickelt wurden, mit unterschiedlichsten Funktionen. Die Ära des Smartphones begann jedoch mit dem „iPhone“, das am 9. Januar 2007 erstmals vorgestellt wurde (vgl. Murtazin 2010). Der Unterschied zu anderen Handhelds war die Bedienung über Touchscreen. Interessant ist, dass Google zur dieser Zeit auch an der Entwicklung eines Smartphones arbeitete, das viele Features hatte, die beim ersten iPhone noch nicht verfügbar waren, wie das Herunterladen von Apps. Jedoch wurde es über eine Hardware-Tastatur bedient und deshalb wurde die Entwicklung abgebrochen und neu begonnen. Das Ergebnis war schließlich das Betriebssystem „Android“ (vgl. Vogelstein 2013). In den folgenden Jahren wurden immer mehr Smartphones auf den Markt gebracht, von den verschiedensten Mobiltelefonherstellern mit vorab noch unterschiedlichen Betriebssystemen.

Tablets oder Tablet-PCs sind die zweite Geräteform, die für mobile Apps relevant ist. Es handelt sich tatsächlich um PCs, jedoch haben sie äußerlich mehr mit Smartphones gemein, da sie nicht über Maus und Tastatur verfügen, sondern auch über einen Touchscreen bedient werden. So könnte man sie als größere Smartphones ohne Telefonfunktion beschreiben. Dieses Konzept ist eigentlich noch älter als das Smartphone, da schon länger verschiedene tragbare Computer, auch mit Touchscreen, erhältlich sind. Auch hier spielte Apple eine große Rolle, da durch die Einführung des iPads 2010 diese PC-Form einen Aufschwung erfuhr (vgl. ITWissen: Tablet-PC). Tablets sollen in Folge nicht weiter eigens genannt werden, da es bei den Plattformen und Apps keine nennenswerten Unterschiede zu Smartphones gibt.

7.1.2 iOS und Android

Wie jeder Computer braucht ein Smartphone ein Betriebssystem. Unter den relevanten Plattformen gibt es proprietäre Systeme, die nur auf den Geräten eines Herstellers verwendet werden, sowie herstellerübergreifende Systeme. Die Plattformen, die insgesamt weitere Verbreitung fanden, sind iOS, Android, Windows Phone, BlackBerry, Symbian und Bada. Bada war ein Betriebssystem für Samsung-Geräte. Da diese mittlerweile alle Android verwenden, wur-

de die Entwicklung eingestellt. Dasselbe gilt für Symbian, das auf Nokia-Geräten installiert war. Dieses Betriebssystem hatte zeitweise durch die Verbreitung der Nokia-Geräte einen hohen Marktanteil, verlor aber schließlich gegen die Konkurrenz, genauso wie die Mobiltelefonsparte des Unternehmens selbst, die 2014 verkauft wurde. BlackBerry 10, Nachfolger von BlackBerry OS ist das proprietäre System, das auf den Produkten der Marke BlackBerry installiert ist. iOS ist schließlich die erfolgreichste unter den hauseigenen Plattformen und läuft auf den iPhones (, iPads und iPods) von Apple. Windows Phone von Microsoft und Android von Google beschränken sich nicht auf einen Hersteller, sondern werden auf einer Vielzahl von mobilen Geräten verwendet (vgl. Shoutmeloud 2014).

Die Kapitelüberschrift beschränkt sich nicht ohne Grund auf zwei dieser Plattformen. Der Marktanteil dieser beiden macht mittlerweile über 96 % (Android: 84,7 %, iOS: 11,7 %) aus (vgl. IDC 2Q14 2014). Diese Tatsache ist auch der Grund, weshalb im Weiteren die anderen Plattformen ausgeklammert werden und die Kapitel über Apps und Lokalisierung von Apps sich auf iOS und Android beschränken.

Der Marktanteil wird berechnet nach dem Prozentsatz an Geräten, die mit dem jeweiligen Betriebssystem ausgeliefert werden. Hier war Android nicht immer führend. 2009 hatte diese Rolle Symbian inne, das auf ungefähr 50 % der verkauften Mobiltelefone installiert war. Infolge der Verdrängung durch iPhone und Co sank der Marktanteil von Symbian stetig. Bada war zwischen 2010 und 2013 aktiv, konnte jedoch nie mehr als 2,7 % erreichen. BlackBerry war 2009 mit über 20 % noch gut vertreten, verlor aber auch langsam aber sicher und liegt heute bei 0,5 %. Ähnlich ist es bei Windows Phone, das 2009 auf guten 10 % lag, mittlerweile aber auf 2,5 % abgestürzt ist. iOS erreichte 2009 bereits über 10 %, schaffte es zwischenzeitlich auf 23,8 % und liegt momentan bei 11,7 %. An den geringen Prozentsätzen der bisher genannten Betriebssysteme lässt sich der Höhenflug von Android erahnen: Nach Markteinführung im Jahr 2008 kletterte Android im Jahr 2009 auf 3,5 %. Die 50 %-Grenze wurde 2011 geknackt, im 2. Quartal 2014 wurden schließlich 84,7 % erreicht, was Android scheinbar zum wichtigsten Player auf dem Markt macht (vgl. Statista Smartphone OS 2014; IDC 2Q14 2014).

Weltweit werden also über 80 % der mobilen Geräte mit Android verkauft. Google dominiert den Markt aber nur scheinbar, da das Betriebssystem Android auf vielen Geräten verschiedener Hersteller installiert ist. So waren es 2012 fast 4000 verschiedene Modelle (vgl. Floemer 2012). Am meisten Profit wird mit den verkauften Smartphones und Tablets erwirtschaftet und hier liegt Apple weit vorn. Der Hersteller, der am meisten Android-Phones verkauft, ist Samsung, gefolgt von Coolpad, Huawei, Lenovo, LG, Xiaomi, und ZTE. Auf diese Weise teilen sich eine Reihe von Herstellern die Gewinne aus den Smartphone-Verkäufen. Android liegt logischerweise auch beim App-Download weit vorn, doch auch hier verdient Apple mehr, da es im Apple App Store mehr Apps gibt, die verkauft werden (vgl. Bradley 2013^b). Im Play Store sind hingegen nur gute 19 % kostenpflichtige Apps, von deren

Erlös Android auch nur ein gewisser Prozentanteil zufließt (vgl. AppBrain Stats Free vs. Paid Android Apps 2014).

Hier können auch die unterschiedlichen Geschäftsstrategien erkannt werden. Apple beschränkt sich auf eigene Produkte, die alle im höheren Preissegment liegen und kontrolliert den App Store stark, indem alle Apps bevor sie angeboten werden überprüft werden. Google vertreibt Android unter einer Open-Source-Lizenz und der App Store ist für alle zugänglich, das heißt, wenn man sich als Entwickler/in anmeldet, kann man nach Lust und Laune Apps verkaufen. Es gibt Android-Smartphones in jedem Preissegment, was auch zu der starken Verbreitung führt, jedoch bedeutet somit die hohe Anzahl an verkauften Geräten nicht, dass der Gewinn höher ist. Zudem schmerzt es Apple nicht, dass viel mehr Android-Smartphones verkauft werden, da diese Kund/innen ohnehin nicht die Zielgruppe ausmachen, das heißt, jemand, der sich um 100 \$ ein Android-Gerät kauft, würde sich nicht um das Fünf- bis Sechsfache ein iPhone leisten (können). Man könnte diese Ansätze als qualitäts- beziehungsweise quantitätsorientiert beschreiben: Apple dominiert das höhere Preissegment während Android den Massenmarkt bedient (vgl. Bradley 2013^a). Die unterschiedlichen Strategien spiegeln sich auch in der Organisation der App-Entwicklung wider.

7.2 Mobile Apps

7.2.1 Grundlegendes

App ist die Abkürzung von „application“, also Anwendung, ein Computerprogramm. Im deutschen Sprachraum wird darunter ein kleines Programm verstanden, das auf einem mobilen Endgerät benutzt werden kann. Applikationen an sich gibt es natürlich schon lange, mobile Apps stellen jedoch trotzdem eine spezielle Klasse dar, da sie an das Medium, das mobile Gerät, angepasst sind. Das heißt, dass die grafische Benutzeroberfläche wegen des kleinen Bildschirms und wegen der Art der Eingabe über Touchscreen und Bewegungssensoren sich grundlegend von der PC-Oberfläche unterscheidet. Ein weiteres Kriterium ist der Speicherplatz: PC-Festplatten verfügen heutzutage üblicherweise über mehrere hundert Gigabyte, die teureren Smartphones über höchstens 64 GB. Wegen der mobilen Nutzung der Smartphones haben die Programme auch andere Funktionen als der Großteil der Desktop-Software. Es hätte keinen Sinn, komplexe Buchhaltungssoftware auf einem Smartphone zu installieren, da man mit diesem ja nicht arbeitet. So beschränken sich Apps auf einzelne Funktionen, im Endeffekt gibt es also für alles eine App und nicht ein einzelnes umfassendes Programm, das alles kann. Dieser Umstand führt dazu, dass jede/r Nutzer/in sein/ihr Smartphone sehr stark personalisieren kann. Durch das einfache Herunterladen von Apps, das in vielen Fällen gratis ist, kann man auch gerade dann eine Funktion nutzen, wenn man sie braucht. Es ist nicht nö-

tig, sich die Investition in ein Softwarepaket gut zu überlegen, will man eine bestimmte Funktion auf seinem Smartphone nutzen, lädt man sich kurzerhand eine App herunter. Dies führt auch zu einer gewissen Kurzlebigkeit vieler Apps und einer Veränderung des Technologieverständnisses zu einer Sicht des „instant need fulfillment“ (vgl. Mayer 2012: 9-13).

Die immerwährende Verfügbarkeit ist ein grundlegender Aspekt. Über eine vorinstallierte App wird der App Store geöffnet (gleichnamig bei Apple, „Play Store“ bei Google). Dieses Programm bietet die verschiedensten Apps zum Download an, mittlerweile gibt es auch E-Books und Musik. Neben der Suchfunktion werden die beliebtesten Apps und die Kategorien angezeigt. Nachdem eine App ausgewählt wird, gelangt man zur Produktseite und kann mittels „Download“-Feld die App herunterladen, die automatisch installiert wird. Zuvor muss (im Play Store) noch der Zugriff auf verschiedene Hardware- und Softwarekomponenten bestätigt werden, so zum Beispiel braucht eine Foto-App Zugriff auf die Kamera, eine Nachrichten-App Zugriff auf das Telefonbuch. Handelt es sich um eine kostenpflichtige App, so muss sie natürlich zuvor bezahlt werden, es gibt dafür verschiedene Möglichkeiten (Kreditkarte, Abrechnung über den Mobilfunkanbieter, Prepaid-Karten oder PayPal). Apple App Store bietet Apps in 155 Ländern an, während Google Play Store in 139 Ländern Apps zum Download anbietet. Kostenpflichtige Apps sind nicht überall verfügbar, beim Play Store sind von den 139 Ländern jedoch nur 4 ausgenommen (vgl. Google Play Store 2014; Google Support: Zahlungsmethoden; Google Support: Standorte 2014; App Store FAQ 2014; Apple Developer: TestFlight 2014).

Welche Funktionen bieten nun Apps? Die Kategorien des Google Play Store geben hierüber Aufschluss (die Kategorien des App Stores unterscheiden sich nicht nennenswert): Grundsätzlich wird zwischen Spielen und anderen Apps unterschieden, was die Bedeutung von Unterhaltung auf dem Smartphone unterstreicht. Für Spiele gibt es eine Reihe von eigenen Kategorien, wie „Action“, „Abenteuer“, „Arcade“, „Strategie“, „Lernspiele“, „Brettspiele“, „Quizzspiele“, „Buchstabenspiele“ und viele mehr. Es ist also alles vertreten, was auch sonst an Spielesoftware erhältlich ist.

Der zweite Typ, also Apps, unterteilt sich in folgende Kategorien:

„Büro“, „Comics“, „Kommunikation“, „Lernen“, „Unterhaltung“, „Finanzen“, „Gesundheit & Fitness“, „Software & Demos“, „Lifestyle“, „Live-Hintergründe“, „Medien & Videos“, „Medizin“, „Musik & Audio“, „Nachrichten & Magazine“, „Personalisierung“, „Fotografie“, „Effizienz“, „Shopping“, „Soziale Netzwerke“, „Sport“, „Tools“, „Verkehr“, „Reisen & Lokales“ und „Wetter“. (Google Support: Kategorien 2014)

Es ist ersichtlich, dass es für alle Lebenslagen Apps gibt, die das Hardware-Potenzial des Endgerätes ausschöpfen. Die Anzahl der Apps, über 1,3 Milliarden allein im Play Store (vgl. AppBrain Stats Google Play Store 2014), deutet allein schon auf die große Vielfalt hin.

Mayer (2012: 14-15) nennt drei Gründe, aus welchen Apps heruntergeladen werden:

- „- Weil sich der Nutzer einen inhaltlichen Mehrwert verspricht, zum Beispiel schnelle Information oder direkten Zugriff auf bestimmte Services.
- Weil der Nutzer Unterhaltung und Abwechslung will, zum Beispiel durch Mobile Games.
- Weil der Nutzer ein Statement abgeben will, indem er zum Beispiel Apps seiner Lieblingsmarke lädt.“

Das heißt, die meisten der oben genannten Kategorien fallen in den ersten Bereich: Büro-, Kommunikations-, und Medizin-Apps, zum Beispiel, bieten den raschen Zugang zu Information oder die Vernetzung mit Freund/innen oder Geschäftspartner/innen. Dass es sich beim Smartphone um ein Kommunikationsinstrument handelt, lässt sich auch an den Download-Statistiken dieser Art von Apps erkennen: In Österreich wird die Liste der frei verfügbaren Apps schon seit Längerem vom „Facebook-Messenger“, „WhatsApp“ und der „Facebook-App“ angeführt. Spiele stehen natürlich auf der Beliebtheitsskala auch ganz oben. Viele User/innen sind sogar bereit, dafür zu bezahlen (vgl. App Annie Google Play Top App Charts 2014). Im Prinzip sind der Fantasie (fast) keine Grenzen gesetzt. Ein nennenswertes Beispiel ist eine App, die von einer libanesischen Studentin programmiert wurde: In ihrem Heimatland kommt es immer wieder zu Selbstmordattentaten und Bombenanschlägen. Wenn dies geschieht, wollen viele Menschen ihre Angehörigen informieren, dass ihnen nichts zugestoßen ist, weswegen dann das Telefon- und Internetnetz überstrapaziert wird. Die App schickt auf Knopfdruck einen Tweet mit dem Text „I am still alive“ aus und informiert so rasch alle Angehörigen und Freund/innen (vgl. Binder 2014). Ob diese App tatsächlich ernst gemeint ist oder eher ein Seitenhieb auf die Situation in Libanon sei dahingestellt. Es zeigt jedoch, wie Apps bereits als eigenes Medium wahrgenommen werden und was sich damit machen lässt.

Apps sind an sich natürlich nicht von Kritik ausgenommen. Viele Apps stellen während der Verwendung automatisch eine Verbindung zu einem Server her. Dies kann meistens nicht abgestellt werden, da vor der Installation bereits das Zugriffsrecht gewährt wurde. Dieser Umstand führt dazu, dass so Sicherheitslücken entstehen und Daten abgezapft werden können (vgl. Ball 2014).

Zudem gibt es Apps, deren Funktion und Qualität infrage gestellt wird. So gibt es eine große Bandbreite an Gesundheits-Apps, mit deren Hilfe es leichter sein soll, gesund zu leben. Sich als Diabetiker/in jedoch auf eine App zu verlassen, wenn es um die Kontrolle des Blutzuckerspiegels geht, kann unter Umständen lebensgefährlich werden. Das Konzept kann natürlich funktionieren, doch ist es wegen der großen Anzahl an angebotenen Apps schwierig, hochwertige Apps von jenen minderwertiger Qualität zu unterscheiden (vgl. Bayer 2014).

7.2.2 Wirtschaftliche Relevanz

Im 2. Quartal 2014 wurden weltweit über 300 Millionen Smartphones verkauft (vgl. IDC 2Q14 2014). Dass Smartphones wirtschaftlich äußerst relevant sind, steht somit außer Frage. Aus dieser Bedeutung lässt sich auch die Bedeutung von Apps ableiten: Der Wert eines Smartphones für den/die Einzelne/n erwächst aus den Apps, die erst wirklich das volle Potenzial des Geräts ausschöpfen. Ohne die Möglichkeit, unterschiedlichste Apps herunterzuladen, ist das Smartphone nicht viel mehr als ein Handy mit Webbrowser. Auf diese Weise ergibt sich eine Symbiose: Smartphones eröffnen die Möglichkeit, Apps zu nutzen, Apps machen das Smartphone erst zu dem, was es ist.

Wirtschaftlich gesehen rentiert sich dies natürlich. Obwohl der Großteil der App-Downloads gratis ist (91 % im Jahr 2013), sind es nicht nur kostenpflichtige Apps, die Geld hereinbringen. Grundsätzlich gibt es vier Möglichkeiten, wie eine App angeboten werden kann: gratis, kostenpflichtig, mit Werbung und mit zusätzlich erwerbbaaren Inhalten (vgl. Apple Developer: First Steps 2014; Google Support: Vertrieb 2014). 2011 kam noch 85,5 % der Einnahmen aus dem Verkauf kostenpflichtiger Apps, dies ist bis 2013 auf 75,9 % gesunken. Die Prognosen sagen ein weiteres Absinken voraus, auf 37,8 % im Jahr 2017. Dafür steigen die Einnahmen aus dem Verkauf von zusätzlich erwerbbaaren Inhalten massiv: 2011 lag dies noch bei 8,6 %, 2013 bereits bei 17,2 % und bis 2017 wird ein Anstieg auf knapp die Hälfte, 48,2 %, prognostiziert. Die Einnahmen aus den sogenannten In-App-Ads steigen hingegen nur moderat: zwischen 2011 und 2013 von 5,6 % auf 6,9 % mit einem prognostizierten Anteil von 14 % 2017.

Prozentangaben sind natürlich interessant, ohne absolute Zahlen ist jedoch nicht ersichtlich, wie viel ein „Stück vom Kuchen“ wert ist. Weltweit wurden mit den Einnahmen aus den verschiedenen App Stores im Jahr 2011 8,381 Milliarden US-Dollar erwirtschaftet. Der weitere Anstieg ist dramatisch: 2012 waren es 18,559 Milliarden Dollar, 2013 bereits 26,683 Milliarden Dollar. Die weiteren Prognosen bis 2017 sagen jährliche Anstiege zwischen zirka 8 und 13 Milliarden Dollar voraus.

Interessant ist auch noch eine andere Verteilung: Diese imposanten Zahlen werden nur durch einen kleineren Prozentsatz aller App-Downloads erwirtschaftet. Die Gesamtanzahl der Downloads lag 2012 bei 63,985 Milliarden, 2013 bei 102,062 Milliarden und wird voraussichtlich bis 2017 auf 268,692 Milliarden steigen. Davon sind jedoch jeweils nur zwischen 5,5 und 10,4 % bezahlte Downloads. Dies reicht jedoch anscheinend, um den Entwickler/innen und Plattformbetreibern reiche Einkünfte zu beschern (vgl. Lunden 2013).

Hierbei muss natürlich beachtet werden, dass es eine große Bandbreite an App-Entwickler/innen beziehungsweise Unternehmen gibt. Manche Entwickler/innen schreiben Apps als Hobby und freuen sich über kleine Erträge. Manche stellen auch Apps zum gratis-Download zur Verfügung, weil sie einfach an der Arbeit Spaß haben und viele Apps auch gar nicht geeignet sind, um verkauft zu werden. Manche Unternehmen bringen auch Apps als

Erweiterung ihrer Dienste in Umlauf, so zum Beispiel Banking-Apps, die von bestehenden Kunden genutzt werden können. Auf der anderen Seite gibt es natürlich jene Unternehmen, die Spiele oder sonstige, meistens komplexere, Apps produzieren, die ihre Haupteinnahmequelle darstellen. Deshalb macht es auch nur bei manchen, vor allem der letzteren Gruppe, Sinn, den (finanziellen) ROI zu berechnen (vgl. Maxwell 2011; Slideshare: ROI 2014).

7.2.3 Vertriebsmodell

Wie bringt man Apps an die Frau beziehungsweise den Mann? Anders als bei PC-Software werden diese Programme ja nicht auf Datenträgern verkauft. In Kapitel 7.3.1 wurde bereits die Verwendung der App Stores durch die User/innen beschrieben. Im Folgenden soll erklärt werden, wie Entwickler/innen vorgehen.

Grundsätzlich funktioniert dies folgendermaßen: Als Entwickler/in braucht man zuerst ein Entwicklerkonto. Dies kostet bei Android einmalig 25 \$, bei Apple für einzelne Entwickler/innen 99 \$ pro Jahr, für Unternehmen 299 \$. Das Entwicklerkonto ist bei Apple auch für den Download der Entwicklungsumgebung notwendig, ferner kann man iOS-Apps nur auf einem Mac programmieren. Bei Android kann die Entwicklungsumgebung von allen heruntergeladen werden, was daran liegt, dass Android und die zugrundeliegende Programmiersprache Java unter einer Open-Source-Lizenz laufen. Nach dem Programmieren, Testen und gegebenenfalls Lokalisieren der App ist der nächste Schritt, sie in den App Store hochzuladen. Dies funktioniert bei Android über die Developer-Konsole des Entwicklerkontos. Die App ist als APK-Datei gespeichert und darf maximal 50 Megabyte groß sein, jedoch gibt es die Möglichkeit, zwei Erweiterungsdateien von jeweils 2 Gigabyte hochzuladen. In der Developer-Konsole wird ein Store-Eintrag, die Produktseite der App, erstellt, das heißt man gibt verschiedene Informationen an (Sprache, Titel, eine Kurzbeschreibung von maximal 80 Zeichen, eine vollständige Beschreibung von maximal 4000 Zeichen, gegebenenfalls neue Features, den Typ: Spiel oder App und die Kategorie). Es ist noch anzugeben, an welchen Standorten die App angeboten wird und welche Support-Kanäle verfügbar sein werden (Website, E-Mailadresse o.ä.). Zudem können Bilder und Videos hochgeladen beziehungsweise verlinkt werden. Die Store-Einträge können auch Übersetzungen in verschiedene Sprachen enthalten, das heißt, für lokalisierte Versionen ist kein zusätzlicher Store-Eintrag nötig. Wenn nicht anders möglich, kann die Langbeschreibung maschinell übersetzt werden. Handelt es sich um eine App, die verkauft wird, Werbung oder zusätzliche Inhalte enthält, müssen auch die Zahlungsdetails geklärt werden. Die Einkünfte werden über Google Wallet, ein Bestandteil des Entwicklerkontos, abgerechnet. Die Preise für Apps variieren nach Land, in Österreich und Deutschland kann zwischen 0,5 und 199 € verlangt werden. Von den Einkünften müssen durchschnittlich 30 % an die App Store-Betreiber abgeführt werden (vgl. Anthes 2011: 16;

Google Support: Apps hochladen 2014; Google Support: Standorte 2014; Google Support: Vertrieb 2014).

Bei Apple funktioniert dies im Prinzip ähnlich, nur wird das App-Bundle auf iTunes Connect hochgeladen und dann erst auch im App Store verfügbar gemacht. Der bedeutende Unterschied ist, dass Apps vor Veröffentlichung von Apple überprüft werden. Dadurch behält das Unternehmen die Kontrolle und kann auch Apps aussieben, die den Qualitätsstandards nicht genügen (vgl. Apple Developer: About App Distribution 2014). Obwohl dieser Ansatz oft kritisiert wurde, da vermutet wurde, dass manche Apps aus Willkür nicht aufgenommen wurden, ist dies durchaus sinnvoll. Beim Play Store, der keiner Kontrolle unterliegt, werden zumindest 17 % als Apps niedriger Qualität eingestuft (vgl. Heise Online 2008; AppBrain Stats Play Store 2014).

7.2.4 App-Programmierung und Internationalisierung

Wie andere Programme auch, laufen Apps auf Betriebssystemen und nicht direkt auf dem Computer. Die in dieser Arbeit beschriebenen Plattformen sind, wie oben bereits genannt, iOS und Android. Im Gegensatz zu PC-Betriebssystemen geschieht die Weiterentwicklung hier sehr rasch. Die aktuelle Version von iOS ist 7.1, das heißt, dass durchschnittlich jedes Jahr eine neue Version veröffentlicht wird. Bei Android ist der Zyklus noch schneller, die aktuelle Version heißt 4.4.4, ist aber schon die 19. Ausgabe des API. Zudem wurde bereits die zukünftige Version, namens „L“, vorgestellt, die im Herbst 2014 erscheinen soll. Für Entwickler/innen gibt es die Entwicklungsumgebung bereits zum Download (vgl. Amadeo 2014; Costello 2014; Android Developer: L Preview 2014). Nicht jedes Update erfindet das Rad neu, es werden aber nach und nach immer mehr Features eingebaut, die teilweise auch für Lokalisierung relevant sind. All dies unterstreicht die Schnelllebigkeit und den raschen Fortschritt im Bereich Smartphones und Apps.

Grundsätzlich ist beim Programmieren von Apps eine größere Abhängigkeit und Vernetzung mit der Plattform zu beobachten also bei anderer Software, zumindest wenn Entwickler/innen dies wünschen. Für gewöhnlich werden Android-Apps in der Programmiersprache Java und iPhone-Apps in Objective-C geschrieben. Für die Entwicklung von Apps kann von den Entwickler-Sites eine integrierte Entwicklungsumgebung (IDE) heruntergeladen werden, bei Android ist dies Eclipse, bei Apple Xcode. Es handelt sich um ein Paket aus Werkzeugen und Anwendungen, mit denen sich Software programmieren lässt. Die Dokumentation ist in Form von Anleitungen und Tutorials auf der jeweiligen Developer-Site frei verfügbar (vgl. Apple Developer: Tutorial Basics 2014; Android Developer: SDK 2014; Android Developer: Training 2014). Es gibt natürlich auch andere Entwicklungsumgebungen und Möglichkeiten, Apps zu entwickeln. Bedenkt man die Unmenge an verfügbaren Apps, ist

klar, dass viele Entwickler/innen nicht nach den beschriebenen Schritten vorgehen und diese Tools für die Programmierung benutzen.

Mit der integrierten Entwicklungsumgebung ist es im Prinzip einfach, eine App zu programmieren und zu internationalisieren. Internationalisierung sollte, wie an früherer Stelle beschrieben, bereits während der Entwicklung geschehen. Wird Eclipse oder Xcode verwendet, wird die Ordner- und Dateistruktur von diesem Programm erstellt. Strings und andere Ressourcen werden von vornherein in eigene Ordner gespeichert oder können extrahiert werden. Es ist möglich, mehrere Ressourcen zu erstellen. Dazu müssen die Ressourcendateien, die lokalisierungsrelevante Elemente enthalten, lediglich kopiert werden. Die Kopien müssen mit dem Sprachcode nach ISO 639-1 und/oder dem Regionscode nach ISO 3166-1-alpha2 gekennzeichnet werden. Hier sollte beachtet werden, dass eines der Ressourcenpakete kein Kürzel enthält und somit die Default-Ressource darstellt. Beim Verwenden der App am Endgerät lädt das Programm automatisch die Ressourcen, die mit den Sprach-/Locale-Einstellungen übereinstimmen. Ist jedoch für die Systemsprache kein Ressourcenpaket vorhanden, wird automatisch auf das Default-Ressourcenpaket zurückgegriffen. Um Speicherplatz zu sparen, kann auch in den lokalisierten Ressourcenpaketen auf Bestandteile verzichtet werden, die nicht lokalisiert werden müssen (zum Beispiel Strings, die nur den Namen der App enthalten). Auch in diesem Fall springt sozusagen die Default-Ressource ein. Dies ist sehr sinnvoll, da der interne Speicher der meisten Smartphones nicht sehr groß ist und meistens eine App mit allen lokalisierten Ressourcen angeboten wird. In der Android-Umgebung gibt es, zusätzlich zu den Sprachen- und Ländercodes, noch andere Möglichkeiten, Ressourcen zu differenzieren: Der „mobile country code“, MMC, steht für das mobile Netzwerk eines Landes, der „mobile network code“, MNC, für ein bestimmtes Netzwerk eines Landes. Diese Codes sind für Lokalisierung dann relevant, wenn zum Beispiel aus rechtlichen Gründen bei einem Anbieter etwas anderes angezeigt werden soll als bei den anderen. Diese beiden Codes sind ranghöher als Sprachen- und Regionscode. Weitere Ressourcenindikatoren gibt es für geräteabhängige Faktoren, wenn zum Beispiel für verschiedene Auflösungen unterschiedliche Layout-Informationen und Grafiken inkludiert werden (vgl. Android Developer: Supporting Different Languages 2014; Android Developer: Resources Overview 2014; Android Developer: Localizing with Resources 2014; Apple Developer: Guidelines for Internationalization 2014; Apple Developer: Internationalization and Localization 2014).

Weiters decken sich die Empfehlungen auf den beiden Developer-Sites mit den Erläuterungen der bisherigen Kapitel: Strings sollten nicht hartkodiert werden, Zahlen, Daten, Zeiten und Währungen sollten auch nicht hartkodiert werden sondern Codes verwendet werden um automatisch die richtige Darstellung zu garantieren und die Unterstützung von Unicode und bidirektionalen Schriften programmiert werden. Letzteres ist bei den neueren Android-Versionen bereits integriert. Ein zusätzlicher Punkt ist das flexible Design des Layouts: Mehr Platz für längere lokalisierte Strings zu lassen wird zwar sonst auch empfohlen, ist hier jedoch aufgrund der begrenzten Displaygröße besonders wichtig. Da, wie oben beschrieben,

verschiedene Ressourcenpakete mitgeliefert werden können, kann auch für jede lokalisierte Version ein eigenes Layout erstellt werden, das dann auch automatisch geladen wird (vgl. Android Developer: Localization Checklist 2014; Apple Developer: Guidelines for Internationalization 2014). Ein interessantes Feature, das seit Version 4.3 im Betriebssystem Android eingebaut ist, ist „Accented English“. Es handelt sich dabei im Prinzip um ein Locale, das für Pseudo-Lokalisierung verwendet werden kann. Wird ein (ausgangssprachliches) Ressourcenpaket mit „zz_ZZ“ als Sprachen- und Regionscode gekennzeichnet, werden alle Zeichen in den Strings automatisch durch Sonderzeichen ersetzt und somit kann überprüft werden, ob Strings nicht externalisiert wurden und ob die Zeichenunterstützung funktioniert. Eine weitere Funktion erzwingt ein rechts-nach-links-Layout. Somit kann überprüft werden, ob das Layout korrekt funktionierte, wenn es für Schriften, die in diese Richtung gehen, umgedreht würde (vgl. Android Developer: Jelly Bean 2014).

Apps müssen, so wie andere Programme, natürlich auch getestet werden. Hierfür gibt es bei den integrierten Entwicklungsumgebungen Emulatoren. Trotzdem sollten sie auch auf Endgeräten getestet werden, um alle Eventualitäten auszuräumen. Dies gestaltet sich natürlich bei Android schwieriger, da es eine sehr große Anzahl an Modellen gibt, die das Betriebssystem verwenden. Deshalb gibt es, so wie auch bei Apple, die Möglichkeit, über den App Store eine neue App von User/innen testen zu lassen (vgl. Android Developer: Localization Checklist 2014; Apple Developer: TestFlight 2014).

7.3 Lokalisierung von Apps

7.3.1 Warum Apps lokalisieren?

Die Vorteile von Lokalisierung wurden in Kapitel 4 umfassend besprochen. Es ist für Unternehmen, die kommerzielle Software herstellen, nicht mehr aktuell, auf dem globalen Markt nicht-lokalisierte Produkte zu verkaufen. Apps sind hier natürlich keine Ausnahme. Bereits 2011 sorgte der US-amerikanische iPhone-Markt nur mehr für 34 % der kostenpflichtigen Downloads (vgl. App Annie Localization 2014). Auch van Genabith (2009: 7) schreibt bereits 2009, dass sich Handheld-Geräte in nicht-Industrieländern sehr stark ausbreiten. Deshalb ist es nur logisch, kommerzielle Apps zu lokalisieren, um einen größeren Markt zu erreichen. Immerhin können Apps über den Apple App Store in 155, über den Play Store in 135 Länder verkauft werden (vgl. App Store FAQ 2014; Google Support: Standorte 2014). Dies spricht im Grunde sehr stark für Lokalisierung: Anders als bei Softwarepaketen, die auf Datenträgern vertrieben werden, fallen hier keine Kosten für deren Herstellung, Verpackung und Vertrieb an.

Es sind natürlich nicht alle Apps für Lokalisierung geeignet, vor allem nicht in viele Sprachen. Es macht wenig Sinn, die Fahrplan-App der ÖBB oder die App der österreichischen Volksbank in vielen verschiedenen Sprachen anzubieten. Ob es sich lohnt, jene 17 % der Play Store-Apps zu lokalisieren, die als minderwertig eingestuft werden, ist auch fraglich.

7.3.2 Lokalisierung für Übersetzer/innen

Wenn eine App erfolgreich internationalisiert wurde, sollte die Lokalisierung kein Problem darstellen. Die übersetzungsrelevanten Dateien werden an Übersetzungs- oder Lokalisierungsdienstleister geschickt, welche diese bearbeiten und zurückschicken. Andere Ressourcen, wie Icons, Grafiken oder Audiodateien, die Sprache enthalten, müssen natürlich auch bearbeitet werden. Die lokalisierten Ressourcendateien werden wieder, zum Beispiel mithilfe der Entwicklungsumgebung, in die App importiert.

Im Idealfall bekommen Übersetzer/innen ein Translation-Kit. Dieses enthält neben den zu übersetzenden Dateien Informationen über die App, frühere Übersetzungen und Glossare, also Referenzmaterial, das für die Übersetzungsarbeit wichtig ist. Werden Apps mit dem Eclipse-IDE programmiert, sind die Strings im XML-Format gespeichert. Diese Dateien können mit einem Editor oder einem CAT-Tool geöffnet und übersetzt werden. Es ist hier, wie bei allen Arten von Strings, auf Codes, Variable und verknüpfte Strings zu achten. Anführungszeichen werden auch hier für die Kennzeichnung von Strings verwendet, müssen also im String vermieden oder doppelt geschrieben werden. Apostrophe müssen nach einem „\“ -Zeichen stehen, um angezeigt zu werden. Formatierungstags für fette, kursive und unterstrichene Zeichen dürfen nicht gelöscht werden. Für Mehrzahlformen bei Variablen gibt es eigene Codierungen mit Klassen für die Anforderungen verschiedener Sprachen. Im Deutschen, zum Beispiel, gibt es hier nur zwei notwendige Formen: Singular und Plural. Manche Sprachen haben jedoch eigene Formen für andere Quantitäten, wie null, zwei, wenige oder viele. Diese müssen gegebenenfalls im zielsprachlichen Dokument ergänzt werden (vgl. Android Developer: String Resources 2014). Es gibt auch ein Tag (xliff:g), das Stringteile umschließt, die nicht übersetzt werden sollen. Zudem gibt es auch hier Kommentare, die Informationen für Übersetzer/innen enthalten (vgl. Android Developer: Localization Checklist 2014).

Bei iOS-Apps sind Strings entweder in .string-Dateien externalisiert oder befinden sich in .nib-Dateien. .string-Dateien können mit Xcode geöffnet werden, oder aber als UTF-8 im Apple-Text-Editor. Auch hier gibt es vergleichbare Tags, Codes und Kommentare. .nib ist das Format des Apple-Programms Interface Builder und enthalten Layout-Informationen. Diese Dateien können mit dem Interface Builder geöffnet und bearbeitet werden (vgl. Apple Developer: Localizing String Resources 2014).

Sonstige übersetzungsrelevante Texte sind die Angaben für die App Stores, Marketingmaterial und gegebenenfalls eine Hilfedatei (vgl. Android Developer: Help 2014; Android Developer: Localization Checklist 2014; Apple Developer: Internationalization and Localization 2014). Hier ist ein Unterschied zur Lokalisierung von Desktop-Software festzustellen: Hilfe und Dokumentation machen üblicherweise das größte Übersetzungsvolumen aus. Bei Apps ist die Hilfe oft direkt integriert, oder auch im HTML-Format gespeichert. Oftmals fehlt sie auch und der Support wird über eine Website zur Verfügung gestellt. Die Dokumentation beschränkt sich im Prinzip auf die Beschreibungen im App Store. Diese Umstände lassen sich aus der Beschaffenheit von Apps herleiten: Da es meistens um wenige Funktionen geht, sind Apps (idealerweise) so aufgebaut, dass sie intuitiv erlernbar sind. Gibt es, wie bei vielen Apps, nur ein Eingabefenster, ist klar, dass die Buttons, so sie gut beschriftet sind, keiner weiteren Erläuterung bedürfen.

Es lässt sich eine interessante Entwicklung bei der Auftragsvergabe von Übersetzungen beobachten. Da es sich bei Apps in der Regel um kleine Programme handelt, sind die gründliche Projektanalyse, Erstellung des Kostenvoranschlags und Verhandlung der Projektdetails nicht notwendig. Der Kontakt zwischen Entwickler/innen und Übersetzungsdienstleister/innen wird durch eine Schnittstelle auf der Website des Unternehmens hergestellt. Hier können die String-Dateien und andere zu übersetzende Dateien hochgeladen werden. Die weiteren relevanten Informationen und die Zielsprache werden über eine Eingabemaske eingegeben. Manche dieser Websites enthalten sogar Tools, die automatisch Wörter zählen und sofort einen Kostenvoranschlag ausrechnen. Meistens wird eine Projektdauer von wenigen Tagen angegeben und ein Kommunikationsweg über eine Nachrichtenfunktion angeboten. Das Managementsystem ist bei diesen Anbietern in der Cloud oder auf sonstigen Servern und sorgt für die Vernetzung der Beteiligten, diese Fälle sind also Beispiele, wie diese relativ neuen Entwicklungen effizient eingesetzt werden können. Bei Android wurde dieses Service bereits in die Developer-Konsole eingebaut (vgl. Android Developer: Localization Checklist 2014; sonstige z.B.: ICanLocalize 2014; Get Localization 2014).

7.3.3 App-Lokalisierung für Entwickler/innen

Für App-Entwickler/innen, die sich die Frage stellen, wie sie ihre Produkte lokalisieren können, geben auch hier die Developer-Sites einen guten Überblick. Die Beschreibung ist hier auf der Android-Site umfassender und soll deshalb hier besprochen werden.

Der erste Schritt nach der Entscheidung, auf internationalen Märkten tätig zu sein, ist die Identifizierung der Zielsprachen und -locales. Dies soll nach verschiedenen Kriterien geschehen, wie der Marktgröße, Vermarktungsmöglichkeit, App-Kategorie, Konkurrenz, lokales Preisniveau und finanziellen Faktoren.

Die Schritte 2 und 3, Lokalisierung bei der Entwicklung im Auge zu behalten und Strings dafür vorzubereiten, ergeben sich aus dem ersten und beschreiben im Prinzip Internationalisierung.

Der vierte Schritt beschreibt Lokalisierung an sich, die Übersetzung und Bearbeitung der Ressourcen sowie deren Wiedereinfügen und Fertigstellung der lokalisierten Version.

Schritt fünf beschreibt die Testphase, also die App auf Funktion und sprachliche Qualität zu überprüfen. Der nächste Schritt umfasst die Veröffentlichung, der letzte empfiehlt, nach der Veröffentlichung Support für die Kund/innen zur Verfügung zu stellen (vgl. Android Developer: Localization Checklist 2014; Apple Developer: Internationalization and Localization 2014).

Diese Informationen für Entwickler/innen geben einen guten Rahmen und passen insgesamt sehr gut zu den Erläuterungen der bisherigen Kapitel. Die Entscheidung, auf internationalen Märkten tätig sein zu wollen, sollte als Erstes getroffen werden. Hier wird das Konzept GILT widerspiegelt, also, nicht bei der Übersetzung zu beginnen, sondern auf höchster Ebene, um die Prozesse optimal planen zu können. Sich nachträglich für Lokalisierung zu entscheiden kann problematisch sein, weil durch nachträgliche Internationalisierung höhere Kosten entstehen. Weiters ist es natürlich wichtig, Software zu internationalisieren, um effizient zu lokalisieren. Umfassende Qualitätskontrollen und Tests sind auch bei Apps nötig, die Entwickler-Plattformen bieten sogar die Organisation von Apps Beta-Testern an.

Die verschiedenen Schritte erklären auch Entwickler/innen, die mit Lokalisierung noch nicht in Berührung gekommen sind, die Konzepte recht verständlich und sind eine praktische Anleitung. Wenn Entwickler/innen diesen Schritten folgen, sollten wenige Probleme auftauchen. Zieht man in Betracht, dass das fehlende Wissen über Lokalisierung bei „traditionellen“ Projekten oft zu größeren Problemen führen kann, ist dies an sich relativ wertvoll. Es sind natürlich die meisten Desktop-Anwendungen ungleich komplexer als Apps, weshalb eine einfache Darstellung, wie sie hier vorliegt, nicht unbedingt funktionieren würde.

8 Fazit

In den ersten Kapiteln wurden die Konzepte der Softwarelokalisierung beschrieben. Lokalisierung ist ein dynamischer Bereich, der sich in gut dreißig Jahren weit verbreitet und gleichzeitig stark verändert hat. Aus der Übersetzung von hartkodierten Strings ist eine umfassende Anpassung von Software geworden, die somit so gut wie möglich die Bedürfnisse der User/innen zufriedenstellt. Gleichzeitig damit wurde immer versucht, die Ressourcen besser zu nutzen und die Prozesse effizienter zu machen. Dies geschieht im Rahmen der Optimierung von Projektmanagement, der Implementierung von Terminologie- und Qualitätsmanagement, der Einsetzung von Tools und Softwarestandards und, nicht zuletzt, der Verbesserung der Kommunikation. So wird aus dem Lokalisierungsprojekt ein komplexes Ganzes, das viele Beteiligte vereint. Neben den Kompetenzen, die die einzelnen Mitarbeiter/innen haben müssen, sind es die Prozesse und Vernetzung, die dies alles möglich machen.

Sind diese Konzepte bei der Lokalisierung von Apps angekommen? Diese Frage kann mit Ja beantwortet werden. Natürlich kann jedes Unternehmen beziehungsweise jede/r App-Entwickler/in so verfahren, wie er/sie möchte, doch zeigen die umfassenden Informationen, Hilfen und Tools der beiden bedeutendsten App-Plattformen, dass die Internationalisierung und Lokalisierung von mobilen Apps ein geradliniger Prozess sein kann. Nicht nur werden Anleitungen und Checklisten angeboten, auch gewisse Dienste wurden bereits in der App Store-Infrastruktur implementiert: So können zum Beispiel, wie beschrieben, Übersetzungsdienste in der Android-Entwicklerkonsole direkt in Anspruch genommen werden. Auch die neueren Versionen der Betriebssysteme selbst sind lokalisierungsfreundlich ausgerichtet, wie zum Beispiel die Unterstützung von bidirektionalen Schriften und Layouts und die eingebauten Features, die für Internationalisierungstests eingesetzt werden können. Diese Dinge sind teilweise recht neu und so bleibt abzuwarten, wie sich die Plattformen in der Zukunft noch mehr auf Lokalisierung einstellen.

Die Unterschiede für die Lokalisierung von Apps ergeben sich aus den Merkmalen des Mediums Smartphone und der Programme selbst. Für Übersetzer/innen stellt sich die besondere Herausforderung der Bildschirmgröße. Da nicht viel Platz vorhanden ist, ist es nicht immer möglich, GUI-Elemente zu vergrößern. Es kann jedoch auf eigene Layout-Pakete ausgewichen werden, die als lokalisierte Ressource mit dem Locale verbunden ist.

Aufgrund ihres beschränkten Umfangs erreichen Lokalisierungsprojekte nur einen Bruchteil des Ausmaßes von herkömmlichen Projekten. Dadurch können beim Projektmanagement mehrere Schritte automatisiert werden, wie zum Beispiel die Angebotserstellung. Die Abwicklung der Aufträge ist somit auch einfacher, da es sich nicht um riesige Datenmengen handelt, die verschickt werden müssen. Über die Websites der Anbieter können die Dateien einfach hochgeladen werden und werden so direkt ins Projektmanagementsystem

aufgenommen. Nach der Lieferung können die lokalisierten Dateien relativ einfach importiert werden, wenn die App internationalisiert wurde.

Die Bedeutung der Plattform-Anbieter ist wohl nicht zu leugnen, da diese den Vertriebskanal kontrollieren. Hier ergibt sich für Lokalisierung auch ein bedeutender Unterschied. Da alle Apps über App Stores angeboten und heruntergeladen werden, gibt es keine weiteren Vertriebsketten, die organisiert werden müssen. Es muss kein Datenträger oder Begleitmaterial für alle Locales beschrieben beziehungsweise gedruckt werden, keine Produktion und kein Versand organisiert werden. Für die App Stores brauchen lediglich die Beschreibungen et cetera übersetzt werden, was im Notfall auch maschinell geschehen kann.

Es ist an der Anzahl der Übersetzungsdienstleister, die Lokalisierung von Apps anbieten, ersichtlich, dass sich die Industrie bereits darauf eingestellt hat. Es sind Informationen verfügbar, wie sich die Lokalisierung anstellen lässt und die Umgebung, die durch die Plattform-Anbieter geboten wird, ist ganz klar lokalisierungsfreundlich. Es ist natürlich so, dass Apps zu einem Zeitpunkt aufkamen, als die Lokalisierungsindustrie bereits weit fortgeschritten war, wie durch den ersten Teil der Arbeit ersichtlich ist. So ist es sieben Jahre nach Beginn der Smartphone-Ära möglich, dass durch die Erkenntnisse der Lokalisierungsindustrie das Lokalisieren von Apps schnell und einfach vorgenommen werden kann und User/innen auf der ganzen Welt die Funktionen und Vorteile nutzen können.

9 Quellen

9.1 Bibliografie

- Abel, Scott. 2010. Adding personalization to the localization mix. *MultiLingual*. #114 Volume 21 Issue 6, 39-44.
- Anthes, Gary. 2011. Invasion of the Mobile Apps. *Communications of the ACM*. Volume 54 No. 9, 16-18.
- Arnsperger, Jeff. 2013. Quantifying and measuring linguistic quality. *MultiLingual*. #139 Volume 24 Issue 7, 44-49.
- Arthur, K.A. & Brandt, B & Fedane, A. & Hannan, D. 2010. Automated text generation for the localisation of an online game. *Localisation Focus – The International Journal of Localisation*. Vol. 9 Issue 1, 36-45.
- Astashkina, Julia. 2005. Working With Virtual Teams. *Localization. The Guide from MultiLingual Computing & Technology*. #75 Supplement, October/November 2005, 14-15.
- Barceló, Curri. 2011. Games localization QA. *MultiLingual*. #122 Volume 22 Issue 6, 36-39.
- Baruch, Talia. 2012. Crafting a request for proposal. *MultiLingual*. #129 Volume 23 Issue 5, 36-38.
- Bass, Scott. 2006. Quality in the real world. In: Dunne, Keiran J. (Hg.), 69-94).
- Bateman, Ben. 2012. Localizing the whole living story. *MultiLingual*. #128 Volume 23 Issue 4, 32-34.
- Beatty, Jeff & Małolepszy, Staś. 2013. Open source localization. *MultiLingual*. #133 Volume 24 Issue 1, 28-32.
- Bell, Terena. 2011. Project management for languages of limited diffusion. *MultiLingual*. #123 Volume 22 Issue 7, 49-52.
- Bernal Marino, Miguel. 2008. What's in a "Game"? *Localisation Focus – The International Journal of Localisation*. Vol. 6 Issue 1, 29-38.
- Bracken, Conor. 2014. Localization maturity in emerging market languages. *MultiLingual*. #143 Volume 25 Issue 3, 18-20.
- Carter, Christoffer S. 2012. Emerging new markets. *MultiLingual*. #131 Volume 23 Issue 7, 25-29.
- Cattelan, Alessandro. 2014. Post-editing MT: Is it worth a discount? *MultiLingual*. #143 Volume 25 Issue 3, 40-41.

- Cerda, Paul. 2010. Beginning a career as a localization project manager. *MultiLingual*. #111 Volume 21 Issue 3, 39-42.
- Childress, Mark D. 2007. Terminology work saves more than it costs. *MultiLingual*. #87 Volume 22 Issue 3, 43-46.
- Corbolante, Licia & Irmeler, Ulrike. 2001. Software Terminology and Localization. In: Wright, Sue Ellen & Budin, Gerhard (Hg.), 516-535.
- Cyr, Dianne & Lew, Richard. 2003. Emerging Challenges in the Software Localization Industry. *Thunderbird International Business Review*. 45, 3, 337-358.
- DePalma, Donald A. 2006. Quantifying the return on localization investment. In: Dunne, Keiran J. (Hg.), 15-36.
- DePalma, Donald A. 2012. Language technology standards should support entire supply chain. *MultiLingual*. #127 Volume 23 Issue 3, 40.
- Dietz, Frank. 2006. Issues in localizing computer games. In: Dunne, Keiran J. (Hg.), 121-134.
- DiFranco, Carla. 2006. Localization Cost. In: Dunne, Keiran J. (Hg.), 47-66.
- Dunne, Keiran J.(Hg.). 2006. *Perspectives on localization*. Amsterdam/Philadelphia: John Benjamins.
- Dunne, Keiran J. 2006a. Introduction: A Copernican revolution: Focusing on the big picture of localization. In: Dunne, Keiran J. (Hg.), 1-11.
- Dunne, Keiran J. 2006b. Putting the cart behind the horse. Rethinking localization quality management. In: Dunne, Keiran J. (Hg.) 95-117.
- Dunne, Keiran J. 2007. Terminology: ignore it at your peril. *MultiLingual*. #87 Volume 18 Issue 3, 32-38.
- Edwards, Tom. 2006a. Flags, culture and politics. *MultiLingual*. #79 Volume 17 Issue 3, 25-27.
- Edwards, Tom. 2006b. Navigating the “Cultural Edge” of game content. *MultiLingual*. #80 Volume 17 Issue 4, 29-31.
- Edwards, Tom. 2007. The power of symbols. *MultiLingual*. #86 Volume 18 Issue 2, 31-33.
- Edwards, Tom. 2008. Time (in)sensitive content. *MultiLingual*. #93 Volume 19 Issue 1, 25-27.
- Edwards, Tom. 2009a. Symbolizing the sacred. *MultiLingual*. #101 Volume 20 Issue 1, 20-21.
- Edwards, Tom. 2009b. Sacred and secular. *MultiLingual*. #106 Volume 20 Issue 6, 28-29.

- Edwards, Kate. 2011. Cultural laws. *MultiLingual*. #119 Volume 22 Issue 2, 24-25.
- Edwards, Kate. 2012. Dealing with regime change. *MultiLingual*. #132 Volume 23 Issue 8, 14-15.
- Edwards, Kate. 2014. The creative difference. *MultiLingual*. #145 Volume 25 Issue 5, 14-15.
- Esselink, Bert. 2000. *A Practical Guide to Localization*. Amsterdam/Philadelphia: John Benjamins.
- Esselink, Bert. 2003. The Evolution of Localization. *Localization. The Guide from MultiLingual Computing & Technology*. #57 Supplement, July/August 2003, 4-7.
- Exton, Chris & Wasala, Asanka & Buckley, Jim & Schäler, Reinhard. 2009. Micro Crowdsourcing: A new Model for Software Localisation. *Localisation Focus – The International Journal of Localisation*. Vol. 8 Issue 1, 81-89.
- Exton, Chris & Spillane, Brendan & Buckley, Jim. 2010. A Micro-Crowdsourcing implementation: the Babel Software Project. *Localisation Focus – The International Journal of Localisation*. Vol. 9 Issue 1, 46-62.
- Fidura, Christie. 2007. The benefits of managing terminology with tools. *MultiLingual*. #87 Volume 18 Issue 3, 39-42.
- Fissgus, Ursula & Seewald-Heeg, Uta. 2005. Softwarelokalisierung. *it – Information Technology*. 47(4), 220-225.
- Ford, Gabrielle & Kotzé, Paula. 2005. Designing Usable Interfaces with Cultural Dimensions. In: *INTERACT'05 Proceedings of the 2005 IFIP TC13 international conference on Human-Computer Interaction*, 713-726.
- Gambín, José. 2014. Evolution of cloud-based translation memory. *MultiLingual*. #143 Volume 25 Issue 1, 46-49.
- Gillam, Richard. 2005. Unicode From 50,000 Feet. *Internationalization. The Guide from MultiLingual Computing & Technology*. #71 Supplement, April/May 2005, 8-9.
- Guillemin, Patrick & Trillaud, Sandrine. 2012. What has become of LISA's OSCAR standards? *MultiLingual*. #127 Volume 23 Issue 3, 38-41.
- Guerberof Arenas, Ana. 2010. Project management and machine translation. *MultiLingual*. #111 Volume 21 Issue 3, 34-38.
- Hall, Bill. 2002. Developing Software with Internationalization in Mind. *Multilingual*. Co-refocus 8-10.
- Harcz, Daniel B. 2011. Translation agency pricing. *MultiLingual*. #117 Volume 22 Issue 1, 24-25.

- He, Z. & Bustard, D.W & Liu, X. 2002. Software Internationalisation and Localisation: Practice and Evolution. In: *PPPJ '02/IRE '02 Proceedings of the inaugural conference on the Principles and Practice of programming, 2002 and Proceedings of the second workshop on Intermediate representation engineering for virtual machines*, 89-94.
- Henderson, Ian. 2010. Streamlining Internationalization and Localization. *Language Technology. Getting Started Guide. MultiLingual*. #111 Volume 21 Issue Supplement April/May 2010, 4-7.
- Helzer, Amir. 2011. Localizing for software, websites and global apps. *MultiLingual*. #119 Volume 22 Issue 3, 34-37.
- Herzog, Gottfried & Mühlbauer, Holger. 2007. *Normen für Übersetzer und technische Autoren*. Berlin/Wien: Beuth.
- Husbands, Gordon. 2005. Survey: Global Companies and Localized Market Communication. *Localization. The Guide from MultiLingual Computing & Technology*. #75 Supplement, October/November 2005, 3-7.
- Jiménez-Crespo, Miguel A. 2010. Web Internationalisation strategies and translation strategy: researching the case of “international” Spanish. *Localisation Focus – The International Journal of Localisation*. Volume 8 Issue 1, 13-25.
- Johnson, Dan. 2003. Localization Project Case Studies. *Localization. The Guide from MultiLingual Computing & Technology*. #57 Supplement, July/August 2003, 21-23.
- Jones, Andrew. 2007. Scheduling successful localization projects. *MultiLingual*. #89 Volume 18 Issue 5, 49-51.
- Karsch, Barbara Inge. 2006. Terminology workflow in the localization process. In: Dunne, Keiran J. (Hg.), 173-191.
- Karsch, Barbara Inge. 2011. Terminology survey results. *MultiLingual*. #119 Volume 22 Issue 3, 45-50.
- Keller, Nicole. 2011. Creating translation-oriented source documents. *MultiLingual*. #123 Volume 22 Issue 7, 43-45.
- Kelly, Nataly. 2009. Myths about crowdsourced translation. *MultiLingual*. #108 Volume 20 Issue 8, 62-63.
- Koehn, Philipp. 2010. Enabling Monolingual Translators: Post-Editing vs. Options. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, 537-545.
- Kuhn, Roland & Isabelle, Pierre & Goutte, Cyril & Senellart, Jean & Simard, Michel & Ueffing, Nicola. 2010. Automatic post-editing. *MultiLingual*. #110 Volume 21 Issue 2, 43-46.

- Lewis, David & Curran, Stephen & Doherty, Gavin & Feeney, Kevin & Karamanis, Niki-foros & Luz, Saturnino & McAuley, John. 2009. Supporting Flexibility and Awareness in Localisation Workflows. *Localisation Focus – The International Journal of Localisation*. Vol. 8 Issue 1, 29-38.
- Lombard, Robin. 2006. A practical case for managing source-language terminology. In: Dunne, Keiran J. (Hg.), 155-172.
- Lutsuk, Andriy. 2014. Getting started with game localization. *MultiLingual*. #144 Volume 25 Issue 4, 27-30.
- Makosiej, Marek. 2013. Three reasons to consider an SLV. *MultiLingual*. #134 Volume 24 Issue 2, 22-23.
- Marcus, Aaron. 2005. User Interface Design and Culture. In: Aykin, Nuray (Hg.). *Usability and Internationalization of Information Technology*. Mahwah, NJ: Lawrence Erlbaum, 43-65.
- Massion, François. 2007. Terminology management: a luxury or a necessity? *MultiLingual*. #87 Volume 18 Issue 3, 47-50.
- Massion, François. 2010. Automating translation quality management. *MultiLingual*. #109 Volume 21 Issue 1, 41-43.
- Massion, François. 2011. Localization of machine software. *MultiLingual*. #119 Volume 22 Issue 3, 40-44.
- Mayer, Ansgar. 2012. *App Economy*. München: mi-Wirtschaftsbuch (Münchner Verlagsgruppe).
- Muddyman, Gary. 2012 The rise of CIVETS economies. *MultiLingual*. #131 Volume 23 Issue 7, 30-33.
- Müller, Eva. 2005. Step-by-step Localization. *Localization. The Guide from MultiLingual Computing & Technology*. #75 Supplement, October/November 2005.
- Orfall, Brad. 2010. Case study: TM economics in project management. *MultiLingual*. #111 Volume 21 Issue 3, 30-33.
- Peng, Wenlin & Yang, Xiaohu & Zhu, Feng. 2009. Automation technique of software internationalization and localization based on lexical analysis. In: *ICIS '09 Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, 970-975.
- Reiß, Katharina & Vermeer, Hans J. 1984. *Grundlegung einer allgemeinen Translationstheorie*. Tübingen: Niemeyer.
- Reuveni, Doron. 2012. Crowdsourcing your localization testing. *MultiLingual*. #125 Volume 23 Issue 1, 47-50.

- Roland, Jessica. 2014. People-powered translation at machine speed. *MultiLingual*. #141 Volume 25 Issue 1, 21-23.
- Rueda, Kathrin. 2010. QA in translation: a process approach. *MultiLingual*. #109 Volume 21 Issue 1, 38-40.
- Safar, Libor. 2014. Localizing content for the UK market. *MultiLingual*. #142 Volume 25 Issue 2, 31-35.
- Sager, Juan C. 1997. Term Formation. In: Wright, Sue Ellen & Budin, Gerhard (Hg.), 25-41.
- Sargent, Benjamin B. 2012. The growing market of global information consumers. *MultiLingual*. #131 Volume 23 Issue 7, 49-50.
- Savourel, Yves. 2014. An introduction to XLIFF 2.0. *MultiLingual*. #144 Volume 24 Issue 4, 42-47.
- Scattergood, Damian. 2003. Documentation Localisation Costs. *Localisation Reader*. 2003-2004, 13-16.
- Schäler, Reinhard. 2003. The Cultural Dimension in Software Localisation. *Localisation Reader*. 2003-2004, 5-8.
- Schäler, Reinhard. 2008. Reverse Localisation. *Localisation Focus. The International Journal of Localisation*. Volume 6 Issue 1, 39-48
- Sikes, Richard. 2010. Five types of localization testing. *MultiLingual*. #109 Volume 21 Issue 1, 33-37.
- Spacinsky, Denise. 2013. Careers in localization. *MultiLingual*. #140 Volume 24 Issue 8, 25-30.
- Spacinsky, Denise. 2014. The Western European localization job market. *MultiLingual*. #142 Volume 25 Issue 2, 16-19.
- Steiert, Afaf & Steiert, Matthias & Mariniello, Elanna. 2010. No recession for translation. *MultiLingual*. #114 Volume 21 Issue 6, 45-47.
- Steiert, Yasin & Steiert, Afaf. 2014. The art of video game translation. *MultiLingual*. #144 Volume 25 Issue 4, 58.
- Stolze, Radegundis. 1999. *Die Fachübersetzung – eine Einführung*. Tübingen: Narr.
- Sweezey, Marcia Rose & Visuri, Stefan. 2013. Money-saving tips for the new localization coordinator. *MultiLingual*. #140 Volume 24 Issue 8, 51-53.
- Tatsumi, Midori & Sun, Yanli. 2008. A Comparison of Statistical Post-Editing on Chinese and Japanese. *Localisation Focus. The International Journal of Localisation*. Volume 7 Issue 1, 22-33.

- Thicke, Lori. 2010. MT's 'perfect storm,' Russian and beyond. *MultiLingual*. #110 Volume 21 Issue 2, 40-42.
- Thicke, Lori. 2011. Improving MT results: a study. *MultiLingual*. #117 Volume 22 Issue 11, 37-40.
- Thicke, Lori. 2013a. Post-editor shortage and MT. *MultiLingual*. #133 Volume 24 Issue 1, 42-45.
- Thicke, Lori. 2013b. Expanding reach through crowdsourcing. *MultiLingual*. #134 Volume 24 Issue 2, 24-27.
- Tishin, Demid. 2014. A quest for quality in video game localization. *MultiLingual*. #144 Volume 25 Issue 4, 35-38.
- van der Meer, Jaap. 2003. Impact of Translation Web Services. *Localisation Reader*. 2003-2004, 9-12.
- van Genabith, Josef. 2009. Next Generation Localisation. *Localisation Focus. The International Journal of Localisation*. Volume 8 Issue 1, 4-10.
- Wasala, Asanka & Schäler, Reinhard & Weerasinghe, Ruwan & Exton, Chris. 2012. Collaboratively Building Language Resources while Localising the Web. In: *Proceedings of the 3rd Workshop on the People's Web Meets NLP, ACL 2012*, Jeju, Republic of Korea, 8-14 July 2012, 15-19.
- Waßmer, Thomas. 2011. SDL Trados Studio 2011. *MultiLingual*. #124 Volume 22 Issue 8, 14-19.
- Westland, J. Christopher. 2002. The cost of errors in software development: evidence from the industry. *The Journal of Systems and Software*. Volume 62(1), 1-9.
- Wittner, Janaina. 2007. Unexpected ROI from terminology. *MultiLingual*. #87 Volume 18 Issue 3, 51-54.
- Wright, Sue Ellen & Budin, Gerhard (Hg.). 1997. *Handbook of terminology management. 1. Basic aspects of terminology management*. Amsterdam/Philadelphia: John Benjamins.
- Wright, Sue Ellen & Budin, Gerhard (Hg.). 2001. *Handbook of terminology management. II. Applications-oriented terminology management*. Amsterdam/Philadelphia: John Benjamins.
- Wright, Sue Ellen. 2001. Terminology and Total Quality Management. In: Wright, Sue Ellen & Budin, Gerhard (Hg.), 488-502.
- Wyld, David C. 2010. Cloud computing around the world. *MultiLingual*. #109 Volume 21 Issue 1, 44-49.
- Yeo, Alvin. 1996. Software Internationalisation and Localisation. In: *Proceedings Sixth Australian Conference on Computer-Human Interaction 1996*, 348-349

- Zerfaß, Angelika. 2005. Assembling a Localization Kit. *Localization. The Guide from MultiLingual Computing & Technology*. #75 Supplement, October/November 2005, 8-11.
- Zerfaß, Angelika. 2008. Localization Technology. *Going Global. Getting Started Guide. MultiLingual*. #95 Volume 19 Issue 3, 8-9.
- Zetzsche, Jost. 2012. Translation technology comes full circle. *MultiLingual*. #127 Volume 23 Issue 3, 50-51.
- Zimmerman, Shannon. 2014. Cloud security for SaaS translation providers. *MultiLingual*. #143 Volume 25 Issue 1, 42-43.
- Zydroń, Andrzej. 2012. Cloud computing, SaaS and translation tools. *MultiLingual*. #125 Volume 23 Issue 1, 20-21.

9.2 Onlinequellen

- Amadeo, Ron. 2014. „The history of Android”.
<http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/25/>, Stand: 08.08.2014.
- Android Developer: First App. URL:
<http://developer.android.com/training/basics/firstapp/index.html>; Stand: 08.08.2014.
- Android Developer: Help. URL: <http://developer.android.com/design/patterns/help.html>;
 Stand: 08.08.2014.
- Android Developer: Jelly Bean. URL: <http://developer.android.com/about/versions/jelly-bean.html>; Stand: 08.08.2014.
- Android Developer: L Preview. URL: <http://developer.android.com/preview/setup-sdk.html>;
 Stand: 10.08.2014.
- Android Developer: Localization Checklist. URL:
<http://developer.android.com/distribute/tools/localization-checklist.html>; Stand:
 08.08.2014.

Android Developer: Localizing with Resources. URL:

<http://developer.android.com/guide/topics/resources/localization.html>; Stand: 08.08.2014.

Android Developer: Resources Overview. URL:

<http://developer.android.com/guide/topics/resources/index.html>; Stand: 08.08.2014.

Android Developer: SDK. URL: <http://developer.android.com/sdk/index.html>; Stand:

08.08.2014.

Android Developer: String Resources. URL:

<http://developer.android.com/guide/topics/resources/string-resource.html>; Stand: 08.08.2014.

Android Developer: Supporting Different Languages. URL:

<http://developer.android.com/training/basics/supporting-devices/languages.html>; Stand: 08.08.2014.

Android Developer: Training. URL: <http://developer.android.com/training/index.html>; Stand:

08.08.2014.

App Annie Google Play Top App Charts. URL: <http://www.appannie.com/apps/google-play/top/austria/>; Stand: 15.08.2014.

App Annie Localization. URL: <http://blog.appannie.com/localization-entry/>; Stand:

15.07.2014.

App Store FAQ. URL: <http://support.apple.com/kb/HT2001>; Stand: 15.08.2014.

AppBrain Stats Free vs. Paid Android Apps. URL: <http://www.appbrain.com/stats/free-and-paid-android-applications>; Stand: 15.08.2014.

AppBrain Stats Play Store. URL: <http://www.appbrain.com/stats/number-of-android-apps>;

Stand: 16.08.2014.

Apple Developer: About App Distribution. URL:

https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html#//apple_ref/doc/uid/TP40012582; Stand: 08.08.2014.

Apple Developer: First Steps. URL:

https://developer.apple.com/library/ios/documentation/LanguagesUtilities/Conceptual/iTunesConnect_Guide/Chapters/FirstSteps.html; Stand: 08.08.2014.

Apple Developer: Guidelines for Internationalization. URL:

https://developer.apple.com/library/ios/documentation/MacOSX/Conceptual/BPInternational/Articles/InternationalGuidelines.html#//apple_ref/doc/uid/20002142-BBCFJBFB; Stand: 08.08.2014.

Apple Developer: Internationalization and Localization. URL:

https://developer.apple.com/library/ios/documentation/MacOSX/Conceptual/BPInternational/Articles/InternatAndLocaliz.html#//apple_ref/doc/uid/20000277-SW1; Stand: 08.08.2014.

Apple Developer: Localizing String Resources. URL:

https://developer.apple.com/library/ios/documentation/MacOSX/Conceptual/BPInternational/Articles/StringsFiles.html#//apple_ref/doc/uid/20000005-SW1; Stand: 08.08.2014.

Apple Developer: TestFlight. URL: <https://developer.apple.com/support/appstore/TestFlight/>; Stand: 08.08.2014.

Apple Developer: Tutorial Basics. URL:

https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/FirstTutorial.html#//apple_ref/doc/uid/TP40011343-CH3-SW1; Stand: 08.08.2014.

Arcade Museum. URL: http://www.arcade-museum.com/game_detail.php?game_id=10530; Stand: 30.07.2014.

Ball, James. 2014. „Angry Birds and ‚leaky‘ phone apps targeted by NSA and GCHQ for user data”. <http://www.theguardian.com/world/2014/jan/27/nsa-gchq-smartphone-app-angry-birds-personal-data>, Stand: 28.01.2014.

Bayer, Florian. 2014. „An app a day keeps no doctor away“.

<http://derstandard.at/1392686437246/An-app-a-day-keeps-no-doctor-away>, Stand: 26.02.2014.

- Binder, Stefan. 2014. „Libanon: Überlebens-App für gefährliche Zeiten“. <http://derstandard.at/1389859080513/Ueberlebens-App-fuer-haertere-Zeiten-im-Libanon>, Stand: 03.02.2014.
- Bradley, Tony. 2013^a. „Apple Can Ignore Android Market Share Stats All The Way To The Bank“. <http://www.forbes.com/sites/tonybradley/2013/10/29/apple-can-ignore-android-market-share-stats-all-the-way-to-the-bank/>, Stand: 02.08.2014.
- Bradley, Tony. 2013^a. „Android Dominates Market Share, But Apple Makes All The Money“. <http://www.forbes.com/sites/tonybradley/2013/11/15/android-dominates-market-share-but-apple-makes-all-the-money/>, Stand: 02.08.2014.
- Byrne, Jody. 2009. „Localisation – When Language, Culture and Technology Join Forces“. <http://ojs.statsbiblioteket.dk/index.php/law/article/view/6190/5378>, Stand: 06.05.2014.
- Byte Notes. URL: <http://www.byte-notes.com/four-types-computers>; Stand: 03.04.2014.
- Costello, Sam. 2014. „iPhone Firmware & iOS History“. http://ipod.about.com/od/iphonesoftwareterms/a/firmw_history.htm, Stand: 10.08.2014.
- Cowie, Jim. 2014. „Syria, Venezuela, Ukraine: Internet Under Fire“. <http://www.renesys.com/2014/02/internetunderfire/>, Stand: 26.02.2014.
- CTV News. URL: <http://www.ctvnews.ca/sci-tech/welcome-to-the-post-pc-world-tablet-sales-to-overtake-desktops-laptops-1.1391171>; Stand: 30.07.2013.
- Duden: App. URL: <http://www.duden.de/rechtschreibung/App>; Stand: 04.08.2014.
- Duden: Smartphone. URL: <http://www.duden.de/rechtschreibung/Smartphone>; Stand: 04.08.2014.
- Floemer, Andreas. 2012. „3.997 Smartphone-Modelle: So fragmentiert ist der Android-Markt“. <http://t3n.de/news/3997-smartphone-modelle-388360/>, Stand: 03.08.2014.
- Forbes GTA 5. URL: <http://www.forbes.com/sites/davidthier/2014/05/13/grand-theft-auto-5-has-sold-nearly-2-billion-at-retail/>; Stand: 13.05.2014.

Free Software Foundation. URL: <http://www.fsf.org/>, Stand: 12.06.2014.

Gartner. URL: <http://www.gartner.com/newsroom/id/2614915>; Stand: 29.10.2013.

Get Localization. URL: <https://www.getlocalization.com/crowdsourcing/>, Stand: 12.06.2014.

Google Play Store. URL: <https://play.google.com/store>; Stand: 08.08.2014.

Google Support: Apps hochladen. URL: <https://support.google.com/googleplay/android-developer/answer/113469>; Stand: 08.08.2014.

Google Support: Kategorien. URL: <https://support.google.com/googleplay/android-developer/answer/113475>; Stand: 08.08.2014.

Google Support: Standorte. URL: <https://support.google.com/googleplay/android-developer/table/3541286?rd=1>; Stand: 08.08.2014.

Google Support: Vertrieb. URL: https://support.google.com/googleplay/android-developer/answer/3502576?hl=de&ref_topic=3450781; Stand: 08.08.2014.

Google Support: Zahlungsmethoden. URL:
<https://support.google.com/googleplay/answer/2651410?hl=de>; Stand: 15.08.2014.

Heise Online. URL: <http://www.heise.de/newsticker/meldung/Programmierer-kritisieren-Apples-App-Store-205628.html>; Stand: 15.09.2008.

Howe, Jeff. 2006. „The Rise of Crowdsourcing”.
<http://archive.wired.com/wired/archive/14.06/crowds.html>, Stand: 10.06.2014.

ICanLocalize. URL: <http://www.icanlocalize.com/site/services/android-localization/>; Stand: 10.08.2014.

IDC 4Q13. URL: <http://www.idc.com/getdoc.jsp?containerId=prUS24676414>; Stand: 02.08.2014.

IDC 2Q14. URL: <http://www.idc.com/getdoc.jsp?containerId=prUS25037214>; Stand: 14.08.2014.

Internet World Stats. URL: <http://www.internetworldstats.com/stats.htm>; Stand: 01.05.2014.

ITWissen: Mooresches Gesetz. URL:

<http://www.itwissen.info/definition/lexikon/Mooresches-Gesetz-Moores-law.html>;

Stand: 01.05.2014.

ITWissen: Tablet-PC. URL: [http://www.itwissen.info/definition/lexikon/Tafel-PC-tablet-](http://www.itwissen.info/definition/lexikon/Tafel-PC-tablet-PC.html)

[PC.html](http://www.itwissen.info/definition/lexikon/Tafel-PC-tablet-PC.html); Stand: 05.08.2014.

LibreOffice Versionshinweise. URL: <http://de.libreoffice.org/download/versionshinweise/>;

Stand: 30.07.2014.

Lunden, Igrid. 2013. „Gartner: 102B App Store Downloads Globally in 2013, \$26B In Sales,

17% From In-App Purchases”. <http://techcrunch.com/2013/09/19/gartner-102b-app-store-downloads-globally-in-2013-26b-in-sales-17-from-in-app-purchases/>, Stand:

10.08.2014.

Mac Developer Library. URL:

[https://developer.apple.com/library/mac/documentation/Carbon/Conceptual/Providing](https://developer.apple.com/library/mac/documentation/Carbon/Conceptual/ProvidingUserAssitAppleHelp/user_help_intro/user_assistance_intro.html)

[UserAssitAppleHelp/user_help_intro/user_assistance_intro.html](https://developer.apple.com/library/mac/documentation/Carbon/Conceptual/ProvidingUserAssitAppleHelp/user_help_intro/user_assistance_intro.html); Stand: 30.07.2014.

Mahajan, Rohit & Shneiderman, Ben. 1997. „Visual and Textual Consistency Checking

Tools for Graphical User Interfaces”. <http://hci12.cs.umd.edu/trs/96-08/96-08.html>,

Stand: 01.07.2014.

Maxwell, Aaron. 2011. „Is Developing a Mobile App Worth the Cost?“

<http://mashable.com/2011/02/24/mobile-app-dev-cost/>; Stand: 10.08.2014.

Mestre, Rosanna. 2009. „Web 2.0 under the light of free software“.

<http://ojs.statsbiblioteket.dk/index.php/law/article/view/6174/5362>, Stand:

01.06.2014.

Microsoft Office-Support. URL: [http://office.microsoft.com/de-at/help/abrufen-von-hilfe-](http://office.microsoft.com/de-at/help/abrufen-von-hilfe-vorlagen-schulungen-und-zusatzlichen-inhalten-HP010354254.aspx#BM1)

[vorlagen-schulungen-und-zusatzlichen-inhalten-HP010354254.aspx#BM1](http://office.microsoft.com/de-at/help/abrufen-von-hilfe-vorlagen-schulungen-und-zusatzlichen-inhalten-HP010354254.aspx#BM1); Stand:

30.07.2014.

Murtazin, Eldar. 2010. „Apple’s Phone: From 1980’s Sketches to iPhone. Part 3“.

<http://mobile-review.com/articles/2010/iphone-history3-en.shtml>, Stand: 03.08.2014.

Netindex. URL: <http://www.netindex.com/>; Stand: 01.05.2014.

OpenOffice. URL: <http://www.openoffice.org>; Stand: 12.06.2014.

Open Source Initiative. URL: <http://opensource.org/>; Stand: 12.06.2014.

PR Newswire. URL:

<http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=2856283>; Stand: 09.04.2014.

Raymond, Eric Steven. 2002. „The Cathedral and the Bazaar”.

<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html#catbmain>; Stand: 12.06.2014.

Sager, Ira. 2012. „Before iPhone and Android Came Simon, the First Smartphone”.

<http://www.businessweek.com/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone>, Stand: 02.08.2014.

Schmitz, Klaus-Dirk. 2009. „Terminological recommendations for software localization”.

<http://ojs.statsbiblioteket.dk/index.php/law/article/view/6191/5379>, Stand: 05.04.2014.

Shoutmeloud. URL: <http://www.shoutmeloud.com/top-mobile-os-overview.html>; Stand: 10.08.2014.

Slideshare: ROI. URL: <http://de.slideshare.net/PocketYourShop/5-mobile-app-roi-metrics-that-actually-matter>; Stand: 10.08.2014.

Springer Gabler: Smartphone. URL:

<http://wirtschaftslexikon.gabler.de/Definition/smartphone.html>; Stand: 02.08.2014.

Stallman, Richard. „Why Open Source misses the point of Free Software”.

<https://www.gnu.org/philosophy/open-source-misses-the-point.en.html>, Stand: 12.04.2014.

Statista Smartphone OS. URL: <http://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>; Stand: 15.07.2014.

Strategy Analytics. URL: <http://blogs.strategyanalytics.com/WSS/post/2014/07/30/Android-Captured-Record-85-Percent-Share-of-Global-Smartphone-Shipments-in-Q2-2014.aspx>; Stand: 30.07.2014.

Techscribe. URL: <http://www.techscribe.co.uk/techw/help-comparison.htm>; Stand: 30.07.2014.

The Hofstede Centre. URL: <http://geert-hofstede.com/national-culture.html>; Stand: 09.04.2014.

Vogelstein, Fred. 2013. „The Day Google Had to ‘Start Over’ on Android”.
<http://www.theatlantic.com/technology/archive/2013/12/the-day-google-had-to-start-over-on-android/282479/>, Stand: 02.08.2014.

Windows Help. URL: [http://msdn.microsoft.com/en-us/library/windows/desktop/ms728460\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms728460(v=vs.85).aspx); Stand: 30.07.2014.

WinFuture. URL: <http://winfuture.de/news,77955.html>; Stand: 21.09.2013.

Kurzfassung

Mobile Apps sind kleine Programme, die auf Smartphones verwendet werden können. Es gibt eine große Vielfalt an Apps. Um ein globales Publikum zu erreichen müssen sie lokalisiert werden.

Um festzustellen, wie dies zu bewerkstelligen ist, wird zuerst der Stand der Lokalisierungsindustrie besprochen. Es werden Lokalisierung und Internationalisierung beschrieben und wie sich die Lokalisierungsindustrie in den letzten Jahren entwickelt hat. Die wichtigsten Aspekte, die im Rahmen eines Lokalisierungsprojekts von Bedeutung sind, werden anschließend erläutert: Projektmanagement, Terminologiemanagement, Übersetzung und Qualitätsmanagement, gefolgt von der Beschreibung der Software, die bei Softwarelokalisierungsprojekten verwendet wird.

Den Ausgangspunkt für die Besprechung der Lokalisierung von mobilen Apps bilden das Medium Smartphone und die Besonderheiten von Apps als Programmen. Schließlich wird beschrieben, wie Apps für die Smartphone-Betriebssysteme Android und iOS lokalisiert werden können und festgestellt, welche Faktoren dies von der Lokalisierung von PC-Software abheben. Die Hauptunterschiede ergeben sich aus der Umgebung, die von den Herstellern der Betriebssysteme geboten wird, dem geringen Umfang von mobilen Apps und deren Verwendung auf Smartphones, die als tragbare Geräte über kleinere Bildschirme und wenig Speicherplatz verfügen.

Abstract

Mobile apps are small applications that run on smartphones. They come with a big variety of features and functions. In order to make them usable by people all over the planet, they have to be localised.

In order to find out how apps can be localised, the first step is to describe the status quo of the localisation industry. Therefore, localisation and internationalisation are discussed, as well as the development of the localisation industry in the past few years. The most important aspects of localisation projects are explained subsequently. These are: project management, terminology management, translation and quality management, followed by a description of the most important tools used in localisation.

Smartphones and mobile apps, as well as the special characteristics of apps are explained, followed by an account of how Android and iOS apps can be localised. Finally, the conclusion is drawn as to how the localisation of mobile apps differs from localising PC software. The main differences are the localisation-friendly environment provided by the providers of the operating systems, the small size of app localisation projects and the fact, that smartphones have limited screen sizes and disk space.

Lebenslauf

Angaben zur Person	
Nachname, Vorname	Rathbauer, David
Titel	Bachelor of Arts
Schul- und Universitätsausbildung	
Zeitraum	seit März 2013
Studienbezeichnung	Masterstudium Dolmetschen
Nähere Angaben zur Ausbildung	Arbeitssprachen: A-Deutsch, B-Englisch, C-Spanisch;
Name und Anschrift der Bildungseinrichtung	Universität Wien - Zentrum für Translationswissenschaften Gymnasiumstraße 50, 1190 Wien
Zeitraum	seit März 2012
Studienbezeichnung	Masterstudium Übersetzen
Nähere Angaben zur Ausbildung	Arbeitssprachen: A-Deutsch, B-Englisch, C-Spanisch;
Name und Anschrift der Bildungseinrichtung	Universität Wien - Zentrum für Translationswissenschaften Gymnasiumstraße 50, 1190 Wien
Zeitraum	Oktober 2007 – Januar 2012
Studienbezeichnung	Bachelorstudium Transkulturelle Kommunikation
Bezeichnung des erworbenen Abschlusses	Bachelor of Arts
Nähere Angaben zur Ausbildung	Arbeitssprachen: A-Deutsch, B-Englisch, C-Spanisch
Name und Anschrift der Bildungseinrichtung	Universität Wien - Zentrum für Translationswissenschaften Gymnasiumstraße 50, 1190 Wien
Zeitraum	1998 – 2006
Bezeichnung des erworbenen Abschlusses	Matura
Nähere Angaben zur Ausbildung	Neusprachliches Gymnasium
Name und Anschrift der Bildungseinrichtung	Bundesgymnasium/Bundesrealgymnasium Wieselburg Erlaufpromenade 1, 3250 Wieselburg

Berufserfahrung	
Zeitraum	Oktober 2012 – April 2013
Funktion	Übersetzer
Nähere Angaben zur Tätigkeit	Untertitelung von Fernsehbeiträgen in den Sprachen Deutsch, Englisch und Spanisch beim Community TV-Sender Okto.
Name und Adresse des Arbeitgebers	Okto. Community TV-GmbH Goldschlagstraße 172/4, 1140 Wien
Zeitraum	seit April 2008
Funktion	Nachhilfelehrer
Nähere Angaben zur Tätigkeit	Begleitende Nachhilfe während des Schuljahres in den Fächern Englisch, Deutsch und Spanisch sowie Intensivkurse zur Vorbereitung auf Nachprüfungen.
Name und Adresse des Arbeitgebers	Christiane Humer – Die Nachhilfe GmbH Grünbach 27, 4623 Gunskirchen
Zusatzqualifikationen	
Zeitraum	Oktober 2010 – Februar 2013
Studienrichtung	Skandinavistik
Nähere Angaben zur Ausbildung	Interessensmodul im Bereich Sprache, Kultur und Gesellschaft Norwegens. Beinhaltet u.a. Norwegisch-Deutsch Übersetzungs-Lehrveranstaltungen.
Name und Anschrift der Bildungseinrichtung	Universität Wien Dr.-Karl-Lueger-Ring 1, 1010 Wien
Auslandsaufenthalte	
Zeitraum	Februar 2010 – Juli 2010
Art des Auslandsaufenthalts	Erasmus-Semester
Nähere Angaben	Auslandssemester in Spanien zur Vertiefung der sprachlichen und kulturellen Kompetenzen.
Name und Anschrift der Bildungseinrichtung	Universitat Autònoma de Barcelona Plaça Cívica, Campus de la UAB, 08193 Sardañola del Vallés, Barcelona, Spanien

Zeitraum	August 2004 – Juni 2005
Art des Auslandsaufenthalts	Schüleraustausch
Nähere Angaben	Austauschjahr im Norden Norwegens.
Name und Anschrift der Bildungseinrichtung	Nordreisa Vidergående Skole Hovedveien 18, 9151 Storslett, Norwegen
Soziale und karitative Tätigkeiten	
Zeitraum	Oktober 2006 – Juni 2007
Art der Tätigkeit	Zivildienst
Name und Anschrift der Organisation	Rotes Kreuz Bezirksstelle Scheibbs Rutesheimer Straße 3, 3270 Scheibbs
Zeitraum	Oktober 2005 – Mai 2009
Art der Tätigkeit	Freiwilliger für AFS Österreich Im Zuge der Tätigkeit als Freiwilliger habe ich Austauschschüler und Gastfamilien betreut und im Rahmen von Wochenendseminaren Austauschschüler auf ihre Aufenthalte vorbereitet sowie nach ihren Auslandsaufenthalten nachbetreut.
Name und Anschrift der Organisation	AFS Austauschprogramme für interkulturelles Lernen Maria-Theresien-Straße 9/6, 1090 Wien