

Der Nutzer – ein unbekanntes Wesen für Software-Tester?

Softwareentwicklung begreift sich als zielorientierter, systematischer Einsatz von Prinzipien, Methoden und Werkzeugen für die Entwicklung und Anwendung von umfangreichen Softwaresystemen (Balzert, 1998). Zuerst setzt man sich Ziele. Anschließend testet man, ob diese erreicht wurden. Obwohl es in der täglichen Arbeit oft unausgesprochen bleibt, das grundlegendste Ziel von Softwareentwicklung dürfte dabei sein, den Nutzern einen Mehrwert zu schaffen. Überrascht es nicht, dass dieses Ziel nach wie vor nur selten systematisch überprüft wird? Wie oft werden die Nutzer zu solchen Tests eingeladen, der möglichst realitätsgetreue aber kontrollierte Bedingungen einhält, und in dem die Nutzer sich mit einem Prototypen auseinandersetzen? Sehr viel Aufmerksamkeit widmen die Unternehmen hingegen der technischen Reife ihrer Produkte. Unbestritten ist die technische Reife eine notwendige Bedingung für ein erfolgreiches Produkt. Für einen überzeugten Verfechter der User Experience¹ ist trotzdem nicht nachvollziehbar, wie wenig der Nutzer in der Softwareentwicklung häufig vorkommt. Dass auch die Bedienung und das Erlebnis bei der Nutzung geprüft und optimiert werden, bevor das Produkt den Markt erreicht, ist eher die Ausnahme als die Regel.

Tests aus Sicht der Softwareentwicklung

Wer sich näher mit dem Thema Testen beschäftigt, trifft auf viele verschiedene Listen und Taxonomien, die das Feld in mehrere Bereiche einteilen. Zuerst kann man das Testen mehr oder weniger systematisch angehen. Das zeigt die Unterscheidung der folgenden Ansätze

- Plausibilitätstests
- Ad-Hoc Tests
- Exploratives Testen
- Geplantes Testen

Wichtig an dieser Unterscheidung ist, dass der Grad der Systematik auch über den Erfolg von Tests entscheidet. Ad-hoc und Plausibilitätstests garantieren bei der Komplexität der meisten Systeme heutzutage kaum eine hinreichende Qualität. Sie sind zwar durchaus anzutreffen und je nachdem, wie eine Kosten-Nutzen-Rechnung ausfällt, haben auch diese Tests ihre Berechtigung. Die folgenden Seiten konzentrieren jedoch

¹ 1. nutzerzentriertes Arbeiten ist ein Kennzeichen von Usability und User Experience gleichermaßen; 2. der Autor betrachtet Usability als Teilmenge von User Experience. Aus beiden Gründen sprechen diese Seiten lediglich von User Experience (UX).

vielmehr auf systematischere Vorgehensweise, die eher dem explorativen oder geplanten Testen zugeordnet werden können.

In der Softwareentwicklung existiert ein ganzes Sammelsurium unterschiedlicher Tests. Allein die vielen Teststufen zeigen wie ausgiebig getestet wird. Alle diese Tests haben ihre Berechtigung. Auch wenn nicht jeder Hersteller jede Art von Test benötigt und durchführt, so wird die Sinnhaftigkeit grundsätzlich nicht in Frage gestellt. Je nach Autor gibt es bis zu acht Teststufen:

- Code Review
- Unit Test
- Integrationstest
- Systemtests
- Akzeptanztests
- Beta-Test
- Abnahmetest
- Deployment-Test

Noch eindrucksvoller wird die Liste, wenn man bedenkt, dass die Teststufen zunehmend differenzierter werden, je weiter der Grad an Integration steigt. Denn kein Test ist in der Lage alle erforderlichen Aspekte zu erfassen. Bei den Systemtests finden wir bereits zahlreiche unterschiedliche Testvarianten, davon unter anderem:

- Performanztests
- Kompatibilitätstests
- Ausnahmetests (exception tests)
- Sicherheitstests (security tests)
- Lasttests, Stress Tests; Crash Tests
- Volumen-/Massentests
- Skalierungstests (scalability tests)
- Installationstests
- Wiederfindungstest (recovery tests)
- Datenkonsistenztests
- Schnittstellentests
- Oberflächentests

Die Erkenntnis, dass mit der Formulierung der Anforderungen nicht alles gesagt ist, und dass Softwareentwicklung folglich iterativ und nicht linear erfolgen muss, hat sich längst durchgesetzt. Trotzdem werden Nutzer häufig erst ab den Systemtests mit der Neuentwicklung konfrontiert ... wenn überhaupt. Und auch unter den Systemtests dominieren die technischen Aspekte bei Weitem. Und auch bei den Systemtests ist die Beteiligung von Endnutzern nicht häufig:

- Usability Tests
- Accessibility testing²

Wo ist das Problem?

Was sofort ins Auge springt, ist die enorme Diskrepanz zwischen Technologie und User Experience. Beispielsweise im Lehrbuch „Basiswissen Softwaretest“ (Spillner & Linz; 2004) findet man den Nutzer nur einmal im Index unter dem Abschnitt „Test auf Benutzerakzeptanz“. Der Abschnitt umfasst insgesamt etwa drei Promille des gesamten Buches. Wohingegen die Technik nach allen Regeln der Kunst getestet wird, bis wirklich fast jede Schwäche entdeckt und behoben ist. Die nutzerorientierten Tests kommen in der Liste zwar vor, sind aber einerseits unterrepräsentiert und andererseits sind sie in der Praxis der Unternehmen nach wie vor eine große Ausnahme. Erfreulich ist nur, dass die Zahlen langsam steigen. Das können wir zumindest erahnen, wenn wir die Branchenreports der deutschen UPA-Sektion (Usability Professional Association) ansehen³ (UPA, 2003-2013).

Die technische Reife ist unbestritten eine notwendige Bedingung für den Erfolg eines Produkts, aber eben keine hinreichende. Hinreichend ist nur die User Experience: der erlebte Umgang mit dem Produkt. Direkten Kontakt mit dem Produkt haben Nutzer allein über das User Interface. Der technische Unterbau ist nicht sichtbar und kaum ein Nutzer versteht, was dort genau vor sich geht. Der Eindruck, den das Produkt hinterlässt, entsteht allein durch die Ereignisse an der Oberfläche. Der entscheidende Unterschied zur Konkurrenz kann nur in der Interaktion zwischen Nutzer und Produkt liegen⁴.

Aufgrund dieser Begründung lässt sich folgende Kernforderung aufstellen:

Systematische und etablierte Testverfahren unter der Beteiligung von **Endnutzern** zu einem möglichst **frühen Zeitpunkt**, an dem Optimierungen der Bedienoberfläche noch möglich sind.

² Test auf Barrierefreiheit (Accessibility) sind natürlich nur relevant, wenn dies als besondere Anforderung festgelegt wurde.

³ Die Zahl der Teilnehmer an der Umfrage nimmt seit Jahren kontinuierlich zu.

⁴ Die Situation ist differenzierter, wenn Kunde/Kaufentscheider und Nutzer nicht dieselben Personen sind. Trotzdem bleibt die UX ein wichtiger Faktor und deshalb wird dieser Fall hier vernachlässigt.

Die Forderung umfasst drei Aspekte. Der Aspekt „**früher Zeitpunkt**“ scheint ohne weitere Argumentation naheliegend und plausibel. Wenn die Ergebnisse eines Tests keine Einfluss mehr auf das Produkt haben können, ist der Test überflüssig. Die Forderung, „echte“ **Endnutzer** zu beteiligen, muss dagegen erläutert werden. Die meisten Unternehmen bringen Wissen über die Nutzer und deren Aufgaben auf Umwegen in den Entwicklungsprozess, indem sie Vertreter einsetzen. Häufig sind das Produktmanager seitens der Hersteller, Anforderungsmanager seitens der Kunden, manchmal auch interne Nutzer beim Hersteller mit ähnlichen Aufgaben. Häufig wird vorausgesetzt, dass diese Vertreter selbst Vorerfahrungen als Endnutzer haben und diese Voraussetzung ist auch absolut sinnvoll. Aber sie ist leider nicht hinreichend für eine Garantie, dass die Produktentwicklung sich tatsächlich nah am Nutzer und dessen Bedürfnissen orientiert. Aus mehreren Gründen:

- Ein Nutzervertreter hat selbst einen einzigartigen Blickwinkel. Schon bei der Auswahl des Vertreters ist häufig nicht gewährleistet, dass dieser einen typischen Blick auf die Nutzung des Produkts hat. Häufig nämlich werden besonders versierte Nutzer ausgewählt, denn die besondere Kenntnis des Produkts wird als Vorteil angesehen. Es kann sich aber auch als Nachteil entpuppen. Wer sich besser mit einem Produkt auskennt, nutzt es auf eine andere Art und Weise. Er geht Abkürzungen, verwendet weniger offensichtliche Features, oder kennt mehrere Wege zum Ziel. Deshalb haben solche „Power-User“ nicht immer einen den Blick für die Nöte der Standardnutzer. Sie vertreten immer auch ihre eigenen Interessen, nicht zwangsläufig die Interessen ihrer Kollegen. Ähnliche Unwägbarkeiten ergeben sich eigentlich immer, wenn die Nutzergruppe nicht vollkommen homogen ist. Der Nutzervertreter kann bis zu einem gewissen Grad nicht aus seiner Haut und wird sich immer schwer tun, die Interessen anderer Nutzergruppen adäquat zu vertreten.
- Je länger ein Nutzervertreter seinen Status als Endnutzer aufgegeben hat, umso schwieriger fällt es ihm, die Endnutzer auf hohem Niveau zu vertreten. Selbst wenn er sporadischer Nutzer bleibt, verliert er nach einiger Zeit das Gefühl für die Langzeitnutzung. Und vor allem wird er, je mehr er mit den Entwicklern kommuniziert, deren Blickwinkel auf die Software kennen lernen und immer mehr über technische Hintergründe erfahren. Dadurch verändert sich sein Blickwinkel zunehmend weg von einer Nutzersicht hin zur Herstellersicht.

Es soll hier nicht der Eindruck entstehen, dass ein Nutzervertreter eine sinnlose Rolle im Projekt darstellt. Einzelne Kollegen in dieser Rolle leisten tatsächlich sogar hervorragende Arbeit. Leider hat dies aber mehr mit der Persönlichkeit, dem Engagement und Einfühlungsvermögen der Personen zu tun als mit ihrer Rolle oder ihrer Ausbildung. So

gut wie nie haben diese Vertreter eine Ausbildung, die nutzerzentriertes Testen – also Tests mit Endnutzern – einschließt. Als Hersteller kann man sich letztlich nie darauf verlassen, dass dieser Vertreter erfolgreich arbeitet. Es wäre allerdings möglich dies zu ändern. Indem man **systematische und etablierte Testverfahren** für die User Experience einsetzt. Seit mehreren Jahrzehnten hat die Forschung zur User Experience Methoden entwickelt und deren Effizienz und Effektivität evaluiert und verglichen. Diese Verfahren sind gut dokumentiert und können genutzt werden. Auf diese Möglichkeiten wird später eingegangen.

Sie haben das Problem nicht!?

Das Optimierungspotential in Sachen User Experience ist ganz leicht zu übersehen. Die Gründe hierfür liegen in der Kommunikation innerhalb des Unternehmens und zwischen Unternehmen und Kunde.

Die meisten Leser werden mit der Kostenkurve in Abbildung 1 wohl vertraut sein. Die Erfahrung des Autors aus der Praxis ist: wenn die Probleme nicht erkannt werden, bevor in der Zieltechnologie implementiert wird, dann bleiben sie sehr häufig im Produkt bis es schließlich vom Markt genommen wird. Es ist schlicht zu teuer, einen grundlegenden Fehler in der Bedienphilosophie nachträglich auszubessern. Teilweise werden nachträglich Workarounds rein gestrickt, teilweise werden die Handbücher optimiert. Die Sünden der Vorgängerversionen werden über Generationen weiter getragen. Spillner & Linz (2004) bringen es auf den Punkt: „Wenn im Abnahmetest gravierende Akzeptanzprobleme sichtbar werden, ist es für über Kosmetik hinausgehende Gegenmaßnahmen allerdings oft zu spät.“ (S. 63). Das gilt definitiv auch bereits auf der Stufe der System- und auch der Integrationstests. Zu diesem Zeitpunkt ist der Code zu wesentlichen Teilen bereits geschrieben.

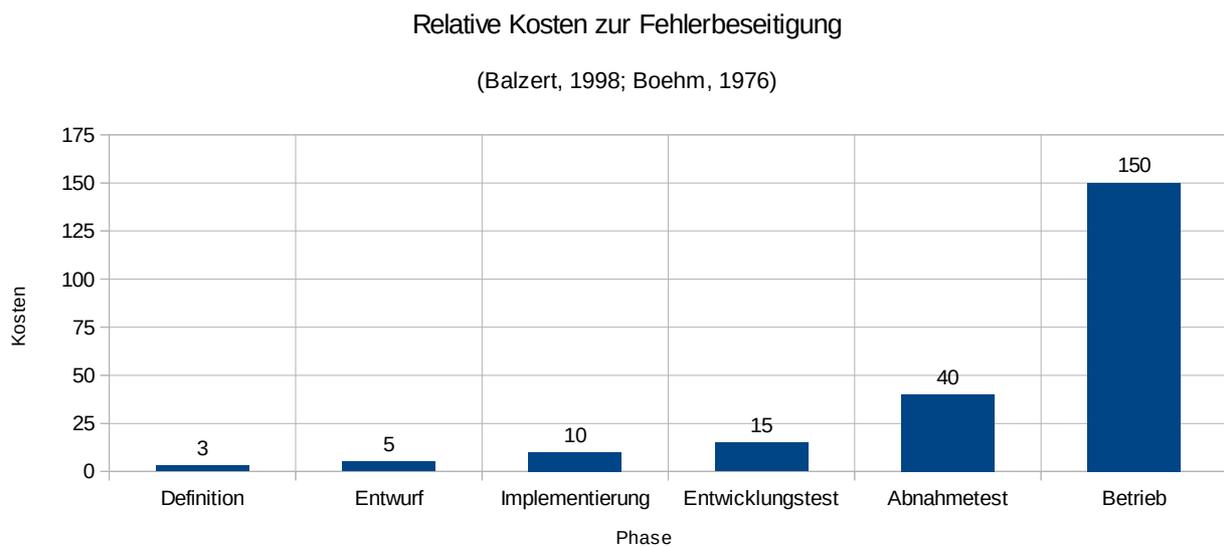


Abbildung 1: Kosten einer Änderung am Produkt abhängig vom Fortschritt des Projekts

Wenn solche Probleme aber nie behoben werden, dann versickern sie allzu leicht in der unternehmensinternen Kommunikation. Nur kleine Kreise im Support oder im Entwicklungsteam selbst bemerken die Problematik überhaupt. Hier entstehen keine zusätzlichen Kosten. Aus der Sicht des Managements ist daher alles in Ordnung und es setzt neue Ziele für das nächste Release. Die Optimierung der User Experience gehört nicht dazu.

Das lässt sich nur verhindern, indem User Experience systematisch getestet wird. Denn von allein kommen Nutzer selten und beschwerten sich über die schlechte Qualität. Mängel werden moniert, wenn das Produkt technisch nicht ausgereift ist. In diesem Fall ist klar, dass der Hersteller seiner Verantwortung nicht nach kommt und er muss nachbessern. Wenn es aber um die Einfachheit der Bedienung geht, ist die Verantwortlichkeit schon weit weniger eindeutig. Viele Nutzer schreiben es den eigenen Fähigkeiten zu, wenn sie etwas nicht bedienen können. Bevor sie sich beim Hersteller beklagen, müssen Nutzer schon sehr leiden. Selbst wenn der Hersteller aktiv auf Nutzer zugeht, zeigen diese eine spürbare Hemmschwelle, offen Kritik zu äußern (Tohidi, Buxton, Baecker & Sellen, 2006). Noch extremer dürfte es sein, wenn andere Faktoren von User Experience ins Spiel kommen. Muss es zum Beispiel wirklich sein, dass ein Produkt so schwer zu lernen ist? Gerade bei Expertensystemen hört man gerne, dass diese komplex seien und sie müssten geradezu schwer erlernbar sein. Das stimmt aber nur zu einem gewissen Grad. Die Erfahrung aus vielen Projekten zeigt da ist häufig noch viel Spielraum. Haben Ihre Nutzer gar eine positive Einstellung zu Ihren Produkten? Sind sie damit zufrieden oder macht es ihnen sogar Spaß? Der Nutzer wird schlicht zu dem Schluss kommen, dass er sich bei der Wahl seines Produkts vertan hat und es zurück schicken oder zum Elektroschrott bringen.

Im Bereich B2B (Business-To-Business Produkte) ist die Situation spezieller. Dort besteht häufig kein direkter Kontakt mit Nutzern. Allenfalls steht ein „Key User“ oder „Power User“ zur Verfügung, der einen Teil seiner Zeit in die Kommunikation mit dem Hersteller investiert. In vielen Fällen sind die Ansprechpartner auf Kundenseite aber keine Nutzer mehr. Der Kunde muss also die Informationen bei den Endnutzern einsammeln und der Hersteller muss diese Information abholen. Von den üblichen Stille-Post Effekten abgesehen, auf beiden Seiten sitzen zwar Experten für die Technik oder für die Fachlichkeit, aber selten sind diese geschult in Techniken zur Gesprächsführung oder Interviews. Für eine systematische Informationsgewinnung ist das aber unabdingbar. So laufen die Gespräche nicht selten eher unstrukturiert. Auch wenn sie sehr interessant sein können, sie sind leider kein Ersatz für einen systematischen Test.

Doch auch wenn der Hersteller all die oben genannten Hürden überwinden könnte, käme er nicht zwangsläufig an die nötigen Informationen. Denn das setzt voraus, dass die Ansprechpartner beim Kunden entsprechend gut informiert sind. Die Situation beim Kunden sieht folgendermaßen aus.

Erstens sind die Endnutzer keine systematischen Tester. Sie testen ihre Tätigkeiten nicht planvoll durch; manche Probleme registrieren sie; sie schreiben ihre Erkenntnisse nicht auf; manches vergessen sie, wenn es zu einem Gespräch darüber kommt. Vieles ist für sie so selbstverständlich, dass sie gegenüber einem Hersteller oder auch einem internen Nutzervertreter nie ein Wort darüber verlieren würden.

Zweitens wird über unzureichende User Experience zwar häufig geschimpft. Aber die Probleme werden deshalb nicht zwangsläufig den Verantwortlichen berichtet. Arbeitnehmer sind bereits zurückhaltend, wenn es darum geht, sich wegen ergonomischer Mängel am Arbeitsplatz beschweren (Fellowes, 2010). Man kann nur vermuten, dass diese Zurückhaltung eher steigt als sinkt, wenn es um Dinge wie „intuitive Gestaltung“ oder andere Aspekte einer positiven User Experience geht. Denn in punkto Ergonomie ist gesetzlich geregelt, welche Rechte ein Arbeitnehmer hat. Aber welche Ansprüche kann er in Sachen User Experience geltend machen? Folglich ist die Kontaktperson beim Kunden (Einkäufer, Anforderungsmanager, etc.) über Schwächen in der User Experience unzureichend informiert und kann den Herstellern darüber keine Auskunft geben.

Das Fazit ist: wenn die Qualität der User Experience nicht systematisch getestet wird, werden Hersteller die Schwächen in vielen Fällen erst dann wahrnehmen, wenn sie bereits ein kritisches Niveau erreicht haben. Und solange nicht regelmäßig und zum richtigen Zeitpunkt getestet wird, können die notwendigen Korrekturen ein Projektbudget schnell sprengen.

Ist Agilität eine Lösung?

Scrum und viele andere agile Methoden weisen einige Ähnlichkeiten zur nutzerzentrierten Gestaltung auf. Beide sind anforderungsgetrieben, iterativ und feedbackorientiert und basieren auf der Einsicht, dass effiziente Kommunikation im Projekt entscheidender ist als Dokumente oder Prozesse. Agile Methoden planen Kundenkontakte in ihre Arbeit ein; User Stories werden aus Nutzersicht geschrieben; regelmäßige Reviews mit Stakeholdern finden statt. Das Ganze ergibt sich zwangsläufig aus dem agile manifesto. Dort heißt es: „Customer collaboration over contract negotiation“ (Beck et al., 2001). Aber die Probleme im Alltag sind häufig dieselben. Meistens wird mit Nutzer-Vertretern geredet, anstatt Endnutzer zu beobachten. Die meisten der Probleme, die im vorigen Abschnitt aufgezählt wurden, werden damit nicht gelöst: Stille-Post, Gedächtniseffekte, unsystematisches Testen, unvollständige Informationen bei den Nutzervertretern, das Zurückhalten von Rückmeldungen bei suboptimaler User Experience, usw. Es ist ein guter Anfang, wenn man einen engen Kontakt zum Kunden sucht. Aber das ist noch keine hinreichende Bedingung.

Dass es einen erheblichen Bedarf gibt, beide Ansätze miteinander zu vereinbaren, das belegen drei Übersichtsarbeiten (Da Silva et al., 2011; Salah, Paige & Cairns, 2014; Sohaib & Khan, 2010). Insgesamt werten die Autoren Publikationen im dreistelligen Bereich aus, die sich alle mit der von Integration agiler Softwareentwicklung und der nutzerzentrierten Gestaltung auseinander setzen. Durch agile Methoden werden die Methoden der nutzerzentrierten Gestaltung nicht überflüssig. Die große Anzahl an Publikationen ist allerdings auch ein Indiz dafür, dass es vermutlich keine einfache Antwort auf die Frage der Integration gibt.

Im Bezug auf das Testing mit Endnutzern werden vor allem abgespeckte Testverfahren empfohlen (Da Silva et al., 2011). Ein herkömmlicher Usability Test mit Planung, Vorbereitung, Durchführung und Auswertung ist nur schwer innerhalb kurzer Sprints durchführbar, insbesondere da nie genau bestimmt werden kann, wann welche Funktionalität zur Verfügung steht (Salah, Paige & Cairns, 2014). Entsprechend berichten nicht einmal die Hälfte der Projekte, dass sie mit Endnutzern testen (Da Silva et al., 2011) oder sie greifen auf Tests mit Kollegen zurück (Salah, Paige & Cairns, 2014), was aber methodisch riskant ist. Salah, Paige & Cairns (2014) geben Empfehlungen, wie man diese Schwierigkeiten umgehen kann. Beendet ist die Diskussion damit allerdings noch nicht.

Tests für User Experience?

Das Wissen, das benötigt wird, um einen iterativen Gestaltungsprozess für das User Interface zu verwirklichen, ist längst vorhanden. Auch die Erfahrungen sind da, wie man diese Vorgehensweisen an die Bedingungen bei verschiedenen Herstellern anpasst. Die

Testverfahren, die benötigt werden, um die Reaktionen von Nutzern auf ein System zu identifizieren, sind seit langem etabliert. Aber nur wenige Unternehmen praktizieren diese Tests. Bestenfalls werden regelmäßig Usability Tests durchgeführt. Und auch das trifft aktuell noch auf die Minderheit der Unternehmen zu. Im Großen und Ganzen ist die User Experience in der Softwareentwicklung immer noch massiv unterrepräsentiert.

Unter die nutzerzentrierten Systemtests lassen sich diese Verfahren zählen:

- Usability Tests: durch einen Moderator angeleiteter Test, der in der Regel auf die Bedienbarkeit abzielt.
- Accessibility testing: teilweise kann Barrierefreiheit durch Heuristiken, Checklisten und sogar automatisiert erfolgen. Einige Aspekte erfordern, dass man eine Variante eines Usability Tests mit Endnutzern durchführt, der speziell auf Barrierefreiheit abzielt.
- User Experience Evaluationen: die User Experience Evaluation geht über den Usability Test hinaus und erfasst neben der Bedienbarkeit auch emotionale, ästhetische, nützlichkeits- oder bedürfnisrelevante Aspekte. Teilweise werden Usability Tests durch Fragebögen oder physiologische Messungen ergänzt; teilweise werden aktuell auch spezielle Testverfahren erprobt (z.B. die Valenzmethode, siehe Burmester et al., 2010).
- Attraktivitäts- und Designevaluation zielen auf die Wirkung, die die äußere Erscheinung des Produkts bei Nutzern (oder anderen Zielgruppen) hinterlässt. Wird üblicherweise als Fragebogenstudie durchgeführt.

Vergleichbar mit den Teststufen (Code review, Komponententest, Integrationstest, etc.) lassen sich auch in der User Experience verschiedene Stufen definieren. Jede Stufe erfordert eine höhere Reife (fidelity) des getesteten Prototypen. Mit jeder Stufe kann man mehrere Aspekte des Systems testen.

UX Teststufe	Erläuterung	Software-Test zum Vergleich
Konzeptreview	Erfasst alle Schwächen im System, die durch die bloße Inspektion identifiziert werden können. Inkonsistente Abläufe und Masken, Layouts etc.	Code Review
Mockup Tests	Bewertet den Ablauf selektiver Nutzeraufgaben mit besonderem Fokus auf Aufgabenabläufe, Informationsverteilung, Verständlichkeit, etc.	Unit Test
Klickdummy Tests	Aufforderungscharakter, Lernbarkeit	-
Interaktiver Prototyp	Auf dieser Ebene können Aspekte untersucht werden wie Feedback (des Systems an den Nutzer), Transitionen/Animationen, Zeitverhalten, etc.	Integrationstest
Validierungsstudie	Ein abschließender (sogenannter summativer) Test, der Abnahmetest prüft, ob die gesetzten Ziele erreicht werden konnten.	

Und der Mehrwert?

50% des Gesamtbudgets eines Softwareprojekts gehen heutzutage auf das Konto von Tests (Sneed, Baumgartner, & Seidl, 2008, nach Sneed & Jungmayr, 2011). Spontan möchte man angesichts dieser Zahlen meinen, dass es für kein Projekt ein Problem darstellen sollte, in irgendeiner Form auch die Interaktion mit dem Produkt zu testen. Wenn bei einer Neuentwicklung die Entwicklungskosten zum Beispiel 500.000 EUR betragen, sollten sich doch irgendwo 10.000 EUR (4% des Testbudgets) für wenigstens einen einzelnen Usability Test organisieren lassen.

Die genauere Betrachtung liefert zwei Aspekte, aus der ein Return-On-Investment entsteht:

1. Die Reduktion von Fehlerkosten. Das beinhaltet ...
 - a) Produktionsstörungskosten: Kosten, die dem Nutzer/Kunden entstehen, weil das Produkt suboptimal ist.
 - b) Fehlerbehebungskosten: die Kosten, die dem Hersteller durch die Korrektur des Systems entstehen. Darin enthalten: Fehler untersuchen, Ursache finden, korrigieren, Korrektur testen; erneutes Rollout.
2. Die Erhöhung der Qualität und damit der Marktchancen des Produkts.

Reduktion von Wartungskosten. 40% der Fehler einer Software sind Anforderungsfehler. Sie sind die häufigste Fehlerart. Zudem sind es die teuersten Fehler (Sneed & Jungmayr,

2011), wenn es um deren Korrektur geht. Da die meisten Mängel in Sachen Usability / User Experience auf die Anforderungen zurückgeführt werden können, sollte sich ein Test leicht bezahlt machen. Hierzu ein Gedankenexperiment:

Ein Hersteller macht in einer frühen Phase einen Test mit einem Klickdummy. Das ist ein einfacher Prototyp ohne wirkliche Funktionalität, der nur so tut als ob. Zu den Kosten für den Test gehört das Erstellen des Prototyps (3000 EUR) und die Testdurchführung plus Auswertung (7.500 EUR)⁵. Gehen wir davon aus, dass wir 5 schwerwiegende Probleme identifizieren können und 20 leichtere. Diese müssen korrigiert werden, was sich lediglich auf die „Papierzeichnungen“ auswirkt, die die Bedienoberfläche beschreiben. Schätzen wir hierfür noch einmal Aufwand im Wert von 4000 EUR. Nun haben Sneed & Jungmayr (2011) verschiedene Studien zusammengestellt, die die Kosten der Fehlerkorrektur untersucht haben. Die Zahlen erlauben die Schätzung, dass es etwa 30 mal so viel kosten würde, wenn diese Probleme erst nach dem Rollout entdeckt würden. 30 mal 4000 EUR ergeben 120.000 EUR.

Die Kosten für den Prototyp und den Test werden also vielfach aufgewogen. Auch wenn die Zahlen grob geschätzt sind, sie sind durchaus realistisch. Wie diese Zahlen dann im konkreten Fall aussehen, ist von vielen Faktoren abhängig. Zum Beispiel der Qualität der Anforderungserhebung, der Frage, wie die Nutzeranforderungen für das Team dokumentiert wurden, wie viel Erfahrung das Team in der Gestaltung von hochwertigen User Interfaces hat, der Komplexität der Anwendung, der Anzahl unterschiedlicher Nutzergruppen, und so weiter.

Erhöhung der Qualität. Wie bereits festgestellt wurde, wenn es am Ende des Projekts Akzeptanzprobleme bei den Nutzern gibt, ist es für grundlegende Änderungen am User Interface bereits zu spät. Vorausgesetzt diese Probleme werden überhaupt bemerkt. Das bedeutet aber natürlich, dass viele Probleme in der Bedienoberfläche beim Hersteller keine Folgekosten nach sich ziehen. Und es bedeutet, dass ein Teil des Mehrwerts von nutzerzentrierten Tests sich nicht dadurch ergibt, dass die Wartungskosten sinken.

Schade ist, dass sich das nutzerzentrierte Testen gerade in dieser Kategorie häufig auszeichnet. Denn das Testen mit Nutzern hilft nicht nur, die bestehenden Anforderungen zu hinterfragen und zu präzisieren. Häufig entstehen auch neue Anforderungen oder bestehende werden als weniger wichtig identifiziert. Denn die anfängliche Sammlung der Anforderungen ist sehr abstrakt. Erst wenn man den Nutzern die Entwürfe zeigt und mit ihnen konkrete Bedienabläufe durchspielt, wird das zukünftige System so konkret und erfahrbar, dass konstruktive Rückmeldungen meistens überhaupt erst realistisch werden. Kein Mensch hat so viel Vorstellungskraft, dass er anhand einer Liste von Features wirklich prüfen kann, ob die Bedienung leicht

⁵ Der Einfachheit halber gehen wir hier davon aus, dass die Tätigkeiten von einem Dienstleister durchgeführt werden mit einem Tagessatz von 1000 EUR pro Tag. Der Test selbst berücksichtigt außerdem Rekrutierung und Aufwandsentschädigung für die Testteilnehmer.

von der Hand gehen oder gar Spaß machen könnte. Selbst einzelne, statische Screenshots helfen da nicht weiter. Ein Kunde des Autors hatte da leidvolle Erfahrungen gemacht. Er hatte einzelne Screenshots in vielen Meetings mit Nutzern diskutiert. Als wir drei Monate später die Zwischenschritte ausgearbeitet haben und ganze Bedienabläufe förmlich „gemalt“ haben, wurde aber sofort klar, dass die Idee in sich unstimmig und gar unlogisch ist. Nicht nur der Produktmanager hatte das schlicht übersehen, auch keinem der Nutzer war das aufgefallen. Wenn man Bedienabläufe Schritt für Schritt ausarbeitet und Nutzer damit konfrontiert, kommt ungenutztes Potential zum Vorschein und die Schwächen springen ins Auge. Das verschafft einen erheblichen Vorteil gegenüber der Konkurrenz. Systematisches nutzerzentriertes Testen verbessert die Qualität der Produkte in zweifacher Hinsicht: sie präzisiert bestehende Anforderungen und eröffnet ganz neue bisher unbeachtete Chancen.

Zusammenfassung & Ausblick

Unter User Experience Experten wird Testen hoch angesehen. Das gilt entsprechend für Unternehmen, in denen diese Themen bereits eine grundlegende Rolle spielen. Das deckt sich unter anderem mit Erfahrungen von Nielsen (2006), dass Usability Testing häufig einer der ersten Schritte ist, den die Unternehmen auf ihrem Weg zum nutzerzentrierten Arbeiten machen. Testen weckt häufig erst das Bewusstsein für dieses Thema und kann die Entwicklung anschließend konstruktiv unterstützen.

Im Web ist Testing heutzutage schon vergleichsweise geläufig. Das hat verschiedene Ursachen. Eine davon mag sein, dass man im Web nur Nutzeraktionen mittels Web Analytics aufzeichnen muss, um an echte Nutzerdaten heran zu kommen. Aber ein viel wichtigerer Grund dürfte sein, dass der Konkurrenzkampf im Web vergleichbar hoch ist. Die Nutzer sind dort vollkommen ungebunden. Das nächste Angebot ist nur einen Klick entfernt. Der harte Konkurrenzkampf erhöht den Druck auf die Anbieter. Wer seinen Nutzern nicht genau „auf die Finger schaut“, der hält sich nicht lange.

Das geht mit nutzerzentriertem Arbeiten. Es erhöht die Qualität des Produkts und steigert den Erfolg. Indem Nutzer in den Entwicklungsprozess eingebunden werden erhält fundierte Rückmeldungen bereits vor Beta-Phase oder der Markteinführung. Risiken werden so effektiv eingedämmt und der Erfolg ist langfristig planbarer.

Es gibt in der User Experience zahlreiche Testverfahren. Sie sind unter hohem Kostenbewusstsein entwickelt worden, denn nutzerzentriertes Arbeiten wird leicht als zusätzlicher Aufwand und Mehrkosten missverstanden. Die Methoden sind schlank und trotzdem aussagekräftig.

Nutzer müssen bereit sein an den Studien teilzunehmen. Vor allem bei B2B-Produkten hat das wichtige Implikationen. Dort müssen auch die Vorgesetzten der Nutzer einer

Studienteilnahme zustimmen. Das erfordert ein gutes Verhältnis zu den Kunden. Führungskräfte beim Kunden müssen Einsicht in den Mehrwert haben, wenn sie ihre Mitarbeiter für Tests frei stellen sollen. Das muss langfristig geplant und strategisch angegangen werden. Ergebnisse von Nutzerstudien sollten veröffentlicht werden, damit die Teilnehmer erfahren, dass ihr Einsatz sich auszahlt. Darüber hinaus müssen Erkenntnisse aus Tests möglichst zeitnah in die Produkte einfließen.

Die andere Möglichkeit wäre es, Studien mit Nutzern auf neuralgische Punkte zu beschränken und zusätzlich mit „nutzerfernen“ Methoden zu arbeiten. Eine Möglichkeit sind Expertengutachten (die ähnlich ablaufen wie die oben angesprochenen Konzeptreviews).

Eine andere Möglichkeit wäre ein höherer Grad an Automatisierung. Aktuell lassen UX Tests sich schwer automatisieren. Eine ähnliche Vorgehensweise kennen wir aus dem World Wide Web. Dort werden Interaktionen der Nutzer routinemäßig protokolliert und ausgewertet. Viele der Maße geben Hinweise auf Probleme seitens der Nutzer. Diesen Hinweisen kann man später mit anderen Methoden gezielt nachgehen und spart so Aufwände. Viele Metriken aus den Web Analytics ließen sich auch auf Desktop-Software oder Mobile Apps übertragen. Um solche Methoden im größeren Maßstab einsetzen zu können muss vorher die Frage der Akzeptanz geklärt werden; eine wichtige Rolle dürften dabei Datenschutz und Betriebsräte spielen. Und schließlich können solche Ansätze nur Hinweise auf Probleme bieten, ohne sie aber zu erklären, was den Einsatz anderer Methoden unumgänglich macht (Sauro, 2010).

Eine andere Option zur Automatisierung bieten Tools, die psychologische Modelle zur Vorhersage von Nutzerverhalten einsetzen⁶. Solche Tools können zwar nur einen begrenzten Teil menschlichen Verhaltens abdecken. Doch trotz dieser Einschränkungen ist es bereits in frühen Entwicklungsphasen möglich, wichtige Erkenntnisse über die Interaktion zwischen Produkt und Nutzer zu gewinnen. Man kann mehrere Entwürfe vergleichen oder sich noch am Reißbrett mit der Konkurrenz messen.

Insgesamt können wir festhalten, dass es bereits zahlreiche etablierte Methoden gibt, um den Nutzer stärker in die Produktentwicklung zu integrieren und somit Fehlentwicklungen frühzeitig zu vermeiden beziehungsweise auch Chancen für das Produkt zu erkennen. Sie müssen eigentlich nur genutzt werden.

Nachwort des Autors

Diese Arbeit enthält an einigen Stellen Aussagen, die ich gerne mit empirischen Arbeiten belegt hätte. Da nach meinem Kenntnisstand hierzu allerdings keine Studien

⁶ Ein prominentes Beispiel wäre [CogTool](http://cogtool.hcii.cs.cmu.edu/) (<http://cogtool.hcii.cs.cmu.edu/>) von der Carnegie Mellon University.

vorliegen, musste ich das durch meine persönliche Erfahrung kompensieren, die ich als Software Dienstleister in fast einem Jahrzehnt und 50 Projekten in der Branche gesammelt habe.

Eine weitere Quelle über den Fortschritt nutzerzentrierten Arbeitens in der Software-Entwicklung ist der Bericht zur Usability Maturity durch die Human Factors International. Dort antworteten 1123 Teilnehmer einem Aufruf, Angaben zur Usability Maturity ihres Unternehmens zu machen. Obwohl fast 50% der Teilnehmer aus dem Web-Bereich kamen und das Thema Usability dort schon eine sehr viel größere Rolle spielen als in anderen Branchen, gab es nur 23% Bejahungen auf die Frage, ob das Unternehmen eine standardisierte Vorgehensweise verfolgt im Bezug auf nutzerzentriertes Testing/Design. Insgesamt scheint das Interesse am Thema allerdings zu steigen, was man den Branchenreports der German UPA entnehmen kann. Die Anzahl der Teilnehmer am Branchenreport hat sich in den letzten 10 Jahren mehr als verdreifacht. Eine allgemeine Tendenz hin zu mehr nutzerzentriertem Arbeiten lässt sich also noch belegen. Bei Detailaussagen ist das weitaus schwieriger. Für sachdienliche Hinweise bin ich jederzeit dankbar.

Quellen

- Balzert, H. (1998) Lehrbuch der Software-Technik. Bd.I. Software-Entwicklung. Heidelberg: Spektrum Akademischer Verlag
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for Agile Software Development Manifesto for Agile Software Development.
- Boehm, B. W. (1981) Software Engineering Economics; Prentice-Hall Advances in Computing Science & Technology Series
- Burmester, M., Mast, M., Jäger, K. & Homans, H. (2010) Valence Method for Formative Evaluation of User Experience. in: Halskov, K.; Graves Petersen, M.G. (eds.): DIS2010 Proceedings, 16.-20. Aug. Arhus, Denmark 2010. New York: ACM, pp. 364-367.
- da Silva, T. S., Martin, A., Maurer, F. & Silveira, M. (2011) User-Centered Design and Agile Methods: A Systematic Review. In Agile Conference (AGILE); S. 77-86
- Fellowes (2010) [Ergonomie an deutschen Arbeitsplätzen](http://fellowes.fleishmaneuropa.de). Marktstudie durchgeführt von Dynamic Markets Ltd., <http://fellowes.fleishmaneuropa.de>, abgerufen am 04. August 2013
- ISO9241-11 (1998) Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability

- Nielsen, J. (2006) Corporate Usability Maturity: Stages 1-4; Jakob Nielsen's Alertbox; Zugriff vom 2. Mai 2013
- Salah, D., Paige, R., Cairns, P. (2014) A systematic literature review on agile development processes and user centred design integration. 18th Int. Conf. on Evaluation and Assessment in Software Engineering, EASE
- Sauro, J. (2010). What five users can tell you that 5000 cannot. Measuring Usability, abgerufen am 22.08.2014, <http://www.measuringusability.com/analytic-testing.php>
- Sneed, H. M., Baumgartner, M. & Seidl R. (2008) Der Systemtest – Requirements-based Testing. Hanser, München
- Sneed, H. M. & Jungmayr, S. (2011). Mehr Testwirtschaftlichkeit durch Value-Driven-Testing; Informatik-Spektrum, 34 (2), S. 192-209
- Sohaib, O. & Khan, K. (2010). Integrating Usability Engineering and Agile Software Development: A Literature Review. In: Computer Design and Applications (ICCCA), 2010 International Conference on, Volume 2, S. 32-38
- Spillner, A. & Linz, T. (2004) Basiswissen Softwaretest. Heidelberg: Dpunkt Verlag
- Straub, K., Patel, M., Bublitz, A. & Broch, J. (2009) HFI Usability Maturity Survey Findings – 2009. <http://de.slideshare.net/kstraub/hfi-usability-practice-maturity-model>; Zugriff vom 22. August 2013
- Tohidi, M., Buxton, W., Baecker, R. & Sellen, A. (2006) Getting the Right Design and the Design Right: Testing Many Is Better Than One. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, S. 1243-1252
- Usability Professionals Association – German Chapter (2003-2012) Branchenreport 2003-2013; <http://www.germanupa.de/berufsverband/branchenreport/>; Zugriff vom 22. August 2013

Über dieses Dokument

Titel: Der Nutzer – ein unbekanntes Wesen für Software-Tester?

Autor: Jan Seifert

Sprache: Deutsch

Version vom 07. Dezember 2014

Schlagnote: Softwaretest, Systemtest, Akzeptanztest; User Research, Usability Test, User Experience Evaluation; Empirie, Empirik, Marktforschung

Weitergabe: <http://creativecommons.org/licenses/by-nd/3.0/de/>

Die Inhalte in diesem Dokument spiegeln ausschließlich die persönlichen Ansichten des Autors wider.

