

„Das ist ein gewachsenes Produkt“ – User Interface Balkonitis

Wie man Usability-Probleme von betrieblich genutzten
Anwendungen nachhaltig verbessern kann

Guido Tesch
DEVK Versicherungen
Riehler Straße 190
50735 Köln
guido.tesch@devk.de

Abstract

In der Realität von betrieblich eingesetzter Software ist der Ausspruch „Das ist ein gewachsenes Produkt“ nur allzu bekannt. Da werden oftmals Anwendungen produktiv genutzt, die eine deutlich verminderte Usability aufweisen. Und dennoch erscheinen die Kräfte, die eine Verbesserung in diesem Bereich verhindern, unüberwindbar, auch wenn es betriebswirtschaftlich sinnvoll wäre. Dieser Erfahrungsbericht bietet einen neuen Blickwinkel auf dieses Phänomen, hilft, die Zusammenhänge zu verstehen und Usability-Verbesserungen in betrieblich eingesetzter Software nachhaltig zu erreichen. Der Autor definiert den Begriff „User Interface Balkonitis“, zeigt die häufigsten Ursachen hierfür auf und gibt Hinweise aus der Praxis, was man sinnvoll dagegen tun kann.

Keywords:

/// User Interface Balkonitis
/// Betrieblich genutzte Software
/// Nachhaltige Usability-
Verbesserungen
/// Anwendungsarchitektur
/// Integrationsarchitektur
/// Featuritis

1. Einführung

Wer als User Interface Designer, Interaction Designer, Usability-Experte oder System-Ergonomie-Architekt an der Gestaltung betrieblich genutzter Anwendungen arbeitet, begegnet dort manchmal User Interfaces, die eine überraschend geringe Gebrauchstauglichkeit (Usability) aufweisen. Dennoch werden die Anwendungen eingesetzt, zum Teil sogar in Kernprozessen des Unternehmens, ohne dass etwas an der Usability geändert wird. Die Ursachen hierfür sind schwer zu erkennen und vielfältig, eine Spielart davon wird in diesem Beitrag beschrieben und als „User Interface Balkonitis“ charakterisiert.

Die User Interface Balkonitis ist nicht spezifisch für ein bestimmtes Unternehmen. Überall, wo die Komplexität des betrieblich genutzten IT-Systems eine gewisse Komplexität erreicht, kann User Interface Balkonitis auftreten.

Es handelt sich bei dem Beitrag um einen Erfahrungsbericht, eine wissenschaftliche Analyse der Ursachen und Nachweise der Wirksamkeit von Maßnahmen bleiben einer zukünftigen Arbeit vorbehalten.

Es ist wichtig festzuhalten, dass es in diesem Beitrag um die Entwicklung **betrieblich** genutzter Anwendungen geht. Eine fundamentale Eigenschaft hiervon ist, dass diejenigen, die diese Anwendungen nutzen dies tun, weil die Geschäftsprozesse des Unternehmens es so vorsehen und nicht, weil sie die Anwendungen für ihre Tätigkeiten selbst gewählt haben. Die Benutzer der betrieblichen Anwendungen haben in der Regel nur einen geringen Einfluss auf die Auswahl sowie die Gestaltung der Anwendungen, die sie tagtäglich gebrauchen. In dieser Hinsicht unterscheidet sich die Betrachtung betrieblich genutzter Anwendungen erheblich von der Betrachtung von Produkten für Endkunden oder für Software-gestützte Consumer-Services.

2. Der Begriff „User Interface Balkonitis“

Der Begriff „User Interface Balkonitis“ (kurz UI Balkonitis) lehnt sich an Begrifflichkeiten in der Medizin an, wobei als „System“ nicht ein Organismus, sondern ein Unternehmen mit einem Informationssystem betrachtet wird. Den Organen entsprechen sowohl IT-Systemkomponenten als auch

organisatorische Einheiten mit ihren Entscheidungsträgern sowie Mitarbeitern.

Die Endung **„-itis“** bezeichnet meist eine entzündliche Krankheit. Am Anfang des Begriffs wird das betroffene Organ/die betroffene Komponente des Unternehmens genannt. Eine Entzündung ist eine charakteristische Antwort von Gewebe / Komponenten auf einen potenziell schädigenden Reiz mit dem Ziel, den Reiz zu beseitigen.

Als **„Balkon“** einer Software-Anwendung wird ein funktionaler Teil bezeichnet, der oft (aber nicht nur) durch „historisches Wachstum“ der Anwendung entsteht. Der Begriff wird in Anlehnung an die Architektur von Gebäuden verwendet, wobei dies nur ein Beispiel (wenn auch ein typisches) ist, wo der zu beschreibende Sachverhalt auftritt. Andere Beispiele wären, wenn ein Raum eines Gebäudes mit Funktionen überfrachtet und dadurch unübersichtlich wird oder wenn die Anordnung von Räumen zu langen oder umständlichen Laufwegen für typische Nutzungsszenarien führt.

„User Interface Balkonitis“ ist die Reaktion des Unternehmens (als Analogon zu dem

Begriff Organismus betrachtet) auf einen schädigenden Reiz. An den „Balkonen“ der IT-Systeme treten Usability-Probleme in der Mensch-Maschine-Kommunikation / im User Interface auf (der schädigende Reiz). Alternativ können auch durch das Zusammenspiel mehrerer Anwendungen Usability-Probleme auftreten, wo eine Anwendung als „Balkon“ einer anderen Anwendung betrachtet wird oder innerhalb einer funktional überfrachteten Anwendung. Die charakterisierende Antwort des Unternehmens besteht in der Erwartung, dass die Belastung gering sei und von den Benutzern – den Mitarbeitern – getragen werden könne. Dadurch spürt das Unternehmen als Ganzes den Reiz oft nicht mehr, obwohl er fortbesteht. In dem Sinne ist UI Balkonitis in der Regel chronisch, die Komponente „Mitarbeiter“ wird dauerhaft belastet, was unter anderem einen erhöhten Aufwand für die Bearbeitung von Geschäftsprozessen bedeutet.

Wie hoch der Aufwand tatsächlich ist, mit dem die Reizbewältigung im Unternehmen betrieben wird, bleibt meist unbekannt. Pro Benutzung eines „Balkons“ ist der Aufwand eher niedrig, kann in der Summe aber substantiell sein. Weil die Benutzer der Anwendungen die Belastung tragen und sie dies als ihre Aufgabe ansehen, ist die Belastung für das Unternehmen als Ganzes vielfach nicht mehr spürbar, was das Erkennen der kausalen Zusammenhänge erschwert. **Die Ungewissheit darüber, wie hoch der Aufwand wäre, um den Reiz zu beseitigen, ist das Hauptproblem, das die Auflösung der UI Balkonitis verhindert.** Darauf wird bei der Diskussion möglicher Maßnahmen eingegangen.

3. Klarstellungen und Abgrenzungen

Der Fokus vieler Unternehmen bei der Weiterentwicklung ihres IT-Systems liegt auf zentralen funktionalen Aspekten der Anwendungen, auf der Gestaltung von Geschäftsprozessen und der Verwendung der Anwendungen in diesen Geschäftsprozessen. Dies ist grundsätzlich sinnvoll, da hier in der Regel mit einem höheren Gewinn in der Effektivität und Effizienz

der Geschäftsprozesse gerechnet werden kann, wodurch wiederum eine Rechtfertigung von entstehenden Kosten für die Weiterentwicklung leichter fällt. Bei der UI Balkonitis geht es um die Fälle, in denen Usability-Probleme in Bezug auf nichtfunktionale Aspekte auftreten und in der Regel nicht die Effektivität betreffen, sondern die Effizienz und die Nutzerzufriedenheit. Wegen des Problems der Ungewissheit über die Höhe der Kosten bleiben diese Verbesserungspotenziale oft ungenutzt.

„Balkone“ in Software-Anwendungen sind nicht grundsätzlich schlecht. Es kann sinnvoll sein, Funktionen anzubieten, indem man „am Rand des Gebäudes“ etwas ergänzt oder bestehende Räume erweitert. Die Alternative wäre, das Gebäude im Inneren umzustrukturieren oder Außenmauern zu versetzen, damit im Inneren mehr Platz entsteht –sprich: Die zusätzliche Funktionalität in die bestehenden Bildschirmabläufe der Anwendung sauber zu integrieren. Ein Balkon kann durchaus Sinn machen, wenn die zusätzliche Funktionalität nur gebraucht wird im Sinne eines „kurz hin, erledigen und zurück“. Allerdings wird typischerweise der „Aufwand im Kleinen“ bei der Benutzung unsinniger „Balkone“ wesentlich geringer eingeschätzt als der Aufwand, den „Balkon“ gar nicht erst entstehen zu lassen und die anzubietende Funktionalität besser zu integrieren.

Erfahrungsgemäß liegen die Ursachen einer UI Balkonitis nicht in Problemen der Geschäftsprozesse, also an ineffizientem Arbeiten wegen der Festlegung der abzuarbeitenden Arbeitsschritte. Hier sind Prozess- und Organisationsmanagement sowie die Facharchitektur gefragt, die Basis zu schaffen, damit die eingesetzten Anwendungen grundsätzlich zu den abzuarbeitenden Arbeitsschritten passen. Die UI Balkonitis bezieht sich auf die Usability der Interaktionsprozesse in den einzelnen Anwendungen sowie auf deren Zusammenspiel.

Der Begriff „Featuritis“ ist verwandt zur UI Balkonitis, sollte aber nicht damit verwechselt werden. Während bei Featuritis das Vorhandensein von zu vielen Funktionen

fokussiert wird, steht bei der UI Balkonitis die Qualität des UI einer einzelnen Funktion sowie deren Integration in das Gesamtsystem im Vordergrund.

4. Typische Ursachen für UI Balkonitis

4.1. Inkorrekte oder extrem unvollständige Anforderungen

Wenn die Anforderungen, die von der Anwendung erfüllt werden sollen, extrem unvollständig bekannt oder inkorrekt erfasst sind, hat dies erfahrungsgemäß folgende Konsequenzen:

- Es gibt keine klare oder eine falsche Zielsetzung für die Entwicklung der Anwendung, besonders bei der Gestaltung der Interaktion für neue Funktionen.
- Wegen der Vielzahl an Gestaltungsmöglichkeiten wird mit gewisser Wahrscheinlichkeit in eine Richtung investiert, bei der die Integration der neuen Funktion zu Usability-Problemen führt.
- Die Investitionen führen zu einer „Richtungsträgheit“: Um eine andere Richtung einzuschlagen, muss ein (oft sehr großer) Widerstand überwunden werden – Stichwort „Investitionsschutz“. Doch dieser kann nur sinnvoll sein, wenn die Investition in die richtige Richtung führt.

Ähnliche Konsequenzen hat es, wenn das Vorgehen beim Entwicklungsprozess bzw. Vorgehensmodell so gewählt wird, dass eine Orientierung hin zu einer hohen Qualität der Anwendung erschwert wird. Dies kann zum Beispiel dadurch entstehen, dass die Anforderungsanalyse nicht genügend Raum und Zeit erhält oder die Erfüllung der Anforderungen nicht konsequent überprüft wird.

4.2. Die „UI-Macht“ liegt in den falschen Händen

Das vorherige Kapitel hat falsche oder extrem unvollständige Anforderungen behandelt. Grundsätzlich kann man in der Praxis



beobachten, dass bei der Gestaltung eines User Interface die dazugehörigen Anforderungen **niemals vollständig** erfasst sind. Auch wenn die wichtigsten Anforderungen bekannt sind, gibt es bei der Realisierung des UI immer Gestaltungsfreiheiten, über die dann jemand entscheiden muss. Und ebenfalls aus der Praxis ist bekannt, dass eine gute Usability korrekte Entscheidungen im Großen genau so wie im Kleinen erfordert: Die beste Anforderungsanalyse führt nicht zum Ziel, wenn in der Detailgestaltung des UI zu viele Fehler gemacht werden.

Diese Entscheidungen bezüglich der UI-Details werden allerdings erfahrungsgemäß oft von Personen verantwortet, deren Fokus nicht eine gute Qualität des UI ist:

- **Manager/Projektleiter** – Ihr Fokus ist die Verwaltung von Ressourcen und die Einhaltung von Verfahrensregeln und Meilensteinen. In der Regel sind sie nicht geschult in der Gestaltung von User Interfaces, und die Qualität des UI ist meist nicht Bestandteil ihrer Projektziele, an denen ihr Erfolg gemessen wird.
- **UI Entwickler** – Ihr Fokus liegt auf der technischen Umsetzung des UI. Wichtig ist ihnen eine schnelle Entwicklung mit Berücksichtigung der bekannten funktionalen Anforderungen und anderer Aspekte wie zum Beispiel Wartbarkeit. Eine Kompetenz in der UI Gestaltung hinsichtlich der Usability ist durchaus manchmal vorhanden, wird aber wegen der genannten Prioritäten oft nicht ausreichend beachtet.
- **Fachverantwortliche** – Ihr Fokus ist oft auf die funktionalen Anforderungen gerichtet. Die Qualität der Ausgestaltung im Ganzen interessiert sie wenig. Oft gehörtes Zitat: „Wie das im Detail gestaltet ist, ist mir eigentlich egal. Hauptsache, die Funktion XY ist da.“ Auch sie haben oft wenig Gestaltungskompetenz bezüglich des User Interface.

4.3. Der Wert von Qualität des UI wird unterschätzt

Selbst wenn alle wichtigen Anforderungen bekannt sind und die Qualität der UI-Details explizit im Fokus liegt (die „UI Macht“ also in den richtigen Händen liegt), muss es außerdem noch genügend Ressourcen (v. a. Zeit und Geld) geben, um das UI hinsichtlich nichtfunktionaler Anforderungen qualitativ hochwertig gestalten und umsetzen zu können. Die Praxis zeigt aber, dass im Rahmen des Projektmanagements dieser Aspekt oft wenig beachtet wird. Der Grund hierfür liegt nach Meinung des Autors darin, dass der Wert der Qualität des UI von den Ressourcenverantwortlichen unterschätzt wird. Das Projektziel ist erreicht, wenn die funktionalen Anforderungen erfüllt werden, die Qualität im UI zu erhöhen gilt als „teures Optimieren der letzten zehn Prozent“. Damit wird der Wert, den ein qualitativ hochwertiges UI für das gesamte Unternehmen hat, eventuell unterschätzt und entsprechend zu wenig Zeit und Geld für die Umsetzung zur Verfügung gestellt.

Ein zweiter Aspekt, der zur Unterschätzung des Wertes von qualitativ hochwertigem UI führt, ist dass die Konsequenzen von schlechten UIs unbekannt bleiben. Es ist oftmals schwer, quantitativ und objektiv nachzuweisen, welche Kosten durch schlechte UIs entstehen. Oft basieren explizite Entscheidungen gegen eine Veränderung des UI auf Annahmen zur Kosten-Nutzen-Relation, die nicht objektiv belegt werden.

Die Folge: Es muss an anderen Stellen investiert werden, um eine gewünschte Qualität der Prozessergebnisse zu erzielen – meist im operativen Alltag durch die Mitarbeiter bei der Benutzung der Anwendungen.

Die Belastung durch UI Balkonitis wird von den Anwendern ausgehalten oder unterschätzt

In der Praxis größerer Unternehmen erlebt man, dass Mitarbeiter sehr belastbar sind

und die Belastung durch UI Balkonitis nicht melden, ja manchmal auch stolz darauf sind, dass sie das komplexe UI trotzdem erfolgreich benutzen können. Hierfür gibt es verschiedene Erklärungsansätze:

- Die zu benutzenden Anwendungen werden im Rahmen des Prozessmanagement festgelegt, und die Mitarbeiter sind oftmals an der Gestaltung der Anwendungen nicht beteiligt. Dementsprechend betrachten die Mitarbeiter die Anwendungen als „gegeben“ und gehen davon aus, dass von ihnen erwartet wird, die Belastung durch die UI Balkonitis auszuhalten. Erst wenn diese Belastung über einen gewissen Schwellenwert hinausgeht, melden sich die Mitarbeiter zu Wort, dann allerdings erfahrungsgemäß eher unspezifisch, sodass eine direkte Ableitung von Handlungsanweisungen zur Neugestaltung der UIs schwierig ist.
- Die Mitarbeiter unterschätzen die Belastung durch UI Balkonitis und beachten sie deshalb nicht oder kommunizieren sie nicht.

In der Konsequenz ist es für das Management – die Entscheider über Ressourcen und Projektpläne – schwierig, die Belastung der Mitarbeiter durch eine schlechte Qualität der UIs in ihre Entscheidungen einzubeziehen.

4.4. Die Qualität von „Standard-Produkten“ wird überschätzt

Manche Unternehmen verfolgen das Prinzip, Standardsoftware der Eigenentwicklung von Anwendungen vorzuziehen. Bezüglich der zu erfüllenden Anforderungen wird auf funktionale Anforderungen hingewiesen. Sonstige Anforderungen inklusive Usability / Qualität der umzusetzenden Funktionen stehen nicht im Fokus. Dahinter steht oft die Annahme, dass die Standard-Software, die bereits bei Wettbewerbern erfolgreich eingesetzt wird, eine ausreichende bis gute Usability haben muss. Solange die Software in die IT-Landschaft des Unternehmens hineinpasst

und technisch gut integrierbar ist, wird sie akzeptiert.

Aus langjähriger Erfahrung als User Experience Architect bei SAP weiß der Autor, dass bei Herstellern betrieblicher Standardsoftware Usability nur ein untergeordneter Aspekt neben vielen anderen Faktoren ist. Die Qualität von Standardsoftware hinsichtlich des UI darf daher nicht überschätzt werden. Auch wenn die Gesamtkosten für hausinterne Software-Entwicklung geringer werden und die Integrationsfähigkeit der Standardsoftware explizit gefordert wird, so kauft man im Endeffekt „die Katze im Sack“, wenn das Thema Usability bzw. UI Balkonitis nicht explizit beachtet wird. Viele kleine Usability-Probleme können sich derart summieren, dass die besten funktionalen Eigenschaften doch nur zur Hälfte hinführen zu der bestmöglichen Lösung.

5. Maßnahmen, die nachhaltig gegen UI Balkonitis helfen

Es gibt verschiedene Ansätze, mit UI Balkonitis umzugehen:

- **Die Belastung durch UI Balkonitis hinnehmen, ohne genau zu wissen, wie hoch sie eigentlich ist.** Diese Praxis verfolgen viele Unternehmen. Verbesserungspotenzial suchen sie oft bei funktionalen Aspekten der Anwendungen, weil sie davon ausgehen, dass die nichtfunktionalen Usability-Probleme keine große Belastung darstellen oder eine Verbesserung an dieser Stelle mit unverhältnismäßigen Kosten verbunden wäre.
- **Punktuell die Belastung durch UI Balkonitis verringern, wenn sie einen fiktiven Grenzwert überschreitet.** Hilft an der spezifischen Stelle, löst aber das Grundproblem nicht, da zu erwarten ist, dass in vielen Fällen der fiktive Grenzwert nicht überschritten wird und damit dauerhafte Einschränkungen vorliegen, die in der Summe substantiell sein können.
- **Überall dort die Belastung durch UI Balkonitis verringern, wo der**

erwartete Nutzen größer ist als der entstehende Aufwand für die Belastungsverringerung.

Voraussetzung für diesen Ansatz ist allerdings, dass die Kosten durch UI Balkonitis bekannt sind. In der Regel ist die Ermittlung des Kosten-Nutzen-Verhältnisses schwierig (siehe Kapitel 2).

- **Strukturell und prozessual vorsorgen, sodass UI Balkonitis gar nicht erst entsteht.** Hier ist vor allem die Gestaltung von Organisationsstrukturen und Vorgehensmodellen etwa beim Entwicklungsprozess und im Projektmanagement gemeint sowie die explizite Schaffung einer Integrationsarchitektur. Auch hier besteht das Problem, dass es in der Regel schwierig ist, die Kosten der Vorgehensmodelle und architekturellen Veränderungen korrekt ins Verhältnis zu setzen zu dem Nutzen, den sie im Hinblick auf UI Balkonitis bringen.

Es ist wichtig zu bemerken, dass alle beschriebenen Umgangsmöglichkeiten legitim sind. Es ist eine Frage der Unternehmenskultur sowie eine Frage des Umgangs mit Unsicherheiten in der Kosten-Nutzen-Abschätzung und welchen Ansatz man verfolgt, um die Anwendungslandschaft zu formen. Die Betrachtung der UI Balkonitis ist ein Ansatz neben anderen möglichen Ansätzen, um die Gesamtqualität des Systems zu verbessern.

Es geht nicht darum, Balkone zu verhindern oder dogmatisch die Perfektionierung des IT-Systems zu verfolgen. Das Ziel sollte sein, die Entscheidung zur Bildung und Gestaltung von Balkonen im Detail zu informieren, sodass das Gesamtsystem – Unternehmen und Mitarbeiter mit verwendeter Anwendungslandschaft – die gesteckten Ziele erreicht. Mit welchem Stellenwert die Gebrauchstauglichkeit in einem Unternehmen betrachtet wird, ist – wie gesagt – eine Frage der Unternehmenskultur. Und ein Grundproblem beim Umgang mit UI Balkonitis ist, dass das Kosten-Nutzen-Verhältnis bei den zu ergreifenden Maßnahmen meist schwer einzuschätzen ist.

Dementsprechend erscheinen dem Autor seiner Erfahrung nach die folgenden Maßnahmen geeignet, bei Entwicklung und Einsatz betrieblich genutzter Software nachhaltig die Usability der Anwendungslandschaft eines Unternehmens zu verbessern und damit die UI Balkonitis zu verringern.

5.1. Die Belastung objektiv und quantitativ nachweisen

Der beste Weg zur nachhaltigen Verbesserung der Situation ist ein objektiver und quantitativer Nachweis der Belastungen durch UI Balkonitis. **Solange dies nicht geschieht, werden Verbesserungsversuche bezüglich der nicht-funktionalen Usability an dem Widerstand der Ressourcenverantwortlichen scheitern.**

Methoden zur Messung von Aufwänden durch Usability-Probleme sind zum Beispiel GOMS-Analysen oder KPI-Messungen im Rahmen von Usability-Tests. Die Herausforderungen sind hierbei:

- Welche Messgrößen sinnvoll sind, hängt maßgeblich von dem zu untersuchenden System ab. Sie müssen sorgfältig gewählt werden, denn sie entscheiden darüber, ob die Ergebnisse einer „Usability-Messung“ für die Ressourcenverantwortlichen nachvollziehbar die Belastungen darstellen.
- Die Messung der Belastung durch UI Balkonitis muss so effizient wie möglich durchgeführt werden, um das Kosten-Nutzen-Verhältnis nicht von vornherein zu sehr zu belasten.
- Es ist oftmals fraglich, mit welchem „Optimum“ die bestehende Lösung verglichen werden soll. Nimmt man als „Maß der Dinge“ eine Referenz-Software (zum Beispiel SAP) oder den Gestaltungsvorschlag eines Usability-Experten?

An diesen Aspekten entscheidet sich, ob die Belastung durch UI Balkonitis von den Entscheidern und Managern korrekt eingeschätzt wird.



Wegen der methodischen Herausforderungen wird diese Maßnahme oftmals vermieden, **die Erfahrung des Autors ist aber, dass ohne einen objektiven und quantitativen Nachweis der Belastung im Kontext der UI Balkonitis keine nachhaltigen Verbesserungen zu erzielen sind.** Insbesondere führt das oft genutzte Instrument des **qualitativen** Usability-Testing nicht zum Ziel, da keine Hilfestellung für die Kosten-Nutzen-Abschätzung gegeben wird.

5.2. Einen direkt erkennbaren Mehrwert beisteuern

Es ist wichtig, dass der Usability-Experte einen **direkt erkennbaren** Mehrwert zu einem Projekt beisteuert, damit sein Beitrag akzeptiert wird. Dies kann auf verschiedene Art erfolgen, zum Beispiel:

- wertvolle Gestaltungsvarianten für die zu betrachtende Anwendung liefern.
- Diskussionen zu Gestaltungsvarianten abkürzen, weil der Usability-Experte mit stichhaltigen Argumenten – vor allem mit objektiven, quantitativen Messungen – eine nachvollziehbare Gewichtung beisteuert.
- Substantiell bei der Umsetzung im Projekt mithelfen.

Erfolgt diese Maßnahme nicht, wird der Usability-Experte in erster Linie als „Störenfried“ wahrgenommen, der den anderen Mitarbeitern im Projekt Mehrarbeit beschert statt bei der Lösung zu helfen. Ohne die Unterstützung derjenigen, die die Umsetzungen im Projekt leisten, ist es für einen Usability-Experten schwer, gegen UI Balkonitis anzukommen.

5.3. Das Thema verkaufen: Kommunikation, Marketing, Erklären, Demonstrieren, Sponsoring

Es ist meist schwierig, die Kosten einer Usability-Messung zu rechtfertigen, da das Kosten-Nutzen-Verhältnis schwer zu ermitteln ist. Dementsprechend müssen solche Maßnahmen verkauft werden, damit Entscheider den Mut finden, sie zuzulassen.

- Die Kommunikation zu dem Thema aktiv betreiben, Leute ansprechen, Ergebnisse präsentieren, „sich unvergesslich machen“.
- Nicht davon ausgehen, dass die Probleme doch offensichtlich sind, sondern erklären, demonstrieren, überzeugen, werben.
- Beispiele aus dem IT-System des Unternehmens selbst heranziehen: Positive Beispiele sind hilfreich; negative Beispiele auch, wenn keiner dabei „sein Gesicht verliert“.
- Fürsprecher / Sponsoren für das Thema gewinnen und diese dafür werben lassen (z. B. Mitbestimmungsgremien).
- Die Bewertung der Belastungen durch UI Balkonitis ist immer diskussionswürdig, also sollte man solche Diskussionen aktiv einfordern.
- Die Sozialverantwortung gegenüber den Benutzern bewusst machen.

5.4. Verbesserungen von hochqualifizierten Experten entwickeln lassen

Um nachhaltig gegen UI Balkonitis ankommen zu können, braucht es überzeugende und wirkungsvolle Umgestaltungen der problematischen „Balkone“. Deshalb ist es notwendig, solche Umgestaltungen von jemandem entwickeln zu lassen, der eine große Kompetenz in Sachen UI Design, Usability sowie Software-Architektur besitzt. Außerdem muss er über eine hohe Sozialkompetenz sowie starke Kommunikationsfähigkeiten, Kompromissbereitschaft und Empathie verfügen. Dies ist keine Aufgabe für einen Praktikanten oder einen Entwickler, der sich nebenbei ein paar Grundkonzepte der Usability-Arbeit angelesen hat.

Sind die genannten Fähigkeiten nicht bei dem Usability-Experten vorhanden, dann muss man damit rechnen, dass seine Vorschläge nicht gut genug sind, um gegen schlechtere Alternativen oder die bereits bestehende Lösung anzukommen, oder sie werden nicht gehört, nicht ernstgenommen oder nicht richtig bewertet.

5.5. Auf lange Sicht: Strategische Einbettung des Themas Usability in Management und Prozesse

In der Theorie ist klar, dass UI Balkonitis nachhaltig vermieden werden kann, wenn die Entwicklungsprozesse, Vorgehensmodelle und Kompetenzzuweisungen entsprechend gestaltet sind. Hier kann die Entstehung schlechter UIs verhindert werden. Wenn das Unternehmen ein Produkt mit Mensch-Maschine-Kommunikation (User Interface) entwickelt und verkauft, dann zeigt sich auch in der Praxis, dass dieser Ansatz nachhaltig zielführend ist.

In der Praxis der **Entwicklung betrieblich genutzter Software innerhalb des nutzenden Unternehmens** (also nicht beim Software-Hersteller) zeigt sich, dass diese strategische Betrachtung nur hilfreich ist, wenn eine weitreichende Akzeptanz und Wertschätzung durch alle beteiligten Entscheidungsträger und Mitarbeiter vorliegt. Es müssen erst alle zuvor genannten Maßnahmen etabliert werden und greifen, bevor eine strategische Einbettung sinnvoll erscheint.

6. Fazit

UI Balkonitis ist eine Spielart (neben manch anderen), wie schlechte Usability entsteht und wo die Usability nicht so leicht zu verbessern ist. Sie ist typisch für betrieblich genutzte Anwendungen und hat Eigenschaften, die eine nachhaltige Verbesserung der Usability erschweren.

Eine der Kernaussagen des Beitrags lautet: Bei UI Balkonitis ist es für eine nachhaltige Verbesserung unverzichtbar, sich mit dem schwierigen Thema des objektiven und quantitativen Nachweises der Belastung auseinanderzusetzen. Wie man dies im Detail umsetzt und wie weitreichend man es umsetzen sollte, hängt von den zu betrachtenden Anwendungen und von der Unternehmenskultur ab. Insofern ist es schwierig, hier allgemeingültige Empfehlungen auszusprechen.

Eine weitere These ist, dass viele Maßnahmen, die sonst geeignet sind, um Usability zu verbessern, im Kontext von betrieblichen Anwendungen und UI Balkonitis nicht wirken. Dementsprechend entsteht bei vielen Usability-Experten bezüglich der Betrachtung betrieblicher Anwendungen eine Resignation angesichts der Widerstände, die der Verbesserung von Usability entgegenstehen. Wer um die Zusammenhänge rund um die UI Balkonitis weiß, hat es leichter, mit der Situation umzugehen und letztendlich geeignete Maßnahmen zu ergreifen, um die Nutzungsqualität nachhaltig zu verbessern.

Bei den DEVK Versicherungen ist das Thema UI Balkonitis bekannt. Zurzeit wird dort eine effiziente Methode entwickelt, um die Belastung durch UI Balkonitis objektiv und quantitativ nachzuweisen. Außerdem arbeitet der Autor als User Interface Designer und Usability-Experte für die DEVK in verschiedenen Projekten mit, um das Thema Usability in die Anwendungsentwicklung einzubringen. Um vorzusorgen, dass UI Balkonitis möglichst gar nicht erst entsteht, wird von Seiten der IT-Architektur eine Integrationsplattform (Enterprise Service Bus) eingeführt, um die funktionale Verknüpfung zwischen Anwendungen zu erleichtern. Letztendlich wird in jedem Einzelfall geprüft, welcher Ansatz zur Verbesserung der Gesamtqualität der Anwendungslandschaft unter Beachtung der Kosten-Nutzen-Relation am sinnvollsten erscheint, wobei das Thema Usability einer der betrachteten Faktoren ist.

7. Über die DEVK Versicherungen

Den DEVK Versicherungen vertrauen bundesweit rund 4 Millionen Kunden mit 13,4 Millionen Risiken in allen Versicherungssparten. Dass sie besonders treue Kunden sind, hängt nicht zuletzt von der persönlichen Nähe ab: rund 1.250 Geschäftsstellen, gut 2.250 hauptberufliche Vertriebspartner und rund 3.400 nebenberufliche Vermittler sprechen für sich. Langjähriger Kooperations- und Vertriebspartner sind zudem die

Sparda-Banken. Nach der Anzahl der Verträge ist die DEVK Deutschlands viergrößter Hausrat-, fünftgrößter Pkw- und sechstgrößter Haftpflichtversicherer.