Rasche Anwendungsentwicklung mit Fokus auf Wartbarkeit am Beispiel einer Software im Bereich Standard Operation Procedures

Bachelorarbeit

Ole Hensel

Betreuer:
Prof. Lutz Prechelt
Christopher Oezbek

AG Software Engineering, Institut für Informatik,
Freie Universität Berlin
in Kooperation mit
Klinik für Anästhesie und operative Intensivmedizin, Campus Virchow-Klinikum, Charité Berlin

Inhaltsverzeichnis

1. Einleitung.....1
1.1 Umfeld.....1
1.2 Motivation.....1
1.3 Gliederung.....2
2. Standard Operation Procedures.....2
2.1 Krankenhaus: Finanznot und ineffiziente Prozesse.....2
2.2 Klinische Behandlungspfade und Standard Operation Procedure.....3
2.3 Erstellung von SOPs.....7
2.4 Softwareanforderungen.....9
3. Content Management Systeme.....10
3.1 Das Internet und seine Webseiten.....10
3.2 Einführung in Content Management Systeme.....13
3.3 OpenCms.....17
4. Softwarewartung.....22
4.1 Einführung in die Softwarewartung.....22
4.2 Verbesserung der Wartbarkeit.....31
5. Ergebnisse der Softwareentwicklung.....32
6. Betrieb und Wartung gewährleisten.....36
7. Zusammenfassung.....45
8. Anhang.....46
9. Literaturverzeichnis.....74

Danksagung

Mein ganzer Dank gilt meinen Eltern, die mir zwei Studiengänge ermöglicht haben.

Ole Hensel

1. Einleitung

1.1 Umfeld

Im Rahmen meiner Bachelorarbeit in der Arbeitsgemeinschaft Software Engineering im Institut für Informatik der Freien Universität Berlin habe ich im Auftrag der Klinik für Anästhesie und operative Intensivmedizin der Charité eine Anwendung zur Erstellung, Ansicht und Pflege von Standard Operation Procedures angefertigt.
Die Arbeitsgruppe Software Engineering ist recht jung. Sie versucht die Differenz zwischen *State of the Art* und herkömmlicher Verwendung der Softwareentwicklung zu minimieren. Die Forschungsschwerpunkte liegen insbesondere in der Untersuchung von Softwarefehlern, im Bereich der Mikroprozessbetrachtung und der Paarprogrammierung. Die Arbeitsgruppe strebt in der Lehre eine realistische und anwendbare Sicht an.
Die Klinik für Anästhesie und operative Intensivmedizin ist am Campus Virchow-Klinikum der Charité Berlin angesiedelt. In der Klinik werden ca. 23 000 Anästhesien pro Jahr betreut. Der wissenschaftliche Schwerpunkt liegt im Bereich Intensivmedizin. Daneben ist die Klinik Referenzzentrum zur Behandlung des schweren akuten Lungenversagens (ARDS).
Im Frühjahr 2005 bildete sich in dieser Klinik für Anästhesiologie und operative Intensivmedizin eine Arbeitsgruppe mit dem Ziel die vorhandenen Ressourcen effizienter einzusetzen.

1.2 Motivation

Meine Erfahrungen während diverser Famulaturen und im Praktischen Jahr (PJ) waren gekennzeichnet durch ineffiziente Krankenhausprozesse. Tagelang war ich ausschließlich damit beschäftigt, nach nicht auffindbaren Röntgenbildern zu fanden oder dasselbe Formular immer wieder manuell auszufüllen.
Aufgrund dieser Erfahrungen nahm ich gern den Auftrag an, im Rahmen meiner Bachelorarbeit eine Anwendung zu entwickeln, die die Optimierung von internen Krankenhausprozessen unterstützen soll.
Wenn ein MRT-Bild nicht aufzufinden war, wurde eine kostspielige Zweituntersuchung durchgeführt. Durch solch ineffiziente Prozessabläufe werden Gelder verschwendet. Noch schlimmer aber ist, dass dadurch den Krankenhausmitarbeiter die Motivation genommen wird, gute Arbeit zu leisten.

1.3 Gliederung

Um eine erfolgreiche Anwendung zu entwickeln, habe ich mich intensiv mit dem Umfeld der Anwendungsentwicklung beschäftigt, um die Anforderungen des Gesundheitswesens besser verstehen zu können (Kapitel 2 „Standard Operation Procedures“).
Des Weiteren gehe ich ausführlicher auf die Grundprinzipien von Content Management Systemen ein (Kapitel 3 „Content Management Systeme“), um diese Technologie und deren Auswirkungen verständlich zu machen. Dazu stelle ich die Anwendung *OpenCms* näher vor.
Die Bedeutung der Softwarewartung als den längsten Prozess der Softwareentwicklung wird im Kapitel 4 „Softwarewartung“ näher betrachtet.
Im Anschluss wird die Webanwendung vorgestellt (Kapitel 5 „Ergebnisse der Anwendungsentwicklung“).
Das Kapitel 6 befasst sich dann mit einigen konkreten Überlegungen zur Wartung der Webanwendung („Wartung und Betrieb der SOP-Anwendung“).
Die Zusammenfassung (Kapitel 7) enthält die von mir gezogenen Schlussfolgerungen zu diesem Themenkomplex.

2. Standard Operation Procedures

2.1 Krankenhaus: Finanznot und ineffiziente Prozesse

Durch die Umstellung auf ein fallpauschalenbasiertes System bei der Krankenhausvergütung im Januar 2004 wurden grundlegende Änderungen im deutschen Krankenhausbereich eingeleitet [1, 3, 5, 6]. Hintergrund dieser Reform waren die enorm gestiegenen Kosten des Gesundheitssystems [13].
Grundlage der neuen Vergütung bilden *Diagnosis Related Groups* (DRG). Diese klassifizieren Behandlungen anhand von medizinischen Kriterien (Art der Erkrankung, Schwere der Erkrankung, usw.) und dem geschätzten Verbrauch von Ressourcen (Arbeitszeit, Material, usw.). Anhand der Einordnung in eine DRG wird die Fallpauschale errechnet, die von der Krankenkasse an die Krankenhäuser gezahlt wird.

Vor der Umstellung wurden Krankenhäuser über den Auslastungsgrad ihrer Bettenbelegung bzw. der Verweildauer der Patienten vergütet [1].
Nun richtet sich die Höhe der Fallpauschalen ausschließlich nach der Patientenerkrankung. So können besonders kostspielige Behandlungsvarianten und unnötige Zweituntersuchungen nicht mehr vom Krankenhaus abgerechnet werden. Die Fallpauschalen berücksichtigen jedoch nicht, dass der Behandlungsaufwand zum Beispiel bei älteren Patienten oft ungleich höher ist als beim jüngeren Patienten.

Durch die Umstellung auf das neue Abrechnungssystem wird ein grundlegendes und rasches Umdenken von den Krankenhausmitarbeitern gefordert. Sich auf ein komplett geändertes Vergütungssystem einzustellen, ist schwierig. Die große Gefahr bestand und besteht immer noch, dass die Krankenhauseinnahmen schneller sinken, als Einsparungen vorgenommen werden können [6].

Vorausgesehen und nun eingetreten ist die Verminderung der Krankenhausvergütung durch die Umstellung auf das neue DRG-Abrechnungssystem [3]. Die Lage ist ernst, Expertenschätzungen zufolge werden 15 % der heutigen Krankenhäuser im Jahre 2010 so nicht mehr existieren [41]. Um zu überleben, müssen Kosten im Krankenhaus eingespart werden.

In vielen Fällen sind Krankenhausprozesse immer noch ineffizient aufeinander abgestimmt [3]. So konnte ich beispielsweise beobachten, dass eine notwendige Ultraschalluntersuchung den behandelnden Arzt mehrere Stunden in Anspruch nahm, weil das notwendige Gerät zu einer anderen Abteilung gehörte. Solche Schnittstellenprobleme blockieren den weiteren Behandlungsablauf des Patienten und verursachen Mehrkosten [6]. Um diese zu verhindern, müssen diese Schnittstellenprobleme identifiziert und durch Strukturänderungen beseitigt werden. So

analysiert Bauer et al. [3] verschiedene Behandlungsverläufe in diesem Bereich der Anästhesie. Die Anästhesie liegt zwischen Station und Operation in der Behandlungskette und ist ein besonders kostenintensiver Bereich der Patientenversorgung [3].

Wenn man den finanziellen Druck durch Einführung der Fallpauschalen nicht nur als Bedrohung empfindet, sondern auch als Chance sieht, interne Arbeitsabläufe und Strukturen zu optimieren, kann es zu einer Patientenversorgung mit weniger Ressourcenverschwendung bei gleichbleibender oder sogar höherer Qualität kommen[3].

2.2 Klinische Behandlungspfade und Standard Operation Procedure

Klinische Behandlungspfade (KBP) beschreiben den optimalen Ablauf einer stationären Behandlung von der Patientenaufnahme bis zur Entlassung. Sie sind „eine Art Verfahrensanweisung, die beschreibt, wie ein Patient, der mit einer definierten Symptomatik beziehungsweise einer definierten Erkrankung in das Krankenhaus kommt, systematisch nach einem vorgegebenen Ablauf diagnostiziert und behandelt wird" [1].

Standard Operation Procedures (SOPs) sind Standardarbeitsanweisungen, welche das Vorgehen innerhalb eines klar abgrenzbaren Arbeitsprozesses beschreiben. Häufig wiederkehrende Arbeitsabläufe werden textlich beschrieben und den Mitarbeitern erklärend an die Hand gegeben. Sie ergänzen einen KBP modular (Abb. 1). Zum Beispiel beschreibt eine SOP in der Anästhesie die Materialien und Vorbereitungstätigkeiten, die notwendig sind, um eine Operation durchzuführen.

Folgende Fragen müssen in einer SOP der Klinik für Anästhesie und operative Intensivmedizin, Charité, Campus Virchow-Klinikum beantwortet werden [55]:

- Welche Medikamente braucht der Patient vor der Operation(Prämedikation), z.B. am Abend zuvor?
- Welche Vorbereitungen sind im Operationsraum zu treffen, welches Material und welche Medikamente werden benötigt?
- Welche Überwachung (Monitoring) soll angewendet werden?
- Wird der Patient eventuell Blut- oder andere Infusionsprodukte im Verlauf der Operation benötigen?
- Wie soll die Narkose eingeleitet und wie durchgeführt werden?
- Wie ist der Patient auf dem Operationstisch zu lagern?
- Welche speziellen Besonderheiten sind bei dieser Art von Operation zu beachten?
- Welche Komplikationen treten gehäuft auf?
- Was muss nach der Operation (postoperatives Management) beachtet werden?

- Welche Literatur wurde zur Erstellung dieser SOP verwendet?

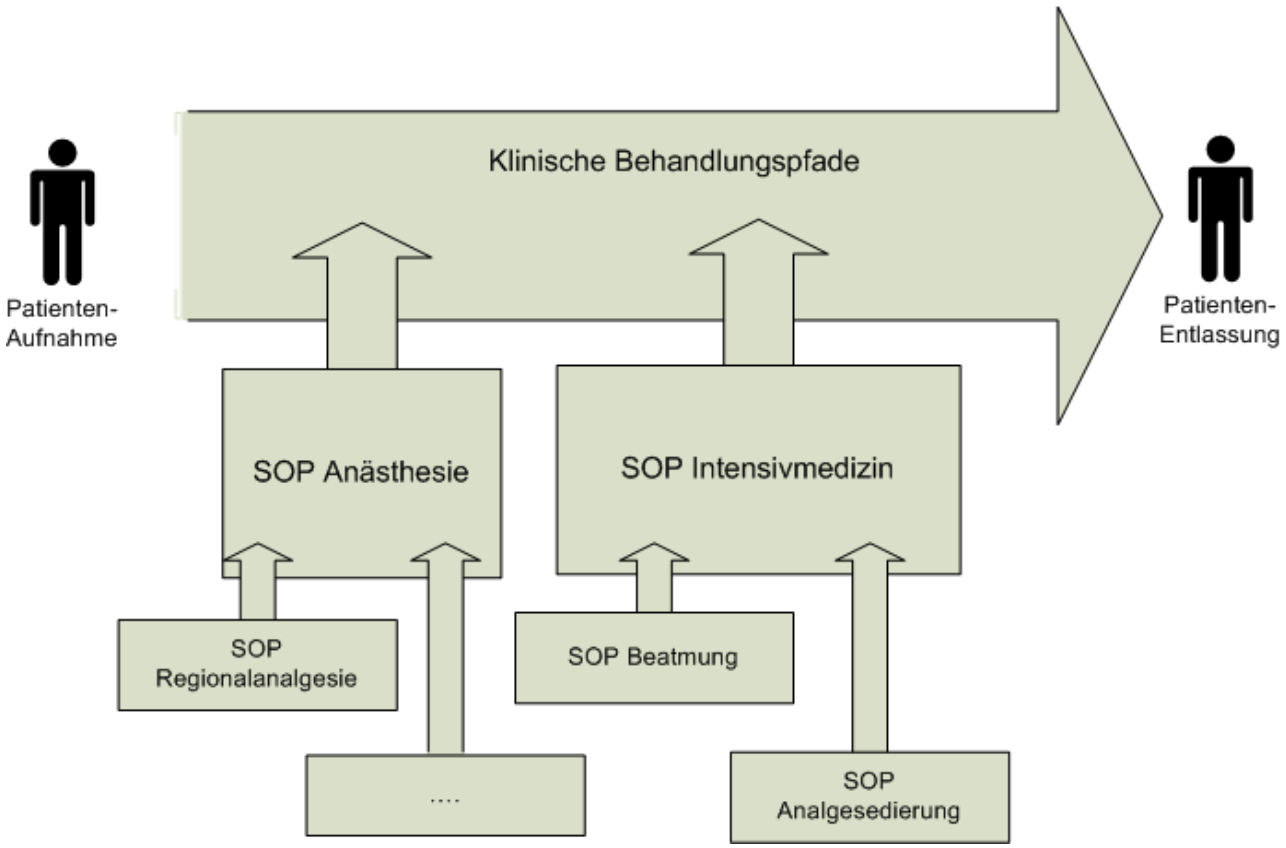

Abb. 1: Anästhesie-SOPs zur Operationsvorbereitung integrieren sich in einem klinischen Behandlungspfad (in Anlehnung an [8]).

Zur Veranschaulichung der Struktur und der Inhalte ist im Anhang eine im Laufe der Bachelorarbeit entstandene SOP aufgeführt (Anhang B).

Das Ziel von KBP und SOPs ist die Standardisierung von Krankenhausprozessen [1]. Ihre konsequente Anwendung führt zu einer koordinierten Vorgehensweise aller Abteilungen (Abb. 2). Die Anwendung zwingt Abteilungen dazu, ihre Arbeitsabläufe mittels interdisziplinärer Denkansätze untereinander abzustimmen, ebenso wie ihre internen Strukturen und Prozesse zu optimieren [6]. Bisherige Erfahrungen zeigen, dass KBP und SOPs wirkungsvoll zur Verbesserung von Prozessabläufen in der Patientenversorgung beitragen [1, 3, 5, 6].

Außerdem fördern sie die notwendige Übernahme von wissenschaftlich-begründeten Wissens in den Klinikalltag, denn es gilt immer noch: „The harsh reality of medicine is that many, if not most, daily clinical decisions are not based on valid scientific fact" [14]. Dies kann geändert werden, wenn Erkenntnisse der *evidenz-basierten Medizin* (EbM) in den klinischen Behandlungspfaden

und somit in den klinischen Alltag mit einbezogen werden. Mit statistischen Methoden untersucht die EbM, ob sich ein bestimmter Sachverhalt (zum Beispiel bessere Therapie des Medikaments xy gegen Medikament yz) erhärtet oder widerlegt werden kann. Dazu werden Ergebnisse großer Studien zusammengefasst [9]. Die konsequente Anwendung der EbM stellt einen Paradigmenwechsel in der Medizin dar [4].

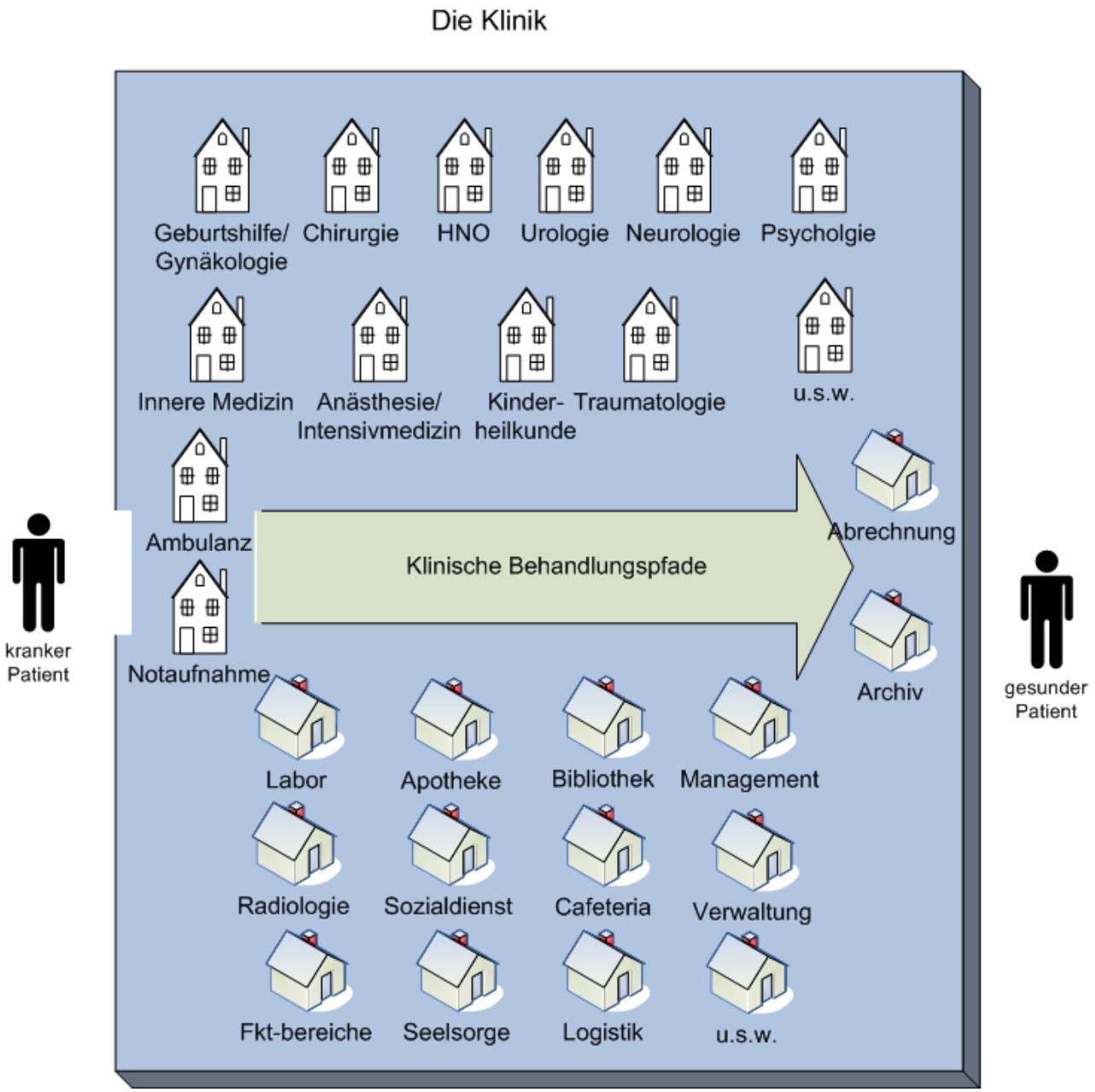

Abb. 2: Die Anwendung eines klinischen Behandlungspfades führt zu einer Koordination aller Klinikabteilungen (in Anlehnung an [11]).

Kox und Spies [2 zitiert nach 8] fassen die Vorteile von KBP wie folgt zusammen:

1. Optimierung von Prozessabläufen,
2. Anwendung evidenz-basierter Medizin,
3. Rationaler Umgang mit Ressourcen,
4. Verbesserung der Ausbildung und Weiterbildung,
5. Vereinfachung der Einarbeitung neuer Mitarbeiter,
6. Verbesserung des Qualitätsmanagements und der Qualitätssicherung,
7. Erhöhung der Kostentransparenz.

Jedoch muss darauf hingewiesen werden, dass nicht in jedem Bereich der Medizin KBP und SOPs gleich gut einsetzbar sind. So ist zum Beispiel der Bereich der Inneren Medizin sehr viel abhängiger von der Diagnostik. Dadurch wird der Behandlungsverlauf sehr viel schwerer als zum Beispiel in der Anästhesie vorhersagbar. Wenn der Patient auf den Anästhesisten trifft, ist die Diagnose des Patienten meist schon abgeklärt. Mit bekannter Diagnose lässt sich der weitere Behandlungsablauf besser standardisieren.

Der Befürchtung einiger Mitarbeiter mit Einführung von KBP komme es zu einer „Kochbuchmedizin" und zu einer Einschränkung ihrer Freiheiten [6, 7], muss entgegen getreten werden. Es geht bei sinkenden Krankenhauseinnahmen darum, die Patientenbehandlung qualitativ und quantitativ weiterhin sicher zu stellen. Der Patient mit seiner Erkrankung muß im Mittelpunkt stehen und nicht die Freiheiten der Mitarbeiter. Die KBP und SOPs müssen einerseits verbindlich sein, anderseits sollen alle Mitarbeiter wissen, dass in jedem begründeten Fall von den KBP bzw. der SOP abgewichen werden kann. Patienten und ihre Erkrankungsverläufe sind individuell, so dass die Behandlung jedes Mal einen speziellen Verlauf nehmen kann. Bauer et al. [3] betonen dazu: KBP und SOPs lassen dem Arzt seinen Entscheidungsspielraum.

Es ist festzuhalten, dass die Anwendung klinischer Behandlungspfade zu einem effizienteren Ressourceneinsatz führt und dadurch Kosten sparen hilft bei gleicher oder höherer Qualität der Patientenbehandlung [1].

2.3 Erstellung von SOPs

Die Erstellung und Anwendung von Behandlungspfaden setzt bei allen Mitarbeitern eine selbstkritische Auseinandersetzung mit ihren täglichen Arbeitsabläufen voraus. Erst wenn erkannt wird, dass Arbeitsprozesse verbesserungsfähig sind, werden gute (d.h. praxisnahe) klinische Behandlungspfade entwickelt und angewendet [1].

Die Entwicklung bzw. Überarbeitung von SOPs ist mit einem großen personellen

und zeitlichen Aufwand verbunden [5]. Dieser kann durch Softwareunterstützung erheblich verringert werden [7].

Fertig erstellte SOPs müssen den jeweiligen Klinikgegebenheiten entsprechen und dürfen nicht einfach von anderen Kliniken übernommen werden [6]. Die gute Anpassung der SOPs trägt maßgeblich zu iherem Erfolg bei [7]. Um dies auch bei Änderungen der Klinikbegebenheiten zu wahren, muss eine zu entwickelnde Software die Möglichkeit bieten, fertige SOPs zu überarbeiten.

Bei der Erstellung bzw. Pflege von SOPs kann man drei beteiligte Gruppen unterscheiden:

- Verfasser
- Prüfer
- Mitglieder des Freigabe-Gremiums

Die Gruppe der *Verfasser* ist verantwortlich für die Ersterstellung bzw. Überarbeitung einer SOP, die Mitglieder dieser Gruppe sollten besonderes Fachwissen auf dem zu beschreibenden Gebiet aufweisen und den klinischen Alltag gut kennen.
Die *Prüfer* kontrollieren den Inhalt der SOPs auf formale und medizinische Korrektheit. Diese Gruppe setzt sich üblicherweise aus Oberärzten zusammen.
Die Mitglieder des *Freigabe-Gremiums* geben nach einer erneuten Prüfung die SOPs frei. In dieser Gruppe sollten Mitglieder aus allen Bereichen wie zum Beispiel Ärzte, Pflege, Einkauf u.s.w. involviert sein.

Alle weiteren Mitarbeiter der Klinik sind sogenannte *Gäste*. Sie können in den fertigen und im Intranet freigegebenen SOPs nachschlagen.

Eine zu entwickelnde Anwendung benötigt eine Benutzer-, Gruppen und Rechteverwaltung.

Martin et al. [12] schlägt folgenden Ablauf zur Entwicklung bzw. Überarbeitung von SOPs vor:

1. Zweck der SOP bestimmen,
2. Ist-Zustand erheben,
3. Verfasser bestimmen,
4. Verantwortlichen festlegen,
5. Erstellung der SOP ,
6. Eindeutige Kennzeichnung der SOP,
7. Evidenz-basierte Literatur angeben,
8. Prüfung der SOPs,
9. Praxistest mit allen Betroffenen,
10. Freigabe der SOP,
11. Änderungen einarbeiten,

12. Außerkraftsetzung der SOP, wenn Inhalt ungültig geworden ist,
13. Laufzeit der SOP festlegen,
14. Information aller Mitarbeiter über das Vorliegen neuer SOPs
15. Überprüfung der Einhaltung von SOPs,

Bei der Erstellung der SOP ist darauf zu achten, dass der Text sich auf wesentliche Informationen konzentriert und klar strukturiert ist. Ein vorgegebenes Format hilft bei der textlichen Strukturierung, so dass die Informationen schneller für den Lesenden erfassbar werden. Es hat sich bewährt, im ersten Teil der SOP eine Übersichtstabelle zu verwenden [2]. Sie liefert komprimiert die wichtigsten Informationen im Überblick, zum Beispiel die Art der Narkose und das benötigtes Material. Als Anschauungsbeispiel für eine SOP sei auf den Anhang B verwiesen.

Jede SOP sollte einen verantwortlichen Prüfer aufweisen. Damit es zu einer Interaktion der Nutzer, Verfasser und Prüfer kommen kann, sind Mechanismen der Rückkopplung (Feedback) einzubauen [10]. Das kann in einer Webanwendung über die Angabe der Email des Verantwortlichen oder aber durch das Anfügen von Kommentaren an der Text erreicht werden.

Deutlich erkennbar muss der Zeitraum der Gültigkeit einer SOP sein. Bisherige interne Erfahrungen aus der Charité zeigen, dass ein Zeitraum von 1-2 Jahre empfehlenswert ist. Nach Ablauf dieser Zeit ist eine inhaltliche Prüfung der SOP dringend notwendig. Im Zeitraum der Freigabe sind natürlich Überarbeitungen der SOP zu berücksichtigen. Neue wissenschaftliche Erkenntnisse sind genauso einzuarbeiten, wie Änderungen im Arbeitsablauf.

Wichtig für die Akzeptanz der SOPs ist, dass alle beteiligten Berufsgruppen bei der Entwicklung neuer SOPs einbezogen werden müssen, dies steigert auch deren Motivation die SOPs später anzuwenden [7]. So sind zum Beispiel Mediziner, Pflegekräfte, Krankenhausapotheker, Physiotherapeuten, Psychologen, Vertreter des Einkaufs/ des Managements und medizinische technische Assistenzen (MTA) mit einzubeziehen. Um eine sinnvolle Anwendung zu erfahren dürfen SOPs nicht von einzelnen Personen oder Gruppen erstellt werden und dann den anderen Mitarbeitern aufgezwungen werden [7].

Die SOPs müssen schnell und einfach von jedem Klinikmitarbeiter einsehbar sein [6].

2.4 Softwareanforderungen

Die Ausführungen in diesem Kapitel ergeben folgende Anforderungen an eine zu entwickelnde Software:

- Erstellen, Überarbeiten und Löschen von Texten, mit der Möglichkeit

Tabellen einzubinden,
- Intranetbasierte Lösung,
- Benutzer- bzw. Gruppenverwaltung,
- Rechteverwaltung,
- Erstellung der Texte durch mehrere Anwender ermöglichen,
- Nach einem Kontrollprozess Veröffentlichung der Texte,
- Die Möglichkeit die Veröffentlichung wieder rückgängig zu machen.
- Alle Mitarbeiter über neue Texte informieren,
- Feedbackmechanismen.

Alle diese Anforderungen kann, meiner Meinung nach ein den Bedürfnissen speziell angepasstes Content Management System, sehr gut erfüllen.

3. Content Management Systeme

3.1 Das Internet und seine Webseiten

Die Anzahl der Webseiten im Internet hat in den letzten 10 Jahren exponentiell zugenommen [15]. Gleichzeitig ist die Komplexität der Webseiten angestiegen, so sind Benutzerinteraktion und Datenbankanbindung inzwischen weit verbreitet. Viele tägliche Arbeitsabläufe sind ohne Informationen aus dem Internet heute nicht mehr vorstellbar. Diese Informationen müssen aktuell und zuverlässig sein. Dies macht eine gute Webseite aus [15]. Unternehmen sind mehr den je darauf angewiesen ihre Webseiten (*Homepages*) aktuell zu halten, um ihr Image nach außen zu wahren [36].

Eine Webseite setzt sich aus vier Bestandteilen zusammen (Abb. 3):

1. *Inhalt* (Content)
2. *Funktionalität*
3. *Präsentation*
4. *Struktur*

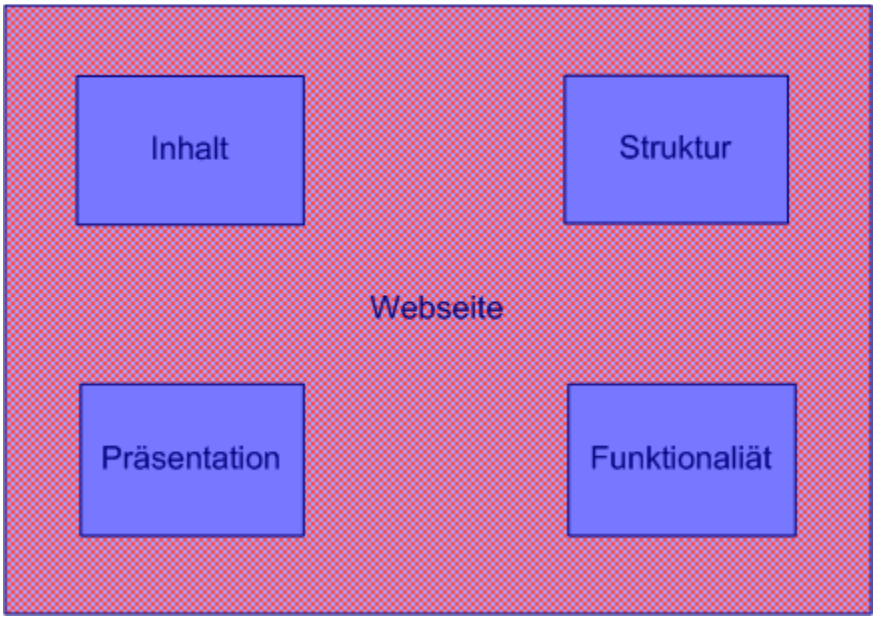

Abb. 3: Die vier Bestandteile einer Webseite.

Content bezeichnet dabei den Inhalt, der sich dem Betrachter auf einem Informationsträger optisch repräsentiert [16]. Informationsträger kann das Internet aber auch eine CD-Rom sein. Vor wenigen Jahren stand im Internet noch die Darstellung von Text im Vordergrund. Durch neue Technologien (breitbandige Inter-

netzugänge, neue Kompressionmethoden) kam es immer mehr zu multimedialen Inhalten. Heute sind im Internet verschiedenste Inhalte verbreitet: Texte, Grafiken, Videos, Audio, Animationen, aber auch Anwenderinteraktionen und Datenbankeninhalte. Insbesondere die Funktionen wie Interaktionsmöglichkeit durch den Anwender und die Einbeziehung von Datenbanken machen die Webseiten immer mehr zu so genannten Webanwendungen [37].

Die *Struktur* definiert die Anordnung einzelner Inhalte. Sie legt beispielsweise die Anordnung von Bildern und Text fest.

Eine häufig anzutreffende *Funktionalität* ist zum Beispiel das Generieren einer druckerfreundlichen Seite oder das Anzeigen aktueller Informationen aus einer Datenbank.

Die *Präsentation* sorgt für den grafischen Eindruck der Webseiten, zum Beispiel können Cascading Style Sheets (CSS) die Schrift in Form und Größe für umfangreiche Webseiten einheitlich modulieren.

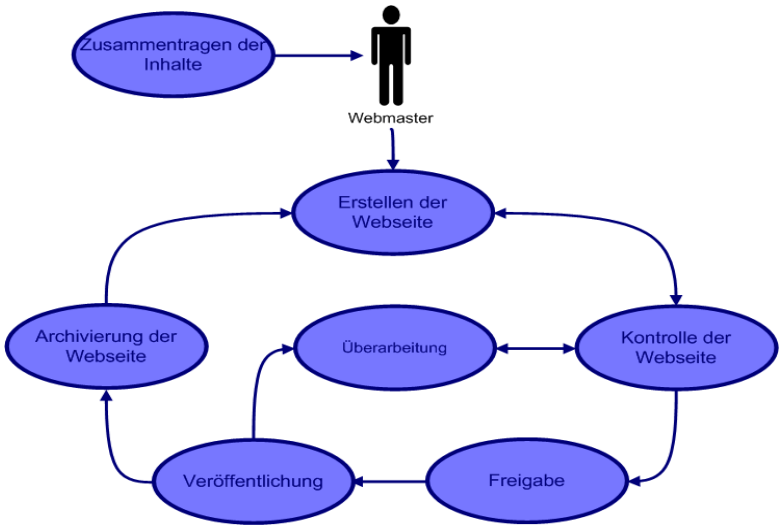

Abb. 4: Eine Webseite durchläuft verschiedene Phasen während ihrer Lebensdauer (Life-Cycle).

Die Lebensdauer (*Life-Cycle*) einer Webseite besteht aus sechs Phasen (Abb. 4):

1. Erstellung,
2. Kontrolle,
3. Freigabe,
4. Veröffentlichung,
5. Überarbeitung,
6. Archivierung (eventuell),

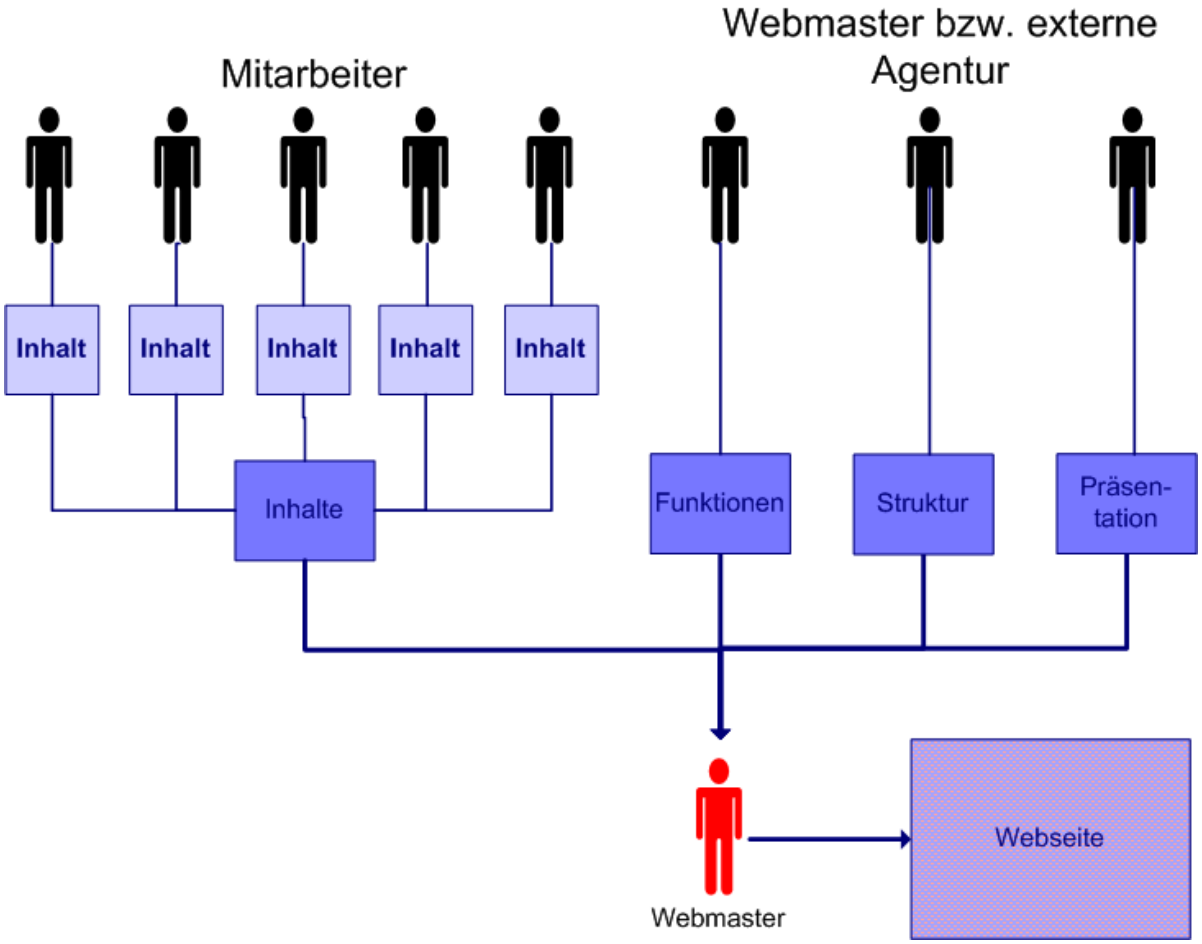

Abb. 5: Der herkömmliche Veröffentlichungsprozess einer Webseite und die zentrale Bedeutung des Webmasters hierbei (angelehnt an [21]).

Die Verantwortung für die Erstellung, Veröffentlichung, Überarbeitung und Archivierung einer Webseite bzw. einer Web-Anwendung obliegt einem Webmaster (Abb. 5). Die Inhalte der Webseite müssen jedoch vom Webmaster in die Webseiten eingearbeitet werden. Der Webmaster und ggf. eine externe Agentur stellen eine Art Flaschenhals dar, weil sie die nötigen Kenntnisse (zum Beispiel HTML-Kenntnisse) besitzen, die für dieses herkömmliche Veröffentlichen von Inhalten notwendig sind [19]. Alle zu veröffentlichenden Informationen laufen bei ihnen zusammen, bevor sie veröffentlicht werden können.

Um einen Nutzwert zu behalten und interessant zu bleiben, müssen die Webseiteninhalte regelmäßig aktualisiert bzw. die Webseite komplett überarbeitet werden (Relaunch). Diese Tätigkeiten verursachen rund 80% der Kosten, die eine Webseite bzw. Webanwendung im Laufe ihrer Existenz verursacht [21, 22]. Der Aufwand der Pflege nimmt mit Umfang der Webseite zu (Diagramm 1, [21]). Die manuelle Pflege ist ab einem bestimmten Umfang der Webseite nicht mehr

ohne Unterstützung durch Werkzeuge zu leisten [21]. Ein solches Werkzeug ist beispielsweise ein Content Management System.

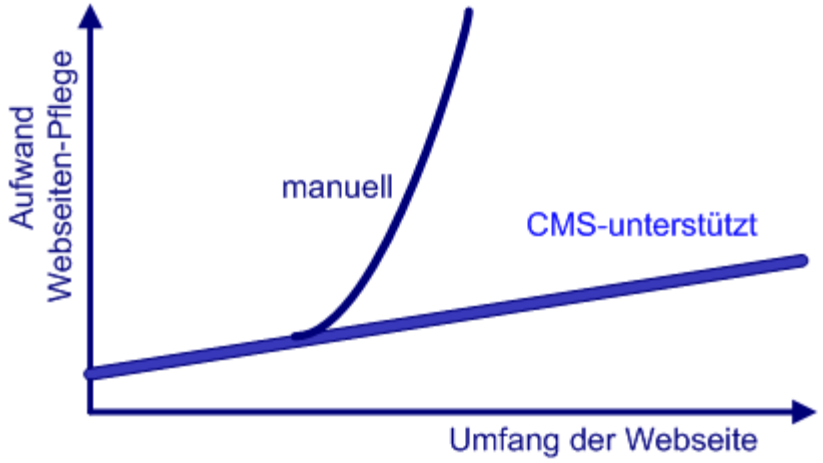

Diagramm 1: Der Aufwand der Webseitenpflege steigt mit dem Umfang der Webseite [21].

3.2 Einführung in Content Management Systeme

Das *Content Management* (CM) befasst sich mit der systematischen Sammlung, Erstellung, Speicherung und Veredelung von strukturierten Inhalten [18]. *Content Management Systeme* (CMS) sind Anwendungen, die im Internet bzw. Intranets das Erstellen, die Kontrolle, die Freigabe, die Publikation und die Archivierung von strukturierten Inhalten erleichtern [19]. Alle Phasen im Lebenszyklus einer Webseite werden von einem CMS unterstützt.

Grundprinzip eines jeden CMS soll die klare Trennung von Webseiteninhalten und Webseitenstruktur sein [22]. Diese Trennung wird mit Vorlagen (*Templates*) erreicht. Diese definieren die Anordnungen von Bestandteilen einer Webseite also die Struktur. Dies hat den Vorteil, große Mengen von Inhalten in einem gleichbleibenden Aussehen (*Layout*) darzustellen. Das einheitliche Auftreten einer Firma durch ein Corporate Design wird dadurch vereinfacht [22]. Der Nachteil von Templates ist die damit einhergehende Unflexibilität, so sind Ausnahmen nur durch Anpassung bzw. Neuerstellung der Templates möglich.

Von besonderer Bedeutung beim Einsatz eines CMS ist, dass nahezu jeder Mitarbeiter Inhalte im Internet bzw. Intranet veröffentlichen kann [22]. Der Prozess ähnelt dem Erstellen von Inhalten mit Office-Produkten. So können Informationen am Ort der Entstehung veröffentlicht werden. Im Sekretariat zum Beispiel laufen viele Informationen (Termine, Hinweise u.s.w.) zusammen. Die Informationen, an dieser Stelle einzupflegen ist sinnvoll. Außerdem kann die Entwicklung

neuer Webseiten (wie zum Beispiel SOP-Webseiten) bzw. deren Überarbeitung von Mitarbeitern durchgeführt werden, die zwar auf ihren Gebiet Fachexperten sind, aber keine speziellen HTML-Kenntnisse besitzen. Der Veröffentlichungsprozess wird vereinfacht, weil mit Hilfe der Templates keine Webseitenprogrammierung mehr benötigt wird. Die Engstelle Webmaster bzw. externe Agentur wird mit einem CMS somit umgangen. Die Inhalte können einfacher und schneller veröffentlicht werden. Die Zeit zwischen der Entstehung der Information bis zur Veröffentlichung (Time-to-Web) kann sich somit wesentlich verkürzen. Dies ist für tagesaktuelle Webseiten von besonderem Interesse.

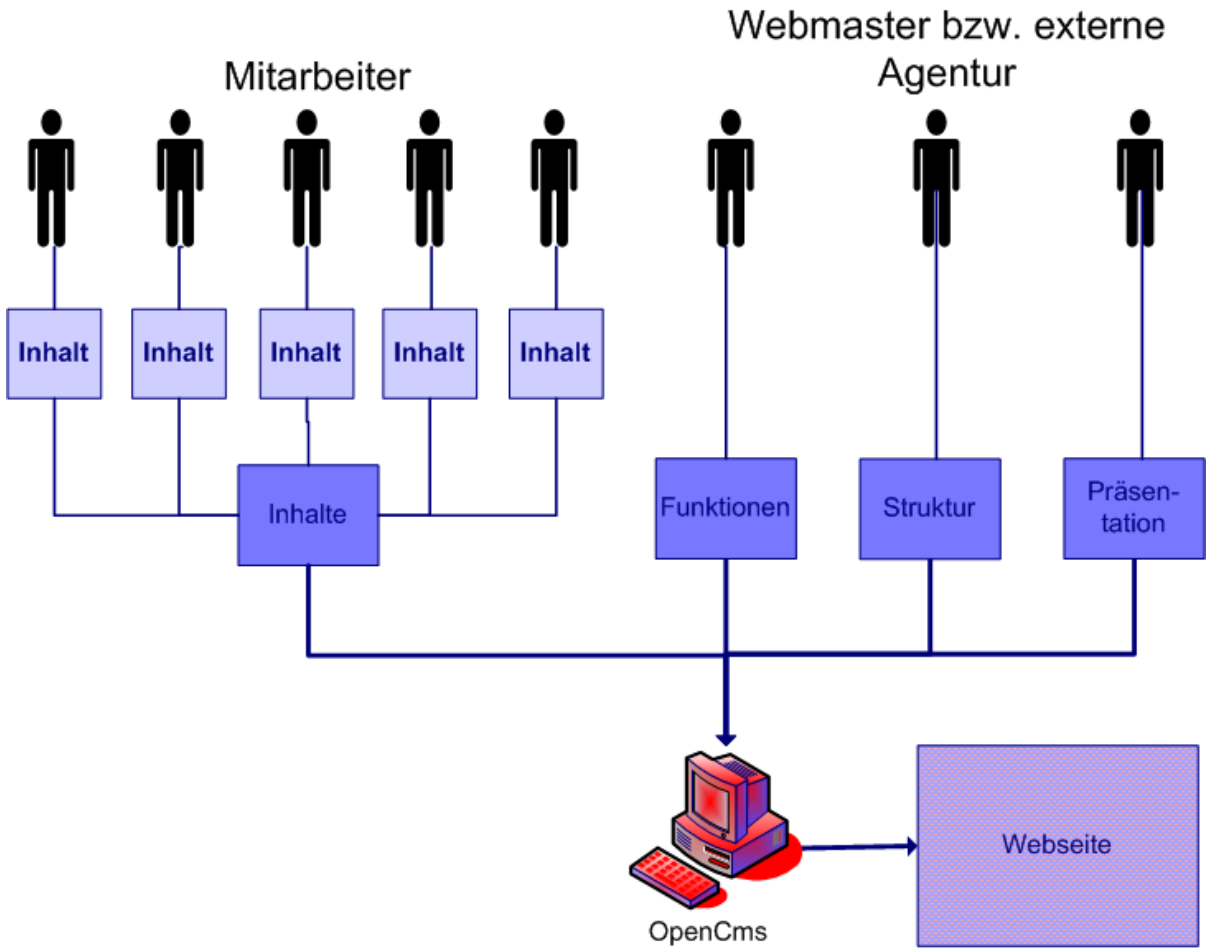

Abb. 6: Der Prozess der Webseiten-veröffentlichung mit Hilfe eines CMS (Anlehnung an [21]).

Die Verwendung eines CMS verhindert auch Konflikte, wenn mehrere Mitarbeiter gleichzeitig an einer Webseite arbeiten [22]. Außerdem werden durch die einheitliche elektronische Kommunikation zwischen den involvierten Rechnern und den Mitarbeitern potentielle Fehlerquellen minimiert [36].

Abb. 6 zeigt die Situation des Webseitenpublizierens mit Hilfe eines CMSs. In der Abbildung sind zwei typische Benutzergruppen dargestellt. Einerseits sind dies die Mitarbeiter, die für den Inhalt, die Kontrolle und die Freigabe der Webseite zuständig sind. Der Webmaster sorgt andererseits für die richtige Struktur, die Präsentation und die Funktionalität, ohne sich inhaltlich Gedanken machen zu müssen. Tabelle 1 fasst die Eigenschaften der beiden Prozesse zusammen.

herkömmliches Webpublishing	**Webpublishing mit einem CMS**
Inhalt und Struktur bzw. Layout einer Webseite ist fest verknüpft.	Inhalt und Struktur bzw. Layout sind getrennt
Meist Webteam oder Webmaster am Ende des Veröffentlichungsprozesses.	Direktes Publizieren auch technisch nicht versierter Mitarbeiter.
Zentralisiertes Arbeiten über Schnittstelle Webmaster.	Dezentrales Arbeiten an einem verteilten System.
Lange Time-to-Web durch Flaschenhals Webmaster.	Kürzere Time-to-Web durch Dezentralisierung.
Überproportional wachsenden Aufwand bei wachsendem Inhalt.	Proportionales Wachstum von Aufwand und Contentmenge.
Manueller Arbeitsablauf.	Automatisierter Arbeitsablauf.
Viele Werkzeuge für Pflege der Webseite.	Automatisierte Webseitenpflege durch CMS.
Technisch nicht versierte Mitarbeiter nicht involviert.	Mitarbeiter entsprechend ihrer Kompetenz eingebunden.
Redesign ist ein aufwendiger Prozess der Umprogrammierung.	Einfacher Relaunch über Änderung der Vorlagen.

Tabelle 1: aus [19]

Die folgenden nicht-repräsentativen Umfrageergebnisse (Stand 12.09.2005)sollen die Bedeutung von Content Management Systemen für die Zukunft verdeutlichen [17].

Wird es langfristig noch große Webseiten ohne Content Management System geben? (abgegebene Stimmen: 966)

Ja	18,94%
Nein	81,06%

Wie wird sich die Anzahl der CMS in Jahre 2005 entwickeln? (abgegebene Stimmen: 437)

wachsen	68,65%
gleich bleiben	9,38%
verringern	21,97%

Wie hoch ist der Bedarf bei kleinen und mittelständischen Unternehmen an Content Management Lösungen? (abgegebene Stimmen: 420)

hoch	45,95%
mittel	32,86%
gering	21,19%

Wie zufrieden sind Sie mit dem Content Management System Ihrer Website? (abgegebene Stimmen: 554)

Sehr zufrieden	32,67%
Größtenteils zufrieden	25,09%
Eher unzufrieden	11,01%
Unzufrieden	9,57%
Kein CMS	21,66%

Die Umfrageergebnisse zeigen, dass die Bedeutung von Content Management Systemen steigen wird und dass rund 50% der Befragten mit ihrem CMS zufrieden sind.

Zusammengefasst kann man festhalten, dass es mit Hilfe von Content Management Systemen möglich ist, rasch Webanwendungen bzw. Webseiten zu erstellen und zu pflegen. Sie helfen dabei die anfallenden Tätigkeiten zu vereinheitlichen, zu automatisieren und somit alle Phasen einer Webseite einheitlich zu steuern. Dies kann der Qualität und Aktualität der Webseite zu gute kommen.

Im nächsten Abschnitt wird ein Content Management System vorgestellt.

3.3 OpenCms

OpenCms ist ein in Java geschriebenes Content Management System unter quelloffener *Lesser GNU Public License* (LGPL). Es ist mit dem Ziel entwickelt worden, umfangreiche Webseiten einfach zu erstellen und zu verwalten [20].

Folgende Softwarevoraussetzungen sind für die Verwendung der aktuellen OpenCms-Version 6.0 notwendig:

1. Java SDK ab 1.4,

2. Java Server Container,
3. Relationale Datenbank,
4. Browser mit deaktivierten Popup-Blocker und aktivierten JavaScript,

OpenCms verwendet folgende Dreischichtenarchitektur (Abb. 7):

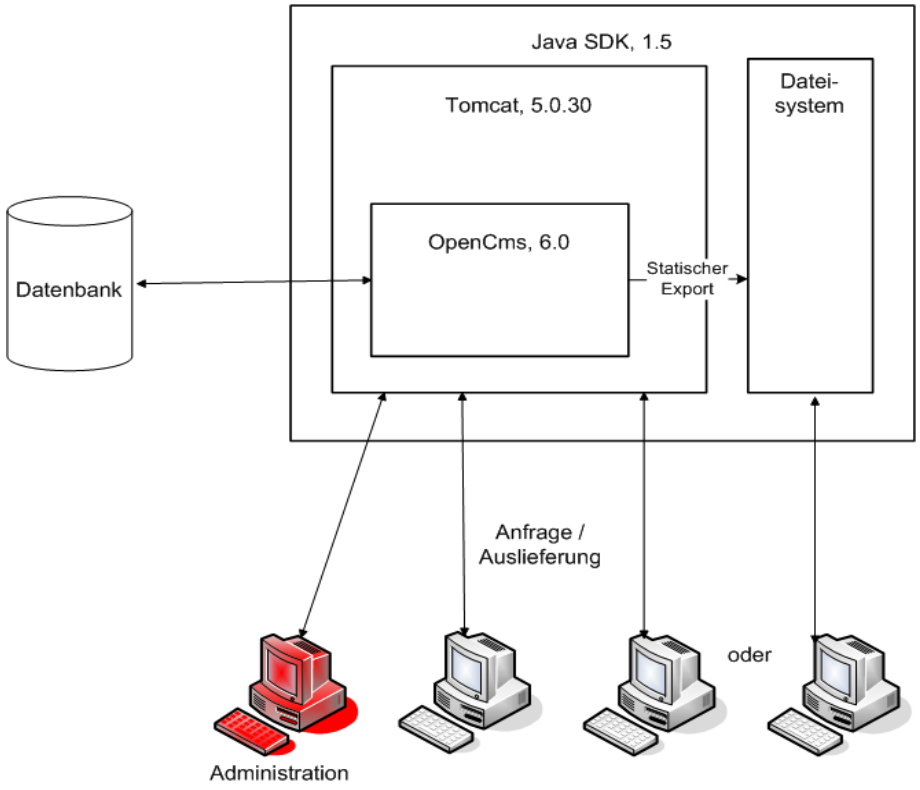

Abb. 7: Die Dreischichtenarchitektur von OpenCms,
(Backend, Middleware, Frontend).

- Über das *Frontend* kann der Anwender mit dem System interagieren, dazu ist auf dem *Clienten* ein Browser (*Internet Explorer, Firefox*) notwendig. Es ist keine weitere Installation auf den Clientrechnern notwendig, der Aufwand für die Wartung wird dadurch stark vermindert. Des Weiteren kann OpenCms über den Browser administriert und die Webseitenpflege durchgeführt werden, dazu bietet OpenCms den *Arbeitsplatz* an.
- Das *Backend* besteht aus einer relationalen Datenbank zur Speicherung aller anfallenden Daten. Es kann aus einer Anzahl Datenbanken frei gewählt werden (zum Beispiel MySQL, Postgres, Microsoft SQL Server, Oracle). Der Webseiten-inhalt wird in der Datenbank als XML-Dateien gespeichert. Die Struktur, Präsentation und Funktionalität einer Webseite und die Benutzerdaten bzw. -rechte werden getrennt davon hinterlegt.

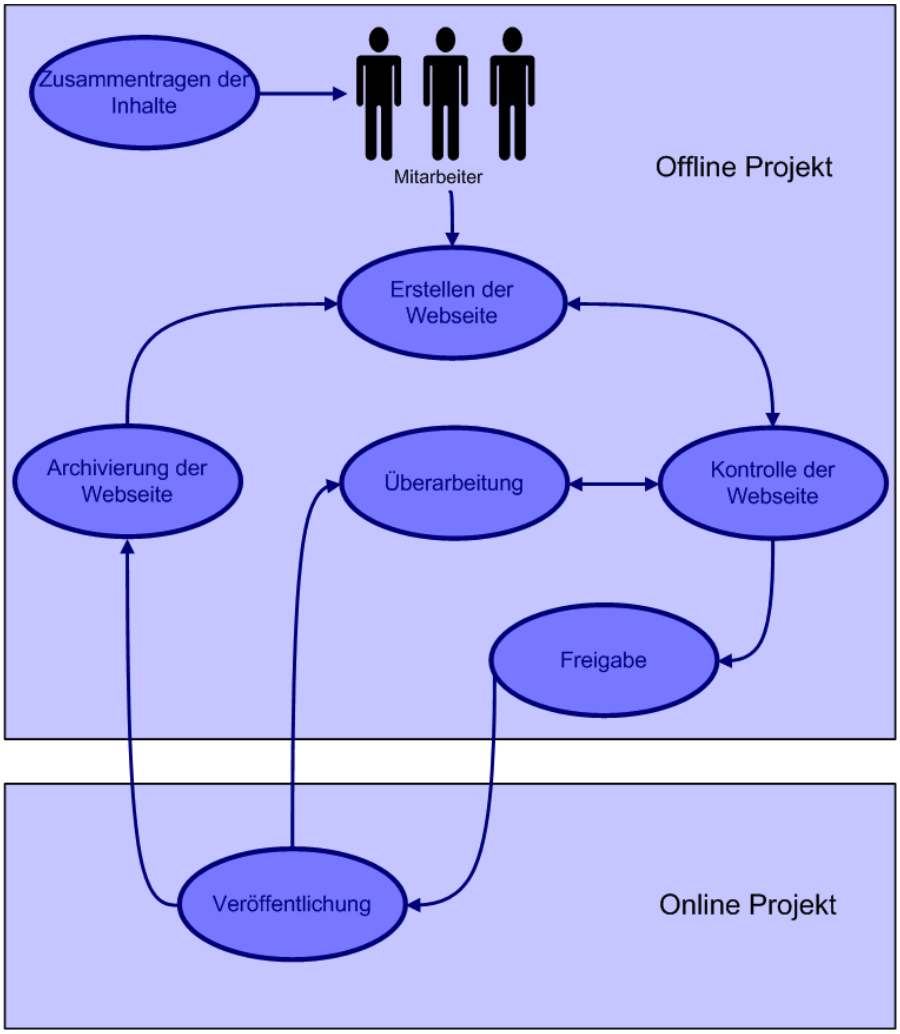

Abb. 8: OpenCms unterscheidet bei der Bearbeitung des Life-Cycles einer Webseite zwei Bereiche: Off-/Online-Projekt.

- Die *Middleware* verbindet Frontend und Backend miteinander. Sie besteht aus dem OpenCms-Programm. Für den Zugriff auf OpenCms-Funktionen steht eine *Application Programming Interface* (API) zur Verfügung, mit der rasch Webanwendungen entwickelt werden können [20]. Da sich OpenCms konsequent an die Java-Standards hält, ist der Einstieg für Javaprogrammierer vereinfacht [52].

OpenCms unterscheidet zwei Dateibereiche (Projekte): online und offline. In dem offline - Projekt werden Webseiten erstellt, überarbeitet, geprüft und getestet, bevor sie durch den Veröffentlichungsprozess in das online - Projekt transferiert werden. Im online - Projekt befinden sich die veröffentlichten und somit im Intranet sichtbaren Webseiten (Abb. 8). Auf beide Bereiche kann man über den

Arbeitsplatz zugreifen, dies ist das administrative Zentrum von OpenCms [20].

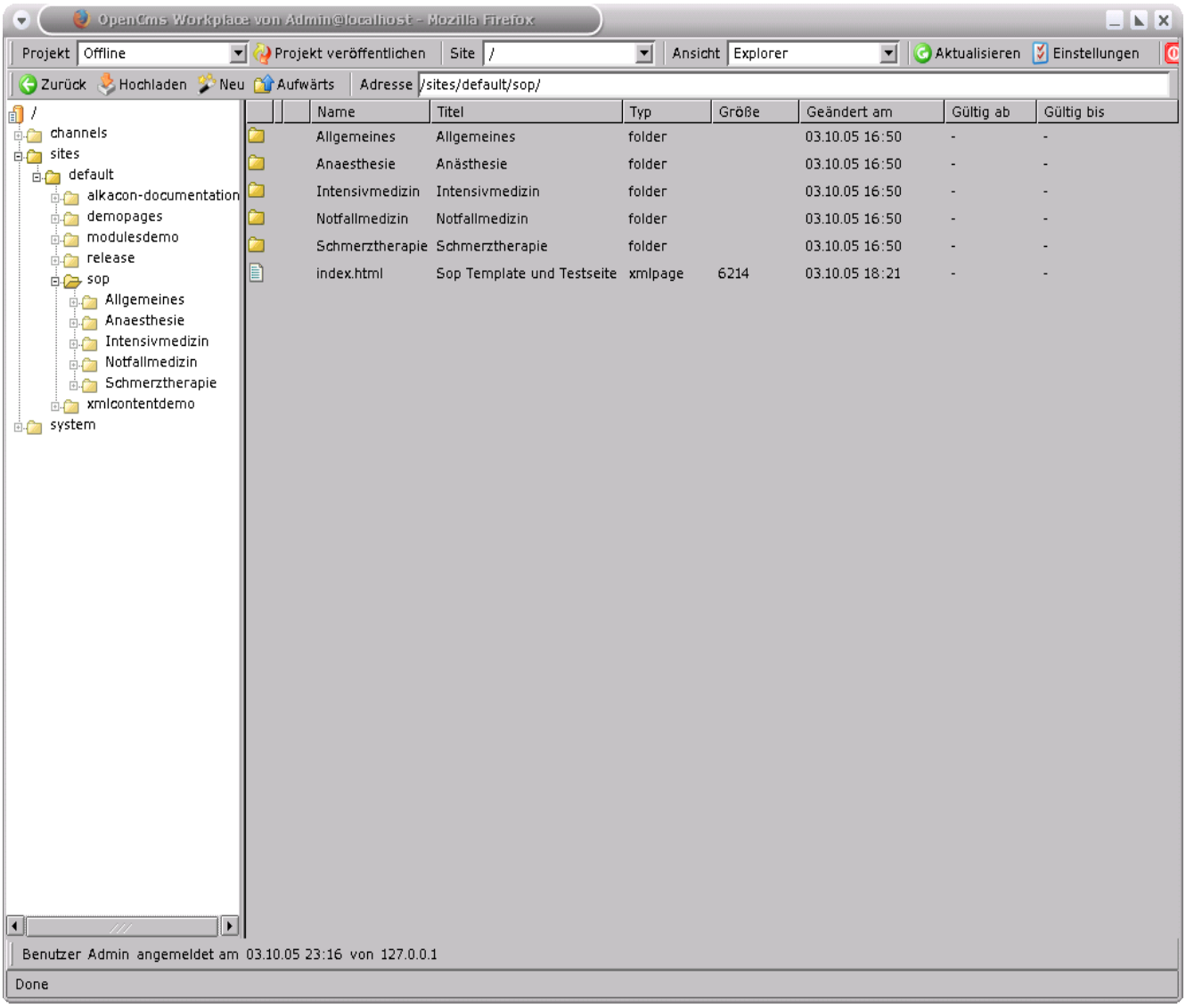

Abb. 9: Die Ordner- und Dateiansicht des OpenCms- Arbeitsplatzes.

Der Arbeitsplatz ist ähnlich dem Windows Explorer aufgebaut (Abb. 9). Auf der linken Seite ist die Ordneransicht, rechts die Dateiansicht. Die Dateien sind jedoch virtuell, das heißt sie werden bei Bedarf aus den einzelnen Webseitenbestandteilen zusammengestellt. Die Entwickler haben dieses Konzept als virtuelles Dateisystems (*virtual file system*, VFS) bezeichnet. Man arbeitet wie mit gewöhnlichen Dateien, was den Umgang mit OpenCms erleichtert. Jedoch wird das Grundprinzip der Trennung von Inhalt, Struktur, Präsentation und Funktionalität eingehalten.

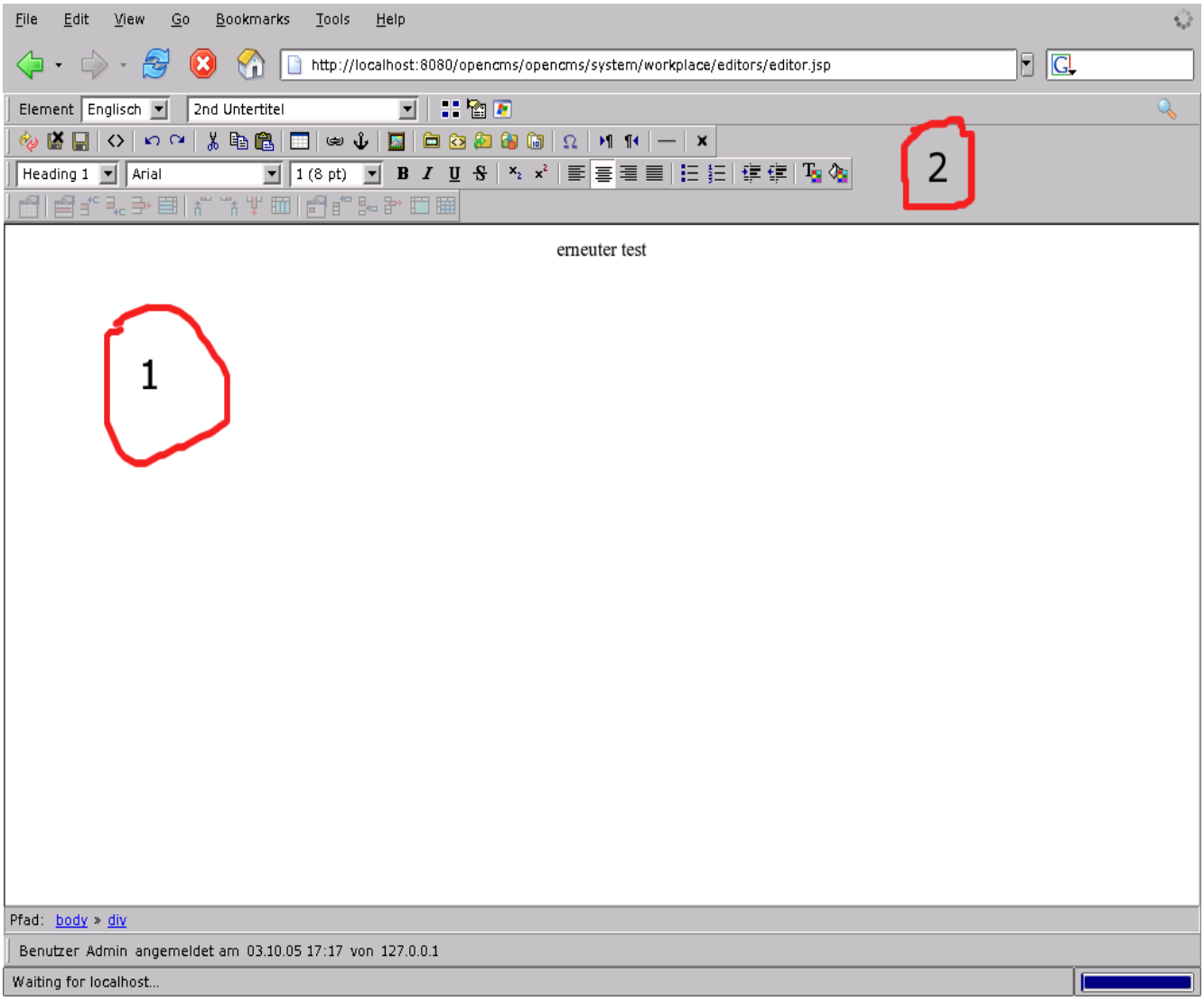

Abb. 10: Mit in dem Browser integriertem WYSIWYG-Editor ist es möglich Inhalte einfach zu erstellen (1: Bereich für Freitext, 2: Funktionsleiste).

Über den Arbeitsplatz kann man mit Hilfe der API neue Templates und Funktionen als Java Server Pages (JSP) erstellen. Das Template in OpenCms stellt einen Rahmen für die Webseiten dar, es enthält im Allgemeinen die Websei tennavigation, Kopf- und Fußzeilen. Neue Funktionalitäten lassen sich sowohl mit der OpenCms-eigenen *JSP-Tag-Library,* als auch mit der *Java-Scriptlet-API* implementieren. Die Installation neuer Module lässt sich ebenfalls über den Arbeitsplatz durchführen. Neue Module[1] können auf einfache Art und Weise die Funktionalität der Anwendung erweitern [20]. Für die Darstellung von strukturierten Webseiteninhalten bietet sich die Verwendung von *XML-Schemas* an, mit diesen lassen sich einfach XML-Dateien generieren [52]. Die Anwendungs-

[1] Einige Module sind zu finden unter „http://www.opencms.org/opencms/en/download/modules/index.html"

entwicklung wird durch die einheitliche Oberfläche (Arbeitsplatz von OpenCms) unterstützt. Diese ermöglicht zum Beispiel neue Funktionalitäten mit dem Texteditor zu erstellen und sofort zu testen, dies ähnelt einer integrierten Entwicklungsumgebung (*IDE*).

Über den Arbeitsplatz kann man neben der Administration von OpenCms auch die Webseiten bearbeiten. Für html-Dateien steht ein extra *WYSIWYG*-Editor zur Verfügung. WYSIWYG steht für „What you see is what you get." und bedeutet: so wie ich die html-Datei (Webseite) jetzt sehe, wird sie später zum Beispiel im Intranet auch aussehen. Die Arbeit mit dem WYSIWYG-Editor von OpenCms ähnelt dem Arbeiten mit bekannten Office-Produkten sehr. In Abb. 10 kann man in den mit 1 kennzeichneten Bereich Texte, Tabellen und Bilder platzieren. Mittels unterschiedlicher Funktionen (Funktionsleiste mit 2 gekennzeichnet in Abb. 10) ist eine einfache Erstellung und Gestaltung von Inhalten möglich.

Für die Bearbeitung von Webseiteninhalten bietet sich noch ein anderer Weg an. Als angemeldeter Nutzer kann man den WYSIWYG- bzw. Texteditor auch direkt über die Webseite aufrufen, diese Funktionalität wird *Direct Edit* bezeichnet.

Zusammenfassend ist zu sagen, OpenCms ist ein professionelles Content Management System, welches sich mit seinen Funktionalitäten nicht vor kommerziellen Produkten zu verstecken braucht. Mit OpenCms ist es vor allen möglich, rasch Webanwendungen zu erstellen und danach einfach zu pflegen.

4. Softwarewartung

4.1 Einführung in die Softwarewartung

Genauso wie eine Webseite gepflegt werden muss, um einen Nutzwert zu haben, muss Software auch gewartet werde. Und so wie der Mensch altert, so altert auch Software [23]. Diese Analogie stimmt bis auf einen Unterschied: Der Alterungsprozess bei Software ist nicht auf Verschleiß oder Abnutzung zurückzuführen, sondern auf Veränderungen in der Umwelt.

Ziel der Softwarewartung ist die Handlungsfähigkeit und Produktivität für den Anwender und deren Arbeitgeber zu erhöhen und die finanziellen Ausgaben zu minimieren. Ian Sommerville sieht Wartung als die Änderung an einem Softwaresystem nach dessen Auslieferung an [24 a]. Die IEEE-Definition (IEEE 1219) fügt hinzu, dass Fehler zu beheben, Performance oder andere Attribute zu verbessern oder Anpassungen an die geänderte Umgebung vorzunehmen sind [25 zitiert nach 45]. Der ISO-Standard 12207 beschreibt Softwarewartung als Änderungen an Software, die auf Grund eines Problems oder Verbesserungsbedarfs notwendig werden [26 zitiert nach 45].
Weidl-Rektenwald untergliedert die Softwarewartung in drei Aspekte [27 a]:

- Wartung als *Tätigkeit* ist hierbei die Änderung des Softwaresystems nach dessen Auslieferung.
- Als *Prozess* hingegen ist Wartung die Durchführung sequenzieller Schritte im Wartungsfall.
- Als *Phase* betrachtet er die Wartung beginnend mit der Auslieferung bis zur Stilllegung der Software.

Balzert trennt hingegen den Begriff der Pflege von dem der Wartung ab [28 a]:

- Pflege reagiert auf Änderungs- bzw. Erweiterungswünsche und ist somit planbar. Änderungen und Erweiterungen können sich hierbei auf alle Teile des Produktes (Produkt-Definition, -Entwurf, -Implementierung) beziehen.
- Wartung in seinem Sinn tritt ein, wenn eine Fehlerwirkung bekannt wird. Sie ist ereignisgesteuert, deswegen nicht planbar. Eine Korrektur des Fehlers im Sinne der Wartung konzentriert sich auf den Quelltext.

Von anderen Autoren wird diese Trennung der Begrifflichkeiten jedoch nicht vorgenommen [23, 27 a]. Ich werde diese Trennung nicht vornehmen. Im englischen Sprachgebrauch wird für die Softwarewartung der Begriff „maintenance" verwendet.

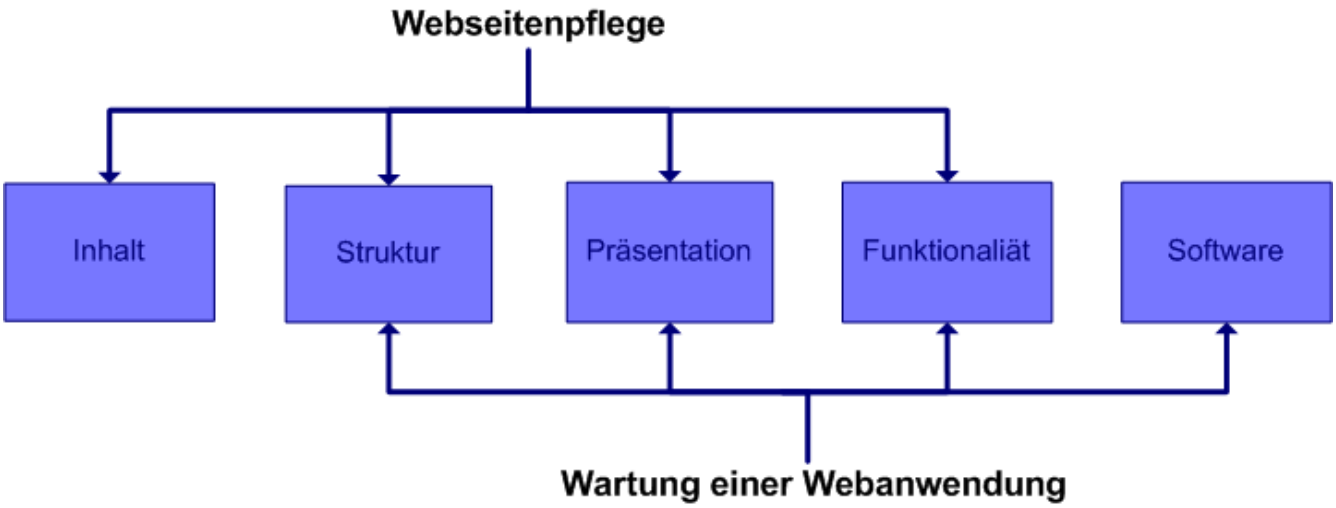

Abb. 11: Unterscheidung Webseitenpflege und Wartung einer Webanwendung.

Um die Bereiche der Webseitenpflege und der Wartung einer Webanwendung klarer unterscheiden zu können, führe ich folgende Unterscheidung ein (Abb. 11):

- *Webseitenpflege* umfasst die Änderung, Erweiterung und Fehlerbehebung in allen Webseitenbestandteilen (Inhalte, Struktur, Präsentation, Funktionalität).
- *Wartung einer Webanwendung* ist eine spezielle Softwarewartung. Sie schließt zusätzlich zur normalen Softwarewartung die Änderung, Erweiterung und Fehlerbehebung von Webseiten in Struktur, Präsentation und Funktionalität mit ein. Aktualisierung von Webinhalten ist nicht Bestandteil der Wartung einer Webanwendung.

Einige Begriff treten häufig im Umfeld der Wartung auf, deswegen will ich kurz auf sie eingehen: Die Softwareevolution beinhaltet alle Änderungen die bei der Entwicklung, Auslieferung, Wartung und der Stilllegung von Software auftreten. *Legacy-Software* bezeichnet Anwendungen, die veraltete Standards verwenden und die aus Kombatibilitäts- bzw. Kostengründen bisher nicht ausgewechselt wurden (in Betrieb meist > 20 Jahre). *Re-Engineering* versucht diese Legacy-Systeme wartbarer zu machen, indem es das System neu dokumentiert, reorganisiert, restrukturiert und in eine moderne Programmiersprache übersetzt. Neue Funktionalitäten werden dabei normalerweise nicht hinzugefügt. *Reverse Engineering* ist dabei ein Analyseprozess, der Systemkomponenten und deren Beziehungen identifiziert und somit Systemdarstellungen in einem höheren Abstraktionsniveau erzeugt. Reverse Engineering modifiziert das System nicht [48].

Nun möchte ich weiter auf die Ursachen für Softwarewartung eingehen. Die Umwelt unterliegt ständigen Änderungen und "Software aging will occur in all successful products" [30]. Deswegen ist Softwarewartung unvermeidlich [27 a]. Die

Ursachen für Softwarewartung können in zwei Bereiche untergliedert werden (Tabelle 2). Der Bereich der menschlichen Faktoren macht rund 20 % der gesamten Wartungsursachen aus [24 a], dabei entstehen Softwarefehler während der Entwicklung und der Wartung. Als Software-Fehler wird dabei jede Abweichung von der ursprünglich geplanten Ausprägung bezeichnet [51].

Menschliche Faktoren		*Veränderungen in der Umwelt*	
Fehler bei der Softwareentwicklung	Fehler durch unsachgerechte Wartung	Veränderungen innerhalb des Systems	Veränderungen außerhalb des Systems

Tabelle 2: Ursachen für Softwarewartung [54].

Anhand der Wartungsursachen kann man auch verschiedene Wartungsarten unterscheiden [28 a].

- Eine Wartung nach dem Auftreten eines Softwarefehlers wird als *korrigierende* Wartung bezeichnet, das Hauptmerkmal ist die reaktive Behebung der Fehler.
- Eine Wartung, die einsetzt bevor Fehler auftreten, wird als *präventive* Wartung bezeichnet. Sie sucht nach Fehlern, bevor sie Auswirkungen zeigen und bessert diese Fehler aus. Sommerville unterscheidet nicht zwischen reaktiver und präventiver Wartung [24 a].
- Durch Änderung der Umgebung (zum Beispiel neues Betriebssystem bzw. neue Geschäftsprozesse) wird die *adaptive* Wartung benötigt. Die Bedeutung dieser Art der Wartung wird mit der Dauer des Softwareeinsatzes immer größer.
- Die *perfektionierende* Wartung beschäftigt sich mit der Verbesserung von Software. Die Parameter Performance, Wartbarkeit und Dokumentation können verbessert werden, aber auch funktionale Erweiterung und neue Leistungsmerkmale können hinzufügt werden. Diese Art der Wartung macht bis zu 65 % aller Wartungen aus [24 a].

In der Praxis ist eine solch klare Trennung der Wartungsarten meist nicht möglich [24 a]. Zum Beispiel, wenn man eine Software an ein neues Betriebssystem anpasst, werden die Vorteile des neuen Betriebssystems oft mitbenutzt (adaptive und perfektionierende Wartung).

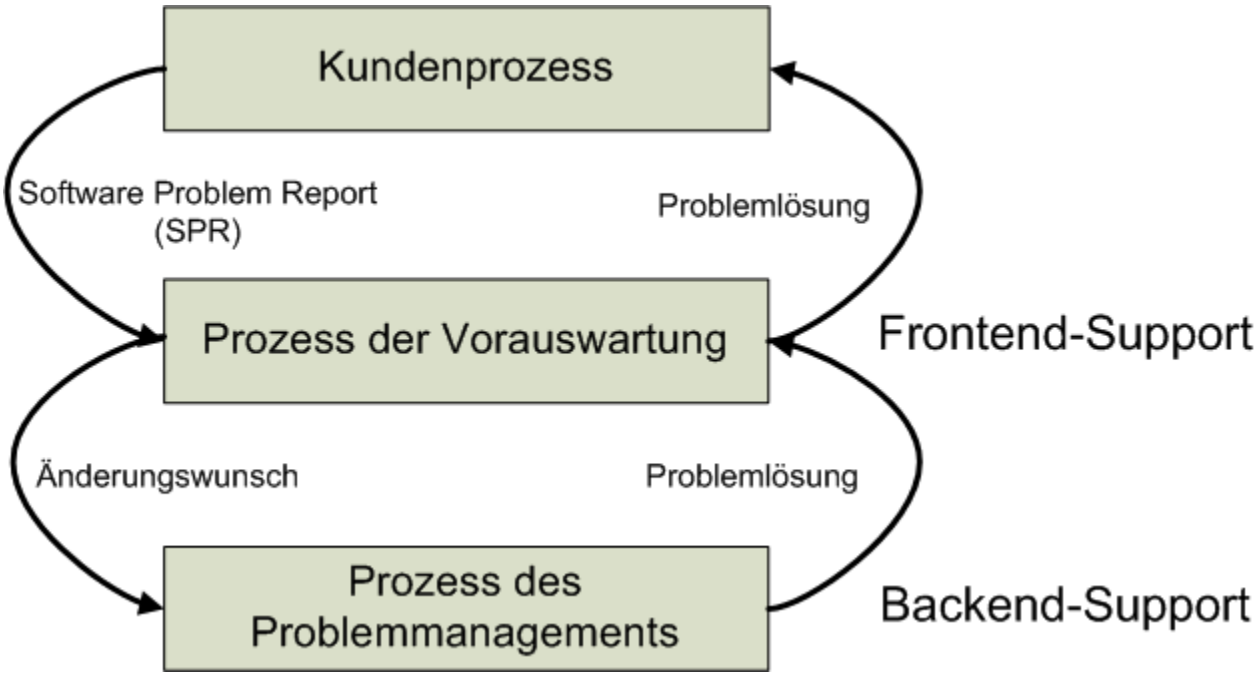

Abb. 12: Der Aufbau des Support-Systems im CM3-Wartungmodell, das Frontend-Support-Team nimmt alle Problemanfragen an und leitet schwierige Anfragen ans Backend-Support-Team weiter (in Anlehnung an [31]).

Es existieren verschiedene Modelle der Softwarewartung, auf die ich im Weiteren eingehen möchte:

1. *Corrective Maintenance Maturity Model* [45, 53, 65]
   Dieses Modell von Kajko-Mattsson ist auf die korrektive Wartung zugeschnitten. Der CM^3-Wartungsprozess ist organisatorisch in 3 Ebenen unterteilt (Abb.12), auf oberster Ebene ist der Anwender (Kundenprozess), der sich im Problemfall an einen Servicemitarbeiter (Prozess der Vorauswartung) wendet. In der Charité exisitert dafür ein HelpDesk. Hier wird versucht das Problem zu lösen, wenn dies auf dieser Ebene nicht möglich ist, wird der Wartungsmitarbeiter mit dem Änderungswunsch (*Modification Request*) kontaktiert.
   Es unterscheidet 7 Prozesse in der Softwarewartung:

   - *Prozess vor der Softwarefreigabe bzw. des nächsten Releases* soll die Softwarewartung während der Softwareentwicklung unterstützen.
   - *Übergangs-/Auslieferungsprozess* soll den reibungslosen Übergang von Entwicklung zu Wartung organisieren.
   - *Prozess des Problemmanagement* beinhaltet das Berichten, Analysieren und Lösen von Softwareproblemen.

- *Testprozess* testet die vorgenommen Änderungen.
- *Documentationsprozess* fertigt eine konsistente Dokumentation an.
- *Prozess der Vorauswartung* (Upfront Maintenance Process) organisiert die Kommunikation zwischen Endnutzer und Wartungsmitarbeiter.
- *Schulung und Training der Wartungsmitarbeiter* sorgt für die kontinuierliche Weiterbildung der Wartungsmitarbeiter.

2. *Quick-Fix-Model* [54]
   Dieses Modell der Softwarewartung wird unter Zeitdruck verwendet, es arbeitet die Veränderungen schnellstmöglich in den Quelltext ein. Die Langzeiteffekte dieser Änderungen werden meist nicht analysiert, ebenso werden diese Veränderungen oft nicht dokumentiert [27 c, 39]. Mit jeder Wartung nach dem Quick-Fix-Model wird das System unübersichtlicher.

3. *Iterative-Enhancement-Model* [54]
   Hierbei werden alle Phasen der Softwareentwicklung im Wartungsfall komplett durchlaufen, wobei jedoch vorausgesetzt wird, dass eine vollständige Dokumentation vorhanden ist. Diese beiden Modelle werden in Abb. 13 gegenüber gestellt.

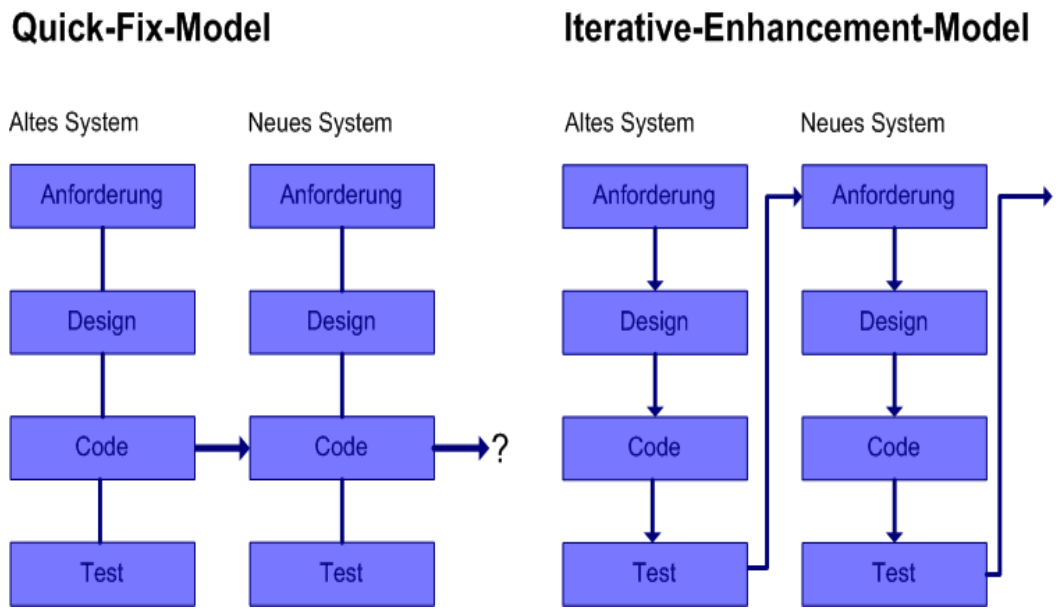

Abb. 13: Das Quick-Fix-Model und das Iterative-Enhancement-Model der Softwarewartung schematisch gegenübergestellt [54].

4. *Modell nach Boehm [39]*
   Mehr auf ökonomischen Modellen und Prinzipien basiert das Modell von Boehm (Abb. 14). Dies soll den Wartungsprozess besser verständlich machen. Im Mittelpunkt steht der Manager, der aus einer Menge vorgeschlagener Lösungen die bewährten Lösungen aussucht [39].

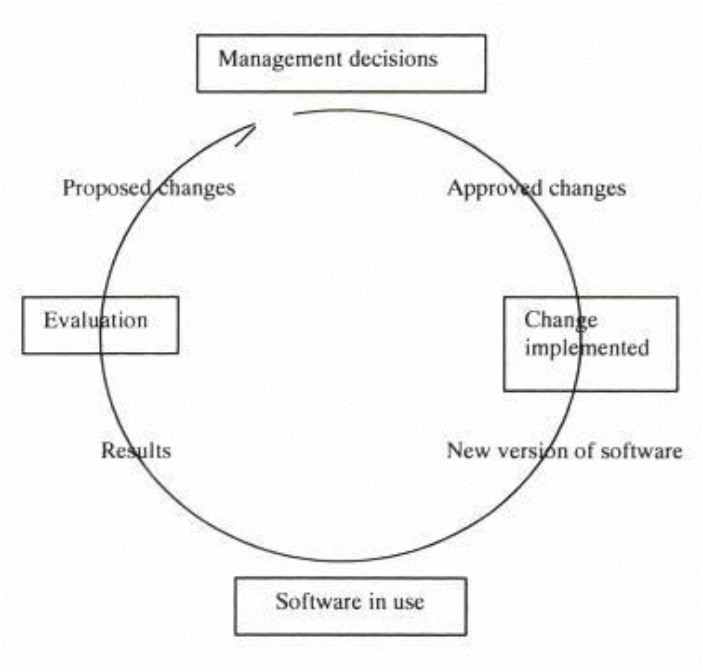

Abb. 14: Die Managemententscheidung steuert den Wartungszyklus im Boehm'schen Modell, aus [39].

5. *Modell nach Osborne [39]*
   Dies ist mehr an der Realität orientiert, so setzt es keine vollständige Dokumentation voraus. Es geht von der Auffassung aus, dass die meisten Wartungsprobleme durch eine inadäquate Kommunikation und Kontrolle des Managements entstehen. Dieses Wartungsmodell schafft in jeden Stadium des Software-Life-Cycles wenn notwendig Voraussetzungen, die die Wartbarkeit erhöhen (Abb. 15). Folgende Forderungen stellt das Osborne Modell [39]:
   - Wartungsanforderungen sind mit einzubeziehen in die Änderungsspezifikationen.
   - Qualitätssicherungsprogramme einsetzen.
   - Wartungsziele erreichen.
   - Als Feedback für die Manager Performancereviews durchführen.

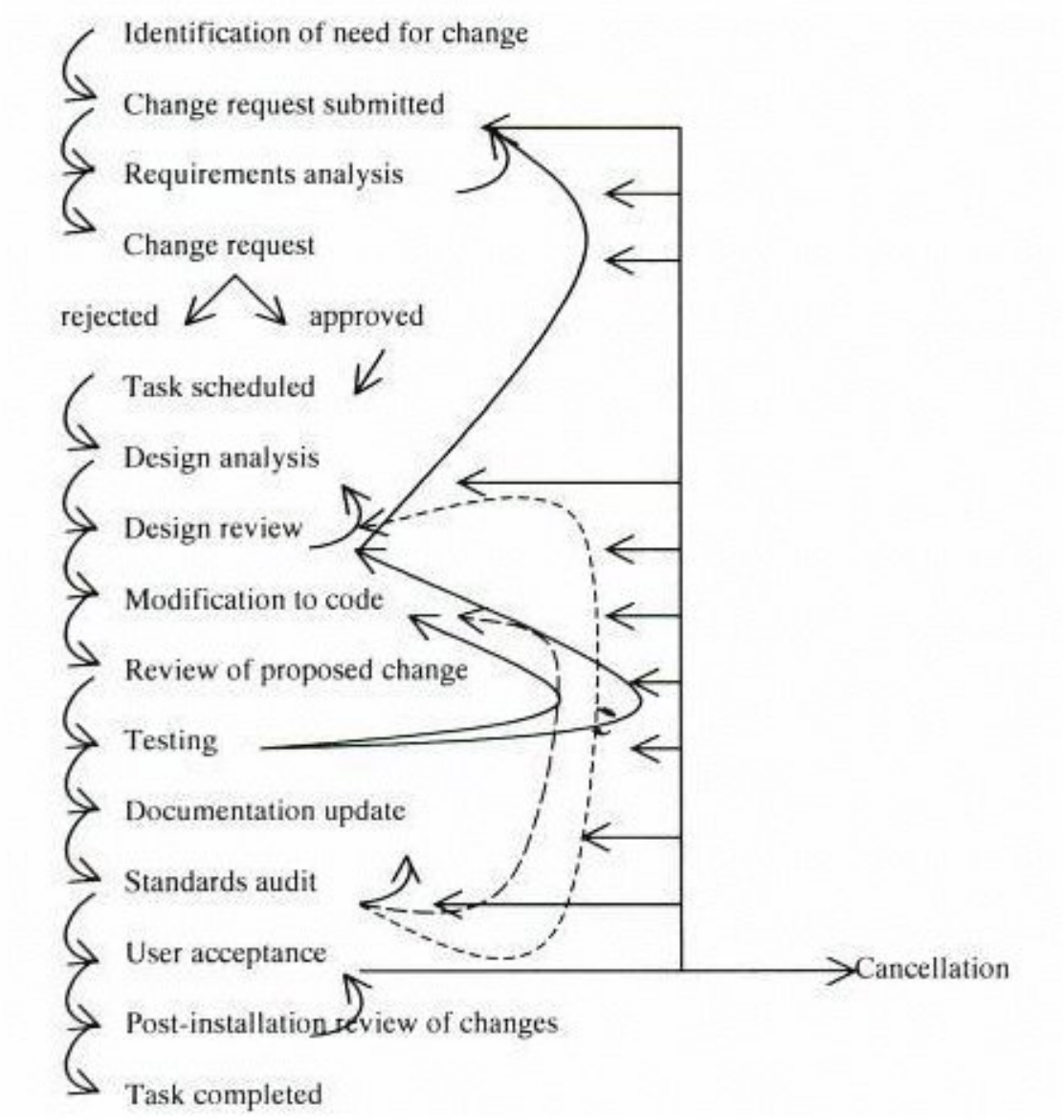

Abb. 15: Das Osborne Modell führt in den Stadien der Software-entwicklung jeweils Tätigkeiten zur Erhöhung der Wartbarkeit durch, aus [39].

6. *Reuse-Oriented-Model* [39]
   Dies bezieht die Wiederverwendung der vorhandenen Programmbestandteile mit ein. Basili beschrieb 1990 dafür vier Schritte :
   - Identifizierung von Softwarebestandteilen des alten Systems, die wieder verwendet werden können.
   - Verstehen der Bestandteile.
   - Die alten Bestandteile den neuen Anforderungen anpassen.

- Integration der geänderten Bestandteile in das neue System.

7. V-Modell nach Harjani und Queill [45]
   Durch einen Trigger in Form einer Problembenachrichtigung wird eine erste grobe Analyse des Problems durchgeführt. Wenn das betroffene Teil identifiziert ist, werden verschiedene Lösungsansätze entworfen und auf Machbarkeit und Auswirkung analysiert. In Abhängigkeit von den erwarteten Auswirkungen wird eine Lösung ausgesucht, implementiert und getestet.

8. IEEE Standard for Software ISO/IEC 14764: Information technology Software maintenance [45]
   Dieser Standard der ISO/IEC ist eng verbunden mit dem Standard 12207 der ISO/IEC. Folgende Wartungsaktivitäten bestimmt der Standard näher:
   - Prozesseinführung, wird einmal zu Beginn der Wartung umgesetzt.
   - Die Migration oder die Ablösung der Software erfolgt einmal am Ende des Wartungsprozesses.
   - Problem- und Änderungsanalyse, Änderungsumsetzung und Review und Abnahme iterativ bei jeder Änderungsanfrage durchlaufen.

Nachdem einige theoretische Modelle vorgestellt wurden, soll das Umfeld der Wartungsaktivitäten betrachtet werden, diese sind [27 b]:

- Analyse bzw. Planung der Änderung
  - „Verstehen der Software" (System, Architektur, Quelltext)
  - Bilden von Hypothesen,
  - Verifizieren von Hypothesen,
  - Analyse der Folgeerscheinungen durch die Änderung,
- Implementierung der Änderung
  - Restrukturierung,
  - Änderungen einarbeiten,
- Verifikation und Validierung
- Re-Dokumentation

Der zeitlich aufwendigste Anteil (50% nach [27 b]) ist dabei das „Verstehen der Software" (Abb. 16). Dieser Aufwand ist jedoch gerechtfertigt, denn jede sachgerechte Wartung beruht auf einem gründlichen Verständnis der Software [23].

Softwarewartung ist unbestritten teuer, bis zu 2/3 der Gesamtsoftwarekosten (Abb. 17) müssen dafür aufgebracht werden [24 a, 28 a, 32].Einige Gründe für den hohen Anteil der Wartungskosten an den Gesamtkosten seien zusammengetragen [24 a, 32]:

- Die Produktivität der Softwarewartung ist geringer als bei der Softwareentwicklung. Dies ist darauf zurückzuführen, dass sich das Vertraut machen mit einer Software viel Zeit kostet („Verstehen der Software"). Hinzu kommt, dass die Wartung meist nicht von den Entwicklern der Software

durchgeführt wird, was die Einarbeitung erschwert.

- Mitarbeiter, die die Wartung übernehmen sind oft unerfahren und kennen sich mit dem Einsatzbereich (Domain) der Software nur wenig aus.

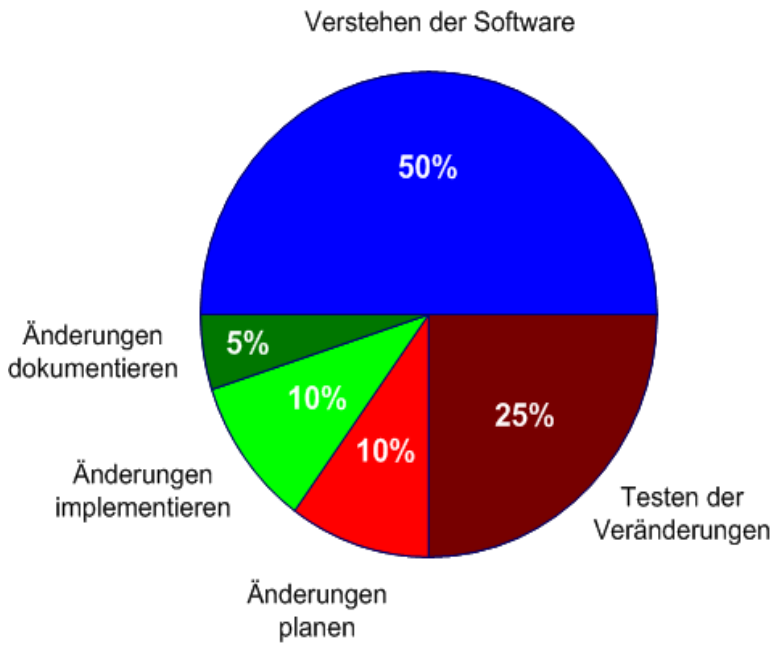

Abb. 16: Die zeitlichen Anteile der Wartungsaktivitäten nach [27 b], man beachte den hohen Anteil für das Verstehen der Software.

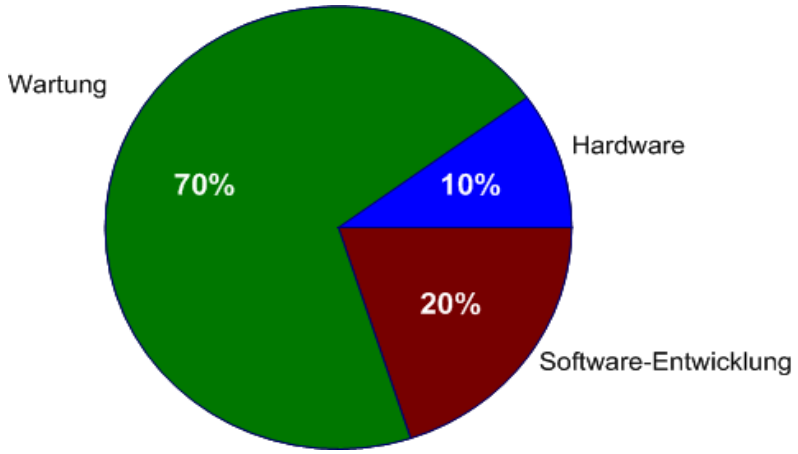

Abb. 17:Softwarewartung verursacht 70% der Kosten im Laufe des Life-Cycles [24, 32].

- Legacy-Software sind mit Programmiersprachen und Techniken entwickelt

worden, die heute nicht mehr verwendet werden und für die Experten rar sind.
- Es stehen selten integrierte Wartungsumgebungen (IME *integrated maintenance enviroment*) zur Verfügung. Die Produktivität ist dadurch geringer. Es ist besonders aufwendig zu analysieren, welche Auswirkungen die Veränderungen am Quelltext nach sich ziehen werden.
- Die Dokumentation ist oft in einem schlechten Zustand, unvollständig oder gar nicht vorhanden.
- Jede Wartungstätigkeit erhöht die Komplexität der Software [42].
- Änderungen in ein laufendes System einzuarbeiten, ist sehr viel aufwendiger als die gleichen Änderungen während der Entwicklung vorzunehmen. Man geht von einem Verhältnis der Kosten von 100:1 (Einarbeitung in laufendes System zu Einarbeitung bei Entwicklung) aus.

Aus Rechtssicht besteht noch keine einheitliche Auffassung darüber, ob Entwicklerfirmen verpflichtet sind, Wartungsverträge anzubieten. Bei besonders hochwertiger Software kann sich jedoch aus §242 BGB eine Verpflichtung zum Abschluss eines Softwarewartungsvertrages ergeben [44]. Wenn Software für den Kunden jedoch einen dauerhaften Nutzen darstellen soll, muss sie kontinuierlich gewartet und weiterentwickelt werden [23]. Die Entwicklung einer neuen Anwendung ist nicht mit der Übergabe an den Kunden beendigt, sondern setzt sich unter geänderten Bedingungen fort [24 a].

4.2 Verbesserung der Wartbarkeit

Balzert sieht zur Verbesserung der Wartung grundsätzlich zwei Wege [28 a]:

1. Produkte mit hoher Qualität entwickeln und freigeben,
2. Erhöhung der Wartungsproduktivität,

Der Softwarewartung muss schon in der Entwicklungsphase Rechnung getragen werden, damit die spätere Wartung erleichtern wird [24 a]. Zum Beispiel kann dies durch die Konzepte der Objektorientierung und der Abstraktion erreicht werden [30]. Gleichzeitig hilft eine gute Dokumentation des Quelltextes dem "Verstehen der Software" und verkürzt somit die Einarbeitungszeit [32]. Frühzeitig ist mit der Erstellung der Dokumentation zu beginnen. Sonst besteht die Gefahr, dass wegen Termindruck die Dokumentation am Ende unterbleibt. Das "Verstehen der Software" kann außerdem durch Anwendung von Kodierkonventionen erleichtert werden, als Beispiele sind einheitliche Formatierung des Quelltextes und sinngemäße Variablenbezeichnung aufzuführen [34].

Durch intensives Testen werden Fehler gefunden, die vor der Auslieferung behoben werden können. Wiederverwendung von Software (*reuse*) hat ebenfalls eine geringere Fehlerrate zu Folge, außerdem führt es zu niedrigerem Wartungsaufwand [24 b]. Zu einer geringeren Fehlerrate führt auch das Einholen einer zweiten Meinung zum Quelltext (Codereview). Die Effizienz und die Strukturierung der Software kann somit ebenfalls erhöht werden [30].

Neben der Erhöhung der Softwarequalität spielt die Wartungsproduktivität eine wichtige Rolle. Die Wartungsproduktivität wird durch die Organisation des Wartungsfalls wesentlich mitbestimmt [28 a]. Balzert bezeichnet dies mit „Konfigurations- und Änderungsmanagements". Er führt darin folgende Punkte auf [28 a, 45]:

- Die Annahme und Verwaltung eingehender Fehlermeldungen bzw. Änderungswünschen (Trigger) muss organisiert werden, bevor der weitere Ablauf entschieden wird.
- Es müssen Überlegungen angestellt werden, ob und wie die Wartungsaktivitäten wirkungsvoll durchgeführt werden können.
- Durchführung der Wartung mit darauf folgender Testung der Änderungen.
- Alle Betroffenen sind zu informieren.

Da Softwarefehler und Änderungswünsche jederzeit auftreten können, müssen die Organisationsstrukturen so angepasst werden, dass sie darauf vorbereitet sind. Zum Beispiel müssen im voraus Wartungsverantwortlichkeiten fest gelegt sein und Gelder für die Softwarewartung auch eingeplant werden [32].

Die Wartungsproduktivität kann durch integrierte Wartungsumgebungen erhöht werden, der Umgang damit muss jedoch auch geschult werden [29].

Der Ausfall der Software im Produktivbetrieb kann hohe Kosten verursachen, was zu einem hohen Termindruck führt. In der Praxis geht man davon aus, dass Wartung immer zeitkritisch ist[23]. Dabei kann es durch übereilte, fehlerhafte Wartungsmaßnahmen zu weiteren Fehlern kommen. Dem soll ein gut eingeführter Wartungsprozess vorbeugen.

5. Ergebnisse der Softwareentwicklung

Der erste Prototyp der Webanwendung konnte in der 4. Woche fertig gestellt werden. Jedoch ging der Entwicklungprozess langsamer als geplant voran. Ich hatte einerseits den zeitlichen Aufwand für das „Verstehen der Software" unterschätzt, andererseits war die OpenCms-Dokumentation nicht sehr hilfreich(Ausnahme [52] und Mailingliste). Die Entwicklung der Webanwendung wird nach Abgabe der Bachelorarbeit noch weitergeführt.

Im Anhang befindet sich das Pflichtenheft (Anhang C) und eine Übersicht über zurzeit auf dem Markt angebotener SOP-Programme (Anhang D).

Die Anwendung soll nun anhand einiger Bilder vorgestellt werden.
Abb. 18 zeigt den Administratorbereich. Neue Verfasser können hier im System angelegt werden bzw. deren Einträge bearbeitet werden (siehe 1 in Abb. 18).

Über den Punkt 2 in Abb.18 erreicht man den Bereich, indem ein neuer SOP-Eintrag erstellt werden kann. Über den Punkt 3 können Verfasser oder Mitglieder des Gremiums den Status eines SOP-Eintrags verändern.

Abbildung 19 zeigt die benötigten Angaben, um einen neuen SOP-Eintrag anzulegen. Punkt 1 fragt in welchen Ordner der neu angelegte SOP-Eintrag gespeichert werden soll. Punkt 2 fragt den Titel des SOP-Eintrags ab. Dieser wird manchmal zu lang sein, um im Menü zu erscheinen. Deswegen wird in Punkt 3 der Abb. 19 der Titel in Kurzform abgefragt. Die Emailadresse des Verantwortlichen soll in Punkt 4 angegeben werden. Die Emailadressen der Verfasser des SOP-Eintrags wird im Punkt 5 eingetragen.

Die Verfasser und der SOP-Verantwortliche werden per Email informiert, nachdem diese Informationen durch ausführen von „Absenden" gespeichert werden. Die Email enthält unter anderen den Link auf die zu bearbeitende SOP. Zusätzlich sind Einstiegshilfen der Email angefügt. Durch Verwendung der Emailadresse wird die Authentizität der Anwender sichergestellt. Im Punkt 6 der Abb. 19 können noch Bemerkungen bzw. Hinweise an die Verfasser hinzugefügt werden.

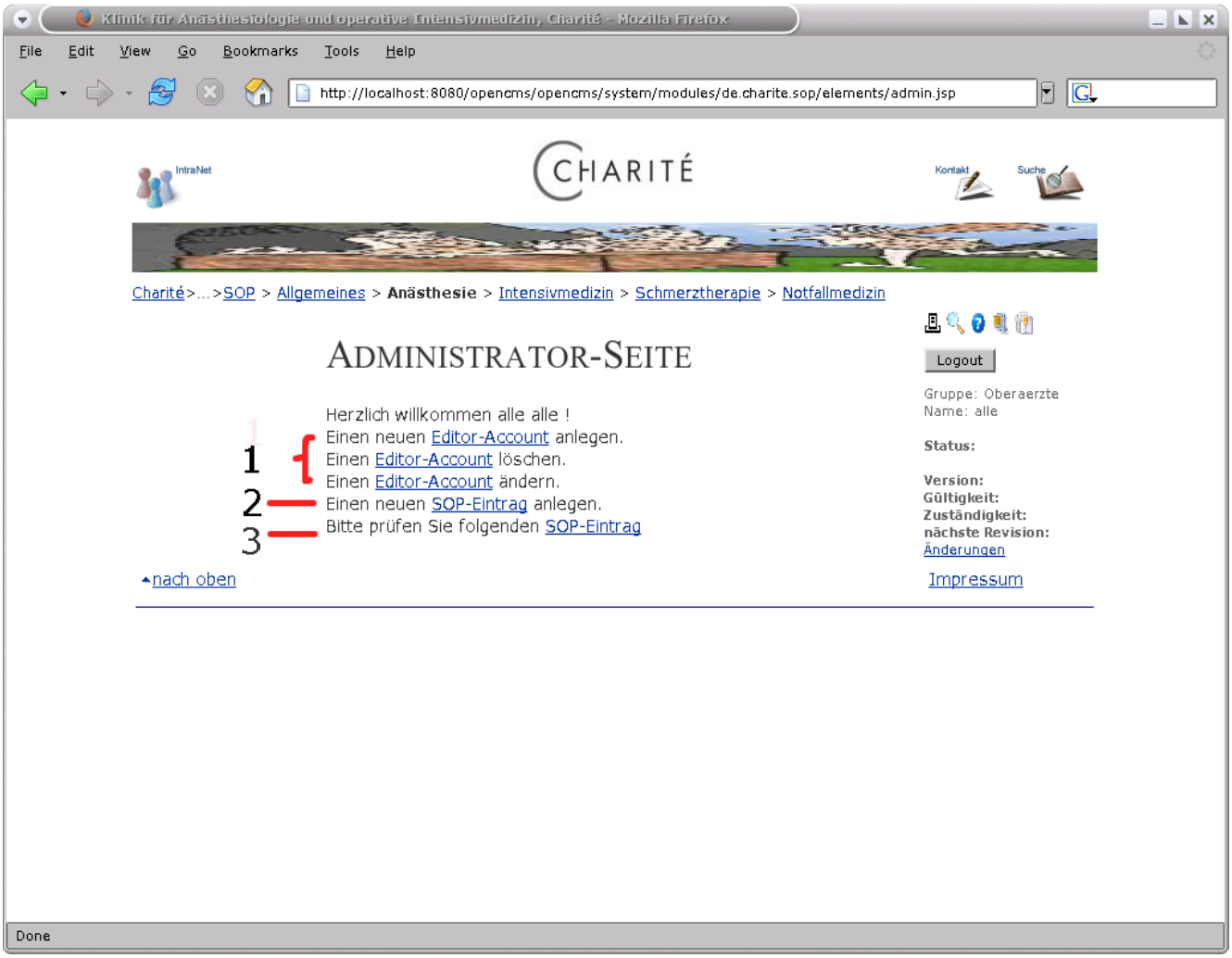

Abb. 18: Administratorbereich (1: Nutzerverwaltung, 2: Neue SOP anlegen 3: SOP-Eintrag prüfen).

Die Verfasser erhalten per Email die nötigen Informationen, um eine neue SOP zu erstellen. Sie bekommen das Passwort für das neu erstellte Benutzerkonto, einen direkten Link zum begonnenen SOP-Eintrag und einige Einstiegshilfen. Die Verfasser können nun den Text beliebig oft überarbeiten.

In Abb. 20 ist eine Beispiel-SOP dargestellt, die mit Hilfe der Anwendung entstanden ist. Mit dem Punkt 1 ist das Menü gekennzeichnet. Dieses wird dynamisch erstellt, so dass neu angelegte SOP-Einträge bzw. Ordner automatisch darin erscheinen. Um so wenig wie möglich von den Inhalten der SOPs abzulenken, klappt das ganze Menü erst bei Mauskontakt auf.

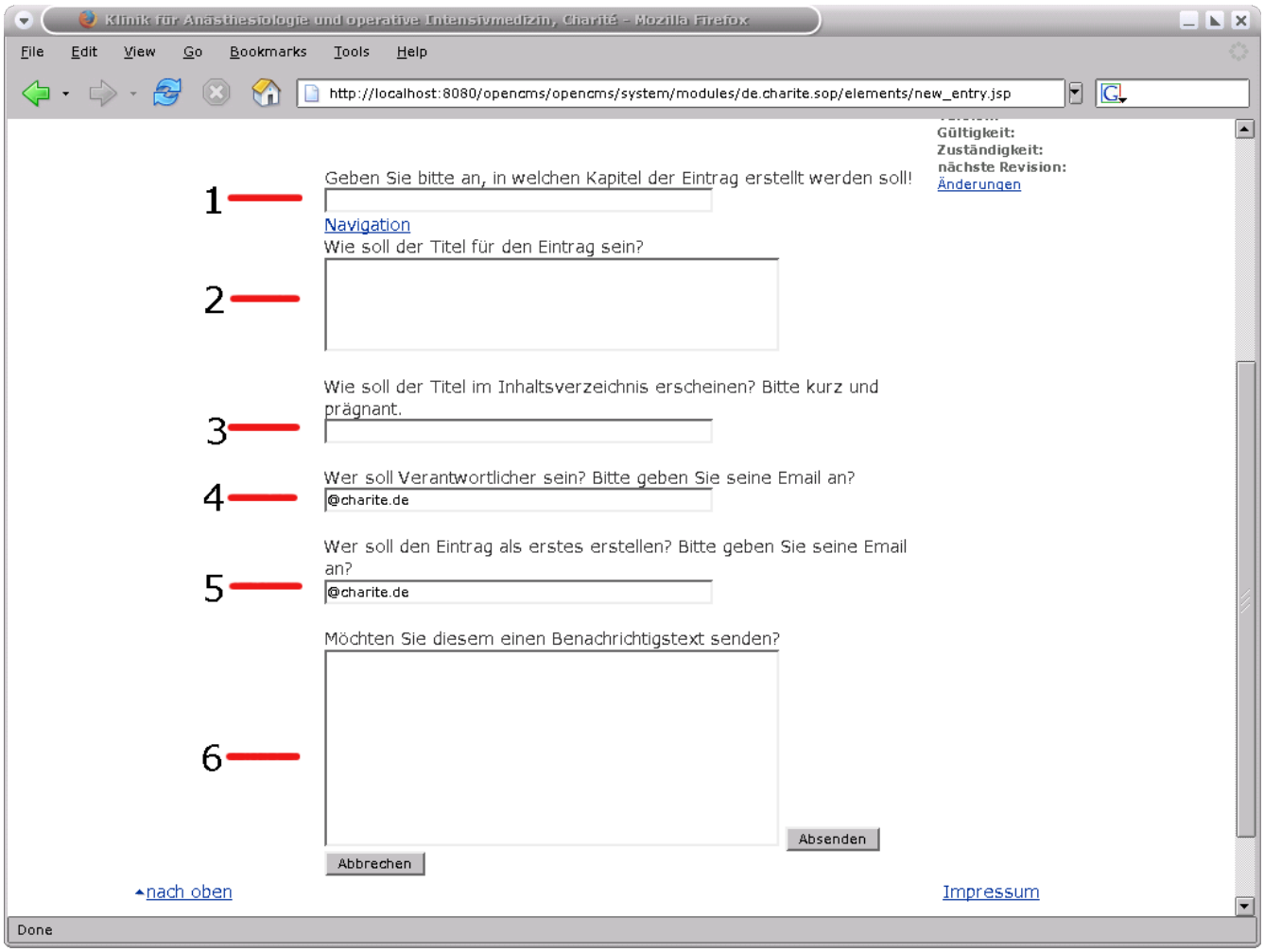

Abb. 19: Einen neuen SOP-Eintrag anlegen kann man im Adminstratorbereich (1: Speicherort, 2: Name der SOP, 3: Name im Menü, 4: Emailadresse des Verantwortlichen, 5: Emailadressen der Verfasser, 6: Hinweise an Verfasser).

Mit Punkt 2 der Abb. 20 sind eine Reihe von Funktionen bzw. Zusatzinformationen gekennzeichnet. Von links nach rechts sind dies:

- Druckversion der Seite,
- Suche in den SOPs,
- Hilfe für den Anwender (Beispiel im Anhang E),
- Medizinisches Abkürzungsverzeichnis,
- Zugang zum administrativen Bereich (Abb. 18),

Der mit Nummer 3 gekennzeichnete Knopf dient zum Editieren der Seite (sogenanntes Direct-Edit) und ist nur für angemeldete Benutzer sichtbar. Durch Betätigen des Buttons gelangt man zu dem WYSIWYG-Editor. Wenn sie ihre Bearbeitung abgeschlossen haben, ändern sie den Status des SOP-Eintrags von „in

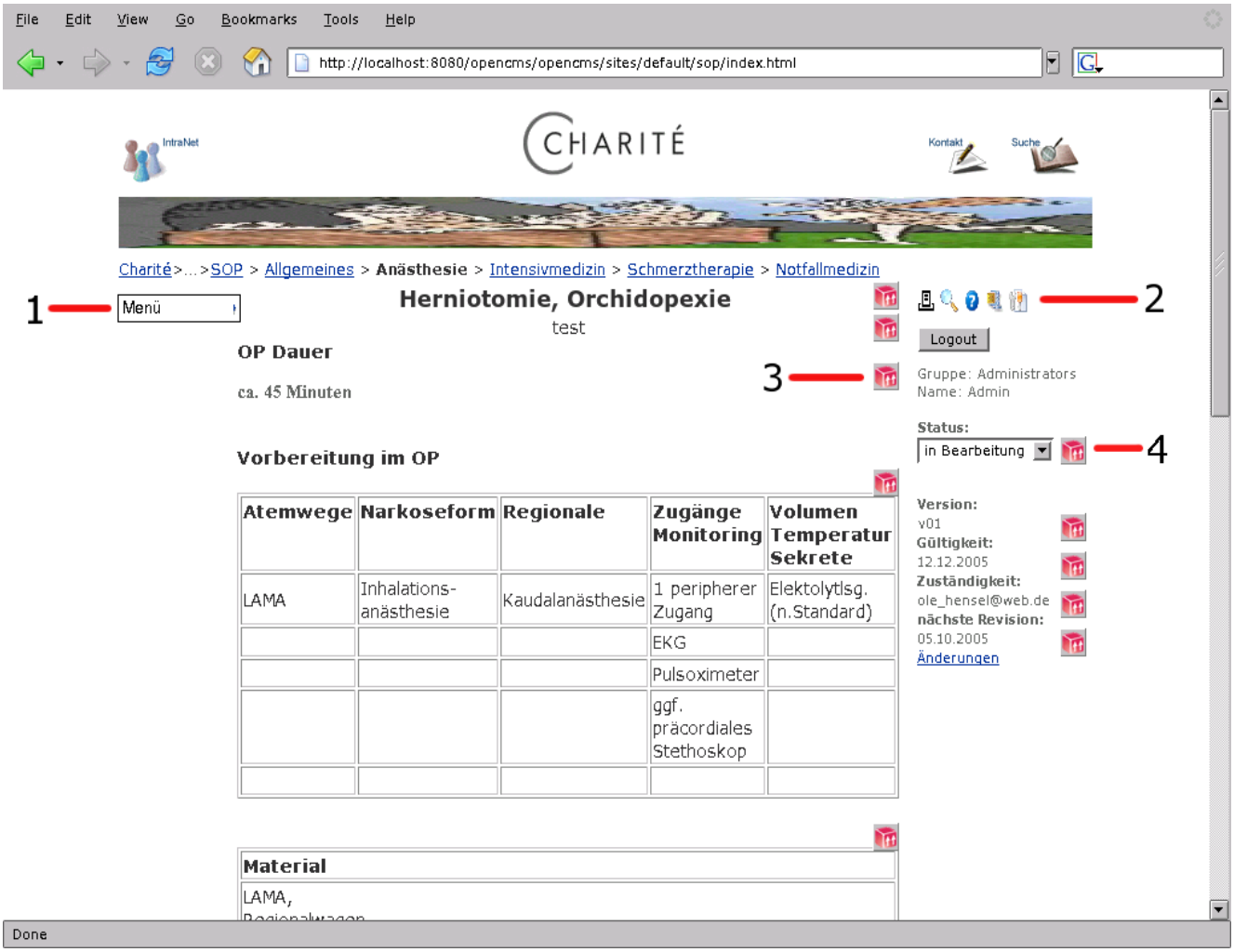

Abb. 20: Eine SOP während ihrer Erstellung bzw. Überarbeitung, der Verfasser ist angemeldet, sonst wären die Direct Edit-Buttons nicht sichtbar (1: dynamisches Menü, 2: Funktionsleiste, 3: Direct Edit, 4: Ändern des SOP-Status).

Bearbeitung" zu „in Prüfung" (Punkt 4). Der Prüfer wird nun per Email benachrichtigt, dass die Erstellung einer neuen SOP abgeschlossen ist. Entweder der Eintrag hält der inhaltlichen und formalen Prüfung stand, dann wird der Status von „in Prüfung" zur „in Freigabe" geändert. Dabei werden die Mitglieder des Gremiums informiert. Oder der SOP-Eintrag geht zurück an die Verfasser, wobei der Status von „in Prüfung" geändert wird zu „in Bearbeitung".
Das Gremium entscheidet endgültig über die Freigabe des neuen SOP-Eintrags, nur sie haben die Möglichkeit Einträge im Intranet zu veröffentlichen bzw. die Veröffentlichung zurückzunehmen.

6. Wartung und Betrieb der SOP-Anwendung

Betrachten wir anfangs die Situation der Softwarewartung und deren Auswirkungen an der Charité etwas genauer. Charité-weit wird die Softwarewartung von zwei Informatikern geleistet. Dies wird per automatisierter Fernwartung realisiert. Voraussetzung ist, dass alle Rechner eine gleiche Softwareausstattung besitzen und eine ähnliche Hardware aufweisen. Um dieses hohe Maß an Effizienz und Sparsamkeit aufrecht zu erhalten, muss jeder Wunsch nach individueller Software abgelehnt werden.

Hinzu kommt die aktuelle Finanznot der Charité. Im Sommer 2005 hat deswegen das Klinikmanagement beschlossen, in den nächsten 5 Jahren bis zu 20% aller Mitarbeiter aus Verwaltung und Service zu entlassen [38]. Dies lässt die Neueinstellungen von Informatikern oder externe Wartungsverträge nicht zu. An der Klinik für Anästhesie gibt es keine angestellten Informatiker und Nicht-Informatiker haben nicht das notwendige Hintergrundwissen eventuelle Wartungen vorzunehmen. Konkret bedeutet dies für meine Anwendung, dass für die Wartung kein Informatiker zur Verfügung steht. Deswegen werden umfangreichere Änderungs- und Erweiterungswünsche nur schwer durchzuführen sein.

Um den laufenden Betrieb der Anwendung zu sichern, muss die Wartung in dieser Situation das gesamte Umfeld beachten. Zur Organisation des reibungslosen Übergangs von Entwicklung zu Wartung (Übergangs-/Auslieferungsprozess des CM^3) werden folgende Punkte vorgeschlagen:

1. Es wird ein Informatiker gebraucht!
   Dieser soll die eventuell auftretenden Fehler und sonstige technische Zwischenfälle bearbeiten. Diesem Informatiker sind optimale Voraussetzungen zu schaffen, so dass er effizient die Wartung durchführen kann. Das heißt ihm/ihr ist die vollständige Dokumentation und alle weiteren Unterlagen (Einstellungen, Passworte usw.) als Kopie zu übergeben, dies betrifft auch Daten und Informationen zu MySQL, Suse Linux, VNC und Tomcat. Zusätzlich sollen Anschauungsbeispiele den Einstieg erleichtern. Alle diese Unterlagen werden vom ärztlichen Administrator verwahrt. Dem Informatiker ist eine ausführliche Einführung in die wichtigsten Aspekte und den Aufbau der Anwendung zu geben, dabei sollte auf den OpenCms-Arbeitsplatz ausführlich eingegangen werden. Dieser kann als Wartungswerkzeug eingesetzt werden. Ideal wäre, wenn der Informatiker einmal im Monat prophylaktisch nach dem Rechten sehen würde, dabei sind zum Beispiel vom Interesse: Log-Files, die MySQL-Datenbank, Festplatten- und Arbeitsspeicherauslastung und die Liste der laufenden Prozesse. Bei längerer Verwendung sollte mindestens jährlich über ein Update der Bestandteile nachgedacht werden. Mit dem Suse YaST Online Update ist dies zum Beispiel für das Betriebssystem einfach möglich.

2. Es wird ein Verantwortlicher für die Anwendung gebraucht!
   Der Verantwortliche gibt hauptsächlich Hilfe und dient als Ansprechpartner für eventuelle Anwenderfragen. Er muss zu Beginn in die Anwendung ausführlich eingewiesen werden, so dass er viele Fragen der Anwender beantworten kann bzw. Workarounds kennt. Zusätzlich zu den Hilfestellungen soll er im Fehlerfall die Wartung einleiten und alle Wartungsunterlagen verwalten (Abb.22). Er muss Sorge tragen, dass diese möglichst vollständig sind. Außerdem soll er sich um die Neuanmeldungen von Prüfern und Mitgliedern der Gremium-Gruppe kümmern.

3. Anleitungen und Kurse für Mitarbeiter!
   Für die Anwender, also Mitglieder des Gremiums, Prüfer und potentielle Verfasser sind Anleitungen und Kurse zu organisieren. Damit sollen Einstiegsprobleme reduziert werden und verdeutlichen, was das Programm kann und was nicht.

4. Softwareauswahl und -konfiguration genau überlegen!
   Aus Gründen der Stabilität als Server ist als Betriebssystem Linux zu empfehlen. Es empfahl sich eine Distribution mit guter und ausführlicher Dokumentation zu wählen, zum Beispiel Suse. Die Software ist so zu konfigurieren, dass nach einem unerwarteten Neustart alle notwendigen Programme selber starten (zum Beispiel Tomcat, MySQL). Beim Mitloggen sollte ein Mittelweg gefunden werden, der nicht zuviel Daten anhäuft, aber auch noch Ereignisse rekapitulierbar bleiben. Sinn macht zum Beispiel die Logeinträge, die älter als zwei Monate sind automatisch löschen zu lassen, so dass die Festplatte nicht zu voll wird.

5. Hardwarekonfiguration genau überlegen!
   Der Einsatz eines RAID-Systems ist anzustreben, diese sollte so konfiguriert werden, dass Benachrichtigungen per Email oder SMS im Bedarfsfall versendet werden. Für ausreichend Festplatten-/Arbeitsspeicherkapazität ist zu sorgen (zurzeit 40 GB Festplatte und 512 MB Arbeitsspeicher). Die Lüfter des Rechners müssen frei bleiben.

6. Datensicherung durchführen!
   Eine sinnvolle Backupstrategie wäre beispielsweise alle 2 Wochen ein Backup durchzuführen, das 5-mal hintereinander, bevor das erste dieser Backups wieder überschrieben wird. Zu beachten ist, dass alle Systemdateien, Benutzerdaten und alle erstellten Webseiten mit gesichert werden.

7. Für sichere Stromzufuhr sorgen!
   Zu Beginn eines jeden Monats wird das zentrale Notstromaggregat der Charite getestet, wobei es zu kurzzeitigem Stromausfall kommt. Aus diesem

Grund ist die Anschaffung einer unterbrechungsfreien Stromversorgung (USV) zu empfehlen. Zusätzlich ist das versehentliche Ziehen des Stromsteckers zu verhindern, sinnvollerweise sollte für diese Aufgabe ein Elektriker kontaktiert werden. Viele Zwischenfälle durch menschliche Interaktion können gemindert werden, wenn der Serverrechner in einen wenig benutzten Raum untergebracht wird (siehe Punkt 9).

8. Für sicheren Netzwerkanschluss sorgen!
   Das versehentliche Ziehen des Netzwerksteckers ist zu verhindern. Das kann erreicht werden, indem der kleine Plastiknippel am Netzwerkstecker entfernt wird.

9. Empfehlungen für die Raumauswahl!
   Am besten den Server in eine verschließbare Abstellkammer mit Strom- und Netzwerkanschluss stellen, die niemand sonst nutzt. Alle Mitarbeiter, die Zugang zu diesen Raum haben, sind darüber zu informieren, dass sich der Server darin befindet. Es ist anzuraten keinen Bildschirm anschließen, damit Mitarbeiter erst gar nicht auf die Idee kommen, den Rechner zweckentfremdet zu nutzen. Perspektivisch ist anzustreben den Server der Webserverfarm der Charité anzugliedern. Wichtig ist, dass der Raum eine Klimaanlage hat oder kräftigen Durchzug, so dass es zu keinen Hitzeschaden kommen kann.

10. Verantwortlichkeiten festlegen!
    Wer ist Ansprechpartner im Wartungsfall, wo und wie ist er erreichbar? Dazu ist auf der Webseite eine Kontaktadresse anzugeben und diese auch beim HelpDesk (zentrale technische Hilfe der Charite) zu hinterlegen.

11. OpenCms Kenner kontaktieren/anstellen!
    In Frage kämen zum Beispiel Studenten aus dem Praktikum OpenCms.

12. Die Fernwartung ermöglichen!
    Dies ist einerseits durch die Webadminstration von OpenCms gewährleistet. Anderseits wird für Systemzugriffe eine Fernwartung per VNC empfohlen.

13. Erreichbarkeit testen!
    Auf einem weiteren Rechner eine Prozedur installieren, die die Erreichbarkeit des OpenCms-Servers und die Tätigkeit von MySQL automatisch testet. Im Fehlerfall soll diese den ärztlichen Administrator per Email oder SMS kontaktieren.

Diese Punkte nehmen keine Änderungen an der Software vor, sind also per Definition nicht Bestandteil der Softwarewartung. Jedoch unterstützen diese Maßnahmen das Ziel der Softwarewartung, die Handlungsfähigkeit und

Produktivität für den Anwender zu sicher bzw. zu erhöhen.

Das „Verstehen der Software" macht rund 50 % des Aufwandes im Wartungsfall aus. Um dies zu unterstützen sind die wichtigsten Dokumentationsquellen zu OpenCms zusammengetragen worden, die das Verstehen des Systems, der Architektur und des Quelltextes erleichtern sollen.

- Als Dokumentation für den Einstieg in OpenCms kommen folgende Quellen in Betracht:
    - Matt Butcher's Buch „Building Web Site with OpenCms" [20], welches 2004 erschienen ist und sich auf die OpenCms Version 5.0 bezieht [2],
    - Handbuch „OpenCms Documentation Version 5.0" und „An Introduction to OpenCms 5.0" sind beide auf der Webseite „www.opencms.org" zu finden [46, 47],
    - Alexander Kandzior's Artikel „Maßkonvektion Strukturierte Inhalte mit OpenCMS 6.0" [52],
    - Mailingliste [3],
    - Administratorhandbuch mit Anleitungen, Quelltextbeispielen usw. (ein Beispiel in Anhang E),
- Als Einstieg für ein Verständnis der Architektur und des Aufbaus von OpenCms sei auf Kapitel 3.3 dieser Bachelorarbeit verwiesen.
- Wenn die Hilfe-Module der Firma Alkacon mit installiert wurden, kann die Alkacon Dokumentation über die Adresse „http://localhost:8080/opencms/opencms/sites/default/alkacon-documentation/index.html" im Browser aufgerufen werden.
- Unter „http://localhost:8080/opencms/opencms/sites/default/alkacon-documentation/javadoc/core/" ist die JavaDoc-Dokumentation zu finden.
- Demonstrationen zum Templatemechanismus findet man unter „http://localhost:8080/opencms/opencms/sites/default/demopages/"
- Demonstrationen zu unterschiedlichen Modulen findet man unter „http://localhost:8080/opencms/opencms/sites/default/modulesdemo/"
- XML Conten Demonstrationen befinden sich unter „http://localhost:8080/opencms/opencms/sites/default/xmlcontentdemo/"

Die Verwendung von bisher bestehenden Komponenten (MySQL, OpenCms) minderte meinen Einfluss auf grundlegende Designentscheidungen. OpenCms bietet jedoch mit seiner API, dem Template-Mechanismus, den XML-Schemas und der einheitlichen Entwicklungsoberfläche gute Voraussetzungen für die Softwarewartung. Dies soll an einigen Szenarien kurz durchgespielt werden. Die Auswahl der Szenarien basieren auf bisherigen Anwenderwünschen.

- *Änderung der Struktur eines SOP-Eintrags*, die SOP-Anwendung verwendet zur Erstellung von SOPs ein strenges Schema. Dies ist auf die speziellen

[2] Probekapitel unter „http://www.packtpub.com/view_book/isbn/1904811043"

[3] Adresse des Archivs: „http://www.opencms.org/opencms/en/development/mailinglist-archive.html"

Bedürfnisse der Anästhesie zugeschnitten. Es kann aber sein, dass dieses für den Praxiseinsatz in der Anästhesie schon zu streng ist. Für benachbarte Fächer wie der Schmerztherapie oder der Intensivmedizin ist dieses Schema schon zu unflexibel. Ich gehe davon aus, dass hier die meisten Änderungswünsche entstehen werden.

- In dem Vorlagenordner /system/modules/de.charite.sop/templates/ befindet sich das bisherige Haupttemplate (template.jsp). In dieser Datei kann mit Hilfe des Texteditors die Struktur der SOP-Einträge angepasst werden.
  Zum Beispiel Änderung der Reihenfolge, erst Untertitel dann Titel:

```
<strong>Untertitel </strong>
<cms:template element="Untertitel">
<cms:include element="Untertitel" editable="true"/>
</cms:template>

<strong>Titel</strong>
<cms:template element="Titel">
<cms:include element="Titel" editable="true"/>
</cms:template>
```

- Ein Anfügen des folgenden Quelltextes in den BODY-Bereich der Datei template.jsp fügt zum Beispiel ein Kommentarteil zu allen SOP-Einträgen hinzu.

```
<strong>Kommentare</strong>
<cms:template element="Kommentar">
<cms:include element="Kommentar" editable="true"/>
</cms:template>
```

- *Neue Funktionen*, auf den bisherigen Vorstellungen des Programm wurde insbesondere die Effizienz und Einfachheit des Veröffentlichens registriert. In der Klinik gibt es verschiedene Anforderungen, die mit einem Content Management System wie OpenCms gelöst werden können. Dies zeigen die Erweiterungswünsche:
  - Im Ordner /system/modules/de.charite.sop/elements/ kann eine neue jsp-Datei erstellt werden, die neue Funktionalitäten bereitstellt. Zur Programmierung neuer Funktionen stehen dazu das OpenCms-API bereit, deren JavaDoc-Dokumentation man im Ordner /sites/default/alkacon-documentation/javadoc/index.html findet.
  - Folgende Erweiterungswünsche sind bisher geäußert worden:
    1. Webseiten mit Informationen zur technische Ausstattung einiger Abteilungen, dies ist insbesondere für den Rettungsdienst wichtig. Wenn eine Reanimation ausgelöst wird, liegen beispielsweise meist keine Informationen über die Ausstattung mit Sauerstoffflaschen

vor.

2. Webseiten mit Informationen zum Verhalten im Notfall, der Ablauf im Notfall muss allen Klinikmitarbeitern bekannt sein. Zur Fortbildung bietet sich hier das Intranet an.
3. Flexible Telefonlisten mit den wichtigsten Ansprechpartnern.
4. Firmennamen von Medikamenten mit Wirkstoffnamen verlinken, bei wechselnden Verträgen mit Arzneimittelfirmen ändert sich häufiger der Präparatename. Dieser wird aber oft benötigt, hier bietet sich der Einsatz eines CMS an.
5. Kosten für die einzelnen Materialien mit einbinden, um aktuelle Kostenvoranschläge machen zu können. Dies wird insbesondere in den Verhandlungen mit den Krankenkassen benötigt.
6. Projekte für mehrere Kliniken erstellen, die im Anhang beschriebene Tauschbörse versucht, den zeitlichen und personellen Aufwand der SOP-Erstellung zu verringern. Der Einsatz eines CMS könnte dies noch weiter rationalisieren.
7. Flexiblere Gestaltung der SOP-Gliederung, um auch ganz andere Klinikbereiche die SOP-Anwendung nutzen zu lassen.
8. Bessere Integration in bestehende Softwarelösungen an der Klinik, verwendete Klinikanwendungen wie Copra und Medvision II arbeiten nicht zusammen , dies führt schon jetzt zu Mehrarbeit.

- Für die Punkte 1 bis 4 müsste jeweils ein neues Template angelegt werden. Dafür kann man die Datei template.jsp kopieren und ein neues Template erstellen. Der SOP-spezifische BODY-Bereich (<cms:template element="body"> bis </cms:template>) muss an die neuen Bedürfnisse angepasst werden. Die entsprechende Ordnerstruktur ließe sich unter /sites/default/sop/Allgemeines/ manuell im Arbeitsplatz erstellen, wobei in den Eigenschaften zu diesem Ordner das neu erstellte Template angegeben werden muss. Für Punkt 4 müsste der Link dann manuell angelegt werden, vom Wirkstoffnamen zum Handelsnamen.
- Zur Realisierung des Punktes 5 müssten alle Materialen zuerst eine eindeutige Kennung erhalten, mit diesem Schlüssel wäre eine Anfrage an die Datenbank des Einkaufs möglich.
- OpenCms ist sehr gut skalierbar [20], einige Sicherheitsaspekte wären insbesonders zu beachten für die Realisierung von Punkt 6,
- Punkt 7 könnte vielleicht mit XML-Schemas realisiert werden oder mit Hilfe von Formular-basierter Templateerstellung,

- *Änderung der Präsentation* (Layout), zur Zeit stellt die Charité ihr Webdesign wieder um. Außerdem müssen die SOP-Einträge aus verschiedenen Abteilungen des Anästhesie auch optisch gut trennbar sein.
  - Im Ordner /system/modules/de.charite.sop/resources/ befindet sich die Datei charite.css. Der Inhalt dieser Datei kann mit dem neuen Design ersetzt werden. Ansonsten ist die Anpassung aller Seiten an das neue Lay-

out über die Datei template.jsp im Vorlagenordner möglich.

- *Neue Umgebung*, perspektivisch soll der Server in die zentrale Webserverfarm der Charité mal umziehen. Auch wichtig um eine Testumgebung zu schaffen.
  - Mit Hilfe des Backups ist es einfach möglich, auf einem neuen System die SOP-Anwendung komplett wieder zum Laufen zu bekommen. Nach Installation von Java SDK (ab 1.4), Tomcat (als einfacher Benutzer) und MySQL kann über die Url „http://localhost:8080/opencms/setup" im Browser der Installationswizard gestartet werden. Nähere Ausführungen dazu im Administratorhandbuch. Nach gelungener OpenCms-Installation kann im Administationsbereich unter Datenbankverwaltung der Import der gesicherten Daten gestartet werden. Die Anwendung ist danach sofort einsatzbereit.
  - Schwierigkeiten für die Wartung können entstehen, wenn die Browser die benötigten Funktionalitäten von OpenCms nicht mehr unterstützen. So könnte es sein, das JavaScript irgendwann nicht mehr unterstützt wird.

Es ist jedoch ausgeschlossen alle zukünftigen Veränderungen der Zukunft und deren Auswirkungen auf die Software vorauszusehen. Umso wichtiger erscheint es, günstige Voraussetzungen zu schaffen, die eine Wartung gewährleisten bzw. noch erleichtern. Dabei kann eine Organisation des Wartungsprozess helfen. Um auf den Wartungsfall vorbereitet zu sein, wird folgender Verlauf vorgeschlagen:

- Wenn ein Fehler bzw. ein Änderungswunsch auftritt, kann der Nutzer diese beschreibend in ein vorgefertigtes Formular (Abb. 21) eingeben. Dieses findet man auf der Webseite im Menü unter Allgemeines. Sollten die SOP-Webseiten nicht vollständig anzeigbar sein, kann der Nutzer das HelpDesk der Charité kontaktieren. Dort sind die Daten des aktuellen Ansprechpartners hinterlegt.
- Die Änderungsanfrage (*Modification Request*) löst die Erstellung eines Wartungsprotokolls aus, dem alle anfallenden Unterlagen zugeordnet werden. Das Protokoll wird durch den ärztlichen Administrator erstellt und soll Online gestellt werden.
- Wenn mehrere Wartungsmitarbeiter existieren, werden diese informiert und ein Verantwortlicher für diesen Wartungsfall bestimmt. Dieser soll die Wartung übernehmen.
- Dieser Wartungsverantwortliche führt eine erste Recherche durch, zum Beispiel indem er im Archiv der OpenCms-Mailingsliste nach ähnlich gelagerten Problemen sucht. In Frage kommen weiterhin das Internet oder die Dokumentationen.
- Der Wartungsverantwortliche organisiert ein Treffen mit dem Anwender, der den Fehler gefunden bzw. Änderungswünsche geäußert hat.
- In einer weiteren Besprechung mit anderen Wartungsmitarbeitern soll das weitere Vorgehen besprochen werden. Je nach Schwere und Situation des Wartungsfalls kann verschieden darauf reagiert werden [48, 49, 50]:

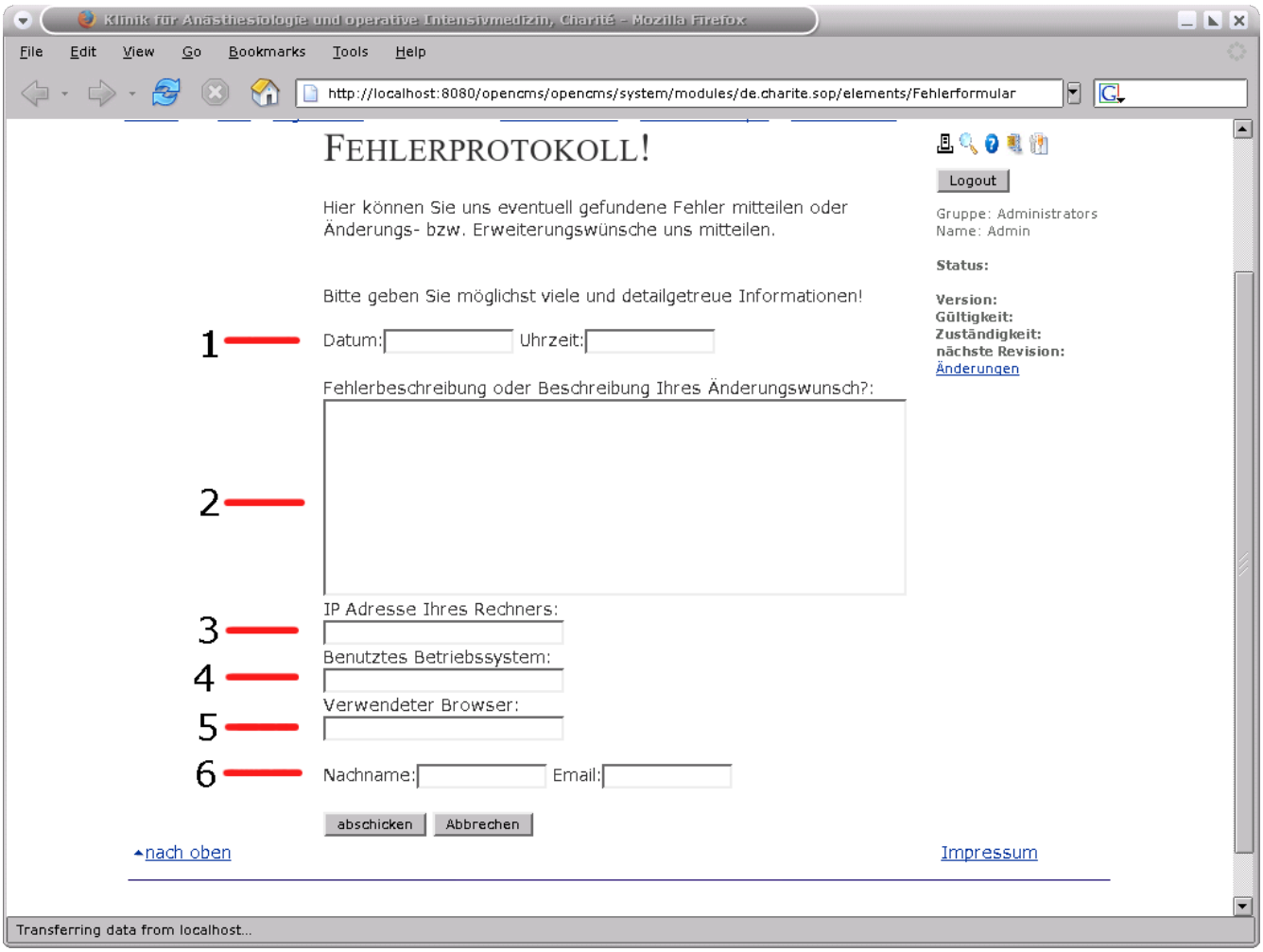

Abb. 21: Meldung von Fehlern bzw. Änderungswünschen (1: Datum, Uhrzeit 2: Beschreibung des Fehlers bzw. Änderungswunsches 3: IP Adresse 4:verwendetes Betriebssystem 5: verwendeter Browser 6: Kontaktdaten)

1. Die Wartung am Programm kann man:
   Selber bzw. von externer Firma durchführen lassen oder mittels bestehender OpenCms-Module beheben.
2. Nichts tun nach dem Spruch „Never changing a running system".
   Dies hat unter bestimmten Situationen seine Berechtigung, wenn zum Beispiel der Aufwand der Änderung den Nutzen nicht aufwiegen würde oder die Folgen einer Änderung nicht abzuschätzen sind, so dass das Risiko eines Totalausfalls als zu hoch angesehen wird. Diese Lösungsvariante sollte aber nicht der Regelfall sein.
3. Ein Re-Engineering durchführen.
   Ein Re-Engineering wird meist bei Legacy-Systemen durchgeführt.
4. Eine völlig neue Anwendung selber entwickeln bzw. erwerben.

Eine Anwendung neu zu entwickeln ist die teuerste und zeitintensivste Lösung.

- Wenn Fehler bzw. Änderungswünsche selber bearbeitet werden sollen, muss sich der Wartungsverantwortliche Verständnis für die Software erarbeiten („Verstehen der Software"). Dafür ist ausreichend Zeit einzuplanen.
- Der Wartungsverantwortliche soll mögliche Lösungen entwerfen und Umsetzbarkeit und Auswirkung der Änderungen analysieren. Eine zweite Recherche soll durchgeführt werden, wenn sich das Problemverständnis vertieft hat.
- Danach soll es zu einer Rücksprache mit anderen Wartungsmitarbeitern und dem Administrator des Systems kommen, um insbesondere Auswirkungen der Änderungen gemeinsam zu besprechen.
- Änderungen planen und einarbeiten, dabei sind Quelltextkonventionen einzuhalten und jede Änderungen ist zu dokumentieren. Die Änderungen sollen an einer Zweitinstallation vorgenommen werden, um die eventuellen Folgen der Änderung in Ruhe analysieren zu können. Diese Zweitinstallation entspricht der Software-Test-Umgebung (STU) aus ISO/IEC 14764 [45].
- Wenn möglich sollen Codereviews von anderen Wartungsmitarbeitern durchgeführt werden.
- Der erste Teil des Testens soll der Wartungsverantwortliche selber durchführen (Systemtest), der Akzeptanztest soll von anderen Mitarbeitern durchgeführt werden, insbesondere von dem Mitarbeiter der den Änderungswunsch geäußert hat (Testprozess des CM^3).
- Nachdem alle Tests positiv sind, sollen die Änderungen in das laufende System eingearbeitet werden.
- Alle Dokumentationen sind zu vervollständigen (Documentationprozess des CM^3).
- Änderungen freigeben im Intranet.
- Alle Beteiligten informieren.
- Vorbereiten auf neue Fehler bzw. Änderungswünsche durch kontinuierliche Weiterbildung der Wartungsmitarbeiter/ärztlicher Administrator, insbesondere in die Verwendung von integrierten Wartungsumgebungen (IME integrated maintenance enviroment).
- Eventuelle Verbesserungsvorschläge für diesen Wartungsprozess einarbeiten.

Die Organisation der Anwendersupports ist in Abb. 22 dargestellt und orientiert sich an dem Corrective Maintenance Maturity Model *(*CM^3).

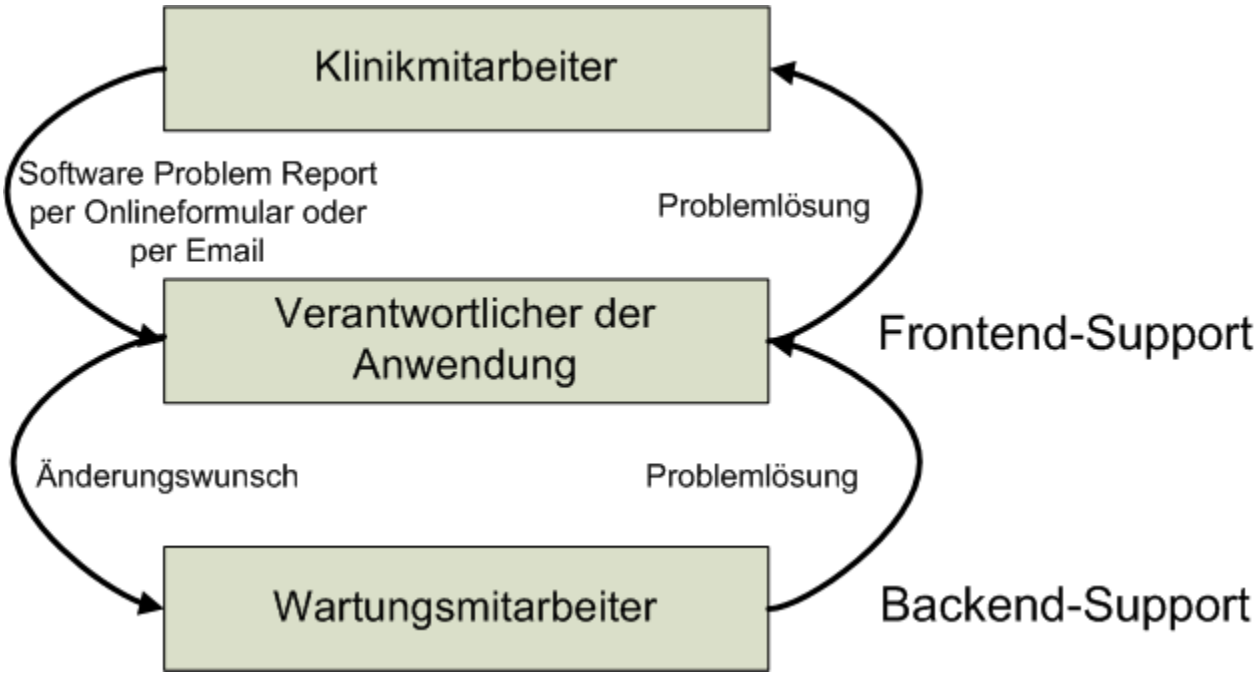

Abb. 22: Organisation des Support für die SOP-Anwendung (in Anlehnung an [65]).

7. Zusammenfassung

Aus meiner früheren Erfahrungen weiß ich , wie groß die Bedeutung von SOPs für die klinische Praxis sind. Nicht klar geregelte Wege und Zuständigkeiten in Kliniken führen zu einem nicht zurechtfertigenden Zeit – und Organisations-aufwand. Die Einführung von „standardisierten Wegen" von Abläufen ist besonders an Schnittstellen in einer Klinik notwenig. Die Einführung der SOPs in Kliniken haben sich mittlerweile auch schon praktisch bewährt. Wichtig bei der Erstellung ist die Einbeziehung aller Personen, die später die SOPs mittragen sollen, um die spätere Einhaltung der standardisierten Wege auch zu gewährleisten.

Meine Aufgabe war es, für die Klinik für Anästhesie und operative Intensivmedizin unter besonderer Beachtung der Wartbarkeit eine Webanwendung zur Verwaltung von SOPs anzufertigen. Mir erschien es in diesem Fall sinnvoll mit OpenCms zu arbeiten. Entsprechendes Know-How mit OpenCms vorausgesetzt, lassen sich mit OpenCms schnell und einfach Webanwendungen erstellen. Es ist in kurzer Zeit eine Anwendung entstanden, mit der es möglich ist, verteilt Standard Operation Procedures zu erstellen und nach einem Kontrollprozess zu veröffentlichen. Die vorhandenen Funktionalitäten von OpenCms bilden eine gute Grundlage zur Steuerung des Life-Cycles einer Standard Operation Procedures.

Nachdem Empfehlungen zum Anwendungbetrieb dargelegt wurden, gab ich bei den Betrachtungen zur Wartbarkeit Einstiegshilfen zum „Verstehen der Software", desweiteren stand die Organisation des Wartungsfalles im Vordergrund.

Wie die bisherige Resonance in der Klinik gezeigt hat, gäbe es für Content Ma-nagement Systeme im Krankenhaus auch andere Einsatzbereiche. Insbesondere die leichte Erstellung von Inhalten und aktuelle Bereitstellung im Intranet stießen auf großes Interesse.

8. Anhang

Anhang A: Glossar

Hmit Wikipedia [40]

Begriff	***Erklärung***
Alkacon	Firma organisiert die Weiterentwicklung von OpenCms und bietet Trainingskurse an.
Änderbarkeit	„Aufwand, der zur Durchführung vorgegebener Änderungen notwendig ist. Änderungen können Korrekturen, Verbesserungen oder Anpassungen an Änderungen der Umgebung, der Anforderungen und der funktionalen Spezifikation einschließen" [28 b]
Anwender	Verwendet die Software in seinem Sinne, gleich Benutzer.
Application Programming Interface (API)	Schnittstelle die Routinen, Protokolle und Dienste für das Erstellen von Software bereitstellt.
Applets	Anwendung, meist in JavaScript geschrieben
Backend	Teil einer Software, läuft auf einen Server und verwaltet die Daten.
Backup	Datensicherung.
Benutzer	Verwendet die Software, gleich Anwender.
Browser	Computerprogramm, welches zum Betrachten verschiedener Inhalte verwendet werden (z.B. Webseiten).
Cascading Style Sheets (CSS)	Deklarative Sprache für strukturierte Dokumente (z.B. HTML, XML), dient der Trennung von Stil und Inhalt.
Content	Bezeichnet mediale Inhalte aller Form (Text, Bild, Klang, Video).
Content-Management-Systeme	Anwendung, die im Internet bzw. Intranets das Erstellen, die Kontrolle, die Freigabe, die Publikation und Archivierung von Inhalten erleichtern [19].
Datenbank	Software, die spezialisiert darauf ist große Mengen von Daten zu verwalten.
Diagnosis Related Groups (DRG)	Klassifizierung von Patienten nach medizinischen Kriterien und ökonomischen Ressourcenverbrauch für

Begriff	***Erklärung***
	die Behandlung.
Effizienz	„Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen.“ [28 b]
Fallpauschalen	Ein Vergütungssystem, dass mit den DRG in den letzten Jahren im Gesundheitswesen eingeführt wurde .
Frontend	Schnittstelle zum Benutzer, dient der Eingabe und dem Anzeigen von Daten.
Gremium	Gruppe, die SOP-Einträge freigeben und aufheben kann.
Gruppen	Dienen der Zusammenfassung von Benutzern mit ähnlichem Profil.
Hypertext Markup Language (HTML)	Beschreibungssprache zur Darstellung von Webseiten.
International Organisation for Standardization (ISO)	Internationale Vereinigung der Standardisierungsgremien von 148 Ländern; die ISO erarbeitet internationale Normen in allen technischen Bereichen, darunter technische, klassifikatorische und Verfahrensstandards.
Java	Objektorientierte, plattformunabhängige Programmiersprache.
JavaDoc	Softwaredokumentationswerkzeug, das aus Java-Quelltexten automatisch Dokumentations-Webseiten erstellt. Dazu müssen spezielle Kommentare im Quelltext eingefügt werden.
JavaScript	Ist eine objektorientierte Programmiersprache, die HTML-Seiten dynamisch gestalten kann. Nicht mit Java zu verwechseln.
Java Virtual Machine (JVM)	Laufzeitumgebung und stellt ein Modell eines Prozessors da, JVM führt den Java-Quelltext bzw. deren Bytecode aus.
Klinische Behandlungspfade (KBP)	Verfahrensanweisung, die beschreibt, wie ein Patient, der mit einer definierten Symptomatik beziehungsweise einer definierten Erkrankung in das Krankenhaus kommt, systematisch nach einem vorgegebenen Ablauf diagnostiziert und behandelt wird
Kochbuch-medizin	Umgangssprachlich für eine Medizin nach einer festen Anleitung.
Komplikation	Unerwünschte Folge einer Erkrankung, eines Unfalls

Begriff	***Erklärung***
	oder eines Eingriffes, das nicht im engeren Sinn zum Krankheitsbild gehört.
Konsole	Einheit zur Befehlsein- und Befehlsausgabe der Ergebnisse.
Layout	Visualisierung bzw. detailliertes Sichtbarmachen eines gedanklichen Bildes.
Lesser GNU Public License (LGPL)	Die LGPL-Lizenz schreibt vor, dass der Quellcode frei verfügbar sein muss. Jede am Quelltext vorgenommene Änderung muss ebenfalss frei verfügbar sein.
Link	Stellt die Verbindung zwischen zwei Webseiten dar.
maligne Lymphome	Bösartige Wucherungen des lymphatischen Systems (z.B. Lymphknoten, Milz).
Middleware	Bietet Dienste zur Vermittlung zwischen Anwendungen (Frontend, Backend) an, so dass die Komplexität verborgen bleibt.
Oberärzte	Leitende Ärzte, ausgewählte Oberärzte bilden die Gruppe der Prüfer.
OCR-Software	Software, die grafische Zeichen erkennt.
Open Source Software (OSS)	Der Quelltext eines Programms ist frei erhältlich, oft als Synonym für freie Software verwendet.
Prüfer	Gruppe, die SOP-Einträge prüfen, außerdem dürfen sie Einträge überarbeiten.
Qualitätsmanagement (QM)	Organisationssystem, dass sicher stellen soll, das Prozesse den Anforderungen entsprechend abgearbeitet werden.
Qualitätssicherung	Teil des Qualitätsmanagements, stellt durch das Erzeugen von Vertrauen sicher, dass Qualitätsanforderungen erfüllt werden.
Re-Engineering	Überarbeitungsprozess unter Einbeziehung bestehender Lösungen.
Server	Dienstleister, der auf Anfragen Webseiten dem Anfragenden zusendet (Webserver).
Servlet	Ist ein in Java geschriebenes Programmmodul, welches auf eine Client-Anfrage dynamisch Webseiten erstellt und diese zurücksendet.
Standard Operation	Standardarbeitsanweisung ist ein Dokument, welches

Begriff	***Erklärung***
Procedure (SOP)	das Vorgehen innerhalb eines Arbeitsprozesses beschreibt. Häufig wiederkehrende Arbeitsabläufe werden textlich beschrieben und den Ausführenden erklärend an die Hand gegeben.
Template	Vorlagen, Schablonen, dienen bei CMS der Trennung von Struktur und Inhalt.
Tier-Architektur	Tier vom Englischen „Schicht", z.B. 3- tier Architektur mit Datenbank, Server und Client.
Übertragbarkeit	„Eignung der Software, von einer Umgebung in eine andere übertragen zu werden. Umgebung kann organisatorische Umgebung, Hardware- oder Software-Umgebung einschließen." [28 b]
Webmaster	Person, die sich mit der Konzeption, der Programmierung und der Wartung einer Webseite beschäftigt.
Wizard	Softwareassistent, der Hilfestellung gibt beim Ausfüllen von Formularen.
Ultraschall	Mittels Schallwellen und deren Reflexion können Bilder von inneren Organen aufgenommen werden.
Workflow	Gleichbedeutend mit Ablauf der Arbeitsschritte.
WYSIWYG	"What you see, is what you get", so wie es jetzt sehe, wird es später veröffentlicht (CMS Kontext).
XML	Extensible Markup Language, ist ein Standard zur Definition von Auszeichnungssprachen.
Zuverlässigkeit	„Fähigkeit der Software, ihr Leistungsniveau unter festgelegten Bedingungen über einen festgelegten Zeitraum zu bewahren." [28 b]

Anhang B: Beispiel-SOP

1.9.3 Laparoskopische Eingriffe in der Gynäkologie

- Diagnostische Laparoskopie
- Sterilisatio
- Benigne Ovarialtumore
- Myomenukleation

***OP-Dauer**: je nach Eingriff: Sterilisatio: ca. 10-20 min, bei benignen Tumoren auch bis ca. 1,5-2h*

Atemwege	**Narkoseform**	**Regionale**	**Zugänge Monitoring**	**Volumen Temperatur Sekrete**
ITN oral	Balanziert	PCEA	Bräunüle	Infu-wärmer
	Kombination		gr. Braunüle	Heißluftdecke
			Arterie	Magensonde
			ZVK	DK
			Temp.sonde	

Vorbereitung im OP

Material
- peripher-venöser Zugang (18G) - Endotrachealtubus (Magill) 7,5 mm - Magensonde mit Mandrin - Blasenkatheter: wird vom Operateur gelegt

Medikamente	
Infusion	Perfusion
- NaCl 0,9 % 10 ml - Atropin 0,5mg/ml - Fentanyl 0,5mg/10ml - Trapanal 250mg/10ml	

- Isoflurane-Verdampfer - Rocuronium 50mg/10ml - ggf. Mivacurium 20mg/20ml - Vollelektrolytlösung	

Monitoring
- Standard-Monitoring

Prämedikation
- nach Standard

Blut/ Transfusionsbedarf
- es werden keine EK im Allgemeinen benötigt

Narkoseeinleitung
- Einleitung nach Standard

Narkoseführung
- balanzierte Anästhesie

Lagerung
- Rückenlage
- Beidseits angelagerte Arme

Besonderheiten
-Besonderheiten der Laparoskopie beachten: CO2-Insufflation in den Bauchraum mit Anlage eines Pneumoperitoneums, dabei evtl. eingeschränkte Ventilation, später vermindertes HZV und verstärkte Auskühlung bei hohen CO2-Flow

Komplikationen
- Wundbrand
- Infektionen (beides als Beispiel)

Postoperatives Management
- Betreuung im Aufwachraum mit Standardmonitoring
- Ein kontinuierliches Schmerztherapieverfahren incl. PCA ist in der Regel nicht erforderlich.
- Verlegung nach Anweisung

Literatur

Anhang C: Pflichtenheft

Bachelorarbeit SOP-Datenbank mit Webinterface

Pflichtenheft
Version 0.5

1. Zielbestimmung

Zur Unterstützung des klinikinternen Qualitätsmanagements sollen Standard Operation Procedures an der Charité (Campus Virchow-Klinikum) eingeführt werden. Das Pilotprojekt soll in der dortigen Anästhesie durchgeführt werden. Die Richtlinien sollen im Intranet jeden Klinikmitarbeiter zur Verfügung stehen, so dass dieser jederzeit dort nachschlagen kann. Des Weiteren soll das Programm Hilfe leisten in der Erstellung von Standard Operation Procedures (SOP).

1.1 Musskriterien

-neue SOP-Richtlinien anlegen, löschen und editieren, bestehend aus:

- eindeutigen Titel mit Nummerierung
- Status mit Gültigkeitsangabe
- Angaben geprüft am, geändert am
- Unter-Überschriften
- OP-Dauer
- Übersichts-Spalte
- Tabelle mit Material, Medikamente, Monitoring
- Prämedikation
- Narkoseeinleitung
- Narkoseführung
- Lagerung
- Besonderheiten
- Komplikationen
- Postoperative Management
- Blut- und Transfusionsbedarf
- Literaturhinweise
- Kommentare

- SOP-Richtlinien suchen, gefundene Seiten anzeigen
- SOP-Richtlinie lassen sich untereinander verlinken
- SOP-Richtlinien als final/in Bearbeitung/in Prüfung markieren mit Zeit-Stempel und wieder aufheben
- SOP-Richtlinie ausdrucken
- neue SOP-Datenbank anlegen
- Kommentare zu Einträgen erstellen, bearbeiten, löschen
- neue Nutzer anmelden, deren Daten bearbeiten, Nutzer löschen
- Hilfefunktionen
- Rechte-Management

- Listen/ Übersichten mit Angaben zu Anzahl der Einträge, der Nutzer usw.

1.2 **Wunschkriterien**

- einfache Konfigurierbarkeit
- hohe Flexibilität in Texterzeugung bei einfacher Handhabung
- statistische Angaben
- Datensicherung
- Teile der SOP-Datenbank ausdrucken
- bestehende Einträge importieren/exportieren
- Selbst-Registrierung von Benutzern

1.3 Abgrenzungskriterien

Die Applikation muss keinen hohen Sicherheitsansprüchen genügen und soll zu keinen Wiki oder Forum werden.

2. Produkteinsatz

Die Applikation soll von Klinikmitarbeitern über ihren Arbeitsplatzrechner genutzt werden.

2.1 Anwendungsbereich

Webgestützte Erstellungshilfe und nachfolgend Nachschlagewerk für SOP's im Klinik-Alltag.

2.2 Zielgruppe

Die Applikation soll von allen Klinikmitarbeitern, vorrangig Ärzten und Pflegepersonal, genutzt werden.

- Gäste haben von jedem Klinikrechner die Möglichkeit die SOP-Einträge zu lesen bzw. auszudrucken.
- Die Gruppe, die mit „Arzt, Schwester“ bezeichnet ist, entspricht der Gruppe der Verfasser. Sie sind verantwortlich für die Erstellung neuer SOP-Einträge.
- Oberärzte prüfen die erstellten SOP-Einträge und sind verantwortlich für einzelne SOP-Einträge. Sie wählen die Verfasser aus.
- Gremium ist die Instanz, die SOP-Einträge freigeben kann im Intranet, nach einer abschließenden Prüfung.
- Administrator sorgt für den reibungslosen Betrieb der Anwendung und kümmert sich um die Datensicherung.

Nicht bei allen Anwendern ist umfangreiche Erfahrung in der Bedienung von Web-Applikationen vorauszusetzen. Browser-Bedienung ist bekannt.

2.3 Betriebsbedingungen

Die Applikation läuft auf einem zentralen Tomcat-Server. Die Applikation soll bis auf Wartungs-

arbeiten rund um die Uhr in Betrieb sein.

3. Produktübersicht

Die Abbildung 1 gibt die beteiligten Akteure mit deren Rechtestrukur wieder. Alle vereinbarten Funktionalitäten werden in Abb. 2 grafisch dargestellt. Im letzten Bild wird der Entstehungsprozess einer SOP dargestellt (Abb. 3).

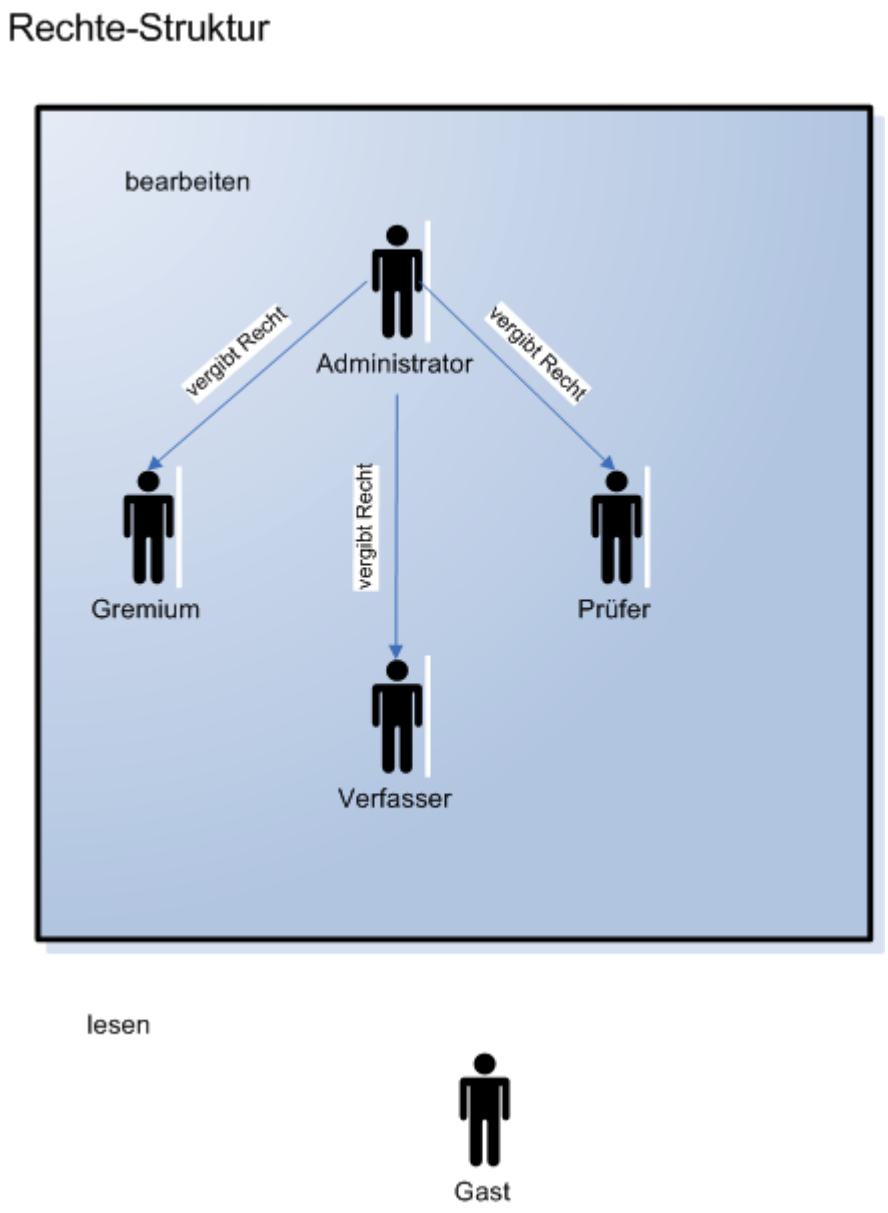

Abb. 1: Rechtestruktur aller Beteiligten.

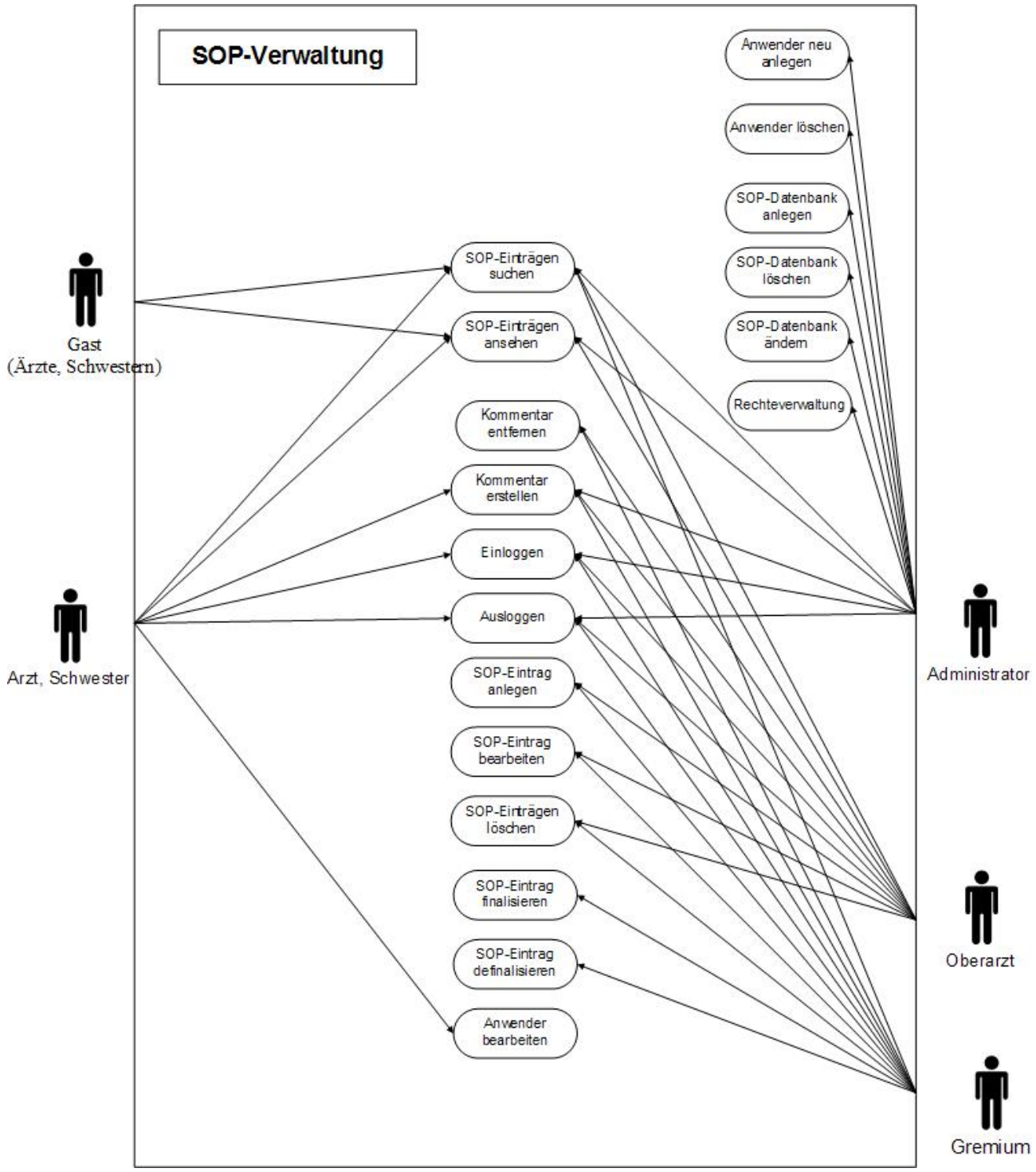

Abb. 2: Vereinbarte Funktionalität.

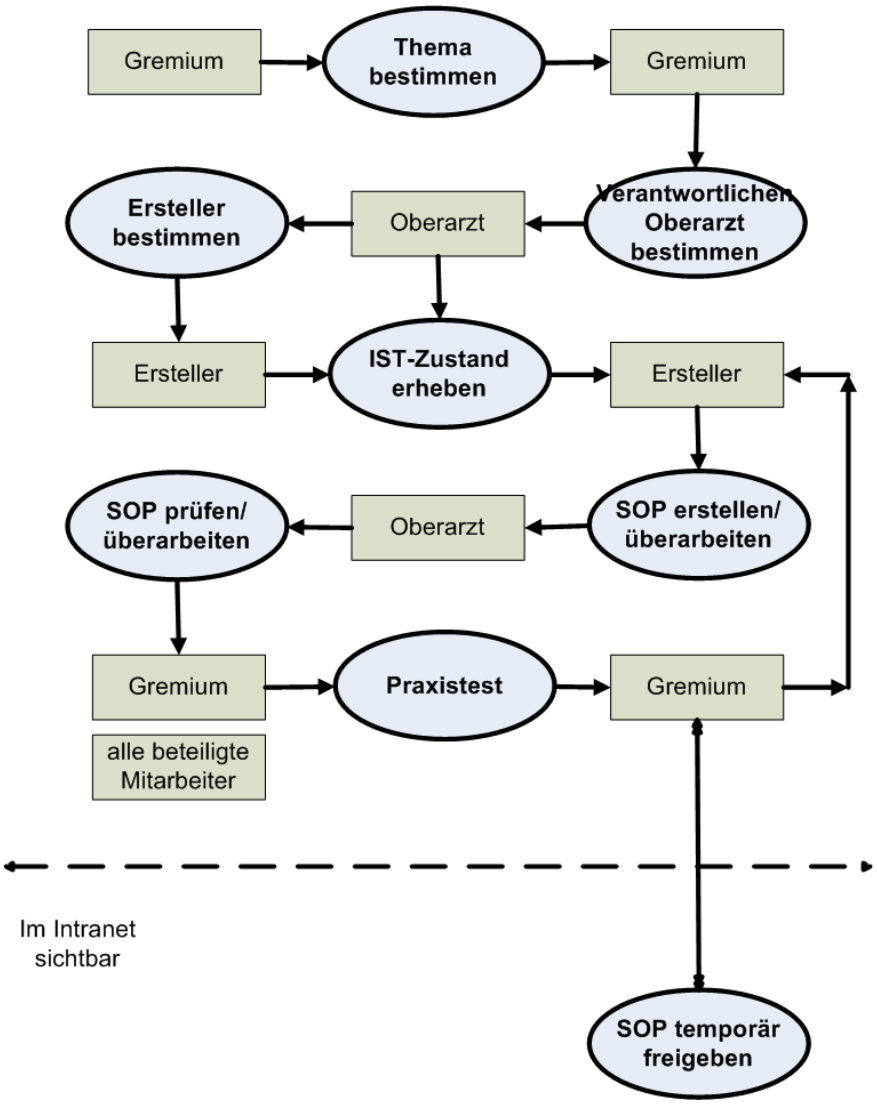

Abb. 3: Erstellungsprozess einer SOP.

4. Produkt-Funktionen

4.1 Geschäftsprozesse

SOP-Anwendung

/F10/ **Geschäftsprozess:** neue SOP-Datenbank anlegen
Akteur: Administrator
Beschreibung: Ein Administrator legt eine SOP-Datenbank neu an und gibt den Namen sowie den Namen des Verantwortlichen an. Danach besteht eine neue SOP-Datenbank, welche noch keine Einträge hat.

/F20/ **Geschäftsprozess:** SOP-Datenbank löschen
Akteur: Administrator

Beschreibung: Ein Administrator löscht eine nicht mehr benötigte SOP-Datenbank mit vorheriger Sicherheitsabfrage. Dabei werden die gespeicherten SOP-Daten und SOP-Einträge aus der zentralen Datenbank gelöscht.

/F30/ **Geschäftsprozess**: SOP-Datenbank ändern
Akteur: Administrator
Beschreibung: Der Administrator verändert bei einem bestehenden SOP-Datenbank den Namen der SOP-Anwendung und den Namen des Verantwortlichen.

SOP-Eintrag

/F40/ **Geschäftsprozess:** SOP-Einträge suchen
Akteur: alle
Beschreibung: Ein Akteur tätigt eine einfache Suche, indem er einen Suchbegriff in SOP-Eintrags-Daten (wie z.B. Titel, Beschreibung). Wenn mehrere gefunden werden, wird eine Liste von gefundenen SOP-Einträgen zurückgegeben, wenn die Suche eindeutige Ergebnisse zurück liefert, wird direkt dahin gesprungen. Es sollen Suchen in verschieden Kategorien möglich sein. Standard-Suche wird Suche in Titel/Unter-Überschrift/eindeutiger Kennung sein. Andere Such-Arten können über Auswahl aktiviert sein.

/F50/ **Geschäftsprozess:** SOP-Einträge erweitert suchen
Akteur: alle
Beschreibung: Ein Akteur tätigt eine erweiterte Suche, indem er sich durch das Inhaltsverzeichnis zur gesuchten Stelle klickt.

/F60/ **Geschäftsprozess:** SOP-Eintrag anlegen
Akteur: Verfasser, Gremium
Beschreibung: Ein Akteur fügt einer bestehenden SOP-Datenbank einen neuen SOP-Eintrag hinzu und gibt zugehörige Eintrags-Daten an. Eintrags-Daten sind:

- eindeutigen Titel mit Nummerierung mit Versionsangabe
- Status mit Gültigkeitsangabe
- Angaben erstellt am/von wem, geprüft am/von wem, geändert am/von wem, freigegeben am, Verantwortlicher,
- Unter-Überschriften
- OP-Dauer
- Übersichtsspalte
- Tabelle mit Material, Medikamente, Monitoring
- Prämedikation
- Narkoseeinleitung
- Narkose-Führung
- Lagerung
- Besonderheiten
- Komplikationen
- Postoperative Management
- Blut- und Transfusionsbedarf
- Pflegerische Besonderheiten mit Tabelle
- Literaturangaben

/F70/ **Geschäftsprozess:** SOP-Eintrag bearbeiten
Akteur: Verfasser, Prüfer
Beschreibung: Ein Akteur nimmt Änderungen an den Eintrags-Daten vor, solange dieser nicht geprüft/freigegeben ist. Dabei wird eine Version-Nummer hochgezählt.

/F80/ **Geschäftsprozess:** SOP-Eintrag freigeben

Akteur: Gremium
Beschreibung: Ein Mitglied des Gremiums gibt einen SOP- Eintrag mit Angabe der Gültigkeit frei. Danach ist der Eintrag nicht mehr änderbar für einen Zeitraum.

/F90/ **Geschäftsprozess:** SOP-Eintrag hebt Freigabe wieder auf
Akteur: Gremium
Beschreibung: Ein Mitglied des Gremiums nimmt Freigabe eines SOP- Eintrag wieder zurück. Danach ist der Eintrag wieder änderbar.

/F100/ **Geschäftsprozess:** SOP-Eintrag kommentieren
Akteur: alle
Beschreibung: Ein Akteur ergänzt einen bestehenden SOP-Eintrag um einen Kommentar.

/F110/ **Geschäftsprozess:** Kommentar entfernen
Akteur: Verfasser, Gremium
Beschreibung: Ein Akteur entfernt einen Kommentar eines SOP-Eintrages.

/F120/ **Geschäftsprozess:** SOP-Eintrag ausdrucken
Akteur: alle
Beschreibung: Ein Akteur druckt sich einen SOP-Eintrag aus..

Anwender

/F200/ **Geschäftsprozess:** Selbst-Registrierung von Anwendern
Akteur: Prüfer
Beschreibung: Ein Akteur aktiviert seinen Anwender-Eintrag in der Benutzer-Datenbank, dabei gibt er Anwender-Daten mit Login-Name, Name, Vorname, Funktion, Email und einem persönlichen Passwort an. Danach ist ein neuer Anwender angemeldet.

/F210/ **Geschäftsprozess:** Anwender-Eintrag bearbeiten
Akteur: Verfasser, Prüfer, Gremium,
Beschreibung: Ein Anwender ändert seine Anwender-Daten; er kann seinen Namen, Vornamen, seine Email-Adresse, sein Passwort oder seine Funktion ändern. Nicht ändern kann er seinen Login-Namen. Danach sind nur noch die neuen Daten gültig.

/F220/ **Geschäftsprozess:** Anwender löschen
Akteur: Administrator
Beschreibung: Der Administrator löscht Anwender aus Benutzer-Datenbank, damit werden die zugehörigen Daten und seine Anwender-Rechte gelöscht.

System

/F300/ **Geschäftsprozess:** Einloggen
Akteur: Verfasser, Prüfer, Gremium, Administrator
Beschreibung: Ein Akteur bekommt durch Angabe von Login-Namen und seines entsprechendem Passwortes das Recht SOP-Einträge editieren zu dürfen.

/F310/ **Geschäftsprozess:** Ausloggen
Akteur: Verfasser, Prüfer, Gremium, Administrator
Beschreibung: Ein Akteur verliert das Recht Kommentare abzugeben, danach ist der Akteur nicht mehr im System angemeldet oder erhält einen Fehlerhinweis.

Rechte-Verwaltung

/F400/ **Geschäftsprozess:** Rechte-Gruppen bearbeiten
Akteur: Oberarzt, Gremium, Administrator
Beschreibung: Ein Akteur fügt Anwender zu Gruppen hinzu. Zum Beispiel kann somit ein Akteur das Recht erhalten, einen SOP-Eintrag zu erstellen. Des Weiteren kann er in eine Gruppe aufgenommen werden, die SOP-Einträge prüfen dürfen. Die Mitglieder der Gremium-Gruppen dürfen einen Eintrag freigeben/wieder schließen. Es können Mitglieder hinzugefügt werden oder

wieder aus der Gruppe entfernt werden.

Kann-Kriterien

/F500/ **Geschäftsprozess:** Oberflächen-Struktur ändern
Akteur: Administrator
Beschreibung: Ein Akteur wählt bei der Neuerstellung einer SOP-Datenbank die für die Ausgabe zu verwendende Oberflächenstruktur.

/F510/ **Geschäftsprozess:** Liste von SOP-Einträgen importieren
Akteur: Administrator
Beschreibung: Ein Akteur fügt der SOP-Anwendung eine Liste von bestehenden SOP-Einträgen hinzu, die aus „komma“-separierten Text-Format bestehen müssen.

/F520/ **Geschäftsprozess:** Liste von SOP-Einträgen exportieren
Akteur: Administrator
Beschreibung: Ein Akteur exportiert eine Liste von SOP-Einträgen in eine „komma“-separierten Datei, dies kann zur Datensicherung dienen.

/F530/ **Geschäftsprozess:** Statistische Angaben
Akteur: alle
Beschreibung: Auf der Startseite werden Angaben über Anzahl der Einträge und Anzahl der angemeldeten Anwender sichtbar.

/F540/ **Geschäftsprozess:** Datensicherung
Akteur: Administrator
Beschreibung: Ein Administrator kann die Datenbank sichern

/F550/ **Geschäftsprozess:** Teile der SOP-Datenbank ausdrucken
Akteur: Gremium
Beschreibung: Ein Mitglied des Gremiums kann Teile der Datenbank ausdrucken.

/F560/ **Geschäftsprozess:** Übersicht mit zur freigegebenen Einträgen einsehen
Akteur: Gremium
Beschreibung: So bekommt man als Mitglied des Gremiums eine Übersicht welche SOP-Einträge den Status Prüfung erreicht haben und auf der nächsten Sitzung zur Diskussion stehen.

/F570/ **Geschäftsprozess:** Übersicht mit zuletzt geänderten SOP- Einträgen einsehen
Akteur: alle
Beschreibung: Auf einer allgemeinen Einstiegsseite findet man eine Übersicht, der in den letzten 3 Monaten freigegebenen SOP-Einträge.

/F580/ **Geschäftsprozess:** Übersicht mit vorgenommen Änderungen
Akteur: alle
Beschreibung: Auf einer SOP-Seite gibt es die Möglichkeit den Verlauf der inhaltlichen Änderungen dieses Artikels zu verfolgen.

/F590/ **Geschäftsprozess:** Hinweise auf Aktualität einsehen
Akteur: alle
Beschreibung: Auf einer SOP-Seite soll es einen optischen Hinweis geben, wenn der Artikel freigegeben wurde in den letzten 3 Monaten, bzw. wenn oberärztliche Änderungen vorgenommen wurden in den letzten 3 Monaten.

5. Produktdaten

/D10/ Daten der Anwender (max. 500 Nutzer in 12 Fachabteilungen)

Daten :

Login-Name
Paßwort
Titel
Name
Vorname
Email-Adresse

/D20/ SOP-Einträge (max 10000)

/D30/ sonstige Einträge (max. 1000)

/D40/ SOP-Datenbanken (max.20 SOP-Datenbanken)

6. Produktleistungen

/L10/ Serveranfragezeit, (max. 3 sec)
/L20/ Serververarbeitungszeit (max. 5 sec)
/L30/ Serverantwortzeit (max. 3 sec)

Gesamt-Reaktionszeit: max. 11 sec

Servicelevel: bei Ausfall am nächsten regulären Arbeitstag wieder in Betrieb,

7. Qualitäts-Zielbestimmung

Produktqualität	***Sehr gut***	***Gut***	***Normal***	***Nicht relevant***
Funktionalität		**x**		
Angemessenheit			**x**	
Richtigkeit			**x**	
Interoperabilität		**x**		
Ordnungsmäßigkeit			**x**	
Sicherheit				**x**
Zuverlässigkeit			**x**	
Reife			**x**	
Fehlertoleranz		**x**		
Wiederherstellbarkeit		**x**		
Benutzbarkeit		**x**		
Verständlichkeit		**x**		
Erlernbarkeit		**x**		
Bedienbarkeit		**x**		
Effizienz		**x**		
Zeitverhalten			**x**	
Verbrauchsverhalten			**x**	
Änderbarkeit	**x**			
Analysierbarkeit			**x**	
Modifizierbarkeit	**x**			
Stabilität		**x**		
Prüfbarkeit			**x**	
Übertragbarkeit			**x**	
Anpassbarkeit			**x**	
Installierbarkeit			**x**	
Konformität			**x**	
Austauschbarkeit			**x**	

8. Benutzerschnittstelle

/B10/ Der Zugriff auf das System erfolgt über das Intranet mit einem aktuellen Internet Explorer (Firefox auch möglich, aber nicht empfohlen).

/B20/ Kommentieren dürfen nur autorisierte Anwender.

/B30/ Standardmäßig ist eine menüorientierte Bedienung vorgesehen.

/B40/ Das Layout kann vom Administrator änderbar sein.

9. Nichtfunktionale Anforderungen

Keine.

10.Technische Produktumgebung

Die Applikation ist eine Webanwendung.

10.1 Software

- Serverseitig wird ein Tomcat 5.0 Server und Java 1.4.2 auf einem Windows oder Linux Betriebssystem vorausgesetzt. Die Datenbank wird eine MySQL sein
- Clientseitig wird eine aktuelle Version der Browser Internet-Explorer vorausgesetzt.

10.2 Hardware

- Serverseitig wird ein Intel-Kompatibler Rechner mit mindestens 1,5GHz, 256 Megabyte RAM und freier Festplattenkapazität je nach SOP-Umfang vorausgesetzt.
- Clientseitig wird ein Rechner mit mindestens 300 MHz und 64 Megabyte RAM vorausgesetzt.

10.3 Orgware

- Der Server muss aus dem Intranet erreichbar sein. Die Clients benötigen Zugang zum Intranet.

10.4 Produktschnittstellen

Keine

11. Spezielle Anforderungen an die Entwicklungs-Umgebung

Keine

12. Gliederung in Teilprodukte

Keine

13. Ergänzungen

13.bisher keine

Anhang D: bestehende Softwarelösungen

Es werden 3 bestehende Lösungen vorgestellt, die die Entwicklung neuer SOPs unterstützen.

- Von der deutschen Gesellschaft für Anästhesiologie und Intensivmedizin (DGAI) und dem Springer Akademischer Verlag betriebene SOP-Tauschbörse,
- „SOP Manager" der Firma chromasoft,
- „SOP-Creator" von der Universität Leipzig,

Die Online-Tauschbörse von der Deutschen Gesellschaft für Anästhesiologie und Intensivmedizin (DGAI) und dem Springer-Verlag bietet die Möglichkeit an, SOPs nicht komplett neu entwickeln zu müssen. Martin et al unterstreicht den Sinn, eine SOP-Tauschbörse aufzubauen [13].

Die Börse bietet fertige und überprüfte SOP's den registrierten Mitgliedern des DGAI an. Die Mitglieder sollen die SOPs als Vorlage verwenden, um somit ihre eigenen SOPs leichter entwickeln zu können.

Damit eine SOP von der DGAI ins Netz gestellt wird, muss ein Kontrollprozess durchlaufen werden. Mit Hilfe eines vorgegebenen Formats (Word-Vorlage) wird die SOP erstellt. Nach Rücksendung an die DGAI prüft ein Gutachter den Inhalt der SOP und entscheidet ob der SOP-Eintrag noch einmal überarbeitet werden muss oder nicht. Wenn die SOP formal und inhaltlich korrekt ist, wird sie den anderen Mitgliedern zum Download zur Verfügung gestellt.

Bei diesem Weg besteht die Gefahr, dass die übernommen SOPs zu wenig an die lokalen Bedürfnisse der Klinik angepasst werden. Sollte dies beachtet werden, kann der Ressourcenaufwand zur Erstellung neuer SOP's in einer Klinik vermindert werden. Die klinikinterne Veröffentlichung müsste noch realisiert werden.

Die Funktionalitäten des „SOP Manager" v 4.0 der Firma chromasoft sollen stichpunktartig aufgezählt werden.

- Übersichtsseite mit freigegebenen SOPs und SOPs im Archiv,
- auf der jeweiligen Übersichtseite,
- Neue SOPs anlegen, ändern, löschen,
- Umfangreiche Suchfunktionen,
- Rechteverwaltung,
- Verteilerliste,

Voraussetzung der Installation ist eine gültige Microsoft Access Lizenz.
Das Programm ist als Einzelplatzanwendung konzipiert. Das bedeutet, alle Rechner auf die das Programm installiert ist, müssten separat gewartet werden. Dieser Punkt sprach gegen den Einsatz des Programms.

Der „SOP-Creator" ist an der Universität Leipzig entwickelt worden, die Anwendung wurde vom Bundesministerium für Bildung und Forschung (BMBF) gefördert.

Die Abbildung 12 zeigt den Ablauf (Workflow) zur Erstellung einer neuen SOP mit Hilfe des „SOP-Creator".

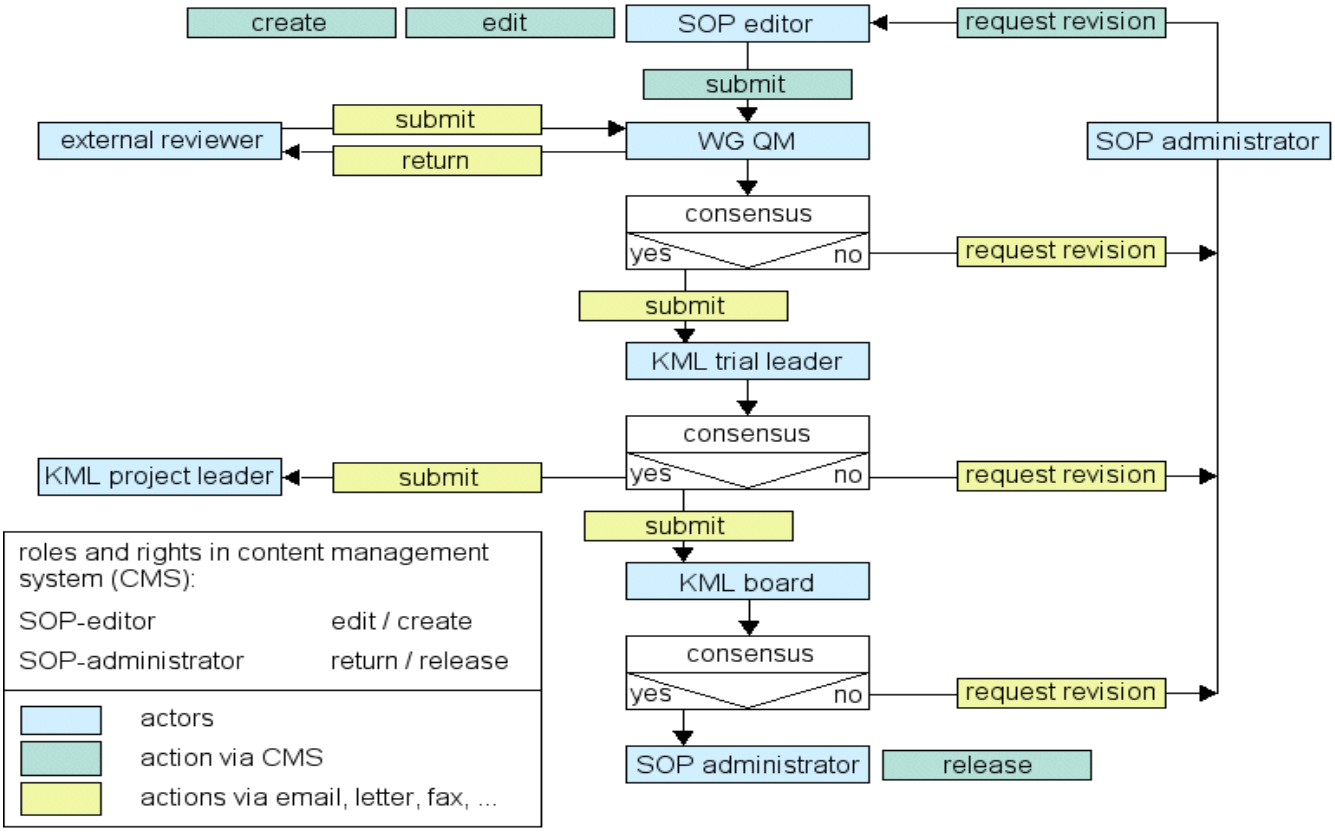

Abb. 4:Der Entwicklungsprozess zur Erstellung neuer SOPs mit Hilfe des SOP-Creators (aus [35]).

Der „SOP-Creator" beruht auf den Kernfunktionalitäten von CMS VIP, als Client

kommt das erweiterte Redakteurswerkzeug VIP HTML zum Einsatz.
Folgende Funktionalitäten werden laut Handbuch unterstützt [43]:

- Erstellen/editieren/löschen von SOPs,
- Freigabe der SOP nach Kontrolle,
- Nutzer- und Rollenkonzept,
- Verschieden Ausgabemöglichkeiten,
- Versionsverwaltung,
- Anlagenverwaltung,

Zurzeit wird der „SOP-Creator" vom Kompetenznetz maligne Lymphome eingesetzt [35].

Weitere Informationen waren auf mehrmalige Nachfrage nicht erreichbar.

Anwendungen, die sich auf die Erstellung von SOPs spezialisiert haben, gibt es bisher wenige. Die Lösung des „SOP-Creator" kommt den gewünschten Anforderungen am nächsten.

Anhang E: Buttonerklärung

Häufig benötigte Buttons sind dick gekennzeichnet.

Symbol	***Name***	***Funktion***
	Elements	Lässt einzelne Bestandteile der Seite an- bzw. abstellen, wenn z.B. kein Untertitel existiert kann dieses abgestellt werden, Cave: Daten werden gelöscht dabei.
	Properties	Zeigt die Eigenschaften an, z.B. verwendetes Template, Titel, Keyword, Beschreibung, Name in Navigation, Stellung in Navigation. Diese Daten lassen sich auch ändern.
	Cleanup HTML Code	Reinigt den HTML Code automatisch.
	Publish direcly	Nur Mitglieder der Gremium-Gruppe können damit den erstellten Inhalt publizieren, d.h. im Intranet sichtbar machen.
	Save and Exit	**Speichern des erstellten Inhalt und gleichzeitig Verlassen des Editors**
	Save	Speichert erstellten Inhalt
	Toggle HTML Source	Umschalten der Ansichten, einmal WYSIWIG oder HTML-Code-Ansicht (für Experten eher gedacht).
	Undoes your last action	Die letzte Aktion wieder rückgängig machen.
	Redoes your last action	Wenn letzte Aktion rückgängig gemacht wurde, diese wieder herstellen.
	Cut selection	Den ausgewählten Bereich ausschneiden und in Zwischenspeicher kopieren.
	Copy selection	Den ausgewählten Bereich kopieren in den Zwischenspeicher.
	Paste from Clipboard	Gespeicherter Inhalt aus Zwischenspeicher wieder einfügen.
	Insert Table	Eine Tabelle einfügen mit Abfrage der Anzahl der Spalten (Cols) und Zeilen (Rows), der Größe der Tabelle, der Anordnung, der Stärke der

Symbol	***Name***	***Funktion***
		Linien.
	Link	Einen internen Link einfügen mit Auswahl des Zieles und wie die neue Seite zu öffnen ist. Zur Benutzung muss vorher Text markiert sein.
	Anchor	Wenn Text markiert ist, kann damit ein sogenannter Anker angelegt werden. Später kann dann direkt auf diesen referenziert werden mit einen Link.
	Insert/Modify Image	Ein neues Bild einfügen oder ein Bild bearbeiten, notwendig ist eine Adresse des Bildes (URL).
	Table gallery	Bestehende Tabellen können ausgewählt werden.
	HTML gallery	Bestehende Bestandteile einer Webseite können ausgewählt werden.
	External links gallery	bestehende externe Links können ausgewählt werden,
	Image gallery	Bestehende Bilder können damit ausgewählt werden.
	Download gallery	Bestehende andere Dateien können ausgewählt werden, wie z.B. pdf oder Excel-Dateien.
Ω	Special characters	Sonderzeichen einfügen, ruft ein Extrafenster auf.
¶	Direction left to right	Textrichtung von links nach rechts einrichten.
¶	Direction right to left	Textrichtung von rechts nach links einrichten.
—	Horizontal Rule	Fügt eine horizontale Linie ein.
	Help	Führt zur OpenCms-Hilfe.
×	**Close**	**Schliesst das Editorfenster ohne Veränderungen zu speichern.**
B	Bold	Markierte Schrift **dick** darstellen.
I	Italic	Markierte Schrift *kursiv* darstellen.

Symbol	***Name***	***Funktion***
	Underline	Markierte Schrift unterstrichen darstellen.
	Strikethrough	Markierte Schrift durchgestrichen darstellen.
	Subscript	Markierte Schrift tiefgestellt darstellen.
	Superscript	Markierte Schrift hochgestellt darstellen.
	Justify Left	Markiertn Text links ausrichten.
	Justify center	Markierten Text mittig ausrichten.
	Justify right	Markierten Text rechts ausrichten.
	Justify Full	Markierten Text beidseits ausrichten.
	Bulleted List	Unnumerierte Aufzählungsliste.
	Ordered List	Nummerierte Aufzählungsliste.
	Increase Indent	Markierten Text komplett nach rechts verschieben.
	Decrease Indent	Markierten Text komplett nach links verschieben.
	Font Color	Textfarbe auswählen.
	Background Color	Hintergrundfarbe der Webseite auswählen
	Table properties	Verschiedene Tabelleneigenschaften.
	Row properties	Verschiedene Zeileneigenschaften.
	Insert row before	Eine Zeile vor Cursor einfügen.
	Insert row after	Eine Zeile nach Cursor einfügen.

Symbol	***Name***	***Funktion***
	Delete row	Eine Zeile löschen.
	Insert column before	Eine Spalte vor Cursor einfügen.
	Insert column after	Eine Spalte nach Cursor einfügen.
	Delete column	Eine Spalte löschen.
	Cell Properties	Verschiedene Zelleneigenschaften.
	Insert cell before	Eine Zelle vor Cursor einfügen.
	Insert cell after	Eine Zelle nach Cursor einfügen.
	Delete cell	Eine Zelle löschen.
	Cell merge	Zellen miteinander vermischen.

Anhang F: Anleitung XML Content

überarbeitet nach Alexander Kandzior [33]

1. Start with the XML schema

You first need to define the XML content structure. To do this, create an XML schema (currently this is just a "plain" file). The XML schema mut follow the layout of the demo files. Check out "/xmlcontent/article.xsd" and all files in "/system/modules/org.OpenCms.frontend.templateone/ schemas/" which give you a couple of examples. Please note that the layout is mandatory, in case you make any mistake it wont word. A fine point is that there's always an "inner" and "outer" node name, e.g. "article" and "articles". The "outer" name _must_ be the inner name with an "s" appended. For the current choice of available data types check the demos and the file "OpenCms-vfs.xml". For the beginning, I recommend you start with a very small schema, maybe based on "article.xsd" where you _remove_ most of the nodes, and rename the schema to "myarticle". Just leave the title data field. The first schema might look like this:

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xsd:include schemaLocation="opencms://opencms-xmlcontent.xsd"/>
  <xsd:element name="myarticles" type="OpenCmsMyarticles"/>
  <xsd:complexType name="OpenCmsMyarticles">
    <xsd:sequence>
      <xsd:element name="myarticle" type="OpenCmsMyarticle" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="OpenCmsMyarticle">
    <xsd:sequence>
      <xsd:element name="Title" type="OpenCmsString" />
    </xsd:sequence>
    <xsd:attribute name="language" type="OpenCmsLocale" use="required"/>
  </xsd:complexType>

  <xsd:annotation>
    <xsd:appinfo>
      <mapping element="Title" mapto="property:Title" />
    </xsd:appinfo>
  </xsd:annotation>
</xsd:schema>
```

2. Create a content with the new schema.

Since the schema is not yet available for creating "new" files, you must do that manually. Create a new file of type "Generic XML content". This will be just an empty file. Now open that file with "Edit controlcode". Then supply a valid XML for the schema, like this:

```
<?xml version="1.0" encoding="UTF-8"?>

<myarticles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="opencms://spielplatz/myfolder/myarticle.xsd">
  <myarticle language="en">
    <title><![CDATA[Sample article 1]]></title>
  </myarticle>
</myarticles>
```

The fine point here's the pointer to the XML schema in the VFS using xsi:noNamespaceSchemaLocation="OpenCms://sites/default/xmlcontent/myarticle.xsd". You must point to the location of your schema here.

Note that the "Edit controlcode" editor will not allow you to save the content if either the schema contains errors, or the XML is not formed according to the schema. So if you can't save, check that both the schema and the XML is vaild. Once you are able to save, you are certain that it worked.

3. Open the created file using "Edit".

Now you should see the "nice" editor, that displays the node name "title" and an input field for that.

4. Create a "detail" page.

The demo contains a "full featured" example. Here's like the "minimum version", without a template:

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms" %>

<cms:contentload collector="singleFile" param="${OpenCms:uri}" editable="true">
<h1><cms:contentshow element="Title" /></h1>
</cms:contentload>
```

Save this file as "mydetail.jsp".

5. Connect the detail page to the XMl content using the "template-elements"

property.

At this point, I recommend moving everything into a new folder, called e.g. "my-folder". This folder should now contain the XML schema, the one "myarticle" created and the detail page "mydetail.jsp". Note that you must manually correct the schema location in the "myarticle" since the file was moven in the VFS.
Now, open the properties of "myfolder", and switch to "Advanced" mode. Locate the "template-elements" property on the lower half of the screen. Into this propety you must point to the location of the detail page, so add "/myfolder/my-detail.jsp" as propety value.
Save the properties and go into the "myfolder" again. Click on the created "myarticle". It should now be displayed using the created detail page. Note that the reason why in our demos all XML files and with ".html" is to make sure the "html" mimetype is used. If your XMl content does not end with .html, there might be issues displaying it correctly because of a wring mime type.

6. Create an overview list page

Now create a JSP page "mylist.jsp" in "myfolder". The mimimum version might look like this:

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms" %>

<cms:contentload collector="allInFolderDateReleasedDesc"
param="/spielplatz/myfolder/myarticle_${number}.html|11" editable="true">
<h3><cms:contentshow element="Title" /></h3>
</cms:contentload>
```

Click on the page. It should now display your single myarticle. Note that you can now create a new article using the "wizard wand". Add some more myarticles. They will be named "myarticle_0001.html" and up.

The "param" attribute controls what files are actually listed. param="/myfolder/myarticle_${number}.html|11" translates to: All contents of type 11 (generic XML content) in "/myfolder/". New files should be named "myarticle_${number}.html". You can also get the folder part from a property by using the "property" attribute and leavin this out of "param", see the provided demo for that.

7. Localizing the XML content editor and providing help texts

Check the folder "/system/modules/org.OpenCms.frontend.templateone. xmlcon-

tentdemo/classes/org/OpenCms/frontend/templateone/xmlcontentdemo/". If contains the workplace translations and lables for the editor. Syntax should be pretty clear. Note that you must restart OpenCms / Tomact if you make changes to one of the properties files. Note that if you create your own module, the "classes" folder must get exported as export point.

8. Add own resource type and workplace context menu

Last thing to do is to add an own resource type for you new content. Check out the "OpenCms-vfs.xml" in the WEB-INF/config folder. There you should find the node for the "article" content. Duplicate this node for your "myarticle". You also see that it points to the location of the schema. You also must give your type a new number. This may look like this:

```
<type class="org.OpenCms.file.types.CmsResourceTypeXmlContent">
   <param name="resource.type.id">13</param>
   <param name="resource.type.name">myarticle</param>
   <param name="schema">/sites/default/myfolder/myarticle.xsd</param>
</type>
```

Then add the context menu, which is done in "OpenCms-workplace.xml", and it may look like this:

```
<explorertype name="myarticle" key="fileicon.myarticle" icon="myarticle.gif"
reference="xmlcontent">
   <newresource
uri="newresource_xmlcontent.jsp?newresourcetype=myarticle" order="25"/>
</explorertype>
```

Note that the "key" refers to a localization entry, so this must be located in a workplace property file (see 7). The image "myarticle.gif" must be located in "/system/workplace/resources/filetypes/".

You need to restart OpenCms after these changes are made. After this, you should be able to create a new "myarticle" type with the "new" function in the workplace.

Alexander Kandzior
Alkacon Software - The OpenCms Experts
http://www.alkacon.com

9. Literaturverzeichnis

[1] Roeder, N.; Hensen,·P.; Hindle, D.; Loskamp, N.; Lakomek, H.-J.: Instrumente zur Behandlungsoptimierung Klinische Behandlungspfade. In: Chirurg 74: Seiten 1149-1155, 2003

[2] Kox, WJ.; Spies,C. (Hrsg.): Check-up Anästhesiologie. 1. Auflage, Berlin: Springer Verlag, 2003

[3] Bauer, M.; Hanß, R.; Schleppers, A.; Steinfath, M.; Tonner, P. H.; Martin, J.: Prozessoptimierung im „kranken Haus". In: Anaesthesist 53:Seiten 414-425, 2004

[4] Das Deutsche Cochrane Zentrum an der Universität Freibung, http://www.cochrane.de/deutsch/index.html, zuletzt gesehen am 20.09.2005

[5] Martin, J. ; Kuhlen, R.; Kastrup, M.; Schleppers, A., Spies,C.: Die Standard-operating-procedures- Tauschbörse Anästhesiologie, Intensivmedizin, Schmerztherapie und Notfallmedizin. In: Anaesthesist 54:Seiten 495-496, 2005

[6] Roeder, N. ; Hindle, D.; Loslamp, N.; Juhra, C.; Hensen, P.; Bunzenmeier, H.; Rochell, B.: Frischer Wind mit klinischen Behandlungsfaden (I), Instrumente zur Verbesserung der Organisation klinischer Prozesse. In: Das Krankenhaus 1, Seiten 20-27, 2003

[7] Roeder, N. ; Hindle, D.; Loslamp, N.; Juhra, V; Hensen, P.; Bunzenmeier, H.; Rochell, B.: Frischer Wind mit klinischen Behandlungsfaden (II), Instrumente zur Verbesserung der Organisation klinischer Prozesse. In: Das Krankenhaus 2, Seiten 124-130, 2003

[8] Martin, J. ; Schleppers, A.; Kastrup, M.; Kobylinski, C.; König, U.; Kox, W.J.; Milewski, P.; Spies, C.: Entwicklung von Standard Operating Procedures in der Anästhesie und Intensivmedizin., In: Anästhesiologie & Intensivmedizin 44: 871-876, 2003

[9] Sackett, D.L.; Rosenberg, W.M.C.; Gray,J.A.: Evidence based medicine: what it is and what it isnt., In: BMJ, Vol.12: pp. 71-72, 1996

[10] Robertson, J.: The value of intranet feedback In: Cm Briefing,17, 2003 www.steptwo.com.au/papers, zuletzt gesehen am 12.9.2005

[11] Hildebrand, R. : Unterlagen von www.hmanage.de, Dr. Hildebrand & Partner GmbH, zuletzt gesehen am 12.09.2005

[12] Martin, J.; Schleppers, A.; Kastrup, M.; Kobylinski, C.; König, U.; Kox, W.J.; Milewski, P.; Spies, C.: Entwicklung von Standard Operating Procedures in der Anästhesie und Intensivmedizin., In: Anästhesiologie & Intensivmedizin, 44: Seiten 871-876, 2003

[13] Bundesministerium für Gesundheit und Soziale Sicherung, Gesundheitspolitik: Rahmenbedingungen, http://www.die-gesundheitsreform.de/gesundheitspolitik/rahmenbedingungen/ zuletzt

gesehen am 20.09.2005
[14] Heffner, JE : Does evidence-based medicine help the development of clinical practice guidelines?.In: Chest, Vol. 113: pp. 172-178, 1998
[15] Gütl,C.: Ansätze zur modernen Wissensauffindung im Internet. Eine Annäherung an das Information Gathering and Organizing System xFIND (Extended Framework for INformation Discovery). 2002, Dissertation an der Technischen Universität Graz, http://www.iicm.edu/thesis/cguetl_diss, zuletzt gesehen am 19.09.2005
[16] Glantschnig, P. : Innovative Content Management Systeme im Betrachtungsfeld von Java 2 Enterprise Edition.2004, Diplomarbeit an der Technischen Universität Graz, http://www.iicm.edu/thesis/pglantschnig/html/da-cms.htm, zulezt gesehen 19.09.2005
[17] ContentManager.de : www.contentmanager.de (Stand 12.09.2005), zuletzt gesehen am 12.09.2005
[18] Rothfuss, G; Ried C.(Hrsg): Content Management mit XML. Kapitel 4.2 Content Management als Prozess, Seite 52, 1. Auflage, Berlin: Springer Verlag, 2001
[19] Albrecht, M.; Cutkovic, O; Jetter,M: Content Management Systeme - Software und deren Einsatzfelder für PR. Fachhochschule Stuttgart, Hochschule der Medien, http://v.hdm-stuttgart.de/projekte/websitepr/content.php, zuletzt gesehen am 12.9.2005
[20] Butcher, M: Building Web Sites with OpenCms. a: chapter 1 What Is a Content Management System?, pp 5; b: chapter 3 Module pp 76; c: chapter 4 JSP Tag Libraries and Scriptlets, pp 91; d: chapter 2 Installing OpenCms. pp 13-24; 1st.ed. , Birmingham: Fackt Publishing, 2004
[21] Jagiela, G.; Kambouz, S.; Ruppert, P.; Zorembsky, D.: Content Management Syteme, Vertiefungsfach Prozessdaten Verarbeitung. 2001 FH Wiesbaden unter Prof. Dr. Linn, http://www.informatik.fh-wiesbaden.de/~linn/vpdv01/cms/start.htm, zuletzt gesehen am 12.9.2005
[22] Gruhn, V.; Heymann, D.; Kleine, M.: Eine Architektur für Content-Management-Systeme auf der Basis von XML. In: Konferenz XML 2000 Germany, Magdeburg, 2000 http://ebus.informatik.uni-leipzig.de/www/media/publications/xml00.pdf, zuletzt gesehen am 12.9.2005
[23] Berg, K. P.: Softwareevolution und Wartung: Situationsanalyse und Entwicklungsmöglichkeiten. ObjektSpektrum, Sigs Verlag, http://www.sigs.de/publications/os/2005/02/berg_OS_02_05.pdf, zuletzt gesehen am am 12.09.05
[24] Sommerville, I.: Software engineering. 7nd ed., a: chapter 21.2, pp 492-498, b: chapter 18, pp.415-438. Boston: Pearson/Addison Wesley, 2004
[25] IEEE – The Institute of Electrical and Electronics Engineers: IEEE Standard for Software Maintenance. In: IEEE Std. 1219-1998: New York, N.Y.

[26] ISO- International Organization for Standardization: ISO IEC 12207 Standard for Information Technology Software Lifecycle Processes, Genf, Schweiz, 1995
[27] Weidl-Rektenwald, J.: Software Wartung und Evolution. a: Lecture 1, b: Lecture 2 , c: Lecture 4, Vorlesungsunterlagen, Web-Skriptum 2004 http://www.infosys.tuwien.ac.at/Teaching/Courses/SWE/swe.html, zuletzt gesehen am 2.9.2005
[28] Balzert, H.: Lehrbuch der Software-Technik Software-Entwicklung, a:Kapitel 6. Die Wartungs & Pflegephase, Seiten 1090-1100; b: Anhang A Richtinien, Konventionen und Normen Abb.A-3a, Seiten 1102-1103; 2. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2001
[29] Kuhlmann,U. ;Winter, A.: Softwarewartung und Prozessmodelle in Theorie und Praxis, http://www.uni-koblenz.de/~kuhlurs/papers/Kuhlmann2004b.pdf, zuletzt gesehen am 19.09, 2005
[30] Parnas, D. L. : Software Aging. In: Proc. ICSE Proceedings of the 16th international conference on Software engineering, Sorrento (Italien), 16-21. Mai 1994, pp. 279-287
[31] Schönleber, T.:Entwicklung eines webbasierten Systems zur Erfassung und Bearbeitung von Modification Requests. Diplomarbeit, http://elib.uni-stuttgart.de/opus/volltexte/2005/2261/pdf/DIP_2257.pdf, zuletzt gesehen am 10.10.2005
[32] Merkert, O.: Softwarewartung Situationsanalyse und Entwicklungsmöglichkeiten, www.merkert.claranet.de/deutscheEssays/spl.html, zuletzt gesehen am 12.09.2005
[33] Alexander Kandzior: [opencms-dev] XML Content Demo. OpenCms Mailingliste, http://mail.opencms.org/pipermail/opencms-dev/2004q4/013609.html, zuletzt gesehen am 10.08.2005
[34] Moser, H.: Auswirkungen von Code Conventions auf Softwarewartung und Evolution, http://www.heinzi.at/texte/codeconv.pdf, zuletzt gesehen 16.09.2005
[35] SOP-Creator, http://www.lymphome.de/Projekte/TP02/SOPCreator/, zuletzt gesehen am 19.09.2005
[36] Weber, M.: Entwurf und prototypische Umsetzung der grundlegenden Funktionen eines generischen Content Management Systems, basierend auf Java-Servlets und XML. http://home.of.the.brave.de/Aktuell/Jobs/Diplomarbeiten/Diplomarbeit_Mathias_Weber.pdf, Diplomarbeit an der Fachhochschule Oldenburg, zuletzt gesehen am 20.09.2005
[37] Martischnig A.; Opelt A.: JAVA Middleware für Web basiertes CMS. http://www2.iicm.edu/cguetl/education/projects/martischnig_opelt/project/Projektbeschreibung_Opelt.pdf , zuletzt gesehen am am 20.09.2005
[38] Kotlorz, T.: Die Charité muß gesundschrumpfen. Die Welt, 21.05.2005, http://www.welt.de/data/2005/05/21/721101.html, zuletzt gesehen am

30.09.2005
[39] Grubb, P.; Takang, A. A.: Software Maintenance Concepts and Practice. chapter 5.5, pp. 71-85, 2 edition, World Scientific 2003
[40] Wikipedia, http://de.wikipedia.org, Stand Herbst 2004
[41] DRG Zeitung: Wenn Krankenhäuser zu Unternehmen werden: http://www.medizininformatik-treffpunkt.de/apps/dertag/medinfoweb_040609.pdf, Online-Ausgabe vom 09.06.2004, zuletzt gesehen am 20.09.2005
[42] Lehman, M. M; Ramil, J. F.; Wernick, P.D.; Perry, D. E.: Metrics and laws of software evolution the nineties view. Proc. Fourth Int. Symp. on Software Metrics, Metrics 97, Albuquerque, New Mexico, 5-7 Nov. 97, pp 20-32
[43] SOP-Creator Handbuch, http://www.onto-med.de/de/publikationen/sopcreator.pdf, zuletzt gesehen am am 20.09.2005
[44] Feil, T.: Abschlusszwang oder Die Pflicht zur Wartung. http://www.recht-freundlich.de/download/Abschlusszwang.pdf, zuletzt gesehen am 24.09.2005
[45] Opferkuch, S.: Software-Wartungsprozesse ein Einblick in die Industrie. http://www.iste.uni-stuttgart.de/se/people/opferkuch/Wartungsprozesse.pdf, zuletzt gesehen am 24.09.2005
[46] OpenCms 5.0 basic documentation, http://www.opencms.org/export/download/opencms/opencms_5.0_doc_full.zip, zuletzt gesehen am 20.09.2005
[47] OpenCms 5.0 user introduction, http://www.opencms.org/export/download/opencms/opencms_5.0_doc_user.zip, zuletzt gesehen am 20.09.2005
[48] Balzert, H.: Lehrbuch der Software-Technik Software-Management Software-Qualitätssicherung Unternehmensmodellierung. Kapitel IV 4. Sanierung, Seiten 663-686 2. Auflage. Heidelberg: Spektrum Akademischer Verlag, 1998
[49] Kern, H. M.: Wartung und Reengineering objektorientierter Software. http://www.iste.uni-stuttgart.de/ps/Lehre/HS_OO_Entwurf/Kern.pdf, zuletzt gesehen am 23.09.2005
[50] Bisbal, J.; Lawless, D.; Wu, B.; Grimson, J.:Legacy Information Systems: Issues and Directions. IN: IEEE Software, Vol.16, No. 5, pp 103-111 1999
[51] Schleifer, S.; Grabmeyer, F.: Wie vermeidet man Fehler in der Software? Präventive Maßnahmen. Seminararbeit, http://www.iivs.de/haag/buerger/fgrabm/studium/Fehlervermeidung.pdf, zuletzt gesehen am 30.09.2005
[52] Kandzior, A.: Maßkonvektion Strukturierte Inhalte mit OpenCms 6.0. In: Linux Magazin, Seiten 9-16, Sonderheft 2/2005
[53] Kajko-Mattsson, M.; Forssander, S.; Olsson, U.: Corrective Maintenance Maturity Model (CM3): Maintainer's Education and Training. Proceedings of

the 23rd international conference on Software engineering, pp.610-619, May 12-19, 2001, Toronto, Ontario, Canada
[54] Thuranszky, S.: Diplomarbeit, http://informatik.fh-nuernberg.de/professors/Holl/Personal/DA-AntiAging.doc, zuletzt gesehen am 27.09.2005
[55] Arbeitsunterlagen der Arbeitsgruppe SOP an der Klinik für Anästhesie und operative Intensivmedizin, Campus Virchow Klinikum, Charité

10. Erklärung

Ich erkläre hiermit, dass ich die Arbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie Zitate als solche gekennzeichnet habe.

Berlin, den