

**Diplomarbeit vom 6. September bis 29. Oktober 2001**

---

Studenten: José Fontanil und Reto Glanzmann

Dozent: Patrick Feisthammel

## **PhPepperShop - Bericht**



## Bericht

Dieser Bericht ist ein Auszug aus der kompletten Diplomarbeitdokumentation. Sie enthält viele Erfahrungen die wir gerne weitergeben möchten.

## Inhaltsverzeichnis

### 1. Bericht

1.1 Kurzbeschreibung in Deutsch und Englisch.....	5
1.1.1 Kurzbeschreibung (Deutsch).....	5
1.1.2 Abstract (English).....	5
1.2 Vorwort.....	7
1.3 Aufgabenstellung.....	8
Umfeld.....	8
Aufgaben.....	8
Aufgabendetails.....	8
Termine.....	9
Kontaktmöglichkeiten.....	9
Abgabe.....	9
Bewertung.....	9
1.4 Analyse des erhaltenen Feedbacks.....	10
1.4.1 E-Mail Auswertung.....	10
Weitere Tipps aus den E-Mails.....	10
1.4.2 Auflistung aller erhaltenen Feedbacks.....	11
1.4.3 Erste Vorentscheidungen.....	12
1.5 Wahl der verwendeten Technologien.....	14
1.6 Zeitplanung.....	16
1.6.1 Aufgabenaufstellung nach Wichtigkeit, 17.09.2001.....	16
1.6.2 Zeitplanung, erstellt zu Projektbeginn.....	18
1.6.3 Effektiver Ablauf des Projekts.....	18
1.7 Tests.....	20
1.7.1 Usability-Test.....	20
1.7.2 Extremwerte Test.....	22
1.7.3 Funktionstest.....	23
Test des CRM's (Customer Relationship Management).....	24
Test des Bestellungsablaufs.....	25
1.8 Erfahrungen.....	26
1.8.1 Installation der Systemumgebung - LAMPS.....	26
Inhalt dieses Kapitels.....	26
Wieso diese Kombination?.....	26
Installation.....	27
Eigentliche Installation.....	27
Entpacken der Pakete.....	27
MySQL.....	28
SSL Installation.....	28
Apache und mod_SSL.....	28
Server-Zertifikat erstellen.....	29
Apache Installation vervollständigen.....	29
PHP.....	30

Konfiguration.....	32
Starten.....	32
1.8.2 Einrichten eines Passwort geschützten Bereichs mit htaccess.....	33
1.8.3 SourceForge einrichten.....	35
Einleitung.....	35
SourceForge einrichten.....	35
CVS einrichten.....	39
Was ist CVS?.....	39
CVS Entwicklungszyklus.....	39
CVS für sein Projekt auf SourceForge einrichten.....	39
Weitere Services von SourceForge.....	42
Fragen zu SourceForge.....	43
1.8.4 Browser Kompatibilität.....	44
1.8.5 PHP und Safe-Mode.....	45
1.8.6 GNU-PG.....	46
1.8.7 Strukturen beim Webprogramming.....	47
Variablentransport zwischen Funktionen.....	47
Andere Fehlermeldungen.....	48
Interaktivität einer HTML-Seite.....	49
Entwurf von neuen Komponenten für den PhPepperShop: (Entwicklungszyklus)....	50
1.8.8 Variablen zwischen den Webpages transportieren.....	52
Übermittlung direkt im Link (GET).....	52
Übermittlung als Feld in einem Formular (POST oder GET).....	52
Welche Daten sollen wie übergeben werden?.....	53
1.8.9 JavaScript.....	56
Gleichzeitige Aktualisierung von Frames mittels JavaScript.....	56
Überprüfung eines Formularfeldes auf Eingabe.....	57
Unsere Meinung zu JavaScript.....	58
1.8.10 Datenbankdesign.....	59
Unterkategorien.....	59
Shop-Settings.....	59
1.9 Zukunft des PhPepperShops.....	61
1.9.1 Veröffentlichung des Final-Release.....	61
1.9.2 Leute, die den Shop weiterentwickeln wollen.....	61
1.10 Schlusswort.....	63
<b>2. Anhang</b>	
2.1 Literaturverzeichnis.....	64
2.1.1 PHP.....	64
2.1.2 Apache Webserver.....	65
2.1.3 CRM – Customer Relationship Management.....	65
2.1.4 Datenbanken und im speziellen MySQL.....	65
2.1.5 HTML, JavaScript und Cascading Style Sheets.....	66
2.1.6 Perl.....	66
2.2 Software Versionen.....	67
2.2.1 Auf dem Server eingesetzte Software.....	67
2.2.2 Auf den Arbeitsstationen eingesetzte Software.....	67
2.3 Projekttagbuch.....	68
2.4 GPL – GNU General Public Licence.....	73
2.4.1 Deutsche Fassung der GPL (Auszug).....	73

# 1. Bericht

## 1.1 Kurzbeschreibung in Deutsch und Englisch

### 1.1.1 Kurzbeschreibung (Deutsch)

Die Entwicklung eines frei verfügbaren, beliebig konfigurier- und erweiterbaren Open Source Webshops war das Ziel, welches wir uns zu Beginn unserer Diplomarbeit setzten.

Entstanden ist ein Shopsystem, welches Artikel mit beliebig vielen Optionen und Varianten unterstützt. Für jeden Artikel kann ein beliebig grosses Artikelbild gespeichert werden, von welchem automatisch ein Thumbnail (kleines Vorschaubild mit fest definierter Grösse) erzeugt wird. Der Shopkunde kann sich bei Bedarf seine Adressdaten benutzername- und passwortgeschützt speichern lassen, damit bei einem späteren Einkauf die Adresse nicht nochmals eingegeben werden muss. Per Suchfunktion kann komfortabel im Shop nach bestimmten Artikeln gesucht werden. Verschiedene frei konfigurierbare Zahlungsarten sind implementiert, so dass mittels Payment-Schnittstelle mit wenig Aufwand die Anbindung an ein Online-Payment Institut möglich ist.

Der Shopadministrator tätigt alle Einstellungen direkt per Webinterface. Ein umfangreiches Layout-Management ermöglicht es, das Shopaussehen bis ins letzte Detail zu konfigurieren (Hintergrundbilder, Framegrösse, Shoplogo, eigene Buttons). Artikel können erstellt, verändert oder gelöscht werden. Ausserdem ist es möglich, einen Artikel mehreren Kategorien zuzuordnen. Per Kategorienmanagement können Kategorien und Unterkategorien erstellt, umbenannt, verschoben und gelöscht werden. Optional kann für das Entgegennehmen der Kreditkartendaten eine SSL-Unterstützung aktiviert werden. Versandkosten werden nach Gewicht, Preis oder Pauschale verrechnet. Ein Mindermengenzuschlag und Nachnahmegebühren können dynamisch zum Rechnungstotal addiert werden.

Der Shop wurde in den Scriptsprachen PHP und JavaScript entwickelt. Die HTML-Seiten werden dynamisch per PHP erzeugt und Benutzereingaben mittels JavaScript auf ihre Korrektheit überprüft. Alle Daten werden in einer MySQL-Datenbank gespeichert. Die Installation erfolgt bequem mittels eines Perl-Scripts.

Wie schon im Titel erwähnt, wurde der Shop als Open Source Projekt auf SourceForge entwickelt. Das heisst, dass der Quellcode frei verfügbar ist und der Shop uneingeschränkt verwendet und weiterentwickelt werden kann. Aktuelle Informationen und einen Beispielshop finden sich auf der Seite <http://www.phpeppershop.com>.

### 1.1.2 Abstract (English)

Our aim was to develop a web shop that could be configured flexibly, distributed freely and is open source under the GNU-General Public Licence. What has arisen is a shopping system that offers articles with infinite options and variations.

A picture of a desired size can be assigned to each article. Due to an automatically generated thumbnail of the before mentioned picture, the customer is given a consistent view of the articles.

The customer can optionally save his address. He can then access his record by simply logging

in with a login name and password. This way he doesn't have to enter his address more than once. The shop provides comfortable search functions to find articles. Several types of payment are supported: cash before delivery, cash on delivery, on account and credit card payment. Thanks to a flexible payment-interface it is possible to link the PhPepperShop to an online-payment institute with ease. PhPepperShop can optionally encrypt the credit card input via SSL.

The shop administrator can access the web shop remotely from any location via a web interface. A flexible layout allows for changing the look of the shop in detail (background images, frame size, shop logo, buttons, etc.). Articles can be created, updated and deleted. An article can even be assigned to more than one category. With the category management the shop administrator can easily create, rename, replace and delete categories and sub categories. Shipping costs can be charged by price, by weight or by an estimated amount. Small amount surcharge and pay on delivery surcharge can be added to the invoice automatically.

The web shop has been programmed using PHP and JavaScript. The HTML pages are generated by PHP, JavaScript validates customer input. The whole shop is saved in a MySQL database. The installation is unlike UNIX with an automatic installation tool written in PERL.

As mentioned in the title, the PhPepperShop is an open source project, which includes free availability of the source code and unrestricted utilisation and development. The PhPepperShop project is maintained on Source Forge. The latest news and a sample shop can be found on [www.phpeppershop.com](http://www.phpeppershop.com).

## 1.2 Vorwort

Im Rahmen unserer Diplomarbeit erweiterten wir unsere erste Projektarbeit "Webshop Framework in PHP" zu einem marktreifen Produkt, dem PhPepperShop.

Das wichtigste Ziel für uns war, den Shop soweit zu entwickeln, dass man ihn ohne Bedenken der Öffentlichkeit übergeben kann. Wir wollten den Shop so programmieren, dass er möglichst vielseitig und branchenunabhängig eingesetzt werden kann. Weiter sollte sich der Shop komplett auf die Bedürfnisse des Shopbetreibers anpassen lassen. Dabei berücksichtigten wir auch die Wünsche und Verbesserungsvorschläge des inzwischen bei uns eingegangenen Feedbacks.

Ein von Anfang an erklärtes Ziel war es, den Shop frei von lizenzrechtlichen Einschränkungen auf einem öffentlich zugänglichen Server gehostet anzubieten. Wir vertreten mit voller Überzeugung die Idee der Open Source Software Entwicklung.

## 1.3 Aufgabenstellung

# Diplomarbeit: Open Source Webshop "PhPepperShop"

J. Fontanil  
R. Glanzmann

### **Umfeld**

Ziel dieser Diplomarbeit ist die Erweiterung des in einer Projektarbeit erstellten *Framework für einen Webshop in PHP*.

Der Shop soll funktionell soweit fertiggestellt werden, dass der praktische Einsatz möglich ist. Dazu soll der Shop als Open Source Projekt publiziert werden und wenn möglich eine *echte (praktisch eingesetzte)* Installation erfolgen. Die daraus gewonnenen Erfahrungen sollen umgesetzt werden.

### **Aufgaben**

Im Umfang der Diplomarbeit sind folgende Punkte zu erfüllen:

- Erstellen des WebShops gemäss *Aufgabedetails*
- Erstellen der Dokumentation

Design-Entscheidungen sind zu dokumentieren und zu begründen.

Die Dokumentation hat zudem mindestens zu umfassen:

- Architektur (Komponenten und deren Funktionalität, sowie Workflow)
- Projekttagbuch (Zeitplanung, Probleme, Entscheidungen, Erfahrungen)
- Ausblick (Verbesserungsmöglichkeiten, Lehren)

Die im Rahmen der Publizierung veröffentlichten Dokumente sind Bestandteil der Dokumentation.

### **Aufgabedetails**

Es geht in dieser Arbeit darum den WebShop zu ergänzen mit:

- Custom Relationship Management (CRM)
- Layout Management mit CSS
- Online Payment
- Implementation der wichtigsten Zusatzfunktionen gemäss Besprechung zu Beginn der Diplomarbeit
- Fertigstellung des Administrationstools

Zudem sollen Erfahrungen mit der Publikation eines Open Source Projekts auf [SourceForge](#) gesammelt und dokumentiert werden.

Falls genügend Zeit vorhanden ist, sind weitere Ergänzungen denkbar:

- Automatische Katalogerzeugung im PDF Format
- Offline Administration
- Erweiterung der Funktionalität

## **Termine**

Donnerstag, 6. September 2001	Ausgabe der Diplomarbeit
Montag, 29. Oktober 2001	Abgabe der Arbeit
Montag, 12. November 2001	Präsentation der Diplomarbeit
Donnerstag, 22. November 2001	Industriepräsentation
Samstag, 24. November 2001	Umzug, Ausstellung

## **Kontaktmöglichkeiten**

Ihr Betreuer ist per Telefon und E-Mail erreichbar. Die Details werden in der ersten Besprechung angegeben.

## **Abgabe**

Alle Daten (Dokumente, Programmcode, etc.) sind vollständig elektronisch abzugeben. Benötigte Zusatzsoftware (falls lizenzrechtlich erlaubt) ist ebenfalls elektronisch abzugeben.

Die Berichte sind zusätzlich in gedruckter Form abzugeben.

## **Bewertung**

Die Bewertung der Arbeit erfolgt mit folgendem Bewertungsschema:

<b>Kriterium</b>	<b>Gewicht</b>
Fleiss, Einsatz, Interesse	1
Methodik, Vorgehen bei der Problemlösung	3
Eigeninitiative, Kreativität, Originalität	4
Gründlichkeit, theoretische Betrachtungen	3
Erarbeitetes Wissen, Literatur, Vollständigkeit	3
<b>Implementation</b>	
Techn. Korrektheit, Bedienbarkeit der Programme	2
Dokumentation, Verständlichkeit Programmcode	2
<b>Vortrag</b>	
Eingehen auf das Wesentliche, Einsatz der Medien	2
Verständlichkeit, Einhalten der Zeit	2
<b>Bericht</b>	
Gliederung, Darstellung	3
Technische Korrektheit, Referenzen	3
Umfang, Vollständigkeit, Sprache	2

Näheres zum Projekt findet man auf: <http://www.phpeppershop.com>  
<http://www.zhwin.ch/~fei/da/>



## 1.4 Analyse des erhaltenen Feedbacks

Unsere Diplomarbeit setzt auf der Projektarbeit 'Framework für einen Webshop in PHP' (PA Fei01/2) auf. Wir veröffentlichten die Arbeit am 19. Juni 2001 und haben seither über 15'000 Hits auf dieser Homepage aus den verschiedensten Ländern verzeichnet. Da ist es klar, dass uns auch einige Leute E-Mails geschrieben haben. Weiter haben wir Feedbacks von mehreren freiwilligen Shop-Testern erhalten. Zudem gaben uns auch Shopbetreiber Tipps.

### 1.4.1 E-Mail Auswertung

Anzahl	Vorschlag
7	Englische Sprache erwünscht (allgemein Mehrsprachensupport)
1	Englischer Code erwünscht (vor allem Kommentare)
9	Provider Probleme ? 1 User, (ev. auch nur eine Datenbank), detaillierte Installationsanleitung
1	htaccess erklären
2	SSL allgemein, und vor allem bei Formular Übertragung zur Bank / yellowpay / SECPay.com
1	Suchfunktionen auf der Homepage
1	Stern-Zeichen als Admin-Login wegtun ? ersetzen durch Admin-Entry-Page
5	Windows Version des Shops (nur NT / 2000)
1	Mehrwertsteuer und Verpackungs-/Frachtkosten in die Berechnung miteinbeziehen
2	Lagerverwaltung: Bestand eingeben, bei Verkauf dekrementieren -> Minderungen Alarm
1	Verfügbarkeit des Artikels angeben und ev. Lieferterminangaben
1	Dynamisches Page Layout (auch: Header und Footer)
2	Unterkategorien (z.B. 300 Kategorien und mehr Produkte)
3	CRM, personalisierte Artikelliste, eigenes Login, autom. ausgefüllte Adresse (weiter: persönlicher Ansprechpartner konfigurieren können, Rechnung, Lieferschein drucken können, Bestellungsfortschritt abrufbar(versendet, ...))
1	Shop muss mit root-Rechten laufen (mühsam, wegen Bild Upload)
1	Developer-Mailing Liste
1	CVS-Server
3	Gleicher Artikel mit verschiedenen Variationen/Optionen können nicht kumuliert werden
1	Optionen/Variationen als Pop-Up Menü implementieren (ev. nur optional)

### Weitere Tipps aus den E-Mails

- Wir sollten Webmastern den Shop empfehlen (Marketing)
- Artikel in der c't oder ähnlichen Magazinen anstreben

- Wir sollten das Problem der Secure-Online-Transactions genau anschauen (P. Fitzpatrick)

### 1.4.2 Auflistung aller erhaltenen Feedbacks

Wir haben folgende Liste mit möglichen Arbeiten aus allem erhaltenen Feedback zusammengestellt und im Rahmen der Zeitplanung (Kapitel 1.6) einer Priorisierung und Zeitaufwandschätzung unterzogen.

<i>Arbeitsvorschläge</i>
Shopname suchen und definieren
Shop Administrationsmodus fertig stellen
Formular Buttons durch grafische Buttons ersetzen
Benutzerinfo, dass ohne E-Mail Adresse keine Bestellbestätigung ausgestellt wird
Admin soll entscheiden können ob der Warenkorb direkt angezeigt werden soll
Aktionspreis implementieren
Artikel in mehreren Kategorien eintragen können
Anleitung für GIF / PNG Bilder Benutzung (für Artikel Bilder)
Warenkorb ist leer Meldung implementieren
Button hinzufügen, der mehr leere Options- / Variationsfelder 'generiert'
Kreditkartenunterstützung ? ev. noch Wrapperklasse für mehrere Payment Institute (Schnittstelle)
Importmöglichkeit (von anderen Quellen Artikeldaten importieren können)
Kategorienmanagement
Offline Administrationstool
Suchresultate optional mit Bilder
Anzahl Artikel pro Suchresultat-Seite festlegen (z.B. 25 Artikel pro Seite), Kunde kann wählen: 10 ... 50 ... alle
Suchanfrage mit mehreren Begriffen (à la Google®, AND-verknüpft)
Eigene Buttons (Buttonsets) ev. mit JavaScript (onmouseover, onclick,...)
Felderkonfiguration bei Kundenadresseingabe dynamisch wählbar gestalten (Admin) ? Reihenfolge wählbar
SSL ? GNU-PG ? E-Mail (Kreditkarten Nr., ... ) Mailorder-Vertrag, allgemein SSL anschauen
Kostenlose Artikel ? z.B. Prospekte bestellen ? Zahlungsmöglichkeiten weglassen können
PLZ kann auch alphanummerisch sein (USA, oder mit voran gestelltem Ländercode)
Backup des Shops (ev. auch automatisiert ? immerhin Anleitung)
Charset im Mailversand ergänzen
Englische Sprache erwünscht (allgemein Mehrsprachensupport)

Englischer Code erwünscht (v.a. Kommentare)
Provider Probleme ? 1 User, (ev. 1 Datenbank), detaillierte Installationsanleitung
htaccess erklären
Suchfunktionen auf der Homepage
Sternchenpixel als Admin-Login wegtun ? ersetzen durch Admin-Entry-Page
Windows (NT / 2000) Version des Shops ? Dozent nach Erfahrung fragen ? Links wegtun
Mehrwertsteuer und Verpackungs-/Frachtkosten in die Berechnung miteinbeziehen
Lagerverwaltung: Bestand eingeben, bei Verkauf dekrementieren -> Mindestmengen Alarm
Verfügbarkeit des Artikels angeben und ev. Lieferterminangaben
Dynamisches Page Layout --> CSS automatisch erzeugen lassen
Unterkategorien erlauben
CRM 1: Eigenes Login, automatisch ausgefüllte Adresse, Login zwingend/nicht zwingend ? Händlermodus
CRM 2: weiter: persönlicher Ansprechpartner konfigurieren können, Rechnung, Lieferschein drucken können, Bestellungsfortschritt abrufbar (versendet, ...)
Shop muss mit root-Rechten laufen (mühsam, wegen Bild Upload) ? Problem lösen
SourceForge: CVS(-Server), Developer-Mailing Liste, ...
Gleicher Artikel mit verschiedenen Variationen/Optionen können nicht kumuliert werden ? Fehler beheben
Variationen als optionales Pulldown-Menü implementieren
Scrolling im Top-Frame unterbinden
Shop-Logo upload ermöglichen ? Als (GIF-)Bilddatei im Bilder-Verzeichnis ablegen
Bilddatum für Cache auf 1.1.1971 korrigieren
Authentifizierung anstatt über htaccess über die PHPLib (PHPAuth)
Neuprogrammierung der Artikel-Update Funktion(-en)
Mehrere Shops in einer DB (Präfix bei Scripts u.s.w.)

### 1.4.3 Erste Vorentscheidungen

Wir haben uns nach einer groben Aufwandschätzung für das Weglassen von verschiedenen Punkten entscheiden müssen. Gegen die Ausführung der Arbeiten haben wir uns entschieden wenn:

- Die Arbeit als reine Fleissarbeit, ohne Lerneffekt angesehen wurde (z.B. Übersetzung des Shops, Windows Portierung).
- Die Implementierung eines funktionsfähigen, einsatzbereiten Shops aufgrund des Aufwands in Frage gestellt werden musste (z.B. CRM 2).

Uns war vor allem wichtig, dass der Shop im praktischen Einsatz bestehen kann. Versand-

kosten zum Beispiel waren einfach ein Muss, denn ohne Versandkostenrechnung kann man nicht wirklich einen Shop betreiben. Erst danach haben wir uns für Zusatzfunktionen wie eine erweiterte Artikelsuche entschieden.

Sind wir ein Problem angegangen, so entschieden wir uns wenn immer möglich dafür, in die jeweilige Teilarbeit auch mehrere Überlegungen einfließen zu lassen. Wir mussten zum Beispiel das Problem lösen, dass mehrere PhPepperShops gleichzeitig in einem DBMS laufen können. Weiter sollte die Installation besser dokumentiert werden, den 1-Datenbank-User Betrieb implementiert und die Möglichkeit geschaffen werden, mit schon bestehenden Datenbank-User arbeiten zu können. Wir haben alle diese Einzelprobleme zusammen gefasst und eine automatisierte Installation erstellt. Auf diese Weise sind alle Probleme elegant gelöst und dem Shopadministrator wird viel mühsame, fehlerträchtige Arbeit abgenommen.

Welche Arbeiten schliesslich ausgewählt wurden, sieht man im Kapitel 1.6, Zeitplanung. Dort ist auch ersichtlich, wie wir den Aufwand der einzelnen Arbeiten eingeschätzt haben.

## 1.5 Wahl der verwendeten Technologien

Ein Framework für einen Webshop lässt sich grundsätzlich auf viele Arten realisieren. In den folgenden Absätzen wollen wir begründen, wieso wir uns für die Kombination PHP4 als Apache-Modul und MySQL als Datenbank entschieden haben.

Ein vielfältiger Webshop sollte ein hohes Mass an Dynamik bieten, sowohl vom Workflow her, als auch innerhalb der HTML-Seite. Weiter sollte er seine Daten, wenn möglich, aus einer zentralen Datenbank beziehen können. Damit hat man auch alle vier Komponenten, die nötig sind um einen dynamischen Webshop realisieren zu können:

- Clientseitige Scriptsprache um statische HTML Seiten dynamisch zu verändern
- Serverseitige Scriptsprache zur Erstellung dynamischer Websites
- Webserver
- Datenbank

Um statischen HTML-Seiten Dynamik zu verleihen fällt die Wahl fast automatisch auf JavaScript. Diese umfassende Scriptsprache erlaubt es, eine bereits komplett herunter geladene Webpage dynamisch zu verändern. Man kann mit JavaScript auch einen Teil der Funktionalität, welche man serverseitig ausführen müsste, auf den Client verlegen, um so die Rechenlast auf dem Server zu minimieren. Es war uns bewusst, dass JavaScript viele Probleme mit sich bringen würde. Wir wollten aber auf die Dynamik, die damit gewonnen wird, nicht verzichten.

Da wir uns für eine Veröffentlichung des Webshops unter der GPL (siehe Anhang) entschieden haben, war es uns auch wichtig Technologien zu verwenden, die selbst Open Source sind. Deshalb wählten wir PHP4 (Personal Home Page Tools Version 4). Unter anderem weil PHP fürs Web erstellt wurde und nicht wie z.B. ASP, das mit VBScript daher kommt, welches nur 'schwer' an die Bedürfnisse eines Webservers angepasst werden kann und ausschliesslich für die Microsoft Plattform verfügbar ist. Perl hingegen bietet sehr mächtige Zeichenketten-Funktionen, hat aber den Nachteil einen eher unleserlichen Code zu eigen zu haben. PHP nimmt sich der Nachteile der oben genannten Technologien an und verbindet diese mit ihren Vorteilen. PHP bietet die von ASP bekannte, einfache Kombination mit HTML, die leistungsstarke und weit verbreitete C-Syntax, die vielen schon von JAVA und JavaScript her bekannt ist. PHP besticht ausserdem durch einen fast schon grandiosen Funktionsumfang. Hier kann keine andere Sprache mithalten. Das sieht man vor allem bei der Datenbank Unterstützung, die uns auch sehr wichtig war.

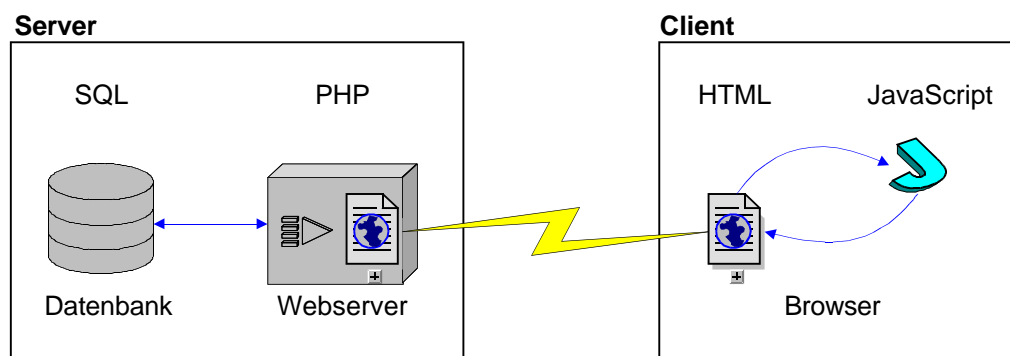
Die Entscheidung welchen Webserver wir verwenden sollten, war einfach zu treffen: Apache ist der heute im Internet am meisten eingesetzte Webserver: zuverlässig, Open Source und für die wichtigsten Plattformen verfügbar. Er hat auch grosse Zukunftsaussichten. Wir wollten den Einsatz des Shops aber nicht grundsätzlich an eine Webserver-Technologie binden. Es braucht nur minimalste Änderungen, um den Shop auf einem anderen Webserver laufen zu lassen (Passwortschutz für Administration und Pfade bei Wechsel von UNIX zur Windows-Plattform).

Bei der Datenbank konnten wir eine gute Idee unseres Dozenten und Diplomarbeitbetreuers, Herr Feisthammel, umsetzen. Unser Webshop wickelt alle Datenbankzugriffe zentral über einen Datenbank-Wrapper ab. Das heisst, der Shop-Betreiber kann wählen, welche Datenbank er benutzen will. Dies ist sicher von Vorteil, wenn in einem bestehenden Unternehmen schon eine Datenbank vorhanden ist. Der Shop ist momentan zu 100% auf MySQL einsetzbar und getestet. Die Programmierung der Sybase Unterstützung ist so gut wie abgeschlossen,

für alle weiteren Datenbanken muss man nur noch zwei Klassen neu schreiben.

MySQL war für uns erste Wahl, weil diese Datenbank für jedermann gratis zur Verfügung steht und bei den Internet Service Providern vielfach schon vorinstalliert ist. Ihr Leistungsumfang kommt zwar nicht an den grosser Datenbanken heran, wir haben uns aber mit dem DB-Wrapper diese Türen offen gehalten.

Abschliessend sei hier noch visuell dargestellt, mit welcher Technologie die Teile des PhPepperShops realisiert wurden. Die Datenbank wird über PHP via SQL Statements angesprochen. PHP generiert danach dynamisch HTML Code. Die HTML-Seiten werden über das Internet an den Browser übertragen. Clientseitig kann ein HTML Dokument per JavaScript dynamisch gestaltet werden. Was im folgenden Bild nicht erwähnt wird, ist das in Perl und Shell-Script geschriebene Installations- und Deinstallations-Script.



## 1.6 Zeitplanung

Nach der Auswertung der Feedbacks und Anregungen, die wir zwischen der ersten Projektarbeit und der Diplomarbeit erhalten hatten, stellten wir eine Liste mit Tätigkeiten und zu implementierenden Funktionalitäten auf. Wir teilten sie in Gruppen unterschiedlicher Dringlichkeit und Wichtigkeit ein. Ausserdem ordneten wir die Tätigkeiten verschiedenen Tätigkeitsgruppen zu. Des weiteren nahmen wir für jeden zu erledigenden Punkt eine grobe Aufwandsschätzung vor. Aufgrund dieser ermittelten Daten wollten wir eine erste detaillierte Zeitplanung erstellen. Die Zeitaufwandssumme der dringend zu erledigenden Arbeiten überstieg jedoch schon fast den gesamten Zeitraum, den wir zur Programmierung zur Verfügung hatten.

Da es unser allerhöchstes Ziel war, den Shop nach Ende der Diplomarbeit funktionsfähig der Öffentlichkeit zu übergeben und wir keine Abstriche in der Funktionalität machen konnten, erstellten wir eine überaus ehrgeizige Zeitplanung. Wir entschieden uns, absolut strikt und ohne Ausnahme nach der Prioritätenliste vorzugehen und Versuche sowie Exkurse weitgehendst zu unterlassen. Da wir unseren Shop auf dem CVS-Server von SourceForge entwickelten, hatten wir fast täglich wieder Commits (hochladen von aktualisierten oder neu erstellten Softwareteilen auf den CVS-Server) und Checkouts (herunterladen der aktuellen Version des Programms). Schnell merkten wir jedoch, dass die nahezu tägliche Neuinstallation des Shops viel Zeit auffrass. So entschlossen wir uns kurzfristig ein Installations- und Deinstallationstool zu erstellen, was rückblickend gesehen eine sehr lohnende Entscheidung war, da wir uns dadurch viel Installationsaufwand und Installationsfehler ersparen konnten.

Durch den Einsatz sämtlicher uns zur Verfügung stehender Kräfte, kamen wir dann bei den meisten Punkten schneller voran, als wir dies eigentlich ursprünglich eingeplant hatten. Wobei hier erwähnt werden muss, dass wir bei der ursprünglichen Planung von einer durchschnittlichen Arbeitsstundenanzahl von 8½ Stunden pro Tag und Person ausgingen, welche wir schlussendlich jedoch meistens um mehrere Stunden überschritten. Ein mühsamer Knackpunkt war das Layoutmanagement, da wir massiv mit der unterschiedlichen Interpretation von HTML, der von uns zu Testzwecken eingesetzten Browser, zu kämpfen hatten.

Gerne hätten wir mehr Zeit in die Zahlungsabwicklung mit Online-Payment-Instituten investiert. Nach einem Besuch dieser Firmen an der Orbit/Comdex, mussten wir jedoch feststellen, dass sich die Zielgruppe dieser Payment-Anbieter nicht mit der unseres Shops deckt. Wir planten dann ein wenig um und entschlossen uns, selbst eine sichere Zahlungsabwicklung zu implementieren.

Erschwerend zur ohnehin schon knapp kalkulierten Zeitplanung kam dazu, dass vor allem in der Start- und Endphase der Diplomarbeit das ZHW-LAN und die Verbindung zum Internet oft und über längere Zeit nicht verfügbar waren, was bei einem Webprojekt nicht gerade förderlich ist.

### 1.6.1 Aufgabenaufstellung nach Wichtigkeit, 17.09.2001

#### Legende:

<b>Priorität (P):</b>	1 = höchste 12 = kleinste	<b>Zeitaufwand (Z):</b>	1 = bis halber Tag 2 = bis zwei Tage 3 = mehr Zeit wird benötigt	<b>Farben:</b>	wichtigste Aufgaben wichtige Aufgaben nicht so wichtige Aufgaben nice to have
-----------------------	------------------------------	-------------------------	--	----------------	--

### ***Bekannte Fehler und Mängel beheben***

P	Z	Aufgabe
1	1	Provider Probleme --> 1 User, (ev. 1 Datenbank), detaillierte Installationsangaben
2	3	Gleicher Artikel mit verschiedenen Var. / Optionen können nicht kumuliert werden
3	3	Unterkategorien (z.B. 300 Kategorien und mehr Produkte)
4	3	Abändern der Artikel-Update Funktionen
5	3	Artikel in mehreren Kategorien eintragen können
6	3	Mehrwertsteuer und Verpackungskosten/Frachtkosten in die Berechnung miteinbeziehen
7	1	Scrollen im Top-Frame unterbinden
8	1	Charset im Mailversand ergänzen
9	1	Bilddatum für Cache auf 1.1.1971 korrigieren
10	1	Kostenlose Artikel --> z.B. Prospekte bestellen --> Zahlungsmöglichkeiten weglassen...
11	1	PLZ auch alphanumerisch (USA)
12	2	Shop muss mit root-Rechten laufen (mühsam, wegen Bild Upload)

### ***Admin-Modus fertig stellen***

1	3	Shop-Administrationsmodus fertig stellen (Zahlungsmöglichkeiten, Kategorienmanagement, Unterkategorien, u.s.w.)
2	3	Kategorienmanagement
3	1	Button hinzufügen, der mehr leere Options/Variantenfelder 'generiert'
4	2	Felderkonfiguration bei Kundenadresseingabe dynamisch wählbar gestalten (Admin) --> Reihenfolge wählbar
5	2	Aktionspreis implementieren
6	1	Formularbuttons durch grafische Buttons ersetzen (überall!)

### ***Implementieren wichtiger Zusatzfunktionalität***

1	3	Kreditkartenunterstützung --> ev. Wrapperklasse für mehrere Institute
2	1	Warenkorb ist leer Meldung implementieren
3	2	Backup des Shops (ev. auch automatisiert --> wenigstens Anleitung)
4	1	htaccess erklären
5	3	Anzahl Artikel pro Suchresultatseite festlegen (z.B. 25 Artikel pro Seite), Kunde kann wählen: 10 ... 50 ... alle
6	2	Suchanfrage mit mehreren Begriffen (wie bei <a href="http://www.google.com">http://www.google.com</a> )
7	1	Administrator soll entscheiden können ob Warenkorb direkt angezeigt werden soll
8	1	Anleitung für GIF/PNG Bilder Benutzung (für Artikelfotos)
9	1	Benutzerinfo, dass ohne E-Mail Adresse keine Bestellbestätigung ausgestellt wird
10	3	SSL --> GNU GP <- PGP-Mail (Kreditkarten Nr., ... ) Mailorder-Vertrag / allgemein SSL anschauen --> wo einsetzen
11	2	Suchresultate optional mit Bilder
12	3	Lagerverwaltung: Bestand eingeben, bei Verkauf dekrementieren --> Mindestmengen Alarm

### ***Layout Management***

1	3	CSS-Einstellungen per Web machen
2	2	Shop-Logo upload ermöglichen --> Als (GIF-)Bilddatei im Bilder-Verzeichnis ablegen
3	2	Eigene Buttons (Buttonsets) ev. mit Java-Script (onMouseOver, onClick,...)
4	3	Dynamisches Page Layout mit einer Auswahl Framesets (auch: Header und Footer dynamisch)

### ***CRM***

1	3	CRM 1: Eigenes Login, automatisch ausgefüllte Adresse, Login zwingend / nicht zwingend --> Händlermodus
---	---	---



### 1.6.2 Zeitplanung, erstellt zu Projektbeginn

■ höchste Dringlichkeit      ■ mittlere Dringlichkeit  
■ hohe Dringlichkeit      ■ nice to have

	36	KW 37 10.9.01-16.9.01					KW 38 17.9.01-23.9.01					KW 39 24.9.01-30.9.01					KW 40 1.10.01-7.10.01					KW 41 8.10.01-14.10.01					KW 42 15.10.01-21.10.01					KW 43 22.10.01-28.10.01				
	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M			
<b>Vorbereitungs- &amp; fest eingeplante Arbeiten</b>																																				
Inbetriebnahme Infrastruktur	■	■																																		
Projektsitzung				■																																
Auswertung Feedback, E-Mails		■																																		
Update Werkzeuge, Programme auf Server		■	■	■	■																															
SourceForge: Projekt und User erstellt				■																																
phpeppershop.com reg. und aufgeschaltet				■																																
minimales Webdesign phpeppershop.com					■	■																														
Orbit (Vorlesung Adori-Shop)																																				
Testen																																				
Erstellen Dokumentation / Online-Doku																																				
Abgabe DA																																				
<b>Bekannte Fehler und Mängel</b>																																				
Provider Probleme --> 1 User																																				
Gleicher Artikel mit versch. Var. / Optionen																																				
Unterkategorien																																				
Abändern der Artikel-Update Funktion(-en)																																				
Artikel in mehreren Kategorien eintragen können																																				
Mehrwertsteuer und Verpackungs-/Frachtkosten																																				
Scrolling im Top-Frame unterbinden																																				
Charset im Mailversand ergänzen																																				
Bilddatum für Cache auf 1.1.1971 korrigieren																																				
Gratisartikel --> Zahlungsmög. weglassen																																				
PLZ auch alphanumerisch (USA)																																				
<b>Admin Modus fertigstellen</b>																																				
Shop-Admin Modus fertigstellen																																				
Kategorienmanagement																																				
Button, mehr leere Options/Variationsfelder																																				
Felderkonfiguration bei Kundenadresseingabe																																				
<b>Implementieren wichtiger Zusatzfunktionalität</b>																																				
Kreditkartenunterstützung (yellowworld)																																				
Warenkorb ist leer Meldung implementieren																																				
Backup des Shops (Anleitung)																																				
htaccess erklären																																				
Anzahl Artikel pro Suchresultatseite festlegen																																				
Suchanfrage mit mehreren Begriffen																																				
<b>Layout Management</b>																																				
CSS-Einstellungen per Web machen																																				
Shop-Logo upload ermöglichen																																				
Buttons ev. mit Java-Script (onmouseover,...)																																				
<b>CRM</b>																																				
CRM 1:Eigenes Login, autom. ausgef. Adresse																																				
<b>nicht so wichtige Aufgaben</b>																																				
Shop muss mit root-Rechten laufen																																				
Aktionspreis implementieren																																				
Warenkorb direkt anzeigen (Admin entscheidet)																																				
Anleitung für GIF / PNG Bilder																																				
Benutzerinfo, dass ohne E-Mail Adresse keine..																																				
<b>nice to have</b>																																				
Formularbuttons durch grafische Buttons ersetz.																																				
SSL --> GNU GP <-- PGP-Mail																																				
Suchresultate optional mit Bilder																																				
Lagerverwaltung: Bestand eingeben																																				
Dynamisches Page Layout (Framesets)																																				

### 1.6.3 Effektiver Ablauf des Projekts

Wie in der folgenden Tabelle ersichtlich ist, haben wir fast alle geplanten Funktionen implementieren können. Die jetzt noch rot unterlegten Arbeiten stellten sich während der Implementierung des Shop für dessen Funktionalität und Bedienbarkeit als nicht elementar heraus, so dass wir während der Diplomarbeit deren Priorität heruntersetzten.

Einige niederprioritäre Funktionen konnten praktisch im gleichen Arbeitsschritt mit einer Funktion, der wir eine höhere Priorität zugeteilt hatten, gelöst werden. Dies erklärt, warum wir nicht konsequent eine Prioritätsstufe nach der anderen abarbeiteten.

- Arbeiten die erfolgreich gelöst werden konnten
- Arbeiten, zu dessen Lösung keine Zeit mehr blieb oder die uns als unwichtig erschienen
- Die Farben in der Wochenübersicht wurden aus der Zeitplanung vom 18.09.2001 übernommen und entsprechen den Prioritäten der Arbeiten

	36	KW 37 10.9.01-16.9.01				KW 38 17.9.01-23.9.01				KW 39 24.9.01-30.9.01				KW 40 1.10.01-7.10.01				KW 41 8.10.01-14.10.01				KW 42 15.10.01-21.10.01				KW 43 22.10.01-28.10.01			
	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	
<b>Vorbereitungs- &amp; fest eingeplante Arbeiten</b>																													
Inbetriebnahme Infrastruktur	■	■																											
Projektsitzung				■				■			■																		
Auswertung Feedback, E-Mails		■																											
Update Werkzeuge, Programme auf Serv.			■	■	■	■	■																						
SourceForge: Projekt und User erstellt			■	■	■	■																							
phpeppershop.com registriert und aufgeschaltet			■																										
minimales Webdesign phpeppershop.com					■	■	■	■	■																				
Erstellung Installations-/Deinstallationscript						■	■	■	■																				
Orbit (Vorlesung Adori-Shop)												■																	
Testen																													
Erstellen Dokumentation / Online-Doku							■		■	■																			
Organisation Raum für DA-Präsentationen																													
Abgabe DA																												■	
<b>Bekannte Fehler und Mängel</b>																													
Provider Probleme --> 1 User							■																						
Gleicher Artikel mit versch. Var. / Optionen													■																
Unterkategorien																													
Abändern der Artikel-Update Funktion(-en)																													
Artikel in mehreren Kategorien eintragen können																													
Mehrwertsteuer und Verpackungs-/Frachtkosten																													
Scrolling im Top-Frame unterbinden																													
Charset im Mailversand ergänzen																													
Bilddatum für Cache auf 1.1.1971 korrigieren																													
Gratisartikel --> Zahlungsmög. weglassen																													
PLZ auch alphanumerisch (USA)																													
<b>Admin Modus fertigstellen</b>																													
Shop-Admin Modus fertigstellen																													
Kategorienmanagement																													
Button, mehr leere Options/Variationsfelder																													
Felderkonfiguration bei Kundenadresseingabe																													
<b>Implementieren wichtiger Zusatzfunktionalität</b>																													
Kreditkartenunterstützung (yellowworld)																													
Warenkorb ist leer Meldung implementieren																													
Backup des Shops (Anleitung)																													
htaccess erklären																													
Anzahl Artikel pro Suchresultatseite festlegen																													
Suchanfrage mit mehreren Begriffen																													
<b>Layout Management</b>																													
CSS-Einstellungen per Web machen																													
Shop-Logo upload ermöglichen																													
Buttonsets ev. mit Java-Script (onmouseover,...)																													
<b>CRM</b>																													
CRM 1.Eigenes Login, autom. ausgef. Adresse																													
<b>nicht so wichtige Aufgaben</b>																													
Shop muss mit root-Rechten laufen																													
Aktionspreis implementieren																													
Warenkorb direkt anzeigen (Admin entscheidet)																													
Anleitung für GIF / PNG Bilder																													
Benutzerinfo, dass ohne E-Mail Adresse keine...																													
<b>nice to have</b>																													
Formularbuttons durch grafische Buttons ersetzt.																													
SSL --> GNU GP <-- PGP-Mail																													
Suchresultate optional mit Bilder																													
Lagerverwaltung: Bestand eingeben																													
Dynamisches Page Layout (Framesets)																													

## 1.7 Tests

Schon als wir mit Programmieren begannen, merkten wir, dass ein kompletter Test aller Funktionen und Zustände schlussendlich nicht mit einem vertretbaren Zeitaufwand möglich sein würde. Wir beschlossen einige Massnahmen, damit der Shop in seiner Schlussversion möglichst wenige, dazu leicht lokalisierbare Fehler enthielte.

1. Funktionen sollen, falls ein Problem auftritt, dieses direkt im Klartext detailliert beschrieben an den Browser ausgeben.
2. Funktionen, die den Rückgabewert eigentlich nicht brauchen, geben bei Erfolg „true“ zurück. Tritt irgend ein Problem auf, wird ein „false“ zurückgegeben. Der Fehler kann also recht genau lokalisiert werden.
3. Funktionen sollen direkt nach der Erstellung mit einigen sinnvollen Extremwerten getestet werden.
4. Schnittstellen zwischen Funktionen müssen genau definiert werden.
5. Jede Zeile, deren Bedeutung auf den ersten Blick nicht gleich erkennbar ist, soll kommentiert werden.

Wir wendeten dieses Konzept konsequent an. Es entstanden jedoch einige Funktionen, die aufgrund ihrer Grösse und Komplexität nicht mehr komplett ausgetestet werden konnten. Durch die oben beschriebenen Massnahmen war (und ist) es jedoch möglich, einen Fehler sehr genau zu lokalisieren.

### 1.7.1 Usability-Test

Da wir nicht von allen Anwendern das technische und logische Verständnis erwarten können, wie wir es mittlerweile haben, entschlossen wir uns, einen umfangreichen Usability-Test durchzuführen.

Da wir natürlich nicht selbst die Usability unseres Shops testen konnten, machten wir uns auf die Suche nach Testpersonen, die sich bereit erklärten, den Shop auf Bedienbarkeit hin zu prüfen und fanden diese dann schliesslich auch.

#### **Andrea Harder, 9500 Wil:**

- Testen des kompletten Administrationsbereichs auf Bedienbarkeit, Korrektheit und Verständlichkeit.
- Einkaufen als normaler Shopuser

#### **Jolanda Jud, 9500 Wil:**

- Betreiben eines bereits installierten und konfigurierten Shops (Artikel und Kategorienmanagement)
- Erstellen eines eigenen Shoplayouts
- Einkaufen als normaler Shopuser

#### **Roland Brühwiler, 8560 Märstetten:**

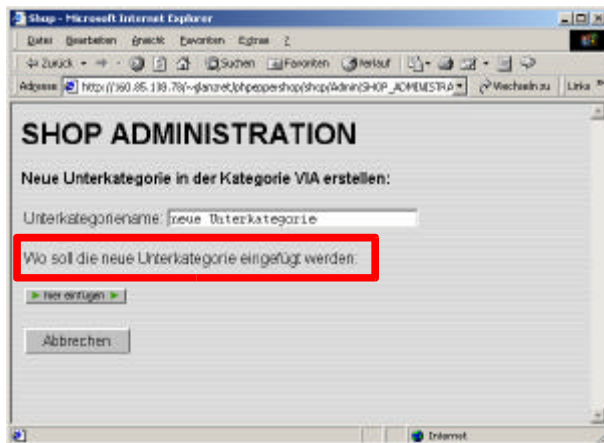
- Installation des Shops
- Verständlichkeit der technischen Einstellmöglichkeiten
- Einkaufen als normaler Shopuser

Was uns erfreute war, dass von unseren Testpersonen keine grösseren Fehler gefunden wur-

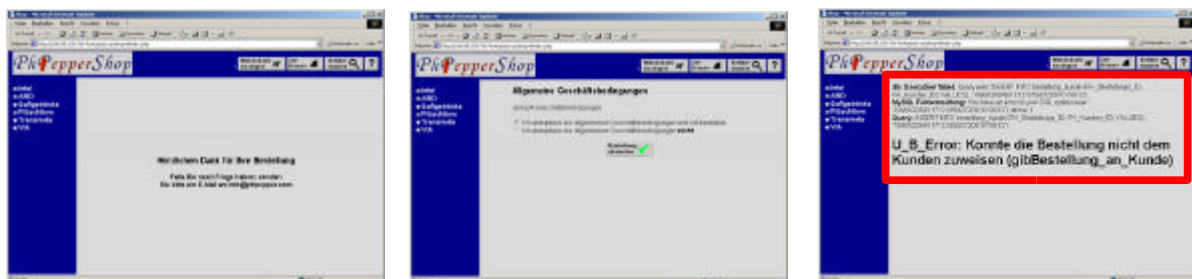
den. Einige kleine Fehler und Probleme konnten von uns schnell lokalisiert und behoben werden.

Betreffend Bedienbarkeit bemängelten alle Testpersonen die gleichen drei Punkte:

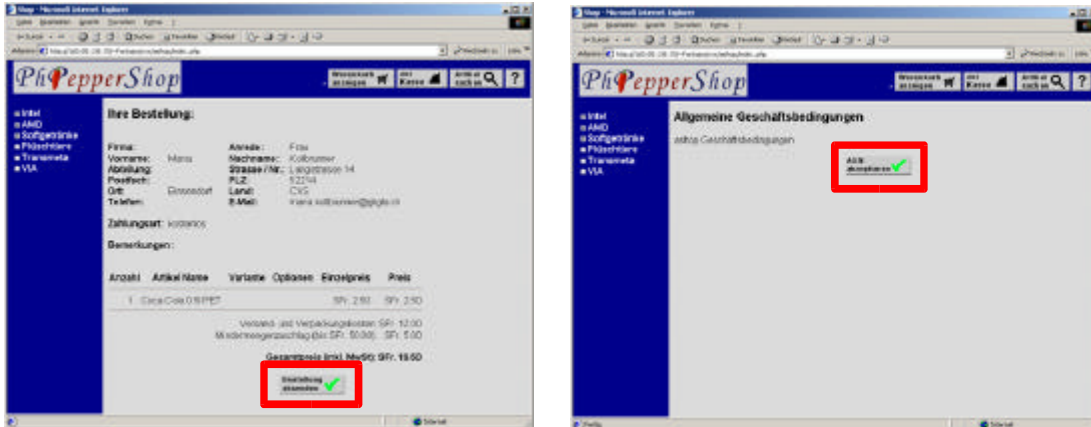
1. Wird eine Unterkategorie in einer Kategorie erstellt, in der noch keine Unterkategorien enthalten sind, so wird dem Administrator mitgeteilt, er könne wählen, wo die Unterkategorie eingefügt werden soll, obwohl eigentlich gar keine Auswahlmöglichkeit besteht:



2. Drückt man nach dem Absenden der Bestellung den Zurück-Button des Browsers, und dann nochmals auf „Bestellung absenden“, wird erneut ein Bestätigungs-E-Mail versendet und eine Fehlermeldung wird an den Browser ausgegeben.



3. Es ist irritierend, dass nach dem Drücken des „Bestellung absenden“-Knopfs noch die Allgemeinen Geschäftsbedingungen angezeigt und akzeptiert werden müssen. Wenn man den Knopf „Bestellung absenden“ drückt, sollte man davon ausgehen können, dass die Bestellung unmittelbar danach auch abgesendet wird.



Leider hatten wir nicht mehr die Zeit, alle drei Mängel zu beheben. Da wesentlich mehr Leute mit dem Shop selbst, als mit dem Administrationsbereich in Berührung kommen werden, legten wir dir Priorität ganz klar auf den Shop selbst.

Wir behoben die beiden letzteren Mängel und entschlossen uns, als Hinweis in die Bedienungsanleitung für den Administrator zu schreiben, dass das Kategorienmanagement vielleicht etwas verwirrend sein kann, wenn man eine Unterkategorie in eine Kategorie einfügt, die keine weiteren Unterkategorien enthält.

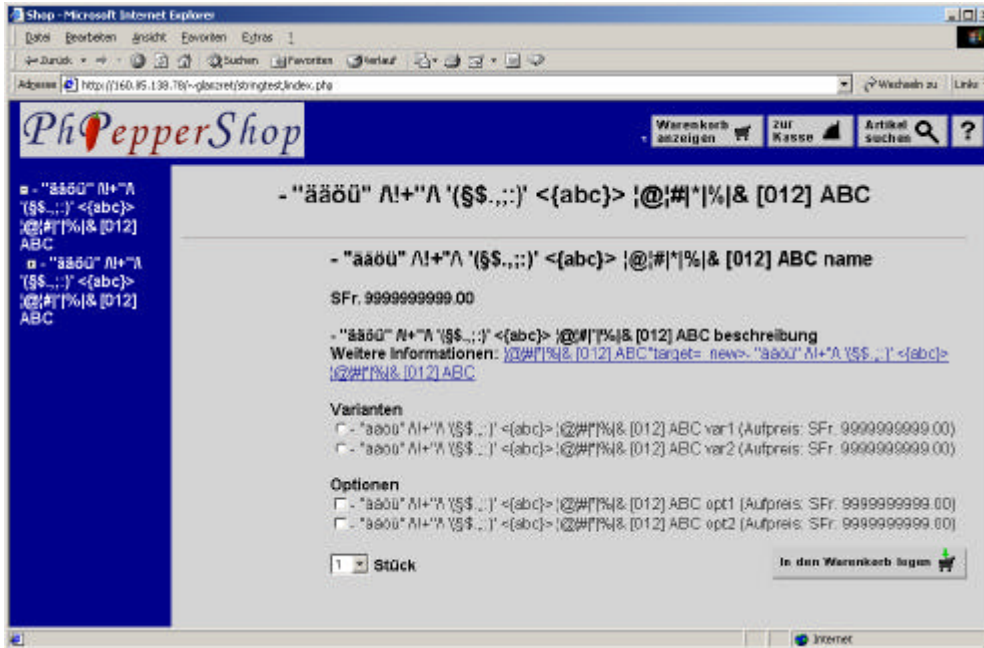
### 1.7.2 Extremwerte Test

Wir stellten bei vielen anderen Webshops fest, dass sich komische Effekte ergaben, wenn man in Formularfeldern gewisse Sonderzeichen einfügte. Dies wollten wir unbedingt verhindern.

Wir erstellten einen ziemlich gemeinen Teststring, der so ziemlich alle erdenklichen Sonderzeichen und Sonderzeichen-Kombinationen enthält:

- "ääöü" /\!+\"/\ ' (§\$. , ; :)' <{abc}> |@|#|\*|%|& [012] ABC

Wir verwendeten darauf ziemlich viel Zeit, um den Shop komplett Sonderzeichen-fähig zu machen. Schlussendlich freuten wir uns, als wir folgendes Bild zu sehen bekamen:



Dies war der Beweis, dass alle Strings korrekt erfasst, verarbeitet und angezeigt wurden. Wir verwendeten den gleichen Test-String dann auch noch zum Sonderzeichentest des CRM- und Bestellungen-Teils. Auch diese Tests fielen positiv aus.

### 1.7.3 Funktionstest

Bei den Funktionstests konzentrierten wir uns hauptsächlich auf die neu hinzugekommenen Funktionen des Shops; das Kundenmanagement, die Eingabeüberprüfung per JavaScript, Unterkategorien, Zahlungsablauf und das Layout-Management. Da die komplette Auflistung aller Tests den Rahmen dieses Dokuments sprengen würde, werden hier exemplarisch die wichtigsten Testfälle des CRM's und des Zahlungsablaufs, abgedruckt. Da es sich in den letzten beiden Projektarbeiten bewährt hatte, die Testfälle zur besseren Übersichtlichkeit zu visualisieren, verfolgten wir dieses Konzept auch hier weiter.




Der Shoptest wurde mit dem Internet Explorer 5.5 durchgeführt (Windows 2000). Vor Aufruf des Shops leerten wir den Browsercache.

Für die Richtigkeit der Testergebnisse,

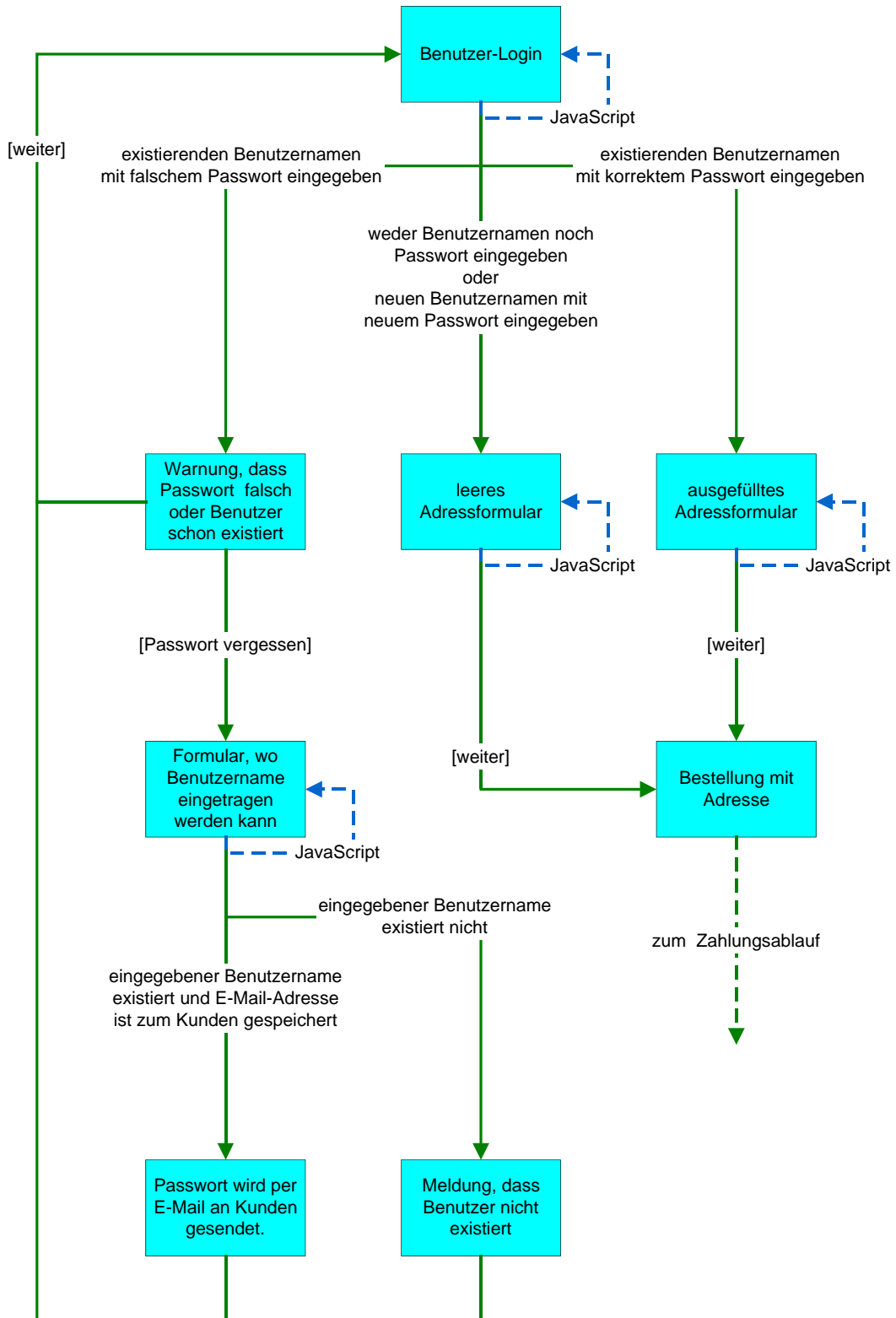
*R. Glanzmann*

R. Glanzmann

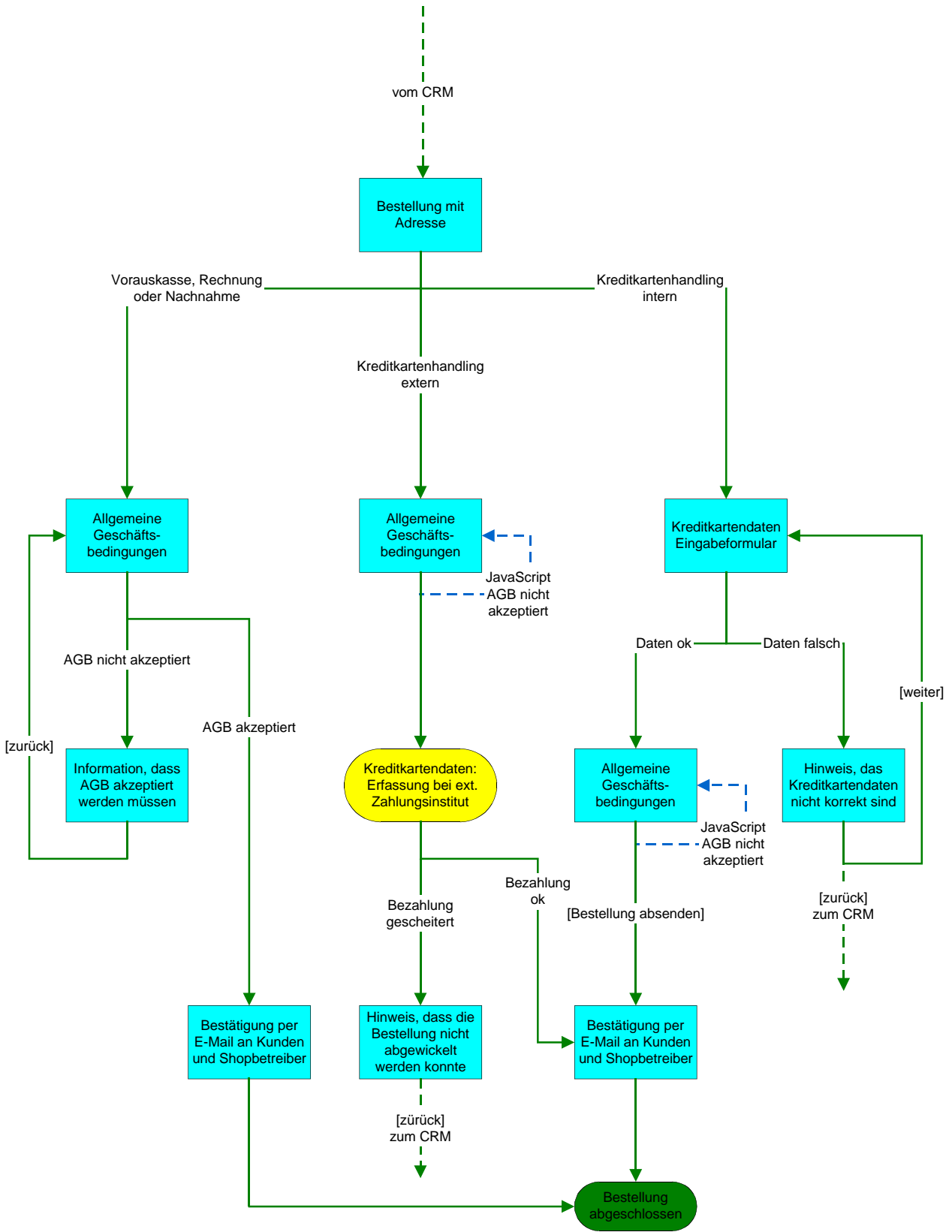
Winterthur 26.10.2001

Testlegende:	
	korrekte Ausgabe an Browser oder E-Mail
	erfolgreich getesteter Funktionsablauf
	erfolgreich getestete JavaScript Eingabeüberprüfung
[text]	Button oder Link gedrückt

### Test des CRM's (Customer Relationship Management)



### Test des Bestellungsablaufs





## 1.8 Erfahrungen

### 1.8.1 Installation der Systemumgebung - LAMPS

#### *Inhalt dieses Kapitels*

Da wir mit der aktuellen Version von PHP (v.4.0.6) arbeiten wollten, haben wir uns entschieden, den Apache Server, PHP und SSL-Unterstützung neu zu installieren. Damit unsere Erfahrungen, die wir hier machten, auch an andere Leute weitergegeben werden können, haben wir diesen Erfahrungsbericht im Stil einer umfassenden Anleitung geschrieben.

#### *Wieso diese Kombination?*

Alle fünf Teile eines LAMPS-Systems: Linux, Apache, MySQL, PHP, (open-, mod\_)SSL sind Projekte, welche gemeinsam haben, dass sie Open Source sind. Sie kosten nichts, sind erprobt und sehr leistungsfähig. Alle Projekte werden aktiv weiterentwickelt. Mit Linux als kostenlosem Betriebssystem, erhält man heute über eine Distribution (SuSE, Redhat,...) ein zu kommerziellen Serverbetriebssystemen konkurrenzfähiges Produkt.

Apache ist ohnehin der am weitesten verbreitete Webserver der Welt. Er lässt hinsichtlich Stabilität und Erweiterbarkeit keinerlei Wünsche offen. Insbesondere die Kombination von Apache mit PHP ist heutzutage sehr populär und auch schon recht häufig anzutreffen. ...Ausserdem ist er nicht anfällig auf Code Red und Nimda...

MySQL ist eine kleine, aber im Web-Umfeld weit verbreitete Datenbank. Sie bietet zwar keine referentielle Integrität und auch sonst kann sie nicht unbedingt mit einer IBM DB2 oder einer Oracle 9i konkurrieren. Ihr grosser Vorteil ist aber die grosse Unterstützung in der Web Open Source Gemeinde. Es gibt unzählige Projekte, welche MySQL als Datenbank nutzen. Dies liegt nicht zuletzt auch daran, dass MySQL sehr einfach zu handhaben ist. Hier sei in diesem Zusammenhang noch das Projekt phpMyAdmin (<http://www.phpwizard.net/phpMyAdmin/>) erwähnt, welches auf PHP basierend eine sehr komfortable Oberfläche zur Datenbankadministration bietet.

PHP ist eine leistungsfähige Skriptsprache, primär dazu ausgelegt im Web-Umfeld verwendet zu werden. PHP geniesst in seiner vierten Inkarnation eine breite Unterstützung, einerseits durch Webserver-Einbindungen, andererseits aber auch durch zahlreiche Funktionen, die inzwischen in PHP implementiert wurden. PHP ist einfach zu erlernen, benutzt die vielfach verwendete ANSI-C Syntax und unterstützt von Haus aus viele Datenbanken. Die für das Webscripting unumgänglichen String-Operationen hat es zum grössten Teil von Perl übernommen. Die Einbindung des PHP Codes in HTML erfolgt analog zu Microsofts ASP. Auf diese Weise wird von allen bekannten Sprachen und Techniken 'das Beste' verwendet und in PHP vereint.

Wenn man mit Geld hantieren will, kommt es auf Sicherheit an, und die Anforderungen an sie sollte man nicht unterschätzen. Die Datenübertragung von Kreditkartennummern, und Benutzerdaten und ihren Einkäufen dürfen für Dritte nicht einsehbar sein. Das heisst, dass man primär die Datenübertragung über das 'unsichere' Internet schützen, sprich verschlüsseln, muss. Eine heute anerkannte Technologie um dies zu realisieren ist SSL (Secure Socket Layer). OpenSSL ist eine Open Source Implementation der Verschlüsselungsfunktionalität, welche vom mod\_SSL (Apache DSO-Modul) benutzt wird.

## Installation

### Quellen beschaffen

Die beschriebenen Quellen beziehen sich auf Pakete, welche im September 2001 aktuell waren. Natürlich werden die Pakete zu einem späteren Zeitpunkt andere Versionsnummern tragen.

### Linux

Linux beschafft man sich am besten als Komplettpaket in Form einer Distribution. Wir arbeiteten mit *SuSE Linux Professional 7.1*. Für welche man sich entscheidet, ist grundsätzlich Geschmackssache und im speziellen für diese Anwendung eigentlich nicht von Bedeutung. Wir benutzen hier weder das X-System, noch distributionsspezifische Eigenheiten. Vorausgesetzt wird lediglich ein lauffähiges Linux-System (natürlich mit gcc, flex, yacc, u.s.w.).

Wir empfehlen die folgenden Pakete gleich ins Verzeichnis /usr/local/src/ zu speichern, dann können wir vom gleichen Pfad ausgehen. Ausserdem ist so durch die Verzeichnisstruktur festgelegt, dass es sich um Sourcepakete handelt.

### Apache

<http://httpd.apache.org/dist/httpd/>

### MySQL

<http://www.mysql.com/Downloads/MySQL-3.23/mysql-3.23.42.tar.gz>

Bei MySQL entstehen ungefähr alle zwei Wochen neue Versionen. Deshalb hier noch die allgemeine Download Adresse (Achtung: Source-Files herunterladen, nicht Binaries):

<http://www.mysql.com/downloads/>

### PHP

<http://www.php.net/downloads.php>

### SSL

openssl: <http://www.openssl.org/source/>

mod\_ssl: <http://www.modssl.org/source/>

### Erweiterungen

GD-Library: <http://www.boutell.com/gd/>

Zlib-Library: <http://www.gzip.org/zlib/>

IJG JPEG Software: <http://www.iijg.org/>

PDFlib: <http://www.pdflib.com/pdflib/download/index.html>

Freetype: [http://sourceforge.net/project/showfiles.php?group\\_id=3157](http://sourceforge.net/project/showfiles.php?group_id=3157)

## Eigentliche Installation

### Entpacken der Pakete

Die Pakete werden wie folgt entpackt:

```
tar xvfz paketename.tar.gz
```

Nun gibt es jeweils Verzeichnisse, in denen sich die entpackten Tarballs (.tar.gz Dateien) befinden.

## MySQL

Wir installieren zuerst MySQL, da die Datenbank unabhängig von den anderen Paketen ist.

Das Verzeichnis, in welchem MySQL seine Datenbanken (data) ablegt, kann mit dem Configure-Parameter `localstatedir` angegeben werden. Wir haben uns hier für den Pfad `/usr/local/mysql/data` entschieden. Mit dem Parameter `prefix` geben wir das Installationsverzeichnis unserer MySQL Datenbank an. (z.B. `/usr/local/mysql/3.23.42/` für unsere aktuelle MySQL Version). Nach der Installation erstellen wir noch einen Softlink. Auf diese Weise findet man nach der Installation einer neueren Version, die aktuelle unter `current`.

Im MySQL Source Verzeichnis:

```
./configure --prefix=/usr/local/mysql/3.23.42/ --localstatedir=/usr/local/mysql/data
make
make install
ln -s /usr/local/mysql/3.23.42 /usr/local/mysql/current
mkdir /usr/local/mysql
mkdir /usr/local/mysql/data
/usr/local/mysql/current/bin/safe_mysqld -u mysqladmin &
```

## SSL Installation

Damit wir mit SSL arbeiten können, müssen wir unser System zuerst mit der SSL Funktionalität ausrüsten. Deshalb installieren wir zuerst openSSL:

Im openSSL Source Verzeichnis:

```
./config--prefix=/usr/local/openssl/0.9.6b
make
make test
make install
ln -s /usr/local/openssl/0.9.6b /usr/local/openssl/current
```

## Apache und mod\_SSL

Wir erstellen nun eine lauffähige Version von Apache, welche mit Dynamic Shared Objects umgehen kann. Dies ist eine überaus praktische Technologie. DSOs werden auch Apache Module genannt und funktionieren in etwa analog zu den bekannten Linux Modulen. Man kann sie einbinden und muss nicht für jedes neue Modul den ganzen Apache neu übersetzen.

Wir patchen die Quellen des Apache Webservers mit dem `mod_SSL` Paket. Gleichzeitig wird die Apache Configure Funktion aufgerufen. Das `datadir` Verzeichnis beherbergt das Webserver Hypertext Document Root. Wir haben uns hier für `/usr/local/apache/htdocs` entschieden.

Im `mod_SSL` Source Verzeichnis (*nicht* im Apache Source Verzeichnis!):

```
./configure --with-apache=../apache_source_verzeichnis \  
--with-ssl=../openssl_source_verzeichnis \  
--prefix=/usr/local/apache/1.3.20 \  
--datadir=/usr/local/apache/htdocs \  
--enable-module-most \  
--enable-shared-max \  
--enable-module-ssl
```

Im Apache Source Verzeichnis:

```
make
```

### Server-Zertifikat erstellen

Wir erzeugen nun ein Zertifikat zur späteren Verwendung von SSL-Sitzungen. Dieses wird von einer Certificate Authority (CA) unterschrieben. Wir legen die CA hier auch gleich mit an.

CA erstellen:

Im Apache Verzeichnis:

```
make certificate TYPE=custom
```

```
Signature Algorithm:      R  
Country Name:            CH  
State or Providence:    ZH  
Locality Name:          Winterthur  
Organisation Name:      PhPepperShop  
Organisational Unit Name: CA  
Common Name:            PhPepperShop CA  
Email Address:          ca@mydomain.com  
Certificate Validity:    365  
  
Certificate Version:     3
```

Nachdem nun der CA-Schlüssel erzeugt wurde, geht es ans Server-Zertifikat. Dabei ist zu beachten, dass der *Common Name* den Rechnernamen darstellt, unter welchem der Server später per SSL erreicht werden kann! (z.B. <https://myshop.mydomain.com>)

```
Country Name:            CH  
State or Providence:    ZH  
Locality Name:          Winterthur  
Organisation Name:      PhPepperShop  
Organisational Unit Name: Webmaster  
Common Name:            myshop.mydomain.com  
Email Address:          info@mydomain.com  
Certificate Validity:    365  
  
Certificate Version:     3
```

Die Schlüssel sollen natürlich *verschlüsselt* werden, dies also bestätigen (zuerst CA, dann Server). Angemerkt sei hier noch, dass bei jedem Neustart des Apache das Passwort für den Server-Key eingegeben werden muss.

## Apache Installation vervollständigen

Im Apache Source Verzeichnis:

```
make install
ln -s /usr/local/apache/1.3.20 /usr/local/apache/current
```

Damit der Apache auch mit SSL-Unterstützung startet, müssen wir noch einen kleinen Eintrag in der Datei: /usr/local/apache/current/bin/apachectl ändern:

```
start)
if [ $RUNNING -eq 1 ]; then
    echo "$0 $ARG: httpd (pid $PID) already running"
    continue
fi
#Original:
#if $HTTPD ; then
#gaendert, -DSSL eingefuegt:
if $HTTPD -DSSL; then
    echo "$0 $ARG: httpd started"
else
    echo "$0 $ARG: httpd could not be started"
    ERROR=3
fi
;;
```

Nun müssen noch alle SSL Einträge aus der httpd.conf.default in die Datei httpd.conf kopiert werden. Es geht v.a. darum, die Module libssl und modssl zu laden (LoadModule).

## PHP

PHP ist ein komplexes System und damit es wie von uns erwartet funktioniert, müssen wir zuerst noch ein paar Hilfslibraries kompilieren und installieren:

### Freetype

Im Freetype Source Verzeichnis:

```
./configure --prefix=/usr/local
make
make install
```

### PDFlib

Im PDFlib Source Verzeichnis:

```
./configure --prefix=/usr/local
make
make install
```

### Zlib

Im Zlib Source Verzeichnis:

```
./configure
make
make install
```

## IJG JPEG Software

Im IJG JPEG Software Source Verzeichnis:

```
./configure
--enable-shared \
--enable-static \
--prefix=/usr/local
make
make test
make install
```

## GD

Bevor wir die GD Library kompilieren können, müssen wir noch das Makefile editieren und abändern. Die Defaulteinstellungen benutzen nicht die libjpeg, libpng, freetype Libraries. Das wollen wir aber und müssen deshalb die vorgegebenen Zeilen kommentieren und die Alternativen auskommentieren (CFLAGS und LIBS).

Bei den INCLUDEDIRS und den LIBDIRS sollte man jeweils einen Zusatz in der Form: `-I/usr/local/include/freetype2` beziehungsweise `-L/usr/local/lib` machen.

Nun sollte noch überprüft werden, ob auch folgende Pfade vorkommen:

```
INSTALL_LIB=/usr/local/lib
INSTALL_INCLUDE=/usr/local/include
INSTALL_BIN=/usr/local/bin
```

Und nun im GD Source Verzeichnis:

```
make
make install
```

## PHP konfigurieren und kompilieren

Wichtig ist das `with-config-file-path`-Attribut. Im angegebenen Pfad wird PHP später nämlich seine `php.ini` suchen.

Im PHP Source Verzeichnis:

```
./configure \  
--with-apxs=/usr/local/apache/current/bin/apxs \  
--with-mysql=/usr/local/mysql/current \  
--with-zlib \  
--with-ftp \  
--with-gd \  
--with-jpeg-dir=/usr/local/lib \  
--with-pdflib \  
--enable-versioning \  
--enable-track-vars=yes \  
--enable-url-includes \  
--enable-sysshm=yes \  
--enable-sysvsem=yes \  
--with-config-file-path=/etc  
make  
make install
```

Nachdem nun PHP installiert wurde, müssen wir nun noch eine Datei ins etc-Verzeichnis kopieren:

```
cp ./php.ini-dist /etc/php.ini
```

Es empfiehlt sich, nun in der `php.ini` den *SafeMode* einzuschalten. Dies macht man, indem man einen Editor (z.B. `pico`) nimmt und die Zeile sucht, in welcher `SafeMode=Off` steht und diesen dann auf `SafeMode=On` stellt.

## Konfiguration

Wir müssen im Verzeichnis `/usr/local/apache/current/conf/` die zentrale Konfigurationsdatei des Apache (`httpd.conf`) editieren. Zuerst sollten alle aktuellen Versionsnummern die im Skript vorkommen (bei uns z.B. 1.3.20) durch `current` ersetzt werden. Nun müssen wir noch die Mime-Types für PHP hinzufügen, beziehungsweise auskommentieren. Dies macht man am besten indem man folgende Zeile sucht, auskommentiert und `.phtml` noch hinzufügt:

```
AddType application/x-httpd-php    .php    .phtml
```

Zusätzlich konfigurieren wir Apache nun so, dass er auch PHP-Dateien als Default-Index Dateien akzeptiert. Dazu suchen wir den Eintrag `DirectoryIndex` und fügen noch die PHP Datei-Endungen hinzu:

```
DirectoryIndex    index.html    index.php    index.htm
```

## Starten

Wir können nun den Apache Webserver starten, indem wir folgende Anweisung eingeben:

```
/usr/local/apache/current/bin/apachectl start
```

Wenn SSL korrekt installiert wurde, wird man an dieser Stelle angehalten seine Passphrase einzugeben. Danach startet der Apache Webserver.

Mit einem Webbrowser (zur Not auch per `lynx`), kann man nun schauen, ob der Webserver auch läuft.

## 1.8.2 Einrichten eines Passwort geschützten Bereichs mit htaccess

Diese Anleitung soll lediglich das Prinzip, wie man ein Verzeichnis mittels htaccess schützt, erklären. Sie kann nicht als detaillierte Installationsanleitung verwendet werden, da sich die notwendigen Schritte je nach eingesetztem Webserver und Hostingprovider unterscheiden. Alle Pfadangaben beziehen sich explizit auf unseren Testserver und müssen durch eigene ersetzt werden.

Das Prinzip des Verzeichnisschutzes per htaccess besteht aus vier Schritten:

1. Im Webserver-Konfigurationsfile (typisch: `httpd.conf`) muss eingestellt werden, dass die Konfigurationsanweisungen in `.htaccess`-Dateien beachtet werden sollen. Dies geschieht mittels des `AllowOverride` Eintrags auf das gewünschte Verzeichnis des Benutzers:

```
<Directory "/home/fontajos/htdocs/">
AllowOverride All
</Directory>
```

**Directory** gibt das Verzeichnis an, in welchem die Konfigurationsdatei beachtet werden soll.

2. In dem Verzeichnis, welches geschützt werden soll, muss die Datei `.htaccess` mit folgendem Inhalt erstellt werden:

```
AuthUserFile /home/fontajos/.htpasswd
AuthName "Administrationsbereich"
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>
```

**AuthUserFile** definiert, wo sich das Passwortfile befindet, welches die Benutzernamen und Passwörter der für den Zugriff berechtigten Benutzer enthält.

**AuthName** enthält den Text, der im Login-Fenster ausgegeben wird.

Mit **AuthType** setzt man den gewünschten Authentifikationstyp fest, wobei entweder Basic oder Digest angegeben werden kann. Prinzipiell sollte man die Digest-Variante bevorzugen, da hier, im Gegensatz zur Basic-Authentifikation, das Passwort niemals im Klartext über die Verbindung geschickt wird und somit von einem potentiellen Angreifer nicht in Erfahrung gebracht werden kann. Obwohl die Digest-Authentifikation sicherer ist, wird sie bisher noch nicht von allen Browsern unterstützt!

3. Mit dem Programm `htpasswd` kann eine Passwortdatei erstellt werden. Um eine Passwortdatei anzulegen, muss zuerst ermittelt werden, wo sich `htpasswd` befindet (`which htpasswd` oder `locate htpasswd`). Danach kann mit folgendem Befehlsaufruf eine Passwortdatei, die einen Benutzer enthält, erstellt werden:

```
verzeichnis_von_htpasswd/htpasswd -c .htpasswd Benutzername
```

**htpasswd** ist der Dateiname, wie die Passwortdatei heissen soll.

Als **Benutzername** kann ein beliebiger Name eingegeben werden.

Mit folgendem Aufruf kann der Passwortdatei noch ein weiterer Benutzer zugefügt, oder das Passwort eines schon enthaltenen Benutzers geändert werden:



```
verzeichnis_von_htpasswd/htpasswd .htpasswd Benutzername
```

4. Schlussendlich muss der Webserver noch dazu bewogen werden, die Konfigurationsfiles neu einzulesen. Entweder man „killed“ die Prozesse des Webservers (`killall -HUP httpd`) oder startet den Webserver neu (`./apachectl restart`).

### 1.8.3 SourceForge einrichten

#### Einleitung

In den folgenden Kapiteln beschreiben wir chronologisch unsere Erfahrungen, wie wir unser Projekt auf SourceForge eingerichtet haben, welche Hürden wir nehmen mussten und auf was man achten muss, wenn man selber ein Projekt auf SourceForge einrichten will. Wir wollen diesen Erfahrungsbericht als in sich geschlossenes Dokument veröffentlichen, sodass auch andere Leute davon profitieren können.

SourceForge bietet Open Source Projekten eine kostenlose, sehr umfangreiche und international ausgerichtete Entwicklungsumgebung an. Geboten wird unter anderem ein CVS Repository, Mailinglisten, eine Homepage, Newsgroups und vieles mehr.

#### SourceForge einrichten

##### Eigener Account

Um ein Open Source Projekt überhaupt erst lancieren zu können, muss man sich zuerst unter <http://ww.sourceforge.net/> anmelden. Dies macht man am einfachsten mit einem Mausklick links oben auf 'Neuer Benutzer (SSL)'. Die folgenden einzugebenden Felder sind selbst erklärend. Wenn man dann sein Login hat und sich einloggt, findet man auf der linken oberen Ecke eine Option 'Neues Projekt'. Auf diese Weise kann man sein neues Projekt einrichten.

##### Ein neues Projekt lancieren

Nun muss man dem SourceForge Terms of Service Agreement zustimmen. Vor allem Punkt 6 und 7 sind interessant. Unter anderem wird dort beschrieben, dass SourceForge keinerlei Haftung für Datenverlust übernimmt. Es ist also trotz allem ratsam immer ein aktuelles Backup zuhause zu haben. Der Punkt 7 der TOS widmet sich der Account- respektive Project Termination. Hierbei sei nur erwähnt, dass man grundsätzlich nicht von SourceForge verbannt wird, wenn man sich halbwegs ordnungsgemäss verhält. Wir wollen hier nicht alle Paragraphen besprechen, aber Punkt 5 finden wir erwähnenswert: Man stimmt mit dem Akzeptieren dieses Vertrags zu, dass man mit den Ressourcen von SourceForge keinerlei kommerzielle Ziele verfolgt.

Nun muss man seinem Projekt zuerst einmal einen Namen geben und es auf Englisch beschreiben. Diese Beschreibung ist Grundlage für die Zulassung des Projekts durch ein Beurteilungsteam von SourceForge. Wenn es unter einer Open Source Lizenz steht und frei verfügbar ist, sollte das aber ohne Probleme klappen. Weiter muss noch eine Lizenz gewählt werden (GPL oder LGPL dürften wohl die am meisten verwendeten Lizenzen sein) und es muss eine kleine Beschreibung angegeben werden. Die Public Description kann nur 255 Zeichen lang sein und wird bei Suchresultaten zu Projekten mit angezeigt. Zu guter letzt muss man noch einen Unix-Namen des Projekts angeben (die beschriebenen Restriktionen beachten).

Jetzt braucht man ein bisschen Geduld. Unser Projekt wurde innerhalb von 12 Stunden akzeptiert und von SourceForge eingerichtet, das kann aber zuweilen auch bis zu 24 Std. dauern. Wenn man sich nun wieder einloggt, findet man unten rechts sein Projekt.

##### Das neue Projekt einrichten

Nach einem Klick auf das Projekt befindet man sich auf der Projekt Übersichtsseite. Von hier

aus komplettiert man zuerst einmal die Projekt-Konfiguration. Man hat eine reiche Auswahl an angebotenen Dienstleistungen, welche alle konfiguriert und eingerichtet werden wollen.

## Übersicht des Projekts

**Summary**

[Überblick](#) | [Admin](#) | [Homepage](#) | [Forums](#) | [Tracker](#) | [Bugs](#) | [Support](#) | [Patches](#) | [Mailinglisten](#) | [Aufgaben](#) | [Dokumentation](#) | [Neuigkeiten](#) | [CVS](#) | [Filesgroup](#)  
[short\\_forum](#) | [Foren](#)

Anfangen sollte man beim 'Admin'-Teil. Im Misc. 'Project Information'-Fenster sieht man in welchem Verzeichnis sich was befindet. Bei uns sah die Verzeichnisstruktur wie folgt aus:

Homepage Link: `phpeppershop.sourceforge.net`  
 Group shell (SSH) server: `phpeppershop.sourceforge.net`  
 Group directory on shell server: `/home/groups/p/ph/phpeppershop`  
 Project WWW directory: `/home/groups/p/ph/phpeppershop/htdocs`

Man hat also sicher schon mal eine Homepage für das Projekt (100MB Webspace, CGI und PHP-Scripts sind auch erlaubt), ein CVS Repository (dazu später mehr) und Zugriff auf einen Mailinglisten-Server. Natürlich darf auch eine Shell nicht fehlen. Mit Telnet kommt man aber nicht weit, SourceForge erlaubt klugerweise nur SSH (SSH1 und SSH2).

Wenn man sich das erste mal auf die Shell des SourceForge Servers einloggt, (`ssh projektname.sourceforge.net`) wird einem der Fingerprint des jeweiligen Servers angezeigt. Man sollte sich kurz die Zeit nehmen und diesen mit dem jeweiligen Fingerprint hier vergleichen. Das beste Sicherheitstool nützt nichts, wenn man es nicht richtig anwendet:

Host	Protocol	Key Fingerprint
shell.sourceforge.net	SSH1	33:50:33:91:b4:ec:7a:b6:fa:39:99:b0:ff:65:63:a7
shell.sourceforge.net	SSH2	4c:68:03:d4:5c:58:a6:1d:9d:17:13:24:14:48:ba:99
cvs.sourceforge.net	SSH1	ff:f1:42:ca:ac:ae:a7:c8:e8:52:45:65:a1:a0:0a:b6
cvs.sourceforge.net	SSH2	02:ab:7c:aa:49:ed:0b:a8:50:13:10:c2:3e:92:0f:42

Bevor man weiterfährt, sollte man sich gleich einmal einloggen und zwar auf dem 'Projekt Server': `projektname.sourceforge.net` und ganz wichtig auch auf dem CVS-Server `cvs.projektname.sourceforge.net`. Man wird das CVS-System *nicht* benutzen können, wenn man sich nicht vorher einmal per SSH darauf eingeloggt hat (erst da wird das benötigte Home-Verzeichnis erstellt).

## Trove Categorisation

Man kann sich jetzt der Trove Categorisation seines Projekts widmen (auf [Edit] klicken). Hier kann man sein Projekt beschreiben und kategorisieren, sodass Leute, die nach Projekten mit gewissen Eigenschaften suchen, auch fündig werden. Wenn ein Projekt nicht eindeutig zugeordnet werden kann, darf man mehrere Einteilungen benutzen. SourceForge erlaubt bis zu drei Einteilungen gleichzeitig.

## Public Information

Im Admin-Bereich klicken wir nun auf 'Edit Public Info'.

[Admin](#) | [User Permissions](#) | [Edit Public Info](#) | [Project History](#) | [VHOSTS](#) |  
[Edit/Release Files](#) | [Post Jobs](#) | [Edit Jobs](#) | [Edit Multimedia Data](#) | [Database Admin](#)

Hier sieht man die öffentlich zugänglichen Dienste für dieses Projekt und die Beschreibung dazu. Hier muss natürlich ein jeder selber wählen, was für sein Projekt alles benötigt wird. Mailinglisten, Foren und CVS wird wohl jeder wählen, Surveys wahrscheinlich eher weniger.

Es ist erstaunlich, was SourceForge einem hier alles anbietet, und zwar gratis und franko. Bleibt nur die Frage, ob das von VA Linux gesponserte System auch für immer bestehen bleibt (216 Mio. \$ Verlust bei 16 Mio. \$ Umsatz im letzten Quartal...) Unserer Meinung nach wäre SourceForge eine Einrichtung, die vom Staat getragen werden müsste.

### Entwickler einrichten

Als nächstes werden alle am Projekt beteiligten Mitarbeiter eingefügt und ihren Usern die entsprechenden Rechte geben:

1. Im Group Members Fenster den entsprechenden Usernamen eingeben und 'Add User' wählen.
2. In den 'User Permissions' (oben) kann man dem jeweiligen User Rechte und Ämter geben.

### Webseiten für das Projekt aufsetzen

Jedes Projekt braucht natürlich seine Homepage. Hier wollen wir nicht darauf eingehen wie man seine Page gestalten sollte, sondern nur, wohin man sie ablegen muss und einige Punkte, welche man nicht vergessen sollte.

In der Projektübersicht (im Admin Modus) sieht man den Project WWW Directory Path. Dorthin muss man seine Seiten ablegen.

Am besten loggt man sich in SourceForge per SSH ein:

```
ssh -l Username Projektname.sourceforge.net
```

Danach wechselt man in das Project WWW Directory. Dort kann man dann mit dem Secure Copy Befehl von SSH die Daten hochladen. Beispiel:

```
scp Datei Username@shell.sourceforge.net:/home/groups/Projektname/htdocs  
scp test.pl fontajos@shell.sourceforge.net:/home/groups/p/ph/phpeppershop/htdocs
```

Das sollte auf einer Projekt-Webpage alles enthalten sein:

- Einen Link auf die SourceForge Projektseite (z.B. <http://sourceforge.net/projects/phpepper-shop/>)
- Links zu den Mailinglisten (die gleich angelegt werden)
- SourceForge Icon
- Einen Link zum Webinterface des CVS-Servers:  
<http://cvs.sourceforge.net/cgi-bin/cvsweb.cgi/?cvsroot=Projektname>)
- Projektspezifische Daten (z.B. ein Howto für CVS oder ähnlich)

Wer direkt Projektdaten auf der Homepage darstellen will, kann dies auch tun. Genaueres findet man hier: [https://sourceforge.net/docman/display\\_doc.php?docid=1502&group\\_id=1](https://sourceforge.net/docman/display_doc.php?docid=1502&group_id=1)

### Mailinglisten anlegen

Mailinglisten und Foren sind die Kommunikationsmittel der verteilten Entwickler. Es wird empfohlen mindestens folgende drei Standard-Listen für sein Projekt anzulegen:

- announce      Hier werden neue Versionen angekündigt
- devel          Eine Mailingliste für Entwickler
- users          Dasselbe für die Benutzer

Mailinglisten anlegen und verwalten kann man über das 'Mailinglisten' Menü in der Projektübersicht. Wenn man auf 'Admin' klickt, kann man neue Mailinglisten anlegen oder be-

stehende administrieren (ist sie öffentlich, Beschreibung des Zwecks, ...).

Man hat noch die Wahl, seine Mailinglisten im 'GNU Mailman' zu administrieren. Dort kann man sich mit dem per E-Mail erhaltenen Passwort einloggen und seine Mailingliste ganz detailliert und komfortabel konfigurieren.

### News-Foren

News-Foren zum Projekt lassen sich per Klick auf 'Forums' in der Projektübersicht administrieren. In den Admin Teil kommt man über 'Admin'. Dort kann man die Foren konfigurieren. Drei News-Foren waren für uns schon automatisch eingerichtet worden.

- Developers
- Help
- Open Discussion

Es wurde noch ein weiteres, spezielles Forum angelegt: **Neuigkeiten**. Der Zweck dieses Forums ist analog zur announce-Mailingliste.

### Dokumentation

Wie koordiniert man die Dokumentation der einzelnen Funktionen und Module, welche von verschiedenen Entwicklern von irgendwo auf der Welt programmiert wurden? SourceForge bietet hier mit der Dokumentationsfunktionalität eine gute Lösung. Jeder Entwickler kann seine Dokumentation in verschiedenen Gruppen organisiert (z.B. Deutsch, Englisch,...) posten und jeder hat danach Einsicht, so muss nicht alles im Code sein.

Eine kleine aber wichtige Anmerkung dazu: Die Dateien die upgeloaded werden, müssen zuerst von einem Administrator begutachtet und für gut befunden werden, erst dann sind sie für alle ersichtlich. Auch muss man aufpassen, wenn man Plain-Text Berichte uploaden will. Alle Umbrüche sind dann weg, weil das Dokument wie eine HTML-Seite betrachtet wird (Stichwort <BR>-Tag). Auf diese Weise hat man aber gute Gestaltungsmöglichkeiten.

### (Bug-, Support-, Patch-, Feature-Request-) Tracker

Der Tracker ist ein Support-Management Tool. Man kann mit ihm 'Tickets' erstellen und an die Entwickler des Projekts senden. Für den betroffenen Entwickler, erscheint dieses mit einer Priorität versehen und nach Themen sortiert auf seiner personalisierten Einstiegsseite.

Um den Tracker zu konfigurieren kann man als Projekt-Administrator von der Projekt Übersichtsseite aus auf 'Tracker' klicken und dort die entsprechende Sparte wählen. Wenn man jetzt auf 'Admin' klickt, kann man den Tracker konfigurieren. Es ist wichtig, dass man das am Anfang macht, da die Leute ihre Meldungen sonst nicht sinnvoll einordnen können. Ausserdem geht unkonfiguriert jedes Ticket an alle Admins, was meistens wohl auch nicht erwünscht ist.

**ACHTUNG:** Einmal eingerichtete Kategorien und Gruppen können *nicht* mehr entfernt werden.

Die einzelnen Tracker-Listen können über 'update preferences' noch zusätzlich konfiguriert werden. Unter anderem kann man dort auch festlegen, ob diese Liste überhaupt zur Verfügung stehen soll.

So, jetzt kommt nur noch CVS.

## CVS einrichten

### Was ist CVS?

Wenn man mit CVS (Current Versions System: <http://www.cvshome.org/>) noch keine Erfahrung hat, braucht man ein bisschen Eingewöhnungszeit. Diese sollte man aber tunlichst auf sich nehmen. Die Vorteile einer CVS-gestützten Software-Entwicklung gegenüber einer ohne CVS, sind vergleichbar mit einem Zirkus-Seiltänzer ... mit und ohne Netz! Wer einmal mit CVS gearbeitet hat, wird es garantiert nicht mehr missen wollen.

Was macht CVS eigentlich? CVS führt Buch über die Veränderungen, die am Sourcecode vorgenommen wurden, zusammen mit einer Beschreibung dessen, was verändert wurde. Man kann sich zu jeder Zeit von jedem Ort die aktuelle und jede vorangegangene Version jeder Datei im Repository ansehen. Das CVS-Repository ist eine Datenbank, in welcher alle Daten abgelegt werden. Durch CVS können mehrerer Entwickler zur gleichen Zeit an der gleichen Datei arbeiten und ihre Änderungen zusammenführen. Wenn zwei Änderungen sich gegenseitig beeinflussen, werden diese Entwickler darauf hingewiesen und es wird sichergestellt, dass der Konflikt behoben wird, bevor die neue Version der Datei im Repository aufgenommen wird.

### CVS Entwicklungszyklus

Im Wesentlichen besteht der Entwicklungszyklus aus folgenden drei Schritten:

1. Man importiert den initialen Sourcecode ins Repository. Anschliessend checkt jeder Entwickler eine Arbeitskopie aus dem Entwicklungsbaum aus.
2. Nun arbeitet jeder Entwickler in seiner gewohnten Umgebung am Programm. Sobald ein neues Feature fertig ist, aktualisiert man zunächst die lokale Kopie, um auf dem laufenden zu sein und schreibt die veränderten Dateien zurück in den Entwicklungsbaum. Siehe dazu auch 'Update-Policy' weiter unten.
3. Sollten beim Zurückschreiben der Veränderungen irgendwelche Konflikte auftreten, wird man darauf hingewiesen und muss anschliessend manuell entsprechende Änderungen vornehmen. In den Dateien werden die veränderten Bereiche mit ">>>>" hervorgehoben und beide Versionen angezeigt. Hier löscht man entweder einfach die alte Version, oder man passt den Bereich an, damit er wie gewünscht arbeitet. Wenn alle Konflikte beseitigt sind, geht man wieder zurück zu Punkt 2 und setzt seine Arbeit fort.

### CVS für sein Projekt auf SourceForge einrichten

Um mit CVS zu arbeiten und es für sein Projekt auf SourceForge einzurichten, benutzt man verschiedene Kommandozeilen Befehle. Wenn es jemand vorzieht unter Windows mit CVS zu arbeiten, so kann er z.B. WinCVS verwenden. WinCVS bietet noch nicht die komplette Funktionalität des UNIX Kommandozeilen Pendant, es kann aber für den täglichen Gebrauch gut verwendet werden. Weitere Angaben zu WinCVS und SourceForge findet man unter: [http://sourceforge.net/docman/display\\_doc.php?docid=766&group\\_id=1](http://sourceforge.net/docman/display_doc.php?docid=766&group_id=1). Für die weitere Beschreibung gehen wir auf die UNIX Kommandozeilen Implementation von CVS ein.

Bevor man nun seinen bestehenden Code ins CVS Repository von SourceForge importiert, noch ein wichtiger Hinweis: Die Verzeichnisstruktur, welche man dem CVS System übergibt, ist 'für die Ewigkeit' gedacht. Man kann nicht einfach wieder ein Verzeichnis löschen, nachdem man es importiert hat! Wir mussten z.B. den Support von SourceForge darum bitten unser CVS-Repository entsprechend zu ändern, nachdem uns ein Faux-pas passiert ist. Man sollte sich deshalb vorher genau überlegen, was man auf CVS veröffentlicht und wie man das

gerne strukturiert haben möchte.

Weiter macht es Sinn, nicht einfach den Source Code in das CVS-Root-Verzeichnis des jeweiligen Projekts abzulegen. Man sollte vielmehr eine kleine Struktur verwenden. Die meisten Projekte haben mehrere Verzeichnisse und manchmal sogar noch Informations-Dateien im CVS-Root. Sinnvoll erscheint uns ein Verzeichnis namens `/src` für den Source Code und ein weiteres für die Dokumentation, meistens `/doc` genannt. Hier lohnt es sich, einmal bei schon bestehenden Projekten vorbeizuschauen und sich anzusehen, wie sich diese Projekte organisiert haben. Wenn man Dateien hinzufügt, welche lediglich informativen Charakter besitzen, sollte man diese mit Grossbuchstaben schreiben, da sie von UNIX dann am Anfang der Dateiaufzählung dargestellt werden.

SourceForge akzeptiert, wie schon weiter oben erwähnt, nur SSH Übertragungen. Folglich muss auch bei CVS SSH verwendet werden. Um CVS mitzuteilen, dass es SSH benutzen soll, ist es, eine Variable zu exportieren (man kann auch `setenv` (Linux: `export`) verwenden):

```
export CVS_RSH=ssh
```

Damit dieser Eintrag nach jedem Login zur Verfügung steht, empfiehlt es sich, diese Zeile der Datei `/etc/profile.local` hinzuzufügen.

### Den Quellcode ins Repository importieren (Import)

Einen CVS Import macht man grundsätzlich nur zu Beginn eines Projekts, um den vielfach schon vorhandenen Code ins CVS-Repository aufzunehmen.

Zuerst wechselt man in das Verzeichnis, dessen Unterverzeichnisse und Dateien ins CVS-Repository importiert werden sollen. Alles was jetzt im aktuellen Verzeichnis und in allen Unterverzeichnissen vorhanden ist, wird ins Repository übernommen!

Man muss folgendes Kommando benutzen um den Code zu importieren:

```
cvs -d:ext:Loginname@cvs.Projektname.sourceforge.net:/cvsroot/Projektname  
import Verzeichnisname vendor start
```

Wobei der `Loginname` und der `Projektname` entsprechend ersetzt werden müssen. Mit dem `Verzeichnisnamen` ist das Verzeichnis gemeint, in welchem die Daten auf dem CVS-Repository abgelegt werden sollen.

Nachdem man das Passwort eingegeben hat, erhält man noch die Möglichkeit, eine Message einzugeben (es wird der UNIX Editor `vi` geöffnet). Wenn man nichts mitgeben möchte, tippt man `:q!` Um den Editor wieder zu verlassen. Wer eine Message eingeben möchte, sollte sich doch zuerst mit dem `vi`-Editor befassen.

### Arbeitskopie aus dem Repository auschecken (Checkout)

Ein grosser Vorteil von CVS ist, dass man von überall auf der Welt an seinem Projekt weiter arbeiten kann. Man muss lediglich ein Checkout machen und schon hat man seine aktuelle Arbeitskopie des Projekts.

*Als eingetragener Entwickler:*

```
cvs -d:ext:Loginname@cvs.Projektname.sourceforge.net:/cvsroot/Projektname  
co Verzeichnisname
```

Wobei hier auch wieder der `Loginname` und der `Projektname` entsprechend ersetzt werden müssen. Mit dem `Verzeichnisnamen` ist das Verzeichnis gemeint, welches man vom Repository herunterladen möchte.

### Als anonymer Benutzer

```
cvsv -d:pserver:anonymous@cvs.Projektname.sourceforge.net:/cvsroot/Projektname  
login
```

Nachdem man sich nun als anonymous eingeloggt hat, kann man das eigentliche Checkout machen:

```
cvsv -d:pserver:anonymous@cvs.Projektname.sourceforge.net:/cvsroot/Projekt-  
name  
co Verzeichnisname
```

Hier muss auch der **Loginname** und der **Projektname** entsprechend ersetzt werden. Mit dem **Verzeichnisnamen** ist das Verzeichnis gemeint, in welchem die Daten auf dem CVS-Repository abgelegt werden sollen.

### Upgedatete Datei zurück ins Repository schreiben (Update, Commit)

Hier geht es darum ein weiter entwickeltes File, oder mehrere Dateien, zurück in das CVS-Repository zu schreiben.

Man sollte sich vorher aber auf eine Update-Policy einigen. Diese Policy sollte regeln, *wann* es sinnvoll ist, eine neue Version ins Repository zurück zu schreiben. Wir haben uns z.B. darauf geeinigt, dass wir immer erst dann ein Update machen, wenn wir eine neue lauffähige Version unseres Moduls oder unserer Funktion haben.

Da man mit dem Checkout-Befehl schon alle Angaben zum Projekt angegeben hat (-d) muss man sie jetzt nicht mehr explizit angeben:

```
cvsv update  
cvsv commit -m "Message"
```

An die **Message** kann man eine kleine Nachricht mitgeben. Hier drin beschreibt man in kurzen Worten, was man am Code geändert hat. Es gibt auch Projekte, welche hier eine Message-Policy eingeführt haben.

### Eine neue Datei ins Repository einfügen (Add, Commit)

Um eine neue Datei ins CVS-Repository aufzunehmen, muss man zuerst einen Checkout des Repositories lokal vorhanden haben. Darin plaziert man zuerst lokal die neue Datei am richtigen Ort im Verzeichnisbaum, programmiert vielleicht schon die Funktionalität hinein und gibt dann folgende Kommandos ein:

```
cvsv add [-kb] Dateiname  
cvsv commit -m "Message"
```

Mit der **Message** kann man wiederum analog zum Update eine beschreibende Nachricht mitgeben. Die Option **-kb** gibt CVS an, dass es sich um eine binäre Datei handelt. Man muss binäre Dateien (z.B. Word-Dateien, ...) als solche kennzeichnen, weil CVS nicht-binäre Dateien liest und gegebenenfalls gewisse Tags ersetzt.

### Webzugriff auf das Repository (Diff, ...)

Um Unterschiede zwischen Versionen von Dateien begutachten zu können, haben wir uns entschieden, das von SourceForge bereitgestellte ViewCVS einzusetzen. Dieser Service erlaubt es, neben weiteren Funktionen, das CVS Repository über das Web anzusehen. Über <http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/Projektname/> oder auch über die Projekt-Übersichtspage, kommt man darauf. Hier kann man bequem zwischen den Versionen der einzel-



nen Dateien hin und her wechseln, sie miteinander vergleichen, ihre Anzeige nach bestimmten Kriterien sortieren und man hat einen sehr schönen Überblick über das Projekt-eigene CVS-Repository. Es gibt natürlich für all diese Funktionen auch entsprechende Kommandozeilen-Befehle (siehe Anhang).

### Eine Datei oder ein Verzeichnis entfernen

**Datei:** Dies ist ein heikler Punkt von CVS. Schliesslich will man, auch wenn man eine Datei löscht, nicht deren Entwicklungsbaum verlieren. CVS bietet deshalb einen remove-Befehl an, der die Datei zwar faktisch als gelöscht markiert, sie aber nicht physikalisch vom Repository entfernt. Folgende Schritte muss man abarbeiten um eine Datei zu 'löschen':

1. Sicher stellen, dass es keine noch nicht commiteten Änderungen für das File gibt.
2. Die Datei lokal löschen (rm Dateiname)
3. cvs remove Dateiname
4. cvs commit Dateiname

**Verzeichnis:** Eigentlich gibt es keinen Weg für SourceForge Benutzer, Verzeichnisse zu löschen, es gibt aber auch hier einen Weg: Man löscht zuerst wie oben beschrieben alle Dateien im Verzeichnis (dabei können auch Wildcards wie z.B. \* benutzt werden). Nun kann man, wenn man entweder einen Checkout oder ein Update macht, die Option -P mit angeben, was für CVS heisst, dass es leere Verzeichnisse von der Arbeitskopie entfernen soll. Es wird empfohlen *immer* mit der Option -P zu arbeiten.

### Eine Datei verschieben

Um eine Datei zu verschieben, muss man lediglich die alte Datei löschen (siehe oben) und die Neue am neuen Ort erstellen. Hier ein Beispiel:

```
mv old new
cvs remove old
cvs add new
cvs commit -m "old zu new umbenannt" old new
```

### CVS Id-Tag

Wenn man ein Programm mit Hilfe von CVS verwaltet, sollte man im Header jeder Programm-Datei folgenden Eintrag als Kommentar einfügen:

```
$Id$
```

CVS wird dann nach jedem Commit an dieser Stelle die aktuelle Versionsnummer, Filenamen, Datum, Zeit und Usernamen einfügen. Bei uns sieht das zum Beispiel so aus:

```
$Id: artikel_def.php,v 1.7 2001/10/26 06:57:48 fontajos Exp $
```

## **Weitere Services von SourceForge**

### **Aufgaben (Task Management)**

Wir sind hier nicht genauer auf das Task-Management eingegangen, weil wir es für die Zeit während unserer Diplomarbeit nicht benutzt haben. Dieser Service ist aber ab einer gewissen Projektgrösse unverzichtbar. Mit ihm kann ein Projekt Administrator noch zu erledigende Aufgaben publizieren und den Entwicklern zuweisen. Auf diese Weise ist immer ersichtlich, wer an welchem Problem arbeitet und was noch zu tun ist.

### **Surveys**

Man kann im Rahmen seines Projekts auch Umfragen starten (Edit Public Info).

### **Compile Farm**

SourceForge bietet seinen Benutzern diesen wirklich aussergewöhnlichen Service an. SourceForge bietet seinen Benutzern Zugriff auf verschiedene Netzwerke und damit auf verschiedene Host Systeme mit unterschiedlichsten Betriebssystemen. Auf diese Weise kann man sein Produkt auch auf anderen Betriebssystemen und Architekturen testen. Weiter lassen sich so bequem spezielle Binaries des eigenen Projekts erstellen.

### **MySQL Datenbank**

Jedem Projekt wird auf Anfrage eine MySQL Datenbank gegeben. Diese kann frei benutzt werden. Oft wird sie dazu benutzt, eine dynamische Webseite des Projekts zu unterstützen.

### **Virtual Hosts Service**

Man kann SourceForge als Hostler benutzen. Per Webinterface kann man eine Anfrage an SourceForge senden, auf einen gewissen DNS Namen zu antworten.

### **Fragen zu SourceForge**

Hier findet man viele Dokumente zu SourceForge: [http://sourceforge.net/docman/?group\\_id=1](http://sourceforge.net/docman/?group_id=1)

## 1.8.4 Browser Kompatibilität

Alle Teile des Webshops wurden von uns mit den folgenden Browsern getestet:

### Browser älterer Generation:

- Netscape 4.7x

### Aktuelle Browser:

- Internet Explorer 5.x / 6
- Mozilla 0.9x
- Netscape 6.0 / 6.1

Bei der Wahl der zu Testzwecken verwendeten Browser beschränkten wir uns auf kostenlos erhältliche Produkte. Den Internet Explorer 5.x/6 sowie Netscape 6.x wählten wir, weil sie momentan die populärsten Browser darstellten. Netscape 4.7 brauchten wir, um eine gewisse Rückwärtskompatibilität zu wahren und Mozilla wird unserer Ansicht nach in Zukunft kontinuierlich an Beliebtheit gewinnen.

Positiv zu erwähnen ist, dass alle Browser neuster Generation die von uns erzeugten HTML-Seiten mit wenigen Ausnahmen korrekt interpretierten und darstellten. Die Definitionen in der Stylesheet-Datei wurden ordnungsgemäss ausgewertet. Der Mozilla-Browser besitzt jedoch eine wesentlich langsamere JavaScript-Engine, was uns beim Kategorienmanagement zwang, anstatt Mouse-Over, Mouse-Click-Ereignisse für die Aktualisierung des Menüs zu verwenden. Die JavaScript Aktivität reduzierte sich dadurch wesentlich, wodurch auch mit Mozilla eine akzeptable Geschwindigkeit erreicht wurde.

Ein weiterer Unterschied herrscht zwischen den Browsern Mozilla und Internet Explorer bei der Übergabe von Funktionsargumenten an eine JavaScript Funktion. Die JavaScript Engine vom Internet Explorer betrachtet die Werte, wie wenn sie mit Escape-Sequenzen codiert wären und entfernt diese selbständig. Da der Mozilla dies nicht macht, resultiert aus dem übergebenen Wert ein völlig anderes Resultat, falls im String ein oder mehrere Backslashes enthalten sind. Als Abhilfe für dieses Problem bleibt entweder die Möglichkeit, auf Sonderzeichen zu verzichten, oder die Funktionsargumente mit der PHP-Funktion `urlencode` zu encodieren (`urlencode` ersetzt alle Zeichen durch die Zeichenfolge `%HH`, wobei HH den Hexcode des betreffenden Zeichens im ASCII-Zeichensatz darstellt). So codierte Funktionsargumente müssen in der JavaScript Funktion dann natürlich wieder decodiert werden.

Wir hätten uns viele Probleme ersparen können, wenn wir auf die Kompatibilität zum Netscape 4.7x Browser verzichtet hätten. Da jedoch ein nicht vernachlässigbarer Teil der Surfer noch Browser dieser Generation einsetzt, entschlossen wir uns, den Shop auch für diese Browsergeneration brauchbar zu machen. Die Probleme begannen damit, dass Netscape 4.7 eine Stylesheet-Datei, die sich nicht im gleichen Verzeichnis wie das Webdokument befand, schlichtweg nicht interpretierte. Diesem Problem entgegneten wir, indem wir die Stylesheet-Datei in jedes Verzeichnis kopierten, in dem sich Webdokumente befanden. Die nächste Schwachstelle von Netscape 4.7x ist, dass Stylesheets nicht ordnungsgemäss vererbt werden. Wenn wir also für eine Tabelle eine Schriftart definierten, mussten wir diese Stylesheet-Angabe für das Spalten- und Zeilen-Tag zusätzlich nochmals definieren (überladen). Ausserdem musste beim Verwenden von CSS-Klassen (z.B. `body.class`, `body.main`, `body.left`) im HTML-Dokument bei jedem Tag noch explizit die Klasse angegeben werden (müsste auch vom Body-Tag geerbt werden!). Das grösste Problem bestand jedoch bei der Interpretation der Stylesheet-Angaben für Links (`<a>`-Tag). Diese wertet der Browser überhaupt nicht aus. Die einzige Lösung, die wir hierfür fanden war, die Stylesheet-Angaben im HTML-Dokument mittels `style=" . . . "` direkt in das Link-Tag zu schreiben.

Da CSS-Stylesheets ausserordentlich nützlich sind, um shopweit ein einheitliches Erscheinungsbild zu erhalten, wollten wir keinesfalls darauf verzichten. Wir nahmen uns deshalb die Mühe, die Stylesheet-Datei so zu erstellen, dass auch Netscape 4.7 damit zurecht kam. Durch mehrfaches Überladen von Styles und Klassen gelang uns dies dann nach langem experimentieren auch. Dies ist insofern sehr wichtig, da wir sonst das Layoutmanagement nur mit sehr viel Zeitaufwand hätten implementieren können.

### **1.8.5 PHP und Safe-Mode**

PHP stellte sich wiederum (wie schon bei der Projektarbeit) als sehr nützliche Scriptsprache heraus, welche die meisten für die Web-Programmierung benötigten Funktionen schon mitbringt und so eine sehr effektive Software-Entwicklung ermöglicht.

PHP besitzt einen sogenannten Safe-Mode. Ist dieser eingeschaltet, läuft PHP in einem sichereren Modus. Viele Operationen auf das Filesystem und die Ausführung von Programmen durch PHP sind dann jedoch nicht mehr möglich. Um eine gewisse Sicherheit zu haben, schalten die meisten Provider diesen Modus ein. Da wir schon in der Projektarbeit konsequent mit eingeschaltetem Safe-Mode entwickelten, behielten wir dies auf für die Diplomarbeit bei.

Theoretisch ist es per Safe-Mode möglich, an allen Stellen auf das Filesystem zu schreiben, bei denen der Inhaber der Datei gleich dem Inhaber des PHP-Scripts ist. Dies erwies sich jedoch in der Praxis als falsch, worauf wir uns nach einer anderen Lösung umschauchen mussten. Wir fanden dann heraus, dass man auf eine Datei, die schon existiert und volle Lese und Schreibrechte für jedermann besitzt, auch bei eingeschaltetem Safemode schreiben kann. Da sich die Anzahl der zum Shop gehörenden Dateien nicht verändert, erstellten wir Dummy-Files, für alles was man per Layoutmanagement und Artikelmanagement hochladen kann. Ein kleines Beispiel: Das Hintergrundbild des Hauptframes ist nach der Installation des Shops in der Datei `bg_main.gif` gespeichert. Trotzdem ist schon ein entsprechendes Pendant im JPG-Format vorhanden (`bg_main.jpg`). Lädt der Shopadministrator nun ein Hintergrundbild vom Typ JPG hoch, so wird das bereits existierende `bg_main.jpg` zum Überschreiben geöffnet. Auf diese Weise können trotz eingeschaltetem Safe-Mode Dateien hochgeladen werden.

Den zweiten Konflikt mit dem Safe-Mode hatten wir, als wir zur Verschlüsselung des Bestätigungs-E-Mails mittels Execute-Kommando das Programm GNU-PG aufrufen wollten. Der

Safe-Mode verweigerte konsequent die Ausführung von Programmen per PHP. Wir fanden dann heraus, dass man in der PHP-Konfigurationsdatei (`php.ini`) ein Verzeichnis angeben kann (`safe_mode_exec_dir`), in dem sich die Programme befinden, die von PHP ausgeführt werden dürfen. Dies klappte dann mit anderen Programmen auch, GNU-PG konnte jedoch die Keyringe nicht richtig einlesen. Dieses Problem konnten wir leider bis jetzt nicht lösen. Da die Ausführung von GNU-PG per Shell funktioniert, wäre es vielleicht ein Lösungsansatz, den GNU-PG Aufruf in ein Shell-Script zu kopieren und dieses dann per PHP auszuführen. Wir werden uns sicher nach Abgabe der Diplomarbeit an Werner Koch, den Programmierer von GNU-PG, wenden und versuchen abzuklären, ob das Problem an GNU-PG oder beim Safe-Mode liegt.

Das letzte Problem, welches wir mit dem Safe-Mode hatten, war, dass ein Shell-Aufruf von PHP aus keine Pipe enthalten durfte. Aufrufe im Stil von `echo hallo welt | irgend_ein_programm` wurden nur bis zum Pipezeichen (`|`) interpretiert, was beim obigen Beispiel eine Ausgabe von „hallo welt“ an den Browser zur Folge hatte. PHP bietet jedoch auch für dieses Problem Abhilfe mit den Funktionen: `popen` (erstellt eine Pipe), `pclose` (schliesst eine Pipe), `fputs` (schreibt auf eine Pipe) und `fgets` (liest von einer Pipe).

Abschliessend kann zu PHP gesagt werden, dass es sicher die Richtige Scriptsprache für Webanwendungen ist. Wie allgemein bekannt ist, hat jedoch auch hier Sicherheit ihren Preis.

### 1.8.6 GNU-PG

Da für einen kleineren Webshop die Kreditkartenabrechnung per Mailorder-Vertrag unserer Meinung nach die lukrativste ist und diese Zahlungsart neu mit Visa- und Eurocard/Mastercard Karten abgewickelt werden kann, entschlossen wir uns, einen Versuch mit GNU-PG zur Verschlüsselung der Bestellungen-E-Mails mit dem PGP-Verfahren zu starten.

Wie bereits im Kapitel 1.8.5 zu lesen ist, gelang es uns jedoch bei eingeschaltetem Safe-Mode nicht, einen beliebigen Text mittels GNU-PG per PHP-Aufruf zu verschlüsseln.

Folgender Aufruf funktionierte in der Shell tadellos:

```
echo "dieser text wird verschlüsselt" | gpg -a --always-trust --homedir ".gnupg"
--batch --no-secmem-warning -e --keyring .gnupg/pubring.gpg --secret-keyring
.gnupg/secring.gpg -r "glanzret@zhwin.ch"
```

<i>Parameter</i>	<i>Erklärung</i>
-a	Ausgabe mit ASCII-Hülle versehen
--always-trust	Nimmt an, dass allen verwendeten Schlüsseln vertraut werden kann.
--homedir	Ort des Homeverzeichnis
--batch	Stapelmodus: Keine Abfragen
--no-secmem-warning	Keine Memory-Sicherheits-Warnung ausgeben
-e	Daten verschlüsseln
--keyring	Als öffentlichen Schlüsselbund mitbenutzen..
--secret-keyring	Als geheimen Schlüsselbund mitbenutzen..
-r	Verschlüsseln für Name..



### 1.8.7 Strukturen beim Webprogramming

Wenn man ein Programm fürs Web entwickelt, muss man sich im klaren sein, dass gewisse starre Aufgabenteilungen und Unpässlichkeiten vorhanden sind. Es gibt viele Unterschiede zu Programmen, welche lokal auf dem Rechner laufen. In diesem Kapitel schildern wir unserer Erkenntnisse, welche wir in dieser Beziehung gewonnen haben.

#### Variablentransport zwischen Funktionen

Um Variablen zwischen Funktionen zu transportieren, arbeiteten wir von der Projektarbeit her viel mit Arrays und assoziativen Arrays. Auf diese Weise kann man bis zu zwei Variablen 'pro Zeile' transportieren. Das Problem entstand für uns immer dann, wenn von einem Objekt plötzlich mehrere Attribute verwendet wurden. Wir standen dann vor der Entscheidung, entweder die Funktionen neu anzupassen oder ein Objekt auf mehrere Array-Einträge zu verteilen. Schwieriger ist es, wenn man eine Funktion hat, welche plötzlich nicht nur ein Objekt, sondern gleich mehrere zurück geben sollte.

Ein Beispiel dafür wäre die `getgesuchterArtikel($Suchstring,...)`-Funktion (`USER_ARTIKEL_HANDLING.php`). Sie gab in der Shop-Version der Projektarbeit einen Array von Artikel-Objekten zurück. Nachdem ein Artikel nun in mehreren Kategorien sein kann, hätten wir die Klasse `Artikel` abändern müssen, was viele schon bestehende und gut funktionierende Funktionen betroffen hätte (`getArtikel,...`). Wir hätten auch den Array zur Übertragung benutzen können (ein Artikel, seine Kategorien, nächster Artikel,...). Wir haben uns aber dafür entschieden, eine neue Klasse 'Artikelmitkategorien' einzuführen. Wir schrieben also lediglich ein anderes Objekt in den Rückgabearray und konnten die Änderungen fast ausschliesslich in dieser Funktion lokal behalten.

Was hier breit an einem Beispiel erklärt wurde, ist genau genommen eine einfache Grundüberlegung: **"Erstelle Klassen für alles was du transportieren willst"**.

Eine weitere elegante Möglichkeit besteht darin, mit Objekten zu arbeiten, welche andere Objekte enthalten, um auf diese Weise die logische Struktur von Daten abzubilden. Ein Beispiel hierfür wäre unser Bestellungsobjekt: Wir haben die `get-` und `set-`Funktionen so programmiert, dass die übergeordneten Objekte die untergeordneten beinhalten und somit die logische Struktur der Daten widerspiegeln: Eine Bestellung enthält immer auch (mindestens) einen Artikel. Wir haben die `getBestellung-`Funktion so programmiert, dass man ein Bestellungsobjekt erhält, welches in seinem Artikelarray alle Artikel beinhaltet.

Unser Merksatz hier lautet: **"Bilde die Datenbankrelationen und logischen Assoziationen in den Objekten nach und erstelle die Transport-Funktionen dafür"**.

Die ganze objektorientierte Entwicklung hat eben schon seine Vorteile. Rückblickend gesehen, hätten wir unseren Shop vollständig objektorientiert entwickeln sollen.

Wenn man am Designen von neuen Funktionen ist, welche mit Objekten interagieren und diese auch weitergeben, so sollte man sich bei der Überlegung, was ich der nächsten Funktion übergeben soll, immer folgende Fragen stellen:

Frage	Design
Sind die übertragenen Daten sicherheitskritisch?	Nur die ID des Objekts übergeben. Bsp. Artikel-Preise: Ein Shop-Kunde könnte den Preis nach seinem Gusto verändert übertragen.

<i>Frage</i>	<i>Design</i>
Ist die Funktion / das Modul Performance kritisch?	<ul style="list-style-type: none"> <li>- Wenn ein Datenbank-Zugriff vermieden werden kann: Nicht einfach ein ganzes Objekt übertragen, sondern nur die wirklich benötigten Attribute eines Objekts weiterreichen.</li> <li>- Wenn sowieso auf die Datenbank zugegriffen werden muss: SQL-Query optimieren (Joins anstatt mehreren SELECTs, SQL-Zugriffe minimieren, Datenmengen minimieren (nur die benötigten Attribute aus der Datenbank auslesen).</li> </ul>
Wenn ich kann, sollte ich:	Mit numerischen IDs arbeiten. Sobald Buchstaben, Benutzereingaben oder sogar Sonderzeichen übertragen werden müssen, wird die Übertragung mühsam und fehleranfällig (siehe Kapitel 1.8.8).

### **Andere Fehlermeldungen**

Wenn man Webprogrammierung mit PHP, JavaScript und HTML betreibt, so muss man sich darüber im Klaren sein, dass man nicht den Komfort einer integrierten Entwicklungsumgebung wie z.B. Microsoft Visual C++ erwarten darf. Dies betrifft vor allem die Fehler-Rückmeldungen vom Programm zum Programmierer.

Wir hatten zu Beginn der Entwicklung zum Teil enorm Mühe, die Programmier-Fehler aufzuspüren, einfach weil wir nicht wussten, wo wir überhaupt Rückmeldungen erhalten. Wir haben ein paar dieser Fehlerfälle zusammengestellt und dokumentiert, wie wir dann jeweils vorgegangen sind.

<i>Fehler</i>	<i>Vorgehen</i>
PHP Code geändert, Reload, Fehler des Browsers: leere Seite.	Quellcode der Page ansehen, oft wird die HTML-Seite nicht angezeigt. Man sieht aber im Quellcode, wo das Programm ins Stocken geriet (SQL-Fehler, o.ä.).
HTML-Code geändert, Reload, Fehler des Browsers: leere Seite.	Quellcode der Page ansehen, überprüfen ob allfällige Tabellen, der Body und das HTML-Dokument selbst geschlossen wurden.
Neue Datei hinzugefügt, erster Zugriff, Fehler des Browsers: Zugriff nicht erlaubt.	Sind die Dateiattribute richtig gesetzt? Dürfen alle User auf die Datei zugreifen? (ideal: chmod 755).
Bild in der Datenbank geändert, Browser zeigt aber immer noch altes an	Meist ist der Cache des Browsers schuld. Vor allem beim Netscap 4.7x muss oft manuell der Cache gelöscht werden, bevor er das gleichnamige neue Bild richtig anzeigt.



<i>Fehler</i>	<i>Vorgehen</i>
Formular gemacht, Submit Button funktioniert aber nicht (kein Ereignis nach Klick).	Ist das Formular komplett? <code>&lt;form action=...&gt;...&lt;input...&gt; &lt;/form&gt;</code> . Wenn eines dieser Elemente fehlt, funktioniert das Formular nicht. Diese Situation kann vorkommen, wenn das Formular dynamisch von verschiedenen Funktionen zusammen gebaut wird.
Internet Explorer zeigt eine Seite mit Tabelle und / oder Formular an, Netscape 4.7x aber nicht.	Meistens ist die Tabelle oder das Formular nicht richtig geschlossen. Netscape 4.7x zeigt eine Tabelle erst an, wenn sie komplett eingelesen wurde.

Da es in den meisten Programmen keinen richtigen Debugger gibt, ist man auf echo "" und die Test Ausgabe der interessanten Variablen angewiesen.

### **Interaktivität einer HTML-Seite**

Der erste grosse Unterschied zwischen lokaler Programmentwicklung und Webprogramming besteht im Funktions-Handling. Wenn ich lokal ein Programm entwickle, schreibe ich eine Eingabemaske und rufe danach mit den eingegebenen Variablen eine Funktion auf. Wenn man dieses Beispiel auf eine Webanwendung mit HTML und einer serverseitigen Scriptsprache wie z.B. PHP überträgt, merkt man, dass es ein gravierendes Problem gibt: Ich kann zwar eine Maske darstellen (HTML-Formular), aber ich kann danach nicht einfach eine PHP-Funktion aufrufen! Weiter geht es nur über einen HTML-Link. Entweder ist die URL im action= Attribut des Formular-Tags `<form ..>` definiert, oder sie ist in Form eines gewöhnlichen Links im Anchor-Tag vorhanden. Man merke sich: "**Von einer Webpage zur anderen geht's nur über eine URL weiter**". Diese Erkenntnis ist ziemlich wichtig!

Um dieser Problematik Herr zu werden, haben wir im PhPepperShop die sogenannte Aufruf-Schicht eingeführt (siehe Kapitel 2.2.1, Übersicht über die PHP-Funktionen und Module). Wir erstellten HTML-Seiten, welche PHP if- und elseif-Verzweigungen beinhalten. Nach jeder HTML-Page ruft man je nach Workflow sich selbst (`$PHP_SELF`) oder eine andere PHP-Datei auf und übergibt die darstellende Variable mit einer Nummer. Je nach Nummer kommt man so in einer Datei (bei uns Modul genannt) an verschiedene Orte, von wo aus dann PHP-Funktionen aufgerufen werden, welche dann auf den Daten operieren.

Ein weiterer Unterschied zwischen lokalen Programmen und Webanwendungen ist, dass man bei lokalen Programmen meistens ereignisgesteuerte Programm-Elemente innerhalb einer dargestellten View (Seite) hat. Hier muss man zuerst einmal ziemlich umdenken. In Webanwendungen arbeitet man primär mit der Seitenbeschreibungssprache HTML. HTML ist nicht dafür entwickelt worden, ereignisgesteuerte Operationen starten zu können. HTML Seiten sind in sich statisch. Wenn ich also z.B. ein Texteingabefeld für einen Artikelpreis habe, so kann ich nicht nach der Eingabe eines Textes prüfen, ob dieser Text auch alphanumerische Zeichen beinhaltet. Die Auswertung mittels PHP kann frühestens nach Aufruf einer weiteren HTML-Seite erfolgen und ist somit für gewisse Aufgaben zu starr. Hier setzt JavaScript an. Mit JavaScript kann man innerhalb einer HTML-Page dynamische, ereignisgesteuerte Elemente benutzen.

Ein weiteres Problem ist die Verzweigung des Workflows, je nach Benutzereingabe auf der gleichen Seite. Hierzu mehrere Problemstellungen und wie man sie lösen kann:

<i>GUI-Elemente</i>	<i>Wie weiter</i>
Ich habe mehrere Knöpfe, der Shopbenutzer wird, je nachdem auf welchen Knopf er drückt, auf eine andere Seite geführt.	Jeden Knopf (IMG-Tag) in einen <code>&lt;a href=".."&gt;</code> Tag setzen. Somit hat man mit jedem Knopf einen eigenen Link erstellt, welcher gegebenenfalls Variablen per GET weiterreichen kann (2kB-Grenze bei GET nicht vergessen!).
Textfelder und / oder Radio-Buttons und / oder Checkboxen und / oder Pull-Down Menüs. Mit einem Knopf geht es weiter.	HTML-Formular erstellen. Man kann den Submit-Knopf auch als Grafik darstellen mit dem Tag: <code>&lt;input type=image src=".."&gt;</code>

### **Entwurf von neuen Komponenten für den PhPepperShop: (Entwicklungszyklus)**

Während der Diplomarbeit erweiterten wir das von der Praktischen Arbeit begonnene Framework für einen Webshop in PHP kontinuierlich zum PhPepperShop. Wir ergänzten den Shop komponentenweise um neue Funktionen. Dabei ist uns bewusst geworden, welche Elemente des PhPepperShops betroffen werden, wenn man ihm eine neue Funktionalität hinzufügen will. Wir haben heraus gefunden, dass folgende Überlegungen und Implementierungsschritte nacheinander abgearbeitet werden müssen, damit man eine neue Funktion oder sogar Komponente effizient und fehlerlos einbauen kann:

#### **1. Sich fragen:**

1. Was brauche ich? (Use-Cases helfen hier!)
  - Datencontainer (Klassen)
  - Datenbankzugriffsfunktionen (get, new, upd, del)
2. Welche bestehenden Funktionen benutze ich?
  - Ausnutzen was es schon gibt
  - Was für Argumente und Rückgabewerte werden dort benutzt?
3. Habe ich Security-relevante oder Performance-kritische Datenübertragungen?
4. Gibt es kapselbare / immer wiederkehrende / redundante HTML-Ausgaben?
  - Darstellen-Funktionen erstellen

#### **5. Datenbank anpassen** (meistens um Tabellen erweitern)

#### **6. SQL Dateien anpassen** (Security-Stufen beachten)

#### **7. Klassen definieren** (Klassenname\_def.php Namenskonvention beachten)

#### **8. Install-Script anpassen** (Links nötig? --> config\_part4.sh)

#### **9. Includes anpassen** (für Objekte und weitere Module)

- Neue Klassen werden je in USER\_ARTIKEL\_HANDLING.php und in USER\_BESTELLUNG.php eingebunden. Wenn es eine reine Administrator Funktion ist, kann man sie auch erst in SHOP\_ADMINISTRATION.php einbinden.

#### **10. Funktionen schreiben** (auch get, new, upd, del neben der eigentlichen Funktionalität)

#### **11. Funktionstests**

#### **12. Formularmasken programmieren** (inkl. JavaScript Eingabeprüfung und Sonderzeichen-Unterstützung)

13. **Aufrufe in bestehende oder neue Module integrieren** (darstellen = ...)
14. **Eingliederung der neuen Funktionalität in die Shop oder Admin-Menüstruktur**
15. **Test auf Sonderzeichen und Extremwertetest** (negative, 0 und grosse positive Werte)
16. **Test der Gesamtfunktionalität**
17. **Commit auf SourceForge**

### 1.8.8 Variablen zwischen den Webpages transportieren

Um einen Variablenwert von einer HTML-Seite zur Nächsten weiterzugeben, können folgende Verfahren verwendet werden:

#### Übermittlung direkt im Link (GET)

Bei der direkten Datenübertragung werden der URL die zusätzlich zu Übertragenen Variablennamen und Werte angehängt. Diese werden dann in der URI dem nächsten Dokument übergeben.

#### Aufbau der URI:

```
Link ? Variablenname_1 = Variablenwert_1 & Variablenname_2 =
  Variablenwert_2 & ...
```

#### Praktisches Beispiel:

```
href=" ./auswertung.php?name=mueller&vorname=hans&darstellen=1 "
```

Es ist zu beachten, dass die maximale URI-Länge von Browser zu Browser unterschiedlich ist. Es sollte keine URI übergeben werden, die länger als 2 KByte ist. Diese Längenbeschränkung grenzt das Einsatzgebiet stark ein. Ausserdem dürfen Sonderzeichen nicht direkt übergeben werden!

Wir haben dieses Verfahren während der ersten Projektarbeit einige Male dazu verwendet, die Variable für die Darstellen-Weiche weiterzugeben. Da der Surfer jedoch die ganze URI sieht und natürlich auch verändern kann, besteht die Gefahr, dass er unvorhersehbare und nicht erwünschte Zustände herstellen kann. Aus diesem Grund haben wir weitgehendst auf den weiteren Einsatz von GET verzichtet.

#### Übermittlung als Feld in einem Formular (POST oder GET)

Bei der Übermittlung der Variablen in einem Formular hat man die Wahl zwischen GET und POST. GET funktioniert, wie oben detailliert beschrieben, durch Anhängen der Daten an die URL. Es gilt hier dementsprechend auch die gleiche Längenbeschränkung (2 KByte). Sonderzeichen werden aber vom Browser ASCII-codiert übergeben und auch wieder decodiert.

Da beim Formularversand mittels GET die URI wiederum vom Browser angezeigt wird und natürlich auch manipuliert werden kann, ist von dieser Methode abzuraten.

Die Variablenweiterleitung per POST-Formular ist die von uns favorisierte Methode, welche wir mit Abstand auch am häufigsten eingesetzt haben. Das ganze Formular ist in seiner Grösse nicht beschränkt. Es können somit auch grössere Datenbestände, wie Bilder, weitergegeben werden. Ein weiterer positiver Aspekt ist, dass dies für den Surfer transparent geschieht. Er kann die Daten nicht so einfach manipulieren und sieht den übermittelten Datensatz nur im HTML-Quellcode.

Die Variablen können in allen möglichen Formularfeldern übermittelt werden:

<i>Formularfeldtyp</i>	<i>Beschreibung</i>
text	Einzeiliges Eingabefeld
checkbox	Kontrollkästchen
radio	Optionsschaltfläche
submit	Sendeknopf

<b>Formularfeldtyp</b>	<b>Beschreibung</b>
reset	Knopf zum zurücksetzen des Formulars
textarea	Mehrzeiliges Texteingabefeld
select	Pull-Down-Menü
passwort	Verdecktes Eingabefeld
button	Schaltfläche
hidden	Unsichtbares Feld

### Aufbau eines Formularfeldes:

```
<input type="Formularfeldtyp" name="Variablennamen" value=" Variablenwert" >
```

### Beispiel eines Formulars mit einem text- und einem hidden-Feld:

```
<form method='post' action='./index.php'>
  <input type="hidden" name="darstellen" value="1">
  <br><B>Geben Sie ihre Land ein:</B>
  <input type="text" name="Land" size="30" value="Schweiz">
  <input type="submit" name="weiter" value="weiter">
</form>
```

Mit dem Attribut value kann den meisten dieser Eingabefelder ein Initialwert zugewiesen werden. Dieser wird dem Surfer dann angezeigt und kann editiert werden. Der editierte Wert wird dann beim Drücken des Sendeknopfs als Variable übermittelt.

Um eine Variable unverändert weiterzugeben (zum Beispiel über mehrere HTML-Seiten), kann sie in ein hidden-Feld geschrieben werden. Dieses kann vom Surfer nicht verändert werden und ist nur im Quellcode ersichtlich. Das hidden-Feld eignet sich auch sehr gut zur Weitergabe der Darstellungsweise (Entscheidung, welche HTML Seite als nächstes ausgegeben wird).

Was unbedingt beachtet werden muss, ist, dass Formularfelder nach der Übermittlung bei gewissen Sonderzeichen (z.B. \,',") einen Backslash vorgestellt haben. Diese Backslashes müssen mit der PHP-Funktion stripslashes entfernt werden. Gibt man eine Variable zum Beispiel in einem Hidden-Feld dreimal weiter, ohne jeweils die stripslashes bei jedem Schritt auszuführen, so wird jedem Sonderzeichen bei jeder Weiterleitung ein Backslash vorangestellt, wobei dieser bei der nächsten Weiterleitung auch wieder als Sonderzeichen behandelt wird!

Folgendes Beispiel verdeutlicht, was passiert, wenn stripslashes vergessen wird:

```
Orginalstring:        /sonderzeichen\ 'string'
Einmal weitergegeben: /sonderzeichen\\ \'string\'
Zweimal weitergegeben: /sonderzeichen\\\\ \\\'string\\\\
Dreimal weitergegeben: /sonderzeichen\\\''string\\\''
```

### Welche Daten sollen wie übergeben werden?

Erst in einer späten Phase der Programmierung haben wir angefangen, die Formulare und deren Verarbeitung auf Herz und Nieren zu prüfen. Wir mussten jedoch schnell feststellen,

dass zwar unsere logischen Überlegungen stimmten, es jedoch trotzdem möglich war, mit extremen Eingaben ein Chaos zu veranstalten.

In folgender Tabelle haben wir zusammengefasst, wie die Daten in den verschiedenen Fällen Codiert/Decodiert werden müssen, damit sie ordnungsgemäss ankommen.

### Legende:

FORM:	Formularfeld; kein Unterschied zwischen sichtbar und hidden.
FORM (sichtbar):	sichtbares, vom Surfer veränderbares Formularfeld
FORM (hidden):	für den Surfer weder sichtbares, noch veränderbares Formularfeld
PHP:	PHP-Variable
DB:	Datenbank

<i>was wird übertragen</i>	<i>Von? Nach</i>	<i>Codier-/Decodier Massnahmen</i>
[a-z], [A-Z], [0-9]	FORM? PHP	keine
[a-z], [A-Z], [0-9]	FORM? DB	keine
[a-z], [A-Z], [0-9]	PHP? DB	keine
[a-z], [A-Z], [0-9]	PHP? FORM (sichtbar)	keine
[a-z], [A-Z], [0-9]	PHP? FORM (hidden)	keine
[a-z], [A-Z], [0-9]	DB? PHP	keine
Alle Zeichen inkl. Sonderzeichen	FORM (sichtbar)? PHP	stripslashes
Alle Zeichen inkl. Sonderzeichen	FORM (hidden)? PHP	1. urldecode 2. addslashes
Alle Zeichen inkl. Sonderzeichen	FORM (sichtbar)? DB	keine
Alle Zeichen inkl. Sonderzeichen	FORM (hidden)? DB	1. urldecode 2. addslashes
Alle Zeichen inkl. Sonderzeichen	PHP? DB	addslashes
Alle Zeichen inkl. Sonderzeichen	PHP? FORM (sichtbar)	htmlspecialchars
Alle Zeichen inkl. Sonderzeichen	PHP? FORM (hidden)	urlencode
Alle Zeichen inkl. Sonderzeichen	DB? PHP	keine

Der Weg vom Formular zur Datenbank geht eigentlich über eine PHP-Variable. Da aber die Daten vom Formular her schon mit vorangestellten Backslashes kommen, kann das stripslashes/addslashes-Prozedere entfallen, weil die Daten sowieso mit vorangestellten Backslashes an die Datenbank übergeben werden müssen.

### **PHP-Funktionen: addslashes und stripslashes**

Mit der Funktion addslashes werden Sonderzeichen behandelt, die oft von Datenbanken interpretiert werden.

' " \ NUL

NUL steht dabei für den absoluten Wert 0. Diesen Sonderzeichen wird ein Backslash voran-

gestellt, damit sie nicht interpretiert werden. Die Funktion `stripslashes` macht die durch `addslashes` vorgenommenen Umwandlungen wieder rückgängig. Der Vorgang wird übrigens auch oft als „escapen“ bezeichnet.

Werden Formulardaten übergeben, bei denen sich eines oder mehrere der oben aufgelisteten Sonderzeichen im String befinden, fügt der Browser automatisch Escape-Zeichen ein,

<i>Eingabestring</i>	<i>Funktion</i>	<i>Ausgabestring</i>
"string" zum 'übertragen'	<code>addslashes</code>	<code>\\"string\\"</code> zum <code>\'übertragen\'</code>
<code>\\"string\\"</code> zum <code>\'übertragen\'</code>	<code>stripslashes</code>	"string" zum 'übertragen'

### **PHP-Funktionen: `urlencode` und `urldecode`**

Die beiden Funktionen `urlencode` und `urldecode` sind eigentlich dazu entwickelt worden, Strings, die direkt an die URL-angehängt werden, so umzuwandeln, dass sie den Konventionen für den Aufbau einer URL genügen.

Dies geschieht, indem Zeichen, welche nicht direkt in der URI übergeben werden können, hexadezimal codiert werden. Dabei handelt es sich um die Zeichenfolge `&HH`, wobei HH für den hexadezimalen ASCII-Wert des speziellen Zeichens steht. Leerschläge werden durch ein `+` ersetzt.

Da URL-encodierte Strings weder Sonderzeichen, die escaped werden könnten, noch Leerschläge beinhalten, eignet sich diese Codierung hervorragend zur unkomplizierten Weitergabe von Daten über mehrere Seiten in Hidden-Feldern.

Enthält ein String viele Sonderzeichen, kann dessen URL-encodierter String viel länger als der Originalstring werden, da jedes Sonderzeichen durch drei Codier-Zeichen ersetzt wird. Dies muss vor allem beachtet werden, wenn an irgend einer Stelle eine Längenbeschränkung verwendet wird.

<i>Eingabestring</i>	<i>Funktion</i>	<i>Ausgabestring</i>
"string" zum 'übertragen'	<code>urlencode</code>	<code>%22string%22+zum+%27%FCbertragen%27</code>
<code>%22string%22+zum+%27%FCbertragen%27</code>	<code>urldecode</code>	"string" zum 'übertragen'

Zu beachten ist, dass der Punkt von `urlencode` nicht als Sonderzeichen behandelt wird, was zu Problemen führen kann, wenn ein Variablenwert als Variablenname verwendet werden soll. Mit der Funktion `str_replace` kann auch der Punkt noch durch die entsprechende ASCII-Sequenz (`%2E`) ersetzt werden.

```
str_replace(".", "%2E", $einsetzen);
```

`urldecode` ersetzt alle ASCII-Sequenzen mit dem entsprechenden Zeichen, wodurch eine gesonderte Behandlung des Punktes wegfällt.

### **PHP-Funktion: `htmlspecialchars`**

Die Funktion `htmlspecialchars` wandelt die folgenden Zeichen so um, dass die Darstellung in HTML korrekt erfolgt:

- & wird als &amp; dargestellt
- " wird als &quot; zurückgegeben
- < wird in &lt; umgewandelt
- > wird als &gt; ausgegeben

<i>Eingabestring</i>	<i>Funktion</i>	<i>Ausgabestring</i>
"string" <zum> &übertragen&	urlencode	&quot;string&quot; &lt;zum&gt; &amp;übertragen&amp;

HTML-Ausgaben müssen unbedingt mit urlencode codiert werden, wenn man verhindern will, dass Zeichenfolgen, die sich in eckigen Klammern (<>) befinden, als Tag interpretiert werden. Wir nutzten diese Funktion vor allem, um Formularfeldwerte mittels value-Attribut auszugeben:

```
<?php
    $variable='enthält "HTML"-Sonderzeichen und <Tags>';
?>

<form method='post' action='./index.php'>
    <br><b>Editieren Sie den Text:</b>
    <input type="text" name="form_data"
        value="<?= htmlspecialchars($variable)" size="40">
    <input type="submit" name="weiter" value="weiter">
</form>
```

### 1.8.9 JavaScript

Ursprünglich war die Verwendung von JavaScript im Shop nicht geplant. Durch der Einführung von Unterkategorien standen wir plötzlich vor dem Problem, dass wegen des dynamischen Menüs im linken Frame mit einem Mausklick zwei Frames aktualisiert werden mussten. Da dies mit reinem HTML nicht möglich ist, standen wir vor der Wahl, entweder den ganzen Shop so umzuschreiben, dass auf Frames verzichtet werden kann oder eine JavaScript Funktion aufzurufen, die beide Frames gleichzeitig aktualisiert. Wir entschieden uns für letztere Lösung, da sie wesentlich weniger Zeitaufwand zur Implementation benötigte.

#### **Gleichzeitige Aktualisierung von Frames mittels JavaScript**

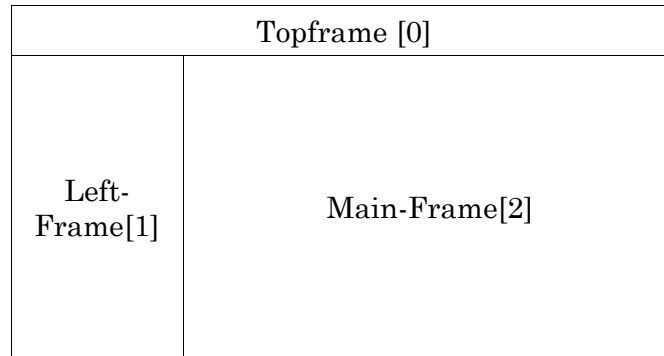
Folgende kleine JavaScript Funktion wird dazu benötigt:

```
<script language="JavaScript">
    function zweiframes(url1,f1,url2,f2) {
        parent.frames[f1].location.href=url1;
        parent.frames[f2].location.href=url2;
    }
</script>
```

#### **Funktionsargumente:**

url1, url2: URL des Dokuments, welches aufgerufen werden soll  
f1, f2: Nummer des zu aktualisierenden Frames





Die Funktion „zweiframes“ wird direkt im Link folgendermassen aufgerufen:

```
<a href='javascript:zweiframes("url1", "f1", "url2", "f2")'>Link-
text</a>
```

### Überprüfung eines Formularfeldes auf Eingabe

Um im Shop einkaufen zu können, muss nun fortwährend im Browser JavaScript eingeschaltet sein. Diesen Umstand nutzten wir gleich dazu, auch noch die Eingabeüberprüfung der Formulardaten per JavaScript zu tätigen. JavaScript hat im Gegensatz zu PHP den Vorteil, dass die Überprüfungsfunktion gleich mit dem Formular selbst zum Browser gesendet werden kann. Die Überprüfung, ob alle Felder ordnungsgemäss ausgefüllt wurden, geschieht demzufolge beim Surfer (Client-seitig) und erfordert weder zusätzlichen Datenverkehr, noch Rechenpower vom Server. Gibt ein Surfer zum Beispiel die Daten fünfmal nicht ordnungsgemäss ein, wird dadurch kein einziges Byte zum Server gesendet. Macht man diese Kontrolle mittels PHP, müssen neun (fünf Fehlermeldungs- und vier Formular-) HTML-Seiten vom Server zum Client geschickt werden, was einem massiven Mehraufwand an Ressourcen entspricht.

Folgendes Beispiel illustriert die minimalste Art der Eingabeüberprüfung. Es wird nur überprüft, ob der Surfer etwas in das Formularfeld eingegeben hat:

```
<script language="JavaScript">
  function chkFormular() {
    /* Überprüfung, ob kein Leerstring übergeben wurde */
    if(document.Formular.Vorname.value == "") ) {
      /* Alert-Warnung ausgeben, wenn ein Leerstring übergeben
        wurde */
      alert("Bitte einen Vornamen eingeben!");
      /* den Fokus auf das Feld Vorname setzen */
      document.Formular.Vorname.focus();
      /* false zurückgegeben, damit das Formular nicht abge-
        schickt wird */
      return false;
    }
  }
</script>

<form name="Formular" onSubmit="return chkFormular()" method='post'
  action= './index.php' >
  <br><b>Geben Sie Ihren Vornamen ein:</b>
  <input type="text" name="Vorname" size="40">
  <input type="submit" name="weiter" value="weiter">
```

</form>

Hat man viele Felder zum Überprüfen in einem Formular, lohnt es sich, die Überprüfung der Felder in einer for-Schleife durchzuführen, da sonst eine nicht vernachlässigbare Menge an JavaScript Code übermittelt werden muss.

### ***Unsere Meinung zu JavaScript***

Wir haben das Programmierprinzip von JavaScript als sehr angenehm empfunden. Dank der konsequenten Objektorientiertheit dieser Scriptsprache, war es problemlos möglich, an alle gewünschten Webelemente zu kommen.

Dies bewog uns dann dazu, JavaScript auch im Hauptmenü des Kategorienmanagements zu verwenden. Leider machte uns hier der Netscape 4.7-Browser wieder einmal einen Strich durch die Rechnung, da er ein schon ausgegebenes Bild nicht durch ein anderes ersetzen kann, wenn es nicht die exakt gleichen Dimensionen besitzt. Wir bauten eine Browserweiche ein und gaben fortan Netscape 4.7-Browsern ein eigenes Menu aus.

Was wir bei der Entwicklung von JavaScript Funktionen als äusserst störend empfanden, war, dass man vom Browser nur mit äusserst spärlichen Informationen versorgt wird, wenn ein Problem mit einer JavaScript Funktion vorliegt. Es ist vorgekommen, dass mitten in der Funktion die Ausführung ohne Fehlermeldung oder Warnung abgebrochen wurde. Etwa zwei Wochen vor dem Ende der Diplomarbeit ist dann Mozilla 0.9.5 freigegeben worden. Dieser Browser enthält eine sehr nützliche JavaScript-Konsole und sogar einen JavaScript-Debugger. Leider mussten wir bis zu diesem Zeitpunkt ohne ein solch komfortables Tool auskommen.

## 1.8.10 Datenbankdesign

### Unterkategorien

Auch im Datenbankdesign haben wir Erfahrungen gesammelt. Als es darum ging Unterkategorien zu implementieren, haben wir uns nach intensivem Gedankenaustausch dafür entschieden, die Unterkategorie-Tiefe auf maximal eine zu beschränken. Dies vor allem, weil in einem Shop im linken Frame die Darstellung der Auswahl-Kategorien mit mehreren Verschachtelungen ungewohnt aussieht und sich nicht intuitiv bedienen lässt, wenn man immer nur einen offenen Kategoriebaum darstellen will (aus Übersicht). Weiter würde mit tieferen Verschachtelungen die ohnehin schon beschränkte Namenslänge weiter eingeschränkt werden (sichtbarer Bereich).

Die einfachste und zeitsparendste Implementationsmethode ist, die Kategorientabelle um das Attribut 'Unterkategorie\_von' zu erweitern. Dieser Designentscheid ist unkonventionell und aus Sicht der Datenbank nicht unbedingt optimal. Die Datenbank bildet hier die logische Verknüpfung von Kategorie und ihren Unterkategorien nicht ab. Um die logische Verknüpfung nachzubilden, hätte man eine Tabelle Unterkategorien einfügen müssen und diese mit einer '1 zu n'-Beziehung an die Kategorien-Tabelle hängen müssen.

Wir haben uns hier für den einfacheren Weg entschieden, im Bewusstsein, dass die Datenbank unser Datenmodell nicht genau nachbildet. Wir können dies folgendermassen begründen:

- Zeitersparnis
- Mehrere Unterkategorie-Ebenen sind bei Bedarf immer noch möglich, indem man im `Unterkategorie_Attribut` eine weitere Unterkategorie angibt. Man ist also nicht definitiv auf nur eine Ebene beschränkt.
- Viele Funktionen konnten unverändert mit der erweiterten Tabelle arbeiten
- Keine weiteren Klassen mussten eingeführt und included werden

Ein Designfehler ist uns aber dennoch unterlaufen: Wir haben nicht weit genug in die Zukunft geschaut. Im Attribut `Unterkategorie_von`, setzten wir den *Namen* der Haupt-Kategorie ein. Wie wir später erfahren mussten ein in mehrfacher Hinsicht schwerwiegender Fehler, welcher uns aber neben den zu umschiffenden Unwägbarkeiten auch einiges an Erfahrung einbrachte.

Wann immer wir performance-kritische Funktionen zu schreiben hatten, übergaben wir die benötigten Attribute in einem Formular als hidden-Field oder per Link im Query-String. Hier kamen nun allerlei Encodierungsprobleme auf uns zu, da der Shop auch Sonderzeichen richtig behandeln können sollte (siehe auch Kapitel 1.8.8). Weiter waren Kategorienamen vor der Implementierung von Unterkategorien eindeutig und einmalig (wir hatten für den Kategorienamen sogar ein UNIQUE-Statement in der Datenbank). Auf einmal kann man aber mehrere Kategorien mit gleichem Namen haben. Da wir hier nicht mit der Kategorie-ID arbeiten, erbt die neue Kategorie alle Unterkategorien der schon bestehenden. Solche Probleme hätten sich vermeiden lassen, wenn man sich auf die Kategorie-ID bezogen hätte.

Wir haben daraus gelernt und werden wohl nie mehr ein anderes Verknüpfungsattribut als eine ID wählen.

### Shop-Settings

Die Shop-Settings Tabelle beinhaltet die allgemeinen Shopeinstellungen. Dies sind inzwischen ganze 40 Attribute, welche auf einer Zeile gespeichert werden. Wir haben uns in der Praktischen Arbeit gedacht, dass man pro Shop eine Zeile in dieser Tabelle hat und auf diese

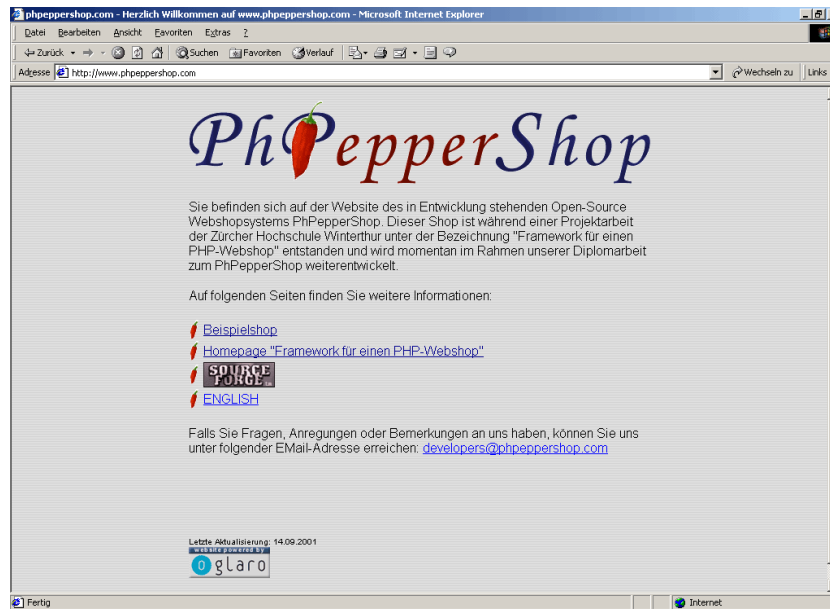
Weise bequem mehrere Shops in dieser Tabelle verwalten kann. Heute wissen wir, dass das so nicht funktioniert. Die parallel laufenden Shops haben alle eine eigene Datenbank. Die Shop-Settings-Tabelle wird zunehmend zu einem Problem, weil die eine Zeile viel zu lang ist. Unser Dozent hat uns darauf hingewiesen, dass z.B. die Datenbank Sybase und auch andere Datenbanken eher restriktiv sind, was ihre maximale Zeilenlänge anbelangt. In nächster Zukunft müsste man also diese Tabelle umstrukturieren, zum Beispiel so:

- `Attributname varchar(256);`
- `Attributwert varchar(256);`

Dies bedingt aber auch ein Anpassen vieler Funktionen und wurde deshalb auf die Zeit nach der Diplomarbeit verschoben. Es handelt sich hier um reine Fleissarbeit ohne weiteren Lerneffekt.

## 1.9 Zukunft des PhPepperShops

Da wir nicht wussten, wie lange wir den Testserver der Schule noch zur Verfügung haben, registrierten wir die Domain **phpeppershop.com**. Diese Domain ist seit dem 13.09.2001 aufgeschaltet und wird seither als offizielle Informationsseite betrieben.



Mit dem Abschluss des Studiums werden wir auch unsere bis anhin verwendeten ZHW E-Mail-Adressen nicht mehr zur Verfügung haben. Damit wir auch danach noch für Fragen und Feedback erreichbar sein werden, richteten wir eine offizielle Kontaktadresse ein:

**developers@phpeppershop.com**

### 1.9.1 Veröffentlichung des Final-Release

Als offizielles Veröffentlichungsdatum haben wir den **15. November 2001** festgesetzt. Bis zu diesem Datum müssen jedoch noch einige Arbeiten erledigt werden:

- Backup des Shops automatisieren
- Bugfixes
- FAQ-Liste und User-Forum auf <http://www.phpeppershop.com> einrichten

### 1.9.2 Leute, die den Shop weiterentwickeln wollen

Folgende Leute haben sich bereit erklärt, unter Umständen an der Weiterentwicklung des Shops mitzuarbeiten, oder den Shop in eine andere Sprache zu übersetzen:

- Uli und Christian / deutsch / Andalusien?? wollen mitentwickeln  
[Uli@prismaworld.com](mailto:Uli@prismaworld.com) oder [Webmaster@fileload.de](mailto:Webmaster@fileload.de)
- Hans, Physiker / englisch /? Security testen  
[thesis@xs4all.nl](mailto:thesis@xs4all.nl)
- Peter Fitzpatrick / englisch /? Secure Online Transaction (ihm Infos schicken)  
[petefitzy@btinternet.com](mailto:petefitzy@btinternet.com)

- Hans-Peter Ambros (Fan von pgmarket) / deutsch /? will mitentwickeln (v.a. spezialisierte Lösung)  
[ambros@planet-interkom.de](mailto:ambros@planet-interkom.de)
- Hans-Jakob Martin / deutsch /? will mitentwickeln, falls er bis dann nicht schon einen anderen Shop benutzt (Lager, Buchhaltung)  
[Hans-Jakob.Martin@gmx.net](mailto:Hans-Jakob.Martin@gmx.net) [www.nordakademie.de](http://www.nordakademie.de)
- Georg Spang / deutsch /? will mitentwickeln (spezialisierte Lösung)  
[georg\\_spang@gmx.de](mailto:georg_spang@gmx.de)
- Alex Henriksen / englisch /? übersetzt englischen Shop ins Dänische  
[alexie@zbox.dk](mailto:alexie@zbox.dk)
- Steven Dowd / englisch /? übersetzt den Shop ins Englische  
[steven.dowd@dowd.co.uk](mailto:steven.dowd@dowd.co.uk)
- Phil / deutsch /? will Shop in PHP-Nuke integrieren  
[p-fs@gmx.de](mailto:p-fs@gmx.de)
- kingluui / deutsch / ? Shop vom Deutschen ins Spanische übersetzen  
[xxxforumbanner@kingluui.de](mailto:xxxforumbanner@kingluui.de)

## 1.10 Schlusswort

Wir haben unser Ziel erreicht! Der Shop kann der Öffentlichkeit übergeben werden. Wir wissen, dass er auch in Zukunft weiter entwickelt werden wird und freuen uns schon auf weitere interessante Features und Feedbacks.

Der PhPepperShop ist etwas Einmaliges. Er bietet eine enorme Flexibilität und Dynamik, welche auch kommerzielle und teure Shop-Lösungen oft vermissen lassen.

Aufgrund des enormen Echos auf unsere Projektarbeit von diesem Frühjahr, waren wir hoch motiviert, den Shop auf die Bedürfnisse des Marktes anzupassen. Dies ist uns unserer Meinung nach auch gelungen.

Bei Herrn Feisthammel möchten wir uns ganz herzlich für seine ausserordentliche Unterstützung, die er uns auch in diesem Projekt wieder gewährleistete, bedanken.

Speziellen Dank möchten wir unseren Freundinnen aussprechen, die uns auch in dieser letzten grossen Arbeit unseres Studiums viel Verständnis und Unterstützung entgegen brachten.

Winterthur, den 28.10.2001

Reto Glanzmann

José Fontanil

## 2. Anhang

### 2.1 Literaturverzeichnis

Nachfolgend eine Liste mit allen Quellen und Literaturverweisen, aufgliedert nach den einzelnen Themen, die durch das Webshop-Programming tangiert werden.

#### 2.1.1 PHP

**PHP 4 Grundlagen und Profiwissen**

Carl Hanser Verlag München Wien  
Jörg Krause  
November 2000  
1161 Seiten  
ISBN 3-446-21546-8

Dies war, wie auch in der Projektarbeit, unser 'Hauptbuch'. Es enthält eine Vielfalt an Themen, die alle ziemlich ausführlich erklärt werden. Aber auch dieses Buch hat seine Grenzen (Bilder in Datenbank speichern, grössere Projekte mit PHP)

**PHP Grundlagen und Profiwissen**

Carl Hanser Verlag München Wien  
Jörg Krause  
März 2000  
1067 Seiten  
ISBN 3-446-21301-5

Dieses Buch ist die Vorgänger-Version des oben genannten. Wir haben es selten konsultiert, aber doch ab und zu etwas daraus brauchen können.

**PHP kurz & gut**

O'Reilly Verlag Köln  
Rasmus Lerdorf  
1. Auflage 2000  
111 Seiten  
ISBN 3-89721-225-0

Eine praktische, kompakte Zusammenfassung, aber leider nur für PHP3.

**PHP 4 Das bhv Taschenbuch**

bhv Verlag Landsberg  
Uwe Hess, Günther Karl  
1. Auflage April 2001  
764 Seiten  
ISBN 3-8287-5057-5

Eine gute Einführung in die Grundlagen von PHP4, der Stoff geht aber nicht all zu tief.

**Linux Enterprise 'PHP Special'-Ausgabe**

Sonderausgabe Herbst 2001  
98 Seiten

Ein Blick in die aktuelle PHP Entwicklung, mit Vorstellung der PHPLib und PEAR. Viele interessante, professionelle Ansätze.

**PHP Archive / -Foren**

<http://www.hotscripts.com/>  
<http://www.phpbuilder.com/>  
<http://www.phparchiv.de/>

Diese drei Pages halfen uns während der Entwicklung oft weiter. Viele Scripts findet man auf hotscripts und im phparchiv. Kompetente Foren in phpbuilder.



## 2.1.2 Apache Webserver

### Apache Web-Server Installation, Konfiguration, Administration

MITP-Verlag GmbH Bonn

Lars Eilebrecht

3. Auflage 2000

671 Seiten

ISBN 3-8266-0612-4

Ein sehr tief gehendes Buch. Half uns sehr gut bei der Apache-Konfiguration und beim Einrichten von htaccess.

## 2.1.3 CRM – Customer Relationship Management

### iX Magazin für professionelle Informationstechnik

*Ausgabe 8/2001, ab Seite 38*

#### **Operatives und analytisches CRM, CRM und Betriebswirtschaft**

*Ausgabe 7/2001, ab Seite 104*

#### **Dienen und Verdienen - CRM: Der Wert eines Kunden**

Wir haben umfassende und sehr aktuelle Artikel in der renommierten Zeitschrift iX gefunden.

Die Artikel geben dem Modewort CRM etwas Farbe. Man lernt, dass CRM nicht einfach eine Technik oder eine Sammlung von Funktionen ist, sondern eine Unternehmensphilosophie. Man muss, um ein umfangreiches CRM einzuführen, weit mehr machen, als es in einen Shop zu integrieren.

Heinz Heise Verlag, 30604 Hannover

## 2.1.4 Datenbanken und im speziellen MySQL

### MySQL in 21 Tagen

Markt + Technik Verlag München

Mark Maslakowski

2001

580 Seiten

ISBN 3-8272-5850-2

Dies war unser 'Hauptbuch' im Bereich Datenbanken. Man findet auch knifflige Aspekte von MySQL hier drin. Vor allem für die MySQL Konfiguration sehr zu empfehlen.

### SQL, der Schlüssel zu relationalen Datenbanken

Rowohlt Taschenbuch Verlag GmbH Reineck bei Hamburg

Gregor Kuhlmann, Friedrich Müllmerstadt

Mai 1999

318 Seiten

ISBN 3-499-60063-3

Dieses Buch half uns vor allem bei schwierigen Queries, aber auch bei Fragen, wie man gewisse Datenbank-Aktionen überhaupt erst angehen soll. Sehr zu empfehlen.

### Datenbanken – Vorlesung an der Zürcher Hochschule Winterthur

Dr. Hans-Walter Buff, zweites Jahr IT

Der Stoff half uns beim Entity Relationship Design, aber auch bei den Create-, Insert- und Del-SQL-Scripts.

## 2.1.5 HTML, JavaScript und Cascading Style Sheets

### HTML 4 – Die Sprache des Web

dpunkt-Verlag Heidelberg

Robert Tolksdorf

3. Auflage 1997

308 Seiten

ISBN 3-920993-77-2

Bietet einen tiefgehenden Einblick ins HTML, leider schon etwas in die Jahre gekommen.

### Jetzt lerne ich HTML

Markt + Technik Verlag München

Harald Taglinger

1997

290 Seiten

ISBN 3-8272-5305-5

Gutes Nachschlagewerk (Farbtabelle, ...) immer wieder praktisch für 'wie ging das jetzt schon wieder?'-Momente. Leider auch schon etwas in die Jahre gekommen.

### Das grosse Buch JavaScript

DATA BECKER GmbH & Co KG Düsseldorf

Rainer Kolbeck

1. Auflage 1997

422 Seiten

ISBN 3-8158-1321-2

Dieses Buch ist unbrauchbar (zumindest wenn man HTML schon kennt und den Netscape Javascript Debugger nicht erklärt braucht...)

### JavaScript

verlag moderne industrie, 53227 Bonn

Michael Seeboerger-Weichselbaum

1. Auflage 2001

767 Seiten

ISBN 3-8266-8106-1

Wir konnten dieses Buch sehr gut als Nachschlagewerk gebrauchen, da es eine umfassende und logisch aufgebaute JavaScript Befehlsreferenz beinhaltet.

Wir können es nur weiter empfehlen, das Preis- / Leistungsverhältnis stimmt.

### Einführung in JavaScript

<http://atlas.ee.fhm.edu/mm-cd/skript/javascript/>

Eine wirklich gelungene Einführung in JavaScript. Vermittelt die Grundlagen von JavaScript in nur 20 Minuten.

### HTML-Referenz

<http://www.teamone.de/selfhtml/>

Selfhtml ist und bleibt das beste Nachschlagewerk in Sachen HTML. Sogar für JavaScript konnten wir einige Informationen von dieser Adresse gebrauchen. Sehr zu empfehlen.

## 2.1.6 Perl

### Einführung in die Perl-Programmierung

<http://userpage.fu-berlin.de/~corff/perl/perl-kurs.html>

Dies ist ein guter Einstieg und trotzdem eine sehr umfangreiche Page. Wir haben vor allem das Kapitel 2, Grundlegende Elemente der Sprache, verwendet.

### Perl – Ein 25 Minuten Crashkurs!

[http://www.opensecure.de/PerlansB/perl\\_crashkurs.htm](http://www.opensecure.de/PerlansB/perl_crashkurs.htm)

Eine nette Einführung. Auf nur 20 Seiten hat man so ziemlich alle Sprachelemente, die man zu Beginn braucht, aufgelistet.

**Reguläre Ausdrücke in Perl**

<http://www.oreilly.de/catalog/regexger/chapter/kap-2d.html>

<http://www.heise.de/ix/artikel/1998/11/178/01.shtml>

Details zur Muster Erkennung mit Regulären Ausdrücken mit Perl.

**2.2 Software Versionen****2.2.1 Auf dem Server eingesetzte Software**

SuSE Linux 7.1

Kernel: 2.4.0-4GB (i686)

Apache

Version 1.3.20 (Unix) (LAMPS Installation)

MySQL

Version 3.23.35

PHP4

Version: 4.0.6

phpMyAdmin

Version 2.2.0

Einzige Linux-Distribution, die zu Projektbeginn schon den Kernel 2.4.0 beinhaltet.

In SuSE Linux Distribution enthalten, genügend aktuell.

Die zu Projektbeginn aktuellste MySQL Datenbank Version aus dem Internet.

Diese damals aktuellste Version von PHP4 war glücklicherweise schon in der SuSE Distribution enthalten.

Datenbank Administrationstool mit Webinterface. Dieses Programm brauchten wir, um SQL-Queries auszutesten und Änderungen in der Datenbank manuell vorzunehmen.

**2.2.2 Auf den Arbeitsstationen eingesetzte Software**

Windows 2000 Professionell

Komfortable Entwicklungsoberfläche

SP1

Apache für Windows

Zu Testzwecken eingerichtet

Version 1.3.19

PHP4

Zu Testzwecken eingerichtet

Version 4.04pl1

PHPed

Sehr nützlicher PHP-Editor mit integriertem FTP-Tool, welches direktes Arbeiten auf dem Server ermöglicht. Leider noch etwas Beta. FTP-Client

Version 2.96.1.2

CUTE FTP

Version 4.2.3 Build 2.14.1

Putty

Telnet / SSH Client

Version 0.5.1

## 2.3 Projekttagbuch

Das Projekttagbuch wurde direkt während der Diplomarbeit jeweils am Ende jedes Arbeitstages weitergeführt. Es ist ziemlich detailliert, erhebt jedoch keinen Anspruch auf Vollständigkeit und soll vor allem den Ablauf des Projekts dokumentieren.

### Mo, 10. September 01

- User pafei auf Webserver erstellt
- Drucker installiert
- PGP-installiert
- Analyse Feedbacks und E-Mails
- Erstellung ToDo-List mit Festlegung der Prioritäten und grober Aufwandabschätzung

### Di, 11. September 01

- phpMyAdmin installiert
- htaccess für phpMyAdmin eingerichtet
- Account bei SourceForge eingerichtet und aktiviert (José Fontanil)
- Apache 1.3.20 installiert
- erfolglose Installation von PHP 4.0.6

### Mi, 12. September 01

- Namen für Shop gesucht und gefunden: PhPepperShop
- Apache 1.3.20 konfiguriert (httpd.conf)
- Projekt PhPepperShop auf SourceForge eingereicht (geht ca. 72 Stunden bis Aktivierung)
- Von Hand installiert und konfiguriert:
  - PHP 4.0.6
  - OpenSSL 0.9.6b / ModSSL 2.8.4
  - Zlib-Library Version 1.1.3
  - IJG JPEG Software-Version 6b
  - Freetype Vers. 1.3.1
  - GD-Library 1.8.4
- Domain phpeppershop.com registriert (www.domainbank.net)
- Webserver für phpeppershop eingerichtet und aufgeschaltet (die Kosten werden momentan von Glarotech Informatikdienstleistungen getragen)

### Do, 13. September 01

- Projektsitzung
- Projekt auf SourceForge eröffnet
- User glanzret auf SourceForge erstellt
- Einarbeitung in CVS
- PHP mit PDF-Unterstützung nochmals kompiliert

### Fr, 14. September 01

- Internet von der ZHW die meiste Zeit nicht oder nur sehr langsam erreichbar -> viel warten!
- provisorische Website <http://www.phpeppershop.com> erstellt
- kleines Logo kreiert
- Mailinglisten auf SourceForge erstellt (devel, users, announce) und Test

- Newsforum auf SourceForge eingerichtet und Test
- Homepage phpeppershop.sourceforge.net -> automatischer redirect auf www.phpeppershop.com

### **Mo, 17. September 01**

- phpeppershop.com bei folgenden Suchmaschinen eingetragen: google.com, altavista.com, yahoo.de, search.ch
- Bericht „Webshop Framework“ im Internet Professionell 10/2001 (S.41)
- Einarbeitung in Perl
- Softlinks per Shell-Skript erstellen
- Script für Erstellung der Datenbank- und Connection-Skripts (1 oder 2 User, User erstellen oder schon erstellt, Shopname, Passwörter, User/Admin)
- Zeitplanung erstellt
- Einteilungen der Arbeiten nach Klassen
- 10'000 Hits auf der Webpage vom „Webshop Framework“ erreicht!

### **Di, 18. September 01**

- Information für Leser der Zeitschrift „Internet Professionell“ auf Webserver veröffentlicht
- Verbindung zum Internet die meiste Zeit nicht möglich!!!
- Konzept Darstellung von Unterkategorien im linken Frame erstellt
- Kategorienmanagement Bedienkonzept entwickelt
- Fertigstellung Perl/Shell-Installationsscript (config.pl, config\_part4.sh)

### **Mi, 19. September 01**

- Kurze Einarbeitung Java Script (mouseover, down, out, gif)
- Java Skript Template erstellt, welches dann einfach in Shop integriert werden kann (dynamische Buttonsets werden damit möglich)
- Scrollen im Topframe unterbunden (CSS-File & Frameset)
- Charset 8859-1 im Mailversand ergänzt
- Expiring-Datum für Cache auf 1.1.1971 korrigiert
- PLZ nun auch alphanumerisch (USA) möglich
- LAMPS-Dokumentation erstellt
- Eingabesicherheit im Installationsscript erhöht
- Deinstallationsscript erstellt

### **Do, 20. September 01**

- Konzept Layoutmanagement erstellt
- Dokumentation Installations/Deinstallationsscript
- yellowpay Anwendungskonzept angeschaut -> Implementation sollte kein Problem sein. Preis an Orbit abklären!
- Bugs in Installationsscript behoben
- Beginn mit Layoutmanagement
- Dokumentation „Einrichten eines Projekts auf SourceForge“

### **Fr, 21. September 01**

- Dokumentation „Einrichten eines Projekts auf SourceForge“
- Layoutmanagement
- Parser für CSS-File-Handling entwickelt

- CSS-Template mit Platzhalter-Tags erstellt

**Mo, 24. September 01**

- Artikel-Update Funktion neu programmiert
- CSS-Layout Management

**Di, 25. September 01**

- Gleicher Artikel mit verschiedenen Varianten und Optionen in Warenkorb
- CSS-Layout Management

**Mi, 26. September 01**

- Ganzer Tag Orbit/Comdex
- Besuch Case Study „Adori Webshop“

**Do, 27. September 01**

- CSS-Layout Management
- Anpassungen Installationsscript (MySQL-Fehler abfangen, Shopnamen Fehler)
- Unterkategorien implementieren

**Fr, 28. September 01**

- CSS-Layout Management
- Unterkategorien implementieren

**Mo, 1. Oktober 01**

- ZHW-LAN nicht verfügbar
- PC (Glanzmann) defekt
- CSS-Layout Management in Shop integrieren

**Di, 2. Oktober 01**

- Anpassung Layoutmanagement an Netscape 4.7
- Root-Problem gelöst!
- Unterkategorien fertig implementiert
- Beginn Artikel in mehrere Kategorien
- Artikelsuche um Unterkategorienfähigkeit und Mehrfachkategorien erweitert. Bilder können wahlweise angezeigt werden.

**Mi, 3. Oktober 01**

- Artikel in mehrere Kategorien
- Bugfixes
- Überarbeitung Admin-Layout und Anpassung Admin-CSS

**Do, 4. Oktober 01**

- Fertigstellung funktionaler Teil Mehrfachkategorien
- Fertigstellung darstellender Teil: Artikel in mehrere Kategorien

**Fr, 5. Oktober 01**

- Fertigstellung Artikelhandling
- Kategorienmanagement Funktionalität
- Anpassung Artikelupdate-Funktion
- Delete-Funktion

**Mo, 8. Oktober 01**

- Möglichkeit zu wählen, wie viele Options und Variantenfelder zur Erfassung und Bearbeitung eines Artikels angezeigt werden (Mindestwert und Anzahl leere Felder)
- Shop-Administrationsmodus überarbeitet und der aktuellen Shopfunktionalität angepasst
- Eingabe-Plausibilitätsprüfung im Shop-Administrationsmodus per Java-Skript.
- Versandkostensettings

**Di, 9. Oktober 01**

- Versandkostensettings fertig (Formular muss nochmals leicht überarbeitet werden)
- Kategorienmanagement GUI (grosse Probleme Netscape 4.xx und Java-Script)

**Mi, 10. Oktober 01**

- Kategorienmanagement

**Do, 11. Oktober 01**

- Kategorienmanagement Sonderzeichen fähig gemacht
- Backup Shop angeschaut
- Kostenlose Artikel implementiert

**Fr, 12. Oktober 01**

- Datenbank auf CRM vorbereitet
- Analyse CRM und Erstellung CRM-Objekte
- Shop Sonderzeichen fähig gemacht

**Mo, 15. Oktober 01**

- CRM-Shop benutzerseitig implementiert

**Di, 16. Oktober 01**

- Dynamische Adresseingabefelder
- Suche erweitert um Anzahl Suchresultate
- Suche mit mehreren Begriffen möglich (wie bei [www.google.com](http://www.google.com))

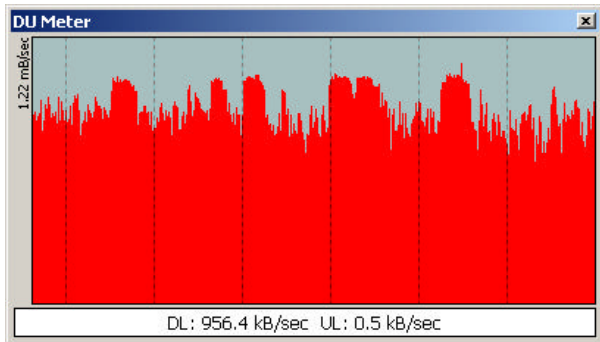
**Mi, 17. Oktober 01**

- GNU-PG Versuch 1
- Zahlungsmöglichkeiten / CRM
- Kreditkarten Datenbank , objektmässig, Administrator-mässig implementiert
- Versandkostensettings Update

**Do, 18. Oktober 01**

- Shopbuttons Upload fertiggestellt
- Nachnahmegebühr/Vorauskauf im Shop integriert.
- Update nicht möglich, da Netz nicht verfügbar

- Ganzer Nachmittag Mailserver nicht verfügbar!
- Totalausfall LAN, nichts geht mehr! Stundenlang!



- Kreditkarten Zahlungsworkflow-Analyse
- GNU-PG Versuch 2
- Artikelhandling überarbeitet

### **Fr, 19. Oktober 01**

- Eigene Pop-Up und Hilfe-Funktionen für Administrationsmodus, die mit dem CSS-File des Administrationsmodus arbeiten.
- Kreditkarten Zahlungsworkflow beendet (Sa.)
- Buttons-Upload um Kategoriebuttons erweitert
- Admin-Buttons einheitlich erstellt (Sa.)
- Testergebnisse abgearbeitet und Bugs behoben (Sa.)
- Bestellmanagement an CRM anpassen
- Raumreservierung für Präsentationen DA (½ Tag)

### **Mo, 22. Oktober 01**

- Festlegung Layout für Dokumentation
- Beginn Dokumentation

### **Di, 23. Oktober 01 – Fr, 26 Oktober**

- Dokumentation Diplomarbeit

### **Mo, 29. Oktober 01**

- Abgabe Diplomarbeit



## 2.4 GPL – GNU General Public Licence

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation herausgegeben, weitergeben und/oder modifizieren, entweder unter Version 2 der Lizenz oder (wenn Sie es wünschen) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE JEDE GEWÄHRLEISTUNG – sogar ohne die implizite Gewährleistung der MARKTREIFE oder der EIGNUNG FÜR EINEN BESTIMMTEN ZWECK.

Details finden Sie in der GNU General Public License. Sie sollten eine Kopie der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

### 2.4.1 Deutsche Fassung der GPL (Auszug)

Übersetzt von Katja Lachmann Übersetzungen ([na194@fim.uni-erlangen.de](mailto:na194@fim.uni-erlangen.de)), modifiziert von Peter Gerwinski ([peter.gerwinski@uni-essen.de](mailto:peter.gerwinski@uni-essen.de)) und Jörg Krause ([joerg@krause.net](mailto:joerg@krause.net)). Es handelt sich nicht um eine offizielle oder im rechtlichen Sinne anerkannte Übersetzung.

Die Free Software Foundation (FSF) ist nicht der Herausgeber dieser Übersetzung, und sie hat diese Übersetzung auch nicht als rechtskräftigen Ersatz für die Original GNU-GPL anerkannt. Da die Übersetzung nicht sorgfältig von Anwälten überprüft wurde, können die Übersetzer nicht garantieren, dass die Übersetzung die rechtlichen Aussagen der GNU-GPL exakt wiedergibt. Wenn Sie sichergehen wollen, dass von Ihnen geplante Aktivitäten im Sinne der GNU-GPL gestattet sind, halten Sie sich bitte an die englischsprachige Originalversion. Die Free Software Foundation möchte Sie darum bitten, diese Übersetzung nicht als offizielle Lizenzbedingungen für von Ihnen geschriebene Programme zu verwenden. Bitte benutzen Sie hierfür stattdessen die von der Free Software Foundation herausgegebene englischsprachige Originalversion. Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung

**§0** Diese Lizenz gilt für jedes Programm und jedes andere Werk, in dem ein entsprechender Vermerk des Copyright-Inhabers darauf hinweist, dass das Werk unter den Bestimmungen dieser General Public License verbreitet werden darf. Im Folgenden wird jedes derartige Programm oder Werk als »das Programm« bezeichnet; die Formulierung »auf dem Programm basierendes Werk« bezeichnet das Programm sowie jegliche Bearbeitung des Programms im urheberrechtlichen Sinne, also ein Werk, welches das Programm, auch auszugsweise, sei es unverändert oder verändert oder in eine andere Sprache übersetzt, enthält. Andere Handlungen als Vervielfältigung, Verbreitung und Bearbeitung werden von dieser Lizenz nicht berührt; sie fallen nicht in ihren Anwendungsbereich. Der Vorgang der Ausführung des Programms wird nicht eingeschränkt und die Ausgaben des Programms unterliegen dieser Lizenz nur, wenn der Inhalt ein auf dem Programm basierendes Werk darstellt (unabhängig davon, dass die Ausgabe durch die Ausführung des Programmes erfolgte). Ob dies zutrifft, hängt von den Funktionen des Programms ab.

**§1** Sie dürfen auf beliebigen Medien unveränderte Kopien des Quelltextes des Programms, wie sie ihn erhalten haben, anfertigen und verbreiten. Voraussetzung hierfür ist, dass Sie mit jeder Kopie einen entsprechenden Copyright-Vermerk sowie einen Haftungsausschluss veröffentlichen, alle Vermerke, die sich auf diese Lizenz und das Fehlen einer Garantie beziehen, unverändert lassen und des Weiteren allen anderen Empfängern des Programms zusammen mit dem Programm eine Kopie dieser Lizenz zukommen lassen. Sie dürfen für den eigentlichen Kopiervorgang eine Gebühr verlangen. Wenn Sie es wünschen, dürfen Sie auch gegen Entgelt eine Garantie für das Programm anbieten.

**§2** Sie dürfen Ihre Kopien des Programms oder eines Teils davon verändern, wodurch ein auf dem Programm basierendes Werk entsteht; Sie dürfen derartige Bearbeitungen unter den Bestimmungen von Paragraph 1 vervielfältigen und verbreiten, vorausgesetzt, dass zusätzlich alle folgenden Bedingungen erfüllt werden:

- (a) Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.
- (b) Sie müssen dafür sorgen, dass jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.
- (c) Wenn das veränderte Programm normalerweise bei der Ausführung interaktiv Kommandos einliest, müssen Sie dafür sorgen, dass es, wenn es auf dem üblichsten Wege für solche interaktive Nutzung gestartet wird, eine Meldung ausgibt oder ausdrückt, die einen geeigneten Copyright-Vermerk enthält sowie einen Hinweis, dass es keine Gewährleistung gibt (oder anderenfalls, dass Sie Garantie leisten), und dass die Benutzer das Programm unter diesen Bedingungen weiter verbreiten dürfen. Auch muss der Benutzer darauf hingewiesen werden, wie er eine Kopie dieser Lizenz ansehen kann. (Ausnahme: Wenn das Programm selbst interaktiv arbeitet, aber normalerweise keine derartige Meldung ausgibt, muss Ihr auf dem Programm basierendes Werk auch keine solche Meldung ausgeben.) Diese Anforderungen betreffen das veränderte Werk als Ganzes. Wenn identifizierbare Abschnitte des Werkes nicht von dem Programm abgeleitet sind und vernünftigerweise selbst als unabhängige und eigenständige Werke betrachtet werden können, dann erstrecken sich diese Lizenz und ihre Bedingungen nicht auf diese Abschnitte, wenn sie als eigenständige Werke verbreitet werden. Wenn Sie jedoch dieselben Abschnitte als Teil eines Ganzen

verbreiten, das ein auf dem Programm basierendes Werk darstellt, dann muss die Verbreitung des Ganzen nach den Bedingungen dieser Lizenz erfolgen, deren Bedingungen für weitere Lizenznehmer somit auf die Gesamtheit ausgedehnt werden – und damit auf jeden einzelnen Teil, unabhängig vom jeweiligen Autor. Somit ist es nicht die Absicht dieses Abschnittes, Rechte für Werke in Anspruch zu nehmen oder zu beschneiden, die komplett von Ihnen geschrieben wurden; vielmehr ist es die Absicht, die Rechte zur Kontrolle der Verbreitung von Werken, die auf dem Programm basieren oder unter seiner auszugsweisen Verwendung zusammengestellt worden sind, auszuüben. Ferner bringt ein einfaches Zusammenstellen eines anderen Werkes, das nicht auf dem Programm basiert, zusammen mit dem Programm oder einem auf dem Programm basierenden Werk auf ein- und demselben Speicher- oder Betriebsmedium das andere Werk nicht in den Anwendungsbereich dieser Lizenz.

**§3** Sie dürfen das Programm (oder ein darauf basierendes Werk gemäß Paragraph 2) als Objectcode oder in ausführbarer Form unter den Bedingungen von Paragraph 1 und 2 vervielfältigen und verbreiten – vorausgesetzt, dass Sie außerdem eine der folgenden Leistungen erbringen:

(a) Liefern Sie das Programm zusammen mit dem vollständigen zugehörigen maschinenlesbaren Quelltext auf einem für den Datenaustausch üblichen Medium aus, wobei die Verteilung unter den Bedingungen der Paragraphen 1 und 2 erfolgen muss. Oder:

(b) Liefern Sie das Programm zusammen mit einem mindestens drei Jahre lang gültigen schriftlichen Angebot aus, jedem Dritten eine vollständige maschinenlesbare Kopie des Quelltextes zur Verfügung zu stellen – zu nicht höheren Kosten als denen, die durch den physikalischen Kopiervorgang anfallen –, wobei der Quelltext unter den Bedingungen der Paragraphen 1 und 2 auf einem für den Datenaustausch üblichen Medium weitergegeben wird. Oder:

(c) Liefern Sie das Programm zusammen mit dem schriftlichen Angebot der Zurverfügungstellung des Quelltextes aus, das Sie selbst erhalten haben. (Diese Alternative ist nur für nicht-kommerzielle Verbreitung zulässig und nur, wenn Sie das Programm als Objectcode oder in ausführbarer Form mit einem entsprechenden Angebot gemäß Absatz b erhalten haben.) Unter dem Quelltext eines Werkes wird diejenige Form des Werkes verstanden, die für Bearbeitungen vorzugsweise verwendet wird. Für ein ausführbares Programm bedeutet »der komplette Quelltext«: Der Quelltext aller im Programm enthaltenen Module einschließlich aller zugehörigen Modulschnittstellen-Definitionsdateien sowie der zur Compilation und Installation verwendeten Skripte. Als besondere Ausnahme jedoch braucht der verteilte Quelltext nichts von dem zu enthalten, was üblicherweise (entweder als Quelltext oder in binärer Form) zusammen mit den Hauptkomponenten des Betriebssystems (Kernel, Compiler usw.) geliefert wird, unter dem das Programm läuft – es sei denn, diese Komponente selbst gehört zum ausführbaren Programm. Wenn die Verbreitung eines ausführbaren Programms oder des Objectcodes dadurch erfolgt, dass der Kopierzugriff auf eine dafür vorgesehene Stelle gewährt wird, so gilt die Gewährung eines gleichwertigen Zugriffs auf den Quelltext als Verbreitung des Quelltextes, auch wenn Dritte nicht dazu gezwungen sind, den Quelltext zusammen mit dem Objectcode zu kopieren.

**§4** Sie dürfen das Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten, sofern es nicht durch diese Lizenz ausdrücklich gestattet ist. Jeder anderweitige Versuch der Vervielfältigung, Modifizierung, Weiterlizenzierung und Verbreitung ist nichtig und beendet automatisch Ihre Rechte unter dieser Lizenz. Jedoch werden die Lizenzen Dritter, die von Ihnen Kopien oder Rechte unter dieser Lizenz erhalten haben, nicht beendet, solange diese die Lizenz voll anerkennen und befolgen.

**§5** Sie sind nicht verpflichtet, diese Lizenz anzunehmen, da Sie sie nicht unterzeichnet haben. Jedoch gibt Ihnen nichts anderes die Erlaubnis, das Programm oder von ihm abgeleitete Werke zu verändern oder zu verbreiten. Diese Handlungen sind gesetzlich verboten, wenn Sie diese Lizenz nicht anerkennen. Indem Sie das Programm (oder ein darauf basierendes Werk) verändern oder verbreiten, erklären Sie Ihr Einverständnis mit dieser Lizenz und mit allen ihren Bedingungen bezüglich der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Werkes.

**§6** Jedesmal, wenn Sie das Programm (oder ein auf dem Programm basierendes Werk) weitergeben, erhält der Empfänger automatisch vom ursprünglichen Lizenzgeber die Lizenz, das Programm entsprechend den hier festgelegten Bestimmungen zu vervielfältigen, zu verbreiten und zu verändern. Sie dürfen keine weiteren Einschränkungen der Durchsetzung der hierin zugestandenen Rechte des Empfängers vornehmen. Sie sind nicht dafür verantwortlich, die Einhaltung dieser Lizenz durch Dritte durchzusetzen.

**§7** Sollten Ihnen infolge eines Gerichtsurteils, des Vorwurfs einer Patentverletzung oder aus einem anderen Grunde (nicht auf Patentfragen begrenzt) Be-.Copyright und Lizenzhinweise 9 Bedingungen (durch Gerichtsbeschluss, Vergleich oder anderweitig auferlegt werden, die den Bedingungen dieser Lizenz widersprechen, so befreien Sie diese Umstände nicht von den Bestimmungen dieser Lizenz. Wenn es Ihnen nicht möglich ist, das Programm unter gleichzeitiger Beachtung der Bedingungen in dieser Lizenz und Ihrer anderweitigen Verpflichtungen zu verbreiten, dann dürfen Sie als Folge das Programm überhaupt nicht verbreiten. Wenn zum Beispiel ein Patent nicht die gebührenfreie Weiterverbreitung des Programms durch diejenigen erlaubt, die das Programm direkt oder indirekt von Ihnen erhalten haben, dann besteht der einzige Weg, sowohl das Patentrecht als auch diese Lizenz zu befolgen, darin, ganz auf die Verbreitung des Programms zu verzichten. Sollte sich ein Teil dieses Paragraphen als ungültig oder unter bestimmten Umständen nicht durchsetzbar erweisen, so soll dieser Paragraph seinem Sinne nach angewandt werden; im Übrigen soll dieser Paragraph als Ganzes gelten. Zweck dieses Paragraphen ist nicht, Sie dazu zu bringen, irgendwelche Patente oder andere Eigentumsansprüche zu verletzen oder die Gültigkeit solcher Ansprüche zu bestreiten; dieser Paragraph hat einzig den Zweck, die Integrität des Vertriebssystems der freien Software zu schützen, das durch die Praxis öffentlicher Lizenzen verwirklicht wird. Viele Leute haben großzügige Beiträge zu dem großen Angebot der mit diesem System verbreiteten Software im Vertrauen auf die konsistente Anwendung dieses Systems geleistet; es liegt am Autor/Geber, zu entscheiden, ob er die Software mittels irgendeines anderen Systems verbreiten will; ein Lizenznehmer hat auf diese Entscheidung keinen Einfluss. Dieser Paragraph ist dazu gedacht, deutlich klarzustellen, was als Konsequenz aus dem Rest dieser Lizenz betrachtet wird.

**§8** Wenn die Verbreitung oder die Benutzung des Programms in bestimmten Staaten entweder durch Patente oder durch urhe-

berrechtlich geschützte Schnittstellen eingeschränkt ist, kann der Urheberrechtsinhaber, der das Programm unter diese Lizenz gestellt hat, eine explizite geographische Begrenzung der Verbreitung angeben, in der diese Staaten ausgeschlossen werden, so dass die Verbreitung nur innerhalb und zwischen den Staaten erlaubt ist, die nicht ausgeschlossen sind. In einem solchen Fall beinhaltet diese Lizenz die Beschränkung, als wäre sie in diesem Text niedergeschrieben.

**§9** Die Free Software Foundation kann von Zeit zu Zeit überarbeitete und/oder neue Versionen der General Public License veröffentlichen. Solche neuen Versionen werden vom Grundprinzip her der gegenwärtigen entsprechen, können aber im Detail abweichen, um neuen Problemen und Anforderungen gerecht zu werden. Jede Version dieser Lizenz hat eine eindeutige Versionsnummer. Wenn in einem Programm angegeben wird, dass es dieser Lizenz in einer bestimmten Versionsnummer oder »jeder späteren Version« unterliegt, so haben Sie die Wahl, entweder den Bestimmungen der genannten Version zu folgen oder denen jeder beliebigen späteren Version, die von der Free Software Foundation veröffentlicht wurde. Wenn das Programm keine Versionsnummer angibt, können Sie eine beliebige Version wählen, die je von der Free Software Foundation veröffentlicht wurde.

**§10** Wenn Sie den Wunsch haben, Teile des Programms in anderen freien Programmen zu verwenden, deren Bedingungen für die Verbreitung anders sind, schreiben Sie an den Autor, um ihn um die Erlaubnis zu bitten. Für Software, die unter dem Copyright der Free Software Foundation steht, schreiben Sie an die Free Software Foundation; wir machen zu diesem Zweck gelegentlich Ausnahmen. Unsere Entscheidung wird von den beiden Zielen geleitet werden, zum einen den freien Status aller von unserer freien Software abgeleiteten Werke zu erhalten und zum anderen das gemeinschaftliche Nutzen und Wiederverwenden von Software im Allgemeinen zu fördern.

**§11** Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, »wie es ist«, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich – aber nicht begrenzt auf – Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.

**§12** In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Programms oder der Unbenutzbarkeit des Programms folgen (einschließlich – aber nicht beschränkt auf – Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Programms, mit irgendeinem anderen Programm zusammenzuarbeiten), selbst wenn ein Copyright-Inhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.