

MARTIN DUSCHEK, 67664, 16MI1-B

EVOLUTION VON CODE BEI MAJOR-RELEASES VON
PROGRAMMIERSPRACHEN

EVOLUTION VON CODE BEI MAJOR-RELEASES
VON PROGRAMMIERSPRACHEN

MARTIN DUSCHEK, 67664, 16MI1-B

HITWK

Hochschule für Technik,
Wirtschaft und Kultur Leipzig

am Beispiel von PHP

Juli 2019

Martin Duschek, 67664, 16MI1-B: *Evolution von Code bei Major-Releases von Programmiersprachen*, am Beispiel von PHP, © Juli 2019

INHALTSVERZEICHNIS

I EINLEITUNG

- 1 EINFÜHRUNG TEST2 CHECK SYNC 2
 - 1.1 Firmenportrait 2
 - 1.2 Entwicklungsumgebung & verwendete Technologien 2
 - 1.3 Organisation 3

II PRAKTIKUMSBERICHT

- 2 TÄTIGKEITEN 5
 - 2.1 Migration des Shopsystems zu php7 5
 - 2.2 Anpassungen im Versand-Workflow 5
 - 2.3 Lokalisation des Shopsystems 7
 - 2.4 Verschiedene Tätigkeiten 10
 - 2.4.1 Austausch des API-Keys im Zahlungsmodul 10
 - 2.4.2 Bugfixes in der Produktpflegetabelle 10
 - 2.4.3 Projektmanagement Produktpflegetabelle 10
 - 2.4.4 Systemadministration 11
- 3 PERSÖNLICHE EINSCHÄTZUNG 12

LISTINGS

Listing 2.1	Versandmodul Beispiel	6
Listing 2.2	Sprachdefinition mittels Konstanten	8
Listing 2.3	Sprachdefinition mittels Arrays	8
Listing 2.4	Extrahieren der Übersetzungen	8
Listing 2.5	Länderübersetzungen	9
Listing 2.6	API-Aufruf geonames.org	9

Teil I

EINLEITUNG

EINFÜHRUNG TEST₂ CHECK SYNC

Dieser Bericht enthält eine Zusammenfassung meiner Tätigkeiten im Rahmen des Pflichtmoduls *Praxisprojekt* im Bachelorstudiengang Medieninformatik an der HTWK Leipzig.

Das Praxisprojekt fand zwischen dem 17. März 2019 und dem 07. Juni 2019 in Kooperation mit dem Unternehmen

TICKETS₇₅
Büsumer Straße 40–44
24768 Rendsburg

am Standort Leipzig statt.

1.1 FIRMENPORTRAIT

Tickets₇₅ ist eine 2006 gegründete Kartenagentur, die sich auf die Vermittlung von begehrten Tickets für Konzerte, Sportveranstaltungen und Unterhaltungsshows spezialisiert hat. Das Unternehmen beschäftigt etwa 30 Mitarbeiter an zwei Standorten. Am Standort Leipzig befinden sich die Abteilungen *Produktmanagement & Verkauf*, sowie *IT & Online-Marketing*. In Rendsburg befinden sich die Versandabteilung sowie der offizielle Firmensitz. Die Entwicklung des Webshops wird zusätzlich durch zwei externe Entwickler unterstützt.

1.2 ENTWICKLUNGSUMGEBUNG & VERWENDETE TECHNOLOGIEN

Der Webshop des Unternehmens basiert auf dem Shopsystem *Gambio*, einer Weiterentwicklung des Open-Source-Projekts *xt:Commerce*. Die eingesetzte Version ist stark an die eigenen Bedürfnisse angepasst und muss deshalb eigenständig weiterentwickelt werden. Basis des Webshops bildet *php*, dem eine Datenbank auf Basis von *MySQL* zur Seite steht. Das Templating von Inhaltsseiten wird durch das Framework *Smarty* abgewickelt. Die lokale Entwicklung kann auf zweierlei Arten stattfinden. Eine Möglichkeit besteht in der Verwendung des Softwarepakets *XAMPP*, welches den Webserver *Apache* und *MySQL* für Windows bereitstellt. Eine andere Möglichkeit besteht in der Verwendung von *Docker*, einer Software zur Containervirtualisierung. Der Vorteil dieser Praxis liegt in der vereinheitlichten Umgebung, die unabhängig von individuellen Systemeinstellungen ist. Das gesamte Projekt wird in *Bitbucket*, einem System zur Versionsverwaltung, vorgehalten und kann lokal durch *git* gesteuert werden. Als Entwicklungsumgebung wird *Visual Studio Code* verwendet, das durch Plugins

auf die eigenen Bedürfnisse angepasst werden kann. Dazu zählen Erweiterungen zur Autovervollständigung und zum Debuggen mittels *XDebug*.

1.3 ORGANISATION

Die Entwicklung am Webshop ist in zweiwöchige Sprints aufgeteilt. Inhalte des Sprints werden zu Beginn in einem *Planning Meeting* besprochen und festgelegt. Dabei wird für neu hinzugekommene Tickets auch der Aufwand abgeschätzt. Der Sprint wird durch tägliche Scrum-Meetings unterstützt, in denen jeder Entwickler einen kurzen Überblick über seine Arbeit des letzten, sowie des kommenden Tages gibt. Gleichzeitig dient dieses Meeting dazu, etwaige Fragen und offene Probleme zu klären. Während eines Sprints erfolgen ein bis zwei Releases fertiggestellter Aufgaben, so dass wöchentlich eine aktualisierte Version produktiv eingesetzt werden kann. Den Abschluss eines Sprints bildet die Retrospektive. Dabei wird von jedem beteiligten Kritik - sowohl positiv, als auch negativ - geäußert und gemeinsam besprochen, wie aufgetauchten Problemen entgegengewirkt werden kann.

Teil II

PRAKTIKUMSBERICHT

TÄTIGKEITEN

Im Folgenden werden die durchgeführten Tätigkeiten beschrieben, sowie dazugehörige Überlegungen dargelegt.

2.1 MIGRATION DES SHOPSYSTEMS ZU PHP7

Dieses Projekt wurde während meiner Werkstudententätigkeit im Unternehmen begonnen und größtenteils fertiggestellt. Da nun der Rollout bevorstand, wurden durch mich hauptsächlich Tests und Bugfixes durchgeführt. Diese bezogen sich zum Großteil auf die Hauptfunktionalitäten des Shops und des administrativen Bereichs, sowie auf den Checkout. Bugfixes wurden für das Session-Handling geschrieben, das durch die Umstellung des Datenbank-Handlers von *mysql* (Diese php-Erweiterung wird mit Version 7 nicht mehr unterstützt) auf *mysqli* noch einige Fehler enthielt. Diese fielen bei der Migration nicht direkt auf, da in der lokalen Entwicklungsumgebung nicht auf ein datenbankbasiertes, sondern auf dateibasiertes Session-Handling gesetzt wurde. Um den Entwicklungsprozess zu vereinfachen, wurde durch einen externen Entwickler ein Docker-Image bereitgestellt, welches durch mich so angepasst wurde, dass ein Debugging mittels der php-Erweiterung *XDebug* auch unter Windows ermöglicht wird. Diese Funktionalität wurde im internen Wiki dokumentiert, um anderen Entwicklern die Einrichtung der Entwicklungsumgebung zu erleichtern. Der Rollout wurde durch einen externen Dienstleister dahingehend vorbereitet, dass Server mit den entsprechenden Softwarepaketen zur Verfügung standen, sodass dort nur das Shopsystem aufgespielt werden musste, sowie der Traffic auf diese Server umgeleitet werden musste. Um möglichst wenige Kunden zu beeinträchtigen und genug Zeit für Tests und einen eventuellen Rollback zu haben, wurde der Rollout früh morgens durchgeführt und durch meinen Vorgesetzten und mich betreut. Beim anschließenden Testen fielen noch einige fehlerhafte Konfigurationen auf, so zum Beispiel eine falsch angegebene Datenbankverbindung. Seitdem diese anfänglichen Probleme behoben sind, läuft das Shopsystem stabil im Produktiveinsatz.

2.2 ANPASSUNGEN IM VERSAND-WORKFLOW

Dieses Projekt hat zum Ziel, den Umgang mit verschiedenen Versandarten vom Bestellen bis zum Versenden zu vereinfachen und gleichzeitig flexibler zu gestalten. Hierzu wurden neue Versandmodule für das Gambio-System erstellt. Diese bestehen hauptsächlich

aus SQL-Anweisungen, die bei der Installation ausgeführt werden und Parameter wie zum Beispiel die erlaubten Versandländer in die Datenbank schreiben. Ein Codebeispiel ist in [Listing 2.1](#) angeführt. Zusätzlich wurde ein neues Feld in die administrative Oberfläche der Versandmodule eingeführt, welches es ermöglicht, pro Versandzone bzw. Land eine individuelle Benennung der Versandart zu wählen.

Listing 2.1: Versandmodul Beispiel

```
function install() {
    xtc_db_query("insert into " . TABLE_CONFIGURATION . " (
        configuration_key, configuration_value,
        configuration_group_id, sort_order, set_function,
        date_added) VALUES ('MODULE_SHIPPING_STANDARDDE_STATUS', '
        True', '6', '0', 'gm_cfg_select_option(array(\'True\', \'
        False\'), ', now())");
    xtc_db_query("insert into " . TABLE_CONFIGURATION . " (
        configuration_key, configuration_value,
        configuration_group_id, sort_order, date_added) values ('
        MODULE_SHIPPING_STANDARDDE_HANDLING', '0', '6', '0', now())
    ");
    for ($i = 1; $i <= $this->num_standardDE; $i++) {
        xtc_db_query("insert into " . TABLE_CONFIGURATION . " (
            configuration_key, configuration_value,
            configuration_group_id, sort_order, date_added) values ('
            MODULE_SHIPPING_STANDARDDE_AFTERBUY_" . $i."', '
            Versicherter Standard-Versand', '6', '0', now())");
    }
}

function keys() {
    $keys = array('MODULE_SHIPPING_STANDARDDE_STATUS',
        'MODULE_SHIPPING_STANDARDDE_HANDLING',
        'MODULE_SHIPPING_STANDARDDE_ALLOWED',
        'MODULE_SHIPPING_STANDARDDE_FREEAMOUNT',
        'MODULE_SHIPPING_STANDARDDE_TAX_CLASS',
        'MODULE_SHIPPING_STANDARDDE_ZONE',
        'MODULE_SHIPPING_STANDARDDE_SORT_ORDER');

    for ($i = 1; $i <= $this->num_standardDE; $i++) {
        $keys[count($keys)] = 'MODULE_SHIPPING_STANDARDDE_COUNTRIES_'
            . $i;
        $keys[count($keys)] = 'MODULE_SHIPPING_STANDARDDE_COST_' . $i
            ;
        $keys[] = 'MODULE_SHIPPING_STANDARDDE_AFTERBUY_' . $i;
    }
    return $keys;
}
```

2.3 LOKALISATION DES SHOPSYSTEMS

Ziel dieses Projekts ist es, Kunden in Fremdmärkten eine simplere Navigation im Shop zu ermöglichen und die Suchmaschinenrelevanz in diesen Märkten zu erhöhen. Bisher sind nur deutsche und englische Sprachversionen des Shops verfügbar, die in einer ersten Umsetzung um zwei weitere Sprachen wichtiger Märkte erweitert werden sollen. Hierfür wurde zuerst eine Bestandsaufnahme der bisherigen Übersetzungen vorgenommen, bei der evaluiert wurde, welche Teile des Shops übersetzt werden müssen und welche Sprachdateien sich nicht mehr in Benutzung befinden. Von dieser Analyse ausgehend wurden aus diesen Dateien die für die Übersetzung wichtigen Bestandteile per *Regular Expression* extrahiert und in das *CSV-Format* überführt, welches von den meisten Übersetzungsbüros gefordert wird. Dieses Skript ist in [Listing 2.4](#) aufgeführt. Eine Herausforderung bestand darin, dass die Sprachdateien in unterschiedlichen Formaten vorlagen. *Gambio* arbeitet vorwiegend mit Konstanten, die nach dem Laden der Datei global verfügbar sind (Ein beispielhafter Ausschnitt ist in [Listing 2.2](#) angefügt), wohingegen die Templating-Engine Smarty bevorzugt mit, auf das jeweilige Template zugeschnittenen, Arrays arbeitet. Für die Übersetzungen wurden Angebote von zwei verschiedenen Übersetzungsbüros angefordert und an jedes ein Auftrag für die Übersetzung in eine Sprache vergeben. Diese Vorgehensweise sollte uns Erfahrungswerte im Umgang mit den zurückgelieferten Dateien, sowie der Übersetzungsqualität liefern. Die übersetzten Dateien lagen im CSV-Format vor und mussten wieder möglichst effizient in die ursprünglichen Formate überführt werden. Hierzu wurde ein Python-Skript geschrieben, welches eine gegebene Ordnerstruktur durchläuft und die Dateien je nach Verzeichnis in die entsprechende Form überträgt.

Eine zusätzliche Herausforderung waren Übersetzungen von Städten und Ländern, die aus Kostengründen nicht an die Übersetzungsbüros gegeben wurden. Um diese zu übersetzen wurden zwei unterschiedliche Skripte geschrieben. Das Skript für die Übersetzung der Länder [Listing 2.5](#) bedient sich einer freien Datenbank mit Übersetzungen aller Länder in unterschiedlichste Sprachen und Dialekte. Um die entsprechenden Übersetzungen zu generieren kann per Kommandozeile die gewünschte Sprache als Parameter angegeben werden (z.B. "he" für hebräisch). Die Ausgabe erfolgt in einer Datei, die direkt nutzbare SQL-Statements enthält, welche nach der Installation des Sprachpakets ausgeführt werden können.

Die Übersetzung der Städte funktioniert nach einem ähnlichen Prinzip, allerdings kommt hier der Open-Source-Webdienst *geonames.org* zum Einsatz, dessen API zu einem gegebenen Ort dessen lokale Bezeichnung liefern kann. Ein Beispiel hierfür ist der API-Aufruf in [Listing 2.6](#), der als Ergebnis der Abfrage nach der niederländischen Bezeichnung der Stadt Köln die Bezeichnung *Keulen* zurückgibt. Die

Ergebnisse müssen allerdings sorgfältig geprüft werden, da Ortsnamen keine eindeutigen Bezeichner sind.

Nach Tests mit mehreren Sprachen mussten noch einige Bugfixes geschrieben werden. Beispielsweise konnten einige Eingabemasken nur mit aufeinanderfolgenden IDs der Sprachen umgehen, an anderen Stellen wurden Masken schon mit drei Sprachen zu unübersichtlich und mussten überarbeitet werden.

Das Projekt ist noch nicht abgeschlossen und wird nach dem Praktikum fertiggestellt.

Listing 2.2: Sprachdefinition mittels Konstanten

```
define('GM_WISHLIST_NOTHING_CHECKED', 'Sie haben keine Artikel
    ausgewählt, die in den Warenkorb gelegt werden sollen!');
define('JS_ERROR_CONDITIONS_NOT_ACCEPTED_AGB', 'Sofern Sie unsere
    Allgemeinen Geschäftsbedingungen nicht akzeptieren,\n können
    wir Ihre Bestellung leider nicht entgegennehmen! \n\n');
define('JS_ERROR_CONDITIONS_NOT_ACCEPTED_WITHDRAWAL', 'Sofern Sie
    unser Widerrufsrecht nicht akzeptieren,\n können wir Ihre
    Bestellung leider nicht entgegennehmen! \n\n');
```

Listing 2.3: Sprachdefinition mittels Arrays

```
$t_language_text_section_content_array = array(
    'page_not_found' => 'Leider konnten wir die von Ihnen gewünschte
        Veranstaltung nicht finden.',
    'page_not_found_introtxt' => 'Die von Ihnen gewünschte Seite
        konnte nicht gefunden werden. Entweder ist die Veranstaltung
        nicht mehr verfügbar oder wurde auf eine andere Seite
        verschoben.',
    'try_following' => 'Dies könnte Ihnen helfen Ihre
        Wunschveranstaltung zu finden:',
    'navigate_to_frontpage' => 'Besuchen Sie unsere <a href="/">
        Startseite</a>',
    'browse_events' => 'Stöbern Sie in unseren <a href="
        upcoming_products.php">Veranstaltungen von A bis Z</a>',
    'use_search_functionality' => 'Nutzen Sie die Suchfunktion:',
    'navigate_to_categories' => 'Navigieren Sie zu einer der
        Kategorien',
    'navigate_to' => 'Navigieren Sie zu'
);
```

Listing 2.4: Extrahieren der Übersetzungen

```
foreach(glob("*.php") as $filename){
    $match = [];
    $langString = "";
    if (is_file($filename) && $filename != 'langExtract.php'){
        //Alle Definitionen von Konstanten
        if (preg_match_all("/define\\W*(\\W*('[^']*')\\W*('[^']*')/",
            file_get_contents($filename), $match)){
            break;
        }
    }
}
```

```

//Alle Definitionen innerhalb des Arrays
} else if (preg_match_all("/\W*'([^\']*)\W*'([^\']*)/",
    file_get_contents($filename), $match)) {
    break;
} else {
    print('Fehler in Datei ' . $filename);
}

for($i = 0; $i < sizeof($match[1]); $i++){
    $langString .= "\" . $match[1][$i] . "\";\" . \"\"\" .
        html_entity_decode($match[2][$i], ENT_HTML401 , '
        cp1252') . "\";\n";
}

$fp = fopen( "csv/" . $filename . ".csv", "w");
fwrite($fp, $langString);
}
}

```

Listing 2.5: Länderübersetzungen

```

$countries = $argv;
unset($countries['0']);
$csv = array_map('str_getcsv', file('IP2LOCATION-COUNTRY-
MULTILINGUAL.CSV'));
$fp = fopen('countries-lang-' . implode(", ", $countries) . '.sql
', 'a');
foreach($countries as $lang_to_insert) {
    fwrite($fp, "SET @var = (SELECT languages_id FROM
        languages WHERE 'code' = '" . $lang_to_insert . "');"
    );
    foreach ($csv as $key => $value){
        if ($lang_to_insert == strtolower($value[0])) {
            fwrite($fp, "UPDATE
                countries_translations
                SET countries_name = '" .
                    str_replace("'", "''", $value
                    [5]) . "'
                WHERE language_id = @var AND
                    countries_iso_code_2 = '" .
                    $value[2] . "';\n");
        }
    }
}
}

```

Listing 2.6: API-Aufruf geonames.org

```

$url = 'http://api.geonames.org/search?q=K%C3%B6ln&country=DE&
    lang=n&username=demo';
$response = file_get_contents($url);

```

2.4 VERSCHIEDENE TÄTIGKEITEN

2.4.1 *Austausch des API-Keys im Zahlungsmodul*

Der Zahlungsdienstleister *Klarna* empfiehlt, den zur Validierung eingesetzten API-Key aus Sicherheitsgründen mindestens ein mal jährlich auszutauschen. Um diesen Anforderungen gerecht zu werden, wurde über die Administrationsoberfläche ein neuer API-Key erstellt und in der lokalen Umgebung erfolgreich getestet. Anschließend wurde dieser in das Shop-System eingepflegt und im internen Wiki dokumentiert.

2.4.2 *Bugfixes in der Produktpflegetabelle*

In der bestehenden Produktpflegetabelle können aktive Produkte bearbeitet werden. Dies umfasst die Zuordnung von Saalplänen, die Bearbeitung von Übersetzungen, SEO-Keywords und Eigenschaften der Produkte, sowie die Erstellung von Filterkategorien, durch welche Produktkategorien sortiert werden können. Durch den großen Funktionsumfang kommt es immer wieder zu kleineren Fehlern die behoben werden müssen.

2.4.2.1 *Zuordnung von Eigenschaften zu Produkten*

Durch diesen Bug war es nicht mehr möglich, einem Produkt Eigenschaften zuzuordnen. Diese sind Voraussetzung um Filter erstellen zu können.

2.4.2.2 *Abspeichern von Hinweisen nicht mehr möglich*

Dieser Bug bewirkte, dass der Speichern-Button für Hinweise bezüglich der Platzwahl nicht mehr angezeigt wurde. Ursache für diesen Bug war, dass das PHP-Skript durch eine Exception abgebrochen wurde, bevor der Button geladen wurde und wurde durch einen früheren Fix ausgelöst, wodurch die betreffende Funktion mit einem leeren Array aufgerufen wurde. Die alte Funktionalität konnte durch kurze Recherche in der Versionsgeschichte der betreffenden Datei wiederhergestellt werden.

2.4.3 *Projektmanagement Produktpflegetabelle*

Für die Entwicklung einer neuen Produktpflegetabelle übernahm ich die Konzeption, sowie das Projektmanagement. Hierbei spielte vor allem die Benutzerfreundlichkeit eine große Rolle, da die aktuell eingesetzte Version eine initiale Ladezeit von über einer Minute aufweist. Um dem entgegenzuwirken und auch zukünftig auszuschließen, wurde die Tabellenstruktur angepasst, wodurch ein *Lazyloading* (Nachladen

von Inhalten erst auf Anfrage) möglich wurde. Ebenso wurde eine Pagination eingeführt, um auch bei einer deutlichen Steigerung des Angebots nachteilige Auswirkungen auf die Geschwindigkeit auszuschließen. Eine spannende und besondere Herausforderung bestand darin, dass sich das unterliegende System (eine modulare Umgebung, basierend auf *Kotlin* und *VueJS*) noch in der Entwicklung befindet und ich bis zum Projektstart keine Erfahrungswerte damit sammeln konnte.

2.4.4 *Systemadministration*

Als Tätigkeiten in der Systemadministration sind vor allem die Konfiguration der Cloud-basierten Telefonanlage, vor allem die Einrichtung neuer Gegenstellen sowie die Behandlung eines Problems nach einem fehlerhaften Update, das durch ein Zurücksetzen der Geräte gelöst werden konnte, sowie die Konfiguration der Firewall. Diese ist sehr restriktiv eingestellt, so dass einige Webseiten und Services nicht auf Anhieb funktionieren und zuerst eine Ausnahme in der Weboberfläche hinzugefügt werden muss.

PERSÖNLICHE EINSCHÄTZUNG

Da ich schon vor der Praktikumsphase in etwa ein Jahr im bei *Tickets75* tätig war, konnte ich sehr gut einschätzen welche Aufgaben und welche Unternehmenskultur mich erwartet. Da ich jedoch im Vorfeld weniger an der Entwicklung des Webshops beteiligt war, waren mir die täglichen Abläufe im Team noch unbekannt. Durch das Praktikum konnte ich tiefe Einblicke in die Organisation des Entwicklungsprozesses gewinnen, angefangen bei täglichen Scrum-Meetings, bis hin zur Durchführung von Releases des eigenen Codes. Ebenso konnte ich in kurzer Zeit viel Erfahrung mit PHP sammeln, einer Programmiersprache, die nach wie vor eine große Bedeutung hat, mit der ich bisher jedoch nur wenige Berührungspunkte hatte. Das mir entgegengebrachte Vertrauen seitens meiner Vorgesetzten und Kollegen führten dazu, dass ich mich mit großer Motivation auch Problemen annehmen konnte, die eine große Ausdauer erforderten. Beispielfhaft seien hier Bugs genannt, die erst durch viele Durchläufe im Debugger gelöst werden können. Diese Aufgaben förderten mein Verständnis für die genutzte Sprache und ihre Eigenheiten. Das gesamte Team brachte mir viel Respekt entgegen, Einwände meinerseits konnten jederzeit besprochen werden und bei Problemen und Fragen wurde mir immer geholfen. Gleichzeitig diente auch ich als Ansprechpartner für andere Teammitglieder in Bezug auf Fragen rund um den Webshop.

Zusammengefasst war das Praktikum eine tolle Erfahrung, für die ich mich an dieser Stelle bei allen Beteiligten noch einmal herzlich Bedanken möchte.

Die Zusammenarbeit mit *Tickets75* wird sowohl nach der Praxisphase, als auch nach dem Studium, auf beidseitigen Wunsch fortgesetzt.

Student

Ausbildungsbeauftragter