

Der für mich „perfekte“ Workflow

Table Of Contents

- [1 XAMPP](#)
- [2 NetBeans](#)
- [3 GIT/Bitbucket/Github](#)
- [4 AllwaySync](#)
- [5 WCF Package Builder](#)
- [6 Abschluss](#)

Generell ist der perfekte Workflow eine Glaubensfrage, wie Pommes mit Ketchup und Mayo, oder nur mit Mayo oder nur Ketchup? Oder in einer der drei Varianten in Kombination mit Zwiebeln? Oder doch Senf dazu?

Niemand wird wohl wirklich „Den perfekten Workflow“ haben, sondern jeder wird den für sich persönlich perfekten Workflow finden müssen und das ist auch gut so. So können wir alle voneinander lernen. Der für mich perfekte Workflow für die Entwicklung von Plugins für das WSC besteht aus insgesamt fünf verschiedenen Programmen und Diensten.

1 XAMPP

Für mich persönlich der erste und bislang einzige Webserver, den ich jemals für die lokale Entwicklung verwendet habe. Leichte Installation, leichte Bedienung, was möchte man für den Anfang mehr? Hinzu kommt, dass dieser kostenlos unter <https://www.apachefriends.org/de/download.html> herunterzuladen ist. Innerhalb weniger Sekunden ist der XAMPP einsatzbereit. Herunterladen, installieren, MySQL und Apache starten => fertig.

Das einzige was ich danach nach einer Neuinstallation nur mache, ist den kompletten htaccess Ordner zu leeren, um mir diesen ganz nach meinen Wünschen einzurichten.

2 NetBeans

2017 musste ich mich schmerzlich von meinem geliebten PHPDesigner verabschieden, da dieser nicht mehr weiter entwickelt wurde und es Zeit war, sich auf PHP 7.x einzulassen. So entdeckte ich für mich die kostenlose IDE Netbeans. Der Nachteil der meisten IDE's sind die Ladezeiten beim Öffnen. Ein Roadrunner wie PHPDesigner ist Netbeans nicht, aber es ist durchaus als eine der schnelleren IDE's zu sehen. Neben den normalen Syntaxhightigter hebt sich Netbeans durch die bessere Code-Vervollständigung hervor und die aus meiner Sicht bessere Übersicht. Ganz interessant fand ich am Anfang die individuelle Code Vervollständigung, sodass z.B. aus einem "debug" bei mir ein `die('<pre>'.print_r($myVariable, true));` wird. Weniger Zeit für mehr Ergebnis eben. Da ich öfters zwischen Desktop PC und Laptop wechsel, ist auch die Import- bzw. Exportfunktion der Einstellungen eine sehr feine Sache, so dass ich überall immer die gleichen Einstellungen nutzen kann. Zu finden ist Netbeans kostenfrei unter: <https://netbeans.org/>

3 GIT/Bitbucket/Github

GIT ist Versionsverwaltungsprogramm. Für diejenigen die davon nie etwas gehört haben oder sich immer gesträubt haben, so etwas zu benutzen, lasst euch gesagt sein „Tut euch selbst den Gefallen und nutzt es. Tut es, wirklich!“. Es gibt diverse Vorteile ein Versionsverwaltungsprogramm zu benutzen. Ein praktisches Beispiel? Ich saß mal vor einigen Jahren an einer Entwicklung eines Plugins, hab dafür einige Wochen gebraucht und irgendwann kam es, wie es kommen musste, die Festplatte meines PCs hat den Geist aufgegeben. Von heute auf morgen. Glaubt mir, ich war glücklich GIT eingesetzt zu haben.

Git ist nicht nur ein Backupsystem, womit man sämtliche Daten sichern kann, sondern es ermöglicht auch eine fast perfekte Zusammenarbeit in einem Team. Person A arbeitet an der Klasse „Pommes“, Person B an der Klasse „Ketchup“ und beide können unabhängig voneinander arbeiten, ohne sich in die Quere zu kommen oder gar die Arbeiten des anderen zu überschreiben. Ich persönlich nutze es als reines Backup System, da ich stets alleine ein Plugin entwickel und nehme mir dennoch die Zeit, immer meine Repros (Ein Repo ist das „Arbeitsprojekt“ innerhalb des Versionsverwaltungsprogramms zu einem Plugin) aktuell zu halten. Ich kann hierdurch einfach herausfinden, was für Änderungen ich vorgenommen habe, wenn eine Funktion plötzlich nicht mehr funktioniert. Hinzu kommt, dass ich viel zwischen Laptop und PC hin und her synchronisieren muss, da ich auf beiden Systemen arbeite. Da kommt mir git sehr gelegen, um schnell die Daten hin und her synchronisieren zu können. Lange Rede: Nutzt es, tut euch selbst den Gefallen!

Da ich in der Windows Generation groß geworden bin, kenne ich mich mit Kommandozeilen kaum aus. Aus diesem Grunde setze ich TortoiseGit ein, das sich wunderbar in mein Windows Menü integriert. Herunterladen könnt ihr dieses unter <https://tortoisegit.org/>. Damit lassen sich die wichtigsten über ein „schönes“ Windowsmenü bedienen.

Ob man nun gitHub.com, Bitbucket.org oder einen eigenen Server als Host für seine git Repros nimmt, ist in erster Linie egal. Ich entschied mich irgendwann zur Nutzung von Bitbucket, da dieses u.a. kostenlose private Projekte unterstützte, was GitHub bis heute nicht tut und man auch kostenlos Projektteams bis zu 5 Personen erstellen konnte. Bis heute konnte ich noch keine nennenswerten Beschränkungen feststellen (Ausnahme bietet hier, dass man bei Teams über 5 Personen etwas zuzahlen muss). Vom Funktionsumfang sind beide Anbieter relativ gleich. Innerhalb der Wollab Community ist allerdings GitHub der Verbreitetste und ich glaube, ich bin bis heute der einzige Entwickler in der Szene, der auf Bitbucket setzt. Aber gerade im Team finde ich die Funktionalitäten von HipChat genial. Innerhalb eines Chattes, sieht man zu jederzeit, wer an was gearbeitet und hochgeladen hat. Wer welche ToDo's erledigt hat etc.

4 AllwaySync

Mein erstes Problem mit dem WSC war, das ich nicht wusste, wie ich ein Zusammenspiel zwischen git und der WSC Installation ermöglichen konnte. Um einen reinen Arbeitsordner zu haben und einen, wo ich live alles testen kann, kam ich sehr schnell mit meinen altmodischen Mitteln an die Grenze. Die eine Möglichkeit besteht darin, alles im Installationsordner des WSC bzw. deren Endanwendungen zu ändern. Die Gefahr besteht beim späteren packen eines Plugins, das man hier Dateien vergisst, die man verändert hat oder neu hinzugefügt hat. Umso größer ein Plugin wird, umso größer wird auch die Wahrscheinlichkeit, das euch dieses passieren wird. Eine andere Möglichkeit ist, dass man einen Arbeitsordner schafft und händisch Veränderungen zwischen den zwei Ordnern vornimmt und die Dateien aus dem Arbeitsordner kopiert und in den Installationsordner hineinkopiert. Nachteil hier ist, dass man ebenfalls gerne Dateien vergisst und das ganze sehr zeit aufreibend ist. Um diese Möglichkeit noch in die größtmögliche Zeitverschwendung ausufernd zu lassen, kann man noch anbringen, dass man nach jeder Änderung das Paket neu packen kann und entsprechend Installieren kann. Eine größere Zeitverschwendung kann ich mir ganz ehrlich nicht vorstellen.

Da alle drei Optionen für mich irgendwann nicht mehr in Frage kamen, entschloss ich mich dazu, folgende Herangehensweise zu nehmen. Auf meiner lokalen Installation sieht es so aus, dass ich unter htdocs (Ordner für XAMPP zur Darstellung und Verarbeitung sämtlicher Dateien) zwei Unterordner habe. Einmal „plugins“ indem ich die reinen Arbeitsrepros habe (Nur in diesen Ordnern nehme ich Änderungen vor und pushe oder

pulle die Änderungen auf Bitbucket). Dann habe ich einen weiteren Ordner „wsc“ in dem die WSC Installation samt WSB oder anderen Endanwendungen installiert habe. Möchte ich eine Änderung live testen (Und diese betrifft eine PHP/JS/TPL Datei), nutze ich AllwaySync. AllwaySync ist ein Synchronisationsprogramm was eigentlich dafür gedacht ist, zwischen unterschiedlichen Speichermedien Dateien synchron zu halten. Das automatische kopieren mache ich mir an dieser Stelle zu Nutze und weiß stets, dass alle Dateien zu 100% innerhalb von zwei Klicks mit meiner Maus binnen der nächsten drei Sekunden zu 100% übereinstimmen. Es reduziert die Fehlerquote Dateien zu vergessen und spart mir unendlich viel Zeit. Man muss sich nur daran gewöhnen, vor einer Änderung alles zu synchronisieren. Der nächste kleinere Nachteil ist, das Programm kostet in der Vollversion zwischen \$15,95 und \$25,95. Es kann aber auch eine Testversion heruntergeladen werden, die eine begrenzte Anzahl an Synchronisationen zulässt. Zum Hersteller kommt ihr hier <https://allwaysync.com/business.html>.

5 WCF Package Builder

Lange Zeit habe ich mit 7 Zip alle Plugins manuell gepackt. Jedes „files“, „templates“ oder „acptemplates“ Archiv und zu gutem Schluss nochmal das ganze Plugin. Ich hab es getan, immer wieder. Bis ich irgendwann zum Glück auf den WSC Package Builder gestoßen bin. Innerhalb von einigen Klicks erstellt mir dieses kleine aber feine Programm ein voll installierbares Plugin. Ihr erhaltet dieses Programm unter <http://sourceforge.net/projects/wcfpb/> zum kostenlosen Herunterladen. Würde das Programm nun noch JS Dateien automatisch komprimieren, wäre ich wunschlos glücklich.

An dieser Stelle möchte ich noch anmerken, dass der WCF Package Builder zwar für das wcf geschrieben wurde, es aber durchaus mit dem WSC kompatibel ist.

6 Abschluss

Generell ist und bleibt der Workflow euch selber überlassen. Es wird jedoch immer bei euch im Kopf die Frage sein müssen, ob alles gut so ist, wie ihr es macht. Mein perfekter Workflow muss nicht der eurige sein, ihr könnt andere Werkzeuge verwenden oder auf den einen oder anderen Schritt verzichten. Das wichtigste ist und bleibt, ihr seid glücklich mit dem was ihr tut. Aber kann man es evt. nicht doch besser machen? !Page not found or type un

Ihr habt Fragen oder Anregungen? Lob oder Kritik? Lasst es mich doch durch einen Kommentar von euch wissen!