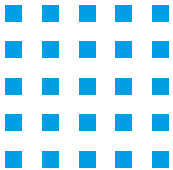
SOTA STUDIO

Refining your Brand.

SOTA STUDIO

Moderne Webentwicklung

Container, Pakete und Versionierung



SOTA STUDIO

Andy Hausmann

Gründer und Inhaber
Dozent und Trainer
Entwicklungsleiter

- Seit 2007 als **Backend Entwickler** tätig
- Seit 2012 selbstständig mit der **Branding-Agentur SOTA Studio**
- Seit 2015 als **freier Dozent** und **Berater** tätig

- Web-Entwickler für **Symfony, TYPO3 CMS, Neos CMS** und **Shopware**
- Konzeption von Web-Anwendung und komplexen Websites
- Setup von Entwicklungs- und Veröffentlichungs-Workflows
- Entwicklung von Extensions für TYPO3 CMS und Neos CMS
- Dozent und Trainer für Webentwicklung

SOTA STUDIO

Themenübersicht

- Vorwort
- Definition »Moderne Webentwicklung«
- Technologien
- Use Case
- Container
- Paketverwaltung
- Versionierung
- »TYPO3 CMS« und »DDEV«
- Fazit
- Links und Quellen

Vorwort – Abgrenzung

Präsentation dreht sich um die **Webentwicklung mit php**.

Erwähnte Tools spiegeln **Best Practice in der Entwickler-Gemeinde** wieder.

Es kann Sinn ergeben, diesen Best Practice auch auf **Bestandsprojekte** anzuwenden.

Das hängt jedoch von der Größe, dem initialen Aufwand sowie der Intensität der Wartung ab.

Definitiv empfehlenswert bei **Neuprojekten** – jedoch: Voraussetzungen beachten!

Vorwort – Voraussetzungen

Optimale Workflows machen da Sinn, wo mehrere Menschen kollaborieren.

Entscheidende Voraussetzungen sind:

- **Parallele Arbeit** an Features und Bugs im Team → Versionierung und Branching
- Teammitglieder benötigen **autarke Entwicklungsumgebungen** → Virtualisierung
- **Veröffentlichungs-Workflow** erwünscht (DEV → STAGE → PROD) → Virtualisierung
- Arbeit mit **Build-Prozessen** (Kompilierung von TypeScript, SCSS u. Ä.) → Paketverwaltung

Definition »Moderne Webentwicklung«

Der **De-Facto-Standard** einer modernen Webentwicklung entwickelt sich stetig weiter.

Somit auch die Auswahl der Tools für spezifische Anwendungsfälle.

Dennoch:

Es gibt ein **Set von Tools**, das sich in der **Entwicklergemeinde** einer **hohen Beliebtheit** erfreut.

Definition »Moderne Webentwicklung«

Doch was bedeutet **Moderne Webentwicklung** für uns – also bei SOTA Studio?

- Auswahl **optimaler Werkzeuge**
- **Best Practice** in der Entwicklergemeinde
- **Unkompliziertes Setup**
- **Teamfähig** und **skalierbar**
- **Zuverlässige Entwicklungsumgebung** / zuverlässiges Deployment
- **Produktive** und **effiziente Arbeit** an einem Projekt

Technologien – Worauf achten?

Es gibt **mehrere Welten zu berücksichtigen**:

- das OS → Linux, macOS, Windows
- Software auf dem Arbeitsgerät
- Software für das Frontend
- Software für das Backend

Fun Fact: Es gibt Lösungen und Tools, die unabhängig von OS und IDE funktionieren.

Technologien – Welche Tools wählen?

- **IDE** mit nützlichen Funktionen (z. B. Visual Studio Code und PhpStorm)
- **Docker** und **DDEV** → Virtualisierung
- **homebrew** → Paketverwaltung für MacOS und Linux
- **Chocolatey** → Paketverwaltung für Windows
- **yarn** → Paketverwaltung für JavaScript-Tools
- **composer** → Paketverwaltung für php
- **git** → Quellvode-Versionierung

Use Case

Angenommen, ihr arbeitet bereits als Team an einem Projekt – und erhaltet Zuwachs.

Wie schnell kann die neue Ressource mit der Arbeit beginnen?

Use Case — Abstrakter Best Case

- Tools installieren
- Quellcode klonen
- Pakete und Abhängigkeiten installieren
- Daten/Datenbank migrieren
- Umgebung starten
- Loslegen

Use Case – Beispiel mit Symfony

```
# Quellcode aus Versionierung klonen
$ git clone git@bitbucket.org:sotastudio/project.symfony ProjectSymfony && cd ProjectSymfony
# Benötigte Pakete + Abhängigkeiten installieren
$ composer install && yarn install
# JavaScript, CSS und Bilder kompilieren
$ yarn dev
# Datenbank und Tabellen erstellen sowie Testdaten importieren (Fixtures)
$ ./bin/console doctrine:database:create && ./bin/console doctrine:migrations:migrate
$ ./bin/console doctrine:fixtures:load
# Lokalen Webserver starten
$ ./bin/console server:start
```

SOTA STUDIO

Use Case – Beispiel mit TYPO3 CMS und Docker

```
# Quellcode aus Versionierung klonen
$ git clone git@bitbucket.org:sotastudio/project.typo3cms ProjectTYPO3CMS && cd ProjectTYPO3CMS
# Container starten und benötigte Pakete + Abhängigkeiten installieren
$ ddev start && ddev composer install && yarn install
# JavaScript, CSS und Bilder kompilieren
$ yarn dev
# Datenbank und Tabellen erstellen sowie Testdaten importieren (Fixtures)
$ ddev import-db < backups/db_snapshots/db.sql.gz
```

Container

- Virtualisierung von Dateisystem, Datenbank und weiteren »Servern« und »Services«
- Abkapselung der Entwicklungsumgebung vom restlichen System
- Vom Betriebssystem unabhängiges Setup
- Host-Betriebssystem nicht kompromittierbar
- Sichere lokale Entwicklung
- Einfach auf anderen Geräten reproduzierbar

Paketverwaltung

- Zentrale Verwaltung von Bibliotheken
- Verwaltung von Abhängigkeiten
- Verwaltung von Paketversionen

- Betriebssystem → homebrew (macOS / Linux) bzw. Chocolatey (Windows)
- JavaScript → yarn (Bibliotheken und Compiler für JSX, SCSS, TypeScript, uvm.)
- php → composer

Versionierung

- Dezentrale Quellcode-Verwaltung
- Unterstützung für Tags (v9.5.11, v9.x-dev, v10.2.0, v10.x-dev, …)
- Unterstützung für Branches (dev/feature-a, dev/feature-b, …)
- Änderungen können rückgängig gemacht werden

  (Beispiel: Update vom TYPO3 CMS Core oder Extensions)
- Eigenhosting (GitLab) oder Cloud Hosting (Bitbucket, Github)

»TYPO3 CMS« und »DDEV«

- DDEV ist ein Toolkit, das auf Docker aufbaut
- kommt mit einem einfachen und mächtigen CLI
- Docker-Container für Webserver und Datenbank werden automatisch gebaut
- nur wenige Schritte nötig
- lässt sich wunderbar versionieren

»TYPO3 CMS« und »DDEV« – Schritte

Einmalig je Arbeitsplatz:

- Docker installieren
- DDEV installieren

Je Projekt:

- TYPO3 CMS Distribution via composer laden
- DDEV initialisieren & starten
- Loslegen

»TYPO3 CMS« und »DDEV« – Beispiel

```
$ mkdir my-typo3-site && cd my-typo3-site
$ ddev config --project-type php --php-version 7.3
$ ddev composer create "typo3/cms-base-distribution:^9" --prefer-dist
$ ddev config --project-type=typo3
$ ddev restart
```

Fazit

Mit nur wenigen Schritten und ein paar Minuten Wartezeit, kann es direkt losgehen.

✓ Ressourcen sparen

✓ Zeit sparen

✓ Kosten sparen

✓ Spaß haben

Links und Quellen

DDEV Dokumentation
https://ddev.readthedocs.io/en/latest/

DDEV und TYPO3 CMS
https://docs.typo3.org/m/typo3/guide-contributionworkflow/master/en-us/Appendix/SettingUpTypo3Ddev.html

Git Workflow
https://www.atlassian.com/de/git/tutorials/comparing-workflows/gitflow-workflow

Yarn – Quickstart
https://yarnpkg.com/en/docs/getting-started

Composer – Quickstart
https://getcomposer.org/doc/00-intro.md

Github
https://github.com/

Bitbucket
https://bitbucket.org/

Gitlab
https://about.gitlab.com/

Homebrew
https://brew.sh/index_de

Chocolatey
https://chocolatey.org/install

Danke.

Für die Aufmerksamkeit.