

# Evolution von Code bei Major-Releases von Programmiersprachen

am Beispiel der Migration zu PHP7

---

Martin Duschek

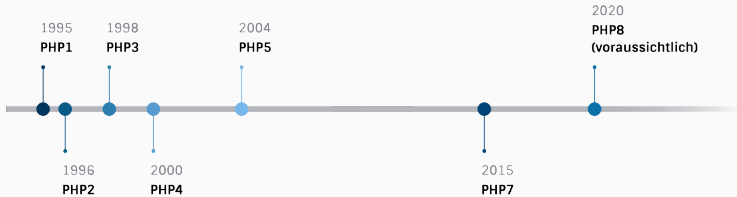
22. Januar 2020

HTWK Leipzig

1. Grundlagen
2. Änderungen der PHP-API
3. Geeignete Mittel
4. Migration
5. Auswertung
6. Ausblick

# Grundlagen

---



**Abbildung 1:** Major-Releases von PHP im Zeitverlauf

„Fahrplan“ zur Migration :

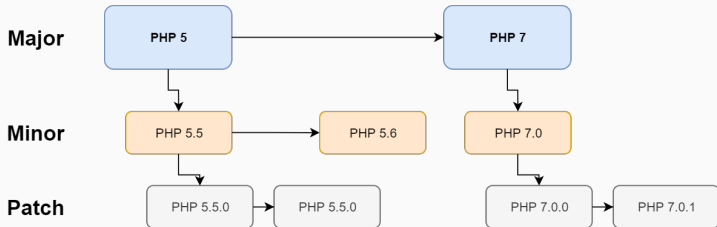
- Anforderungsanalyse und Definition der Migration
- Entwicklung von Werkzeugen zur Migration
- Entwicklung der angepassten Software
- Verifikation der Migration
- Support der alten Umgebung

Semantic Versioning :

Major.Minor.Patch[-Pre-Release]

- **Major:** inkompatible API-Änderungen
- **Minor:** abwärtskompatible Änderungen oder *Deprecations*
- **Patch:** abwärtskompatible Bugfixes

# Versionierung von Software



**Abbildung 2:** Semantic Versioning am Beispiel von PHP

# Änderungen der PHP-API

---



# Arten von Änderungen

- Abwärtsinkompatible Änderungen
- Veraltete Funktionen
- Geänderte Funktionen
- Neue Funktionen
- Entfernte Erweiterungen

# Abwärtsinkompatible Änderungen

## Indirekter Variablenzugriff

Ausdruck: `$foo→$bar['baz']`

### Interpretation in PHP 5

`$foo→$bar['baz']`

### Interpretation in PHP 7

`$foo→$bar['baz']`

## Implizite Konstruktoren

```
<?php
class foo {
    function foo($a) {
        echo("Instance Created");
    }
}
?>
```

## Anonyme Klassen

```
<?php
$foo = new class {
    public function bar() {
        echo("Hello World");
    }
}

$foo->bar();
?>
```

## mysql

- keine Prepared Statements
- Weiterentwicklung aufgegeben
- modernere Alternativen

- Erhöhung der Sicherheit
- Verständlichkeit des Codes
- Höhere Ausführungsgeschwindigkeit

## Geeignete Mittel

---

## Probleme bei komplexen Strukturen:

```
<?php
switch(1) {
    default:
        echo("Never evaluated");
        break;
    default:
        echo("Evaluated")
        break;
}
?>
```



## Vorteile:

- Erkennung komplexen Codes
- Beliebige Projektgröße
- Abschätzung über den Aufwand möglich
- Bericht als Arbeitsplan

## Vorteile:

- Protokollierte Änderungen
- Historische Versionen zurückverfolgbar
- Einfache Bearbeitung durch mehrere Personen
- Wartung alter Versionen weiter möglich

## Lokale Umgebungen

- Unabhängige Installation
- keine Abhängigkeiten
- keine Versionsverwaltung

## Docker-Integration

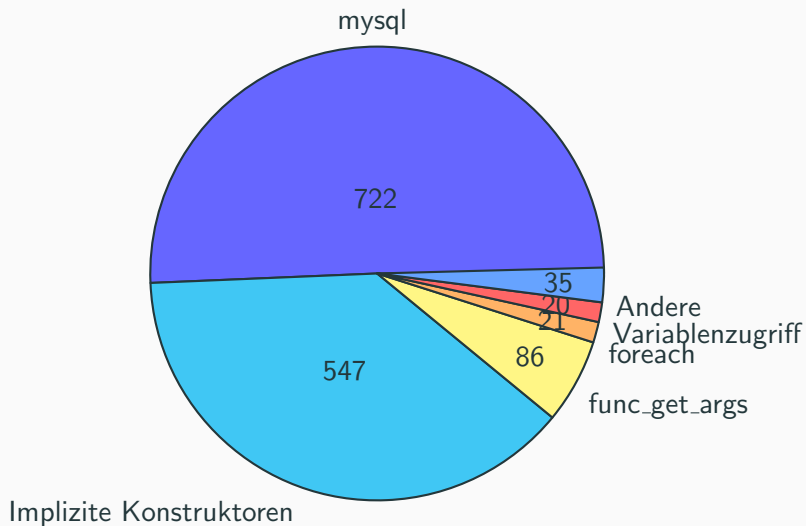
- Zentrale Konfiguration
- komplexes Ökosystem
- Versionsverwaltung per *git*

# Migration

---

**Tabelle 1:** Anteil zu migrierender Codeteile an der gesamten Codebasis

	Gesamt	Betroffen	Anteil
<b>Dateien</b>	5732	690	12,04%
<b>Codezeilen</b>	596198	1431	0,24%



- Weiterentwicklung von *php7mar*
- Einführung von Docker
- Konfiguration von *XDebug*

```
//veralteter Aufruf von mysql
$result = mysql_query($query);
if(!$result) $output .= mysql_error();

//Ersatz durch mysqli
$result = mysqli_query($db_link, $query);
if(!$result) $output .= mysqli_error($db_link);
```



# Migration der Software

```
<?php
class order {
    function __construct($order_id) {
        $this->order($order_id);
    }

    function order($order_id) {
        print($order_id);
    }
}
?>
```

- Manuelles Testen
- Monitoring mittels *New Relic*
- keine Unit Tests

- Für haus eigene Software weniger relevant
- Parallelbetrieb während Übergangszeit
- Historischer Code weiter in Versionsverwaltung

# Auswertung

---

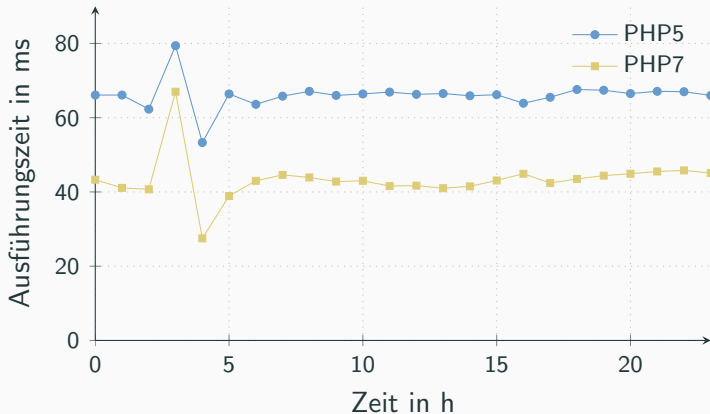
## Sicherheit:

- Entfernung von *mysql*
- Entfernung potentieller Risiken

## Klarheit des Codes:

- Einheitliche Konstruktoraufrufe
- Einheitliche Variablenaufrufe

## Geschwindigkeit:



# Ausblick

---



C. Anderson.

**Docker [Software engineering].**

*IEEE Software*, 32(3):102–c3, May 2015.



PHP Group.

**PHP: Backward incompatible changes - Manual -  
Changes to variable handling.**



ISO/IEC.

**ISO/IEC/IEEE International Standard for Software  
Engineering - Software Life Cycle Processes -  
Maintenance.**

*ISO/IEC 14764:2006 (E) IEEE Std 14764-2006 Revision of  
IEEE Std 1219-1998*), pages 1–58, September 2006.





Levi Morrison.

**PHP: rfc:remove\_php4\_constructors, November 2014.**



Oracle.

**MySQL :: MySQL 8.0 Reference Manual :: 13.5 Prepared SQL Statement Syntax.**



Tom Preston-Werner.

**Semantic Versioning 2.0.0.**