



## Software Craftsmanship: Software professionell entwickeln

Warum die Prinzipien des Handwerks in der Anwendungsentwicklung heute wichtiger sind denn je



**Wie viel Aufmerksamkeit schenken Sie Ihrer Software? Was ist Ihnen am Endprodukt wichtig? Nur die 'Oberfläche' und die geforderten Funktionen? Oder sind es auch die inneren Werte der Software, die Sie als Firma oder einzelner Softwareentwickler erstellen? Investieren Sie aktiv in die innere und äußere Qualität der Software oder nehmen Sie bewusst oder unbewusst „technische Schulden“ auf? Sind Sie als Firma daran interessiert, dass Ihre Entwickler ihre Fähigkeiten kontinuierlich verbessern, ihre Erkenntnisse teilen, über ihre Arbeit reflektieren und Verbesserungen einbringen, um den Wert der entwickelten Software zu erhalten und zu erhöhen? Dann könnten Sie genau wie uns die Prinzipien des Software Craftsmanship interessieren.**

## Methoden und Techniken im Wandel

In den letzten Jahren hat sich gezeigt, dass agile Methoden klassische Projektentwicklungsmethoden ergänzen können und in einigen Bereichen überlegen sind. Agile Methoden haben den Menschen wieder in den Vordergrund gerückt als wichtigen Erfolgsfaktor für das Gelingen eines Projekts. Aber hat unser technisches Verständnis in der Softwareentwicklung und -programmierung bei diesem Wandel mitgehalten?

Wir denken, dass dies gerade im Bereich der Individualentwicklung von Geschäftsanwendungen nicht der Fall ist. Vielfach erwarten insbesondere agile Vorgehensweisen eine umfassende, professionelle Haltung zur Softwareentwicklung und zum produzierten Ergebnis, dem Code. Das Bild des Software Craftsman (also eines Handwerkers) ergänzt das sehr wissenschaftlich orientierte Vorgehen, wie es an Hochschulen gelehrt wird. Mit einem Handwerk sind Werte, Techniken und Einstellungen zum eigenen Handeln verbunden.

## Prinzipien des Handwerks

Craftsmanship bezeichnet also die Einstellung zur eigenen Profession als Softwareentwickler. Die Regeln und Fähigkeiten eines Handwerks werden von Person zu Person, von erfahrenen Handwerkern an junge und durch Austausch untereinander weitergegeben. Und handwerkliche Fähigkeiten erwirbt man durch Übung. Aus Sicht der Software-Craftsmanship-Bewegung ist dieses Verhältnis von "Lehrlingen" und "Meistern" auch auf die Softwareentwicklung zu übertragen. Es betont den praktischen Aspekt der Programmierung im Gegensatz zur theoretisch ausgerichteten Informatik.

Programmierung und individuelle Softwareentwicklung haben viel mit der Herstellung von handgefertigten Unikaten unter Anwendung großer Erfahrung gemein. Es wäre also eine Verkürzung, Software Craftsmanship auf die Code-Qualität zu reduzieren. Software Craftsmanship legt den Fokus auf **drei Prinzipien**: Qualitätsanspruch, professionelles Handeln sowie eine Gemeinschaft von Experten, die lebenslanges Lernen fördert.

Bei OPITZ CONSULTING legen wir großen Wert auf diese Prinzipien sowohl für interne Entwicklungen als auch für die Anwendungsentwicklung bei unseren Kunden.

## Qualitätsanspruch

Häufig wird in der Softwareentwicklung ein Fokus auf die externe Qualität gelegt. Tester betrachten Software als eine Black Box und prüfen, ob sie das erwartete leistet. Dies ist aber, insbesondere wenn es um Investitionsschutz geht, zu kurz gedacht. Dann ist die interne Qualität von Software genauso wichtig.

Software wird viel häufiger gelesen als geschrieben. Der größte Anteil an Kosten für Software fällt in der Wartungsphase an. Software, die einfach zu verstehen und zu restrukturieren ist, ist günstiger in der Wartung. Daher sollte Software nicht nur funktionieren, sondern sie sollte auch gut gefertigt sein. Dies heißt für uns, dass sich in der Software etablierte Muster und Vorgehensweisen wiederfinden, die sich am aktuellen Stand der Technik orientieren.

Nach diesen Grundsätzen sichern unsere Entwicklerteams die Qualität ihrer Arbeit:

- Wir entwickeln Software testgetrieben und tolerieren keine vermeidbaren Defekte
- Wir verwenden die in einer Sprache gebräuchlichen Idiome
- Wir beherrschen verschiedenen Techniken und Methoden und lassen uns nicht durch unbekannte Techniken von der Auswahl der besten Methoden und Techniken abhalten
- Wir nutzen die Technik so, dass die Lösung so einfach wie möglich und so kompliziert wie nötig ist und setzen keine unnötig komplizierten Mittel ein.

Zu unserem Qualitätsanspruch gehört es auch, dass wir diese Mindeststandards von "produzierter" Software leben und diese nicht verhandelbar sind. Genau so wie ein Elektriker einem Kunden klar machen muss, dass es keine Option ist, aus Kostengründen Kabel diagonal zu verlegen.

## Professionelles Handeln

Wir wollen das Geschäft unserer Kunden verstehen und nicht techniker-  
liebt sein, d. h. insbesondere schlagen wir keine Lösung vor, bis wir uns  
sicher sind, das Problem verstanden zu haben. Dies bedeutet, dass wir auch  
die Wirtschaftlichkeit einer Lösung im Blick haben. Wir bieten passgenaue  
Lösungen an und verzichten auf „goldene Wasserhähne“. Unser Verständ-  
nis von einer guten Lösung geht damit über die reine Machbarkeit hinaus.  
So finden wir die für den Kunden auch langfristig beste Lösung.

Unser Verständnis gleichen wir regelmäßig mit dem Kunden ab und kön-  
nen auch Kurskorrekturen vornehmen. Kurskorrekturen können aber auch  
dann notwendig sein, wenn es zu Abweichungen vom ursprünglichen Plan  
kommt und weniger Features als geplant in der vorgesehenen Zeit umge-  
setzt werden können.

Zum professionellen Handeln gehört es auch, den Kunden zu beraten,  
wenn er „technische Schulden“ aufnimmt bzw. nicht genug in die innere  
Qualität der Software investiert. Wir können hier abwägen, wann es kurz-  
fristig sinnvoll sein kann, den Qualitätsanspruch etwas niedriger zu bewer-  
ten. Wir zeigen den Auftraggebern die Konsequenzen auf, vereinbaren  
einen Investitionsplan für die Wiederherstellung der Qualität und sagen  
dort bewusst Nein zur Aufnahme „technischer Schulden“, wo es uns unver-  
antwortlich erscheint.

## Eine Gemeinschaft von Experten

Als Gemeinschaft von Experten lernen wir miteinander und voneinander.  
Wir teilen unser Wissen und üben gemeinsam. Wir streben nach ständiger  
Verbesserung und hinterfragen dabei auch kritisch vorhandene Gewohn-  
heiten.

Dieses Prinzip praktizieren wir z. B. in Leserunden zu ausgewählten The-  
men, um unsere Kenntnisse über Methoden und Prinzipien zu erweitern.  
Neben der ständigen Weiterentwicklung unserer Methodenkenntnisse  
wollen wir auch den Umgang mit unseren Werkzeugen, d. h. insbesondere  
Programmiersprachen, Frameworks und Entwicklungsumgebung ständig  
üben und verbessern. Dazu führen wir regelmäßig Programmierabende  
durch.

Der Austausch mit anderen Experten außerhalb unserer Firma ist ebenso  
wichtig. Wir lernen durch Kenntnis anderer Vorgehensweisen, Denkmus-  
tern und konstruktiver Kritik. Gleichzeitig geben wir unser Wissen durch  
die Teilnahme an User-Gruppen und Konferenzen weiter."

### Trainieren Sie mit unseren Spezialisten!

#### Java Code Camp

Sie möchten einmal live erleben,  
was Software-Craftsmanship bedeuten  
kann und wie sich Code Dojos anfühlen?  
Dann melden Sie sich zu einem Code Camp  
mit unseren Java-Experten an.  
Details und Anmeldung:  
[www.opitz-consulting.com/code-camp](http://www.opitz-consulting.com/code-camp)



#### Inhouse-Trainings

Gerne kommen wir in Ihr Haus und führen mit Ihren Mitarbeitern  
maßgeschneiderte Trainings im Bereich Software-Craftsmanship durch.  
Sprechen Sie uns an.

#### Sprechen Sie uns an:



**Richard Attermeyer**

**Senior Solution Architect**

Telefon: +49 2261 6001-0

Telefax: +49 2261 6001-4200

E-Mail: [richard.attermeyer@opitz-consulting.com](mailto:richard.attermeyer@opitz-consulting.com)

#### Mehr zu unserem Leistungsbereich Anwendungsentwicklung:

[www.opitz-consulting.com/anwendungsentwicklung](http://www.opitz-consulting.com/anwendungsentwicklung)



#### Folgen Sie uns



[www.opitz-consulting.com/newsroom](http://www.opitz-consulting.com/newsroom)