

# Java aktuell



iJUG  
Verbund  
www.ijug.eu

## Topaktuell

Alle Neuheiten  
von Java 10

## Aus der Praxis

Bastlerfreundliche  
Heim-Automatisierung

## Der neue Trend

Serverless Java  
mit Fn Project

# Java ist cool





# Soft Skills – nur etwas für Dampfplauderer?

Christoph Menzel, Method Park Engineering GmbH

*Sind Soft Skills wirklich nur etwas für Dampfplauderer oder sind sie heutzutage für jeden von uns wichtig, auch für den normalen, einfachen Software-Entwickler? Der vorliegende Artikel geht dieser Frage nach und zeigt im Anschluss zu einigen ausgewählten Soft Skills, wie man an ihnen arbeiten kann.*

Was sind Soft Skills eigentlich? Wie sehen ihre Gegenstücke, die Hard Skills, aus? Für Hard Skills (Fachkompetenzen) findet sich eine der treffendsten Definitionen bei Wikipedia: „... die Fähigkeit von Personen, berufstypische Aufgaben und Sachverhalte den theoretischen Anforderungen gemäß selbstständig und eigenverantwortlich zu bewältigen ...“ (siehe „<https://de.wikipedia.org/wiki/Fachkompetenz>“).

Fachkompetenzen sind also Fähigkeiten, die sich von Beruf zu Beruf unterscheiden: Ein Schreinermeister benötigt andere Fachkompetenzen als ein Bauingenieur oder ein Software-Entwickler. Diese theoretischen Fähigkeiten lassen sich zum Beispiel durch das Lesen von Büchern, das Absolvieren einer Ausbildung oder in der Zusammenarbeit mit Kollegen erlernen.

Das selbstständige und eigenverantwortliche Bewältigen von Aufgaben mithilfe der Hard Skills ist ein Bestandteil unserer Ausbildung. In der Software-Entwicklung wird diese Kompetenz beispielsweise bei agilen Vorgehensweisen sehr gestärkt. Welche Fachkompetenzen muss ein Software-Entwickler konkret beherrschen? Dazu einige Beispiele (siehe Abbildung 1):

- Programmiersprachen
- Versionsverwaltung
- Frameworks
- Algorithmen und Datenstrukturen
- Testen
- Debugging
- Patterns

## Soft Skills

Was aber sind Soft Skills, die oft auch „weiche Fähigkeiten“ genannt werden? Aus der Vielzahl der Definitionen erscheint wieder die bei Wikipedia passend: „... combination of interpersonal people skills, social skills, communication skills, character traits, attitudes, ... that enable people to ... work well *with others* ... achieve their goals with *complementing* hard skills.“ (siehe „[https://en.wikipedia.org/wiki/Soft\\_skills](https://en.wikipedia.org/wiki/Soft_skills)“). Die kursiv gesetzten Keywords geben Aufschluss über die Bedeutung von Soft Skills:

#### ■ **Combination**

Kombination bedeutet, dass Soft Skills selten alleine stehen und nur schwer voneinander abzugrenzen sind. Ganz im Gegenteil, sie gehen ineinander über und die Kombination vieler Fähigkeiten ist besonders wichtig.

#### ■ **Enable**

Soft Skills befähigen uns, gut mit anderen zusammenzuarbeiten. Soft Skills sind also vor allem dann wichtig, wenn es um die Zusammenarbeit mehrerer Menschen geht.

#### ■ **Complementing**

Soft Skills ergänzen und vervollständigen die Hard Skills. Jede Fähigkeit für sich alleine genommen ist nutzlos. Erst ihre Kombination macht Hard und Soft Skills extrem wert- und machtvoll. Nachfolgend einige Beispiele, die Soft Skills noch greifbarer machen:

### Warum braucht man in der Software-Entwicklung Soft Skills?

Ein Software-Entwickler muss hauptsächlich eine Programmiersprache beherrschen und performante, sichere Software schreiben können. Wozu braucht er Soft Skills? Die wichtigsten Gründe lassen sich in vier Schlagwörtern zusammenfassen:

- Komplexität
- Zunehmend rascherer Technologiewandel
- Qualitätsansprüche
- Teamwork

### Komplexität

Produkte und die dazugehörige Software werden immer komplexer. Ein Beispiel ist die smarte Zahnbürste: Sie stellt fest, wie gut oder schlecht und wie lange man Zähne putzt. Die gesammelten Daten kann man sich in einer iOS oder Android App ansehen, vermutlich auch in der Cloud speichern und vielleicht sogar per Facebook oder Twitter teilen.

Bei so etwas Simplem wie einer Zahnbürste kommen schon sehr viele verschiedene Technologien und Software-Produkte zum Einsatz. Bei einem Auto oder einem Medizinprodukt, wie etwa einem Computertomografen, steigt die Komplexität um ein Vielfaches. Um mit dieser Komplexität besser zurechtzukommen, helfen Soft Skills wie zum Beispiel Diszipliniertheit, Gründlichkeit, Lernbereitschaft oder Ordnungssinn.

### Technologiewandel

Die Software-Entwicklung ist wohl eine der Branchen, die sich am schnellsten verändert. Dies zeigt beispielsweise die Zahl neuer Java Script Frameworks, die jeden Monat entstehen. Auch neue Technologien, wie etwa 3D-Druck, Virtual und Augmented Reality oder Drohnen, stellen immer wieder andere, neue Herausforderungen an einen Software-Entwickler. Um mit diesem schnellen Wandel mithalten zu können, braucht es Soft Skills wie Lernbereitschaft, Lern-, Anpassungs- oder Begeisterungsfähigkeit.

### Qualitätsanspruch

Vor allem in sicherheitskritischen Branchen wie der Automobilindustrie, der Medizintechnik und dem Energie-Sektor werden besonders hohe und einmalige Qualitätsansprüche an Software gestellt. Oftmals geht es dort um Menschenleben oder um sehr viel Geld. Aber auch in anderen, weniger sicherheitssensiblen Branchen soll-

te man hohe Qualitätsansprüche verfolgen, denn Qualität ist ein „Unique Selling Point“. Soft Skills vermitteln das dafür notwendige Bewusstsein; sie unterstützen Qualität durch Gründlichkeit, Ordnungssinn und Organisationsfähigkeit.

### Teamwork

Kaum jemand arbeitet heutzutage nicht im Team oder nicht mit anderen Menschen zusammen. In der Software-Entwicklung schreiben vor allem neue Entwicklungspraktiken wie Scrum oder SAFe Teamwork und Kommunikation sehr groß. Unterstützende Soft Skills sind in diesem Fall Selbstorganisation, Eigenverantwortlichkeit und Kommunikationsfähigkeit.

Alleine mit Hard Skills lassen sich die gerade genannten Gründe nicht befriedigen. Genau deshalb werden Soft Skills immer wichtiger und ein entscheidender Faktor für den Erfolg von Teams oder sogar Unternehmen.

Hard Skills werden, wie erwähnt, vor allem durch Bücher und Ausbildung vermittelt. Wie erlernt man Soft Skills? Sie sind weder expliziter Teil einer Berufsausbildung noch eines Studiums. Soft Skills werden oftmals erst im Lauf einer beruflichen Entwicklung aktiv gefördert, etwa wenn man eine höhere Position erreicht. Dann ist es jedoch eigentlich zu spät, denn bereits ein Junior-Software-Entwickler benötigt viele der vorher genannten Soft Skills in seinem Alltag. So sind heutzutage zum Beispiel Kommunikationsfähigkeit, Lernbereitschaft, Gründlichkeit und Organisationsfähigkeit ein täglicher Begleiter jedes guten Software-Entwicklers.

Soft Skills werden jedoch weniger erlernt als vielmehr adaptiert: Man schaut sich Soft Skills von Eltern, Lehrern, Freunden, Kollegen oder Mitmenschen ab. Allerdings können Soft Skills auch trainiert und weiterentwickelt werden. Hierzu einige Tipps und Tricks.

### Social Skills

Unter Social Skills versteht man eine Vielzahl an Soft Skills wie zum Beispiel Kommunikation, Wortwahl, aktives Zuhören, Körpersprache, Augenkontakt, Empathie, Gefühlskontrolle oder Selbstbeherrschung. Sie helfen dabei, selbstbewusst aufzutreten, neue Leute kennenzulernen – dadurch vielleicht sogar den Traumjob zu bekommen –, Beziehungen aufzubauen und zu pflegen oder das Arbeitsklima zu verbessern. Um Social Skills zu fördern und sich häufiger aus der eigenen Komfortzone zu bewegen, bieten sich zwei Übungen an: das 60-Seconds-Game und das Compliments Game.

Im 60-Seconds-Game geht es darum, sich jemandem persönlich vorzustellen, wenn man einen neuen Ort betritt: eine Veranstaltung, ein Café oder einen Bus. Es ist egal wem, es ist egal wie – man muss es nur innerhalb von 60 Sekunden tun, denn nach dieser ersten Minute wird die Angst sich vorzustellen zu groß.

Diese Angst ist jedoch unbegründet, denn man hat nichts zu verlieren. Im Gegenteil, man kann nur gewinnen und sich durch diese Übungen für die wirklich wichtigen Momente vorbereiten. Man stärkt das eigene Selbstbewusstsein und lernt, sich in solchen doch eher ungewöhnlichen Situationen wohlfühlen und souverän aufzutreten.

Das Compliments-Game ist ähnlich: Dreimal täglich sollte man ein aufrichtiges Kompliment vergeben. Es ist vollkommen egal wem, die



ohne ihm in die Augen sehen zu müssen. Oder man schaut auf eine andere Partie des Gesichts, zum Beispiel die Nase oder den Mund.

## Zuhören

Neben dem Augenkontakt ist auch das richtige Zuhören sehr wichtig für eine gute und zielführende Kommunikation. Oftmals nimmt man das Gesagte nicht wirklich auf. Ein aktives Zuhören hilft, mehr aufzufassen und dem Gegenüber ein Feedback zu senden. So kann es hilfreich sein, das Gehörte mental zu rekapitulieren, Fragen zu stellen, zusammenzufassen, zu wiederholen oder zuzustimmen.

Aber auch schon kleine Gesten wie ein Nicken oder ein Lächeln geben Feedback und zeugen von Aufmerksamkeit. Wichtig dabei ist es den Gesprächspartner niemals zu unterbrechen, seine Sätze nicht für ihn zu beenden und nicht die Kontrolle über das Gespräch an sich zu reißen. Das gibt dem Redenden nur ein Gefühl von Unselbstständigkeit. Der Zuhörende muss seinem Gegenüber die Zeit geben, die dieser braucht, um eine These oder ein Argument zu entwickeln.

Ein in der Software-Entwicklung sehr bekanntes Prinzip ist „Keep it simple, stupid“, kurz „KISS“. Dieses Prinzip lässt sich auch sehr gut bei alltäglichen Kommunikationen verwenden. Vor allem bei schriftlicher Kommunikation ist es extrem hilfreich.

Oftmals zeugt das absichtliche Verwenden komplizierter Wörter und Sätze nicht von Intelligenz, sondern vielmehr von Hochnäsigkeit und Arroganz. Deshalb sollte man immer versuchen, die Kommunikation einfach zu halten, eine unkomplizierte Sprache zu verwenden, nur maximal drei bis fünf verschiedene Punkte anzusprechen und auf direktem Weg mit dem richtigen Empfänger zu kommunizieren. Ist Kommunikation einfach und simpel, dann ist sie auch effizient.

## Fokussieren

In der heutigen Zeit gibt es so viele Störfaktoren wie noch nie. Um nur ein paar zu nennen: Facebook, Twitter, Handy, Mail, Chat, Kollegen oder Smart Watches. All dies und noch mehr hindert daran, uns zu fokussieren und konzentriert an einer Sache zu arbeiten. Der Versuch des Multitaskings verschärft das Problem noch. Es kann aufschlussreich sein, Störfaktoren aufzuschreiben und zu sammeln. Um seinen Fokus zu finden und ihn beizubehalten, gibt es einige Tricks.

Eine Methode ist „Pomodoro“, relativ einfach mit einem hohen Wirkungsgrad. Zu Beginn formuliert man seine Aufgabe schriftlich, um sie festzuhalten und sich klar zu machen, woran man arbeiten möchte. Danach stellt man sich einen Wecker auf 25 Minuten; es geht auch kürzer, jedoch keinesfalls länger. In diesen 25 Minuten bearbeitet man die vorher formulierte Aufgabe und lässt keinerlei Störungen zu. Man liest keine Mails, öffnet Twitter nicht, führt keine Telefonate und lässt auch keine Ablenkung von Kollegen zu, außer eines davon wäre Teil der Aufgabe.

Oftmals muss man seine Kollegen zu Beginn darüber informieren und ihnen zum Beispiel durch eine Lampe oder ein Schild signalisieren, dass gerade nicht gestört werden darf. Natürlich kann es Ausnahmen geben, jedoch sollte es dann dafür einen sehr triftigen Grund geben.

Sind die 25 Minuten vorbei, wird eine Fünf-Minuten-Pause eingelegt, um neue Kraft für die nächste Runde zu sammeln. Je nach

Aufgabengröße kann an der gleichen Aufgabe weitergearbeitet oder eine neue formuliert werden. Nach vier Runden sollte man eine längere Pause machen, etwa 20 bis 30 Minuten.

Manche Störfaktoren lassen sich leichter, andere schwieriger blockieren. Das Aufsetzen von Kopfhörern kann ein Signal für Kollegen sein, dass man sich gerade in einer Pomodoro-Runde befindet, und schafft gleichzeitig einen Raum ohne Störgeräusche. Auch die Benachrichtigungen von Handy, Facebook, Twitter oder Mail-Client sollte man ausschalten. Es genügt normalerweise völlig, diese Medien zwei-, drei- oder viermal am Tag zu lesen und zu beantworten. Hierfür gibt es verschiedene Apps und Plug-ins für alle gängigen Browser und Handy-Plattformen.

Pausen werden leider bei der Fokussierung auf eine Arbeit oder Aufgabe meist sehr unterschätzt. Deswegen sind bei der Pomodoro-Technik die Pausen ein fester Bestandteil, da sie extrem hilfreich und förderlich für die Konzentration sind. Als Faustregel sollte man nach 45 Minuten Arbeit für fünf Minuten Pause, nach 90 Minuten zehn Minuten Pause und nach vier Stunden 30 Minuten Pause machen.

Man kann die Pausen auch gut nutzen, um mit Kollegen einen Kaffee oder Tee trinken zu gehen oder zusammen einen kleinen Spaziergang zu unternehmen. Im Gespräch mit Kollegen kommt man entweder gemeinsam auf eine Lösung oder ein Kollege hatte bereits das gleiche Problem und eine gute Lösung dafür. Pausen sind nicht unproduktiv, ganz im Gegenteil, sie erhöhen die Produktivität und den Fokus enorm.

## Fazit

Soft Skills sind keinesfalls soft. Sie sind vielmehr ein wichtiger Bestandteil unseres alltäglichen Lebens, sowohl beruflich als auch privat, und können den Unterschied zwischen Scheitern und Erfolg ausmachen. Sie sind nicht nur etwas für Dampfplauderer, sondern werden auch in der Software-Entwicklung immer wichtiger. Jeder muss an seinen Soft Skills arbeiten, Fortbildungen dazu gibt es mindestens genauso viele wie für die sogenannten „Hard Skills“.



**Christoph Menzel**

Christoph.Menzel@methodpark.de

Christoph Menzel ist bei Method Park als Senior Software Engineer tätig. Seine Themen-Schwerpunkte sind agile Entwicklungsmethoden und die Testautomatisierung. Als agiler Coach und zertifizierter Scrum-Master unterstützt er seine Kunden bei der Einführung von Scrum oder bei der Optimierung von Entwicklungsprozessen. Außerdem hält er Hands-on-Workshops zu Themen wie „Unit Tests“, „Clean Code“ oder „Continuous Integration & Delivery“. Darüber hinaus begleitet Christoph Menzel seine Kunden aktiv bei der Entwicklung von Software und engagiert sich beim Software Engineering Camp Erlangen.