

THOMAS MARBELLA

# Leseprobe Senior

# CODE TO CASHFLOW

Die 5 Level zu mehr Marktwert  
& Freiheit im AI-Zeitalter

# Gehaltsverhandlungen

Willkommen beim Thema, auf das viele (meist introvertierte) Devs und Nerds so gar keinen Bock haben: Gehaltsverhandlungen! Hier gut zu punkten ist ein erlernbarer Skill, den viele gar nicht als solches sehen oder gleichermaßen denken, dass alle 1-2 Jahre die 10 Minuten Dialog ausreichen. Die Realität könnte kaum anders sein.

Da nur wenige Firmen automatische, jährliche Gehaltserhöhungen anbieten (\*OneCode \*Eigenwerbung) und Tarifverträge unter Entwicklern nur sehr selten vorkommen, obliegt es deinem Skill und vor allem deinem Mut, das richtig anzugehen.

Ohne große Umschweife, die Top 5 Faktoren für deine erfolgreiche Verhandlung:

1. Verstehe die Sicht deines Chefs und deiner Firma, sei empathisch
  - Beispiel: „*Was muss ich machen / Was brauchst du ...?*“
  - Gegenbeispiel: „*Ich will, weil ...*“
2. Dein Mehrwert und Impact - fürs Team, fürs Produkt
  - *„Ich schreibe sehr guten Code“* ist Standard, kein Grund für mehr Gehalt
  - Deine Arbeit beschert deiner Firma +100k Gewinn? 10% für dich!
  - Deinen monetären Mehrwert zu bestimmen ist schwierig, aber möglich!
3. Dein Gehaltsband samt Firmengröße und Struktur
  - Große Konzerne: etliche Level & Bänder, oft mit Regeln
  - Kleine Firmen: proaktiver Dialog mit dem Chef, Mehrwert = mehr Gehalt
4. Dein Marktwert
  - Beispiel: du bist Cobol Experte, einer von 100 in DE -> hoher Marktwert
  - Gegenbeispiel: Senior React Dev -> austauschbar und ersetzbar (leider)
  - Konkrete (höhere) Angebote vorliegen zu haben, schafft dir Optionen
5. Deine Unverzichtbarkeit im Team & Projekt
  - Du bist ein Urgestein der Codebase und jeder im Team baut auf dich?
    - ... weiß dein Chef und CTO das auch?

- Selbst 2-3 neue Devs könnten dich mittelfristig kaum ersetzen? Gut!
- Du machst die Arbeit von 2-3 Devs, zum Gehalt von einem? Auch gut!

Wichtig zu verstehen ist auch, dass obige 5 Punkte der Priorität nach sortiert sind. Egal wie unverzichtbar du bist, egal was dein Marktwert ist - wenn du nicht empathisch deinen Chef verstehst oder deine Firmengröße und deren Gehaltsbänder nichts hergeben, wird es eng für dich.

Was nahezu immer funktioniert, aber genau deswegen nicht so einfach ist, ist Tipp #2: Schaffe so viel Mehrwert und Impact in deiner Arbeit, dass dein Chef dein Gehalt erhöhen „muss“. Das beste Beispiel sind hierfür oft Sales-Jobs. Zugegeben, es könnte kaum weiter entfernt sein vom nerdigen Entwickler Dasein, aber die Analogie ist treffend: Bringe mehr Umsatz rein, bekomme deine Provision - und alle haben gewonnen. Damals fand ich Vertriebler-Gehälter unverschämt hoch (100-200k als Senior sind da keine Seltenheit!), mittlerweile kann ich es umso mehr nachvollziehen.

Wenn dir die Ideen fehlen, wie du das als Dev umgesetzt kriegst, frag proaktiv nach, gerne auch beim CEO oder CTO. Die beiden werden sich über deine Ambition freuen und dir Input geben - wahrscheinlich nicht ganz in dem Format, wie du es gerne hättest, aber es ist ein Anfang und besser als Nichts.

Damit du damit nicht allein dastehst, anbei erprobte Beispiele:

- Mehr User in der Software, durch bessere Bewertungen / Features
- Weniger Serverkosten oder Ladezeiten durch optimierten Code
- Geringere Crash Rates / Ausfallzeiten bei App und Servern
- Mehr Umsatz durch Conversion Optimierung über Split Tests
- Mehr Projekte/Umsatz (bei Agenturen) durch dein eigenes Netzwerk
- Neue Team Member über dein Netzwerk (spart teure Recruiter)
- Auch gut: Mehr Reichweite / Reputation durch Conference Talks

Wie du siehst: Ein einfaches „*ich schreib' guten Code*“ reicht da selten aus, sondern ist eher die Baseline. Viele Entwickler halten naiv an dem Glaubenssatz fest und wundern sich dann, dass es mit der Karriere nur so mittelmäßig klappt. Aber nicht du! ;)

## **Senior Level: 70k+**

“*Welcome to the senior league!*“ Spätestens hier trennt sich die Spreu vom Weizen und spätestens hier zeigt sich, ob du *wirklich* ein Senior Dev bist und ob dein Gehalt das auch wieder spiegelt. Jeder Uni Absolvent kann sich als Senior bezeichnen, aber ob dein Titel deinen Skills gerecht wird, ist eine andere Frage. Realistisch am Markt sind mittlerweile mindestens 5 Jahre, damit du als Senior durchgehst, damals waren es mal Zehn. Ich kenne aber auch Devs, die nach der Zeit gerade als Mid Level durchgehen.

Aber kommen wir erstmal zu den Basics, was du als Senior drauf haben solltest:

- Du kriegst komplexe Features & Probleme alleine gelöst
  - ... mit sehr guter und nachhaltiger Code Qualität
  - ... die trotzdem dein Junior im Team auch versteht (!)
- Architekturentscheidungen zur Skalierbarkeit sind dein täglich Brot
- Du stellst die Best Practices fürs Team zusammen, im Team
- Mentoring & Pairing deines Teams ist dir wichtig
- Fachlichen Diskurs auf hohem Niveau und Tiefe kriegst du hin

Kurzum: du bist zwar (noch) nicht das „Mädchen für Alles“, aber deine Rolle wird im Team und Unternehmen immer wichtiger. Deine Entscheidungen haben große Auswirkungen und Gewicht (neudeutsch „Impact“) und damit einhergehend kommt einiges an Verantwortung.

So oder so wirst du merken, dass gerade Letzteres ein wichtiger Faktor für deine Karriere und vor allem dein Gehalt sein wird. Mehr Impact und mehr Verantwortung werden (fast)

immer entsprechend entlohnt. Gleichermaßen könntest du als Senior den besten Code der Welt schreiben - wenn dein Team sich mit Mittelmaß zufriedengibt und kein Mehrwert oder Impact im Produkt dabei herauskommt, wird's eng mit dem Top-Gehalt. Jetzt, wo du im Senior Level angekommen bist, kannst du langsam (langsam!) vom Learn zum Earn übergehen. Lernen solltest du immer noch weiterhin, die ersten Lorbeeren für die jahrelange Arbeit stehen nun aber für dich bereit. Glückwunsch!

Konkret heißt das: du hast (mindestens!) deine ersten 5 Jahre Karriere als Dev gemacht, gut gelernt, hast Verantwortung im Team und Projekt und deine Skills spiegeln obige Auflistung wider. Viele neigen dazu, sich zu überschätzen (Stichwort: „Dunning-Kruger“), einige wenige zum ganzen Gegenteil (Imposter-Syndrom). Um weder das eine noch das andere zu verkörpern, solltest du dein Team und Umfeld nach einer kompetenten, aber (wirklich!) kritischen Meinung zu deiner Arbeit und Schwachstellen fragen. Dann weißt du, wo du stehst und welche Baustellen du noch hast.

**Gehaltsspanne:** Kommen wir damit zum Teil, der Spaß macht: dein Gehalt! Die Spanne im Senior Level ist wirklich sehr groß! Allein von der Differenz zwischen 70k zu 120k passt locker ein schicker Premium Oberklasse Wagen. Besser deswegen: vorab verstehen, welche Faktoren wichtig sind, damit du planbar ans obere Ende kommst.

Faktoren für dein Gehalt als Senior

- Dein Arbeitgeber und dessen Geschäftsmodell und Margen
  - Kleine Firma oder wenig Marge -> wenig Luft nach oben
- Deine „Unverzichtbarkeit“ im Team und Projekt
  - Wie teuer wäre deine Kündigung? Ist deinem Team das bewusst?
- Deine anderen Optionen auf dem Markt
  - Du hast Angebote zu +20k vorliegen? Go for it!
  - Du findest keine Alternativen? Dig Deeper - oder lerne weiter!
- Deine Expertise am Markt selbst

- Wie gefragt ist dein Stack/Spezialisierung am Markt **wirklich?**

Damit geht vieles einher, wie konkret du gut verhandelst und wie du dein Gehaltspotenzial am besten auslotest, folgt am Kapitelende. Viel wichtiger: Du hast die letzten Jahre (hoffentlich) die richtige Basis für eine finanziell lukrative Zukunft gelegt! Rein aus Spaß & Nächstenliebe bist du kein Entwickler geworden und die Miete bezahlt sich auch (noch) nicht von selbst.

## **Beyond Senior: 100k+**

Du bist gut im Senior Game drin, dein Gehalt kratzt an den 100k ... aber war's das jetzt? Getrost dem Motto „*only the sky is the limit!*“

Vieles, was danach folgt, ist sehr individuell und von vielen Faktoren abhängig. Um dir dennoch einen guten Überblick zu geben, hier das Ganze in drei Kategorien unterteilt:

1. Dev Experte: primär Coden, kaum Personal (Lead, Staff, Principal)
2. Management: kein Code, nur Teamführung & Strategie (Manager, Head, VP)
3. C-Level, CTO: ein bunter Blumenstrauß an Aufgaben, je nach Firmengröße

... und als Extrapunkt danach: die Silicon Valley Experience, die gehaltlich mehr bietet, als jeder Normalverbraucher an Geld kaum ausgeben kann ... (hoffentlich).

Fakt ist: Alles, was jetzt kommt, ist Oberliga!

Die guten Plätze sind natürlich begehrt und heiß umkämpft, bieten dafür aber auch entsprechend viel! Eine Jobsuche in diesem Bereich kann durchaus einige Monate oder gar ein Jahr dauern. Das Gute für dich: Du brauchst nur **einen** Treffer landen, damit du viele Jahre gutes Geld verdienen kannst. (Im Gegensatz zum Freelancer- oder Unternehmer-Dasein, wo parallele Gigs keine Seltenheit sind).

Da du als erfahrener Senior schon einige Jahre Erfahrung hast und dein „Learn or Earn“ jetzt primär auf letzterem steht, ist es auch nicht unüblich, in den folgenden Positionen viele Jahre, teils bis zur Rente zu verbringen. Quasi der „Herbst“ deiner Karriere, der anbricht: etwas gediegener, der Grind der letzten Jahre hinter sich - dafür stetiger und berechenbarer. Auch finanziell.

Zwar dreht sich das Werk primär um den finanziellen Erfolg, Erfüllung, Sinnhaftigkeit und andere Faktoren sind natürlich trotzdem wichtig, nicht zuletzt der eigenen Glücklichkeit wegen. Gerade weil du in diesem Abschnitt deiner Karriere viel Zeit in einer Firma verbringen kannst, ist es umso wichtiger, dass diese Faktoren stimmen. Mal eben „*Augen zu und durch*“ auf 10+ Jahre wäre da denkbar schlecht.

### **Principal / Staff / Lead: 120k+**

In manchen größeren Firmen kommt nach dem Senior Level noch der „Principal“ Status on top. Oder eben die Ebene des Lead Developers, mit Personalverantwortung. Übersetzt heißt Principal nichts anderes als „Rektor“ aus dem schulischen - das sagt schon einiges aus. Quasi der Head-Senior, der alle Tätigkeiten und Skills des Senior Levels nochmal aufs nächste Level bringt. Mehr Verantwortung, mehr High Level, mehr Impact. Konkret gehen damit oft firmenübergreifende Best Practices einher.

Der Begriff des „Staff Engineer“ zielt auf ein ähnliches Level ab, ist gängiger in den USA und wird oft zwischen Lead/Principal angesiedelt - genormt ist das aber nicht.

Der „Lead Engineer“ wiederum sagt da schon mehr aus: Verantwortung für dein Team, oft pro Plattform und der erste Schritt zum Manager und etwas weg vom Coden.

Da die 3 Level je nach Firma und Kultur anders gehandhabt werden, ist es folglich auch schwer, konkrete Zahlen zu Gehältern zu finden oder Vergleichbarkeit zu schaffen. In

einem Startup kannst du Lead-Dev sein und trotzdem nur 60k verdienen, wohingegen du in einem Konzern für denselben Titel locker das Doppelte bis Dreifache verdienst.

Einige Firmen nutzen den Lead-Dev Titel auch als Ersatz für Engineering Manager (und umgekehrt), was das Thema kaum übersichtlicher oder transparenter macht - leider.

Kleiner Wermutstropfen für dich: du bist schon so weit oben in der Karriereleiter als Dev, dass dich all das nicht stören soll, da dein Gehalt eh schon Oberliga ist. Die Titel Principal und Staff wiederum kommen eher in größeren Firmen vor, sodass mit diesem Level auch die entsprechenden Aufgaben und Verantwortungen einhergehen und die Gehaltsbänder entsprechend gelegt sind.

**Gehaltsspanne:** Dein Spielraum ist mit ca. 120k-300k außer Frage sehr groß und obliegt nahezu immer der Firma selbst, wie weit das Band geht oder wo und unter welchen Kriterien du dort eingeordnet wirst. Meiner Erfahrung nach haben große Firmen, die diese Titel vergeben, aber auch (zum Glück!) konkrete KPIs und Performance Reviews, die das ganze sehr transparent machen.

## **Manager / Head / VP: 120k+**

Der Vollständigkeit halber müssen wir noch über 3 Rollen und Titel sprechen:

Engineering Manager, Head of Engineering und VP of Engineering.

Mit steigender Firmengröße und Headcount werden mehr Management-Ebenen im eigentlichen Entwicklungsprozess eingezogen. Die klassische BWL Regel ist, dass ein Mensch sich effektiv um 7-10 andere im Team kümmern kann - und da ist mehr dran, als du denkst oder mir damals lieb war. Natürlich gibt es CTOs mit hunderten Mitarbeitern in der Verantwortung, direktes Reporting sind aber selten mehr als zehn.

Konkret am Beispiel: oft sind pro Scrum Team 5 Devs am Coden. Die wiederum haben ihren verantwortlichen Engineering Manager (oder Lead Dev), der wiederum an den CTO berichtet, oder je nach Größe an einen Head of Engineering (oft pro Departement) oder eben an den VP, wenn der CTO nur noch strategisch tätig ist.

Das ganze mal in Kürze am Beispiel erklärt. :

- StartUp: 10 Devs -> Report direkt an den CTO
- ScaleUp: 100 Devs -> 5 Devs \* 5 EngManager \* 4 HeadOf -> CTO
- Mittelstand: 200 Devs -> 5 Devs \* 8 EngManager \* 5 HeadOf -> CTO
- Konzern: 1000 Devs -> 5 Devs \* 20 EngManager \* 10 HeadOf \* 1 VP -> CTO

Auch hier wieder: je nach Firma und Kultur anders gehandhabt. Beispielsweise ist es in Software Beratungen nicht selten so, dass die Hierarchie und Titel nochmals anders ist:  
Manager -> Senior Manager -> Associate Partner -> Partner

**Gehaltsspanne:** Selbiges für die Gehaltsbänder, ähnlich wie beim Principal Level: du bist schon recht weit oben angekommen, finde proaktiv heraus, welche Faktoren in deiner Firma und deinem Team für deinen Chef wichtig sind - und nutze dein Potenzial, diese Ziele zu erreichen!

## **FAAMNG Special: 180k+**

*"Welcome to the Valley!"* - könnte das Intro für diesen Abschnitt heißen. Womöglich hast du längst vom Silicon Valley und den (teils) absurd hohen Gehältern gehört - aber auch den damit verbundenen hohen Lebenskosten. Fakt ist: Für viele Entwickler ist „das Valley“ der Heilige Gral als Entwickler. Einerseits der finanziellen Möglichkeiten wegen, andererseits der Chance, bei den weltgrößten, namhaften Tech-Konzernen zu arbeiten, die unsere moderne Welt nicht unmaßgeblich bestimmen.

Woher das komisch anmutende Akronym „FAAMNG“ kommt?

Von den Anfangsbuchstaben der jeweiligen Big Tech Konzerne:

Facebook, Apple, Amazon, Microsoft, Netflix, Google.

Je nach Konstellation und Akronym (und davon gibt es einige!) fällt Microsoft raus, NVidia oder OpenAI rein, oder es wird gänzlich anders geordnet, wie MAANG oder sonstige Varianten. Einzig wichtig: Die großen Firmen im Valley mit Top-Gehältern.

Über die Lebenshaltungskosten in San Francisco und der Bay Area lässt sich natürlich streiten, aber darum soll's hier nicht gehen. Eher interessant ist die Frage, ob das Thema für dich interessant ist und du es dir vorstellen kannst - und wie der Weg dazu aussieht.

Als ich damals 2009 das erste Mal in San Francisco war und danach unbedingt ins Valley auswandern wollte (aus obigen Gründen und dem sonnigen Wetter wegen), stand genau das Thema auf dem Tisch. Vorab die größte Hürde: das Visum! Als Student kommst du recht einfach an ein J1, zum Geld verdienen bedarf es aber einem H1B, L1 oder O1.

Kurze Visums Erklärung dazu:

H1B: mit deinem Jobangebot/Vertrag bewirbt sich dein AG darauf, die Quote ist mäßig

L1: du hast einen AG außerhalb des US, der eine hiesige Niederlassung hat

O1: du hast eine „extraordinary ability“ und bist Weltklasse in dem, was du tust

Gängigerweise ist aber das H1B, damals war die Quote ca. 1:8 und erforderte einen akademischen Grad - weswegen ich noch „fix“ in 3 Jahren einen Bachelor nachgeholt habe. Die Quote und gesetzlichen Gegebenheiten ändern sich stetig - deswegen keine Garantie auf Gewähr. ;)

Jenseits der Visum-Modalitäten: Welche Skills brauchst du fürs Valley?

Die Antwort: sehr spezielle, oft stark theorielastige, in Data Science und Algorithmen. Warum? Weil die Mentalität vor Ort eine andere ist und eher auf mathematisches und

logisches Verständnis, samt Algorithmen und Datenstrukturen setzt, statt pragmatischer Entwicklung in einer Sprache und Stacks deiner Wahl.

In der Realität sieht das dann oft so aus, dass man an einem Interview-Tag vor Ort, oder einer Interview-Reihe remote meist 5-7 Stages durchlaufen muss, um sein Können zu beweisen. Themen wie Binary Tree Inversion, String Palindromes, Traveling Salesman Problem, Rekursion und Matrizen sind dabei die Grundlage für alles, was dir bevorsteht.

Praxisnah? Kaum. Dein Problem, um den Job zu bekommen? Leider wahr.

Mittlerweile gibt es unzählige Memes und Parodien deswegen:

The Job Interview: *„Invert this binary tree in log-n“*

First Day at the Job: *„Move this Button 10px to the right and make it green“*

Um dich darauf vorzubereiten, solltest du dir 3-6 Monate Zeit nehmen und den Klassiker „Cracking the Coding Interview“ als Buch durcharbeiten und bei Leetcode etliche solcher Probleme gelöst haben. Das steigert deine Chancen enorm, Pech mit einer schweren Frage kannst du dennoch immer haben.

Wichtig auch bei diesen Interviews ist die Herangehensweise: proaktiv mit dem Interviewer im Dialog den Austausch zum Approach besprechen, Hints heraushören und gleichzeitig iterativ sich an die Lösung heranarbeiten - und am Ende die Komplexität optimieren. Klingt komplex und ist es auch, dafür winkt am Ende aber einer der bestbezahltesten Jobs als Entwickler auf der ganzen Welt. Hoffentlich.

Was im Valley und bei den Big Tech Firmen ebenfalls etwas anders funktioniert, ist dein CV. In Deutschland strotzen viele vor lauter Prosatexten auf etlichen Seiten, im US ist der „Résumé“ oft nur 1 Seite und beinhaltet vor allem eins: Erfolgsgeschichten mit KPIs und Impact! Natürlich ist es nicht immer einfach, seine Dev-Arbeit so zu quantifizieren, wichtig für den Eintritt bei US-Konzernen ist es dennoch. Be Prepared!

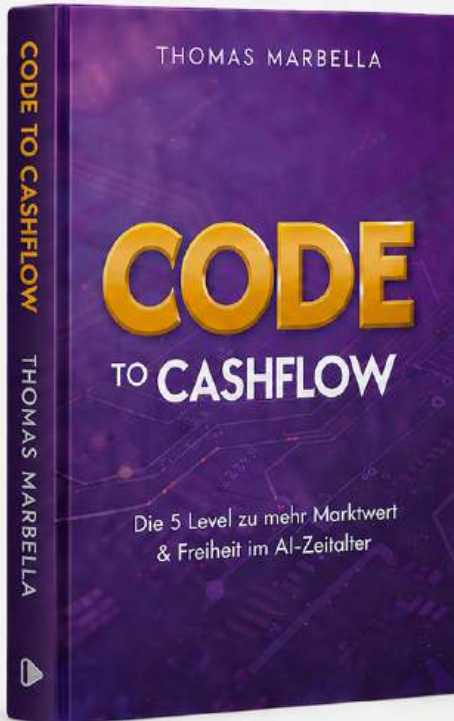
**Gehaltsspanne:** Kommen wir zum spannenden Teil: deinem Gehalt im Silicon Valley, bei FAAMNG Firmen! Anno 2012, nach meinem Studium war ein Seniorsalary ca. 120-150k, in Deutschland dabei gerade mal die Hälfte. Mittlerweile (2026) rangiert es grob zwischen 250k+ für einen Senior und geht steil nach oben. Je nach Level und Verantwortung sind 300k, 400k oder gar 500k+ möglich, wenn man Staff- und Principal-Engineer sowie Engineering Manager und andere Rollen berücksichtigt. Die Gehaltsstruktur selbst ist oft ein Mix aus Base-Salary, Bonus und Stocks/Equity Options und als Gesamtheit zu betrachten. Die Geschichten aus den frühen 2000ern, in denen zum Beispiel viele Microsoft Entwickler über Stock-Options damals schon Millionäre geworden sind, ist nur eine von vielen Anekdoten aus dem Bereich. (Inflationsbereinigt und nach heutiger Zeit gerechnet, kommt da einiges zusammen!)

Lifehack: den Job nach Valley Maßstäben bekommen und dann nach Deutschland zu gleichen Bezügen versetzen lassen. Ich habe 2 gute Freunde, die das geschafft haben. Mit etwas Glück kann das durchaus der Dev-Jackpot werden.

Kurz: Das Silicon Valley kann einiges bieten, auch wenn die Einstiegshürden hoch sind. Ob es das wert ist, kannst du gerne für dich entscheiden - mal einen Abstecher zu machen und den „Vibe des Valleys“ in Palo Alto & Co mitzunehmen, kann ich dir nur wärmstens empfehlen.

Weitere Senior Themen im Buch:

## Inhalte & Guides, Part II



### Level 1: Berufseinstieg

Learn or Earn  
Erfüllung im Job: individuelles Muss-Kriterium  
Theorie vs. Realität: Studium, Zertifikate & Abschlüsse  
Soft Skills, die den Unterschied machen  
Coding in 2026: The AI Hype is real  
Wo Anfangen?  
Branche, Stack, Spezialisierung?  
Vita Keypoints & CV Beispiele  
Github Profile & Open Source  
Bootcamps  
Praxis: der Einstieg  
TL:DR, Berufseinstieg

### Level 2: Angestellter

Gehalt vs. Perks vs. Arbeitsklima  
Gehaltsverhandlungen  
Junior Level: 40k+  
Mid Level: 50k+  
Senior Level: 70k+  
Beyond Senior: 100k+  
FAAMNG Special: 180k+  
TL:DR, Angestellter

jetzt auf  
**amazon**

### Die Leseprobe hat dir gefallen?

<https://www.amazon.de/Code-Cashflow-Marktwert-Freiheit-AI-Zeitalter/dp/3987552042/>

Bonus für PreOrder:

- Exklusiver Zugang zu unserer Community und -Jobmarkt
- Kostenlose 1:1 Beratungs Session mit uns (auf 50 limitiert)
- Dein Karriere und Netzwerk Boost - du musst den Weg nicht alleine gehen!