

Diese Leseprobe haben Sie beim
 edv-buchversand.de heruntergeladen.
Das Buch können Sie online in unserem
Shop bestellen.
[Hier zum Shop](#)

Einleitung

Clean Code ist seit vielen Jahren fester Bestandteil der agilen Softwareentwicklung. Code wird häufiger gelesen als geschrieben, daher ist es essenziell, dass Code leicht verständlich ist und seine Ziele klar erkennbar sind. Mit seinem Fokus auf Lesbarkeit und Verständlichkeit trägt Clean Code wesentlich zur Wartbarkeit und Testbarkeit von Programm-Code bei.

Dieses Buch ist eine Sammlung von Regeln, Richtlinien und Empfehlungen, die Ihren ABAP-Code »sauber« machen: lesbar, verständlich, wartbar und testbar. Es bespricht Entwurfsmuster und Sprachmuster ebenso wie generelle Abwägungen beim Softwaredesign. Auch Prozesse und organisatorische Feinheiten bei der Umsetzung von Clean Code in einem Unternehmen werden wir im Laufe des Buches besprechen.

An wen sich dieses Buch richtet

Dieses Buch haben wir vorrangig für ABAP-Entwicklerinnen und -Entwickler sowie Testerinnen und Tester mit einem Hang zur Entwicklung geschrieben. Sie finden hier viele verschiedene Code-Beispiele mit technischen Details.

Clean ABAP ist für Neulinge und erfahrene Programmiererinnen und Programmierer gleichermaßen geeignet. Je nach Kenntnisstand werden Sie möglicherweise von unterschiedlichen Themen profitieren oder sich mit bestimmten Bereichen näher auseinandersetzen wollen. Die meisten Ideen und Empfehlungen setzen jedoch keine großen Vorkenntnisse voraus, und Sie werden sie von der ersten Zeile Code an einsetzen können.

Wir ermutigen insbesondere auch Softwarearchitektinnen und -architekten, sich mit diesem Buch und Clean ABAP generell auseinanderzusetzen. Einige Themen in diesem Buch beschäftigen sich direkt mit Fragen, die typischerweise in den Aufgabenbereich der Architektinnen und Architekten fallen, z. B. Pakete und Objektorientierung. Selbst wenn Sie selbst nicht (mehr) programmieren, sollten Sie auch über die anderen Bereiche zumindest grundlegend Bescheid wissen, um sich mit Ihrem Team darüber austauschen zu können.

Wir haben in dieses Buch gezielt Kapitel aufgenommen, die Ihnen helfen sollen, mit Clean ABAP anzufangen, egal wann, egal in was für einer Art von Projekt und egal in welcher Art von Team Sie mit ABAP zu tun haben.

Wie dieses Buch aufgebaut ist

Dieses Buch beginnt mit einer Einführung in Clean ABAP und einer generellen Betrachtung der ABAP-Programmiersprache in Kapitel 1, »Einführung«, und Kapitel 2,

»Die Programmiersprache ABAP«. In Kapitel 3, »Klassen und Interfaces«, und Kapitel 4, »Methoden«, besprechen wir Methoden, mit denen Sie größere Entwicklungsteile wie Klassen, Interfaces und Methoden sauber strukturieren können. Kapitel 5, »Namen«, beschäftigt sich mit dem zentralen Thema der Namensgebung. Kapitel 6 bis Kapitel 11 gehen auf Detailbereiche ein, von Variablen über Literale bis zur Fehlerbehandlung. Unit Tests in ABAP werden im gleichnamigen Kapitel 12 genauer untersucht, gefolgt von Kapitel 13, »Pakete«. Kapitel 14, »Wie Sie Clean ABAP umsetzen«, beschäftigt sich schließlich eingehender damit, wie Sie Clean ABAP auf individueller, Team- oder Organisationsebene einführen können.

Die Kapitel 1 und 2 legen die Grundlage für die weiteren Erklärungen in diesem Buch, und Sie sollten sie daher nicht überspringen. Die Kapitel 3 und 4 gehören eng zueinander, da Klassen, Interfaces und Methoden natürlich ineinander übergehen. Die übrigen Kapitel sind unabhängiger, und Sie können sie problemlos in beliebiger Reihenfolge lesen oder auch als Referenz für später nutzen. Abschnitt 1.2, »Wie kann ich mit Clean ABAP anfangen?«, macht einen genaueren Vorschlag, welche Kapitel sich besonders für Anfängerinnen, Skeptiker oder Alleingelassene eignen. Das letzte Kapitel können Sie jederzeit heranziehen, wenn Sie Probleme mit der Umsetzung von Clean ABAP in Ihrem Arbeitskreis haben.

Hier eine genauere Inhaltsangabe der Kapitel:

■ **Kapitel 1, »Einführung«**

Dieses Kapitel führt in die Konzepte von Clean ABAP ein. Warum ist es wichtig, sauberen Code zu schreiben? Wie fange ich damit an? Sie erfahren auch, wie sich Clean ABAP zu anderen Leitfäden verhält und mit welchen Ressourcen Sie Ihr Verständnis von Clean ABAP vertiefen können.

■ **Kapitel 2, »Die Programmiersprache ABAP«**

Dieses Kapitel betrachtet die Programmiersprache ABAP an sich und ihre grundlegenden Besonderheiten. Sie erfahren mehr darüber, wie Sie mit Legacy-Code und Performance umgehen können. Allgemeine Fragen zu den Themen Objektorientierung, Sprachelementen und Entwurfsmustern werden geklärt.

■ **Kapitel 3, »Klassen und Interfaces«**

Dieses Kapitel beschreibt, wie sich Clean ABAP auf Klassen und Interfaces auswirkt. Wir erforschen, welche Teile der Objektorientierung Ihnen dabei helfen, ein sauberes Design zu erhalten, und welche Sie besser vermeiden sollten.

■ **Kapitel 4, »Methoden«**

Dieses Kapitel untersucht Methoden. Wie können Sie sie übersichtlich definieren? Wie kurz müssen sie sein, um verständlich zu bleiben? Wie können Sie Komplexität durch sorgsamem Umgang mit Parametern reduzieren?

■ **Kapitel 5, »Namen«**

Dieses Kapitel zeigt Ihnen, wie extrem sich gute Namen auf Lesbarkeit auswirken. Wir vergleichen gute mit schlechten Namen und geben Ihnen Ratschläge im Umgang mit Kürzeln und Abkürzungen.

■ **Kapitel 6, »Variablen und Literale«**

Dieses Kapitel bespricht, wie Variablen und Konstanten mit minimalem Aufwand für Übersichtlichkeit sorgen. Wir zeigen Ihnen auch, wie Code leichter verständlich wird, wenn Sie komplexe Literale und Ausdrücke in einfachere Bestandteile zerlegen.

■ **Kapitel 7, »Interne Tabellen«**

Dieses Kapitel wiederholt die Tabellenarten in ABAP und erklärt, welche wann geeignet sind. Anschließend vergleichen wir typische Muster, um mit Tabellen zu interagieren, und arbeiten heraus, welche Varianten davon besonders lesbar sind.

■ **Kapitel 8, »Kontrollfluss«**

Dieses Kapitel betrachtet, wie Zweige und alternative Ausführungspfade die Komplexität von Code verschlechtern können und wie Sie dem durch sorgsamem Umgang mit Bedingungen und Fallunterscheidungen entgegenwirken können.

■ **Kapitel 9, »Kommentare«**

Dieses Kapitel erforscht, wann sich Kommentare lohnen und wann nicht. Wir gehen darauf ein, wo Kommentare am besten stehen, wie ausführlich sie sein sollten und warum ihr Weglassen überraschend oft die beste Wahl ist.

■ **Kapitel 10, »Formatierung«**

Dieses Kapitel zeigt auf, wie Formatierung die Lesbarkeit von Code verbessert. Neben generellen Erwägungen zu Konsistenz und Lesen-vor-Schreiben schlagen wir Ihnen einige einfache Regeln vor, wie Sie zu gut lesbarem Code kommen können.

■ **Kapitel 11, »Fehlerbehandlung«**

Dieses Kapitel untersucht das Thema Fehlerbehandlung im Detail. Wir vergleichen Meldungen, Rückgabewerte und Ausnahmen, und vertiefen bei letzteren, wie Sie sie am besten aufbauen, auslösen und abfangen.

■ **Kapitel 12, »Unit Tests«**

Dieses Kapitel geht detailliert auf die Prinzipien und Techniken automatischer Tests ein. Wir beschreiben das ABAP Unit Test Framework und weitere Werkzeuge, die bei der Implementierung helfen. Dazu kommen grundlegende Empfehlungen, wie Sie Tests übersichtlich gestalten und langfristig am Leben halten können.

■ **Kapitel 13, »Pakete«**

Dieses Kapitel zeigt Ihnen, wie Sie ABAP-Code übersichtlich in Paketen organisieren können. Wir beschreiben die Grundlagen der Paketverwaltung und gehen näher auf Paketschnittstellen und -hierarchien ein.

■ Kapitel 14, »Wie Sie Clean ABAP umsetzen«

Dieses Kapitel beschreibt typische Herausforderungen wie Unverständnis, Skepsis und Ablehnung bei der Einführung von Clean ABAP und wie Sie damit umgehen können. Auch erfahrenere Teams und Clean-Code-Fans bekommen Tipps, die den Arbeitsalltag erleichtern können.

In hervorgehobenen Informationskästen finden Sie in diesem Buch Inhalte, die wissenswert und hilfreich sind, aber etwas außerhalb der eigentlichen Erläuterung stehen. Damit Sie diese Informationen sofort einordnen können, haben wir die Kästen mit entsprechenden Symbolen gekennzeichnet:



■ In Kästen, die mit diesem Symbol gekennzeichnet sind, finden Sie Informationen zu *weiterführenden Themen*.



■ Dieses Symbol weist Sie auf *Besonderheiten* hin, die Sie beachten sollten. Es *warnet Sie* außerdem vor häufig gemachten Fehlern oder Problemen, die auftreten können.



■ Die mit diesem Symbol gekennzeichneten *Tipps* geben Ihnen spezielle Empfehlungen aus unserer Projektpraxis, die Ihnen die Arbeit erleichtern können.

Wir wünschen Ihnen viel Vergnügen beim Lesen dieses Buches. Wir hoffen, dass wir Ihnen den Einstieg in das komplexe Thema Clean Code erleichtern können und dass Sie nach der Lektüre in der Lage sein werden, mit Clean Code zu arbeiten und sich selbstständig in diesem Gebiet weiterzubilden.

Dieses Buch werden Ihnen wichtige Konzepte und Design- und Sprachmuster vermitteln, die Ihren ABAP-Code sauberer machen. Sie werden erfahren, was »Sauberkeit« in diesem Kontext bedeutet, warum sie Ihnen wichtig sein sollte und wie Sie sie in Ihrer täglichen Arbeit einsetzen können.

Clean ABAP entwickelt sich beständig weiter, wie sich auch die Programmiersprache selbst und unser Verständnis davon weiterentwickelt. Wir werden Ihnen deswegen auch zeigen, wie das Hinterfragen von alten Gewohnheiten zu besserem Code führen kann und dass es manchmal besser ist, die Regeln zu ignorieren.

Clean ABAP ist auch für Teams gedacht. Deshalb möchten wir Ihnen auch zeigen, wie Sie die Konzepte in Ihrem Team und Ihren Projekten einführen, nämlich mit Respekt und greifbaren Beweisen.

Danksagungen

Lesbaren, testbaren und wartbaren Code zu schreiben, ist etwas, das allen professionellen Entwicklerinnen und Entwicklern am Herzen liegt. Danke an alle, die sich für dieses Thema stark machen, ihr Wissen mit anderen teilen und helfen, angehende Generationen von Entwicklerinnen und Entwicklern zu Profis auszubilden.

Dieses Buch wäre nicht möglich gewesen ohne die Unterstützung einer riesigen Gemeinschaft von Entwicklern und Entwicklerinnen. Mein Dank geht deshalb an alle, die zum Open-Source-Projekt Clean ABAP beitragen. Besonderer Dank geht an Thomas Hammer und Jakub Filak und die anderen direkt Mitwirkenden, sowie an Knut Stargardt dafür, dass er sich für das Thema eingesetzt hat, an die Basis für ihr kritisches Feedback, an die Rechtsabteilung für die schnelle Lizenzentscheidung und an Brian Bernard, der geholfen hat, das öffentliche Repository aufzubauen.

Auch ohne den Einsatz und die Unterstützung meiner früheren Teams der Abteilung Governance, Risk, and Compliance bei SAP wäre dieses Buch nicht möglich gewesen. Besonderen Dank möchte ich meinem ehemaligen Kollegen Florian Hoffmann für sein besonderes Engagement beim Clean-ABAP-Projekt aussprechen. Auch meinem früheren Manager Eckhardt Mentz gebührt Dank dafür, dass er mir die zeitliche Mitarbeit ermöglicht hat, und der zu jeder Zeit hinter unseren Ideen und unserer Arbeit stand.

Ein dickes Dankeschön geht auch an meine Mitautoren Florian Hoffmann, Rodrigo Jordão, Michel Martin, Anagha Ravinarayan und Kai Westerholz, die so viel Zeit und Arbeit in dieses Buch gesteckt haben. Ich schätze die vertrauensvolle und inspirierende Zusammenarbeit sehr.

Dank schließlich an Megan Fuerst aus dem Bostoner Lektorat von SAP PRESS, die uns bei unserem ersten Buchprojekt überhaupt unterstützt hat und uns mit allen Fragen geholfen hat, und auch an Simone Bechtold und Janina Schweitzer, die diese deutsche Übersetzung ermöglicht haben.

Klaus Häuptle

Clean ABAP ist eine Gemeinschaftsleistung. Viele haben dazu beigetragen, es auf den Weg zu bringen und wachsen zu lassen. Danke an alle, die geholfen haben, dieses Projekt bekannt zu machen, zu seinem Inhalt beigetragen und sich jeden Tag auf Neue für guten Code einsetzen.

Florian Hoffmann

Ich möchte Florian Hoffmann und Klaus Häuptle dafür danken, dass sie das Thema Clean ABAP bei SAP aufgebracht und weiterentwickelt haben und die lebhaft Open-Source-Gemeinschaft moderieren, die sich um Florians erste Texte gebildet hat. Clean ABAP und Clean Code allgemein haben mir als Themen immer am Herzen gelegen, und ihr Beispiel und ihre Führung haben mich dazu inspiriert, ebenfalls etwas zu diesem wichtigen Projekt beizutragen. Dieses Buch wäre ohne die Unterstützung von SAP selbst nicht möglich, mit einer Kultur, die uns zu ständigem Lernen ermutigt und uns herausfordert, jeden Tag ein bisschen besser zu werden. Ich wäre heute nicht da, wo ich bin, ohne die überwältigende Hilfe der Leute, die ich in meinem Leben am

meisten liebe: meine Eltern, meine Söhne und meine geliebte Frau Nara. Ihr inspiriert mich jeden Tag, die beste Version meiner selbst zu sein!

Rodrigo Jordão

Ich bin dankbar für die tollen Karrieremöglichkeiten, die ich in den vergangenen 17 Jahren bei SAP hatte. SAP macht sich für seine Mitarbeiter stark, spornt zum Lernen an und erlaubt uns, verschiedene Rollen auszuprobieren – Dinge, die uns allen helfen zu wachsen. Als Mitarbeitende bei SAP haben wir die Freiheit, eigene Interessengemeinschaften zu gründen, Wissen zu teilen und uns zu Veränderungen zu ermutigen. Ohne die Unternehmenskultur bei SAP hätten wir es nur schwer geschafft, ein Autorenteam zusammenzustellen.

Michel Martin

Ich bin meiner Familie (sowohl den Blutsverwandten als auch den Nichtblutsverwandten) dankbar, dass sie mich bei all meinen Unternehmungen unterstützen und mich ermutigen, Hindernisse zu überwinden. Ein dickes Dankeschön auch an meine Kolleginnen und Kollegen bei SAP, von denen ich jeden Tag Neues lerne. Ihre hilfsbereite und auf Zusammenarbeit ausgelegte Einstellung schätze ich sehr.

Vielen Dank an Florian Hoffmann und Klaus Häuptle für die Gelegenheit, zu diesem Buch beizutragen. Und danke an all meine Koautoren für ihren Einsatz und ihre Gesprächsbereitschaft. Ich habe eine Menge von euch allen gelernt und habe diese gemeinsame Reise genossen.

Anagha Ravinarayan

Nur dieses Buch zu lesen, wird nicht genügen, um Clean ABAP in Ihr Team einzuführen, aber es ist auf jeden Fall ein guter Startpunkt. Sie werden mit Ihrem Team über die Aspekte in diesem Buch diskutieren müssen, und Sie werden Zeit und mehrere Iterationen brauchen, um besser zu werden. An dieser Stelle möchte ich meinem Team danken, dass es gemeinsam mit mir diese Reise gemacht hat, und Tag für Tag ein bisschen besser wird, Erfahrungen teilt und offen ist für neue Ideen.

Kai Westerholz