

Clean Code Developer – Initiative für professionelle Softwareentwicklung

Anfang 2009 wurde von Ralf Westphal und Stefan Lieser die Initiative »Clean Code Developer (CCD)« ins Leben gerufen. Die Idee dazu basiert auf dem Buch »Clean Code« von Robert

C. Martin¹. Da beide aus dem Lager der .NET Entwickler kommen, erfreut sich CCD dort einer großen Beliebtheit und schlägt sich inzwischen auch in den Medien nieder.

VKSI: Stefan, ich freue mich dieses Interview mit Dir für den VKSI machen zu dürfen. CCD hat meine Arbeitsweise im vergangenen Jahr stark beeinflusst. Für mich persönlich stellt es einen Baustein auf dem Weg in Richtung des Karlsruher Manifests² des VKSI dar. Kannst Du bitte den Lesern, welche noch nicht mit Eurer Initiative vertraut sind, kurz vorstellen, was die wesentlichen Ziele bei CCD sind?

Stefan Lieser: Am Anfang der Initiative stand für Ralf Westphal und mich die Frage, was einen professionellen Software Entwickler ausmacht. So haben wir begonnen, Bausteine zusammen zu tragen, von denen wir glauben, dass sie zu innerer Qualität von Software beitragen. Als wir ungefähr 40 Prinzipien und Praktiken zusammengetragen hatten, kamen wir darauf, diese Bausteine unter einem Wertesystem zusammenzufassen. Zusammengefasst ist also das Ziel der Clean Code Developer Initiative, die innere Qualität von Software zu verbessern, indem die Entwicklung von Software professionalisiert wird.

VKSI: Es gibt schon einige andere Ansätze, die Qualität in der Softwareentwicklung zu verbessern. Was ist der Unterschied bei CCD?

Stefan Lieser: Laut ISO 9000 wird mit Qualität der Grad bezeichnet, mit dem Merkmale eines Produktes Anforderungen erfüllen. Ein Merkmal von Software ist beispielsweise ihre Sicherheit. Ohne Anforderungen an dieses Merkmal zu stellen, kann man jedoch keine Aussage über die Qualität in diesem speziellen Merkmal treffen. Folglich müssen zur Betrachtung der Qualität nicht nur Merkmale definiert werden, sondern auch die jeweiligen Anforderungen an diese Merkmale. Daraus entsteht dann die zentrale Frage: wer definiert die Anforderungen an Software? Die reflexartige Antwort lautet natürlich: König Kunde.

Diese Antwort ist jedoch nicht ausreichend. Würden die Anforderungen ausschließlich von Kunden definiert, wären nur Qualitätsmerkmale zu betrachten, die dem Kunden vertraut sind. Daher müssen den völlig berechtigten Anforderungen der Kunden die Anforderungen derjenigen hinzugefügt werden, die hinter die Kulissen schauen. Noch deutlicher wird das Problem, wenn man an Ergänzungen oder Änderungen von Features denkt. Natürlich geht der Kunde davon aus, dass er zu einem späteren Zeitpunkt weitere Features einfordern kann. Dass wir als Softwareentwickler dafür jedoch Vorsorge tragen müssen, kann und soll ihn nicht interessieren. Dies ist unsere

Verantwortlichkeit, weil nur wir die Details der Softwareentwicklung kennen.

Die Clean Code Developer Initiative ist damit ein Gegengewicht zu den Anforderungen der Kunden. Unser Fokus liegt auf der inneren Qualität. Wir wollen versuchen, einen Ausgleich zu schaffen zwischen den Anforderungen der Kunden und solchen, die sich nur aus der Kenntnis der inneren Details ergeben können.

VKSI: Im Zusammenhang mit CCD wird von »professioneller« Softwareentwicklung gesprochen. Was unter scheidet die CCD Definition von »Professionell« von der etablierten Vorstellung?

Stefan Lieser: Für uns verfügt ein professioneller Softwareentwickler sowohl über ein Wissensgebäude als auch über ein Wertesystem. Dass Softwareentwickler über ein gewisses Maß an Kenntnissen verfügen müssen, dürfte allgemein anerkannt sein. Ein Wertesystem konnten wir allerdings bislang nicht ausmachen. Das führte uns dazu, die vier Werte Evolvierbarkeit, Korrektheit, Produktionseffizienz und Reflexion zu formulieren. Damit weichen wir meiner Einschätzung nach jedoch mit unserem Begriff von „Professionell“ nicht von etablierten Vorstellungen ab. Unter Softwareentwicklern etabliert sich langsam die Ansicht, dass automatisiertes Testen eine nützliche Praktik ist. Aber eben nicht zum Selbstzweck, sondern weil damit die Werte Evolvierbarkeit, Korrektheit, Produktionseffizienz und Reflexion positiv beeinflusst werden.

Inhaltlich steht natürlich die Frage im Raum, ob denn ausgerechnet das von uns vorgeschlagene Wertesystem ausreicht bzw. zum Ziel der höheren inneren Qualität führt. Dieser Diskussion stellen wir uns gerne und fordern sie sogar ein. An dieser Stelle sei auf die Diskussionsmöglichkeiten bei google (<http://groups.google.de/group/clean-code-developer>) und xing (<https://www.xing.com/net/ccd>) hingewiesen.

VKSI: Bei Entwicklerveranstaltungen trifft man immer mehr Entwickler mit CCD Armbändern an. Was hat es mit den Bändern auf sich?

Stefan Lieser: Die Armbänder können mehrere Funktionen übernehmen. Zum einen dienen sie dazu, denjenigen der eines trägt daran zu erinnern, dass er sich mit den Bausteinen der Clean Code Developer Initiative auseinandersetzen will. Bei über 40 Bausteinen ist es nicht damit getan, sich mal ein paar Tage damit zu beschäftigen. Es erfordert in vielerlei Hinsicht ein

¹ Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin), Prentice Hall International, 2008 ISBN 0-13-235088-2.

² VKSI Karlsruher Manifest für Software Engineering: <http://www.vksi.de/karlsruher-manifest.html>



Umdenken und Lernen. Solche Veränderungsprozesse erfordern Disziplin und da kann es helfen, wenn man durch ein Armband immer wieder daran erinnert wird, sich mit den Bausteinen und Werten auseinander zu setzen. An der Stelle möchte ich betonen, dass wir das Armband als eine Möglichkeit unter vielen verstehen. Wer sich durch ein Knoten im Taschentuch erinnert, verzichtet eben auf das Armband.

Da es sehr viele Prinzipien und Praktiken zu erlernen und umzusetzen gilt, haben wir sie in fünf sogenannte Grade eingeteilt. Dabei haben wir darauf geachtet, die Bausteine in eine sinnvolle Reihenfolge zu bringen. Ferner haben wir jeden Grad mit einer Farbe versehen und haben uns dabei am Regenbogen orientiert. Die Kernidee der Grade ist, sich für jeweils etwa drei Wochen schwerpunktmäßig mit den Bausteinen eines Grades vertraut zu machen. Danach wechselt man zum nächsten Grad. Irgendwann muss man dann mal alle Bausteine in ihrer Gesamtheit berücksichtigen, dazu dient der weiße Grad. Er vereinigt sozusagen alle Grade in sich, so wie Weiß alle Farben enthält. Nachdem man dies ebenfalls für einige Zeit getan hat, sollte man übrigens wieder von vorne beginnen. Insofern ist der Begriff Grad vielleicht etwas irreführend. Wir wollen damit nicht implizieren, dass jemand, der sich mit dem gelben Grad auseinandersetzt besser ist, als jemand der am roten Grad arbeitet. Unsere Horrorvorstellung wäre, wenn Personalabteilungen plötzlich auf die Idee kämen, Entwickler zu suchen die mindesten den gelben Grad absolviert haben. Die Grade dienen nur der Einteilung der Bausteine in handhabbare Teilmengen.

Eine weitere Funktion der Armbänder ist das Marketing bzw. die Kennzeichnung. Wir möchten die Idee der Clean Code Developer Initiative natürlich verbreiten, weil wir glauben, dass die gesamte Branche sich langfristig auf ein gemeinsames Wertesystem verständigen sollte. Ob das am Ende das Wertesystem der Clean Code Developer Initiative ist, sei dahin gestellt, wir möchten gerne auf das Problem mangelnder innerer Qualität hinweisen und eine mögliche Lösung anbieten.

Wer nun mit den Armbändern nichts anfangen kann, lässt sie halt weg. Auch das ist uns wichtig zu betonen.

VKS: Das hört sich nach viel Aufwand für den Entwickler an. Welchen Vorteil hat er in der täglichen Arbeit?

Stefan Lieser: Das Anwenden der Clean Code Developer Prinzipien und Praktiken führt meiner Beobachtung nach zu größerer Ruhe und Zufriedenheit. Es gibt ein sicheres Gefühl zu wissen, dass die Software die man gerade erstellt, für zukünftige Anforderungen gerüstet ist. Ferner sind die Rückläufer durch Fehler geringer. All dies wirkt sich positiv auf den Alltag des Entwicklers aus, so dass sich der Aufwand, sich mit den Bausteinen auseinanderzusetzen, in mehrfacher Hinsicht lohnt.

VKS: Vom Management hört man entsprechend die Frage nach den höheren Kosten. Was »kostet« CCD Deiner Meinung nach?

Stefan Lieser: Meine Gegenfrage lautet: was kostet es, wenn man CCD oder vergleichbare Konzepte nicht einsetzt? Die Kosten, die nach der eigentlichen Entwicklung der Software anfallen, werden in aller Regel nicht näher betrachtet. Ich habe Teams →

Clean Code Developer – Interview mit Stefan Lieser



→ erlebt, in denen es ein Verhältnis von Supportmitarbeitern zu Entwicklern von 1:1 gab. Und jeder Softwareentwickler kann von einem Projekt erzählen, bei dem nach einiger Zeit nur noch Neuschreiben in Frage kam. Das sind Kosten, die in keinem Verhältnis stehen zu den Kosten, die mit der Beschäftigung mit CCD oder vergleichbarem anfallen.

VKSI: Welche Änderungen hat die Einführung von CCD für das Management? Bzw. kann das Management die Einführung von CCD unterstützen?

Stefan Lieser: Das Management kann ruhiger schlafen. Schließlich stehen auch sie in der Schusslinie, wenn die innere Qualität der Software zu wünschen übrig lässt. Daher sollte eine angemessene innere Qualität im Interesse aller sein. Managementunterstützung bedeutet hier zuerst einmal, das absolute Vertrauen, dass die Entwickler nach Kräften versuchen, das Beste zu geben. Dies ist vor allem in der Phase wichtig, in der die Weiterentwicklung an den Produkten langsamer voranschreitet. Die Einführung von CCD ist ein Veränderungsprozess. Solche Veränderungen bringen es mit sich, dass einige lieb gewordene Angewohnheiten in Frage gestellt werden, dass eine Zeit lang Unruhe herrscht. Gerade in dieser Phase muss das Management zu den Entwicklern stehen. Und am Ende muss natürlich Zeit sein, sich mit CCD auseinanderzusetzen. Leider wird die Weiterbildung von Entwicklern oft als persönliches „Problem“ der Mitarbeiter angesehen. Auch hier kann das Management also unterstützen.

VKSI: Was kannst Du einem Entwickler ans Herz legen, wenn er jetzt gleich mit CCD beginnen möchte? Welche Voraussetzungen gibt es?

Stefan Lieser: Zunächst sollte sich ein interessierter Entwickler einen Überblick über die CCD Initiative verschaffen. Dazu kann er sich z.B. auf clean-code-developer.de umschauen, dort gibt es auch zahlreiche Literaturhinweise. Vermutlich werden nicht alle Bausteine neu für ihn sein, das relativiert den Eindruck, dass es eine Menge neuesNeues zu lernen gilt. Tja und dann hilft nur, anfangen! Sich mit den Bausteinen im ersten, dem roten Grad, vertraut machen und sie über einen Zeitraum von drei Wochen im Blick behalten. Wer sich mit anderen austauschen möchte, kann sich im Kollegenkreis umschauen oder in der örtlichen Usergroup nachhören ob es dort CCD Interessierte gibt. Und natürlich kann in den Foren diskutiert werden. Ich wünsche viel Erfolg!

Stefan Lieser ist freiberuflicher Berater/Trainer/ Autor aus Leidenschaft. Sein Interesse gilt den agilen Methoden der Softwareentwicklung und er sucht ständig nach Verbesserung und neuen Wegen. Bereits während des Informatikstudiums setzte er seinen Schwerpunkt im Bereich Software Technik. Über die Jahre hat er sich intensiv mit Patterns und Principles auseinandergesetzt. Dies spiegelt sich auch in seinen zahlreichen Vorträgen wider.



Das Interview führt **Ralf Schoch**, Mitglied des Präsidiums des VKSI und Mitglied der .NET User Group Karlsruhe. Er ist freiberuflicher Softwareentwickler im .NET Umfeld und Certified Scrum Product Owner. Sein Schwerpunkt liegt im Bereich der Entwicklung von Webapplikationen, der Verbesserung der Entwicklungsprozesse und der Code Qualität.

