# Einführung Version 1.0

## Problemstellung

Die Probleme moderner Software sind sehr vielseitig. Der Kunde hat einen Fehler gefunden und dieser sollte innerhalb kürzester Zeit behoben werden. Ein weiterer Kunde möchte eine Einstellung zum ausblenden zahlreicher Details in einem Report haben. Der dritte Kunde beschwert sich, dass der Dialog welchen er mehrmals am Tag öffnet viel zu lange lädt und er dadurch sehr viel Zeit verliert. Bei all diesen Problemen gibt es einen gemeinsamen Nenner. Die Lösung dieser Probleme erfordert eine Änderung der Codebasis. Bei Änderungen der aktuellen Codebasis gibt es eine zentrale Frage.

* Wo genau muss die Änderung stattfinden?

Diese lässt sich meist nicht so leicht beantworten.

Vor allem Produkte welche über mehrere Jahre hinweg von verschiedenen Personen entwickelt worden sind, leiden meist an ähnlichen Problem. Der Code ist sehr schwer zu Lesen. Die Folgen die durch diese Unlesbarkeit entstehen sind sehr weitreichend und dürfen nicht unterschätzt werden. In vielen Fällen hat unlesbarer Code schon zum „Tod“ von ganzen Programmen geführt, da die Weiterentwicklung unmöglich wurde.

## Die Probleme moderner Software

Die Entwicklung von Software ist sehr ähnlich wie in der folgenden Grafik erläutert.

Dabei wird die Codebasis immer mehr erweitert und es werden immer weitere Features hinzugefügt. Oft werden auch Fehler behoben und dadurch verändert sich die Codebasis ebenfalls. An den meisten Softwareprojekten arbeiten auch zahlreiche verschiedene Entwickler und dadurch ergibt sich ein weiterer Faktor, der zur Unterschiedlichkeit der Codebasis beiträgt.

Aus diesen Voraussetzungen welche für nahezu jede Software, welche über einen längeren Zeitraum unterstützt und weiterentwickelt wird, zutreffen ergeben sich folgende Probleme:

* Es werden immer mehr Features hinzugefügt wodurch der Code immer mehr wird

## Lösungsvorschlag – Clean Code Development

Clean Code Development (CCD) ist eine Bewegung die in den letzten Jahren immer mehr an Wichtigkeit gewinnt. Der Ausgangspunkt dieser Bewegung war das Buch „Clean Coder“ von Robert C. Martin, in welchem er seine Ansichten zum Thema Clean Code erläutert. Diese Arbeit wird des Öfteren auf Teile seines Buches verweisen. Im deutschsprachigen Raum kümmern sich vor allem Herr Ralf Westphal und Herr Stefan Lieser um die Verbreitung dieser Strömung. Es gibt dabei unter anderem eine Seite mit Grundsätzen und grundlegenden Erklärungen zum Thema Clean Code Development (<http://clean-code-developer.com/>), eine XING – Gruppe in der momentan 4245 (Stand November 2015) mehr oder weniger aktive Mitglieder sind und zahlreiche Podiumsdiskussionen und Workshops zu diesem Thema.

Eine Eigenschaft der Codebasis von Programmen ist, dass der Code viel öfters gelesen als geschrieben wird. Meist wird er das auch von mehreren verschiedenen Personen. Damit es dabei zu möglichst wenig Missverständnissen kommt, der Code gut lesbar ist und dies vor allem nach mehreren Monaten oder sogar Jahren müssen gewisse Verhaltensregeln beachtet werden. Robert C. Martin beschreibt diese Verhaltensregeln in seinem bereits oben genannten Buch „Clean Coder“. In dieser Arbeit sollten anhand der Verhaltensregeln Codeteile aus Open-Source Frameworks so umgestaltet werden, dass sie den Verhaltensregeln des CCD entsprechen. Dabei sollten die beiden Varianten immer verglichen werden. Es sollten aber auch Beispiele gezeigt werden bei denen die Verhaltensregeln eingehalten werden.

Die CCD – Bewegung ist eine in den letzten Jahren immer stärker werdende Strömung die sich vor allem dafür einsetzt, Code so zu gestalten, dass die Aspekte Wartbarkeit, Lesbarkeit und die sich dadurch ergebende Zukunftssicherheit gegeben sind. Dadurch wurden bereits zahlreiche Publikationen, Artikel und auch Bücher veröffentlicht in denen diese Praktiken beschrieben werden. Es geht beim CCD in erster Linie um die Qualität des Codes hinsichtlich Lesbarkeit. Dieses Thema sollte in meiner Bachelorarbeit näher behandelt werden. Es wird dabei nicht auf das Thema Architektur eingegangen.