

## Aufgabe 1

Was halten Sie davon?

Das Glossar ist durch die verdrehte Denkweise und den Sarkasmus sehr unterhaltsam. An einigen Stellen ist es jedoch sehr speziell und an diesen Stellen bietet es Lesern, die allgemeinere Informationen zu schlechtem Code bzw. Refactoring suchen, nicht wirklich das, was sie vielleicht suchen. Die vielen allgemeingültigen Tipps und Tricks finde ich jedoch sehr hilfreich.

Präsentieren Sie 5 wichtige, lehrreiche Hinweise/Fehler/Fallstricke

1. Benennung von Variablen, Funktionen usw.: Hier gibt es viele Beispiele, die einem Programmierer das Leben schwer machen können. Z.B. die Verwendung von nichtsprechenden Variablennamen, Verwendung von Namen mit nur einem Zeichen, Verwendung von anderen Bezeichnern z.B. bei Laufvariablen in Schleifen usw.
2. Falsche Dokumentation vs. keine Dokumentation: Manchmal ist dokumentierter Code schlechter als nicht-dokumentierter, weil z.B. bei einem Update vom Code, die dazugehörige Dokumentation nicht angepasst wird. Oder weil die Dinge dokumentiert werden, die sowieso klar sind, nicht-durchschaubare Algorithmen werden dann aber z.B. nicht dokumentiert usw.
3. Programm Design: Beim Programm Design können ebenfalls viele Fehler gemacht werden. Ein Punkt wäre z.B. wenn eine Methode oder Funktion zu viele Dinge auf einmal macht. Es wäre besser wenn jede Methode/Funktion immer nur eine einzige Aufgabe erledigt. Soll z.B. das Quadrat einer Zahl berechnet werden soll nur die Zahl zurückgegeben werden aber nicht in der Funktion ausgegeben werden.
4. Testen: Man sollte seine Programme immer testen. Vor allem für mich als Web-Entwickler ist dieser Punkt sehr wichtig, da es z.B. bei Skriptsprachen keinen Compiler gibt, der gewisse Fehler abfängt. Es sollte immer verschiedene Testfälle eingebaut werden. Dabei ist nicht nur die korrekte Ausführung eines Programms wichtig, sondern auch dessen Performance.
5. Ein wichtiger Punkt ist, dass man immer so programmieren sollte, man spätere Änderungen nur an einer Stelle durchführen muss. Das heißt mit Konstanten, Funktionen usw. arbeiten.
6. Ein für mich zusätzlich sehr wichtiger Punkt ist der, dass man grobe Änderungen an der API (Funktionen die wegfallen, Veränderung von Zugriffen, usw.) vermeiden sollte. Bzw. dass nicht laufend Dinge verändert werden. Ein bekanntes Beispiel hierfür wäre Facebook, die leider laufend etwas ändern und so manchen Code „zerstören“.