

1 Zielkonflikte in der Softwareentwicklung

1. Nein, die vier aufgeführten Ziele sind nicht alle gleichermaßen erreichbar, da sie sich auch gegenseitig bedingen. Wenn man nämlich z.B. versucht die Qualität zu erhöhen, dann zieht das logischerweise nach sich, dass der Zeitaufwand steigt und auch die Kosten, da z.B. vermehrt getestet werden muss und man im Zuge dessen womöglich das Projekt auf eine lange Testphase anlegt. Zusätzlich kann es sein, dass der Umfang auch größer wird, da man zur Gewährung der Qualität eventuell noch mehr Funktionalität umsetzen muss.
2. Die *Qualität* hat Vorrang bei Projekten, die z.B. sicherheitsrelevante Software umsetzen sollen. So sollte Software, die in einem Flughafen-Tower eingesetzt wird, natürlich 100% funktionieren.
Wenn man hingegen ein Profi-Tool zum Videoschnitt programmieren will, dann sollte dies durch den großen *Funktionsumfang* brillieren, und hier sollte auch der Hauptaugenmerk drauf gelegt werden.
Software, die aus irgendeinem Grunde schnell auf den Markt gebraucht werden soll, z.B. um eventuellen Konkurrenzprodukten zuvor zu kommen, sollte in der Entwicklung natürlich hier den Fokus drauf legen, dass der *Zeitaufwand* möglichst niedrig ist.
Schlussendlich sollte man versuchen die Kosten niedrig zu halten, wenn man womöglich eine Software entwickelt, die auch zu einem niedrigen Preis angeboten werden soll. Denkbar ist dies bei kleinen Helfertools.
3. Wenn man mehr Entwickler zu Verfügung hat, dann sinkt die Entwicklungszeit pro Kopf natürlich. Man darf aber auch nicht vergessen, dass bei ansteigender Entwicklerzahl Zeit für die Koordination, Meetings usw. gebraucht wird, weswegen zu viele Entwickler eventuell sogar mehr Zeit benötigen könnten.
Konkret mit den gegebenen Werten könnte man schätzen, dass zwei Entwickler noch ziemlich gut die Entwicklungszeit halbieren könnten, da zwei Personen noch gut zu koordinieren sind. Bei fünf Entwicklern könnte es schon etwas problematischer werden, hier benötigt man sicherlich nur gut 4 Monate für das ganze Projekt, also ein Drittel der Zeit, wie wenn nur einer es umsetzt. Aber bei 150 Entwicklern braucht man wohl wieder länger, eventuell sogar ein Jahr, aufgrund der oben erwähnten Koordinationen.

2 Wiederholung von Begriffen

1. **Klasse** ist ein Konstrukt, in dem man gleichartige Attribute und Methoden über einen bestimmten Sachverhalt zusammenfasst.
Vererbung ist das Prinzip, dass man Klassen von anderen Klassen ableiten kann um somit den Funktionsumfang sinnvoll zu erweitern oder zu spezifizieren.
Attribut ist eine Eigenschaft, ein Merkmal einer Klasse, das durch eine Variablen dargestellt wird.
Operation ist ein Befehl in einer Methode.

Methode ist ein Algorithmus, eine Funktion, die man zum Arbeiten auf einer Klasse und deren Objekte aufrufen kann.

Exemplar

Objekt ist eine Instanz einer Klasse, also das bei der Ausführung des Programms erstellte Abbild der Klasse, die benutzt wird

Bibliothek Eine Sammlung von Funktionalitäten, die z.B. in eine Java-Klasse eingebunden (importiert) werden können.

Spezifikation ist eine formale Definition einer Software.

Implementierung das ist die Umsetzung einer Problemlösung in Code, dabei muss einer bestimmten Spezifikation gefolgt werden.

Verifikation ist das formale oder auch mathematische Beweisen, dass die Spezifikationen eines Programms korrekt umgesetzt worden sind.

Geheimnisprinzip ist das Prinzip, dass man Daten bzw. Funktionalitäten nach außen hin verbirgt, so dass kein direkter Zugriff hierauf möglich ist. Der Zugriff kann so nur durch definierte Schnittstellen erfolgen.

2. Durch die oben genannten Begriffe wird nicht der Bereich des Managements taxiert und auch nicht die Kosten-Nutzen-Analyse abgedeckt. Und zwar deckt der Großteil der Begriffe die Phase der Implementierung ab, einige wie Attribut und Methode oder Klasse sind auch Vokabeln, die im Entwurf eine Rolle spielen. Zusätzlich wird durch den Begriff Verifikation auch auf die Qualitätssicherung und das Testen hingewiesen, wo wie das Wort schon sagt, die Lösung verifiziert und validiert werden muss. Auf die Anforderungen wird kurz mit dem Begriff Spezifikation eingegangen, bzw. kann man ihn hier einordnen, da es bei den Anforderungsdefinitionen darum geht, zu klären was umgesetzt werden soll.
3. Eine Klasse definiert wie Objekte aussehen sollen, wie sie sich verhalten sollen etc. wohingegen eine Menge von Objekten sich auf eine oder mehrere Klassen zurückführen lassen, dessen Typ sie besitzen. Im Klartext heißt dass, dass in einer Klasse quasi die Bauanleitung für ein Objekt steht und das Objekt das fertige Bauteil ist. Die Klasse definiert die Parameter, die das Objekt später mal haben soll und welche Funktionalität es bereitstellen soll und das Objekt ist das konkrete Abbild, welches einen Zustand und das gewünschte Verhalten aufzeigt.