

## ACDK als Methode

### Abstrakt

Als wir ACDK entworfen haben, verfolgten wir eine bestimmte Idee, wie Softwareentwicklung auszusehen hat. ACDK soll auch als Angebot verstanden werden, die Softwareentwicklung im heutigen Arbeits-, Projekt- und Produktzusammenhang zu verstehen.

### Autor

Roger René Kommer

### Version

1.0

### Datum

2000-11-02

### Seiten

3

### Kontakt

Ingenieurbüro Kommer, artefaktur  
Dörnbergstrasse 11  
34119 Kassel  
Tel.: 0561-780-150  
Fax.: 0561-780-135  
<http://www.artefaktur.com>  
<mailto:kommer@artefaktur.com>

## **ACDK als Methode**

Als wir ACDK entworfen haben, verfolgten wir eine bestimmte Idee, wie Softwareentwicklung auszusehen hat. ACDK soll auch als Angebot verstanden werden, die Softwareentwicklung im heutigen Arbeits-, Projekt- und Produktzusammenhang zu verstehen.

## **Native Programming**

Musiker denken in Partituren, Architekten in Aufrissplänen und Programmierer in den Programmiersprachen, mit denen sie täglich entwickeln. ACDK unterstützt die Entwickler mit einem Sprachkonzept, Standards, Bibliotheken und Werkzeugen, die die Sicht des Programmierers mit den Projektbedingungen zusammenführen.

## **Softwareentwicklung als Ingenieurleistung**

Durch die Integration von Standards, Methoden und Werkzeugen in ein Projektenvironment wird Softwareentwicklung als Ingenieurleistung möglich und durch Methoden des Projektmanagements integrativ verwaltbar.

Das Projektenvironment besteht aus

- einem homogenen Objektmodell
- Komponenten & Bibliotheken
- Test- und Debugverfahren (Selbsttests)
- Dokumentationssystem für Konzepte, Spezifikationen, Klassen, Implementationen und Module
- Konfigurations-, Release- und Versionsmanagement
- Tasktracking, Projektmanagement

## **Extreme Programming**

Grob 80% des Programmcodes werden im Laufe eines Projekt- und Produktlebens wieder verworfen und ersetzt. Traditionelle Ansätze des Software-Engineering versuchen, diesen Faktor soweit wie möglich zu reduzieren.

Die oft vergessene Kehrseite des Paradigma des wiederverwendbaren Codes ist das des wegwerfbaren Codes. Programmteile müssen auch weggeworfen werden können, ohne dass das Gesamtsystem gefährdet werden würde (minimalinvasiver Eingriff).

Programme, von denen nie Teile weggeworfen und reimplementiert wurden, sind in der Regel schlecht getestet und gewartet - es hat sich einfach niemand getraut, den halbwegs laufenden Code anzufassen. ACDK versucht sowohl das Wiederverwenden als auch das Wegwerfen zu erleichtern, indem auf der einen Seite ein tragfähiges entkoppelndes

Komponentenmodell zur Verfügung gestellt wird, und auf der anderen Seite ein Projektenvironment geboten wird, in dem das Risiko beim Wegwerfen von Codes minimiert wird. Durch die Verbindung von Quellcode-Dokumentation und zugeordnete Selbsttest-Programme lassen sich auch bei größeren Änderungen die Programme auf ihre ursprünglich spezifizierte Funktionsweise testen.

## **Time to Market**

Software wird in aller Regel unter Zeit- und Personalknappheit entwickelt. Der Kunde will Software-Produkte "Time to Market" entwickelt, im Industriebereich sollen Inbetriebnahmezeiten reduziert werden, Produkte sollen auch während des Betriebes nachgebessert werden.

Dies ist alles nur dann möglich, wenn das klassische Wasserfallmodell der Softwareentwicklung aufgelöst wird. ACDK unterstützt eine Entwicklung, bei der ein fließender Übergang von der Spezifikation der Schnittstellen, der Entwicklung eines Prototypen, der Implementation und der Inbetriebnahme möglich wird.

## **Projekt- und Produktgeschäft**

ACDK macht in seinen Methoden keinen Unterschied zwischen Projekt- und Produktgeschäft, vielmehr werden von beiden die Tugenden in einer Software-Engineeringmethode zusammengefasst.

Durch eine Unterstützung des Projektenvironment werden sehr kurze Release-Zyklen möglich, so dass zu jeder Zeit im Projekt eine Version verfügbar ist, an der man schon die Grundzüge und Qualität des Endproduktes ermessen kann. Tendenziell gibt es keinen Unterschied mehr zwischen Prototyp und Produkt. Durch die weitgehende Java-Kompatibilität wird das Projekt/Produkt auch flexibel gegenüber einem technologischen Wechsel.

Durch eine Auskoppelung von Produktlinien aus dem Projektgeschäft auf der einen Seite und das Einbinden vorgefertigter Produktkomponenten in die Projektsoftware auf der anderen Seite bekommt man einen positiven Effekt auf alle Parameter der Softwarequalität und steigert die Marktfähigkeit der Softwareentwicklung.