

---

# 1 Einleitung

Ein einheitliches Verständnis für die »Produktivität in der Softwareentwicklung« ist in der Softwareindustrie heute nicht sichtbar. Die Produktivität steht noch nicht im Zentrum der Betrachtung des Software Engineering und des Projektmanagements. Das betrifft einerseits die akademische Forschung, aber noch viel mehr die Projekt- und Entwicklungspraxis in den Unternehmen.

Aus einem allgemeinen wirtschaftlichen Verständnis abgeleitet, steht Produktivität für das Verhältnis von Ertrag zu Aufwand. Aber wie sind Aufwand und vor allem der Ertrag in der Softwareentwicklung und für Softwareprojekte sinnvoll zu definieren, sodass die daraus bestimmte Produktivität auch eine praktische Aussagekraft erhält? Ein noch größerer »weißer Fleck« tut sich auf, wenn es um die Frage geht, wie Produktivität in der Praxis der Softwareentwicklung gemessen, bewertet und verbessert werden kann.

Wir stellen Standards und den »State-of-the-Art« in der produktivitätsorientierten oder »produktiven« Softwareentwicklung dar. Mithilfe von Praxisbeispielen, empirischen Daten und konkreten Empfehlungen werden wir unsere Aussagen verdeutlichen und den Leser bei der konkreten Umsetzung unterstützen. Die Erkenntnisse und Empfehlungen beruhen auf unseren eigenen beruflichen Erfahrungen. Wir haben diese immer wieder kritisch geprüft und verglichen mit dem, was im »klassischen« Projektmanagement bereits etabliert ist. Da wir keine wissenschaftliche Arbeit, sondern ein Praxishandbuch schreiben wollten, verweisen wir dabei auf Literatur vor allem an den Stellen, die für den Leser auch zum »Weiterlesen« und für seine eigene Praxis interessant sein könnten.

Produktivität und Qualität sind im besten Sinne Ausdruck von »Peopleware«. Unser Buch richtet sich deshalb an alle Menschen in der Softwareentwicklung – Entwickler, Designer, Architekten, Projektleiter, Teamleiter, Methodenexperten, Controller, Manager. Linienmanager lernen, wie sie optimale Rahmenbedingungen schaffen können. Projektleiter erfahren, wie sie Projekte produktiv führen können. Und Projektmitarbeiter erhalten Hinweise, wie sie zu einer hohen Produktivität beitragen und von ihr profitieren können.

### Drei Anmerkungen zum »Scope« dieses Buches:

Es ist ein Allgemeinplatz, dass zu einer praxisgerechten Betrachtung der Produktivität mehr als »Output« und »Input« gehört. Will man Produktivität im Umfeld des komplexen Entstehungsprozesses von Software verstehen, muss man auch Zielerreichung und Zeit sowie möglicherweise weitere Aspekte betrachten. Auch wenn wir vordringlich den Begriff »Produktivität« verwenden, werden wir diese Aspekte nicht ausblenden. Zur Produktivität kommen für uns Qualität und Geschwindigkeit als gleichberechtigte Erfolgsmerkmale hinzu. Unsere Hauptperspektive bleibt dabei aber die Produktivität.

Weiterhin beschränken wir uns auf die Entwicklung von Software beziehungsweise auf Softwareprojekte. Wir sind uns bewusst, dass für die Beurteilung der Wirtschaftlichkeit eines Softwareprodukts letztlich auch dessen Pflege und Wartung betrachtet werden muss. Vieles, was wir zur Messung der Produktivität sagen werden, lässt sich auch darauf übertragen. Andererseits: Wenn es darum geht, wie eine hohe Produktivität erreicht werden kann, sind die erheblichen Unterschiede zwischen Softwareentwicklung und Softwarewartung – einerseits in Projektform, andererseits häufig als Organisationseinheit – zu berücksichtigen. Wir sind der Meinung, dass das Thema Produktivität in der Softwarewartung durchaus eine eigene umfassende Arbeit rechtfertigt, und haben diesen Aspekt deshalb in unserem Buch ausgeklammert.

Zum Dritten schließlich: Beide Autoren haben ihren beruflichen Erfahrungsgrund primär in der Entwicklung betrieblicher Informationssysteme. Ob unsere Aussagen auch für die Entwicklung technischer Informationssysteme interessant und anwendbar sind, können Praktiker aus diesen Bereichen sicher sinnvoller beurteilen. Wir, die Autoren, hoffen es und freuen uns über Rückmeldungen.

### Zur Gliederung

Wir haben unser Buch in vier Hauptteile gegliedert:

- In **Teil I: Softwareentwicklung und Produktivität** fassen wir zunächst den aktuellen Stand der Softwareentwicklung unter dem Aspekt der *Professionalisierung* zusammen. Hier geht es uns vor allem darum, ein gemeinsames Verständnis dafür zu schaffen, wie heute Software (im Umfeld betrieblicher Informationssysteme) entwickelt wird und welche Herausforderungen es für eine weitere Professionalisierung der Softwareentwicklung gibt. Dazu gehört auch die Frage, welche Bedeutung die Produktivität für die Softwareentwicklung hat und wie Professionalisierung und Produktivität zusammenhängen.

- In **Teil II: Produktivität messen und bewerten** behandeln wir die praktische, konkrete Einführung und Durchführung von *Messungen* für wichtige Kennzahlen zur Produktivität und Qualität in Softwareprojekten. Welche *Kennzahlen* sind wirklich sinnvoll? Wie werden sie gemessen? Und welche Voraussetzungen sind dafür zu schaffen? Und schließlich: Wie sind die Ergebnisse auszuwerten und zu interpretieren?
- In **Teil III: Produktivitätsfaktoren** gehen wir der Frage nach, welche Faktoren eigentlich Produktivität im negativen und im positiven Sinne beeinflussen. Die Vielfalt möglicher Einflussfaktoren auf die Produktivität von Softwareprojekten ist legendär und erschlagend. Wir stellen deshalb ein einfaches Modell für *Produktivitätsfaktoren* vor und beschreiben damit die in der Praxis entscheidenden acht Produktivitätsfaktoren. Daraus resultieren die *acht Gebote produktiver Softwareentwicklung*.
- In **Teil IV: Produktiver werden** beschreiben wir für die acht Produktivitätsfaktoren, welche Maßnahmen konkret in der Organisation und in den Projekten zu entscheidenden Verbesserungen führen. Jedem der acht Gebote produktiver Softwareentwicklung ist dabei ein eigenes Kapitel gewidmet. Wir zeigen die *Richtungen der Beschleunigung* auf und verdeutlichen, durch welche konkreten *produktiven Praktiken* diese erreicht werden können.