



Aus der Reihe

# Management

## Intelligente Nutzung der Intelligenz

Thema	Seite
Prolog.. . . . .	3
Die Phasen. . . . .	3-4
Fehlerquote. . . . .	4
Arbeitsfortschritt durch Nichtstun. . . . .	4-5
Softwareentwickler sind Künstler. . . . .	5
Effektive Arbeitsleistung. . . . .	5-7
Emotionale und rationale Intelligenz. . . . .	7
Epilog. . . . .	7-8
Der Autor. . . . .	8

---

**Dr. Erik Wischnewski**

Heinrich-Heine-Weg 13 • D-24568 Kaltenkirchen

Tel: 04191-7509 • Fax: 04191-770509 • Mobil: 0170-3251666

E-Mail: [info@proab.de](mailto:info@proab.de) • Internet: <http://www.wischnewski-online.de>

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors und seinen Lizenzgebern unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Copyright © 2002-2008 Dr. Erik Wischnewski und seinen Lizenzgebern.  
Alle Rechte vorbehalten.

Version: 1. Juli 2008, 14:11:14



## Prolog

Der junge Kollege saß mir im Raum gegenüber und hackte bei seinem morgendlichen Erscheinen von der ersten Minute an auf seinem Computer, und das bis zur Letzten seines Arbeitstages. Ununterbrochen bemühte er die Tastatur. Es entstand der Eindruck, dass er fleißig und erfolgreich wäre. Ich hingegen machte häufig Erholungs- und Besinnungspausen. Im Endeffekt hatte der junge Mitarbeiter nur ein Zehntel meiner Effektivität nachweisen können. Mein Prinzip war, dass ich nur dann wirklich kreativ arbeitete, wenn ich mich dazu optimal in der Lage sah. In der übrigen Zeit machte ich Pausen oder erledigte Routi-

nearbeiten. Damit erreichte ich, dass ich in den entscheidenden Schritten kaum Fehler machte, die ich später in den ebenfalls Optimalphasen wieder hätte ausbügeln müssen. Der Fortschritt an einem Tag war zwar oftmals nur gering, aber dafür **nachhaltig**, da das Ergebnis fast fehlerfrei war. Beim meinem jungen Kollegen konnte es durchaus vorkommen, dass er an einem Tag eine neue Aufgabe sehr weit vorantrieb, dann aber tagelang damit beschäftigt war, die Fehler zu beseitigen, um schließlich vielleicht sogar nochmals von Neuem zu beginnen.

### Leistungssteigerung durch intelligente IQ-/EQ-Nutzung

Auch wenn diese Erfahrungen bei der Software-Entwicklung gemacht wurden und sich der nachfolgende Beitrag deshalb darauf konzentriert, diese Erfahrungen zu quantisieren und greifbar und somit begreifbar zu machen, gelten Sie meines Erachtens auch in eini-

gen anderen Bereichen, in denen Kreativität eine entscheidende Rolle spielt (z.B. dem Schreiben von Büchern, dem Malen eines Gemäldes oder beim Komponieren).

## Die Phasen

Um meine Hypothese quantitativ zu untermauern, habe ich für die drei Arbeitsphasen *Kreativphase*, *Produktivphase* und *Routinephase* jeweils die Dauer der Phase, meine Arbeitsleistung und die Fehlerquote in der jeweiligen Phase abgeschätzt. Ferner habe ich zur Beseitigung der Fehler einen Ansatz gemacht,

den ich später erläutern möchte. Für meine persönliche Arbeitsleistung in der Software-Entwicklung mit Visual Basic konnte ich folgende Arbeitsleistung, ausgedrückt in durchschnittlich erzeugten Zeilen Quellcode, ermitteln:

Ich bin bei einem 8-Stunden-Arbeitstag durchschnittlich eine halbe Stunde in optimaler Verfassung für kreative Arbeiten. In dieser Kreativphase beträgt meine Leistung 80 Zeilen pro Stunde, und mache dabei ein Fehler pro Stunde. Unter der Annahme, dass ich die übrige Zeit des Arbeitstages nicht mehr arbeite ergibt dies eine Tagesleistung von 40 Zeilen bei durchschnittlich einem halben Fehler.

Die produktiven Phasen ergeben in der Summe etwa 2 Stunden pro Tag. In dieser Zeit erreiche ich stündlich 20 Zeilen bei gleichzeitig 4 Fehlern. Das ergibt ebenfalls 40 Zeilen pro Tag bei 8 Fehlern.

In der übrigen Zeit des Arbeitstages von 5.5 Stunden beträgt meine Arbeitsleistung nur 5 Zeilen pro Stunden bei geschätzten 16 Fehlern pro Stunde (damit sind auch kleinere Fehler gemeint wie Flüchtigkeitsfehler). Hierdurch ergeben sich 28 Zeilen pro Tag bei 88 Fehlern. In dieser Zeit fehlt es mir an Konzentration, bin vielleicht von Müdigkeit beherrscht oder arbeite einfach nur oberflächlich.

	Zeilen pro Stunde	Fehler pro Stunde
Kreativphase	80	1
Produktivphase	20	4
Routinephase	5	16

*Normalansatz*

Der oben gemachte Ansatz funktioniert nur, wenn alle Aktivitäten gut vermischt und gleichverteilt vorkommen. Ansonsten ist mit dem Maßstab ›Zeilen Quellcode pro Stunden‹ nicht viel anzufangen (siehe letzten Absatz in ›Effektive Arbeitsleistung‹). Zudem

sollte man wissen, was für mich typischen Aktivitäten einer Phase sind. Dabei habe ich noch nicht einmal Trivialtätigkeiten wie Aktenablage und Aufräumen mitgezählt.

Phase	Typische Aktivitäten
Kreativphase	„erfinden“, komplizierte Algorithmen
Produktivphase	Spezifikation, Handbuch, Tests
Routinephase	Korrekturlesung, Besprechungen

*Beispiele für typische Aktivitäten einer Phase*

## Fehlerquote

Ich möchte den Sachverhalt nicht zu mathematisch gestalten und führe deshalb vereinfachte Betrachtungen durch. So interessieren mich vor allem die kreativen Aktivitäten eines Softwareprojektes. Ich kann diese z.B. nur in der Kreativphase durchführen, d.h. ich arbeite nur etwa eine halbe Stunde pro Tag. Ich kann diese aber zusätzlich auch in der Produktivphase durchführen. Drittens möchte ich den akademischen Fall betrachten, dass ich die kreativen Aktivitäten nur in der Produktivphase ausübe. Schließlich könnte ich diese Arbeiten auch in der verbleibenden Zeit durchführen, die zwar länger ist, aber von geringerer Leistungsfähigkeit geprägt ist.

Bereits beim Vergleich der erreichten, noch fehlerbehafteten Ergebnisse zeigen sich bei den oben genannten Arbeitsmethoden deutliche Unterschiede: Die Spanne reicht von 28 bis 80 Zeilen pro Tag. Der wesentliche Unterschied liegt aber in der erzeugten Fehlerquote. Diese Fehler müssen in der nachfolgenden Zeit beseitigt werden. Sind es einfache Codierungsfehler, werden sie unmittelbar beim nachfolgenden Test erkennbar; sind es Konzeptfehler, so werden sie meist erst sehr spät erkannt und führen dazu, dass sehr viel getane Arbeit vergeblich war. Die Korrekturzeit ist dann umso länger. Ich habe für mich fest-

gestellt, dass die Korrekturzeit im Projektmittel bei der Zeit liegt, die ich für die Erstellung von 2.5 Zeilen Quellcode benötige. Die absolut benötigte Zeit hängt dann davon ab, in welcher Phase ich die Korrektur durchführe (Kreativphase, usw.) und liegt somit in der Spanne von zwei bis 30 Minuten. Ferner handelt es sich im günstigsten Fall um einfache Codierungsfehler, die zwischen einer halben und fünf Minuten benötigen, und im schlimmsten Fall um Konzeptfehler, deren Behebung zwischen zehn Minuten und mehreren Stunden dauert.

Gerade der letzte Fall zeigt oftmals deutlich die Bedeutung der Kreativphase: Ein Verfahren möge nicht so funktionieren wie man es sich wünscht. Befindet man sich gerade in der optimalen Phase, dann kommt einem der springende Gedanke häufig in minuten-schnelle, nur eine einzige trickreiche Maßnahme ist notwendig und das gesamte bisherige Entwicklungsergebnis kann so bleiben. In der ungünstigen Phase probiert man herum, codiert laufend neuen „Mist“ hinzu, stößt immer wieder erneut auf Hindernisse und wird einfach nicht fertig. Sie kennen dann den Effekt, dass man die Arbeit weglegt und am nächsten Morgen frisch ausgeruht und entspannt (!) sofort die Lösung für das aufgetretene Problem hat.

## Arbeitsfortschritt durch Nichtstun

Warum soll man nun nicht immer so arbeiten, wie man es bei derartigen Fehlern oder Problemen tut? Wer arbeitet macht Fehler! Diese zu beseitigen kostet aber die Hauptzeit eines Projektes. Also ist es sinnvoll, keine (oder zumindest möglichst wenig) Fehler zu machen. Somit arbeite ich nur in der Zeit, wo ich wenig Fehler mache, nämlich in der Kreativphase.

Also arbeite ich nur eine halbe Stunde pro Tag und ruhe mich die übrige Zeit aus. Das Ergebnis ist, dass ich mit dem Projekt effektiv betrachtet relativ schnell voran komme. Es klingt wie schlechter Hohn: Wer arbeitet macht Fehler, also arbeite ich nicht, sodass mein Projekt fehlerfrei ist.

Obwohl diese eben formulierten Grundgedanken prinzipiell richtig sind, hat es den Anschein, als würde da noch ein Haken bei sein. Es kann doch nicht sein, dass jemand bei nur 30 Minuten Arbeitszeit pro Tag mehr schaffen soll wie jemand, der die ganzen acht Stunden schafft?

Es gibt zwei Erklärungsansätze für dieses scheinbare Paradoxon: Zum einen weiß man im voraus nicht, wann diese genialen Momente sind. Es ist ja keine fest definierte Zeitspanne, z.B. von 9:00 bis 9:30 Uhr. Es sind vielmehr einzelne Minuten, vielleicht

auch mal fünf oder zehn im Block, in denen die persönliche Leistungsfähigkeit stark anwächst. Man muss sich also ständig beobachten, ständig auf der Hut sein, um sofort los arbeiten zu können. Dann muss man auch die optimalen Arbeitsbedingungen haben. Bloß keine Maus, die hakt und klemmt, bloß keinen Kollegen, der in dem Moment los redet, und bloß kein knurrender Magen. So betrachtet, muss ich den ganzen Arbeitstag in Bereitschaft sein. Ich tue in dieser Zeit auch etwas, nämlich warten und damit Fehler vermeiden.

## Softwareentwickler sind Künstler

Sie meinen, dass hört sich theoretisch an? Nein, denn ich spreche hier von Softwareentwicklern, die eine kreative Arbeit ausüben. Ein Maler benötigt vielleicht sechs Monate für ein Bild und hat nur jeden zweiten Tag für zehn Minuten daran gemalt. Immer dann nämlich, wenn er die richtige Inspiration hatte. Das sind nur 15 Stunden Arbeitszeit. Ein Auftrag-

geber könnte nun ja auf den Gedanken kommen und den Maler bitten, bis übermorgen ein ebenso gutes Bild für ihn zu malen. Dafür ist er dann auch bereit, zwei Tagessätze Honorar zu zahlen. Softwareentwickler sind wie Maler und Komponisten, es sind Künstler und keine Maschinen. Ihre Kreativität ist gefragt.

## Effektive Arbeitsleistung

Eine andere Rechnung ergibt die durchschnittliche, effektive Arbeitsleistung in Zeilen pro Tag des kreativen Anteils, wenn die Arbeit ausschließlich in der

Kreativ-, in der Produktiv- oder in der übrigen Phase (Routinephase) durchgeführt wird:

	Zeilen pro Tag
Kreativphase	39
Produktivphase	27
Routinephase	3

*Effektive Arbeitsleistung eines Tages beim Normalansatz unter ausschließlicher Nutzung der jeweiligen Phase*

Der zweite Wert (Produktivphase) ist noch einigermaßen gut, weil die Produktivphase pro Tag länger ist (bei mir im Mittel typischerweise viermal so lang wie in der Kreativphase), und andererseits die Fehlerquote noch halbwegs im Rahmen bleibt.

Etwas verwunderlich scheint es zu sein, dass man in der Routinephase noch im langfristigen Mittel 3 Zeilen pro Tag schaffen soll, obwohl mehr Zeit für Fehlerbeseitigung drauf geht als man pro Stunde schafft. Das liegt am gewählten Ansatz, dass für die gänzliche Beseitigung eines Fehlers die Zeit für 2.5

Zeilen Quellcode benötigt wird. Für die Arbeitsleistung eines Tages werden also 8 Tage benötigt, um die Fehler des ersten Tages wieder zu bereinigen (wenn man nur in der Routinephase arbeitet). Man hat also in neun Tagen 27.5 Zeilen fehlerfreien Quellcode erzeugt, das sind 3 Zeilen pro Tag.

Zur Sicherheit meiner Berechnungen möchte ich auch noch eine optimistische Variante durchrechnen. Hierbei schneidet die Kreativphase nicht ganz so gut und die Routinephase nicht ganz so schlecht ab (vgl. Tabelle S.3):

	Zeilen pro Stunde	Fehler pro Stunde
Kreativphase	40	2
Produktivphase	20	5
Routinephase	10	10

*Optimistischer Ansatz*

Die Werte der einzelnen Phasen liegen enger zusammen. Hierbei soll die Korrektur von fehlerhaften Zeilen die Zeit in Anspruch nehmen, die normaler-

weise zur Programmierung von vier Zeilen erforderlich ist. In diesem Fall kommen folgende Werte für den Kreativanteil heraus:

	Zeilen pro Tag
Kreativphase	17
Produktivphase	20
Routinephase	11

*Effektive Arbeitsleistung eines Tages beim optimistischen Ansatz*

Analoges gilt natürlich auch für den Produktivanteil eines Projektes, den man entweder optimal in der Produktivphase ausübt oder ineffektiv in der Routinephase. Die Kreativphase sollte natürlich dem Kreativanteil vorbehalten sein, würde man aber sinnloserweise den Produktivanteil in dieser Phase durchführen, so wäre damit kein Vorteil verbunden. Die Kreativphase degeneriert zur Produktivphase. Das ist Ressourcenverschwendung im Sinne von Qualitätsressource.

Was passiert nun, wenn der Kreativanteil eines Projektes besonders hoch ist und man im Laufe eines Tages nicht mit der halben Stunden auskommt, und eine Fortsetzung am nächsten Tag zu uneffektiv wäre, weil für die Produktiv- und Routinephase keine entsprechenden Aktivitäten vorliegen. Welche Durchschnittsleistung erreicht man, wenn man entweder nur in der Kreativphase, in der Kreativ- und Produktivphase oder den ganzen Tag die Kreativarbeiten ausüben würde?

Die Ergebnisse lauten:

	Zeilen pro Tag
Kreativphase	39
Produktivphase	63
Routinephase	33

*Effektive Arbeitsleistung eines Tages beim Normalansatz unter Ausnutzung aller besseren Phasen bis zur jeweiligen*

Ganz eindeutig ist es sinnvoll, kreative Arbeiten sowohl in der Kreativ- als auch in der Produktivphase auszuüben. Ich bin also etwa 2.5 Stunden pro Tag tätig und entspanne die restlichen 5.5 Stunden eines Arbeitstages. Werden die richtigen Arbeitsbedingungen geschaffen, die in meinem Fall ein Büro im eigenen Heim sind, dann kann der Arbeitstag auch zwölf Stunden lang sein, wodurch sich die Kreativphase oftmals auf 45 Minuten erweitert und die Produktivphase durchaus drei Stunden dauern kann. Das

ergäbe dann eine Gesamttagesleistung von etwa 90 Zeilen pro Tag.

Kehren wir wieder zum Acht-Stunden-Tag zurück. Die „tolle“ Leistung von 63 Zeilen pro Tag wird allerdings im Vergleich zur Kreativphase allein in der fünffachen Zeit erreicht. Wenn man also in der Produktivphase nichts anderes tun kann, lohnt es sich, die kreativen Arbeiten in ihr fortzuführen. Dies gilt aber nicht mehr für die Routinephase.

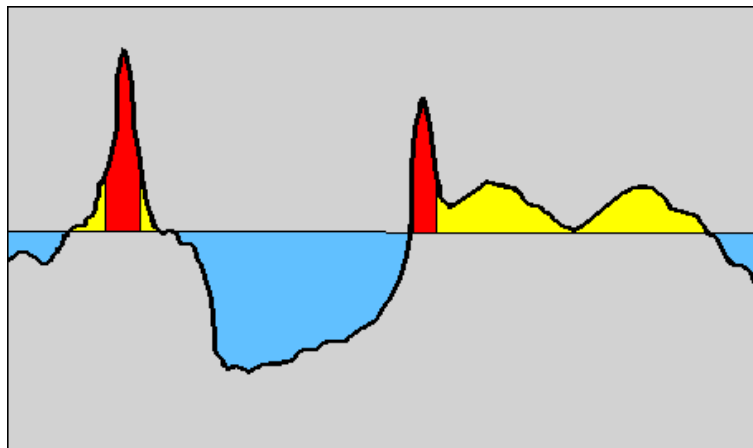
Ich möchte an dieser Stelle noch einmal betonen, dass die Angabe von Zeilen/Tag nicht bedeutet, dass es sich hierbei um reine Codierungsarbeiten handelt. Es ist lediglich die hier gewählte Messlatte und ist im Falle der Softwareentwicklung als Durchschnittswert über alle Komponenten, von der Konzeption bis zur Installation und Abnahme, zu betrachten. Dieses Maß ist genau dann brauchbar, wenn die Codierung eini-

germaßen parallel zur Gesamtentwicklung läuft. Wer also zuerst völlig trocken Studien anfertigt und Konzepte entwirft, ohne auch nur eine Zeile Code entworfen zu haben, der wird mit dieser Berechnungsmethode für sich persönlich nicht viel anfangen können. Das allerdings ändert nichts an dem grundsätzlichen Ergebniss dieser Betrachtungen. Die sind übertragbar.

## Emotionale und rationale Intelligenz

Woran liegt es, dass man eine Kreativ-, eine Produktiv- und eine Routinephase hat? Die persönliche Leistungsfähigkeit ist von enorm vielen Faktoren abhängig. Bekannt ist der tägliche Bio-Zyklus, bei dem man in der späten Mittagszeit ein Tief erreicht

und mehrere Hochs am Tag durchläuft. Dem überlagern sich weitere Einflüsse, sodass es nicht nur zu langsamen Veränderungen kommt, sondern auch zu sehr kurzfristigen, nur wenige Minuten andauernden Spitzen oder Tälern.



*IQ-Kurve im Laufe eines Tages (symbolisch)*

Die Graphik zeigt nur glatte Verläufe, die auch nichts mit dem täglichen Bio-Zyklus zu tun haben. Deshalb wurden auch die Achsenbeschriftungen weggelassen. Die Kurve soll nur symbolisch andeuten, dass der persönliche IQ (Intelligenzquotient) und auch der EQ (Emotionale Intelligenz) sehr stark variieren. So kann ein Mensch mit einem durchschnittlichen IQ von 120 durchaus Spitzenwerte von 200 erreichen und auch Zeiten erleben, in denen sein IQ nicht mehr als 70 beträgt. Nehmen wir an, dass der IQ für insgesamt 30 Minuten eines Tages über 140 steigt, dann könnten das die Minuten der Kreativphase sein. Die Zeit, in

der der IQ dann zwischen 110 und 140 liegt, könnte die Produktivphase sein und die restliche Zeit mit einem IQ unter 110 wäre bei diesen Menschen geeignet für Routinearbeiten. Genau genommen überlagert sich dieses noch mit dem EQ. Insbesondere die Kreativphase wird dann gegeben sein, wenn der IQ und der EQ gleichzeitig hoch sind. Die Produktivphase wird durch einen etwas überdurchschnittlichen IQ bei gleichzeitig durchschnittlichem EQ gekennzeichnet sein. Ist der EQ hoch und der IQ in dem Moment niedrig, nützt einem das für die Kreativität vermutlich nichts, und umgekehrt.

## Epilog

In den obigen Betrachtungen wurde immer nur davon gesprochen, wann man kreative Aktivitäten ausüben sollte. Es wurde diskutiert, wie die effektive Arbeitsleistung ist, wenn kreative Arbeiten nur in der Kreativphase oder auch zusätzlich in der Produktivphase

oder gar in allen Phasen erfolgen würde. Dabei wurde aber ein ganz wichtiger Aspekt außer Acht gelassen. Wenn man auch in der Produktivphase kreative Aktivitäten durchführt, wann erledigt man dann die Tätigkeiten, die für die Produktivphase vorzubehalten

sinnvoll ist? Oder noch krasser gefragt: Wenn ich in allen Phasen versuche, kreativ zu sein, bleiben alle anderen Arbeiten wie das Erstellen von Spezifikationen, das Schreiben von Handbüchern, die Durchführung von Tests und die Teilnahme an Besprechungen, für die man dann plötzlich keine Zeit mehr hat, unerledigt. Das hat weitreichende Konsequenzen für eine erfolgreiche Projektdurchführung.

Zusammenfassend darf festgestellt werden, dass es höchst sinnvoll im Sinne der Gesamteffektivität ist, die unterschiedlichen Aktivitäten eines Projektes hinsichtlich ihrer geistigen Anforderungen (kreativ, produktiv, Routine) möglichst synchron zur persönli-

chen Verfassung, gekennzeichnet durch den momentanen IQ und EQ, zu erledigen. Dadurch werden Zeiten großer Schaffungskraft nicht für Routinearbeiten vergeudet und umgekehrt werden schwierige Aufgaben nicht in Tiefphasen angepackt, was in erster Linie nur zu zahlreichen Fehlern führt, deren anschließende Beseitigung den Fortschritt des Projektes behindert. Umgekehrt dienen leichte Tätigkeiten in der Routinephase zur Erholung und bieten die notwendige schöpferische Pause. Als hilfreiche Randbedingungen erweisen sich: Ruhige Arbeitsumgebung, kein unmittelbarer Zeitdruck und mentale Stärke, auch mal nichts zu tun.

## Der Autor

Dr. Erik Wischnewski, geboren 1952 in Hamburg, Studium der Physik an der Universität Hamburg, Promotion an der Technischen Universität Graz über Projektmanagement.

Wissenschaftlicher Assistent an der Universitätssternwarte in Hamburg-Bergedorf, Industrietätigkeit bei AEG: Entwicklungsingenieur, Vertriebsleiter für Studien und Entwicklung, Projektleiter und -berater und schließlich Fachabteilungsleiter für Entwicklungssteuerung.

Seit 1989 selbständiger Berater und Trainer in der freien Wirtschaft, bei Behörden und Banken, Dozent an mehreren Hochschulen. Referent auf nationalen und internationalen Kongressen und Workshops in Rostock, in Wien und in Polen. Zahlreiche Fachbücher zu den Themen Software-Engineering und Projektmanagement. Seit 1998 überwiegend tätig als IT-Consultant.