

„SOFTWARE-ENGINEERING IST EINE IDEE, DEREN ZEIT GEKOMMEN UND AUCH WIEDER GEGANGEN IST“

Tom DeMarco zum Thema *Metriken im Speziellen und Software-Engineering im Allgemeinen*

Tom DeMarco verfasste in den frühen 80er Jahren ein Plädoyer für bessere frühzeitige Schätzmethoden und Metriken (siehe **Kasten**), das auch heute noch oft zitiert wird. Anlässlich 40 Jahre Software-Engineering wollte OBJEKTSpektrum von DeMarco wissen, wie er heute über die Arbeiten von damals denkt. Das Gespräch führte Peter Hruschka.

? Sind Sie noch immer mit dem Kerngehalt Ihres Metrikbuchs „*Controlling Software Projects: Management, Measurement, and Estimation*“ zufrieden? Und glauben Sie noch daran, dass Metriken ein integraler Bestandteil jeder erfolgreichen Softwareentwicklung sein müssen?

! Nein und nein. Ich sehe das Buch heute als eine eigenartige Mischung aus vielen wahren Aussagen auf jeder Seite, die zusammen genommen aber eine ziemlich falsche Botschaft vermitteln. Es kommt mir vor, als ob der damals junge Autor nie auf eine Metrik gestoßen ist, die er nicht leiden konnte. Die fundamentale Botschaft des Buchs scheint zu lauten: „Metriken sind gut, mehr Metriken wären besser, und ganz viele Metriken optimal.“ Heute verstehen wir alle, dass Softwaremetriken Geld und Zeit kosten und daher sehr vorsichtig eingesetzt werden sollten. Außerdem ist Softwareentwicklung doch etwas grundsätzlich anderes als Naturwissenschaften wie die Physik. Und Softwaremetriken sind viel weniger perfekt, um das zu messen, was man eigentlich überwachen möchte. Man muss sie mit mehr Fingerspitzengefühl interpretieren, statt den Zahlen bedenkenlos zu vertrauen.

? Was denken Sie heute über das berühmt gewordene Zitat, mit dem das

Buch beginnt: „Was man nicht messen kann, kann man auch nicht überwachen und steuern“?

! Auch da ist viel Wahres dran, aber die Art, wie ich es benutzt habe, empfinde ich heute mehr und mehr als peinlich. Implizit in dem Zitat (und auch im Buchtitel) steckt die Idee, dass Überwachen ein sehr wichtiger Aspekt jedes Softwareprojekts ist, vielleicht sogar der wichtigste. Aber das stimmt nicht. Es gab und gibt viele Projekte, die ohne viel Steuerung durchgeführt wurden und wunderbare Produkte wie GoogleEarth oder Wikipedia geschaffen haben. Um die wahre Rolle von Überwachen und Steuern zu verstehen, sollte man zwei verschiedene Projekttypen betrachten:

- Projekt A wird am Ende 1 Million Dollar kosten und einen Wert von ca. 1.1 Millionen Dollar erwirtschaften.
- Projekt B wird auch ca. 1 Million Dollar kosten und einen Wert von mehr als 50 Millionen Dollar erwirtschaften.

Jetzt ist sofort klar, dass man in Projekt A viel Überwachung und Steuerung braucht, während das im Projekt B völlig unwichtig ist. Das führt mich zu der überraschenden Schlussfolgerung, dass Überwachen und Steuern nur bei relativ nutzlosen Projekten sehr, sehr wichtig ist und bei sehr nützlichen Projekten eher weniger wichtig. Daraus kann man schließen, dass Personen, die sich stark auf Überwachen und Steuern konzentrieren, meist an Projekten arbeiten, die etwas abliefern, was relativ bedeutungslos für den Geschäftserfolg ist. Die wesentlich wichtigere Frage als „Wie überwache und steuere ich ein Softwareprojekt?“ ist daher: „Warum arbeiten wir



Tom DeMarco und Peter Hruschka sind beide Principals der Atlantic Systems Guild. DeMarco ist Autor bzw. Koautor von zwölf Büchern. Das neueste darunter ist „Adrenalin-Junkies und Formular-Zombies: Typisches Verhalten in Projekten“, erschienen im Carl-Hanser Verlag (englische Fassung bei Dorset House, New York).

an so vielen Projekten, die so wenig Wert abliefern?“

? Kann man wirklich behaupten, dass man ein Projekt ohne Steuerung oder mit relativ wenig Steuerung laufen lassen kann?

! Ja, so ungefähr. Zuerst sollten wir nur Projekte starten, bei denen Überwachen und Steuern nicht so wichtig sind. Und dann sollten wir unsere Erwartungshaltung, wie gut man ein Projekt überhaupt – trotz heftiger Bemühungen – überwachen und steuern kann, etwas zurückschrauben. Betrachten wir dazu eine Metapher: Stellen Sie sich vor, dass wir nicht ein Softwareprojekt lenken wollen, sondern uns über die Erziehung unserer Kinder Gedanken machen. Schon allein die Idee, dass wir das Heranwachsen unserer Kinder gezielt lenken und steuern wollen, sollten wir als ein wenig peinlich empfinden. Aber es geht bei der Erziehung doch um unheimlich viel. Wenn man dabei versagt, völlig versagt, kann das Leben vieler Personen ruiniert sein. Es ist also wichtig, dass einem die Erziehung nicht völlig aus der Hand gleitet. Man fühlt sich ungefähr wie ein Fechter, der lernt, sein Schwert so zu halten, als ob es ein Vogel wäre: zu fest und man verletzt den Vogel, zu locker und er wird wegfliegen. Wenden Sie jetzt meinen Spruch „Was man nicht messen kann, kann man auch nicht überwachen und steuern“ auf Teenager an. Das, was wirklich wichtig ist – Ehre, Würde, Disziplin, Persönlichkeit, Werte, Ethik, Einfallsreichtum, Loyalität, Humor, Großzügigkeit – wird überhaupt nicht gemessen. Man muss Jugendliche so gut wie möglich in diese Richtung lenken, ohne jegliches zahlenmäßiges Feedback. Das ist schwer, aber Erziehung ist halt schwer. Ein bisschen Feedback bekommt man viel-



1982 erschien das Buch „Controlling Software Projects – Management, Measurement & Estimation“ von Tom DeMarco bei Prentice Hall, NY. Die deutsche Fassung „Was man nicht messen kann ...“ ist in zweiter Auflage 2008 im mitp-Verlag, Bonn erschienen. In dem Buch führt DeMarco Kennzahlen auf der Basis von Analyse- und Designmodellen ein, um frühzeitig – d.h. schon lange, bevor man Codezeilen zählen kann – zu Metriken zu kommen, die etwas über die Komplexität der Aufgabenstellung und daher über die erwartete Laufzeit und den erwarteten Aufwand eines Projekts aussagen und die im Projektverlauf gezielt für die Steuerung der Teilprojekte und zur Fortschrittskontrolle verwendet werden können.

leicht in Form von Schulnoten und dafür ist man auch dankbar. Aber wir wissen, dass die Mathematiknote mehr Aussagekraft hat als die Englischnote, da man mathematisches Können besser messen kann als Sprachbeherrschung. Und die „Note“ in

Betragen und Sozialverhalten sagt meist mehr über den Lehrer als über das Kind aus.

? *Wie halten Sie dann ein Projekt im Griff, ohne es zu überwachen und zu steuern?*

! Du betreust die Leute und überwachst Zeit und Geld. Du sagst dem Projektleiter zum Beispiel „Ich habe ein Abgabedatum im Kopf, aber ich verrate es Ihnen nicht. Wenn ich eines Tages hereinkomme und Ihnen mitteile, dass das Projekt in einer Woche vorbei ist, dann müssen Sie bereit sein, das, was Sie haben, als Endprodukt zusammenzupacken und auszuliefern. Ihre Aufgabe ist es, jeweils die Teile zu dem Ganzen hinzuzufügen, die den größten Wertzuwachs bringen, und das Ganze ständig zu integrieren, zu testen und zu dokumentieren.“

? *Das klingt sehr nach einem agilen Ansatz, den Sie da vertreten.*

! Ja, ich bin für Managementmethoden, die das Team in Richtung Agilität bewegen, zumindest in Richtung iterative und inkrementelle Verfahren.

? *Wir feiern jetzt den 40. Geburtstag des Begriffs „Software-Engineering“, 40 Jahre nach der NATO-Konferenz in Garmisch. Sie sind scheinbar nicht mehr so begeistert von Metriken – aber darf ich fragen, ob Sie auch weniger an das ganze Thema Software-Engineering glauben?*

! Ja, das trifft zu. Ich glaube, Software-Engineering ist eine Idee, deren Zeit gekommen und auch wieder gegangen ist. Ich glau-

be noch immer, dass es sinnvoll ist, Software nach Ingenieurprinzipien zu entwickeln. Aber das ist nicht mehr die Richtung, in der sich das Software-Engineering bis heute weiter entwickelt hat. Der Begriff schließt heute eine Menge Teildisziplinen ein, unter anderem Prozessdefinitionen, Inspektionen und Walkthroughs, Requirements-Engineering, Traceability, Metriken, Qualitätsmanagement, Planungs- und Überwachungsmethoden sowie Codier- und Dokumentationsstandards. In jeder dieser Disziplinen strebt man nach konsistenten Praktiken und Vorhersagbarkeit. Das sind auch nach wie vor erstrebenswerte Ziele, aber ich zweifle, dass es wirklich das Wichtigste ist. In den letzten 40 Jahren haben wir uns damit herumgequält, Projekte termingerecht und innerhalb des Budgetrahmens fertigzustellen. Aber – wie ich schon angedeutet habe – hätte das nie unser Hauptziel sein sollen. Das wichtigere Ziel wäre Wandel – Software zu entwickeln, die die Welt verändert oder die Art und Weise, wie wir unser Geschäft betreiben. Softwareentwicklung ist ein experimentelles Feld und wird es auch immer bleiben. Die eigentliche Konstruktion ist nicht experimentell, aber die Konzeption ist es. Und darauf sollten wir unser Hauptaugenmerk richten. Eigentlich sollten wir darauf schon immer unser Hauptaugenmerk gerichtet haben.

? *Herr DeMarco, vielen Dank für das Gespräch.*

Tom DeMarco wird als Keynote-Speaker bei der OOP 2009 in München dabei sein.