Buchbesprechungen

Tom DeMarco: The Deadline - A Novel About Project Management *(Dorset House Publishing, New York, 1997, ISBN 0-932633-39-0, 310 Seiten, US$ 29.70)*

Software-Management als Krimiroman

Paradiesische Zustände, die sich jeder Software--Manager wünscht. Genügend qualifiziertes und motiviertes Personal, jegliche benötigte Infrastruktur, vernünftiger Zeitrahmen und unbeschränktes Budget für Projekte, deren erwartete Ergebnisse, die Softwareprodukte eindeutig spezifiziert sind. So paradiesisch, dass man sich sogar leisten kann, jeweils drei Teams mit der gleichen Aufgabe zu betrauen, damit man mit unterschiedlichen Vorgehen für die Entwicklung und ihr Management experimentieren kann.

Ist das der Stoff, aus dem man den ersten Roman schneidert, der Software Engineering zum Thema hat? Natürlich nicht. Der grosszügige Auftraggeber und Sponsor des Unternehmens geht auf Reisen und sein Stellvertreter ist auf dem Software-Auge blind und sieht nur mit dem Finanz-Auge. Seine erste Anweisung: Laufzeit der Projekte verkürzen, damit die Einnahmen früher kommen. Seine zweite: Aufstocken eines der Projektteams für das Schlüsselprodukt. Seine dritte ist nicht sehr originell: Endtermin nochmals auf einen früheren Zeitpunkt legen. Höllische Zustände? Normal auf Erden, sagen die meisten Software-Manager unter uns.

Die Hauptfigur des Romans ist Webster Thompkins, der kurz vor seiner Pensionierung unverhofft und auf eine abenteuerliche Weise zu seinem Job kommt. In einem fernen unbekannten Land durchwandert Thompkins alle Stadien zwischen dem Manager-Paradies und der Manager-Hölle. Von uns wirklichen Software-Managern unterscheidet ihn die unerschöpfliche Quelle der besten Berater der Welt, die ihm in den brenzligen Situationen immer mit dem richtigen Rat beistehen. Belinda Binda, eine US-Amerikanerin, die sich in dieses Land zurückzog, nachdem sie einige grosse Projekte gemeistert hat, ist seine wichtigste Beraterin, oder vielmehr sein Coach. Sie erklärt ihm manch ein Phänomen, das er zwar kennt aber zu deren Ursachen er nie vorgedrungen ist. Und der Leser lernt brav mit Webster mit.

In Silikon Valejit von Morovia, da spielt sich nämlich dieses Wechselbad der Managergefühle ab, gibt es noch andere weise Menschen, die Thompkins immer zur rechten Zeit über den Weg laufen. Etwa Gabriel Markov, der unter dem früheren sowjetischem Regime General war und, daher eher überraschend, virtuos mit Menschen umgehen kann. Oder der bescheidene Waldo Montifiore, der als Sekretär von Webster beginnt und zum Manager der Metrik-Gruppe wird, nachdem er mit Akribie und Windeseile alle verfügbaren Zahlen in Silikon Valejit ausgegraben hat. Und vor allem Aristotle Kenoros, der erste Programmierer Morovias, der Thompkins die Grundsätze der Software-Entwicklung nahebringt.

Für kurzfristige Einsätze stehen ihm so bekannte Persönlichkeiten als Berater zur Verfügung wie Dr. Hector Rizzoli, Abdul Jamid, T. Johns Caporous und Dr. Larry Boheme. Oder haben Sie sie etwa nicht erkannt? Sind Sie neugierig? Grund genug das Buch zu lesen, um sie zu entlarven. Vielleicht hilft ihnen beim ersten Namen das folgende Zitat:

"Aber da muss etwas sein, was ich tun kann, irgendeine kurzfristige Massnahme zur Verbesserung der Produktivität, zum Beispiel ... "

Hector schüttelte energisch seinen Kopf. "Es gibt kein solches Ding wie eine kurzfristige Massnahme in unserem Geschäft. Es gibt keine Möglichkeit, die Produktivität kurzfristig zu verbessern. Letztlich, ist die von Ihnen erreichte Produktivität ein direktes Ergebnis der langfristigen Investition Ihrer Vorgänger. Und zugunsten Ihrer Nachfolger zu investieren ist der einzige wirkliche Einfluss, den Sie jetzt auf die Produktivität haben können."

Herr Thompkins seufzte. "Ich glaube, ich wusste es. Trotzdem ist es erfrischend, es so frei heraus gesagt zu bekommen."

"Die kalte Dusche der Wirklichkeit auf das Thema der Produktivitätsverbesserung."

Natürlich kennen und erkannten Sie ihn. Inzwischen ist er bereits an einer Konferenz und Webster Thompkins noch immer in Morovia und schaut weiteren anstrengenden Tagen entgegen. Und lernt fleissig weiter. Am Ende jedes aufregenden Tages notiert er Anregungen für uns in sein Notizbuch. Hier sind einige, mit Bedacht ausgewählte:

Management umfasst Herz, Bauch, Seele und Nase.

Vermeiden von Fehlern, viel eher als das Optimieren des Erfolgs, steigert die Gesamtleistung.

Erhöhte Überzeit reduziert die Produktivität.

Konflikt verdient Beachtung. Konflikt ist kein Zeichen unprofessionellen Verhaltens.

Tödlich ist nicht, was man nicht weiss ... sondern was man weiss aber dem ist nicht so.

Ein Projekt benötigt beides, Ziele und Schätzungen; sie sollten unterschiedlich sein.

Klingen sie für Sie kontraintuitiv? Die Geschichten hierzu sind ein ausgezeichnetes Training für Ihre Intuition. 18 Mal mehr solcher Erkenntnisse konnte Webster Thompkins in seinem Notizbuch vermerken, bevor er in den verdienten Ruhestand ging. Und uns, die noch nicht ruhen dürfen oder wollen, sollten diese Notizen täglich beunruhigen: Sind wir nicht in der gleichen verzwickten Situation? Kann uns ein Blick nach Morovia aus ihr heraushelfen?

Warum es ein Krimiroman ist? Aus zwei Gründen. Erstens, es ist die sich etwas mühsam anbahnende kriminelle Tat einer gescheiten Frau, die unserem Helden die Lektion in Software-Management ermöglicht. Und wie in jedem anständigem Krimi gibt es auch hier ein Happy End. Zweitens, liest sich die in diesem äusseren Rahmen eingebettete Geschichte der Software-Projekte und ihrer Führung wie ein Krimi. Die Spannung steigt von Kapitel zu Kapitel und regt die Phantasie des Lesers an. Ich empfehle Ihnen, zwischen den Kapiteln bewusst Pausen einzulegen und jeweils die Geschichte selbst weiterzuspinnen bevor Sie weiterlesen.

Klingt alles sehr positiv? Gut, ich bekenne: Am Anfang war ich hin (weiterlesen) und her (aufgeben) gerissen. Das Buch beginnt nicht mit dem Knall. Langsam, zum Teil holprig und plakativ wird die Geschichte aufgebaut. Weil ich Tom DeMarco und sein bisheriges Werk kenne, riss mich meine Neugier hin. Zum Glück. Sobald er auf dem heimischen Boden ist (ich meine Software Engineering, nicht Morovia), gewinnt die Geschichte an Fahrt, die Dialoge fliessen wie von selbst und es entstehen Bilder des Projektalltags. Eines Projektalltags, den wir erträglicher gestalten können, wenn wir uns von diesem Buch inspirieren lassen.

Karol Frühauf, INFOGEM AG, 5401 Baden

G. Kösters, B.-U. Pagel und H.-W. Six: Software Engineering - Aufgaben und Lösungen

Wer schon einmal Übungsaufgaben zu einer Software-Technikvorlesung stellen mußte, weiß wie arbeitsaufwendig das Formulieren guter Aufgaben und die Erstellung von Musterlösungen sind. Deshalb werden Lehrende in der Software-Technik ein Buch mit Aufgaben und Lösungen dankbar annehmen. Auch für Lernende, die sich im Selbststudium in neue Gebiete der Software-Technik einarbeiten wollen, könnte ein derartiges Buch hilfreich sein.

Das vorliegende Buch beinhaltet eine Sammlung von Aufgaben mit Musterlösungen, die im Rahmen der Software-Technikausbildung an der FernUniversität Hagen verwendet wurden. Die behandelten Themen orientieren sich stark an dem Lehrbuch Pagel/Six: Software Engineering - Band 1: Die Phasen der Software-Entwicklung.

Das Aufgabenbuch ist in die fünf Kapitel Analyse und Definition, Entwurf, Modulare Software-Entwicklung mit ANSI-C, Implementation und Testen eingeteilt. Zu jedem Kapitel gibt es eine Reihe von recht komplexen Aufgaben mit Unterpunkten. In jedem Kapitel werden verschiedene Methoden und Verfahren der Software-Technik behandelt. Die Aufgaben in den einzelnen Kapiteln bauen teilweise aufeinander auf, so daß ein Problem in verschiedenen Phasen des Lebenszyklus bearbeitet wird und verschiedene Methoden eingesetzt werden.

Die Lösungen zu den Aufgaben sind sehr umfangreich. Sie werden Schritt für Schritt entwickelt, sind gut kommentiert und eignen sich zum Selbststudium.

Erfahrungen mit dem Buch

Für die Übungen zur Software-Technologievorlesung im WS 97/98 haben wir in Dortmund das Buch eifrig studiert und zwei Aufgaben daraus eingesetzt. Es sind nicht mehr geworden, obwohl wir uns gern mehr Arbeit erspart hätten, weil es stark auf das Lehrbuch von Pagel und Six zugeschnitten ist, und dieses Buch nicht die einzige Grundlage der Vorlesung war.

Als Implementierungssprache liegt Modula 2 zugrunde, davon ist nicht nur das Implementierungskapitel sondern auch das Entwurfskapitel geprägt. Es beschäftigt sich mit modularem Entwurf, während unserer Vorlesung die Konzepte der rollenbasierten und der objektorientierten Software-Entwicklung zugrunde lagen.

Das Kapitel über Analyse und Definition scheint uns am vielfältigsten einsetzbar, da es programmiersprachenunabhängig ist. Dort finden sich Aufgaben zu ER-Modellierung, strukturierter Analyse und objektorientierter Analyse. Insbesondere Aufgaben zur Analyse, in denen ein in sich geschlossener Ausschnitt der realen Welt beschrieben werden muß, sind aufwendig zu erstellen und schwierig zu formulieren, da schnell wichtige Teilaspekte vergessen werden. Deshalb haben wir dankbar die Gelegenheit ergriffen und eine dieser Aufgaben erfolgreich eingesetzt.

Das Entwurfskapitel ist stark durch die im Lehrbuch definierte Spezifikationssprache geprägt. Das trifft insbesondere auf die Lösungen zu. Wenn man die Aufgaben ein wenig umschreibt, können sie wohl auch in einer anderen Begriffswelt benutzt werden.

Wer einen modularen Entwurf in ANSI-C realisieren möchte, findet in Kapitel 3 mehrere Übungsaufgaben zur Umsetzung des Modulkonzepts.

Im Kapitel zur Implementierung stehen die Codierung in Modula 2 und Effizienzbetrachtungen im Vordergrund. Die Themen der Aufgaben stammen fast ausschließlich aus der Mathematik. Diese Aufgaben könnten auch in einem Modula-Programmierkurs eingesetzt werden.

Eine der Aufgaben zum Testen haben wir benutzt, obwohl die zu testende Prozedur in Modula 2 angegeben war. Von Studierenden im Hauptstudium wird erwartet, daß sie den Transfer von einer Sprache in eine andere leisten können. Die anzuwendenden Testverfahren wurden in der Vorlesung behandelt. Leider enthielt die Lösung der Aufgabe drei kleine Fehler.

Die Aufgaben sind nach Schwierigkeitsgrad und Umfang klassifiziert. Nach unserer Erfahrung sind diese Einschätzungen realistisch.

Fazit

Wenn eine Software-Technikvorlesung ähnlich aufgebaut ist wie das Lehrbuch von Pagel/Six und Modula 2 oder ANSI-C als Implementierungssprache benutzt wird, bietet das vorliegende Buch eine gute Basis zum Nacharbeiten des Stoffs in Form von Übungen.

Doris Schmedding und Corina Kopka
Universität Dortmund, Informatik 10 (Software-Technologie), D-44221 Dortmund,
schmedding|kopka@cs.uni-dortmund.de