

## Professionalisierte Rentnerverwaltung lohnt sich

Eine Kosten-Nutzen-Analyse unter Einsatz moderner Projektmanagement-Methoden

**Martin Schriefer, Filderstadt**

Das Umfeld der betrieblichen Altersvorsorge unterliegt insbesondere in den letzten Jahren einem rapiden Wandel, ebenso wie die Professionalisierung im Projektmanagement. Am Beispiel einer Verwaltungsfunktionalität für die Rentenzahlungsphase bei Pensionskassen, Pensionsfonds oder Lebensversicherern, im Folgenden kurz Rentnerverwaltung genannt, soll gezeigt werden, wie mit modernen Projektmanagementwerkzeugen solide Investitionsentscheidungen getroffen werden können.

Die durch das neue Alterseinkünftegesetz eingeführten Entwicklungen in der betrieblichen Altersversorgung ziehen insbesondere Änderungen in der Besteuerung, aber auch der Beitragsermittlung zur Sozialversicherung von Renten- bzw. Pensionsleistungen nach sich. Die neuen Regelungen werden durch heute bestehende Systeme meist nur unzureichend bis gar nicht abgedeckt. Somit werden die Versorgungsanbieter immer häufiger vor der Aufgabe stehen, auch in solchen Randgebieten des Tagesgeschäftes unter strengen ökonomischen Gesichtspunkten Investitionsentscheidungen zu treffen.

Betrachtet man zum Beispiel Rentenauszahlungen, so konnten hier bisher die meisten Rahmenbedingungen der Auszahlungsmodalitäten organisatorisch abgeklärt werden, auch ohne intensive maschinelle Unterstützung. Die Änderungen in der Sozialversicherungsbeitragsabwicklung und die unterschiedlichen Besteuerungsmethoden von Rentenleistungen werden hier aber zukünftig komplexere Fragestellungen aufwerfen. Daher stehen viele Unternehmen vor der Frage, wie diese neuen Anforderungen effizient bewerkstelligt werden können.

Im anderen Umfeld des Projektmanagements hält eine zunehmende Professionalisierung Einzug. Neue Technologien stellen zwar neue Anforderungen, beispielsweise an die Methodik der Aufwandschätzung. Allerdings erlauben grundlegende Untersuchungen unter älteren Technologien oftmals einen Transfer von Ergebnissen auch auf diese neue Technologien.

Exemplarisch werden hier zwei Methoden zur Aufwandschätzung betrachtet. Zum Ersten sei die Function-Point-Methode genannt, die immer größere Verbreitung findet. Grundlage ist hierbei die fachliche Beschreibung von Anforderungen, zum Beispiel in Form der strukturierten Analyse oder von Use Cases (siehe [Bur97]). Diese wird dann hinsichtlich ihrer Komplexität bewertet, woraus die Anzahl der sogenannten ungewichteten Function-Points (siehe [Bun04]) folgt. Einfluss finden dann die Rahmenbedingungen des Produktes, der Hardware und der Organisation, woraus sich die Function-Points ableiten, die somit als Maß des funktionalen Umfanges dienen. Im Weiteren kann daher der Begriff der Function-Points mit dem Funktionsumfang gleichgesetzt werden. Auf Basis der Function-Points wird aus Erfahrungswerten früherer Projekte der Entwicklungsaufwand abgeleitet.

Zum Zweiten wird COCOMO II („CONstructive COst MOdel“) betrachtet. Dieses Modell leitet aus der geschätzten Programmgröße, die in einer normierten Zählung von jeweils 1000 Zeilen Programmcode (KLOC – „Kilo Lines of Code“) gemessen wird, den Aufwand des Projektes ab. Auch hierbei werden individuelle Projekt- und Produktrahmenbedingungen berücksichtigt.

## **Methode**

Eine der Hauptschwierigkeiten bei der Anwendung von COCOMO II ist die Abschätzung der Programmgröße. Unter der Annahme, dass beide oben genannten Methoden vergleichbare Aufwände liefern sollten, können sie derart kombiniert werden, dass man eine Formel für die Programmgröße abhängig von den Function-Points erhält. Dieses Vorgehen nennt man „Backfiring“. Es fließt hierbei auch ein linearer Faktor ein, der die zugrunde liegende Programmiersprache berücksichtigt. In der Praxis wird in ausreichender Vereinfachung ein linearer Zusammenhang zwischen ungewichteten Function-Points und der Programmgröße angenommen.

Im konkreten Fall einer Rentnerverwaltung wird der funktionale Umfang mit Hilfe von Use Cases beschrieben und dann mit ungewichteten Function-Points bewertet. Eine Produktanalyse liefert den Korrekturfaktor für die Ermittlung der Function-Points, welche der Vollständigkeit wegen angegeben werden.

Mittels „Backfiring“ wird dann die für das COCOMO II–Verfahren notwendige Programmgröße ermittelt. COCOMO II liefert den erwarteten Projektumfang, die erwartete Projektdauer und die Verteilung des Aufwands auf einzelne Aktivitäten..

Wir werden das COCOMO II – Modell in der Sichtweise der frühen Schätzung („Early Design“) nutzen. Eine detailliertere Schätzung für eine späte Schätzphase („Post Architecture“) lässt sich erst im konkreten Projektumfeld sinnvoll nutzen und ist deshalb nicht mehr Bestandteil dieser Betrachtung.

In diesem Artikel werden für die genannten Punkte nur die wesentlichen Ergebnisse dargestellt, ohne Formelwerk und Details zu Bewertungen. Die genaue Herleitung kann gerne beim Autor erfragt werden.

## **Fachliche Beschreibung der Anforderung**

Es wird folgende Auswahl grundlegender Verarbeitungen für eine Rentnerverwaltung getroffen:

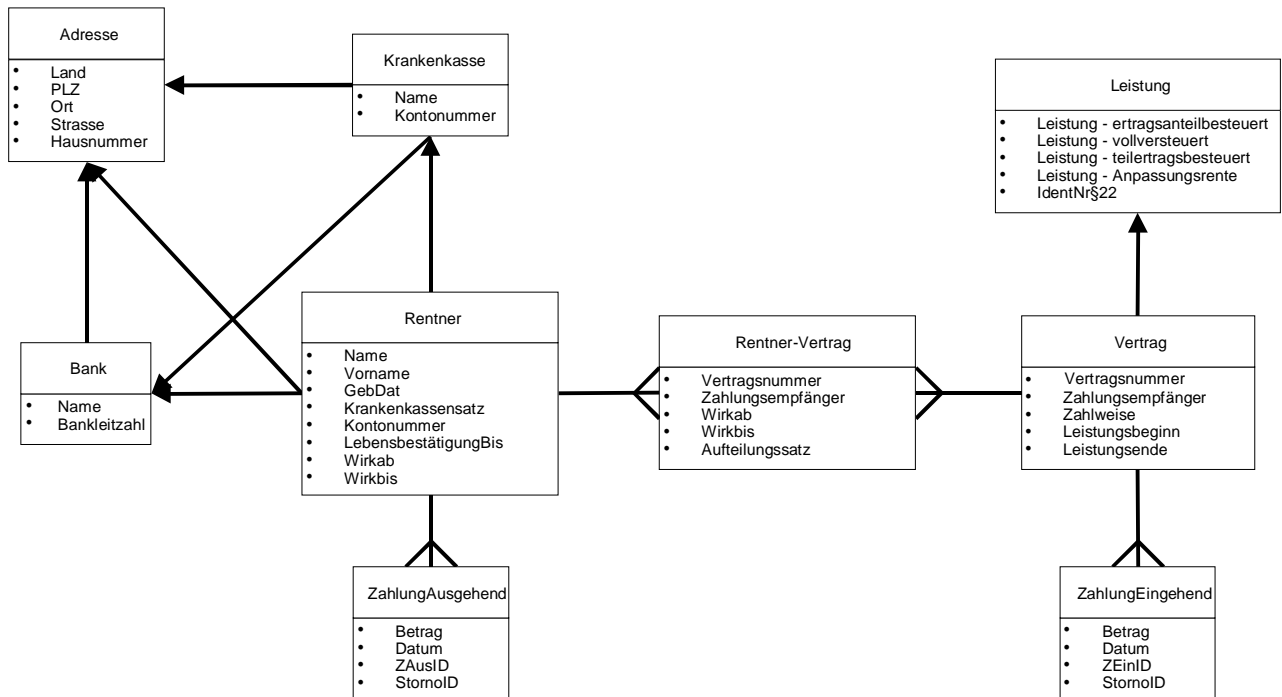
Adressen verwalten	Adresse anlegen
	Adresse ändern
	Adresse suchen / anzeigen
Bankverbindung verwalten	Bankverbindung anlegen
	Bankverbindung ändern
	Bankverbindung suchen / anzeigen
Krankenkassendaten verwalten	Krankenkasse anlegen
	Krankenkasse ändern
	Krankenkasse suchen / anzeigen
Rentner – Vertragsdaten verwalten	Rentner anlegen
	Rentner ändern / stornieren
	Rentner suchen / anzeigen
Mitteilung nach EStG	Mitteilung kommt vom Produktgeber und wird gemäß § 22 Nr. 5 Satz 7 EStG an den Rentner sowie evtl. gemäß § 22a EStG an die BfA (ZfA) verteilt
Zahlungsflüsse	Auszahlungsbetrag kommt vom Produktgeber und wird an Rentner und Krankenkasse verteilt
	Zahlungsfluss suchen / anzeigen
	Zahlungsfluss stornieren über die Zuordnung der zu stornierenden Zahlung in einem separaten Feld
Lebensbestätigung	Prüfen, ob Lebensbestätigung ausläuft, wenn nein, Brief an Rentner und Liste an Sachbearbeiter ausgeben. Das Einpflegen der zurückgemeldeten Lebensbestätigung erfolgt über „Rentner ändern“.

Weitere fachlich wichtige Geschäftsvorfälle bzw. Verarbeitungen lassen sich dann technisch einfach auf die genannten Basisverarbeitungen zurückführen:

- „Tod des Rentners“ → Rentner ändern, Rentner anlegen  
(Wirksamkeit des alten Rentners beenden, dann neue Rentner, die Erben, einrichten)
- „Scheidung“ → analog „Tod Rentner“
- Rückwirkende Änderungen (Krankenkassendaten, Rentnerdaten, Storno von Änderungen) → werden über „Rentner ändern“ realisiert

Zur Ablage der relevanten Daten wird folgendes fachliche Datenmodell genutzt:

## Fachliches Datenmodell



## Zählung der Function-Points

Zuerst werden die ungewichteten Function-Points gezählt. Hierzu müssen – aus Datensicht – interne und externe Datenbestände (ILF und ELF) sowie – aus funktionaler Sicht – externe Eingaben (EI), Ausgaben (EO) und Abfragen (EQ) analysiert werden. Dann wird daraus ihre Komplexität abgeleitet. Aus der Komplexität der Datenbestände und Funktionen ergeben sich schließlich die ungewichteten Function-Points.

Exemplarisch wird hier die Zählung für die Anwendung „Krankenkasse anlegen“ (siehe Tabelle 1) und „Mitteilung § 22“ (siehe Tabelle 2) dargestellt. Die Notation ist an [Bun04] angelehnt, dabei ist FTR („file type record“) die Anzahl der referenzierten Datenbestände, RET („record element type“) sind die Datenuntergruppen und DET („data element type“) ist die Anzahl der Datenelemente (s. Bewertung).

Use Case	Zähl-objekt	FTR/RET/DET	Einzelwert	Σ
Krankenkasse anlegen	ILF	RET	• Eigenes ILF Krankenkasse	1
		DET	• Krankenkasse - Name • Krankenkasse - Kontonummer • Verweis auf ELF Adresse • Verweis auf ELF Bank	4
	EI	FTR	• Eigenes ILF Krankenkasse pflegen • ELF Adresse lesen • ELF Bank lesen	3
		DET	• Krankenkasse - Name • Krankenkasse - Kontonummer • Systemmeldungen	3

Tabelle 1

Use Case	Zähl-objekt	FTR/RET/DET	Einzelwert	Σ
Mitteilung nach EStG	ILF	RET	• Eigenes ILF Leistung	1
		DET	• Leistung - ertragsanteilbesteuert • Leistung - vollversteuert • Leistung - teilertragsbesteuert • Leistung - Anpassungsrente • IdentNr § 22 • Verweis auf ELF Rentner-Vertrag	6
	EI	FTR	• Eigenes ILF Leistung pflegen • ELF Renter-Vertrag lesen	2
		DET	• Leistung - ertragsanteilbesteuert • Leistung - vollversteuert • Leistung - teilertragsbesteuert • Leistung - Anpassungsrente • IdentNr § 22 • Systemmeldungen	6
	EO	FTR	• Eigenes ILF Leistung pflegen • ELF Adresse lesen • ELF Renter-Vertrag lesen	3
		DET	• Leistung - ertragsanteilbesteuert • Leistung - vollversteuert • Leistung - teilertragsbesteuert • Leistung - Anpassungsrente • IdentNr § 22 • Vertrag - Vertragsnummer • Vertrag - Zahlweise • Vertrag - Leistungsbeginn • Vertrag - Leistungsende • Rentner - Name • Rentner - Vorname • Rentner - GebDat • Adresse - Land • Adresse - PLZ • Adresse - Ort • Adresse - Strasse • Adresse - Hausnummer • Systemmeldungen • Rentner-Vertrag - Aufteilungssatz	19

Tabelle 2

## Die Bewertung der Komplexität liefert für diese beiden Beispiele

- Krankenkasse anlegen – ILF: Niedrige Komplexität 7 ungewichtete Function-Points
- Krankenkasse anlegen – EI: Mittlere Komplexität 4 ungewichtete Function-Points
- Mitteilung § 22 – ILF: Niedrige Komplexität 7 ungewichtete Function-Points
- Mitteilung § 22 – EI: Mittlere Komplexität 4 ungewichtete Function-Points
- Mitteilung § 22 – EO: Mittlere Komplexität 5 ungewichtete Function-Points

Für den Geschäftsvorfall „Krankenkasse anlegen“ erhält man also 11 und für die „Mitteilung § 22“ 16 ungewichtete Function-Points. Betrachtet man alle Geschäftsvorfälle, so ergibt sich eine Gesamtsumme von 134 ungewichteten Function-Points.

## Übergang zu COCOMO

Allgemein liefert COCOMO II eine Aufwandschätzung für einen normierten Aufwand; das ist im Wesentlichen der Aufwand ohne Plan- und Anforderungserstellung laut Wasserfallmodell. Der Aufwand in Personenmonaten ist exponentiell von der Programmgröße und linear von Korrekturfaktoren abhängig. Der Exponent wird dabei durch die Projekteffizienz feinjustiert, während in die Korrekturfaktoren die Bewertung der Produkt- und Projektrahmenbedingungen einfließt.

Wichtig für die weitere Betrachtung ist, dass explizit die Änderung des Aufwands angegeben werden kann, wenn sich die Programmgröße linear und der Exponent additiv ändern. Weiter liefert COCOMO II auf ähnliche Weise wie für den Aufwand eine Formel für die Projektdauer in Monaten – auch hier ohne Plan- und Anforderungserstellung laut Wasserfallmodell.

Über das eingangs erwähnte „Backfiring“ lässt sich nun ein Zusammenhang zwischen den in Kapitel 3.2 ermittelten ungewichteten Function-Points und der erwarteten Programmgröße herstellen. Die aktuellen Umrechnungsfaktoren können unter <http://www.spr.com/library/OLangtbl.htm> abgerufen werden. Wir gehen von einer Implementierung in Java aus, für die als Faktor 53 SLOC/UFP (SLOC = „Source Lines of Code“, UFP = ungewichtete Function-Points) gilt. Hieraus folgt, dass die Programmgröße etwa 7,1 KLOC („Kilo Lines of Code“) beträgt. Dies erlaubt nun eine Darstellung des normierten Aufwandes und der normierten Projektdauer abhängig von ungewichteten Function-Points und des durch die Projekteffizienz bestimmten Exponenten. Natürlich kann dann auch hier explizit die Änderung angegeben werden, wenn sich die ungewichteten Function-Points linear und der Exponent additiv ändern. Die Projekteffizienz wird durch fünf Kriterien bewertet. Bei einem durchschnittlich effizienten Projekt ergibt sich mit deren Summe für den Exponenten in der Aufwandsberechnung nach [Boe00] der Wert 1,0997.

Die Projekt- und Produktrahmenbedingungen werden durch Korrekturfaktoren repräsentiert. Es handelt sich in der frühen Schätzphase („Early Design“) um sieben Kriterien. Von diesen werden hier nur diejenigen bewertet, die unabhängig vom Projekt sind und nur an Produktanforderungen hängen (siehe [Boe00]). Die anderen Kriterien werden „neutral“ mit dem Faktor 1 bewertet. Man erhält dann den Wert 1,33 als linearen Korrekturfaktor.

Wendet man nun das Formelwerk an, so erhält man als **normierte** Werte für den Aufwand 33,76 Personenmonate und für die Entwicklungsdauer 11,24 Monate. Weiter liefert [Boe00] auch eine Verteilung der Dauer und des Aufwands nach Phasen, sowie eine Verteilung der Aufwände je Aktivität in den einzelnen Phasen. Im Detail ergibt sich die Aufteilung aus Tabelle 3.

	Projektphase	-1-	-2-	-3-	-4-
	Aktivität	Pläne und Anforderungen	Produktentwurf	Programmierung	Integration und Test
<b>Aufwand (PM)</b>	-1- Anforderungsanalyse	1,06	0,75	0,91	0,19
	-2- Produktdesign	0,40	2,29	1,82	0,37
	-3- Programmierung	0,07	0,73	12,01	2,43
	-4- Testplanung	0,07	0,28	0,91	0,16
	-5- Verifikation und Validierung	0,14	0,36	1,48	2,25
	-6- Projektbüro	0,33	0,66	1,42	0,54
	-7- Konfiguration und QS	0,06	0,14	1,37	0,55
	-8- Handbücher	0,13	0,43	1,18	0,53
	<b>Summe</b>	<b>2,26</b>	<b>5,64</b>	<b>21,09</b>	<b>7,03</b>
<b>Dauer (Monate)</b>		1,76	2,59	6,13	2,52

Tabelle 3

Als Gesamtsummen hat man damit – nun **inklusive** Aufwand für Plan- und Anforderungserstellung – für den Gesamtaufwand 36,03 Personenmonate und die Gesamtdauer 12,99 Monate.

Diese Darstellung ist aus den Tabellen in [Boe00] abgeleitet und somit diskret. Eine stetige Weiterentwicklung unter Anwendung einfacher mathematischer Näherungen ist möglich und durchaus sinnvoll. Insbesondere führt dies zu exakteren Aussagen in einer Kosten-Nutzen-Analyse oder ermöglicht eine Sensitivitätsanalyse des angenommenen Projektszenarios zur Kontrolle der Stabilität des zugrunde gelegten Modells.

### Kosten-Nutzen-Analyse

Für eine beispielhafte Kosten-Nutzen-Analyse auf Basis obiger Grundlagen werden folgende Größen betrachtet:

- die Anzahl der Rentner, die mit dem System im Jahr verwaltet wird

- der Aufwand in Personenmonaten pro Jahr, der benötigt wird, um einen Rentner mit den bisherigen Möglichkeiten (z.B. manuelle Errechnung der Auszahlungsbeträge, manuelle Erstellung der §22-Mitteilungen) zu verwalten
- der Aufwand, der mit Hilfe des neuen Systems hierfür anfällt.

Die Effizienzsteigerung pro Rentner und Jahr ergibt sich dann durch die Differenz zwischen dem bisherigen und dem neuen Aufwand zur Verwaltung eines Rentners pro Jahr. Die Verwaltungstätigkeiten und die Entwicklung der neuen Rentnerverwaltung werden mit unterschiedlichen Tagessätzen bewertet. Der Break-Even wird errechnet durch das Verhältnis des Entwicklungsaufwandes zur Effizienzsteigerung für alle verwalteten Rentner – mit den jeweiligen Tagessätzen bewertet.

Unter den plausiblen Annahmen, dass der Tagessatz für die Entwicklung ca. das 1,7-fache des Tagessatzes für die Verwaltung und das Einsparungspotential etwa 1/2 Stunde pro Rentner und Jahr durch die Rentnerverwaltung betragen, erhält man für den oben errechneten Entwicklungsaufwand von etwa 36 Personenmonaten ungefähr 20.000 Jahre/Rentner. als Break-Even-Größe. Das heißt, bei etwa 20.000 Rentnern hätte sich der Aufwand bereits nach einem Jahr amortisiert.

Als zweites Szenario unter sehr schlechten Bedingungen kann angenommen werden, dass man sich beim funktionalen Umfang um den Faktor zwei verschätzt hat und die Projekteffizienz sehr schlecht ist. Dann erhält man als Break-Even etwa 60.000 Jahre/Rentner. Bei den angenommenen 20.000 verwalteten Rentnern hätte man somit eine Amortisationszeit von drei Jahren.

## Fazit

Moderne Projektmanagement- und Analysemethoden liefern ein umfangreiches und in der Praxis auch verwendbares Instrumentarium, um einen gewünschten Funktionsumfang auch formal in einer frühen Phase zu beschreiben sowie fundierte Schätzungen über den erwarteten Entwicklungsaufwand abzuleiten. Zudem sind – ist ein solches Instrumentarium einmal installiert – unterschiedliche Szenarien leicht zu untersuchen, sodass auch während des Projektesverlaufes sehr gut Kennzahlen des Projektes ermittelt werden können.

Einfache Kosten-Nutzen-Betrachtungen können dann zu fundierten Investitionsentscheidungen führen, die einen vordergründig hohen Entwicklungsaufwand von mehreren Jahren wieder in Relation setzen zum erwarteten Nutzen, der in der frühen fachlichen Analyse entsprechend verankert sein muss.

Gerade das Umfeld des behandelten Beispiels der Rentnerverwaltung liefert noch vielfältige Möglichkeiten der Effizienzsteigerung. So kann das System ausgebaut werden für die Verwaltung von Lohnsteuerkarten und Lohnsteuerbescheinigungen, es kann um die Verwaltung von Pfändungen und Sterbekassenbeiträge sowie um weitere automatisierte Kommunikation erweitert werden. Hier gibt es noch erhebliches Potenzial für Aufwandsreduktion, das sicherlich genutzt werden muss, um dem – gerade im Finanzdienstleistungsbereich – steigenden Kosten

druck begegnen zu können. In jedem Fall sollte dieses Beispiel eingehend verdeutlichen, dass die Einführung moderner Projektmanagement- und Analysemethoden lohnt.

*Der Autor: Martin Schriefer ist Senior-Consultant bei der viadico ag, Filderstadt.*

### **Literaturverzeichnis**

- [Boe00] BOEHM, Barry W.; et al.: *Software Cost Estimation with COCOMO II*, Upper Saddle River, New Jersey, USA: Prentice Hall, 2000
  
- [Bun04] BUNDSCHUH, Manfred; FABRY, Axel: *Aufwandschätzung von IT-Projekten*, Bonn: mitp-Verlag, 2004
  
- [Bur97] BURKHARDT, Rainer: *UML – Unified Modeling Language*, Bonn: Addison Wesley Longman Verlag GmbH, 1997
  
- [Hen02] HENRICH, Dr. Andreas: *Management von Softwareprojekten*, München: Oldenbourg Wissenschaftsverlag GmbH, 2002