

## Artikelserie Kommunalsoftware auf dem Prüfstand<sup>1</sup>

(Auszug aus der Veröffentlichung im OKKSA-Newsletter Nr. 16 vom 04.12.07)

### Teil 6: Währungen, Prozente und Fristen – Kann der Rechner richtig rechnen?

(von Dr. Uwe Schwochert)

Die Frage, ob eine einfache arithmetische Berechnung auf dem Computer zum richtigen Ergebnis führt, klingt zunächst obsolet. Ja, da gab es mal die Meldung mit dem Pentium Bug, aber die Auswirkungen dieses und anderer Prozessorfehler auf ein Finanzverfahren im täglichen Einsatz waren praktisch nicht relevant.

Aus Sicht der fachlichen Verfahrensprüfung spielen tatsächlich weniger die Rechenkünste des Prozessors eine Rolle. Viel wichtiger für den finanzwirksamen Verfahrenseinsatz ist die programmtechnische und algorithmische Umsetzung der durch Gesetze und Regelungen vorgeschriebenen Berechnungsvorschriften.

Das entsprechende Kriterium lautet:

[FÜ3.1] Berechnungen, die ein Programm durchführt, erfolgen mathematisch korrekt und folgen den geltenden Gesetzhkeiten. (MUSS-Kriterium)

Damit ist eigentlich eine Selbstverständlichkeit formuliert. Es liegt allerdings nicht nur an der Kriteriensystematik, dass diese Forderung explizit in den OKKSA Katalog aufgenommen wurde.

Tatsächlich, das zeigt sich in Prüfungen immer wieder, ist die algorithmische Abbildung gesetzlicher Vorgaben nach wie vor ein häufiges Problem auch modernster Finanzprogramme. Es würde an dieser Stelle zu weit führen, auf alle aus der Prüfpraxis bekannten Problemfälle einzugehen. Beispielhaft seien die Berechnung von Abschreibungen und resultierenden Restbuchwerten, die Ermittlung von Gewerbesteuerzinsen unter Berücksichtigung geleisteter Zahlungen und die Korrektur von Stundungszahlungen bei geänderter Abgabenschuld genannt.

In diesen und anderen Fällen wird immer wieder deutlich, wie wichtig nicht nur der Test der mathematischen Richtigkeit der entwickelten Programme sondern auch die Prüfung der richtigen Umsetzung gesetzlicher Vorgaben ist. Dabei geht es vor allem um folgende "Schwachstellen":

#### *Datums- und Fristberechnungen*

Für die Zuverlässigkeit von Finanzprogrammen ist neben der *Plausibilität von Datumswerten* (siehe Artikel Nr. 4 dieser Serie) auch die richtige Fristberechnung wichtig. Dabei sind weniger die einfachen Datumsberechnungen (1. Februar + 30 Tage = ?) als vielmehr die durch Gesetze und Normen vorgegebenen Spielregeln dieser Berechnungen problematisch. Typische Fragestellungen sind:

Was ist ein "voller" Monat?

Wie sind drei Arbeitstage in einem föderalen Feiertagssystem zu ermitteln?

Welche Rolle spielt die Uhrzeit bei taggenauen Berechnungsgrundlagen?

Entwicklerseitig ist neben der Kenntnis der Datumsmechanismen der jeweiligen Softwareentwicklungsumgebung auch eine hinreichende Sensibilität bzgl. der jeweiligen Rege-  
lungsgrundlagen gefragt.

---

<sup>1</sup>Hinweis: Die in der Artikelserie genannten Kriterien stammen aus der aktuellen Fassung des "OKKSA Anforderungskatalogs für Fachprogramme in der Öffentlichen Verwaltung - Teilbereich Fachübergreifende Programmanforderungen", Version 3.1

Im Zweifelsfall muss aber auch der Anwender des Programms die zu Grunde liegenden Regelungen kennen, damit er kritische Sonderfälle erkennen und die Ergebnisse des Programms überwachen kann.

### *Zinsberechnungen*

Zeitbezogene Zins- und Abgabeberechnungen setzen zunächst natürlich auch eine richtige Fristenberechnung voraus. So muss z. B. bei Tageszinsen die komplexere 30/360-Tage-Regel angewendet werden, wodurch vom Kalender abweichende Fristen entstehen. Zusätzlich kommt es hier regelmäßig zum Problem der Berechnungstransparenz. Da in den meisten Fällen ein Kunde bzw. ein Bürger betroffen ist, muss diesem der entstehende summarische Betrag anhand der einzelnen Berechnungssätze und Fristen hergeleitet werden. Das erfordert eine tabellarische Übersicht mit Zeitintervallen, Beträgen, termingebundenen Zins-/Abgabesätzen und den einzelnen Teilbeträgen.

Genau hier liegt dann häufig das Problem, wenn es nachträglich zu Änderungen und Rückrechnungen kommt: ab einer bestimmten Zahl von verschiedenen Berechnungssätzen und Änderungen sprengt die schriftliche Herleitung (z. B. auf dem Steuerbescheid) die vom Programmierer vorgesehenen Möglichkeiten des Ausdrucks.

Für den Programmanwender heißt das: Bei der Softwarebeschaffung möglichst komplexe Anwendungsfälle testen! Denn im täglichen automatisierten Betrieb führt (fast) jeder entsprechende Formfehler zum Unverständnis beim Adressaten und damit meist zu komplizierten "Reklamationen" des Endkunden / Bürgers.

Reklamationen und Rückfragen kann es natürlich auch bei richtigen Berechnungen und übersichtlicher Herleitung geben, für den Sachbearbeiter ist es aber ein großer Unterschied, ob er sich im Beschwerdefall auf seine Software verlassen kann oder nicht.

### *Rundungen*

Mit der Währungsumstellung wurde in vielen Finanzabteilungen der Begriff "Summenrundungsfehler" bekannt. Diese Fehler sind ein Spezialaspekt der Tatsache, dass in einem Finanzprogramm immer wieder Werte übereinstimmen müssen, die auf verschiedenem Wege ermittelt und an unterschiedlichen Stellen auf ganze Cent-Beträge gerundet werden. Buchhalterisch ist zwar die Ausbuchung von rundungsbedingten Kleinbeträgen kein großer Akt, das Problem ist aber, dass diese im Verlauf des Jahres an verschiedensten Stellen und auch sehr häufig auftauchen können.

Hier ist die Sensibilität des Programmierers für dieses Thema gefragt. Er muss die möglichen Rundungsabweichungen vorhersehen und deren Ausbuchung automatisiert unterstützen. Nur so wird sein Verfahren auch in größeren Verwaltungen praktikabel sein.

### *Fazit*

Auch wenn arithmetische Berechnungen an sich kein besonderes Problem für moderne EDV-Verfahren darstellen, steckt doch der Teufel im Detail. Speziell bei gesetzesbezogenen Fristen und darauf aufbauenden Finanzberechnungen sowie bei Rundungsabweichungen ist nicht zuletzt auch die Kompetenz und Sensibilität des Anwenders gefragt, um mögliche Problemfelder zu erkennen. Entscheidend ist jedoch die Fähigkeit des Programms, komplexe Berechnungen und Herleitungen übersichtlich und prüfbar zu dokumentieren.

**Nächster Teil der Serie: Passwortrestriktionen – darf Sicherheit erzwungen werden?**