

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

(Auszug aus der Veröffentlichung im OKKSA-Newsletter Nr. 14 vom 06.09.2006)

## Teil 4: Dateneingabe – muss ein Programm alles akzeptieren?

(von Dr. Uwe Schwochert)

Es ist heute selbstverständlich und im Sinne einer rechtssicheren Programmnutzung unumgänglich, dass Fachprogramme die Plausibilität von Nutzereingaben überwachen und so helfen, Datenerfassungsfehler zu vermeiden. Das dient nicht nur dem Anwendungskomfort sondern soll auch verhindern, dass der Anwender mit seinen Eingabebefehlen Daten im Programm unbrauchbar macht.

Was genau hat ein Programm im Rahmen der Plausibilitätskontrollen zu leisten? Die in der Privatwirtschaft anzuwendende Prüfnorm IDW PS 880 fordert die Prüfung der Felder auf gültige Formate und Einhaltung von Grenzwerten sowie die Prüfung der Existenz der erfassten Konten- und sonstiger Nummern (Punkt (18)). Im Bereich des öffentlichen Rechnungswesens könnte der Anspruch des Anwenders auf Plausibilitätsprüfungen u. a. aus § 27 Abs. 5 Pkt. 1 GemHVO NRW abgeleitet werden, wonach die Daten in den Programmen "vollständig und richtig erfasst, eingegeben, verarbeitet..." werden sollen. Auch wenn diese Formulierung sich zunächst auf die Verfahrensanwendung bezieht, setzt sie doch implizit auch das Vorhandensein geeigneter Software voraus.

Im fachübergreifenden Anforderungskatalog wurde versucht, das Thema Plausibilitätsprüfungen zu präzisieren. Dazu wurde zwischen formaler und inhaltlicher Plausibilität unterschieden. Das Kriterium für die formale Plausibilität lautet:

*[FÜ2.5] Das Programm erkennt formale Fehleingaben und verhindert deren Weiterverarbeitung. Dies betrifft insbesondere die Eingabe von Werten außerhalb formal gültiger Wertebereiche, wie*

- *zu große und zu kleine Zahlen,*
- *zu lange Zeichenfolgen,*
- *falsche Datumswerte,*

*sowie die Verwendung der richtigen Eingabezeichen und -formate:*

- *Buchstaben/Zeichen/Zahlen/Ja-Nein-Angaben,*
- *Eingabeformate, zum Beispiel für Kontonummern.*

*Offensichtliche logische Beziehungen zwischen verschiedenen Eingabefeldern werden dabei berücksichtigt. (MUSS-Kriterium)*

Es sollen also alle Eingaben, die ohne besondere "Hintergrundintelligenz" als falsch erkannt werden können, auch abgewiesen werden. Das betrifft zu einem großen Teil Tippfehler. Auch die Jahreszahl 1006 sollte durch ein Buchhaltungsprogramm nicht als Datumsangabe akzeptiert werden. "... verhindert deren Weiterverarbeitung" meint eine möglichst frühzeitige Fehlermeldung bzw. die sofortige Abweisung beim Eingeben. Im Zusammenhang damit steht auch die Anforderung, dass das Programm Interpretationen der eingegebenen Werte für den Anwender sichtbar macht.

---

<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".

Im Kriterienkatalog sind neben den Kriterien auch Rechts- und Normungsgrundlagen genannt, aus denen die genannten Kriterien abgeleitet werden.

Momentan (März 2006) wird der Anforderungskatalog entsprechend den Vorgaben des OKKSA-Vereins durch das Fachgremium aktualisiert.

Beispiel: In ein Feld "Ende der Betriebszugehörigkeit" wird "5.9" eingegeben.

Das Programm soll

- (1) diese Eingabe direkt nach Verlassen des Feldes abweisen, wenn Datumseingaben in dieser Form nicht gültig sind,
- (2) andernfalls die Eingabe in eine vollständige und eindeutige Datumszahl (z. B. "05.09.2006") umwandeln und dies dem Anwender direkt nach der Eingabe sichtbar machen,
- (3) überprüfen, ob das eingegebene Datum evtl. im Widerspruch zu einem Wert "Beginn der Betriebszugehörigkeit" auf der gleichen Eingabemaske steht.

Es ist beim aktuellen Stand der Programmierertechnik nicht mehr akzeptabel, wenn ein Programm formale Eingabefehler der Anwender akzeptiert oder dadurch sogar eine fehlerhafte Datenverarbeitung / Inkonsistenzen zulässt.

Etwas komplizierter ist die Frage der inhaltlichen Plausibilität der Benutzereingaben. Das Kriterium dazu lautet:

*[FÜ2.6]Das Programm erkennt und meldet, wenn die eingegebenen Daten trotz formaler Richtigkeit nicht korrekt weiterverarbeitet werden können. Gründe dafür können sein:*

1. Die Weiterberechnung führt zu Werten außerhalb gültiger Wertebereiche;
2. es werden offensichtlich gesetzliche Vorgaben verletzt;
3. der eingegebene Wert steht im Widerspruch zu sonstigen im Programm hinterlegten Daten (zum Beispiel im Widerspruch zu Wertetabellen oder auf vorherigen Masken erfassten Werten). (KANN-Kriterium)

Hier spielt die Intelligenz der Programme (bzw. ihrer Programmierer) eine größere Rolle. Nahe liegend ist, dass z. B. eingegebene Nummern von Konten und Kostenstellen mit den vorhandenen abgeglichen werden. Etwas komplizierter wäre z. B. eine Überwachung, ob eingegebene Buchungswerte zu einer Überschreitung verfügbarer Mittel führen würden. Die Komplexität dieser Plausiprüfung erschließt sich auch Menschen nicht immer... Bei OKKSA ist dies ein Thema der spezielleren Kriterienkataloge für Finanzsoftware.

Bei der **internetorientierten Programmnutzung** sind Fehlermeldungen bei der Eingabe nicht immer sofort möglich. Die für die Plausiprüfung notwendigen Programmmodule können sehr komplex sein und stehen damit im Widerspruch zum Gebot der Vermeidung aktiver Programmkomponenten auf dem Client-PC. Hier ist eine maskenorientierte Plausiprüfung durchaus akzeptabel. Umso wichtiger ist jedoch, dass der Anwender nach dem Absenden der Eingabewerte ein **eindeutiges Feedback** über die erfassten Werte bzw. deren Fehler erhält.

Beispiel: Bei einer E-Government-Anwendung erfasst der Kunde Werte der Zählerablesung. Die Zählernummern haben das Format "xxx.xxxx". Wenn das Online-Formular das Eingabeformat nicht direkt "erzwingt", sollte es doch im Ergebnis die Interpretation von Werten wie "000056" oder "56" anzeigen (formale Plausi). Darüber hinaus sollte die Existenz einer entsprechenden Zählernummer beim Kundenkonto überwacht werden (inhaltliche Plausi).

Betrachtet man die Konsequenzen, die sich aus dem Einsatz von Programmen mit unzureichenden Plausiprüfungen ergeben, so kann die Ausstattung der Software mit entsprechenden Mechanismen nicht allein Sache des Programmentwicklers sein. Der Schaden

entsteht zunächst für die Anwenderorganisation. Und hier wird entschieden, welche Rolle Plausibilitätsprüfungen als Aspekt der Softwarequalität bei Beschaffung der Verfahren spielen.

**Nächster Teil der Serie: Wie langsam darf ein Programm sein?**