

Kommunalsoftware auf dem Prüfstand

(Auszug aus der Veröffentlichung im OKKSA-Newsletter Nr. 22 vom 14.12.2011)

*Teil 11: Zuverlässigkeit, Datensicherung, Service: Wer kann das überprüfen?
(von Dr. Uwe Schwochert).*

Neben den Programmfunktionen "ab Werk" spielt auch das **Nutzungsumfeld** eines Programms eine nicht geringe Rolle für eine ordnungsgemäße und sichere Vorgangsbearbeitung seitens des Anwenders. So müssen z. B. Daten regelmäßig gesichert und Programmfehler behoben werden. Langfristig bestimmen sich hieraus wesentliche Erfolgs- und Risikofaktoren der Datenverarbeitung.

Diese Aspekte werden natürlich primär vor Ort und in der Vertragsgestaltung zwischen Programmanwender und Programmanbieter geregelt. Vor Ort wird entschieden, wann und wie Sicherungen durchgeführt werden. Im Vertrag wird geregelt, wie seitens des Programmentwicklers auf entdeckte Programmfehler reagiert wird.

Trotzdem spielen diese Aspekte auch in den Kriterien des fachübergreifenden Anforderungskataloges eine Rolle. Hier geht es darum festzustellen, ob "ab Werk" grundsätzliche Konzepte für diese Vor-Ort-Funktionen bereitstehen.

Dies beginnt mit der Frage, ob programmseitig ein Konzept zur Datensicherung vorgesehen ist:

[FÜ07.01] Zusammen mit dem Programm wird ein dokumentiertes **Konzept zur Datensicherung und -wiederherstellung** der im Programm gespeicherten Fachdaten und Einstellungen bereitgestellt. [MUSS-Kriterium] ¹

Es ist klar, dass die Datensicherung in den meisten Fällen mit Hilfe von datenbank- oder systemseitigen Dienstprogrammen erfolgt. Doch diese Sicherung muss auch programmseitig vorgesehen sein. Dabei geht es insbesondere auch darum, dass im Fall einer **Rücksicherung** das Programm mit der Situation eines "veralteten" Datenbestands konfrontiert wird und hier eine inhaltliche Nachbearbeitung ermöglicht. Wie diese erfolgt, hängt von der Art der bearbeiteten (Finanz-)vorgänge ab. Mit **FÜ07.01** wird zunächst erwartet, dass es ein dokumentiertes Konzept nicht nur für die Datensicherung sondern auch für die konsistente Wiederherstellung von Daten gibt.

Neben dem Schutz vor Datenverlust spielt aber auch die **konsistente Daten- und Codebasis** eine wichtige Rolle für die Vor-Ort-Programmanwendung. Insbesondere nach Fehlern und Programmabstürzen soll der Benutzer verlässlich mit dem Programm weiterarbeiten können. Auch bei dieser Situation geht es nicht nur um das Management der Programmanwendung vor Ort. Mit OKKSA FÜ.B wird verlangt, dass solche Situationen auch seitens des Programmentwicklers vorher angedacht werden.

Die Problematik besteht hier darin, dass Bedien- und Programmfehler, die vorhergedacht sind, meist gar nicht auftreten. Denn jeder gute Programmierer baut in den Code Abfangroutinen für falsche Eingaben und sonstige vorhersehbare Fehlersituationen ein.

¹ Die in der Artikelserie genannten Kriterien stammen aus der aktuellen Version 4.00 des "OKKSA Anforderungskatalogs für Fachprogramme in der Öffentlichen Verwaltung - Teilbereich Fachübergreifende Programmanforderungen (FÜ.B)". Im Kriterienkatalog sind neben den Kriterien auch weitere Rechts- und Normungsgrundlagen genannt, aus denen die genannten Kriterien abgeleitet werden.

Mithin geht es hier vor allem um die **Fehlersituationen**, die der Programmierer nicht vorhergesehen hat. Hier muss ein anderer Ansatz her, der es ermöglicht, den Zustand vom Daten und Programm unabhängig von einer vorab bekannten Fehlersituation zu beurteilen.

Seitens des Programmentwicklers sind also Fragen zu stellen, wie Welche **IT-Komponenten** könnten unerwartet zu einem ungünstigen Zeitpunkt ausfallen?

Welche Arten von **Inkonsistenzen** können im Datenbestand und in der Programmumgebung entstehen?

Welche Situationen sind für den Nutzer nicht überschaubar?

Ein Beispiel für den letzten Punkt stellt bei der **internetorientierten Programmnutzung** der durchaus wahrscheinliche Ausfall der Internetverbindung dar. Hieraus ergibt sich die Notwendigkeit, dass (1) jeder per Nutzereingabe ausgelöster Vorgang für den Nutzer sofort sichtbar bestätigt wird (z. B. Zusammenfassung nach einer Online-Bestellung) und (2) auch falls diese Bestätigung nicht mehr angezeigt werden konnte, der Benutzer nachträglich den erzielten Status überprüfen kann (z. B. Bestellübersicht nachträglich anschauen usw.).

Egal ob Programmabsturz, Stromausfall, Netzwerkausfall – jeder Anwender erlebt fast täglich solche Situationen. Beim verantwortungsvollen Softwareeinsatz spielt es deshalb eine große Rolle, dass der Programmentwickler den Anwender bei der Überprüfung des ordnungsgemäßen Programmablaufs unterstützt:

[FÜ07.03] *Das Programm unterstützt Möglichkeiten, mit angemessenem Aufwand die **Daten wiederherzustellen**, die durch Programmfehler, Systemabstürze, Stromausfälle und sonstige Ereignisse beschädigt werden. Dort, wo folgenschwere Inkonsistenzen im Datenbestand entstehen und unbemerkt bleiben könnten, bietet das Programm Möglichkeiten, den Datenbestand auf Defekte hin zu überprüfen.*

Der Benutzer wird durch die Dokumentation über das Verhalten bei Systemabstürzen und Fehlern informiert. [MUSS-Kriterium]

Falls also durch einen Systemausfall, einen Programmfehler oder auch durch einen manuellen Eingriff (z. B. seitens des Programmsupports) die Konsistenz des Programms oder seiner Datenbasis in Frage gestellt ist, soll es (halb-)automatische Mechanismen oder zumindest Anleitungen geben, wie der Anwender wieder zu einer verlässlichen Anwendungs-umgebung kommen kann. Elemente dieser Mechanismen könnten sein:

- automatische Überprüfung der Datenbankindizes,
- automatische Überprüfung der Programmdateien,
- Zugriff auf Internet-Ressourcen (FAQ, ...),
- Hinweise zu einer Reparaturinstallation.

Bei all diesen Unterstützungsmechanismen wird es jedoch auch bei einem zuverlässigen Entwicklungsprozess Programmfehler vor Ort geben, die der Anwender nicht mit eigener Hilfe beheben kann. Hier sind erweiterte Systemreparaturen, Programmierarbeiten oder das Einspielen von Patches erforderlich. Auch wenn die Regelungen dieser Fälle zunächst Sache eines belastbaren **Supportvertrages** sind (s. z. B. EVB-IT), setzt die verlässliche Bearbeitung entsprechender Situationen seitens des Supports doch auch entsprechende anwenderübergreifende Konzepte voraus. Genau um diese geht es mit dem nächsten Kriterium:

[FÜ07.05] Für das Programm existiert ein dokumentiertes Verfahren zur Behebung von Programmfehlern unter Zugriff auf die Supportwege des Programmentwicklers inklusive definierter Eskalationsoptionen im Fall des Misserfolgs. [KANN-Kriterium]

Für eine Verfahrensprüfung soll also dokumentiert werden,

- über welche **Kommunikationswege**,
- mit welchen **Supportmaßnahmen** und
- unter Verfügbarkeit welcher **Eskalationsoptionen**

der Anwender seinen Anspruch auf funktionsfähige Software umsetzen kann. Es ist nahe liegend, dass die entsprechenden Mechanismen auch genutzt werden, um nicht nur Fehler sondern auch Programmverbesserungen und -weiterentwicklungen anzuregen.

Die zum Tragen kommenden konkreten Supportoptionen (Welche Kommunikationswege stehen jeweils zur Verfügung? Mit welchen Antwortzeiten ist zu rechnen?) sind Sache der konkreten Vertragsgestaltung. Mit dem Kriterium wird aber vorab gefordert, dass es dokumentierte Kommunikationswege, Reaktionszeiten und Eskalationsoptionen gibt.

Nächster Teil der Serie: Produktbeschreibung, Bedienerhandbuch, Systemdoku – will das jemand lesen?