



Whitepaper

Modernisierung geschäftskritischer Microsoft-Access-Anwendungen

Ein neuer Lebenszyklus für Software im Unternehmen

Große Microsoft-Access-Anwendungen sind in vielen Unternehmen ein stiller, aber tragender Bestandteil operativer Prozesse. Sie steuern Planung, Controlling, Freigaben, Auswertungen oder Reporting und sind über Jahre hinweg in den Fachabteilungen gewachsen. Oft funktionieren sie zuverlässig – und werden gerade deshalb selten strategisch hinterfragt.

Die eigentlichen Risiken liegen heute weniger in der Datenhaltung, die häufig bereits auf SQL Server basiert, sondern in Architektur, Technologie und personeller Abhängigkeit. Access-Anwendungen sind lokale Windows-(Fat)-Clients, basieren auf VBA und nutzen eine Vielzahl impliziter, Access-spezifischer Mechanismen, die sich nicht ohne ein spezialisiertes Verfahren und bewährte Basisbibliotheken in moderne Webarchitekturen übertragen lassen. Unternehmen stehen damit vor einer strategischen Entscheidung: Ersatz durch Standardsoftware, Neuentwicklung oder Migration. Während eine Neuentwicklung für kleine, überschaubare Access-Anwendungen sinnvoll sein kann, ist sie bei großen, individuell gewachsenen Systemen mit erheblichen Risiken verbunden. Die Migration verfolgt hier einen anderen Ansatz: Sie bewahrt funktionierende fachliche Logik und überführt sie in eine moderne Architektur – als Basis für einen neuen Software-Lebenszyklus.



Unsichtbare Schlüsselsoftware im operativen Kern

Abseits strategischer Programme existieren in vielen Organisationen Anwendungen, deren Bedeutung sich erst dann vollständig zeigt, wenn man sie sich wegdenkt. Sie erscheinen in keiner Digitalstrategie, stehen in keinem Investitionsplan und tauchen in keiner IT-Roadmap auf. Und doch würde ohne sie der operative Betrieb spürbar beeinträchtigt. Sehr häufig handelt es sich dabei um Anwendungen auf Basis von Microsoft Access. Was als pragmatische Lösung in einer Fachabteilung begann, hat sich über Jahre – teilweise über Jahrzehnte – zu einem zentralen Anwendungskern entwickelt. Diese Systeme sind akzeptiert, vertraut und tief in die Arbeitsweise des Unternehmens eingebettet. Ihre Zuverlässigkeit erzeugt ein Gefühl von Stabilität, das selten hinterfragt wird. Gerade diese Stabilität ist trügerisch. Denn sie beruht auf Annahmen, die in einer modernen IT-Landschaft zunehmend nicht mehr gelten.

Wo die eigentlichen Risiken liegen

In vielen Unternehmen wurde die Datenhaltung der Access-Anwendungen in die zentrale IT verlagert. Die Migration auf SQL Server sorgt für Stabilität, Backup-Strategien und kontrollierten Betrieb. Diese Maßnahmen sind sinnvoll, adressieren jedoch nur einen Teil des Problems. Das eigentliche Risiko liegt nicht in den Daten, sondern in der Anwendung selbst. Große Access-Anwendungen bestehen nicht nur aus Tabellen und Abfragen. Sie nutzen eine Vielzahl Access-spezifischer Mechanismen, die über Jahre tief in die Anwendung eingewoben wurden. Formular- und Ereignislogik, implizite Datenbindungen, automatische Kontextmechanismen sowie Abfragen, die Logik und Datenzugriff miteinander vermischen, prägen das Verhalten der Anwendung. Für Anwender ist dieses Verhalten selbstverständlich, technisch ist es jedoch hochspezifisch. Access stellt diese Funktionalität implizit zur Verfügung. Genau diese Implizitheit macht die Anwendungen im Alltag effizient – und ihre Ablösung so anspruchsvoll.

Personelle Abhängigkeit als strukturelles Risiko

Hinzu kommt die personelle Dimension. In vielen Fällen liegt das Wissen über Struktur, Sonderfälle und fachliche Feinheiten bei wenigen Personen, häufig in den Fachabteilungen. Die Dokumentation ist unvollständig, die Einarbeitung zeitaufwendig, eine Weiterentwicklung riskant. Solange diese Personen verfügbar sind, wirkt die Situation beherrschbar. Fällt dieses Wissen jedoch weg, entsteht ohne Vorwarnung ein akutes Unternehmensrisiko. Dieses Szenario ist kein Ausnahmefall, sondern ein wiederkehrendes Muster.

Neuentwicklung: sinnvoll – aber nur bis zu einer klaren Grenze

Wird dieses Risiko erkannt, liegt der Gedanke an eine Neuentwicklung nahe. Moderne Technologien, saubere Architekturen und langfristige Wartbarkeit sprechen dafür. Für kleine Access-Anwendungen ist dieser Weg heute realistisch. Systeme mit wenigen Dutzend Formularen und klar umrissener Logik lassen sich gut neu entwickeln. Moderne KI-Werkzeuge können dabei unterstützen, bestehende Strukturen zu analysieren, Code zu generieren sowie Tests und Dokumentation zu beschleunigen.

Bei großen, gewachsenen Access-Systemen verändert sich die Ausgangslage grundlegend. Der vollständige Funktionsumfang ist häufig nicht mehr explizit beschrieben, sondern nur noch implizit im System selbst vorhanden. Fachliche Regeln sind historisch gewachsen, Ausnahmen wurden schrittweise ergänzt, das Verhalten hat sich an die Organisation angepasst. In diesen Fällen scheitert eine Neuentwicklung selten an der Technik, sondern an unterschätzter Komplexität und fehlender Transparenz. KI kann Entwicklungsaufwand reduzieren, sie kann jedoch kein vollständiges fachliches Verständnis rekonstruieren.

Standardsoftware: strukturell oft keine Alternative

Auch der Ersatz durch Standardsoftware erscheint auf den ersten Blick attraktiv. Klare Produkte, definierte Roadmaps und kalkulierbare Lizenzmodelle sprechen aus Management-Sicht dafür. In der Praxis zeigt sich jedoch, dass große Access-Anwendungen selten Standardprozesse abbilden. Sie sind hochgradig individualisiert und eng mit der Organisation verwoben. Standardsoftware ersetzt diese Logik nicht, sondern erzwingt Anpassung – organisatorisch, prozessual und kulturell.

Software-Migration: Bewahren und einen neuen Lebenszyklus schaffen

An dieser Stelle rückt die Migration in den Fokus – nicht als technische Notlösung, sondern als strategisches Instrument. Software-Migration bedeutet allerdings nicht, Altes blind zu konservieren. Es bedeutet, funktionierende fachliche Logik bewusst zu bewahren und in eine Architektur zu überführen, die einen neuen Lebenszyklus für die Anwendung ermöglicht.

Erhalten wird nicht das „Alte“, sondern das fachlich Richtige: Regeln, Abläufe und Verhalten, die den Betrieb tragen. Gleichzeitig wird die technische Basis vollständig erneuert. Aus einem lokalen Windows-Client wird eine Webanwendung, VBA wird durch C#/.NET ersetzt, lokale Installationen weichen einem zentralen Betrieb. Software-Migration ist daher kein „Garbage-in-Garbage-out“-Prozess. Schlechte Architektur wird nicht reproduziert, technische Altlasten nicht verewigt. Fehlerhafte Logik wird stattdessen identifiziert und bereinigt.

Architektur als Voraussetzung für funktionale Identität

Eine zentrale Rolle spielt dabei die Wahl der Zieltechnologie. fecher setzt bewusst auf Wisej.NET, weil dieses Framework zustandsbehaftete, ereignisgetriebene Oberflächen ermöglicht und damit das Verhalten komplexer Access-Anwendungen strukturgleich im Web abbilden kann. Implizite Logik wird nicht aufgelöst, sondern tragfähig weitergeführt. Das Ergebnis ist eine Anwendung, die sich für Anwender vertraut anfühlt, für die IT beherrschbar ist und dem Unternehmen einen neuen technologischen Lebenszyklus eröffnet.

KI-gestützte Verfahren werden dabei gezielt eingesetzt – zur Optimierung von Migrationswerkzeugen, zur Mustererkennung in VBA-Logik, zur Fehleranalyse, Qualitätssicherung und UI-Optimierung. KI unterstützt und beschleunigt, ersetzt jedoch weder Verfahren noch Erfahrung oder Verantwortung.

Spezialisierung statt Standardansatz

Die Migration großer Access-Anwendungen ist kein Standardprojekt. Sie erfordert Spezialisierung, Erfahrung mit sehr großen, gewachsenen Systemen und ein methodisches Vorgehen, das auf Wiederholbarkeit und Kontrolle ausgelegt ist. Entscheidend ist dabei eine technische Basis, die die strukturellen Unterschiede zwischen Microsoft Access und modernen .NET-Webarchitekturen systematisch berücksichtigt.

fecher hat sich bewusst auf dieses Szenario spezialisiert und dafür eigene Werkzeuge und Basisbibliotheken entwickelt, die zentrale Access-Konzepte und -Verhaltensweisen in .NET/C# strukturgleich abbilden. Diese spezialisierte Grundlage ermöglicht eine weitgehend automatisierte, nachvollziehbare und risikoarme Software-Migration mit funktionaler Identität.

Ohne eine solche Basis wird jedes Migrationsprojekt zu einem hochgradig individuellen Vorhaben, bei dem zentrale Access-Funktionalität explizit neu modelliert, implementiert und abgesichert werden muss – mit entsprechendem Einfluss auf Aufwand, Dauer und Risiko.

Fazit und strategische Konsequenz

Große Access-Anwendungen sind kein Altlastproblem. Sie werden es erst, wenn sie ignoriert werden. Kleine Anwendungen lassen sich neu entwickeln. Große, geschäftskritische Systeme hingegen nicht ohne erhebliches Risiko. Hier ist Software-Migration keine Option unter vielen, sondern der operative Kern einer gezielten Modernisierung, um Stabilität, Kontrolle und Zukunftsfähigkeit miteinander zu verbinden.

Die entscheidende Frage lautet daher nicht, ob modernisiert werden soll, sondern wann und unter welchen Bedingungen. Jetzt – mit Wissen, Zeit und Optionen. Oder später – unter Druck. Die strukturierte Grobanalyse von fecher schafft die notwendige Transparenz, ohne Eingriff in den laufenden Betrieb und ohne Vorabinvestition. Ohne diesen Schritt verschärft sich die Ausgangslage unnötig.

IMPRESSUM & KONTAKT

Herausgeber: fecher GmbH

Otto-Lilienthal-Str. 12, 63322 Rödermark, Germany

Telefon: +49 6074 80577-00

E-Mail: info@fecher.eu

Web: www.modernizing-applications.de

Geschäftsführer: Günter Hofmann, Andreas Glomm