Mainframe-Anwendersoftware-Endspurt mit Trombose-Strümpfen

24.01.2017, 13:58 | IT, New Media & Software

Pressemitteilung von: *TOOLIB*

Mainframe-Anwendersoftware-Endspurt mit Trombose-Strümpfen
Experten aus dem Altersheim sourcen

Kritische Bestandteile der Systeme von Finanzdienstleistern wie Banken, Versicherungen, Bausparkassen usw. basieren oft auf einigen Systemen, die 30 oder sogar 40 Jahre alt sind.

Vom Gesetzgeber geforderte oder von der Fachabteilung gewünschte Erweiterungen stellen die IT vor große Herausforderungen. Sie muss einen reibungslosen Betrieb gewährleisten und die neuen Funktionalitäten fehlerfrei integrieren.

Da die Altsysteme in Programmiersprachen wie z.B. Cobol geschrieben sind, sind nur noch wenige Experten verfügbar. Die meisten dieser Spezialisten sind bereits im Rentenalter. Die „Expertenlücke“ verknappt das Angebot und diese sind, falls sie überhaupt verfügbar sind, nur zu hohen Stundensätzen zu engagieren.

Gefahren des Nichthandelns

Ein Verbleiben in den Altsystemen birgt Risiken:

1. Non Compliance - Die Gefahr, dass die Altsysteme nicht an neue Gesetzgebungen angepasst werden können oder nur zu prohibitiv hohen Kosten.
2. Zeitweilige Non Compliance - Die Gefahr, dass Altsysteme nicht schnell genug an Gesetzgebungen angepasst werden können, da keine Altsystem-Experten zur Verfügung stehen.
3. Wartungskostensteigerung - Die Gefahr, dass die Wartungskosten der Altsysteme durch gesetzlich oder technisch

notwendige Anpassungen eine kritische Größe überschreiten.
4. Experten nicht verfügbar - Die Gefahr, dass ein Umstieg auf moderne Systeme so spät angegangen wird, dass keine Altsystem-Experten mehr zur Verfügung stehen und dadurch das Risiko der Umstellung massiv steigt.
5. Verpasste Marktchancen - Die Gefahr Markchancen zu verpassen, da die Altsysteme nicht, nur sehr langsam und kostenaufwendig angepasst werden können.

Schwierige Lösung „Nachtrainieren“

Manche mögen sich denken: „Dann hole ich mir halt fähige Programmierer und einen alten Hasen und lasse ihn die Neuen auf die Altsysteme trainieren.“

Die Realität sieht aber leider so aus, dass der Programmierer nicht einfach nur eine neue Sprache lernen, sondern auch in eine neue Programmier-Philosophie einstiegen muss. Die Denke dieser alten COBOL-Welt ist, etwa im Vergleich zu der heutigen Java Welt, sehr unterschiedlich. Ohne eine entsprechende Basis ist von einem erheblichen Aufwand auszugehen, um einen modernen Programmierer „rückwärtskompatibel“ mit Sprachen wie COBOL zu machen. Viele Programmierer, die mit einer „höheren“ 4GL-Sprache angefangen haben, werden auch nicht den Schritt zurück zu einer 3GL-Sprache machen wollen. Diese ist viel mühsamer als die neuen Sprachen. Zudem wirkt sich die zu erwartende kurze Halbwertzeit der Verwendbarkeit des neuen Wissens auch negativ auf die Bereitschaft aus, diese alte Sprache zu lernen.

Schwierige Lösung Automatisierte Konvertierung

Einige Anbieter bieten eine automatische Konvertierung des alten COBOL-Codes in Java an. Bei dieser 1:1 Übertragung werden allerdings alle Schwächen des alten Codes mit übertragen. Die Systemkomponenten, die in COBOL geschrieben sind, aber auch Ergänzungen in Natural oder RPG oder Fortran, sind nicht objektorientiert entwickelt worden.

Zusätzlich ist davon auszugehen, dass noch weitere Arbeiten zu erledigen sind, bevor die neuen Systeme produktiv geschaltet werden können. Man sollte von einem nicht unerheblichen Aufwand ausgehen, für die man teilweise auch wieder Mainframe Experten braucht. Dies kommt u.a. auf die Menge der in COBOL hinterlegten logische Funktionalitäten an, die nach der Übertragung alle kontrolliert und oft auch nachgebaut werden müssen.

Der hauptsächliche Vorteil der automatisierten Konvertierung liegt darin, dass der Softwarecode nach der Konvertierung im Java Code vorliegt. Dadurch können die wesentlich häufiger am Markt vorhandenen Java Developer die Aufgaben übernehmen. Der Nachteil ist, dass dieser Code aber weit davon entfernt ist, die Vorzüge der Programmiersprache zu nutzen und eine skalierbare und flexible Systembasis zu sein.

Die Alternative – Komplettumstellung in geordneten Schritten

Die saubere und langfristig stabilste Alternative ist die schrittweise Umstellung der Systeme nach fachlich zusammenhängenden Funktionsbereichen.

Es macht wenig Sinn, ein Programm nach dem anderen auszuwechseln. Wechselwirkungen zwischen komplex verflochtenen Einzelprogrammen würden diese Art der Umstellung um eine Vielfaches aufwendiger machen. Es geht darum, sowohl die o.g. Wechselwirkungen als auch den Gesamtumfang der Teilumstellung beherrschbar zu halten.

Die Bestimmung dieser logisch zusammenhängenden Bereiche ergeben sich auf der einen Seite funktional, z.B. 37 Programme, die sich mit der Verarbeitung von „Hypothekardarlehen“ befassen. Auf der anderen Seite geht es um die Betrachtung der Einheiten, die oft miteinander interagieren, insb. wenn diese bidirektional Daten austauschen.

Aufbauend auf einer guten Basis an technischen und kaufmännischen Klassen kann ohne Altlasten gestartet werden. Die Vorteile der modernen Java Programmiersprache können vollumfänglich für das neue System genutzt werden.

Für die Umsetzung und die spätere Weiterentwicklung kann das Unternehmen auf die vielen am Markt verfügbaren Java Entwickler zurückgreifen. Mit einer guten Architektur ist das neue System flexibel erweiterbar und zukunftssicher. Die kritischen Altsysteme können nach und nach ersetzt und abgeschaltet werden.

Herausforderungen der schrittweisen Umstellung

Wenn man sich dazu entschließt, eine schrittweise Ablösung vorzunehmen, dann ist es wichtig, dass die Basis stimmt. Ermöglichst diese kein sauberes und effektives Vorgehen, so sind in einer späteren Phase der Ablösung erhebliche Mehraufwendungen die Folge.

Zwei der wichtigsten Bestandteile sind die Architektur und die Basis an kaufmännischen Klassen.

Für die Architektur gilt, dass man hier nur die besten und fachlich versiertesten Experten beschäftigen sollte. Jeder Euro, den man hier vermeintlich spart, wird sich später mindestens 10-fach rächen.

Der Grundstock der kaufmännischen Klassen sichert ab, dass die neuen Programme auch alle relevanten gesetzlichen Bestimmungen berücksichtigen. Dieser ist die gemeinsame Sprache der Rechts-, Fach- und IT-Abteilung und erspart diesen viele Mannmonate an Diskussionen. Der gesetzlich vorgeschriebene Teil steht fest. Erste funktionale Basis-Prototypen können innerhalb weniger Tage erstellt werden. Diese können dann durch die Zusatzanforderungen der Fachabteilungen ergänzt und ausgebaut werden.

Natürlich sind noch viele andere Aspekte zu berücksichtigen. Nicht zuletzt ist das ein umfangreiches Testen der neuen Systeme bevor diese produktiv geschaltet werden. Eine ausreichende Zeit der Parallelverarbeitung und der Vergleich der Ergebnisse beider Systeme ermöglicht einen sicheren Übergang zum neuen System.

TOOLIB als Starthilfe zur effizienten Entwicklung von gesetzeskonformen Programmen

Leider ist es sehr oft so, dass die IT auf keine umfassende Sammlung an konsistenten kaufmännischen Klassen im eigenen Hause zurückgreifen kann. Eine Vergleichsmöglichkeit mit einem externen Benchmark ist auch nicht möglich.

Eine gute Lösung für eine saubere Basis für die Ablösung der Mainframe System ist TOOLIB (Tool and Object Library for Insurances and Banks). TOOLIB ist eine einzigartige Sammlung an kaufmännischen Klassen für die objektorientierte Entwicklung. Die Sammlung umfasst über 1.000 nach einem einheitlichen Schema vormodellierte Java-Klassen, die auf den gesetzlichen Vorgaben (HGB, BGB, AktG, WG, DepotG, VVG etc.) beruhen. Die bestehenden Klassen können sofort eingesetzt und in alle gängige Modellierungstools importiert werden. Die weitere Entwicklung von Klassen nach der TOOLIB-Entwicklungsmethode ermöglicht eine effiziente Entwicklung von gesetzeskonformen und revisionssicheren Systemen.

Portrait

TOOLIB ist ein Zusammenschluss von Mainframe Anwendungssoftware Umstellungs-Experten, die ihre Kunden von der alten Cobol-Welt sicher und effektiv in die neue Java-Welt begleiten.

Der Initiator und Schöpfer von TOOLIB, Eckhardt Krummacker, hat mehr als 30 Jahre Erfahrung bei

Finanzdienstleistern im In- und Ausland als Projektleiter in der Funktion als Business Analyst. Während seiner Arbeit für viele unterschiedliche Finanzdienstleister wurde er als Spezialist für Altsysteme des öfteren mit der Problematik von unflexiblen und wartungsintensiven Altsystemen und deren aufwendige Erweiterung oder Umstellung konfrontiert.

So entstand die Idee zu TOOLIB. Auf Basis der vorhandenen kaufmännischen Klassen von TOOLIB können funktionale Prototypen in Tagen statt in Wochen erstellt werden. Der Gesamtaufwand sinkt erheblich. Die Softwarequalität ist gleichbleibend hoch.
Darüber hinaus bietet TOOLIB eine effektive Vorgehensweise zur gesetzeskonformen Erstellung von Anwendersoftware. Endlich haben Fach-, Rechts- und IT-Abteilung eine gemeinsame Sprache. Mit TOOLIB als Basis können die üblichen notwendigen Abstimmungsrunden massiv reduziert werden. Die Anwendersoftware kann so bedeutend schneller in Betrieb genommen werden.

News-ID: 935593 • Views: 577 (Stand: 24.06.2026)

Link zur Pressemitteilung:
https://www.openpr.de/news/935593/Mainframe-Anwendersoftware-Endspurt-mit-Trombose-Struempfen.html