



EISBERG VORAUSS

Wie Sie Ihre COBOL- oder PL/I-Anwendungen erfolgreich ohne Eisberg in die Zukunft bringen

Was tun, wenn Ihre Spezialisten in Rente gehen? Wenn App Entwickler innerhalb kürzester Zeit ein vorzeigbares Programm auf dem Tablet zeigen oder die Buchhaltung den Kopf schüttelt über die Wartungsrechnung für den Mainframe. Wenn der Vorstand dann auch noch alles strategisch in Frage stellt, wird es Zeit zu handeln. Ihre Aufgabe ist es, dem Eisberg auszuweichen. Lesen Sie in dieser fünfteiligen Serie, wie Sie Ihre „Alt“-anwendungen ohne den Eisberg zukunftssicher machen.

Eisberg! Welcher Eisberg?

Meine eigene Mainframe „Karriere“ startete „erst“ 1998 bei einem amerikanischen Anbieter für Mainframe Performance Tools. Ich erinnere mich an meinen ersten Tag, welcher mit einer Schulung zu IBM IMS/DB und DC geschmückt war. Ich erinnere mich ebenfalls genau daran, dass der Referent den Vortrag endete mit dem Satz: „Das heute Präsentierte ist nur von kurzer Dauer. Weder IMS noch den Mainframe wird man in signifikanter Größe in 10 Jahren noch finden“. Nun ja. Ganz so schlimm ist es doch nicht gekommen. Die ersten Abgesänge auf den Mainframe erschienen allerdings schon viel früher seitens des MIT und erschienen 1984 in der New York Times¹:

„BAILING OUT OF THE MAINFRAME INDUSTRY“

Heute, mehr als dreißig Jahre später, hört man diese Weisheiten immer noch. Regelmäßig wird versucht, die Mainframe-Plattform in den Ruhestand zu schicken. In regelmäßigen Abständen tauchen die sogenannten Mainframe Killer auf. SAP, SOA, HANA, JAVA, LINUX, etc.. Die Lebenserwartung dieser Trends ist nicht immer so lang, wie man prognostiziert. Die Frage die sich den Unternehmen stellt, ist schlichtweg, ob eine COBOL oder PL/I basierte



Plattform überhaupt zukunftsfähig ist. Die Entscheidung den Mainframe für eine offenere und verteilte Umgebung zu verlassen und damit fertig entwickelte und gereifte Anwendungen aufzugeben, ist ein ständiger Kampf zwischen den Verteidigern von eingespielten, bewährten Systemen und denen, die eine Modernisierung und stetige On-the-Edge Technologie vertreten.

Der erste Schritt in jeder Entscheidung den Mainframe zu verlassen ist, die Applikationen von einer zentralisierten, in eine dezentrale Umgebung zu bewegen. Die konstante Erhöhung der Kosten einer Mainframe-Umgebung (hier speziell das MIPS Hochpreis Modell), simultan den Verlust von Mitarbeitern mit Mainframe Know-how und die Verlockungen von angesagten Trends wie z.B. Cloud-basierten Technologien bieten die stärksten Argumente für einen Wechsel.

Im Jahr 2017 führte Harris Interactive² eine Umfrage bei Mainframe-Anwendern zum Thema Zukunft von COBOL und/oder PL/I durch. Unter den Befragten waren Entwickler, IT-Architekten, Berater, Ingenieure und IT-Manager. 1.200 COBOL- oder PL/I-Nutzer antworteten mit folgenden Ergebnissen:

- 85% dieser Anwender geben an, dass COBOL- oder PL/I-Anwendungen strategisch für Ihr Geschäft sind.
- 44% dieser Anwender sagen, dass diese Anwendungen geschäftskritisch sind.
- 90% der Anwender haben Anwendungen im Einsatz, die > 1. Mill LoC (Lines of Code) aufweisen
- 50% dieser Anwendungen haben eine verbleibende Lebensdauer von => 10 Jahren

Warum werden diese Sprachen und der Mainframe immer noch als strategisch wichtig angesehen, wenn es doch angeblich bessere und billigere Alternativen gibt? Und wenn Modernisierungsprojekte durchgeführt werden, warum sind nur wenige davon wirklich erfolgreich?

Eisberg?

Der große Vorteil heutiger IT-Verantwortlicher ist, dass sie im Gegensatz zu Edward John Smith, dem Kapitän der Titanic, volle Sicht haben. Somit ist es für sie wesentlich einfacher den drohenden Eisberg zu umfahren und ihre IT auf erfolgreichem Kurs zu halten.

Seit Jahrzehnten laufen die geschäftskritischsten Anwendungen unter COBOL oder PL/I auf dem Mainframe. Sie werden angepasst, erweitert und vielleicht auch performanter gemacht. Aber grundsätzlich ist es noch das gleiche Stück Code wie vor 20, 30 oder vielleicht 40 Jahren. Wir sehen in diesen geschäftskritischen Applikationen zwei unterschiedliche Ebenen. Zum einen gibt es da die Präsentationsschicht, diese wurde zumeist neugestaltet und entspricht modernsten Standards. Zum anderen gibt es das Fundament der Applikation. Dieses Fundament ist das, was seit Jahrzehnten gewachsen ist und die eigentliche Geschäftslogik beinhaltet. Viele Modernisierungsansätze sehen vor, dieses Fundament neu zu erstellen. Aber genau diese Projekte sprengen oft den geplanten Zeit- und Budgetrahmen, kosten mehr als was sie später als Gewinn einbringen oder scheitern ganz.

Der Grund dafür ist oft, dass das für eine Modernisierung notwendige Wissen darüber was in der Anwendung wie umgesetzt wurde und warum, schlicht und einfach nicht mehr vorhanden ist.

Die Diskussion wird ab dem Zeitpunkt der Anwendungsanalyse sehr interessant. Die erstaunten Augen der Teilnehmer, wenn der COBOL- oder PL/I-Code analysiert wird und sich herausstellt, dass ein erheblicher Teil des Codes ohne funktionellen

Wert mitgeschleppt wird. Üblicherweise zeigen Modernisierungskritiker gerne auf die Risiken eines Re-writings oder einer Modernisierung mit den einhergehenden Abbrüchen oder den daraus resultierenden Gefahren für das Geschäft.

Die Schlussfolgerung

Was also tun, wenn die Spezialisten in Rente gehen? Wenn App Entwickler innerhalb kürzester Zeit ein vorzeigbares Programm auf dem Tablet zeigen oder die Buchhaltung den Kopf schüttelt über die Wartungsrechnung für den Mainframe. Wenn der Vorstand dann auch noch alles strategisch in Frage stellt, wird es Zeit zu handeln.

Der logische Ansatz ist es, das Fundament noch profitabler zu machen und zu optimieren. Zusätzlich muss die funktionierende, bewährte Lösung zukunftssicher sein. Mit dieser vierteiligen Serie möchte ich Ihnen eine mögliche Strategie an die Hand geben, welche Ihnen eine Modernisierung auf der Code-Seite, eine Teilüberführung Ihrer Umgebung oder einen vollständigen Plattformwechsel so effektiv und kostengünstig wie möglich macht.

Lesen Sie in den nächsten Teilen, wie Sie

- **Wissenstransfer:** Das Fundament sichern, wiederverstehen, optimieren und weitergeben.
- **Assessment:** Das Risiko eines Wechsels abschätzen und minimieren.
- **Kostenreduktion:** Die Kosten durch Modernisierung reduzieren.
- **Mehrwert:** Technologiewechsel

Zum Autor

Hans Nickessen, Jahrgang 1966 ist seit 30 Jahren in der IT tätig. Anfangs als Datenbank-Entwickler, später über leitende Vertriebspositionen hin zur aktuellen Tätigkeit als Senior Consultant bei Delta Software Technology GmbH. Als Trusted Advisor betreut er heute Anwender in generellen Fragen der Software-Modernisierung und als spezieller Teilbereich, die Ablösung von IMS-Datenbanken.

Kontaktieren Sie uns

Delta Software Technology GmbH
Eichenweg 16, 57392 Schmallenberg
Deutschland

Telefon +49 2972 9719-0
E-Mail info@delta-software.com

delta-software.com

¹ <https://www.nytimes.com/1984/02/05/business/bailing-out-of-the-mainframe-industry.html>

² <https://www.microfocus.com/future-cobol-apps/>