



ICEBERG AHEAD

How you can successfully bring your COBOL or PL/I Applications into the Future without hitting an Iceberg

What to do if your specialists retire? If app developers show a presentable App on a tablet within a very short time or the accounting department shakes its head about the maintenance invoice for the mainframe again? If the management also questions everything strategically, it is time for you to act. Your job is to avoid the iceberg ahead. In this five-part series you can read how you can future-proof your “old” applications without hitting the iceberg.

Iceberg. Which Iceberg?

My own mainframe “career” only started in 1998 with an US based company with mainframe performance tools in it’s portfolio. At that time a highly profitable business. I remember my first day, which was decorated with a training course on IBM IMS/DB and DC. I also clearly remember that the speaker ended the presentation with the quote: “*What has been presented today will be only short-lived. Neither IMS nor the mainframe will be found in a significant amount in 10 years.*” Well. It didn’t turn out that bad. The first swan songs on the mainframe appeared much earlier by the MIT and appeared in the New York Times in 1984 ¹:

„BAILING OUT OF THE MAINFRAME INDUSTRY“

Today, more than thirty years later, you can still hear these phrases. Attempts are made regularly to retire the mainframe platform. The so-called mainframe killers appear periodically. SAP, SOA, HANA, JAVA, LINUX, etc.. The life expectancy of these trends is not always as long as predicted. The question that



companies have to ask is simply whether a COBOL or PL/I-based platform is sustainable for them at all. The decision to leave the mainframe for a more open and distributed environment and thus abandon fully developed and mature applications is a constant battle between the defenders of well-established, proven systems and those who represent modernization and constant on-the-edge technology.

The first step in any decision to leave the mainframe is to move the applications from a centralized to a decentralized environment. The constant increase in the of costs of for a mainframe environment (especially the MIPS high-price model in this case), the simultaneous loss of employees with mainframe know-how and the temptation of popular trends such as cloud-based technologies offer the strongest arguments for a change.

In 2017 Harris Interactive² conducted a survey of mainframe users on the future of COBOL and/or PL/I. Among the interviewees were developers, IT architects, consultants, engineers and IT managers. 1200 COBOL or PL/I users answered with the following results:

- 85% of these users state that COBOL or PL/I applications are strategic for their business.
- 44% of these users say that these applications are business critical.
- 90% of the users use applications that have >1. Mill LoC (Lines of Code)
- 50% of these applications have a remaining lifetime of => 10 years

Why are these languages and the mainframe still seen as strategically important when there are supposedly better and cheaper alternatives? And when modernization projects are implemented, why are only a few of them really successful?

Iceberg?

The great advantage of today's IT managers is that, unlike Edward John Smith, the captain of the Titanic, they have a full 360° view. This makes it much easier for them to bypass the impending iceberg and keep their IT on a successful course.

The most business-critical applications are running under COBOL or PL/I on the mainframe for decades. They are adapted, extended and perhaps also made more performant. But basically, it's the same piece of code as it was 20, 30 or maybe 40 years ago.

We see two different levels in these business-critical applications. On the one hand there is the presentation layer, which has mostly been re-designed and corresponds to the most modern standards. On the other hand, there is the foundation of the application. This foundation is what has grown for decades and contains the actual business logic. Many modernization approaches provide this foundation to be rebuilt. But exactly these projects often exceed the planned time and budget framework, cost more than what they later profit, or fail completely.

The reason for this is often that the knowledge necessary for a modernization about what has been implemented in the application and why this is simply no longer available.

The discussion becomes very interesting from the point of application analysis. The astonished eyes of the participants when the COBOL or PL/I code is analysed and it becomes clear that a

considerable part of the code is dragged along with no functional value. Modernization critics usually like to point out the risks of rewriting or modernization with the associated crashes or the resulting dangers for the business.

The Conclusion

What to do if your specialists retire? If app developers show a presentable program on the tablet within a very short time or the accounting department shakes its head about the maintenance invoice for the mainframe once again. If the management also questions everything strategically, it is time to act for you.

The logical approach is to make the foundation even more profitable and to optimize it. In addition, the functioning, proven solution must be future-proof. With this four-part series I would like to give you a possible strategy that will make a modernization on the code side, a partial transfer of your environment or a complete platform change as effective and inexpensive as possible.

Read in the next parts how you

- **Knowledge transfer:** Secure, re-understand, optimize and pass on the foundation
- **Assessment:** Assess and minimize the risk of a change
- **Cost reduction:** Reduce costs through modernization
- **Added value:** technology change

About the Author:

Hans Nickessen, born in 1966, has been working in the IT for 30 years. Initially as a database developer, later from managerial sales positions to the current position as a Senior Consultant at Delta Software Technology GmbH. As a Trusted Advisor, he now supports users in general questions of software modernization and, as a special sub-area, the replacement of IMS Databases.

Get in touch with us

Delta Software Technology GmbH
Eichenweg 16, 57392 Schmallenberg
Germany

Phone +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/>