



TEXT SERIES ICEBERG AHEAD

Part 1: 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

TEXT SERIES

ICEBERG AHEAD



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.

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

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

Part 2: Knowledge Transfer: Securing, Re-understanding and Passing on the Foundation

These three procedures are the prerequisite for securing the investments you have made so far, maintaining your applications efficiently and using them optimally. And finally, to successfully modernize these applications and keep them future-proof.

You know the situation: The original developers of the COBOL and PL/I applications are rarely or no longer available. The successors, new teams or external service providers are responsible for the maintenance, modernization or re-implementation. But I have seen the shrug of shoulders so often when questions were asked about the basis of the legacy applications. In order to be able to guarantee smooth day-to-day business, you need a comprehensive and in-depth understanding of your existing applications.

Targeted Application Analysis for Knowledge Transfer, Modernization and Re-implementation

It is very time-consuming to understand large and complex application systems. Figuring out what they're doing is nearly impossible without profound help. With the existing COBOL and PL/I applications, further hurdles is the ongoing development that has been implemented for decades (i.e. software evolution) and the very own language architecture. The older an application gets, the harder it is to understand it. Knowledge transfer, further development or modernization are almost impossible without the understanding.

In each of these cases, however, you not only need reliable information about the current structure and the interaction of the individual components, you must also and above all understand the implemented logic. Static information from compilers or information from data dictionaries is usually not sufficient for this. Aspects such as the orders of

TEXT SERIES

ICEBERG AHEAD

With our solution, the real knowledge of the application is restored by formal and logical abstraction as well as conclusions from the obtained information and the application logic is separated from the technical infrastructure. The knowledge gained from this is presented in different perspectives clearly and language-neutral in a user-friendly interactive front end.

Customized analyses and reports are also possible. Large and complex COBOL or PL/I applications are analysed automatically (adapted to your requirements). In this way you regain the knowledge of the application and, at the same time, you receive a current documentation.

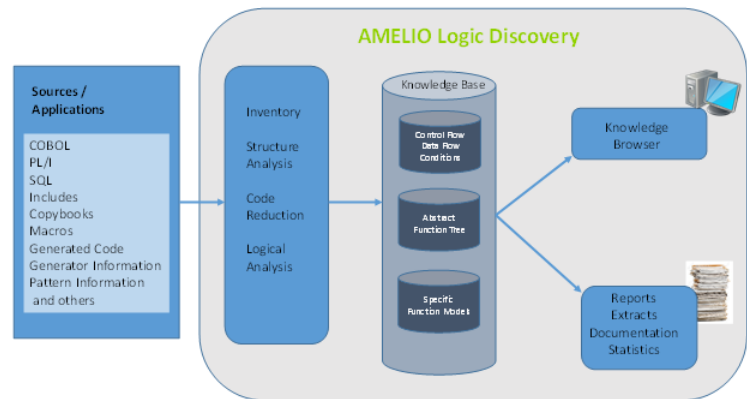
Your Advantages

AMELIO Logic Discovery can be perfectly tailored to your needs.

- You receive a visualization of the application logic as well as reliable statements about the quality, size and complexity of the applications.
- AMELIO Logic Discovery supports you in eliminating the deficits caused by dwindling know-how and ensures the transfer of knowledge. Your dependence on external specialists is reduced and the productivity of development is improved.
- With this reliable basis for decision-making, you are able to reduce costs and risks for modernization and re-implementation as well as for development, maintenance and quality assurance.
- AMELIO Logic Discovery provides a sound foundations for the requirements engineering if you want to refactor or rewrite the application or sub-applications.

The added value for you:

The regained knowledge and understanding of your old application secures the foundation and enables the necessary and loss-free transfer of knowledge.



In this way you secure your decades of investments and can continue to efficiently maintain/optimally use your applications, modernize them and keep them future-proof.

Part 3: Assessment or how to assess and minimize the Risk of a Change

Your COBOL or PL/I applications contain enormous investments, a lot of company-specific know-how and decades of experience just like this famous ship back in the 20th century which was the crown of development. Why put it on risk by not doing what needs to be done. The mature applications run error-free and resource-optimized. They support business-critical processes, are adapted to new requirements and new functions are added time and again. If such an application should be cleaned-up, refactored, modernized or transformed, extensive knowledge of the application is absolutely mandatory. In order to regain the fundamentally important knowledge about your applications and to gain an understanding of the functionalities, you have to perform an individualized analysis. Standard analyses do not help you understand the application, since every modernization has its own starting point and objective. Usually these are a platform change, an architecture transformation, a clean-up, a technology exchange, or something else. These different objectives each affect certain but always different parts of the applications and places in the code and have different effects on the application components, their interfaces and their behaviour.

For a reliable assessment of the measures, procedures and risks, you need targeted analyses that



provide the necessary knowledge about the applications regarding the project goal. You need well-structured facts instead of unmanageable data collections. Without this knowledge, the project threatens to become a case that exceeds the time and budget frame or, in the worst case, fails completely..

Don't guess assess

A tailor-made assessment of your application offers the possibility of reliably determining the critical points and evaluating them precisely. This avoids the risk of your modernization projects getting out of hand in terms of budget and target date.

70-80% of the projects exceed exactly these requirements and fail. In order that your planning and decisions are not only based on estimates and assumptions, but on facts and reliable knowledge, you should have performed a comprehensive assessment of the applications by independent experts before starting a project.

From HOW to WHAT as required

In the process of understanding an application, gathering information about how it was implemented is the first necessary step. Statements about the actual functions cannot yet be made on this basis. Here we use our AMELIO solution in a personalized assessment tailored to your needs. Not only information about how the application was implemented is collected and evaluated, but also what was implemented.

The determination and evaluation of all “sticking points” (so-called points of interest - POI) is extremely important. Every project, either a platform change, an architecture transformation, a clean-up, a technology exchange, or something else, affects specific but always different places in the application components, their interfaces and their behaviour. These effects define the POIs. They are fully determined, analysed and evaluated.

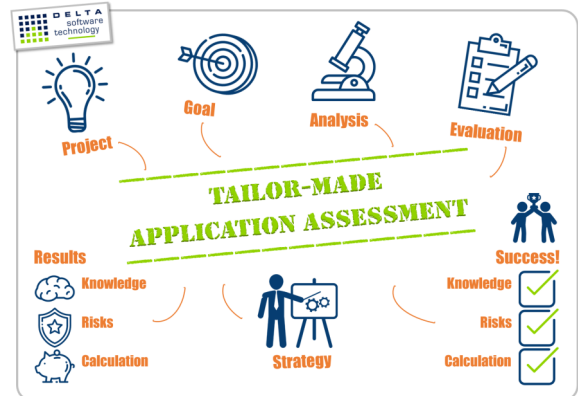
With our experience and the comprehensive and in-depth analyses by using our innovative tools, we help you to develop the best project strategy and to plan your project realistically. We tailor the assessment precisely to your goals and requirements. All applications with all its source components are analysed comprehensively and

target-oriented with regard to the project task. All connections and dependencies (internal and external) are clearly presented and reliably assessed. You receive all analysis results as a clear and meaningful documentation:

- An overview of the applications, all components, the connections and the dependencies.
- Statistics help you to evaluate the complexity of

your applications and / or individual components.

- Proposals for clean-up (cleaning, refactoring) based on the POI's help you to improve readability and maintainability and to reduce the code
- Overview of all POI to be processed in the project
- ... and much more ...



Your Advantages

With these targeted analyses and a tailor-made assessment of your applications, we help you plan your projects reliably and reduce the risks and costs.

- Secure choice of a project strategy
- Realistic scheduling
- Reliable calculation of costs
- Secure planning and implementation of the project
- Well-founded selection of a suitable test strategy
- Comparison of technical alternatives
- Audit-proof implementation of the project

If you would like to find out more about the advantages, you can find more information here:

<https://delta-software.com/link.php?en=6287>

Part 4: Reducing costs through modernization

Modernization of COBOL or PL/I applications

Changing large, productive IT applications is associated with high costs, risks and an enormous amount of time. The race in the areas of agility, adaptability, and the adoption of new requirements is becoming increasingly difficult. The costs for the maintenance and further development of the application are increasing.

In discussions, my customers repeatedly highlight two topics as the reason for this:

1) Benefits and costs diverge from each other :

The maintenance nightmare due to missing documentation, the loss of experienced employees and missing specialists prevent the further development of the applications. Time-critical changes, for example, have to be laboriously integrated into the system. A lack of know-how is compensated for by more and more workarounds. The downward spiral

begins. Changed requirements such as legal regulations are becoming major challenges. The costs increase due to increased maintenance effort and further development. Due to the deteriorating quality, the existing landscape can no longer be flexibly adapted to the changes that are being demanded ever more quickly. New required business models suffer as a result. The legacy systems are becoming more and more complex and, in the worst case, hinder the progress of the entire company.

However, flexibility and quick adaptability of the core applications are the most important prerequisites for software business processes to be optimally supported. However, the COBOL and PL/I applications have grown over decades, and they have been continually adapted and expanded to meet new requirements. What the user needs now is a clean-up of these applications.

2) The technology change

The implementation of new technologies promises more flexibility, more options and lower costs. To do this, however, the existing technology has to be replaced. Such projects were often started in the past, but the old technology was never completely replaced. Finally, several technologies are used for one task. However, such a coexistence of different technologies can lead to problems over time: The performance of the application decreases, the effort for maintenance increases and, if necessary, a migration to a new platform that is necessary for other reasons is prevented. To counteract this, it is necessary to exchange or consolidate technologies. But often, the technologies to be exchanged are firmly anchored in the existing applications.

However, the challenge with clean-ups and technology changes does not necessarily lie in the changes themselves. As a rule, the problem is the bulk of the application modules. In addition, there is the difficulty (and sometimes even the impossibility) of isolating individual applications from one another in such a way that the changes can be performed out safely.



And these applications are typically large, complex, and business-critical. A manual implementation of the clean-up or the technology change is too time-consuming and risky. A manual implementation would mean an enormous testing effort and impair regular maintenance. Thus, a process must be implemented that automatically provides you with a change and, if necessary, also allows a partial procedure.

AMELIO Modernization Platform

The AMELIO Modernization Platform offers you fully automated, step-by-step modernization that is perfectly tailored to your individual modernization requirements and that provides you with measurable results at an early stage. The targeted transformation tailored to your needs is performed in such a way that it also takes your future application architecture into account. In this way, you can safely replace old techniques, development paradigms and components. You get clearly structured, stable and flexible applications. Applications that can cope with the next changes and the changes after that and are therefore future-proof.



Your advantages:

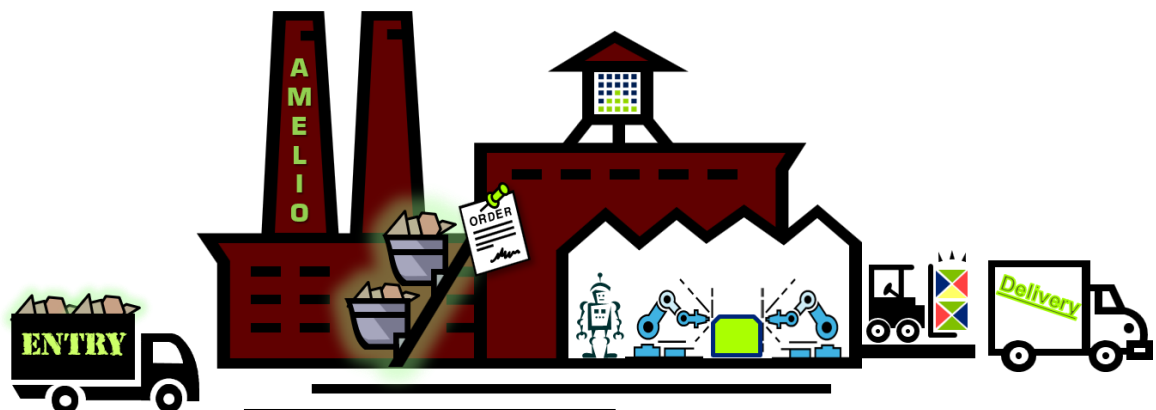
- Strong automation, up to 100% in most areas
- Future security for your applications
- No vendor lock-ins, no external dependencies
- Tool-based solutions instead of anonymous

outsourcing solutions

- Step-by-step approach with early and clear results
- Excellent quality

AMELIO supports your modernization project safely and flexibly with the following options:

- **Clean-up**
Take advantage of the opportunity to not only examine your applications for dead code. See the superfluous parts of your application, visualize their links, discover and clean-up code from the past that is no longer required or let them examine how syntax problems affect your applications.
- **Change or consolidation of technologies**
A task such as the transfer from hierarchical IMS/DB to relational DB2 seemed impossible for a long time. Nevertheless, several of our customers took this step successfully with our help. Or use AMELIO to make your applications available to new areas of application quickly and easily.
- **Modernization of the application architecture**
Even if this option does not save costs in the short term, in the long term it will mean that your application landscape can maintain the pace that is required. This includes changes to individual application layers, e.g. the modernization of the front ends for the use of new (web) technologies, the implementation of a



TEXT SERIES

ICEBERG AHEAD



separate data server layer, as well as extensive changes as are necessary when implementing service-oriented architectures (SOA).

The AMELIO Modernization Platform not only helps you to change your COBOL or PL/I applications. It shows you how big the problem actually is. It analyzes all sources including all meta-level components such as macros, copybooks, includes etc. AMELIO performs all changes automatically and 100%. This drastically reduces the test effort required and guarantees the full integrity and functionality of your applications at all times.

The productive lifetime of business functions is often measured in decades. Technologies, on the other hand, have a much shorter lifespan. For such transformation projects, you therefore need flexible tools that change your applications in a targeted manner - without endangering functionality and thus the important day-to-day business.

delta-software.com/amelio

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 97190

delta-software.com