DELTA software technology

AMELIO Logic Discovery

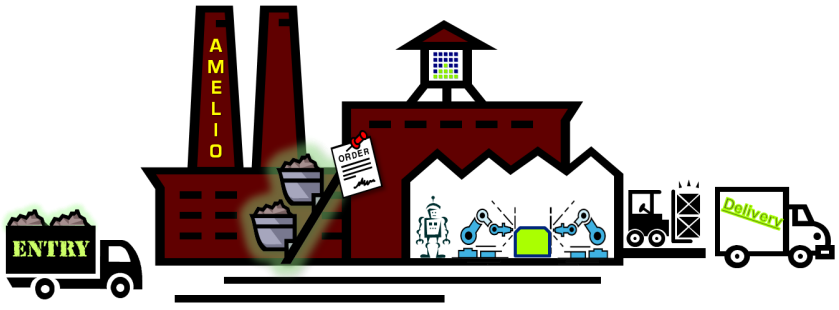
Alles neu macht der Mai?

Legacy-Anwendungen werden von vielen Unternehmen zunehmend als Altlast empfunden, deren Wartung und Weiterentwicklung fehleranfällig und kostenintensiv ist. Ein Neuschreiben der Anwendung ist aufgrund der Größe und ihrer Komplexität jedoch zu teuer und zu risikobehaftet. Mittels Bereinigung, Restrukturierung und Modernisierung wird aus der Altlast ein wertvolles Erbe, das effizient weiter verwendet werden kann. Mit den Werkzeugen der AMELIO-Familie können diese Modernisierungsmaßnahmen passgenau und automatisiert durchgeführt werden, so dass Aufwand und Risiko überschaubar bleiben.

1 Legacy – Altlast oder wertvolles Erbe?

Viele Unternehmen haben über Jahrzehnte ihre unternehmenskritischen Kernanwendungen in COBOL oder PL/I entwickelt. Durch die Pflege und Weiterentwicklung sind große und komplexe Anwendungen entstanden. Diese Anwendungen sind oft unflexibel, ihre Wartung und Weiterentwicklung fehleranfällig und zeitaufwändig. Deshalb werden Legacy-Anwendungen heute oft als Altlast empfunden, obwohl sie viel Wissen über die Unternehmensprozesse enthalten. Die Gründe dafür sind vielfältig: Die Anwendung ist über Jahrzehnte gewachsen, wurde von Entwickler zu Entwickler weitergereicht, nach verschiedenen Paradigmen, Methoden und Versionen einer Sprache entwickelt, verwendet veraltete Technologien und vieles mehr.

Die Anwendungen in einer modernen Sprache nach aktuellen Paradigmen neuzuschreiben ist aufgrund ihrer Größe und Komplexität ein enorm kostspieliges und risikoreiches Unterfangen. Der zu erzielende Gewinn ist dagegen vergleichsweise gering, zumal „nur“ die alte Funktionalität wiederhergestellt wird, aber keine neue hinzugefügt würde.



Factory zur Modernisierung von Legacy-Anwendungen

Statt also die bestehenden Anwendungen „weg zu schmeissen“ und neu zu implementieren, ist es sinnvoller den Code gründlich aufzuräumen, zu bereinigen, zu refaktorisieren und zu modernisieren und so die Les- und Wartbarkeit zu verbessern und eine effiziente Weiternutzung zu ermöglichen.

Mit den Tools der AMELIO Familie können Bereinigungs- und Modernisierungsaufgaben schrittweise und zu 100 Prozent automatisiert

erfolgen, wodurch Aufwand und Risiko für ein solches Projekt minimiert werden.

2 Fit für die Zukunft

Möglichkeiten eine Legacy-Anwendung wieder flexibel zu gestalten und effizient warten zu können gibt es viele. Sie reichen von Bereinigungen, über Refaktorisierungen bis hin zur Modernisierung. In manchen Fällen reicht es aus, einzelne Maßnahmen vorzunehmen, in anderen sollten mehrere schrittweise nacheinander oder kombiniert durchgeführt werden.

2.1 Wissen wiedergewinnen – Verstehen Sie Ihre Anwendung

Bevor weitreichende Änderungen an einer Anwendung geplant und vorgenommen werden können, ist es unablässig zu verstehen was die Anwendung tut, wie sie es tut und welche Zusammenhänge und Abhängigkeiten existieren. Dokumentation, soweit vollständig und aktuell vorhanden, dient eher als Gedächtnisstütze für den ursprünglichen Entwickler der Anwendung. In den meisten Fällen ist sie aber als Grundlage für ein Modernisierungsprojekt ungeeignet.

AMELIO Logic Discovery hilft in solchen Fällen die Anwendungen zu analysieren und Abstraktionen durchzuführen, die das Verständnis der Anwendung erleichtern. Zudem können unnötiger Ballast, Vorschläge für eine Refaktorisierung und die Knackpunkte für eine Modernisierung automatisch ermittelt werden.

2.2 Unnötigen Ballast loswerden

Wird eine Anwendung über Jahrzehnte weiterentwickelt, geändert und gepflegt, so entsteht oft unnötiger Ballast. Code, der eigentlich nicht mehr benötigt wird, aber mit dem bloßen Auge nicht als solcher erkannt wird und deshalb immer weiter mitgepflegt wird. Mittels automatischer Analysen kann solcher Code, Statements als auch Datendefinitionen, erkannt und entfernt werden. In COBOL und PL/I spielen Copybooks und Includes eine wichtige Rolle, deshalb muss auch ermittelt werden, ob der Code ggf. aus einem solchen Modul stammt und ob er immer tot ist oder nur in einigen Programmen.

Auch die verwendeten Programmiersprachen wurden im Laufe der Zeit weiter entwickelt. Die Anwendungen enthalten deshalb oft einen Mix aus neuen Sprachkonstrukten und den alten und weniger gut lesbaren Konstrukten, diese sollten vereinheitlicht werden.

Durch Wartung und Erweiterungen ändern sich auch Schnittstellen. Diese Änderungen wurden aber nicht immer in der gesamten Anwendung nachgezogen. Funktioniert die Anwendung dennoch korrekt, so ist dies eher Zufall. Deshalb gilt es genau zu analysieren, welche Schnittstellen nicht korrekt bedient werden und diese zu bereinigen.

Auch wenn dies auf den ersten Blick nur viele kleine Änderungen sind, die Auswirkung auf die Lesbarkeit ist enorm.

2.3 Neue Strukturen

Jahrzehntelange Wartungsarbeiten und Weiterentwicklungen haben das ursprünglich saubere Design verändert. Zudem haben sich Programmierparadigmen weiterentwickelt. Waren früher monolithische Anwendungen State-of-the-Art, gelten sie heute als schwer wartbar. Mittels Refaktorisierung können besser wartbare Strukturen

geschaffen werden, die heutigen Standards entsprechen. In COBOL- und PL/I-Programmen trifft man besonders häufig auf die Notwendigkeit mehrere kleine Paragraphen oder Prozeduren unter bestimmten Bedingungen zusammen zu fassen oder eigenständige Teile aus besonders großen Programmen in separate Unterprogramme auszulagern. Die Einführung einer Service-Schicht für Zugriffe auf Datenbanken und Files verbessert nicht nur die Lesbarkeit der Programme, sondern erhöht auch die Flexibilität beim Austausch oder Umstrukturierung der zugrundeliegenden Datenbank- und File-Systeme.

2.4 Modernisierung

Im Laufe der Jahre wurden oft neue Technologien, z.B. Datenbanksysteme, eingeführt ohne dass die alten vollständig abgelöst wurden. Daten müssen dann in verschiedenen Systemen konsistent gehalten werden, was die Anwendung inperformant und unflexibel macht. Zudem ist der Wartungsaufwand unnötig hoch. Durch die Vereinheitlichung der Technologien oder den Austausch durch eine neue Technologie kann Performance und Flexibilität gewonnen werden.

3 Ganz automatisch und Schritt für Schritt

Auch Bereinigungs- und Modernisierungsaufgaben erfordern viele und zum Teil tiefe Eingriffe in die Anwendung. AMELIO führt diese Maßnahmen zu 100% automatisiert, nach einem Factory-basierten Ansatz, durch und ermöglicht es so Aufwand und Risiko zu minimieren. Dabei arbeitet AMELIO modell- und regelbasiert. Alle Sourcen einer Anwendung werden zunächst eingelesen und mittels formaler und logischer Abstraktion in Modelle überführt. Die Modelle sind dann die Grundlage für weitere Analysen oder die eigentlichen Transformationen, die mittels Regeln spezifiziert werden. So ist es möglich eine Reihe von Standardanalysen und -transformationen, speziell für die Code-Bereinigung, anzubieten. Zum anderen können projektspezifische Analysen und Transformationen realisiert werden. Das regelbasierte und automatisierte Vorgehen bietet aber vor allem Vorteile in Bezug auf Sicherheit und Qualität. Analysen und Transformationen können jederzeit nachvollzogen und reproduziert werden. Alle Transformationen erfolgen absolut gleichförmig. Transformiert eine Regel den Code einmal korrekt, so tut sie dies immer. Damit müssen im Test nicht alle geänderten Code-Stellen getestet werden, sondern nur die Korrektheit der Regel, womit der Testaufwand drastisch reduziert wird. Alle vorgenommenen Änderungen werden automatisch im Code und in zusätzlichen externen Dokumenten revisionssicher festgehalten. Ein automatisierter, Factory-basierter Ansatz ermöglicht es auch, die Bereinigung und Modernisierung einer Anwendung schrittweise vorzunehmen. So kann man beispielsweise die Anwendung zunächst bereinigen, bevor man weitere Maßnahmen beschließt und umsetzt. Statt einem „Big Bang“ wird die Anwendung Schritt für Schritt fit für die Zukunft gemacht. Am Ende jedes Schritts hat man ein lauffähiges System und kann den Gewinn direkt nutzen.

AMELIO
LOGIC DISCOVERY

DELTA software technology

4 Alles neu macht der Mai?

Jein! Bei einer vollständigen Neuentwicklung einer bestehenden Anwendung steht der Gewinn meist in keinem Verhältnis zu Risiko und Aufwand. Stattdessen ist es viel günstiger die Legacy-Anwendung mittels automatisierter Verfahren zu analysieren, bereinigen, refaktorisieren und ggf. modernisieren. Auf diese Weise wird das in der Anwendung implementierte Wissen wieder effizient nutzbar gemacht. Sollten anschließend funktionale Anforderungen ein Neuschreiben erzwingen, so reicht es aus einzelne Komponenten neu zu entwickeln statt der gesamten Anwendung.

Bestellnummer: MT11088.01 – April 2026

Delta Software Technology GmbH • Eichenweg 16 • 57392 Schmallenberg, Deutschland • Tel +49 2972 9719-0 • E-Mail: info@delta-software.com • www.delta-software.com