

Das Human Capital

Der am meisten unterschätzte Wert bei Modernisierungsprojekten liegt im Human Capital. Viele Unternehmen haben zu spät erkannt, dass sie mit dem Ersatz von Systemen wertvolle Investitionen in Schulung und Wissen ihrer Mitarbeiter verloren haben. Üblicherweise beleben Mitarbeiter das Tagesgeschäft, aber sie schreiben selten Arbeitsprozesse nieder.

Durch Modernisierung in einem System, das die Weiterentwicklung fördert, kann das Unternehmen die Human Resources schützen. Nach Ersetzen eines Systems braucht es zum Teil Jahre, bis dieses Wissen wieder bei den Mitarbeitern aufgebaut worden ist.

Das Idealziel

der Modernisierung ist jede der vier Säulen zu stärken oder zumindest abzusichern. Wenn so vorgegangen wird, sichert das Unternehmen alle kritischen Aktivposten und steigert den Wert der Investitionen durch moderne Lösungen. Das Ergebnis der Modernisierung ist nicht nur gestiegene Produktivität und Innovation im IT-System. Der wichtigste Aspekt der Modernisierung gegen den Ersatz ist, dass dem ein evolutionärer Prozess zugrunde liegt, der Schritt für Schritt gegangen werden kann und dadurch zeitlich und budgetär plan- und kontrollierbar ist.

Migration ist der Prozess, bei dem Daten und Geschäftsprozesse gekapselt in Anwendungen und Programmen sowie der dazugehörige Support von einem Altsystem in eine moderne Entwicklungsumgebung gehiftet wird.

Migration ermöglicht die Sicherung von Ressourcen und Wissen auf einer neuen Plattform. Das Beste dabei ist, dass der Weg der Migration Anwendern und Entwicklern erlaubt, ihr Wissen und ihre Erfahrung weiter auszubauen. Somit ist Migration der attraktivste Weg, die vier Säulen Ihres Unternehmens zu modernisieren.

Wege zur Modernisierung

Der ideale Weg lässt zu, dass bestehende Applikationen weiterarbeiten können und dabei offen für zukünftige Veränderungen sind. Folgende Optionen stehen zur Verfügung:

- Das System auf einem anderen System zu emulieren.
- Neuschreiben und Ersetzen von Anwendungen.
- Webfacing einsetzen, um Anwendungen eine neue Optik zu geben.
- Programme in Sprachen wie Java oder C# transferieren.
- Programme mit einem neuen Compiler umwandeln, der auf bekannter Syntax aufsetzt, aber modernen Code und moderne Applikationen produziert.

Wir wissen, dass die ersten beiden Optionen keine Modernisierung darstellen. Emulationen lösen keine fundamentalen Änderungen in der Situation des Unternehmens aus. Die Anwendung arbeitet wie bisher und ist nicht modernisiert. Zusätzlich ist das Unternehmen an eine Plattform gebunden und von deren Anbieter abhängig. Von dieser Plattform wegzugehen, ist nur mehr durch Ersatz möglich, der all die oben beschriebenen Risiken beinhaltet.

ScreenScraping oder Webfacing ist eine Methode die einige Zeit sehr populär war. Die Anwendungen wurden über ein grafisches Interface angezeigt. Die Bildschirmmasken wurden in ein Web-ähnliches Look-and-Feel konvertiert, ohne sie zu verändern. Diese teure Möglichkeit der Erneuerung macht sich kaum bezahlt, da sie sich in einer scheinbaren Erneuerung erschöpft. Sie ermöglicht keinen oder spärlichen Nutzen und bietet auch keine weitere Erneuerungsmöglichkeit. Wenn die Anwendung geändert wird, müssen alle betroffenen Bildschirme neu erstellt werden. Scraping hat keinen Modernisierungseffekt auf Anwendungen, Daten oder zusätzliche Möglichkeiten, nur das User Interface erscheint modern.

Erneuerung an der Basis

Einzig die Umsetzung des Source-Codes in eine moderne Sprache ermöglicht echte Modernisierung. Wenn das Ergebnis der Code ist, der in einer modernen Entwicklungsumgebung genutzt werden kann, ist man in der Lage, die aktuellen und zukünftigen Anforderungen zu erfüllen.

Die Wahl des Compilers muss gut überlegt werden. Mit vielen Compilern ist man in der Lage, Anwendungen zu produzieren, die auf alle gängigen Datenbank-Plattformen zugreifen können. Beachten Sie bei Ihrer Entscheidung die folgenden wichtigen Fragen:

- Kann die Migration durch eigenes Personal durchgeführt werden oder braucht man dazu externe Entwickler, die die Anwendung nicht kennen?
- Können aus bestehenden iSeries-Anwendungen Web-Services gemacht werden?
- Wie viel Kontrolle und Einfluss lässt der gewählte Migrationsweg zu? Muss die Migration in einem Zug umgesetzt werden oder erlaubt der Prozess einen evolutionären Ansatz, in dem Parameter – wie Anforderungen, Planung und Budget – eine Rolle spielen?
- Was ist für die spätere Wartung nötig? Wird das eigene Personal in der Lage sein, die Wartung durchzuführen? Wenn die Anwendung in eine fremde Sprache wie Java oder C# umgesetzt wird, ist die Wahrscheinlichkeit hoch, dass das Ergebnis von der Entwickler- oder Betreuungsmannschaft nicht gewartet werden kann.

- Können die Daten auf der iSeries bleiben? Wenn sie auf eine andere Plattform geschifft werden müssen, stellt sich die Frage nach den nötigen Werkzeugen und der Schwierigkeit des Unternehmens.
- Welche Programmiersprachen lässt die Umgebung zu? Wird die neue Umgebung das Wissen der Mitarbeiter unterstützen oder müssen RPG-Programmierer geschult oder ersetzt werden?
- Wird der neue Code einfach oder schwierig zu warten sein? Wie sehr hängt das vom Code selbst und den Mitarbeitern ab?
- Wird die entstandene Applikation die Erwartungen in Bezug auf Beweglichkeit, Wartungsfreundlichkeit und Erweiterungsfähigkeit erfüllen?
- Wird die Entwicklungsumgebung die in sie gesetzten Erwartungen erfüllen?

Migrieren statt Ersetzen

Die Migration in Umgebungen wie DotNet oder Java sind eine erstklassige Alternative zu Emulation, ScreenScraping und Ersetzen von Legacy-Systemen. Durch die Weiterentwicklung der Informationstechnologie wird die Erneuerung auch eine Notwendigkeit für Unternehmen, die eine iSeries einsetzen. Unabhängig davon, welcher Weg gewählt wird, müssen die vier Säulen des Unternehmens – Menschen, Prozesse, Anwendungen und Daten – von Anfang an mit in die Überlegungen eingebunden werden.

Fachautor: Christian Neißl