Wichtige Aspekte bei der Wahl des Entwicklungswerkzeuges

Anwendungsentwicklungswerkzeuge entscheiden über den Erfolg der Unternehmens-IT. Deshalb reicht es nicht Aspekte wie Leistungsfähigkeit, Wartbarkeit, Performanz und Personal zu berücksichtigen. Die Entscheidung für ein Werkzeug soll sich langfristig als richtig und erfolgreich beweisen. Dazu müssen auch strategische Aspekte betrachtet werden. Aber welche?

Programmierung für IBM i statt auf IBM i

Maßstab für die Anwendungsentwicklung ist die Qualität der Ergebnisse. User beurteilen diese bei klassischen RPG-Anwendungen eher mangelhaft. Vor allem der Bedienkomfort aber auch der Funktionsumfang und die Integration externer Software, Hardware, Datenbanken, Bilder und Dokumente lassen Wünsche offen.

Auch IBM sieht die Rolle der Power Systems im Wandel. Immer mehr rückt der Aspekt eines intelligenten Datenbankservers in den Fokus (Beispiel: Watson). Diese neue Sichtweise eröffnet der Anwendungsentwicklung spannende Perspektiven: objektorientierte Programmierung, 4GL-Sprachen, Plattformunabhängigkeit, integrierte Entwicklungsumgebungen, komfortable Debugger, Versionskontrolle uvm. Manche Entwicklungswerkzeuge und-komponenten, wie die iNEXT Suite, sind darauf spezialisiert die Vorzüge der DB2 wie z.B. Sicherheit, Schnelligkeit, Skalierbarkeit, Administrierbarkeit auch bei der Anwendungsentwicklung im .NET-Umfeld zu nutzen.

Softwareentwicklung & Modernisierung

Natürlich ist es effizient, den Lebenszyklus der vorhandenen RPG-Anwendungen durch Modernisierung zu verlängern. Für eine mittel- bis langfristige Perspektive reicht ein einfaches GUI oder Web-Interface nicht. Funktionaler Mehrwert ist der Schlüssel zum Erfolg.

Klick-Mich-Werkzeuge nach dem Player-Prinzip zeigen meist einen einfachen Weg. Ihre schnelle Erlernbarkeit erkaufen sie durch einen begrenzten Funktionsumfang. Individuelle Anforderungen scheitern oft an diesen Leistungsgrenzen. Außerdem ergibt sich eine starke Abhängigkeit vom Anbieter. Dieser entscheidet über die verfügbaren Funktionen, was sich im eingeschränkten Leistungsvermögen der erstellten Lösungen widerspiegelt. Verschwindet der Player oder der Anbieter vom Markt, ist eine Weiterentwicklung der erstellten Lösungen nahezu unmöglich.

Im Gegensatz dazu bieten API-basierte Konzepte uneingeschränkte Möglichkeiten für funktionalen Mehrwert. Sie nutzen programmierbare Schnittstellen für den Informationsaustausch zwischen Zielsystem und IBM i. Die iNEXT Suite greift so auf Systemfunktionen (CL), RPG-Anwendungen und Daten des IBM i zu. Diese Informationen werden auf der Zielplattform (.NET) verarbeitet. Auch hierfür gibt es keine Einschränkungen. Jede Neuerung im .NET-Framework steht automatisch in der iNEXT Suite zur Verfügung. Das gibt langfristig Sicherheit in der Anwendungsentwicklung.

Vorteilhaft ist auch die geringe Abhängigkeit vom Anbieter. iNEXT-Kunden erhalten beispielsweise den Sourcecode ihrer Anwendungen und des iNEXT-Frameworks. Damit können sie ihre Lösungen individuell weiterentwickeln. iNEXT-Lösungen bieten so passgenaue Modernisierung und Softwareentwicklung mit funktionalem Mehrwert. So werden die Ansprüche der User, die aus dem täglichen Umgang mit Anwendungen wie Adobe Acrobat, Microsoft Office, Facebook, Amazon, WhatsApp resultieren, wirklich zufriedengestellt.

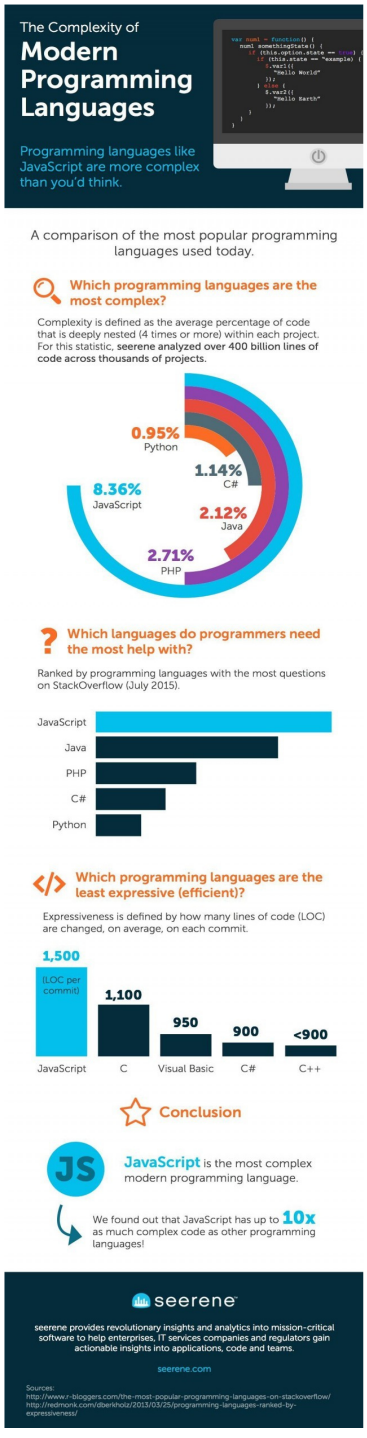
Desktop – Web – Mobile?

Desktopapplikationen bestechen durch Leistungsfähigkeit, Integration und Performance. Diese sogenannten Rich Clients eignen sich deshalb vor allem für umfangreiche Businesslösungen wie ERP, PPS, WaWi, CRM etc. im internen Betrieb. Sie können problemlos in Citrix-Umgebungen betrieben werden. Selbst im Außendienst sind sie heute in Verbindung mit Notebooks oder Hybrid-Geräten einer Web-Tablet-Lösung überlegen.

Soll hingegen vielen externen Usern ein begrenzter Funktionsumfang zur Verfügung gestellt werden, dann bieten sich Weblösungen an. Typische Beispiele sind Webshops, Kfz-Konfiguratoren, Vergleichsportale, Wissensdatenbanken. Hierbei werden lokale Installationen vermieden. Eine gute Performance zu erreichen, ist oftmals aufwändig und hängt stark von Hardware und Infrastruktur ab.

Mobile Applikation beschränken sich ebenfalls auf wenige Kernaufgaben, die per Smartphone überall nutzbar sein sollen. Sie werden einfach mittels AppStore installiert. Bekannte Beispiele sind Wetterberichte, Börsenkurse, Benzinpreise, Nachrichten. Über Webservices lassen sich so Unternehmensinformationen auch vom System i bereitstellen. Komfortable Generatoren, wie in der iNEXT Suite, vereinfachen den Einstieg in diesen zukunftsträchtigen Entwicklungsbereich.

Desktop, Web, Mobile - Jedes Userinterface hat seine Berechtigung. Erstrebenswert ist daher ein Entwicklungswerkzeug, das alle drei bedient wie z.B. ML-Softwares iNEXT Suite. Selbst wenn kurzfristig z.B. keine Weblösungen und mobilen Apps geplant sind, kann dies mittelfristig ein dringendes Thema werden. Warum also bei der Wahl des Entwicklungswerkzeuges Einschränkungen hinnehmen, wenn es flexiblere Alternativen gibt?



Welche Programmiersprache bringt Kontinuität für die Zukunft?

Um sämtliche funktionalen Anforderungen der User erfüllen zu können, reicht RPG trotz verschiedener Weiterentwicklungen nicht mehr aus. Perspektivisch fehlt zudem der plattformübergreifende Einsatz. Die Folge: Wird das IBM System i irgendwann abgelöst, ist die RPG Software wertlos. Die Alternative: neue Funktionalitäten unter .NET entwickeln, denn dieser Code kann zukünftig auch für andere Plattformen angepasst werden.

Hinzu kommt, dass sich IBM auf den Hardwaremarkt konzentriert und die Plattform für Open Source Angebote öffnet. IBM übernimmt hierfür auch die Versionspflege was bei Releasewechseln zusätzliches Gefahrenpotential birgt.

Abgesehen davon bieten Open Source Sprachen wie JavaScript auf dem System i oftmals auch keine befriedigende Lösung. Laut einer Analyse von Seerene ist JavaScript 10 mal komplexer als andere Sprachen. Im Vergleich zu C# sind die Supportanfragen der Entwickler 3 mal häufiger und der Codierungsaufwand fast doppelt so hoch.

Bleibt der Blick über den IBM-Tellerrand in Richtung Java, C, C++, C#, Visual Basic uvm. Unter den modernen, objektorientierten Sprachen gehören Java und C# (mit den Vorgängern C++, C) zu den bedeutendsten. **Java** entstand 1995 und ist **eine Sprache für viele Plattformen** (Unix/Linux, Windows, MacOS ..). Das 2001 ins Leben gerufene **C# vereint hingegen viele Sprachen auf einer Plattform, .NET** (Windows, Linux).

C# als die jüngere der beiden Sprachen verfügt über nahezu alle Features des Konkurrenten Java sowie zahlreiche Neuerungen wie z.B. systemnahes Programmieren und Kompatibilität mit anderen .NET-Sprachen wie VB, C++ u.a.. Sie ist im Vergleich zu Java mächtiger, bequemer und griffiger. Sowohl der produzierte Code als auch der Compiler bieten ein besseres Laufzeitverhalten. Vor allem unter Windows ist C# daher die bessere Wahl.

Java's Vorteile liegen primär in der Portabilität. Problematisch ist die stagnierende Weiterentwicklung. Damit fehlt die technologische Perspektive, um für die nächsten 5 – 10 Jahre gewappnet zu sein.

Bei C# betrachtet man die zukünftige Weiterentwicklung als sichergestellt. Microsoft ist mit knapp 86,83 Milliarden US-Dollar (Juni 2015) Weltmarktführer der Softwarehersteller. Der Marktanteil vom Windows-Betriebssystem wird auf 92 % geschätzt. Auch die Vergangenheit zeigt mit C und C++ eine langjährige Historie ähnlich wie RPG. Hersteller IBM konzentriert sich inzwischen auf den Hardwaremarkt. Viele Jahre war IBM Weltmarktführer im Serverbereich und gehört noch immer zu den Top 3. Im Bereich Enterprise-Flash-Speicher-Systeme ist sie bis heute Weltmarkführer.

Welches ist die beste Lösung?

Warum nicht die Hard- und Software-Qualitäten von IBM und Microsoft kombinieren?

Genau das leistet die iNEXT Suite. Sie nutzt die vorhandene RPG-Businesslogik. Die UI's werden wahlweise als Rich Client, Web oder Mobile bereitgestellt. Der Zugang zu externen Quellen wie SQL Server, Cloudservices Microsoft Azure, Amazon S3, Google erschließt zusätzliche Informationen.

iNEXT Suite setzt für die Softwareentwicklung auf Visual Studio, C# und .NET von Weltmarktführer Microsoft. Damit steht ein nahezu unerschöpflicher Pool an Technologien und Funktionen bereit, der durch fertige Komponenten von Drittanbietern komplettiert wird. Eine programmierbare Schnittstelle (API) stellt die Verbindung zur IBM i her. Hierrüber werden die vorhandenen Anwendungen, Daten und Systemfunktionen integriert.

Die iNEXT Suite ist das umfassendste Paket für Modernisierung und Neu-/Weiterentwicklung. Funktionale Erweiterungen für den erforderlichen Mehrwert können wahlweise auf der IBM- oder Microsoftseite entwickelt werden. Dieses harmonische Miteinander ermöglicht einen schrittweisen Prozess.

In der Praxis hat sich dieser Weg bereits vielfach bewährt. Viele Unternehmen und Softwarehäuser nutzen iNEXT für die Weiterentwicklung ihrer ERP-, WaWi-, CRM-, PPS- und anderen Lösungen. Neben dem Mehr an Funktionalitäten können .NET-Lösungen lokal, unter Citrix (u.ä.) und in der Cloud betrieben werden. Diese Optionen werden auch für die Unternehmen zunehmend relevant.

ML-Software GmbH
Hertzstraße 26
76275 Ettlingen

+49(0)7243-56550
info@ml-software.com
www.inext-suite.com