



Whitepaper

Anwendungsmodernisierung VEDA JUMP

Die Brücke zwischen Klassik und Moderne

Autor: Harald Kleutgens, VEDA GmbH, harald.kleutgens@veda.net

Stand: 26.12.2012

VEDA - Ihr Lösungsanbieter

Die VEDA GmbH ist einer der führenden Anbieter von Unternehmenslösungen in den Bereichen Human Resources, Finance & Accounting und IT. Von der Beratung über innovative Softwareanwendungen bis hin zu optimalen Infrastrukturkonzepten sowie Outsourcing bietet VEDA einen ganzheitlichen Ansatz, der sich an den Kundenprozessen orientiert.

Inhaltsverzeichnis

Inhaltsverzeichnis	i
1. Prämisse.....	3
2. Strategie - Einsatzgebiete von JUMP.....	4
3. Modernisierung – JUMP der sanfte Modernisierungsprozess	5
4. Migration – „Big Bang“ vermeidbar.....	7
5. Architektur.....	8
6. Transformer	9
7. Entwicklungsumgebung	10
8. Einzelbearbeitung	11
9. Ausführungsumgebung (Runtime).....	11
10. Wartung	12
11. Projektablauf.....	13
12. Benefits.....	13
13. USP - Alleinstellungsmerkmale	15
14. Neuentwicklung - Weiter- und Neuentwicklung	17

Vorbemerkung

Die in diesem Dokument enthaltenen Informationen können ohne Vorankündigung geändert werden und stellen keine Verpflichtung seitens des Verkäufers dar.

Die in diesem Dokument beschriebene Software wird unter einem Lizenzvertrag geliefert und darf nur nach Maßgabe der darin enthaltenen Bedingungen und Vereinbarungen benutzt oder kopiert werden.

Ohne ausdrückliche schriftliche Erlaubnis der VEDA GmbH dürfen für andere Zwecke als den persönlichen Gebrauch des Käufers diese Dokumentation und/oder Software in keinerlei Form mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie, durch Aufzeichnung oder mit Informationsverarbeitungssystemen reproduziert oder übertragen werden.

Die in diesem Dokument gemachten Angaben beziehen sich auf das Produkt

VEDA JUMP – Version 52.20.

VEDA®, VEDAware® und j-ware® sind eingetragene Marken der VEDA GmbH, Alsdorf. Alle anderen verwendeten Namen oder Bezeichnungen können Marken oder eingetragene Marken ihrer jeweiligen Eigentümer sein. Änderungen vorbehalten.

Die VEDA GmbH übernimmt keine Verantwortung für die Inhalte von fremden Websites, die über in diesem Dokument enthaltene Links erreicht werden. Diese Links werden bei der Aufnahme nur provisorisch geprüft und bewertet. Eine kontinuierliche Prüfung der Inhalte ist weder beabsichtigt noch möglich. Die VEDA GmbH distanziert sich ausdrücklich von allen Inhalten, die möglicherweise straf- oder haftungsrechtlich relevant sind oder gegen die guten Sitten verstoßen.

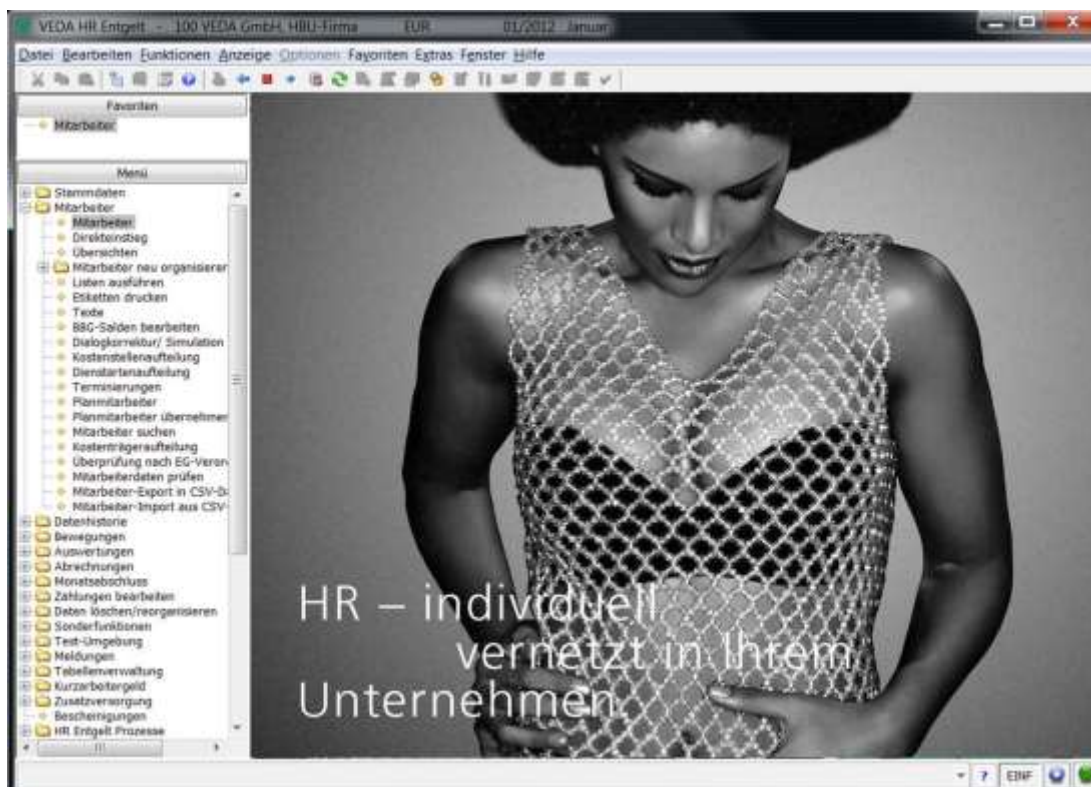
© 2012 VEDA GmbH. Alle Rechte vorbehalten.

JUMP – Java™ unter Midrange Power

Als Software-Hersteller, mit langer Tradition auf IBM i kennen wir die Herausforderung, Anwendungen auf dem Stand der Technik zu halten. Neuentwicklung, aber auch Modernisierung der bewährten Programme ist unser Tagesgeschäft. Wir kennen den Markt, auch weil wir unseren Kunden immer die beste Lösung bieten wollen. Daher ist VEDA JUMP nach Meinung unserer Kunden das beste verfügbare Modernisierungswerkzeug für IBM i Anwendungen nach Java.

VEDA ist seit vielen Jahren ISO 9001 zertifiziert und IBM Premier Business Partner.

Die guten Verbindungen zur IBM und die fast 35-jährige Projekterfahrungen haben es uns ermöglicht, mit VEDA JUMP und iJUMP eine optimale und zukunftssichere Lösung bereit zu stellen.



VEDA schafft Werkzeuge (Tools), die den Anwendungsentwickler entlasten. Neue Technologien nutzbar zu machen bedeutet nicht zwangsläufig die Komplexität zu steigern.

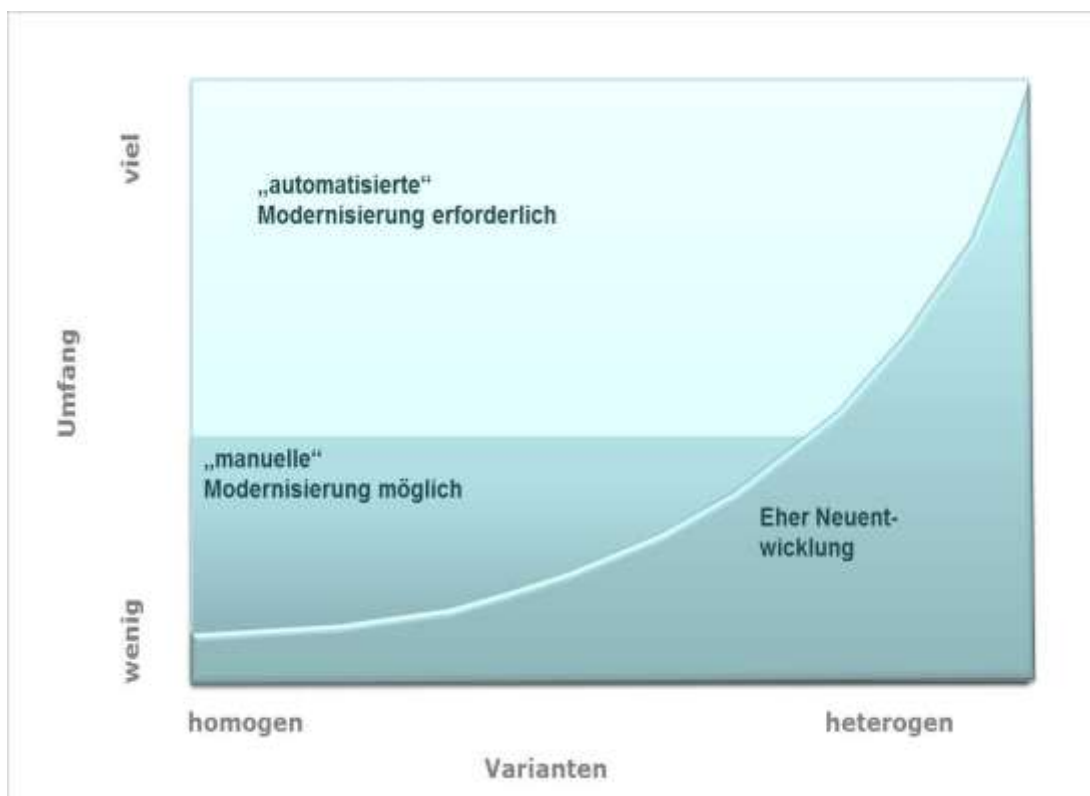
Kernaufgabe der Entwickler ist die Abbildung der Geschäftsprozesse. Dazu bedarf es eines ausgereiften Business-Frameworks.

VEDA hat ein solches Framework geschaffen und entwickelt es fortwährend weiter. Es ist bewährt, weil hundertfach im Einsatz, und allgemeingültig, denn die unterschiedlichsten Anwendungen wurden bereits damit modernisiert.

Ob AS/400, iSeries, System i oder IBM i, diese Plattform begeistert bis heute. Durch Java ist VEDA auch unter Unix, Linux, Windows und selbst auf Apple zu Hause. Java sichert nicht nur die Plattformunabhängigkeit, sondern eröffnet auch den Zugang zu unzähligen Ressourcen und Standardlösungen. Keine andere Entwicklungsumgebung hat diese Verbreitung und Akzeptanz insbesondere für unternehmenskritische Anwendungen.

1. Prämisse

Wenn die Portfolioanalyse der Altsysteme ergibt, dass der Business-Nutzen hoch ist, aber die Benutzerakzeptanz und/oder die Flexibilität nicht mehr ausreichen, dann ist eine Modernisierung sinnvoll.



Eine Eigenentwicklung schafft Differenzierung und damit Marktvorteile. Beides kann eine gute Modernisierungsstrategie erhalten. In die Jahre gekommenen AS/400-Anwendungen fehlt oft eine ausgereifte grafische Benutzeroberfläche (GUI) und die Integration in Fremdsysteme ist schwierig. Beides ist heute zwingend notwendig.

Zur Anpassung des GUI gibt es viele Lösungen, die jedoch oft in Sackgassen führen. Integration beschränkt sich oft auf das, was mit erträglichem Aufwand auf der IBM i lösbar ist. Selten wird eine saubere Brücke nach Java gebaut, die hilft den Weg von der Behelfslösung zum Standard sauber und dauerhaft zu beschreiten.

Ein Werkzeug und Framework, das diese Probleme zukunftsweisend löst und gleichzeitig „State of the Art“ sein will, sorgt bei IBM i-Entwicklern in der Regel für Bauchschmerzen. Zu oft haben sie die Erfahrung gemacht, dass die offenen Technologien mehr Probleme machen als Vorteile bringen.

Woher die zusätzliche Kapazität nehmen, diese neuen Themen zu erlernen und zu beherrschen?

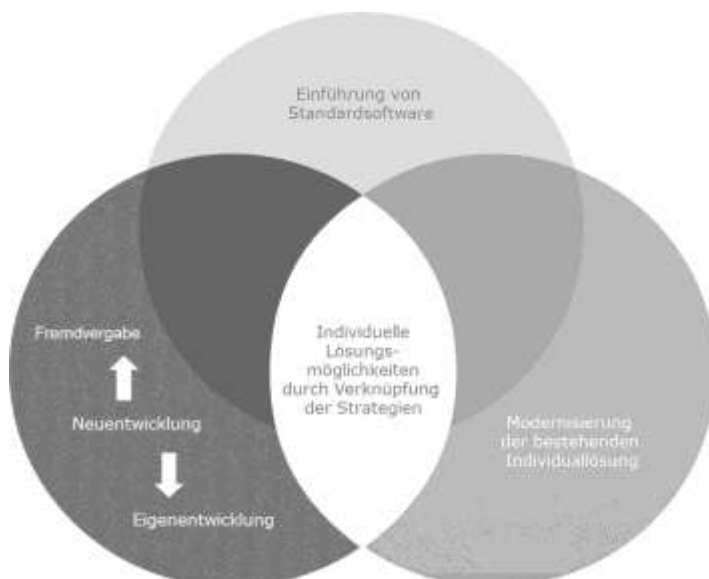
- ▶ Für diese Entwickler ist ein Modernisierungswerkzeug nur nützlich und erfolgreich, wenn es einen schnell erkennbaren Nutzen bringt,
- ▶ den Rahmen der einsetzbaren Ressourcen einhält und
- ▶ zukunftsweisend ist.

2. Strategie - Einsatzgebiete von JUMP

Kunden, die die Plattform IBM i nicht in Frage stellen und die noch über ausreichend Entwicklerkapazitäten für die klassische Anwendung verfügen, nutzen VEDA iJUMP oft nur zur Bereitstellung einer komfortablen, zeitgemäßen Benutzeroberfläche und zur Integration von Client-Komponenten, wie z. B. Microsoft Office. Bei dieser Strategie wird die Altanwendung genauso weiterentwickelt wie bisher. Zusätzlich wird die neue Oberfläche durch die automatische Migration erzeugt bzw. aktualisiert.

JUMP kann ebenfalls die komplette Anwendung nach Java portieren. Mit VEDA JUMP erfolgt die Überführung der notwendigen Quellen und Objekte nach Java vollständig automatisch, kann aber individuell beeinflusst werden.

Ein weiteres Einsatzgebiet für JUMP: Software-Hersteller (ISV), die ihre bewährte IBM i Lösung auch auf anderen Plattformen anbieten möchten, können in RPG weiterentwickeln, testen aber bereits in der automatisch erzeugten Java Welt.



Durch das one-source Konzept wird die Anwendungen plattformübergreifend angeboten. Dabei kann die Anwendung weiterhin in RPG, CLP und DDS entwickelt werden und steht unmittelbar in Java zur Verfügung.

Der migrierte Java-Code ist lesbar und kann direkt weiterentwickelt werden. Generell gilt: Jede Migration kann beliebig oft wiederholt werden, wobei alle Änderungen erhalten bleiben.

Wer vorhandene Lösung übergangsweise einbetten möchte, steckt keine Arbeit mehr in die reine Verschönerung der Altanwendung, sondern nutzen JUMP als Brücke zur Neuentwicklung.

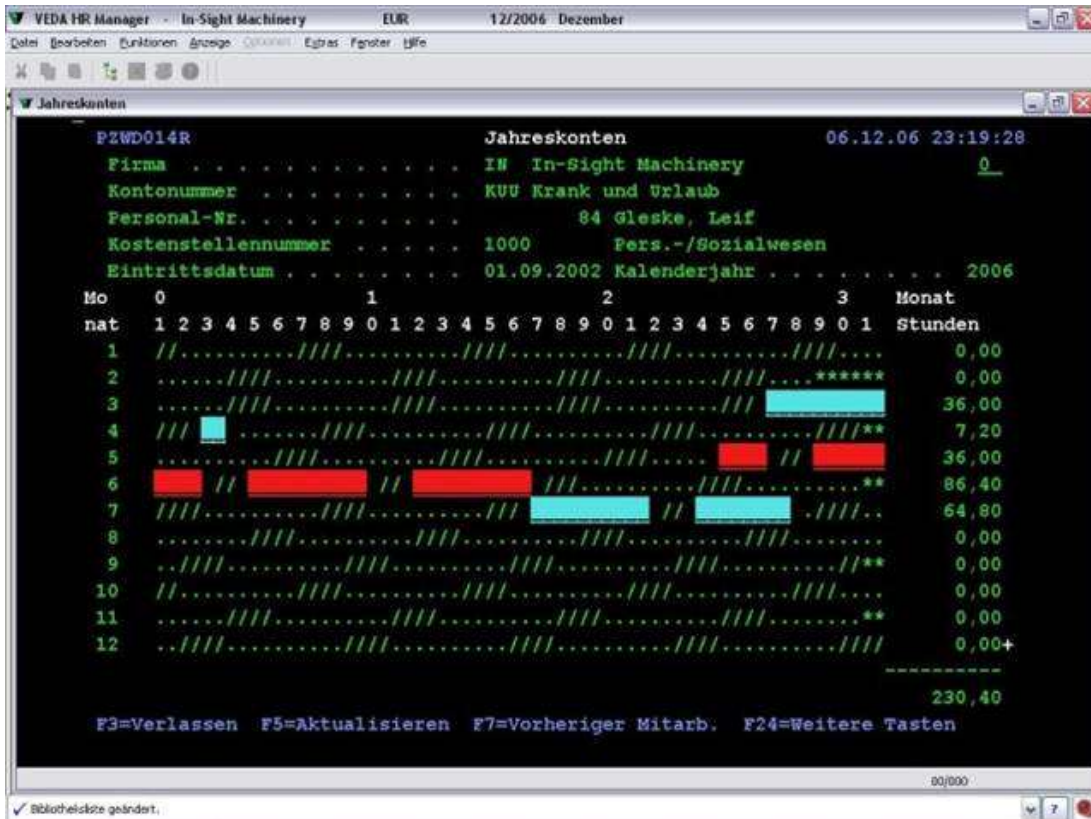
3. Modernisierung – JUMP der sanfte Modernisierungsprozess

Die GUI muss einen deutlichen Mehrwert bringen, sonst ist der schnelle Einstieg mit einer oberflächlichen GUI-fizierung zwar mit JUMP möglich, empfiehlt sich aber nicht. Wenn das GUI nicht bereits beim ersten Kontakt dem Anwender einen deutlichen Mehrwert bringt, hat man eine große Chance zur Akzeptanz des Modernisierungsprozesses vertan. Dies ist das Dilemma vieler Screen Scraping Schnellschüsse, die regelmäßig scheitern.

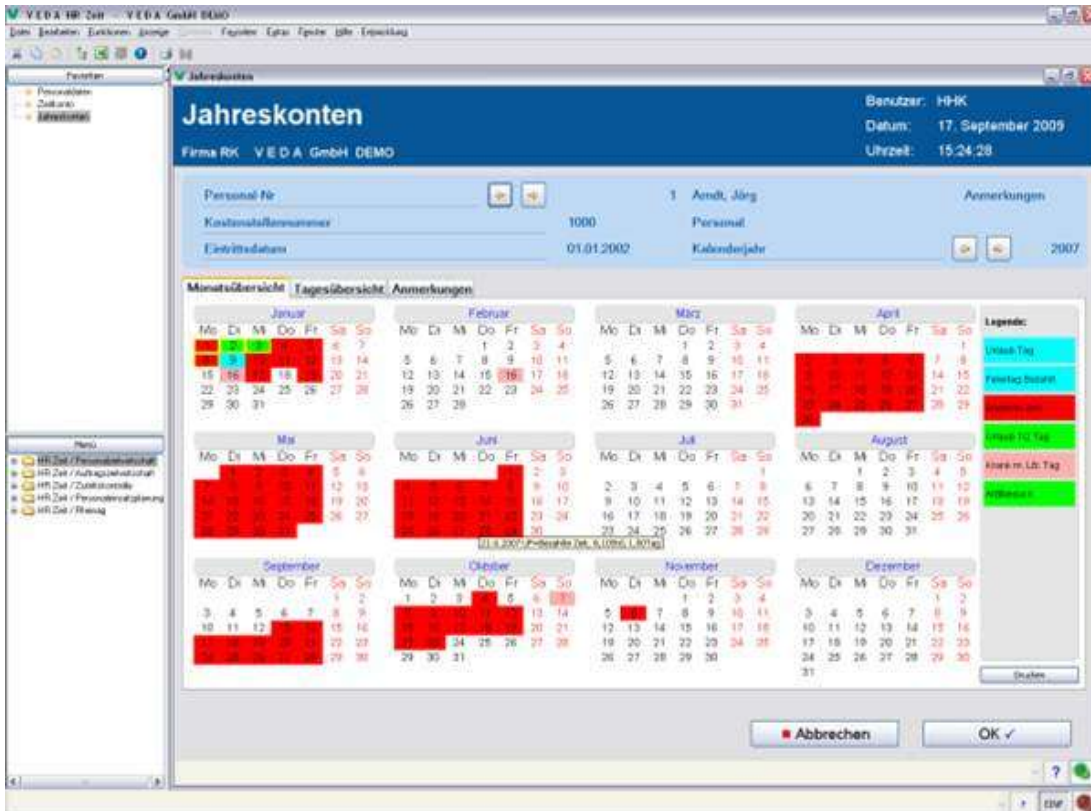
Aktualisierungen gezielt und schnell durchführen.

Mit JUMP kann man einen Modernisierungsprozess starten, indem gezielt und schnell einzelne Module oder Funktionsblöcke auf den aktuellen Stand von benutzerfreundlicher Software bringt. Der Rest der Anwendung kann währenddessen noch einfach migriert bleiben, wozu auch alle Masken des Betriebssystems oder fremder Software gehören können.

Anhand der modernisierten Teile der Anwendung können ausgewählte Benutzer ihre Wünsche einbringen, damit den Prozess weiter vorantreiben und gleichermaßen als Key-User positive Stimmung zur neuen Oberfläche verbreiten.



Vorher



Nachher

Oberflächenmodernisierung ist aber nur ein Bruchteil der eigentlichen Aufgabenstellung. **Integration** ist gefragt. Ob es dabei um die Einbindung von Office oder Groupware geht oder die Anbindung von Mashups (Google & Co.), der JUMP Client bringt bereits alles Notwendige mit. Sie können Teile Ihrer Anwendung als Web-Service zur Verfügung stellen oder konsumieren andere Services im Rahmen der SOA Architektur. Egal ob Sie dabei Ihre RPG-/Cobol-Programme nutzen wollen oder nach Java migrierte Teile der Anwendung.

4. Migration – „Big Bang“ vermeidbar

Die Erweiterbarkeit von gewachsenen Anwendungen ist in der Regel sehr eingeschränkt. Ein Re-Design ist oft so aufwändig, dass sich die Frage nach dem Ersatz durch Standardsoftware oder Neuentwicklung stellt. Beide Alternativen bedeuten den Verlust der gewachsenen, oft maßgeschneiderten Anwendung! Mit JUMP können Sie neue Standardsoftware oder neu entwickelte Anwendungskomponenten allein über die Oberfläche in Ihre Altanwendung integrieren. Auf diese Weise können (moderne) Teile der Anwendung neu geschrieben – oder zugekauft – werden, während die vorhandene, bekannte und beliebte Individualsoftware weiter im Einsatz bleibt. Diese Methodik bleibt dem Benutzer weitgehend verborgen. Somit ist der „**Big-Bang**“ eines Austauschs **vermeidbar**.

JUMP schafft Freiräume für intuitives Arbeiten.



Mit JUMP verlängert man das Leben der Altanwendungen und löst die akuten Beschwerden. Das verschafft Luft, um neue Wege zu gehen. Denn über eines darf eine automatisierte Modernisierung nicht hinwegtäuschen: Auf lange Sicht muss jede herkömmliche Anwendung ersetzt werden.

VEDA hat dazu das Neuentwicklungswerkzeug "JLand" geschaffen, das unsere Ansprüche an Effizienz und Flexibilität erfüllt. Es basiert wie JUMP auf Java Standards und integriert z. B. mit JUMP modernisierte Anwendungen. Somit ebnet dieses Werkzeug bereits nahtlos den nächsten Schritt in die Zukunft.

Mit JUMP und JLand bietet VEDA einen konsequenten Weg, der aktuelle Software mit maximaler Investitionssicherheit und minimalem Ressourceneinsatz verknüpft. Mit VEDA gehen Sie eine Partnerschaft ein, die auf Dauer ausgelegt ist. Letztlich kommt es auf das Vertrauen an – das Vertrauen in eine Lösung aber auch in ein Unternehmen.

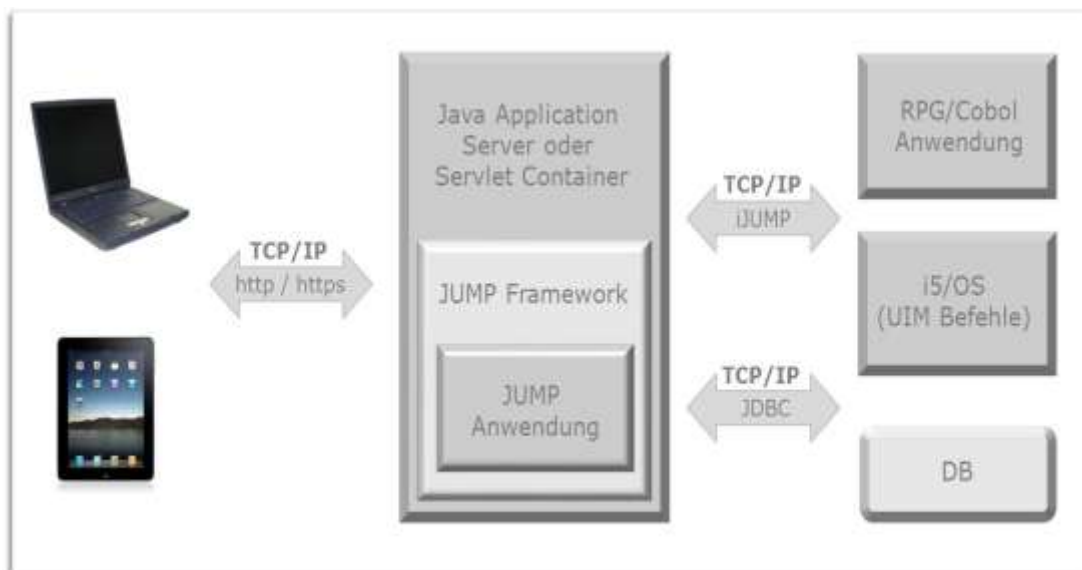
5. Architektur

Die Architektur einer mit JUMP modernisierten Anwendung ist auf die Bedürfnisse heutiger Geschäftsanwendungen ausgerichtet. Die Anwender müssen sicher, komfortabel und überall auf der Welt arbeiten können. JUMP-Anwendungen können somit als SAAS-, Cloud, oder ASP-Anwendungen betrieben werden.

Der Client installiert und aktualisiert sich automatisch (Java Webstart). Alle Geschäftsregeln werden nur auf dem Server ausgeführt. Der Client dient nur zur Darstellung und als Schnittstelle zum Server sowie zu anderer Client-Software wie z. B. Microsoft Office.

Auch auf Serverseite kommt ausschließlich Java zum Einsatz, wodurch JUMP-Anwendungen plattformunabhängig sind und somit auch mit einem IBM i-Server betrieben werden können.

Es werden etablierte Standards wie LDAP, SSL oder WSDL eingesetzt. Als Server-Umgebung können beliebige Java Applikation-Server eingesetzt werden vom Tomcat bis hin zu Websphere.



Weitgehende Datenbankunabhängigkeit wird einerseits durch JDBC, andererseits mit einer Dialektschicht in JUMP erreicht.

In der offenen Architektur ist die Integration von Business Process Management (BPM) oder eines Enterprise Service Bus (ESB) bereits möglich.

„Business follows IT“ war gestern - heute bestimmen die Geschäftsprozesse den Funktionsumfang der Software.

6. Transformer



Durch den regelbasierten Ansatz von JUMP können auch umfangreiche Anwendungen schnell transformiert werden. Je konsistenter der Programmierstil der zu modernisierenden Anwendung ist, desto weniger Einzelbearbeitung ist notwendig.

Die Entwicklungsmethode von JUMP berücksichtigt die Erfahrungen von IBM i-Programmierern und öffnet die IBM i-Welt für Java-Entwickler.

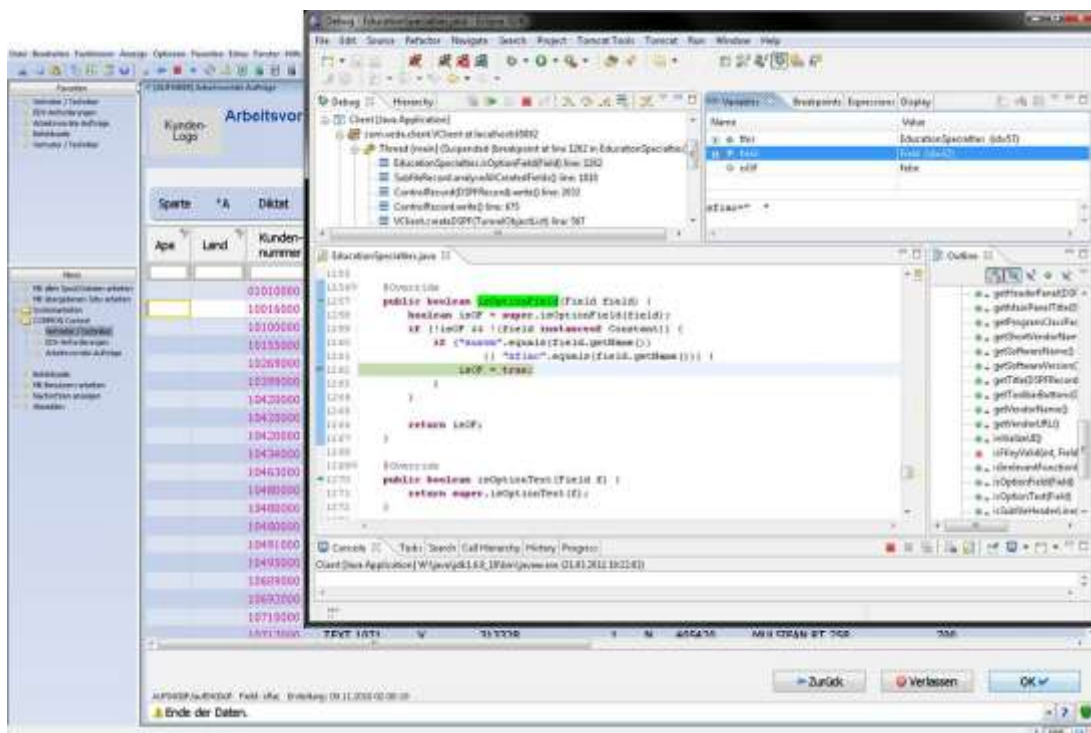
Die Migration und damit auch die Bereitstellung einer grafischen Oberfläche kann vollständig in den bisherigen Entwicklungsprozess eingebunden werden. Dazu sind nur ein paar zusätzliche i5/OS-Befehle und DDS-Schlüsselwörter notwendig. Der komplette Prozess – von der Entwicklung bis zur Installation – läuft auf IBM i und ist somit für IBM i Entwickler selbstverständlich.

Die Regeln hinter dem Prozess werden – für jeden Programmierer leicht verständlich – in Java programmiert und zu Beginn des Projektes in Zusammenarbeit mit VEDA erstellt.

Wir haben sogar die Erfahrung gemacht, dass oft schon die Voreinstellungen ausreichen.

7. Entwicklungsumgebung

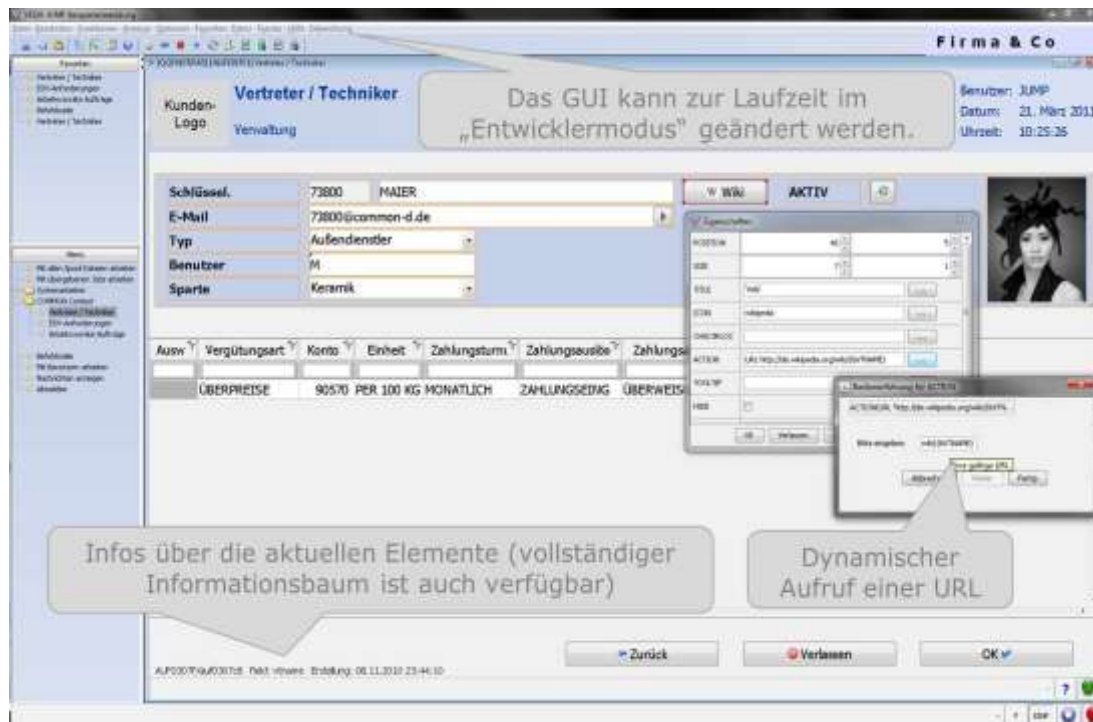
Im Rahmen des Projektes ergeben sich durch den Einsatz von Java neue Möglichkeiten, wie z. B. die Einbindung freier Darstellungskomponenten. Erst wenn diese genutzt werden sollen, um mit der modernisierten Lösung weitere Türen zu öffnen, sind Java-Kenntnisse nötig.



Die Java-Entwicklung erfolgt in der IDE Ihrer Wahl. Da die Architektur von JUMP sich an Standards hält, ist keine spezielle Entwicklungsumgebung notwendig. Alle üblichen Methoden zum Ändern, Debuggen oder Verteilen von Java Web-Anwendungen können verwendet werden. Jeder Java Entwickler findet sich schnell darin zurecht und kann mit Unterstützung der IBM i Entwickler auch schnell Erweiterungen vornehmen. JUMP stellt Schnittstellen und Methoden für die typischen Problemstellungen bereit.

Für den normalen Migrationsprozess, z. B. nach Änderungen der DDS oder RPG Quellen, wird keine grafische Entwicklungsumgebung benötigt, sondern lediglich ein i5/OS-Befehl ausgeführt.

8. Einzelbearbeitung



Die Weiterentwicklung kann also in der vorhandenen Anwendung erfolgen und die Migration läuft im Hintergrund automatisch. Wurden Erweiterungen für das migrierte Ergebnis vorgenommen, bleiben diese natürlich erhalten. Erweiterungen und Änderungen am grafischen Frontend erfolgen z. B. durch neue DDS Schlüsselwörter (DDX), wie es ein IBM i Entwickler gewohnt ist.

Die Änderungen können aber auch über den JUMP Client im Design-Modus bei laufender Anwendung erfolgen. So sieht man sofort die Auswirkung und kann diese Änderungen auch wieder als DDX speichern.

9. Ausführungsumgebung (Runtime)

Eine migrierte Anwendung verhält sich ohne Anpassungen genauso, wie die originale Anwendung. Dazu ist es notwendig, dass auch in der Java Umgebung i5/OS Konzepte wie Bibliothekssuchlisten, temporäre Dateien/Bibliotheken, Überschreibungen, Jobs, Spools, u. v. m. vorhanden sind. Dies alles liefert die Laufzeitumgebung von JUMP bzw. integriert die Lösungen der ausführenden Plattform.

Damit eine IBM i Anwendung auf anderen Plattformen unverändert ausgeführt werden kann bedarf es weit mehr als der Migration der Programme. JUMP portiert auch Datenbanken und i5/OS Objekte wie Nachrichtendateien oder Datenbereiche und liefert eine Laufzeitumgebung, die nahezu alle notwendigen Konzepte der IBM i abbildet. So gibt es natürlich eine QTEMP, Bibliothekssuchlisten, Jobs, OUTQs,

MSGQs, DTAQs und vieles mehr. In JUMP sind sehr viele CL-Befehle und APIs verfügbar, so dass IBM i Anwendung wirklich komplett portiert werden können. Neuschreiben von Programmen ist sehr selten notwendig. Es gibt immer wieder mal Funktionen die JUMP noch nicht abgebildet hat. Aber dank des objektorientierten Java Frameworks lassen sich diese Funktionen schnell nachrüsten. Selbst komplexe Konzepte der IBM i wie CGI, FTP, RPG-Zyklus oder OPNQRYP sind bereits im Standard enthalten.



JUMP ermöglicht den Zugriff auf nahezu beliebige Datenbanken aus einem Programm heraus und das so einfach wie im RPG. Definieren Sie ganz normal Dateien und Lesen/Schreiben mit bekannten Methoden wie READ oder WRITE. Das Framework weiß wo sich die Tabellen befinden und leitet den Zugriff auf das jeweilige Datenbanksystem weiter. Daten können ebenso einfach von einem Datenbanksystem auf ein anderes übertragen werden. Sichern Sie z. B. einfach eine Bibliothek unter DB2/UDB und speichern Sie unter MS SQL zurück.

10. Wartung

Da eine durch JUMP modernisierte Anwendung noch die Strukturen der originalen Anwendung aufweist, finden sich die bisherigen Entwickler schnell darin zurecht. Tests können effizient z. B. in Form von automatisierten Regressionstests durchgeführt werden.

Analysewege sind wie gewohnt vorhanden. Beispielsweise gibt es weiterhin einen Joblog mit der Anzeige des Programmaufrufstapels oder der Liste der offenen Dateien. Die Wartung von JUMP Anwendungen ist daher so komfortabel, wie bei einer klassischen IBM i Anwendung. Die zusätzliche Komplexität durch die Web-Architektur ist weitgehend versteckt und muss nur in Ausnahmefällen untersucht werden.

11. Projektablauf

Die Modernisierung mit JUMP erfolgt in individueller Geschwindigkeit und Umfang.

Möglicher Ablauf bei Integration

Als erstes sollten die funktionalen Anforderungen an das grafische Frontend sowie dessen Design definiert werden. Die Oberfläche eines JUMP-Clients lässt sich auf jedes Corporate Design anpassen. Mit der Migration des Frontends erfolgt der erste Modernisierungssprung. Die Anwendung kann nun mit grafischer Oberfläche und vielen neuen Funktionen eingesetzt werden. Dabei werden die unveränderten IBM i-Dialogprogramme weiterverwendet.

Als nächstes sollte an der Schaffung von Zusatznutzen gearbeitet werden. Beispielsweise kann nun eine Einbindung von Microsoft Office erfolgen oder die Integration von Geoinformationen (Landkarten). Bei einzelnen Bildschirmen kann es auch sinnvoll sein, individuelle Komponenten z. B. zur grafischen Ressourcenplanung zu erstellen.

Bereits in diesem Stadium verschmilzt die Anwendung einem Java Framework, das die Stärken dieser offenen Welt nutzbar macht. Die Verarbeitung von XML oder das Konsumieren von Webservices stellen nun kein Problem mehr dar.

Auch die gesamte Anwendung kann nach Java portiert werden. Dadurch ist die Plattform IBM i nicht mehr zwingend, aber natürlich auch noch möglich. Alleine diese Option sichert Ihre Investition in eigene Software!

12. Benefits

Eine JUMP Modernisierung eröffnet aber völlig neue Möglichkeiten. Bei der Umsetzung der täglichen Anforderungen können effizienteste Wege eingeschlagen werden. Durch die Fortführung der Eigenentwicklung bleibt die Flexibilität erhalten und wird durch die offene Architektur gesteigert. Fertige Business-Funktionen erweitern Ihre Lösung und bringen dem Anwender erheblichen Zusatznutzen. Neue Anwender finden sich schneller zurecht und Ihre Entwickler können zukunftsicher arbeiten.

JUMP schöpft Ressourcen aus. Ob es die Plattformunabhängigkeit ist oder einfach die enorm verbreitete Programmierumgebung: Die Zeiten der Insellösungen sind jedenfalls vorbei.

Damit wird die Grundidee der Investitionssicherung beim System IBM i durch JUMP konsequent weitergeführt.

Wichtig ist, dass der Anwender seinen Job mit Ihrer Anwendung schnell und zuverlässig erledigen kann. Dazu bietet JUMP alles Notwendige und mehr:

- ▶ Intuitive Bedienung mit Kontextmenüs und Drag & Drop (sehr wichtig für Einsteiger)
- ▶ Übersichtlichkeit durch einheitliche Gestaltung und Hervorhebungen
- ▶ Menübaum, Favoritenmenü, Checklisten und Desktop helfen beim Organisieren wiederkehrender Funktionsaufrufe
- ▶ Datumskomponenten, Comboboxen, Checkboxes vereinfachen die Eingaben und vermeiden Tippfehler
- ▶ Integration von beliebigen Mashups (online Dienste) wie Google Maps, Wetter, Flug-, Personensuche, ...
- ▶ Integration von Mail, Office, GIS (graphische Informationssysteme)
- ▶ Integration von Bildern (z. B. Artikelbilder, Passfotos)
- ▶ Integration in Workflows oder einer Business Process Engine
- ▶ Darstellung von Hierarchien in Form von Baumstrukturen
- ▶ Tabellendarstellung mit
 - ▶ freier Sortierfunktion
 - ▶ umfangreicher Filterfunktion
 - ▶ freier Anordnung der Spalten
 - ▶ schneller Diagrammanzeige
 - ▶ einfachem Excel-Export
 - ▶ Summenzeile
- ▶ Verarbeitung von Subfile-Zeilen über Kontextmenü (wahlweise auch weiter über Eingabe von Auswahlen)
- ▶ Integration von Uploads (Import externer Dateien direkt über die Oberfläche)
- ▶ Personalisierungsmöglichkeiten (Farben, Verhalten, Darstellung)
- ▶ Vorteile aus der GreenScreen-Welt bleiben erhalten (Tastaturpuffer, Navigation)
- ▶ Dynamische Bildschirmauflösung

Beispielansicht Arbeitsvorräte Aufträge



Unternehmen, die sich für eine Neuentwicklung entschieden haben, integrieren die vorhandene Lösung über JUMP mit der neuen Software. Denn kaum jemand kann eine Neuentwicklung komplett parallel zur Wartung des Altsystems leisten.

13. USP - Alleinstellungsmerkmale

Erfahrung

VEDA hat als Software- und Lösungsanbieter JUMP auf Basis hoher Anforderungen und fast 35-jährigen **Erfahrung** entwickelt.

Durch den Einsatz bei Partnern und Kunden ist JUMP weiter gewachsen. Inzwischen deckt das Modernisierungswerkzeug die Bedürfnisse und Anforderungen weitreichender ab, als die meisten Produkte von reinen Tool-Herstellern. Das liegt daran, dass VEDA die eigenen „lessons learned“ und die Ideen unserer Kunden und Partner immer wieder in unsere Lösungen einfließen lässt. **Wir verstehen Sie!**

Verbreitung

Die JUMP Technologie ist sehr weit **verbreitet** und das Business Framework löst schon heute Probleme, auf die Sie zwangsläufig in der vernetzten, komplexen Welt stoßen werden.

Eine durchdachte und starke Anwendung

- ▶ Die grafische Benutzeroberfläche von JUMP Anwendungen ist außerordentlich leistungsstark. Kein anderes Werkzeug verfügt out of the box über so viele durchdachte Komponenten. Für das GUI müssen die Programme, anders als bei RPG OA oder ähnlichen alten Techniken, nicht modifiziert werden. Trotzdem hat JUMP Zugriff auf hidden Felder und den ganzen Subfile. Es erfolgt also kein Screen Scraping.
- ▶ Die verwendete Schnittstelle zu dem interaktiven Job (ohne OLTP Kosten), ermöglicht die Darstellung aller Masken. Ob aus CL, RPG, Cobol oder aus dem Betriebssystem, alle Bildschirme werden im grafischen Client innerhalb des gleichen Jobs angezeigt. Bildschirme zu denen ein DDS vorliegt, können dabei so umgesetzt werden, dass sie nicht von guten Windows-Benutzeroberflächen zu unterscheiden sind.
- ▶ Die Migration kann beliebig oft wiederholt werden, wobei alle Modifikationen erhalten bleiben. Diese **Re-Migrationsfähigkeit** ist bei anderen Werkzeugen oft nicht oder schlecht gelöst.
- ▶ Das Ergebnis der Migration kann stark beeinflusst werden. Dennoch erfolgt dies leicht nachvollziehbar. **Erweiterungen** sind an allen wichtigen Stellen möglich. Es gibt keine Grenzen wenn man bereit ist, eigenen Java Code zu schreiben.
- ▶ Die verwendete Architektur und die gewählte Technik, wie z. B. der HTTP-Tunnel, ist **State of the Art**. JUMP hält sich so weit wie möglich an Standards. Es ist keine proprietäre Lösung. Beispielsweise können JUMP Anwendungen mit beliebigen Entwicklungswerkzeugen bearbeitet und analysiert werden. Somit gibt es auch sofort ausreichend Entwickler, die sich damit auskennen.
- ▶ Für die Entwickler der bestehenden Anwendung ist der Einstieg in die neuen Technologien so einfach wie möglich. Da durch JUMP zeitgemäße Programmierstandards zum Einsatz kommen, sehen auch die klassischen IBM i Programmierer eine bessere Zukunft. Dies **motiviert** so manchen und erleichtert es neue Mitarbeiter zu finden und zu binden.
- ▶ Auch ohne Java programmieren zu müssen, lässt sich eine JUMP Oberfläche schön gestalten. Dabei helfen die Weiterführung des DDS-Konzeptes und die Möglichkeit, zur Laufzeit der Anwendung, in einen **Entwurfsmodus** zu wechseln, der WYSIWYG Änderungen erlaubt.
- ▶ Der Client installiert und aktualisiert sich **automatisch** (zero administration) und der Server ist einfach zu installieren. Änderungen in Ihrer Anwendung können auch bei laufendem Server ausgeliefert werden.
- ▶ Der Umstieg der **Anwender** von der klassischen Oberfläche auf die JUMP GUI wird durch viele Funktionen erleichtert. In Erweiterung zu einer typischen grafischen Oberfläche bietet JUMP Bedienmöglichkeiten, die nur aus der zeichenbasierten Oberfläche bekannt sind wie Tastaturpuffer und vollständige Bedienung mit der Tastatur inklusive Navigation. Selbst feldübergreifendes Copy/Paste ist möglich. Trotzdem gibt es alle Windows-üblichen Funktionen, die die Anwendung aufwerten, wie Filter-, Sortier-, Export-, Diagramm-Möglichkeiten in Subfiles u. v. m. Der JUMP Client ist selbst für die beim Green Screen geschätzte „Schnellerfassung“ geeignet, weil entsprechende Tastenbelegungen vorgesehen sind.

- ▶ JUMP eröffnet eine neue Welt für die **Entwickler** ohne Sie zu stark mit neuen Themen zu belasten. Selbst beim GUI kann der Programmierer weiter DDS gestalten und muss sich nicht mit JSP, JSF, HTML oder Flash rumschlagen. Dies gehört zu den Prinzipien von JUMP: „keep it simple“. Der Entwickler soll sich weiterhin um die fachlichen Anforderungen kümmern und nicht mit zunehmend komplexen Systemen befassen.
- ▶ JUMP sichert die **Zukunft**. Ob es der komplette Plattformwechsel oder die Integration in eine Neuentwicklung ist, alle Investitionen in Ihre Eigenentwicklung behalten ihren Wert.

14. Neuentwicklung - Weiter- und Neuentwicklung

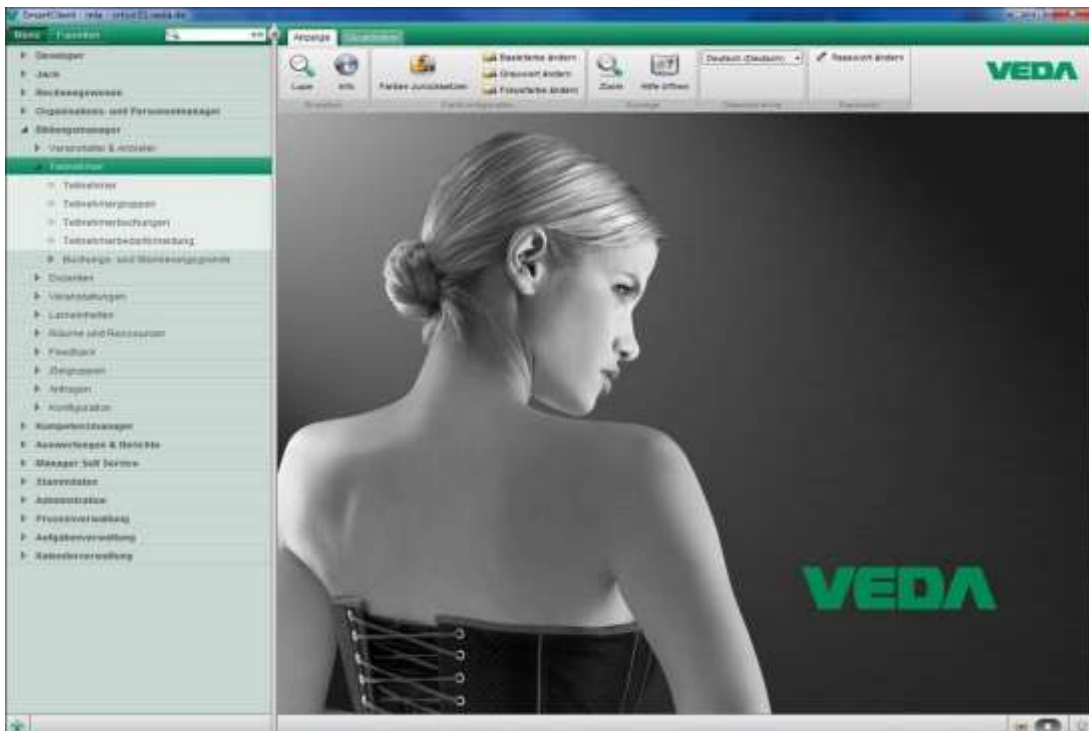
Individuallösungen haben neben Standardsoftware ihre Berechtigung. Wer gute Erfahrungen mit der dadurch entstandenen Flexibilität und Unabhängigkeit gemacht hat, wird auch in Zukunft eine eigene Lösung in Betracht ziehen. Software muss permanent aktualisiert werden. Bleibt die fortlaufende Umsetzung aktueller Bedürfnisse eine Zeit lang aus, wird früher oder später eine aufwändige Modernisierung notwendig.

Eine Modernisierung kann gute Ergebnisse liefern, dennoch ist sie nicht in jedem Fall der beste Weg. Oft werden komplett andere Strukturen gefordert oder es liegen ganz neue Anforderungen der Anwender vor. Wie sollten solche Entwicklungen erfolgen?

Moderne Software hat eine andere Architektur als historische Anwendungen. Außerdem muss sie in immer kürzeren Zyklen angepasst und verändert werden können: „Software follows business“.

Mit JLand hat VEDA ein anwendungsunabhängiges Business-Framework entwickelt, das Ihrem „Business“ perfekt folgt. Das Framework stellt Funktionen für die Abbildung von Geschäftsprozessen bereit. JLand basiert dabei aber nicht auf generiertem Programmcode, sondern interpretiert Konfigurationen und stellt auf diese Weise die Anwendung dynamisch dar.

JLand kann als dynamisches CASE-Tool bezeichnet werden. Die Anwendung wird in einem Repository definiert. Nur einzelne Bausteine, die dann dynamisch eingebunden werden, verlangen eine Java-Programmierung. Zukunftssicherer und flexibler können Sie Ihre Anwendung nicht erstellen.



JLand bietet:

- ▶ Plattformunabhängigkeit
- ▶ Datenbankunabhängigkeit (MS SQL, Oracle, ...)
- ▶ Skalierbarkeit
- ▶ leichte Wartbarkeit
- ▶ einfache und releasefähige Anpassbarkeit (Varianten)
- ▶ austauschbare Präsentationsschicht (Rich Client, HTML, Mobile)
- ▶ schnelle Entwicklung
- ▶ automatisierte Tests
- ▶ einheitliche, schöne, intuitive und benutzerfreundliche Oberfläche

JLand differenziert sich von anderen Frameworks / Entwicklungstools durch:

- ▶ Einfachheit
- ▶ Nutzen von Standardtechnik
- ▶ Verwendung einer gemeinsamen logischen Datenbank: "Joker-Repository"
- ▶ Bestehende und neue Anwendungen können integriert unter einer Oberfläche laufen
- ▶ Integration von business process management
- ▶ umfangreiche business features



JLand schafft die Brücke von gewachsenen Systemen zu moderner Software:

Mit VEDA JUMP modernisierte Anwendungen kann JLand integrieren, so dass bei Bedarf der Übergang von „alt“ zu „neu“ schrittweise erfolgen kann.



Deutschland:

VEDA GmbH
Hauptsitz
Carl-Zeiss-Str. 14
52477 Alsdorf
Telefon: +49 2404 5507-0
Telefax: +49 2404 5507-2133
E-Mail: info@veda.net
Internet: www.veda.net

VEDA HR Informationssysteme AG
Carl-Zeiss-Str. 14
52477 Alsdorf
Telefon: +49 2404 5507-0
Telefax: +49 2404 5507-2133
E-Mail: info@net.de
Internet: www.veda.net

VEDA GmbH
Niederlassung Offenburg
Okenstr. 23a
77652 Offenburg
Telefon: +49 2404 5507-1700
Telefax: +49 2404 5507-2770
E-Mail: info@veda.net
Internet: www.veda.net

VEDA Zeit GmbH
Okenstr. 23a
77652 Offenburg
Telefon: +49 2404 5507-1700
Telefax: +49 2404 5507-2770
E-Mail: info@veda.net
Internet: www.veda.net

Schweiz:

VEDA (Switzerland) AG
Im Langhag 5
8307 Effretikon
Telefon: +41 52 347-0212
Telefax: +41 52 355-3233
E-Mail: info@veda.net
Internet: www.veda.net