



Nr.: FIN-10-2008

## **Bewertung von VLBA- Architekturvarianten:**

### **Eine Fallstudie zur Einführung von OpenCms und der SAP® e-Recruiting Plattform**

Sebastian Herden, André Zwanziger

*Arbeitsgruppe Wirtschaftsinformatik*



Fakultät für Informatik  
Otto-von-Guericke-Universität Magdeburg

# Technical Report

**Impressum** (§ 10 MDStV):

*Herausgeber:*  
Otto-von-Guericke-Universität Magdeburg  
Fakultät für Informatik  
Der Dekan

*Verantwortlich für diese Ausgabe:*  
Otto-von-Guericke-Universität Magdeburg  
Fakultät für Informatik  
Sebastian Herden  
Postfach 4120  
39016 Magdeburg  
E-Mail: [sebastian.herden@ovgu.de](mailto:sebastian.herden@ovgu.de)

<http://www.cs.uni-magdeburg.de/Preprints.html>

*Auflage:* 51

*Redaktionsschluss:* Oktober 2008

*Herstellung:* Dezernat Allgemeine Angelegenheiten,  
Sachgebiet Reproduktion

*Bezug:* Universitätsbibliothek/Hochschulschriften- und  
Tauschstelle

# **Bewertung von VLBA-Architekturvarianten: Eine Fallstudie zur Einführung von OpenCms und der SAP® e-Recruiting Plattform**

Sebastian Herden, André Zwanziger

VLBA Lab Magdeburg  
Fakultät für Informatik  
Institut für betriebliche und technische Informationssysteme  
Otto-von-Guericke Universität Magdeburg  
Postfach 4120  
39016 Magdeburg  
sebastian.herden@iti.cs.uni-magdeburg.de  
andre.zwanziger@iti.cs.uni-magdeburg.de

**Abstract:** Bei der Planung neuer IT-Systeme für Very Large Business Applications (VLBA) müssen in frühen Phasen fundierte Architekturentscheidungen getroffen werden, welche einen weit reichenden Einfluss auf das weitere Vorgehen, Zeit und Kosten im Entwicklungsprojekt haben. In diesem Artikel wird am Beispiel der Bayer AG beschrieben, wie eine Architekturentscheidung mit Hilfe der Function-Point-Analyse zur weltweiten Einführung der SAP® e-Recruiting Plattform und dem Content Management System OpenCms vorbereitet wurde. Dafür werden die möglichen Architekturvarianten mit Vor- und Nachteilen vorgestellt und bewertet.

**Abstract:** Planning new IT-Systems for Very Large Business Applications (VLBA) requires informed decisions about possible architectures in early phases, which may have large-scale influences on further approaches, time and cost considerations during a development project. This article describes how an architectural decision has been prepared based on a Function-Point-Analysis approach to rollout the SAP® e-Recruiting platform and the content management system OpenCms at Bayer AG worldwide. Therefore the possible kinds of architecture will be introduced and discussed.

## **1 Einführung**

Im Jahr 2006 wurde bei der Bayer AG das Projekt „Transforming Human Resources“ (THR) aufgesetzt, welches zum Ziel hatte, die Prozesse in der Personalverwaltung an den weltweiten Standorten der Bayer AG zu standardisieren und zu verbessern [Fey06]. Dazu sollten sowohl die organisatorischen Abläufe restrukturiert, als auch eine global integrierte IT-Landschaft im HR-Bereich aufgebaut werden. Die organisatorische Restrukturierung des Personalwesens hatte zum Ziel die bisher einzelnen Personalabteilungen der jeweiligen Teilkonzerne und Standorte in Shared-Service-Center-Strukturen zu überführen, um somit eine gemeinsame zentrale Personalabteilung zu etablieren. Auf technischer Seite sollte die bisher heterogene IT-Landschaft im HR-Bereich harmonisiert werden, sodass nur noch ein globales IT-System für die Personalverwaltung vorhanden ist [Fey06]. Der neu strukturierte Personalbereich der Bayer AG erfüllt somit sämtliche Eigenschaften einer VLBA [Grab07].

Ein Bestandteil dieser VLBA für die Personalverwaltung ist eine so genannte e-Recruiting-Plattform. Sie ermöglicht die webbasierte Suche nach offenen Stellen in einer Organisation und die Abwicklung von Online-Bewerbungen auf Vakanzen als Self-Services. Nachdem sich eine Person online beworben hat und eingestellt wurde, können diese persönlichen Daten dann automatisch in das Bayer HR Employee Portal übernommen werden, welches ebenfalls auf dem globalen HR-System basiert [Bay06d]. Durch die Eingabe der persönlichen Daten durch den Bewerber selbst, werden die klassischen Forderungen der integrierten Informationsverarbeitung (frühzeitige Dateneingabe, redundanzfreie Datenhaltung) vollständig umgesetzt.

Ein Teilprojekt im Rahmen des THR-Projektes zielte dabei auf die Ablösung einer bestehenden web-basierten e-Recruiting-Lösung in der Personalbeschaffung ab, in der Content-Management-System-Funktionen (Anlegen und Bearbeiten von Webseiten) und e-Recruiting-Funktionen (Stellensuche, Online-Bewerbung) abgebildet waren. Diese Lösung sollte durch das Content Management System (CMS) OpenCms und durch SAP® e-Recruiting ersetzt werden. Die Substitution der bestehenden Anwendung durch zwei in ihrem Gebiet spezialisierte Produkte, zog folglich die Frage der Integration von OpenCms und SAP® e-Recruiting nach sich [Bay06a; Fey06].

## **2 Varianten der Basisarchitektur als Entscheidungsgrundlage**

Im Rahmen des THR-Projektes wurden bei der Bayer AG zwei grundlegende Varianten einer möglichen Basisarchitektur für die Integration des CMS OpenCms und der SAP® e-Recruiting Plattform erarbeitet. Neben diesen beiden grundsätzlich unterschiedlichen Architekturvarianten waren auch so genannte hybride Architekturen möglich, die eine Mischform der beiden anderen darstellten [Bay06a; Bay06b]. Sämtliche Architekturvarianten mussten die geforderten Komponenten enthalten, die in Abbildung 1 als UML-Komponentendiagramm grob dargestellt sind.

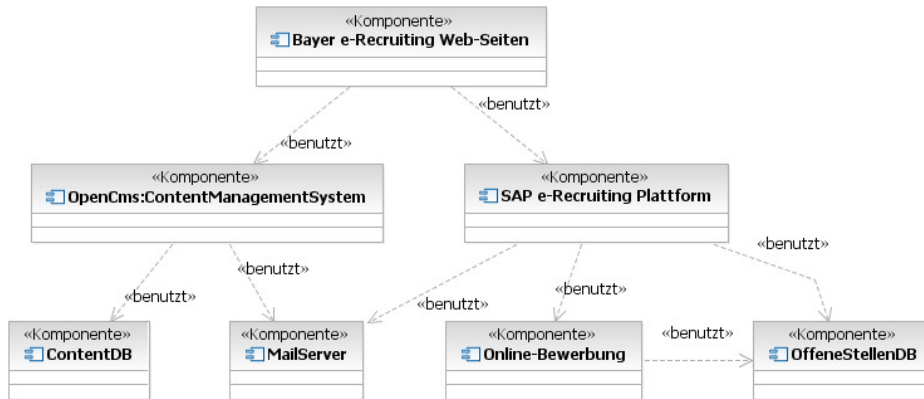


Abbildung 1: Modell der Basiskomponenten nach [Bay06a; Bay06b]

Die e-Recruiting Web-Seiten<sup>1</sup> bieten mit Hilfe des Content Management Systems Informationen über das Arbeiten bei der Bayer AG an. Neben diesen Inhalten werden Dienste zur Online-Bewerbung und zur Suche von offenen Stellen in der e-Recruiting Plattform zusammengefasst.

Ziel war es die Bayer e-Recruiting Web-Seiten so zu gestalten, dass diese beide Kernkomponenten nahtlos miteinander verbunden sind. Hierbei musste ein Trade-Off gefunden werden, zwischen zu erwartendem Entwicklungsaufwand und größtmöglicher Flexibilität bei der Gestaltung der Benutzerschnittstelle. Dieser Zusammenhang der beiden grundlegenden Varianten der Basisarchitektur ist in Abbildung 2 dargestellt.

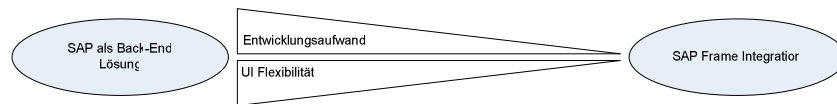


Abbildung 2: Zusammenhang der grundlegenden Varianten der Basisarchitektur [Bay06a]

Die erste Variante (SAP® Frame Integration - Abbildung 3) stellt eine direkte Einbindung der web-basierten Benutzerschnittstelle des SAP® e-Recruitings mit Hilfe von HTML-Rahmen (insbesondere iframes- bzw. object-tags [W3C]) dar.

In der zweiten Variante (SAP® als Back-End Lösung - Abbildung 4) wird die SAP® e-Recruiting Plattform komplett gekapselt und der Benutzer interagiert nicht direkt mit dem SAP® System.

Diese Architekturvarianten dienen als Entscheidungsgrundlage und werden im Folgenden mit Hilfe von UML-Diagrammen kurz vorgestellt.

<sup>1</sup> <http://www.mybayerjob.de>

## 2.1 Integration durch HTML-Frames (SAP® Frame Integration)

Die SAP® e-Recruiting Plattform besitzt eine anpassbare HTML-Oberfläche, über die verschiedene Dienste für Arbeitssuchende und Bewerber angeboten werden können. Das bedeutet, diese Plattform kann auf Grund ihrer eigenen unabhängigen Oberfläche direkt durch Einbindung mit Hilfe von HTML-Frames verwendet werden. In Abbildung 3 ist der Zusammenhang der Komponenten dargestellt.

Der Benutzer fragt mit Hilfe eines Web-Browsers beim Content Management System die entsprechenden Web-Seiten mit dem Bayer HR-Inhalten an. Diese besitzen dann eine Referenz auf die entsprechenden SAP® Inhalte, die dann vom Web-Browser direkt bei der SAP® e-Recruiting Plattform abgerufen werden. Der Browser besitzt die Aufgabe diese beiden Inhalte zu den Bayer e-Recruiting Web-Seiten zusammenzusetzen. Benutzerinteraktionen im SAP-Frame werden direkt über die SAP® e-Recruiting Plattform abgewickelt und gesteuert.

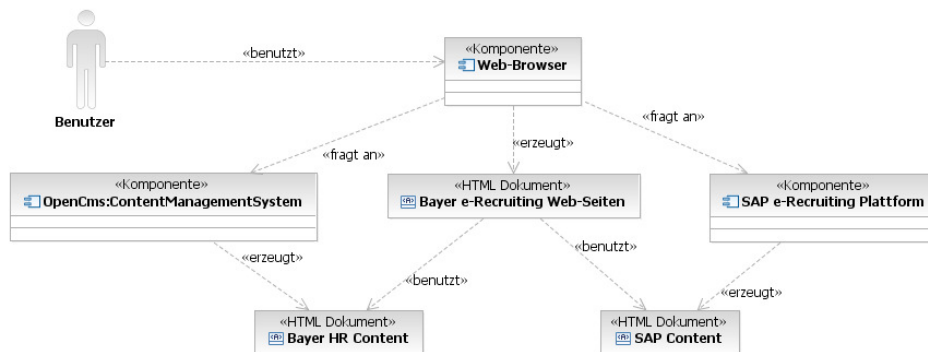


Abbildung 3: SAP® Frame Integration nach [Bay06a]

### Konsequenzen dieses Ansatzes

Bei diesem Ansatz würden die Dienste für die Abwicklung des Bewerbungsprozesses und der Suche von offenen Stellen (insgesamt 35 Use Cases [Bay06c]) alleinig durch die SAP® e-Recruiting Plattform gesteuert. Zusätzliche Funktionalität, die nicht durch diese direkt angeboten wird, muss separat entwickelt werden.

Zudem ergibt sich Entwicklungsbedarf bezüglich des Designs der SAP® Oberfläche, die an die konzerneigenen Richtlinien für das Corporate Design der Bayer AG angepasst werden muss. Zudem erhöht sich der zukünftige Pflegeaufwand, falls es Änderungen gibt. Hierbei muss dann die gesamte Anpassung der Bayer HR-Webseiten in zwei Systemen nachvollzogen werden.

Zusätzlich bedingt die Abhängigkeit des CMS von der e-Recruiting Plattform eine Fehlerbehandlung über beide Systeme hinweg. Dies ist mit der hier vorgeschlagenen Variante nicht ohne zusätzlichen Entwicklungsaufwand möglich, da bei einem Ausfall der e-Recruiting Plattform das CMS keine Kontrolle über diese besitzt.

## 2.2 Integration durch vollständige Kapselung (SAP® als Back-End-Lösung)

Bei dieser Variante wird dem Benutzer der Bayer HR-Webseiten nur ein System für die Kommunikation angeboten. Die Funktionalität der SAP® e-Recruiting Plattform wird hierbei durch OpenCms gekapselt, welches somit alleinig für die Erstellung der Bayer HR-Webseiten verantwortlich ist. Diese Architekturvariante ist in Abbildung 4 dargestellt.

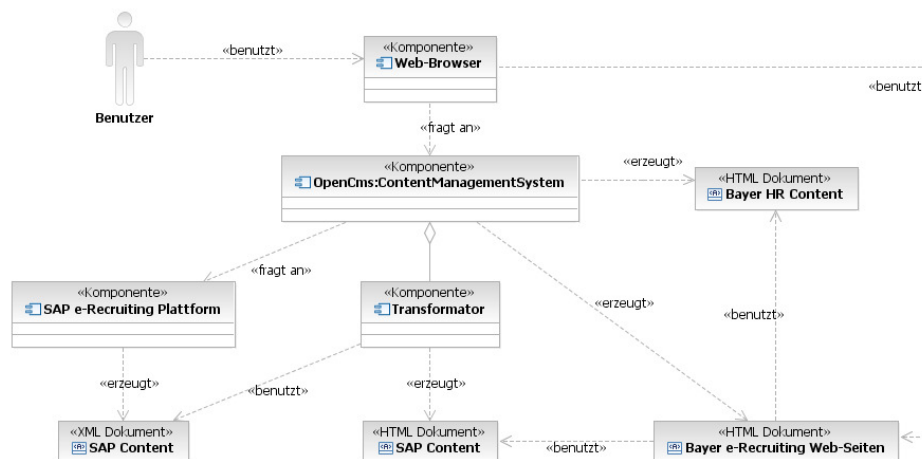


Abbildung 4: SAP® als Back-End-Lösung nach [Bay06a]

Sämtliche Anfragen an die Bayer HR-Webseiten werden von OpenCms entgegengenommen und beantwortet. Sollten SAP® e-Recruiting Funktionen angefragt werden, wird ein RFC (Remote Function Call) beim SAP®-System angestoßen, welcher die entsprechenden Inhalte des Bewerbungsprozesses in Form eines XML-Dokuments zurückliefert. OpenCms verarbeitet diese Daten mit Hilfe eines Transformators und erzeugt zusammen mit den Bayer HR-Inhalten die angefragte Bayer e-Recruiting Web-Seite.

### Konsequenzen dieses Ansatzes

Durch die zentrale Zuständigkeit des CMS für das Layout und das Design existieren keine Einschränkungen bei der Gestaltung der Bayer HR-Webseiten durch das SAP®-System. Änderungen am Corporate Design können somit an zentraler Stelle verwaltet und angepasst werden.

In der Entwicklung wird die Hauptaufgabe dann bei der Gestaltung der Schnittstelle zwischen OpenCms (Anfrageverarbeitung und Transformator) und SAP® (Zurücklieferung von XML-Daten) für die Remote Function Calls gesehen.

Zusätzlich werden Vorteile bezüglich der Wiederverwendbarkeit der Schnittstelle durch das Design einer serviceorientierten Architektur (SOA) erwartet [Her06]. So sind weitere Dienste, die in Form von Web-Services für die Bayer HR-Webseiten aus den Basisdiensten der SAP® e-Recruiting Plattform komponiert werden, denkbar.

## 2.3 Hybrider Ansatz

Zwischen den vorgestellten grundsätzlichen Architekturvarianten existiert eine Menge von hybriden Lösungen. Dies bedeutet, dass einige Funktionen direkt über die SAP® e-Recruiting Plattform angeboten und andere wiederum gekapselt werden. In Abbildung 5 wird der Zusammenhang der Komponenten für eine hybride Lösung grafisch dargestellt.

Zur Vermeidung von Irritationen bezüglich der Zuordnung von Funktionen innerhalb der hybriden Lösung war es notwendig, eine klare Grenze zu schaffen. Diese wurde zwischen dem öffentlichen Bereich und einem personalisierten Bereich gezogen.

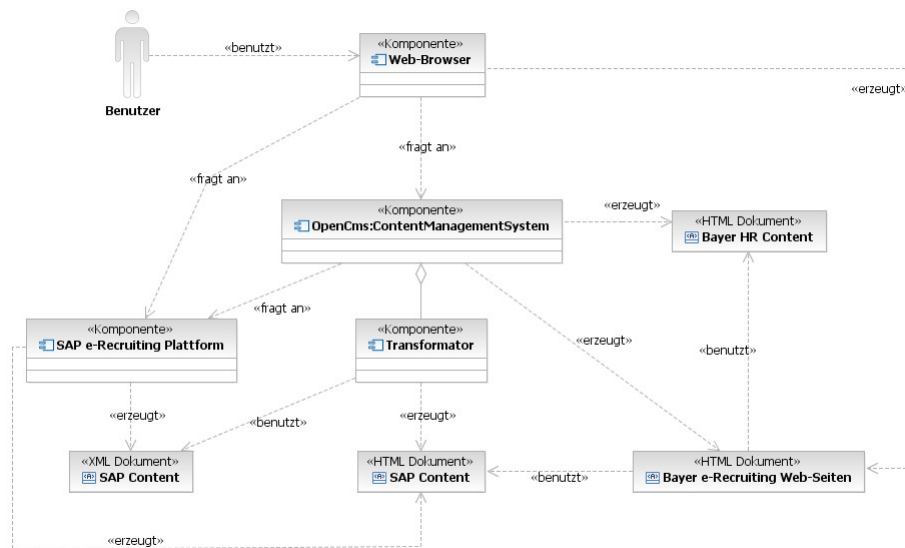


Abbildung 5: Hybrider Ansatz nach [Bay06a]

Der öffentliche Bereich umfasst sämtliche Dienste und Informationen, die uneingeschränkt von jedem Benutzer einsehbar sind (z. B. Job-Search und Job-Navigator). Zum geschützten Bereich hingegen haben nur registrierte Benutzer Zugriff, um Dienste für den Bewerbungsprozess nutzen zu können. Somit sind eine klare Abgrenzung der Funktionen (Use Cases) und deren Verteilung in der Basisarchitektur möglich, welche die Vorteile beider grundlegenden Varianten vereint.



### 3 Entscheidungsunterstützung durch die Function-Point-Analyse

Im Bereich des Software-Engineerings ist es für das Projektmanagement bedeutsam frühzeitig Risikofaktoren bzw. Bereiche mit hohem Risiko zu bestimmen, um geeignete Maßnahmen zur Sicherung abzuleiten [Du01]. Gerade in der Entwurfsphase des Entwicklungszyklus von Software- und IT-Systemen müssen mögliche Architekturvarianten erstellt und bewertet werden können [He04].

Eine Bewertungsmethode aus dem Bereich der Aufwandschätzungen im Software-Engineering stellt die Function-Point-Analyse (FPA) dar. Hierbei wird der funktionale Gehalt einer Software aus Endbenutzersicht für die Schätzung als Ausgangspunkt verwendet. Daher ist es möglich, dass die Größe einer Software an Hand der Aufgabenstellung und nicht am fertigen Endprodukt ermittelt wird [Al79; Dr89; Du01]. Abbildung 6 zeigt den Ablauf der FPA als UML-Aktivitätsdiagramm.

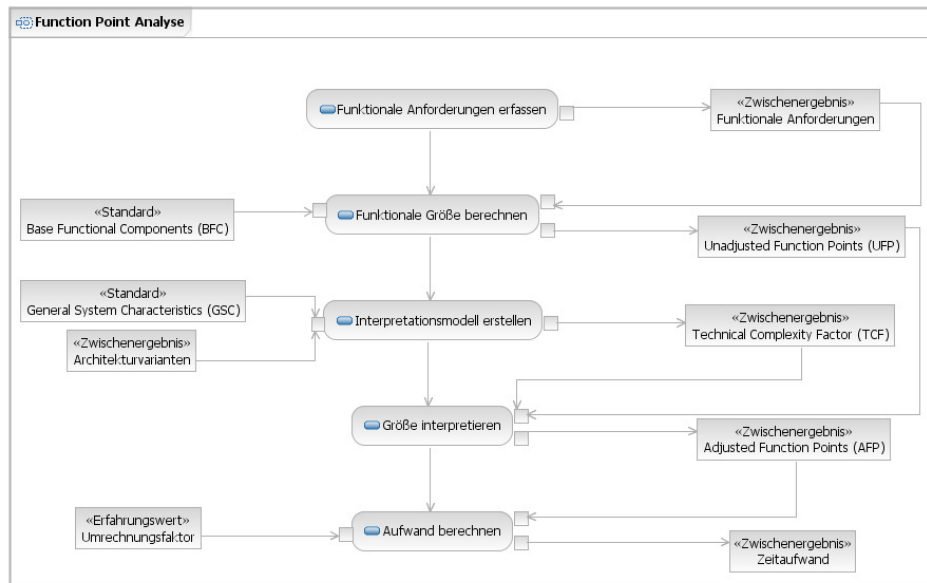


Abbildung 6: Vorgehen bei der Function-Point-Analyse nach [Du01, Hue99, Dr89]

Das Vorgehen der FPA teilt sich in fünf Aktionen auf. Zunächst werden quantitative Aussagen über funktionale Anforderungen durch eine Einteilung in Basisfunktionskomponenten (BFC) getroffen und als gewichtete, aber unbewertete, Function Points (UFP) erfasst. Diese sind dann, mit Hilfe eines Interpretationsmodells, so zu interpretieren, dass die Möglichkeit existiert qualitative Aussagen, erfasst durch bewertete Function Points (AFP), abzuleiten. Hierfür werden die UFPs mit einem technischen Komplexitätsfaktor (TCF) bewertet. Im letzten Schritt werden, auf Basis von Erfahrungswerten, diese AFPs in Zeitaufwand für die weitere Projektplanung umgerechnet.

Obwohl die Function-Point-Analyse nicht direkt zum Ziel hatte, mögliche Architekturvarianten zu vergleichen, sondern lediglich die Abschätzung eines Entwicklungsaufwandes an Hand der Anforderungen [Dr89; Du01], bestand die Vermutung, dass sie für die Bewertung von VLBA-Architekturvarianten in abgewandelter Form eingesetzt werden kann. Grundannahme ist es, dass die ermittelten UFPs als quantitative Größe der Anforderungen, die aus den Benutzeranforderungen abgeleitet werden, mit jeder Variante auf unterschiedliche Art und Weise realisiert werden. Der Aufwand für die Umsetzung ist demnach für jede Architekturvariante unterschiedlich zu bewerten. Im folgenden Kapitel soll das Vorgehen am Beispiel der Bayer AG verdeutlicht.

#### 4 Bewertung und Interpretation der Architekturvarianten bei der Bayer AG

Die Wichtungsfaktoren der Basisfunktionskomponenten (Tabelle 1) basieren auf einem Standard (ISO/IEC 20926) zur FPA der durch die International Function Point User Group (IFPUG) festgelegt wurde, um eine möglichst einheitliche Function-Point-Berechnung zu schaffen. Dieser Standard wurde durch die IEC/ISO 14143-1 zertifiziert und ist damit international anerkannt [Du01, Hue99, Dr89].

BFC-Typ	Gewicht		
	einfach	mittel	schwer
Eingabe	3	4	6
Ausgabe	4	5	7
Abfrage	3	4	6
Interne Geschäftsentität	7	10	15
Externe Geschäftsentität	5	7	10

Tabelle 1: Wichtungsfaktoren bei der FPA nach ISO/IEC 20926 [Du01; Hue99; Dr89]

Die Summe der zugeordneten (einfach, mittel, schwer) und gewichteten funktionalen Anforderungen ergeben die unbewerteten Function Points (UFP – unadjusted function points) [Al79]. Die Erfassung der funktionalen Anforderungen und deren Gewichtung für die Bayer AG ist in der Tabelle 2 dargestellt.

ID	Funktionsanforderung	Komplexität	BFC-Typ	Gewicht
E-001-1	Extended Job Search	einfach	Eingabe	3
E-001-2	Job List	einfach	Abfrage	3
E-001-3	Job Details	einfach	Ausgabe	4
E-001-4	Send to a Friend	mittel	Ausgabe	5
E-002	Quick Job Search	einfach	Eingabe	3
E-003-1	Job Navigator (school leavers)	schwer	Abfrage	6
E-003-2	Job Navigator (graduates)	schwer	Abfrage	6
E-003-3	Job Navigator specialists	schwer	Abfrage	6
E-004	Current jobs	einfach	Abfrage	3
E-005	Current job on start page	einfach	Ausgabe	4
I-001	Register User	einfach	Eingabe	3

<b>I-002</b>	Overview page	einfach	Ausgabe	4
<b>I-003-1</b>	Candidate Profile	einfach	Eingabe	3
<b>I-003-1-1</b>	Interest Group	einfach	Eingabe	3
<b>I-003-1-2</b>	Work Experience	schwer	Eingabe + Ausgabe	6+7
<b>I-003-1-3</b>	Education	schwer	Eingabe + Ausgabe	6+7
<b>I-003-1-4</b>	Qualifications	schwer	Eingabe	6
<b>I-003-1-5</b>	Attachments	schwer	Eingabe	6
<b>I-003-1-6</b>	Desired Job	mittel	Eingabe	4
<b>I-003-1-7</b>	Data Overview	schwer	Ausgabe	7
<b>I-003-1-8</b>	Profile Release	einfach	Eingabe	3
<b>I-003-2</b>	Delete Registration	mittel	Eingabe	4
<b>I-004-1</b>	Apply Directly	mittel	Eingabe + Ausgabe	4+5
<b>I-004-2</b>	Search for Jobs	einfach	Eingabe	3
<b>I-004-2-1</b>	Job list	einfach	Abfrage	3
<b>I-004-2-2</b>	Job detail	einfach	Ausgabe	4
<b>I-004-3</b>	Favourites	einfach	Ausgabe	4
<b>I-004-4</b>	Applications	einfach	Ausgabe	4
<b>I-005-1</b>	Personal data	einfach	Eingabe	3
<b>I-005-2</b>	Communication data	einfach	Eingabe	3
<b>I-006-1</b>	Personal Settings	einfach	Eingabe	3
<b>I-006-2</b>	Change User Name	einfach	Eingabe	3
<b>I-007-1</b>	Application Wizard	einfach	Ausgabe	4
<b>I-007-2</b>	Cover Letter	einfach	Ausgabe	4
<b>I-007-3</b>	Submit Application	einfach	Eingabe	3
			<b>Function Points (UFP):</b>	<b>162</b>

Tabelle 2: Anforderungen bei der Bayer AG aus Benutzersicht [Bay06c]

In Abbildung 6 ist zu erkennen, dass diese UFPs durch ein Interpretationsmodell bewertet werden. Zur Vorbereitung einer endgültigen Entscheidung über die Architekturvarianten müssen demnach Eigenschaften bestimmt werden, welche den jeweiligen technischen Schwierigkeitsgrad (TCF) je Architekturvariante genauer beschreiben. Hierfür werden im IFPUG Standard verschiedene Merkmale (GSC – general system characteristics) benannt. Diese müssen ebenfalls innerhalb einer durch diesen Standard festgelegten Skala eingeschätzt werden. Der TCF ist ein zusammenfassendes Maß für entwicklungs- und produktumfeldbezogene Einflussgrößen des Aufwands. Er bezieht sich im Gegensatz zu den UFPs auf die gesamte Anwendung und nicht auf einzelne Funktionalitäten [Dr89, Hue99, Du01]. Die zu bewertenden Einflussgrößen des IFPUG Standards sind allerdings qualitative Einflussgrößen bezüglich des Aufwands [Hue99]. Der TCF enthält demnach sämtliche Einflussgrößen, die den gemessenen Funktionsumfang der Software nach oben oder unten korrigieren, um eine Aufwandsabschätzung für das Entwicklungsprojekt durchzuführen [Du01].

Im Vordergrund einer Architekturentscheidung steht allerdings nicht unbedingt der Entwicklungsaufwand, sondern vielmehr die Vergleichbarkeit der Fähigkeit mit der eine Variante den geforderten Funktionsumfang umsetzt, um daraus die adäquateste Wahl zu treffen. Es ist demnach notwendig ein nutzenbezogenes Qualitätsmodell dem TCF entgegenzusetzen, welches sämtliche Einflussgrößen erfasst die den Nutzen und damit den Funktionsgehalt der geforderten Lösung senken oder erhöhen.

Aus diesem Grund wurde ebenfalls der Nutzen betrachtet. Hierfür wurde angenommen, dass in der reinen Frame-Integration nur Anpassungsaufwand entsteht und dieser keinen weiteren Nutzen stiftet, da bereits vorhandene Funktionalität verwendet wird [Bay06a]. Um den reinen Entwicklungsaufwand vom Anpassungsaufwand zu trennen, wurde zunächst auf Grund der Architekturbeschreibungen eine Verteilung der UFPs nach Entwicklung und Anpassungsanforderungen durchgeführt (Tabelle 3). Dabei ist zu erkennen, dass für die Back-End Lösung (Architekturvariante 2) im Bereich der Entwicklung sämtliche UFPs neu entwickelt werden müssen. Dies deckt sich mit den Annahmen aus dem Kapitel 2.

Zudem hat sich, an Hand der Benutzeranforderungen herausgestellt, dass bei der Architekturvariante 1 (Frame-Integration) sieben Funktionsanforderungen nicht direkt durch die SAP® e-Recruiting Plattform angeboten werden können und hier ebenfalls Entwicklungsaufwand besteht [Bay06a]. Daher wird die reine Frame-Integration nicht realisierbar sein, da sämtliche geforderte Anforderungen umgesetzt werden müssen. Aus technischer Sicht stellt sie nunmehr, in abgewandelter Form, eine andere Variante der hybriden Lösung dar.

<b>Architektur-variante</b>	<b>Entwicklung</b>	<b>Enthaltene Anforderungen</b>	<b>Anpassung</b>	<b>Enthaltene Anforderungen</b>
Frame-Integration	33	E-001-4, E-002, E-003-1 bis E-003-3, E-004, E-005	129	E-001-1, E-001-2, E-001-3, alle im geschützten Bereich
Back-End Lösung	162	alle	0	keine
Hybride Lösung	43	alle im öffentlichen Bereich	119	alle im geschützten Bereich

Tabelle 3: Verteilung der UFP nach [Bay06a]

Für die Entscheidung zwischen den in Tabelle 3 dargestellten Architekturvarianten wurde angenommen, dass der Entwicklungsaufwand höher zu bewerten ist als der Anpassungsaufwand. Des Weiteren wurde angenommen, dass die Entwicklung einen abnehmenden Grenznutzen besitzt und somit die Architekturvariante der reinen Back-End Lösung ebenfalls nicht gewählt werden sollte. Hier werden 100% der bereits vorhandenen Funktionalitäten der SAP® e-Recruiting Plattform erneut implementiert, wobei auf Grund des abnehmenden Grenznutzens anzunehmen war, dass der Entwicklungsaufwand mit dem Nutzen nicht gerechtfertigt werden konnte.

Die beiden übrigen Lösungsvorschläge unterscheiden sich nur geringfügig im Entwicklungsaufwand (Tabelle 3) – 7% Unterschied in der Anzahl der neu zu entwickelnden Function Points. Auf Grund der klaren und dem Endbenutzer ersichtlichen Abgrenzung der in Variante 3 vorgeschlagenen Trennung zwischen öffentlichem und geschützten Bereich und dem sich nur gering unterscheidenden Entwicklungsaufwand, wurde die Variante 3 empfohlen [Bay06a].

## **6 Zusammenfassung und Ausblick**

In diesem Artikel wurde ein Entscheidungsproblem bei VLBA-Architekturvarianten vorgestellt, welches bei der Bayer AG im Rahmen der Umstrukturierung des weltweiten HR-Bereiches aufgetreten war. Hier sollte ein bereits existierendes Altsystem durch die Integration der SAP® e-Recruiting Plattform und OpenCms abgelöst werden. Dafür wurden zu Beginn des Projektes drei Architekturvarianten vorgestellt, die zur Entscheidung vorlagen.

Auf Grund der technologieneutralen Bewertung des Funktionsumfangs durch Function Points wurde die Function-Point-Analyse gewählt, um die Architekturvarianten zu bewerten. Hierbei wurde angenommen, dass sich die Architekturvarianten auf Grund ihrer Verteilung der gleichen Function Points in der Umsetzung unterscheiden. Dabei konnte festgestellt werden, dass die klassische FPA nicht direkt geeignet ist, um Architekturvarianten zu vergleichen, da sie aufwandsorientiert ist.

Dennoch wurde die Function-Point-Analyse für die Bewertung von VLBA-Architekturvarianten eingesetzt. Dafür musste neben der aufwandsorientierten, auch eine nutzenorientierte Betrachtung erfolgen. Im vorliegenden Artikel wurde diese nutzenorientierte Betrachtung zum einen durch die Aufteilung in Entwicklungs- und Anpassungsaufgaben, sowie durch die Annahme eines abnehmenden Grenznutzens, den eine Entwicklung bezüglich einer Wiederverwendbarkeit bzw. bezüglich der Layout-Flexibilität stiftet, erreicht.

Abschließend bleibt anzumerken, dass ein einheitliches nutzenorientiertes Qualitätsmodell auf Basis der Eigenschaften von VLBAAs, gerade für den Vergleich von Architekturvarianten von Vorteil ist und dies weiterverfolgt werden sollte.

Zu dem gibt es eine Reihe von weiteren Methoden, die sich mit dem Thema Software-Architekturbewertung befassen, wie zum Beispiel die „Software Architecture Analysis Method“ (SAAM) und „Architecture Tradeoff Analysis Method“ (ASAM) [CI01].

## Literaturverzeichnis

- [Al79] Albrecht, A. J. (1979): Measuring Application Development Productivity, Proceedings SHARE/GUIDE IBM Applications Development Symposium, Monterey, CA., Oct 14-17, 1979.
- [Bal98] Balzert, H. (1998): Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Spektrum Akademischer Verlag, Heidelberg, 1998
- [BayST] Lüttge, S. (2006): Bayer HR Web Styleguide. Internes Arbeitspapier der Bayer AG.
- [Bay06a] Zwanziger, A. (2006): Transforming Human Resources – Software Architectures to integrate the SAP e-Recruiting platform and OpenCms. Internes Arbeitspapier der Bayer AG.
- [Bay06b] Osterkamp, A.; Zwanziger, A. (2006): Transforming Human Resources – Fallback Alternatives. Internes Arbeitspapier der Bayer AG.
- [Bay06c] Zwanziger, A.; Herden, S. (2006): Transforming Human Resources – Excel-Sheet zur Erfassung und Bewertung der Function-Points. Internes Arbeitspapier der Bayer AG.
- [Bay06d] Bergmann, B.; Herden, S.; Jäger, D.; Schmitz, A.; Schulte, T. (2006): Transforming Human Resources – HR Portal Basic Architecture. Internes Arbeitspapier der Bayer AG. Version 1.14
- [Cl01] Clements, P.; Kazman, R.; Klein, M.: Evaluating Software Architectures. Methods and Case Studies.: Methods and Case Studies (SEI Series in Software Engineering). Addison-Wesley Longman, Amsterdam, 2001
- [Dr89] Dreger, J. B. (1989): Function Point Analysis. Prentice Hall Advanced Reference Series. Prentice Hall, New Jersey. 1989
- [Du01] Dumke, R. (2001): Software Engineering: Eine Einführung für Informatiker und Ingenieure: Systeme, Erfahrungen, Methoden, Tools. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden, 2001
- [Fey06] Fey, C. (2006): Transforming the Human Resources Function. HRO World Europe 2006, 6-8. November 2006, Brüssel
- [Gra07] Grabski, B.; Günther, S.; Herden, S.; Krüger, L.; Rautenstrauch, R.; Zwanziger, A.: Very Large Business Applications. In: Informatik Spektrum, 20. Jg., Heft 4, 2007; S. 259-263.
- [He04] Heinrich, L.; Heinzl, A.; Roithmayr, F. (2004): Wirtschaftsinformatik-Lexikon. Oldenbourg Verlag München, Wien. 2004
- [Her06] Herden, S.; Gomez J.; Rautenstrauch, C.; Zwanziger, A. (2006): Software-Architekturen für das E-Business. Springer-Verlag, 2006
- [Hue99] Hürten, R. (1999): Function-Point Analysis – Theorie und Praxis: Die Grundlage für ein modernes Softwaremanagement. expert-Verlag Renningen-Malmsheim. 1999
- [W3C] World Wide Web Consortium (W3C): XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition); <http://www.w3.org/TR/xhtml1/>