

# WHITEPAPER

## Innovative Integration von Software-Entwicklungssystemen

### MANAGEMENT SUMMARY

Wurden und werden Integrationen zwischen Software-Entwicklungswerkzeugen noch häufig mithilfe von unflexiblen und aufwändig zu pflegenden Punkt-zu-Punkt-Skripten realisiert, so existiert mittlerweile eine wesentlich vielversprechendere Lösung. **Effizienter mit serviceorientierter Architektur** lautet die aus IT- und ökonomischer Sicht sinnvolle Alternative! Anforderungen wie beispielsweise Flexibilität, Skalierbarkeit und die einfache Integration von Systemen und externen Partnern werden damit spielend abgedeckt.

Dieses Whitepaper gibt Hintergrundinformationen zur Technologie, stellt Anwendungsbeispiele vor und zeigt die technischen und wirtschaftlichen Vorzüge im Vergleich zu anderen Ansätzen auf.

Autoren:

Prof. Dr.-Ing. Peter Fromm, Hochschule Darmstadt  
Ralf Klimpke, agosense GmbH

### VITA

#### Prof. Peter Fromm

Dr. Fromm hat in Aachen Elektrotechnik studiert. Nach seiner Promotion für Mechatronik 1999 baute er den Bereich Software und System Engineering bei der Firma VDO/Division PKW Informationssysteme (später Siemens und Continental) auf. Zu den Schwerpunkten seiner Aufgaben gehörten die erfolgreiche Einführung von CMMi Level 3 sowie die Definition und Implementierung einer weltweit standardisierten Engineering Toolchain. Inhaltliche Schwerpunkte seiner Tätigkeiten waren die Prozessbereiche Konfigurations- und Change Management, Requirements Engineering und Software Engineering. In 2008 übernahm er eine Professur für Microcontroller und Informationstechnik an der Hochschule Darmstadt.

### VITA

#### Ralf Klimpke

Ralf Klimpke studierte Wirtschaftsingenieurwesen an der Fachhochschule für Technik in Esslingen. Seit 1995 war er unter anderem bei der MKS GmbH (seit 2011 PTC), einem Anbieter für Application Lifecycle Management Werkzeuge, in verschiedenen Positionen tätig. Dort verantwortete er den Aufbau eines indirekten Vertriebskanals für eine Produktlinie. Später etablierte er als Key Account Manager durch die Gewinnung zahlreicher namhafter Großkunden maßgeblich die starke Präsenz des Unternehmens in der Automobil-Branche. Seit 2009 ist Ralf Klimpke als Mitgründer und Geschäftsführer der agosense GmbH tätig und bringt insgesamt mehr als 15 Jahre Erfahrung in Vertrieb und Marketing in das Unternehmen ein.

**Die effektive Integration von Software Engineering Systemen sowie der automatisierte Austausch von Entwicklungsdaten sind heiß diskutierte Themen in IT- und Entwicklungsabteilungen. Es gilt, häufig über Unternehmensgrenzen hinweg, Systeme aus verschiedenen Prozessbereichen zu einer durchgängigen Toolkette zu integrieren. Angefangen beim Requirements Management über das Test Management sowie Change- und Defect Management bis hin zum Konfigurationsmanagement. Speziell in prozessorientierten Branchen, z. B. in der Automobilindustrie, Medizintechnik oder auch der Finanzbranche, ist eine integrierte Werkzeugkette sowohl für eine effiziente als auch für eine aus Prozesssicht nachvollziehbare Software- und Systementwicklung unabdingbar geworden.**

Die Realität zeigt allerdings, dass nicht jedes Unternehmen eine in sich integrierte ALM Suite<sup>1</sup> eines einzelnen Herstellers als Standard einsetzen kann und möchte. So werden immer wieder Prozesslücken entstehen, beispielsweise im Bereich der Modellierung oder wenn bestimmte Anwendergruppen hochspezialisierte Werkzeuge einsetzen, welche nicht Teil einer ALM Suite sind (Best-of-Breed-Strategie)<sup>2</sup>.

Darüber hinaus machen Entwicklungsprozesse nicht an den eigenen Unternehmensgrenzen halt. Dort wo Kunden-Lieferanten-Verhältnisse oder Entwicklungspartnerschaften bestehen, werden häufig Informationen (elektronisch) ausgetauscht. Und auch hier gilt die Anforderung, Abläufe zu automatisieren und zu dokumentieren, deren Einhaltung sicherzustellen und jederzeit kontrollieren zu können.

Wurden diese Integrationsanforderungen bislang mithilfe von so genannten Punkt-zu-Punkt-Integrationen realisiert, soll hier ein innovativer und vielversprechender Ansatz vorgestellt werden.

### **Bisheriger Ansatz: Punkt-zu-Punkt Integration**

Wie wurden (und werden) Integrationen bisher realisiert? Üblicherweise werden Synchronisationsskripte und -programme im Unternehmen bzw. der Fachabteilung selbst oder von externen Beratungsunternehmen erstellt – mit dem Vorzug, dass diese in aller Regel den gewünschten Anwendungsfall abbilden. Doch was passiert, wenn der Entwicklungsprozess modifiziert wird oder neue Versionen der integrierten Tools eingesetzt werden sollen? Wenn bestimmte Gruppen/Abteilungen mit verschiedenen Arbeitsabläufen operieren? Dann müssen Änderungen an den Skripten erstellt bzw. beauftragt, getestet und vielleicht sogar in unterschiedlichen Varianten an eine große Anzahl von Anwendern verteilt werden.

Darüber hinaus werden diese Integrationen häufig von den Fachbereichen selbst erstellt und gepflegt und haben sich in vielen Unternehmen zwischenzeitlich zu einer kostenintensiven Parallel-IT entwickelt. Die Gründe sind auf der einen Seite einleuchtend, denn die Entwicklungsabteilungen sind in der Lage, auf Änderungsanforderungen wesentlich schneller zu reagieren als eine zentrale IT-Abteilung. Auf der anderen Seite müssen die Fachabteilungen von ihren eigentlichen Tätigkeiten - nämlich der Produktentwicklung und Projektarbeit - immer mehr Zeit für die Pflege, Wartung und Anpassung dieser Integrationen aufwenden und können üblicherweise auch kein optimales Betriebskonzept sicherstellen.

Teilweise werden Plug-Ins zur Datensynchronisation auch von Drittanbietern bzw. den Herstellern der Entwicklungswerkzeuge selbst angeboten. In Bezug auf die benötigte Funktionalität und Anpassbarkeit an die eigenen Abläufe und funktionalen Anforderungen stellen diese jedoch immer Kompromiss-Lösungen dar. Vor allem wenn neue Produktversionen der eingesetzten Werkzeuge implementiert werden sollen, wird die Abhängigkeit zum Werkzeug-Hersteller deutlich. Denn gerade wenn Updates durchgeführt werden (müssen), unterstützt die jeweilige Integrationstechnik häufig noch nicht diese neuen Versionen.

Je verflochtener die Werkzeuglandschaft integriert ist, desto stärker ist natürlich diese Abhängigkeit. Notwendige Produktaktualisierungen können deshalb verzögert oder gar verhindert werden.

Die Nachteile solcher starren Integrationen, die meistens nur zwei spezifische Tools verbinden, sind also offenkundig. Sie sind in der Erstellung und in der Pflege höchst aufwändig und dazu sehr anfällig gegen Änderungen und Updates einzelner Toolkomponenten. Hinzu kommt, dass die Werkzeuglandschaft mit steigender Anzahl an Tools und dementsprechenden Punkt-zu-Punkt-Integrationen zunehmend unübersichtlicher und schwerfälliger in der Wartung und Pflege wird.

<sup>1</sup> ALM Suite – in sich hochintegrierte Werkzeugplattform, die möglichst viele Prozessbereiche im Application Lifecycle Management abdeckt.

<sup>2</sup> Best-of-Breed-Strategie: beschreibt den Ansatz, für den jeweiligen Prozessbereich das für das eigene Unternehmen am besten geeignete bzw. spezialisierte Werkzeug einzusetzen.

Auch die Handhabung in Bezug auf den Betrieb und die Prozess-Sicherheit ist komplex: wie gut skaliert dieser Ansatz bei großen Datenmengen oder einer großen Anzahl von Endanwendern? Wie aufwändig ist das Ausrollen der Skripte und Plug-Ins an die einzelnen Arbeitsplatzrechner? Muss der Anwender oder ein Administrator mögliche Konflikte manuell auflösen? Wer stellt sicher, dass ein Anwender seine aktualisierten Daten auch für den nächsten Prozess-Schritt synchronisiert hat? Dies sind nur einige der bekannten Fragestellungen, denen hauptsächlich mit viel Disziplin und manueller (Nach-)Arbeit begegnet wird.

Offenbar wird der insgesamt betriebene Aufwand für die Aufrechterhaltung dieser Integrationsansätze also unterschätzt. Doch welche Alternativen gibt es, heterogene Werkzeuglandschaften wirklich effizient und flexibel zu integrieren?

### Die Suche nach einer effizienten Alternative

Bevor die Frage nach Alternativen beantwortet werden kann, sollen zunächst die häufigsten Anforderungen an eine moderne Integrationstechnologie definiert werden. Dabei werden hauptsächlich genannt:

- Flexible Infrastruktur, die sich leicht in bestehende IT-Landschaften einbetten lässt
- Einfache Integration weiterer (auch selbstentwickelter) Werkzeuge
- Integrationsmöglichkeit weiterer Prozessbereiche wie z. B. Product Lifecycle Management, Projekt Management, ...
- Skalierbarkeit und Verarbeitungsgeschwindigkeit (Echtzeit-Synchronisation)
- Automatisierung von Prozessen und Reduktion manueller Fehlerquellen
- Robustheit, Behandlung von Fehlern und Konflikten, Reporting/Logging
- Kein Zusatzaufwand oder zusätzliche Bedienoberflächen für den Endanwender
- Zentraler IT-Betrieb bei gleichzeitiger dezentraler Konfiguration für Gruppen oder Projekte
- Vollständiges, von einem unabhängigen Hersteller unterstütztes Produkt
- Leichte Erlernbarkeit bzw. Verwendung von Standardtechnologien

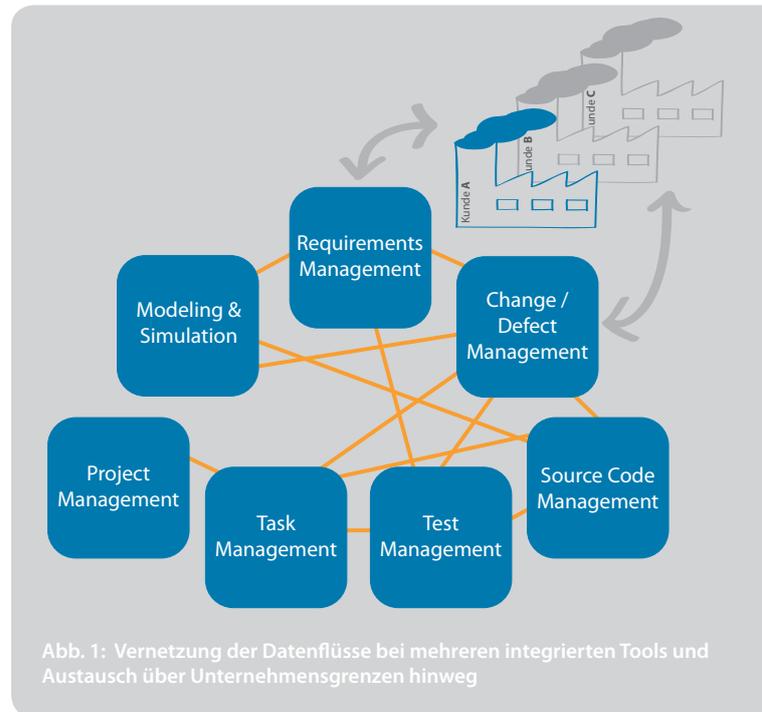
Für den Datenaustausch mit externen Partnern:

- Automatisierungsmöglichkeiten für das Abholen und Bereitstellen von Daten in Unternehmensportalen
- Protokollierung/Aufzeichnung von Austauschvorgängen in beide Richtungen zur Sicherstellung von Compliance Anforderungen für „externe“ Prozesse
- Unterstützung standardisierter Datenaustauschformate, wie XML, ASAM, RIF/ReqIF, etc.
- Vom Hersteller supportetes Produkt - keine Individualprogrammierung oder Dienstleistung

### Effizienter mit serviceorientierter Architektur

Ein Blick über den Tellerrand der ALM-Welt und der Softwareentwicklung hinaus zeigt, dass zum Beispiel im Bereich der Geschäftsprozesse bereits seit langem etablierte Standardtechnologien eingesetzt werden.

Beispielhaft für die Integration von Businessanwendungen und Geschäftsprozessen stehen sogenannte serviceorientierte Architekturen (SOA), welche bereits seit Jahren erfolgreich eingesetzt werden und dementsprechend ausgreift sind. Und letztendlich sind Prozesse im Software Engineering nichts anderes als spezialisierte Geschäftsprozesse. So drängt es sich nahezu auf, diese Standardtechnologie auch im Bereich der Softwareentwicklung zu adaptieren.



Doch was genau verbirgt sich hinter diesem Begriff? Eine Definition lautet beispielsweise: „Eine serviceorientierte Architektur (SOA) ist ein Konzept, welches das Geschäft und die IT eines Unternehmens nach Diensten strukturiert, welche modular aufgebaut sind und flexibel zur Umsetzung von Geschäftsprozessen genutzt werden können. Dabei lässt sich eine serviceorientierte Architektur nicht fertig auf dem Markt kaufen, vielmehr ist SOA ein Konzept, das jeweils auf die individuellen Gegebenheiten – sowohl hinsichtlich der Organisation als auch der bestehenden Anwendungslandschaft – eines Unternehmens angepasst werden muss.“<sup>3</sup>

Im Wesentlichen gehören dazu die folgenden Komponenten:

- Basisdienste zur (webbasierten) Kommunikation
- Entkoppelte Dienste zur Abbildung der Geschäftsprozesse
- Adapter, welche die Datenschnittstellen zu den einzelnen Applikationen implementieren
- Applikation für die grafische Modellierung der Geschäftsprozesse
- Ausführungsschicht für die Geschäftsprozesse (Prozessdefinition und -abbildung)
- Administrations- und Präsentationsfunktionen

Auf die Welt der Softwareentwicklung übertragen, kann dieses Konzept folgendermaßen dargestellt werden: Abbildung 2 verdeutlicht schematisch den Aufbau einer mittels SOA integrierten Werkzeugkette. Über eine Kollaborationsschicht werden die Daten, in der Regel aus dem Projektmanagement, Requirements Management, Change Management und Test Management zwischen den externen Beteiligten ausgetauscht. Die Integrationslösung ersetzt hier zunächst primär den klassischen Datenaustausch per Papier, E-Mail oder Datenträger. Die Daten werden von einem Quellsystem, z.B. einem FTP-Server oder Portalsystem gelesen, bei Bedarf gefiltert, syntaktisch und semantisch interpretiert, gewandelt und dann in das Zielsystem eingespielt. Auf dieselbe Art und Weise werden die internen Werkzeuge zu einer nahtlos verbundenen Toolkette integriert. Allerdings mit dem Unterschied, dass hier die Daten nicht über ein Zwischenformat ex- und importiert werden müssen. Hier werden die einzelnen Werkzeuge über Adapter direkt an ihren Integrationsschnittstellen angesprochen. Punkt-zu-Punkt-Integrationen werden somit überflüssig.

Die einzelnen Dienste einer service-orientierten Architektur können dabei den drei Schichten Process Layer, Middleware und Adapter zugeordnet werden.

Abbildung 3 zeigt hier die beteiligten Komponenten und Dienste der in Abbildung 2 dargestellten „Integration Plattform“.

Die Adapterschicht Data Exchange ist für den physikalischen Austausch von Daten zuständig und bildet die direkte technische Anbindung an die einzelnen Werkzeuge ab. Sie ist eng mit der unteren Middleware-Schicht Data Integrity verzahnt, die für den fehlerfreien und vollständigen Transfer Sorge trägt, aber auch für Performanceoptimierungen verantwortlich ist.

Die obere Middleware-Schicht Data Conversion ist für die semantische Umwandlung der Daten zuständig und somit eng mit der eigentlichen Prozessschicht verzahnt. Hier sind etwa Dienste für

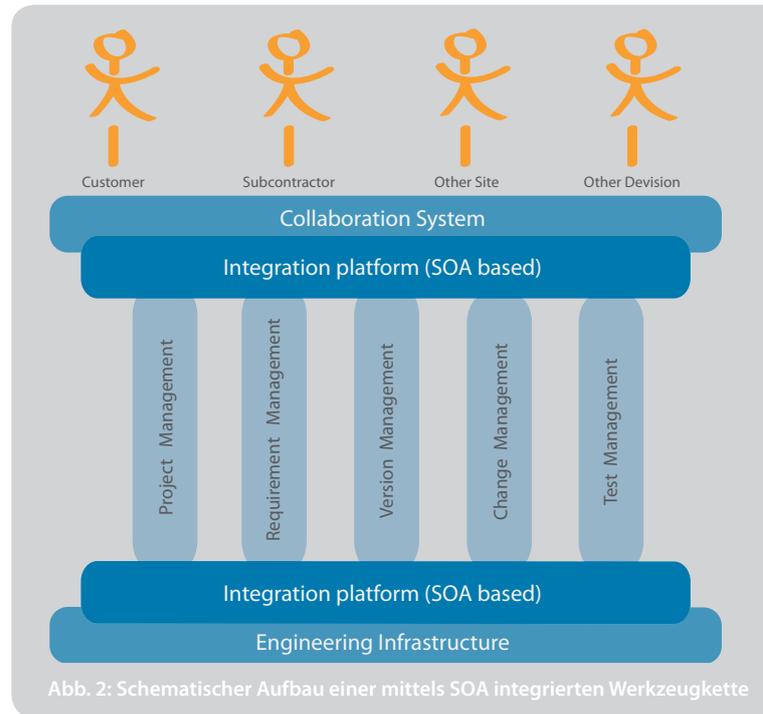


Abb. 2: Schematischer Aufbau einer mittels SOA integrierten Werkzeugkette

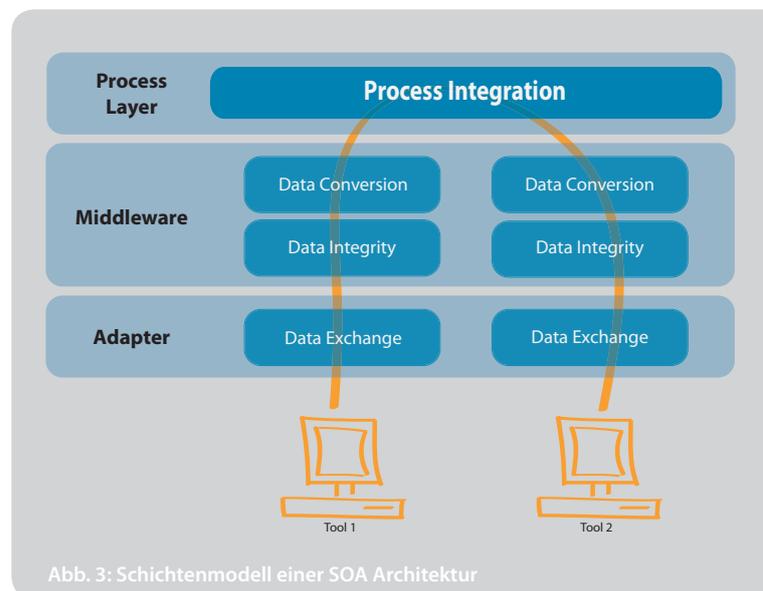


Abb. 3: Schichtenmodell einer SOA Architektur

das Mapping von Attributwerten bzw. Datenfeldern (z.B. Statusinformationen, ...) sowie Dienste für die Umwandlung von Datensätzen in firmenspezifische Formate angesiedelt.

Die so vereinheitlichten Daten werden dann auf der Ebene der Process Integration verarbeitet und an das Zielsystem übertragen. Dies erfolgt mittels der grafischen Modellierungssprache „BPEL“ (Business Process Execution Language). Über diese Prozessmodellierung und mithilfe geeigneter Funktionen der Adapter lassen sich so definierte Prozesse einfach dokumentieren und 1:1 ausführen.

**Ziele und Anwendungsfälle**

Ein wichtiges Ziel einer solchen Integration ist die Bereitstellung eines weitestgehend automatisierten Datenaustausches zwischen den verschiedenen beteiligten Systemen über Netzwerk- und Unternehmensgrenzen hinweg. Einhergehend mit dieser Automatisierung des Austausches ist der Effizienzgewinn ein wesentlicher Treiber für diese Lösung. Ebenso bedeutsam ist aber auch die äußerst flexible Integrationsmöglichkeit der internen, heterogenen Werkzeuglandschaft. Mithilfe der Prozessmodellierung kann diese genau an die Bedürfnisse des Unternehmens angepasst werden. Durch diese Automatisierung werden manuelle Tätigkeiten und somit die Kosten stark reduziert. Zudem steigen die Qualität der Daten, sowie die Prozessqualität (Nachvollziehbarkeit) und die Performance erheblich.

Typische Anwendungsfälle für die Zusammenarbeit externer Partner sind etwa der Austausch von Anforderungsdokumenten, Änderungsanfragen und Fehlermeldungen zwischen Automobilhersteller und Zulieferern. Standardisierte Datenformate wie RIF/ReqIF oder ASAM ermöglichen hier einen automatisierten und werkzeugunabhängigen Datenaustausch über Unternehmensgrenzen hinweg.

**Anwendungsfall der Marquardt GmbH, Automobilzulieferer und Spezialist für Schaltersysteme mit Hauptsitz in Rietheim-Weilheim:**

*Nachdem ein Produktmusterstand an den Kunden (Daimler AG) geliefert wurde, beginnt dieser mit ersten Funktions- und Integrationstests für das entsprechende Fahrzeugprojekt. Dabei auftretende Fehler oder Abweichungen zur vereinbarten Spezifikation werden von den Daimler Ingenieuren erfasst und über das Lieferantenportal „DanTe“ in Form einer XML-Datei bereit gestellt.*

*Marquardt verfügt über einen eigenen Zugang zu diesem Portal und kann hier die entsprechenden Dateien abholen. Dabei wird das Portal von agosense.symphony in konfigurierbaren Zeitintervallen auf neue Nachrichten geprüft. Liegen neue Daten vor, werden diese für das jeweils entsprechende Entwicklungsprojekt automatisch in das Änderungsmanagement (in diesem Falle Serena Dimensions) intern bei Marquardt importiert und dem zuständigen Projekt-Team zugeordnet. Während der Bearbeitung der Meldungen innerhalb der Entwicklungsabteilung bei Marquardt werden definierte Zwischenstände an den Kunden ebenfalls vollständig automatisiert über agosense.symphony zurückgemeldet, bis letztlich die endgültige Fehlerbehebung stattgefunden hat. Parallel dazu ist es auch möglich, dass entweder der Kunde selbst oder Marquardt nach internen Tests weitere Fehler oder Status-Updates über die Plattform zurückmelden. Somit haben alle Projektbeteiligten zu jedem Zeitpunkt einen aktuellen Status über den Reifegrad des in der Entwicklung befindlichen Produkts und damit eine zuverlässige Grundlage für die Planung der Aktivitäten.*

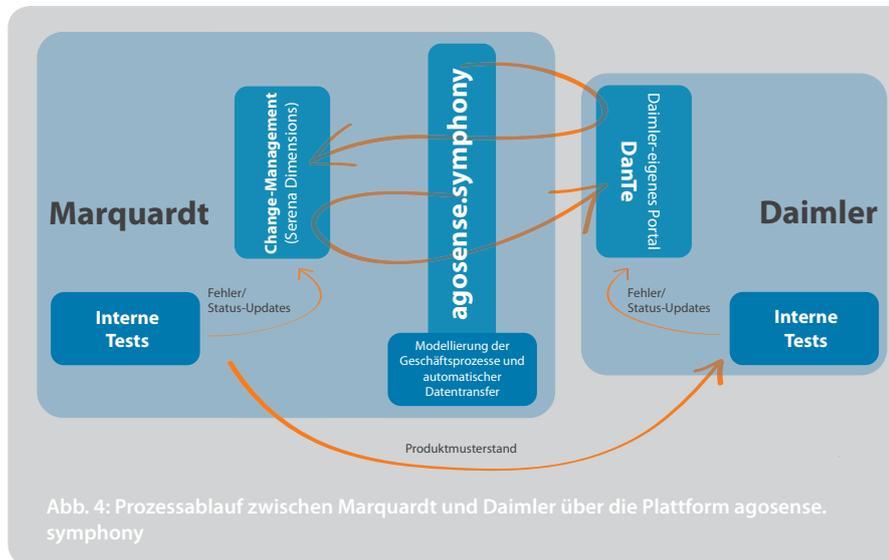


Abb. 4: Prozessablauf zwischen Marquardt und Daimler über die Plattform agosense.symphony

Damit die ausgetauschten Daten stets konsistent bleiben, müssen die jeweiligen internen Bearbeitungs- und Informationsprozesse sehr gut aufeinander abgestimmt werden. Diese Prozesse werden in agosense.symphony direkt modelliert und ausgeführt. Dadurch wird die Zuverlässigkeit der automatischen Synchronisierung sichergestellt.

**Anwendungsfall eines Schweizer Finanzdienstleisters:**

Um im Bereich Private Banking die Applikationsentwicklung zu steuern, werden neue Produktfeatures und Änderungsanfragen aus unterschiedlichsten Quellen, z.B. aus den Geschäftsbereichen, dem Betrieb oder aus der Qualitätssicherung zunächst in die zentrale Project Collaboration Applikation auf Basis von Atlassian Jira und Confluence eingestellt. Das Request Management ordnet, priorisiert und weist die für die jeweiligen Produkte bzw. Produktfreigaben relevanten Anträge so zu, dass diese - nach festgelegten Prozessen - automatisch an das für die Implementierung ausgewählte Projekt, die betroffenen Anwendungen und die zuständigen Anforderungsingenieure übergeben werden.

Unabhängig von der angewandten Entwicklungsmethode – iterativ, klassisch nach Wasserfall-Modell, etc. - die Entwicklungsdisziplinen Requirements Engineering, Solution Architecture, Software Engineering, Test Engineering und das zugehörige Configuration Management sind immer involviert, um die betroffenen Anwendungen im Rahmen des Produkt-Lebenszyklus anzupassen. Nur so ist die Bank in der Lage, mehrere hundert Anwendungen (Java, Mainframe und weitere Plattformen) nicht nur initial zu erstellen, sondern auch über die gesamte Lebensdauer zu pflegen.

Dies wird gewährleistet, indem den Anforderungsingenieuren die Änderungsanträge im System für Anforderungs-Management, Polarion Requirements, zunächst als Änderungsanfragen zugeordnet werden, damit diese die für die weitere Entwicklung notwendigen Spezifikationen und Anwendungsfälle erstellen können. Zugehörige UML-Modelle werden vom Anforderungsingenieur in Sparx Enterprise Architect erstellt und automatisch mit Polarion Requirements abgeglichen. Neue Statusinformationen zu den Anfragen werden regelmäßig an die Collaboration Anwendung in Atlassian Jira zurückgemeldet, so dass die ursprünglichen Antragsteller auch den Status der Umsetzung ihrer Anfragen jederzeit verfolgen können.

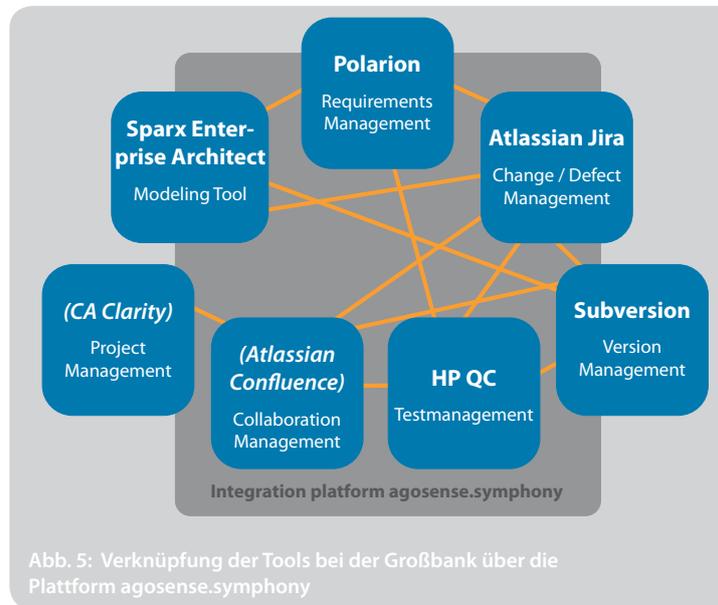


Abb. 5: Verknüpfung der Tools bei der Großbank über die Plattform agosense.symphony

Das Softwarearchitektur-Team benutzt die Anforderungen aus Polarion Requirements und weitere Informationen aus der technischen Analyse in Sparx Enterprise Architect, um die Architektur der Anwendung zu validieren und gegebenenfalls anzupassen. Dadurch wird eine konsistente Dokumentation der Anwendungsarchitektur über den gesamten Lebenszyklus gewährleistet. Softwareingenieure nutzen die Anforderungsspezifikation, die Architektur und Designdokumentation für die eigentliche Softwareentwicklung. Dabei können die Entwickler aus Eclipse heraus Projekt-Aufgaben und Anträge aus Atlassian Jira direkt sehen und ändern sowie mit Code-Änderungen verknüpfen. Parallel werden freigegebene Anforderungen in den Testbereich nach HP Quality Center/ALM übertragen. Die Testingenieure erstellen in diesem Werkzeug die korrespondierenden Testfälle für die neuen Anforderungen oder passen vorhandene Anwendungs-Testfälle an. Die Planung der Testaktivitäten wird mit der Projektplanung in Atlassian Jira synchronisiert. Ist die Codierung abgeschlossen, werden die Applikationen gemäß den Testfällen und Testplänen in HP QC/ALM geprüft. Für nicht bestandene Tests wird eine Fehlermeldung erstellt, die dann automatisiert nach Atlassian Jira an das Projektteam zur Bearbeitung zurückgespielt wird. Dabei werden auch notwendige Testergebnisse übergeben, so dass die Entwicklung in der Lage ist, Fehler qualifiziert zu analysieren und zu beheben – so lange, bis der Entwicklungszyklus letztendlich mit erfolgreichem Test abgeschlossen werden kann.

Die Tools der verschiedenen Rollen in diesem Softwareentwicklungsprozess werden durchgängig mit agosense.symphony modelliert und gesteuert, wobei selbstverständlich auch Fehler- und Konfliktsituationen behandelt werden. Darüber hinaus ist es je nach Prozessdefinition möglich, dass Daten entweder direkt durch Benutzerinteraktion (z.B. durch Statuswechsel) oder zeitgesteuert synchronisiert werden. So ist gewährleistet, dass jeder Prozessbeteiligte in seiner Rolle mehrheitlich in seinem spezialisierten Werkzeug arbeiten kann und dennoch alle notwendigen und persönlich relevanten Informationen erhält, ohne Tools oder Benutzeroberflächen wechseln zu müssen.

Für einen weiteren Ausbau dieser Werkzeugkette wird die Integration von Atlassian Confluence, CA Clarity sowie weiteren Anwendungen über die Integrationsplattform analysiert.

### Der Markt von SOA-Anbietern

Neben Produkten von Firmen wie z.B. Oracle, SAP, Microsoft oder IBM, die häufig einen Schwerpunkt auf die Integration der hauseigenen Tools legen, existieren nur sehr wenige Anwendungen mit speziellem Fokus auf Entwicklungssysteme, wie etwa die Integrationsplattform agosense.symphony der agosense GmbH. Im Gegensatz zu der einleitenden Definition einer service-orientierten Architektur ist agosense.symphony ein fertiges, in vielen Unternehmen bereits eingeführtes Produkt, welches bereits alle notwendigen Komponenten und Dienste enthält. Über die Methodik der grafischen Prozessmodellierung bietet die Plattform die größtmögliche Flexibilität für die Umsetzung der jeweils unternehmensspezifischen Regeln für die Definition der Datenflüsse zwischen Werkzeugen und beteiligten Parteien. Dabei stellt die Verwendung und Unterstützung offener Standards, wie z.B. BPEL, ASAM, RIF/ReqIF<sup>4</sup> etc. sicher, dass die Plattform leicht in bestehende Systemwelten eingebettet und an nahezu jedes beliebige Drittsystem angekoppelt werden kann.

Dabei erfüllt agosense.symphony alle hier dargestellten Anforderungen und bietet darüber hinaus noch einige weitere Vorteile gegenüber alternativen Ansätzen.

So kann durch die direkte Kopplung der Adapter an die jeweiligen APIs/Schnittstellen ein größtmöglicher Funktionsumfang der zu integrierenden Werkzeuge bereit gestellt werden. Andere Versuche, Schnittstellen herstellerübergreifend zu standardisieren, beschneiden grundsätzlich die individuellen Möglichkeiten der Tools. Die integrierte Ausführungsschicht für die Prozessmodelle erlaubt es, Abläufe ohne Programmierkenntnisse zu definieren und zu steuern. Außerdem erfährt agosense als unabhängiges Unternehmen eine breite Unterstützung seitens der Systemanbieter für die Entwicklung und Pflege einer Vielzahl von Adaptern aller namhaften Hersteller.

Abbildung 6 stellt die Architektur dieser Integrationsplattform schematisch dar. Im Adapter Framework werden - hier nur auszugswise dargestellt - einige der unterstützten Werkzeuge und Formate aufgeführt. Diese werden mittels spezifischen Adaptern an ein internes Kommunikationsbus-System angekoppelt. Die Adapter wiederum werden über das Bus-System durch die Process Execution gesteuert. Dabei werden unter Verwendung der grafischen Modellierungssprache BPEL Prozesse zur Regelung der Datenflüsse und Aktionen definiert und abgebildet. Oberhalb der Applikation sind so genannte funktionale Module integriert, die dem Administrator anhand von grafischen Browser-Oberflächen die einfache Konfiguration des Systems ermöglichen.

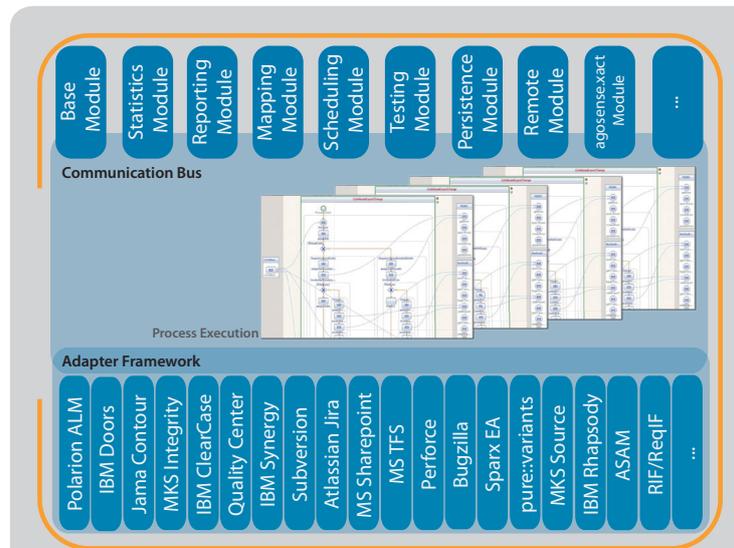


Abb. 6: Architektur der Integrationsplattform agosense.symphony

4 BPEL (Business Process Execution Language): moderner Standard für die grafische Modellierung und Ausführung von Geschäftsprozessen.  
 ASAM: ASAM e.V., Organisation die es sich zum Ziel gesetzt hat, eine Plattform für die Entwicklung offener Standards bereitzustellen.  
 Beispiele sind ASAM AEIssue, ASAM CC.  
 RIF/ReqIF (Requirements Interchange Format): OMG-Standardformat für den verlustfreien und toolneutralen Austausch von Anforderungsdokumenten.

## Fazit

„Innovative Integration von Software-Entwicklungswerkzeugen“ - Wie im Titel angekündigt, wurde im vorliegenden Whitepaper ein alternativer innovativer Ansatz zur Integration von Entwicklungstools vorgestellt. Mithilfe sogenannter serviceorientierter Architekturen lassen sich Prozesse in der Softwareentwicklung effizient abbilden. Eine mittel- bis langfristige Loslösung von den bisherigen unflexiblen und aufwändigen Punkt-zu-Punkt-Integrationen ist aus wirtschaftlicher und IT-Sicht unabdingbar.

## Über agosense

Als Anbieter der Integrations- und Austauschplattform [agosense.symphony](https://www.agosense.com) positioniert sich die agosense GmbH im Bereich der „Development Process Integration“. Das Unternehmen integriert heterogene Application Lifecycle Management (ALM) Umgebungen in der Software- und Systementwicklung zu einer durchgängigen Prozesskette.

Dabei richtet sich das Angebot hauptsächlich an Unternehmen, die ihre Entwicklungs- und Managementsysteme nachhaltig und unabhängig integrieren möchten, beziehungsweise eng mit ihren Geschäftspartnern vernetzt sind.

Die Plattform ermöglicht den prozessgesteuerten und automatisierten Datenaustausch im Rahmen des Entwicklungsprozesses – auch über Unternehmensgrenzen hinweg. Dabei werden unter anderem Standardformate wie RIF/ReqIF, ASAM, usw. für verschiedenste Entwicklungs-Werkzeuge unterstützt. Die Vielzahl verfügbarer Adapter für die gängigen Tools stellt eine schnelle Implementierung sicher. Die Architektur von [agosense.symphony](https://www.agosense.com) ermöglicht eine individuelle Anpassung der Integrations-Szenarien an die jeweiligen Kundenanforderungen. Unter anderem zählen folgende Unternehmen aus den unterschiedlichsten Branchen zu den Kunden:

- [Amadeus Germany GmbH](#)
- [BMW AG](#)
- [Bury GmbH & Co. KG](#)
- [Continental AG](#)
- [Daimler AG](#)
- [Intel Mobile Communications GmbH](#)
- [Robert Bosch GmbH](#)
- [Volkswagen AG](#)

Neben den Kunden, sind auch die Partnerschaften zu Werkzeug-Herstellern von großer Bedeutung für agosense. Nur durch eine enge Zusammenarbeit ist es möglich, die Adapter für die jeweiligen Applikationen und deren Versionen in entsprechender Qualität zu entwickeln beziehungsweise schnellstmöglich auf Änderungen zu reagieren.

Daneben bietet agosense auch diverse Lizenzmodelle für Werkzeug-Hersteller an. Spezifische Wünsche in Bezug auf Packaging, Branding und Customizing werden dabei gerne berücksichtigt. Die OEM-Partner bieten damit ihren eigenen Kunden durch standardisierte Technologien einen entscheidenden Mehrwert, wenn diese vor Integrationsherausforderungen zu Dritt-Werkzeugen stehen.

Ebenso zählen Beratungsunternehmen zu den Partnern von agosense. Mit verschiedenen Geschäftsmodellen, z.B. für Solution Delivery oder Service-Leistungen rund um agosense Produkte, erweitern diese ihr Service-Portfolio. So profitiert der Endkunde von standardisierten Lösungen, die dennoch flexibel an spezielle Bedürfnisse angepasst werden können.

## Kontakt

agosense GmbH  
Stammheimer Str. 10  
70806 Kornwestheim



fon +49.7154.99951.0  
fax +49.7154.99951.99

[info@agosense.com](mailto:info@agosense.com)

Weitere Informationen finden Sie auch im Web unter [www.agosense.com](http://www.agosense.com).