

Christian Zinke und Kyrill Meyer (Herausgeber/Eds.)

**Produkt- und Dienstleistungslebenszyklus-Management**  
*Theorie und Praxis für produktbezogene Dienstleistungen am Beispiel des  
Sondermaschinenbaus*

Leipziger Beiträge zur Informatik: Band/Volume XLVI  
Klaus-Peter Fährnich (Series Editor)

## Herausgeber/Volume Editors

**Christian Zinke**

**Dr. Kyrill Meyer**

Universität Leipzig

Abteilung Betriebliche Informationssysteme

Hainstraße 11

04109 Leipzig, Deutschland

*Finanziert aus Mitteln der Europäischen Union und des Freistaates Sachsen durch das Programm FuE-Projektförderung der SAB*



Gefördert aus Mitteln  
der Europäischen Union

Europa fördert Sachsen.



Weitere Informationen zu diesem Buch und dem Projekt PDLM unter:

<http://bis.informatik.uni-leipzig.de/de/Projekte/pdlm>

***Produkt-, und Dienstleistungslebenszyklus Management. Theorie und Praxis für produktbezogene Dienstleistungen am Beispiel des Sondermaschinenbaus***

***Christian Zinke und Kyrill Meyer***

***InfAI, Leipzig, 2014.***

ISBN: 978 -3-941608-34-4

## Vorwort Series Editor

In der Buchreihe „Leipziger Beiträge zur Informatik“ erscheinen Berichte aus Forschungsvorhaben, Herausgeberbände im Bereich innovativer und sich etablierender Forschungsgebiete, Habilitationsschriften und Dissertationen sowie Konferenz-Proceedings und herausragende studentische Arbeiten. Der Wert dieser durch den „Leipziger Informatik Verbund“ (LIV) als Zusammenschluss und Interessenverbund verschiedener Informatik-Einrichtungen im Jahr 2003 begründeten Reihe liegt darin, zeitnah und umfassend über abgeschlossene oder laufende wissenschaftliche Arbeiten sowie über neu entstehende Forschungsfelder zu berichten. Die Reihe stellt die innovative Themenvielfalt in den Herausgeberbänden neben die hohe wissenschaftliche Durchdringung in Habilitationen und Dissertationen. Zudem ergänzt sie forschungsrelevante Bereiche mit praxisorientierten technischen Beiträgen und Dokumentationen.

Die verschiedenen Beiträge entstammen dem Hintergrund der Angewandten Informatik und Wirtschaftsinformatik. Schwerpunkte liegen dabei in den Bereiche betriebliche Informationssysteme, Content- und Wissensmanagement sowie IT-gestützter intra- und interorganisationale Kooperation mittels Internet-Technologien. Besondere Beachtung findet auch – entsprechend seiner Bedeutung für den Standort – der Dienstleistungssektor, wobei ein besonderer Schwerpunkt auf Dienstleistungen mit einem hohen IT-Anteil gelegt wird.

Das Institut für Angewandte Informatik (InfAI) veröffentlicht Vorträge und fördert die weitere Entwicklung der Buchreihe. Das InfAI als An-Institut der Universität Leipzig verfolgt die gemeinnützige **Förderung von Wissenschaft und Forschung** auf den Gebieten der **Informatik und der Wirtschaftsinformatik**.

**Der vorliegende 47. Band der Reihe** beschäftigt sich mit dem Management und den technischen Unterstützungsmöglichkeiten von produktbezogenen Dienstleistungen in der Domäne des Spezialmaschinenbaus.

Klaus-Peter Fähnrich

Leipzig, im Juli 2014

Series Editor

Vorstandsvorsitzender InfAI

## ***Vorwort***

Deutschland ist noch immer ein führender Produktionsstandort, in dem Technologie von Weltruf für den globalen Markt entwickelt und zur Serienreife geführt wird. Gut ausgebildete Ingenieure und Fachkräfte ermöglichen dies und sorgen immer wieder für Innovationen. Nicht zuletzt aus diesem Umstand heraus erzielt das Land jährlich wieder führende Quoten beim Export von Gütern in die anderen Länder der Welt.

Diese globale Marktposition ist jedoch nicht unangefochten und macht Deutschlands Industrie anfällig für Nachahmer. Auch der Umstand, dass entsprechende Fachkräfte in Deutschland ein höheres Gehaltsniveau bedingen als in anderen Teilen der Welt, kann zumindest in Teilen als Nachteil gewertet werden.

Umso wichtiger ist es für die produzierende Industrie in Deutschland, sich im Rahmen ihrer Möglichkeiten Alleinstellungsmerkmale zu erarbeiten, die sich nicht einfach nachahmen und kopieren lassen oder durch eine günstigere Fertigung auf andere Art und Weise zu einer Konkurrenz führen.

Seit einiger Zeit hat die Industrie hier die enge Verzahnung ihrer Güterproduktion mit Dienstleistungen, auch oft als Services oder Lösungen bezeichnet, als strategische Option erkannt. Der Vorteil einer engen Verflechtung von Produkt und Dienstleistung ist, dass insbesondere im Pre- und Aftersales-Bereich neue Wertschöpfungsmöglichkeiten entstehen, die auf dem einzigartigen Know-How und dem Wissen der anbietenden Unternehmen basieren. Zunehmend wird über solche Lösungen sogar der größere Anteil der Wertschöpfung für das Unternehmen erzielt, so dass die Bedeutung solcher Angebote beständig zunimmt.

Ein bisher in diesem Zusammenhang nur unzureichend adressiertes Problem ist die Frage, wie diese neuartigen Lösungsangebote in ihrer Kombination aus Güterprodukt und Dienstleistung auf der Seite der Unternehmen verwaltet und gestaltet werden können. Während für das Güterprodukt umfangreiche technologische Unterstützungsmöglichkeiten existieren (z.B. CAD-Systeme, ERP-Systeme, PLM-Systeme), um von der Konstruktion bis zur Auslieferung systematisch und planmäßig koordinieren und steuern zu können, fehlen diese Ansätze auf der Dienstleistungsseite häufig noch. Insbesondere ist die Verknüpfung von Dienstleistungen und Produktion in den Managementsystemen häufig noch nicht oder unzureichend abgebildet.

Der vorliegenden Band dokumentiert die Arbeit des Verbundvorhabens „Produkt-Dienstleistung-Lifecycle-Management“, welches von 2011-2014 den Lehrstuhl für Betriebliche Informationssysteme an der Universität Leipzig zusammen mit Unternehmen ATB Arbeit, Technik und Bildung GmbH, CADsys GmbH, Amtech GmbH und SITEC GmbH aus Chemnitz realisiert und mit Mitteln der Europäischen Union und des Freistaates Sachsen durch das Programm FuE-Projektförderung der SAB

finanziert wurde. Erstmals werden integrierte Konzepte und Werkzeuge für ein Lebenszyklusmanagement in der Verbindung von Produktion und begleitenden Dienstleistungen vorgestellt, die in entsprechenden Use-Cases in der Praxis zur Anwendung geführt werden konnten.

Allen Autoren, Projektpartnern sowie den Fördergeber sei ausdrücklich für die umfangreiche und weitreichende Arbeit gedankt, die mit dem vorliegenden Band in kompakter Form zusammengefasst wird. Dem geneigten Leser sei eine anregende Lektüre gewünscht, die im eigenen Kontext zu einer Nutzung und Anwendung der Ergebnisse führt.

Dr. Kyrill Meyer

Projektleiter der Universität Leipzig im Juli 2014

## Inhaltsverzeichnis

---

Abschnitt 1	Theorie und Konzept	9
Kapitel 1	Das Konzept eines Service-Lifecycle-Managements: Vom Produzenten zum Dienstleister ( <i>Michael Thieme, Kyrill Meyer</i> )	10
Kapitel 2	Dienstleistungslebenszyklen in Produktlebenszyklen ( <i>Christian Zinke</i> )	21
Abschnitt 2	Umsetzung und Werkzeuge	32
Kapitel 3	Review und Bewertung der Eignung von Open-Source-Lösungen als technische Lösung für ein Life-Cycle-Management produktbezogener Dienstleistungen ( <i>Christian Zinke und Florian Golemo</i> )	33
Kapitel 4	Abbildung von Dienstleistungen im Aras Innovator ( <i>Christian Zinke</i> )	65
Kapitel 5	Unterstützung der produktbezogenen Dienstleistung durch die Integration des Service Modeller ( <i>Florian Golemo</i> )	87
Kapitel 6	Kundenintegration und Kollaborationsmöglichkeiten im Aras Innovator ( <i>Christian Zinke, Lars-Peter Meyer</i> )	117
Abschnitt 3	Anwenderfallstudien	122
Kapitel 7	Unterstützung von Pre-Sales Dienstleistungen am Beispiel der Erarbeitung eines technischen Lösungsvorschlag ( <i>Frieder Swoboda, Egbert Mauersberger, Christian Zinke</i> )	123
Kapitel 8	Unterstützung von After-Sales Dienstleistungen am Beispiel des Ersatzteilmanagement ( <i>Frieder Swoboda, Egbert Mauersberger, Christian Zinke</i> )	130
Kapitel 9	Evaluation des PDLM Konzepts und des Unterstützungssystems ( <i>Christian Zinke</i> )	143

### *Einführung (Christian Zinke)*

Produzierende Unternehmen (KMU) im Hochtechnologiebereich in Sachsen hatten in jüngster Zeit das Problem, ihre Dienstleistungen systematisch und effizient weiterzuentwickeln und auf den Stand der Technik zu bringen - d.h. diese zu automatisieren und zu digitalisieren. Noch bevor die ersten Ansätze zu Industrie 4.0 formuliert wurden, entwickelte die Arbeitsgruppe Service Innovation and Management 2011 die Idee eines integrierten Dienstleistungslebenszyklusmanagementansatzes, der mittels moderner kollaborativer IT umgesetzt werden sollte. Ein entsprechender Antrag wurde formuliert und in einem 36-monatigen Projekt realisiert. Während der Laufzeit wurde das existierende PLM Konzept um ein Dienstleistungslebenszyklusmanagement hin zum **Produkt-Dienstleistung-Lebenszyklus-Management (PDLM)** erweitert. Ziel war es, Dienstleistungen in den Produktlebenszyklus und Kunden systematisch in den Dienstleistungsprozess, mittels digitaler Techniken, zu integrieren. Gleichzeitig wurden Referenzdienstleistungen in den Partnerunternehmen aufgenommen und mittels des PDLM Ansatzes weiterentwickelt und digitalisiert. Diese Weiterentwicklung umfasste die prototypische Digitalisierung von Dienstleistungsportfolien (inkl. KPI Zuordnung), Dienstleistungsprozessabläufen sowie zugeordneten Ressourcen. Weitere Teile der prototypischen PDLM Umsetzung sind die Automatisierung von Prozessschritten (wie Benachrichtigungen), digitale Kollaborationsmöglichkeiten für Mitarbeiter, die digitale Dokumentation der zugeordneten Ressourcen (z.B. Produktkomponenten), die Integration von verschiedenen Systemen (PLM, ERP) sowie die Digitalisierung von Kundenschnittstellen (Kundenportal).

Ziel des beschriebenen Vorhabens war kleine und mittlere Unternehmen (KMU) unterschiedlicher Hochtechnologiebereiche mit ihren stark unikal, d.h. kundenindividuellen Leistungen und Produkten bei der Steigerung des Niveaus der technologischen Lösungen entlang des Wertschöpfungsprozesses zu unterstützen. Dabei war das Ziel des Projektes neue Marktsegmente und Marktchancen für dienstleistungsintegrierte Sachgüter aus dem Hochtechnologiebereich zu erschließen.

### **Zielerreichung**

Nach der prototypischen Umsetzung des PDLM Ansatzes sowie Implementation eines IT-Supportsystems, wurden folgenden fundamentale Ziele erreicht: Aus analogen und vereinzelt Prozessmodellen wurden standardisierte Prozessmodelle innerhalb von integrierten Dienstleistungsmodellen (incl. Workflowanbindung) entwickelt. Darüber hinaus wurde aus einem wenig dokumentierten und unsystematischen Dienstleistungsangebot ein detailliertes, digitales Dienstleistungsportfolio mit der Möglichkeit der Zuordnung von KPIs und direkter Implementierung in ein Produkt-Dienstleistungs-Lebenszyklus-Management Tool geschaffen. Weiterhin wurden digitale Verknüpfungsmöglichkeiten von Dienstleistungen und betroffenen Produktkomponenten (inklusive der Integrationsmöglichkeit in ERP Systeme) prototypisch umgesetzt. Die erhobenen

Referenzdienstleistungsprozesse wurden optimiert und einzelne Prozessschritte automatisiert. Zusätzlich wurde mit Hilfe des prototypischen Unterstützungssystems eine digitale Kollaborationsmöglichkeit für Mitarbeiter der Unternehmen und Kunden geschaffen. Diese Kundenintegration wurde durch eine prototypische Kundenplattform ergänzt, auf welcher Kunden Dienstleistungen anstoßen und ständig den Status der geordneten Dienstleistungen einsehen können.

### ***Nutzen***

Die Anwendungspartner haben im Verlauf des Projektes das Potential des digitalen Dienstleistungsmanagements und der Dienstleistungsunterstützung erkannt (bis zu 20% Zeitersparnis durch digitale Dienstleistungsunterstützung). Sie wollen die eingesetzten Prototypen weiter vorantreiben und im operativen Geschäft einsetzen. Die Projektpartner finalisieren zurzeit ein Geschäftsmodell, welches im Anschluss an das Projekt umgesetzt werden soll. Der Projektpartner im IT Bereich (Systemhaus) plant sein Portfolio, um die integrative Dienstleistungsunterstützung im PLM-Bereich zu erweitern.

Das vorgestellte Forschungsprojekt zeigt, welches digitale Innovationspotential im Bereich der produzierenden KMU liegt. Es dient als Vorreiter für den anstehenden Digitalisierungsschub und wird Impulsgeber für den digitalen Wandel in der Industrie- und Dienstleistungsbranche in den nächsten Jahren sein. Das Projekt zeigt wichtige Digitalisierungspotentiale auf und verdeutlicht die Wichtigkeit der Forschung und Entwicklung bei der Digitalisierung von Dienstleistungen, nicht nur in der Industrie.

### ***Das Buch***

Das vorliegende Buch gliedert sich in 3 Abschnitte mit 9 Kapiteln. Im ersten Abschnitt wird die theoretische Basis des Projektes vorgestellt, mit Beiträgen von Michael Thieme und Kyrill Meyer zur Motivation und Reifegrads eines Service-Lifecycle-Managements sowie eines Beitrages von Christian Zinke zu Dienstleistungslebenszyklen in Produktlebenszyklen. Der zweite Abschnitt und größte Teil dieses Buches gliedert sich in vier Beiträge zu Umsetzung und Werkzeugen des im Projekt PDLM umgesetzten Konzepts. Im dritten und letzten Teil dieses Buches sind zwei Anwendungsfälle sowie ein kleiner Ausblick zu finden.



## Abschnitt 1    Theorie und Konzept

---

*„Es gibt nichts Praktischeres als eine gute Theorie.“*

*Immanuel Kant (1724 - 1804), deutscher Philosoph*

## Kapitel 1 Das Konzept eines Service-Lifecycle-Managements: Vom Produzenten zum Dienstleister (*Michael Thieme, Kyrill Meyer*)

### Motivation und Reifegrade des Service-Lifecycle-Managements

---

#### *Motivation eines systematischen Vorgehens*

Die unsystematische Betrachtungsweise von Dienstleistungen verursacht im Unternehmen, wie bereits erwähnt, zahlreiche Schwierigkeiten bei der effizienten Erbringung, gewinnbringenden Vermarktung und Qualitätssicherung der Leistung. Für die effiziente und gewinnbringende Erbringung von Dienstleistungen ist es notwendig im Unternehmen die herrschende produktorientierte Denkweise um eine dienstleistungsorientierte Denkweise zu ergänzen. Hierzu ist es notwendig das reaktive Vorgehen bei der Dienstleistungserbringung in einen systematischen aktiven Managementansatz zu überführen. Der Grad der Dienstleistungsorientierung im Unternehmen bestimmt die Wichtigkeit des Einsatzes dedizierter Managementmethoden für die Dienstleistungserbringung.

Basierend auf den Arbeiten von Beyer (2007) und Becker et al (2010), betrachten wir drei Stufen der Dienstleistungsorientierung in Unternehmen, den reinen Produzenten, den dienstleistenden Produzenten und den produzierenden Dienstleister. Zur Beschreibung der drei Stufen, verbinden wir die sechs aus der wissenschaftlichen Literatur bekannten Reifegradmodelle zu einem holistischen Ansatz (siehe Abbildung 1).

Diese Ansätze werden im Folgenden kurz zusammengefasst: Hildenbrand et al. (2006) verwendet zur Unterscheidung der fünf Stufen der Dienstleistungsorientierung das Grundkonzept der sogenannten Transformationslinie. Die Transformationslinie beschreibt ein Kontinuum zwischen den zwei Extrempunkten des reinen Produktverkäufers und des produzierenden Dienstleisters, welche mit unterschiedlichen Nutzenpotenzialen der industriellen Dienstleistungen verbunden werden. Das Reifegradmodell nach Olivia und Kallenberg (2003) differenziert vier Stufen der Dienstleistungsorientierung anhand der Integration und Orientierung der Dienstleistungen im Leistungsportfolio. Das Reifegradmodell nach Spath und Demuß (2006) differenziert vier Stufen der Dienstleistungsorientierung anhand der Art der angebotenen Dienstleistungen. Das Reifegradmodell nach Müller (1998) differenziert vier Stufen der Dienstleistungsorientierung anhand der Art der Vermarktung angebotener Dienstleistungen. Das Reifegradmodell nach Nägele und Vossen (2006) differenziert fünf Stufen der Dienstleistungsorientierung anhand der Art der Kundenintegration in der Entwicklungsphase von Dienstleistungen.

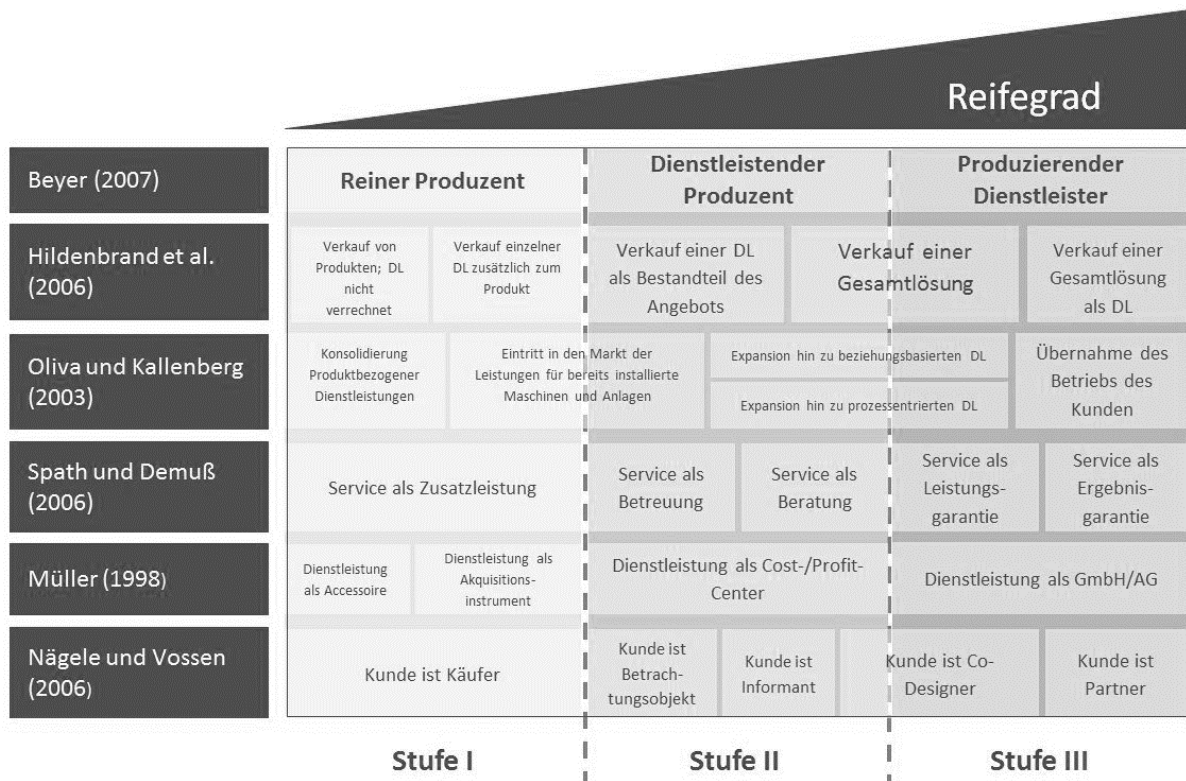


Abbildung 1 Die 3 Stufen der Dienstleistungsorientierung; basierend auf Becker (2010) und Beyer (2007)

Ist die gewinnorientierte Erbringung einer qualitativ hochwertigen Dienstleistung ein gestecktes Unternehmensziel, muss diese Dienstleistung ähnlich dem Produkt systematisch entwickelt, vermarktet, erbracht und kontrolliert werden, um die größtmögliche Effizienz zu erreichen. Mit anderen Worten: je höher die gesetzten Standards an die Dienstleistung, desto höher muss der Reifegrad der Dienstleistungsorientierung im Unternehmen sein, um diese Standards zu erreichen. Hierzu bietet das Service-Lifecycle-Management einen Methoden- und Werkzeugsatz, welcher in 4.2. dargestellt und erläutert wird. Nachfolgend werden die in Abbildung 1 dargestellten drei Stufen des Reifegradmodells kurz vorgestellt.

### Reifegrade der Dienstleistungsorientierung in Unternehmen

#### Reiner Produzent

In der **ersten Stufe** (Reiner Produzent) ist das Unternehmen gekennzeichnet durch:

- ein reaktives Servicemandat in der Strategieformulierung,
- geringe Kundenorientierung und geringe Kundenintegration, Kunde wird primär als Abnehmer der Produkte und Dienstleistungen gesehen; er wird in seiner passiven Rolle als Verbraucher oder Nutznießer betrachtet,

- eine unsystematische Dienstleistungsentwicklung ohne Bezug zum bestehenden Leistungsportfolio,
- zunächst reiner Verkauf von Produkten, Konzentration liegt deutlich auf Produkt und Technologie,
- zunächst wird das Akquisitionspotenzial von Dienstleistungen als Instrument zur Steigerung des Verkaufs der materiellen Produkte benutzt,
- Übergang zur nächsten Entwicklungsstufe durch Konsolidierung der produktbezogenen Dienstleistungen und Angebot als integrierte Nebenleistung (als Add-On auf ein Produkt),
- Vermarktung erfolgt nebenbei durch den Produktvertrieb,
- Dienstleistungen werden nicht verrechnet, Preisbündelung mit Goodwill-Verrechnungsstrategie,
- kaum vorhandene Qualitätssteuerung,
- bei der Organisationsgestaltung ist die Serviceeinheit extern,
- die Organisationsstrukturen sind primär verrichtungsorientiert,
- traditionelle Entlohnungssysteme basierend auf quantitativen Kenngrößen,
- Führungsbewusstsein gekennzeichnet durch Produktorientierung,
- produktbezogene Fachkompetenz.

#### Dienstleistender Produzent

In der **zweiten Stufe** (dienstleistender Produzent) ist das Unternehmen gekennzeichnet durch:

- aktives Servicemandat in der Strategieformulierung,
- im Leistungsportfolio werden die Dienstleistungen prozessunterstützend entwickelt,
- Verkauf von Dienstleistungen ist integraler Bestandteil des Leistungsangebots
- Dienstleistungen werden als integrierte Mehrwertleistung angeboten,
- Kundenbeziehungen werden aus einer transaktionsbasierten in eine beziehungsbasierte Interaktion gewandelt und darauf aufbauend Dienstleistungen entwickelt und angeboten,
- die Bedürfnisse des Kunden werden bei der Entwicklung von Leistungen explizit berücksichtigt, Hersteller versucht aus der Sicht der Kunden herauszufinden, welche Dienstleistungen die Kundenprobleme lösen können,
- der Kunde wird immer mehr als Koproduzent wahrgenommen; der Kunde verliert zunehmend seine passive Rolle und wird als Experte und Informant (z.B. durch Interviews oder Umfragen) für das Serviceangebot zur Lösung seiner Geschäftsprobleme entdeckt,
- Entwicklungspfad hin zum Angebot einer Gesamtlösung aus Produkt und Dienstleistung,

- aktiver Eintritt in den Markt mit Dienstleistungen für bereits installierte Maschinen und Anlagen,
- das Unternehmen ist in der Lage, die Prozesse der Kunden durch professionelle Beratung zu optimieren,
- Identifikation von Gewinnchancen im Markt,
- Lernprozess zum Verkauf, Lieferung und Abrechnung von Dienstleistungen,
- das Dienstleistungsgeschäft wird langfristig als Profit-Center betrieben; Ziel ist die Trennung des Produktions- und Servicegeschäfts,
- Anreizmechanismen für den Verkauf von Dienstleistungen durch den Produktvertrieb.

#### Produzierender Dienstleister

In der **dritten Stufe** (produzierender Dienstleister) ist das Unternehmen gekennzeichnet durch:

- ein expansives Servicemandat in der Strategieformulierung,
- der Kunde wird als Koproduzent angesehen, die einseitige Informationsbeschaffung der vorherigen Stufe entwickelt sich zu einem offenen Dialog. Die Entwicklungspartnerschaften werden langfristig gepflegt; es bestehen dauerhafte und intensive Beziehungen zwischen Unternehmen und Geschäftskunden,
- Systematisierung der Dienstleistungsentwicklung innerhalb des Leistungsportfolios,
- Dienstleistungen sind kundenunterstützend,
- Angebot der Dienstleistung als Komplettlösung mit separater Verrechnung von Sachleistungen,
- Verkauf der Gesamtlösung als Dienstleistung; Angebot innovativer Geschäftsmodelle. In der höchsten Ausprägung übernimmt das Unternehmen den operativen Betrieb des Kunden,
- Steuerung der Qualität durch Service-Level-Management, Einführung eines Key-Account-Managements,
- zunehmende Bedeutung von prozessbezogener Fachkompetenz sowie Interaktions-/Sozialkompetenz und Selbstkompetenz,
- Entlohnungssystem ist flexibel und basiert auf quantitativen sowie qualitativen Kenngrößen,
- Führungsbewusstsein und Organisationsgestaltung gekennzeichnet durch Dienstleistungsorientierung (Service Business Unit),
- Prozessgestaltung im Spannungsfeld von Komplexität und Kundennähe,
- Die betrachteten Nutzenpotenziale sind: Akquisitionspotenzial, Differenzierungspotenzial, Diffusionspotenzial, Ertragspotenzial, Kundenbindungspotenzial, Diversifikationspotenzial, Informationspotenzial, Imagepotenzial, Beschäftigungspotenzial,
- rechtlich selbständige Service-Firma, die die Dienstleistungen als ihr Kerngeschäft bezeichnet.

## Methoden und Werkzeuge eines Service-Lifecycle-Managements

---

Das vorgestellte Stufenmodell zeigt, dass die Anwendung eines Service-Lifecycle-Managements (SLM) mit wachsendem Einsatz von verschiedenen Managementmethoden und der dahinterliegenden informationstechnischen Systeme sinnvoll ist. Dies bedeutet, dass mit steigender Zahl der angewandten Werkzeuge und Methoden, die Notwendigkeit einer ganzheitlichen Sicht – wie sie ein Service-Lifecycle-Management bietet – steigt. Diese Notwendigkeit entspringt der Einsicht, dass auf diesem Wege ein optimales Zusammenspiel aller Managementelemente gewährleistet werden kann. Ziel ist es, die Daten zentral verfügbar zu machen, so dass eine konsistente Verwendung aller Systemelemente ohne Medien- bzw. Technologiebrüche möglich wird und somit eine händische Anpassung vermieden werden kann.

Das SLM setzt sich – analog zum Produkt-Lebenszyklus-Management (PLM) – aus verschiedenen Managementmethoden zusammen, die durch technische Werkzeuge unterstützt und in einer integrierten Entwicklungsumgebung zusammengeführt werden können. Wie auch bei Produkten müssen für Dienstleistungen Ressourcen geplant und Prozesse optimiert werden. Die Arten und Eigenschaften der Ressourcen und Prozesse unterscheiden sich jedoch – teils erheblich - von denen eines Produktes. Dienstleistungen haben besondere Maßgaben, zum Beispiel die nicht vorhandene Lagerfähigkeit oder die Einbeziehung von personenbezogenem Wissen in den Dienstleistungsprozess, welche systematisch einbezogen werden müssen, wenn es um die Identifizierung von – technischen – Werkzeugen geht. Weitere Eigenschaften von Dienstleistungen, die beachtet werden müssen, sind: die Integration des externen Faktors; die dauerhafte Leistungsbereitschaft; der fehlende Patentschutz; der menschliche Faktor im Allgemeinen (DIW 1995; Bullinger und Scheer 2006). Aus diesem Grund werden speziell für produktbezogene Dienstleistungen Methoden und Konzepte herausgearbeitet. Auf dieser Basis erfolgt ein Abgleich der Anforderungen mit bestehenden IT-Werkzeugen für ein Service-Lifecycle-Management sowie eine Identifikation möglicher bestehender Lücken. Beispiele und Anwendungen für SLM Ansätze sind in Meier (2012) sowie Zinke (2013) zu finden.

Im Folgenden werden die im Projekt identifizierten SLM Methoden systematisiert und die theoretischen Methoden und möglichen Werkzeuge eines SLM im PDLM Kontext vorgestellt.

- Servicestrategie- und Dienstleistungsportfolio-Management

„Unter einer Dienstleistungsstrategie soll ein bedingter, langfristiger, globaler Verhaltensplan zur Erreichung der Unternehmens- und Marketingziele eines Dienstleistungsunternehmens verstanden werden.“ (Bruhn und Meffert 2001) Sie „bestimmt die Richtung, das Ausmaß und die Struktur der Entwicklung eines Unternehmens zum Dienstleister“ (Produktbegleitende Dienstleistungen 2002). Die Dienstleistungsstrategie und das Unternehmensleitbild stehen

damit in engem Zusammenhang. Die jeweilige Ausrichtung der Dienstleistungsstrategie sind sowohl auf den Kunden gerichtet, als auch auf die Aufbau- und Ablauforganisation. Darüber hinaus steht sie auch in engem Zusammenhang zum Portfoliomanagement.

Unter Portfoliomanagement ist die aktive Planung, Realisierung, Kontrolle und Veränderung eines Portfolios zu verstehen. Portfolio ist gleichbedeutend mit einer Sammlung von Objekten eines bestimmten Typs – in diesem Falle eine Bestandsaufnahme von Dienstleistungen, die ein ganzheitliches, aufeinander abgestimmtes Angebot von Dienstleistungen abbildet.

Weiterhin ist ein Dienstleistungsportfolio-Management sehr nützlich, um die Leistungen intern sowie extern zu definieren und zu organisieren. Es liefert auf diese Weise Informationen über die Ergebnisebene der Dienstleistung. Für eine solche Managementmethode kann das Werkzeug des Leistungsbaums verwendet werden, um einen Überblick über die einzelnen Leistungen und ihre hierarchische Ordnung zu bekommen.

- **Qualitätsmanagement und Service-Level-Management**

Eine wesentliche Methode für das Qualitätsmanagement speziell für Dienstleistungen ist das Service-Level-Management (SLM). Ursprünglich ist SLM ein Ansatz aus der Informationstechnik, welcher innerhalb der Service Science auf die Anwendung im Bereich der nicht technischen Dienstleistungen überführt und etabliert wurde. „Mit Hilfe des Service Level Managements (SLM) werden mit Kunden gemeinsam verbindliche Leistungszusagen (sog. Service Level Agreements) vereinbart, gepflegt und kontrolliert.“ (Ellis und Käuferstein 2004) So wird das Marketing und die Kommunikation (mit dem Kunden) des Dienstleistungsunternehmens vereinfacht und effizienter. Dabei ist ein dem Kunden zugesichertes Service Level die Basis für ein internes Service Level Agreement – ein sogenanntes Operational Level Agreement – welches der Absicherung der Erreichung des übergeordneten SLAs dient. „Die Überwachung und Auswertung dieser Operation Level stellt sicher, dass die Prozesse ständig auf ihre Zielerreichung hin, nämlich den optimalen Kundennutzen, beurteilt und ggf. optimiert werden.“ (Ebd.)

Ein Service Level Agreement bietet eine Grundlage für ein systematisches Qualitätsmanagement. Die Qualität der Dienstleistung bemisst sich am Grad der Zielerreichung einer Dienstleistung. So kann im Zuge eines Service Level Agreements ein Servicekatalog entstehen, der die Zielerreichung in objektiver Weise messbar macht. Das Qualitätsmanagement ist bei Dienstleistungen ebenso definiert, wie es bei Produkten der Fall ist. Es erweitern sich nur die Faktoren, die dem Management unterliegen. So spielt bei der Dienstleistungsqualität nicht nur die objektive Erbringung der Leistung eine Rolle, sondern auch die subjektive, also vom

Kunden subjektiv wahrgenommene, Dienstleistungserbringung. Auch diese „weichen“ Faktoren (Kompetenz, Höflichkeit, Sicherheit, materielles Umfeld, etc.) müssen geplant, organisiert und optimiert werden.

- **Prozessmanagement, Prozessmodellierung und Konfigurationsmanagement**

Um eine Dienstleistung effizient zu realisieren, ist es notwendig, die dazugehörigen Prozesse und Prozessabläufe zu managen (Prozessmanagement). Dieses Management umfasst mindestens eine Prozessmodellierung, aber auch Prozessentwicklung, Prozesssimulation u.v.m.. Diese Methoden können technisch unterstützt werden. Entsprechende Prototypen (ServCASE) wurden entwickelt, haben aber bisher keine Verwendung in der Praxis erfahren. Besonderheiten einer Prozessmodellierung von Dienstleistungen sind zum Beispiel Kundenschnittstellen. Ein anderes Beispiel wäre eine umfangreiche Standardisierung und Hierarchisierung von Fehlerbehandlungen (Bsp.: Third-Level-Support), deren Prozesse mit umfangreicher Rechte-, Pflichte- und Aufgabenverteilung geplant werden müssen.

- **Ressourcenmanagement**

Weiterhin ist für das SLM auch ein Ressourcenmanagement erforderlich, welches u.a. Technik, Personal (in Rollen) und Materialien beinhaltet. Um eine bestimmte Dienstleistung durchführen zu können, müssen bestimmte Ressourcen zur Verfügung stehen, d.h. ein 24h Wartungsservice muss entsprechend so geplant werden, dass dieser Erbringungszeitraum zu jeder Zeit gewährleistet ist. Dies bedeutet nicht nur, dass das notwendige Personal vorhanden ist, sondern darüber hinaus auch, ob die benötigten Informationen entsprechend schnell zugänglich sind. Solche Vorgänge werden klassischerweise von einem ERP unterstützt. Bei Dienstleistungen müssen, im Unterschied zum Produkt, auch nicht-materielle Ressourcen mitgedacht werden, wie zum Beispiel Wissensressourcen. Dieses Beispiel veranschaulicht auch die Verknüpfungen zu anderen Methoden. So fällt eine vorher vereinbarte, pünktliche Umsetzung der Dienstleistung in das Service-Level-Management und die Zugänglichkeit von Informationen in das – nachfolgende – Lieferkettenmanagement.

- **Lieferkettenmanagement**

Ebenso ist das Lieferkettenmanagement ein Teil eines SLM-Konzeptes und steht in einem direkten Zusammenhang mit dem Ressourcenmanagement. So wird eine Dienstleistung durch ein gutes Lieferkettensystem unterstützt, insofern vielleicht bestimmte Lösungsmodule für ein vorliegendes Problem von unterschiedlichen Partner zeitgleich geliefert werden und nicht nacheinander. Die technischen Lösungen für diese Methode sind SCM-Systeme.

- **Kundenmanagement**



Besonders herauszustellen ist das Kundenmanagement, denn der Kunde spielt im SLM mit steigender Reife der Dienstleistung während des gesamten Lebenszyklus eine zunehmend aktivere Rolle. Die Entwicklung der Sichtweise von passiven Kunden hin zum Kunden als Koproduzent (Prosumer) stellt enorme Ansprüche an die technischen Lösungen eines CRM und dessen Flexibilität. Der Anspruch besteht darin den Kunden in den Prozess der Dienstleistung und der Dienstleistungserbringung systematisch mit einzubinden. Es muss also immer wieder Schnittstellen geben, die im Entwicklungsprozess bedacht werden müssen. Es ist also mehr als ein Betreuungs- und Anforderungsmanagement; es ist ein Interaktionsmodell, welches organisiert und gesteuert werden muss.

Diese Ausarbeitungen können wie folgt dargestellt werden:

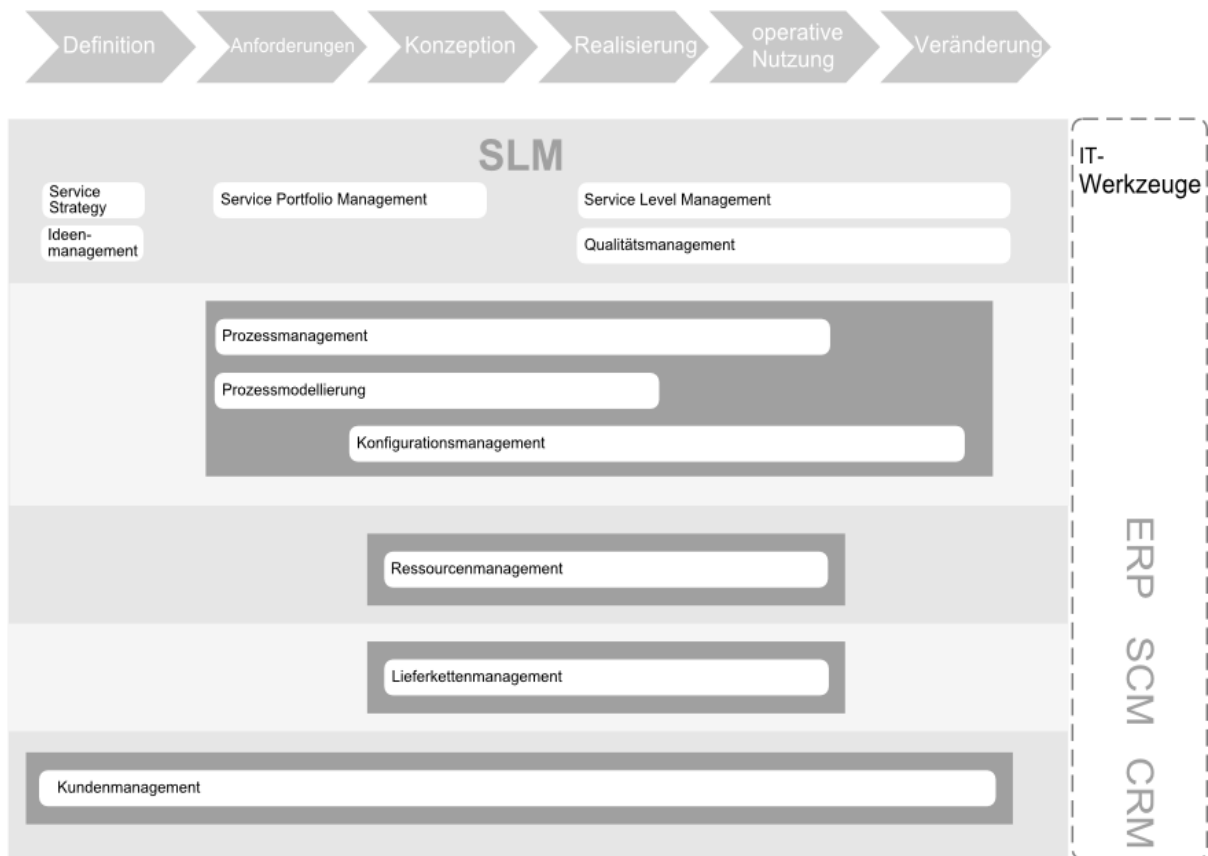


Abbildung 2 SLM Methoden und Werkzeuge

## Literatur

---

DIW (Innovationen im Dienstleistungssektor, 1995): Innovationen im Dienstleistungssektor, in: Deutsches Institut für Wirtschaftsforschung (DIW), Wochenbericht, Vol. 65 (1998), Nr. 29, S. 519-526. (1995).

Produktbegleitende Dienstleistungen. Konzepte und Beispiele erfolgreicher Strategieentwicklung (2002). Berlin: Springer.

Becker, Jörg; Knackstedt, Ralf; Pöppelbuß, Jens (2010): Vergleich von Reifegradmodellen für die hybride Wertschöpfung und Entwicklungsperspektiven. MKWI 2010 – Integration von Produkt und Dienstleistung - Hybride Wertschöpfung. Göttingen: GOEDOC - Dokumentenserver der Georg-August-Universität Göttingen. Online verfügbar unter [http://webdoc.sub.gwdg.de/univerlag/2010/mkwi/03\\_anwendungen/integration\\_von\\_produkt\\_und\\_dienstleistung\\_-\\_hybride\\_wertschoepfung/09\\_vergleich\\_von\\_reifegradmodellen.pdf](http://webdoc.sub.gwdg.de/univerlag/2010/mkwi/03_anwendungen/integration_von_produkt_und_dienstleistung_-_hybride_wertschoepfung/09_vergleich_von_reifegradmodellen.pdf).

Becker, Michael; Klingner, Stephan (2012): Formale Modellierung von Komponenten und Abhängigkeiten zur Konfiguration von Product-Service-Systems. In: Dienstleistungsmodellierung 2012.

Beyer, Mark (2007): Servicediversifikation in Industrieunternehmen - Kompetenztheoretische Untersuchung der Determinanten nachhaltiger Wettbewerbsvorteile Gabler. Wiesbaden.

Brocke, Jan vom; Rosemann, Michael (2010): Handbook on business process management. Berlin ;, London: Springer.

Bruhn, Manfred; Meffert, Heribert (2001): Handbuch Dienstleistungsmanagement. Von der strategischen Konzeption zur praktischen Umsetzung. 2. Aufl. Wiesbaden: Gabler.

Bullinger, Hans-Jörg; Scheer, August-Wilhelm (Hg.) (2006): Service Engineering: Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen. Berlin/Heidelberg: Springer-Verlag.

Eigner, Martin; Stelzer, Ralf (2009): Product Lifecycle Management. Berlin Heidelberg: Springer-Verlag.

Ellis, Avy; Kaufenstein, Michael (2004): Dienstleistungsmanagement. Erfolgreicher Einsatz von prozessorientiertem Service Level Management. Berlin, Heidelberg, New York, Hongkong, London, Mailand, Paris, Tokio: Springer.

Engelfried, Justus (2011): Nachhaltiges Umweltmanagement. 2. Aufl. München [u.a.]: Oldenbourg.

Fischermanns, Guido (2008): Praxishandbuch Prozessmanagement. 7. Aufl. Giessen [i.e. ] Wettenberg: Schmidt.

Hildenbrand, Katharina; Gebauer, Heiko; Fleisch, Elgar (2006): Strategische Ausrichtung des Servicegeschäfts in produzierenden Unternehmen. In: Karim Barkawi, Andreas Baader und Sven Montanus (Hg.): Erfolgreich mit After Sales Services - Geschäftsstrategien für Servicemanagement und Ersatzteillogistik. Berlin Heidelberg, S. 73–94.

Horst Geschka, Martina Schwarz-Geschka (2007): Management von Innovationsideen. In: Peter Gentsch Edelbert Dold (Hg.): Innovation möglich machen. Düsseldorf, S. 147–169.

Kamiske, Gerd F. (2008): Qualitätsmanagement. Eine multimediale Einführung mit einer CD-ROM "Lernprogramm Qualitätsmanagement". 4. Aufl. München: Fachbuchverl. Leipzig.

Meyer, Kyrill; Thieme, Michael; Meyer, Lars-Peter; Zinke, Christian (2012): Computer Aided Lifecycle Management for Product-Related Services. In: Services and Economic Development: Local and Global Challenges. ASE.

Müller, Roland (1998): Kommerzialisierung industrieller Dienstleistungen. Schesslitz: Rosch-Buch.

Müller-Prothmann, Tobias; Dörr, Nora (2009): Innovationsmanagement. Strategien, Methoden und Werkzeuge für systematische Innovationsprozesse. München: Hanser.

Nägele, Rainer; Vossen, Ilga (2006): Erfolgsfaktor kundenorientiertes Service Engineering — Fallstudienenergebnisse zum Tertiärisierungsprozess und zur Integration des Kunden in die Dienstleistungsentwicklung. In: Kristof Schneider, Hans-Jörg Bullinger und August-Wilhelm Scheer (Hg.): Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Berlin Heidelberg (2. Aufl).

Neckel, Hartmut (2004): Modelle des Ideenmanagements. Intuition und Kreativität unternehmerisch nutzen. Stuttgart: Klett-Cotta.

NIST/ECMA (1993): Reference Model for Frameworks of Software Engineering Environments: ECMA Technical Report TR/55, NIST Special Publication 500-211, NIST ISEE Working Group und ECMA TC33 Task Group on the Reference Model.

Oliva, Rogelio; Kallenberg, Robert (2003): Managing the transition from products to services. In: *International Journal of Service Industry Management* 14 (2), S. 160–172.

Scheer, August-Wilhelm (2002): ARIS - vom Geschäftsprozess zum Anwendungssystem. 4. Aufl. Berlin [u.a.]: Springer.

Schuh, Günther; Klappert, Sascha (2010): Produktion und Management 2. Technologiemanagement. 2. Aufl. Berlin: Springer Berlin.

Sendler, Ulrich (2009): Das PLM-Kompodium - Referenzbuch des Produkt-Lebenszyklus-Managements. Berlin Heidelberg: Springer-Verlag.

Spath, Dieter; Demuß, Lutz (2006): Entwicklung hybrider Produkte - Gestaltung materieller und immaterieller Leistungsbündel. In: Kristof Schneider, Hans-Jörg Bullinger und August-Wilhelm Scheer (Hg.): Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Berlin Heidelberg (2. Aufl), S. 463–502.

Zinke, Christian; Meyer, Lars-Peter; Meyer, Kyrill (2013): Modeling Service Life Cycles within Product Life Cycles. In: Camarinha-Matos Luis Scherer M.; Raimar J. (Ed.): Collaborative Systems for Reindustrialization, PRO-VE 2013. Springer Verlag.

## Kapitel 2 Dienstleistungslebenszyklen in Produktlebenszyklen (*Christian Zinke*)

### Einleitung<sup>1</sup>

---

Um Stagnation im güterproduzierenden Industriesektor, welcher in vielen Fällen durch niedrige Preisspannen und einen hohen Wettbewerbsdruck charakterisiert ist, zu vermeiden, müssen erfolgreiche Produzenten industrieller Güter nach neuen Verkaufswegen Ausschau halten. Eine Möglichkeit ist meist das Angebot von Hybridlösungen, bestehend aus der Integration von Dienstleistungen und materiellen Gütern (Sachleistungen) (Becker et al. 2010; Böhmman et al. 2007).

Um Hybridlösungen anzubieten, müssen Produzenten industrieller Güter einen größeren Fokus auf das Erweitern ihres Produktportfolios durch Dienstleistungen und die Verknüpfung selbiger in einer angemessenen Weise legen, um sich von der Konkurrenz abzuheben, die Kundenzufriedenheit zu steigern und finanzielle Vorteile zu erlangen (Becker et al. 2010; Kersten et al. 2006).

Dienstleistungen, die ein materielles Produkt ergänzen, wirken sich auf verschiedene Bereiche des Produktlebenszyklus oder den gesamten Lebenszyklus des materiellen Produktes aus. Zum Beispiel kombiniert ein Produzent im Spezialmaschinenbau seine Produkte mit beratenden Dienstleistungen, Schulungen, Instandhaltungen sowie Reparaturen / Nachrüstungen und dem Recycling. Diese Dienstleistungen müssen während des gesamten Produktlebenszyklus in Betracht gezogen werden. Beispielsweise muss in der Planungsphase die Wartbarkeit des Produktes berücksichtigt werden (bspw. durch Sensorik und LAN Schnittstellen), um den Wartungs- und Instandhaltungsdienst möglichst effizient gestalten zu können (bspw. Fernwartung durch Remote Services).

Wenn ein Unternehmen produktnahe Dienstleistungen und die dadurch unterstützten Produkte erfolgreich verwalten möchte, ist es notwendig zu verstehen, wie Dienstleistungen und materielle Güter unter Berücksichtigung ihrer Lebenszyklen kombiniert werden können. Basierend auf einer theoretischen Beschreibung wird hier ein Weg präsentiert, Dienstleistungen im Kontext materieller Güter und ihre Lebenszyklen zu modellieren. Ziel ist die Beschreibung der Verbindungen zwischen materiellen Gütern und Dienstleistungen über ihre Lebenszyklen hinaus in einer theoretischen, allgemeinen Art und Weise.

Unsere Arbeit basiert auf einer Fallstudie aus dem Bereich des Maschinenbaus, insbesondere im Bereich der Spezialmaschinen. Die Reichweite unserer Arbeit ist auf industrielle Güter und Dienstleistungen

---

<sup>1</sup> Dieser Abschnitt ist angelehnt an Modeling Service Life Cycles within Product Life Cycles von Christian Zinke, Lars-Peter Meyer und Kyrill Meyer erschienen in: Collaborative Systems for Reindustrialization (Editors: Luis M. Camarinha-Matos und Raimar J. Scherer), PRO-VE 2013. Springer Verlag. 2013

begrenzt, die einen hohen Individualisierungsgrad, einen ausgeprägten Informationsaustausch mit dem Kunden und geringe Produktionsmengen sowie High-Tech Know-how aufweisen.

### Problem- und Zielstellung

---

*“The traditional boundary between manufacturing and services is fast becoming obsolete. Manufacturing has traditionally meant the production of tangible goods, but for today’s customers it is the bundling together of the tangible object with an array of intangible services that makes for the most desirable, ‘service-enhanced product’”*<sup>2</sup> (Lester 1998). Zusammenfassend gilt, dass obwohl materielle Güter und immaterielle Dienstleistungen zusammen gehandhabt werden müssen, um erfolgreich zu sein, noch viele Unternehmen diese Dienste vernachlässigen und sie nicht in Betracht ziehen, wie dies für materielle Güter geschieht. Dieses Phänomen wird “Servitization” genannt und wurde erstmals von Vandermerwe und Rada (1988) eingeführt und ist immer noch ein wichtiges Thema in der Forschung und der Praxis (Steunebrink 2012; Nelly et al 2010; Baines et al. 2009), auch wenn dessen Existenz nicht so neu ist, wie es scheint (Wildemann 2009). “Servitization” meint “[t]he emergence of product-based services which blur the distinction between manufacturing and traditional service sector activities”<sup>3</sup> (White et al. 1999).

Andererseits gibt es verschiedene Werkzeuge und Methoden, um Produkte und Dienstleistungen während des gesamten Lebenszyklus zu handhaben (Sendler 2009). Zur Unterstützung solcher Methoden werden häufig Informationssysteme wie PLM/PDM genutzt und implementiert. Diese PLM/PDM Systeme sind, durch die Unterstützung von Materialstücklisten, traditionellen Versorgungsketten und unterstützenden Werkzeugen, wie zum Beispiel CAD, primär auf materielle Güter ausgerichtet. Diese Informationssystem-Lösungen für materielle Güter sind anspruchsvoll und können Effizienz und Qualität zu erschwinglichen Kosten steigern.

Allerdings ist weitere Forschung im Bereich der Unterstützung (PLM/PDM-Systeme) von produktbezogenen Dienstleistungen notwendig. Dienstleistungen sind nicht nur praktisch vernachlässigt, sondern werden auch in den betroffenen Informationssystemen, die diese unterstützen sollten, nicht berücksichtigt. Dienstleistungswerkzeuge sind oftmals eigenständige Anwendungen ohne Integration in PDM/PLM-Lösungen. Außerdem sollten die Modellierung von PDM/PLM-Systemen überdacht werden, so dass diese ebenso für produktbezogene Dienstleistungen wie materielle Güter

---

<sup>2</sup> Frei übersetzt: Die traditionelle Grenze zwischen Produktion und Dienstleistung verschwinden zunehmend. Produktion im traditionellen Sinne ist das Herstellen greifbarer Güter, für den heutigen Kunden ist es jedoch die Verknüpfung des materiellen Produktes mit einer Reihe von Dienstleistungen, die es zu dem wünschenswerten 'Dienstleistungs-erweiterten Produkt' machen.

<sup>3</sup> Frei übersetzt: die Entstehung von produktbezogenen Dienstleistungen, die die Grenzen zwischen den Aktivitäten des Produktionssektors und des traditionellen Dienstleistungssektors verwischen.

nutzbar werden. Dafür müssen die speziellen Eigenschaften von Dienstleistungen (wie Immaterialität, Vergänglichkeit, Untrennbarkeit und Simultanität (Zeithaml 1985)) beachtet werden.

Bis hierhin haben wir die Wichtigkeit produktbezogener Dienstleistungen für gut produzierende Unternehmen erklärt. Zusammenfassend: Dienstleistungen werden zu einem wichtigen Teil vieler Unternehmen und unser Ziel ist es, Probleme mit produktbezogenen Dienstleistungen zu verringern.

Um diese Probleme zu verringern benötigen wir:

- einen theoretischen Rahmen für die Verknüpfung zwischen Produkten und produktbezogenen Dienstleistungen, die deren Lebenszyklen berücksichtigt.
- ein Modell für diesen theoretischen Rahmen.

Sowohl die theoretische Herangehensweise als auch das resultierende Modell werden essentielle Bestandteile dieses Kapitels sein. Insbesondere wird auf die Modellierung von Dienstleistungen und deren Lebenszyklen innerhalb eines Produktlebenszyklus fokussiert. Das Modellieren produktbezogener Dienstleistungen hilft dabei Methoden und Werkzeuge in Herangehensweisen, die auf traditionell materiellen Gütern basieren, einzubetten. Auf diese Art und Weise können wir Probleme von Dienstleistungen, wie zum Beispiel Kostenschätzungen, Qualitätskontrollen und interne sowie externe Kommunikation, mit dem Kunden lösen.

Um unsere Ziele festzuhalten, formulieren wir Forschungsfragen. Die folgenden Forschungsfragen werden im Rahmen dieser Arbeit beantwortet:

- Wie können produktbezogene Dienstleistungen in die Herangehensweisen des Produktlebenszyklus integriert werden?
- Wie können produktbezogene Dienstleistungen für PLM/PDM-Systeme modelliert werden?

### Produkt-Dienstleistungs-Lebenszyklus Ansatz

---

Bevor produktbezogene Dienstleistungen auf einer technischen Ebene modelliert werden können, wird ein theoretischer Rahmen für diese Modellierung benötigt. In diesem Kapitel wird dieser theoretische Rahmen als Produkt-Dienstleistungs-Lebenszyklus Ansatz bezeichnet. Produktbezogene Dienstleistungen werden als Dienstleistungen in einem Produktlebenszyklus verstanden, die jedoch auch ihren eigenen Lebenszyklus haben (siehe Abb. 3) (Meyer et al. 2012).

Der Lebenszyklen beider Forschungsgegenstände – des Produktes und der Dienstleistung – enthalten verschiedene Phasen. Für “Produkte” werden die folgenden Phasen klassifiziert: Anforderungen, Produktplanung, Entwicklung, Prozessplanung, Produktion, Betrieb und Wiederverwertung. Für “Dienstleistung” werden die folgenden Phasen definiert: Definition, Anforderungen, Konzeption, Realisierung, betriebliche Nutzung und Anpassung (DIN 1998). Auf jeder Ebene des

Produktlebenszyklus haben wir verschiedene Dienstleistungen identifiziert, die einen eigenen Lebenszyklus aufweisen. Die verschiedenen Dienstleistungs- und Produktlebenszyklen laufen nicht synchron. Das Verfahren einen Kunden für die Produktplanung heranzuziehen wird, zum Beispiel, in der Produktplanungsphase realisiert und verwendet. Die Instandhaltung jedoch sollte in der Anforderungs- und Produktplanungsphase definiert und in der Betriebsphase des Produktes ausgeführt werden. Nach dem präsentierten theoretischen Rahmen müssen wir produktbezogene Dienstleistungen in einem PDM/PLM-System modellieren.

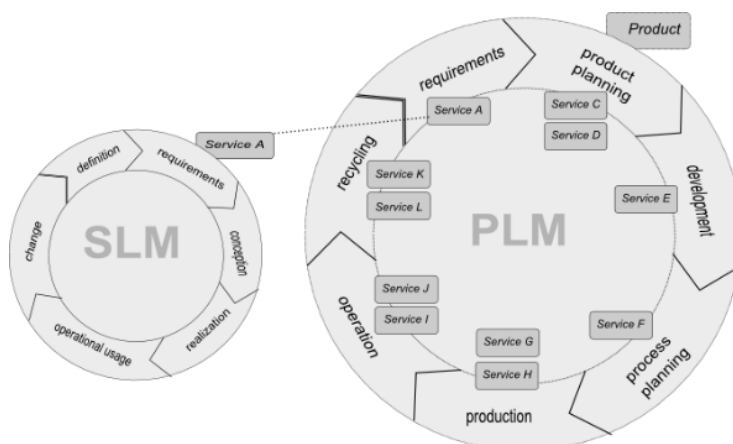


Abbildung 3 Produkt-Dienstleistungs-Lebenszyklen (Zinke et al 2013)

Diese Herangehensweise liegt nah bei anderen wissenschaftlichen Forschungen, wie der Forschung an Produkt-Dienstleistungs-Systemen (PSS) (Sendler 2009) und der domänenspezifischen Forschung an Industriellen Produkt-Dienstleistungs-Systemen (IPS<sup>2</sup>). Sowohl PSS als auch IPS<sup>2</sup> haben einen Lebenszyklus (Sadek et al. 2010; Katja 2013) mit verschiedenen Ebenen auf denen PSS oder IPS<sup>2</sup> Dienste und Produktkomponenten definiert, geplant und umgesetzt werden. Das Hauptziel ist es, das Anbieten von Produkt-Dienstleistungs-Bündeln als kundenspezifische Lösung zu unterstützen. Die Design- und Modellierungsherangehensweise fokussiert sich hauptsächlich auf PSS oder IPS<sup>2</sup> Komponenten (Abramovici 2009) als neue Objekttypen. In diesem Buch und diesem Kapitel haben Produkte und Dienstleistungen unterschiedliche Lebenszyklen, die integriert und verknüpft werden können.

### Model von produktbezogenen Dienstleistungen

Unser Modellierungsansatz basiert auf verschiedenen Fallbeispielen und orientiert sich an den im Projekt gefundenen Referenzdienstleistungen, für welche eine technische Umgebung als Fallanwendung ausgewählt wurde. Für diese Herangehensweise wurde ein technisches Werkzeug (Aras Innovator) ausgewählt, siehe Kapitel 3 Review und Bewertung der Eignung von Open-Source-Lösungen als technische Lösung für ein Life-Cycle-Management produktbezogener Dienstleistungen. Der Aras Innovator ist die Fallanwendung, um zu zeigen, wie Dienstleistungen und materielle Güter modelliert



werden können. Zunächst wird das Produktmodell des Aras Innovator eingeführt. Es folgt die Modellierung von produktbezogenen Dienstleistungen und die Kombination beider Ansätze.

Materielle Güter

Im Aras Innovator sind Produkte – materielle Güter – als hierarchisches Produktmodell dargestellt. Hierbei sind Produkte realisierte Produktmodelle, die wiederum aus Produktkomponenten (“parts”) zusammengesetzt sind. Einzelne Komponenten können verschiedene Komponententypen (z.B. “assembly”) repräsentieren. Verschiedene Komponententypen haben verschiedene Verbindungen zu anderen Elementtypen (z.B. hat der Typ “assembly” eine BOM während der Typ “material” keine BOM hat). Das Produktmodell ist komponentenbasiert, da die Produkte aus einzelnen, klar definierten (modularen) Komponenten mit standardisiertem Interface bestehen<sup>4</sup>. Das beschriebene Modell ist in Abb. 4 visualisiert. Dieses Schema kann in einer normierten Form, als UML (Unified Modeling

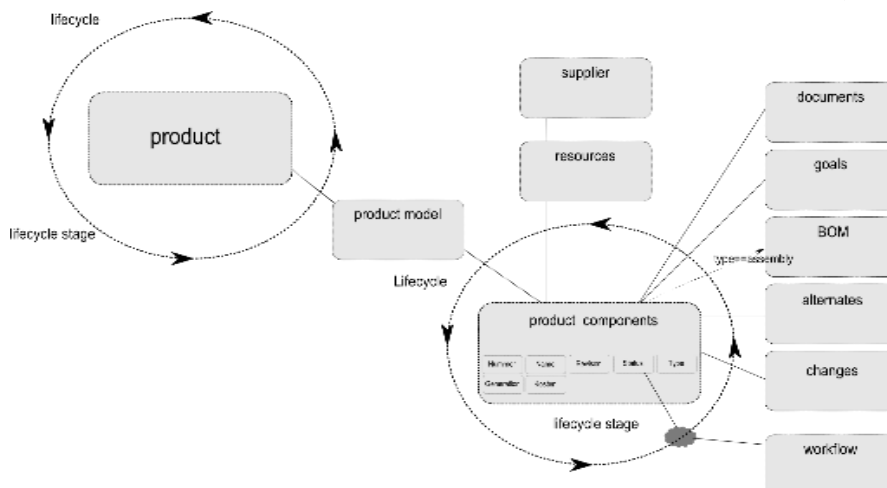


Abbildung 4 Produkt Modell (Zinke et al 2013)

Language) dargestellt werden (Abb. 5).

<sup>4</sup> Siehe Aras Corporation Self Help Guide, <http://www.aras.com/university/SelfHelpGuides/Print%20of%20Online%20Help%20-%20Aras%20Product%20Engineering.pdf>

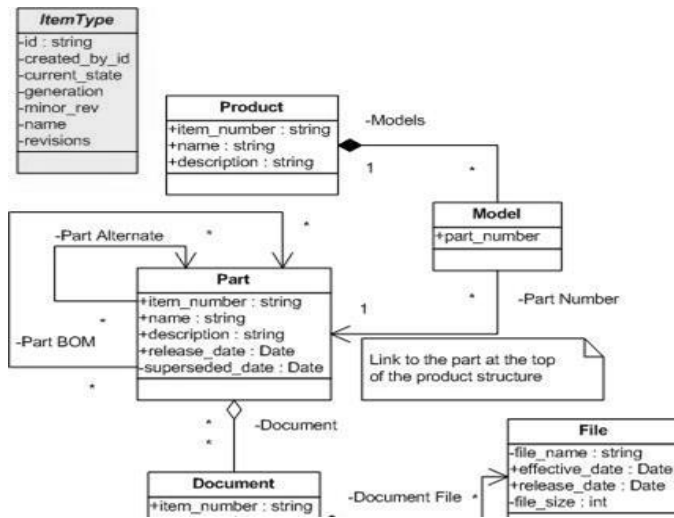


Abbildung 5 Produkt-Teile UML (Zinke et al 2013)

An dieser Stelle sehen wir, dass das Produkt aus Modellen, die jeweils genau eine Einheit darstellen, besteht. Diese Einheiten können einzelne Teile, komplexe Teile oder voll funktionierende Teile sein. Deshalb können verschiedene Teile in verschiedenen Produkten wiederverwendet werden, was zu einem modularen Produktmodell führt.

### Immaterielle Dienstleistungen

Die beschriebene Architektur des Produktmodells ist potenziell mit Dienstleistungsmodellen kombinierbar. Dafür wird ein Modell für Dienstleistungen benötigt (siehe Abb. 4).

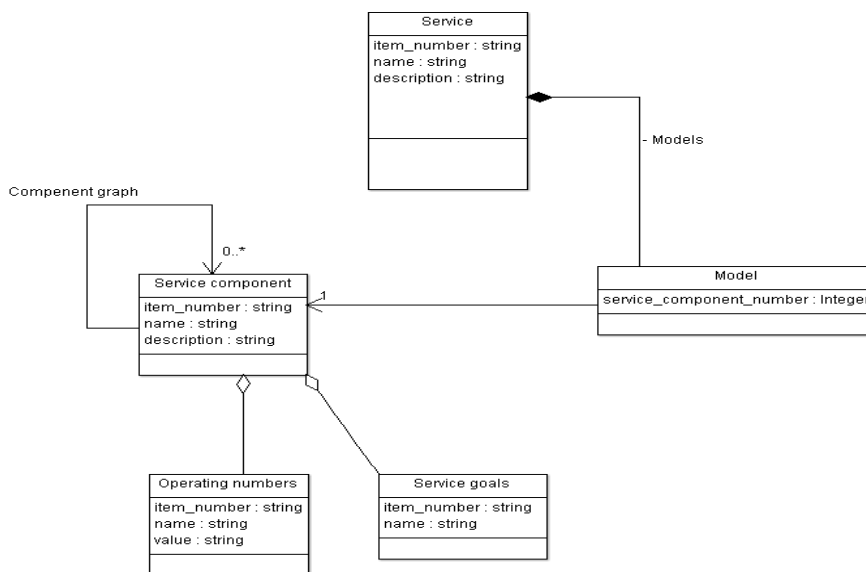


Abbildung 6 Dienstleistungskomponenten (Zinke et al 2013)

Die Modularität garantiert die Möglichkeit, Dienstleistungen mit Produkten zu verbinden. Dienstleistungen im Allgemeinen können modularisiert (Böttcher und Klingner 2011) werden, so dass Dienstleistungen als Komponenten repräsentiert werden können.

Es ist ersichtlich, dass die Entität ‚Dienstleistungen‘ Dienstleistungskomponenten beinhaltet und Dienstleistungskomponenten mit sich selbst verbunden werden können. Daher wird eine Baumstruktur aus Dienstleistungskomponenten entwickelt (siehe Abb. 6).

Das führt zu einem vereinfachten Modell von Dienstleistungen, veranschaulicht jedoch die Modularisierung, wodurch es ermöglicht wird, Dienstleistungsmodule in die PLM/PDM-Lösung, also den Aras Innovator, zu integrieren. Dienstleistungen zu integrieren erlaubt uns andere Komponenten des Aras Innovator, wie das Lebenszyklus-Konzept, das Dokumentenmanagement und die Workflow-Engine, zu nutzen.

Produktbezogene Dienstleistungen

Um die PLM-Lösung für hybride Produkte nutzen zu können, müssen beide Modelle integriert werden. Eine solche Integration wurde innerhalb des ARAS Innovator PLM Systems realisiert. Zunächst wird jedoch die Integration der Modelle visualisiert (siehe Abb. 7).

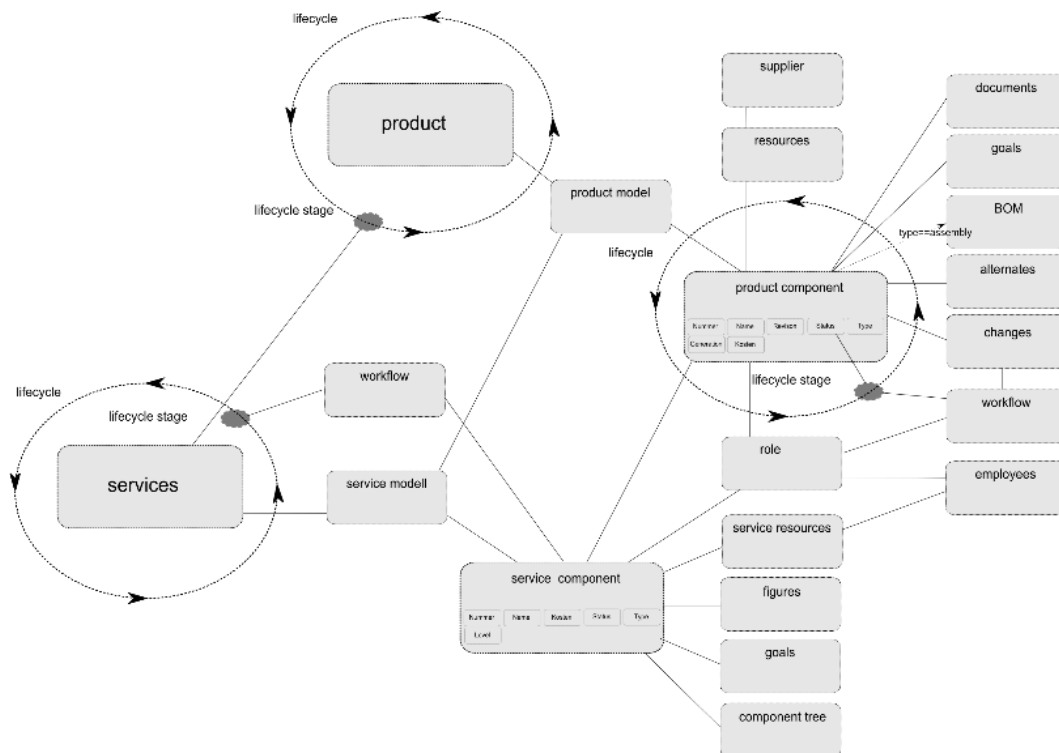


Abbildung 7 Modell von integrierten Dienstleistungen und materiellen Produkten (Zinke et al 2013)

In Anlehnung an unseren theoretischen Rahmen werden produktbezogene Dienstleistungen als Dienstleistungen in den einzelnen Lebenszyklus-Ebenen des Produktes modelliert. Diese Dienstleistungen beinhalten Komponenten, die mit Produktkomponenten, Dienstleistungszielen und so weiter verknüpft werden können (siehe Abb. 7). Des Weiteren kann dies erneut auf eine formalere Art mit UML, wie in Abb. 8, dargestellt werden.

Als Ergebnis wurde exemplarisch gezeigt, wie das Dienstleistungsmodell, unter Berücksichtigung der verschiedenen Lebenszyklen, in ein existierendes PLM/PDM-System integriert werden kann. Nur ein kleiner Teil der Integration wurde beschrieben, um einen Fokus auf die Verknüpfung von Dienstleistungen, Dienstleistungskomponenten, Produkten und Produktkomponenten zu legen. Dies wurde wie beschrieben durchgeführt, da es sich um die Basisanforderungen für ein komplexeres Modell von um Dienstleistungen erweiterte Produkte handelt. Basierend auf unserem Anwendungsfall haben wir eine mögliche Herangehensweise für die Integration von Dienstleistungen in einen Produktlebenszyklus gezeigt. Des Weiteren haben wir produktbezogene Dienstleistungen modelliert. Insgesamt konnten wir Antworten auf unsere Forschungsfragen mit Hilfe der ausgewählten Fallstudienherangehensweise finden.

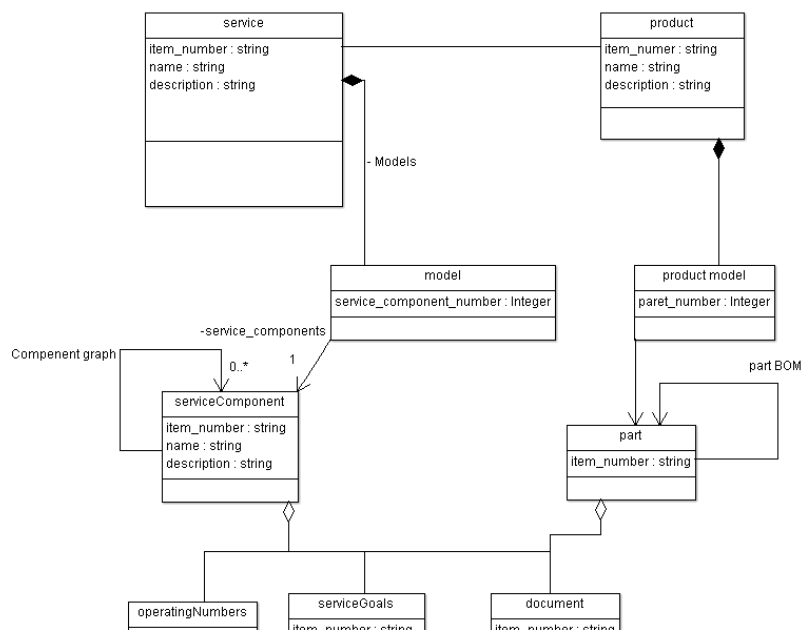


Abbildung 8 UML produktbezogene Dienstleistungen (Zinke et al 2013)

### Perspektiven und zukünftige Arbeiten

Das vorgestellte Modell hilft im Allgemeinen Dienstleistungen in PDM/PLM-Systeme zu integrieren und zu handhaben. Diese Integration ist ein entscheidender Schritt, um Computer-gestützte Hilfe für eine „Servitization“ in Richtung der Produkt-Dienstleistungs-Systeme (PSS) zu ermöglichen. Mit gemäß des Entwicklungsmodells erweiterten PDM/PLM-Systemen stehen Betrieben IT-Tools zum

Erstellen und Handhaben der Dienstleistungs- und Produktportfolios für interne und externe (Kunden-) Nutzung zur Verfügung.

Wir haben zwei weitere Schritte identifiziert, die notwendig sind, um produktbezogene Dienstleistungen zu modellieren. Einerseits muss unsere Arbeit allgemeiner gestaltet werden, um unseren theoretischen Rahmen und das Modell für weitere Anwendungsbereiche und Applikationen nutzen zu können. Andererseits arbeiten wir an einer gesamtheitlichen, praktischen Integration in einen Prototyp, um den Mehrwert für Betriebe mit dienstleistungsunterstützten Produkten zu evaluieren. Hierfür integrieren wir aktuell Dienstleistungs-Tools, wie den "Service-Modeller" (Böttcher und Klingner 2011), in den Aras Innovator (siehe Kapitel 5 Unterstützung produktbezogener Dienstleistungen durch die Integration des Service Modeller). Des Weiteren werden wir die dargestellte Implementierung für eine Prototypen einer Ersatzteillieferungs-Dienstleistung im High-Tech Maschinenbereich nutzen (siehe Kapitel 8 Unterstützung von After-Sales Dienstleistungen am Beispiel des Ersatzteilmanagement).

In der Zukunft wird sich der Schwerpunkt auf materielle Güter in einer allgemeineren Weise ändern, und die unterstützenden Anwendungen werden Dienstleistungs- und Produktlebenszyklusmodelle einschließen. Wir sind zuversichtlich, dass diese Arbeit etwas zu dieser Entwicklung beisteuern kann.

## Quellenverzeichnis

---

Abramovici, M., Neubach, M.; Schulze, M.; Spura, C.: Metadata Reference Model for IPS<sup>2</sup> Lifecycle Management. In: Proceedings of the 1st CIRP IPS<sup>2</sup> Conference. Cranfield, UK (2009)

Aras Corporation Self Help Guide, <http://www.aras.com/university/SelfHelpGuides/Print%20of%20Online%20Help%20-%20Aras%20Product%20Engineering.pdf> (letzter Zugriff 01.03.2014)

Aras Corporation Discussion File, [http://www.aras.com/Community/cfs-filesystemfile.ashx/\\_\\_\\_key/CommunityServer.Discussions.Components.Files/7/0361.aras\\_5F00\\_product\\_5F00\\_model.jpg](http://www.aras.com/Community/cfs-filesystemfile.ashx/___key/CommunityServer.Discussions.Components.Files/7/0361.aras_5F00_product_5F00_model.jpg) (letzter Zugriff 01.03.2014)

Aras Corporation Aras Platform, <http://www.aras.com/technology/platform/> (letzter Zugriff 01.03.2014)

Baines, T., Lightfoot, H., Peppard, J., Johnson, M., Tiwari, A., Shehab, E., Swink, M.: Towards an operations strategy for product-centric servitization. International Journal of Operations & Production Management. Vol. 29 Iss: 5. pp.494 – 519 (2009)

Becker J., Beverungen D., Knackstedt R., Müller O., Müller S.: TCO-as-a-Service – Servicebasierte Lebenszyklusrechnung für hybride Leistungsbündel. In Becker J., Knackstedt R., Müller O., Winkelmann, A.: Vertriebsinformationssysteme. Pp 161-174. Springer. Berlin Heidelberg (2010)

Böhmman T., Krcmar H: Hybride Produkte: Merkmale und Herausforderungen. In Bruhn M., Stauss B. (eds.) Wertschöpfungsprozesse bei Dienstleistungen – Forum Dienstleistungsmanagement. Pp 239-255. Wiesbaden (2007)

DIN: Service Engineering: Entwicklungsbegleitende Normung für Dienstleistungen. Berlin (1998)

Katja L.: Wandel des traditionellen Dienstleistungsverständnisses. In: Thomas, O., Nüttgens, M.: Dienstleistungsmodellierung 2012 – Product-Service Systems und Produktivität. Springer Gabler. Wiesbaden (2013)

Kersten W., Zink T., Kern E.-M.: Wertschöpfungsnetzwerke zur Entwicklung und Produktion hybrider Produkte: Ansatzpunkte und Forschungsbedarf, In: Becker T., Gemünden H.G. (eds.): Wertschöpfungsnetzwerke. Festschrift für Bernd Kaluza. pp. 189-291. Berlin (2006)

Klingner S., Böttcher M., Becker M., Döhler A., Schumacher, F.: Managing complex service portfolios: A business case from a full service provider. In: Reser 2011 Proceedings. Hamburg (2011)

Lester, R.: The Productive Edge – How US Industries are Pointing the Way to a New Era of Economic Growth. Norton. New York (1998)

Meyer K., Thieme M., Meyer L.-P., Zinke, Chr.: Computer Aided Lifecycle Management for Product-Related Services. In: Service and Economic Development: Local and Global Challenges. The 22nd RESER International Conference. pp. 17-40. Bukarest. (2012)

Neely, A., Benedittini, O., Visnjic, I: The Servitization of Manufacturing: Further Evidence. EuOMA Conference. Cambridge (2011)

Sadek, T., Köster, M.: Eine modellorientierte Methodik zur Unterstützung der Konzeptentwicklung industrieller Produkt-Service Systeme. In: Thomas, O., Nüttgens, M.: Dienstleistungsmodellierung 2010. Springer-Verlag. Berlin Heidelberg 2010

Sendler, U.: Das PLM-Kompodium - Referenzbuch des Produkt-Lebenszyklus-Managements. Springer-Verlag, Berlin Heidelberg (2009)

Schmenner, R. W: Manufacturing, service, and their integration: some history and theory. International Journal of Operations & Production Management, Vol. 29 Iss: 5 (2009)

Steunebrink, G.G.B.: The servitization of product-oriented companies, <http://purl.utwente.nl/essays/62039> (2012)

Vandermerwe, S., Rada, J.: Servitization of Business: Adding Value by Adding Services. *European Management Journal* 6. pp. 314-324 (1988)

Wildemann, H.: *Produkte und Services entwickeln und managen: Strategien, Konzepte, Methoden*. TCW Transfer-Centrum-Verlag, München (2009)

White, A.L., Stoughton, M., Feng, L.: *Servicizing: The quiet transition to extended product responsibility*. Tellus Institute. Boston (1999)

Zeithaml, V., Parasuraman, A., Berry, L.: "Problems and strategies in services marketing", *Journal of Marketing*, Vol. 49, Spring, pp. 33-46 (1985)

Zinke, Christian, Meyer, Lars-Peter, Meyer, Kyrill: *Modeling Service Life Cycles within Product Life Cycles* In: *Collaborative Systems for Reindustrialization* (Editors: Luis M. Camarinha-Matos und Raimar J. Scherer ), PRO-VE 2013 . Springer Verlag. (2013)

## Abschnitt 2      Umsetzung und Werkzeuge

---

*„Wer gute Arbeit leisten will, schärfe zuerst das Werkzeug.“*

*Spruchwort*



## Kapitel 3 Review und Bewertung der Eignung von Open-Source-Lösungen als technische Lösung für ein Life-Cycle-Management produktbezogener Dienstleistungen (*Christian Zinke und Florian Golemo*)<sup>5</sup>

### Vorwort

---

Das Ziel des vorliegenden Kapitels ist es, verschiedene Open-Source-Lösungen für die technische Lösung für den PDLM Ansatz zu identifizieren und die erfolgversprechendsten Lösungen zu qualifizieren. Anschließend werden diese auf ihre Eignung für eine Softwarelösung, als technische Unterstützung für die verschiedenen Methoden eines Produkt-Dienstleistungs-Lebenszyklus-Management-Ansatzes, hin zu prüfen sein. Es soll bewertet werden, ob die existierenden Open-Source-Modelle für einen effizienten Einsatz im Projekt geeignet sind. Für diesen Zweck werden verschiedene Kriterien herausgearbeitet. Zum einen werden Suchkriterien herausgearbeitet, auf deren Basis eine engere Auswahl von Open-Source-Lösungen stattfindet. Diese Kriterien werden im Zuge der Auswahl verfeinert, um eine aussagekräftige Bewertung der Software-Systeme zu erhalten.

### Praktische Suche

---

Gesucht wurde zunächst mittels einer einfachen Google-Suche, später wurde auch auf Sourceforge.net gesucht. Die zugrundeliegenden Prämissen für die Suche blieben jedoch dieselben. Diese können wie folgt beschrieben werden:

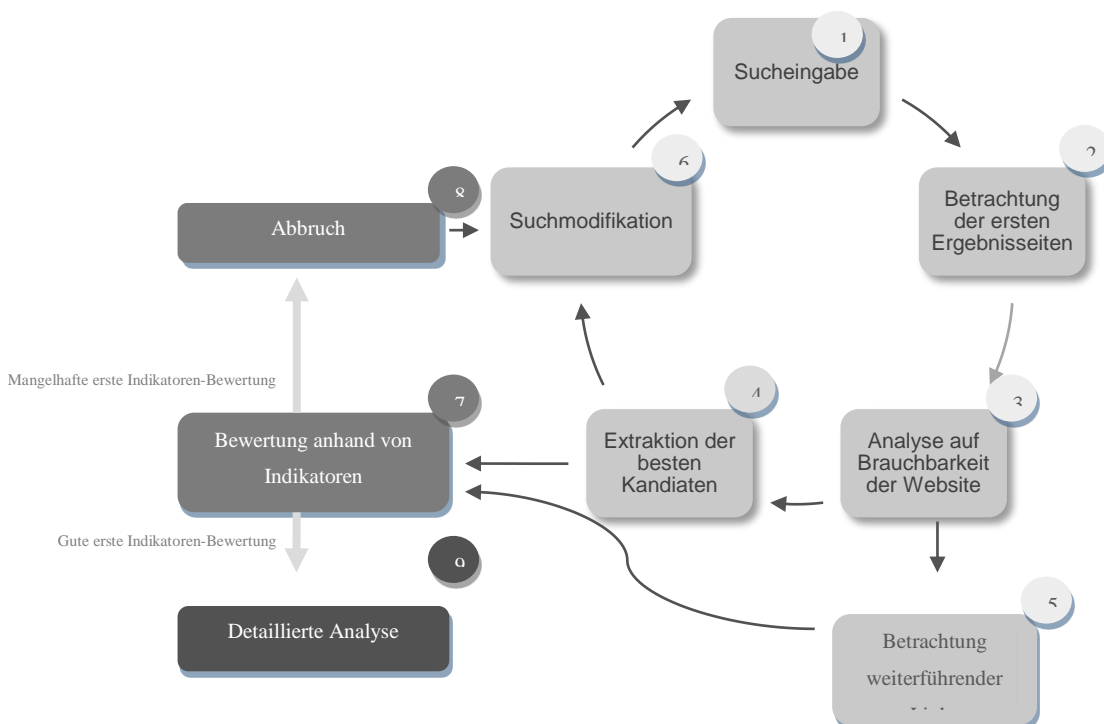
- Die gefundenen Websites benötigten eine gewisse Eignung, um für eine PLM-Softwarelösung anwendbar zu sein, beispielsweise durch ein PLM-Szenario auf der Produktwebsite oder durch Anwendung von Managementansätzen, die ein PLM umfasst (bspw. Dokumentenmanagement). D.h. ein erkennbarer Bezug zu PLM ist Basis für die Auswahl.
- Die gefundene Software muss Open Source sein oder zumindest kostenlos verwendet werden können.
- Auf der Website kann das Produkt ohne oder mit kostenloser Anmeldung direkt heruntergeladen werden.
- Es ist erkennbar, dass wesentliche Komponenten der angebotenen Software im Browser arbeiten.

Das konkrete Vorgehen staffelte sich in die Eingabe der Suche (1), die Sichtung der ersten Ergebnisse (2) und die Analyse auf Brauchbarkeit der Website (3). Es folgt die direkte Extraktion von Kandidaten

---

<sup>5</sup> Die vorgestellten Ergebnisse erscheinen in kürzerer Form in der Bachelor-Arbeit von Florian Golemo (Golemo 2013)

(4) oder das Verfolgen von weiterführenden Links (5), die auf potentielle Kandidaten verlinkt waren. Nach der Extraktion wurde die Suche modifiziert (6) und wieder eingegeben. Dieses Verfahren wurde durchgeführt bis genügend Kandidaten gefunden wurden. Die gewählten Kandidaten wurden einer oberflächlichen, ersten Bewertung unterzogen (7), um die Ergebnisse weiter zu filtern. Waren während dieser Bewertung schon sehr schnell Mängel zu erkennen, wurde der Kandidat entfernt (8) und die Suche fortgesetzt. Sollten die Indikatoren auf dem ersten Blick gut bzw. ausreichend erscheinen, wurden diese später einer detaillierteren Analyse unterzogen (9).



Ein Beispiel für dieses Vorgehen:

- Suchanfrage bei google.de mit dem String „plm software“ (1).
- Diese Anfrage liefert zu einem großen Teil kommerzielle – proprietäre – Software. Einzig Aras Innovator konnte als kostenfreie Software identifiziert werden (2).
- Die Website ist übersichtlich und aktuell, bietet jedoch keine erfolgversprechenden weiterführenden Links (3).
- Der Kandidat Aras Innovator wird extrahiert (4).
- Eine erste Bewertung verweist auf keine großen Mängel (7). Die Software wird zu den Kandidaten hinzugefügt und später einer detaillierten Analyse (9) unterzogen.
- Modifikation des Suchstrings: „plm software open source“ (6).
- Suchanfrage über google.de (1).

- Nach Prüfung der Vorschautexte und vorgeschlagenen Treffer-URLs wird bei dieser Anfrage wieder der Aras Innovator gefunden und darüber hinaus die Software openPLM (Platz 4) gefunden (2).
  - Die Seite ist aktuell und verweist auf Sourceforge<sup>6</sup>(3).
  - Nach Identifizierung von openPLM und dem Testen des Downloads auf Sourceforge, wird unter dem Punkt „Recommended Projects“ das Projekt Oberon und „plm pdm bom dms“ empfohlen (5), welches die Kriterien eines PLM erfüllt
    - Die Oberon Plattform zeigt keine gravierenden Mängel (7). Sie wird in die Kandidatenliste angefügt und später einer detaillierten Analyse (9) unterzogen.
    - „plm pdm bom dms“ wird wegen einer Sprachbarriere nicht betrachtet<sup>7</sup> (8).
    - Allgemein: Suchstring kann um die Modifikation BOM und DMS erweitert werden.
  - Änderung des Suchportals auf Sourceforge (6).
  - Sichten der Kategorie „Product Lifecycle Mangement (PLM)“ auf Sourceforge<sup>8</sup> (1).
  - Ein Überblick dieser Kategorie liefert weitere Treffer (2).
  - OSSWorkx ist seit 06/2010 nicht mehr weiterentwickelt worden und die Projektwebsite ist offline (3) und wird nicht mit aufgenommen (4).
  - Ranchbe wird in die vorläufige Auswahl mit aufgenommen<sup>9</sup> (4).
- openPLM wird extrahiert (4).
- Modifikation des Suchstrings: „open source product lifecycle management“ (6).
- Suchanfrage über google.de (1).
- Diese Suche ergab den Treffer <http://www.plmportal.de>. (2).
- Auf dieser Seite findet man eine deutsche Aufbereitung des PLM-Gedanken sowie zahlreiche kostenpflichtige Anbieter. In der Rubrik Systeme > Open-Source PDM konnten zwei interessante Angebote gefunden werden (3).
- Die weiterführenden Links werden betrachtet (5):
  - FIDAB PDM ist eine Desktop-, und keine Websoftware (7), die Basisprüfkriterien werden damit nicht erfüllt.
  - Die weitere Analyse wird abgebrochen (8).
  - Da PDMWeb nicht mehr betreut wird und es keinen funktionierenden Download der Software gibt (7), werden damit die Basisprüfkriterien nicht erfüllt.
  - Die weitere Analyse wird abgebrochen (8).

---

<sup>6</sup> <http://sourceforge.net/projects/open-source-plm/?source=directory> (Stand: 29.04.2012)

<sup>7</sup> Projektsprache Chinesisch

<sup>8</sup> <http://sourceforge.net/directory/business-enterprise/enterprise/plm/freshness:recently-updated/> (Stand: 29.04.2012)

<sup>9</sup> Dieses Beispiel wird später noch detailliert dargestellt.

- Umordnen der Suchbegriffe (6).
- Suchanfragen bei Google (1).
- Resultat ist eine geänderte Trefferreihenfolge, aber keine neuen Treffer (2).
- .....

Als dieser und viele weitere Suchvorgänge abgeschlossen wurden, folgte die detaillierte Analyse, deren Kriterien und Indikatoren im Folgenden dargelegt werden. Auf diesen Teil aufbauend folgt dann die eigentliche Analyse.

### Suchkriterien

---

Um eine geeignete technische Software zu identifizieren, wurden Suchkriterien festgelegt, welche für die Recherche richtungswesend waren. Zunächst erfolgt eine Bewertung anhand der einzelnen Kriterien auf Basis der Internetauftritte und der dort veröffentlichten Informationen der identifizierten Softwareangebote. In einem zweiten Schritt wurden Programme, welche die sich der ersten Begutachtung als nützlich erwiesen haben, für eine tiefergehende Evaluation in eine Testumgebung installiert. Ziel war es die Funktionalitäten, das praktische Benutzen (falls z.B. keine Online-Demonstration vorhanden war) und die Erweiterbarkeit um weitere Module zu beleuchten.

#### Formale Suchkriterien

Am Anfang der Suche wurden zunächst formale Suchkriterien festgelegt und anschließend gewichtet. Diese Suchkriterien galt es – aufgrund ihres eher allgemeinen Charakters – zu spezifizieren, um sie für die Online-Recherche praktikabel zu gestalten. Die ursprünglichen Ansprüche an die Software waren:

- **Open-Source-Lösung** (*Gewichtung: Hoch*)

Das Kriterium, das die Software Open-Source sein soll, wurde im Laufe der Recherche modifiziert. Der Grund hierfür war, dass nur eine erwähnenswerte Open-Source-Software für ein PLM identifiziert werden konnte und dies kein befriedigendes Ergebnis darstellte. Darüber hinaus garantiert eine Open-Source-Lösung nicht, dass diese Software kostenfrei ist, sondern nur das der Code frei zugänglich ist. Wichtig ist, dass eine kostenfreie Entwicklungsumgebung zur Verfügung gestellt wird. Dafür ist es nicht notwendig, dass der zugrunde liegende Code (des Kernprogramms) verfügbar ist. Daraus ergab sich eine Modifikation, welche zwischen Open-Source und Freeware (Closed-Source) unterscheidet.

- **Entwicklungsumgebung** (*Gewichtung: Hoch*)

Zur Entwicklungsumgebung gehören zunächst die vorhandenen technischen Komponenten – also die Architektur und der darauf aufbauende Funktionsumfang. Hinzu kommt die Möglichkeit der

Erweiterung der Software, also ob und welche Schnittstellen existieren. Um effektiv entwickeln zu können, ist es darüber hinaus nötig auf eine strukturierte Dokumentation der Software und ihrer Möglichkeiten zurückgreifen zu können. Darüber hinaus spielt auch der Entwicklungsstand der Software eine Rolle. Dieser muss operationalisiert und mit einbezogen werden, um eine generelle Nutzbarkeit der Software zu gewährleisten, d.h. über den Use-Case hinaus nutzbar zu sein. All diese Punkte gehen in das Kriterium der Wiederverwendbarkeit ein.

Diese Merkmale werden im Folgenden operationalisiert und gewichtet und anschließend – auf Basis der Gewichtung – in ein Punktesystem überführt, um auf deren Basis eine Nutzwertanalyse der identifizierten Software durchzuführen.

### Operationalisierte Suchkriterien

- **Freeware** (*Gewichtung: Voraussetzung*)

Die Software soll prinzipiell kostenfrei zur Verfügung stehen, egal ob diese Closed- oder Open-Source ist.

❖ Indikator: kostenfreie Nutzung des Programms

- **Reife** (*Gewichtung: Mittel*)

Die Reife einer Software sagt etwas über die Entwicklungsstufe der Software und deren Nutzung (ohne Nutzung/Finanzierung keine Weiterentwicklung) aus. Sie spricht für die praktische Erfahrung der Programmierer und der Erweiterung der Software durch Features und Debugs. Eine hohe Reife senkt die Wahrscheinlichkeit von kritischen Fehlern und erhöht die Wahrscheinlichkeit des effektiven Einsatzes.

❖ Indikatoren: aktueller Stand; Geschäftsmodell

- **Regelmäßige Weiterentwicklung** (*Gewichtung: Mittel*)

Eine Software, die nicht regelmäßig weiterentwickelt wird, ist nicht immer auf dem Stand der Zeit und spricht nicht unbedingt für die Qualität der Software. Wenn zum Beispiel eine Software nicht aktualisiert wird und aus diesem Grund nicht auf Windows 9 installiert werden oder nicht mit Windows IIS 25 interagieren kann, entsteht ein nicht unerheblicher Mehraufwand der Anpassung und das Risiko der Benachteiligung gegenüber der Konkurrenz.

❖ Indikatoren: Zeitabstände der Aktualisierungen; Userforen; Bugtracker

- **Wiederverwendbarkeit** (*Gewichtung: Hoch*)

Die Wiederverwendbarkeit eines Systems ist wichtig, um die Anwendbarkeit über einen speziellen Use-Case hinaus zu gewährleisten. Es stellt sich die Frage, ob die Software

geeignet ist für eine generelle Nutzung als PDLM-Werkzeug. Dieses Kriterium kann untergliedert werden in die Unterkriterien Dokumentation, Schnittstellen und Funktionsumfang der Software.

- **Dokumentation** (*Gewichtung: Hoch*)

Die Dokumentation ist sowohl für einen Endnutzer als auch für einen Programmierer Grundvoraussetzung für das Arbeiten mit oder an einer Software. Mit einer Dokumentation können zum Beispiel Kosten und Nutzen einer Software und einer Software-Weiterentwicklung besser eingeschätzt werden.

- ❖ Indikatoren: Quantität (Umfang); Qualität (Inhalt und Struktur der Informationen; Realitätsnähe der Informationen)

- **Erweiterbarkeit** (*Gewichtung: Hoch*)

Die Erweiterbarkeit von Softwares gliedert sich in zwei Punkte. Es gibt zum einen die Möglichkeiten der Erweiterung durch Schnittstellen, welche die Möglichkeiten der Modifikation von einzelnen Modulen wie auch der Implementierung eigener Module beinhaltet, und zum anderen die Möglichkeit der Veränderung und Anpassung des Programmkerns selbst (Open-Source). Die Qualität der Schnittstellen kann an dieser Stelle nicht bestimmt werden, weil hierfür umfangreiche Tests und Praxis benötigt werden. Die Qualität von Schnittstellen kann nur indirekt über die Dokumentation bestimmt werden, da die Wahrscheinlichkeit von qualitativ hochwertigen Schnittstellen durch eine sehr gute Dokumentation steigt.

- **Schnittstellen** (*Gewichtung: Hoch*)

- ❖ Indikatoren: Quantität (Welche gibt es?); Qualität (kann nicht bestimmt werden)

- **Open-Source** (*Gewichtung: Niedrig*)

Eine niedrige Gewichtung entsteht durch die Unterscheidung von Freeware und Open-Source. Der Faktor Open-Source hat in diesem Falle nur einen geringen Vorteil gegenüber freien Mixed- oder Closed-Source-Lösungen.

- ❖ Indikator: offener Quelltext

- **Funktionsumfang der Software** (*Gewichtung: Mittel-Hoch*)

Hier steht die Frage im Vordergrund, welche Funktionen von der Software für ein PDLM schon abgedeckt werden und wie diese umgesetzt wurden. Dementsprechend ist nicht nur entscheidend, wie viele Funktionen es gibt, sondern wie diese umgesetzt wurden, d.h. welche Anforderungen an sie gestellt wurden und wie diese wirklich funktionieren. Dies

umfasst darüber hinaus die Frage, ob diese Funktionen vielleicht auch generisch genug sind, um SLM Methoden zu unterstützen. Obwohl die Qualität des Funktionsumfangs eine sehr wichtige Einflussgröße ist, kann diese nicht einfach ermittelt werden. Nur in der Praxis kann dies bestimmt werden, welche hier jedoch fehlt. Indirekt fließt aber diese Größe mit in Bewertung der Dokumentation ein, denn wenn diese ausführlich und gut ist, ist die Wahrscheinlichkeit hoch, dass die Qualität der Funktionen auch gut ist.

❖ Indikatoren: Quantität (Anzahl der Funktionen); Qualität der Funktionen (kann nicht bestimmt werden)

• **Benutzerfreundlichkeit** (*Gewichtung: Mittel*)

Die Benutzerfreundlichkeit ist ein nicht zu vernachlässigter Faktor der Softwareauswahl. Es unterliegt in einem hohen Maße einer subjektiven Einschätzung. Es wird davon ausgegangen, dass die hier beteiligten Tester über das technische Know-How verfügen, um eine prinzipielle Unbenutzbarkeit identifizieren zu können. Zugleich soll dieses Kriterium aber nicht überbewertet werden, da ein gutes UI auch durch eine eigene Entwicklung gewährleistet werden könnte – was im Gegenzug ein Mehraufwand ist und entsprechend bewertet werden muss.

❖ Indikator: subjektive Einschätzung

• **Support** (*Gewichtung: Niedrig*)

Ein Support ist für die vorliegende Suche kein wichtiges Kriterium für eine geeignete Softwarelösung.

❖ Indikator: Supportanbindung

Aus den vorgestellten Kriterien und deren Indikatoren wurde eine Skala entwickelt, mittels derer die gefundenen Software Lösungen bewertet werden. Diese Skala richtet sich an den schon getroffenen Gewichtungen aus und ist entscheidend für die endgültige Auswahl.

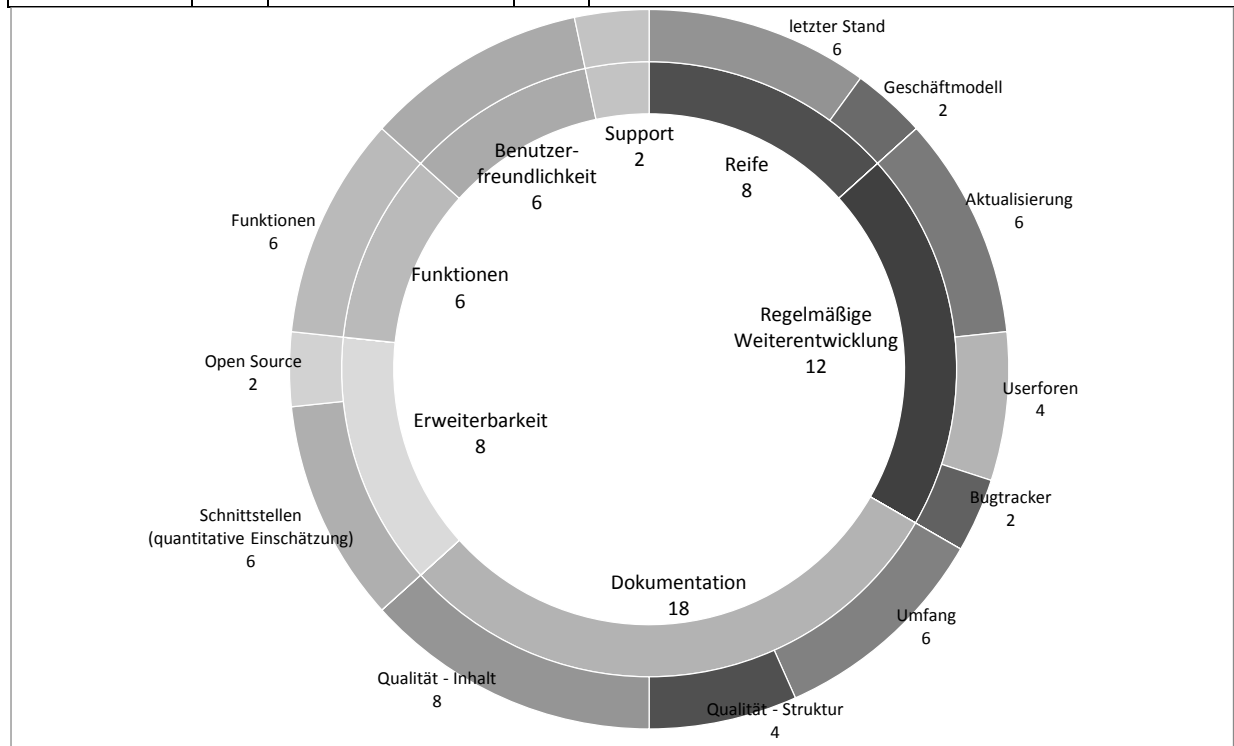
Kriterium	Punkte	Indikator	Punkte	Punktvergabe
Reife	8	<u>aktueller Stand</u>	6	Die Punkte werden für Zeitabstände vergeben und sind wie folgt gestaffelt: zwölf Monate (0); acht Monate (1); sechs Monate (2); vier Monate (3); zwei Monate (4); einen Monat (5) und aktueller Monat (bzw. die letzten 30 Tage) (6). Weiterhin kann sich eine niedrige, gar keine, oder Beta Version negativ auf die Bewertung aus.
		<u>Geschäftsmodell</u>	2	Wenn ein Geschäftsmodell deutlich erkennbar ist, werden diese Punkte vergeben (Cross-Selling; angegliederte Produkte, Services (oder Support))

<b>Weiter- entwicklung</b>	12	<u>Aktualisierung</u>	6	Die Punkte werden nach den Intervallen der Aktualisierungen vergeben. So entsprechen 2 Punkte einer Aktualisierung mehrfach pro Jahr; 3 – mehrfach pro Halbjahr; 4 – mehrfach pro Vierteljahr und 6 mehrfach pro Monat. Art und Umfang der Aktualisierungen werden mit in die Bewertung aufgenommen.
		<u>Userforen</u>	4	Es werden Punkte vergeben – sofern es ein Forum gibt – nach Nutzeraktivitäten, so werden 2 Punkte für eine monatliche Aktivität im Forum vergeben und 4 Punkte für eine tägliche Aktivität von Nutzern im Forum.
		<u>Bugtracker</u>	2	Punkte gibt es hier, falls ein Bugtracker eingerichtet (1) und aktiv (2) ist.
<b>Dokumentati on</b>	18	<u>Umfang</u>	6	Die Grenzen der Punktevergabe sind hier sehr fließend, deshalb werden folgende Eckpunkte der Bewertung wie folgt festgelegt: zwei Punkte werden vergeben, falls die Dokumentation unzureichend ist, d.h. es werden nur einzelne Aspekte beleuchtet und es ergibt sich allein aus der Doku kein schlüssiges Bild, wie man die Software benutzt oder damit entwickelt. Vier Punkte werden vergeben, wenn große Teile und Aufgaben der täglichen Arbeit und übliche Entwicklungsfragen abgedeckt sind. Sechs Punkte werden vergeben, wenn alle Aspekte der Software vollends beleuchtet werden und mit hilfreichen und weiterführenden Links und Quellen gefüllt sind.
		<u>Qualität – Struktur</u>	4	Die Struktur einer Dokumentation wird wie folgt bewertet: einen Punkt, falls diese unklar und verstreut ist. Zwei Punkte, wenn diese einigermaßen zusammenhängend ist, drei Punkte für eine erkennbare aber sehr umständliche Struktur (viele Barrieren) und vier Punkte für eine zentrale, klare, intuitive Struktur und gute interne Verlinkung.



		<u>Qualität – Inhalt</u>	8	Für die Qualität von Dokumentationen sind die folgenden Maßgaben richtungsweisend. Eine Dokumentation mit zahlreichen Anweisungen, die falsch, veraltet oder unklar formuliert sind, wird mit einem Punkt bewertet. Unterdurchschnittliche Leistung (2-3) sind Dokumentationen deren Anweisungen Erfahrung zur Umsetzung bedürfen. Befriedigend (4-5) sind Dokumentationen bei denen alle Anweisungen und Beispiele ohne oder mit klar-definierten Änderungen umsetzbar sind. Als gut (6-7) sind Dokumentationen zu bewerten, die Codebeispiele oder ergänzende Hinweise angeben. Als exzellent werden Dokumentationen bewertet, die auf häufige Fehler eingehen und weiterführendes Material empfehlen, also hilfreiches Wissen an vielen Stellen mit einfließen lassen, ohne die Übersichtlichkeit zu verlieren.
<b>Erweiterung</b>	8	<u>Schnittstellen (quantitative Einschätzung)</u>	6	Falls es Schnittstellen gibt, werden folgenden Bewertungskriterien als Eckpunkte angelegt. 2 Punkte werden vergeben, wenn es Schnittstellen von außen gibt, das heißt das Programm kann in andere eingebunden werden. 4 Punkte gibt es, wenn es Schnittstellen von und nach außen gibt, diese aber nur ein sehr begrenztes Repertoire von Funktionen / Datennutzung verfügen, bzw. über die Schnittstellen erreichbar sind. 6 Punkte – also exzellente Schnittstellen - werden erreicht, wenn nahezu alle wichtigen Funktionen und Daten in die Software geholt und von der Software abgerufen werden können. Ausschlaggebend für die Bewertung ist also die Anzahl der Möglichkeiten, die sich durch die Schnittstellen bieten.
		<u>Open-Source</u>	2	Die Verteilung erfolgt relativ simpel. Ein reines Closed-Source Projekt wird keine Punkte erhalten. Sind Teile des Programms zugänglich, so kann ein Punkt vergeben werden. Vollwertige Open-Source Projekte erhalten 2 Punkte.
<b>Funktion</b>	6	<u>Quantitative Einschätzung</u>	6	Die Punktevergabe erfolgt auf Basis der Anzahl von Funktionen, welche die Software umfasst oder umfassen kann. Eine hohe Anzahl von Funktionen für ein PLM ist entsprechend hoch bewertet. Über die Qualität kann ohne detaillierte Tests keine Aussage getroffen werden.
<b>Benutzerfreundlichkeit</b>	6	<u>Subjektive Einschätzung</u>	6	Die Benutzerfreundlichkeit unterliegt einem hohen subjektiven Grad. Um die Bewertung dennoch nachvollziehbar zu halten, werden folgende Kriterien festgelegt: Null Punkte gibt es, wenn das Produkt über keine Benutzeroberfläche verfügt. Zwei Punkte, wenn die Oberfläche kompliziert und unübersichtlich ist sowie eine erhebliche Einarbeitungszeit notwendig ist und dennoch Unklarheiten über bestimmten Symbole und Schaltflächen

				herrschen. Drei Punkte werden vergeben, wenn die Oberfläche zwar unübersichtlich ist, aber nach ein wenig Einarbeitung flüssig damit gearbeitet werden kann. Vier Punkte entsprechen einer Oberfläche, die einfach ist und nur eine geringfügige Einarbeitung benötigt. Die höchste Punktzahl gibt es für intuitive Oberflächen, welche sehr einfach strukturiert sind und nur wenig Einarbeitung benötigt. Dabei erschließen sich die Funktionen aller Schaltflächen und Symbole dem Nutzer sehr schnell.
<b>Support</b>	2	<u>Anbindung</u>	2	Null Punkte gibt es, wenn kein erkennbarer Support oder Möglichkeit zum Feedback geben erkennbar ist. Einen Punkt erhalten Produkte, die zumindest über einen passiven Support verfügen (Forum etc.) oder einen kostenpflichtigen Support anbieten. Höchste Punktzahl bekommen Produkte mit aktivem Support (Email; Formulare; Servicetelefon; offene Ticketsysteme o.ä.), welcher bestenfalls kostenfrei ist.



### Identifizierte Lösungen

Mehrere Lösungen wurden gesichtet und bewertet. Um das praktische Suchvorgehen zu veranschaulichen, sollen zwei Software Systeme vorgestellt werden, welche die Bewertungskriterien nicht ausreichend erfüllt haben. Hierfür wurde Ranchbe und Xingo DMS ausgewählt. Anschließend werden die drei Systeme vorgestellt, die den Bewertungskriterien am besten gerecht wurden. Es handelt sich hierbei um den Aras Innovator, openPLM und die Oberon Plattform.

## Ranchbe

Ranchbe wurde auf Sourceforge unter der Kategorie „Product Lifecycle Mangement (PLM)“ gefunden. Die Website des Anbieters ist komplett Französisch, ohne Sprachauswahl<sup>10</sup>. Die Übersetzung wurde zunächst so schmal wie möglich gehalten, um zuerst einmal nur eine prinzipielle Eignung feststellen zu können. Die Sprachbarriere an sich gibt aber einen großen Minuspunkt. Die Software basiert auf Modulen von Tikiwiki<sup>11</sup> - eine bekannte Open-Source-Groupware.

## Bewertung

- Reife
  - Indikator: aktueller Stand

Die Software befindet sich auf dem Stand der Version 0.8.1. Die letzte zu erkennende Aktualisierung war 2010 und zwischen 2008 und 2010 gab es keinerlei Entwicklungen<sup>12</sup>. Die Projektseite ist zudem unvollständig und verlinkt zu nicht existierenden URLs.<sup>13</sup>

❖ Bewertung: 0 / 6
  - Indikator: Geschäftsmodell

Es wird zwar auf einen Support verwiesen, bzw. eine Firma, jedoch gibt es die verlinkte Website nicht.<sup>14</sup>

❖ Bewertung: 0 / 2
  - ❖ Bewertung: 0 / 8
- Funktionsumfang
  - Indikator: quantitative Einschätzung

Ranchbe ist ein erweitertes Dokumentenmanagement, was auch für CAD-Daten eingesetzt werden kann. Es ist auch ein Kundenkontakt-Management erkennbar. Grundsätzliche Funktionalitäten, wie ein Lebenszyklus oder Komponentenmanagement, ist nicht erkennbar<sup>15</sup>.

❖ Bewertung: 2 / 6
- Abbruch der Betrachtung

---

<sup>10</sup> <http://www.ranchbe.com/spip.php?rubrique1> (Stand:19.04.2012)

<sup>11</sup> <http://info.tiki.org/tiki-index.php> (Stand:19.04.2012)

<sup>12</sup> <http://www.ranchbe.com/spip.php?page=breves-toutes> (Stand:19.04.2012)

<sup>13</sup> <http://www.ranchbe.com/spip.php?rubrique9> (Stand:19.04.2012)

<sup>14</sup> <http://www.ranchbe.com/spip.php?rubrique8> (Stand:19.04.2012)

<sup>15</sup> <http://ns37734.ovh.net/~plmpdm/ranchbe-demo/accueil.php> (Stand:19.04.2012)

Da Basisindikatoren (Reife/Funktion) schon sehr mangelhaft sind und eine nicht unerhebliche Sprachbarriere existiert, wurde die weitere Analyse als nicht lohnend angesehen und die Bewertung wurde abgebrochen.

Ranchbe ist ein erweitertes Dokumentenmanagementsystem, welches schon in einer sehr frühen Version nicht weiterentwickelt wurde und stellt kaum die Funktionalitäten, die eine PLM-Software haben müsste, zur Verfügung.

### Xinco DMS

Auf der Suche nach Open-Source-Lösungen für das PDM wurde nicht nur nach explizit ausgewiesenen PLM- oder PDM-Systemen gesucht, sondern auch nach Lösungen, die diesen Softwaretypen nahe stehen. Dies wurde notwendig durch die Einsicht, dass nur wenig Open-Source Software für PLM zur Verfügung steht. Daher wurden andere Managementsoftwaresysteme mit in die Suche einbezogen, so u.a. auch Informations- und Dokumentenmanagementsysteme, welche zur Verwaltung und Steuerung von Daten in Unternehmen dienen<sup>16</sup>. Bei Xinco DMS<sup>17</sup> handelt es sich um ein Ergebnis dieser Suchmodulation und ist entsprechend ein Beispiel für ein Open-Source Dokumentenmanagement-System. Im Folgenden wurde die Software gesichtet und anhand der Indikatoren bewertet. Diese Bewertung führte zu einem relativ schnellen Abbruch, da wichtige Indikatoren eine sehr ungenügende Bewertung erhielten und eine tiefere Untersuchung als wenig sinnvoll erschienen ließen. Folgenden Bewertungen wurden vorgenommen:

#### Bewertung

- Funktionsumfang

- Indikator: quantitative Einschätzung

Die Software verfügt über zahlreiche Features, wie Dokumentenverwaltung oder auch Volltextindexierung von Dokumenten<sup>18</sup>. Die Software legt mit den schon integrierten Funktionen den Fokus auf interne Verwaltung von Dokumenten. Dieser Fokus lässt wenig Raum für eine schnelle Integration von anderen Softwareelementen, die ein PDM-System, oder PLM-System benötigt. So ist keine Nutzerverwaltung oder Kundenverwaltung explizit aufgeführt, Tools zum Organisieren von Prozessen und Abläufen sind nicht vorhanden. Der Funktionsumfang ist damit eindeutig ungenügend.

- ❖ Bewertung: 1 / 6

---

<sup>16</sup> Ein solches System kann als PDM-Werkzeug in einem Unternehmen dienen.

<sup>17</sup> [http://www.xinco.org/index\\_de.php](http://www.xinco.org/index_de.php) (Stand:19.04.2012)

<sup>18</sup> [http://www.xinco.org/index\\_de.php](http://www.xinco.org/index_de.php) (Stand:19.04.2012)

- Erweiterung

- Indikator: Open-Source

Es handelt sich um eine vollwertige Open-Source-Lösung, welche unter Apache License V2.0 und GNU General Public License version 2.0 (GPLv2) veröffentlicht wurde<sup>19</sup>.

❖ Bewertung: 2 / 2

- Indikator: Schnittstellen (quantitative Einschätzung)

Besonders schwer fallen die fehlenden Schnittstellen zu anderen Systemen ins Gewicht. Das bedeutet, dass Xinco DMS nicht darauf ausgerichtet ist, mit anderen Systemen zu interagieren<sup>20</sup>. Jede Anbindung zu anderen System muss von Hand programmiert und in den Quellcode eingebunden werden.

❖ Bewertung: 0 / 6

- ❖ Bewertung: 2 / 8

- Benutzerfreundlichkeit

- Indikator: subjektive Einschätzung

Während des Testens des Systems fällt auf, dass das Browser-Frontend bei weitem nicht ausgereift ist<sup>21</sup>. Einziger Vorteil ist, dass es überhaupt ein Browser-Frontend gibt, welches Plattform-unabhängig gestaltet werden kann.

- ❖ Bewertung: 1 / 6

- Abbruch der Betrachtung

Da Basisindikatoren (Erweiterung/Funktion) schon mangelhaft bewertet wurden, wurde die weitere Analyse als nicht lohnend angesehen und die Bewertung wurde abgebrochen.

Xinco DMS, wie auch Ranchbe, sind exemplarische Beispiele für Software, die sich schon nach wenigen Bewertungsindikatoren als unbrauchbar herausstellte. Die erhoffte Offenheit von Usermanagement und erweiterbaren Informationsmanagement zu Gunsten eines PLM wurde nicht gefunden.

### openPLM

Die Software openPLM ist eine sehr gut zu findende Open-Source-Lösung für eine PLM-Software. Zu finden ist sie über die Stichwörter „open source plm“ (Platz 4) und „plm open source“ (Platz 6)<sup>22</sup> bei

---

<sup>19</sup> <http://sourceforge.net/projects/xinco/> (Stand:18.04.2012)

<sup>20</sup> <http://sourceforge.net/p/xinco/wiki/Home/> (Stand:18.04.2012)

<sup>21</sup> <http://xinco.org:8080/xinco> (Stand:18.04.2012)

<sup>22</sup> <http://www.semager.de/keywords/url-analyse.php?url=http%3A%2F%2Fwww.openplm.org%2F&mainkey=> (Stand: 11.04.2012)

Google. openPLM bietet, laut eigenen Angaben<sup>23</sup>, eine sichere, offene und flexible Lösung für Produktdaten-Management und ist weiterhin geeignet für alle Managementansätze, die mit einem PLM einhergehen. Die Software ist nicht nur Open-Source, sondern auch entsprechend kostenfrei zu beziehen.

### Bewertung

- Reife

- Indikator: aktueller Stand

Die letzte Aktualisierung war laut offizieller Timeline am Tag der Bewertung (14.04.2012)<sup>24</sup>. Allerdings befindet sich die Software in der Version 0.5. Laut System ist Version 1.0 zu 81% vollständig<sup>25</sup>. Die niedrige Version wirkt sich negativ auf die Bewertung aus, da die Software sich auf einem sehr jungen Stand befindet.

❖ Bewertung: 3 / 6

- Indikator: Geschäftsmodell

Für die französische Firma linObject<sup>26</sup> ist openPLM zum einen eine Demonstration ihrer Fähigkeiten für die Schaffung von Software Lösungen und zum anderen ein Programm zu dem sie Support, Hilfe und Workshops anbieten. Es ist kaum einzusehen, welcher Teil überwiegt. Es legt aber die Befürchtung nahe, dass openPLM ein Showroom für einen Software-Entwickler ist, u.a. auch deshalb, weil keine Referenzen oder Anwendungspartner angegeben werden.

❖ Bewertung: 1 / 2

- ❖ Bewertung: 4 / 8

- Weiterentwicklung

- Indikator: Aktualisierungen

Die Aktualisierungen sind zwar stetig, aber nur auf das Jahr gerechnet. So liegen beispielsweise zwischen Release 0.5 und Release 0.4 vier Monate<sup>27</sup>.

❖ Bewertung: 2 / 6

- Indikator: Userforen

---

<sup>23</sup> <http://www.openplm.org/> (Stand: 11.04.2012)

<sup>24</sup> <http://wiki.openplm.org/trac/timeline> (Stand: 11.04.2012)

<sup>25</sup> <http://wiki.openplm.org/trac/roadmap> (Stand: 11.04.2012)

<sup>26</sup> <http://www.linobject.com/> (Stand: 19.04.2012)

<sup>27</sup> <http://wiki.openplm.org/trac/timeline?from=04%2F19%2F12&daysback=90&authors=&milestone=on&update=Update> (Stand: 19.04.2012)

Es ist ein Forum vorhanden und dieses wird auch benutzt. Jedoch beschränkt sich die Nutzung auf ein bis zwei Beiträge im Monat<sup>28</sup>.

❖ Bewertung: 2 / 4

- Indikator: Bugtracker

Es ist ein Ticketsystem (trac) eingerichtet, dieses ist offen zugänglich und aktiv<sup>29</sup>.

❖ Bewertung: 2 / 2

❖ Bewertung: 6 / 12

- Dokumentation

- Indikator: Umfang

Die Dokumentation ist auf den ersten Blick sehr umfangreich<sup>30</sup>. Die Übersicht bietet eine Bandbreite von Themen und Gebieten, die abgedeckt sind. Besonders gut ist, dass es nicht nur auf Programmierer zugeschnitten ist, sondern eine explizite Dokumentation für Administratoren und User zur Verfügung steht. Die multimediale Unterstützung könnte noch ausgebaut werden, wobei der Umfang dennoch als sehr gut bewertet werden kann.

❖ Bewertung: 5 / 6

- Indikator: Qualität – Struktur

Die Struktur der Dokumentation ist auf einem sehr guten Stand. Sie ist gut strukturiert, zentral und intuitiv zu bedienen.

❖ Bewertung: 4 / 4

- Indikator: Qualität – Inhalt

Die Dokumentation ist mit vielen Beispielen und Anwendungsfällen bestückt und gut beschrieben<sup>31</sup>. Zu einer sehr guten Dokumentation fehlen nur noch weiterführende Links und Wissen über diese Basis hinaus, die eingebunden werden könnte.

❖ Bewertung: 6 / 8

❖ Bewertung: 15 / 18

- Erweiterung

- Indikator: Open-Source

---

<sup>28</sup> <http://wiki.openplm.org/trac/discussion/forum/1?order=lastreply&desc=1> (Stand: 11.04.2012)

<sup>29</sup> <http://wiki.openplm.org/trac/report> (Stand: 19.04.2012)

<sup>30</sup> <http://wiki.openplm.org/docs/index.html> (Stand: 19.04.2012)

<sup>31</sup> Beispiel: <http://wiki.openplm.org/docs/devel/architecture.html> (Stand: 19.04.2012)

Es handelt sich um eine vollwertige Open-Source-Lösung, unter Apache und GNU veröffentlicht<sup>32</sup>. Der Quellcode kann über einen SVN Kanal bezogen werden<sup>33</sup>.

❖ Bewertung: 2 / 2

- Indikator: Schnittstellen

Die Software verfügt über fest definierte Schnittstellen zu anderen Systemen. Es werden Plugins mitgeliefert, die mit anderen Open-Source-Software-Lösungen (Free-CAD; Open-Office) kommunizieren können<sup>34</sup> – z.B. den Versand von Dokumenten von Open-Office an openPLM. Diese Schnittstellen sind aber sehr spezifisch und nur für Nutzer dieser speziellen Programme interessant. Andere Schnittstellen sind nicht aufgeführt, müssten entsprechend selbst erstellt und eingebunden werden. Die Schnittstellen sind daher als Mangelhaft zu bewerten.

❖ Bewertung: 2 / 6

❖ Bewertung: 4 / 8

- Funktionsumfang

- Indikator: quantitative Einschätzung

Die Software verfügt über zahlreiche Features, wie Dokumenten-, Konfigurations- und Lebenszyklus-Verwaltung. Hinzu kommen eine Historienverwaltung für alle Objekt (erleichtert die Nachvollziehbarkeit) und eine BOM (Bill of Materials) Verwaltung<sup>35</sup>. Auch ist ein Usermanagement (mit Rechtesystem) enthalten, dieses muss jedoch separat (im Django-Backend) durchgeführt werden. Nicht implementiert sind eine Kundenverwaltung oder ein übergeordnetes Projektmanagement. Daher nur eine mäßige Bewertung.

❖ Bewertung: 3 / 6

- Benutzerfreundlichkeit

- Indikator: subjektive Einschätzung

Die Oberfläche ist optisch nicht ansprechend, jedoch nach kurzer Eingewöhnung gut benutzbar<sup>36</sup>. Fehlermeldungen werden nicht deutlich genug hervorgehoben. Die stets-verfügbare Suche fällt positiv auf. Bei der Orientierung in den Datensätzen entpuppt sich die eingesetzte Zweiteilung zwischen “Navigate” und “Study” als sehr hilfreich. Mit Hilfe dieser Funktionen kann man durch die Teilebäume navigieren und wenn man an

---

<sup>32</sup> <http://www.openplm.org/> (Stand: 19.04.2012)

<sup>33</sup> [http://wiki.openplm.org/docs/admin/ht\\_1\\_install\\_server.html](http://wiki.openplm.org/docs/admin/ht_1_install_server.html) (Stand:19.04.2012)

<sup>34</sup> <http://wiki.openplm.org/docs/user/plugins.html> (Stand:19.04.2012)

<sup>35</sup> <http://www.openplm.org/> (Stand: 19.04.2012)

<sup>36</sup> <http://standard.openplm.org/accounts/login/?next=/> (Stand: 19.04.2012)



interessanten Teilen oder Dokumenten angelangt ist, kann man diese schnell sichten (mit “Study”). Bei Betrachtung von größeren Objektmengen ist die Navigation allerdings nur eingeschränkt von Nutzen, da der Objektbaum unübersichtlich wird und die Orientierung schwerer fällt.

❖ Bewertung: 4 / 6

- Support

- Indikator: Anbindung

Support erhält man entweder kostenpflichtig über die Projektinitiatoren linObject<sup>37</sup> oder über das vorhandene Forum bzw. Wiki-System<sup>38</sup>.

❖ Bewertung: 1 / 2

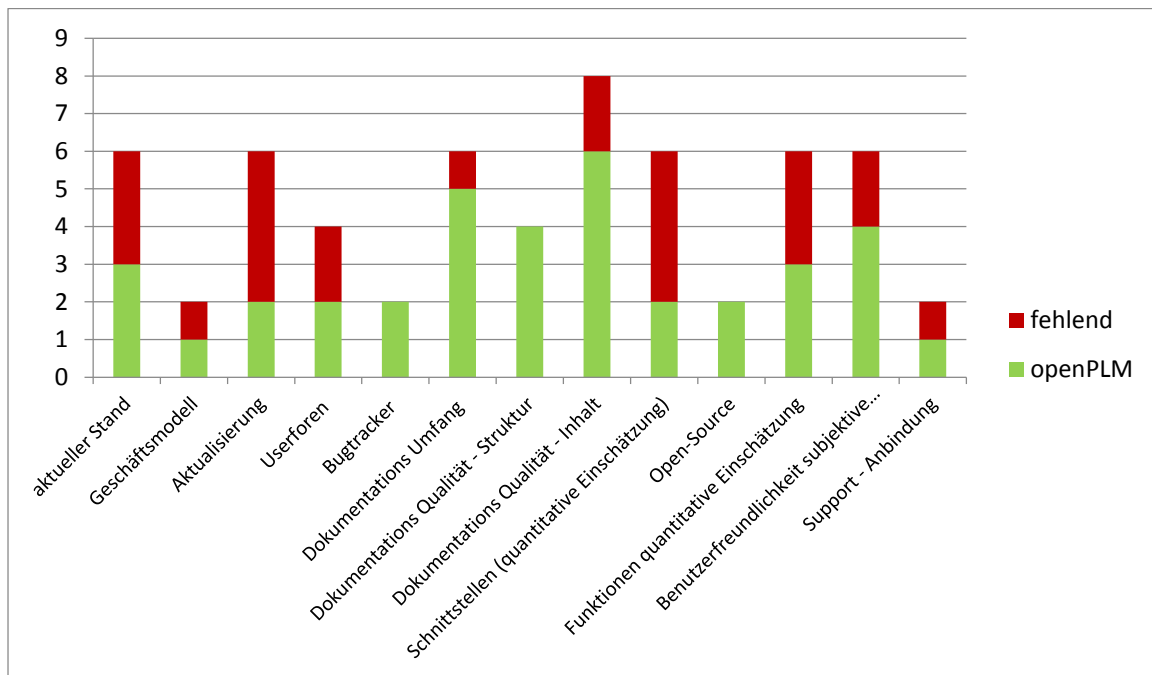


Abbildung 9 Bewertung openPLM

### Architektur

openPLM ist ein Framework aus zahlreichen OS-Komponenten, von denen erst alle installiert und eingerichtet werden müssen, bevor das Gesamtsystem funktioniert. Als Umgebung wird dringend Unix/Linux empfohlen, da nicht gewährleistet werden kann, dass alle Komponenten auf einem anderen Server – wie z.B. Windows – installiert werden können.

<sup>37</sup> <http://www.linobject.com/services/> (Stand: 19.04.2012)

<sup>38</sup> <http://wiki.openplm.org/trac/discussion> (Stand: 19.04.2012) und <http://wiki.openplm.org/trac/wiki> (Stand 19.04.2012)

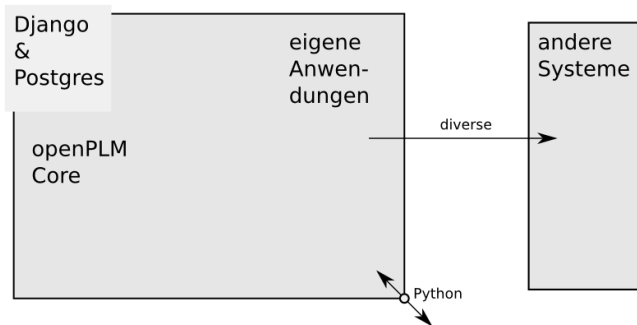


Abbildung 10 Architektur openPLM

Die Software basiert auf Python und wird auf einem Django-Webserver betrieben<sup>39</sup>. Dieser wird im praktischen Einsatz meist hinter einen Apache o.ä. gebräuchlicheren Webserver gelegt (bzw. ist dieses Szenario vom Entwickler empfohlen<sup>40</sup>). Der Django-Server kommuniziert mit einem Datenbankserver, im entsprechenden Tutorium wird dafür PostgreSQL

empfohlen<sup>41</sup>. Der Django-Server bzw. die Python-Umgebung benötigt verschiedene Komponenten. Dazu gehören<sup>42</sup>:

- Celery, ein Job-Warteschlangen-Server, der Prozesse wie Mailversand, Cronjobs und weitere im Hintergrund übernimmt<sup>43</sup>
- Haystack, ein Django-Such-Plugin mit Xapian als Backend<sup>44</sup>
- Graphviz, das Toolkit zur programmatischen Erzeugung von Grafiken<sup>45</sup>

Nachdem alle Komponenten einmalig eingerichtet wurden sind sie relativ wartungsfrei. Einzig der Django-/Apache-Server könnte weiterer Anpassung bedürfen.

### Fazit

Obwohl openPLM viele vielversprechende Ansätze besitzt, kommt es über eine Gesamtbewertung von 37/(60) Punkten nicht hinaus. Das offene Trac-System und die gut ausgebaute Dokumentation sind besonders gut. Besonders schwer wiegen hingegen der unausgereifte Stand, die ungleichmäßigen Weiterentwicklungen und die fehlenden Schnittstellen, welches eine große Schwachstelle der Software ist. Auch die Plattformabhängigkeit ist kein Pluspunkt für die Software.

### Aras Innovator

Aras Innovator bietet – laut eigenen Angaben – eine breite Palette von Softwarelösungen für ein Produkt-Lebenszyklus-Management<sup>46</sup>. Sie sollen Innovation, Zusammenarbeit und Koordination auf

<sup>39</sup> <http://wiki.openplm.org/docs/devel/architecture.html#main-dependencies> (Stand:19.04.2012)

<sup>40</sup> [http://wiki.openplm.org/docs/admin/ht\\_1\\_install\\_server.html#requirements](http://wiki.openplm.org/docs/admin/ht_1_install_server.html#requirements) (Stand: 11.04.2012)

<sup>41</sup> [http://wiki.openplm.org/docs/admin/ht\\_1\\_install\\_server.html#requirements](http://wiki.openplm.org/docs/admin/ht_1_install_server.html#requirements) (Stand: 11.04.2012)

<sup>42</sup> <http://wiki.openplm.org/docs/devel/architecture.html#main-dependencies> (Stand:19.04.2012)

<sup>43</sup> <http://celeryproject.org/> (Stand:19.04.2012)

<sup>44</sup> <http://haystacksearch.org/> (Stand:19.04.2012) und <https://github.com/notanumber/xapian-haystack> (Stand: 19.04.2012)

<sup>45</sup> <http://www.graphviz.org/> (Stand:19.04.2012)

<sup>46</sup> <http://www.aras.com/solutions/> (Stand:11.04.2012)

hoher Ebene verbessern. Darüber hinaus wird, laut Entwickler, ein großes Repertoire an Methoden für ein PLM abgedeckt werden, einschließlich einem „Program-Management“, „Product Engineering“ und einem „Quality Planning“<sup>47</sup>. Die Entwickler bezeichnen das von ihnen entwickelte System als Mixed-Source bzw. „Enterprise Open Source“ – mit dieser Bezeichnung soll verdeutlicht werden, dass einige Programmteile Closed-Source sind um den (Sicherheits-)Ansprüchen von geschäftskritischen Bereichen gerecht zu werden. Diese Mixed-Source-Lösung hat keine weiteren Konsequenzen, was bedeutet, dass es sich bei dieser Software um Freeware handelt<sup>48</sup>.

### Bewertung

- Reife

- Indikator: aktueller Stand

Die Software liegt in Version 9.3 SP2 vor. Laut Website wird im April 2012 SP3 geliefert, welches aber noch nicht zum Download bereitsteht.<sup>49</sup>

❖ Bewertung: 5 / 6

- Indikator: Geschäftsmodell

Laut eigenen Angaben gehören zu den Kunden Firmen wie Hitachi, Xerox und General Electric<sup>50</sup>. Darüber hinaus zeichnet sich ein deutliches Geschäftsmodell über Support-Dienstleistungen und Workshops ab<sup>51</sup>. Aras Innovator ist das Hauptprodukt, welches durch viele Dienstleistungen um diese Software finanziert wird.

❖ Bewertung: 2 / 2

- ❖ Bewertung: 7 / 8

- Weiterentwicklung

- Indikator: Aktualisierungen

In einem Zeitraum von einem Monat wurde der Aras Innovator zweimal aktualisiert (Sicherheitsupdates). Die letzte vollwertige Veröffentlichung (9.3) war vor einem dreiviertel Jahr<sup>52</sup>, Stand März 2012. Aus diesen Daten ergibt sich eine durchschnittliche Bewertung. Positiv angerechnet werden die geplanten Releases im Monatstakt und das diese, inklusive Features, transparent gehalten werden<sup>53</sup>.

---

<sup>47</sup> <http://www.aras.com/solutions/> (Stand: 11.04.2012)

<sup>48</sup> <http://www.aras.com/technology/open-source.aspx> (Stand: 11.04.2012)

<sup>49</sup> <http://www.aras.com/plm-roadmap/> (Stand: 16.04.2012)

<sup>50</sup> <http://www.aras.com/customers/> (Stand: 11.04.2012)

<sup>51</sup> <http://www.aras.com/services/> (Stand: 11.04.2012)

<sup>52</sup> <http://www.aras.com/support/documentation/documentation.aspx?Version=9.3.0> (Stand: 11.04.2012)

<sup>53</sup> <http://www.aras.com/plm-roadmap/> (Stand: 11.04.2012)

❖ Bewertung: 4 / 6

- Indikator: Userforen

Es gibt ein Forum für User, welches regelmäßig benutzt wird. Es gibt mehrere Einträge pro Woche, wobei die Reaktionszeit zwischen Stunden und Tagen schwankt<sup>54</sup>.

❖ Bewertung: 3 / 4

- Indikator: Bugtracker

Ein Tracksystem wurde nicht entdeckt und ist nicht transparent. Das liegt wahrscheinlich an der Mixed-Source-Lösung, die es quasi unmöglich macht, ein solches System sinnvoll zu implementieren, da die Teile, die Closed-Source sind, als vertraulich gehandhabt werden und die Open-Source-Parts so individuell sind, dass nur Support am Kunden Sinn macht. Es gibt nur eine Roadmap, deren einzelne Teile von Usern bewertet werden können<sup>55</sup>.

❖ Bewertung: 1 / 2

❖ Bewertung: 8 / 12

- Dokumentation

- Indikator: Umfang

Es steht eine Dokumentation<sup>56</sup> bereit, außerdem Selbstlernmaterial, das jeweils verschiedene Themenbereich bzw. Installation oder Konfiguration abdeckt<sup>57</sup>. Auch die API ist gut dokumentiert.

❖ Bewertung: 5 / 6

- Indikator: Qualität – Struktur

Die Dokumente sind auf der Website teils sehr verstreut und der Zugang zu einer zusammenhängenden Dokumentation muss erkauft werden. Außerdem liegen die Dokumentationen im pdf-Format vor, was zwar gut ist für die Offline-Verwendung, aber auf Kosten der Übersichtlichkeit geht. Dies ist sicher so gewünscht<sup>58</sup>, muss jedoch hier negativ bewertet werden.

❖ Bewertung: 1 / 4

- Indikator: Qualität – Inhalt

---

<sup>54</sup> <http://www.aras.com/community/forums/> (Stand: 11.04.2012)

<sup>55</sup> <http://www.aras.com/plm-roadmap/> (Stand: 11.04.2012)

<sup>56</sup> <http://www.aras.com/support/documentation/> (Stand:19.04.2012)

<sup>57</sup> U.a. <http://www.aras.com/getting-started/videos-demos.aspx> (Stand:19.04.2012) und <http://www.aras.com/university/self-study.aspx> (Stand:19.04.2012)

<sup>58</sup> Grund ist der Anreiz zum kaufen einer vollständigen Version.

Die Dokumentation ist mit vielen Code-Beispielen gefüllt und gut beschrieben. Die angebotenen Materialien sind von hoher Qualität, decken mit Bildschirmfotos und Zusatzinformationen alle relevanten Bereiche der jeweiligen Aufgabe ab. Inhaltlich wird die Dokumentation als sehr gut bewertet.

❖ Bewertung: 8 / 8

❖ Bewertung: 14 / 18

- Erweiterung

- Indikator: Open-Source

Der Hauptquellcode liegt nicht offen, aber bei Einrichtung eines Demo-Systems werden dem Nutzer zahlreiche Quelltexte aus Oberfläche und Backend zur Verfügung gestellt, an denen man ablesen kann, wie das System funktioniert. Der Mixed-Source-Ansatz erlaubt die Vergabe von einem Punkt an dieser Stelle<sup>59</sup>.

❖ Bewertung: 1 / 2

- Indikator: Schnittstellen

Da die Software in vielerlei Hinsicht angepasst werden kann, kann sie nicht die eine Schnittstelle anbieten, sondern ermöglicht den Entwicklern Funktionen zu schreiben und diese dann nach außen durch das Protokoll SOAP verfügbar zu machen<sup>60</sup>. Es sind keine Schnittstellen vorgefertigt, sie müssen erstellt oder entsprechende gesucht werden. Die Standardisierung der Schnittstelle erlaubt eine gute Bewertung.

❖ Bewertung: 4 / 6

❖ Bewertung: 5 / 8

- Funktionsumfang

- Indikator: quantitative Einschätzung

Nach erstmaliger Installation können bereits Produkte, Teile, Dokumente, Arbeitsabläufe usw. im System gespeichert und über dieses kollaboriert werden. Der Softwareentwickler selbst nennt eine ganze Reihe von Funktionalitäten, die mit diesem Programm abgedeckt werden. Genannt werden u.a. BOM Management, Document Management, Engineering Change Management, Global Product Development, Lean Product Development, Manufacturing Process Planning, Product Analytics, Product Costing, Product Data Management, Product Engineering, Quality Planning, Requirements Management, Risk

---

<sup>59</sup> <http://www.aras.com/technology/open-source.aspx> (Stand: 19.04.2012)

<sup>60</sup> <http://www.aras.com/technology/model-based-soa.aspx> (Stand: 19.04.2012)

Management, Supply Chain Management und Systems Engineering<sup>61</sup>. Dieser Umfang ist als sehr gut zu bewerten.

❖ Bewertung: 6 / 6

- Benutzerfreundlichkeit

- Indikator: subjektive Einschätzung

Die Oberfläche des Aras Innovators legt keinen Fokus auf eine ästhetische Erscheinung. Sie ist mit Buttons überfrachtet, die kaum gebraucht werden oder funktionieren, aber immer sichtbar sind. Das Design orientiert sich an MS Windows 2000 und aus der Ordneransicht dieses Betriebssystems<sup>62</sup>. Dennoch lässt sich nach einiger Eingewöhnung die Oberfläche gut bedienen und man findet meist schnell, was man sucht.

❖ Bewertung: 4 / 6

- Support

- Indikator: Anbindung

Die Aras University<sup>63</sup> ist zentrale Anlaufstelle für alle Formen des Supports, bezahlt und kostenfrei, jedoch befindet sich, wie bereits erwähnt, die kostenlose Dokumentation auf verschiedenen Unterseiten verteilt. Zu erkennen ist ein aktiver Support, der gut zu finden ist, aber eben nur teilweise kostenfrei zu Verfügung steht. Diese Splittung erlaubt eine sehr gute Bewertung in Sachen Anbindung.

❖ Bewertung: 2 / 2

---

<sup>61</sup> <http://www.aras.com/solutions/> (Stand: 19.04.2012)

<sup>62</sup> <http://www.aras.com/screenshots/> (Stand: 19.04.2012)

<sup>63</sup> <http://www.aras.com/university/> (Stand: 19.04.2012)

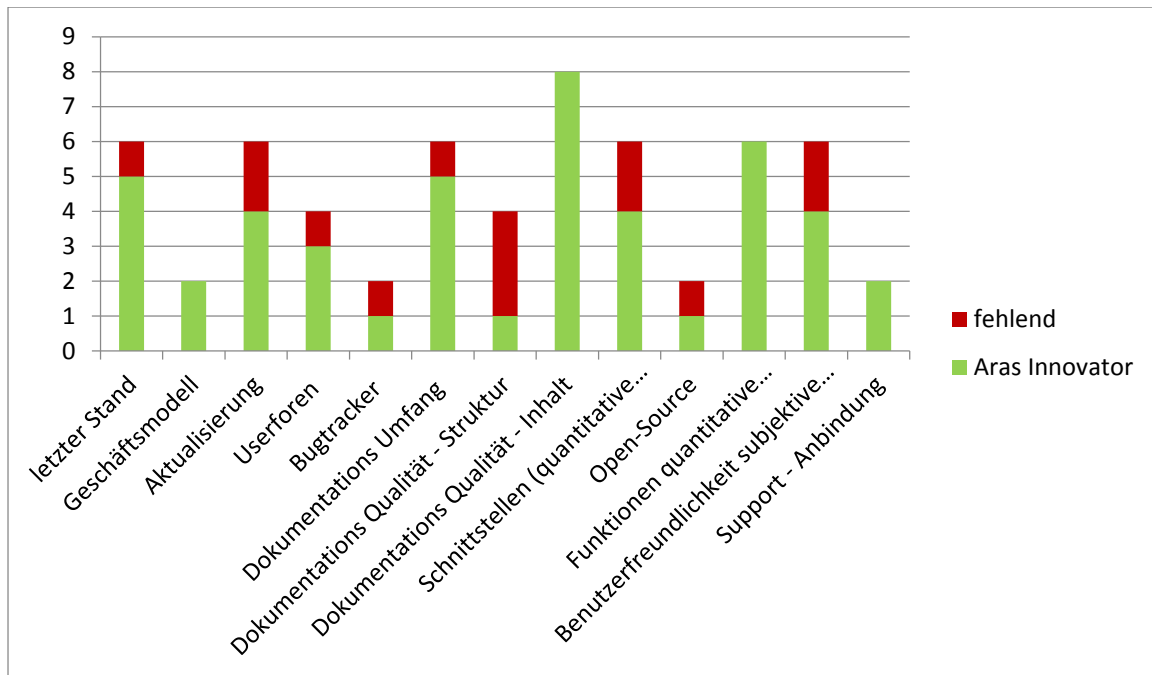


Abbildung 11 Bewertung Aras Innovator

### Architektur

Die Software benötigt einen Microsoft Windows Server mit IIS und Microsoft SQL Server sowie vergleichsweise ein hohes Maß an Hardware-Ressourcen und wird als MS-Installer geliefert. Die Server-Komponente wird in den IIS integriert. Clients können im Netzwerk ausschließlich über den Internet Explorer auf die Oberfläche zugreifen. Administratoren haben die Möglichkeit, innerhalb der Oberfläche alles anzupassen, was die Datenmodelle oder Prozesse angeht sowie Projekte, Nutzer und Teams zu verwalten<sup>64</sup>.

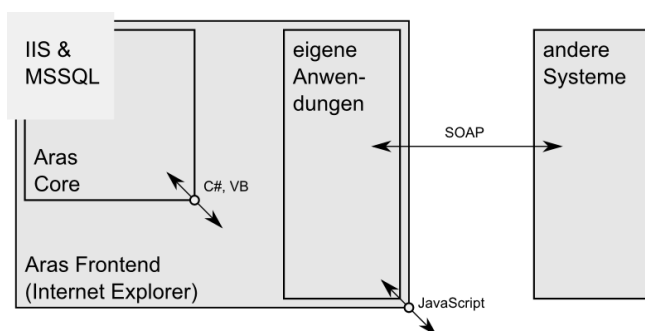


Abbildung 12 Architektur Aras Innovator

Entwickler können das System bearbeiten oder erweitern, indem sie (innerhalb der Oberfläche) bestehende, Server- wie Client-seitige Scripts bearbeiten oder neu anlegen können (JS für den Client, C#/VB/J# für den Server)<sup>65</sup>. Darüber können auch Skripte für Schnittstellen zu anderen Systemen oder für die eigene SOAP-

<sup>64</sup> <http://www.aras.com/support/documentation/9.3.0/Installation%20and%20Configuration/Aras%20Innovator%209.3%20-%20Installation%20Guide.pdf> (Stand: 19.04.2012)

<sup>65</sup> <http://www.aras.com/support/documentation/9.3.0/Other%20Documents/Aras%20Innovator%209.3%20-%20Programmers%20Guide.pdf> (Stand: 19.04.2012)

Schnittstelle geschrieben und konfiguriert werden<sup>66</sup>.

Der Aras Innovator speichert seine Datenbankdaten im Microsoft SQL Server und die Datendateien in einem sog. "Aras Vault", d.h. einem Ordner, der auf beliebige Laufwerke gelegt werden kann. Zur Sicherung können mehrere Speicherorte angegeben werden, um Redundanz zu erlauben.

#### Fazit

Der Aras Innovator überzeugt vor allem durch seinen hohen mitgelieferten Funktionsumfang, der einfachen Installation/ Einrichtung und der vergleichsweisen einfachen Integration mit vorhandenen Microsoft-Diensten in einem Unternehmen (z.B. Sharepoint, Reporting Services). Die Benutzeroberfläche ist zunächst erschlagend, aber nach kurzer Einarbeitung handhabbar. Auffällig negativ ist die Bindung an den Internet Explorer und die verstreute Dokumentation sowie die hohe Ressourcenanforderungen der Software.

#### Oberon Plattform

Die Oberon Plattform ist eine Java-Freeware, die vom Entwickler unter einer eigenen Lizenz kostenlos genutzt und weitergegeben werden kann<sup>67</sup>. Laut eigenen Angaben, bietet es ein leistungsfähiges Werkzeug zur Verwaltung für „Business Objects“<sup>68</sup> und deren Informationsflüsse. Die Software kann u.a. den Lebenszyklus für „Domain Objects“ verwalten und ermöglicht das Verwalten von Nutzprofilen<sup>69</sup>.

#### Bewertung

- Reife

- Indikator: aktueller Stand

Die Software liegt seit 01/2012 in Version 2.6 vor, Stand April 2012. Es handelt sich um eine relative junge Software, die keine Community vorweisen kann.

❖ Bewertung: 3 / 6

- Indikator: Geschäftsmodell

Der Autor nennt keine Kunden und es ist kein Geschäftsmodell erkennbar.

---

<sup>66</sup> <http://www.aras.com/technology/model-based-soa.aspx> (Stand: 19.04.2012)

<sup>67</sup> <http://www.oberonplattform.com/index.php?show=false> (Stand: 19.04.2012)

<sup>68</sup> Business Objects (Geschäftsobjekte) sind Objekte in einer objektorientierten Anwendung, welche die Einheiten eines Geschäftsbereichs (Business Domain) bzw. der geschäftlichen Welt repräsentieren. Diese bestehen nicht nur aus Daten, sondern auch aus Verarbeitungslogiken.

<sup>69</sup> <http://www.oberonplattform.com/purpose.php> (Stand: 19.04.2012)



❖ Bewertung: 0 / 2

❖ Bewertung: 3 / 8

- Weiterentwicklung

- Indikator: Aktualisierungen

In einer Stichprobe wurden letzte Aktivitäten im Januar festgestellt. Eine Komponente wurde seit 2011 nicht mehr aktualisiert, Stand April 2012. Da es sich um Closed-Source-Software handelt, steht kein Code-Repository zur Verfügung und entsprechende Transparenz ist nicht gegeben. Der Change Log verweist jedoch auf regelmäßige Aktualisierungen<sup>70</sup>, welche min. vierteljährlich vollzogen werden.

❖ Bewertung: 4 / 6

- Indikator: Userforen

Im Forum gab es in 3 Monaten keinerlei Aktivitäten und kaum bis keine Einträge<sup>71</sup>.

❖ Bewertung: 0 / 4

- Indikator: Bugtracker

Der Bugtracker wird zum überwiegenden Teil nur von zwei Nutzern verwendet, einer davon ist der Autor.

❖ Bewertung: 1 / 2

❖ Bewertung: 5 / 12

- Dokumentation

- Indikator: Umfang

Es gibt Tutorials für die erstmalige Einrichtung sowie für bestimmte Aufgaben wie Konfiguration oder wiederkehrende Arbeiten mit der Oberfläche<sup>72</sup>. Außerdem JavaDocs zu den verwendeten Klassen und API-Dokumentation zur Verfügung<sup>73</sup>. Auch ein FAQ und verschiedene Videotutorials stehen zur Verfügung<sup>74</sup>.

❖ Bewertung: 6 / 6

- Indikator: Qualität – Struktur

Die Struktur ist übersichtlich und zentral gelegen<sup>75</sup>.

---

<sup>70</sup> [http://www.oberonplatform.com/bugtrace/changelog\\_page.php](http://www.oberonplatform.com/bugtrace/changelog_page.php) (Stand: 19.04.2012)

<sup>71</sup> <http://www.oberonplatform.com/obforum/index.php> (Stand: 19.04.2012)

<sup>72</sup> <http://www.oberonplatform.com/install.php> (Stand: 19.04.2012)

<sup>73</sup> <http://www.oberonplatform.com/javadoc.php> (Stand: 19.04.2012)

<sup>74</sup> <http://www.oberonplatform.com/tutorial.php> (Stand: 19.04.2012)

<sup>75</sup> <http://www.oberonplatform.com/documents.php> (Stand: 19.04.2012)

❖ Bewertung: 4 / 4

- Indikator: Qualität – Inhalt

Für Anwendungsentwickler gibt es eine gute Dokumentation, mit zahlreichen Bildschirmfotos und Codebeispielen. Negativ ist die nur mäßige Dokumentation für Einrichtung und Inbetriebnahme der Server-Funktionalitäten. So werden zum Beispiel einige Bereiche, wie auch eine grobe Übersicht über die Schritte, die nötig sind zur Bereitstellung von Oberon im Netzwerk, nicht ausreichend dokumentiert – ein Mangel der sich erst in der praktischen Anwendung zeigt. Auch werden keine weiterführenden Informationen gegeben oder auf häufige Fehler hingewiesen. Trotz des sehr guten Umfangs der Dokumentation kann inhaltlich dieser Standard nicht gehalten werden. Besonders kritisch ist, wenn die Dokumentation an so sensiblen Punkten, wie der Inbetriebnahme des Systems, nicht ausreichend abgedeckt ist. Daher gibt es an dieser Stelle nur eine mäßige Bewertung.

❖ Bewertung: 4 / 8

- ❖ Bewertung: 14 / 18

- Erweiterung

- Indikator: Open-Source

Der Kern ist Closed-Source, aber man kann die Quelltexte einer Beispielapplikation auf Basis der Plattform herunterladen, die tiefe Einblicke in die Funktionsweise von Oberon liefern. Clients und Webapplikationen sind Open-Source – laut eigenen Angaben<sup>76</sup>.

❖ Bewertung: 1 / 2

- Indikator: Schnittstellen

Für Schnittstellen werden SOAP- und XML-Prozessoren mitgeliefert, jedoch muss jede Anbindung händisch programmiert werden<sup>77</sup>. D.h. der mitgelieferte Designer lässt Schnittstellen anlegen, benennen und Funktionen sowie Variablen zuordnen. Diese Funktionen benötigten in jedem Falle Java zur Ausfüllung. Die standardisierten Schnittstellen erlauben eine gute Bewertung.

❖ Bewertung: 4 / 6

- ❖ Bewertung: 5 / 8

- Funktionsumfang

- Indikator: quantitative Einschätzung

---

<sup>76</sup> Hier gibt es leider keine weiteren Informationen: <http://www.oberonplatform.com/downloads.php> (Stand: 13.04.2012)

<sup>77</sup> <http://www.oberonplatform.com/architecture.php> (Stand: 19.04.2012)

Die Bandbreite der Funktionen wird nicht spezifiziert. Klar ist nur, dass ein Life-Cycle-Management (für Business Objects) mit einem entsprechenden Nutzersystem und Nutzerverwaltung implementiert werden kann<sup>78</sup>. Um die Funktionalitäten testen zu können, wird eine vorkompilierte Demo-Anwendung “PLM” zum Download angeboten<sup>79</sup>. Da diese jedoch mit einer älteren Version von Oberon kompiliert wurde, kann sie nicht gestartet werden. Der Funktionsumfang kann durch diesen Faktor nur als mittelmäßig eingeschätzt werden.

❖ Bewertung: 3 / 6

- Benutzerfreundlichkeit

- Indikator: subjektive Einschätzung

Die Plattform verfügt über keine Oberfläche, da sie nur ein Framework für die Entwicklung von Oberflächen und Anwendungen zur Verfügung stellt. Die Werkzeuge zur Erstellung und Verwaltung der Applikationen auf Basis von Oberon sind jedoch sehr übersichtlich, einfach bedienbar, bieten ein großes Funktionsspektrum und basieren auf dem Eclipse Framework.

❖ Bewertung: 5 / 6

- Support

- Indikator: Anbindung

Ein Support ist aktiv eingebunden, jedoch kostenfrei kaum genutzt – was sich bereits in anderen Indikatoren (Userforum) widerspiegelt und hier nicht wieder mit aufgenommen wird<sup>80</sup>.

❖ Bewertung: 2 / 2

---

<sup>78</sup> <http://www.oberonplatform.com/features.php> (Stand:19.04.2012)

<sup>79</sup> [http://www.oberonplatform.com/casestudies\\_plm.php](http://www.oberonplatform.com/casestudies_plm.php) (Stand:19.04.2012)

<sup>80</sup> <http://www.oberonplatform.com/support.php> (Stand:19.04.2012)

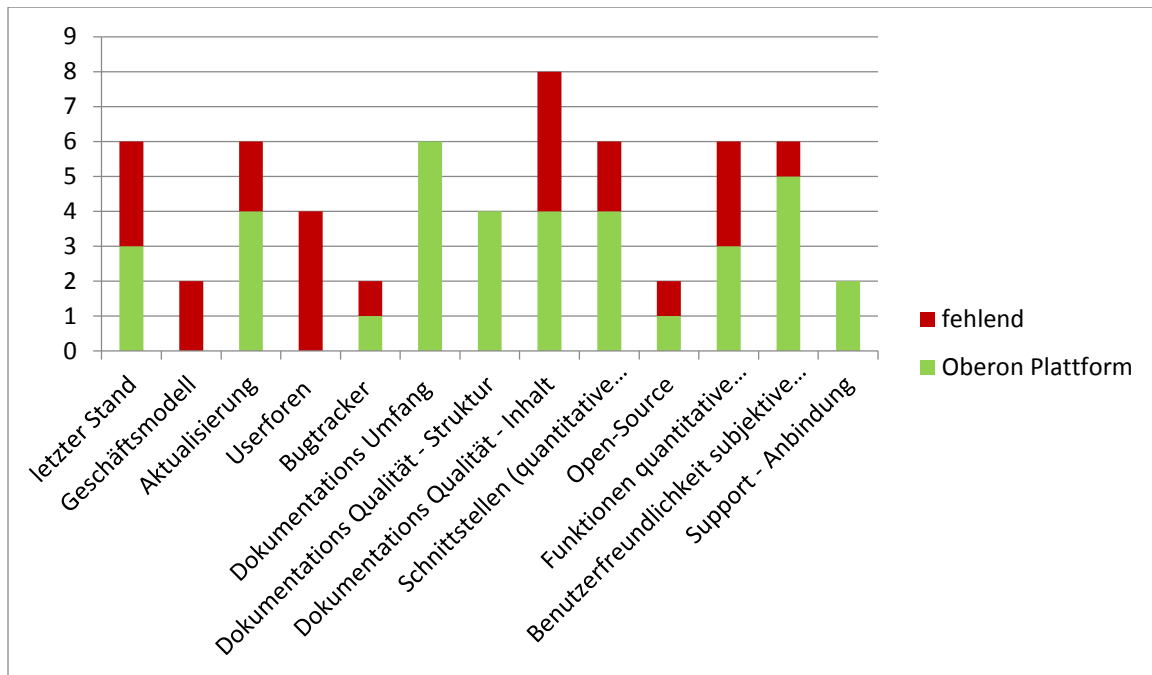


Abbildung 13 Bewertung Oberon Plattform

Architektur

Die Oberon Plattform stellt eine Entwicklungsumgebung für Java-Programme dar, die zum einen eine Server-Client-Hierarchie aufweisen und zum anderen bestimmte wiederkehrende Elemente – wie zum Beispiel Businessobjekte, Dateien, Benutzer, Workflows u.ä. – verwendet<sup>81</sup>. Der Kern von Oberon ist eine Laufzeitumgebung für die zahlreichen bereitgestellten Funktionen. An sich ist dies noch kein Server, anders ausgedrückt: die Umgebung hat keine Server-Funktion.

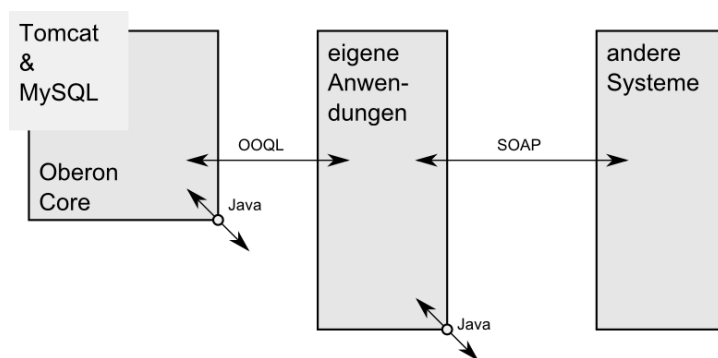


Abbildung 14 Architektur Oberon Plattform

Man kann jedoch (mit den mitgelieferten Werkzeugen) eine Applikation auf Basis von Oberon entwickeln. Hierzu wird der Oberon-Kern zu einer Jar-Bibliothek hinzugefügt und mit der Anwendung auf einem Java-Anwendungsserver (bsw. Apache Tomcat) bereitgestellt. Schnittstellen zu anderen

Softwarewerkzeugen kann man im mitgelieferten Designer entwerfen, und danach ausprogrammieren<sup>82</sup>.

<sup>81</sup> <http://www.oberonplatform.com/features.php> (Stand: 19.04.2012)

<sup>82</sup> <http://www.oberonplatform.com/configure.php> (Stand: 19.04.2012)

Oberon ist in Java geschrieben, was Betriebssystemunabhängigkeit bedeutet. Bei einer Probeinstallation wurde jedoch festgestellt, dass der Einrichtungs- und Wartungsaufwand bei Windows-Systemen wesentlich höher als bei Unix-System ist<sup>83</sup>.

Klassischerweise wird Oberon mit MySQL oder Postgres eingerichtet, kann jedoch auch mit OracleDB betrieben werden, laut Website<sup>84</sup>. Man kann bei der Einrichtung/ Entwicklung einen Dateispeicherort frei wählen (für eventuelle Dokumente und einen separaten für Caching), und die Tabellen und variablen Daten werden in der gewählten Datenbank gespeichert. Die Software stellt für die Abfrage und Manipulation der Businessobjekte sowie deren Relationen eine eigene Abfragesprache "OOQL" zur Verfügung<sup>85</sup>. Diese wurde jedoch nicht näher untersucht.

### Fazit

Oberon bietet eine sehr gute Dokumentation für Anwendungsentwickler, welche klar strukturiert und gut zugänglich ist. Darüber hinaus wird ein gutes Werkzeug („Designer“) als Unterstützung der Anwendungsentwicklung mitgeliefert, welches den praktischen Einsatz erleichtert. Weiterhin positiv zu erwähnen ist die Unabhängigkeit der Software gegenüber dem Betriebssystem, des Browsers und der Datenbank<sup>86</sup>. Negativ fällt auf, dass wenige „out-of-the-box“-Funktionen mitgeliefert werden. Selbst für eine statische Website mit Benutzeranmeldung ist Einarbeitung und nennenswerter Programmieraufwand nötig. Weiterhin fallen die wenigen Nutzer und das inaktives Forum negativ auf.

### Gegenüberstellung

Die bisherigen Ausarbeitungen lassen sich wie folgt zusammenfassen: Drei Software-Lösungen wurden mittels der vorgestellten Kriterien als gut befunden. In der Gesamtbewertung hat der Aras Innovator die besten Bewertungen bekommen. Darüber hinaus schneidet der Aras Innovator in 4 von 7 Kriterien besser ab, als die anderen vorgestellten Softwarelösungen. Auch wenn einzelne Facetten der anderen Lösungen beim Aras Innovator wünschenswert wären (Plattformunabhängigkeit, kostenfreie Dokumentation), so überzeugt die Reife und der Funktionsumfang der Software. Diese Einschätzungen spiegeln sich auch in der Tabelle 2 und Abbildung 7 wider.

		Aras	Oberon	openPLM
Reife	letzter Stand	<b>83%</b>	50%	50%
	Geschäftsmodell	<b>100%</b>	0%	50%

<sup>83</sup> <http://www.oberonplatform.com/architecture.php> (Stand:19.04.2012)

<sup>84</sup> <http://www.oberonplatform.com/install.php> (Stand: 19.04.2012)

<sup>85</sup> <http://www.oberonplatform.com/ooql.php> (Stand: 19.04.2012)

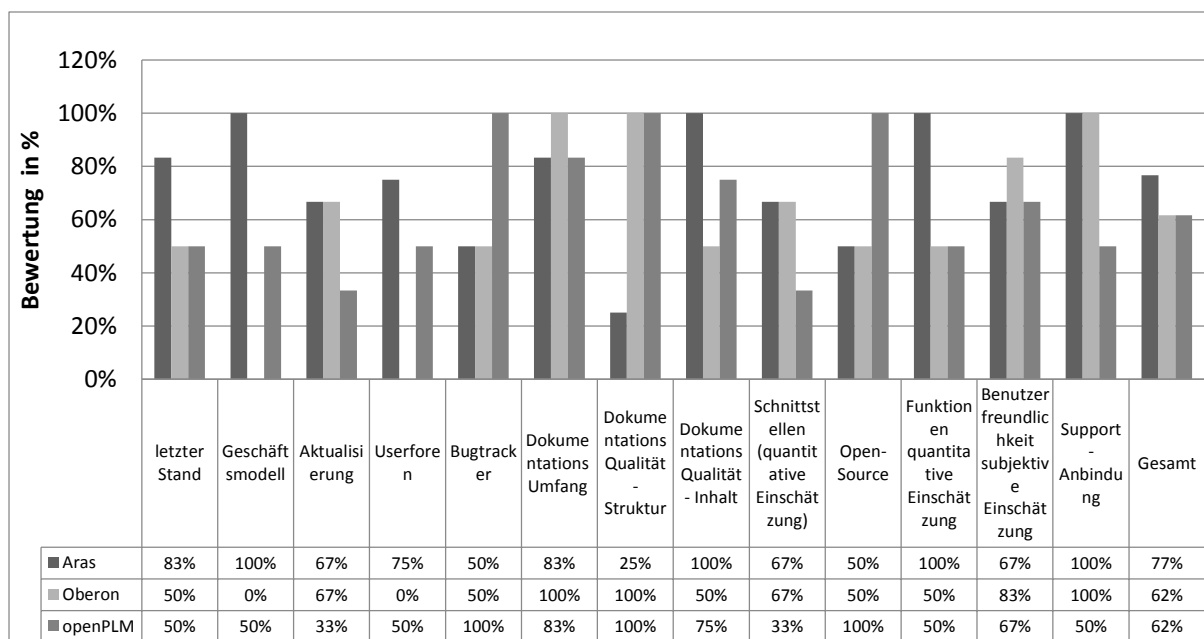
<sup>86</sup>Theoretisch sind viele andere Datenbanksysteme möglich, solange diese JDBC unterstützt.

	<u>Reife gesamt</u> <sup>87</sup>		<b>88 %</b>	38%	50%
Weiter- entwicklung	Aktualisierung		<b>67%</b>	<b>67%</b>	33%
	Userforen		<b>75%</b>	0%	50%
	Bugtracker		50%	50%	<b>100%</b>
	<u>Weiterentwicklung gesamt</u>		<b>67%</b>	42%	50%
Dokumentation	Dokumentationsumfang		83%	<b>100%</b>	83%
	Dokumentationsqualität - Struktur		25%	<b>100%</b>	<b>100%</b>
	Dokumentationsqualität - Inhalt		<b>100%</b>	50%	75%
	<u>Dokumentation gesamt</u>		77 %	78%	<b>83%</b>
Erweiterbarkeit	Schnittstellen (quantitative Einschätzung)		<b>67%</b>	<b>67%</b>	33%
	Open-Source		50%	50%	<b>100%</b>
	<u>Erweiterbarkeit gesamt</u>		<b>63%</b>	<b>63%</b>	50%
Funktionsumfang	Funktionen quantitative Einschätzung		<b>100%</b>	50%	50%
Benutzerfreund- lichkeit	Benutzerfreundlichkeit Einschätzung	subjektive	67%	<b>83%</b>	67%
Support	Support - Anbindung		<b>100%</b>	<b>100%</b>	50%
	<b>Gesamt</b>		<b>77%</b>	62%	62%

In der Reife überzeugt Aras durch ein überzeugendes Geschäftsmodell, mit entsprechenden Kunden und einer weit entwickelten Version. Die anderen Kandidaten schneiden hier unterdurchschnittlich bis durchschnittlich ab – wobei die Oberon Plattform eine sehr mangelhafte Reife besitzt durch das fehlende Geschäftsmodell, welches für eine nachhaltige Entwicklung unumgänglich ist. Weiterhin schneidet der Aras Innovator, durch seine sehr regelmäßigen Aktualisierungen und des guten Userforums im Bereich der „Weiterentwicklung“, am besten ab. Obwohl kein Bugtracker – wie vorbildlich bei openPLM zu finden ist – implementiert ist, kann eine gute Bewertung erreicht werden und grenzt sich gegenüber der Konkurrenz durch die beiden Indikatoren „Aktualisierung“ und „Userforum“ ab. Diese haben erheblichen Nachholbedarf in beiden Bereichen. Das Kriterium „Dokumentation“ konnte die Oberon

<sup>87</sup> Die Zahlen entstehen nicht aus den Mittelwerten der Prozente, sondern aus den erreichten Punkten für das gesamte Kriterium.

Plattform für sich entscheiden, dicht gefolgt von openPLM; beide erhielten eine sehr gute Bewertung, da die Dokumentation nicht nur umfangreich ist, sondern darüber hinaus auch gut strukturiert und kostenfrei. Der Aras Innovator kann hier nur eine gute Bewertung erhalten, da eine gute Struktur der Dokumentation nicht gegeben ist bzw. erkaufte werden muss. Bei der Bewertung für das Kriterium der Erweiterbarkeit liegen Oberon und Aras auf einer Ebene, welche als gut bezeichnet werden kann. Die Erweiterbarkeit ist eingeschränkt durch den Mixed-Source-Ansatz und durch die teils aufwendige Implementierung. OpenPLM erreicht durch eine durchschnittliche Bewertung aufgrund seiner Spezialisierung auf Schnittstellen zu anderen Open-Source Projekten. Im Bereich des mitgelieferten Funktionsumfangs konnte Aras absolut überzeugen, die Liste der Features ist sehr lang und liefert so eine sehr umfangreiche Bandbreite von Funktionalitäten, mit der die anderen Kandidaten nicht mithalten können. In der Kategorie Benutzerfreundlichkeit konnte die Oberon Plattform durch seinen guten Designer Punkte sammeln. Die anderen beiden Kandidaten sind als benutzbar eingestuft worden, aber nicht überdurchschnittlich gut. Die Supportanbindung ist bei Oberon und bei Aras optimal und nur bei openPLM Durchschnitt.



## Quellenverzeichnis

<http://www.oberonplatform.com/> (Stand: 04.06.2012)

<http://www.aras.com/> (Stand: 04.06.2012)

<http://sourceforge.net/> (Stand: 04.06.2012)

<http://www.plmportal.de> (Stand: 04.06.2012)

<http://www.ranchbe.com/> (Stand: 04.06.2012)

<http://info.tiki.org/> (Stand: 04.06.2012)

<http://ns37734.ovh.net/~plmpdm/ranchbe-demo/accueil.php> (Stand: 04.06.2012)

<http://www.xinco.org/> (Stand: 04.06.2012)

<http://www.semager.de/> (Stand: 04.06.2012)

<http://www.openplm.org/> (Stand: 04.06.2012)

<http://www.linobject.com/> (Stand: 04.06.2012)

<http://celeryproject.org/> (Stand: 04.06.2012)

<http://haystacksearch.org/> (Stand: 04.06.2012)

<https://github.com/notanumber/xapian-haystack> (Stand: 04.06.2012)

<http://www.graphviz.org/> (Stand: 04.06.2012)



## Kapitel 4 Abbildung von Dienstleistungen im Aras Innovator (*Christian Zinke*)

### Einleitung

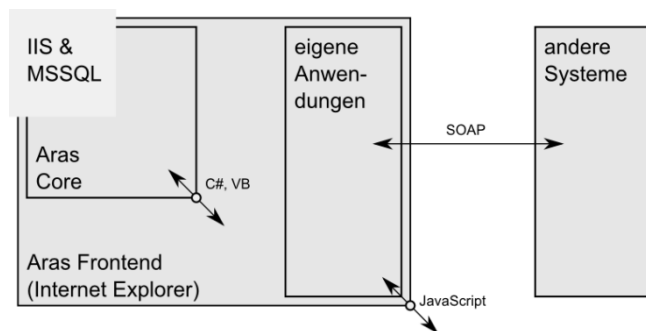
In diesem Kapitel wird zunächst auf die generelle Architektur<sup>88</sup> des Aras Innovators eingegangen, das schließt u.a. die Systemvoraussetzung und Erweiterungsmöglichkeiten ein. Im Anschluss daran werden die Grundbausteine des Aras Innovators vorgestellt, darunter zählt der „ItemType“, die „Workflow Maps“ und das LifeCycle-Konzept. Diese drei Bausteine bilden aus der Sicht des Autors die Basis, um das ausgearbeitete Konzept einer technischen Lösung für das Life-Cycle-Management produktbezogener Dienstleistungen und dessen angestrebte Funktionen in das PLM-Unterstützungssystem Aras Innovator zu implementieren und ein PDLM-Unterstützungssystem zu schaffen. Entsprechend diesem Programm folgt das Kapitel zur generellen Architektur des Aras Innovators.

### Architektur des Aras Innovator

#### Generelle Architektur

Der Aras Innovator wird mit MS-Installer geliefert und benötigt einen Microsoft Windows Server mit IIS und Microsoft SQL Server sowie vergleichsweise ein hohes Maß an Hardware-Ressourcen. Die Serverkomponente wird in den IIS integriert. Clients können im Netzwerk ausschließlich über den Internet Explorer auf die Oberfläche zugreifen. Der Aras Innovator speichert seine Datenbankdaten im Microsoft SQL Server und die Datendateien in einem sog. „Aras Vault“, d.h. einem Ordner, der auf beliebige Laufwerke gelegt werden kann. Zur Sicherung können mehrere Speicherorte angegeben werden, um Redundanz zu erlauben.

Administratoren haben die Möglichkeit innerhalb der Oberfläche sowohl alles anzupassen, was die Datenmodelle oder Prozesse betrifft, als auch Projekte, Nutzer und Teams zu verwalten<sup>89</sup>.



Entwickler können das System bearbeiten oder erweitern, indem sie (innerhalb der Oberfläche) bestehende, server- wie clientseitige Scripts bearbeiten oder neu anlegen (JS für den Client, C#/VB/J# für den

Abbildung 15 Architektur Aras Innovator

<sup>88</sup> Diese stellt eine Zusammenfassung und Erweiterung zu Kapitel 3 dar.

<sup>89</sup> <http://www.aras.com/support/documentation/9.3.0/Installation%20and%20Configuration/Aras%20Innovator%209.3%20-%20Installation%20Guide.pdf> (Stand: 19.04.2012)

Server)<sup>90</sup>. Darüber können auch Skripte für Schnittstellen zu anderen Systemen oder für die eigene SOAP-Schnittstelle geschrieben und konfiguriert werden<sup>91</sup>. Diese flexible Lösung erlaubt es Modelle und Funktionen clientseitig zu implementieren und den Programmkern unberührt zu lassen. Die Implementierung eines PDLM-Konzepts kann so größtenteils durch die modifizierte Nutzung der Kernelemente realisiert werden. Dahinter liegt das Aras Innovator Konzept: „Everything is an Item“. Dieses Konzept findet sich auch in der Kommunikation zwischen Client und Server wieder, denn diese Kommunikation findet nur durch den XML-Dialekt AML (Adaptive Markup Language) statt. Diese Kern- oder Grundbausteine werden im Folgenden vorgestellt.

### Grundbausteine des Aras Innovators

Die Grundbausteine des Aras Innovators sind die „ItemTypes“, die „Workflow Maps“ und das LifeCycle-Konzept. Alle Bausteine sind clientseitig im Aras Innovator Admin Menü zu finden (siehe Abbildung 16) und können dort eingesehen werden, bzw. neu instanziiert (als „Items“) werden. Diese Bausteine können entsprechend der Architektur auch in AML abgebildet werden.

#### ItemType

Der „ItemType“ ist der kleinste Baustein des Aras Innovators. Er ist selbstreferentiell und entsprechend in der Liste aller ItemTypes zu finden (siehe Abbildung 17). Durch diese Eigenschaften bildet der „ItemType“ das grundlegendste Element, so sind auch „Workflow Maps“ und „Life Cycle Maps“ spezielle „ItemTypes“. „ItemType“ wird in AML wie folgt abgebildet:

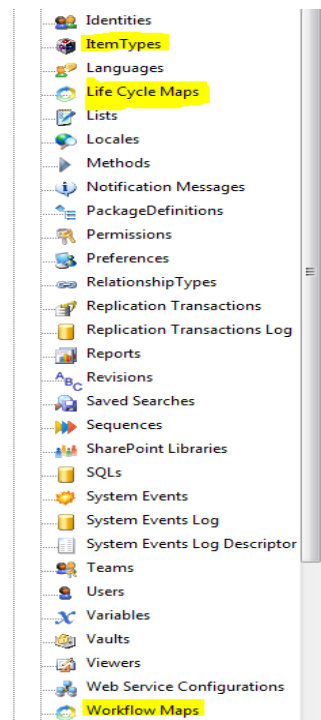


Abbildung 16 Admin Menü

```
<AML>
  <Item type="ItemType" id="SomeEIdHereIn">
    ...
  </Item>
  ...
</AML>
```

<sup>90</sup><http://www.aras.com/support/documentation/9.3.0/Other%20Documents/Aras%20Innovator%209.3%20-%20Programmers%20Guide.pdf> (Stand: 19.04.2012)

<sup>91</sup><http://www.aras.com/technology/model-based-soa.aspx> (Stand: 19.04.2012)

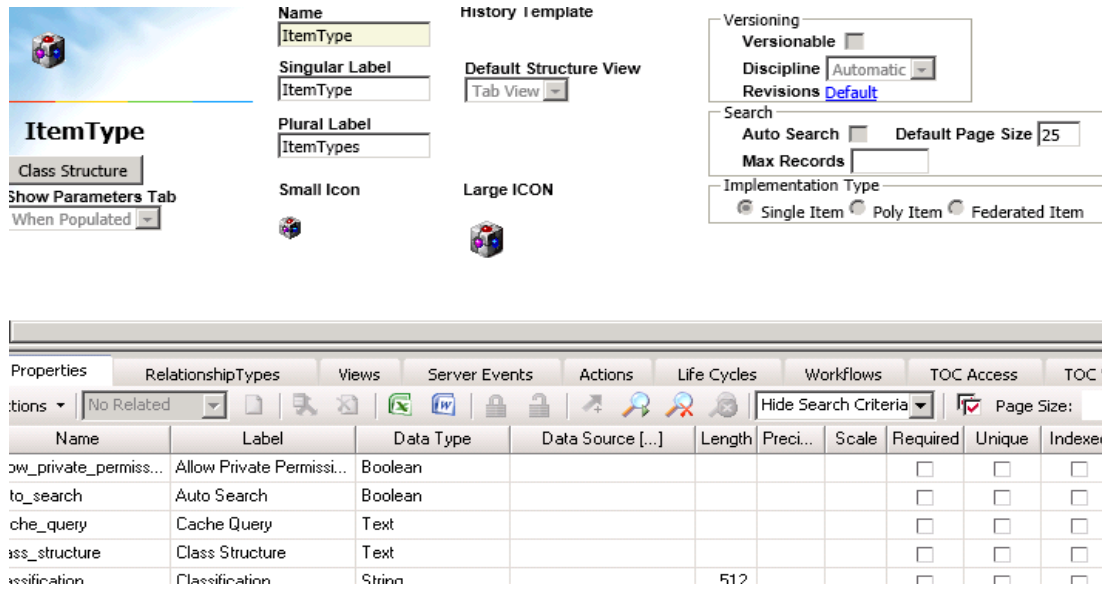


Abbildung 17 Screenshot ItemType: ItemType

Der „ItemType“ ist ein Objekt, welches angelegt und bearbeitet werden kann. Er besitzt eine Reihe von Eigenschaften (Properties) und verschiedene Relationen, welche u.a. bei der Erstellung eines neuen „ItemTypes“ zu beachten sind. Diese Relationen können als Klassendiagramm abstrahiert werden, wie in Abbildung 18 geschehen.

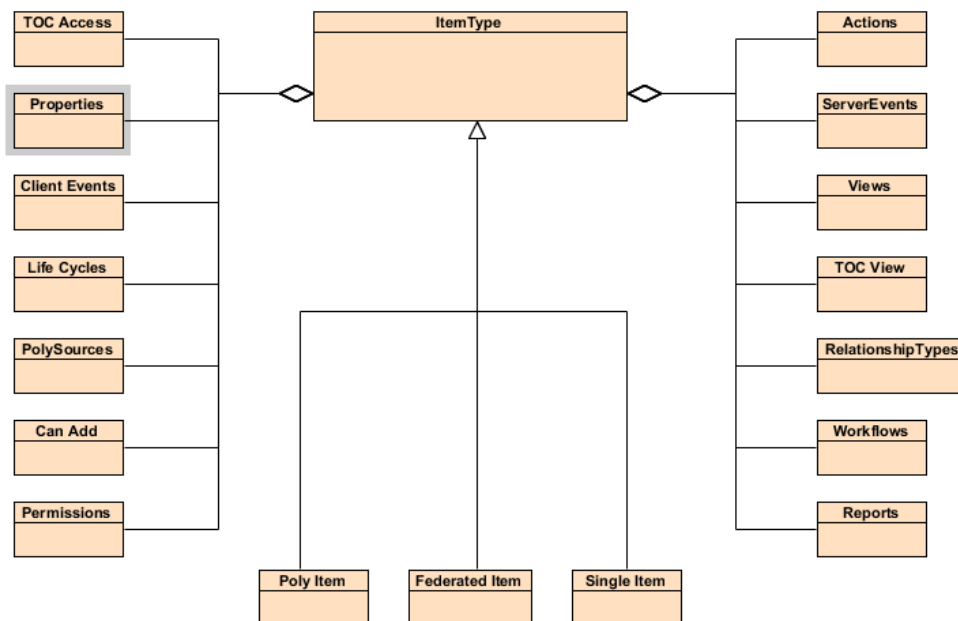


Abbildung 18 Abstrahiertes Klassendiagramm ItemType

Neben diesen Verknüpfungen gibt es, wie schon erwähnt, eine ganze Reihe von Eigenschaften. Eine Eigenschaft ist als besonders herauszustellen, weil es die Art eines neuen Objektes des Typs „ItemType“

bestimmt. Es handelt sich um die sogenannte Implementierungsspezifikation, welche im nächsten Abschnitt erläutert wird. Weiterhin wichtig erscheint die Möglichkeit der Versionierung von „ItemTypes“, eine kurze Erläuterung hierzu folgt im weiteren Verlauf.

ItemType Spezifikation (Implementierungsspezifikation)

Die Implementierungsarten der „ItemTypes“ können beim Aras Innovator noch spezifiziert werden. Normale „ItemTypes“ sind „Single Items“, es gibt aber auch andere Möglichkeiten (siehe Abbildung 19).

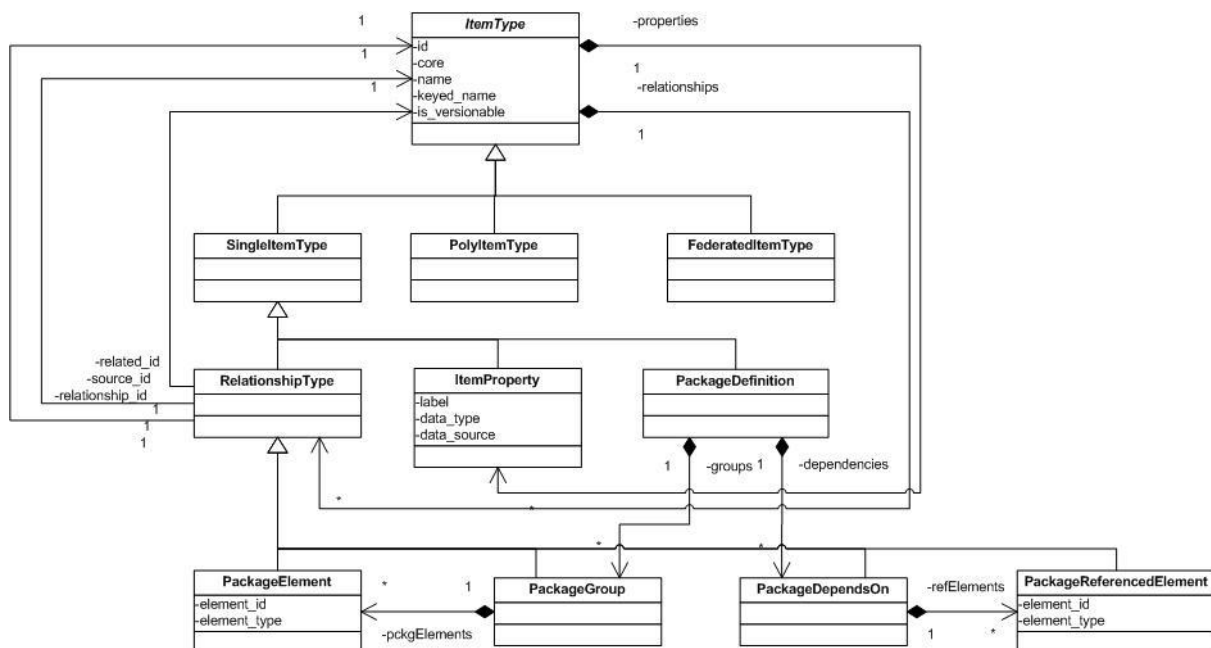


Abbildung 19 ItemType Spezifikation<sup>92</sup>

Diese Möglichkeiten werden in der nachfolgenden Tabelle kurz erläutert.

Objekt	Beschreibung
Single Item	Das „Single Item“ ist das Standard Geschäftsobjekt, das gesetzt wird. Es wird in einer Datenbank durch eine einfache Tabelle repräsentiert.
Federated Item	Es handelt sich dabei um ein normales Objekt, dass aber Eigenschaften besitzen kann, welche von externen Quellen stammen (beispielsweise CAD-Daten) <sup>93</sup> .

<sup>92</sup> [http://www.aras.com/Community/cfs-filesystemfile.ashx/\\_key/CommunityServer.Discussions.Components.Files/7/8284.Abstract-UML-Model\\_5F00\\_v1.1.jpg](http://www.aras.com/Community/cfs-filesystemfile.ashx/_key/CommunityServer.Discussions.Components.Files/7/8284.Abstract-UML-Model_5F00_v1.1.jpg) (Stand: 11.12.2012)

<sup>93</sup> „Innovator uses the term ‘Federated’ to mean data shared between different information systems.“ (http://www.aras.com/Community/wikis/kb/how-to-use-federation.aspx; Stand: 23.11.2012)

Poly Item	Hierbei handelt es sich um eine sogenannte „polymorphic“ Klasse. Sie repräsentiert eine Zusammenstellung von anderen ItemTypes. Diese Klasse hat viele Limitierungen gegenüber den anderen Klassen und ist, einmal eingestellt, nicht rückgängig zu machen.
-----------	---

Versionierung von ItemTypes

Mit den Eigenschaften „Versionable Enable“ und „Manual Versioning“ kann eine Versionierung von bestimmten „ItemTypes“ sichergestellt werden. Dabei beinhaltet die Eigenschaft „Versionable Enable“ eine Änderungsverfolgung von Veränderungen an bestimmten Items durch eine Generationierung und die Eigenschaft „Manual Versioning“ eine manuelle Änderungsverfolgung von Veränderungen<sup>94</sup>.

Workflow Maps

Wie im vorherigen Abschnitt beschrieben, können für jeden „ItemType“ „Workflow Maps“ hinterlegt werden. „Workflow Maps“ beschreiben ein einfaches Prozessmodell mit drei Bausteinen: den Prozessvariablen (Process Variables), den Aktivitäten (Activity) und den Pfaden (Path). Aus diesen Bausteinen können komplexe Prozesse modelliert werden (Beispiel in Abbildung 20).

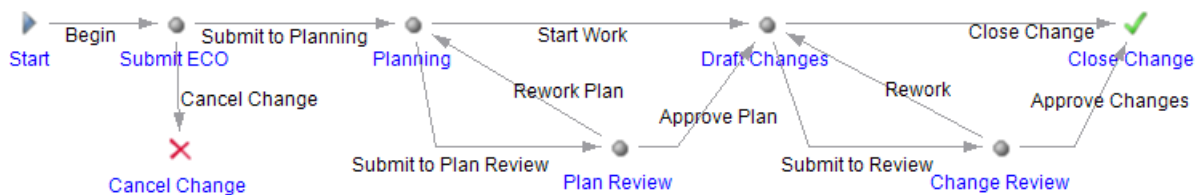


Abbildung 20 Beispiel Workflow

Eine „Workflow Map“ ist als ein spezieller „ItemType“ modelliert und kann entsprechend angepasst werden (siehe Abbildung 21).

<sup>94</sup> Siehe [http://www.aras.com/University/SelfHelpGuides/Basic\\_Admin\\_Navigation.pdf](http://www.aras.com/University/SelfHelpGuides/Basic_Admin_Navigation.pdf) (Stand: 07.12.2012)

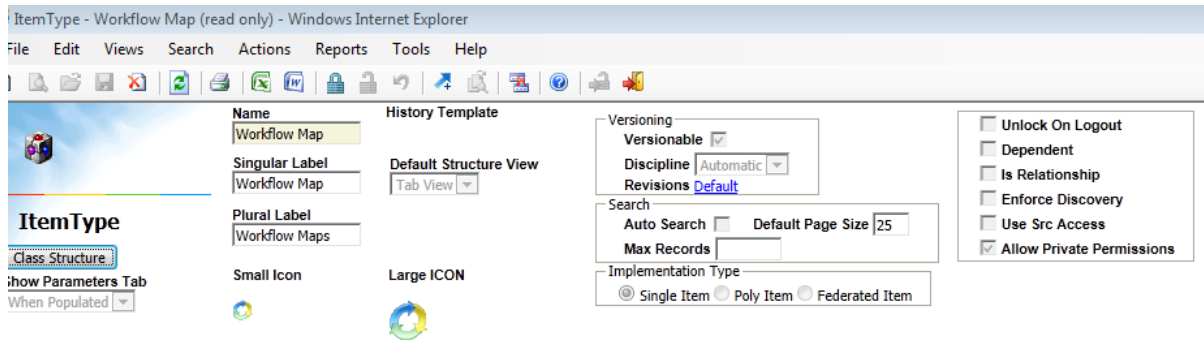


Abbildung 21 ItemType Workflow Map

Darüber hinaus besitzt eine „Workflow Map“ spezifische Eigenschaften: einen Namen, ein Label, eine Beschreibung und einen Besitzer (Process Owner). Da „Workflow Maps“ auf Aktivitäten beruhen, sollen darüber hinaus noch deren Eigenschaften thematisiert werden. In der folgenden Tabelle werden diese dargestellt:

Wichtige Eigenschaften von Aktivitäten:

Objekt	Typ	Beschreibung
Name	String	Der Name der Aktivität.
Label	String	Definiert, wie diese Aktivität im Client erscheint.
Message	String	Hier wird die Nachricht für die ausführenden Personen definiert, wenn die Aktivität aufgerufen wird.
Expected Duration	Integer	Hier wird festgelegt, in welchem Zeitraum eine Fertigstellung, Beendigung der Aktivität erwartet wird (für die zeitliche Abfolge siehe Abbildung 7).
Timeout Duration	Integer	An dieser Stelle kann definiert werden, ab wann eine Eskalationszeit erreicht ist. Später können Methoden für den Event „onEscalate“ entsprechend zugeordnet werden (für die zeitliche Abfolge siehe Abbildung 7).
Reminder Interval	Integer	An dieser Stelle kann definiert werden, welcher Zeitraum zwischen den einzelnen Reminder Nachrichten liegen soll (für die zeitliche Abfolge siehe Abbildung 7).
Reminder Count	Integer	Die Zahl der Reminder, die gesendet werden sollen. In der Abbildung 22 sind $RC = 3$ und $RI = 1$ , wobei 1 für einen Tag steht.



Abbildung 22 Ablauf Activity<sup>95</sup>

Escalate To	Identity	Es kann definiert werden, welcher Nutzer oder Nutzergruppe im Falle einer Eskalation zuständig ist, diese ggf. zu beenden.
Managed By	Identity	Hier wird der Manager der Aktivität zugeordnet.
Role	Identity	Es kann darüber hinaus festgelegt werden, aus welchen Rollen der „Managed By“ gewählt werden kann.
Subflow	Workflow	Es kann für jede Aktivität ein Untergeordneter Workflow definiert werden, der ausgeführt werden muss.
Start Activity	Bool	Es wird definiert, ob es sich um eine Start Aktivität handelt.
End Activity	Bool	Es wird definiert, ob es sich um eine End-Aktivität handelt.
Automatic Activity	Bool	Es wird definiert, ob es sich um eine automatische Aktivität handelt.
Can Refuse	Bool	Definiert wird an dieser Stelle, ob die Aktivität zurückgewiesen werden darf.
Can Delegate	Bool	Definiert wird an dieser Stelle, ob die Aktivität delegiert werden darf.
Consolidate Delegated	Bool	Definiert wird an dieser Stelle, ob die Aktivität gemeinsam delegiert werden darf.
Wait For All Inputs	Bool	Definiert wird an dieser Stelle, ob die Aktivität erst beendet werden darf, wenn alle Inputs getätigt worden sind.
Wait For All Votes	Bool	Definiert wird an dieser Stelle, ob die Aktivität erst beendet werden darf, wenn alle Votes getätigt worden sind.

Die Aktivitäten haben wiederum Verknüpfungen (Abbildung 23) und Eigenschaften. So können bestimmte Aufgaben (Tasks) oder Benachrichtigungen (Notifications) mit einer Aktivität verbunden werden.

<sup>95</sup>

<http://www.aras.com/University/SelfHelpGuides/Print%20of%20Online%20Help%20-%20Aras%20Innovator%20Application%20Framework.pdf> (Stand 11.12.2012)

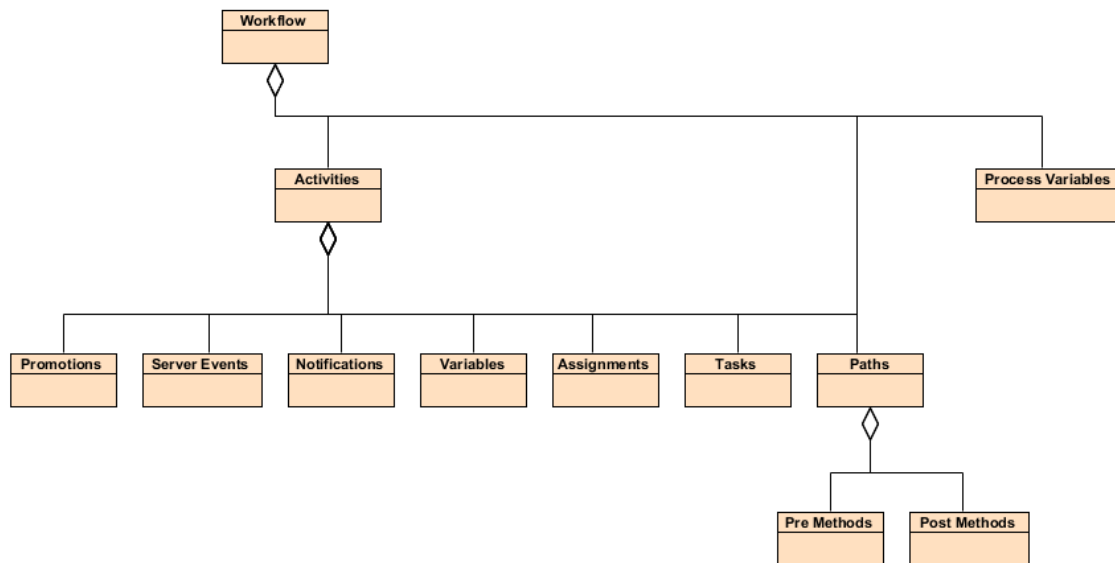


Abbildung 23 Verknüpfungen Workflow/Activities<sup>96</sup>

Das LifeCycle-Konzept

Wie auch die “Workflow Maps” sind die “Life Cycle Maps” als Prozesse modellierbar. Zu einem späteren Zeitpunkt wird auch auf eine dritte Form der Prozesse eingegangen: den Projekten. Jedoch gibt es Unterschiede bei den jeweiligen Formen der Prozesse. Diese Unterschiede können den entsprechenden Definitionen entnommen werden:

Definition: Ein Lebenszyklus ist eine Reihe von sich gegenseitig ausschließenden Zuständen eines Objektes, welche die Stadien (State) dieses Objektes repräsentiert. Darüber hinaus sind die Zustände durch bestimmte Übergänge – darunter auch Rückkopplung bzw. Schleifen – verbunden.

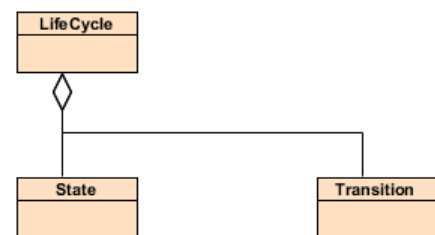


Abbildung 24 UML - LifeCycle

Im Vergleich dazu sind Workflows als ein Satz von Aktivitäten, die durchgeführt werden müssen, definiert. Diese Aktivitäten haben eine bestimmte Ausführungslogik, eine bestimmte Funktion sowie einen Akteur.

Ein Projekt wiederum ist definiert als eine graduell erarbeitete Serie von Aktivitäten (Arbeitsschritten) und Aktionen, die zu einem einzigartigen Resultat führen. Im Projekt werden die Fragen des Umfangs, der Zeit und Kosten gestellt und beantwortet, um dieses Ergebnis zu erreichen.

<sup>96</sup><http://www.aras.com/university/SelfHelpGuides/Print%20of%20Online%20Help%20-%20Aras%20Innovator%20Application%20Framework.pdf>



Wie all diese Prozesse sich im Aras Innovator darstellen, kann der folgenden Tabelle entnommen werden.

	LifeCycle	Workflow	Project
Zweck	Definieren von sich ausschließenden Zuständen	Abläufe standardisieren und organisieren, wer diese durchführt	Management der Abhängigkeiten von Umfang, Kosten und Zeit
Knoten heißen	State	Activity	Activity2
Links heißen	Transition	Path	Predecessor
Instanziierung	Eine Instanz wird von vielen Items geteilt	Individuelle Workflow Prozess Instanzen	Verschieden in verschiedenen Instanzen
Parallele Pfade	Nein, nur ein "State"	Ja	Ja
Mehrdimensional	Nein	Ja	Nein
Knoten Methoden	Nein	Ja	Nein
Link Methoden	Ja, Pre und Post	Ja, Pre und Post	Nein

97

Auf die Eigenschaften von LifeCycle Maps wird an dieser Stelle nicht näher eingegangen. Eine vollständige Liste ist in Anhang 3 zu finden.

### Zusammenfassung Architektur

Der Aras Innovator überzeugt vor allem durch seinen hohen mitgelieferten Funktionsumfang, der einfachen Installation/ Einrichtung und der vergleichsweise einfachen Integration mit vorhandenen Microsoft-Diensten in einem Unternehmen (z.B. Sharepoint, Reporting Services).

### Dienstleistungserweiterung im Aras Innovator

Wie in Kapitel 2 dargelegt wurde, können PLM Systeme um Dienstleistungen und Dienstleistungsbausteine erweitert werden. Im PDLM Projekt wurden die Dienstleistungen in den Aras Innovator auf verschiedene Art und Weisen integriert. Diese Integrationen beziehen sich direkt auf die verschiedenen Ebenen von Dienstleistungen, welche im Projekt PDLM abgedeckt wurden. Grundlage ist hierfür die Modularisierung von Dienstleistungen. Diese Modularisierung wird im vorliegenden Fall durch den Service Modeller unterstützt. Diese Dienstleistungskomponenten dienen als Grundlage für

97

<http://www.aras.com/university/SelfHelpGuides/Print%20of%20Online%20Help%20-%20Aras%20Program%20Management.pdf> (Stand: 11.12:2012)

die Hinterlegung von Prozessen und Ressourcen für die zu erbringenden Dienstleistungen. So müssen für Dienstleistungen Prozessmodelle im Aras Innovator hinterlegt werden, welche die Ausführung der entsprechenden Dienstleistung steuern. Im Aras Innovator bilden die „Workflow Maps“ diese Möglichkeit. Nach dieser Definition steht eine „Workflow Map“ in einer direkten Relation zu einem „ItemType“.

Die Erstellung eines „ItemTypes“ kann noch erweitert werden, zum Beispiel um die Zuordnung zu dem „Poly Item“ als Dienstleistungsausführungen, oder als extra Listenelement zum Menü<sup>98</sup>. Dies ist eine Frage der Benutzeroberfläche, welche an dieser Stelle nicht weiter erörtert werden soll.

Nachdem die Komponenten durch ein Prozessmodell ergänzt werden können, ist es auch möglich entsprechende Ressourcen für die Dienstleistung zuzuordnen. Diese Ressourcen müssen noch näher bestimmt werden.

#### *Implementierung von Prozessmodellen/ Workflows für Dienstleistungen*

Eine Dienstleistung verfügt über einen Prozess mit verschiedenen Arbeitsschritten, welcher ausgeführt werden muss. Dieser Prozess kann auf verschiedene Art und Weise modelliert werden. Den vorliegenden Dienstleistungen wird in unserem Beispiel ein Workflow zugeordnet. Dies wurde bereits beschrieben: jedem „ItemType“ kann ein Workflow zugeordnet werden. Diese „Workflow Map“ kann formal als Prozessmodell bezeichnet werden. Diesem sind Aktivitäten zugeordnet, welche wiederum eine Funktion erfüllen. Zur Umsetzung dieser Funktionen werden Ressourcen auf verschiedene Weisen betroffen oder benutzt. Eine dieser Ressourcen sind bestimmte Parameter, aber auch Dokumente oder ähnliches können dies sein. Diese werden als Input und Output einer Funktion bezeichnet. Diese Parameter finden sich im Aras Innovator als Variables wieder. Sie können dem Prozess oder einzelnen Aktivitäten zugewiesen werden.

---

<sup>98</sup> Siehe Abschnitt: Dienstleistungen im Aras Innovator abbilden

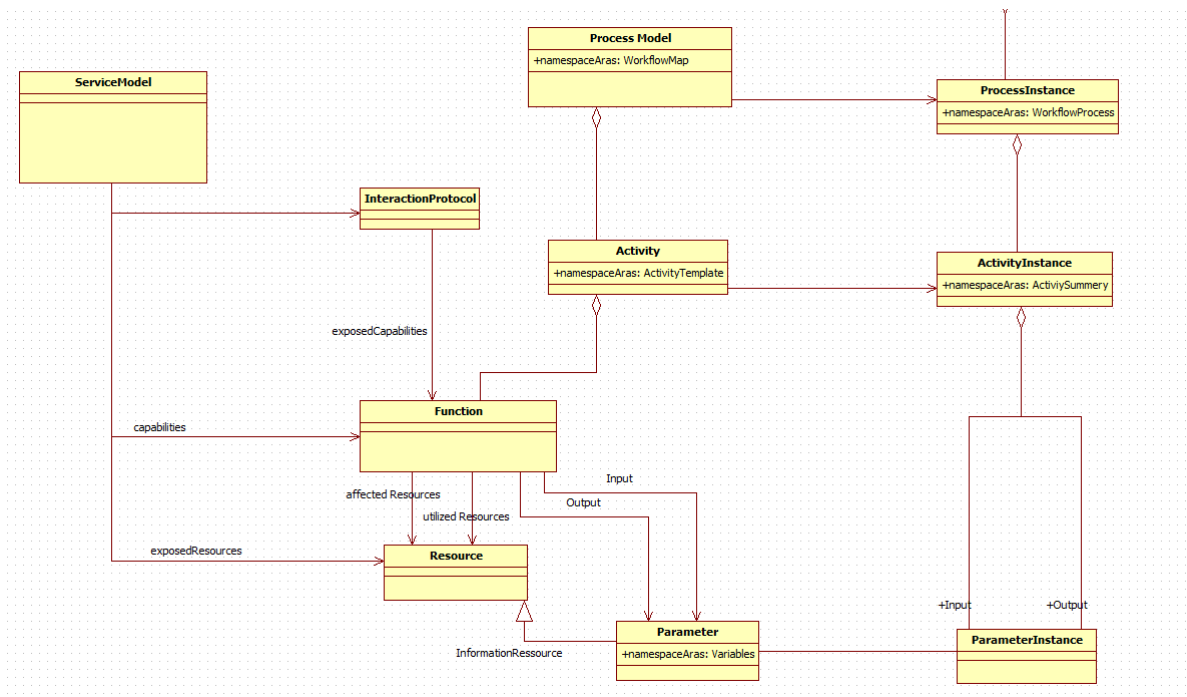


Abbildung 26 Dienstleistungsmodell und Prozessmodell

Auf diese Art und Weise können komplexe Prozesse mit entsprechenden Parametern im Aras Innovator abgebildet und den entsprechenden Dienstleistungen zugeordnet werden.

Verschiedene Ressourcen und deren Verknüpfung mit Zulieferern zu einer Dienstleistung

Ressourcen werden als Potenzialfaktoren definiert, welche für die Erstellung von Sachgütern bzw. Leistungen notwendig sind. Diese Definition findet auch im Bereich der Dienstleistungsmodellierung Anwendung (Böttcher 2009). Potenzialfaktoren können auch als Input bezeichnet werden, welche wiederum einen Output – ein Ergebnis generieren. Diese Begriffe finden ihre Anwendung in der Produktion (Produktionsmanagement) und in technischen Dienstleistungen (vgl. Böttcher 2009).“Produktion umfasst die Elemente (1) Potentiale: Bereitstellung von Ressourcen (Input), (2) Prozesse: Kombination von Ressourcen (Throughput) und (3) Produkte: Ergebnis (Output)“ (Palupski 2002). Eine Ressourcenplanung und -management ist daher unumgänglich für ein integriertes PDL-System.

Wie jedoch Ressourcen spezifiziert werden, hängt von den jeweiligen verfolgten Zielen ab. So unterscheidet man beispielsweise im Produktionsmanagement zwischen Personal, Betriebsmittel, Material, Zeit und Informationen, wobei die Ressourcen Zeit und Informationen eine Sonderstellung einnehmen, da sie in allen Bereichen geplant werden müssen (Gienke und Kämpf 2007).

Ein Ressourcenmanagement ist dafür zuständig, die „richtige Menge einer Ressource zum richtigen Zeitpunkt am vorgegebenen Ort bereitzustellen.“ (Gienke und Kämpf 2007). Eine Verwaltung muss

weitergehend Auskunft über Menge und Zustand (inkl. Verfügbarkeit) geben können. Im Folgenden wird sich mit der Verwaltung der Ressourcen und deren Verknüpfungspunkte zu den schon bestehenden Dienstleistungs- und Sachleistungsmodellen im Aras Innovator beschäftigt.

Während der Aras Innovator für Material und zum Teil für Betriebsmittel eine explizite Verwaltungsmöglichkeit in Form des Zulieferer- und Ressourcenmodells bietet, ist eine Personalplanung indirekt durch das beschriebene Projektmanagement und die eingesetzte Workflowsteuerung möglich. Diese Planungen betreffen jedoch primär die Produktion von Sachgütern.

Auch wenn Teile der Dienstleistungsressourcenplanung über das Projektmanagement gesteuert werden könnten, muss danach gefragt werden, wie es um die Verwaltungsmöglichkeiten von Dienstleistungsressourcen im Aras Innovator bestellt ist. So wird das Beispiel der Dokumenteneinbindung (welches als nächstes folgt) in den Aras Innovator zeigen, dass die Planung von Ressourcen und insbesondere der Ressourcenzugang überdacht werden müssen.

Unter Dienstleistungsressourcen zählen sicher auch Ressourcen, welche für Sachgüter eingesetzt werden. Darüber hinaus gibt es weitere Ressourcen, die es abzudecken gilt, wie Informationen, welche ad-hoc vom Kunden kommen oder systematisch durch vorgegebene Dokumente oder Formulare erhoben werden, aber auch die Dokumententemplates, die es zu verwalten gilt.

Im Prinzip sind Ressourcen Objekte, welche im Dienstleistungslebenszyklus zur Verfügung stehen müssen. Diese prinzipielle Verfügbarkeit dieser Ressourcen muss durch ein mit der entsprechenden Dienstleistung verknüpftes Objekt ermöglicht werden. Diese Objekte gilt es für jede einzelne Dienstleistungen zu bestimmen und zu implementieren. An dieser Stelle kann aber keine generelle Aussage darüber getroffen werden, welche konkreten Objekte das sind und wie diese implementiert werden können.

In der Dienstleistungsplanung müssen solche Anforderungen an Ressourcen mit aufgenommen werden – zum Beispiel über das als nächstes vorgestellte Requirement-Management-System. So kann eine Prozessanalyse, wie diesem Projekt zugrunde liegend, die Ressourcen, die innerhalb der Prozesse benötigt werden, aufgezeigen und später implementiert werden. In unserem Fall gilt es die Anforderungen, Standarddokumente und externe ERP Daten im Dienstleistungsprozess zur Verfügung zu stellen. Diese Anforderungen wurden exemplarisch und prototypisch umgesetzt.

Diese Arbeit leistet kein voll umfassendes Ressourcenmanagement, es werden jedoch Potentiale aufgezeigt, wie dies zu leisten ist. Es werden Ressourcen betrachtet, welche für die Referenzprozesse am entscheidendsten sind. Über die Einbindung von Daten hinaus ist dies der Einsatz von Personal, bzw. Abteilungen für Prozesse, die über ein entsprechendes Monitoring innerhalb des Aras Innovators überwacht werden können. Dies wird unter anderem durch den vorgestellten Eskalationsmodus

unterstützt. So ist die Unterstützung des Managements von Zeit, Personal und Prozessdaten innerhalb des Projektes prototypisch umgesetzt, aber gewiss noch erweiterbar. Darüber hinaus wird durch das Workflowsystem die Dienstleistung und deren Ressourcen an einen Ablauf gebunden, der gewährleisten soll, dass die „richtige Menge einer Ressource zum richtigen Zeitpunkt am vorgegebenen Ort [bereitzustellt]“ (Gienke und Kämpf 2007) wird.

### *Implementierung von Dokumenten als Ressourcen in den Workflow*

Eine besondere Anforderung, welche sich im Laufe des PDLM Projektes herausstellte, ist die Anbindung von Dokumenten (als Ressourcen) für Workflows und Dienstleistungen. Komplexe Informationen und Dokumentationen der Partner liegen in Dokumentenform vor und werden während der Dienstleistung immer wieder verwendet und verändert. Damit wurden Dokumente und Dokumentenvorlagen für dieses Projekt eine wichtige Ressource, die es in die Dienstleistung und die Workflows zu integrieren gilt. Zunächst müssen aber die Begriffe, die hier verwendet werden, geschärft werden. Wenn hier von Dokument die Rede ist, so meint dies, die reine Datei (Aras Namespace: File) und nicht ein Dokument, welches schon einer Dokumentverwaltung, wie sie Aras zur Verfügung stellt (Aras Namespace: Document), zugeordnet ist. Kurz: Aras-„Files“ werden hier als Dokumente bezeichnet und Aras-„Documents“ werden als Dokumentvorlagen bezeichnet.

Wie Dokumente mit Dienstleistungen und ihren entsprechenden Workflows verbunden werden können, hängt von unterschiedlichen Nutzungsarten ab. Zwei Formen sollen hier vorgestellt werden. Es ist zum einen die Anforderung, dass Dokumente ad-hoc während der Ausführung der Dienstleistung hinzugefügt werden können, und zum anderen, dass Dokumente als Vorlage für eine bestimmte Dienstleistung angegliedert werden können.

### Dokumente an einen Dienstleistungsworkflow anbinden

---

Für die Angliederung von Dokumenten an einen Dienstleistungsworkflow muss zunächst beschrieben werden, wie dies modelliert werden kann. Zu diesem Zweck wird das Workflowmodell entsprechend erweitert. Dies bedeutet, dass ein neuer RelationshipType erstellt werden muss, welcher die ItemTypes „Activity“ und „File“ miteinander verknüpft. In unserem Fall heißt diese Relationship „Activity File“<sup>99</sup>.

---

<sup>99</sup> Eigenschaften: Source ItemType: Activity; Related ItemType: File; Create Related: true; Behavior: Float; Grind View: Intermix; Pick&Create: true; Requires Related: true;

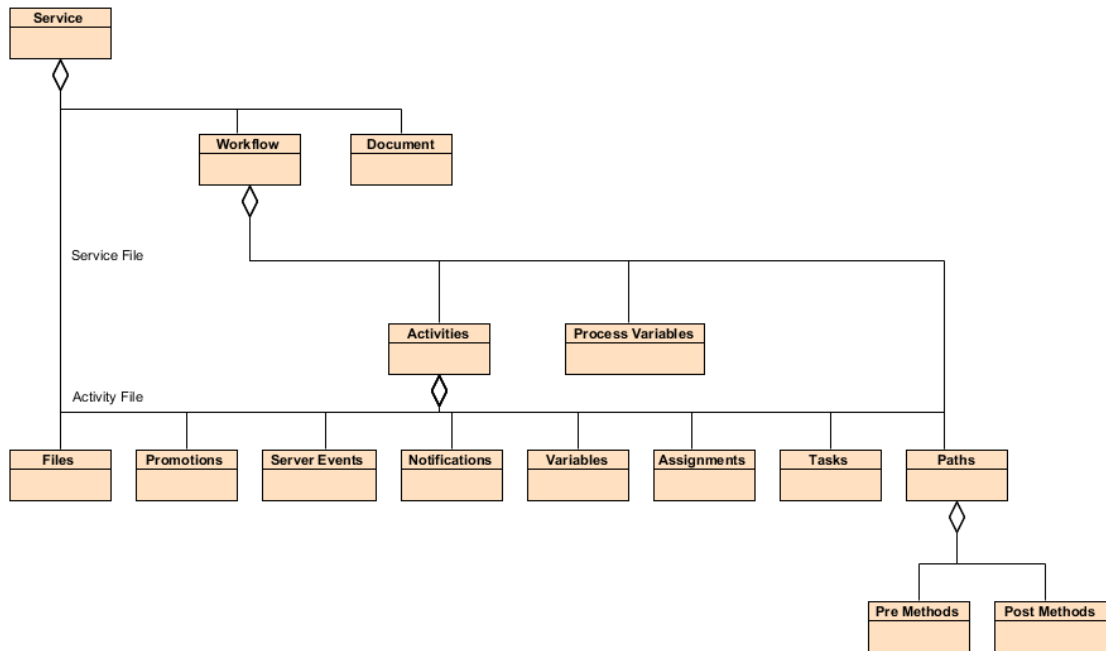


Abbildung 27 Verknüpfung von Dokumenten und Dienstleistungsworkflows

Nach der Modellierung folgt die Implementierung. Für die Implementierung ist es notwendig das Formular für die Erbringung einer Aktivität zu manipulieren. Dieses liegt in Form einer ASP.NET Server Page (*InBasket-VoteDialog.aspx*) im Installationspfad: *C:\Program Files (x86)\Aras\Innovator\Innovator\Client\scripts\InBasket* vor. Für den Upload von Dateien wird dabei ein neues Input-Feld gesetzt, auf welches dann bestimmte Methoden angewendet werden können. Dieses könnte zum Beispiel am Ende des Formulars erfolgen. Der Quellcode lautet wie folgt:

```

<tr id="FileSection">
<td align="center" class="logicalSection">
    <fieldset>
        <p></p>
        <input type="file" id="file-selected"
            name="file-selected" onchange="doAttachFile()" />
    </fieldset>
</td>
</tr>

```

Wenn eine Datei ausgewählt wurde, wird die Methode „doAttachFile()“ aufgerufen, welche ein Dokument in den Vault lädt und die entsprechenden Verknüpfungen setzt.

```

function doAttachFile() {
    var filePath = document.getElementById("file-selected").value;
}

```

```

// Activity
var actId = aras.getItemProperty(MyActivity, "id");
var actName = aras.getItemProperty(MyActivity, "name");
var ActItem = myInnovator.newItem('Activity', 'update')
ActItem.setId(actId);
ActItem.setProperty('name', actName);
ActItem.lockItem();

// Create File
var fileItem = myInnovator.newItem("File", "add");
fileItem.setFileName(filePath);

// Create the relationship between the Document and File
var relItem = myInnovator.newItem("Activity File", "add");
ActItem.addRelationship(relItem);
relItem.setRelatedItem(fileItem);

var results = ActItem.apply();
if (results.isError()) {
    top.aras.AlertError(results.getErrorDetail());
} else {
    // Show the new Document
    top.aras.uiShowItemEx(results.getItemByIndex(0).node, 'tab
        view', true);
}
}

```

Um diese Funktion ausführen zu können, sind noch folgende Ergänzungen nötig. Zum einen muss eine weitere JS-Bibliothek geladen werden:

```

<script type="text/javascript"
src="../../javascript/iom.js"></script>

```

Und zum anderen muss die Klasse „myInnovator“ initiiert werden, bevor die Funktion ausgeführt wird:

```

var myInnovator = new Innovator();

```

So können ad-hoc Dateien während der Ausführung einer Dienstleistung hinzugefügt werden. Nun müssen diese Dokumente noch entsprechend in der Übersicht der jeweiligen Dienstleistung sichtbar sein. Dafür ist es notwendig, eine entsprechende Verknüpfung zwischen der jeweiligen Dienstleistung und dem Dokument zu erstellen. Um von der Aktivität eines Workflows auf die zugeordnete Dienstleistung zu gelangen, bedarf es einiger Schritte.

1. Es muss der Workflow anhand des Parameters abgerufen werden.
2. Dieser Workflow besitzt die Parameter "source\_id, source\_type".
3. Der Parameter "source\_type" ist eine Id, welche auf den ItemType verweist. Dementsprechend muss dieser wiederum abgerufen werden, um den Namen des ItemTypes (der Dienstleistung) zu erfassen.
4. Nun kann die zugeordnete Dienstleistung aufgerufen werden und mit Parametern gefüllt werden.

```
/**
 * Query Workflow to get Source_id and Source_type
 */
var qryWorkflow = myInnovator.newItem("Workflow", "get");
qryWorkflow.setAttribute("select", "source_id,
    source_type");
var workProcess = new Item("Workflow Process", "get");
workProcess.setAttribute("select", "id");
workProcess.setProperty("id", workflowId);
qryWorkflow.setRelatedItem(workProcess);

var results = qryWorkflow.apply();
if (results.isError()) {
    top.aras.AlertError(results.getErrorDetail());
}
if (results.isCollection()) {
    top.aras.AlertError('workflow ist nicht zuzuordnen');
}
var sourceId = results.getProperty('source_id');
// get sourceType as String
var sourceTypeId = results.getProperty('source_type');
var sourceTypeObject = top.aras.getItemById("ItemType",
    sourceTypeId, 1);
var sourceType = aras.getItemProperty(sourceTypeObject,
    "name");
// get sourceObj and prepar it (lock)
var sourceObj = top.aras.getItemById(sourceType, sourceId, 1);
var source = myInnovator.newItem(sourceType, 'update');
source.setId(sourceId);
source.setProperty('owned_by_id',
    aras.getItemProperty(sourceObj, "owned_by_id"));
source.lockItem();
```



Package: aras.pdlm.workflowFiles

### Dokumentvorlagen für eine Dienstleistung definieren

Um eine Dokumentenvorlage für eine Dienstleistung bzw. für den Workflow anzubinden, ist es darüber hinaus notwendig, den Workflow zu erweitern. Für Dokumentenvorlagen wird das Dokumentenverwaltungssystem in den Workflow eingebunden. Zu diesem Zweck wird das vorhergehende Modell entsprechend erweitert.

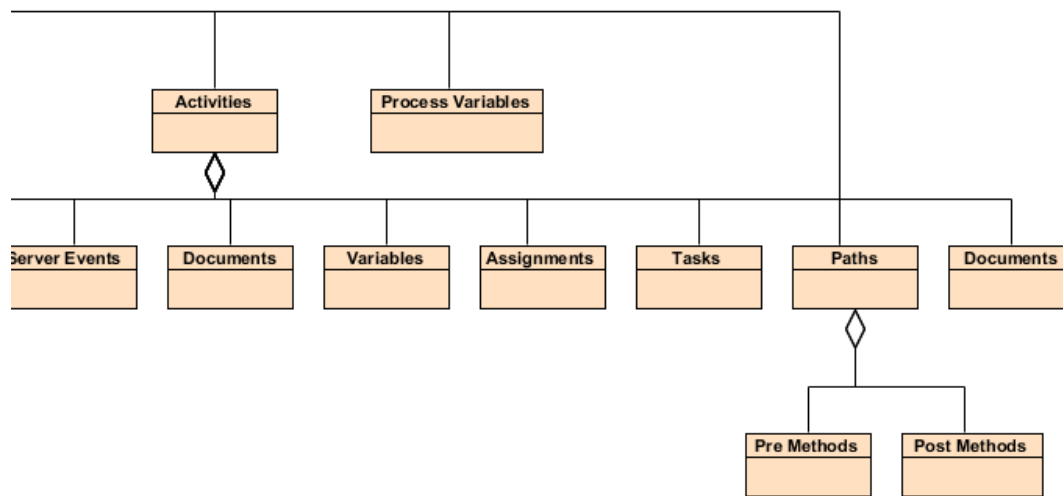


Abbildung 28 UML Modell Workflow Erweiterung Aras

Durch eine entsprechende Relationship wird dies in Aras umgesetzt. Im Folgenden können nun Dokumente für einzelne Aktivitäten des Dienstleistungsworkflows zur Verfügung gestellt werden.

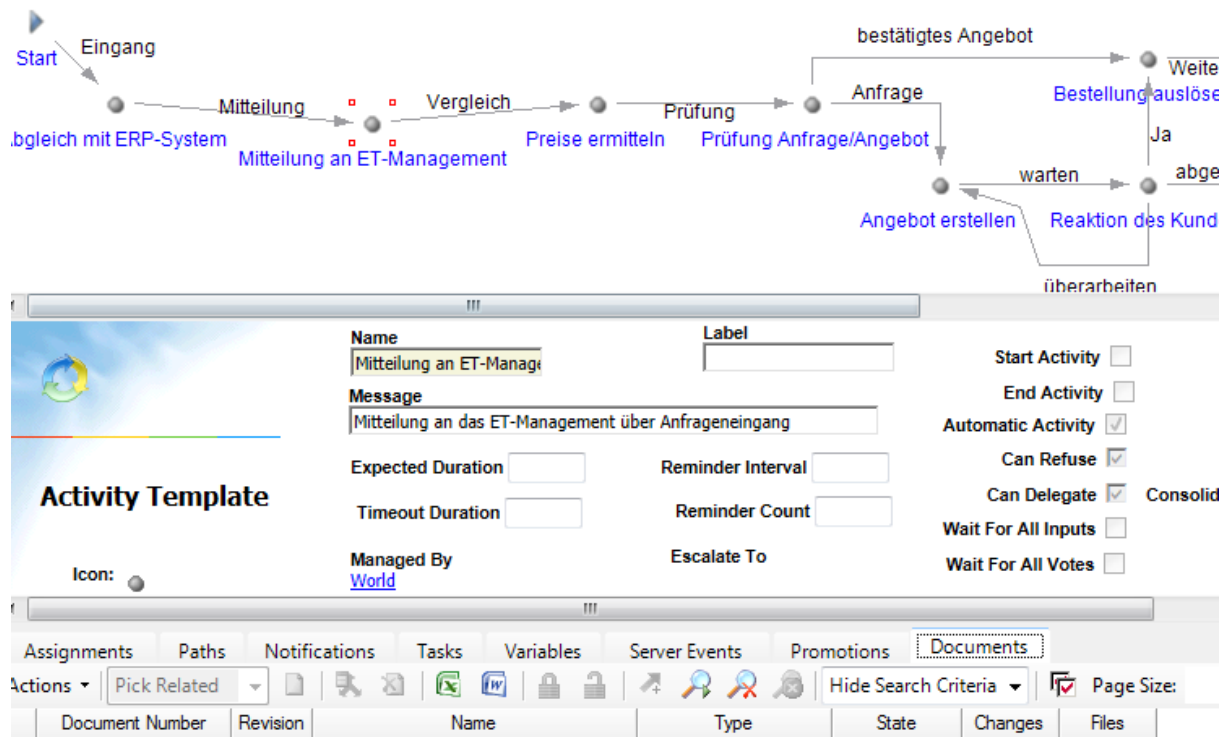


Abbildung 29 Aras Umsetzung Dokumentvorlage

Auf diese Dokumentenvorlagen muss nun in der *InBasket-VoteDialog.aspx* zugegriffen werden. Der entsprechend implementierte Code kann im Folgenden nachvollzogen werden:

```
function getDocumentTemplates() {

var actTem = aras.getItemByKeyedName('Activity Template',
top.aras.getItemProperty(MyActivity, "keyed_name"));
var actTem2 = aras.getItemById('Activity Template',
top.aras.getItemProperty(actTem, "id"));

varList = ActTem2.selectNodes('Relationships/Item[@type="Activity
Template Document"]');

var varId = false;
var arr = getSequenceOrderArray(varList);
for (var i = 0; i < varList.length; i++)
{
actFile = varList.item(arr[i].ind);
varID = aras.getItemProperty(actFile, "related_id");
temp = top.aras.getItemById("Document", varID, 1);
var varTempId = aras.getItemProperty(temp, "id");
this.vars.document.write('<p onclick="showSomeDocuments(\'
+ varID + \'')">');
}
```

```
        this.vars.document.write(aras.getItemProperty(temp,
            "name"));
        this.vars.document.write('</p>');
    }
}
```

Die Implementierung zusätzlicher Ressourcen in den Dienstleistungsprozess muss den Prozessen der Unternehmen angepasst werden und ist für die Dienstleistungsplanung wichtig. Welche Ressourcen wann zur Verfügung stehen, muss vorher definiert und durch IT unterstützt werden können.

Damit wurden an dieser Stelle zwei Dinge demonstriert. Zum einen wurde gezeigt, dass in der Definitionsphase der Dienstleistung die Ressourcen, auf die während der Erbringung zugegriffen werden müssen, definiert werden und innerhalb der IT Unterstützung implementiert werden müssen. Zum anderen wird hier exemplarisch aufgezeigt, wie sich Ressourcen integrieren lassen, die für die Erbringung von Dienstleistungen entscheidend sind, und dass der Dienstleistungsprozess und die Informationsflüsse transparent gehalten werden und gut dokumentiert abgelegt werden können.

*Implementierung eines einfachen Requirement-Management-Unterstützungswerkzeuges und dessen Verknüpfung mit Dienstleistungen*

„Eine Anforderung ist eine Aussage über eine Eigenschaft oder Leistung eines Produkts, eines Prozesses oder der am Prozess beteiligten Ressourcen“ (van Husen 2007)<sup>100</sup>.

Die Aufgabe eines Anforderungs- oder Requirement-Managements ist das Managen von Anforderungsänderungen sowie der Verknüpfungen einzelner Anforderungen, Anforderungsdokumenten und anderen Dokumenten, welche zum Beispiel im Entwicklungsprozess erstellt wurden. (vgl. van Husen 2007; Kotonya und Sommerville 1998).

Um diese Aufgabe zu übernehmen wurde ein Requirement-Management-Tool als ein Community Projekt von Rolf Laudenbach entwickelt. Dieses Werkzeug befähigt den Aras Benutzer Produkthanforderungen aller Art – also Kundenanforderungen, technische oder gesetzliche Anforderungen – in einer dynamischen Datenbank zu definieren, zu entwickeln und systematisch zu erfassen.

---

<sup>100</sup> “requirement. (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation A documented representation of a condition or capability as in (1) or (2)” (IEEE standard glossary of software engineering terminology 1990 S.62)

Es beinhaltet die Analyse der Auswirkungen und Rückverfolgbarkeit durch das Änderungsmanagement und die Versionsgeschichte, welche der Aras Innovator zur Verfügung stellt. Durch einfache Installation des Pakets stehen die entsprechenden ItemTypes zur Verfügung. Die Aras Struktur ermöglicht dann das Requirement-Management-Tool mit entsprechenden Dienstleistungen und Sachleistungen zu verknüpfen<sup>101</sup>.

Das Management-Tool erfasst dabei, wo diese Anforderungen genutzt worden sind, was eine Nachvollziehbarkeit sicherstellt. In der gelieferten Version können zusätzlich zum „Where used“ auch die „Related Parts“ hinzugefügt werden. Entsprechend den PDLM Vorgaben ist hier eine Erweiterung eines RelationshipTypes, des ItemType „Requirement“ um die Dienstleistungskomponenten erforderlich.

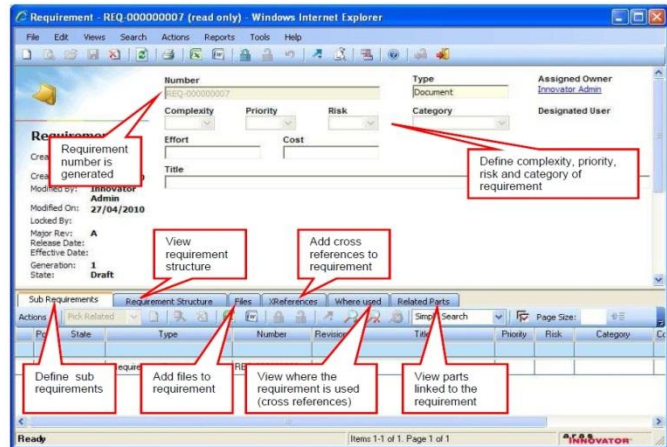


Abbildung 30 Requirement-Management-Tool

Ob und wie dieses Requirement Tool im Speziellen auf Dienstleistungen zugeschnitten werden muss, kann an dieser Stelle nicht erörtert werden. Dazu wären detailliertere Evaluationen notwendig, welche im Rahmen dieses Projektes nicht abgedeckt werden konnten.

Die Integration eines Requirement-Management-Tools und deren einfache Anpassung in Hinblick auf die Nutzung für Dienstleistungen ist jedoch exemplarisch gezeigt worden.

### Zusammenfassung und Ausblick

In diesem Kapitel wurde beschrieben, wie Dienstleistungen im Aras Innovator abgebildet werden können, bzw. wie der Aras Innovator modifiziert und angepasst werden kann, um Dienstleistungen abbilden zu können. Zu diesem Zweck wurde kurz auf die interne Logik des Aras Innovators eingegangen, um verstehen zu können, wie eine Dienstleistungsunterstützung auf verschiedenen Ebenen innerhalb der Aras Innovator Logik aussehen kann. Im Speziellen wurde hierbei auf Dienstleistungsprozesse, Dienstleistungsressourcen und ein einfaches Requirement-Management für Dienstleistungen eingegangen. Das vorliegende Kapitel fokussierte sich auf diese Ebenen von Dienstleistungen, weil sie direkt mit Hilfe des angepassten Aras Innovators unterstützt werden konnten. Damit wurde gezeigt, dass PLM/PDM Systeme prinzipiell in der Lage sind, Dienstleistungen auf bestimmten Ebenen (Prozessebene, Ressourcenebene) zu unterstützen. Dieses Potential ist ein

<sup>101</sup>

<http://myinnovator.com/vault/vaultserver.aspx?dbname=MyInnovator&fileid=1864C45824594539B17319FED11E701E&filename=RequMgmt%20screen%20shot1.jpg> (Stand: 04.01.2013)

Transformationspotential in Richtung der informationstechnischen Unterstützung von Servitizationprozessen (Baines) und der systematischen Unterstützung von Produkt-Service Systemen (Mont). Diese Unterstützung kann trotz der vielen Möglichkeiten noch ausgebaut und stärker auf Dienstleistungen fokussiert werden (z.B. integriertes Ressourcenmanagement innerhalb der Prozesse, Standardisierung der Prozessmodelle (z.B. durch Integration von BPMN oder EPK)).

Bisher wurde noch nicht auf die Produktebene von Dienstleistungen und die Kundenintegration eingegangen. Die nachfolgenden Kapitel beschäftigen sich mit der Produktsicht auf Dienstleistung und der systematischen und digitalen Kundenintegration in den Dienstleistungsprozess mit Hilfe des Aras Innovators.

## Quellenverzeichnis

---

IEEE standard glossary of software engineering terminology (1990). New York, N.Y., USA: Institute of Electrical and Electronics Engineers.

Aras (2013): Aras Innovator Programmers Guide v.9.3.0. Hg. v. Aras. Online verfügbar unter <http://www.aras.com/support/documentation/9.3.0/Other%20Documents/Aras%20Innovator%209.3%20-%20Programmers%20Guide.pdf>, zuletzt geprüft am 01.10.2013.

erdomke (2013): (Anwendung) AML Studio, AUTOR. Online verfügbar unter <http://amlstudio.codeplex.com/>, zuletzt aktualisiert am 01.10.2013, zuletzt geprüft am 01.10.2013.

Gienke, Helmuth; Kämpf, Rainer (2007): Handbuch Produktion. Innovatives Produktionsmanagement: Organisation, Konzepte, Controlling. München: Hanser.

Golemo, Florian (2013): Integration eines Service-Portfolio-Managers (SM) in ein Product-Lifecycle-Management-System (PLM). Bachelorarbeit.

Kotonya, Gerald; Sommerville, Ian (1998): Requirements engineering. Processes and techniques. Chichester,, New York: J. Wiley.

Löper, Christian; Schäfer, Wilhelm: Anforderungsanalyse und Anforderungsdefinition für sicherheitskritische Systeme (inkl. formale Methoden).

Palupski, Rainer (2002): Management von Beschaffung, Produktion und Absatz. Leitfaden mit Praxisbeispielen. 2. Aufl. Wiesbaden: Gabler.

Project Management Institute (2008): A guide to the project management body of knowledge (PMBOK® Guide). 4. Aufl. Newtown Square, Pa: Project Management Institute.

Schlick, Gerhard H. (1999): Projektmanagement, Gruppenprozesse, Teamarbeit. Wege, Hilfen und Mittel zu schnittstellen-minimierter Problemlösungskompetenz. 3. Aufl. Renningen-Malmsheim: Expert-Verl.

van Husen, Christian (2007): Anforderungsanalyse für produktbegleitende Dienstleistungen. Heimsheim: Jost-Jetter.

## Kapitel 5 Unterstützung der produktbezogenen Dienstleistung durch die Integration des Service Modeller (*Florian Golemo*)

### Einleitung<sup>102</sup>

---

In vielen Branchen ist es üblich, dass Produkte oder Dienstleistungen nicht voneinander losgelöst, sondern gemeinsam in einem Produkt-Dienstleistungs-Bündel angeboten werden. Der Kunde hat möglicherweise beim Kauf die Wahl zwischen verschiedenen Produktvariationen, die dem Verkäufer wiederum verschiedene Dienstleistungen ermöglichen. Die Daten aus beiden Bereichen werden in Service-Portfolio-Management-Systemen (SM) verwaltet. Diese Systeme werden in aller Regel durch Software unterstützt. Dass es erhebliche ökonomische Vorteile für ein Unternehmen bietet, die gesamten Lebenszyklen seiner Produkte in einem System zu verwalten, kann als Common Sense bezeichnet werden. Gleiches gilt für die elektronische Dienstleistungsverwaltung, die für zahlreiche Firmen bereits zur Notwendigkeit geworden ist. Ein zentrales Paradigma des Produkt-Lebenszyklus-Management (PLM) ist es, alle Unternehmensdaten in einem System zu integrieren. Bisher geschieht dies überwiegend nur für Produkte, während die Servicedaten-Integration weitgehend übersehen wird. Ziel dieses Kapitels ist es, die Machbarkeit durch eine exemplarische Integration aufzuzeigen. Dazu wurden Daten aus dem SM-System "Service-Modeller" in den Aras Innovator integriert. Zu diesem Zweck wurde eine Software entwickelt, welche die Daten aus dem SM extrahiert und im PLM/PDM verfügbar macht. Die Entwicklung fand in PHP statt und nutzt die SOAP-Schnittstellen beider Systeme. Das Ergebnis der Arbeit, das Konverter-Werkzeug, erfüllt das gesteckte Ziel und steht als Open-Source-Anwendung online zum Herunterladen bereit.

### Vorstellung des SM-Systems "Service Modeller"

---

Im Rahmen des KoProServ-Projektes ist eine Prototyp-Anwendung zur computergestützten Modellierung eines Dienstleistungsportfolios entstanden. Diese Software, der "Service Modeller" (SM), ist eine Webanwendung, die auf Silverlight basiert (Ergebnisse – KoProServ 2013). Die Verwendung des Service Modellers für diese Arbeit wurde vom betreuenden Lehrstuhl als Teil der Aufgabenstellung vorgegeben. Dieses Kapitel soll einen Überblick über die Funktionsweise des SM geben, seine grundlegende Architektur aufzeigen und in Vorarbeit zum nächsten Kapitel die Schnittstellen untersuchen.

---

<sup>102</sup> Dieses Kapitel ist Teil der Bachelor Arbeit von Florian Golemo 2013 (Golemo 2013), verfügbar unter <http://lips.informatik.uni-leipzig.de/files/thesis-final.pdf> (Stand: 19.06.2014)

Theoretischer Hintergrund

Klingner et al. stellen in (2011) fest, dass im Zuge des zunehmenden Individualisierungsbedarfs der Gesellschaft wachsende Anforderungen an die Quantität und Flexibilität des Serviceangebots von Unternehmen gestellt werden. Die Autoren entwickeln eine Möglichkeit zur standardisierten Darstellung des Service-Angebots und zur besseren Kostenschätzung. Ihr Ansatz umfasst zwei Phasen: (a) die Service-Modellierung und (b) die Service-Konfiguration. Die Modellierung (a) zielt darauf ab, das gesamte Repertoire an möglichen Dienstleistungen zu erfassen und Relationen aufzuzeigen. Die möglichen Dienstleistungen werden "Servicekomponenten" (engl. "service components") genannt. Mit Relationen ist gemeint, dass Dienstleistungen gebündelt angeboten werden können oder abhängig von anderen. Wenn der Kunde, nach Buchung einer Komponente, aus mehreren Servicekomponenten eine oder mehrere auswählen kann oder muss, so können diese Komponenten in einer Relation zusammengefasst werden. So entstehen baumartige Netzwerke, wie in Abbildung 31 dargestellt.

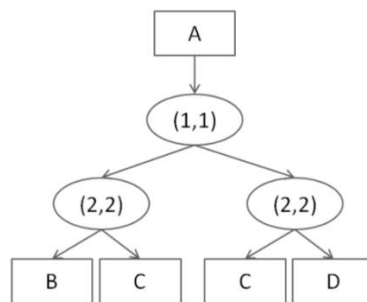


Figure : Defining logical, hierarchical dependencies using cardinalities

**Abbildung 31** Diese Abbildung stammt aus Klingner et al. (2011) und stellt ein Beispiel-Service-Portfolio nach dem Modell von Klingner et al. vor: Wenn man Service-Komponente A kauft, muss man eine Wahl treffen. Die Wahl beläuft sich entweder auf die linke oder rechte Seite (B,C) oder (C,D) des nachfolgenden Baumes. Man muss mindestens eines von beiden wählen, aber darf nur maximal eines wählen, daher die Notation "(1,1)". Je nach Wahl muss man jeweils genau zwei aus den Folgeelementen im Baum wählen, daher die Kardinalität "(2,2)". Je nachdem, wie man sich also entscheidet, erhält man die Dienstleistungskomponenten A,B,C oder A,C,D.

In Phase zwei (b) wird je nach Kunde eine Servicekonfiguration (engl. "service configuration") gefunden, indem bestimmte Services aus dem Modell aus- oder abgewählt werden. Wenn man die Komponenten in der ersten Phase mit Kenngrößen, wie Zeitaufwand oder Kosten versieht, so kann man nach der Servicekonfiguration präzise abschätzen, wie der Gesamtaufwand für die gewählten Optionen ausfällt. Dieser Modellierungsansatz wurde im Service Modeller implementiert.

Überblick

Der Service Modeller funktioniert als Silverlight-Applikation in beliebigen Webbrowsern unter Windows, für die das Silverlight-Plugin Version 5 oder neuer zur Verfügung steht. Der Prototyp unter (ServiceModeller 2013) bedarf keiner Anmeldung und kann für beide Phasen des theoretischen Modells genutzt werden. Abbildung 32 stellt die Benutzeroberfläche dar.



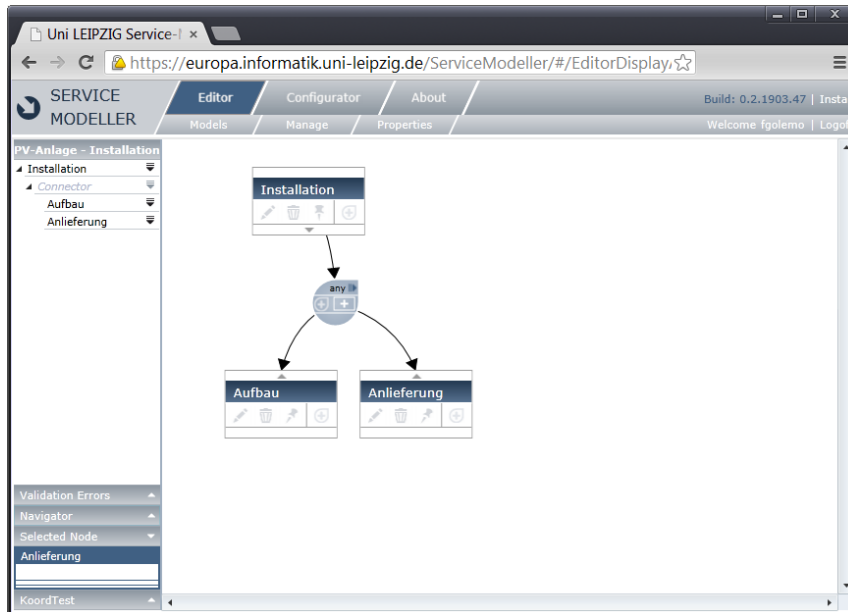


Abbildung 32 Das Frontend des Service Modellers ist hier abgebildet. Am oberen Rand befinden sich die Hauptnavigationselemente für die zwei Phasen. Die Modellierungsphase ist im “Editor” repräsentiert und die Konfigurationsphase im “Configurator”. In der horizontale Leiste darunter, über “Models”, kann man, nach Login, jeweils Modelle oder Konfigurationen speichern und später erneut laden und bearbeiten. Links bietet sich eine Übersicht über alle Komponenten und deren jeweilige Relation (“Connector”). Im Hauptfenster werden diese grafisch dargestellt und man kann mit Hilfe der Maus weitere Komponenten oder Relationen hinzufügen oder bestehende ändern bzw. löschen.

Der Service Modeller lässt sich für beide Phasen der Modellierung des Dienstleistungsportfolios nutzen. Nachdem die erste Phase abgeschlossen ist, kann man das Dienstleistungsmodell konfigurieren und einzelne Komponenten hinzu- oder abwählen. Eine solche Konfiguration kann im Anschluss gespeichert werden. Konfigurationen werden intern automatisch mit dem zugehörigen Modell assoziiert.

### Architektur

Auf der Server-Seite benötigt der Modeller eine Microsoft-Umgebung mit .NET-Framework Version 3 oder höher, IIS-Webserver und MSSQL-Datenbankserver (ServiceModeller 2013). Die Windows Communication Foundation (WCF, Teil des .NET-Framework seit Version 3 (Bustamante 2013)) stellt eine Schnittstelle zwischen Datenbank und Silverlight-Komponente zur Verfügung [WCF allgemein: (Microsoft Corporation.Windows Communication Foundation 2013)]. Das bedeutet, die Datenbank ist abstrahiert als WCF-Server-Komponente und die Silverlight-Anwendung fungiert als WCF-Client-Anwendung, welche die Server-Dienste konsumiert.

Clientseitig wird lediglich ein Browser benötigt, der das Microsoft-Silverlight-Plugin unterstützt. Dieses wird derzeit nur für das Windows-Betriebssystem XP SP2 oder neuer angeboten, unterstützt dafür

jedoch ein breites Spektrum an Browsern<sup>103</sup>. Das Linux-Äquivalent zu Silverlight, Novells “Moonlight” (Inc. Novell 2013), wurde in Version 3.99.0.3 getestet und funktioniert nicht: Die SM-Silverlight-Komponente wird vom Moonlight-Plugin heruntergeladen, aber nicht ausgeführt.

### Schnittstellen

Laut der technischen Dokumentation<sup>104</sup> stellt der SM-Server auch eine SOAP-Schnittstelle zur Verfügung. Diese Schnittstelle stellt 4 Funktionen bereit:

- “**GetAvailableModels**” ruft alle verfügbaren Modelle ab und gibt diese jeweils mit Name, ID und Beschreibung zurück. Die Funktion erwartet keine Parameter.
- “**GetModelByID**” liest genau ein Modell aus, anhand des übergebenen ID-Parameters. Die Syntax entspricht dem Beispiel in der technischen Dokumentation (wie auch bei “GetConfigurationByID”). Die Komponenten, aus denen das Modell zusammengesetzt ist, sind bereits enthalten.
- “**GetAvailableConfigurations**” hat eine ähnliche Funktionsweise wie “GetAvailableModels”, nur dass stattdessen Dienstleistungskonfigurationen ausgegeben werden.
- “**GetConfigurationByID**” ist das Äquivalent zu “GetModelByID” für Dienstleistungskonfigurationen. Weiterhin wird das Modell angegeben, von dem die Konfiguration stammt.

### Integration des SM “Service Modeller” in das PLM “Aras Innovator”

---

Die technische Natur des Service Modeller wurde bereits beleuchtet. Für die Lösung des Problems der Integration ist es notwendig, die konkreten Anforderungen an die Implementierung zu erfassen. Dies wird in Form eines Lastenhefts durchgeführt. Auf ein Pflichtenheft wird verzichtet, da das folgende Kapitel ebenso als Konzept der Umsetzung fungieren kann. Bevor dieses Konzept jedoch aufgebaut wird, muss untersucht werden, wie genau sich das gewählte PLM mit anderen Systemen verknüpfen lässt. Eine vorläufige Analyse ist bereits während der Vorauswahl passiert, aber nun müssen die technischen Gegebenheiten genauer betrachtet werden. Im Anschluss wird anhand von UML-Grafiken das Konzept der Umsetzung erläutert.

Der letzte Teil dieses Kapitels beschäftigt sich mit der tatsächlichen Umsetzung des Projektes in PHP. Dabei werden zuerst die beteiligten Anwendungen aufgelistet aus Gründen der Reproduzierbarkeit

---

<sup>103</sup> Microsoft Corporation. Get Silverlight | Microsoft Silverlight. URL: <http://www.microsoft.com/getsilverlight/Get-Started/Install/Default.aspx> (besucht am 20. 07. 2013).

<sup>104</sup> Universität Leipzig. Service Modeller - SOAP Info. URL: [http://europa.informatik.uni-leipzig.de:8095/soap\\_info.pdf](http://europa.informatik.uni-leipzig.de:8095/soap_info.pdf) (besucht am 20. 07. 2013).

sowie die Einrichtung der Entwicklungsumgebung. Am Ende wird auf Stolpersteine und Schwierigkeiten während des Entwicklungsprozesses eingegangen, damit diese in zukünftigen Entwicklungen in diesem Bereich umschifft werden können.

### Anforderungen

Die Erstellung des Lastenheftes ist ein Standardverfahren, um die Anforderungen eines Auftraggebers an ein Softwareprojekt aus dessen Sicht aufzunehmen (Balzert 2009). Für das Lastenheft wurde eine Dokumentenschablone gewählt, die aus dem Fach "Softwaretechnik" von Prof. Dr. Klaus-Peter Fähnrich<sup>105</sup> stammt und auf einem Standardwerk von Balzert (2009) aufbaut. Da es sich um ein eigenständiges Dokument handelt, wurde es in diese Arbeit nicht aufgenommen (siehe dazu Golemo 2013). Auf der Grundlage dieser Spezifikation wird im folgenden Kapitel der Aras Innovator näher untersucht und das Konzept für die Implementierung abgeleitet.

### Technisches Konzept

Wie bereits beschrieben, soll vor dem Hintergrund der Integration des Dienstleistungsportfolios der Aras Innovator erweitert werden. Hierfür gibt es mehrere Möglichkeiten: (a) Man könnte eine Funktion innerhalb des Aras Innovators schreiben, die regelmäßig Verbindung zum Service Modeller herstellt, die Modelle und Konfigurationen abrufen und in das eigene Innovator-System integriert. Oder (b) man könnte eine eigenständige Software schreiben, die auf beide Systeme von außen zugreift und aus dem SM die Daten extrahiert und in das PLM einspielt.

Variante (a) hätte den Nachteil, dass man an die Programmierumgebung des Innovators gebunden ist und im Internet Explorer entwickeln und debuggen muss. Dies ist ein Nachteil, da diese browsereingebettete Entwicklungsumgebung langsam ist und als unangenehm in der längeren Arbeit empfunden wurde. Außerdem hätte auf eine der Programmiersprachen C#, Visual Basic oder JavaScript zurückgegriffen werden müssen. Der Autor ist mit den ersten beiden nur eingeschränkt vertraut. JavaScript wäre eine Option gewesen, jedoch ist nicht gewährleistet, dass alle API-Funktionen des Innovators in ausreichendem Maße hätten genutzt werden können. Diese Möglichkeit wurde in Anbetracht der anderen Nachteile nicht tiefer untersucht und stattdessen Variante (b) gewählt.

Dazu muss untersucht werden, wie man außerhalb des Aras Innovators auf dessen Daten lesend und schreibend zugreifen kann. Davon hängt maßgeblich ab, wie die technische Konzeption ausfällt. Diese Untersuchung findet im nächsten Kapitel statt.

---

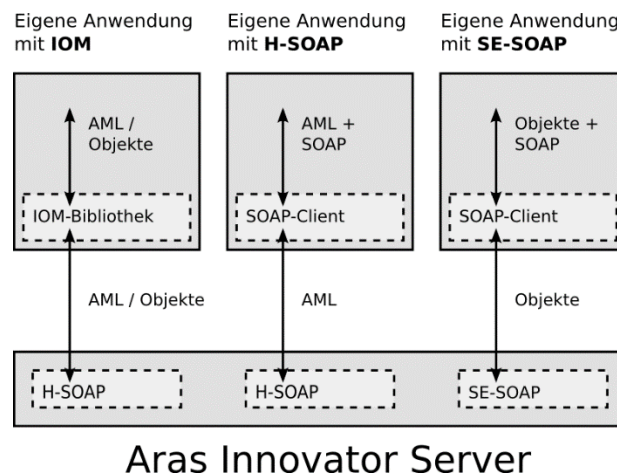
<sup>105</sup> Modul: Softwaretechnik (10-201-2321). URL: <http://bis.informatik.uni-leipzig.de/de/Lehre/BAMA/Softwaretechnik?v=y56> (besucht am 20. 07. 2013).

Schnittstellen des Aras Innovator aus technischer Sicht

Die Innovator-Serverkomponente bietet drei mögliche Schnittstellen für den Datenaustausch: die "IOM API" als .NET-Bibliothek, die Haupt-SOAP-Schnittstelle (nachfolgend "H-SOAP") und selbst-eingerichtete SOAP-Schnittstellen (nachfolgend "SE-SOAP"). Die IOM-API ist eine Softwarebibliothek, die von dem PLM mitgeliefert wird. H-SOAP und SE-SOAP sind Schnittstellen, die am Webserver, dem IIS, entweder bei der Installation (H-SOAP) oder nachträglich von Administratoren (SE-SOAP) eingerichtet werden.

Jegliche Kommunikation mit dem Aras-Server findet über AML, einem XML-Dialekt, statt. Alle Einstellungen, Objekte, Relationen, Befehle werden zwischen dem Aras-Server und zum Beispiel der mitgelieferten Clientsoftware (ASP.NET- Browserkomponente) in AML kommuniziert. Für die eigene Verwendung kapselt Aras diese AML-Sprache in eine API, die einfache Erstellung und Manipulation von Objekten (Daten) zulässt. Jedoch erlaubt sie auch weiterhin die Nutzung von AML. Diese API heißt "Innovator Object Model API" oder kurz "IOM API"<sup>106</sup>.

Wenn man eigene Software schreiben möchte, die mit dem Aras-Server kommuniziert, so kann man AML nutzen (mittels IOM oder H-SOAP) oder eine höhere Abstraktionsebene (IOM oder SE-SOAP). Die Methoden werden im nächsten Abschnitt erläutert. Abbildung 33 stellt die Schnittstellen grafisch gegenüber.



**Abbildung 33** Im Bild sind im oberen Teil die verschiedenen Möglichkeiten der Anbindung zu sehen. Jeweils an einem Pfeil steht, welche Daten übertragen werden. Graue Boxen stehen für Anwendungen. In gestrichelten Boxen sind die genutzten Schnittstellen bzw. -Konsumenten verzeichnet. "Objekte" bedeutet dabei XML-serialisierte Objekte, in Abgrenzung zu AML.

<sup>106</sup> Aras Corporation. Aras Innovator 9.3 - Programmers Guide. URL: <http://www.aras.com/support/documentation/9.3.0/Other%20Documents/Aras%20Innovator%209.3%20-%20Programmers%20Guide.pdf> (besucht am 18. 07. 2013).

Die von Aras bevorzugte Möglichkeit zur Datenübertragung ist die IOM API. Auf diese Weise muss der Entwickler nur kleine Teile des AML selbst erstellen und die Kommunikation ist weniger fehleranfällig. Allerdings kann diese Möglichkeit nur in Microsoft-Entwicklungsumgebungen genutzt werden. In der Theorie ist eine Nutzung der .NET-DLL in Mono, und damit plattformübergreifend, denkbar<sup>107</sup>. Dies ist jedoch mit zusätzlichem Einrichtungsaufwand verbunden, wurde nicht getestet und könnte in künftigen Innovator- oder Mono-Versionen nicht mehr funktionieren. Die IOM-Bibliothek ist nur ein Wrapper um die H-SOAP-Schnittstelle. Im Hintergrund wird aus den IOM-Befehlen AML zusammengesetzt und in SOAP gehüllt an den Server übertragen.

Die SOAP-Schnittstellen hingegen sind plattform- und programmiersprachenübergreifend nutzbar. Zum einen bietet sich die Möglichkeit, die bestehende Hauptschnittstelle (H-SOAP) zu nutzen. Über die Hauptschnittstelle wird ausschließlich in AML kommuniziert. Alle Funktionen des Servers können darüber genutzt werden.

Zum anderen kann man am Server neue Schnittstellen einrichten (SE-SOAP). Diese akzeptieren dann nicht mehr das gesamte Funktionsspektrum, sondern nur den Teil, den man am Server gewählt hat. Man muss kein AML schreiben, sondern kann jeweils mit einer Funktion ein Objekt (und dessen Beziehungen) anlegen, ändern oder löschen.

Eines der Argumente für eine eigenständige Anwendung außerhalb des Aras Innovators war im vorherigen Kapitel zu finden: keine praktischen Programmierkenntnisse in C# sind vorhanden. Diese oder Kenntnisse in .NET-Entwicklung wären nötig, um die IOM-Bibliothek zu nutzen. Stattdessen werden die programmiersprachenunabhängigen Alternativen H-SOAP und SE-SOAP betrachtet. Die SE-SOAP-Schnittstelle ist gut dokumentiert<sup>108</sup>, während bei H-SOAP nur bekannt ist, dass ausschließlich AML übertragen werden kann. Daher war SE-SOAP die erste Wahl für die technische Konzeption.

Jedoch zeigte sich im Laufe der Entwicklung eine große Schwachstelle in der Nutzung von SE-SOAP. Deshalb wurde die Entwicklung unterbrochen, große Code-Teile verworfen und auf H-SOAP umgestellt. Das Problem mit dieser Schnittstelle ist, dass sie nicht WSDL-beschrieben ist. Das bedeutet, während normale SOAP-Schnittstellen eine Beschreibung mitliefern, welche Funktionen an dieser Schnittstelle aufgerufen werden können, welche Parameter man angeben muss, und welche Daten man zurückbekommt, ist diese Schnittstelle gar nicht beschrieben. Wenn man jedoch die externe Software "AML Studio" nutzt, kann man durch Probieren lernen, wie die Schnittstelle nutzbar ist. Im

---

<sup>107</sup> stack exchange inc. Using Precompiled .NET Assembly DLL in Mono? URL: <http://stackoverflow.com/q/1160722> (besucht am 21. 07. 2013).

<sup>108</sup> Aras Corporation. Aras Innovator 9.3 - Programmers Guide. URL: <http://www.aras.com/support/documentation/9.3.0/Other%20Documents/Aras%20Innovator%209.3%20-%20Programmers%20Guide.pdf> (besucht am 18. 07. 2013).

Wesentlichen ist die Methode “applyAML( param1 )” vollkommen ausreichend. “param1” muss valides AML sein und die Methode liefert AML zurück. Die AML-Syntax ist im Aras Entwicklerhandbuch dokumentiert<sup>109</sup>. Die H-SOAP-Schnittstelle kann unter folgender URL erreicht werden:

`http://[Serveradresse]/InnovatorServer/Server/InnovatorServer.aspx`

### Konzeption

In diesem Kapitel werden die Planungsschritte für die Implementierung erläutert. Es werden die Wahl der Programmiersprache, die Ablaufmodellierung und das Klassendiagramm vorgestellt. Während der Umsetzung kam es zu einigen Anpassungen an den letzten beiden Punkten.

#### Wahl der Programmiersprache

Vor der technischen Konzeption steht die Frage der zu wählenden Programmiersprache. Die Konzeption ist losgelöst von der Implementierung und könnte daher programmiersprachenunabhängig passieren. Jedoch weisen alle Sprachen bestimmte Eigenarten auf, die bei der Modellierung beachtet werden müssen. In PHP z.B. kann man ohne Objektorientierung eine Liste von Tripeln erstellen, während man in Java JavaBeans, also eigene Klassen, dafür verwenden muss. Und z.B. in Python gibt es keinen Unterschied zwischen Interfaces und abstrakten Klassen.

Kriterien für die Wahl der Programmiersprache sind (a) Kenntnis des Autors, (b) Sprachkomplexität, (c) Güte von verfügbaren SOAP-Bibliotheken und (d) Einsatzaufwand als Desktop-Anwendung. Nach Kriterium (a) sind folgende Sprachen erlaubt: C++, Java, JavaScript, PHP und Python. Durch den hohen relativen Entwicklungsaufwand gegenüber den Scriptsprachen entfallen Java und C++. JavaScript entfällt nach Kriterium (d), da es zwar für viele Plattformen vorgefertigte Binärdateien zur Ausführung von JavaScript-Code gibt, jedoch fehlt die praktische Erfahrung im Umgang mit diesem Code. Bei der Untersuchung von Python hat sich ergeben, dass Python 2 nicht weiterentwickelt und von der Nutzung abgeraten wird. Stattdessen wird nahegelegt, auf Python 3 umzusteigen<sup>110</sup>. Alle Python-SOAP-Bibliotheken benötigen jedoch Python 2<sup>111</sup>. PHP hingegen enthält in der Standardbibliothek der Sprache

---

<sup>109</sup> Aras Corporation. Aras Innovator 9.3 - Programmers Guide. URL: <http://www.aras.com/support/documentation/9.3.0/Other%20Documents/Aras%20Innovator%209.3%20-%20Programmers%20Guide.pdf> (besucht am 18. 07. 2013).

<sup>110</sup> Python Software Foundation. Should I use Python 2 or Python 3 for my development activity? URL: <http://wiki.python.org/moin/Python2orPython3> (besucht am 23. 07. 2013).

<sup>111</sup> stack exchange inc. What's the best SOAP library for Python 3.x? URL: <http://stackoverflow.com/questions/7817303/whats-the-best-soap-library-for-python-3-x> (besucht am 23. 07. 2013).

bereits einen SOAP-Client<sup>112</sup> und bietet eine weitere Möglichkeit über das Zend Framework 2<sup>113</sup>. Weiterhin lässt sich PHP sehr gut als Desktop-Applikation einsetzen, allein durch die Installation der offiziellen PHP-Laufzeitumgebung<sup>114</sup>. Der PHP-Interpreter steht danach als Kommandozeilenprogramm zur Verfügung und kann Skripte ausführen. Daher wurde PHP als Programmiersprache gewählt. Bei der Wahl der Versionsnummer gibt es keine Einschränkungen, weshalb die aktuelle Serie (5.5) gewählt wurde.

#### Ablaufplan und Datenstruktur

Zum Entwurf des Programmablaufplans ist es wichtig, den allgemeinen Ablauf zu kennen, in dem die Anwendung eingesetzt werden soll. Dies wurde in (Zinke et al. 2013) bereits beleuchtet und wird hier kurz zusammengefasst. Zu beachten ist, dass es, wie bei der computergestützten Modellierung von Services, auch bei Produktdaten eine Trennung zwischen Produktmodell und Produktkonfiguration gibt. Produktmodelle enthalten, wie auch Servicemodelle, jeweils alle verfügbaren Optionen bzw. Wahlmöglichkeiten. Produktkonfigurationen basieren auf Produktmodellen. Bei ersteren wurden einige der Modell-verfügbaren Optionen ausgewählt. Daher ist nur noch eine Teilmenge der Optionen enthalten.

Abbildung 34 stellt eine Übersicht über den allgemeinen Produkt-Dienstleistungs-Kopplungsprozess aus (Zinke et al. 2013) dar. Sie visualisiert den Prozess, in welchem Produkt- und Dienstleistungsmodelle separat erstellt werden. Sobald beide vorhanden sind, können die Dienstleistungsmodelle an die Produktmodelle “gebunden”, also als Wahlmöglichkeit hinterlegt werden. Wenn ein Kundenauftrag eingeht, können die beiden Modelle separat instanziiert und konfiguriert werden.

---

<sup>112</sup> The PHP Group. PHP: SoapClient - Manual. URL: <http://www.php.net/manual/en/class.soapclient.php> (besucht am 23. 07. 2013).

<sup>113</sup> Zend Technologies Ltd. Zend/Soap/Client - Zend Framework 2 2.2.2 documentation - Zend Framework. URL: <http://framework.zend.com/manual/2.2/en/modules/zend.soap.client.html> (besucht am 23. 07. 2013).

<sup>114</sup> The PHP Group. PHP: Introduction - Manual. URL: <http://at1.php.net/manual/en/features.commandline.introduction.php> (besucht am 23. 07. 2013).

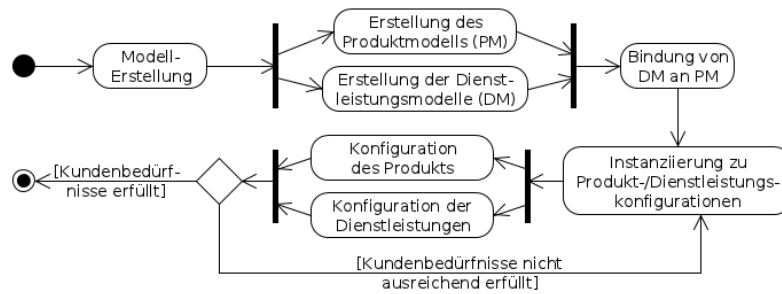


Abbildung 34 Diese Abbildung stammt aus Zinke et al. 2013 und wurde aus dem Englischen übersetzt. Abgebildet ist ein UML2-standardisiertes Aktivitätsdiagramm. Dieses stellt die Schritte in dem Prozess dar, für den die hier entwickelte Software zum Einsatz kommen soll. Alle abgebildeten Aktivitäten sollen softwaregestützt stattfinden. In dem Schritt “Bindung von DM an PM” ist gemeint, dass zu den Produktmodelldaten hinterlegt wird, welche Dienstleistungsmodelle gewählt werden können.

Dieser Prozessablauf und die Wahl der beteiligten Softwaresysteme impliziert eine naive Datenstruktur, die in Abbildung 35 dargestellt ist.

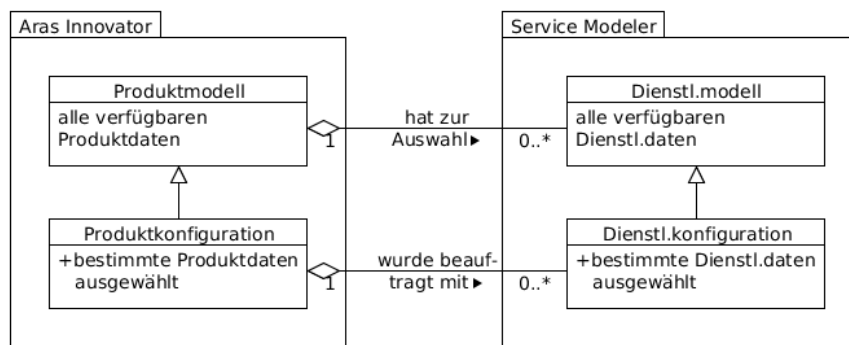


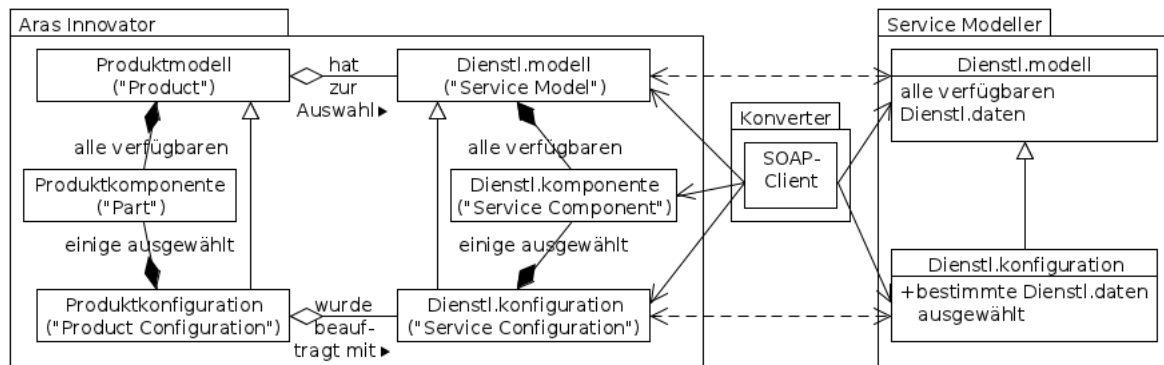
Abbildung 35 Die minimal notwendigen Komponenten zur Modellierung des Ablaufes, wie in Abbildung 12 dargestellt. Die Grafik hat das Format eines UML2-Klassendiagramms mit eingezeichneten Paketen für die zugehörigen Systeme.

Dieses Datenformat erfüllt für die Implementierung jedoch nicht das Kriterium der Exemplarität, da ein wesentlicher Teil in beiden Systemen fehlt. Alle in Abbildung 35 sichtbaren Datentypen sind zusammengesetzt, entweder aus Produkt- oder Dienstleistungskomponenten. Diese müssen bei der Implementierung auch berücksichtigt werden.

Der Service Modeller verfügt nicht über die Möglichkeit, URLs zu erstellten Modellen oder Konfigurationen anzubieten. Da es also keine Möglichkeit gibt, innerhalb des Aras Innovators eindeutig auf Dienstleistungsmodelle oder -konfigurationen zu verlinken, müssen die Dienstleistungsdaten auch im Innovator hinterlegt werden. Dies stellt die Hauptaufgabe der Implementierung dar: Objekte aus dem Service Modeller extrahieren und je nach Bedarf, also bei Neuerstellung, Aktualisierung, etc., in den Aras Innovator einfügen. Die überarbeitete Datenstruktur dieses Szenarios ist in Abbildung 36 dargestellt. Der Aras Innovator verfügt bereits über zahlreiche Objektklassen (im Innovator “ItemTypes”), darunter “Product” und “Part”. Diese beiden erfüllen den gleichen Zweck wie



Produktmodell und -komponenten und stehen bereits in einer assoziativen Relation. Sie können also direkt für den Zweck dieser Arbeit herangezogen werden. Die anderen Datentypen, Product Configuration, Service Model, Service Component und Service Configuration sind noch nicht vorhanden und müssen erstellt werden. Weiterhin müssen sie, wie abgebildet, mit Relationen verknüpft werden.

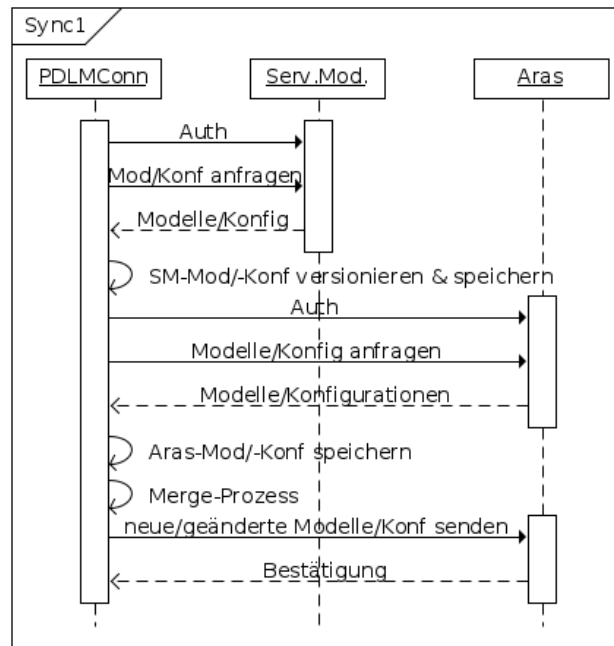


**Abbildung 36** Diese Abbildung wurde aus Zinke et al (2013) übersetzt. Angezeigt wird die modifizierte Datenstruktur, gegenüber Abbildung 35, mit den beteiligten Systemen in einem UML2-Klassendiagramm mit verzeichneten Paketen für Systeme. Weiterhin wurde der Konverter zwischen Innovator und Service Modeller eingezeichnet, um die Rolle zu verdeutlichen. Zu beachten ist, dass im Service Modeller die Dienstleistungskomponenten implizit enthalten sind. Das bedeutet, sie sind gegenüber dem Aras Innovator, nicht als eigenständige Datenstruktur verfügbar, sondern nur als Teil von entweder Modell oder Konfiguration

Der Konverter muss die Objekte des Service Modellers mittels SOAP auslesen. Das Format, in dem die Daten ankommen, ist in jedem Falle XML, da dies vom SOA-Protokoll gefordert ist. Weiterhin sollten vom Innovator die gleichen (Service-)Klassen durchsucht werden, ob eventuell schon Dienstleistungsdaten vorhanden sind.

In der Praxis kann es auch von Bedeutung sein, den Fall zu behandeln, dass im Service Modeller Objekte gelöscht wurden. Diese können nicht gefahrlos auch im Innovator gelöscht werden, denn sie könnten in Produktmodellen oder -konfigurationen bereits eingebunden sein. Daher müssen diese Daten auch abgefragt werden. Vom Aras-System kommen die Daten ebenfalls in XML bzw. in dem Dialekt AML. Um die Daten aus beiden Systemen vergleichbar zu machen, müssen diese in eine interne Datenstruktur umgewandelt werden. Hierfür wurden PHP-Objekte gewählt, da deren Attribute für den Vergleich leicht zugänglich sind und die Objekte als Ganzes serialisierbar sind.

Sind die Daten aus beiden Systemen als PHP-Objekte im gleichen Format verfügbar, können sie verglichen und das Changeset (deutsch: Änderungsmenge) errechnet werden. Dieses Changeset kann im Anschluss wieder über SOAP auf den Aras Innovator angewendet werden. Dieser Ablauf ist in Abbildung 37 dargestellt.



**Abbildung 37** Dieses UML2-Sequenzdiagramm stellt den angedachten Programmablauf der Konverter-Anwendung dar. Diese wurde in dieser Grafik “PDLmconn” bezeichnet. Der Ablauf ist dreiphasig und umfasst die beiden Systeme Service Modeller (“Serv.Mod.”) und Aras Innovator (“Aras”). Die drei Phasen sind das Laden der Daten aus beiden Systemen, das Errechnen des Changesets (hier als “Merge-Prozess” beschrieben) und seine Anwendung auf das PLM-System.

Anhand der allgemeinen Ablaufkonzeption wurde die Programmstruktur des Konverters entwickelt. Im folgenden Kapitel werden die Klassen erläutert, die vermutlich notwendig sind, um die beschriebene Funktionalität zu implementieren.

### Klassendiagramm

An der Universität Leipzig wird überwiegend objektorientierte Programmierung gelehrt, weshalb auch in dieser Arbeit das Paradigma angewendet wird. Das Unternehmen, welches für die Entwicklung von PHP verantwortlich ist, die Zend Technologies Ltd. (Zend Technologies Ltd. 2013), hat einen Leitfaden für die Benennung von Klassen, Attributen, Methoden, usw. in PHP herausgegeben (Portnow 2013). Bei der nachfolgenden Klassen-Modellierung werden diese Standards beachtet.

Nachdem der Ablauf bekannt ist, müssen die Klassen für die Konverter-Anwendung modelliert werden. Begonnen wird bei den Klassen, in denen die Daten der beiden Systeme abgelegt und verglichen werden. Dies betrifft die sechs Klassen:

- ProductComponent (für Produktkomponenten)
- ProductConfig (für Produktkonfigurationen)
- ProductModel (für Produktmodelle)
- ServiceComponent (für Dienstleistungskomponenten)

- ServiceConfig (für Dienstleistungskonfigurationen)
- ServiceModel (für Dienstleistungsmodelle)

Weiterhin wurde die Klasse “Customer” hinzugefügt und zu Product- sowie ServiceConfig in Relation gesetzt. Damit wird angedeutet, dass jede Produkt- und Dienstleistungskonfiguration zu einem Kunden gehört. Jede der vorgenannten Klassen wurde mit einer Reihe von Attributen ausgestattet. Die Attribute der Service-Klassen sind die Schnittmenge der Attribute, die von beiden beteiligten Systemen zur Verfügung gestellt wurden. Die Produkt-Klassen wurden analog modelliert und zusätzlich mit Attributen versehen, welche die Dienstleistungen referenzieren. Bei den Config-Klassen wurden weiterhin Datumsangaben für den Verkauf eingebunden, um die praktische Situation besser zu reflektieren und Abbildung 32 gerecht zu werden. Um die Produkt- und Service-Klassen jeweils zu gruppieren, wurde für beide eine abstrakte Oberklasse eingeführt, AbstractProductObject und AbstractServiceObject. Um alle Daten zu gruppieren, die über SOAP-Schnittstellen ausgetauscht werden, wurde eine nächsthöhere abstrakte Klasse AbstractPdlmObject eingeführt. Diese vereint unter sich die beiden abstrakten und die Customer-Klasse. Die vorläufige Klassenstruktur ist in Abbildung 38 dargestellt.

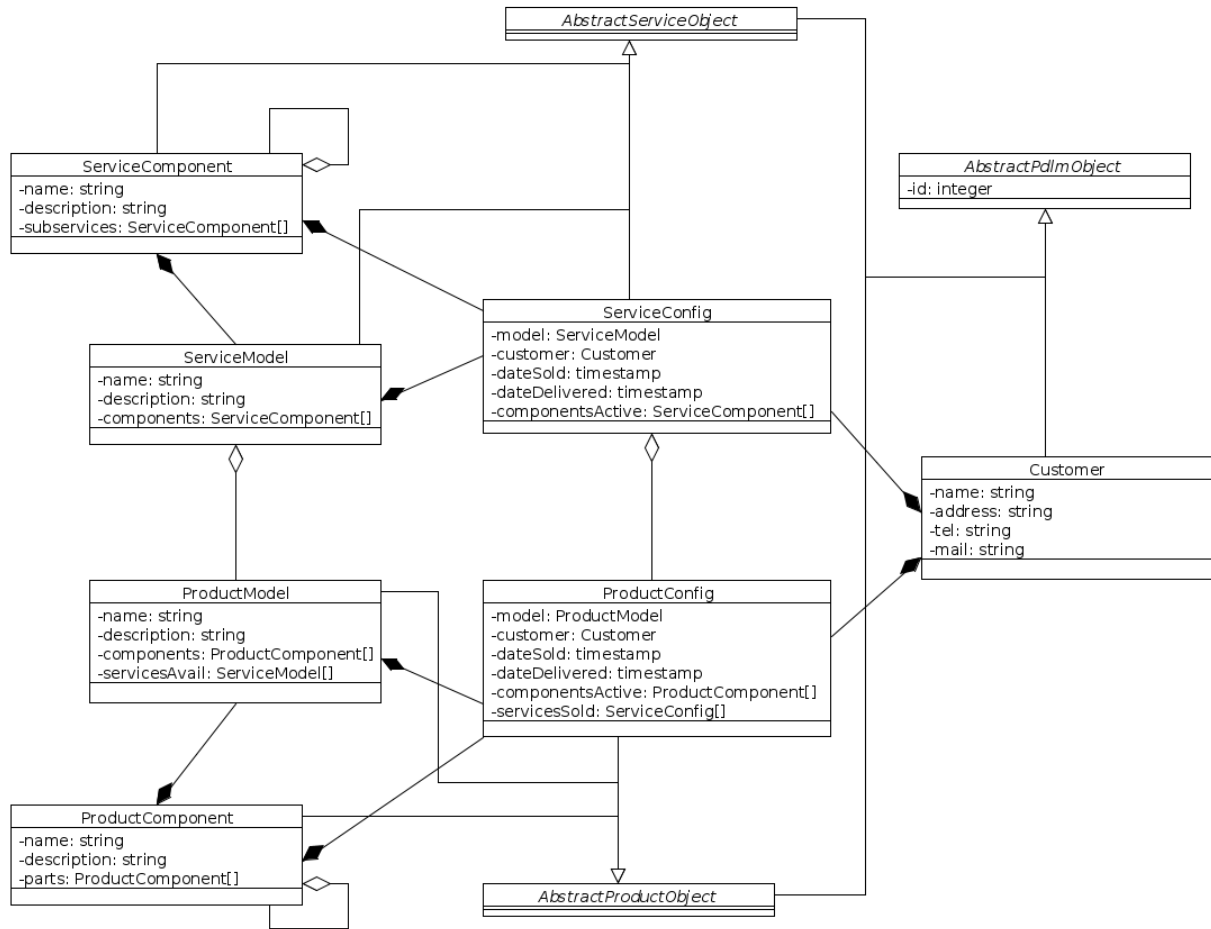


Abbildung 38 Dargestellt ist das UML2-Klassendiagramm für die Daten aus den beiden Systemen Aras Innovator und Service Modeller. Über allem steht die abstrakte Klasse AbstractPdlmObject (rechts oben). Darunter teilt sich der Klassenbaum in die Produkt- und Service-Klassen (jeweils oben mittig und unten mittig) und weiterhin in die einzelnen nicht-abstrakten Klassen. Produkt- und ServiceComponents sind selbstreferenziell, denn sie können Unter-Elemente des eigenen Typs enthalten.

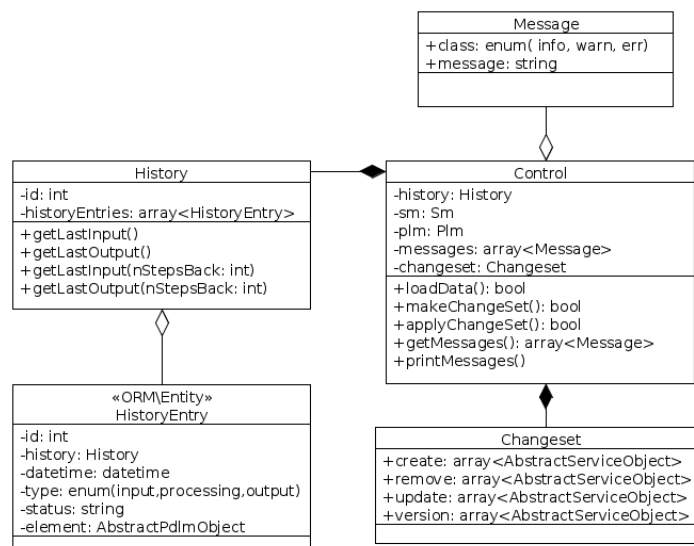
Neben den reinen Datenaustauschklassen werden weiterhin Klassen benötigt, die den Programmfluss regulieren. Dies erfordert mindestens eine Hauptklasse (“Control”), welche die drei Phasen starten kann. Die Phasen lauten: Daten aus beiden Systemen laden (“loadData”), Changeset errechnen (“makeChangeSet”) und Changeset anwenden (“applyChangeSet”). Diese sind als Methoden der Control-Klasse zugeordnet. Um das Changeset erstellen zu können, muss eine Historie darüber geführt werden, welche Daten aus welchem System gekommen sind. Hierzu wird eine Klasse “History” eingeführt mit zugehörigen Einträgen HistoryEntry. Die History-Klasse selbst ist nur ein Container für die Einträge und soll diese durchsuchen können, möglicherweise auch Objekte aus der Vergangenheit, um z.B. Löschungen festzustellen. HistoryEntry-Objekte (HE) speichern jeweils ein AbstractPdlmObject, eingehend oder ausgehend aus einem der Systeme, also z.B. eine ServiceConfig oder eine ProductComponent. Objekte dieser HE-Klasse sollten daher in der Datenbank abgelegt werden

und das jeweilige Element referenzieren, sowie den Typ (“eingehend”, “in Bearbeitung” oder “ausgehend”) und das System aus dem sie kommen (hier im “status”-Attribut festgehalten).

Um einen Container für die Changeset-Anweisungen zu bieten, wurde eine Klasse “Changeset” angelegt und mit vier Listen versehen: jeweils eine für zu erstellende Objekte (“create”), zu löschende Objekte (“remove”), zu aktualisierende Objekte (“update”) und eine spezielle Variante der Aktualisierung, zu versionierende Objekte (“version”). Diese letzte Unterscheidung wird durchgeführt, da mit “update” das Überschreiben der Daten gemeint ist, während “version” eine neue Version des Datensatzes erstellt, den alten jedoch bewahrt.

Um eine Art einfaches Logbuch zu bieten, wurde eine Hilfsklasse “Message” erstellt, die, ähnlich einem JavaBean, lediglich die Art der Nachricht (Information, Warnung oder Fehler) und den Nachrichtentext enthält. Diese Klasse kann vom Hauptprogramm genutzt werden, um den Fortschritt zu speichern und Kontrolle darüber zu haben, wann dieser ausgegeben wird.

Diese Steuerklassen wurden in einem separaten Klassendiagramm erstellt (Abbildung 39) und danach in die Gesamtansicht zusammengeführt (Abbildung 40). In diesen beiden Abbildungen ist nun auch eingezeichnet, welche Klassen in der Datenbank gehalten werden. Die betreffenden Klassen sind mit dem Stereotyp “<<ORM\Entity>>” versehen, da dies die Markierung des objektrelationalen Mappers “Doctrine 2” ist, der später Verwendung findet. Weitere Informationen zu Datenbank und Mapping folgen im nächsten Kapitel.



**Abbildung 39** Angezeigt wird der Abschnitt des UML2-Klassendiagramms, welcher die Steuer- bzw. Hilfsklassen beinhaltet. In den Klassen Changeset und Message wurden die Attribute bewusst als “public” markiert, um schnelleren Zugriff zu erlauben, da die zugehörigen Objekte vermutlich häufig manipuliert werden. Diese Darstellung ist ein Bestandteil von Abbildung 40, wurde aber aus Gründen der Übersichtlichkeit auch separat dargestellt.

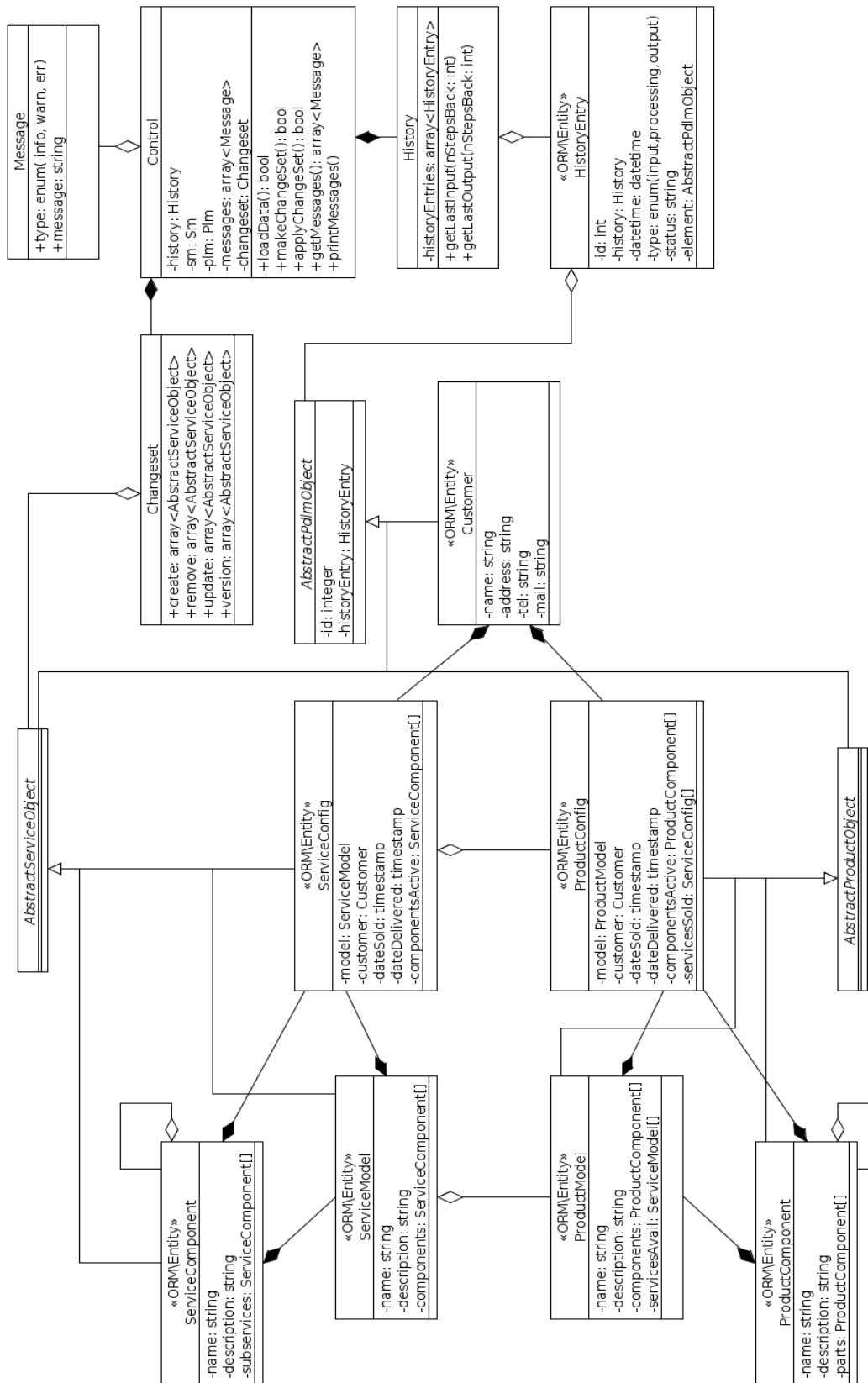


Abbildung 40 Die Grafik zeigt das komplette UML2-Klassendiagramm der Klassen, welche für die Konverter-Anwendung benötigt werden. Es wurde hochkant in diese Arbeit eingebunden, um den Platz

auf der Seite besser auszufüllen und lesbar zu bleiben. Im linken Teil sind die Datenaustauschklassen aus Abbildung 38 und im rechten Teil die Hilfsklassen aus Abbildung 39 angeordnet.

### Implementierung

Im Folgenden wird der Implementierungsprozess dokumentiert. Dabei wird Wert auf Reproduzierbarkeit gelegt. Um diese zu gewährleisten, wird auf die verwendeten Systeme und Softwareressourcen eingegangen. Gefolgt wird dies von der Dokumentation des prinzipiellen Entwicklungsprozesses und der Beschreibung des tatsächlichen Vorgehens bei der Implementierung. Abschließend wird auf Schwierigkeiten während des Prozesses eingegangen, damit diese von weiterführenden Projekten umgangen werden können.

### Verwendete Systeme

Dieses Kapitel listet die wesentlichen Software-Merkmale der verwendeten Systeme auf. Für die weitere Nutzung der entwickelten Anwendung wird davon ausgegangen, dass mindestens diese Voraussetzungen in den betreffenden Systemen erfüllt sein müssen, damit eine korrekte Funktionsweise gewährleistet werden kann. Möglicherweise sind auch ältere Versionen der beteiligten Softwareprodukte ausreichend, aber dies wurde nicht getestet.

#### Aras-Innovator-Server:

- Betriebssystem: Microsoft Windows Server 2008 R2 Enterprise, 64 Bit, SP1<sup>115</sup>
- Software:
  - Microsoft IIS 7.5.7600<sup>116</sup>
  - Microsoft SQL Server 2008 R2, 64 Bit<sup>117</sup>
  - Microsoft .NET Framework 4<sup>118</sup>
  - Aras Innovator 9.3.0 Build 5711A<sup>119</sup>

#### Service-Modeller-Server:

- Betriebssystem: Windows Server 2008 R2 Enterprise, 64 Bit<sup>120</sup>

---

<sup>115</sup> Microsoft Corporation. Windows Server 2008 R2 Standard/Enterprise. URL: <http://www.microsoft.com/de-de/kmu/Produkte/Seiten/Windows-Small-Business-Server-2011-Standard-Enterprise.aspx> (besucht am 28. 07. 2013).

<sup>116</sup> LLC. Neudesic. IIS Website. URL: <http://www.iis.net> (besucht am 28. 07. 2013).

<sup>117</sup> Microsoft Corporation. Microsoft SQL Server Website. URL: <http://www.microsoft.com/de-de/server/sql-server/2012/default.aspx> (besucht am 28. 07. 2013).

<sup>118</sup> Microsoft Corporation. Microsoft .NET Framework 4. URL: <http://www.microsoft.com/de-de/download/details.aspx?id=17718> (besucht am 28. 07. 2013).

<sup>119</sup> Microsoft Corporation. Microsoft .NET Framework 4. URL: <http://www.microsoft.com/de-de/download/details.aspx?id=17718> (besucht am 28. 07. 2013).

<sup>120</sup> Microsoft Corporation. Windows Server 2008 R2 Standard/Enterprise. URL: <http://www.microsoft.com/de-de/kmu/Produkte/Seiten/Windows-Small-Business-Server-2011-Standard-Enterprise.aspx> (besucht am 28. 07. 2013).

- Software:
  - Microsoft IIS 7 (Die genaue Version ist unbekannt. Es handelt sich um eine virtuelle Maschine (VM) vom Institut für Informatik der Universität Leipzig. Bei diesen VMs sind automatische Updates voreingestellt. Und daher wird davon ausgegangen, dass diese Software aktuell gehalten wird, Stand 24.07.2013)<sup>121</sup>
  - Microsoft SQL Server 2008 Enterprise Edition, 64 Bit<sup>122</sup>

#### Service-Modeller-Client:

- Betriebssystem: Microsoft Windows XP SP2 oder neuer
- Software:
  - Beliebiger Browser, insofern er das Silverlight-Plugin unterstützt
  - Silverlight-Browser-Plugin Version 5 oder neuer

#### Anwendungsentwicklung:

- Betriebssystem: Linux Mint 15 "Olivia", 64 Bit<sup>123</sup>
- Software
  - XAMPP für Linux 1.8.3<sup>124</sup>
  - PHP 5.5.0 (cli)
  - Zend Framework 2.2.1<sup>125</sup>
  - zsh 4.3.11<sup>126</sup>

#### Beschreibung und Einrichtung Entwicklungsumgebung

PHP-Anwendungen lassen sich in allen gängigen Betriebssystemen (Mac OS, Linux und Windows) entwickeln, da die Laufzeitumgebung vorkompiliert bereitsteht und nur noch installiert werden muss. Ebenso stehen die beiden großen Entwicklungsumgebungen Eclipse<sup>127</sup> und Netbeans IDE<sup>128</sup> als Installationspakete für zahlreiche Plattformen bereit. In der Regel wird nicht nur PHP installiert, sondern auch eine komplette Webentwicklungsumgebung mit Webserver und Datenbank. Dies kann unter Umständen zu Schwierigkeiten bei der Konfiguration führen, da die Systeme aufeinander abgestimmt werden müssen. Für dieses Problem gibt es eine Reihe Alternativen, darunter das vorkonfigurierte Paket

---

<sup>121</sup> LLC. Neudesic. IIS Website. URL: <http://www.iis.net> (besucht am 28. 07. 2013).

<sup>122</sup> Microsoft Corporation. Microsoft SQL Server Website. URL: <http://www.microsoft.com/de-de/server/sql-server/2012/default.aspx> (besucht am 28. 07. 2013).

<sup>123</sup> Linux Mint. Linux Mint Website. URL: <http://www.linuxmint.com/index.php> (besucht am 28. 07. 2013).

<sup>124</sup> Apache Friends. XAMPP für Linux. URL: <http://www.apachefriends.org/de/xampp-linux.html> (besucht am 28. 07. 2013).

<sup>125</sup> Zend Technologies Ltd. Zend Framework 2 Website. URL: <http://framework.zend.com/> (besucht am 28. 07. 2013).

<sup>126</sup> Zsh Web Page Maintainers. Z Shell. URL: <http://zsh.sourceforge.net> (besucht am 28. 07. 2013).

<sup>127</sup> The Eclipse Foundation. Eclipse - The Eclipse Foundation open source community website. URL: <http://www.eclipse.org/> (besucht am 25. 07. 2013).

<sup>128</sup> Oracle. Welcome to NetBeans. URL: <https://netbeans.org/> (besucht am 25. 07. 2013).



“XAMPP” von [apachefriends.org](http://www.apachefriends.org)<sup>129</sup>. Dieses muss nur entpackt werden und stellt einen sofort einsatzbereiten Apache-Webserver mit PHP und MySQL zur Verfügung. Weiterhin sind zahlreiche PHP-Bibliotheken und Erweiterungen enthalten, darunter auch gängige Datenbank-Treiber (z.B. MySQL und SQLite).

Für die Entwicklung wurde aus persönlicher Präferenz Linux, Netbeans IDE und XAMPP gewählt. Diese Wahl ist willkürlich in dem Sinne, dass die anderen Optionen keine signifikanten Verbesserungen oder Verschlechterungen für dieses Projekt mit sich gebracht hätten. XAMPP ist unnötig, da nur PHP auf der Kommandozeile benötigt wird. Jedoch war zu Projektbeginn nicht klar, dass auf den Webserver komplett verzichtet werden kann. Das Betriebssystem war bereits installiert, wurde jedoch aus Sicherheitsgründen durch automatische Updates auf den neuesten Stand gebracht.

Nachdem das XAMPP-für-Linux-Paket heruntergeladen und mit Standardeinstellungen installiert wurde, wurde das Binärverzeichnis zum Systempfad hinzugefügt, um Befehle wie

```
> php -v
```

zu erlauben, gegenüber

```
> /opt/lampp/bin/php -v
```

Daneben wurde “Composer”, das Paketverwaltungsprogramm für PHP, systemweit installiert<sup>130</sup>. Das Programm wird häufig in Projekten verwendet, die Fremdbibliotheken einbeziehen und ermöglicht eine schnelle Installation und semi-automatische Updates. Bei diesem Projekt wurde davon ausgegangen, dass Fremdbibliotheken zum Einsatz kommen könnten. Außerdem wurde bereits an das Zend Framework 2 als “Softwarebaukasten” gedacht, welches auch Composer einsetzt. Netbeans bietet einen Installationsassistenten auf der offiziellen Website an, der in der PHP-Variante heruntergeladen und ausgeführt wurde. Nach der Installation wurde die Entwicklungsumgebung gestartet und konfiguriert. Letzteres bedeutet, dass in den Optionen die Pfade für die ausführbaren Dateien von PHP und Composer gesetzt wurden. Weiterhin wurde Mercurial<sup>131</sup> als Versionierungswerkzeug installiert und in der Konfigurationsdatei der Benutzername angegeben. Um den entwickelten Code öffentlich bereitzustellen und externe Backups zu erlauben, wurde entschieden, ein Online-Code-Repository [Glossar] einzurichten und zu nutzen. Für Mercurial bietet der Dienst Bitbucket<sup>132</sup> gutes, kostenloses Hosting. Gegenüber anderen Diensten bietet Bitbucket auch kostenlose “private” Repositories. Das bedeutet, dass man jederzeit umstellen kann, ob andere öffentlich den Quellcode sehen dürfen oder nicht. Wenn der

---

<sup>129</sup> Apache Friends. XAMPP. URL: <http://www.apachefriends.org/de/xampp.html> (besucht am 25. 07. 2013).

<sup>130</sup> Composer. Composer Website. URL: <http://getcomposer.org/doc/00-intro.md> (besucht am 25. 07. 2013).

<sup>131</sup> Mercurial Community. Mercurial SCM. URL: <http://mercurial.selenic.com> (besucht am 25. 07. 2013)

<sup>132</sup> Inc. Atlassian. Free source code hosting for Git and Mercurial by Bitbucket. URL: <https://bitbucket.org> (besucht am 25. 07. 2013)

Quellcode in jedem Stadium der Entwicklung offen liegt, kann das bei Software, die sich bei Fremddiensten anmeldet, möglicherweise Sicherheitslücken hervorrufen oder Angriffe begünstigen. Denn es kann vorkommen, dass in frühen Entwicklungsstadien Zugangsdaten für die Fremdsysteme im Code enthalten sind. Es ist schlechter Programmierstandard, Logindaten in größeren Programmteilen, und nicht in separaten Konfigurationsdateien zu speichern, aber es kann, aus Unachtsamkeit oder zum Testen, durchaus vorkommen.

Wenn das Entwicklungsprojekt eine bestimmte Größe überschreitet und wenn bestimmte wiederkehrende Funktionen benötigt werden, macht man sich ein Framework zu Nutze. Solche Frameworks unterstützen die Entwicklung und bieten hilfreiche Funktionen für Probleme, die bei Standard-Projekten häufig auftreten können. Außerdem sind viele Frameworks erweiterbar und bei einigen gibt es eine breite Bibliothek an Funktionalitäten, die nur heruntergeladen und installiert werden müssen, wie z.B. Benutzerverwaltung. Für PHP gibt es eine Reihe etablierter Frameworks, siehe PHPFrameworks.com 2013)), McLeod (2013) und Romanenko (2013). Es ist wünschenswert, eines der Frameworks (FW) aus den jeweils besten 5 zu wählen, da es eine Verbindung gibt, zwischen Verbreitung des FW und Anzahl der Problemen mit dem FW zu denen Lösungen im Internet gefunden werden können. Damit ist gemeint, ein FW wird angenommen von drei Personen weltweit genutzt. Dann ist die Wahrscheinlichkeit sehr gering, dass man durch Internetsuche Lösungen zu den Problemen findet, in die man während der Entwicklung möglicherweise kommen kann. Im Umkehrschluss ist die Chance sehr groß, bestehende Lösungen finden zu können, wenn es eine sehr große Nutzergemeinschaft gibt, da die Chance höher ist, dass sich darin Nutzer befinden, die bereitwillig ihr Wissen teilen.

Zu den besten PHP-Frameworks gehören meist CakePHP (Inc. Cake Software Foundation 2013), Symfony (Sensio Labs 2013), Yii (Yii Software LLC. 2013) und das Zend Framework (Zend Technologies Ltd. 2013).

Für dieses Projekt wurde das Zend Framework 2 genutzt, da es einen einfachen alternativen SOAP-Client anbietet und das Kriterium der hohen Verbreitung erfüllt. Weiterhin haben eigene Erfahrungen mit dem FW gezeigt, dass es dem Entwickler sehr viele Freiheiten gibt, im Sinne von Konfigurationsmöglichkeiten und Einsatzszenarien. Es bietet außerdem Kommandozeilenunterstützung (Zend Technologies Ltd. 2013).

Um das Entwicklungsprojekt zu starten, wurde bei Bitbucket ein neues Repository angelegt (Bitbucket fgolemo 2013), dieses auf den Entwicklungscomputer heruntergeladen, Zend Framework 2 installiert und das Projektverzeichnis in Netbeans als neues Projekt geöffnet.

### Durchführung der Implementierung

In Vorbereitung wurden im Aras Innovator alle benötigten Datentypen angelegt und miteinander verknüpft, wie in Abbildung 14 dargestellt. Die eigentliche Implementierung hat in 3 Schritten stattgefunden: Anbindung der beteiligten Systeme, Erstellung des Hauptprogrammes ohne die beteiligten Systeme und Zusammenführung des Hauptprogramms mit den externen Daten. Die genaue Entwicklung der Anwendung ist in der Commit-Historie des Bitbucket-Repositorys (Bitbucket fgolemo 2013) zu lesen und wird hier nur zusammengefasst. Die Innovator-ItemTypes wurden exportiert und sind ebenfalls im Repository verfügbar.

Zuerst wurde nur versucht, die Daten der externen Systeme über SOAP abzurufen und in eigene PHP-Objekte umzuwandeln. Dies hat ohne die zuvor genannten Steuerklassen stattgefunden. Der Prozess wurde zuerst beim Aras Innovator, dann beim Service Modeller vollzogen. Angefangen wurde damit, dass alle Funktionen und Daten in einer Klasse pro externem System vereint wurden. Nachdem die prinzipielle Funktionalität hergestellt war, wurde beim Aras Innovator die Hauptfunktion der Analyse des AMLs in eine eigene Klasse, den Aras\Reader ausgelagert. Dieser verfügt über die Fähigkeit, anhand einer ini-Konfigurationsdatei, beliebige Daten aus einer beliebigen Innovator-Installation auszulesen und in PHP-Objekte zu speichern. Damit soll gewährleistet werden, dass dieses Projekt auch in anderen Szenarien zum Einsatz kommen kann.

Im Anschluss, ab Commit 3034450, wurde mit dem Hauptprogramm begonnen, und damit, die selbstgeschriebenen Komponenten in die Verzeichnisstruktur des Zend Framework einzufügen. Das bedeutet, der Aras\Reader wird weiterhin als eigenständige Bibliothek in einem entsprechenden neutralen Verzeichnis belassen. Das restliche Programm wurde in ein Zend-Modul umgewandelt, welches per Kopieren und Einfügen auch in anderen Zend-Framework-Projekten genutzt werden kann. Besagte SOAP-Komponenten wurden gegenüber dem Hauptprogramm vorerst ruhen gelassen, da die prinzipielle Funktion in der ersten Phase bereits erprobt wurde. Das Hauptprogramm wurde mit Demonstrationsdaten implementiert und während des Vorgangs gab es eine Reihe Änderungen gegenüber der ursprünglichen Konzeption.

Enums (engl. "enumerations", Aufzählungen) sind Listen von Konstanten, wie z.B. eine Liste der deutschen Namen aller Wochentage oder aller Monate. Dieses Konzept kommt z.B. in Java vor<sup>133</sup>, wird jedoch von PHP nicht direkt unterstützt im Sinne von statischen Konstanten einer Klasse. Daher wurden Hilfsklassen eingeführt für die Klassen, die Enums benötigt hätten, Message und HistoryEntry. Die beiden Hilfsklassen MessageType und HistoryEntryType dienen dazu anzugeben, welche

---

<sup>133</sup> Oracle. Enum Types. URL: <http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html> (besucht am 25. 07. 2013).

wohldefinierte Dringlichkeit eine Message hat bzw. ob die HistoryEntries für eingehende oder ausgehende Daten sind.

Damit es möglich ist, die Klassen auszutauschen, die in diesem Projekt für die Kommunikation mit externen Systemen zuständig sind. Dabei wurden zwei neue Abstrakte Klassen eingeführt, AbstractReader und AbstractWriter. Diese definieren eine Reihe von Funktionen, die von beliebigen Klassen implementiert werden können, um einen lesenden oder schreibenden Zugriff auf ein externes PLM oder SM zu geben. Anders ausgedrückt, wenn man z.B. außer dem Service Modeller ein anderes SM-System nutzen möchte, muss man dafür nur eine neue Klasse erstellen, die alle AbstractReader-Methoden implementiert. Dies erlaubt erhöhte Wiederverwendbarkeit. Ähnliches gilt, wenn man andere PLM-Systeme nutzen möchte, jedoch muss hier zusätzlich der AbstractWriter implementiert werden. Beide neuen abstrakten Klassen erben von der weiteren neuen Oberklasse AbstractSoap, um für alle Reader- und Writer-Komponenten einheitliche Login-Mechanismen zur Verfügung stellen zu können. Alle Komponenten, die in Verbindung zur SOAP-Kommunikation stehen, wurden in ein eigenes Verzeichnis sowie eigenen PHP-Namespaces "Soap" verschoben.

In zahlreichen Frameworks und Software-Projekten kommt der Mechanismus "Dependency Injection" zum Einsatz (Potencier 2013). Dieser bietet höhere Flexibilität bei der Wiederverwendung von Klassen, in denen Variablen gesetzt werden können. Um diesen Mechanismus besser nutzen zu können, wurden zwei weitere Hilfsklassen eingeführt, Plm und Sm. Beide verfügen über ein Attribut, welches eine Instanz der eben genannten AbstractReader-Klasse beinhaltet und Plm-Objekte erhalten außerdem ein AbstractWriter-Attribut. Beide Objekte werden dem Control-Objekt beim Start übergeben.

In der Control-Klasse wurden die drei geplanten Verhalten implementiert. Dabei wurde Wert darauf gelegt, dass diese auch unabhängig voneinander aufgerufen werden können. Die erste Methode ruft die Objekte aus den Systemen ab und persistiert diese in der Datenbank. Wenn alle Schritte nacheinander ausgeführt werden, sind diese Ereignisse noch in der Historie gespeichert. Daraus kann das Changeset kalkuliert werden. Wenn die loadData-Methode separat ausgeführt wurde, stellen die nachfolgenden Schritte die Historie aus der Datenbank wieder her.

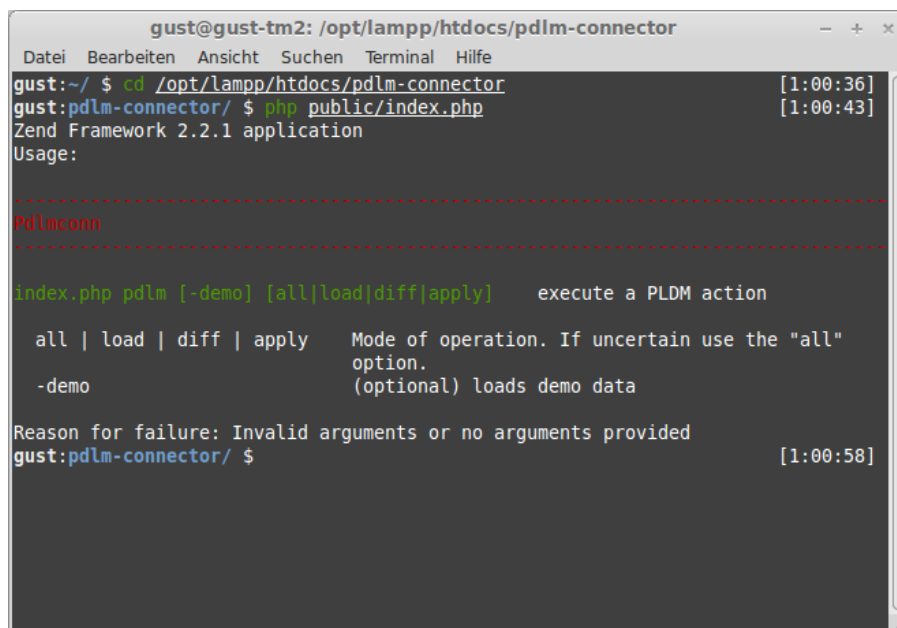
Bei der Changeset-Berechnung wurden vorerst nur die Fälle "create" und "update" implementiert. Jedoch sind die weiteren Fälle "remove" und "version" im Code markiert und können nachträglich eingefügt werden. Für die Berechnung wird zuerst geprüft, ob die jeweiligen Elemente ungefähr gleich sind, danach ob sie vollkommen gleich sind. Beide Prüfungen wurden in jedem der Service-Objekte einzeln implementiert, um bestmöglichen Vergleich zu gewährleisten. Bei Service-Komponenten ist z.B. ein hinreichendes Kriterium, dass der Name gleich ist. Für vollkommene Gleichheit werden außerdem die weiteren Attribute normal und die SubServices rekursiv geprüft. Dieses Verfahren ist jedoch anfällig für Namensänderungen. In zukünftigen Versionen könnte ein Schwellwert für die

Ähnlichkeit gegeben werden und jede Ähnlichkeit addiert einen Wert auf eine Punktzahl, die dann mit dem Schwellwert verglichen wird.

Es wurde festgestellt, dass die Customer-Klasse gänzlich unnötig für das Changeset ist. Die Produktdaten sind derzeit auch noch ohne Einfluss, da die remove-Funktion noch nicht implementiert ist. Für zukünftige Versionen könnten Einstellungen in einer ini-Datei gespeichert werden, wie in verschiedenen Szenarien in Bezug auf die Änderungen vorgegangen werden soll. Zum Beispiel steht die Frage offen, ob Dienstleistungsdaten kaskadierend gelöscht werden sollten. Das heißt, wenn man z.B. ein Dienstleistungsmodell gelöscht hat, dann auch Produktmodelle entfernt werden, in denen es vorkommt.

Für den letzten Schritt, den Versand der Änderungen, werden einfach die unterschiedlichen Changeset-Listen an den Writer des PLM-Objektes übergeben, welches diese je nach verwendetem System abarbeiten kann. Im Falle des Aras Innovators wird jedes Element der Liste in eine AML-Anfrage umformuliert und an die SOAP-Schnittstelle übergeben. Diese Funktion wird erneut durch eine selbstgeschriebene externe Bibliothek unterstützt, dem Aras\Writer, die nun auch von anderen Softwareprojekten genutzt werden kann.

Die drei Control-Aktionen werden von einem Zend-Controller gesteuert. Dieser ist eingerichtet, ausschließlich auf der Kommandozeile zu funktionieren. Wenn der Nutzer die Startseite auf der Konsole öffnet, wird die Syntax des Konnektor-Werkzeuges beschrieben. Dies ist in Abbildung 41 dargestellt.



```
gust@gust-tm2: /opt/lampp/htdocs/pdlm-connector
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
gust:~/ $ cd /opt/lampp/htdocs/pdlm-connector [1:00:36]
gust:pdlm-connector/ $ php public/index.php [1:00:43]
Zend Framework 2.2.1 application
Usage:
-----
Pdlmconn
-----
index.php pdlm [-demo] [all|load|diff|apply] execute a PLDM action

all | load | diff | apply Mode of operation. If uncertain use the "all"
                        option.
-demo                   (optional) loads demo data

Reason for failure: Invalid arguments or no arguments provided
gust:pdlm-connector/ $ [1:00:58]
```

Abbildung 41 Hier werden die möglichen Befehle für die Konverter-Anwendung dargestellt. Mit dem “-demo”-Präfix wird die Funktion der Anwendung demonstriert. Dabei wird ausschließlich auf Demodaten gearbeitet, keine Verbindung nach außen hergestellt und nichts in den externen Systemen geändert. Das

Hauptargument für den Aufruf ist eines von entweder “all”, “load”, “diff” oder “apply”. “all” steht dabei für alle nachfolgenden. Wenn kein Parameter übergeben wird, nur das “pdlm”-Wort, wird “all” angenommen. Jedes der anderen Argumente startet eine Phase der Control-Klasse.

Nachdem die Funktion des Steuerprogramms gewährleistet ist, wurde dieses mit den eigentlichen Daten aus den externen Systemen versorgt. Dazu wurden die anfangs testweise angelegten Klassen zur Kommunikation mit dem Innovator und Service Modeller refaktoriert, um in die neue Klassenhierarchie einzupassen.

Das Softwarewerkzeug steht nun als Kommandozeilenprogramm zur Verfügung (siehe Abbildung 42) und kann automatisch in bestimmten Zyklen ausgeführt werden. Dies wird z.B. unter Linux durch Cron Jobs ermöglicht<sup>134</sup>. Das Ergebnis ist eine Synchronisation der Dienstleistungskomponenten, Dienstleistungsmodelle und Dienstleistungskonfigurationen vom Service Modeller in den PLM Aras Innovator.

```

gust@gust-tm2: /opt/lampp/htdocs/pdlm-connector
Datei Bearbeiten Ansicht Suchen Terminal Hilfe

gust:pdlm-connector/ $ php public/index.php pdlm -demo all [3:39:19]
info: ### start loading data
info: found 3 SM objects
info: found 2 PLM objects
info: ### end loading data
info: ### start calculating changes
info: will create Pdlmconn\Model\Data\ServiceConfig with ID SCONF001
info: will update Pdlmconn\Model\Data\ServiceModel with ID SMOD001 (Service Model 1)
info: ### end calculating changes
info: ### start applying changes
info: did create ServiceConfig SCONF001
info: did update ServiceModel SMOD001
info: ### end applying changes
gust:pdlm-connector/ $ [3:40:36]
    
```

Abbildung 42 : Zu sehen ist das Konverter-Programm mit Demodatens und dem “all”-Parameter. Deutlich zu erkennen sind die drei Phasen der Operation, jeweils markiert durch drei “#”-Symbole und die Meldung über den Start.

Die Einrichtung und Benutzung der fertigen Anwendung ist auch auf der Bitbucket-Seite des Repositories<sup>135</sup> und in der README.rst im Hauptverzeichnis des Quellcodes dokumentiert.

Sollte der Konverter verwendet werden und eine andere oder modifizierte Repräsentation von Dienstleistungen unterliegen, so muss der Quellcode teilweise refaktoriert werden. Die Klassen “Sm”, “Plm”, “AbstractReader” und “AbstractWriter” enthalten Strukturen, die auf dem Modell von Klingner et. al basieren. Diese müssen ausgetauscht werden. Im Aras Innovator müssen andere ItemTypes und Relations angelegt werden. Die Konfiguration “config/aras.ini” für den Aras\Reader und Aras\Writer müssen angepasst werden und es müssen neue Klassen für das Datenbankmapping geschrieben werden. Der Umfang der Änderungen ist jedoch, je nach Änderung der Dienstleistungsabbildung, relativ gering.

<sup>134</sup> nixCraft. HowTo: Add Jobs To cron Under Linux or UNIX? URL: <http://www.cyberciti.biz/faq/how-do-i-add-jobs-to-cron-under-linux-or-unix-oses/> (besucht am 25. 07. 2013).

<sup>135</sup> BitBucket. fgolemo / pdlm-connector. URL: <https://bitbucket.org/fgolemo/pdlm-connector> (besucht am 25. 07. 2013).

Wenn man zum Beispiel neben Dienstleistungskomponenten, -modellen und -konfigurationen auch noch “Dienstleistungsfeedback”, also Kundenmeinungen, modellieren möchte, so belaufen sich die Änderungen auf: eine geänderte Zeile jeweils in Sm, Plm, AbstractReader und AbstractWriter, eine neue PHP-Klasse “ServiceFeedback” für das Datenbankmapping, einen Eintrag in der aras.ini über 5-10 Zeilen und eine neue Klasse im Aras Innovator.

### Schwierigkeiten

Während des Entwicklungsprozesses sind zahlreiche Probleme aufgetreten. Da ein Ziel dieser Arbeit Reproduzierbarkeit ist, wird in diesem Kapitel darauf eingegangen, was in ähnlichen Projekten hinderlich sein könnte.

- Zunächst wurde versucht, die SE-SOAP-Schnittstelle des Aras Innovators zu nutzen. Diese ist komfortabler in der Nutzung und WSDL-dokumentiert. Das heißt, die Funktionen, die an dieser Schnittstelle zur Verfügung stehen, sind formal dokumentiert. Aufgrund dieser Tatsache kann man mit Testprogrammen, wie SOAPui<sup>136</sup>, automatisch Tests der einzelnen Funktionen erstellen lassen und sieht, wie die Schnittstelle funktioniert. Damit ist die Erlernbarkeit der SE-Schnittstelle wesentlich höher als bei H-SOAP. Es ist außerdem sehr einfach zu kontrollieren, welche Datentypen über die Schnittstelle zur Verfügung gestellt werden und welche Zugriffsrechte bestehen. Damit ist auch das Rechtmanagement einfacher. Jedoch hat die Schnittstelle eine entscheidende Schwachstelle: darüber lassen sich zwar einzelne Datensätze holen, jedoch noch nicht deren, in Relation stehende, Elemente. Zum Beispiel hat ein Service Model laut Definition mindestens eine Servicekomponente. Wenn das Model nun abgeholt werden soll, so erhält man zwar alle Daten des Models, jedoch keinerlei Auskunft darüber, ob und wie viele Servicekomponenten verknüpft sind. Man kann sich auch eine Liste der Servicekomponenten ausgeben lassen, erhält aber keinerlei Informationen darüber, ob diese eventuell mit Modellen verknüpft sind. Dieses Fehlen der relationalen Angaben macht die Nutzung dieser Schnittstelle unmöglich. Und nach anfänglichen Erfolgen mit SE-SOAP wurde der Code gänzlich umgeschrieben und stattdessen auf H-SOAP gesetzt. Das steigerte den Entwicklungsaufwand enorm und machte es erforderlich, einen AML-Parser (den Aras\Reader) und -Generator (den Aras\Writer) zu schreiben.
- Bei der erstmaligen Einrichtung des objektrelationalen Mappers “Doctrine 2” kam es zu Schwierigkeiten. Die Installation verlief problemlos. Das Plugin installiert zwei ausführbare Anwendungen unter “\$PROJECTROOT/vendor/bin”, namentlich “doctrine” und “doctrine-module”. Diese werden benötigt, um zum Beispiel die Datenbank zu erstellen, zu löschen oder

---

<sup>136</sup> SmartBear Software. What is soapUI? | About SoapUI. URL: <http://www.soapui.org/About-SoapUI/what-is-soapui.html> (besucht am 26. 07. 2013).

allgemein zu überprüfen, ob die Klassen korrekt gemappt werden können. In der offiziellen Dokumentation ist immer nur von der Nutzung von “vendor/bin/doctrine” die Rede<sup>137</sup>, also wurde das auch hier versucht, z.B. mit dem Befehl “vendor/bin/doctrine orm:validate-schema”. Dieser Befehl, wie auch jeder andere, schlägt jedoch mit einer kryptischen Fehlermeldung fehl. Nach einiger Recherche hat sich herausgestellt, dass die Anwendung “doctrine” in der Zend-Framework-Umgebung nicht genutzt werden kann, dafür jedoch “doctrine-module”. Letztere Anwendung verfügt über den gleichen Funktionsumfang und kann äquivalent genutzt werden.

- Während dieses Projektes wurden mehrere Versuche unternommen, eigene Methoden innerhalb der Innovator-Entwicklungsumgebung zu verfassen. Dies stellte sich als äußerst schwierig heraus, da teilweise nicht im vollen Umfang auf Innovator-Objekte zugegriffen werden konnte und da sich mitunter Klassen anders verhielten als dokumentiert. Daher ist nach wie vor ungeklärt, ob es überhaupt technisch möglich gewesen wäre, die Konverter-Anwendung in diesem Umfang direkt im Aras Innovator zu implementieren.
- Während der objektorientierten Entwicklung in PHP wurden die meisten Klassenattribute als “private” oder “protected” markiert, damit darauf nicht öffentlich zugegriffen werden kann oder weil Doctrine 2 das erfordert. Da diese Werte jedoch auf irgendeine Art gelesen und geschrieben werden müssen, muss man “Getter- und Setter-Methoden” heranziehen. Diese Methoden erlauben es, jeweils ein Attribut zu lesen oder zu schreiben und in der Regel werden zwei Methoden für jedes Attribut erstellt (ein “Getter”, ein “Setter”). Dies führt jedoch zu vielen Codezeilen, die Unübersichtlichkeit in relative einfache Klassen bringen können. PHP bietet eine Alternative an, “Magic Methods”<sup>138</sup>. Damit kann man erreichen, dass man nur noch eine get- und eine set-Methode schreibt und damit alle Attribute bedienen kann. Dies fördert zwar die Lesbarkeit des Codes, birgt aber zahlreiche Probleme, darunter schlechtere Performance, schlechtere automatisch-generierte Dokumentation und keine Autovervollständigung in der Entwicklungsumgebung für magische Getter und Setter [LINK68]. Daher wurde auf Magic-Method-Getter und -Setter verzichtet und ein längerer Code akzeptiert. Die zahlreichen Getter- und Setter-Methoden wurden durch die Netbeans-IDE automatisch erstellt und bedurften keiner weiteren Anpassung.
- Es wurde versucht, mit Hilfe des Programms “ApiGen”<sup>139</sup>, automatisch Dokumentationen für das Softwareprojekt erzeugen zu lassen. Leider kam es dabei stets zu einer Fehlermeldung. Diese konnte darauf zurückgeführt werden, dass Taxy die XAMPP-Installation als solche erkannte und eine Bibliothek (“Taxy”) im falschen Verzeichnis suchte. Dies ist ein allgemeiner

---

<sup>137</sup> Doctrine Project. 25. Tools - Doctrine 2 ORM 2 documentation. URL: <http://docs.doctrine-project.org/en/latest/reference/tools.html> (besucht am 26. 07. 2013).

<sup>138</sup> The PHP Group. Magic Methods. URL: <http://php.net/manual/en/language.oop5.magic.php> (besucht am 26. 07. 2013).

<sup>139</sup> Ondrej Nesporek & Nette Foundation Jaroslav Hanslik. ApiGen | API documentation generator for PHP 5.3+. URL: <http://apigen.org/> (besucht am 26. 07. 2013).



Fehler in ApiGen Version 2.8. Dieser kann behoben werden, indem man die ausführbare apigen-Datei editiert und den Pfad zu Texy korrekt einstellt.

## Ergebnis und Ausblick

---

Nachdem die letzten Kapitel den Verlauf der Arbeit dokumentiert haben, soll dieses Kapitel evaluieren, ob und inwiefern die gestellten Ziele erreicht wurden. Weiterhin wird darauf eingegangen, wo noch Verbesserungsbedarf besteht und in welche Richtung die weitere Entwicklung im Bereich der PLM-SM-Integration verlaufen könnte.

### Ergebnis und Zielerreichung

Die ursprüngliche Aufgabenstellung lautete, eine exemplarische Integration eines SM-Systems in ein PLM zu erzielen. Das Ziel besteht aus zwei Teilen, der Exemplarität und der Integration. Nach dem “Digitalen Wörterbuch der deutschen Sprache” bedeutet “exemplarisch”, dass etwas als Beispiel bzw. als Vorbild dient. Das Wort steht in einem Lernkontext. Das Objekt oder der Vorgang ist exemplarisch, wenn es sich um ein Beispiel für ein unterliegendes Prinzip handelt oder wenn der Vorgang als Muster herangezogen werden kann für ähnliche Prozesse<sup>140</sup>.

Es wurden umfassende Recherchen angestellt, welche PLM-Systeme kostenlos verfügbar sind und einen reichhaltigen Katalog an Kriterien erfüllen, der sie praktisch nützlich macht. Dadurch soll gewährleistet werden, dass auch Außenstehende die Ergebnisse dieses Projektes nutzen können und es Vorbild-Charakter erfüllt. Daneben wurde bei der Softwareentwicklung stets darauf geachtet, dass eine hohe Modularität eingehalten wird. Dadurch sind die Programmteile, die speziell auf die beteiligten Systeme, den Aras Innovator und den Service Modeller zugeschnitten sind, sehr lose an den Rest gebunden. Sie können leicht durch Entwickler ausgetauscht werden, zugunsten anderer PLM- und SM-Systeme oder modifiziert, um abweichenden Anforderungen zu genügen. Durch diese Wiederverwendbarkeit ist ein weiterer Schritt in Richtung des ersten Ziels getan.

Für das Ziel der Integration wurde besagte Anwendung entwickelt. Wenn diese ausgeführt wird und in beiden Systemen relevante Daten vorhanden sind, so geschieht eine Datenkonvertierung und -übertragung aus dem SM in das PLM. Diese gewährleistet, dass die Daten aus beiden Systemen in einem integriert werden und so entsteht ein PDLM bzw. PSLM. Das Konverter-Werkzeug in Verbindung mit den kostenlosen PLM- und SM-Anwendungen ermöglicht es beliebigen kleinen und mittelständigen Unternehmen, ihr eigenes integriertes System aus produkt- und dienstleistungsbezogenen Daten zu schaffen.

---

<sup>140</sup> Berlin-Brandenburgische Akademie der Wissenschaften. DWDS | Suchergebnisse 'exemplarisch'. URL: <http://www.dwds.de/?qu=exemplarisch> (besucht am 26. 07. 2013).

Auf der anderen Seite muss festgestellt werden, dass diese Arbeit nicht nur auf der reinen Softwareentwicklung für den Konverter besteht. Die Schaffung des PDLM-Ökosystems im Aras Innovator hat weit mehr Zeit in Anspruch genommen. Dieses ist zwar exportierbar, jedoch ist das offizielle Export-/Import-Werkzeug von Aras sehr fehleranfällig. Außerdem interferiert die Import-Funktion mit bestehenden Klassen im Innovator. Es kann also nicht gewährleistet werden, dass die hier geschaffene Umgebung unter allen Umständen in ein bestehendes System eingefügt werden kann. Um dies ausgleichen, besteht stets die Möglichkeit, dass die Klassen im Innovator manuell angelegt werden oder dass die Konfigurationsdateien des Konverters angepasst werden.

Auch die Anwendung an sich ist noch nicht vollständig. Es fehlen noch, wie beschrieben, die Behandlung von Löschungen im SM-System und die Unterstützung der Versionierungsfunktion, die von einigen PLM, darunter auch dem Innovator, angeboten wird. Diese Funktionen wurden aber im Code und bei der Softwareplanung bereits bedacht und müssen nur noch an der entsprechenden Stelle eingefügt werden. Der Konverter kann auch ohne diese Funktionen eingesetzt werden.

Wenn bei der Dienstleistungsmodellierung ein Paradigma verfolgt wird, das grob von dem hier vorgestellten (Zinke et al. 2013), abweicht, so muss die Anwendung refaktoriert werden. Der Aufwand ist jedoch gering und die entsprechenden Stellen, die das erforderlich machen könnten, wurden in dieser Arbeit genannt.

Alle eben genannten Einschränkungen sind nicht derartig gravierend, dass die Zielerreichung gefährdet wäre. Also wird das Projekt positiv bewertet in Hinblick auf die Erfüllung der Aufgabenstellung.

### Ausblick

Was die nächsten Entwicklungsschritte dieses Projektes betrifft, gibt es mehrere Möglichkeiten:

- Zunächst kann das Konverter-Programm erweitert werden und die weiteren angedeuteten Szenarien behandeln (“Services wurden gelöscht” und “Services sollen versioniert werden”). Dem Anwender könnte man weiterhin die Möglichkeit geben, mit einer Konfigurationsdatei das Konverterverhalten in Konfliktsituation zu steuern.
- Derzeit ist die Erkennung von geänderten Serviceobjekten zu großen Teilen an den Namen gebunden. In Zukunft könnte man eine Heuristik entwerfen, die Punkte vergibt für verschiedene Elemente der Übereinstimmung. Dazu könnte man zwei Schwellwerte festlegen, einen für Ähnlichkeit, einen für Übereinstimmung.
- Man könnte weiterhin die Anbindung an andere Systeme erweitern. Zum einen sind damit die Open-Source-Systeme gemeint, die in dieser Arbeit neben dem Aras Innovator untersucht

wurden. Zum anderen können auch kommerzielle Systeme in Betracht gezogen werden, um die Anzahl der Einsatzszenarien zu erhöhen.

In jedem Falle wurde gezeigt, dass sich mit zwei kostenlosen, bestehenden Softwaresystemen und einem einfachen Softwarewerkzeug von knapp 2000 Zeilen Code ein komplexes Produkt-Dienstleistungs-Lebenszyklus-Management-System schaffen lässt. Damit ist der Grundstein für weitere praktisch-integrative Arbeiten gelegt.

Außerdem wird erwartet, gestützt z.B. durch (Meier et al. 2012), dass in Zukunft die Kopplung von Produkten und Dienstleistungen wesentlich stärker wird. Man spricht von einer Transformation zu "hybriden Leistungsbündeln" (Meier et al. 2012). In diesem Zuge ist damit zu rechnen, dass früher oder später auch "konventionelle" PLM-Systeme Einflüsse von Dienstleistungen erhalten und damit den Konverter hoffentlich überflüssig machen.

## Quellenverzeichnis

---

BitBucket. fgolemo / pdlm-connector. URL: <https://bitbucket.org/fgolemo/pdlm-connector> (besucht am 25. 07. 2013).

Bustamante, Michele Leroux: Hosting WCF Services. URL: {<http://www.code-magazine.com/articleprint.aspx?quickid=0701041&printmode=true>} (besucht am 20. 07. 2013).

Doctrine Project. 25. Tools - Doctrine 2 ORM 2 documentation. URL: <http://docs.doctrine-project.org/en/latest/reference/tools.html> (besucht am 26. 07. 2013).

Golemo, Florian (2013): Integration eines Service-Portfolio-Managers (SM) in ein Product-Lifecycle-Management-System (PLM). Bachelorarbeit.

Inc. Cake Software Foundation. CakePHP: the rapid development php framework. URL: <http://cakephp.org/> (besucht am 25. 07. 2013).

Inc. Novell. Moonlight. URL: <http://www.go-mono.com/moonlight> (besucht am 20. 07. 2013).

Klingner, Stephan, et al. "Managing complex service portfolios: A business case from a full service provider." *reser. net* (2011): 1-13.

Linux Mint. Linux Mint Website. URL: <http://www.linuxmint.com/index.php> (besucht am 28.07.2013).

McLeod, Gavin (2013): Best PHP Frameworks for Developers | Code Geekz. URL: <http://codegeekz.com/best-php-frameworks-for-developers/> (besucht am 25. 07. 2013).

Meier, Horst, and Eckart Uhlmann, eds. *Integrierte Industrielle Sach- und Dienstleistungen: Vermarktung, Entwicklung und Erbringung hybrider Leistungsbündel*. Springer, 2012.

Microsoft Corporation. Windows Communication Foundation (WCF). URL:<http://msdn.microsoft.com/de-de/vstudio/aa663324.aspx> (besucht am 20. 07. 2013).

nixCraft. HowTo: Add Jobs To cron Under Linux or UNIX? URL: <http://www.cyberciti.biz/faq/how-do-i-add-jobs-to-cron-under-linux-or-unix-oses/> (besucht am 25. 07. 2013).

Oracle. Enum Types. URL: <http://docs.oracle/javase/tutorial/java/javaOO/enum.html> (besucht am 25.07.2013).

Oracle. Welcome to NetBeans. URL: <https://netbeans.org/> (besucht am 25. 07. 2013).

PHPFrameworks.com. Top 10 Ranking PHP Frameworks. URL: <http://www.phpframeworks.com/top-10-php-frameworks/> (besucht am 25. 07. 2013).

Portnow, Denis (2013): Coding Standards - Zend Framework 2.0 - Zend Framework Wiki. URL: <http://framework.zend.com/wiki/display/ZFDEV2/Coding+Standards> (besucht am 24. 07. 2013).

Romanenko, Helen (2013): Top 5 PHP Frameworks Infographic | PHP Zone. URL:<http://php.dzone.com/articles/top-5-php-frameworks> (besucht am 25. 07. 2013).

Sensio Labs. High Performance PHP Framework for Web Development - Symfony. URL: <http://symfony.com/> (besucht am 25. 07. 2013).

Universität Leipzig. Ergebnisse - KoProServ. URL: <http://koproserv.uni-leipzig.de/ergebnisse> (besucht am 17. 07. 2013).

Universität Leipzig. ServiceModeller. URL: <https://europa.informatik.uni-leipzig.de/ServiceModeller> (besucht am 19. 07. 2013).

Yii Software LLC. Yii PHP Framework: Best for Web 2.0 Development. URL: <http://www.yiiframework.com/> (besucht am 25. 07. 2013).

Zend Technologies Ltd. Zend - The PHP Company - Zend.com. URL: <http://www.zend.com/de/> (besucht am 24. 07. 2013).

Zend Technologies Ltd. Introduction to Zend/Console - Zend Framework 2 2.2.2 documentation - Zend Framework. URL: <http://framework.zend.com/manual/2.2/en/modules/zend.console.introduction.html> (besucht am 25. 07. 2013).

Zinke, Christian; Meyer Kyrrill; Golemo, Florian and Lars-Peter Meyer: The Use of a Service Modeler Together with a PLM Software for the Management of Product-Related Services: A First Use-case-based Approach to Configure Service Components for Product-Related Services In: Product Lifecycle Management for Society . Springer . 2013

## Kapitel 6 Kundenintegration und Kollaborationsmöglichkeiten im Aras Innovator (*Christian Zinke, Lars-Peter Meyer*)

### Einleitung

---

Die Einbindung des Kunden ist eines der konstitutiven Momente von Dienstleistungen. Die Unterstützung zur systematischen und digitalen Einbindung von Kunden in Dienstleistungsprozesse ist daher eine der wichtigsten Herausforderungen eines PDLM-Systems. Diese Herausforderungen werden in diesem Kapitel angegangen. Hierbei werden die Möglichkeiten der aktiven und digitalen Kundeneinbindung in den Aras Innovator untersucht, erweitert und letztlich implementiert. Klares Ziel ist es Unternehmen in die Lage zu versetzen, den Kunden in den Dienstleistungsprozess stärker mit einzubeziehen und gleichzeitig den eigenen Aufwand für die Dienstleistungserbringung möglichst zu minimieren. Die digitale Integration des Kunden und deren Integration in das PLM/PDM System schafft eine entsprechende Informationsbasis und Systematik für beide Seiten, um diese Ziele zu erreichen. Im Folgenden werden zunächst die Möglichkeiten der Kundenintegration in den Aras Innovator betrachtet. Anschließend wird auf die Möglichkeit der Integration einer separaten Kundenplattform untersucht und dessen Umsetzung vorgestellt.

### Möglichkeiten der Einbindung des Kunden

---

Im Standardpaket von Aras Innovator hat der Kunde eine eher passive Rolle. So können Kundendaten gespeichert und im System abgelegt werden, ohne jede weitere Verknüpfungen. Direkte Kundensichten oder Ähnliches sind nicht implementiert und müssen erst neu geschaffen werden.

#### Einbindung über den Aras Innovator

Die Einbindung des Kunden kann direkt über das PLM/PDM System geschehen, welcher damit zur direkten und digitalen Kollaborationsplattform wird. Dies kann über verschiedenste Anpassungen geschehen und wird im Folgenden beschrieben.

Um den Kunden einzubinden ist es notwendig, dem Kunden überhaupt Zugriff zum System zu geben und den Table of Content (TOC), die Permissions sowie weitere Filterfunktionen zu implementieren. Dies wäre eine Möglichkeit den Kunden zumindest Daten über den Aras Innovator bereitzustellen. Vorstellbar sind hier Produkt- oder Dokumentationsdaten, welche so zur Verfügung gestellt werden könnten. Dies entspricht jedoch weniger einem aktiven Dienstleistungsmanagement, da es sich um ein reines zur Verfügung stellen von Information handelt. Zudem ist die Einstellung einer solchen Kundensicht im Aras Innovator relativ aufwendig, da das Permissionmanagement im Aras Innovator sehr komplex ist. Zusätzlich zum eher kleinen Nutzen einer solchen Einrichtung steht außerdem der

Aufwand beim Kunden den Zugang zum Aras Innovator über den Internet Explorer einzurichten (Aras Innovator Version 9<sup>141</sup>).

In einer solchen Einrichtung von Kundenzugängen zum PLM/PDM System ist der eingeloggte Kunde jedoch noch unabhängig vom integrierten CRM. Die Aras Modellierung muss, um diesen Zustand zu beheben, in der Weise angepasst werden, dass ein Kunde im CRM einer Kundenidentity zugeordnet wird. Mit Hilfe dieser Zuordnung können Kunden leichter in das System integriert werden und kann als Grundlage eines aktiven Kundenmanagements gesehen werden.

Soll der Kunde darüber hinaus in die Prozesse und Dienstleistung einbezogen werden, ist der Aras Innovator nur sehr schwer konfigurierbar. Im Rahmen des PDLM Projektes wurde eine solche Schnittstelle für das ETM (Ersatzteilmanagement) zu Testzwecken eingerichtet. Hierzu mussten mehrere komplexe Funktionen geschrieben werden, die die Kundensicht so einschränkt, dass dieser immer nur das „richtige“ sieht. Zudem musste ein zusätzliches Formular geschrieben werden, welches kundenspezifische Daten auslesen und zur Verfügung stellen sollte. Die zahlreichen Verknüpfungen auf reiner Darstellungsebene sind extrem aufwendig in das Formular einzupflegen.

Allein der Austausch von Informationen innerhalb des Aras Innovators war aus dieser Perspektive sehr aufwendig. Eine Dienstleistung ist aber nicht nur ein statischer Moment des Informationsaustauschs, sondern zumeist komplexe Prozesse mit vielseitigen Interaktionen. Aus dieser Sicht heraus ist das Ausfüllen eines Formulars eine Aktivität im Dienstleistungsprozess, welcher mit Absenden des Formulars weiterläuft. Die Einbindung der Kundeninformationen in den Workflow hat dabei oberste Priorität. Auch dies ist mit einem nicht unerheblichen Zeitaufwand umgesetzt wurden. Die jeweiligen Anpassungen haben prototypisch gezeigt, dass der Kunde in den Aras Innovator auf Dienstleistungs- und Prozessebene mit einem sehr hohen Aufwand eingebunden werden kann, aber noch mit vielen Restriktionen einhergeht und innerhalb des Aras Innovators kaum standardisiert werden kann.

#### *Einbindung des Kunden über eine externe Schnittstelle - Plattformanbindung*

Da der Aras Innovator in den System-Anforderungen für einen Kunden, und damit auch für den Hersteller, nicht besonders attraktiv ist, stellt sich die Herausforderung eine Möglichkeit zu schaffen, Kunden besser zu integrieren. Dies kann über eine externe Schnittstelle, welche mit dem Aras Innovator verbunden ist, realisiert werden, ohne auf die umfangreichen Funktionalitäten des Aras Innovators verzichten zu müssen.

Wie schon vorgestellt besteht die Möglichkeit, Daten des Aras Innovators via AML Abfragen abzurufen, aber auch Daten an den Aras Innovator zu senden (siehe Kapitel 5 Unterstützung der produktbezogenen

---

<sup>141</sup> Dies trifft auf die aktuelle Version des Aras Innovators 10 nur noch bedingt zu (Stand 19.06.2014).

Dienstleistung durch die Integration des Service Modeller). Diese Möglichkeiten können genutzt werden, um eine browserunabhängige Plattform zu schaffen und dabei bestimmte Funktionalitäten, welche für den Kunden entscheidend sind, extern abzubilden. Dies schafft darüber hinaus ein auf den Kunden abgestimmtes User Interface und sich an dessen Bedürfnissen zu orientieren, dabei trotzdem mit dem eigentlichen PLM/PDM System verbunden zu sein und einen gut dokumentierten und systematischen Ablauf der Dienstleistungen zu garantieren. Abstimmungsprozesse können digital vollzogen werden, und sind damit lückenlos dokumentiert. Im Rahmen des Projektes wurde aus den genannten Gründen ein Prototyp mit der Zielsetzung der direkten und individuellen Integration von Kunden entwickelt. Dabei wurde eine Schnittstelle zum Aras Innovator geschaffen, welche eine systematische Erweiterung der Schnittstelle darstellt, welche in Kapitel 5 Unterstützung der produktbezogenen Dienstleistung durch die Integration des Service Modeller vorgestellt wurde. Der entwickelte Prototyp kann auf den entsprechenden Dienstleistungsprozess innerhalb des Aras Innovators zugreifen und der Kunde bekommt eine benutzerfreundliche Oberfläche, um beispielsweise eine Anfrage abzuschicken und den Dienstleistungsprozess aus seiner Sicht zu beobachten. Weiterhin ist dieser Prototyp prinzipiell in der Lage, alle Dienstleistungen mit Kundenintegration zu unterstützen. Die Funktionalitäten werden im Use-Case ETM (Kapitel 8 Unterstützung von After-Sales Dienstleistungen am Beispiel des Ersatzteilmanagement) detailliert vorgestellt. Umgesetzt wurden:

- Browserunabhängiges Portal
- Integration des User-Systems des Aras Innovator zur Zuordnung der Kundenrolle
- Integration des Aras Prozess Models – Anstoßen des entsprechenden Workflows der Dienstleistung
- Integration der dienstleistungserforderlichen, kundenspezifischen Produktdaten
- Nutzfrendliches und intuitives UI

Die vorgestellte Plattform ist speziell auf die Kundensicht angepasst, darüber hinaus gibt es natürlich noch andere Akteure (wie zum Beispiel Lieferanten) und Informationen, die für den Dienstleistungsprozess eine erhebliche Rolle spielen. Dies soll der nachfolgende Fall verdeutlichen.

#### Dienstleistungskomponenteninformationen – Vernetzung von unterschiedlichen Informationen

In einem Dienstleistungsprozess werden zumeist unterschiedlichste Informationen aus vielen verschiedenen Quellen benötigt bzw. werden in Kooperation miteinander erstellt. Diese Informationen, Anforderungen u.ä. müssen nachhaltig dokumentiert werden, damit sie als Ressource für andere Dienstleistungen oder Dienstleistungskomponenten zur Verfügung stehen. So müssen die Aktivitäten und Informationen, die der Kunde im Dienstleistungsprozess durchführt, gespeichert und dokumentiert werden. So ist es durch die Integration des Kundenportals zum Aras Innovator möglich, Informationen

und Inputs vom Kunden automatisiert mit dem Workflow zu verknüpfen und damit nachvollziehbar zu halten. Im Beispiel des ETMs wurde darüber hinaus die Verbindung der Dienstleistung, der Kundeneingabe, der Ersatzteile und der externen ERP Datenbank über geschaffen. Diese Schritte sind für jede einzelne Dienstleistung dokumentiert.

Wie später noch detaillierter beschrieben werden soll (Kapitel 8 Unterstützung von After-Sales Dienstleistungen am Beispiel des Ersatzteilmanagement), ist das ETM so geprägt, dass der Kunde einzelne Produktkomponenten auswählen und neu bestellen kann. Kunden und Projektdaten werden dabei vom System ausgelesen und die richtigen Informationen werden an den Kunden herausgegeben. Über eine einfache Bestellfunktion können die Daten abgeschickt werden. Daraufhin werden diese Daten automatisch mit einem externen ERP-System verglichen und auf Preis und Aktualität des Preises hin geprüft. Die Automatisierung solcher Vorgänge stellt einen erheblichen Mehrwert für das Unternehmen dar. Zusätzlich zur Automatisierung wird eine Kundennähe und einheitliche Dokumentationsbasis der Kundeninteraktion geschaffen. Im Managementbereich können diese Prozesse nachvollzogen werden und darüber hinaus können durch entsprechende Reports die Zeiten der einzelnen Dienstleistungen und Dienstleistungsaktivitäten eingesehen werden. Jede weitere Automatisierung kann also durch das Management direkt durch einen Zeitfaktor bewertet werden.

Wichtig an dieser Stelle bleibt die Anbindung von Produktdaten (als obligatorische Ressource) an die Aktivitäten des Kunden und die Möglichkeiten der direkten Weiterverarbeitung von Kundendaten zu den Produktdaten. Eine Bestellung von Neuteilen wäre nur ein Beispiel. Dasselbe Prinzip kann auch auf ein Change Management, oder ein Re-sampling von Komponenten angewendet werden. Der Kunde würde in dem Falle die Funktionsänderungswünsche für einzelne Komponenten angeben und abschicken. Die Anbindung erlaubt eine direkte Kommunikation über dieselbe Datengrundlage, was Abstimmungsprozesse erheblich verkürzen kann.

Das eben beschriebene und noch sehr einfach gehaltene Beispiel zeigt deutlich die möglichen Vernetzung von Informationen aus unterschiedlichsten Quellen auf und deren Potentiale im Dienstleistungsprozess.

### Kollaborationsmöglichkeiten des Aras Innovator – ein Ausblick

---

Die Beschreibung, wie ein Kunde in einen einfachen oder komplexen Prozess im Unternehmen eingebunden werden kann, ist nur eine Form der Kollaborationsmöglichkeiten, welche der Aras Innovator bietet. Der Aras Innovator bietet viele Möglichkeiten der Zusammenarbeit. Hierbei ist es unerheblich, ob es sich um einen Kunden, Lieferanten oder Mitarbeiter handelt. Die Einbindungsmöglichkeiten sind dank des komplexen User- und Permissionsystems dieselben.



Aufwendig ist jedoch die jeweilige Anpassung an den eigentlichen Prozessablauf auf die einzelnen Rollen im System.

Die Aras-Strukturen können so genutzt bzw. umgestaltet werden, so dass eine Zusammenarbeit einer Vielzahl von Beteiligten über mehrere Prozessschritte oder innerhalb eines Prozessschrittes möglich ist und abgebildet werden kann. Die Zuordnung mehrerer personeller Ressourcen auf eine Aktivität macht eine Abstimmung zwischen den Mitarbeitern nötig. Weiterhin können Ressourcen, wie Dokumente, digital weitergereicht und bearbeitet werden. Voraussetzung für eine solche umfangreiche Kollaboration ist nicht nur eine Abbildung der internen Strukturen, sondern auch die Abbildung der externen Strukturen und eine entsprechende Verknüpfung zu den Funktionalitäten des Aras Innovator. Dies kann an dieser Stelle nur als Ausblick beschrieben werden, da der Fokus dieses Projektes auf der Integration des Kunden lag und nicht in der digitalen Vernetzung aller an allen Prozessen beteiligter Partner. Es wurde jedoch exemplarisch aufgezeigt, wie eine solche Vernetzung aussehen könnte und welche Vorteile diese hat.

## Abschnitt 3      Anwenderfallstudien

---

*“Ideas won't keep. Something must be done with them.”*

*Alfred North Whitehead (1861-1947), engl. Mathematiker u. Philosoph*

## Kapitel 7 Unterstützung von Pre-Sales Dienstleistungen am Beispiel der Erarbeitung eines technischen Lösungsvorschlag (Frieder Swoboda, Egbert Mauersberger, Christian Zinke)

In vielen praktischen Fällen muss vor der Produkterstellung eine genaue Abstimmung über die Lösungsvorstellung des Kunden vorgenommen werden. Diese Leistung wird erbracht, um zielgenau die Kundenwünsche erfüllen zu können und ist damit Basis für weitere Leistungen bis hin zum Pflichtenheft, dem Produktdesign und der Erstellung. In dem Projekt PDLM wurde diese Dienstleistung analysiert und ein Prozessmodell mit entsprechender Ressourcenanbindung erstellt. Über die Ressourcenanbindung hinaus wurden Schnittstellen mit dem prototypischen PDLM System herausgearbeitet. Diese dienen hier als Vorlage.

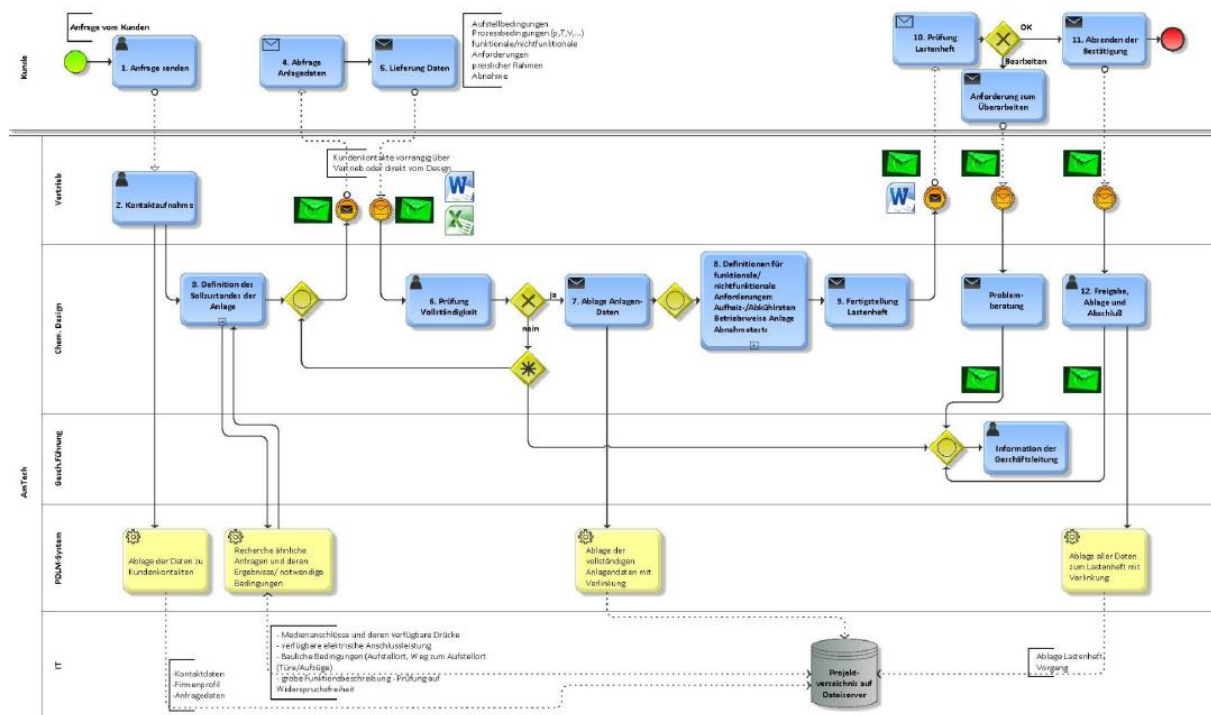


Abbildung 43 Modellierter Sollprozess BPMN

Der abgebildete Sollprozess ist der erste Schritt zur systematischen Planung einer Dienstleistung. Dieser Prozess musste, wie auch schon im Fall des ETM, entsprechend als Workflow in das System eingebunden werden. Diese Einbindung beinhaltete über den eigentlichen Prozessablauf hinaus das Einbinden von Rollen und verschiedenen Dokumenten, die während des Prozesses von den jeweiligen Bearbeitern benötigt werden.

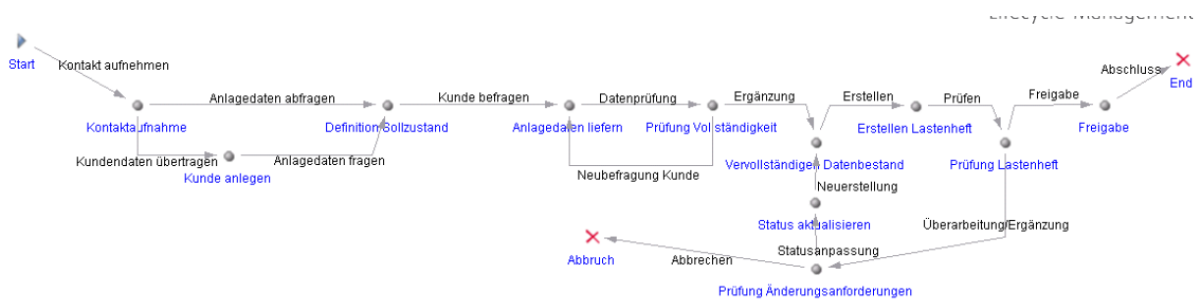


Abbildung 44: Aras Workflow

Auch müssen während der Erstellung des Workflows alle anderen Ressourcen beachtet werden. In diesem Beispiel wurde prototypisch aufgezeigt, wie standardisierte Dokumente systematisch an den Dienstleistungsprozess gebunden werden können, und eine ganzheitliche Dokumentation des Dienstleistungsprozess gewährleistet werden kann. Die nachfolgende Grafik stellt die Verbindung von Sollprozess und Workflow noch einmal systematisch dar.

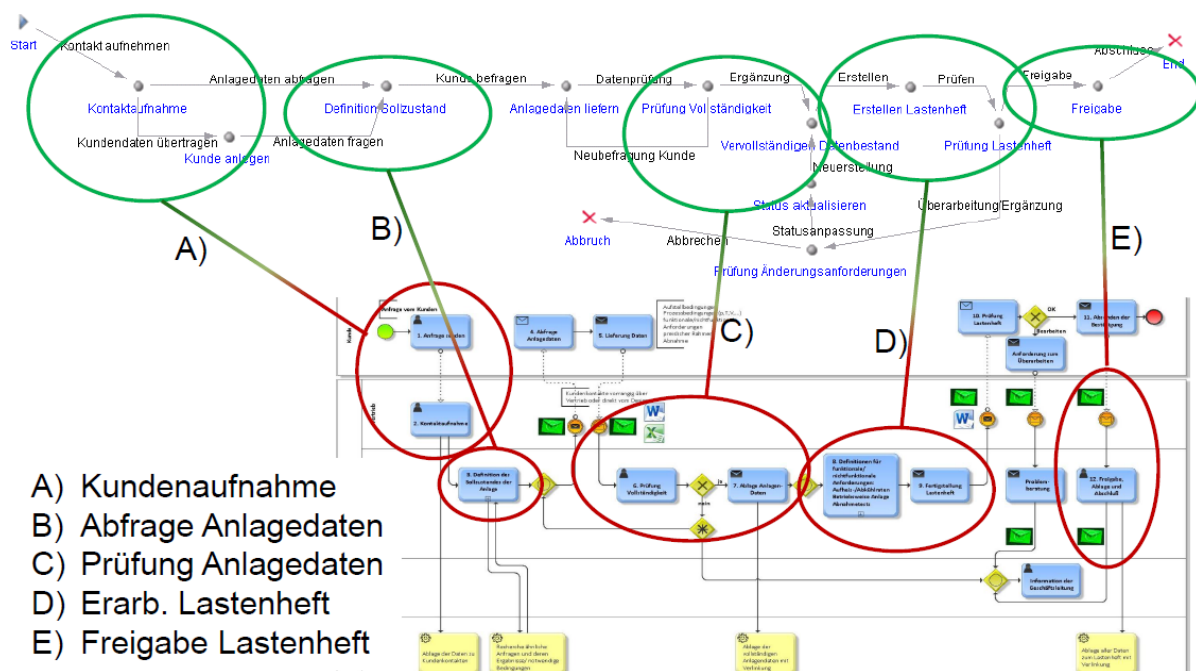


Abbildung 45: Verbindung zwischen Aras Workflow und BPMN Modell

**Beteiligte Mitarbeiter - User im Workflow:**

- Gerd Blau Vertriebsmitarbeiter von Amtech (Vertrieb)
- Fred Meier Mitarbeiter im Chemical Design von Amtech (Chemiker)
- Paul Müller Mitarbeiter des Kunden Amtech1 (Kunde Amtech1)

Nr.	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
1	Anfrage zu Mehrphasenparallelreaktor zur Untersuchung von Hydrierungs-Reaktionen per Email an Amtech	Paul Müller (Kunde Amtech1)	Email Email als Dok <i>Anfrage-Kunde1.msg</i>	
2	Prüfung Kundenkontaktdaten und Abgleich dieser mit dem Datenbestand in Outlook bzw. Neuanlage Kontaktdaten (Firma, Name, Position, Telefon, Email, ...)  Firma "KundeAmtech1" ist im Outlook vorhanden, Ergänzung neuer Kontaktperson "Paul Müller" (Name, Position,...)	Gerd Blau (Vertrieb)	Kontaktdaten aus Kunden-Email	
2	Anlegen neue Workflow-Instanz im Aras Innovator zur Lastenhefterstellung für Kunde "KundeAmtech1"	Gerd Blau (Vertrieb)	Workflow-Instanz mit Kundendaten  Kunden-Email <i>Anfrage-Kunde1.msg</i> in Aras hochladen, mit Workflow verknüpfen	
2	Vertrieb leitet Anfrage an Chemical Design weiter zur Bearbeitung	Gerd Blau (Vertrieb)	Anfragedaten aus weitergeleiteter Kunden-Email	Parallelreaktor Mehrphasen Hydrierung
3	Recherche im Aras Innovator nach Dokumenten oder Dateien zu Lastenheften für Mehrphasenparallelreaktor (z.B. SPR16) zur Untersuchung von Hydrierungs-Reaktionen und damit verknüpfte Workflows zur Lastenhefterstellung	Fred Meier (Chemiker)		Mehrphasen, SPR16, Parallelreaktor, Hydrierung

Nr.	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	<ul style="list-style-type: none"> <li>Suche nach Workflows zur Lastenhefterstellung mit geeigneten Stichworten und deren Relationen zu Dateien bzw. Dokumenten (Property Suchkriterien (search_keys) im ItemType Lastenheft_AMTECH)</li> </ul>			Keine Resultate, da DB leer
3	<p>CUSTOMER_REQUEST-Vorlage-Datei aus dem Aras Innovator herunterladen</p> <ul style="list-style-type: none"> <li>manueller Aufruf einer Item-basierten Action zum automatischen Download der Datei in lokales Dateisystem (Working Directory) mit Öffnen der Excel-Datei</li> <li>CUSTOMER_REQUEST-Vorlage-Datei liegt als Dokument vor</li> </ul>	Fred Meier (Chemiker)	CUSTOMER_REQUEST.xls - Download aktuelle Version des Dokuments	
3	<p>Zusammenstellen einer Abfrage zu den Anlagendaten aus den Dokumenten/Dateien der Recherche-Ergebnisse und der CUSTOMER_REQUEST-Datei</p> <ul style="list-style-type: none"> <li>Eintrag von vorliegenden Daten in die CUSTOMER_REQUEST-Datei</li> </ul>	Fred Meier (Chemiker)	CUSTOMER_REQUEST.xls ausfüllen als Datenblatt.xls Recherche-Daten ex. nicht Spec_SPR16.pdf als Bsp. hinzufügen	

Nr.	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	<ul style="list-style-type: none"> <li>Übermittlung ergänzender Notizen mit Bemerkungen und Informationen zum Anlagentyp</li> </ul>			
3	Verknüpfung der Dateien zur Abfrage im Aras Innovator mit der Workflow-Instanz <ul style="list-style-type: none"> <li>Dateien hochladen und Relationen mit Workflow erstellen</li> </ul>	Fred Meier (Chemiker)	ausgefüllte Excel-Datei ( <a href="#">Datenblatt.xls</a> ), beispielhafte Beschreibung <a href="#">Spec_SPR16.pdf</a> hochladen/ verknüpfen	
3	Übermittlung der Abfrage zu den Anlagendaten an den Vertrieb zur Weiterleitung an den Kunden "KundeAmtech1"	Fred Meier (Chemiker)	<a href="#">Datenblatt.xls</a> <a href="#">Spec_SPR16.pdf</a> Email <a href="#">Abfrage-Anlagendaten.msg</a> mit Anlagen	
4 5	Vom Kunde ausgefülltes Datenblatt und evtl. zusätzliche Dokumente zur Beschreibung der Anforderungen an den Mehrphasenparallelreaktor an den Vertrieb Weiterleitung an den Chemiker	Paul Müller (Kunde Amtech1)  Gerd Blau (Vertrieb)	<ul style="list-style-type: none"> <li>ausgefülltes Datenblatt (<a href="#">Anforderungen.xls</a>)</li> <li>erläuternde Word-Datei (<a href="#">Anforderungen.docx</a>)</li> <li><a href="#">Kunden-Email</a> (<a href="#">Anforderungen-Kunde1.msg</a>)</li> </ul>	

Nr.	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
6 7	<ul style="list-style-type: none"> <li>• Prüfung der Kunden-Anforderungs-Daten auf Vollständigkeit</li> <li>• Anlegen der Datei(en) im Aras Innovator mit Verknüpfung zum aktuellen Workflow                             <ul style="list-style-type: none"> <li>○ Dateien hochladen und Relationen mit Workflow erstellen</li> </ul> </li> </ul>	Fred Meier (Chemiker)	ausgefüllte Datenblatt ( <i>Anforderungen.xls</i> ) als neue Version zu Datenblatt.xls erläuternde Word-Datei ( <i>Anforderungen.docx</i> )	
6	<p>Bei Unvollständigkeit der benötigten Daten Spezifikation noch zu liefernder Informationen und Fortsetzung Workflow in Schritt (3)</p> <ul style="list-style-type: none"> <li>• Dokumentation der Unvollständigkeit der Daten, in Aras Innovator als Protokoll hochgeladen und mit dem Workflow verknüpft</li> </ul>	Fred Meier (Chemiker)	Dokumentationsprotokoll der Unvollständigkeit <i>Luecken.docx</i> hochladen/ verknüpfen	
8 9	Erarbeiten der vollständigen Spezifikation der Anlage für den Kunden "KundeAmtech1" und Erstellen des Lastenhefts (Dokument)	Fred Meier (Chemiker)	<i>Lastenheft.docx</i>	
9	Hochladen Lastenheft-Dokument in Aras Innovator und versionsabhängige Ablage mit Verknüpfung zum aktuellen Workflow	Fred Meier (Chemiker)	<i>Lastenheft.docx</i>	Suchbegriffe aus LH-Inhalt:



Nr.	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	<ul style="list-style-type: none"> <li>Wird über eine Methode (Script im Client) realisiert, die das LH-Dokument im Workflow sucht oder ein Neues erzeugt und mit dem Workflow verknüpft, eine neue Version anlegt und die Datei hoch lädt</li> </ul>			Rührkessel-reaktor Rührerdrehzahl Gasdosierung Massendurchflussregler ...
9	Übermittlung Lastenheft-Dokument an den Vertrieb zur Weiterleitung an Kunden "KundeAmtech1" zur Überprüfung und Bestätigung	Fred Meier (Chemiker)  Gerd Blau (Vertrieb)  Paul Müller (Kunde Amtech1)	<a href="#">Email mit Anlage Lastenheft.docx</a> <i>Uebertragung_Lastenheft.msg</i>	
12	Freigabe der letzten Version des Lastenheft-Dokuments im Aras Innovator bei Bestätigung des Lastenheftes durch den Kunden "KundeAmtech1"	Fred Meier (Chemiker)	<a href="#">Bestätigung Lastenheft</a> <i>Lastenheft-Kunde1.msg</i>	

## Kapitel 8 Unterstützung von After-Sales Dienstleistungen am Beispiel des Ersatzteilmanagement (*Frieder Swoboda, Egbert Mauersberger, Christian Zinke*)

Das Liefern und Bereitstellen von Ersatzteilen ist ein Standardprozess für einen Spezialmaschinenbauer. Innerhalb des Projektes wurde diese Dienstleistung analysiert und ein Prozessmodell mit entsprechender Ressourcenanbindung erstellt. Über die Ressourcenanbindung hinaus wurden Schnittstellen mit dem prototypischen PDLM System herausgearbeitet. Diese dienen hier als Vorlage.

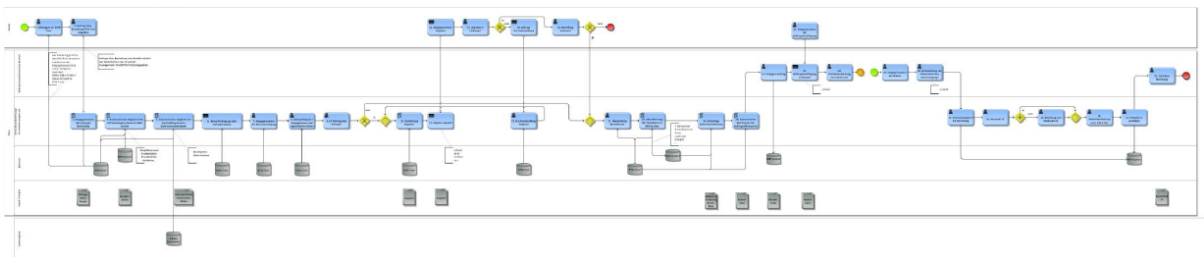


Abbildung 46: Sollprozess in BPMN modelliert

Der abgebildete Sollprozess ist der erste Schritt zur systematischen Planung einer Dienstleistung. Dieser Prozess musste entsprechend als Workflow in das System eingebunden werden. Diese Einbindung beinhaltete über den eigentlichen Prozessablauf hinaus das Einbinden von Rollen und den dazugehörigen personellen Ressourcen, was ein Ressourcenmanagement darstellt.

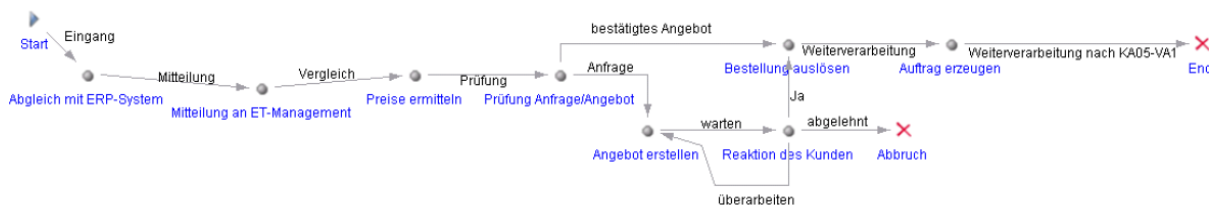


Abbildung 47: Aras Workflow für ETM

Auch müssen während der Erstellung des Workflows alle anderen Ressourcen beachtet werden. In diesem Beispiel wurde prototypisch aufgezeigt, wie Daten von externen Systemen systematisch an den Dienstleistungsprozess gebunden werden können. Die nachfolgende Grafik stellt die Verbindung von Sollprozess und Workflow noch einmal systematisch dar.

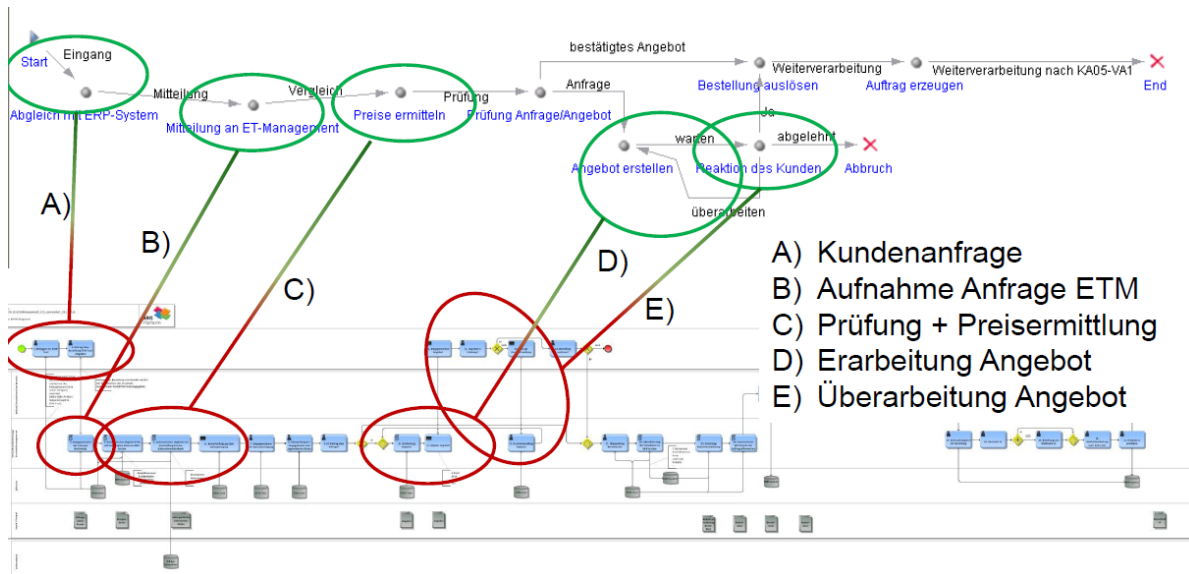
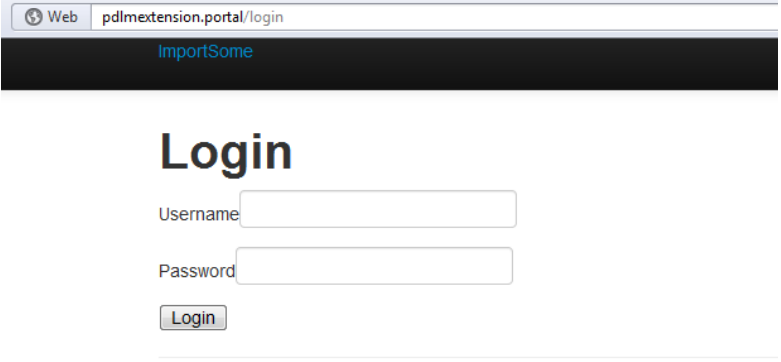


Abbildung 48: Verbindung zwischen Aras Workflow und BPMN Modell

Zur Verdeutlichung des Use-Cases werden im Folgenden systematisch die Abläufe und der Dokumentenaustausch beschrieben.

Dabei beteiligte Mitarbeiter bzw. User im Workflow sind:

- Kurt Muster Mitarbeiter im Ersatzteilmanagement von Sitec (ETManager1)
- Frank Mayer Mitarbeiter des Kunden Sitec1 (Kunde Sitec1)

Nr	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
1	Einloggen von Kunde Sitec1 in Seite ETM von Aras Innovator bei Sitec (zur Zeit direktes Einloggen in Aras als User)	Frank Mayer (Kunde Sitec1) als User im Aras	Login-Formular in Aras für ETM  (Filterfunktion für TOC für Kunde)	
 <p data-bbox="165 898 555 927"><b>Abbildung 49: Login Kundeportal</b></p>				
2	Eingabe der Anfrage-/Bestelldaten für Ersatz-/ Verschleißteil durch Start einer neuen Workflow-Instanz mit Produktdaten von Kunde Sitec1: <ul style="list-style-type: none"> <li>• Projektnummer: ANLX3_Mari Aue</li> <li>• Anlagenbezeichnung und -nummer: ANLX3</li> </ul>	Frank Mayer (Kunde Sitec1)	Filterfunktion für Produktdaten Kunde Sitec1  Bestellung: 2* Aufnahme 1* Mini-Schlitten	

Nr .	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	<ul style="list-style-type: none"> <li>• Stücklistennummer aus Blatt M13: 886-212:00</li> <li>• Benennungen: <ul style="list-style-type: none"> <li>Aufnahme (886-212:04)</li> <li>Mini-Schlitten (DGSL-8-80-P1A)</li> <li>Näherungsschalter (SMT-10F-PS-24V-K0,3L-M8D)</li> <li>Lichttaster (Faseroptik) (BFO D13-XB-RB-EAK-10-02)</li> <li>Basisgerät (BOS 74K-UU-1FR-B0-Z-S49-0,2)</li> </ul> </li> </ul>			

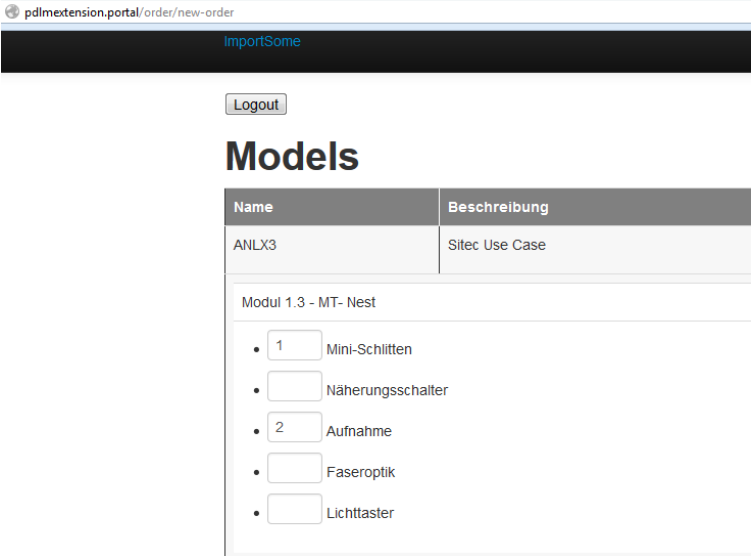
Nr .	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten				
	 <p>The screenshot shows a web interface for 'Models'. It includes a 'Logout' button, a table with the following data:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ANLX3</td> <td>Sitec Use Case</td> </tr> </tbody> </table> <p>Below the table, there is a section titled 'Modul 1.3 - MT- Nest' containing a list of items:</p> <ul style="list-style-type: none"> <li>1 Mini-Schlitten</li> <li>Näherungsschalter</li> <li>2 Aufnahme</li> <li>Faseroptik</li> <li>Lichttaster</li> </ul>	Name	Beschreibung	ANLX3	Sitec Use Case			
Name	Beschreibung							
ANLX3	Sitec Use Case							
3	Übernahme der Bestell- und Login-Daten in den Workflow automatisch nach Abschicken der Anfrage	automatisch	Kunde "KundeSitec1" Anfragedatum "30.07.13"	Kundenname Anlagen- bezeichnung ...				

Abbildung 50: Externes Bestellformular

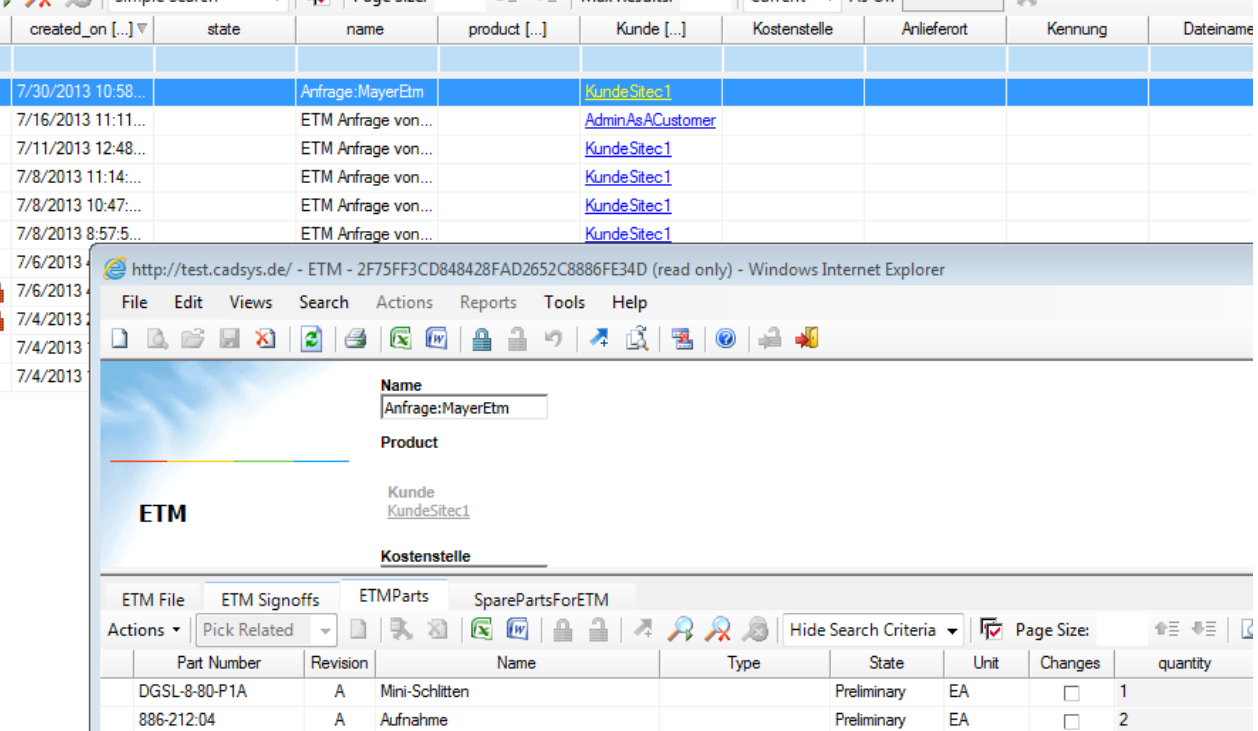
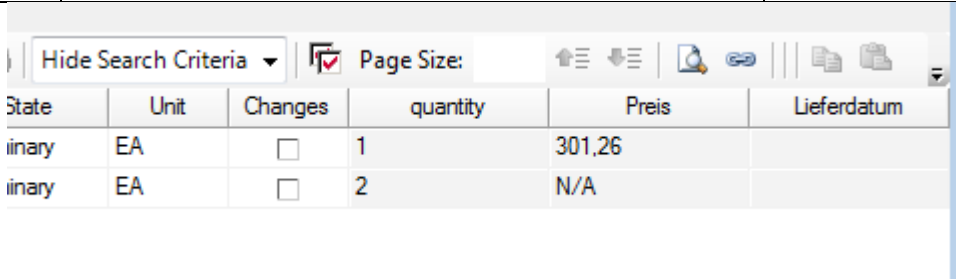
Nr	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	 <p>The screenshot shows a table of ETM requests with columns: created_on, state, name, product, Kunde, Kostenstelle, Anlieferort, Kennung, and Dateiname. Below the table is a detailed view of an ETM request with fields for Name (Anfrage:MayerEtm), Product (Kunde, KundeSitec1), and Kostenstelle. At the bottom, there is a table of parts with columns: Part Number, Revision, Name, Type, State, Unit, Changes, and quantity.</p>			
4	<p>Datenabgleich mit ERP Zugriff über Federated Item auf externe ERP-DB ProAlphaPDLM mit Suche nach:</p> <ul style="list-style-type: none"> <li>• Projektnummer</li> </ul>	automatisch	externe SQL-DB	

Abbildung 51: Anfrage in Aras Innovator angekommen

Nr	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	<ul style="list-style-type: none"> <li>Ersatzteilliste</li> <li>Stückliste</li> </ul>			
				
<p>Abbildung 52: Automatische Abfrage der Preise und letzten Lieferungen</p>				
6	Benachrichtigung über Anfrage-/Bestell-Eingang	automatisch	Email	
7	Entgegennahme Benachrichtigung durch Lesen der Email	Kurt Muster (ETManager1) als Sitec-Mitarbeiter	Email zur Benachrichtigung	
5	Datenabgleich mit Lieferanten → manueller Abgleich durch ET-Manager nach Benachrichtigung (Schritt (6))	Kurt Muster (ETManager1)	<ul style="list-style-type: none"> <li>Einzelpreise Ersatzteile</li> <li>Liefertermine</li> </ul>	Ersatzteil- nummern
8	Manuelle Prüfung der Daten Anfrage/ Bestellung;	Kurt Muster (ETManager1)	<ul style="list-style-type: none"> <li>Kundendaten</li> <li>Projektnummer</li> <li>Ersatzteilliste</li> </ul>	Kundendaten



Nr.	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	ET-Management überprüft, ob die eingegebenen, ermittelten und zugeordneten Daten korrekt und plausibel sind		<ul style="list-style-type: none"> <li>• Stückliste</li> <li>• Einzelpreise Ersatzteile</li> <li>• Liefertermine</li> <li>• ...</li> </ul>	

**Workflow Process**

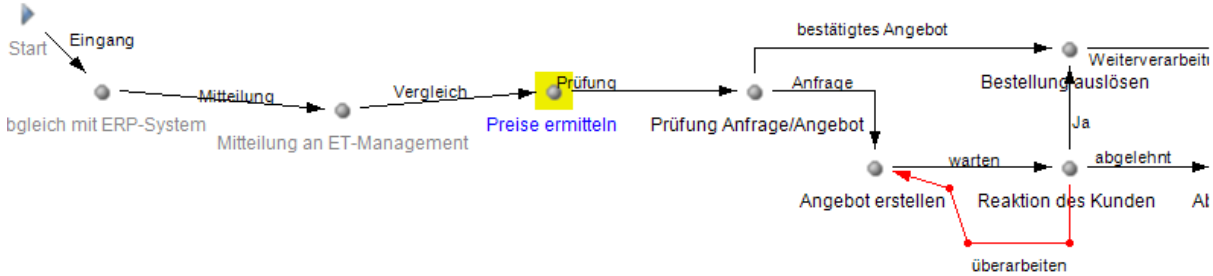


Abbildung 53: Workflow ETM

9	<p>Prüfung auf Vorlage Anfrage oder Bestellung</p> <ul style="list-style-type: none"> <li>• für Anfrage nur Einkaufspreis pro Stück aufgelistet</li> <li>• für Bestellung zusätzlich Gesamt-, Verkaufspreis und Zuschlag</li> </ul>	Kurt Muster (ETManager1)	<ul style="list-style-type: none"> <li>• Einkaufspreise</li> <li>• Gesamtpreise</li> <li>• Verkaufspreise</li> <li>• Zuschlag</li> </ul>	
---	---	-----------------------------	--	--

Nr .	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	es wird eine Bestellung angenommen → Fortsetzung mit Nr. 17 bzw. "Bestellung auslösen" im Workflow			
10	Erarbeitung Angebot zur Anfrage anhand Stücklisten- und Ersatzteillistendaten	Kurt Muster (ETManager1)	<a href="#">Angebotsdokument</a> <i>Angebot</i> <i>KundeSitec1.docx</i> hochladen und verknüpfen mit Workflow	
11	Abgabe Angebot an Kunde	Kurt Muster (ETManager1)	<a href="#">Email</a> <i>Angebot_ETM.msg</i> zum <i>Angebot</i> mit <a href="#">Angebotsdokument</a> <i>Angebot</i> <i>KundeSitec1.docx</i>	Angebots- nummer

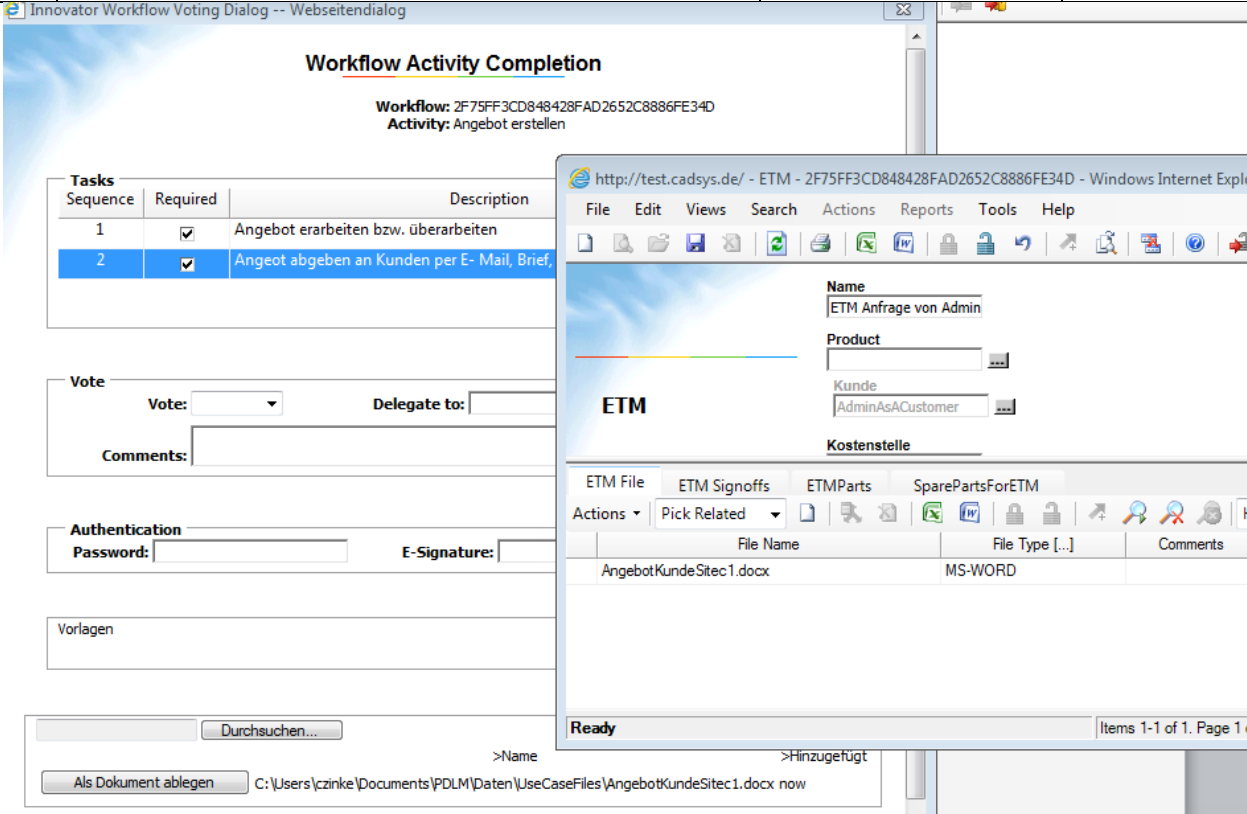
Nr	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
				
12	Prüfung des Angebots durch Kunde,	Frank Mayer (Kunde	• Angebotsdokument	
13	bei Einverständnis (16) Bestellung auslösen	Sitec1)	<i>Angebot KundeSitec1.docx</i>	

Abbildung 54: Dokument der Dienstleistung zugeordnet und an Kunden versendet

Nr .	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
	sonst (14) Anfrage zu Nachverhandlung an Sitec stellen oder Abbruch des Workflows <ul style="list-style-type: none"> <li>• bei Abbruch Notizen zu Abbruchbedingungen an Workflow anhängen</li> </ul>	Kurt Muster (ETManager1)	<ul style="list-style-type: none"> <li>• Email mit Bestellung <i>Bestellung_ETM.msg</i></li> <li>• Email mit Anfrage Nachverhandlung <i>Nachfrage_zu_Angebot_ETM.msg</i></li> <li>• Notiz zu Abbruchbedingungen hochladen <i>AbbruchWorkflow.txt</i></li> </ul>	
15	Nachverhandlung Angebot und Fortsetzung mit Schritt (12)	Kurt Muster (ETManager1)	<a href="#">Notiz zu Nachverhandlung</a> <i>Nachverhandlung.txt</i> hochladen und verknüpfen mit Workflow	
17	Bestellung vom Kunden entgegennehmen Abgleich Bestelldaten mit Angebotsdaten	Kurt Muster (ETManager1)	<a href="#">Email</a> <i>Bestellung_ETM.msg</i> zur Bestellung mit <a href="#">Bestelldaten</a> hochladen und verknüpfen mit Workflow	
18	Aktualisieren der Teiledaten im ERP-System (externe DB - Federated Item zum Schreiben zulassen)	Kurt Muster (ETManager1)	<ul style="list-style-type: none"> <li>• Teilenummern</li> <li>• Bestellnummer</li> <li>• Preise</li> <li>• Lieferzeit</li> <li>• Rabatte</li> </ul>	

Nr .	Aktivität	Bearbeiter	Dokumente Objekte	Suchbegriffe Anfragedaten
19	Erstellen der Bedarfsanforderung im ERP-System durch ET-Management  (Daten in externer DB ProAlphaPDLM zur ERP-Simulation erstellen - siehe Spaltennamen in Klammer)	Kurt Muster (ETManager1)	<ul style="list-style-type: none"> <li>• Kostenstelle Datum (<i>ZugTermin</i>)</li> <li>• Ersteller</li> <li>• Anlieferort</li> <li>• Kennungen Kauf-/Fertigteil, Investition, Neuanlage Teil, Auftrag (<i>Teil</i>)</li> <li>• Auftragsnummer (<i>LiefNr</i>)</li> <li>• Gesamtpreis (Warenwert_FW)</li> <li>• Liefertermin (<i>LiefTermin</i>)</li> </ul>	
20	Weitergabe Auftrags-Informationen an Einkauf durch ET-Management (automatisch - wie? über ERP-Verweis, z.B. Aufgabenzuordnung oder Workflow?)	Kurt Muster (ETManager1)	Auftrags-Informationen (Daten Workflow)	

<p><b>21</b></p>	<p>Erzeugen Auftrag durch Einkauf und Weiterbearbeitung nach KA05-VA1 und Abschluss des Workflows Auftragserzeugung</p> <ul style="list-style-type: none"> <li>• Auftragsbestätigung an Kunde per Email senden</li> <li>• Auftragsbestätigung als Dokument an den Workflow anhängen</li> </ul>	<p>Einkauf Sitec</p>	<p><a href="#">Auftragsbestätigung</a>  <a href="#">Email Auftragsbestätigung_ETM.msg</a> mit Auftragsbestätigung senden, hochladen und verknüpfen mit Workflow</p>	
------------------	--	----------------------	---	--

## Kapitel 9 Evaluation des PDLM Konzepts und des Unterstützungssystems (*Christian Zinke*)

Am Ende eines Konzepts und dessen IT-Unterstützungssystems steht die Frage, welchen Nutzen und Mehrwert diese Arbeit hat. Diese einfache Frage hat keine einfachen Antworten. Es muss danach gefragt werden, ob die erstrebte Dienstleistungsunterstützung für produktbezogene Dienstleistungen tatsächlich unterstützt, an welcher Stelle sie unterstützt, wo die Potentiale liegen, aber auch was Barrieren für eine solche Unterstützung – besonders bei KMU – sind.

### Methode

---

Zunächst gilt es zu beschreiben, was genau, mit welchen Methoden erhoben wurde. Um die Projektergebnisse zu evaluieren, wurde ein semi-strukturierter Fragenbogen entworfen. Dieser Fragebogen umfasste Fragen zu Barrieren bei der Einführung von PDLM Konzepten, zu Kommunikations- und Kollaborationsaspekten (intern und extern), zur Effizienz der Dienstleistungsunterstützung, zur Bedienbarkeit, Verständlichkeit, Fehlerrobustheit und Erweiterbarkeit der Lösung, zu Innovationspotentialen und Operationalisierung der Dienstleistungs- und IT-Prozesse, sowie Fragen zu den speziellen Szenarien, welche in den jeweiligen Unternehmen durchgeführt wurden. Die hier befragten Personen gelten für die Evaluation als Experten, welche fachlichen Aufschluss über die gestellten Fragen geben sollen. Daher kann die Befragung als Experteninterview bezeichnet werden.

Diese Fragen wurden in Gruppengesprächen mit den für die Fälle zuständigen Mitarbeitern gestellt und die Antworten dokumentiert. Fokussiert wurden hierbei die inhaltlich-sachlichen Antworten der jeweiligen Befragten, welche im weiteren Vorgehen zusammengefasst und analysiert wurden. Die Analyse erfolgte mit dem Fokus auf der Identifikation von qualitativen und quantitativen Faktoren und deren Ausprägung für die Umsetzung des PDLM Konzeptes und des Unterstützungssystems.

### Ergebnisse

---

#### *Barrieren bei der Einführung*

Alle praktischen Partner (Geschäftsleitung und ausgewählter Mitarbeiter Kreis) sehen die Notwendigkeit des neuen Systems und den entsprechenden Nutzen. Zum Erhebungszeitpunkt konnten keine Barrieren im Bereich der Mitarbeiter Akzeptanz erkannt werden. Zu diesem Zeitpunkt haben entsprechende Tests mit dem Prototypen stattgefunden. Unterstützungs- und Trainingsarbeit erfolgt durch die CADsys Chemnitz. Allgemein wurde ein Schulungsbedarf bei der Einführung von neuen IT unterstützten Abläufen festgestellt. Ein Problem zum Einsatz ist die bisher unverschlüsselte Datenübertragung beim Aras Innovator. Weiterhin wurde die Notwendigkeit der Individualisierbarkeit der IT Lösung in den Vordergrund gerückt, d.h. die Unterstützung bei der Anpassung des IT

Unterstützungssysteme auf die Besonderheiten im Prozessablauf und der Konfiguration der Informationen. Daher kann abstrahiert werden, dass Barrieren bei der Einführung eines Unterstützungssystems folgende Ebene betrifft:

- Schulungs- und Einarbeitungsaufwand (z.T. Hoch)
- Gewährleistung der Sicherheit im Datenverkehr
- Individuelle Anpassungsarbeiten des IT Systems

#### *Kommunikations- und Kollaborationsaspekte*

Das User Interface des Aras Innovators wurde generell als verständlich, wenn auch als sehr komplex beschrieben. Die umgesetzte prototypische Lösung war zum Teil noch Englisch, was bei einigen Mitarbeitern Sprachprobleme mit sich brachte. Bei der Kommunikation unter den beteiligten Mitarbeiter, unterstützt durch das Unterstützungssystem, konnten zum Erhebungszeitpunkt keine (positiven oder negativen) Veränderungen der Effizienz der Kommunikation festgestellt werden. Bei der Kommunikation mit dem Kunden wurde der Einsatz des Aras Innovators und des entwickelten Kundenportals positiv aufgenommen. Jedoch konnte die Akzeptanz seitens der Kunden nicht eingeschätzt werden. Es wurde angemerkt, dass fehlendes Internet beim Kunden bzw. am Arbeitsplatz des Kunden ein Problem sein könnte. Über den produktiven Einsatz hinaus haben einige Befragte angemerkt, dass über die neuen Möglichkeiten der Kundenintegration neue Marketing Optionen entstehen würden. Dennoch sind weitere Tests nötig, um die Akzeptanz der Beteiligten und die Effizienz des Unterstützungssystems einschätzen zu können.

#### *Effizienz der Dienstleistungsunterstützung*

Die Vorteile der Dienstleistungsunterstützung wurden klar erkannt, so wurde auf die Vorteile durch die klaren Rollenverteilungen der Mitarbeiter, der klaren Schnittstellen, der Übersichtlichkeit über erstellte Dokumentationen, der Vorteile beim Suchen und der Kommunikation sowie genereller Organisationsvorteile aufgezählt. Durch diese Vorteile kommt es überwiegend zu einer schnelleren und routinierteren Arbeitsprozess (u.a. durch Template Nutzung) sowie der neuen Möglichkeit der Parallelisierung von Prozessen. Diese Zeitersparnis quantifizieren die Partner zwischen 10 und 25%. Weitere Zeitersparnisse entstehen durch die umfangreichen Suchmöglichkeiten und den entsprechenden Verknüpfungen. Dieser Aspekt konnten jedoch nicht quantifiziert werden.

Darüber hinaus wird die IT-Lösung so eingeschätzt, dass sie prinzipiell in der Lage ist, Fehler im Ablauf zu reduzieren, so könnten u.a. Rückfragen eingespart werden. Weiterhin wurde die Erinnerungsfunktion positiv hervorgehoben. Die Zeitersparnis durch Fehlerreduktion wurde von einem Partner auf 20% beziffert. Besonders positiv wurde die automatische Dokumentation der Dienstleistungshistorie hervorgehoben, welche eine Engstellenanalyse ermöglicht und eine entsprechende Reaktion folgen



kann. Generell sind das Versionieren und das „Einfrieren“ von Zuständen für die Produktentwicklung von Vorteil.

*Bedienbarkeit, Verständlichkeit, Fehlerrobustheit und Erweiterbarkeit der Lösung*

Obwohl der Prototyp nur kurz im Einsatz war, wurden Fragen zur Bedienbarkeit, Verständlichkeit, Fehlerrobustheit und Erweiterbarkeit der Lösung gestellt. Es wurde sehr schnell darauf verwiesen, dass für einige Informationen erst intensiv und lange mit der IT gearbeitet werden muss, um diese beantworten zu können. Trotzdem wurden die entwickelten Lösungen als verständlich und nachvollziehbar bezogen, auch die Erweiterbarkeit wurde als prinzipiell gut bezeichnet. Darüber hinaus konnten keine weiteren Informationen gewonnen werden.

*Innovationspotentiale und Operationalisierung der Dienstleistungs- und IT-Prozesse*

In diesem Frageblock wurde sich damit beschäftigt, ob noch weitere Innovationspotentiale für die umgesetzte Lösung vorhanden sind. So zum Beispiel, an welchen Stellen noch mehr automatisiert werden kann, oder wo es weitere Zeitersparungen geben könnte. Die Befragten haben angegeben, dass es noch weiteres Potential bei der Kundeneinbindung durch aktuelle Statusmitteilungen und Benachrichtigungen gibt. Viele der Potentiale sind jedoch nicht quantifizierbar. Auch schon vorhandene Effekte, wie zum Beispiel die Datenaktualität, konnten von den Befragten nicht quantifiziert werden. In weiteren Forschungsarbeiten sollte zu diesem Zweck stärker auf die Operationalisierung solcher durchaus nützlichen Effekte beim Qualitätsmanagement der Dienstleistung geachtet werden.

*Anmerkungen zu den speziellen Szenarien*

Die Befragten hatten unterschiedliche Unternehmensszenarien, welche durch die entstandene PDLM Lösung unterstützt wurde. Zu diesen Szenarien wurden gesondert noch einmal Fragen gestellt, um den Nutzen innerhalb der Szenarien noch einmal genauer zu betrachten.

Zusätzlich zu den schon gezeigten Ergebnissen wurde an dieser Stelle die Übersichtlichkeit der ablaufenden Dienstleistungsprozesse und der Möglichkeit der Analyse dieser betont, welche das Management von aktuellen Dienstleistungen wesentlich unterstützt. Darüber hinaus trägt die Qualität der Dokumente und der Nachvollziehbarkeit der Änderungen zu einem besseren und schnelleren Verständnis zwischen Kunden und der Firma im Pre-Sales Bereich bei.

Ausblick

---

Wie in diesem Abschnitt aufgezeigt wurde, gibt es sowohl einen qualifizierbaren als auch einen quantifizierbaren Nutzen des PDLM Unterstützungssystem und der ganzheitlichen und lebenszyklusorientierten Sichtweise von Dienst- und Sachleistungen. Der PDLM Ansatz muss weiter ausgebaut, verfeinert, verglichen und evaluiert werden, um noch bessere Ergebnisse erzielen zu können.

Neuere Ansätze aus dem PSS und Servitization Bereich müssen genutzt werden, um noch bessere Unterstützung von produktbegleitenden Dienstleistungen schaffen zu können und die unterstützenden IT-Systeme auf diese besser und zukunftsfähiger auszurichten. So wird die Weiterentwicklung der Industrie nicht nur von technologischen Innovationen abhängen, sondern wird zunehmend von Innovationen im Bereich der produktbegleitenden Dienstleistungen bestimmt sein, welche eine noch bessere Individualisierung der Sachleistungen zulässt und den Kunden aktiv einbindet. Besonders KMU sind hier gefragt, sich strategisch auf Dienstleistungen auszurichten und dieses ganzheitlich einzubinden. Die vorliegende Arbeit hat einen kleinen Schritt in diese Richtung leisten können.

## Leipziger Beiträge zur Informatik

In der Reihe "Leipziger Beiträge zur Informatik" erscheinen Forschungsberichte aus Forschungsvorhaben, Herausgeberbände im Bereich innovativer und sich etablierender Forschungsgebiete, Habilitationsschriften und Dissertationen sowie herausragende Beiträge von Studierenden. Die Publikationsreihe wird im Eigenverlag der Universität Leipzig vom Institut für Angewandte Informatik e.V. herausgegeben.

- Band I FÄHNRICH, K.-P.; HERRE, H. S. (Hrsg.):  
*Content- und Wissensmanagement. Arbeiten aus dem Forschungsvorhaben Pre BIS und Beiträge auf den Leipziger Informatik-Tagen 2003.* Leipziger Beiträge zur Informatik: Band I. Leipzig, 2003.  
ISBN 3-934178-26-X
- Band II FÄHNRICH, K.-P.; MEIREN, T. (Hrsg.):  
*Computer Aided Engineering. Arbeiten aus dem Forschungsvorhaben Serv -Case.* Leipziger Beiträge zur Informatik: Band II. Leipzig, 2004.  
ISBN 3-934178-39-1
- Band III FÄHNRICH, K.-P.; THRÄNERT, M.; WETZEL, P. (Hrsg.):  
*Umsetzung von kooperativen Geschäftsprozessen auf eine internetbasierte IT-Struktur.* Arbeiten aus dem Forschungsvorhaben Integration Engineering. Leipziger Beiträge zur Informatik: Band III. Leipzig, 2005.  
ISBN 3-934178-52-9
- Band IV FÄHNRICH, K.-P.; KÜHNE, S.; SPECK, A.; WAGNER, J. (Hrsg.):  
*Integration betrieblicher Informationssysteme – Problemanalysen und Lösungsansätze des Model- Driven Integration Engineering.* Leipziger Beiträge zur Informatik: Band IV. Leipzig, 2006.  
ISBN: 978-3-934178-66-3
- Band V FÄHNRICH, K.-P.; HÄRTWIG, J.; KIEHNE, D.-O.; WEISBECKER, A. (Hrsg.):  
*Technologien und Werkzeuge für ein rollen- und aufgabenbasiertes Wissensmanagement. Zusammenfassender Bericht aus dem Forschungsprojekt Pre BIS.* Leipziger Beiträge zur Informatik: Band V. Leipzig, 2007.  
ISBN: 978-3-934178-76-2

- Band VI FÄHNRICH, K.-P.; THRÄNERT, M.; WETZEL, P. (Hrsg.):  
*Integration Engineering. Motivation – Begriffe – Methoden – Anwendungsfälle.* Leipziger Beiträge zur Informatik: Band VI. Leipzig, 2007.  
ISBN: 978-3-934178-78-6
- Band VII AUER, S.:  
*Towards Agile Knowledge Engineering: Methodology, Concepts and Applications.* Dissertation an der Fakultät für Mathematik und Informatik der Universität Leipzig. Leipziger Beiträge zur Informatik: Band VII. Leipzig, 2007.  
ISBN: 978-3-934178-73-1
- Band VIII FÄHNRICH, K.-P.; HEYER, G. (Hrsg.):  
*Games Summer Camp 2007. Interdisziplinäres Blockseminar zum Thema Digitale Spiele.* Eine Dokumentation. Leipziger Beiträge zur Informatik: Band VIII. Leipzig, 2007.  
ISBN: 978-3-934178-77-9
- Band IX ASLAM, M. A.:  
*Towards Integration of Business Processes and Semantic Web Services.* Leipziger Beiträge zur Informatik: Band IX. Leipzig, 2008.  
ISBN: 978-3-934178-89-2
- Band X FÄHNRICH, K.-P.; MÜLLER, R.; MEYER, K.; FREITAG, M. (Hrsg.):  
*Entwicklung internationaler produktbezogener Dienstleistungen – Ein Handlungsleitfaden für kleine und mittlere Unternehmen.* Leipziger Beiträge zur Informatik: Band X. Leipzig, 2008.  
ISBN: 978-3-934178-98-4
- Band XI FÄHNRICH, K.-P.; KÜHNE, S.; THRÄNERT, M. (Hrsg.):  
*Model-Driven Integration Engineering. Modellierung, Validierung und Transformation zur Integration betrieblicher Anwendungssysteme.* Leipziger Beiträge zur Informatik: Band XI. Leipzig, 2008.  
ISBN: 978-3-941152-02-1
- Band XII MAICHER, L.; GARSHOL, L. M. (Hrsg.):  
*Subject-centric Computing. Fourth International Conference on Topic Maps Research and Applications, TMRA 2008.* Leipziger Beiträge zur Informatik: Band XII. Leipzig, 2008.  
ISBN: 978-3-941152-05-2

- Band XIII FÄHNRICH, K.-P.; SCHUMACHER, F. (Hrsg.):  
*Digitale Spiele in Forschung und Lehre. Beiträge zum Games Summer Camp 2008.* Leipziger Beiträge zur Informatik: Band XIII. Leipzig, 2009.  
ISBN: 978-3-941608-00-9
- Band XIV HEYER, G. (Ed.):  
*Text Mining Services – Building and applying text mining based service infrastructures in research and industry. Proceedings of the conference on Text Mining Services – TMS 2009 at Leipzig University.* Leipziger Beiträge zur Informatik: Band XIV. Leipzig, 2009.  
ISBN: 978-3-941608-01-6
- Band XV THRÄNERT, M.:  
*Integration-Engineering – Grundlagen, Vorgehen und Fallstudien.* Leipziger Beiträge zur Informatik: Band XV. Leipzig, 2009.  
ISBN: 978-3-941608-02-3
- Band XVI FÄHNRICH, K.-P.; ALT, R.; FRAN CZYK, B. (Eds.):  
*Practitioner Track – International Symposium on Services Science (ISSS'09).* Leipziger Beiträge zur Informatik: Band XVI. Leipzig, 2009.  
ISBN: 978-3-941608-03-08
- Band XVII MEYER, K.:  
*Software – Service – Co-Design: Eine Methodik für die Entwicklung komponentenorientierter IT-basierter Dienstleistungen.* Leipziger Beiträge zur Informatik: Band XVII. Leipzig, 2009.  
ISBN: 978-3-941608-04-7
- Band XVIII AUER, S.; LAUENROTH, K.; LOHMANN, S.; RIECHERT, T. (Hrsg.):  
*Agiles Requirements Engineering für Softwareprojekte mit einer großen Anzahl verteilter Stakeholder.* Leipziger Beiträge zur Informatik: Band XVIII. Leipzig, 2009.  
ISBN: 978-3-941608-05-4
- Band XIX MAICHER, L.; GARSHOL, L. M. (Eds.):  
*Linked Topic Maps. Fifth International Conference on Topic Maps Research and Applications, TMRA 2009.* Leipziger Beiträge zur Informatik: Band XIX. Leipzig, 2009.  
ISBN: 978-3-941608-06-1

- Band XX HÄRTWIG, J.:  
*Konzept, Realisierung und Evaluation des semantischen Informationsraums. Dissertation.*  
Leipziger Beiträge zur Informatik: Band XX. Leipzig, 2010.  
ISBN: 978-3-941608-07-8
- Band XXI MORGENSTERN, U.; RIECHERT, T. (Hrsg.):  
*Catalogus Professorum Lipsiensis. Konzeption, technische Umsetzung und Anwendungen für Professorenkataloge im Semantic Web.* Leipziger Beiträge zur Informatik: Band XXI. Leipzig, 2010.  
ISBN: 978-3-941608-08-5
- Band XXII LEHMANN, J.:  
*Learning OWL Class Expressions.* Leipziger Beiträge zur Informatik: Band XXII. Leipzig, 2010.  
ISBN: 978-3-941608-09-2
- Band XXIII MEYER, K.; THIEME, M. (Hrsg.):  
*Vom Bedarf zum Innovationserfolg – In 6 Schritten gemeinsam Potentiale aktivieren.* Leipziger Beiträge zur Informatik: Band XXIII. Leipzig, 2010.  
ISBN: 978-3-941608-10-8
- Band XXIV MAICHER, L.; GARSHOL, L. M. (Eds.):  
*Information wants to be a Topic Map. Sixth International Conference on Topic Maps Research and Applications, TMRA 2010.* Leipziger Beiträge zur Informatik: Band XXIV. Leipzig, 2010.  
ISBN: 978-3-941608-11-5
- Band XXV HEYER, G.; LUY, J.-F.; JAHN, A. (Hrsg.):  
*Text- und Data Mining für die Qualitätsanalyse in der Automobilindustrie.* Leipziger Beiträge zur Informatik: Band XXV. Leipzig, 2010.  
ISBN: 978-3-941608-12-2
- Band XXVI FÄHNRIICH, K.-P.; SCHUMACHER, F.; THIEME, M.; GROSS, J. (Hrsg.):  
*(Über-)Leben in der Kreativwirtschaft - Beiträge zum Creative Summer Camp 2011.* Leipziger Beiträge zur Informatik: Band XXVI. Leipzig, 2011.  
ISBN: 978-3-941608-13-9

- Band XXVII AUER, S.; RIECHERT, T.; SCHMIDT, J. (Hrsg.):  
*Studentenkonferenz Informatik Leipzig 2011*. Leipziger Beiträge zur Informatik: Band XXVII.  
Leipzig, 2011.  
ISBN: 978-3-941608-14-6
- Band XXVIII GEBAUER, M.; STEFAN, F.:  
*Systemintegration - Eine qualitative Erhebung aus der Sicht von Integrationsdienstleistern*.  
Leipziger Beiträge zur Informatik: Band XXVIII. Leipzig, 2011.  
ISBN: 978-3-941608-15-3
- Band XXIX MEYER, K.; BÖTTCHER, M. (Hrsg.):  
*Entwicklungspfad Service Engineering 2.0 – Neue Perspektiven für die  
Dienstleistungsentwicklung*. Leipziger Beiträge zur Informatik: Band XXIX. Leipzig, 2011.  
ISBN: 978-3-941608-16-0
- Band XXX KERN, H.; KÜHNE, S. (Hrsg.):  
*Integration betrieblicher Informationssysteme und modellgetriebene Entwicklung*. Leipziger  
Beiträge zur Informatik: Band XXX. Leipzig, 2012. ISBN: 978-3-941608-17-7
- Band XXXI BÖTTCHER, M.; KLINGNER, S.; BECKER, M.; SCHUMANN, K. (Hrsg.):  
*Produktivität von Dienstleistungssystemen – Ergebnisse eines Arbeitskreises*. Leipziger Beiträge  
zur Informatik: Band XXXI. Leipzig, 2012.  
ISBN: 978-3-941608-18-4
- Band XXXII GRÄBE, H.-G.; GROEPLER-ROESSER, I. (Hrsg.):  
*MINT – Zukunft schaffen – Innovation und Arbeit in der modernen Gesellschaft*. Leipziger  
Beiträge zur Informatik: Band XXXII. Leipzig, 2012.  
ISBN: 978-3-941608-19-1
- Band XXXIII RIECHERT, T.:  
*Eine Methodologie für agiles und kollaboratives Requirements-Engineering (Dissertation)*.  
Leipziger Beiträge zur Informatik: Band XXXIII. Leipzig, 2012.  
ISBN: 978-3-941608-20-7

- Band XXXIV SCHMIDT, J.; RIECHERT, T.; AUER, S. , (Hrsg.):  
*SKIL 2012 – Dritte Studentenkonferenz Informatik Leipzig 2012*. Leipziger Beiträge zur Informatik: Band XXXIV. Leipzig, 2012.  
ISBN: 978-3-941608-21-4
- Band XXXV MEYER, K.; THIEME, M. (Hrsg.):  
*High-Tech-Services, Clustermanagement und Dienstleistungsengineering – Potentiale, Trends und Perspektiven*. Leipziger Beiträge zur Informatik: Band XXXV. Leipzig, 2012.  
ISBN: 978-3-941608-22-1
- Band XXXVI MEYER, K.; ABDELKAFI, N. (Hrsg.):  
*Smart Services and Service Science, Proceedings of the 4th International Symposium on Services Science, Leipzig (Germany), September 25, 2012*. Leipziger Beiträge zur Informatik: Band XXXVI. Leipzig, 2012.  
ISBN: 978-3-941608-23-8
- Band XXXVII STREHL, B.:  
*Innovationsmanagement im Service Center: Anforderungen, Konzeption und Realisierung einer informationstechnischen Unterstützungsleistung*. Leipziger Beiträge zur Informatik: Band XXXVII. Leipzig, 2012.  
ISBN: 978-3-941608-24-5
- Band XXXVIII KÜHNE, S.; SCHMIDT, J. (Hrsg.):  
*Betriebsführung und Instandhaltung regenerativer Energieanlagen. Fachtagung BIREA am 24. und 25. September 2012 in Leipzig*. Leipziger Beiträge zur Informatik: Band XXXVIII. Leipzig, 2012.  
ISBN: 978-3-941608-25-2
- Band XXXIX KLINGNER, S.; MEIREN, T.; BECKER, M. (Hrsg.):  
*Produktivitätsorientiertes Service Engineering – Komponenten, Kennzahlen, Anwendungen*. Leipziger Beiträge zur Informatik: Band XXXIX. Leipzig, 2012.  
ISBN: 978-3-941608-26-9
- Band XL HUMMEL, A.; KERN, H.; PRINZ, T.; DÖHLER, A. (Hrsg.):  
*Simulation im E-Commerce – Ergebnisse aus dem Forschungsprojekt SimProgno*. Leipziger Beiträge zur Informatik: Band XL. Leipzig, 2013.  
ISBN: 978-3-941608-27-6
- Band XLI MEYER, K.; THIEME, M. (Hrsg.):  
*Theory and Practice for System Services Providers in Complex Value and Service Systems – ISSS 2013 Proceedings*. Leipziger Beiträge zur Informatik: Band XLI. Leipzig, 2013.  
ISBN: 978-3-941608-28-3
- Band XLII WERNER, A.; KÜHNE, S., ARNOLD, G., SCHMIDT, J. (Hrsg.):  
*Energy EcoSystems Conference 2013, Leipzig, Germany, 23–24 September 2013, Proceedings*. Leipziger Beiträge zur Informatik: Band XLII. Leipzig, 2013.  
ISBN: 978-3-941608-29-0

Weitere Informationen und Bestellungen über: <http://www.infai.org/>.



## **Autorenverzeichnis**

*Dr. Kyrill Meyer*

Universität Leipzig, Abteilung Betriebliche Informationssysteme

*Christian Zinke*

Universität Leipzig, Abteilung Betriebliche Informationssysteme

*Michael Thieme*

Universität Leipzig, Abteilung Betriebliche Informationssysteme

*Lars-Peter Meyer*

Universität Leipzig, Abteilung Betriebliche Informationssysteme

*Florian Golemo*

Universität Leipzig, Abteilung Betriebliche Informationssysteme

*Frieder Swoboda*

Cadsys, Chemnitz

*Egbert Mauersberger*

Cadsys Chemnitz