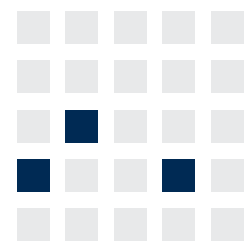


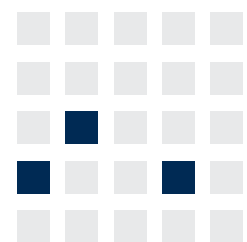


EAM-Tools im Vergleich

Praxisbeispiel eines Finanzdienstleisters



**Lehrstuhl für Wirtschaftsinformatik
Prozesse und Systeme**
Universität Potsdam



**Chair of Business Informatics
Processes and Systems**
University of Potsdam

Univ.-Prof. Dr.-Ing. habil. Norbert Gronau
Lehrstuhlinhaber | Chairholder

Mail August-Bebel-Str. 89 | 14482 Potsdam | Germany
Visitors Digitalvilla am Hedy-Lamarr-Platz, 14482 Potsdam
Tel +49 331 977 3322

E-Mail ngronau@lswi.de
Web lswi.de

Lernziele

- Was ist die Rolle von Enterprise Architecture Management in modernen Unternehmen?
- Warum benötigen Organisationen spezialisierte EAM-Tools?
- Welche Herausforderungen entstehen bei der Verwaltung komplexer IT-Landschaften (z. B. Microservices, Schatten-IT, fehlende Datenbasis)?
- Wie verläuft ein strukturiertes Auswahlvorgehen für EAM-Tools?
- Welche Kriterien eignen sich, um EAM-Tools signifikant zu vergleichen?
- Wie lassen sich Stärken und Schwächen verschiedener Tooltypen (klassisch, graphbasiert, BPM-orientiert, Portfolio-orientiert) bewerten?
- Wie kann das Fallbeispiel des Dienstleisters genutzt werden, um praktische Anforderungen an EAM-Tools abzuleiten?



Einführung

Problemlage des Fallbeispiels

Vorgehen bei der Toolauswahl

Überblick: EAM-Tool-Landschaft

Vergleich der Tools

Deep Dives: Shortlist-Tools

Empfehlung und Ausblick



Einführung

Problemlage des Fallbeispiels

Vorgehen bei der Toolauswahl

Überblick: EAM-Tool-Landschaft

Vergleich der Tools

Deep Dives: Shortlist-Tools

Empfehlung und Ausblick

Einführung und Kontext

Warum Enterprise Architecture Management (EAM)?

- EAM schafft **Transparenz** über Systeme, Daten, Prozesse und Verantwortlichkeiten.
- EAM ermöglicht die **Steuerbarkeit** komplexer IT-Landschaften.
- EAM sichert **Konsistenz** und vermeidet Redundanzen in Architektur- und Betriebsinformationen.
- EAM-Tools dienen als **zentrale Informationsdrehscheibe**, auf die alle Bereiche zugreifen können.

Herausforderungen moderner Organisationen

- **Viele heterogene Systeme**, oft historisch gewachsen
- **Unklare Verantwortlichkeiten** für Anwendungen, Schnittstellen und Daten
- **Schnelle technologische Veränderungen** (Cloud, Microservices, neue Plattformen)
- **Hoher Integrationsaufwand** und fehlende Übersicht über Abhängigkeiten
- **Steigende Anforderungen** an Sicherheit, Compliance und Datenqualität



Einführung

Problemlage der Fallbeispiels

Vorgehen bei der Toolauswahl

Überblick: EAM-Tool-Landschaft

Vergleich der Tools

Deep Dives: Shortlist-Tools

Empfehlung und Ausblick

Problemlage des Finanzdienstleisters

Sehr große IT-Landschaft

- viele verschiedene Anwendungen und Services, historisch gewachsen

Fehlende Gesamtübersicht

- unklare Abhängigkeiten, inkonsistente Informationsbasis

Uneinheitliche Modellierung

- unterschiedliche ArchiMate-Verständnisse und Strukturen

Unvollständige Dokumentation

- Systeme, Schnittstellen, Verantwortlichkeiten nicht vollständig erfasst

Kein zentrales EAM-Repository

- Informationen verteilt in Excel, Confluence, Teams, Köpfen

Mangelnde Governance

- keine verbindlichen Modellierungsstandards, Rollen, Pflegeprozesse

Hohe operative Risiken

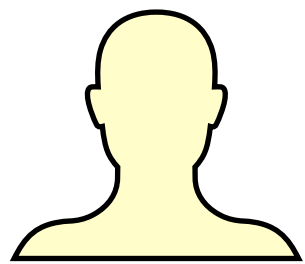
- Sicherheitslücken, Redundanzen, erschwerte Wartung und Planung

Ausgangslage

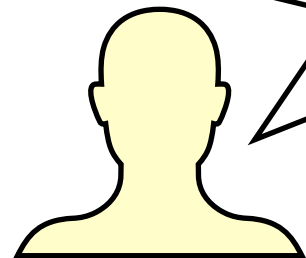
Das Framework des alten *Kundenportal 1.0* wird bald nicht mehr unterstützt.
Welche Systeme – z. B. *Kreditmanagementsystem*, *Konto-Backend*, *Identitätsdienst* – hängen noch davon ab?

Im *Reporting- und Risikocockpit* ist wieder ein Fehler aufgetreten – ist das ein Problem in der *Data Warehouse DB* oder in der Schnittstelle zum *Transaktionsmonitoring*?

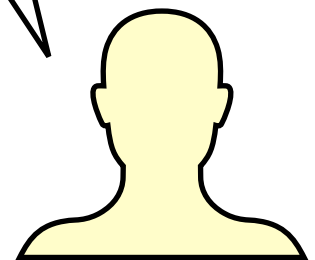
Müssen wir die Kundendaten wirklich erneut manuell erfassen?
Die liegen doch bereits im *Customer Master Data System* und im *KYC/AML-System*.



Systemarchitekt



IT-Betrieb



PO/RE

Beispiele aus dem Alltag eines Finanzdienstleisters: Fehlende Transparenz über Systeme, Verantwortlichkeiten und Datenflüsse führt zu Verzögerungen, Fehlerquellen, Nacharbeiten und erhöhtem Risiko.

Lösung für diese Ungewissheiten:

Enterprise Architektur Management System

Offene Fragen

- Wie hängen Anwendungen miteinander zusammen? Welche Daten werden zwischen diesen ausgetauscht?
- Werden die Daten eines Systems auch in einem anderen System gespeichert?
- Bei welchen Systemen sind unsere geschäftskritischen Prozesse direkt betroffen?
- Wer ist der Ansprechpartner für ein System oder einen Service?
- Wie gehe ich bei einem Systemausfall vor?
- Welche Daten welcher Systeme werden auf diesem Server verwaltet?

Funktionen EA-System

- Modellierung und Dokumentation der genutzten Anwendungen sowie deren Datenaustausch.
- Risikobewertung der Systeme bezüglich Einfluss und Ausfallrisiko.
- Einheitliche Dokumentation von Verantwortlichkeiten über die Systeme.
- Definition von Reportprozessen.

Ziel eines EA-Tools: Klare Verantwortlichkeiten und transparente Entscheidungsprozesse ermöglichen und etablieren.



Einführung

Problemlage der Fallbeispiels

Vorgehen bei der Toolauswahl

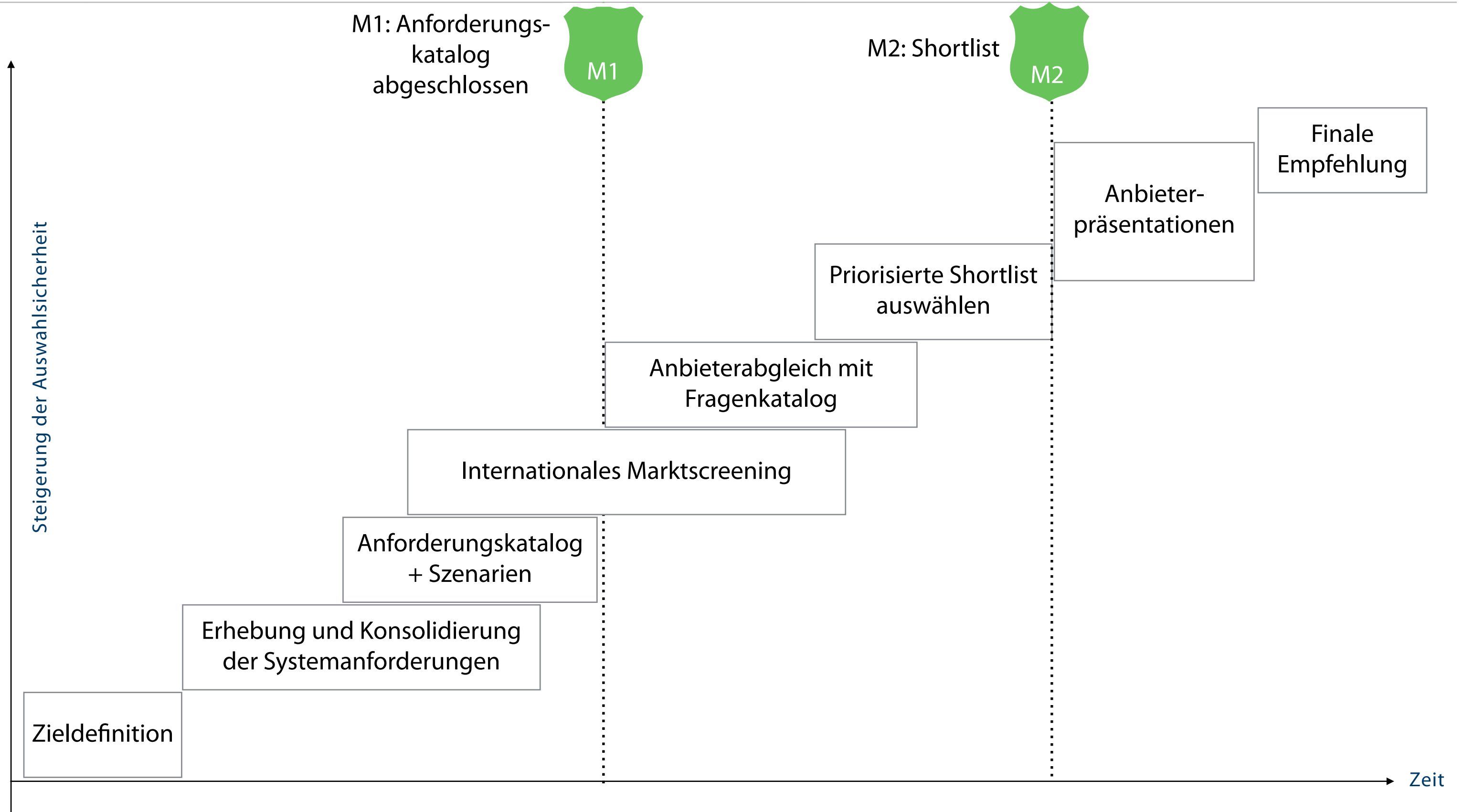
Überblick: EAM-Tool-Landschaft

Vergleich der Tools

Deep Dives: Shortlist-Tools

Empfehlung und Ausblick

Vorgehen bei der Toolauswahl



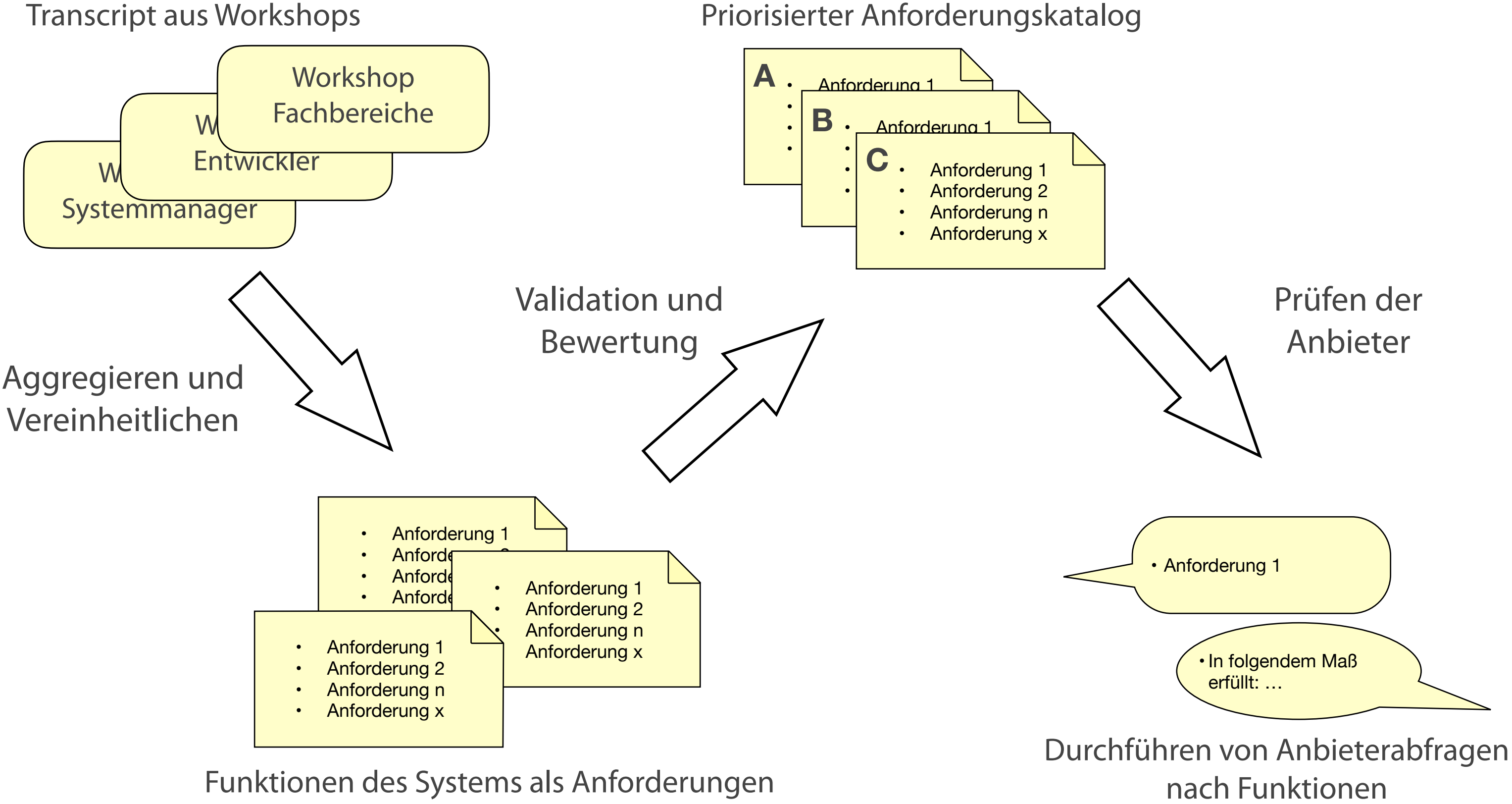
Zieldefinition

- Der Finanzdienstleister benötigt für die digitale Weiterentwicklung ein zukunftsfähiges Modell zur Steuerung ihrer Anwendungs- und Prozesslandschaft.
- Ziel ist die Auswahl eines Enterprise Architecture (EA)-Tools zur ganzheitlichen Sicht auf Systeme, Datenflüsse und Verantwortlichkeiten.

Warum dieser Schritt jetzt wichtig ist:

- Wachsende Komplexität der IT-Landschaft und steigender Bedarf an Transparenz über Systeme, Prozesse und deren Abhängigkeiten.
- Eine fundierte Toolauswahl ist mit dem aktuellen Datenstand (nach gezielter Anreicherung) bereits möglich.
- Die spätere Nutzung des Tools setzt eine strukturierte Erhebung und Dokumentation der Architekturinformationen voraus.
- Synergieeffekte zwischen Prozessintegration und EA-Aufbau können jetzt aktiv genutzt werden.

Anforderungserhebung



Methodisches Vorgehen in den Interviews

Ermitteln des Mengengerüsts

Mengengerüst

- Identifikation von Architektur-Objekten, die im Tool gepflegt werden sollen (z. B. Systeme, Schnittstellen, Verantwortlichkeiten)
- Abfrage vorhandener und fehlender Attribute je Objekt (z. B. Technologie, Eigentümer:in, Beziehungen)
- Einordnung der Datenquellen: Wo liegen diese Informationen heute (z. B. GitLab, Confluence, Excel)?

Konzept zur Vorbereitung

- Strukturierte Leitfragen pro Rolle zur Erhebung des Informationsbedarfs (z. B. „Welche Informationen fehlen Ihnen im Alltag?“)
- Use-Case-basierter Einstieg: Erhebung über User Stories mit Systembezug und Attributsicht
- Dokumentation im Template durch das Projektteam während des Gesprächs; Rückgabe zur Freigabe durch Stakeholder

Das Mengengerüst entsteht schrittweise aus realen Nutzungsszenarien – keine Vollerhebung, sondern gezielte Verdichtung anhand priorisierter Rollen und Use Cases.

Beispiel - Fachliche Anforderungen (EAM-Tool)

| Nr. | Anforderung |
|-----|--|
| 1 | Abbildung der vollständigen Applikationslandschaft mit Systemen, Modulen, Services und Abhängigkeiten. |
| 2 | Darstellung von technischen und fachlichen Verantwortlichkeiten (Business Owner / Technical Owner). |
| 3 | Modellierung von Geschäftsprozessen und deren Verknüpfung mit IT-Systemen. |
| 4 | Möglichkeit zur Dokumentation von Datenflüssen zwischen Systemen. |
| 5 | Hinterlegung von Betriebs-, Sicherheits- und Compliance-Informationen pro System. |
| 6 | Erstellung von Impact-Analysen bei Änderungen (z. B. Abschaltung eines Systems). |
| 7 | Unterstützung einer Systemklassifizierung (z. B. Kritikalität, Datenschutz, Verfügbarkeit). |
| 8 | Modellierung von Schnittstellen inkl. Datentypen, Protokollen und Verantwortlichkeiten. |
| 9 | Möglichkeit zur Bewertung der Systemlandschaft anhand definierter Kriterien (z. B. Reifegrad, Risiko, Kosten). |
| | |

Beispiel - Technische Anforderungen (EAM-Tool)

| Nr. | Anforderung |
|-----|--|
| 1 | Ist eine Anbindung an Jira zur Synchronisation von User Stories, Epics und Tickets möglich? |
| 2 | Können Architekturdokumentationen in Confluence automatisch verlinkt bzw. eingebunden werden? |
| 3 | Können Confluence-Modelle in das Tool importiert und aus dem Tool exportiert werden? |
| 4 | Werden Systemdaten automatisch aus der CMDB aktualisiert (z. B. Nightly Import)? |
| 5 | Können BI-Datenquellen angebunden und im EAM zur Analyse genutzt werden? |
| 6 | Können externe Dokumente im Tool referenziert oder eingebettet werden? |
| 7 | Lässt sich im Tool ein zentrales Schnittstelleninventar (System-zu-System) aufbauen und pflegen? |
| 8 | Lässt sich im Tool der Lifecycle von Schnittstellen (Planung, Betrieb, Außerbetriebnahme) modellieren? |
| 9 | Lassen sich im Tool Zuständigkeiten für Schnittstellen (Owner, technische Kontaktperson) hinterlegen? |
| | ... |

Beispiel - Governance-Anforderungen (EAM-Tool)

| Nr. | Anforderung |
|-----|--|
| 1 | Unterstützung eines rollenbasierten Berechtigungskonzepts (Lesen, Bearbeiten, Freigeben). |
| 2 | Definition und Verwaltung von Modellierungsrichtlinien und Namenskonventionen. |
| 3 | Möglichkeit zur Einrichtung von Review- und Freigabeprozessen für Modelle und Objekte. |
| 4 | Versionierung von Architekturobjekten und Modellen inkl. Änderungsnachverfolgung. |
| 5 | Automatische Qualitätsprüfungen (z. B. fehlende Attribute, ungültige Beziehungen). |
| 6 | Benachrichtigungsfunktionen bei Änderungen an kritischen Architekturartefakten. |
| 7 | Unterstützung eines Lifecycle-Managements für Systeme, Prozesse und Schnittstellen. |
| 8 | Möglichkeit zur Definition verbindlicher Verantwortlichkeiten (z. B. System Owner, Data Owner). |
| 9 | Etablierung von Standards zur Dokumentationspflege (z. B. Pflichtattribute, Mindestinformationen). |
| | ... |

Einwahldaten

- URL: <https://quiz.lswi.de/login>
- Lecture Code: aba19





Einführung

Problemlage der Fallbeispiels

Vorgehen bei der Toolauswahl

Überblick: EAM-Tool-Landschaft

Vergleich der Tools

Deep Dives: Shortlist-Tools

Empfehlung und Ausblick

Überblick: EAM-Tool-Landschaft

Breites Spektrum an EAM-Werkzeugen

- klassische EAM-Suiten
- modellgetriebene Plattformen
- graphbasierte, API-orientierte Tools
- BPM- und Governance-orientierte Lösungen
- Portfolio- und Strategietools mit EAM-Erweiterungen

Zentrale Unterscheidungsmerkmale

- Metamodell-Flexibilität
- Integrations- und API-Fähigkeiten
- Visualisierung von Abhängigkeiten
- Governance- und Kollaborationsfunktionen
- Kosten- und Betriebsmodelle

Marktdynamik

- Übergang von statischen Modellen zu lebenden Datenplattformen
- stärkere Automatisierung durch APIs und Integrationen
- wachsender Bedarf durch Microservices- und Cloud-Landschaften



Einführung

Problemlage der Fallbeispiels

Vorgehen bei der Toolauswahl

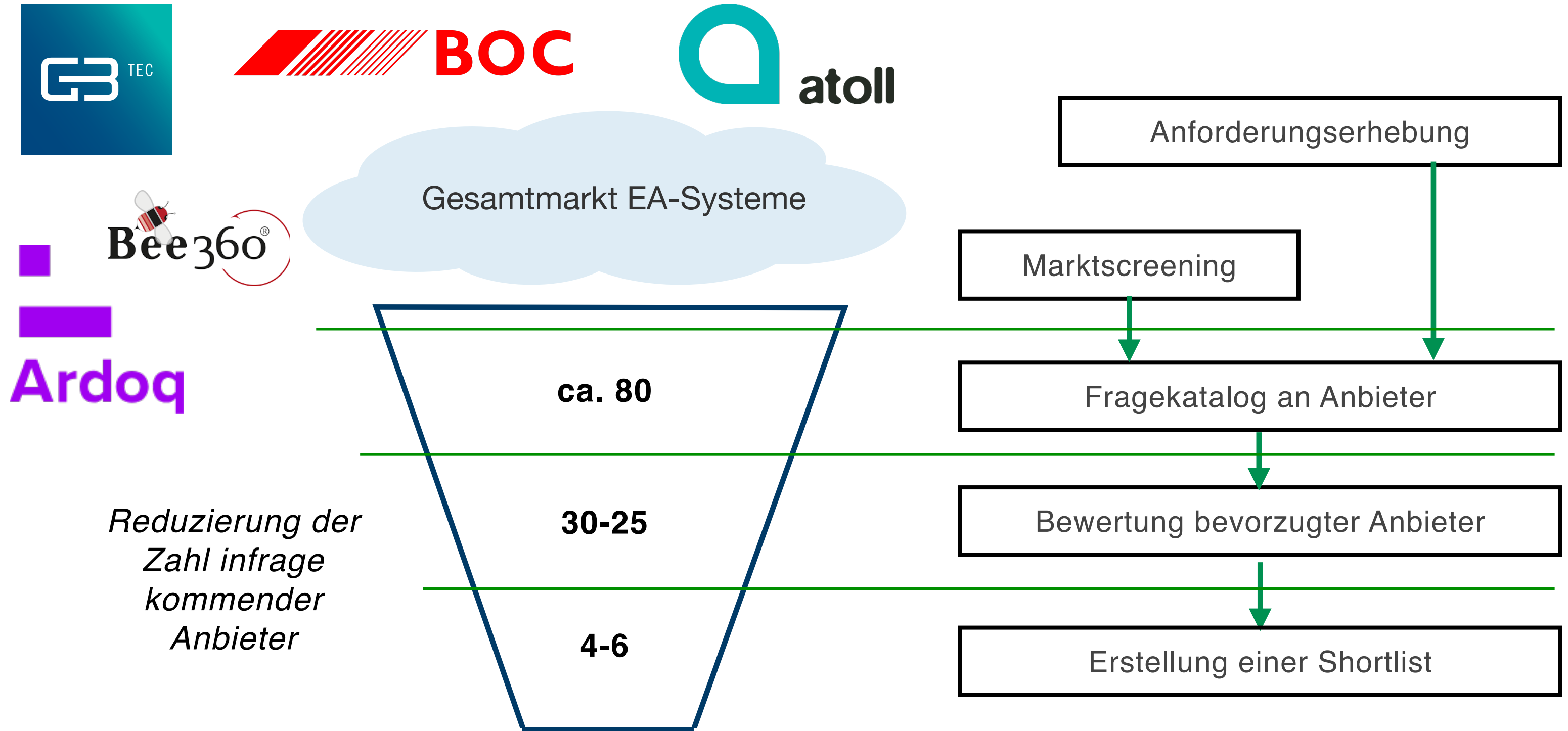
Überblick: EAM-Tool-Landschaft

Vergleich der Tools

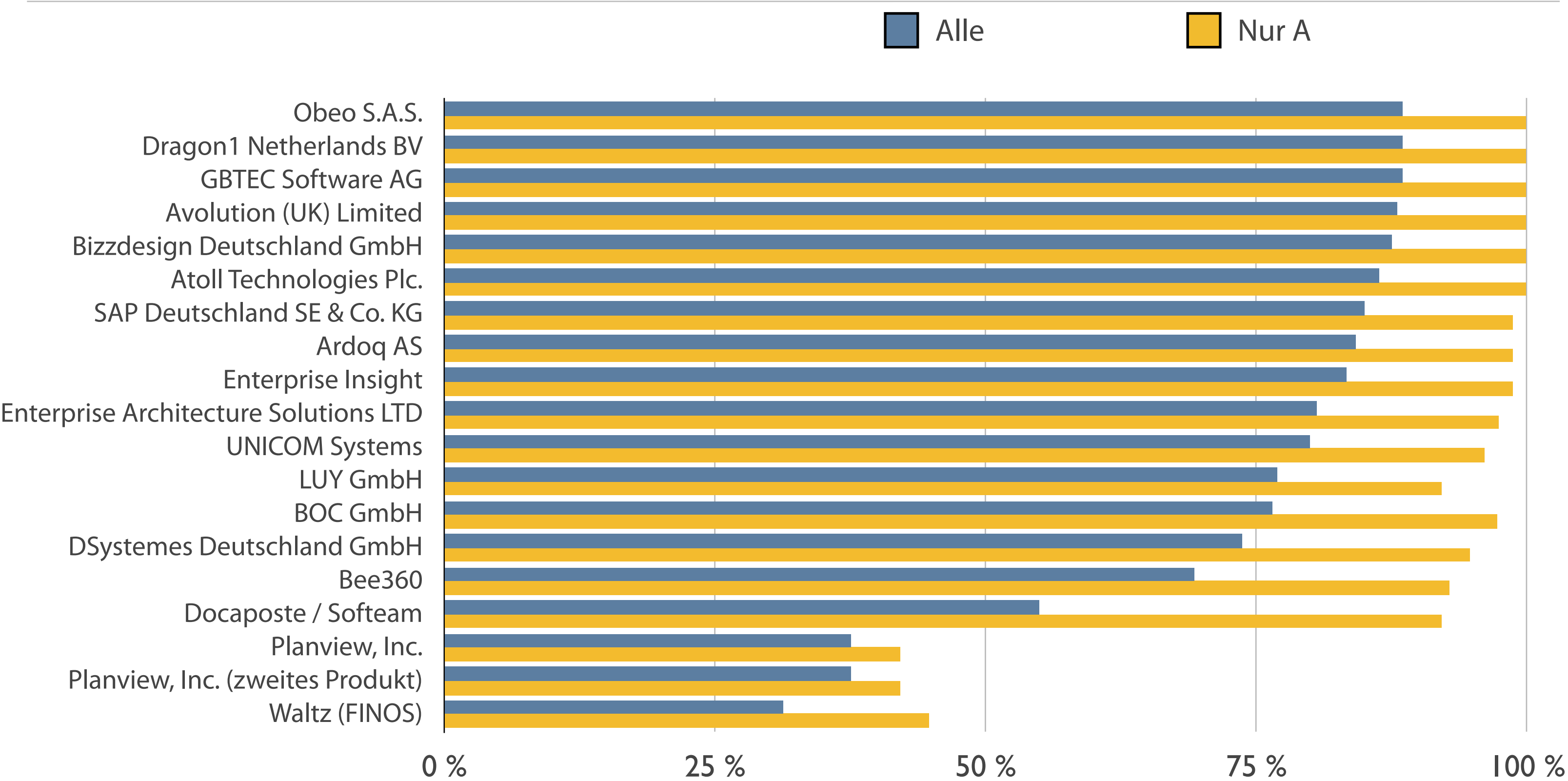
Deep Dives: Shortlist-Tools

Empfehlung und Ausblick

Weiteres Vorgehen nach der Anforderungserhebung

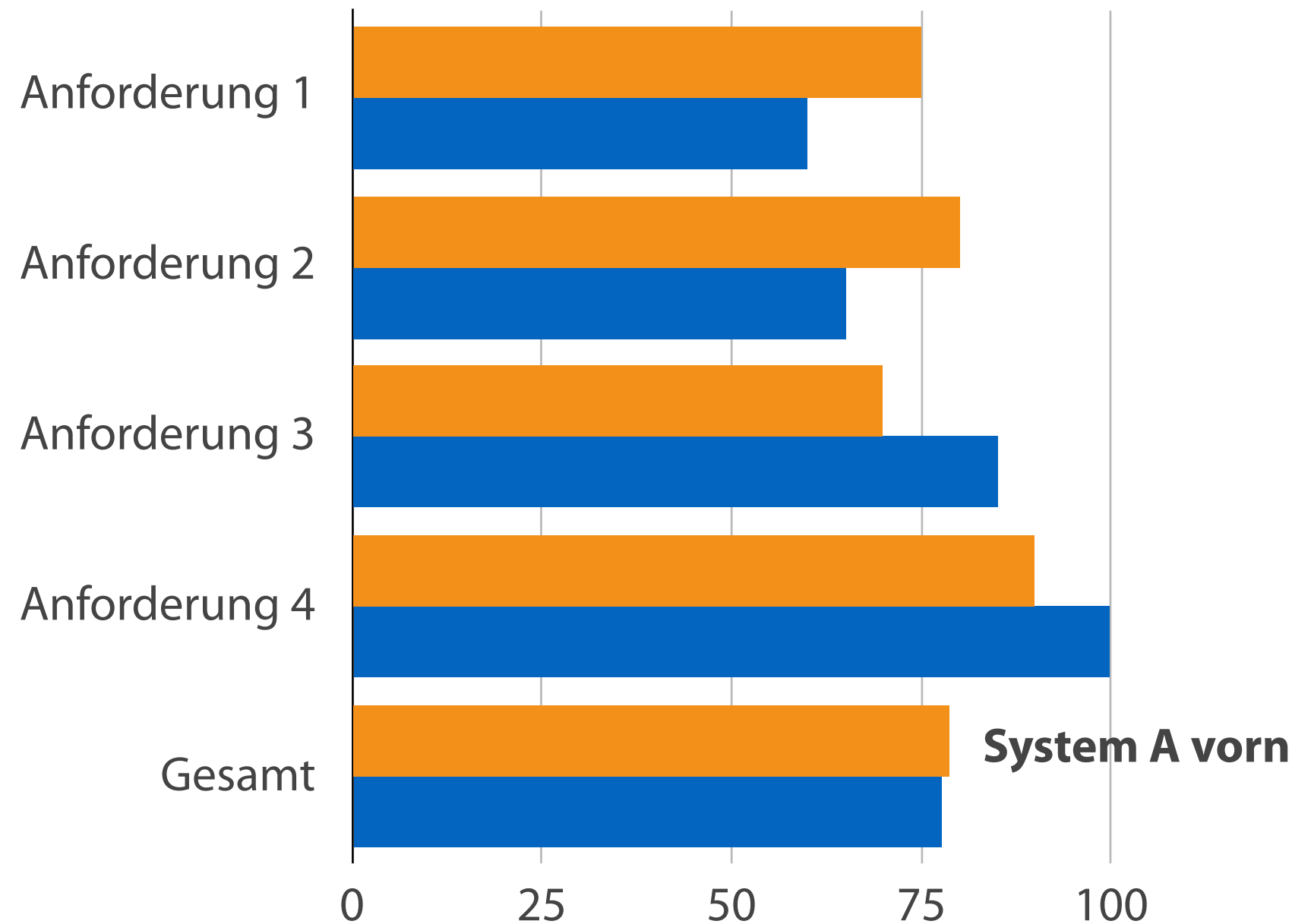


Ergebnisse bei Betrachtung der Anforderungen

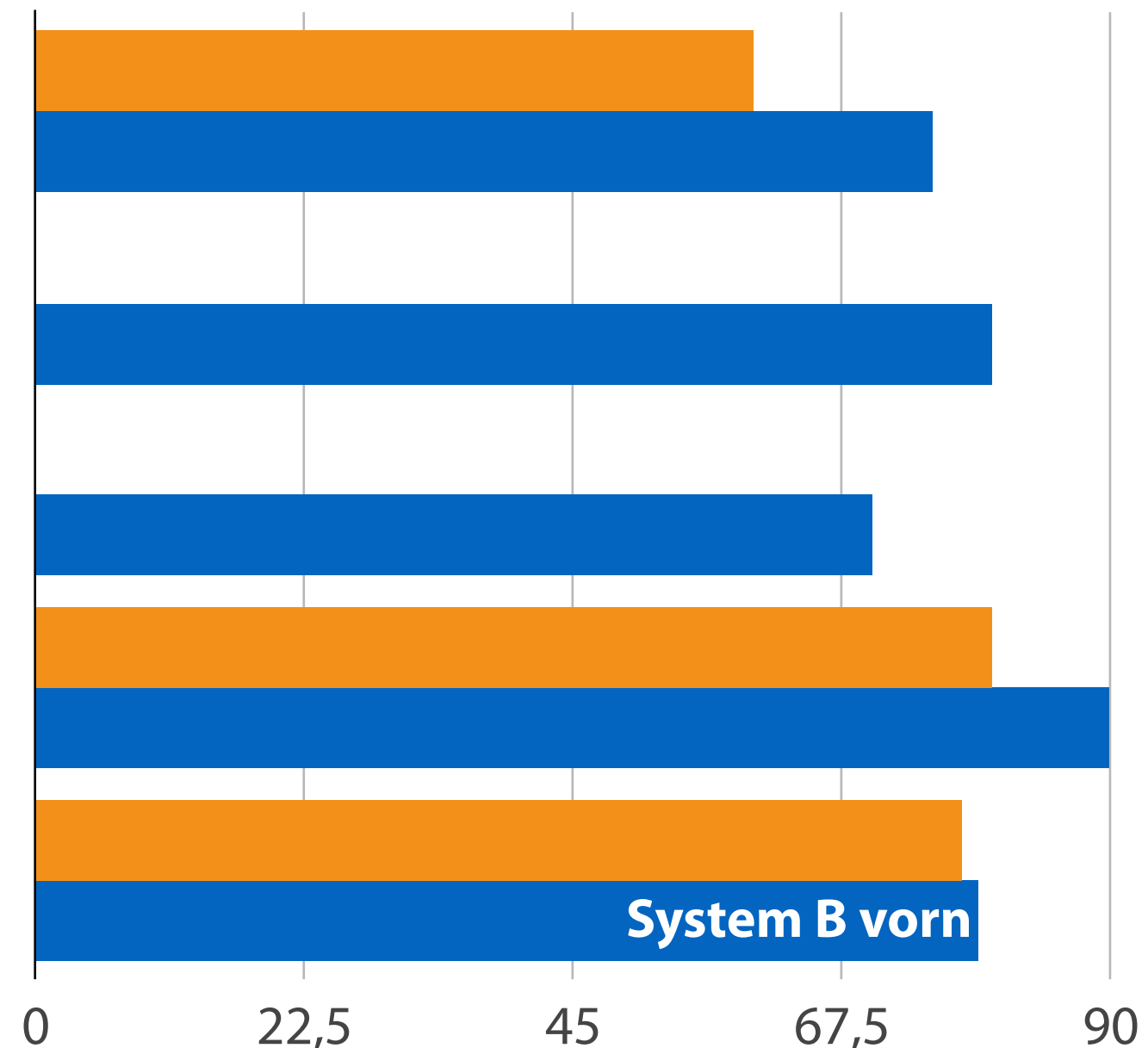


Unterschied zwischen einer Systemauswahl nach Anforderungen und einer Auswahl nach Priorisierung

Erfüllung der Anforderungen



Priorisierung



Während nach Funktionen System A bevorzugt würde, ist es nach Priorisierung B.



Einführung

Problemlage des Fallbeispiels

Vorgehen bei der Toolauswahl

Überblick: EAM-Tool-Landschaft

Vergleich der Tools

Deep Dives: Shortlist-Tools

Empfehlung und Ausblick

Kurzprofil

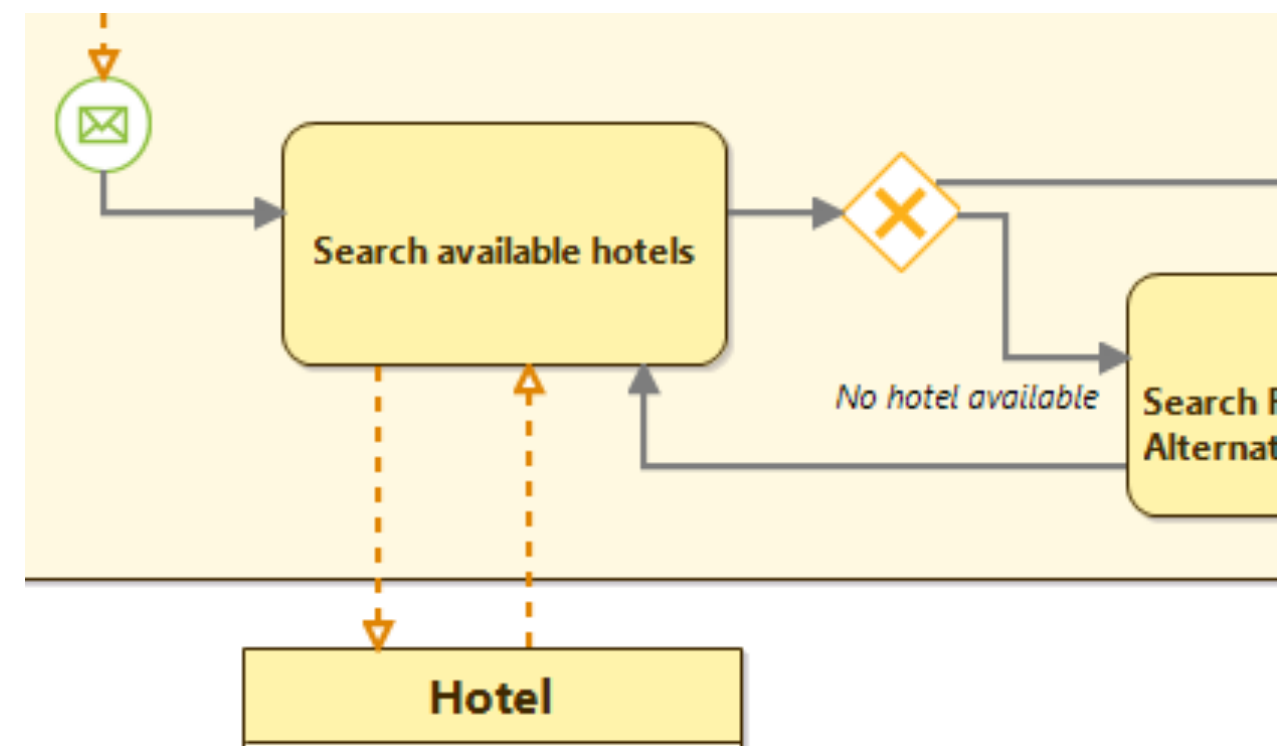
- Produkt: Obeo SmartEA
- Gründung 2005, ca. 60 Mitarbeitende, ~100 Kunden
- Schwerpunktmarkt: Frankreich, v. a. große Konzerne (z. B. Orange, EDF, Engie)

Schwächen/Risiken

- Eher technik- und entwicklerlastiges Tool → höherer Einführungsaufwand
- Weniger „Mainstream“ im deutschen Markt; Community kleiner als bei ADOIT / GBTEC
- UX / Benutzerfreundlichkeit noch zu verifizieren

Technologische Basis und API

- Modellgetriebener Ansatz (Eclipse-Ökosystem)
- REST-API vorhanden; Integrationsfähigkeiten gegeben, aber nicht im Detail standardisiert ausgewiesen



Kurzprofil

- Produkt: *BIC EAM*
- Gründung 2005, ~350 Mitarbeitende, ~2000 Installationen
- Starke Referenzen: AOK, Siemens, Otto Group, RWE

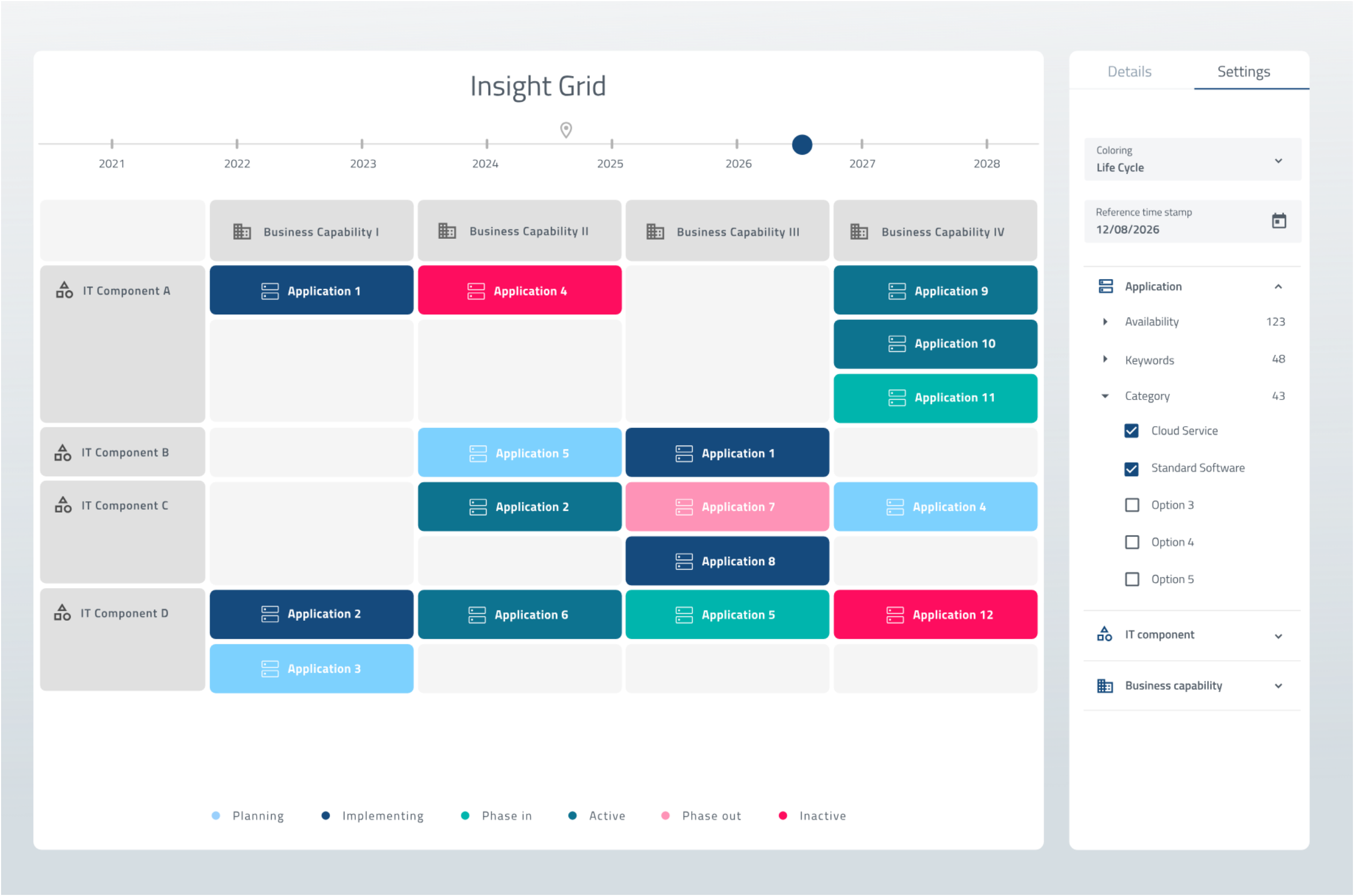
Technologische Basis und API

- Umfassende REST-API, tiefe Integration in das BIC-BPM-Ökosystem
- Gute Kopplung zu Prozessmanagement-/Workflow-Tools von GBTEC

Schwächen/Risiken

- Eher BPM-/Prozessfokus; Microservice-/Service-Sichten müssen geprüft werden
- Data-Quality-/Analytics-Funktionen nur im mittleren Bereich → Ausbaupotenzial
- Metamodell-Flexibilität nicht so hoch wie bei Ardoq/Obeo

Einordnung von Anwendungen und Business-Capabilities über Zeiträume hinweg



Quelle: <https://www.gbtec.com/de/software/bic-eam/application-management/>

Kurzprofil

- Produkt: *Dragon1*
- Gründung 2005, sehr kleines Team (~5 MA), ~60 Kunden weltweit
- Referenzen u. a. IBM, Eurobank, Staatsolie

Stärken

- Sehr visualisierungsstark: gut für Kommunikation mit Management
- Robuste Basis in Domänen-, Prozess- und Systemmodellierung
- Gut geeignet für schnelle Architektur- und Risikoübersichten

Technologische Basis und API

- Starke Visualisierungsplattform
- API: CRUD-Operationen, JSON/XML, Echtzeit-Datenaustausch, Modell- & Architekturzugriff

Schwächen/Risiken

- Sehr kleines Unternehmen → Liefer- und Zukunftssicherheit kritisch
- Governance-, Reporting- und Data-Quality-Funktionen weniger ausgereift
- Proprietärer Ansatz, geringere Marktverbreitung

Fazit für den Anwendungsfall: Spannend als visualisierungsstarke Lösung, aber mit Risiko bzgl. Stabilität und Governance-Reife

Dragon1

Detaillierte Modellierung von Architektur-, Prozess- und Capability-Strukturen

The screenshot displays the Dragon1 Enterprise Architecture tool interface. The top navigation bar includes a language selector (English (US)), a search bar, and various utility icons. Below this, a tabbed interface shows 'Repository' as the active tab. The main workspace is divided into three panels:

- Explorer (Left):** A tree view showing the hierarchy of the '123 Enterprise Architecture' model, including 'Architecture Design Book', 'Data', 'Applications', 'Business Processes', and 'Capabilities'. The 'Manage Risk' capability is selected under 'Capabilities'.
- Entities (Center):** A form for editing the 'Manage Risk' capability. It includes fields for 'System ID' (50498), 'System GUID', 'Name' (Manage Risk), 'Title' (Manage Risk), 'Reference' (BC001), 'Image Location' (/images/business-capability-risk-management.p), and 'Description / Specification' (A bank manages risk.). A circular diagram titled 'Risk Management' is displayed, showing a cycle of four stages: Measure, Assess, Evaluate, and Manage.
- List Content Items (Right):** A list of available content items for linking to the model, including Organizations, Goals, Domains, Areas, Functions, Stakeholders, Issues, Needs, Requirements, Products, Services, Capabilities, Processes, Applications, and Interfaces.

The bottom of the interface features a Windows taskbar with various application icons and a system tray showing the time (09:00) and date (30-10-2018).

Kurzprofil

- Produkt: *ADOIT*
- Gründung 1995, ~80 Mitarbeitende, >10.000 Kunden weltweit
- Referenzen: u. a. Vienna Airport, Emmi Group, Molson Coors

Stärken

- Sehr hohe Marktverbreitung, viele Best Practices
- Starke Governance-/Rollenmodelle, ausgereifte EAM-Analysen
- Metamodell ist flexibel und gut anpassbar
- Gut zur Abdeckung komplexer Organisations- und Applikationslandschaften

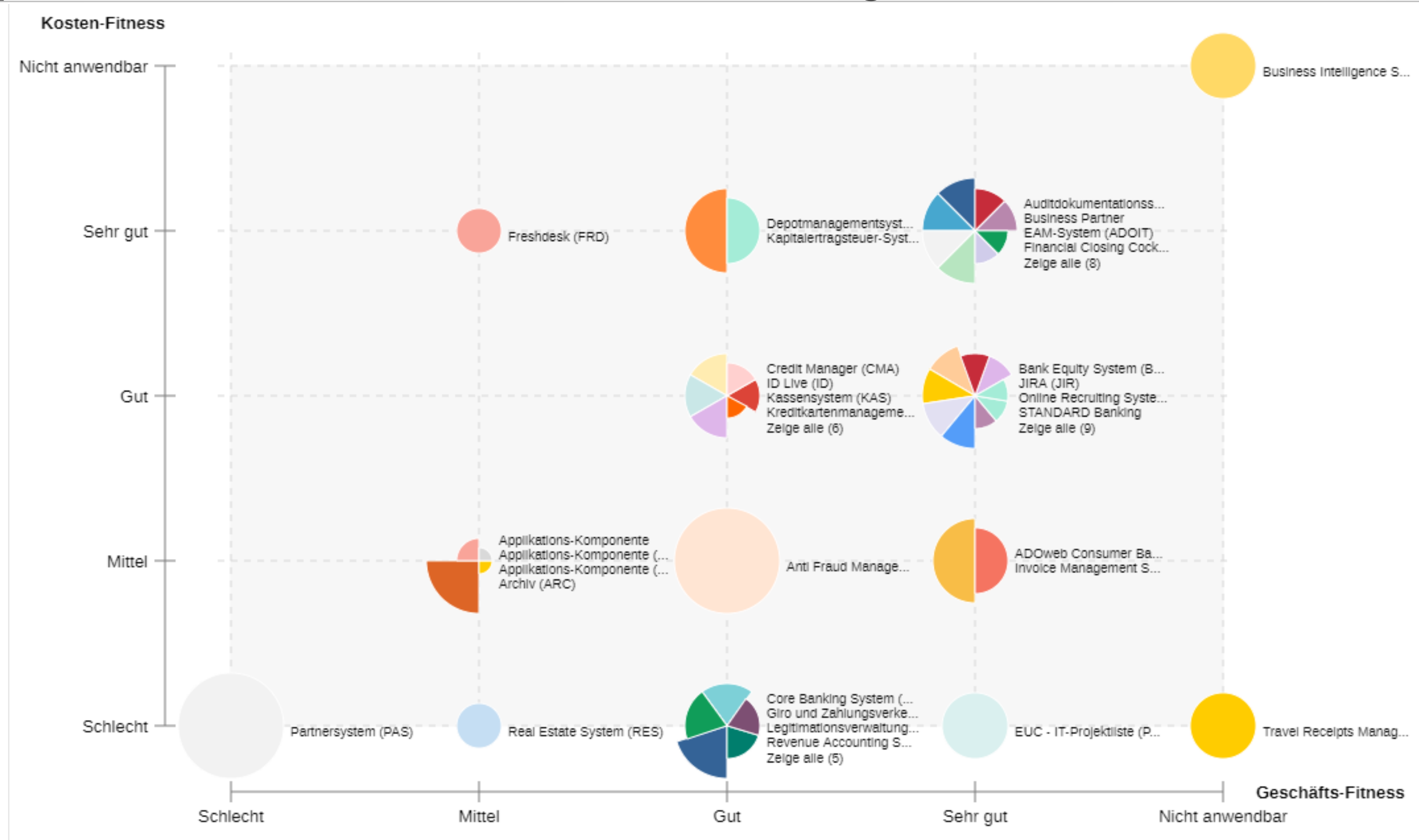
Technologische Basis und API

- Vollständig dokumentierte REST-API (CRUD, JSON/XML)
- Reifes Repository- und Governance-Modell, etablierter EAM-Standard

Schwächen/Risiken

- Einführung und Customizing tendenziell aufwendiger
- Lizenz- und Betriebskosten im oberen Segment

Application-Assessment: Governance- und Bewertungsfunktionalität



Kurprofil

- Produkt: *Ardoq*
- Gründung 2013, ~250 Mitarbeitende, ~500 Kunden
- Referenzen: u. a. STIHL, UMCG, SmartestEnergy, Folksam

Stärken

- Top in API- & Integrationsfähigkeit → ideal für automatisierte Datensynchronisation
- Starke graphbasierte Visualisierungen (Abhängigkeiten, Impact-Analysen)
- Modernes technisches Fundament, gut skalierbar für >250 verschiedene Anwendungen und >200 Microservices

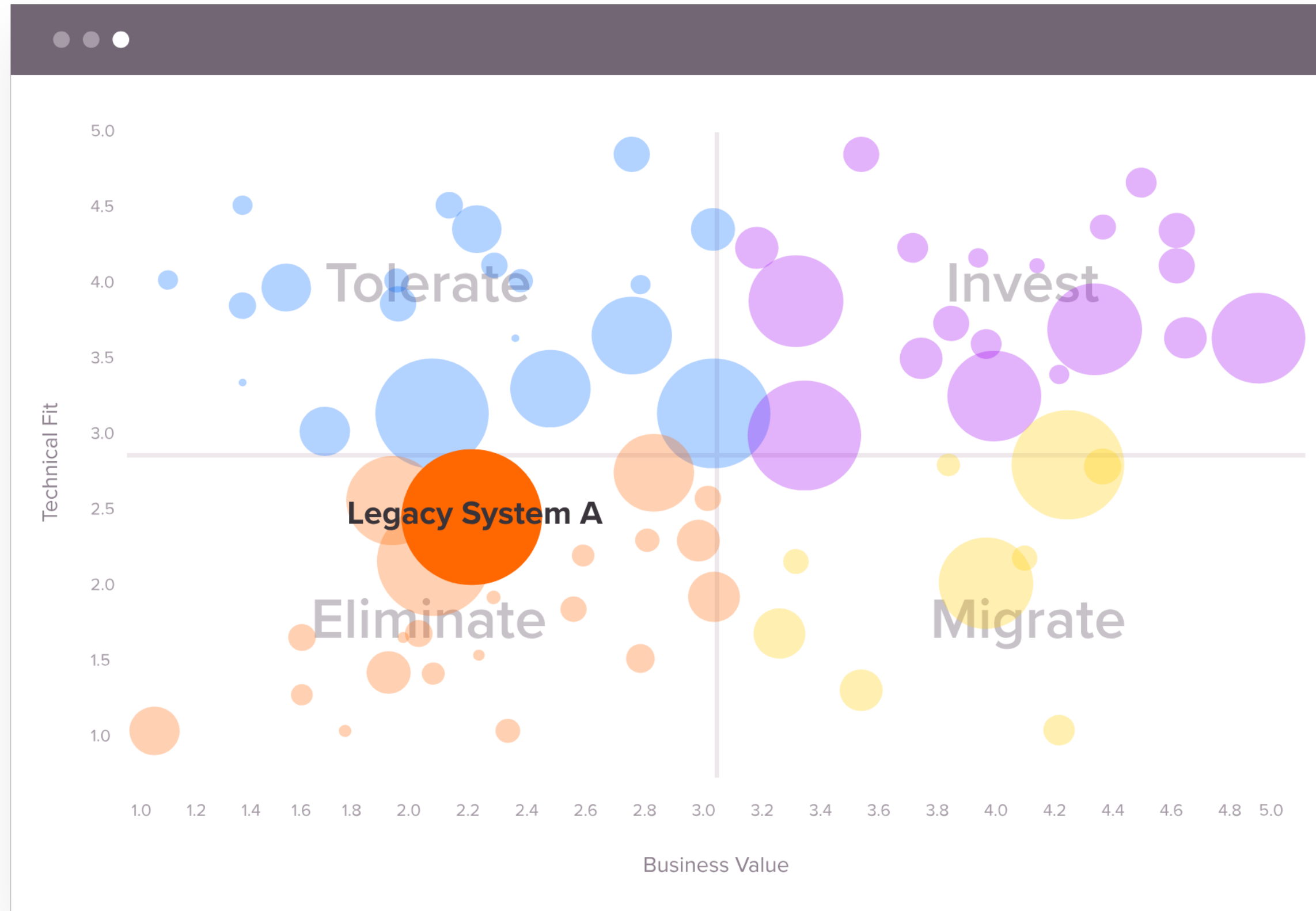
Technologische Basis und API

- Sehr umfangreiche REST-API (CRUD, Bulk, Filter, Webhooks, OAuth2, Rate Limiting)
- Vollständig graphbasierte Datenhaltung, moderne Cloud-native Plattform

Schwächen / Risiken

- Governance-/Prozessfunktionen weniger formal ausgeprägt als bei ADOIT/GBTEC
- Starker SaaS-Fokus → On-Prem-Optionen und Datenschutzerfordernungen prüfen
- Lernkurve für Fachbereiche (neues Paradigma, Graphdenken)

Portfolio-Bubble-Plot verbindet Business- und IT-Bewertungskriterien miteinander



Kurzprofil

- Produkt: SAMU EA Tool
- Gründung 2001, ~40 Mitarbeitende, ~40 Kunden
- Referenzen: OTP Bank, Magyar Telekom, Erste Bank, Wizz Air

Stärken

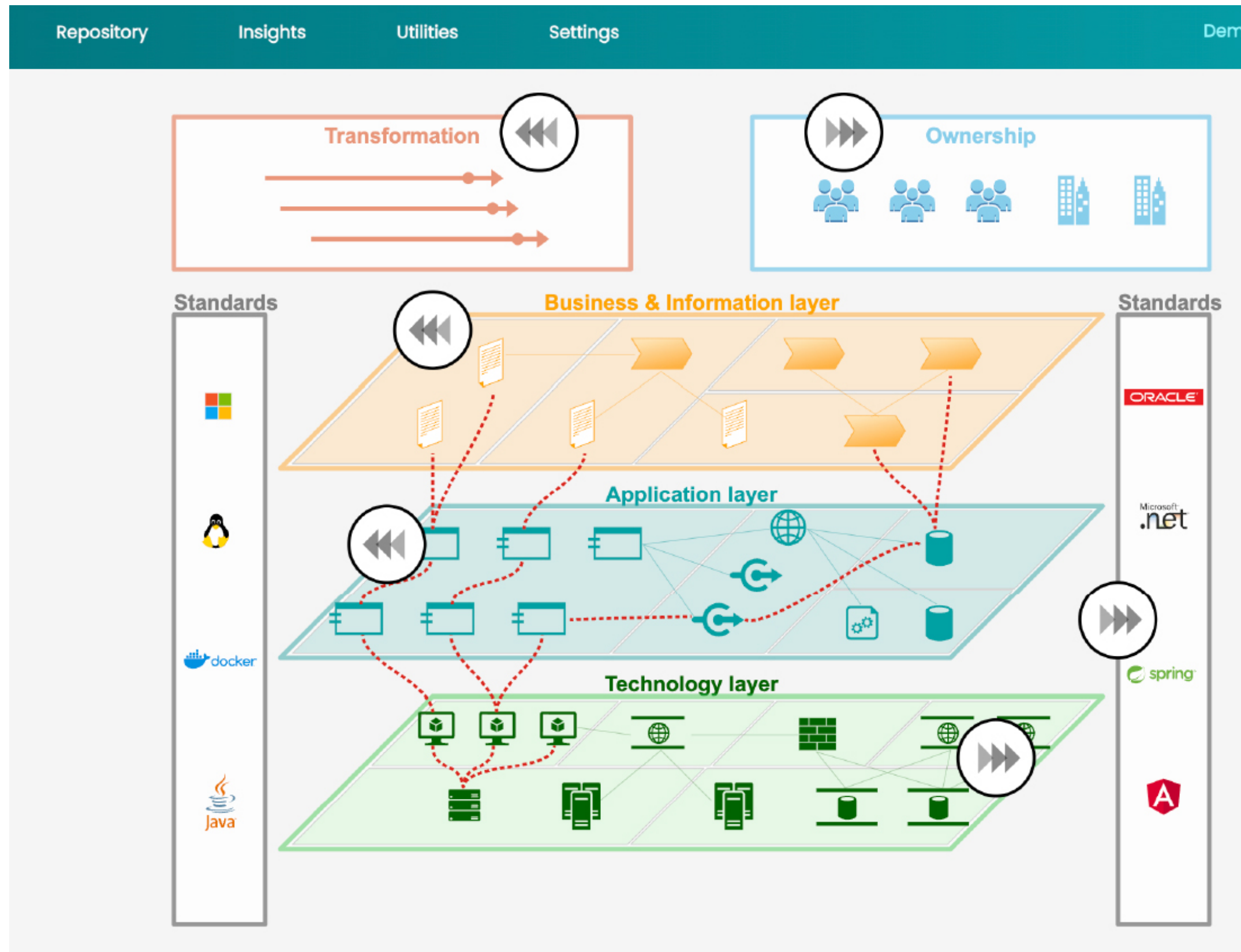
- Gut geeignet für Integration in Datenplattformen
- Betonung von Metamodell und Beziehungen; strukturiertes Repository
- Preislich meist unter den großen EAM-Suites

Technologische Basis und API-Fähigkeiten

- REST-API für Datenimport/-export, Metamodellabfragen, Echtzeitabfragen und CRUD
- Fokus auf Metamodellierung und Objektbeziehungen

Schwächen/Risiken

- Kleinere Marktpräsenz und Community
- UX, Reporting und Governance-Funktionen weniger prominent
- Weniger out-of-the-box Best Practices für große Enterprises



Kurzprofil

- Produkt: *Bee360*
- Gründung 2003, ~85 Mitarbeitende, Kunden v. a. im Versicherungs- und Handelsumfeld
- Referenzen: LVM, Zurich Gruppe, Witt-Gruppe

Stärken

- Sehr stark im Enterprise Portfolio Management (Projekte, Budgets, Strategien)
- Gute Governance- und Steuerungsfunktionen für IT-Portfolio & Projekte
- Eignet sich für strategische Steuerung von Transformationen

Technologische Basis und API-Fähigkeiten

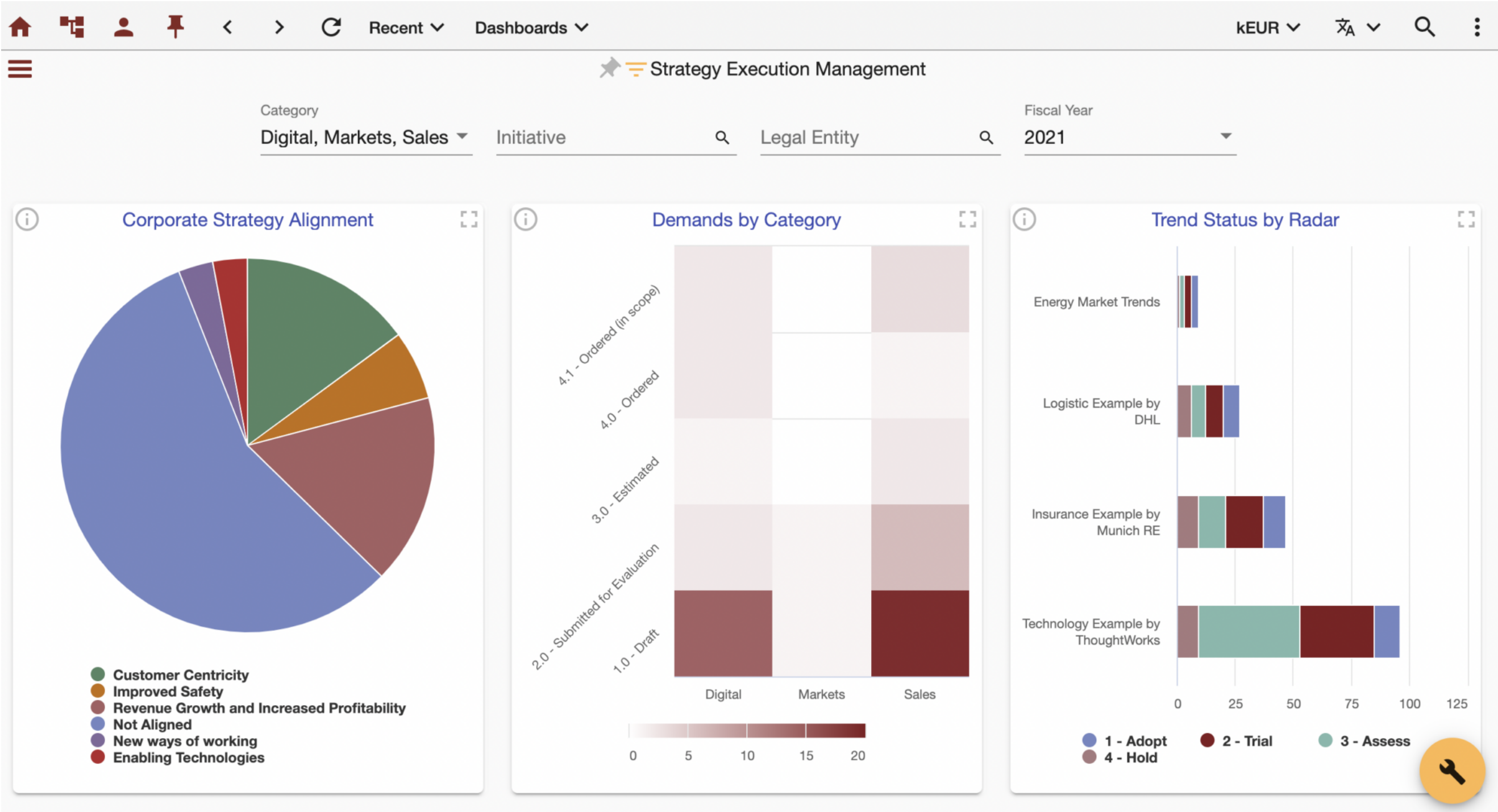
- Grundlegende REST-API (Reporting, Projektendpunkte, ausgewählte Datenobjekte)
- Fokus auf Portfolio-, Projekt- und Strategieprozesse

Schwächen/Risiken

- EAM-Metamodell nicht vollständig offen dokumentiert
- Eher Portfolio- als klassisches EAM-Tool; Microservice-/Architektur-Sicht begrenzt
- Integrations- und Datenmodell-Tiefe im Architekturkontext zu prüfen

Bee360

Strategischen Portfolio- und Demand-Management



Vergleichstabelle: EAM-Tools im Überblick

| Tool | Metamodell-Flexibilität | API/Integration | Visualisierung/Analyse | Governance/Collaboration | Kosten/Implementierungs-aufwand |
|---------------------------------|---|-----------------------------------|--|------------------------------------|-------------------------------------|
| Obeo | sehr flexibel (modellgetrieben, Eclipse-basiert) | technisch gut, aber hoher Aufwand | möglich, aber nicht out-of-the-box | gering ausgeprägt, Entwicklerfokus | hoher Aufwand, teuer in Setup |
| Dragon1 | flexibel, aber proprietär | begrenzte Integrationen | stark visuell | begrenzte Governance | geringe Kosten, moderater Aufwand |
| GBTEC (BIC) | eingeschränkt anpassbar | solide, v. a. BPM-orientiert | gut für Prozesse, weniger für EAM | sehr gute Governance | mittlere Kosten |
| BOC Group (ADOIT) | sehr flexibel, EAM-fokussiert | ausgereifte Integrationen | starke EAM-Analysen | starke Governance & Rollenmodelle | höherer Aufwand und Kosten |
| Ardoq | hochgradig flexibel | exzellente API, automationsstark | graphbasierte moderne Visualisierungen | kollaborativ, aber weniger formal | mittlere Kosten, geringe Setup-Zeit |
| Atoll (HOPEX/AtollTools) | gering flexibel | geringe Integrationsfähigkeit | standardisierte Sichten, wenig dynamisch | schwach ausgeprägt | niedrigere Kosten |
| Bee360 | moderat flexibel | solide API | Portfolio- und Managementfokus | starke Governance, Portfolio-fähig | teuer, hoher Einführungsaufwand |

Einwahldaten

- URL: <https://quiz.lswi.de/login>
- Lecture Code: aba19



Quelle:





Einführung

Problemlage des Fallbeispiels

Vorgehen bei der Toolauswahl

Überblick: EAM-Tool-Landschaft

Vergleich der Tools

Deep Dives: Shortlist-Tools

Empfehlung und Ausblick

Empfehlung und Ausblick

Fokus für den Finanzdienstleister:

- Tool mit starker API, hoher Automatisierbarkeit und moderner Visualisierung
- Unterstützung großer, dynamischer IT-Landschaften
- Möglichkeit zum schrittweisen Governance-Aufbau

Kurzfristige Empfehlung

- Prüfung API-starker, graphbasierter Tools (z. B. Ardoq)
- Validierung der Datenintegration und Modellierungskompatibilität
- Aufbau eines initialen Architektur-Repositorys

Mittelfristige Empfehlung

- Einführung klarer Modellierungsstandards und Verantwortlichkeiten
- Governance-Prozesse (Freigaben, Versionierung) etablieren
- Evaluierung ergänzender Funktionen (Reporting, Dashboards)

Langfristige Perspektive

- Vollständiger Rollout als zentrales EAM-Repository
- Kontinuierliche Datenpflege und Quality Gates
- Integration in Portfolio-, Risiko- und Prozessmanagement

Diskussion

- Welche Kriterien sind aus Ihrer Sicht am wichtigsten bei der Auswahl eines EAM-Tools? (Technische Integrationen, Usability, Governance, Skalierbarkeit?)
- Wie viel Standardisierung braucht ein EAM-Tool – und wie viel Flexibilität ist notwendig?
- Welche Rolle sollte EAM in einer historisch gewachsenen IT-Landschaft spielen? (Dokumentation, Steuerung, Entscheidungsunterstützung?)
- Ab wann ist ein graphbasiertes Tool (z. B. Ardoq) einem klassischen EAM überlegen – und umgekehrt?
- Wie lässt sich die Akzeptanz eines EAM-Tools in Fachbereichen sicherstellen?
- Was wäre ein sinnvoller erster Anwendungsfall für ein EAM-Tool bei einem Finanzdienstleister/ Unternehmen Ihrer Wahl?
- Welche Risiken bestehen, wenn kein zentrales EAM etabliert wird?

Literatur

Gronau, N. (2023). Enterprise Resource Planning: Grundlagen, Einführung, Anwendung. 3. Auflage. Berlin: Springer Vieweg.

Scheer, A.-W. (1998). ARIS – Modellierungsmethoden, Metamodelle, Anwendungen. Berlin: Springer.

Scheer, A.-W. (2002). ARIS in der Praxis: Gestaltung, Implementierung und Optimierung von Geschäftsprozessen. Berlin: Springer.

Weske, M. (2019). Business Process Management: Concepts, Languages, Architectures. 3. Auflage. Berlin: Springer.